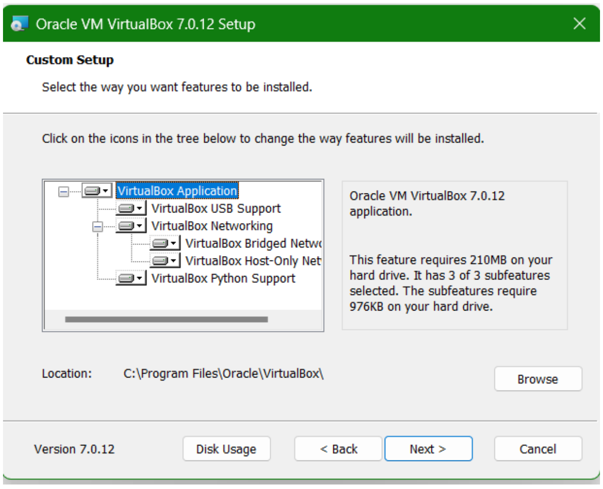
Detailed Documentation **Sheikh Muhammed Tadeeb**

Step1)

For testing my codes, I will install Oracle Virtual Box (Type-2 Hypervisor) on my windows machine:



Oracle VM VirtualBox 7.0.12 Setup

Custom Setup

Select the way you want features to be installed.

Click on the icons in the tree below to change the way features will be installed.

- VirtualBox Application
  - VirtualBox USB Support
  - VirtualBox Networking
    - VirtualBox Bridged Netw
    - VirtualBox Host-Only Net
  - VirtualBox Python Support

Oracle VM VirtualBox 7.0.12 application.

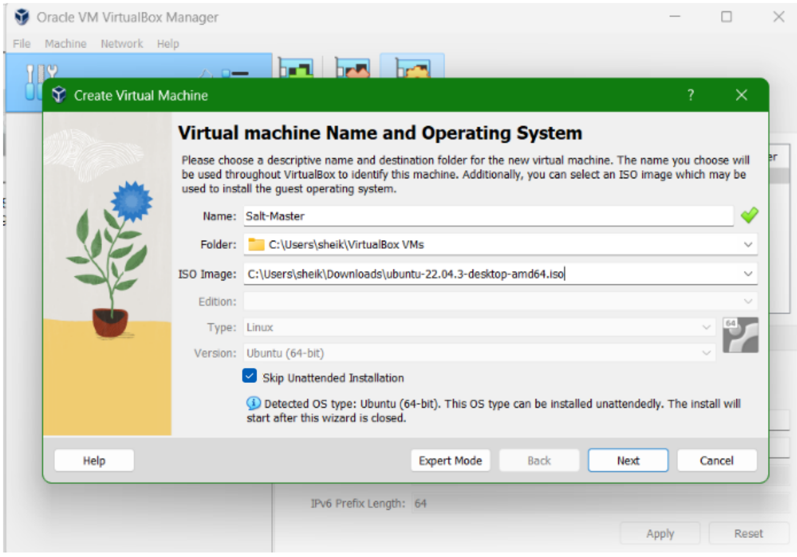
This feature requires 210MB on your hard drive. It has 3 of 3 subfeatures selected. The subfeatures require 976KB on your hard drive.

Location: C:\Program Files\Oracle\VirtualBox\

Version 7.0.12

Step2) Inside this Oracle VirtualBox I will launch:

- ➔ Vm-1 (Salt Master)
- ➔ Vm-2 (Minion-1 with Ubuntu as OS)
- ➔ Vm-3 (Minion-3 with CentOS as OS)



Oracle VM VirtualBox Manager

Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: Salt-Master

Folder: C:\Users\sheik\VirtualBox VMs

ISO Image: C:\Users\sheik\Downloads\ubuntu-22.04.3-desktop-amd64.iso

Edition:

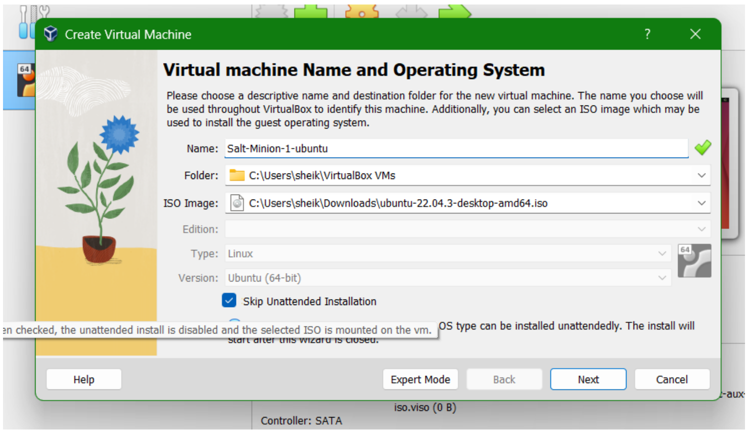
Type: Linux

Version: Ubuntu (64-bit)

☑ Skip Unattended Installation

Detected OS type: Ubuntu (64-bit). This OS type can be installed unattendedly. The install will start after this wizard is closed.

IPv6 Prefix Length: 64



Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: Salt-Minion-1-ubuntu

Folder: C:\Users\sheik\VirtualBox VMs

ISO Image: C:\Users\sheik\Downloads\ubuntu-22.04.3-desktop-amd64.iso

Edition:

Type: Linux

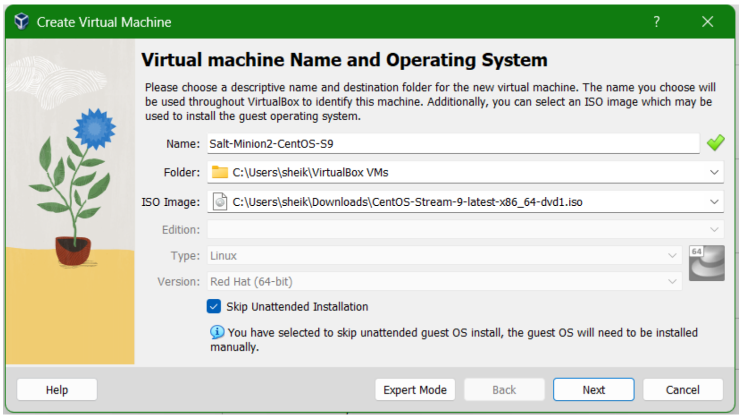
Version: Ubuntu (64-bit)

☑ Skip Unattended Installation

en checked, the unattended install is disabled and the selected ISO is mounted on the vm. OS type can be installed unattendedly. The install will start after this wizard is closed.

iso.viso (0 B)

Controller: SATA



Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: Salt-Minion2-CentOS-S9

Folder: C:\Users\sheik\VirtualBox VMs

ISO Image: C:\Users\sheik\Downloads\CentOS-Stream-9-latest-x86_64-dvd1.iso

Edition:

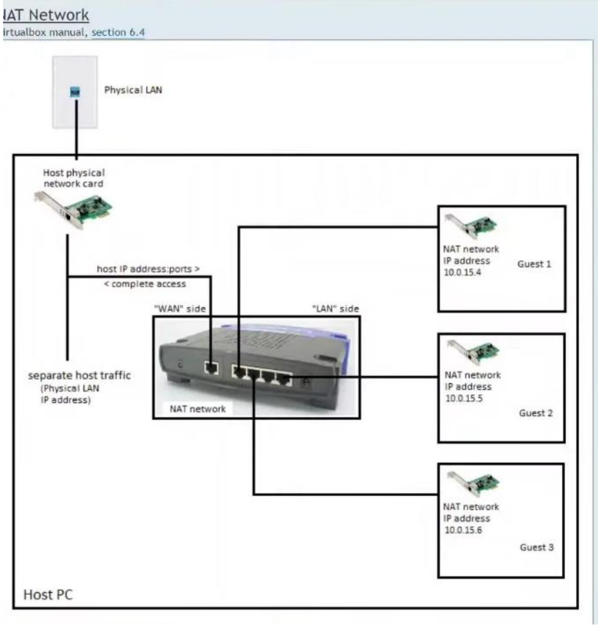
Type: Linux

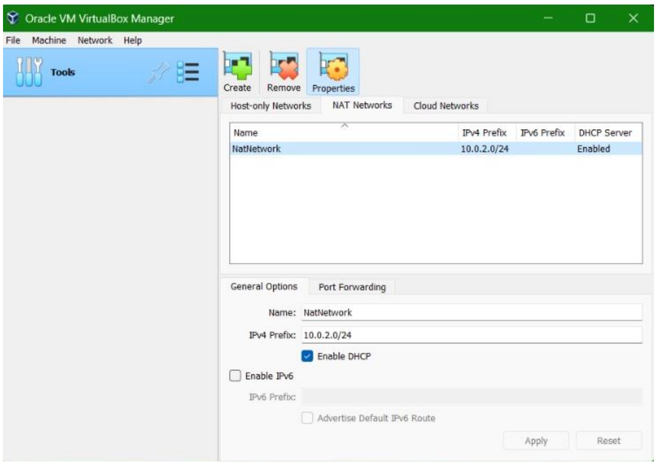
Version: Red Hat (64-bit)

☑ Skip Unattended Installation

You have selected to skip unattended guest OS install, the guest OS will need to be installed manually.

Step3) All these machines will be **NAT network**:

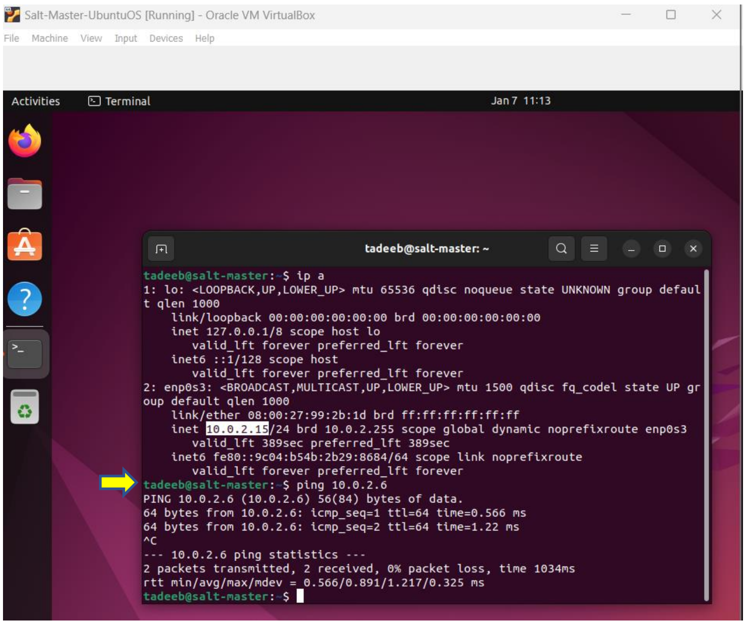

IAT Network
irtualbox manual, section 6.4
Physical LAN
Host physical network card
host IP address:ports >
< complete access
"WAN" side
"LAN" side
separate host traffic (Physical LAN IP address)
NAT network
NAT network IP address 10.0.15.4
Guest 1
NAT network IP address 10.0.15.5
Guest 2
NAT network IP address 10.0.15.6
Guest 3
Host PC



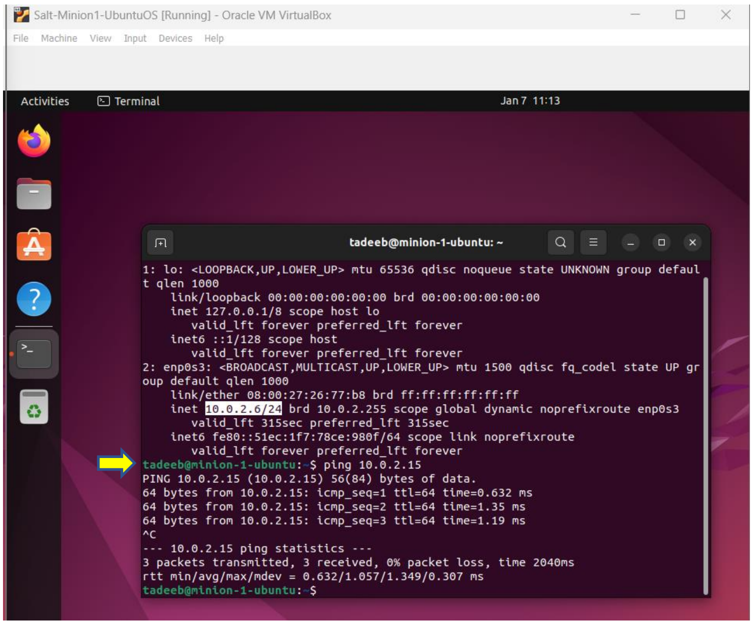
Name	IPv4 Prefix	IPv6 Prefix	DHCP Server
NatNetwork	10.0.2.0/24		Enabled

Name: NatNetwork
IPv4 Prefix: 10.0.2.0/24
☑ Enable DHCP
□ Enable IPv6
IPv6 Prefix:
□ Advertise Default IPv6 Route

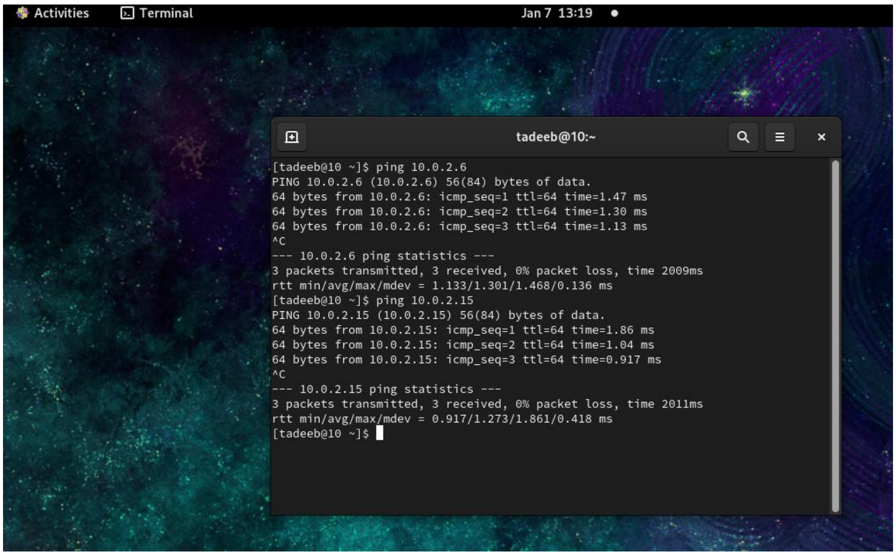
We can see machines are in same LAN.


