

Dave Mangot

Video 4.1

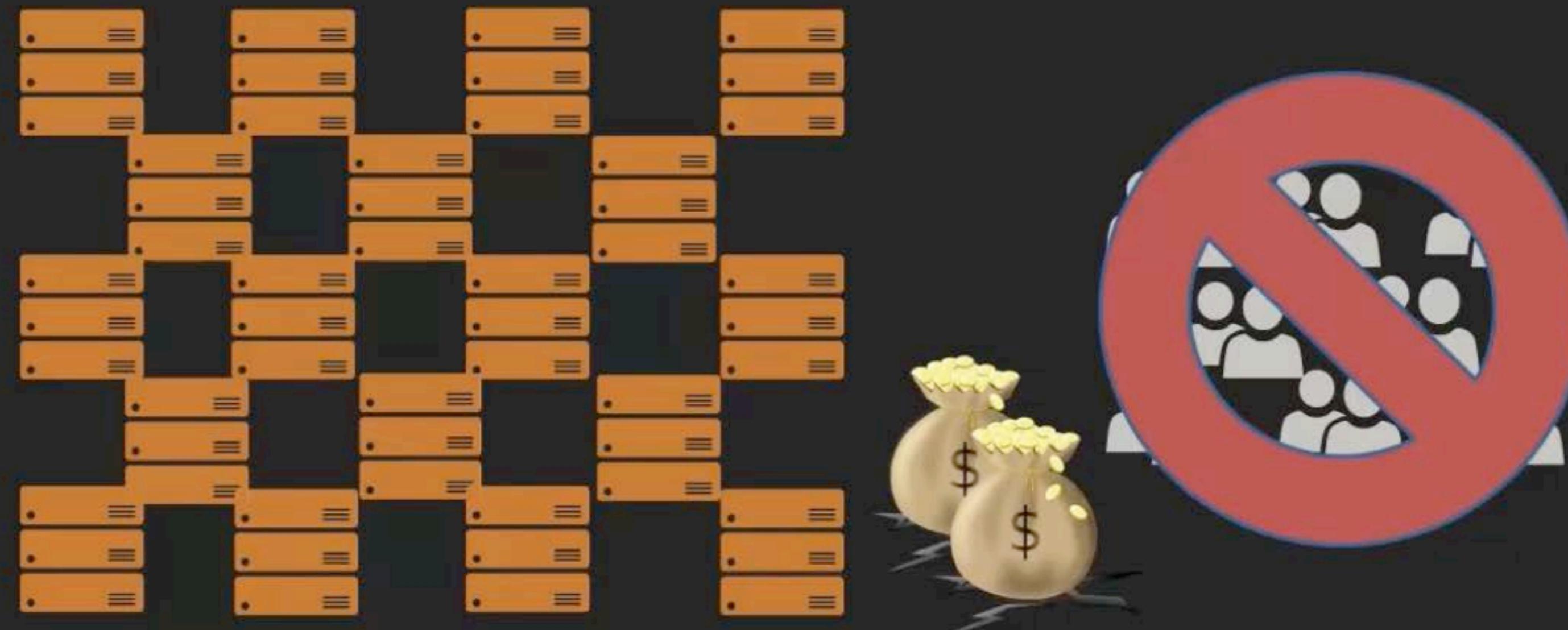
Why Automate?



In this Video, we are going to take a look at...

- Why automate
- Kickstart/Jumpstart
- Systems in production
- How CM solves the problem

Why Automate?



Benefits of Automation

- Consistency
- Compliance
- Repeatability
- Testability
- Saves time building
- Saves time maintaining

Configuration Management



Next Video

Configuration Management with SaltStack

Dave Mangot

Video 4.2

Configuration Management with SaltStack



In this Video, we are going to take a look at...

- How to use configuration management
- Our first Salt state
- Your first Salt state
- Where to go from here

What You Will Need?

- Unix virtual machine (preferably Ubuntu 14.04 LTS)
- Network connectivity

Why a Virtual Machine?



- Can recreate at will
- Allows you to experiment
- Practice with ephemeral infrastructure
- Failure happens

How to Use Configuration Management?

- Enforce states/compliance with states/manifests/recipes
- No snowflakes
- Infrastructure as Code
 - Revision Control
 - Unit Tests
 - Integration Tests
 - Code Review

Our First Salt State



SALTSTACK

Our First Salt State

- Documentation
 - <https://docs.saltstack.com/en/latest/topics/tutorials/quickstart.html>
 - <https://docs.saltstack.com/en/latest/ref/states/all/salt.states.pkg.html>
 - <https://docs.saltstack.com/en/latest/ref/states/all/salt.states.sysctl.html>
 - <https://www.digitalocean.com/community/tutorials/how-to-install-and-use-graphite-on-an-ubuntu-14-04-server>
 - <https://github.com/dmangot/Mastering-DevOps/>

Next Video

Configuration Management for Distributed Systems

Dave Mangot

Video 4.3

Configuration Management for Distributed Systems



EC2 Management Con Salt Masterless Quic How To Install and Us salt.states.pkg salt.states.sysctl Mastering-DevOps/... Dave

https://us-west-2.console.aws.amazon.com/ec2/v2/home?region=us-west-2#Instances:sort=instanceId

AWS Services Edit Dave Mangot Oregon Support

EC2 Dashboard

Events

Tags

Reports

Limits

INSTANCES

Instances

Spot Requests

Reserved Instances

Scheduled Instances

Dedicated Hosts

IMAGES

AMIs

Bundle Tasks

ELASTIC BLOCK STORE

Volumes

Snapshots

NETWORK & SECURITY

Security Groups

Elastic IP

Launch Instance

Connect

Actions

Filter by tags and attributes or search by keyword

?

<

1 to 1 of 1

>

>>

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS
	i-950d588d	t2.micro	us-west-2b	running	2/2 checks ...	None	ec2-54-187-28-

Instance: i-950d588d Public DNS: ec2-54-187-28-63.us-west-2.compute.amazonaws.com



Description

Status Checks

Monitoring

Tags

Instance ID i-950d588d

Public DNS ec2-54-187-28-63.us-west-2.compute.amazonaws.com

Instance state running

Public IP 54.187.28.63

Feedback

English

© 2008 - 2016, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Privacy Policy

PACKT
VIDEO

Terms of Use



SALTSTACK

Overview Tutorials Documentation Downloads Develop

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

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

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

BOOTSTRAP SALT MINION

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

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

BASH

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

TELLING SALT TO RUN MASTERLESS

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

2015.8.12 2016.3.3 ✓ Develop



Preseed Minion with Accepted Key

The MacOS X (Maverick) Developer Step By Step Guide To Salt Installation

running salt as normal user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

Configuring Salt

ubuntu@ip-172-31-47-14: ~ (s...)

ubuntu@ip-172-31-47-14:~\$ ls -ack.com/vagrant-tutorials/master/minion

ubuntu@ip-172-31-47-14:~\$ curl -L https://bootstrap.saltstack.com -o bootstrap_salt.sh

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
	Dload	Upload	Total	Spent	Left	Speed	
100	264	100	264	0	0	591	591
100	241k	100	241k	0	0	190k	421k

ubuntu@ip-172-31-47-14:~\$

Standalone minion can be used to do a number of things:

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

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

BOOTSTRAP SALT MINION

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

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

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

TELLING SALT TO RUN MASTERLESS

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

2. ubuntu@ip-172-31-47-14: ~ (ssh)

Pressed Minion with Accepted Key

The MacOS X (Maverick) Developer Step By Step Guide To Salt Installation

running-salt-as-normal-user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt To Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

Configuring Salt

Section 4, Lecture 17

2. ubuntu@ip-172-31-47-14: ~ (ssh)

```
ubuntu@ip-172-31-47-14: ~ (s... | ubuntu@ip-172-31-47-14: ~ (s...
ubuntu@ip-172-31-47-14:~$ ls
ubuntu@ip-172-31-47-14:~$ curl -L https://bootstrap.saltstack.com -o bootstrap_salt.sh
% Total % Received % Xferd Average Speed Time Time Current
          Dload Upload Total Spent Left Speed
100 264 100 264    0     0  591      0 --:--:-- --:--:-- 591
100 241k 100 241k   0     0 190k      0 0:00:01 0:00:01 --:--:-- 421k
ubuntu@ip-172-31-47-14:~$ ls
bootstrap_salt.sh
ubuntu@ip-172-31-47-14:~$ sudo sh bootstrap_salt.sh
```

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

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

BOOTSTRAP SALT MINION

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

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

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

TELLING SALT TO RUN MASTERLESS

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

Passed Minion with Accepted Key

The MacOSX (Mavericks) Developer Step By Step Guide To Salt Installation

running salt as normal user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

EC2 Management Con Salt Masterless Quick How To Install and Us salt.states.pkg salt.states.sysctl Mastering-DevOps/... Dave

https://docs.saltstack.com/en/latest/topics/tutorials/quickstart.html

SALTSTACK Overview Tutorials Documentation Downloads Develop

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

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

Note

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

1. Create the `top.sls` file:

/srv/salt/top.sls:

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

2015.8.12 2016.3.3 ✓ Develop



Preseed Minion with Accepted Key

The MacOS X (Maverick) Developer Step By Step Guide To Salt Installation

running salt as normal user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

Configuring Salt



2. Create the webserver state tree:

/srv/salt/webserver.sls:

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

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...

```
Setting up mysql-common (5.5.50-0ubuntu0.14.04.1) ...
Setting up libmysqlclient18:amd64 (5.5.50-0ubuntu0.14.04.1) ...
Setting up libpgm-5.1-0:amd64 (5.1.118-1~dfsg-0.1ubuntu3) ...
Setting up libsodium13:amd64 (1.0.3-1) ...
Setting up libzmq3:amd64 (4.0.5+dfsg-3) ...
Setting up dctrl-tools (2.23ubuntu1) ...
Setting up debconf-utils (1.5.51ubuntu2) ...
Setting up libjs-jquery (1.7.2+dfsg-2ubuntu1) ...
Setting up libjs-underscore (1.4.4-2ubuntu1) ...
Setting up libjs-sphinxdoc (1.2.2+dfsg-1ubuntu1.1) ...
Setting up python-concurrent.futures (3.0.3-1) ...
Setting up python-crypto (2.6.1-4build1) ...
Setting up python-dateutil (1.5+dfsg-1ubuntu1) ...
Setting up python-markupsafe (0.18-1build2) ...
Setting up python-jinja2 (2.7.2-2) ...
Setting up python-mako (0.9.1-1) ...
Setting up python-msgpack (0.4.6-1build1) ...
Setting up python-mysqldb (1.2.3-2ubuntu1) ...
Setting up python-tornado (4.2.1-1~ds+1) ...
Setting up python-zmq (14.0.1-1build2) ...
Setting up salt-common (2016.3.3+ds-1) ...
Setting up salt-minion (2016.3.3+ds-1) ...
salt-minion start/running, process 2913
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
  * INFO: Running install_ubuntu_stable_post()
  * INFO: Running install_ubuntu_check_services()
  * INFO: Running install_ubuntu_restart_daemons()
salt-minion stop/waiting
salt-minion start/running, process 2976
  * INFO: Running daemons_running()
  * INFO: Salt installed!
ubuntu@ip-172-31-47-14:~$ sudo mkdir -p /srv/salt
ubuntu@ip-172-31-47-14:~$ cd !$
cd /srv/salt
ubuntu@ip-172-31-47-14:/srv/salt$
```

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Events](#)

2015-07-20 20:51:47 | Developer

[Pending Minion with Accepted Key](#)[The MacOS X \(Maverick\) Developer Step By Step Guide To Salt Installation](#)[running-salt-as-normal-user tutorial](#)[Standalone Minion](#)

Salt Masterless Quickstart

[Bootstrap Salt Minion](#)[Telling Salt to Run Masterless](#)[Create State Tree](#)[Salt-call](#)[Dependencies](#)[Optional Dependencies](#)[Upgrading Salt](#)[Building Packages using Salt Pack](#)[Configuring Salt](#)

EC2 Management Con Salt Masterless Quick How To Install and Us salt.states.pkg salt.states.sysctl Mastering-DevOps/tu Dave

https://docs.saltstack.com/en/latest/topics/tutorials/quickstart.html

SALTSTACK

Overview Tutorials Documentation Downloads Develop

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

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

Note

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

1. Create the `top.sls` file:

```
/srv/salt/top.sls:
```

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

YAML

2. Create the webserver state tree:

```
/srv/salt/webserver.sls:
```

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

YAML

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

Configuring Salt



ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...)

```
Setting up mysql-common (5.5.50-0ubuntu0.14.04.1) ...
Setting up libmysqlclient18:amd64 (5.5.50-0ubuntu0.14.04.1) ...
Setting up libpgm-5.1-0:amd64 (5.1.118-1~dfsg-0.1ubuntu3) ...
Setting up libsodium13:amd64 (1.0.3-1) ...
Setting up libzmq3:amd64 (4.0.5+dfsg-3) ...
Setting up dctrl-tools (2.23ubuntu1) ...
Setting up debconf-utils (1.5.51ubuntu2) ...
Setting up libjs-jquery (1.7.2+dfsg-2ubuntu1) ...
Setting up libjs-underscore (1.4.4-2ubuntu1) ...
Setting up libjs-sphinxdoc (1.2.2+dfsg-1ubuntu1.1) ...
Setting up python-concurrent.futures (3.0.3-1) ...
Setting up python-crypto (2.6.1-4build1) ...
Setting up python-dateutil (1.5+dfsg-1ubuntu1) ...
Setting up python-markupsafe (0.18-1build2) ...
Setting up python-jinja2 (2.7.2-2) ...
Setting up python-mako (0.9.1-1) ...
Setting up python-msgpack (0.4.6-1build1) ...
Setting up python-mysqldb (1.2.3-2ubuntu1) ...
Setting up python-tornado (4.2.1-1~ds+1) ...
Setting up python-zmq (14.0.1-1build2) ...
Setting up salt-common (2016.3.3+ds-1) ...
Setting up salt-minion (2016.3.3+ds-1) ...
salt-minion start/running, process 2913
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
  * INFO: Running install_ubuntu_stable_post()
  * INFO: Running install_ubuntu_check_services()
  * INFO: Running install_ubuntu_restart_daemons()
salt-minion stop/waiting
salt-minion start/running, process 2976
  * INFO: Running daemons_running()
  * INFO: Salt installed!
ubuntu@ip-172-31-47-14:~$ sudo mkdir -p /srv/salt
ubuntu@ip-172-31-47-14:~$ cd !$
cd /srv/salt
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi top.
```

