



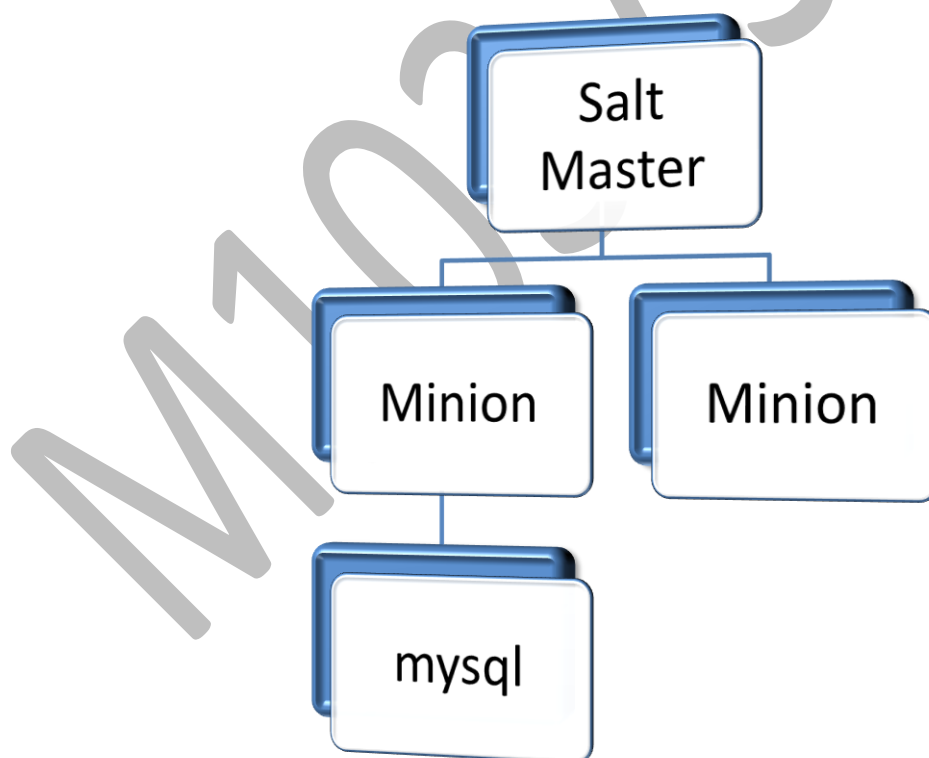
SALTSTACK



Saltstack

Saltstack is a Open project infrastructure management tool which helps in using infrastructure as a code. It delivers dynamic infrastructure communication used for orchestration, remote execution, configuration management. It is based on master and minion architecture, master commands the minions to perform task or it may be event based tasks.

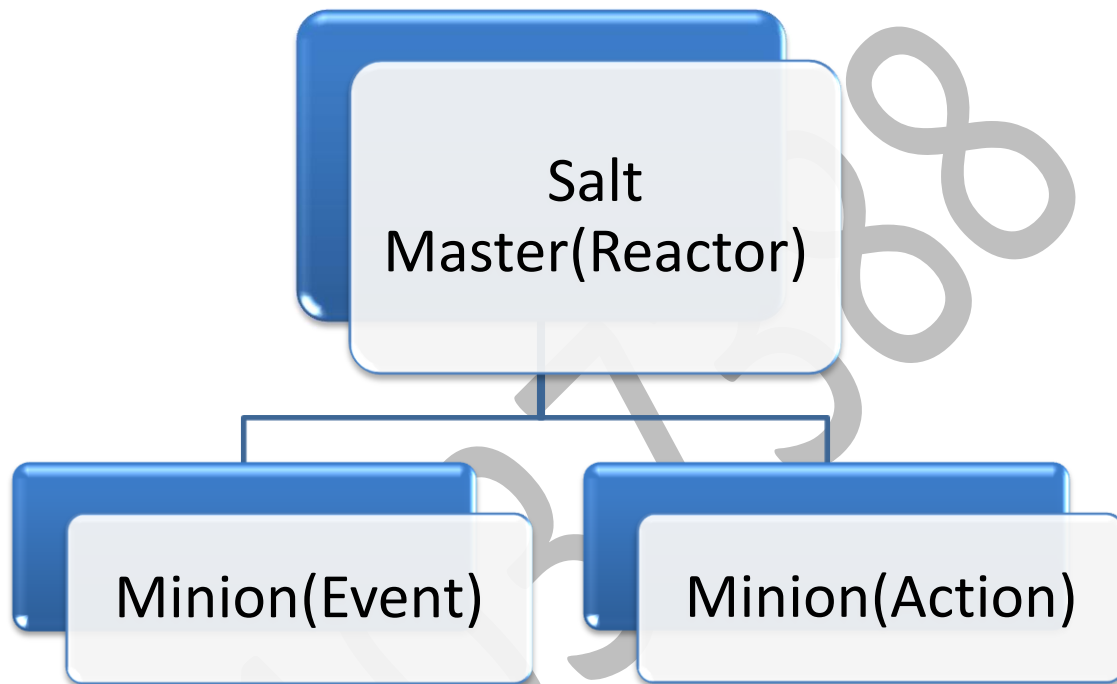
Architecture



Salt master can make the minion communicate with other service.