Salt-Master-UbuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Jan 7 11:13
tadeeb@salt-master: ~

```
tadeeb@salt-master:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:99:2b:1d brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 389sec preferred_lft 389sec
    inet6 fe80::9c04:b54b:2b29:8684/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
tadeeb@salt-master:~$ ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
64 bytes from 10.0.2.6: icmp_seq=1 ttl=64 time=0.566 ms
64 bytes from 10.0.2.6: icmp_seq=2 ttl=64 time=1.22 ms
^C
--- 10.0.2.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1034ms
rtt min/avg/max/mdev = 0.566/0.891/1.217/0.325 ms
tadeeb@salt-master:~$
```

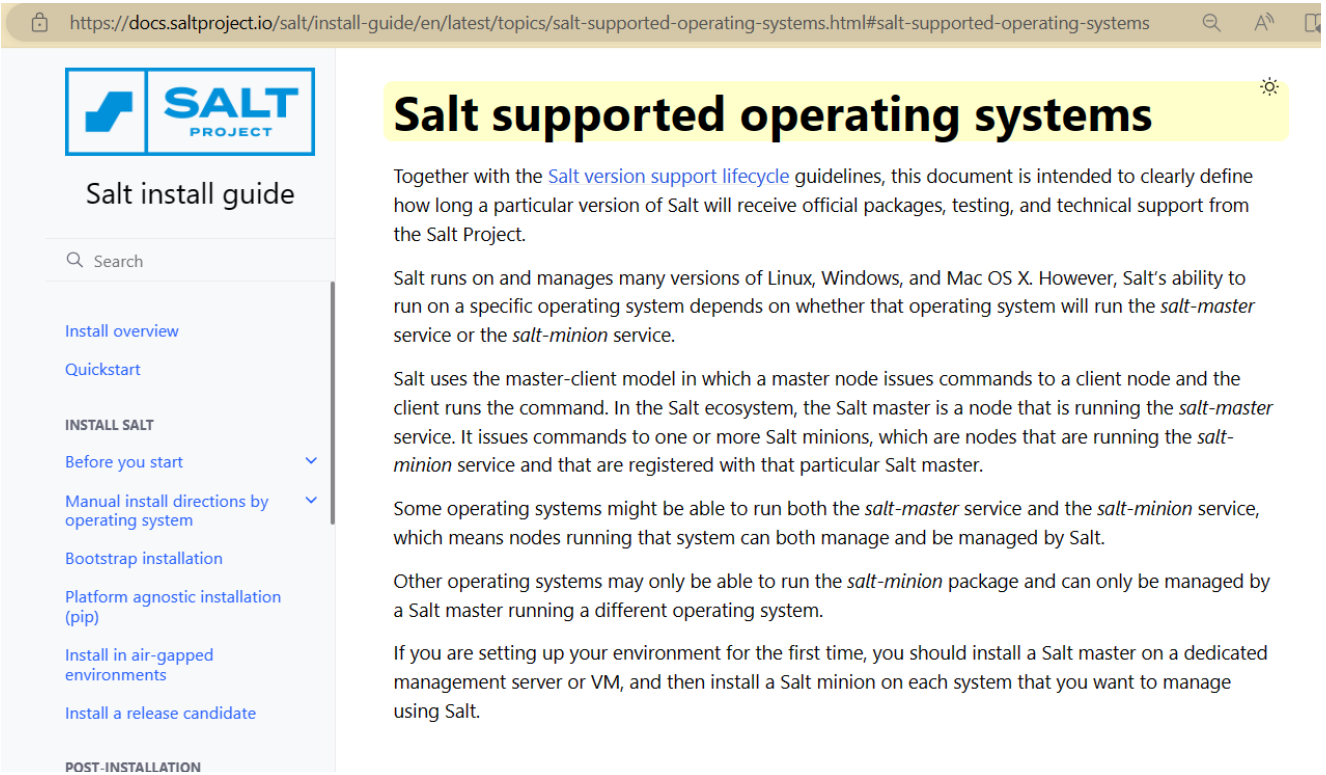

Salt-Minion1-UbuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Jan 7 11:13
tadeeb@minion-1-ubuntu: ~

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
oup default qlen 1000
    link/ether 08:00:27:26:77:b8 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.6/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 315sec preferred_lft 315sec
    inet6 fe80::51ec:1f7:78ce:980f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
tadeeb@minion-1-ubuntu:~$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=0.632 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=1.35 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=1.19 ms
^C
--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.632/1.057/1.349/0.307 ms
tadeeb@minion-1-ubuntu:~$
```