Overview Tutorials Documentation Downloads Events

2015.8.12 · 2016.8.1 · Download

Proseed Minion with Accepted Key

The MacOS X (Maverick) Developer Step By Step Guide To Salt Installation

running-salt-as-normal-user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

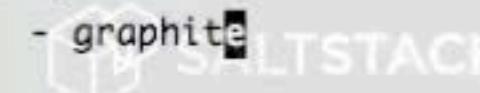
Building Packages using Salt Pack

Configuring Salt

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...)

base: https://docs.saltstack.com/en/latest/topics/minion-commands.html



Overview Tutorials Documentation Downloads Devs

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

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

Note

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

1. Create the top.sls file

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

```
base:
  -> webserver
```

2. Create the webserver state tree

```
/srv/salt/webserver.sls
```

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

[Privated Minion with Accepted Key](#)[The MacOS X\(Maverick\) Developer Step By Step Guide To Salt Installation](#)[running-salt-as-normal-user tutorial](#)[Standalone Minions](#)

Salt Masterless Quickstart

[Bootstrap Salt Minion](#)[Telling Salt to Run Masters](#)[Create State Tree](#)[Salt-call](#)[Dependencies](#)[Optional Dependencies](#)[Upgrading Salt](#)[Building Packages using Salt Pack](#)[Configuring Salt](#)

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

```
ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...  
Setting up libmysqlclient18:amd64 (5.5.50-0ubuntu0.14.04.1) ...  
Setting up libpgm-5.1-0:amd64 (5.1.118-1~dfsg-0.1ubuntu3) ...  
Setting up libsodium13:amd64 (1.0.3-1) ...  
Setting up libzmq3:amd64 (4.0.5+dfsg-3) ...  
Setting up dctrl-tools (2.23ubuntu1) ...  
Setting up debconf-utils (1.5.51ubuntu2) ...  
Setting up libjs-jquery (1.7.2+dfsg-2ubuntu1) ...  
Setting up libjs-underscore (1.4.4-2ubuntu1) ...  
Setting up libjs-sphinxdoc (1.2.2+dfsg-1ubuntu1.1) ...  
Setting up python-concurrent.futures (3.0.3-1) ...  
Setting up python-crypto (2.6.1-4build1) ...  
Setting up python-dateutil (1.5+dfsg-1ubuntu1) ...  
Setting up python-markupsafe (0.18-1build2) ...  
Setting up python-jinja2 (2.7.2-2) ...  
Setting up python-mako (0.9.1-1) ...  
Setting up python-msgpack (0.4.6-1build1) ...  
Setting up python-mysqldb (1.2.3-2ubuntu1) ...  
Setting up python-tornado (4.2.1-1~ds+1) ...  
Setting up python-zmq (14.0.1-1build2) ...  
Setting up salt-common (2016.3.3+ds-1) ...  
Setting up salt-minion (2016.3.3+ds-1) ...  
salt-minion start/running, process 2913  
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...  
Processing triggers for ureadahead (0.100.0-16) ...  
* INFO: Running install_ubuntu_stable_post()  
* INFO: Running install_ubuntu_check_services()  
* INFO: Running install_ubuntu_restart_daemons()  
salt-minion stop/waiting  
salt-minion start/running, process 2976  
* INFO: Running daemons_running()  
* INFO: Salt installed!  
ubuntu@ip-172-31-47-14:~$ sudo mkdir -p /srv/salt  
ubuntu@ip-172-31-47-14:~$ cd !$  
cd /srv/salt  
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi top.sls  
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi graphite.sls
```

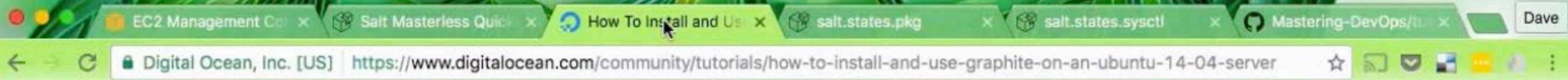
[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Devices](#)[2015.8.17](#) [2016.3.10](#) [Develop](#)[Pending Minion with Accepted Key](#)[The MacOS X \(Maverick\) Developer Step By Step Guide To Salt Installation](#)[running-salt-as-normal-user tutorial](#)[Standalone Minions](#)

Salt Masterless Quickstart

[Bootstrap Salt Minion](#)[Telling Salt to Run Masterless](#)[Create State Tree](#)[Salt-call](#)

Dependencies

[Optional Dependencies](#)[Upgrading Salt](#)[Building Packages using Salt Pack](#)[Configuring Salt](#)



To get started, we need to download and install the Graphite components. If you looked at our introduction to graphing software, you will have noticed that Graphite is made of several components: the web application, a storage backend called Carbon, and the database library called whisper.

Graphite used to be fairly difficult to install. Luckily, in Ubuntu 14.04, all of the components that we need can be found in the default repositories.

Let's update our local package index and then install the necessary packages:

```
sudo apt-get update  
sudo apt-get install graphite-web graphite-carbon
```

During the installation, you will be asked whether you want Carbon to remove the database files if you ever decide to purge the installation. Choose "No" here so that you will not destroy your stats. If you need to start fresh, you can always manually remove the files (kept in `var/lib/graphite/whisper`).

When the installation is complete, Graphite will be installed. We need to do some additional configuration though to get everything off the ground and running.

Configure a Database for Django

Although the Graphite data itself is handled by Carbon and the whisper database library, the web application is a Django Python application, and needs to store its data somewhere.

SCROLL TO TOP

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...)

graphite: https://docs.saltstack.com/en/latest/ref/states/all/salt.states.pkg.html

pkg.installed:

- pkgs:
 - graphite-web
 - graphite-carbon

Multiple packages can also be installed with the use of the pkgs state module

dotdeb.repo:

pkgrepo.managed:

- humanname: Dotdeb
- name: deb http://packages.dotdeb.org wheezy-php55_all
- dist: wheezy-php55
- files: /etc/apt/sources.list.d/dotdeb.list
- KeyID: 800F9A77
- keyserver: keys.gpgpg.net
- refresh_db: true

php_packages:

pkg.installed:

- fromrepo: wheezy-php55
- plus:
 - php5-fpm
 - php5-odbc
 - php5-curl

Warning

Package names are currently case-sensitive. If the minion is using a package manager which is not case-sensitive (such as pipenv), then this state will fail if the package name is not used. This will be addressed in a future release of Salt.

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Events](#)

2015.8.10 · 2016.3.0 · 2016.11.0 · Develop

[salt.states.pkg_group_installed](#)[salt.states.pagerduty_service](#)[salt.states.pagerduty_user](#)[salt.states.pkg](#)[group_installed](#)[installed](#)[latest](#)[mod_aggregate](#)[mod_watch](#)[merged](#)[removed](#)[uptodate](#)

Installation of packages using OS package managers such as yum or apt-get

salt.states.pkg_group_installed (name=skin-Nano, include=None, **kwargs)

EC2 Management Con Salt Masterless Quick How To Install and Use salt.states.pkg salt.states.sysctl Mastering-DevOps/... Dave

https://docs.saltstack.com/en/latest/topics/tutorials/quickstart.html

SALTSTACK Overview Tutorials Documentation Downloads Develop

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

2015.8.12 2016.3.3 Develop

SALT-CALL

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

`salt-call --local state.apply`

BASH

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

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

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

BASH

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

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

The minion should now have Apache installed, and the next step is to begin learning how to write [more complex states](#).



Preseed Minion with Accepted Key

The MacOS X (Maverick) Developer Step By Step Guide To Salt Installation

running salt as normal user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

Configuring Salt

ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...)

```
Setting up libsodium13:amd64 (1.0.3-1) ...
Setting up libzmq3:amd64 (4.0.5+dfsg-3) ...
Setting up dctrl-tools (2.23ubuntu1) ...
Setting up debconf-utils (1.5.5ubuntu2) ...
Setting up libjs-jquery (1.7.2+dfsg-2ubuntu1) ...
Setting up libjs-underscore (1.4.4-2ubuntu1) ...
Setting up libjs-sphinxdoc (1.2.2+dfsg-1ubuntu1.1) ...
Setting up python-concurrent.futures (3.0.3-1) ...
Setting up python-crypto (2.6.1-4build1) ...
Setting up python-dateutil (1.5+dfsg-1ubuntu1) ...
Setting up python-markupsafe (0.18-1build2) ...
Setting up python-jinja2 (2.7.2-2) ...
Setting up python-mako (0.9.1-1) ...
Setting up python-msgpack (0.4.6-1build1) ...
Setting up python-mysqldb (1.2.3-2ubuntu1) ...
Setting up python-tornado (4.2.1-1~ds+1) ...
Setting up python-zmq (14.0.1-1build2) ...
Setting up salt-common (2016.3.3+ds-1) ...
Setting up salt-minion (2016.3.3+ds-1) ...
salt-minion start/running, process 2913
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
* INFO: Running install_ubuntu_stable_post()
* INFO: Running install_ubuntu_check_services()
* INFO: Running install_ubuntu_restart_daemons()
salt-minion stop/waiting
salt-minion start/running, process 2976
* INFO: Running daemons_running()
* INFO: Salt installed!
ubuntu@ip-172-31-47-14:~$ sudo mkdir -p /srv/salt
ubuntu@ip-172-31-47-14:~$ cd !$
cd /srv/salt
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi top.sls
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi graphite.sls
ubuntu@ip-172-31-47-14:/srv/salt$ sudo salt-call --local state.apply
```

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

[Overview](#) [Topics](#) [Documentation](#) [Downloads](#) [Search](#)[Tutorials](#) [Advanced](#) [Glossary](#)

Pressed Minion with Accepted Key

The MacOSX (Maverick) Developer Step By Step Guide To Salt Installation

running salt as normal user tutorial

Standalone Minion

Salt Masterless Quickstart

[Bootstrap Salt Minion](#)[Telling Salt To Run Masterless](#)[Create State Tree](#)[Salt-call](#)[Dependencies](#)[Optional Dependencies](#)[Upgrading Salt](#)[Building Packages using Salt Pack](#)[Configuring Salt](#)

```
ubuntu@ip-172-31-47-14: /spu
```

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

```
new: 2012c-1ubuntu0.1
old:
python-whisper:
The only type of file provision our minion using salt-call.
new:
SALT-CALL 0.9.12-1
old:
The python2.7-cairo: used to run remote execution functions locally on a minion instead of executing it over SSH to the master. Normally the salt-call command checks into the master to retrieve file server and pillar data, but when running standalone salt-call needs to be instructed to not check the master for data.
new: 1
old: 0
python2.7-pyparsing:
-----
salt-call new: state.state.apply
    1
old:
The --local flag tells the salt minion to look for the state tree in the local file system and not to contact a SaltMaster for instructions.
new:
To provide verbose output, use -l debug
old:
-----  
salt --local state.apply -l debug  
ary for local
-----
needed: 1 (changed=1) examines the __env.state file and determines that it is a part of the group matched by *  
ed: 0 and that the webserver SLB should be applied.
-----
states run: 1
run time: 11.899 s
tu@ip-172-31-47-14:/srv/salt$ dpkg -l | grep graphite
graphite-carbon          0.9.12-3           all      backend da
graphite-web              0.9.12+debian-3   all      Enterprise
tu@ip-172-31-47-14:/srv/salt$
```

backend data caching and persistence daemon for Graphite
Enterprise Scalable Realtime Graphing



ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...)

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

base:

*:
- graphite
- tuning

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

SALT-CALL

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

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

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

To provide verbose output, use -l debug

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

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

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

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

Overview Tutorials Documentation Downloads Releases

2015.8.12 · 2016.3.7 · Dev

Provision Minion with Accepted Key

The Mac OS X (Maverick) Developer Step By Step Guide To Salt Installation

running-salt-as-normal-user tutorial

Standalone Minions

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masters

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

Building Packages using Salt Pack

Configuring Salt

```
ubuntu@ip-172-31-47-14: /srv/salt$ ls -l | grep graphite
total 0
ubuntu@ip-172-31-47-14: /srv/salt$
```

2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)



SALTSTACK
python-whisper:

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

new:

0.9.12-1

SALT-CA old:

python2.7-cairo:

The salt-call command is used to run remote execution functions locally on a minion instead of executing them from the master. Normally the salt-call command connects into the master to retrieve files

new:

1

server and return data, but when running standalone salt-call needs to be instructed to not check the

old:

python2.7-pyparsing:

new:

salt-call --local state.apply

old:

python2.7-simplejson:

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

new:

1

To provide verbose output, use -l debug

old:

Summary for local

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

Succeeded: 1 (changed=1)

Failed: 0 minion first examines the .cons.sls file and determines that it is a part of the group matched by %

and that the whenever SLS should be applied.