- **Peer-Interface:** the ability of a minion to command other minions.
- **Event-reactor interface:** where the master reacts to the event taken place in minion to perform any action.



- There is key based authentication between the master and minion.
- To command the minion we use **salt formula** which is written in **jinja**.
- **Jinja** is a template engine for python.



Installation

- Before you install update your VM by
apt-get update
- To install Salt master on VM.

apt-get install salt-master

- To install salt minion on VM.

apt-get install salt-minion

- Now we should make sure that minion identifies its master by configuring the Salt master IP in
/etc/salt/minion.

vi /etc/salt/minion

```
root@ip-172-31-35-161:~# vi /etc/salt/minion
##### 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

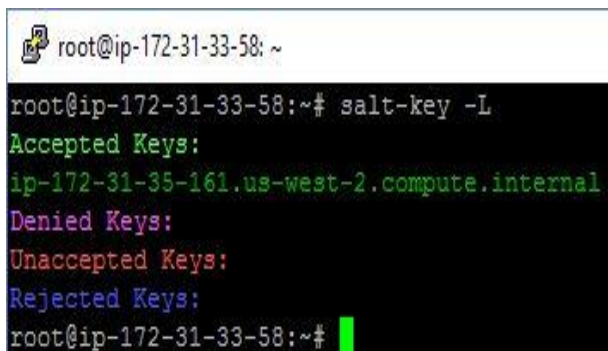
# Set the location of the salt master server. If the master server cannot be
# reached, then the minion will fail to start.
master: 35.165.239.108
#master_finger: 4c01:d3:bu:3e:73:d3:4u:04:48:1b:58:0e:1f:6d:6d:fa:1e:34:89:9b:77:nd:79:0e:0d:2b:d3:d3:b4"
# Set http proxy information for the minion when doing requests
#proxy_host:
#proxy_port:
#proxy_username:
#proxy_password:

# 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
```



- Now salt needs to register to its master by **Salt-minion**
- Salt master should accept the key, we can see the list of keys by **salt-key -L**
- We can accept the key from the minion by **salt-key -A**



```
root@ip-172-31-33-58: ~  
root@ip-172-31-33-58:~# salt-key -L  
Accepted Keys:  
ip-172-31-35-161.us-west-2.compute.internal  
Denied Keys:  
Unaccepted Keys:  
Rejected Keys:  
root@ip-172-31-33-58:~#
```

- Now the salt master and minions are connected.
- To check we can give a simple command to the minion **Salt * test.ping**
This command will ping all the minions present under that salt master.



```
root@ip-172-31-33-58:/srv/salt# salt '*' test.ping
ip-172-31-35-161.us-west-2.compute.internal:
    True
root@ip-172-31-33-58:/srv/salt#
```

Salt formula to install apache in minion

- Let us create a directory in salt master by **mkdir -p /srv/salt/**
- In this directory we create a file name **apache.sls** **vi apache.sls** and write the salt formula

```
root@ip-172-31-33-58: /srv/salt
```

```
apache:
  pkg.installed:
    - name: apache2
  service.running:
    - enable: True
    - require:
      - pkg: apache
```



- To execute this salt formula

salt '*' state.sls apache

this command will apply the salt formula to every minion under it.

```
root@ip-172-31-33-58: /srv/salt
root@ip-172-31-33-58:/srv/salt# salt '*' state.sls apache
ip-172-31-35-161.us-west-2.compute.internal:
-----
ID: apache
Function: pkg.installed
Name: apache2
Result: True
Comment: Package apache2 is already installed
Started: 18:05:22.118713
Duration: 316.062 ms
Changes:
-----
ID: apache
Function: service.running
Result: False
Comment: The named service apache is not available
Started: 18:05:22.435363
Duration: 17.847 ms
Changes:
-----
Summary for ip-172-31-35-161.us-west-2.compute.internal
-----
Succeeded: 1
Failed: 1
-----
Total states run: 2
Total run time: 333.909 ms
ERROR: Minions returned with non-zero exit code
root@ip-172-31-33-58:/srv/salt#
```

- We can get to know whether it was successful or unsuccessful.



- We can test this by checking IP of minion on 80 port in web browser.

<IP of minion>:80