```
[tadeeb@10 ~]$ ping 10.0.2.6
PING 10.0.2.6 (10.0.2.6) 56(84) bytes of data.
64 bytes from 10.0.2.6: icmp_seq=1 ttl=64 time=1.47 ms
64 bytes from 10.0.2.6: icmp_seq=2 ttl=64 time=1.30 ms
64 bytes from 10.0.2.6: icmp_seq=3 ttl=64 time=1.13 ms
^C
--- 10.0.2.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2009ms
rtt min/avg/max/mdev = 1.133/1.301/1.468/0.136 ms
[tadeeb@10 ~]$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
64 bytes from 10.0.2.15: icmp_seq=1 ttl=64 time=1.86 ms
64 bytes from 10.0.2.15: icmp_seq=2 ttl=64 time=1.04 ms
64 bytes from 10.0.2.15: icmp_seq=3 ttl=64 time=0.917 ms
^C
--- 10.0.2.15 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 0.917/1.273/1.861/0.418 ms
[tadeeb@10 ~]$
```

Step4) Before installing Salt. I just checked pre-requisites.


https://docs.saltproject.io/salt/install-guide/en/latest/topics/salt-supported-operating-systems.html#salt-supported-operating-systems
SALT PROJECT
Salt install guide
Search
Install overview
Quickstart
INSTALL SALT
Before you start
Manual install directions by operating system
Bootstrap installation
Platform agnostic installation (pip)
Install in air-gapped environments
Install a release candidate
POST-INSTALLATION

Salt supported operating systems

Together with the Salt version support lifecycle guidelines, this document is intended to clearly define how long a particular version of Salt will receive official packages, testing, and technical support from the Salt Project.

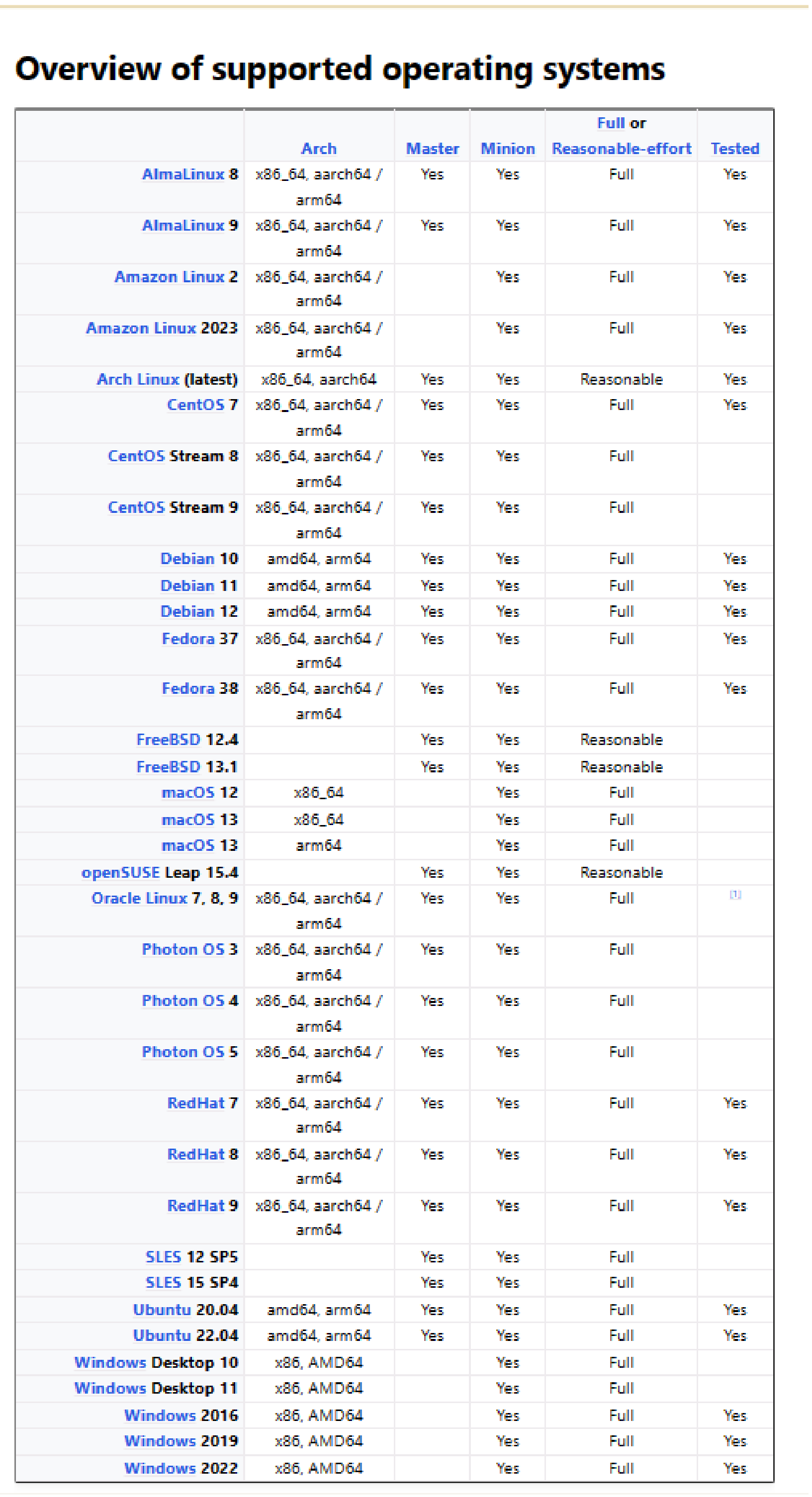
Salt runs on and manages many versions of Linux, Windows, and Mac OS X. However, Salt's ability to run on a specific operating system depends on whether that operating system will run the *salt-master* service or the *salt-minion* service.

Salt uses the master-client model in which a master node issues commands to a client node and the client runs the command. In the Salt ecosystem, the Salt master is a node that is running the *salt-master* service. It issues commands to one or more Salt minions, which are nodes that are running the *salt-minion* service and that are registered with that particular Salt master.

Some operating systems might be able to run both the *salt-master* service and the *salt-minion* service, which means nodes running that system can both manage and be managed by Salt.

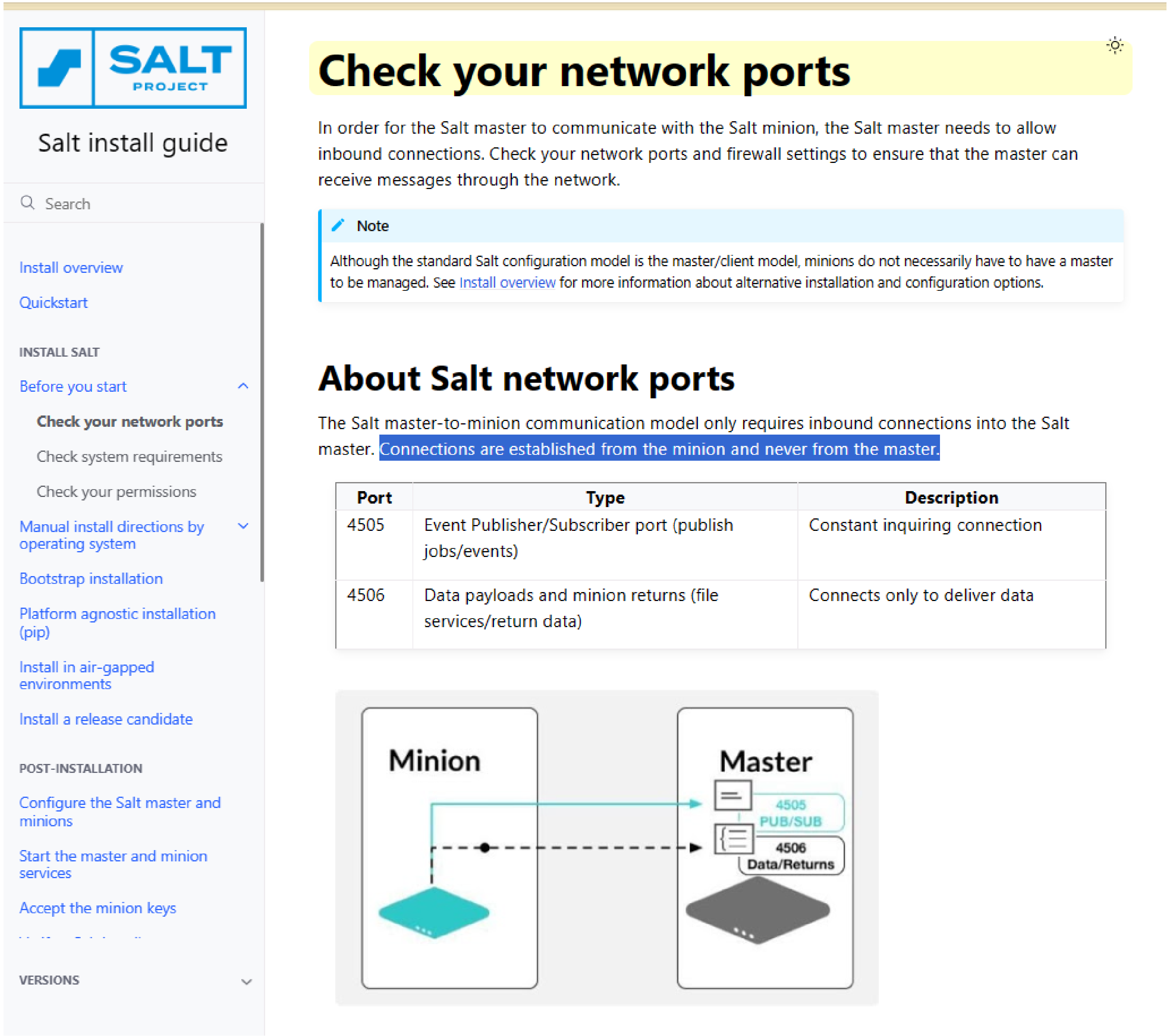
Other operating systems may only be able to run the *salt-minion* package and can only be managed by a Salt master running a different operating system.

If you are setting up your environment for the first time, you should install a Salt master on a dedicated management server or VM, and then install a Salt minion on each system that you want to manage using Salt.



Overview of supported operating systems

	Arch	Master	Minion	Full or Reasonable-effort	Tested
AlmaLinux 8	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
AlmaLinux 9	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
Amazon Linux 2	x86_64, aarch64 / arm64		Yes	Full	Yes
Amazon Linux 2023	x86_64, aarch64 / arm64		Yes	Full	Yes
Arch Linux (latest)	x86_64, aarch64	Yes	Yes	Reasonable	Yes
CentOS 7	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
CentOS Stream 8	x86_64, aarch64 / arm64	Yes	Yes	Full	
CentOS Stream 9	x86_64, aarch64 / arm64	Yes	Yes	Full	
Debian 10	amd64, arm64	Yes	Yes	Full	Yes
Debian 11	amd64, arm64	Yes	Yes	Full	Yes
Debian 12	amd64, arm64	Yes	Yes	Full	Yes
Fedora 37	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
Fedora 38	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
FreeBSD 12.4		Yes	Yes	Reasonable	
FreeBSD 13.1		Yes	Yes	Reasonable	
macOS 12	x86_64		Yes	Full	
macOS 13	x86_64		Yes	Full	
macOS 13	arm64		Yes	Full	
openSUSE Leap 15.4		Yes	Yes	Reasonable	
Oracle Linux 7, 8, 9	x86_64, aarch64 / arm64	Yes	Yes	Full	[1]
Photon OS 3	x86_64, aarch64 / arm64	Yes	Yes	Full	
Photon OS 4	x86_64, aarch64 / arm64	Yes	Yes	Full	
Photon OS 5	x86_64, aarch64 / arm64	Yes	Yes	Full	
RedHat 7	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
RedHat 8	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
RedHat 9	x86_64, aarch64 / arm64	Yes	Yes	Full	Yes
SLES 12 SP5		Yes	Yes	Full	
SLES 15 SP4		Yes	Yes	Full	
Ubuntu 20.04	amd64, arm64	Yes	Yes	Full	Yes
Ubuntu 22.04	amd64, arm64	Yes	Yes	Full	Yes
Windows Desktop 10	x86, AMD64		Yes	Full	
Windows Desktop 11	x86, AMD64		Yes	Full	
Windows 2016	x86, AMD64		Yes	Full	Yes
Windows 2019	x86, AMD64		Yes	Full	Yes
Windows 2022	x86, AMD64		Yes	Full	Yes


SALT PROJECT
Salt install guide
Search
Install overview
Quickstart
INSTALL SALT
Before you start
Check your network ports
Check system requirements
Check your permissions
Manual install directions by operating system
Bootstrap installation
Platform agnostic installation (pip)
Install in air-gapped environments
Install a release candidate
POST-INSTALLATION
Configure the Salt master and minions
Start the master and minion services
Accept the minion keys
VERSIONS

Check your network ports

In order for the Salt master to communicate with the Salt minion, the Salt master needs to allow inbound connections. Check your network ports and firewall settings to ensure that the master can receive messages through the network.

> **Note**
>
> Although the standard Salt configuration model is the master/client model, minions do not necessarily have to have a master to be managed. See Install overview for more information about alternative installation and configuration options.

About Salt network ports

The Salt master-to-minion communication model only requires inbound connections into the Salt master. Connections are established from the minion and never from the master.

Port	Type	Description
4505	Event Publisher/Subscriber port (publish jobs/events)	Constant inquiring connection
4506	Data payloads and minion returns (file services/return data)	Connects only to deliver data

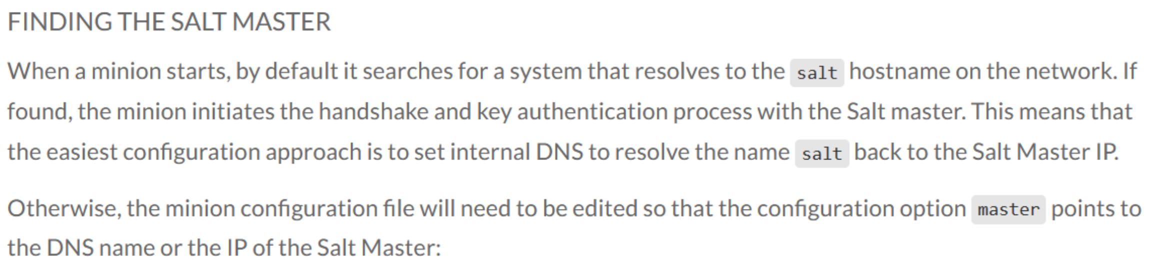
Step5) Install Salt on Master and minions:

I used these 2 links below from official Salt website only:

Salt install guide (saltproject.io)

Bootstrap installation - Salt install guide (saltproject.io)

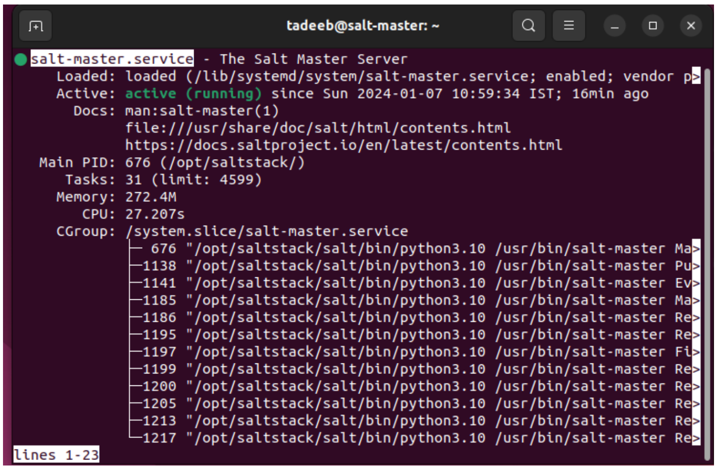
Step6) Just start Master and minion services and do necessary changes in minion machines /etc/hosts file so that they can find Master.



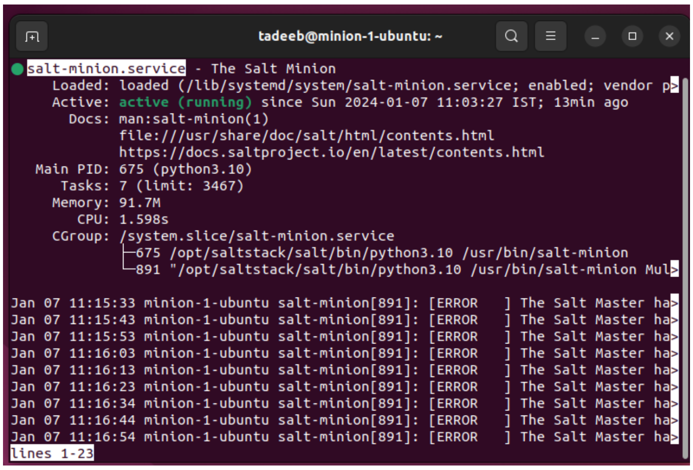
FINDING THE SALT MASTER

When a minion starts, by default it searches for a system that resolves to the `salt` hostname on the network. If found, the minion initiates the handshake and key authentication process with the Salt master. This means that the easiest configuration approach is to set internal DNS to resolve the name `salt` back to the Salt Master IP.

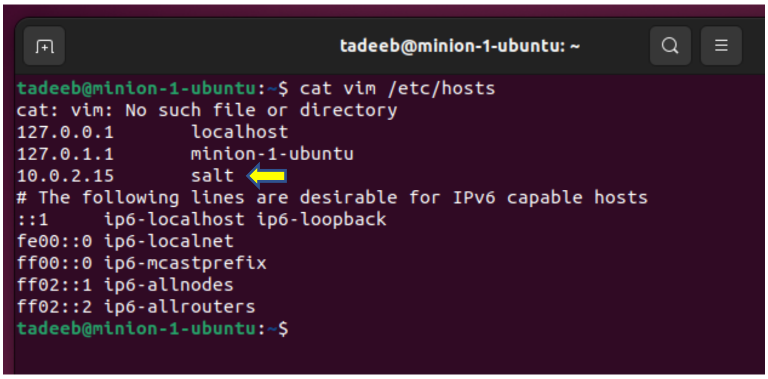
Otherwise, the minion configuration file will need to be edited so that the configuration option `master` points to the DNS name or the IP of the Salt Master:



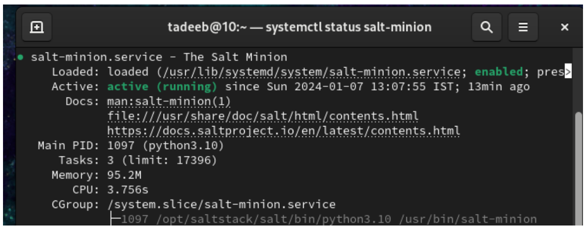
```
● salt-master.service - The Salt Master Server
     Loaded: loaded (/lib/systemd/system/salt-master.service; enabled; vendor p>
     Active: active (running) since Sun 2024-01-07 10:59:34 IST; 16min ago
       Docs: man:salt-master(1)
             file:///usr/share/doc/salt/html/contents.html
             https://docs.saltproject.io/en/latest/contents.html
   Main PID: 676 (/opt/saltstack/)
      Tasks: 31 (limit: 4599)
     Memory: 272.4M
        CPU: 27.207s
     CGroup: /system.slice/salt-master.service
             ├─ 676 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Ma>
             ├─1138 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Pu>
             ├─1141 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Ev>
             ├─1185 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Ma>
             ├─1186 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
             ├─1195 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
             ├─1197 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Fi>
             ├─1199 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
             ├─1200 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
             ├─1205 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
             ├─1213 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
             └─1217 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-master Re>
lines 1-23
```



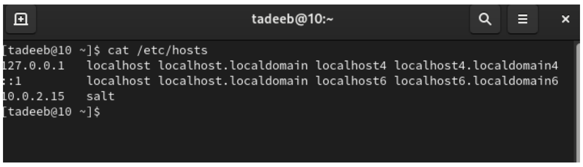
```
● salt-minion.service - The Salt Minion
     Loaded: loaded (/lib/systemd/system/salt-minion.service; enabled; vendor p>
     Active: active (running) since Sun 2024-01-07 11:03:27 IST; 13min ago
       Docs: man:salt-minion(1)
             file:///usr/share/doc/salt/html/contents.html
             https://docs.saltproject.io/en/latest/contents.html
   Main PID: 675 (python3.10)
      Tasks: 7 (limit: 3467)
     Memory: 91.7M
        CPU: 1.598s
     CGroup: /system.slice/salt-minion.service
             ├─675 /opt/saltstack/salt/bin/python3.10 /usr/bin/salt-minion
             └─891 "/opt/saltstack/salt/bin/python3.10 /usr/bin/salt-minion Mul>

Jan 07 11:15:33 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:15:43 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:15:53 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:16:03 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:16:13 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:16:23 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:16:34 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:16:44 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
Jan 07 11:16:54 minion-1-ubuntu salt-minion[891]: [ERROR   ] The Salt Master ha>
lines 1-23
```



```
tadeeb@minion-1-ubuntu:~$ cat vim /etc/hosts
cat: vim: No such file or directory
127.0.0.1       localhost
127.0.1.1       minion-1-ubuntu
10.0.2.15       salt
# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
tadeeb@minion-1-ubuntu:~$
```

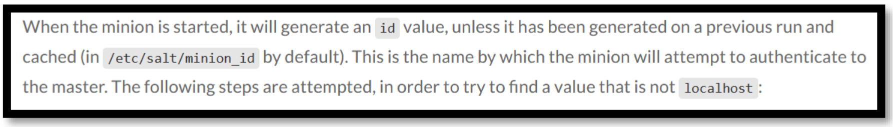


```
● salt-minion.service - The Salt Minion
     Loaded: loaded (/usr/lib/systemd/system/salt-minion.service; enabled; pres>
     Active: active (running) since Sun 2024-01-07 13:07:55 IST; 13min ago
       Docs: man:salt-minion(1)
             file:///usr/share/doc/salt/html/contents.html
             https://docs.saltproject.io/en/latest/contents.html
   Main PID: 1097 (python3.10)
      Tasks: 3 (limit: 17396)
     Memory: 95.2M
        CPU: 3.756s
     CGroup: /system.slice/salt-minion.service
             ├─1097 /opt/saltstack/salt/bin/python3.10 /usr/bin/salt-minion
```

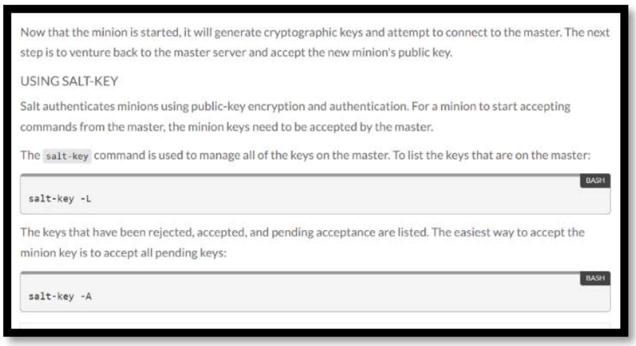