Total states run: 1

Total run time: 11.899 s

ubuntu@ip-172-31-47-14:/srv/salt\$ dpkg -l | grep graphite

ii graphite-carbon 0.9.12-3

ii graphite-web 0.9.12+debian-3

ubuntu@ip-172-31-47-14:/srv/salt\$ sudo vi top.sls

ubuntu@ip-172-31-47-14:/srv/salt\$ sudo vi tuning.sls

all
all

backend data caching and persistence daemon for Graphite

Enterprise Scalable Realtime Graphing

Configuring Salt

should now have Apache installed and the next step is to begin learning how to

backend data caching and persistence daemon for Graphite

Enterprise Scalable Realtime Graphing

Configuring Salt

2015-09-01 2016-09-01 Develop

Provision Minion with Accepted Key

The MacOS X(Maverick) Developer Step By Step Guide To Salt Installation

running salt as normal user tutorial

Standalone Minion

Salt Masterless Quickstart

Bootstrap Salt Minion

Telling Salt to Run Masterless

Create State Tree

Salt-call

Dependencies

Optional Dependencies

Upgrading Salt

2. ubuntu@ip-172-31-47-14: ~ (ssh)

```
ubuntu@ip-172-31-47-14: /srv...  ubuntu@ip-172-31-47-14: ~ (s...
ubuntu@ip-172-31-47-14:~$ sudo sysctl -a | grep file-max
fs.file-max = 99947
ubuntu@ip-172-31-47-14:~$
```

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Events](#)

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

SALT-CALL

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

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

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

To provide verbose output, use -l debug

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

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

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

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

[Provision Minion with Accepted Key](#)

[The MacOS X \(Maverick\) Developer Step By Step Guide To Salt Installation](#)

[running-salt-as-normal-user tutorial](#)

[Standalone Minions](#)

Salt Masterless Quickstart

[Bootstrap Salt Minion](#)

[Telling Salt to Run Masters](#)

[Create State Tree](#)

[Salt-call](#)

[Dependencies](#)

[Optional Dependencies](#)

[Upgrading Salt](#)

[Building Packages using Salt Pack](#)

[Configuring Salt](#)

EC2 Management Co. Salt Masterless Quick How To Install and Us salt.states.pkg salt.states.sysctl Mastering-DevOps/... Dave

https://docs.saltstack.com/en/latest/ref/states/all/salt.states.sysctl.html

SALTSTACK Overview Tutorials Documentation Downloads Develop

SALT STATES.SYSCONTL

CONFIGURATION OF THE LINUX KERNEL USING SYSCONTL

Control the kernel sysctl system.

```
vm.swappiness:  
    sysctl.present:  
        - value: 20
```

YAML

salt.states.sysctl. **present** (*name, value, config=None*)

Ensure that the named sysctl value is set in memory and persisted to the named configuration file.

The default sysctl configuration file is /etc/sysctl.conf

name

The name of the sysctl value to edit

value

The sysctl value to apply

config

The location of the sysctl configuration file. If not specified, the proper location will be detected based on platform.

2015.8.12

2016.3.3 ✓

Develop



salt.states.saltstack

salt.states.status

salt.states.stormpath_account

salt.states.supervisord

salt.states.svn

salt.states.sysctl

present

Configuration of the Linux kernel using sysctl

salt.states.syslog_ng

salt.states.sysrc

salt.states.telemetry_alert

salt.states.test

salt.states.timezone

salt.states.tls

salt.states.tomcat

```
ubuntu@ip-172-31-47-14: /srv/salt$ ssh 2. ubuntu@ip-172-31-47-14: ~ (s...  
 2. GitHub 2012c-lubuntu0.1 ub.com/dmangot/Mastering-DevOps/blob/master/Section4/Video2/tuning.sls  
  old:  
  python-whisper:  
    ---- Personal Open source Business Explore Pricing Blog Support This repository Sign in GitHub  
  new:  
  0.9.12-1  
  old:  
  python2.7-cairo: Mastering-DevOps  
  -----  
  new:  
  < Code Issues 0 Pull requests 0 Pulses Graphs  
  old:  
  python2.7-pyparsing:  
  ----- Mastering-DevOps / Section4 / Video2 / tuning.sls Find file Copy path  
  new:  
  1 dmangot more than 3 years ago  
  old:  
  python2.7-simplejson:  
  -----  
  new:  
  1 dmangot more than 3 years ago  
  old: 1 (3.8MB) 58 bytes Raw blame History  
  
Summary for local  
-----  
Succeeded: 1 (changed=1)  
Failed: 0  
  
Total states run: 1  
Total run time: 11.899 s  
ubuntu@ip-172-31-47-14:/srv/salt$ dpkg -l | grep graphite  
ii graphite-carbon 0.9.12-3 all backend data caching and persistence daemon for Graphite  
ii graphite-web 0.9.12+debian-3 all Enterprise Scalable Realtime Graphing  
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi top.sls  
ubuntu@ip-172-31-47-14:/srv/salt$ sudo vi tuning.sls  
ubuntu@ip-172-31-47-14:/srv/salt$ dpkg -l | grep graphite
```



2. ubuntu@ip-172-31-47-14: /srv/salt (ssh)

ubuntu@ip-172-31-47-14: /srv... ubuntu@ip-172-31-47-14: ~ (s...)

ubuntu@ip-172-31-47-14:/srv/salt\$ sudo salt-call --local state.apply blob/master/Section4/video2/tuning.sls

[WARNING] /usr/lib/python2.7/dist-packages/salt/grains/core.py:1493: DeprecationWarning: The "osmajorrelease" will be a type of an integer.

local:



Personal Open source Business Explore

Pricing Blog Support This repository

Sign in



```
ID: graphite
Function: pkg.installed
  Result: True
  Comment: All specified packages are already installed
  Started: 05:55:49.650091
  Duration: 240.693 ms
  Changes:
```

[Watch](#) 1 [Star](#) 0 [Fork](#) 0

```
ID: fs.file-max
Function: sysctl.present
  Result: True
  Comment: Updated sysctl value fs.file-max = 65536
  Started: 05:55:49.891260
  Duration: 19.668 ms
  Changes:
```

[Find file](#) [Copy path](#)

```
fs.file-max:
  value: 65536
```

[Raw](#) [Blame](#) [History](#) [Issues](#) [Pull requests](#) [Graphs](#)

```
Summary for local
```

```
Succeeded: 2 (changed=1)
```

```
Failed: 0
```

```
Total states run: 2
```

```
Total run time: 260.361 ms
```

```
ubuntu@ip-172-31-47-14:/srv/salt$ sudo cat tuning.sls
```

```
fs.file-max:
```

```
  sysctl.present:
```

```
    - value: 65536
```

```
ubuntu@ip-172-31-47-14:/srv/salt$
```

Mastering DevOps

Dave Mangot

Video 4.3

Configuration Management for Distributed Systems



0:02 / 7:19

In this Video, we are going to take a look at...

- Reviewing our Salt state
- Installing a Salt master and minion
- Verifying connectivity
- Improving our Salt state
- Serving our state from a salt master

Our Current State

- Installs two packages
- Tunes the OS
- Runs locally only

Salt Example

- https://docs.saltstack.com/en/latest/topics/tutorials/salt_bootstrap.html#ubuntu-and-derivatives
- <https://docs.saltstack.com/en/latest/ref/states/all/salt.states.service.html>
- <https://github.com/dmangot/Mastering-DevOps>

Compute Engine - Dave Mangot Salt Bootstrap salt.states.service GitHub - dmangot/Mastering- Dave

https://console.cloud.google.com/compute/instances?project=dave-mangot&graph=GCE_CPU&duration=PT1H

Dave Mangot

Compute Engine VM instances CREATE INSTANCE CREATE INSTANCE GROUP REFRESH START STOP

VM instances Instance groups Instance templates Disks Snapshots Images Metadata Health checks Zones Operations Quotas Settings

Filter by label or name Columns Labels Recommendations

CPU utilization 1 hour 6 hours 12 hours 1 day 2 days 4 days 7 days 14 days 30 days

CPU

% CPU

50
40
30
20
10

Sep 3, 4:30 PM Sep 3, 4:45 PM Sep 3, 5:00 PM Sep 3, 5:15 PM Sep 3, 5:28 PM

CPU: 0.483

Name	Zone	Machine type	Recommendation	In use by	Internal IP	External IP	Connect
mastering-devops	us-central1-c	1 vCPU, 3.75 GB			10.128.0.2	104.197.162.140	SSH

[PACKT] VIDEO

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
dave@mastering-devops:~$ git clone https://github.com/dmangot/Mastering-DevOps.git
Cloning into 'Mastering-DevOps'...
```

```
remote: Counting objects: 108, done.
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.
```

```
Resolving deltas: 100% (20/20), done.
```

```
Checking connectivity... done.
```

```
dave@mastering-devops:~$
```

```
dave@mastering-devops:~$ cd Mastering-DevOps/Section4/Video2
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ls
```

```
graphite.sls top.sls tuning.sls
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

```
[REDACTED]
```

CPU

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
dave@mastering-devops:~$ git clone https://github.com/dmangot/Mastering-DevOps.git
Cloning into 'Mastering-DevOps'...
```

```
remote: Counting objects: 108, done.
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.
Resolving deltas: 100% (20/20), done.
```

```
Checking connectivity... done.
```

```
dave@mastering-devops:~$
```

```
dave@mastering-devops:~$ cd Mastering-DevOps/Section4/Video2
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ls
```

```
graphite.sls top.sls tuning.sls
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo mkdir /srv/salt
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo cp * !$
```

```
sudo cp * /srv/salt
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

Name	Zone	Machine type	Recommendation	In use by	Internal IP	External IP	Actions
mastering-devops	us-west-1a	t2.micro 0.06			10.108.1.10	100.102.162.100	SSH

Compute Engine - Dave Mangot X Salt Bootstrap X salt.states.service X GitHub - dmangot/Mastering- X

Dave

https://docs.saltstack.com/en/latest/topics/tutorials/salt_bootstrap.html#ubuntu-and-derivatives

SALTSTACK

Overview Tutorials Documentation Downloads Develop

Using curl to install latest development version from GitHub:

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

If you want to install a specific release version (based on the Git tags):

```
PYTHON
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh git v2015.8.8
```

To install a specific branch from a Git fork:

```
PYTHON
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -g https://github.com/myuser/salt.git git mybranch
```

If all you want is to install a salt-master using latest Git:

```
PYTHON
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -M -N git develop
```

If your host has Internet access only via HTTP proxy:

```
PYTHON
PROXY='http://user:password@myproxy.example.com:3128'
curl -o bootstrap_salt.sh -L -x "$PROXY" https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -M -N git
```

Establishing secure connection...

2015.8.12 2016.3.3 ✓ Develop

Configuration Management

Events & Reactor

Orchestration

Salt SSH

Salt Cloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

[PACKET] VIDEO

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
dave@mastering-devops:~$ git clone https://github.com/dmangot/Mastering-DevOps.git
Cloning into 'Mastering-DevOps'...
```

```
remote: Counting objects: 108, done.
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.
Resolving deltas: 100% (20/20), done.
```

Checking connectivity... done.

dave@mastering-devops:~\$

dave@mastering-devops:~\$ cd Mastering-DevOps/Section4/Video2

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ls

graphite.sls top.sls tuning.sls

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo mkdir /srv/salt

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo cp * !\$

sudo cp * /srv/salt

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

To install a specific branch from a Git fork:

```
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -g https://github.com/myuser/salt.git mybranch
```

If all you want is to install a salt-minion using latest Git:

```
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -H -N git develop
```

If your host has internet access only via a HTTP proxy:

```
PROXY='http://user:password@proxy.example.com:3128'
curl -o bootstrap_salt.sh -L -x "$PROXY" https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -G -H "$PROXY" git
```

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Develop](#)
[2015-2016](#) [2014-2015](#) [Development](#)
[Configuration Management](#)
[Events & Reactor](#)
[Orchestration](#)
[Salt-SSH](#)
[Salt Cloud](#)
[Salt Proxy Minion](#)
[Salt-Virt](#)
[Command Line Reference](#)
[Salt Module Reference](#)
[APIs](#)
[Architecture](#)
[Windows](#)
[Developing Salt](#)
[Release Notes](#)

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
dave@mastering-devops:~$ git clone https://github.com/dmangot/Mastering-DevOps.git
Cloning into 'Mastering-DevOps'...
```

```
remote: Counting objects: 108, done.
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.
Resolving deltas: 100% (20/20), done.
```

```
Checking connectivity... done.
```

dave@mastering-devops:~\$

dave@mastering-devops:~\$ cd Mastering-DevOps/Section4/Video2

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ls

graphite.sls top.sls tuning.sls

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo mkdir /srv/salt

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo cp * !\$

sudo cp * /srv/salt

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current	
			Dload	Upload	Total	Spent	Left	Speed
100	264	100	264	0	0	1044	0	--::--- --::--- --::--- 1047
100	241k	100	241k	0	0	434k	0	--::--- --::--- --::--- 434k

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

```
sudo sh bootstrap_salt.sh -q https://github.com/dmangot/salt.git git mybranch
```

If all you want is to install a salt-minion using latest Git:

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

If your host has Internet access only via HTTP proxy:

```
PROXY='http://user:password@myproxy.example.com:3128'
curl -o bootstrap_salt.sh -L -x "$PROXY" https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -G -H "$PROXY" git
```

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Desktop](#)
[2015.8.2](#) [2015.8.1](#) [Desktop](#)
[Configuration Management](#)
[Events & Reactor](#)
[Orchestration](#)
[Salt SSH](#)
[Salt Cloud](#)
[Salt Proxy Minion](#)
[Salt Virt](#)
[Command Line Reference](#)
[Salt Module Reference](#)
[APIs](#)
[Architecture](#)
[Windows](#)
[Developing Salt](#)
[Release Notes](#)

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
dave@mastering-devops:~$ git clone https://github.com/dmangot/Mastering-DevOps.git
Cloning into 'Mastering-DevOps'...
```

```
remote: Counting objects: 108, done.
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.
Resolving deltas: 100% (20/20), done.
```

```
Checking connectivity... done.
```

dave@mastering-devops:~\$

dave@mastering-devops:~\$ cd Mastering-DevOps/Section4/Video2

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ls

graphite.sls top.sls tuning.sls

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo mkdir /srv/salt

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo cp * !\$

sudo cp * /srv/salt

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com

% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current	
			Dload	Upload	Total	Spent	Left	Speed

100	264	100	264	0	0	1044	0	--::--- --::--- --::--- 1047
-----	-----	-----	-----	---	---	------	---	------------------------------

100	241k	100	241k	0	0	434k	0	--::--- --::--- --::--- 434k
-----	------	-----	------	---	---	------	---	------------------------------

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo sh bootstrap_salt.sh -M stable

- * INFO: Running version: 2016.08.16 https://github.com/saltstack/salt.git mybranch
- * INFO: Executed by: sh
- * INFO: Command line: 'bootstrap_salt.sh -M stable'

```
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -M -N git develop
```

If your host has Internet access only via HTTP proxy:

```
PROXY='http://user:password@myproxy.example.com:3128'
curl -o bootstrap_salt.sh -L -x "$PROXY" https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -M -N "$PROXY" git
```

[Features](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Editions](#)
[2016.08.16](#) [2016.07.26](#) [Development](#)
[Configuration Management](#)
[Events & Reactor](#)
[Orchestration](#)
[Salt SSH](#)
[SaltCloud](#)
[Salt Proxy Minion](#)
[Salt Virt](#)
[Command Line Reference](#)
[Salt Module Reference](#)
[APIs](#)
[Architecture](#)
[Windows](#)
[Developing Salt](#)
[Release Notes](#)

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
Setting up dctrl-tools (2.23ubuntu1) ...
Setting up debconf-utils (1.5.51ubuntu2) ...
Setting up libjs-jquery (1.7.2+dfsg-2ubuntu1) ...
Setting up libjs-underscore (1.4.4-2ubuntu1) ...
Setting up libjs-sphinxdoc (1.2.2+dfsg-1ubuntu1.1) ...
Setting up python-concurrent.futures (3.0.3-1) ...
Setting up python-crypto (2.6.1-4build1) ...
Setting up python-dateutil (1.5+dfsg-1ubuntu1) ...
Setting up python-async (0.6.1-1) ...
Setting up python-smmap (0.8.2-1) ...
Setting up python-gitdb (0.5.4-1) ...
Setting up python-git (0.3.2~RC1-3) ...
Setting up python-markupsafe (0.18-1build2) ...
Setting up python-jinja2 (2.7.2-2) ...
Setting up python-mako (0.9.1-1) ...
Setting up python-msgpack (0.4.6-1build1) ...
Setting up python-mysqldb (1.2.3-2ubuntu1) ...
Setting up python-tornado (4.2.1-1~ds+1) ...
Setting up python-zmq (14.0.1-1build2) ...
Setting up salt-common (2016.3.3+ds-1) ...
Setting up salt-master (2016.3.3+ds-1) ...
salt-master start/running, process 4568
salt-minion start/running, process 4604
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
• INFO: Running install_ubuntu_stable_post()
• INFO: Running install_ubuntu_check_services()
• INFO: Running install_ubuntu_restart_daemons()
salt-minion stop/waiting
salt-minion start/running, process 5269
salt-master stop/waiting
salt-master start/running, process 5313
• INFO: Running daemons_running()
• INFO: Salt installed!
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

Overview Releases Documentation Downloads Details

2016.3.2 2016.3.1 2016.3.0 Details

Configuration Management

Events & Reactor

Orchestration

SaltSSH

SaltCloud

Salt Proxy Minion

SaltVirt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

```
dave@mastering-devops: ~/M...  
Setting up python-tornado (4.2.1-1~ds+1) ...  
Setting up python-zmq (14.0.1-1build2) ...  
Setting up salt-common (2016.3.3+ds-1) ...  
Setting up salt-master (2016.3.3+ds-1) ...  
salt-master start/running, process 4568  
Setting up salt-minion (2016.3.3+ds-1) ...  
salt-minion start/running, process 4604  
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...  
Processing triggers for ureadahead (0.100.0-16) ...  
  * INFO: Running install_ubuntu_stable_post()  
  * INFO: Running install_ubuntu_check_services()  
  * INFO: Running install_ubuntu_restart_daemons()  
salt-minion stop/waiting  
salt-minion start/running, process 5269  
salt-master stop/waiting  
salt-master start/running, process 5313  
  * INFO: Running daemons_running()  
  * INFO: Salt installed!  
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ps -ef | grep salt-master  
root      4732      1  1 00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      4744      1  1 00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5313      1  1 00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5324    5313  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5325    5313  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5326    5313  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5329    5313  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5380    5313  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5381    5313  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5382    5381  0  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5383    5381  2  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5394    5381  2  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5396    5381  2  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5400    5381  2  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
root      5401    5381  2  00:31 ?        00:00:00 /usr/bin/python /usr/bin/salt-master  
dave     5931  2005  0  00:32 pts/0  00:00:00 grep --color=auto salt-master  
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

Overview Tutorials Documentation Downloads Details

2016.3.2 2016.3.0 Devday

Configuration Management

Events & Reactor

Orchestration

Salt SSH

SaltCloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

Setting up salt-common (2016.3.3+ds-1) ...

Setting up salt-master (2016.3.3+ds-1) ...

salt-master start/running, process 4568

Setting up salt-minion (2016.3.3+ds-1) ...

salt-minion start/running, process 4604

Processing triggers for libc-bin (2.19-0ubuntu6.9) ...

Processing triggers for ureadahead (0.100.0-16) ...

* INFO: Running install_ubuntu_stable_post()

* INFO: Running install_ubuntu_check_services()

* INFO: Running install_ubuntu_restart_daemons()

salt-minion stop/waiting

salt-minion start/running, process 5269

salt-master stop/waiting

salt-master start/running, process 5313

* INFO: Running daemons_running()

* INFO: Salt installed!

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

root 4732 1 1 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 4744 1 1 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5313 1 1 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5324 5313 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5325 5313 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5326 5313 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5329 5313 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5380 5313 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5381 5313 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5382 5381 0 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5383 5381 2 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5394 5381 2 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5396 5381 2 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5400 5381 2 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

root 5401 5381 2 00:31 ? 00:00:00 /usr/bin/python /usr/bin/salt-master

dave 5931 2005 0 00:32 pts/0 00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo pkill -9 -f salt-master

Killed

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

[Overview](#) [Topics](#) [Documentation](#) [Downloads](#) [Editions](#)[2016.3.3](#) [2015.8.8](#) [Development](#)[Configuration Management](#) [Events & Reactor](#) [Orchestration](#) [Salt SSH](#) [Salt Cloud](#) [Salt Proxy Minion](#) [Salt Virt](#) [Command Line Reference](#) [Salt Module Reference](#) [APIs](#) [Architecture](#) [Windows](#) [Developing Salt](#) [Release Notes](#)

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

Setting up salt-minion (2016.3.3+ds-1) ...

salt-minion start/running, process 4604

Processing triggers for libc-bin (2.19-0ubuntu6.9) ...

Processing triggers for ureadahead (0.100.0-16) ...

* INFO: Running install_ubuntu_stable_post()

* INFO: Running install_ubuntu_check_services()

* INFO: Running install_ubuntu_restart_daemons()

salt-minion stop/waiting

salt-minion start/running, process 5269

salt-master stop/waiting

salt-master start/running, process 5313

* INFO: Running daemons_running()

* INFO: Salt installed!

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

User	PPID	PID	TTY	Time	Command
root	4732	1	1	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	4744	1	1	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5313	1	1	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5324	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5325	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5326	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5329	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5380	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5381	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5382	5381	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5383	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5394	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5396	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5400	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5401	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
dave	5931	2005	0	00:32 pts/0	00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo pkill -9 -f salt-master

Killed

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

dave 5951 2005 0 00:32 pts/0 00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

Overview Themes Documentation Downloads Details

2015.8.22 - 2016.10.27 DevOps

Configuration Management

Events & Reactor

Orchestration

Salt SSH

Salt Cloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

Processing triggers for libc-bin (2.19-0ubuntu6.9) ...

Processing triggers for ureadahead (0.100.0-16) ...

- INFO: Running install_ubuntu_stable_post()
- INFO: Running install_ubuntu_check_services()
- INFO: Running install_ubuntu_restart_daemons()

salt-minion stop/waiting

salt-minion start/running, process 5269

salt-master stop/waiting

salt-master start/running, process 5313

- INFO: Running daemons_running()
- INFO: Salt installed!

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

User	Pid	PPid	State	Time	Command
root	4732	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	4744	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5313	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5324	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5325	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5326	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5329	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5380	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5381	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5382	5381	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5383	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5394	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5396	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5400	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5401	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
dave	5931	2005	0 00:32 pts/0	00:00:00	grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo pkill -9 -f salt-master

Killed

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

dave 5951 2005 0 00:32 pts/0 00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo service salt-master start

salt-master start/running, process 5959

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ su

[Overview](#) [Topics](#) [Documentation](#) [Downloads](#) [Get Help](#)

2015-02-20 2015-02-20 2015-02-20

Configuration Management

Events & Reactor

Orchestration

Salt SSH

Salt Cloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

Processing triggers for ureadahead (0.100.0-16) ...

- INFO: Running install_ubuntu_stable_post()
- INFO: Running install_ubuntu_check_services()
- INFO: Running install_ubuntu_restart_daemons()

salt-minion stop/waiting

salt-minion start/running, process 5269

salt-master stop/waiting

salt-master start/running, process 5313

- INFO: Running daemons_running()
- INFO: Salt installed!

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

User	PPID	PID	TTY	Time	Command
root	4732	1	1	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	4744	1	1	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5313	1	1	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5324	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5325	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5326	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5329	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5380	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5381	5313	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5382	5381	0	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5383	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5394	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5396	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5400	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
root	5401	5381	2	00:31 ?	00:00:00 /usr/bin/python /usr/bin/salt-master
dave	5931	2005	0	00:32 pts/0	00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo pkill -9 -f salt-master

Killed

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

User	PPID	PID	TTY	Time	Command
dave	5951	2005	0	00:32 pts/0	00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo service salt-master start

salt-master start/running, process 5959

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo vi /etc/salt/minion

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Issues](#)[salt](#) [saltstack](#) [saltminion](#) [saltapi](#)

Configuration Management

Events & Reactor

Orchestration

Salt SSH

Salt Cloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

Processing triggers for libc-bin (2.19-0ubuntu6.9) ...

Processing triggers for ureadahead (0.100.0-16) ...

- INFO: Running install_ubuntu_stable_post()
- INFO: Running install_ubuntu_check_services()
- INFO: Running install_ubuntu_restart_daemons()

salt-minion stop/waiting

salt-minion start/running, process 5269

salt-master stop/waiting

salt-master start/running, process 5313

- INFO: Running daemons_running()
- INFO: Salt installed!

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

User	Pid	PPid	State	Time	Command
root	4732	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	4744	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5313	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5324	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5325	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5326	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5329	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5380	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5381	5313	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5382	5381	0 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5383	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5394	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5396	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5400	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5401	5381	2 00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
dave	5931	2005	0 00:32 pts/0	00:00:00	grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo pkill -9 -f salt-master

Killed

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

dave 5951 2005 0 00:32 pts/0 00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo service salt-master start

salt-master start/running, process 5959

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo vi /etc/salt/minion

Overview Tutorials Documentation Downloads Examples

2015-02-20 20:25:57 Devday

Configuration Management

Events & Reactor

Orchestration

SaltSSH

SaltCloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

```
dave@mastering-devops: ~/M...  
##### Primary configuration settings #####  
#####  
# This configuration file is used to manage the behavior of the Salt Minion.  
# With the exception of the location of the Salt Master Server, values that are  
# commented out but have an empty line after the comment are defaults that need  
# not be set in the config. If there is no blank line after the comment, the  
# value is presented as an example and is not the default.  
  
# Per default the minion will automatically include all config files  
# from minion.d/*.conf (minion.d is a directory in the same directory  
# as the main minion config file).  
#default_include: minion.d/*.conf  
  
# Set the location of the salt master server. If the master server cannot be  
# resolved, then the minion will fail to start.  
master: localhost  
  
# Set http proxy information for the minion when doing requests  
#proxy_host:  
#proxy_port:  
#proxy_username: bootstrap_salt.sh -L https://bootstrap-saltstack.com  
#proxy_password: bootstrap_salt.sh -g https://github.com/myuser/salt.git git mybranch  
  
# If multiple masters are specified in the 'master' setting, the default behavior  
# is to always try to connect to them in the order they are listed. If random_master is  
# set to True, the order will be randomized instead. This can be helpful in distributing  
# the load of many minions executing salt-call requests, for example, from a cron job.  
# If only one master is listed, this setting is ignored and a warning will be logged.  
# NOTE: If master_type is set to failover, use master_shuffle instead.  
#random_master: False  
  
# Use if master_type is set to failover.  
#master_shuffle: False  
#PROXY='http://user:password@myproxy.example.com:3128'  
# Minions can connect to multiple masters simultaneously (all masters  
-- INSERT --
```

Overview Topics Documentation Downloads Edition

2015-01-20 2015-01-21 Develop

Configuration Management

Events & Reactor

Orchestration

Salt-SSH

Salt-Cloud

Salt Proxy Minion

Salt-Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

* INFO: Running install_ubuntu_restart_daemons()

salt-minion stop/waiting

salt-minion start/running, process 5269

salt-master stop/waiting

salt-master start/running, process 5313

* INFO: Running daemons_running()

* INFO: Salt installed!