```
[tadeeb@10 ~]$ cat /etc/hosts
127.0.0.1   localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.15   salt
[tadeeb@10 ~]$
```

Step7) Accepting Minions keys from master



When the minion is started, it will generate an `id` value, unless it has been generated on a previous run and cached (in `/etc/salt/minion_id` by default). This is the name by which the minion will attempt to authenticate to the master. The following steps are attempted, in order to try to find a value that is not `localhost`:



Now that the minion is started, it will generate cryptographic keys and attempt to connect to the master. The next step is to venture back to the master server and accept the new minion's public key.

USING SALT-KEY

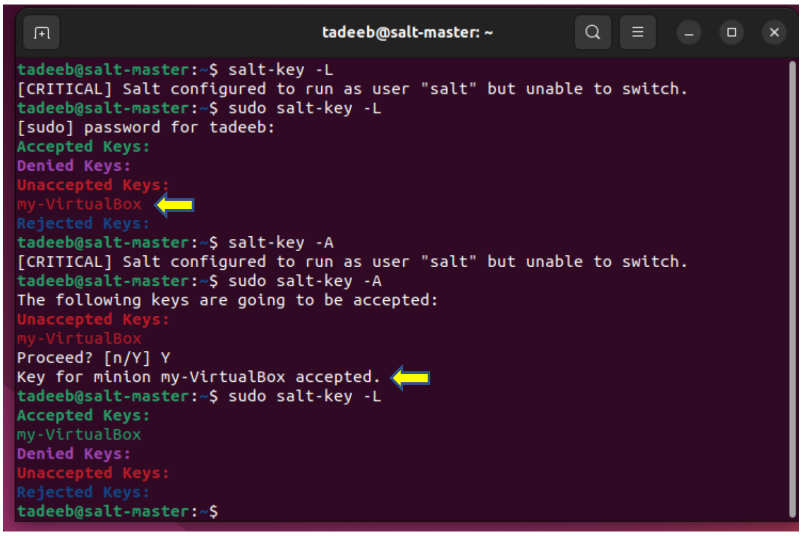
Salt authenticates minions using public-key encryption and authentication. For a minion to start accepting commands from the master, the minion keys need to be accepted by the master.

The `salt-key` command is used to manage all of the keys on the master. To list the keys that are on the master:

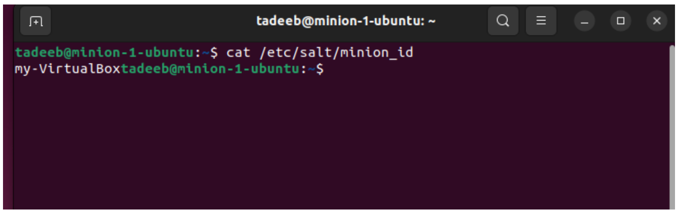
```
salt-key -L
```

The keys that have been rejected, accepted, and pending acceptance are listed. The easiest way to accept the minion key is to accept all pending keys:

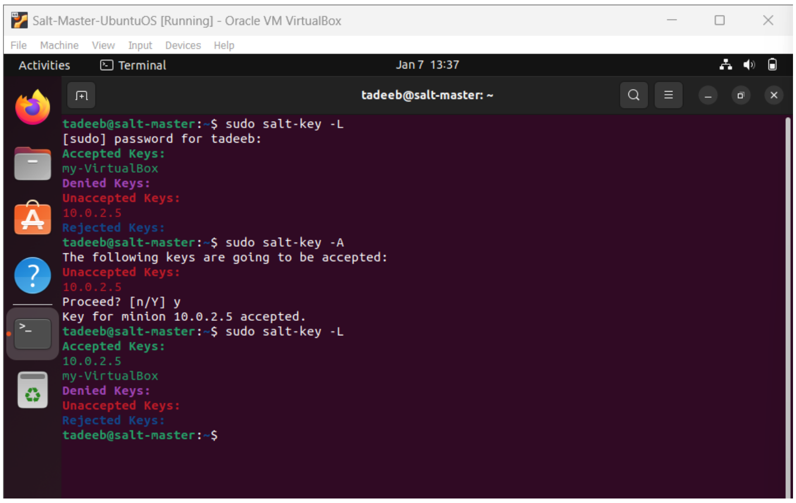
```
salt-key -A
```



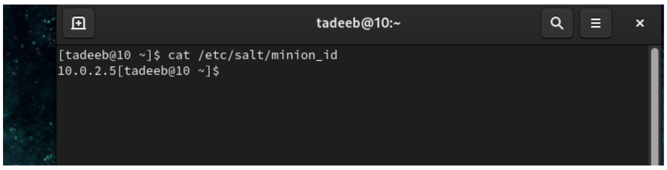
```
tadeeb@salt-master:~$ salt-key -L
[CRITICAL] Salt configured to run as user "salt" but unable to switch.
tadeeb@salt-master:~$ sudo salt-key -L
[sudo] password for tadeeb:
Accepted Keys:
Denied Keys:
Unaccepted Keys:
my-VirtualBox
Rejected Keys:
tadeeb@salt-master:~$ salt-key -A
[CRITICAL] Salt configured to run as user "salt" but unable to switch.
tadeeb@salt-master:~$ sudo salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
my-VirtualBox
Proceed? [n/Y] Y
Key for minion my-VirtualBox accepted.
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
my-VirtualBox
Denied Keys:
Unaccepted Keys:
Rejected Keys:
tadeeb@salt-master:~$
```



```
tadeeb@minion-1-ubuntu:~$ cat /etc/salt/minion_id
my-VirtualBoxtadeeb@minion-1-ubuntu:~$
```

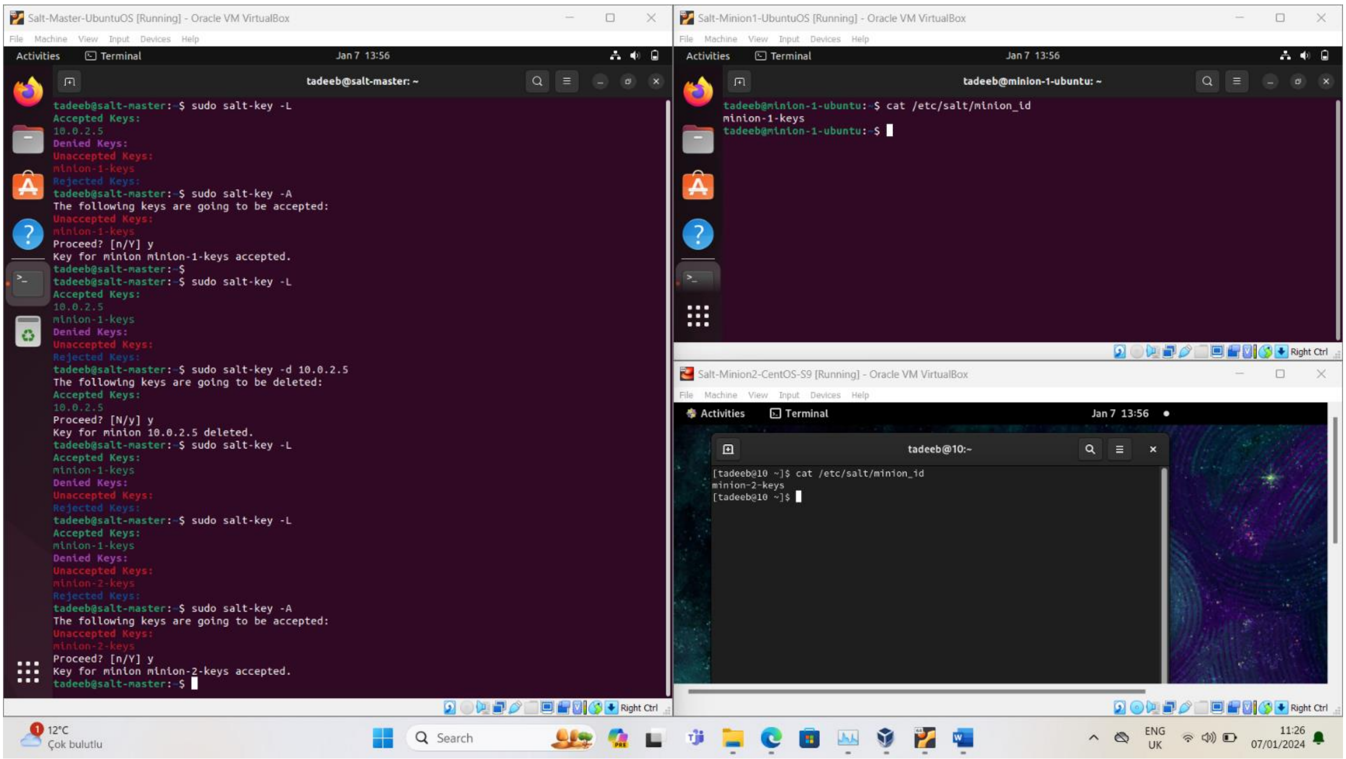


```
tadeeb@salt-master:~$ sudo salt-key -L
[sudo] password for tadeeb:
Accepted Keys:
my-VirtualBox
Denied Keys:
Unaccepted Keys:
10.0.2.5
Rejected Keys:
tadeeb@salt-master:~$ sudo salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
10.0.2.5
Proceed? [n/Y] y
Key for minion 10.0.2.5 accepted.
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
10.0.2.5
my-VirtualBox
Denied Keys:
Unaccepted Keys:
Rejected Keys:
tadeeb@salt-master:~$
```



```
[tadeeb@10 ~]$ cat /etc/salt/minion_id
10.0.2.5[tadeeb@10 ~]$
```

I changed the keys for my convenience:



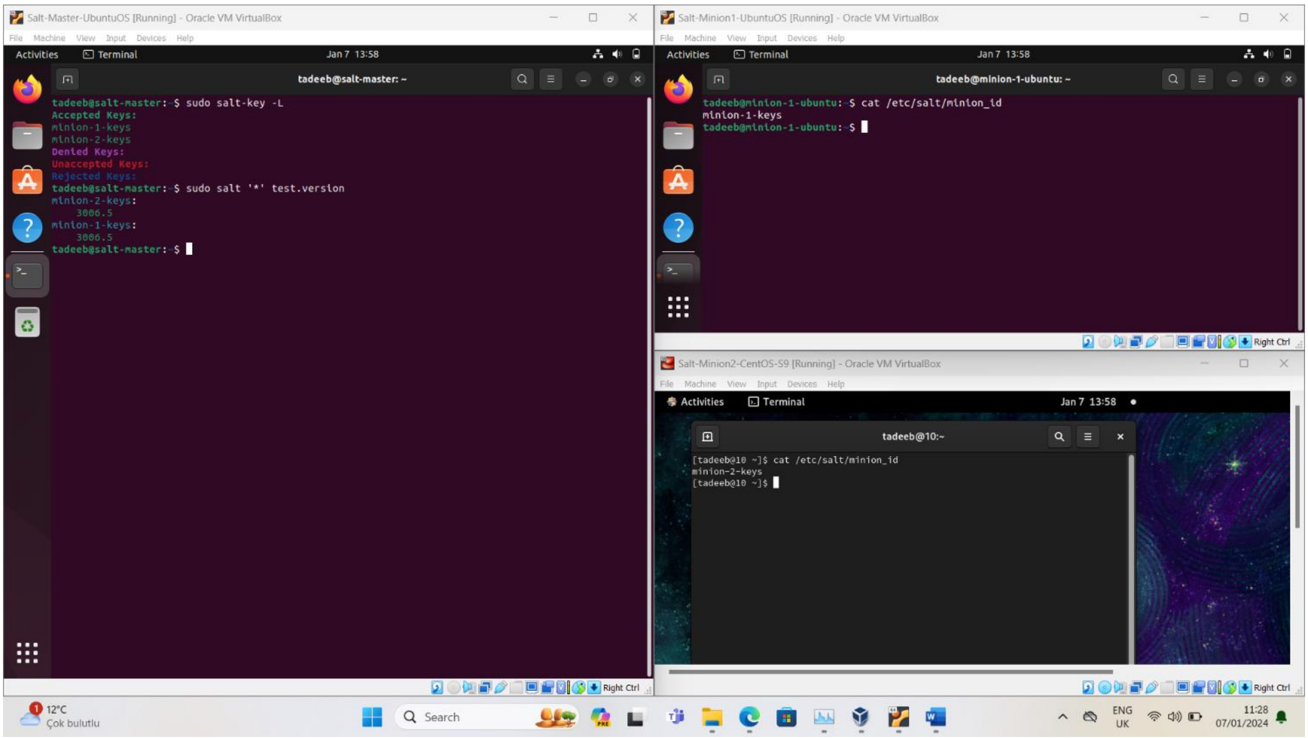
```
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
10.0.2.5
Denied Keys:
Unaccepted Keys:
minion-1-keys
Rejected Keys:
tadeeb@salt-master:~$ sudo salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
minion-1-keys
Proceed? [n/Y] y
Key for minion minion-1-keys accepted.
tadeeb@salt-master:~$
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
10.0.2.5
minion-1-keys
Denied Keys:
Unaccepted Keys:
Rejected Keys:
tadeeb@salt-master:~$ sudo salt-key -d 10.0.2.5
The following keys are going to be deleted:
Accepted Keys:
10.0.2.5
Proceed? [N/y] y
Key for minion 10.0.2.5 deleted.
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
minion-1-keys
Denied Keys:
Unaccepted Keys:
Rejected Keys:
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
minion-1-keys
Denied Keys:
Unaccepted Keys:
minion-2-keys
Rejected Keys:
tadeeb@salt-master:~$ sudo salt-key -A
The following keys are going to be accepted:
Unaccepted Keys:
minion-2-keys
Proceed? [n/Y] y
Key for minion minion-2-keys accepted.
tadeeb@salt-master:~$
```

Salt-Minion1-UbuntuOS [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
Jan 7 13:56
tadeeb@minion-1-ubuntu: ~
Right Ctrl

```
tadeeb@minion-1-ubuntu:~$ cat /etc/salt/minion_id
minion-1-keys
tadeeb@minion-1-ubuntu:~$
```

```
[tadeeb@10 ~]$ cat /etc/salt/minion_id
minion-2-keys
[tadeeb@10 ~]$
```

Step8) Checking whether our Minions are connected with master or not:

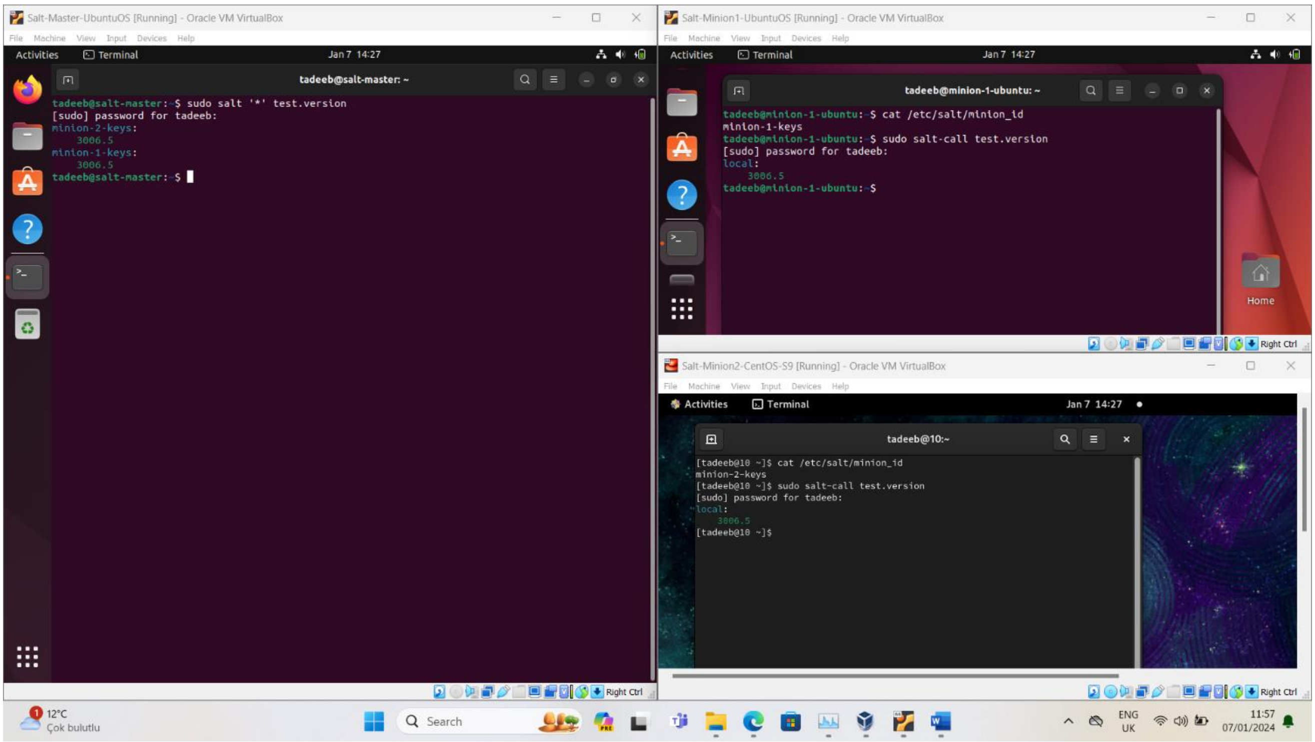

Salt-Master-UbuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jan 7 13:58
tadeeb@salt-master: ~
Salt-Minion1-UbuntuOS [Running] - Oracle VM VirtualBox
tadeeb@minion-1-ubuntu: ~
Salt-Minion2-CentOS-S9 [Running] - Oracle VM VirtualBox
tadeeb@10:~
Right Ctrl
12°C
Çok bulutlu
Search
ENG
UK
11:28
07/01/2024

```
tadeeb@salt-master:~$ sudo salt-key -L
Accepted Keys:
minion-1-keys
minion-2-keys
Denied Keys:
Unaccepted Keys:
Rejected Keys:
tadeeb@salt-master:~$ sudo salt '*' test.version
minion-2-keys:
    3006.5
minion-1-keys:
    3006.5
tadeeb@salt-master:~$
```

```
tadeeb@minion-1-ubuntu:~$ cat /etc/salt/minion_id
minion-1-keys
tadeeb@minion-1-ubuntu:~$
```

```
[tadeeb@10 ~]$ cat /etc/salt/minion_id
minion-2-keys
[tadeeb@10 ~]$
```

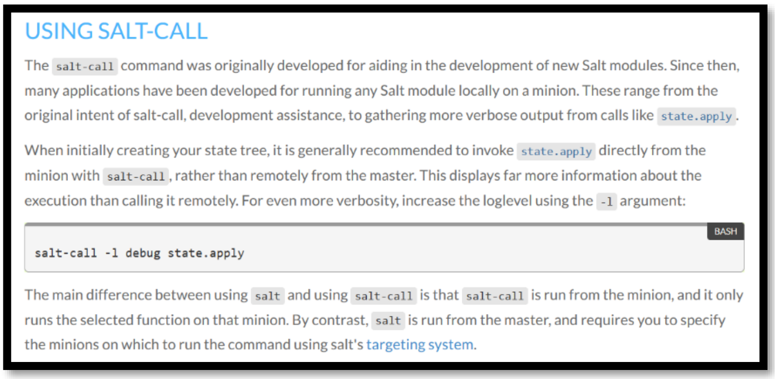
Cross confirm from minions:


Salt-Master-UbuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal Jan 7 14:27
tadeeb@salt-master: ~
Salt-Minion1-UbuntuOS [Running] - Oracle VM VirtualBox
tadeeb@minion-1-ubuntu: ~
Home
Salt-Minion2-CentOS-S9 [Running] - Oracle VM VirtualBox
tadeeb@10:~
Right Ctrl
12°C
Çok bulutlu
Search
ENG
UK
11:57
07/01/2024

```
tadeeb@salt-master:~$ sudo salt '*' test.version
[sudo] password for tadeeb:
minion-2-keys:
    3006.5
minion-1-keys:
    3006.5
tadeeb@salt-master:~$
```

```
tadeeb@minion-1-ubuntu:~$ cat /etc/salt/minion_id
minion-1-keys
tadeeb@minion-1-ubuntu:~$ sudo salt-call test.version
[sudo] password for tadeeb:
local:
    3006.5
tadeeb@minion-1-ubuntu:~$
```

```
[tadeeb@10 ~]$ cat /etc/salt/minion_id
minion-2-keys
[tadeeb@10 ~]$ sudo salt-call test.version
[sudo] password for tadeeb:
local:
    3006.5
[tadeeb@10 ~]$
```



USING SALT-CALL

The `salt-call` command was originally developed for aiding in the development of new Salt modules. Since then, many applications have been developed for running any Salt module locally on a minion. These range from the original intent of salt-call, development assistance, to gathering more verbose output from calls like `state.apply`.

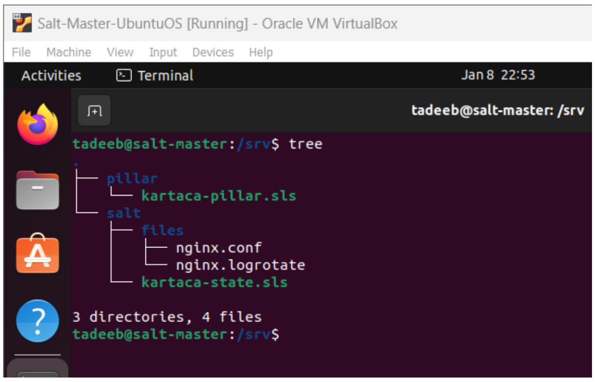
When initially creating your state tree, it is generally recommended to invoke `state.apply` directly from the minion with `salt-call`, rather than remotely from the master. This displays far more information about the execution than calling it remotely. For even more verbosity, increase the loglevel using the `-l` argument:

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

The main difference between using `salt` and using `salt-call` is that `salt-call` is run from the minion, and it only runs the selected function on that minion. By contrast, `salt` is run from the master, and requires you to specify the minions on which to run the command using salt's targeting system.

Step9) Performing the given task:

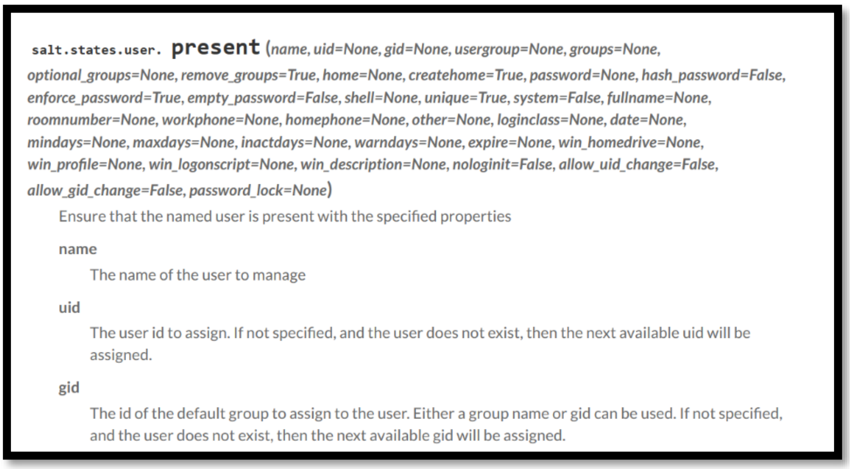
Directory Structure:


Salt-Master-UbuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Jan 8 22:53
tadeeb@salt-master: /srv

```
tadeeb@salt-master:/srv$ tree
.
├── pillar
│   └── kartaca-pillar.sls
└── salt
    ├── files
    │   ├── nginx.conf
    │   └── nginx.logrotate
    └── kartaca-state.sls