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

User	Pid	PPid	State	Start Time	CPU %	Command
root	4732	1	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	4744	1	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5313	1	1	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5324	5313	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5325	5313	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5326	5313	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5329	5313	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5380	5313	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5381	5313	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5382	5381	0	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5383	5381	2	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5394	5381	2	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5396	5381	2	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5400	5381	2	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
root	5401	5381	2	00:31 ?	00:00:00	/usr/bin/python /usr/bin/salt-master
dave	5931	2005	0	00:32 pts/0	00:00:00	grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo pkill -9 -f salt-master

Killed

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ ps -ef | grep salt-master

dave 5951 2005 0 00:32 pts/0 00:00:00 grep --color=auto salt-master

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo service salt-master start

salt-master start/running, process 5959

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo vi /etc/salt/minion

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo service salt-minion restart

salt-minion stop/waiting

salt-minion start/running, process 6589

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

Overview Tutorial Documentation Downloads Releases

2016-07-20 2016-08-01 2016-08-02 2016-08-03 2016-08-04 2016-08-05 2016-08-06 2016-08-07 2016-08-08 2016-08-09 2016-08-10 2016-08-11 2016-08-12 2016-08-13 2016-08-14 2016-08-15 2016-08-16 2016-08-17 2016-08-18 2016-08-19 2016-08-20 2016-08-21 2016-08-22 2016-08-23 2016-08-24 2016-08-25 2016-08-26 2016-08-27 2016-08-28 2016-08-29 2016-08-30 2016-08-31 2016-09-01 2016-09-02 2016-09-03 2016-09-04 2016-09-05 2016-09-06 2016-09-07 2016-09-08 2016-09-09 2016-09-10 2016-09-11 2016-09-12 2016-09-13 2016-09-14 2016-09-15 2016-09-16 2016-09-17 2016-09-18 2016-09-19 2016-09-20 2016-09-21 2016-09-22 2016-09-23 2016-09-24 2016-09-25 2016-09-26 2016-09-27 2016-09-28 2016-09-29 2016-09-30 2016-10-01 2016-10-02 2016-10-03 2016-10-04 2016-10-05 2016-10-06 2016-10-07 2016-10-08 2016-10-09 2016-10-10 2016-10-11 2016-10-12 2016-10-13 2016-10-14 2016-10-15 2016-10-16 2016-10-17 2016-10-18 2016-10-19 2016-10-20 2016-10-21 2016-10-22 2016-10-23 2016-10-24 2016-10-25 2016-10-26 2016-10-27 2016-10-28 2016-10-29 2016-10-30 2016-10-31 2016-11-01 2016-11-02 2016-11-03 2016-11-04 2016-11-05 2016-11-06 2016-11-07 2016-11-08 2016-11-09 2016-11-10 2016-11-11 2016-11-12 2016-11-13 2016-11-14 2016-11-15 2016-11-16 2016-11-17 2016-11-18 2016-11-19 2016-11-20 2016-11-21 2016-11-22 2016-11-23 2016-11-24 2016-11-25 2016-11-26 2016-11-27 2016-11-28 2016-11-29 2016-11-30 2016-12-01 2016-12-02 2016-12-03 2016-12-04 2016-12-05 2016-12-06 2016-12-07 2016-12-08 2016-12-09 2016-12-10 2016-12-11 2016-12-12 2016-12-13 2016-12-14 2016-12-15 2016-12-16 2016-12-17 2016-12-18 2016-12-19 2016-12-20 2016-12-21 2016-12-22 2016-12-23 2016-12-24 2016-12-25 2016-12-26 2016-12-27 2016-12-28 2016-12-29 2016-12-30 2016-12-31 2016-12-32 2016-12-33 2016-12-34 2016-12-35 2016-12-36 2016-12-37 2016-12-38 2016-12-39 2016-12-40 2016-12-41 2016-12-42 2016-12-43 2016-12-44 2016-12-45 2016-12-46 2016-12-47 2016-12-48 2016-12-49 2016-12-50 2016-12-51 2016-12-52 2016-12-53 2016-12-54 2016-12-55 2016-12-56 2016-12-57 2016-12-58 2016-12-59 2016-12-60 2016-12-61 2016-12-62 2016-12-63 2016-12-64 2016-12-65 2016-12-66 2016-12-67 2016-12-68 2016-12-69 2016-12-70 2016-12-71 2016-12-72 2016-12-73 2016-12-74 2016-12-75 2016-12-76 2016-12-77 2016-12-78 2016-12-79 2016-12-80 2016-12-81 2016-12-82 2016-12-83 2016-12-84 2016-12-85 2016-12-86 2016-12-87 2016-12-88 2016-12-89 2016-12-90 2016-12-91 2016-12-92 2016-12-93 2016-12-94 2016-12-95 2016-12-96 2016-12-97 2016-12-98 2016-12-99 2016-12-100 2016-12-101 2016-12-102 2016-12-103 2016-12-104 2016-12-105 2016-12-106 2016-12-107 2016-12-108 2016-12-109 2016-12-110 2016-12-111 2016-12-112 2016-12-113 2016-12-114 2016-12-115 2016-12-116 2016-12-117 2016-12-118 2016-12-119 2016-12-120 2016-12-121 2016-12-122 2016-12-123 2016-12-124 2016-12-125 2016-12-126 2016-12-127 2016-12-128 2016-12-129 2016-12-130 2016-12-131 2016-12-132 2016-12-133 2016-12-134 2016-12-135 2016-12-136 2016-12-137 2016-12-138 2016-12-139 2016-12-140 2016-12-141 2016-12-142 2016-12-143 2016-12-144 2016-12-145 2016-12-146 2016-12-147 2016-12-148 2016-12-149 2016-12-150 2016-12-151 2016-12-152 2016-12-153 2016-12-154 2016-12-155 2016-12-156 2016-12-157 2016-12-158 2016-12-159 2016-12-160 2016-12-161 2016-12-162 2016-12-163 2016-12-164 2016-12-165 2016-12-166 2016-12-167 2016-12-168 2016-12-169 2016-12-170 2016-12-171 2016-12-172 2016-12-173 2016-12-174 2016-12-175 2016-12-176 2016-12-177 2016-12-178 2016-12-179 2016-12-180 2016-12-181 2016-12-182 2016-12-183 2016-12-184 2016-12-185 2016-12-186 2016-12-187 2016-12-188 2016-12-189 2016-12-190 2016-12-191 2016-12-192 2016-12-193 2016-12-194 2016-12-195 2016-12-196 2016-12-197 2016-12-198 2016-12-199 2016-12-200 2016-12-201 2016-12-202 2016-12-203 2016-12-204 2016-12-205 2016-12-206 2016-12-207 2016-12-208 2016-12-209 2016-12-210 2016-12-211 2016-12-212 2016-12-213 2016-12-214 2016-12-215 2016-12-216 2016-12-217 2016-12-218 2016-12-219 2016-12-220 2016-12-221 2016-12-222 2016-12-223 2016-12-224 2016-12-225 2016-12-226 2016-12-227 2016-12-228 2016-12-229 2016-12-230 2016-12-231 2016-12-232 2016-12-233 2016-12-234 2016-12-235 2016-12-236 2016-12-237 2016-12-238 2016-12-239 2016-12-240 2016-12-241 2016-12-242 2016-12-243 2016-12-244 2016-12-245 2016-12-246 2016-12-247 2016-12-248 2016-12-249 2016-12-250 2016-12-251 2016-12-252 2016-12-253 2016-12-254 2016-12-255 2016-12-256 2016-12-257 2016-12-258 2016-12-259 2016-12-260 2016-12-261 2016-12-262 2016-12-263 2016-12-264 2016-12-265 2016-12-266 2016-12-267 2016-12-268 2016-12-269 2016-12-270 2016-12-271 2016-12-272 2016-12-273 2016-12-274 2016-12-275 2016-12-276 2016-12-277 2016-12-278 2016-12-279 2016-12-280 2016-12-281 2016-12-282 2016-12-283 2016-12-284 2016-12-285 2016-12-286 2016-12-287 2016-12-288 2016-12-289 2016-12-290 2016-12-291 2016-12-292 2016-12-293 2016-12-294 2016-12-295 2016-12-296 2016-12-297 2016-12-298 2016-12-299 2016-12-300 2016-12-301 2016-12-302 2016-12-303 2016-12-304 2016-12-305 2016-12-306 2016-12-307 2016-12-308 2016-12-309 2016-12-310 2016-12-311 2016-12-312 2016-12-313 2016-12-314 2016-12-315 2016-12-316 2016-12-317 2016-12-318 2016-12-319 2016-12-320 2016-12-321 2016-12-322 2016-12-323 2016-12-324 2016-12-325 2016-12-326 2016-12-327 2016-12-328 2016-12-329 2016-12-330 2016-12-331 2016-12-332 2016-12-333 2016-12-334 2016-12-335 2016-12-336 2016-12-337 2016-12-338 2016-12-339 2016-12-340 2016-12-341 2016-12-342 2016-12-343 2016-12

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

- INFO: Running daemons_running()
- INFO: Salt installed!

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ps -ef | grep salt-master
root    4732  1  1 00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    4744  1  1 00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5313  1  1 00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5324  5313  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5325  5313  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5326  5313  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5329  5313  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5380  5313  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5381  5313  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5382  5381  0  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5383  5381  2  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5394  5381  2  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5396  5381  2  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5400  5381  2  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
root    5401  5381  2  00:31 ?      00:00:00 /usr/bin/python /usr/bin/salt-master
dave    5931  2005  0  00:32 pts/0  00:00:00 grep --color=auto salt-master
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo pkill -9 -f salt-master
```

```
Killed curl -o DOCKER_0.11.0.tgz https://DOCKER-0.11.0.tgz.000
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ps -ef | grep salt-master
dave    5951  2005  0  00:32 pts/0  00:00:00 grep --color=auto salt-master
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo service salt-master start
salt-master start/running, process 5959
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo vi /etc/salt/minion
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo service salt-minion restart
salt-minion stop/waiting
```

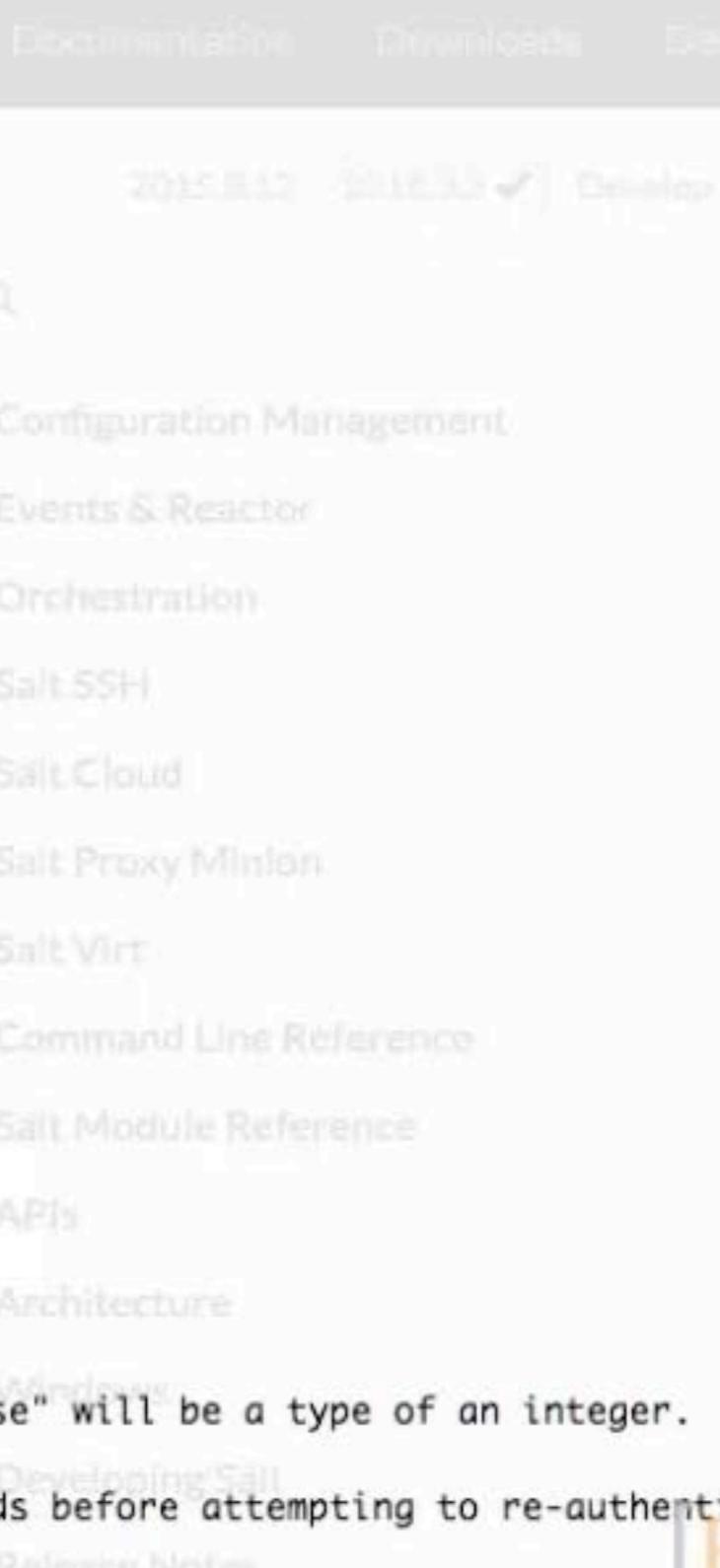
```
salt-minion start/running, process 6589
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo salt-call state.highstate
```