3 directories, 4 files
tadeeb@salt-master:/srv$
```

On both servers:

- Create a user named kartaca with user and group ID 2023, home directory, default shell, password carthage2023. (Keep the user password information on the pillar data, not the state file.)`/home/krt/bin/bash`



`salt.states.user.` **`present`** *(name, uid=None, gid=None, usergroup=None, groups=None, optional_groups=None, remove_groups=True, home=None, createhome=True, password=None, hash_password=False, enforce_password=True, empty_password=False, shell=None, unique=True, system=False, fullname=None, roomnumber=None, workphone=None, homephone=None, other=None, loginclass=None, date=None, mindays=None, maxdays=None, inactdays=None, warndays=None, expire=None, win_homedrive=None, win_profile=None, win_logonscript=None, win_description=None, nologinit=False, allow_uid_change=False, allow_gid_change=False, password_lock=None)*

Ensure that the named user is present with the specified properties

name

The name of the user to manage

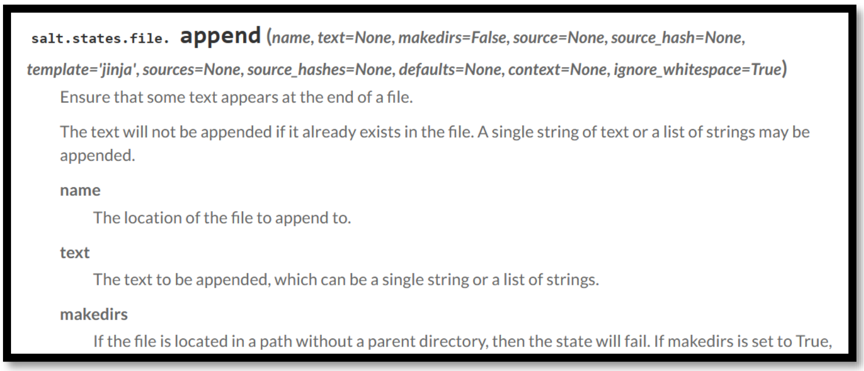
uid

The user id to assign. If not specified, and the user does not exist, then the next available uid will be assigned.

gid

The id of the default group to assign to the user. Either a group name or gid can be used. If not specified, and the user does not exist, then the next available gid will be assigned.

- Give sudo privileges to the Carthage user and this user can run his command on Ubuntu and his command on Centos without entering a password.`sudo aptsudo yum`



`salt.states.file.` **`append`** *(name, text=None, makedirs=False, source=None, source_hash=None, template='jinja', sources=None, source_hashes=None, defaults=None, context=None, ignore_whitespace=True)*

Ensure that some text appears at the end of a file.

The text will not be appended if it already exists in the file. A single string of text or a list of strings may be appended.

name

The location of the file to append to.

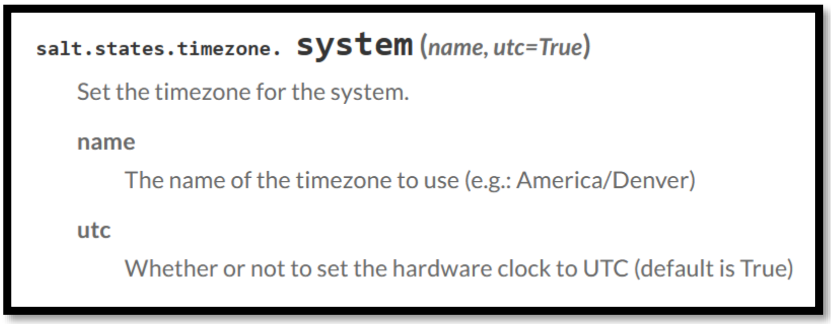
text

The text to be appended, which can be a single string or a list of strings.

makedirs

If the file is located in a path without a parent directory, then the state will fail. If makedirs is set to True,

- Set the server timezone to Istanbul.



`salt.states.timezone.` **`system`** *(name, utc=True)*

Set the timezone for the system.

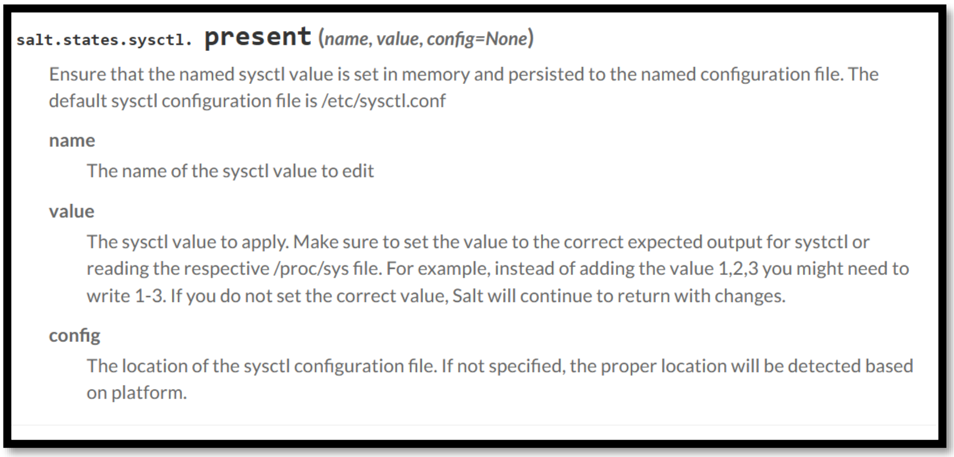
name

The name of the timezone to use (e.g.: America/Denver)

utc

Whether or not to set the hardware clock to UTC (default is True)

- Enable IP Forwarding permanently.



`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

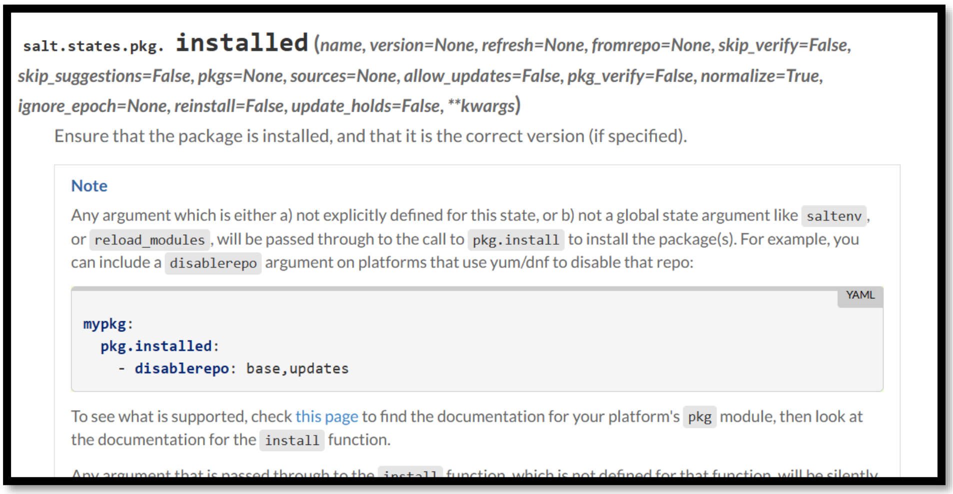
value

The sysctl value to apply. Make sure to set the value to the correct expected output for systctl or reading the respective /proc/sys file. For example, instead of adding the value 1,2,3 you might need to write 1-3. If you do not set the correct value, Salt will continue to return with changes.

config

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

- Install the necessary packages to be able to run the , commands from the terminal.`htop tcp traceroute ping dig iostat mtr`



`salt.states.pkg.` **`installed`** (*name, version=None, refresh=None, fromrepo=None, skip_verify=False, skip_suggestions=False, pkgs=None, sources=None, allow_updates=False, pkg_verify=False, normalize=True, ignore_epoch=None, reinstall=False, update_holds=False, **kwargs*)

Ensure that the package is installed, and that it is the correct version (if specified).

Note

Any argument which is either a) not explicitly defined for this state, or b) not a global state argument like `saltenv`, or `reload_modules`, will be passed through to the call to `pkg.install` to install the package(s). For example, you can include a `disablerepo` argument on platforms that use yum/dnf to disable that repo:

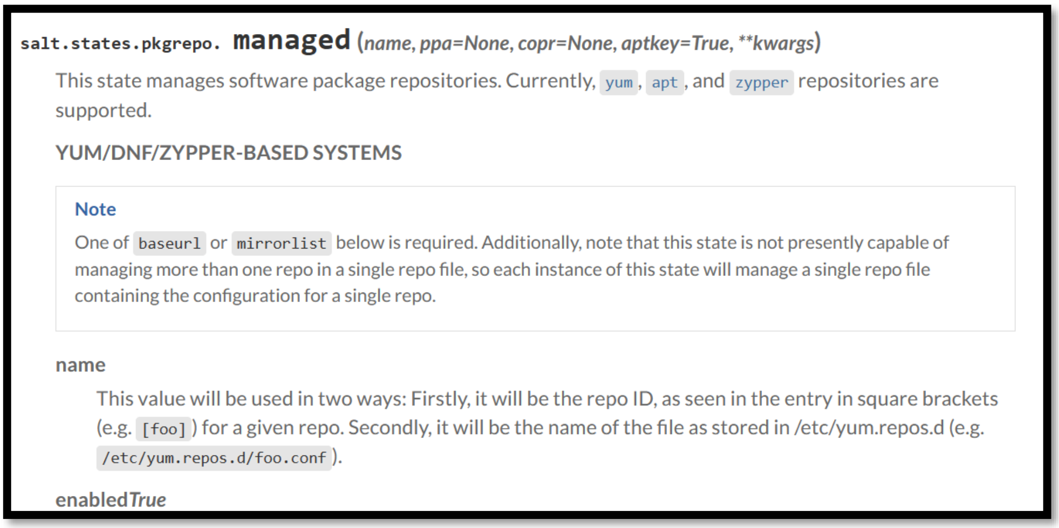
YAML

```
mypkg:
  pkg.installed:
    - disablerepo: base,updates
```

To see what is supported, check this page to find the documentation for your platform's `pkg` module, then look at the documentation for the `install` function.

Any argument that is passed through to the `install` function, which is not defined for that function, will be silently

- [Add](#) the Hashicorp repo to the system with the information at https://www.hashicorp.com/official-packaging-guide and install the v1.6.4 version of the Terraform package.



`salt.states.pkgrepo.` **`managed`** (*name, ppa=None, copr=None, aptkey=True, **kwargs*)

This state manages software package repositories. Currently, `yum`, `apt`, and `zypper` repositories are supported.

YUM/DNF/ZYPPER-BASED SYSTEMS

Note

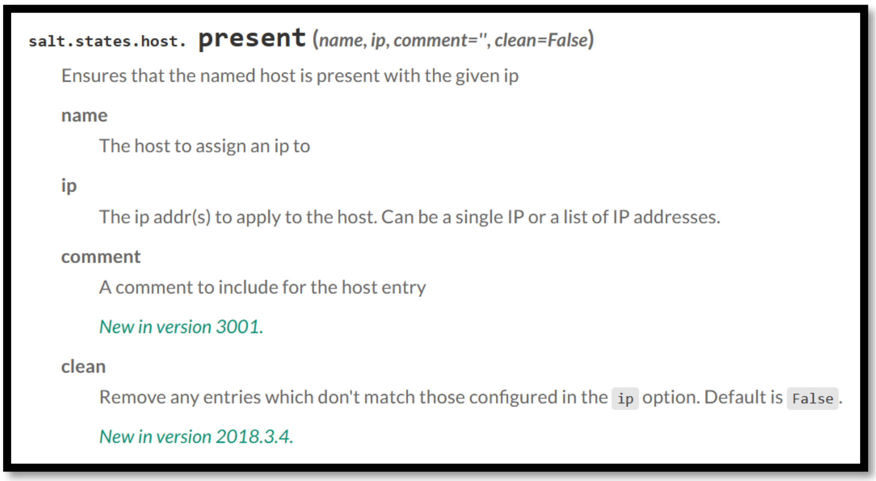
One of `baseurl` or `mirrorlist` below is required. Additionally, note that this state is not presently capable of managing more than one repo in a single repo file, so each instance of this state will manage a single repo file containing the configuration for a single repo.

name

This value will be used in two ways: Firstly, it will be the repo ID, as seen in the entry in square brackets (e.g. `[foo]`) for a given repo. Secondly, it will be the name of the file as stored in /etc/yum.repos.d (e.g. `/etc/yum.repos.d/foo.conf`).

enabled*True*

- For each IP address in the 192.168.168.128/28 IP block, add the host record to the file to resolve the kartaca.local address. Make this change with the *for* loop in the Salt state file.`/etc/hosts`



`salt.states.host.` **`present`** *(name, ip, comment='', clean=False)*

Ensures that the named host is present with the given ip

name
The host to assign an ip to

ip
The ip addr(s) to apply to the host. Can be a single IP or a list of IP addresses.

comment
A comment to include for the host entry

New in version 3001.

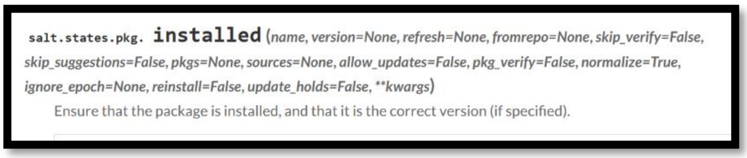
clean
Remove any entries which don't match those configured in the `ip` option. Default is `False`.

New in version 2018.3.4.

Step10)

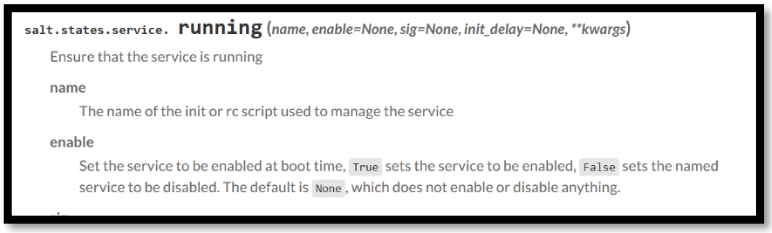
On the Centos server:

- Install Nginx web server.
- Configure the Nginx service to start automatically every time the server restarts.



`salt.states.pkg.` **`installed`** *(name, version=None, refresh=None, fromrepo=None, skip_verify=False, skip_suggestions=False, pkgs=None, sources=None, allow_updates=False, pkg_verify=False, normalize=True, ignore_epoch=None, reinstall=False, update_holds=False, **kwargs)*

Ensure that the package is installed, and that it is the correct version (if specified).



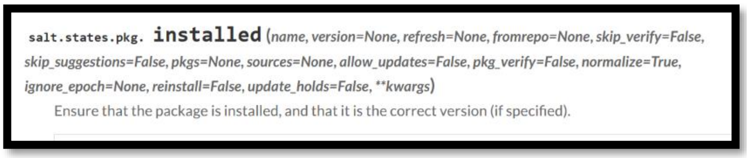
`salt.states.service.` **`running`** *(name, enable=None, sig=None, init_delay=None, **kwargs)*

Ensure that the service is running

name
The name of the init or rc script used to manage the service

enable
Set the service to be enabled at boot time, `True` sets the service to be enabled, `False` sets the named service to be disabled. The default is `None`, which does not enable or disable anything.

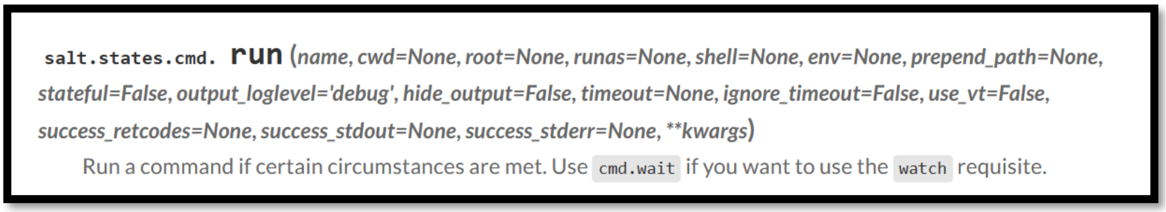
- Install the necessary PHP packages to run WordPress on the server, make the necessary Nginx/PHP configurations.



`salt.states.pkg.` **`installed`** *(name, version=None, refresh=None, fromrepo=None, skip_verify=False, skip_suggestions=False, pkgs=None, sources=None, allow_updates=False, pkg_verify=False, normalize=True, ignore_epoch=None, reinstall=False, update_holds=False, **kwargs)*

Ensure that the package is installed, and that it is the correct version (if specified).

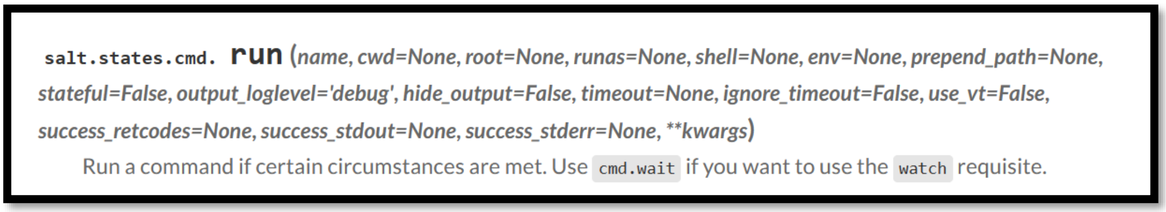
- Download the WordPress archive file from https://wordpress.org/download to its directory.`/tmp`



`salt.states.cmd.` **`run`** *(name, cwd=None, root=None, runas=None, shell=None, env=None, prepend_path=None, stateful=False, output_loglevel='debug', hide_output=False, timeout=None, ignore_timeout=False, use_vt=False, success_retcodes=None, success_stdout=None, success_stderr=None, **kwargs)*

Run a command if certain circumstances are met. Use `cmd.wait` if you want to use the `watch` requisite.

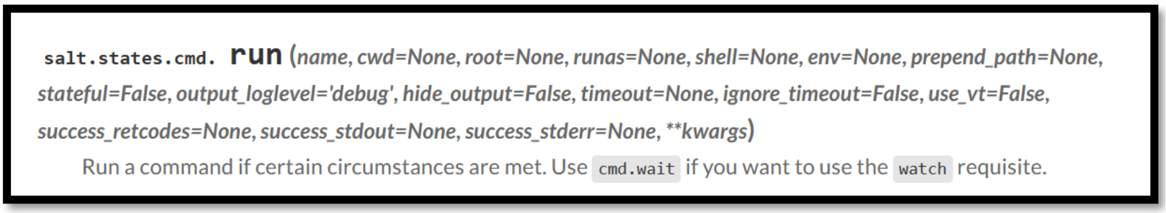
- Unzip the WordPress archive file to its directory. `/var/www/wordpress2023`



`salt.states.cmd.` **run** (*name, cwd=None, root=None, runas=None, shell=None, env=None, prepend_path=None, stateful=False, output_loglevel='debug', hide_output=False, timeout=None, ignore_timeout=False, use_vt=False, success_retcodes=None, success_stdout=None, success_stderr=None, **kwargs*)

Run a command if certain circumstances are met. Use `cmd.wait` if you want to use the `watch` requisite.

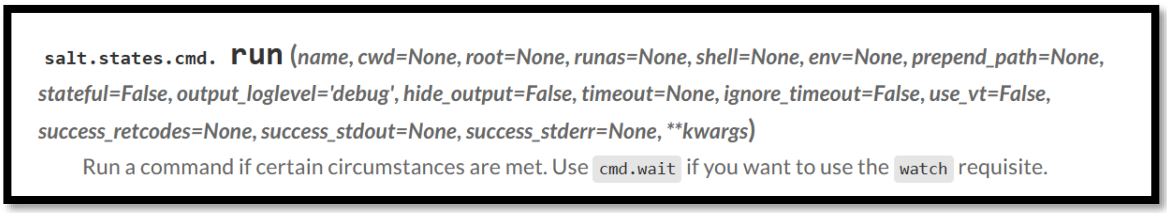
- Configure the Nginx service to reload each time the contents of the /etc/nginx/nginx.conf file are updated.



`salt.states.cmd.` **run** (*name, cwd=None, root=None, runas=None, shell=None, env=None, prepend_path=None, stateful=False, output_loglevel='debug', hide_output=False, timeout=None, ignore_timeout=False, use_vt=False, success_retcodes=None, success_stdout=None, success_stderr=None, **kwargs*)

Run a command if certain circumstances are met. Use `cmd.wait` if you want to use the `watch` requisite.

- `wp-config.php` Enter the MySQL database and user information you created on the Ubuntu server in the database information in the file.
- `wp-config.php` Fill in the file by pulling the necessary secrets and keys for WordPress from the https://api.wordpress.org/secret-key/1.1/salt/ address.



`salt.states.cmd.` **run** (*name, cwd=None, root=None, runas=None, shell=None, env=None, prepend_path=None, stateful=False, output_loglevel='debug', hide_output=False, timeout=None, ignore_timeout=False, use_vt=False, success_retcodes=None, success_stdout=None, success_stderr=None, **kwargs*)

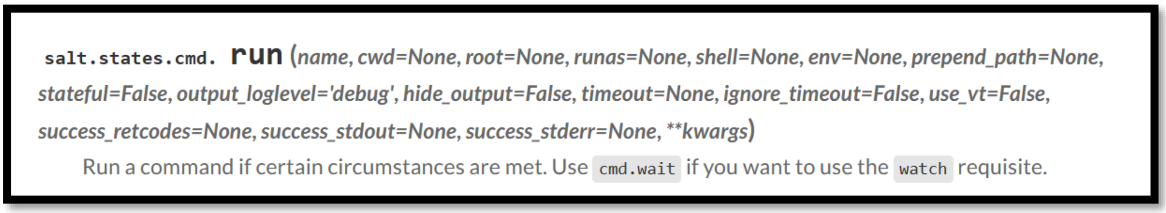
Run a command if certain circumstances are met. Use `cmd.wait` if you want to use the `watch` requisite.

{{ pillar['mysql']['db_name'] }} will be replaced with the value of db_name from the pillar file.
{{ pillar['mysql']['db_user'] }} will be replaced with the value of db_user from the pillar file.
{{ pillar['mysql']['db_password'] }} will be replaced with the value of db_password from the pillar file.

In the sed command, database_name_here, username_here, and password_here are just placeholders. They need to match the placeholders used in your wp-config.php file that you want to replace

The /g at the end of the substitution expression stands for "global." It tells sed to replace all occurrences of the pattern in each line, not just the first occurrence.

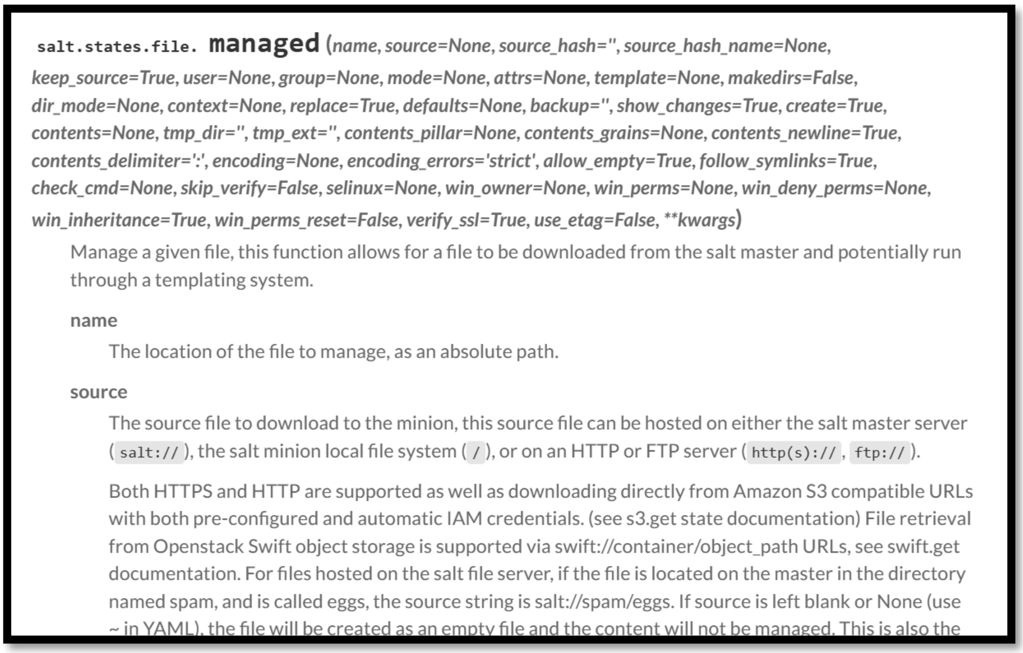
- *Create a self-signed SSL* certificate and include it in the Nginx configuration.



`salt.states.cmd.` **run** (*name, cwd=None, root=None, runas=None, shell=None, env=None, prepend_path=None, stateful=False, output_loglevel='debug', hide_output=False, timeout=None, ignore_timeout=False, use_vt=False, success_retcodes=None, success_stdout=None, success_stderr=None, **kwargs*)

Run a command if certain circumstances are met. Use `cmd.wait` if you want to use the `watch` requisite.

- Manage the Nginx configuration with Salt, each time Salt state is applied, the file on the server is updated from the file in the files directory, which is in the same directory as the Salt state file. `/etc/nginx/nginx.confnginx.conf`



`salt.states.file.` **managed** *(name, source=None, source_hash='', source_hash_name=None, keep_source=True, user=None, group=None, mode=None, attrs=None, template=None, makedirs=False, dir_mode=None, context=None, replace=True, defaults=None, backup='', show_changes=True, create=True, contents=None, tmp_dir='', tmp_ext='', contents_pillar=None, contents_grains=None, contents_newline=True, contents_delimiter=':', encoding=None, encoding_errors='strict', allow_empty=True, follow_symlinks=True, check_cmd=None, skip_verify=False, selinux=None, win_owner=None, win_perms=None, win_deny_perms=None, win_inheritance=True, win_perms_reset=False, verify_ssl=True, use_etag=False, **kwargs)*

Manage a given file, this function allows for a file to be downloaded from the salt master and potentially run through a templating system.

name

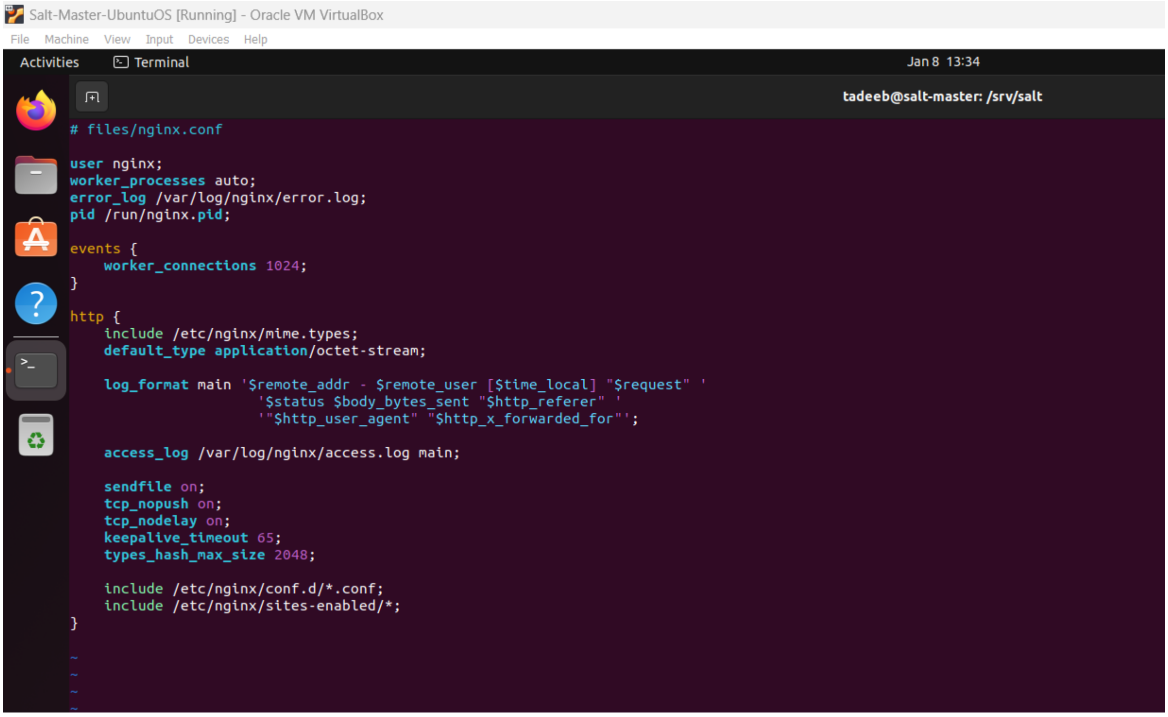
The location of the file to manage, as an absolute path.

source

The source file to download to the minion, this source file can be hosted on either the salt master server (`salt://`), the salt minion local file system (`/`), or on an HTTP or FTP server (`http(s)://`, `ftp://`).

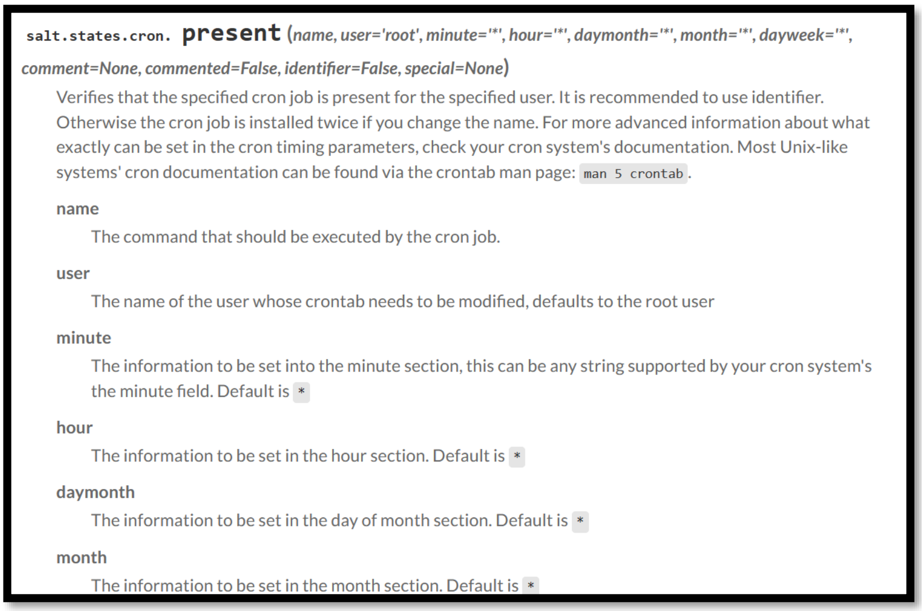
Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials. (see s3.get state documentation) File retrieval from Openstack Swift object storage is supported via swift://container/object_path URLs, see swift.get documentation. For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs. If source is left blank or None (use ~ in YAML), the file will be created as an empty file and the content will not be managed. This is also the

I added this file at /srv/salt/files/nginx.conf:


Salt-Master-UbuntuOS [Running] - Oracle VM VirtualBox
File
Machine
View
Input
Devices
Help
Activities
Terminal
Jan 8 13:34
tadeeb@salt-master: /srv/salt