```
[WARNING] /usr/lib/python2.7/dist-packages/salt/grains/core.py:1493: DeprecationWarning: The "osmajorrelease" will be a type of an integer.
```

```
[ERROR] The Salt Master has cached the public key for this node, this salt minion will wait for 10 seconds before attempting to re-authenticate
Minion failed to authenticate with the master, has the minion key been accepted?
```

```
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```



2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

```
dave@mastering-devops: ~/M...
root      5324  5313  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5325  5313  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5326  5313  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5329  5313  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5380  5313  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5381  5313  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5382  5381  0 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5383  5381  2 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5394  5381  2 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5396  5381  2 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5400  5381  2 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
root      5401  5381  2 00:31 ?    00:00:00 /usr/bin/python /usr/bin/salt-master
dave     5931  2005  0 00:32 pts/0  00:00:00 grep --color=auto salt-master
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo pkill -9 -f salt-master
Killed
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ps -ef | grep salt-master
dave    5951  2005  0 00:32 pts/0  00:00:00 grep --color=auto salt-master
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo service salt-master start
salt-master start/running, process 5959
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo vi /etc/salt/minion
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo service salt-minion restart
salt-minion stop/waiting
salt-minion start/running, process 6589
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo salt-call state.highstate
[WARNING] /usr/lib/python2.7/dist-packages/salt/grains/core.py:1493: DeprecationWarning: The "osmajorrelease" will be a type of an integer.

[ERROR] The Salt Master has cached the public key for this node, this salt minion will wait for 10 seconds before attempting to re-authenticate
Minion failed to authenticate with the master, has the minion key been accepted?
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo salt-key -L
Accepted Keys:
Denied Keys:
Unaccepted Keys:
mastering-devops.c.dave-mangot.internal
Rejected Keys:
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

Configuration Management

Events & Reactor

Orchestration

SaltSSH

SaltCloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

```
dave@mastering-devops: ~/M...
root      5382  5381  0 00:31 ?          00:00:00 /usr/bin/python /usr/bin/salt-master
root      5383  5381  2 00:31 ?          00:00:00 /usr/bin/python /usr/bin/salt-master
root      5394  5381  2 00:31 ?          00:00:00 /usr/bin/python /usr/bin/salt-master
root      5396  5381  2 00:31 ?          00:00:00 /usr/bin/python /usr/bin/salt-master
root      5400  5381  2 00:31 ?          00:00:00 /usr/bin/python /usr/bin/salt-master
root      5401  5381  2 00:31 ?          00:00:00 /usr/bin/python /usr/bin/salt-master
dave     5931  2005  0 00:32 pts/0    00:00:00 grep --color=auto salt-master
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo pkill -9 -f salt-master
Killed
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ ps -ef | grep salt-master
dave    5951  2005  0 00:32 pts/0    00:00:00 grep --color=auto salt-master
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ 
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo service salt-master start
salt-master start/running, process 5959
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo vi /etc/salt/minion
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo service salt-minion restart
salt-minion stop/waiting
salt-minion start/running, process 6589
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo salt-call state.highstate
[WARNING] /usr/lib/python2.7/dist-packages/salt/grains/core.py:1493: DeprecationWarning: The "osmajorrelease" will be a type of an integer.
[ERROR] The Salt Master has cached the public key for this node, this salt minion will wait for 10 seconds before attempting to re-authenticate
Minion failed to authenticate with the master, has the minion key been accepted?
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo salt-key -L
Accepted Keys:
Denied Keys:
Unaccepted Keys:
mastering-devops.c.dave-mangot.internal
Rejected Keys:
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$ sudo salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
mastering-devops.c.dave-mangot.internal
Proceed? [n/Y] y
Key for minion mastering-devops.c.dave-mangot.internal accepted.
dave@mastering-devops:~/Mastering-DevOps/Section4/Video2$
```

[Configuration Management](#)[Events & Reactor](#)[Orchestration](#)[Salt SSH](#)[Salt Cloud](#)[Salt Proxy Minion](#)[Command Line Reference](#)[Salt Module Reference](#)[APIs](#)[Architecture](#)[Windows](#)[Developing Salt](#)[Release Notes](#)

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...

```
old: saltstack.com:latest.tar.gz -> bootstrap_salt.sh
python2.7-cairo:
```



new:

```
1
Using curl to pull latest development version from GitHub
```

old:

python2.7-pyparsing:

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

new:

1

old:

python2.7-simplejson:

new:

```
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh git v2015.0.6
```

old:

- ID: fs.file-max
Function: sysctl.present

Result: True

Comment: Updated sysctl value fs.file-max = 65536

Started: 00:34:39.537394

Duration: 32.733 ms

Changes:

If you want to install a salt-master using latest Git:

fs.file-max:

```
65536
curl -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
sudo sh bootstrap_salt.sh -N git develop
```

Summary for local

Succeeded: 2 (changed=2)

Failed: 0

Total states run: 2

Total run time: 14.333 s

dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2\$

[Overview](#) [Tutorials](#) [Documentation](#) [Downloads](#) [Bundles](#)
[2015.07](#) [2016.07](#) [Development](#)

Configuration Management

Events & Reactor

Orchestration

Salt SSH

Salt Cloud

Salt Proxy Minion

Salt Virt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

2. dave@mastering-devops: ~/Mastering-DevOps/Section4/Video2 (ssh)

dave@mastering-devops: ~/M...



```
new: saltstack.com/salt/_bootstrap/bootstrap.py --proxy https://bootstrap.saltstack.com
```

1

old:

python2.7-pyparsing:

Using curl to pull latest development version from GitHub.

new:

1

old:

```
curl -L https://bootstrap.saltstack.com/salt -o /tmp/salt-*.tar.gz
git clone https://github.com/saltstack/salt.git develop
```

new:

If you want to install a specific release version (based on the Git tags):

old:

```
ID: fs.file-max
Function: sysctl.present
    - salt.sh git v2015.8.9
Result: True
Comment: Updated sysctl value fs.file-max = 65536
Started: 00:34:39.537394
Duration: 32.733 ms
```

```
Changes: -o bootstrap_salt.sh -L https://bootstrap.saltstack.com
         sudo -E bootstrap_salt.sh -g https://github.com/myuser/salt.git mybranch
         fs.file-max:
```

65536

If all you want is to install a salt-minion using latest Git:

Summary for local

```
Succeeded: 2 (changed=2)
Failed: 0
```

Total states run: 2

Total run time: 14.333 s

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo salt '*' test.ping

mastering-devops.c.dave-mangot.internal:

True

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$

Overview

Tutorials

Documentation

Downloads

Design

2015.8.12 · 2015.8.13 · Details

Configuration Management

Events & Reactor

Orchestration

SaltSSH

SaltCloud

Salt Proxy Minion

SaltVirt

Command Line Reference

Salt Module Reference

APIs

Architecture

Windows

Developing Salt

Release Notes

Compute Engine - Dave Mangot × Salt Bootstrap × salt.states.service × GitHub - dmangot/Mastering... ×

Dave

https://docs.saltstack.com/en/latest/ref/states/all/salt.states.service.html

SALTSTACK

Overview Tutorials Documentation Downloads Develop

Note

The current status of a service is determined by the return code of the init/rc script status command. A status return code of 0 it is considered running. Any other return code is considered dead.

2015.8.12

2016.3.3 ✓

Develop

```
httpd:  
  service.running: []
```

The service can also be set to be started at runtime via the enable option:

```
openvpn:  
  service.running:  
    - enable: True
```

By default if a service is triggered to refresh due to a watch statement the service is by default restarted. If the desired behavior is to reload the service, then set the reload value to True:

```
redis:  
  service.running:  
    - enable: True  
    - reload: True  
    - watch:  
      - pkg: redis
```

salt.states.splunk

salt.states.splunk_search

salt.states.sqlite3

salt.states.ssh_auth

salt.states.ssh_known_hosts

salt.states.stateconf

salt.states.status

salt.states.stormpath_account

salt.states.supervisord

salt.states.svn

salt.states.sysctl

salt.states.syslog_ng

salt.states.sysrc

salt.states.telemetry_alert

salt.states.test

salt.states.timezone

Note

dave@mastering-devops: /srv...

2. dave@mastering-devops: /srv/salt (ssh)

https://old:saltstack.com/en/latest/ref/states/all/salt.states.service.html
python2.7-pyparsing:



new:

1

Note old:

python2.7-simplejson:

The current status of each service is determined by the return code of the init/rescanstatus command. A status

return code of 0 is considered running. Any other return code is considered dead.

new:

1

old:

ID: fs.file-max

Function: sysctl.present

Result: True

Comment: Updated sysctl value fs.file-max = 65536

Started: 00:34:39.537394

Duration: 32.733 ms

Changes:

fs.file-max:

65536

Summary for local

the desired behavior is to reload the service, then set the reload value to True:

Succeeded: 2 (changed=2)

Failed: 0

Total states run: 2

Total run time: 14.333 s

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo salt '*' test.ping

mastering-devops.c.dave-mangot.internal:

True

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ cd /srv/salt/

dave@mastering-devops:/srv/salt\$ sudo vi graphite.sls

dave@mastering-devops:/srv/salt\$ cd /srv/salt/

2015-09-03 05:36:53 [Desktop]



salt.states.splunk_search

salt.states.sqlite3

salt.states.ssh_auth

salt.states.ssh_known_hosts

salt.states.stateconf

salt.states.status

salt.states.stormpath_account

salt.states.supervised

salt.states.syn

salt.states.sysctl

salt.states.syslog_ng

salt.states.sync

salt.states.telemetry_alert

salt.states.test

salt.states.timezone

dave@mastering-devops: /srv...

2. dave@mastering-devops: /srv/salt (ssh)

python2.7-pyparsing: en/latest/ref/states/all/salt.states.service.html



new:

old:

Now python2.7-simplejson:

The current status of a service is determined by the return code of the init/rcscript/status command. A status

new:

return code of 0 is considered running. Any other return code is considered dead.

old:

ID: fs.file-max

Function: sysctl.present

Result: True (running: 1)

Comment: Updated sysctl value fs.file-max = 65536

Started: 00:34:39.537394 To be started at runtime via the enable option.

Duration: 32.733 ms

Changes:

open(-----)

fs.file-max:

65536

True

Summary for local

If a service is triggered to refresh due to a watch statement the service is by default restarted. If

Succeeded: 2 (changed=2)

r is to reload the service, then set the reload value to True.

Failed: 0

Total states run: 2

Total run time: 14.333 s

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ sudo salt '*' test.ping

mastering-devops.c.dave-mangot.internal:

True

dave@mastering-devops:~/Mastering-DevOps/Section4/Video2\$ cd /srv/salt/

dave@mastering-devops:/srv/salt\$ sudo vi graphite.sls

dave@mastering-devops:/srv/salt\$ sudo salt '*' state.highstate

Overview Themes Documentation Downloads Details

salt.states

DevLog

salt.states.salt	salt.states.salt
salt.states.splunk_search	salt.states.splunk_search
salt.states.sqlite3	salt.states.sqlite3
salt.states.ssh_auth	salt.states.ssh_auth
salt.states.ssh_known_hosts	salt.states.ssh_known_hosts
salt.states.stateconf	salt.states.stateconf
salt.states.status	salt.states.status
salt.states.stompmail_account	salt.states.stompmail_account
salt.states.supervisord	salt.states.supervisord
salt.states.syn	salt.states.syn
salt.states.sysctl	salt.states.sysctl
salt.states.syslog_ng	salt.states.syslog_ng
salt.states.sync	salt.states.sync
salt.states.telemetry_alert	salt.states.telemetry_alert
salt.states.test	salt.states.test
salt.states.timezone	salt.states.timezone

2. dave@mastering-devops: /srv/salt (ssh)

```
dave@mastering-devops: /srv/salt$ sudo vi graphite.sls
dave@mastering-devops:/srv/salt$ sudo salt '*' state.highstate
mastering-devops.c.dave-mangot.internal:
```

```
-----[SNIP]-----[SNIP]
ID: graphite
Function: pkg.installed
Result: True
Comment: All specified packages are already installed
Started: 00:36:22.898657
Duration: 301.799 ms
Changes:
```

```
-----[SNIP]-----[SNIP]
ID: carbon-cache
Function: service.running
Result: True
Comment: The service carbon-cache is already running
Started: 00:36:23.201033
Duration: 23.931 ms
Changes:
```

```
-----[SNIP]-----[SNIP]
ID: fs.file-max
Function: sysctl.present
Result: True
Comment: Sysctl value fs.file-max = 65536 is already set
Started: 00:36:23.225449
Duration: 12.38 ms
Changes:
```

```
Summary for mastering-devops.c.dave-mangot.internal
```

```
-----[SNIP]-----[SNIP]
enable: True
Succeeded: 3
Failed: 0
watcher:
-----[SNIP]-----[SNIP]
```

```
Total states run: 3
```

```
Total run time: 338.110 ms
```

```
dave@mastering-devops:/srv/salt$
```

```
2015-09-03 17:36:52 [salt.statefunck_search]
```

```
salt.states.salt3
```

```
salt.states.salt_auth
```

```
salt.states.ssh_known_hosts
```

```
salt.states.stateconf
```

```
salt.states.status
```

```
salt.states.stompsh_account
```

```
salt.states.supervisord
```

```
salt.states.svnm
```

```
salt.states.sysctl
```

```
salt.states.syslog_ng
```

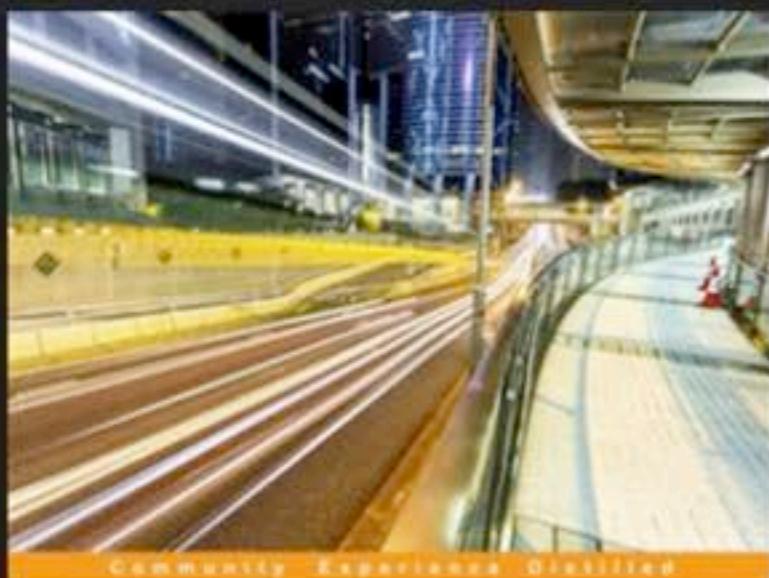
```
salt.states.sync
```

```
salt.states.telnetv_alert
```

```
salt.states.test
```

```
salt.states.timezone
```

For More Information



Learning SaltStack *Second Edition*

Build, manage and secure your infrastructure by utilizing the benefits of SaltStack

Colton Myers

PACKT
open source*

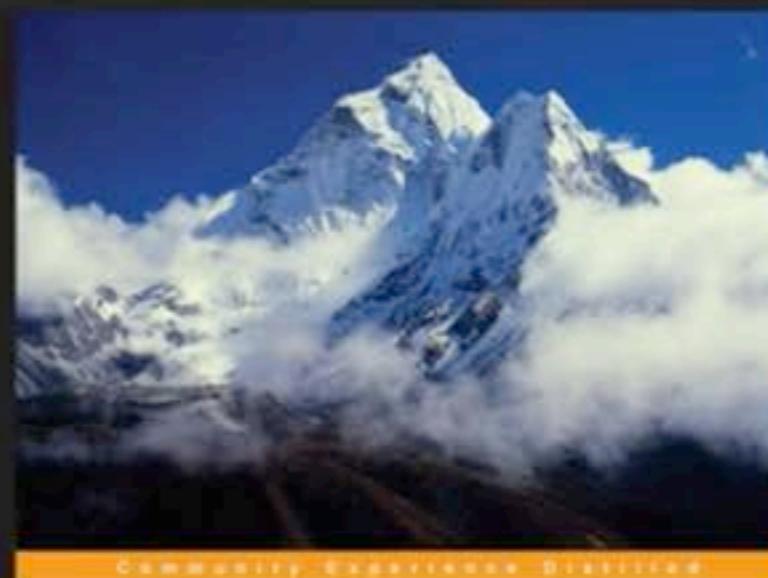
Anirban Saha

[PACKT] open source*



Salt Cookbook

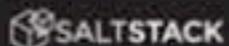
Over 80 hands-on recipes to efficiently configure and manage your infrastructure with Salt



Mastering SaltStack

Take charge of SaltStack to automate and configure enterprise-grade environments

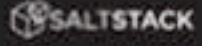
Foreword by Thomas S. Hatch - Founder and CTO, SaltStack



Extending SaltStack

Extend the power of your infrastructure and applications with Salt modules

Foreword by Eric Johnson - Author of SaltStack's Google Compute Engine cloud module



Next Video

What About Containers?

Dave Mangot

Video 4.4

What About Containers?



What Are Containers?



- Optimize flow
- Get faster feedback
- Run more experiments

What Are Containers?

DEBS

JAILS

LXC

RPMS

ZONES



Advantages of Containers



- Have all requirements
- Same bits
- Have some isolation
- Start/Stop in fraction of a second

When Do I Use Which?

- Configuration management
 - Configure the pipeline
 - Configure the host
 - Configure the repositories
- Containers
 - Package up all the dependencies
 - Configure the application
 - Configure the runtime

Next Video

Building and Running a Docker Container

Dave Mangot

Video 4.5

*Building and Running a
Docker Container*



In this Video, we are going to take a look at...

- Installing Docker
- Building a Docker image
- Running a Docker container
- Connecting to a Docker container

Installing, Building, and Running Docker



- o <https://docs.docker.com/v1.8/installation/ubuntulinux/#installation>
- o <https://github.com/dmangot/Mastering-DevOps>

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

```
ubuntu@ip-172-31-38-74: ~...  
ubuntu@ip-172-31-38-74:~$ git clone https://github.com/dmangot/Mastering-DevOps.git  
Cloning into 'Mastering-DevOps'...  
remote: Counting objects: 108, done.  
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108  
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.  
Resolving deltas: 100% (20/20), done.  
Checking connectivity... done.  
ubuntu@ip-172-31-38-74:~$ cd Mastering-DevOps/Section4/Video5  
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ uname -a  
Linux ip-172-31-38-74 3.13.0-92-generic #139-Ubuntu SMP Tue Jun 28 20:42:26 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux  
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$
```

<https://docs.docker.com/v1.8/installation/ubuntu/>#installation
<https://github.com/dmangot/Mastering-DevOps>

EC2 Management Console Installation on Ubuntu GitHub - dmangot/Mastering

Dave

https://docs.docker.com/v1.8/installation/ubuntulinux/#installation

3. Get the latest Docker package.

```
$ curl -sSL https://get.docker.com/ | sh
```

The system prompts you for your `sudo` password. Then, it downloads and installs Docker and its dependencies.

Note: If your company is behind a filtering proxy, you may find that the `apt-key` command fails for the Docker repo during installation. To work around this, add the key directly using the following:

```
$ curl -sSL https://get.docker.com/gpg | sudo apt-key add
```

1. Verify `docker` is installed correctly.

```
$ sudo docker run hello-world
```

This command downloads a test image and runs it in a container.

Optional configurations for Docker on Ubuntu

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

```
ubuntu@ip-172-31-38-74: ~...  
ubuntu@ip-172-31-38-74:~$ git clone https://github.com/dmangot/Mastering-DevOps.git  
Cloning into 'Mastering-DevOps'...  
remote: Counting objects: 108, done.  
remote: Total 108 (delta 0), reused 0 (delta 0), pack-reused 108  
Receiving objects: 100% (108/108), 27.28 KiB | 0 bytes/s, done.  
Resolving deltas: 100% (20/20), done. 1. Verify docker is installed correctly.  
Checking connectivity... done.  
ubuntu@ip-172-31-38-74:~$ cd Mastering-DevOps/Section4/Video5  
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ uname -a  
Linux ip-172-31-38-74 3.13.0-92-generic #139-Ubuntu SMP Tue Jun 28 20:42:26 UTC 2016 x86_64 x86_64 x86_64 GNU/Linux  
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ curl -sSL https://get.docker.com/ | sh  
modprobe: FATAL: Module aufs not found.  
+ sudo -E sh -c sleep 3; apt-get update  This command downloads a test image and runs it in a container.
```

Optional configurations for Docker on Ubuntu

This section contains optional procedures for configuring your Ubuntu to work better with Docker.

- Create a docker group
- Limit memory and swap accounting
- Enable UPnP forwarding
- Configure a DNS server for use by Docker
- Configure Docker to start on boot

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~...

```
Processing triggers for man-db (2.6.7.1-1ubuntu1) ...
Processing triggers for ureadahead (0.100.0-16) ...
Setting up libltdl7:amd64 (2.4.2-1.7ubuntu1) ...
Setting up libsystemd-journal0:amd64 (204-Subuntu20.19) ...
Setting up aufs-tools (1:3.2+20130722-1.1) ...
Setting up docker-engine (1.12.1-0~trusty) ... docker is installed correctly.
docker start/running, process 29504
Setting up cgroup-lite (1.9) ...
cgroup-lite start/running
$ sudo docker run hello-world
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
+ sudo -E sh -c docker version
Client:
Version: 1.12.1
API version: 1.24
Go version: go1.6.3
Git commit: 23cf638
Built: Thu Aug 18 05:22:43 2016
OS/Arch: linux/amd64
```

Server:

```
Version: 1.12.1
API version: 1.24
Go version: go1.6.3
Git commit: 23cf638
Built: Thu Aug 18 05:22:43 2016
OS/Arch: linux/amd64
```

If you would like to use Docker as a non-root user, you should now consider adding your user to the "docker" group with something like:

```
sudo usermod -aG docker ubuntu
```

Remember that you will have to log out and back in for this to take effect!

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$
```

Optional configurations for Docker on Ubuntu

This section contains optional procedures for configuring your Ubuntu to work better with Docker.

* Create a Docker group

* Shared memory and swap accounting

* Configure a Docker daemon to use by Docker

* Using Docker daemon on port

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~/

```
Setting up libltdl7:amd64 (2.4.2-1.7ubuntu1) ...
Setting up libsystemd-journal0:amd64 (204-5ubuntu20.19) ...
Setting up aufs-tools (1:3.2+20130722-1.1) ...
Setting up docker-engine (1.12.1-0~trusty) ...
docker start/running, process 29504
Setting up cgroup-lite (1.9) ... 1. Verify docker is installed correctly.
cgroup-lite start/running
Processing triggers for libc-bin (2.19-0ubuntu6.9) ...
Processing triggers for ureadahead (0.100.0-16) ...
+ sudo -E sh -c docker version
```

Client:

```
Version: 1.12.1
API version: 1.24
Go version: go1.6.3
Git commit: 23cf638
Built: Thu Aug 18 05:22:43 2016
OS/Arch: linux/amd64
```

Server:

```
Version: 1.12.1
API version: 1.24
Go version: go1.6.3
Git commit: 23cf638
Built: Thu Aug 18 05:22:43 2016
OS/Arch: linux/amd64
```

If you would like to use Docker as a non-root user, you should now consider adding your user to the "docker" group with something like:

```
sudo usermod -aG docker ubuntu
```

Remember that you will have to log out and back in for this to take effect!

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo service docker status
docker start/running, process 29504
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$
```

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~/

adding your user to the "docker" group with something like:

```
sudo usermod -aG docker ubuntu
```

Remember that you will have to log out and back in for this to take effect!

```
1. Verify docker is installed correctly.  
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo service docker status  
docker start/running, process 29504  
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo docker run hello-world  
Unable to find image 'hello-world:latest' locally  
latest: Pulling from library/hello-world  
c04b14da8d14: Pull complete  
Digest: sha256:0256e8a36e2070f7bf2d0b0763dbabdd67798512411de4cdcf9431a1feb60fd9  
Status: Downloaded newer image for hello-world:latest
```

Hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker Hub account:

<https://hub.docker.com>

For more examples and ideas, visit:

<https://docs.docker.com/engine/userguide/>

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~/

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

\$ sudo docker run hello-world

To try something more ambitious, you can run an Ubuntu container with:

\$ docker run -it ubuntu bash

This command downloads a test image and runs it in a container.

Share images, automate workflows, and more with a free Docker Hub account:

<https://hub.docker.com>

For more examples and ideas, visit:

<https://docs.docker.com/engine/userguide/>

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ ls

Dockerfile

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ cat Dockerfile

what operating system

FROM ubuntu

some metadata

LABEL Description="Mastering DevOps" Version="1.0"

install the packages

RUN apt-get update && apt-get install -y \

graphite-carbon

which TCP port for carbon

EXPOSE 2003

where to write the data

VOLUME /var/lib/graphite/whisper

--nodaemon because we'll detach in 'docker run'

ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~/...

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash           $ sudo docker run hello-world
```

Share images, automate workflows, and more with a free Docker Hub account:

<https://hub.docker.com>

This command downloads a test image and runs it in a container.

For more examples and ideas, visit:

<https://docs.docker.com/engine/userguide/>

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ ls

Dockerfile

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ cat Dockerfile

what operating system

FROM ubuntu

some metadata

LABEL Description="Mastering DevOps" Version="1.0"

install the packages

RUN apt-get update && apt-get install -y \ graphite-carbon

which TCP port for carbon

EXPOSE 2003

where to write the data

VOLUME /var/lib/graphite/whisper

--nodaemon because we'll detach in 'docker run'

ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ telnet localhost 2003

Trying 127.0.0.1...

telnet: Unable to connect to remote host: Connection refused

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~/...

Share images, automate workflows, and more with a free Docker Hub account:
<https://hub.docker.com>

For more examples and ideas, visit:

<https://docs.docker.com/engine/userguide/> Verify docker is installed correctly.

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ ls

Dockerfile

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ cat Dockerfile

what operating system

FROM ubuntu

some metadata

LABEL Description="Mastering DevOps" Version="1.0"

install the packages

RUN apt-get update && apt-get install -y \ graphite-carbon

which TCP port for carbon

EXPOSE 2003

where to write the data

VOLUME /var/lib/graphite/whisper

--nodaemon because we'll detach in 'docker run'

ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ telnet localhost 2003

Trying 127.0.0.1...

telnet: Unable to connect to remote host: Connection refused

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker build -t dmangot/graphite .

Sending build context to Docker daemon 2.048 kB

Step 1 : FROM ubuntu

latest: Pulling from library/ubuntu

952132ac251a: Downloading [=====>] 30.47 MB/49.73 MB

82659f8f1b76: Download complete

c19118ca682d: Download complete

8296858250fe: Download complete

24e0251a0e2c: Download complete

Optional configurations for Docker
on Ubuntu

This section contains optional procedures for configuring your Ubuntu to

use Docker.

Install Docker

Configure Docker

Run Docker

Stop Docker

Uninstall Docker

Update Docker

Install Docker Compose

Configure Docker Compose

Run Docker Compose

Stop Docker Compose

Uninstall Docker Compose

Update Docker Compose

Install Docker Swarm

Configure Docker Swarm

Run Docker Swarm

Stop Docker Swarm

Uninstall Docker Swarm

Update Docker Swarm

Install Docker Machine

Configure Docker Machine

Run Docker Machine

Stop Docker Machine

Uninstall Docker Machine

Update Docker Machine

Install Docker Volume

Configure Docker Volume

Run Docker Volume

Stop Docker Volume

Uninstall Docker Volume

Update Docker Volume

Install Docker Network

Configure Docker Network

Run Docker Network

Stop Docker Network

Uninstall Docker Network

Update Docker Network

Install Docker Buildx

Configure Docker Buildx

Run Docker Buildx

Stop Docker Buildx

Uninstall Docker Buildx

Update Docker Buildx

Install Docker Compose

Configure Docker Compose

Run Docker Compose

Stop Docker Compose

Uninstall Docker Compose

Update Docker Compose

Install Docker Machine

Configure Docker Machine

Run Docker Machine

Stop Docker Machine

Uninstall Docker Machine

Update Docker Machine

Install Docker Volume

Configure Docker Volume

Run Docker Volume

Stop Docker Volume

Uninstall Docker Volume

Update Docker Volume

Install Docker Network

Configure Docker Network

Run Docker Network

Stop Docker Network

Uninstall Docker Network

Update Docker Network

Install Docker Buildx

Configure Docker Buildx

Run Docker Buildx

Stop Docker Buildx

Uninstall Docker Buildx

Update Docker Buildx

Install Docker Compose

Configure Docker Compose

Run Docker Compose

Stop Docker Compose

Uninstall Docker Compose

Update Docker Compose

Install Docker Machine

Configure Docker Machine

Run Docker Machine

Stop Docker Machine

Uninstall Docker Machine

Update Docker Machine

Install Docker Volume

Configure Docker Volume

Run Docker Volume

Stop Docker Volume

Uninstall Docker Volume

Update Docker Volume

Install Docker Network

Configure Docker Network

Run Docker Network

Stop Docker Network

Uninstall Docker Network

Update Docker Network

Install Docker Buildx

Configure Docker Buildx

Run Docker Buildx

Stop Docker Buildx

Uninstall Docker Buildx

Update Docker Buildx

Install Docker Compose

Configure Docker Compose

Run Docker Compose

Stop Docker Compose

Uninstall Docker Compose

Update Docker Compose

Install Docker Machine

Configure Docker Machine

Run Docker Machine

Stop Docker Machine

Uninstall Docker Machine

Update Docker Machine

Install Docker Volume

Configure Docker Volume

Run Docker Volume

Stop Docker Volume

Uninstall Docker Volume

Update Docker Volume

Install Docker Network

Configure Docker Network

Run Docker Network

Stop Docker Network

Uninstall Docker Network

Update Docker Network

Install Docker Buildx

Configure Docker Buildx

Run Docker Buildx

Stop Docker Buildx

Uninstall Docker Buildx

Update Docker Buildx

Install Docker Compose

Configure Docker Compose

Run Docker Compose

Install Docker Machine

Configure Docker Machine

Run Docker Machine

Stop Docker Machine

Uninstall Docker Machine

Update Docker Machine

Install Docker Volume

Configure Docker Volume

Run Docker Volume

Stop Docker Volume

Uninstall Docker Volume

Update Docker Volume

Install Docker Network

Configure Docker Network

Run Docker Network

Stop Docker Network

Uninstall Docker Network

Update Docker Network

Install Docker Buildx

Configure Docker Buildx

Run Docker Buildx

Stop Docker Buildx

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~/

You may want to keep these database files even if you completely remove graphite-carbon, in case you plan to reinstall it later.

Remove database files when purging graphite-carbon? [yes/no]

Use of uninitialized value \$_[1] in join or string at /usr/share/perl5/Debconf/DbDriver/Stack.pm line 111.

invoke-rc.d: could not determine current runlevel

invoke-rc.d: policy-rc.d denied execution of start.

Use of uninitialized value \$val in substitution (s///) at /usr/share/perl5/Debconf/Format/822.pm line 83, <GEN6> line 1.

Use of uninitialized value \$val in concatenation (.) or string at /usr/share/perl5/Debconf/Format/822.pm line 84, <GEN6> line 1.

Setting up python-pam (0.4.2-13.2ubuntu2) ...

Setting up python-serial (3.0.1-1) ...

Processing triggers for libc-bin (2.23-0ubuntu3) ...

Processing triggers for python-twisted-core (16.0.0-1) ...

Processing triggers for systemd (229-4ubuntu7) ...

---> 3196644019ac

Removing intermediate container 70a93de1e972

Step 4 : EXPOSE 2003

---> Running in cbf23270c318

---> eab2cd377eb7

Removing intermediate container cbf23270c318

Step 5 : VOLUME /var/lib/graphite/whisper

---> Running in 53241b04157f

---> b155b33daeaa

Removing intermediate container 53241b04157f

Step 6 : ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start

---> Running in 68a97399e8ea

---> b1c4cee02229

Removing intermediate container 68a97399e8ea

Successfully built b1c4cee02229

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dmangot/graphite	latest	b1c4cee02229	15 seconds ago	233.2 MB
ubuntu	latest	bd3d4369aebc	10 days ago	126.6 MB
hello-world	latest	c54a2cc56cbb	9 weeks ago	1.848 kB
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$				

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~...

graphite-carbon, in case you plan to reinstall it later.

Remove database files when purging graphite-carbon? [yes/no]

Use of uninitialized value \$_[1] in join or string at /usr/share/perl5/Debconf/DbDriver/Stack.pm line 111.

invoke-rc.d: could not determine current runlevel

invoke-rc.d: policy-rc.d denied execution of start.acker is installed correctly

Use of uninitialized value \$val in substitution (\$///) at /usr/share/perl5/Debconf/Format/822.pm line 83, <GEN6> line 1.

Use of uninitialized value \$val in concatenation (.) or string at /usr/share/perl5/Debconf/Format/822.pm line 84, <GEN6> line 1.

Setting up python-pam (0.4.2-13.2ubuntu2) ... sudo docker run hello-world

Setting up python-serial (3.0.1-1) ...

Processing triggers for libc-bin (2.23-0ubuntu3) ...

Processing triggers for python-twisted-core (16.0.0-1) ...

Processing triggers for systemd (229-4ubuntu7) ... command downloads a test image and runs it in a container.

----> 3196644019ac

Removing intermediate container 70a93de1e972

Step 4 : EXPOSE 2003

----> Running in cbf23270c318

----> eab2cd377eb7

Removing intermediate container cbf23270c318

Step 5 : VOLUME /var/lib/graphite/whisper

----> Running in 53241b04157f

----> b155b33daeaa

Removing intermediate container 53241b04157f better with Docker.

Step 6 : ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start

----> Running in 68a97399e8ea

----> b1c4cee02229

Removing intermediate container 68a97399e8ea

Successfully built b1c4cee02229

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dmangot/graphite	latest	b1c4cee02229	15 seconds ago	233.2 MB
ubuntu	latest	bd3d4369aebc	10 days ago	126.6 MB
hello-world	latest	c54a2cc56cbb	9 weeks ago	1.848 kB

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker run -it -d -p 2003:2003 --name devops dmangot/graphite

439a0065d376fe18fd8cdf9fdc71661c35270eceb7e1ec7689e7dcf37a5db61b

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~...

Use of uninitialized value \$_[1] in join or string at /usr/share/perl5/Debconf/DbDriver/Stack.pm line 111.
invoke-rc.d: could not determine current runlevel
invoke-rc.d: policy-rc.d denied execution of start.
Use of uninitialized value \$val in substitution (\$///) at /usr/share/perl5/Debconf/Format/822.pm line 83, <GEN6> line 1.
Use of uninitialized value \$val in concatenation (.) or string at /usr/share/perl5/Debconf/Format/822.pm line 84, <GEN6> line 1.
Setting up python-pam (0.4.2-13.2ubuntu2) ... docker is installed correctly.
Setting up python-serial (3.0.1-1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Processing triggers for python-twisted-core (16.0.0-1) ...
Processing triggers for systemd (229-4ubuntu7) ...

---> 3196644019ac

Removing intermediate container 70a93de1e972

Step 4 : EXPOSE 2003

---> Running in cbf23270c318
---> eab2cd377eb7

Removing intermediate container cbf23270c318

Step 5 : VOLUME /var/lib/graphite/whisper

---> Running in 53241b04157f
---> b155b33daea

Removing intermediate container 53241b04157f

Step 6 : ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start

---> Running in 68a97399e8ea
---> b1c4cee02229

Removing intermediate container 68a97399e8ea

Successfully built b1c4cee02229

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dmangot/graphite	latest	b1c4cee02229	15 seconds ago	233.2 MB
ubuntu	latest	bd3d4369aebc	10 days ago	126.6 MB
hello-world	latest	c54a2cc56cbb	9 weeks ago	1.848 kB

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker run -it -d -p 2003:2003 --name devops dmangot/graphite
439a0065d376fe18fd8cdf9fdc71661c35270eceb7e1ec7689e7dcf37a5db61b

ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5\$ sudo docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
439a0065d376	dmangot/graphite	/bin/sh -c '/usr/bin'	12 seconds ago	Up 12 seconds	0.0.0.0:2003->2003/tcp	devops

2. ubuntu@ip-172-31-38-74: ~/Mastering-DevOps/Section4/Video5 (ssh)

ubuntu@ip-172-31-38-74: ~...

```
Setting up python-pam (0.4.2-13.2ubuntu2) ...
Setting up python-serial (3.0.1-1) ...
Processing triggers for libc-bin (2.23-0ubuntu3) ...
Processing triggers for python-twisted-core (16.0.0-1) ...
Processing triggers for systemd (229-4ubuntu7) ...
--> 3196644019ac
```

```
Removing intermediate container 70a93de1e972
```

```
Step 4 : EXPOSE 2003
```

```
--> Running in cbf23270c318      $ sudo docker run hello-world
--> eab2cd377eb7
```

```
Removing intermediate container cbf23270c318
```

```
Step 5 : VOLUME /var/lib/graphite/whisper
```

```
--> Running in 53241b04157f      This command downloads a test image and runs it in a container.
--> b155b33daaaa
```

```
Removing intermediate container 53241b04157f
```

```
Step 6 : ENTRYPOINT /usr/bin/carbon-cache --config /etc/carbon/carbon.conf --nodaemon start
```

```
--> Running in 68a97399e8ea
--> b1c4cee02229
```

```
Removing intermediate container 68a97399e8ea
```

```
Successfully built b1c4cee02229
```

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
dmangot/graphite	latest	b1c4cee02229	15 seconds ago	233.2 MB
ubuntu	latest	bd3d4369aebc	10 days ago	126.6 MB
hello-world	latest	c54a2cc56cbb	9 weeks ago	1.848 kB

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo docker run -it -d -p 2003:2003 --name devops dmangot/graphite
```

```
439a0065d376fe18fd8cdf9fdc71661c35270eceb7e1ec7689e7dcf37a5db61b
```

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
439a0065d376	dmangot/graphite	/bin/sh -c '/usr/bin'	12 seconds ago	Up 12 seconds	0.0.0.0:2003->2003/tcp	devops

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
439a0065d376	dmangot/graphite	/bin/sh -c '/usr/bin'	22 seconds ago	Up 22 seconds	0.0.0.0:2003->2003/tcp	devops
12c42d591be3	hello-world	/hello	4 minutes ago	Exited (0) 4 minutes ago		hopeful_stonebreaker

```
ubuntu@ip-172-31-38-74:~/Mastering-DevOps/Section4/Video5$
```

Summary

- Configuration Management (CM)
- CM with SaltStack
- CM for distributed systems
- Containers
- Build and ran a Docker container

Next Section

Automation – Continuous Delivery