```
# files/nginx.conf

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include /etc/nginx/conf.d/*.conf;
    include /etc/nginx/sites-enabled/*;
}
~
~
~
~
```

- Create a cron that will stop and restart the Nginx service on the first day of each month.



`salt.states.cron.` **present** (*name, user='root', minute='*', hour='*', daymonth='*', month='*', dayweek='*', comment=None, commented=False, identifier=False, special=None*)

Verifies that the specified cron job is present for the specified user. It is recommended to use identifier. Otherwise the cron job is installed twice if you change the name. For more advanced information about what exactly can be set in the cron timing parameters, check your cron system's documentation. Most Unix-like systems' cron documentation can be found via the crontab man page: `man 5 crontab`.

name

The command that should be executed by the cron job.

user

The name of the user whose crontab needs to be modified, defaults to the root user

minute

The information to be set into the minute section, this can be any string supported by your cron system's the minute field. Default is `*`

hour

The information to be set in the hour section. Default is `*`

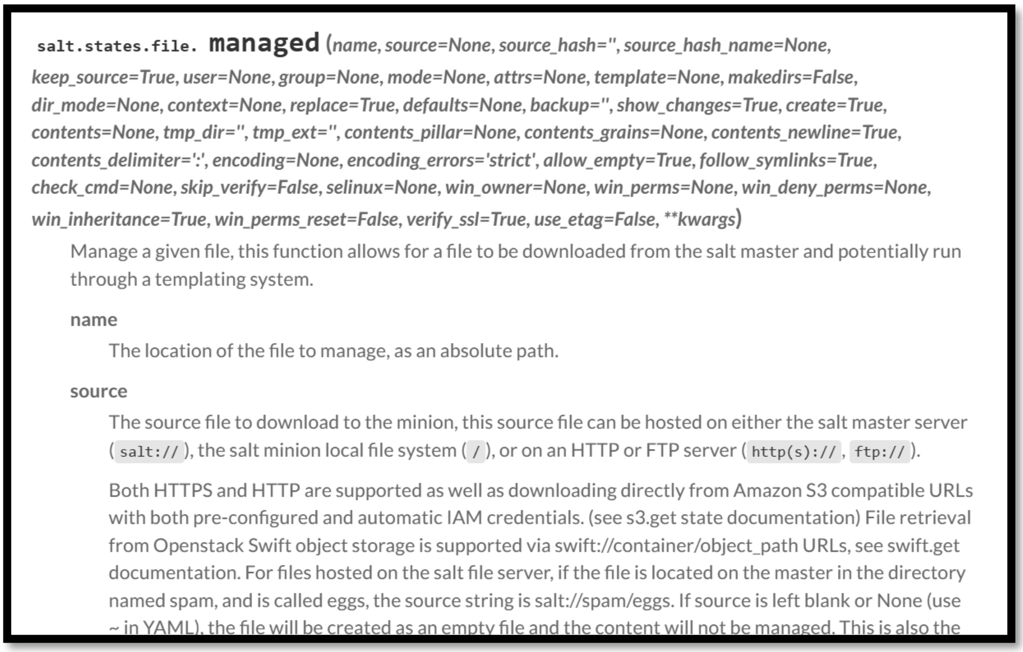
daymonth

The information to be set in the day of month section. Default is `*`

month

The information to be set in the month section. Default is `*`

- Make a *logroate* configuration that will rotate nginx logs hourly, compress rotated log files with gzip, and store only the last 10 files.



`salt.states.file.` **managed** (*name, source=None, source_hash='', source_hash_name=None, keep_source=True, user=None, group=None, mode=None, attrs=None, template=None, makedirs=False, dir_mode=None, context=None, replace=True, defaults=None, backup='', show_changes=True, create=True, contents=None, tmp_dir='', tmp_ext='', contents_pillar=None, contents_grains=None, contents_newline=True, contents_delimiter=':', encoding=None, encoding_errors='strict', allow_empty=True, follow_symlinks=True, check_cmd=None, skip_verify=False, selinux=None, win_owner=None, win_perms=None, win_deny_perms=None, win_inheritance=True, win_perms_reset=False, verify_ssl=True, use_etag=False, **kwargs*)

Manage a given file, this function allows for a file to be downloaded from the salt master and potentially run through a templating system.

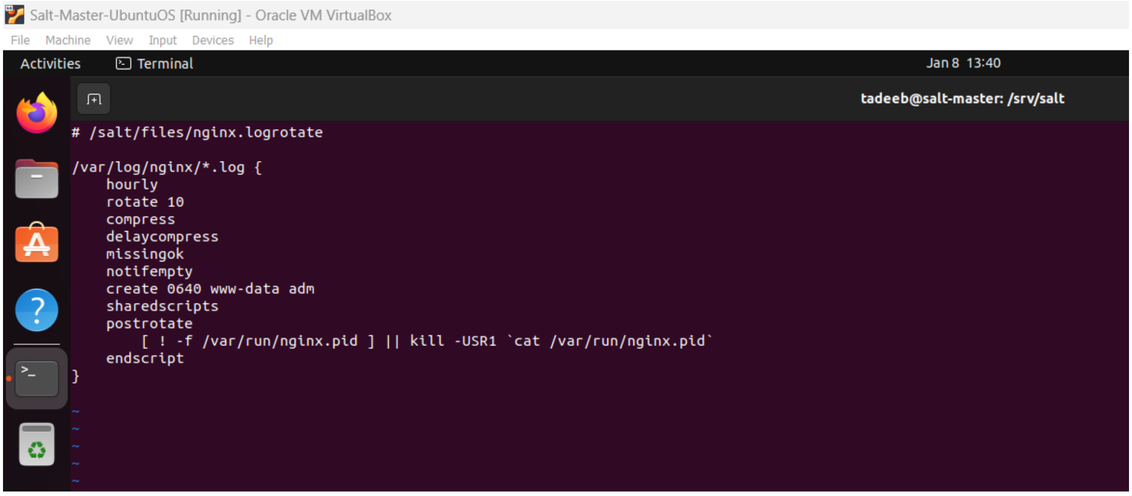
name

The location of the file to manage, as an absolute path.

source

The source file to download to the minion, this source file can be hosted on either the salt master server (`salt://`), the salt minion local file system (`/`), or on an HTTP or FTP server (`http(s)://`, `ftp://`).

Both HTTPS and HTTP are supported as well as downloading directly from Amazon S3 compatible URLs with both pre-configured and automatic IAM credentials. (see s3.get state documentation) File retrieval from Openstack Swift object storage is supported via swift://container/object_path URLs, see swift.get documentation. For files hosted on the salt file server, if the file is located on the master in the directory named spam, and is called eggs, the source string is salt://spam/eggs. If source is left blank or None (use ~ in YAML), the file will be created as an empty file and the content will not be managed. This is also the

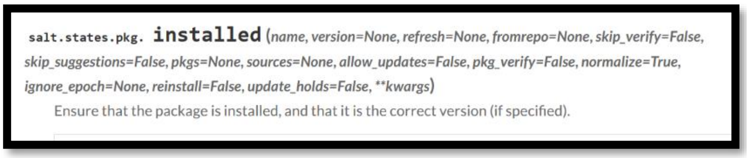


Salt-Master-UbuntuOS [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Jan 8 13:40
tadeeb@salt-master: /srv/salt

```
# /salt/files/nginx.logrotate

/var/log/nginx/*.log {
    hourly
    rotate 10
    compress
    delaycompress
    missingok
    notifempty
    create 0640 www-data adm
    sharedscripts
    postrotate
        [ ! -f /var/run/nginx.pid ] || kill -USR1 `cat /var/run/nginx.pid`
    endscript
}
~
~
~
~
~
```

On the Ubuntu server:

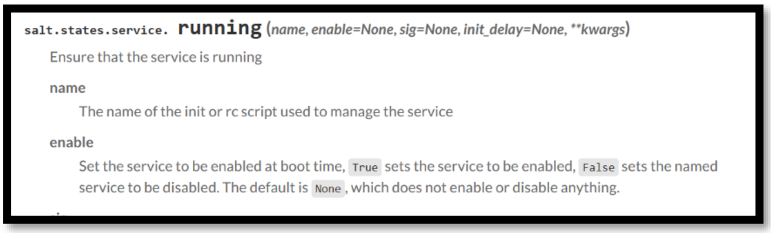
- Set up a MySQL database.



salt.states.pkg. **installed** (*name, version=None, refresh=None, fromrepo=None, skip_verify=False, skip_suggestions=False, pkgs=None, sources=None, allow_updates=False, pkg_verify=False, normalize=True, ignore_epoch=None, reinstall=False, update_holds=False, **kwargs*)

Ensure that the package is installed, and that it is the correct version (if specified).

- Configure the MySQL service to start automatically when the server restarts.



salt.states.service. **running** (*name, enable=None, sig=None, init_delay=None, **kwargs*)

Ensure that the service is running

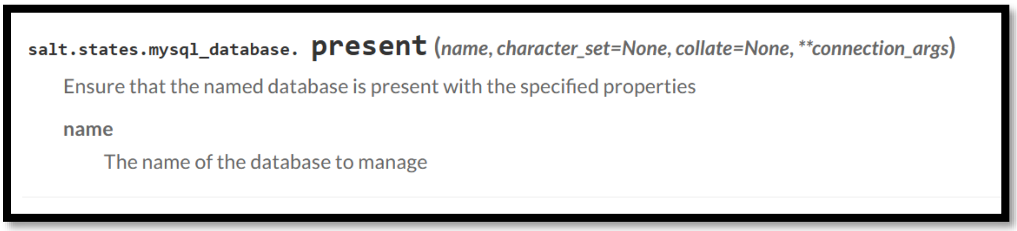
name

The name of the init or rc script used to manage the service

enable

Set the service to be enabled at boot time, `True` sets the service to be enabled, `False` sets the named service to be disabled. The default is `None`, which does not enable or disable anything.

- Create a database and user on MySQL for WordPress installation, define the necessary database privileges for the user you created. (Get the required information for the database name, user, and password from the pillar data, not from the Salt state file.)

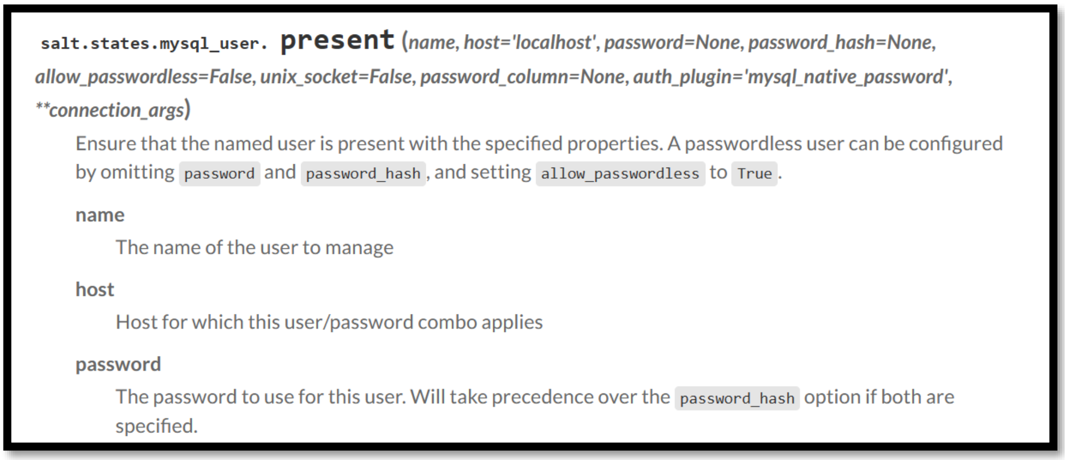


salt.states.mysql_database. **present** (*name, character_set=None, collate=None, **connection_args*)

Ensure that the named database is present with the specified properties

name

The name of the database to manage



salt.states.mysql_user. **present** (*name, host='localhost', password=None, password_hash=None, allow_passwordless=False, unix_socket=False, password_column=None, auth_plugin='mysql_native_password', **connection_args*)

Ensure that the named user is present with the specified properties. A passwordless user can be configured by omitting `password` and `password_hash`, and setting `allow_passwordless` to `True`.

name

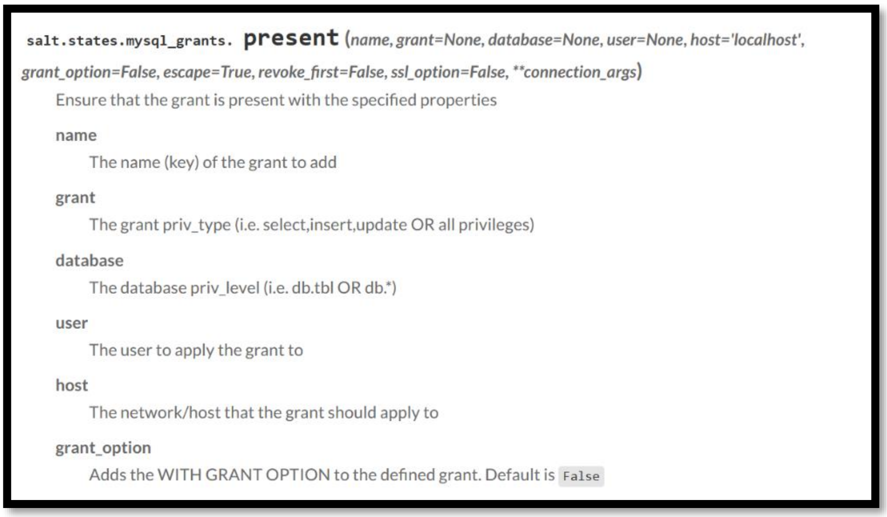
The name of the user to manage

host

Host for which this user/password combo applies

password

The password to use for this user. Will take precedence over the `password_hash` option if both are specified.



salt.states.mysql_grants. **present** (*name, grant=None, database=None, user=None, host='localhost', grant_option=False, escape=True, revoke_first=False, ssl_option=False, **connection_args*)

Ensure that the grant is present with the specified properties

name

The name (key) of the grant to add

grant

The grant priv_type (i.e. select,insert,update OR all privileges)

database

The database priv_level (i.e. db.tbl OR db.*)

user

The user to apply the grant to

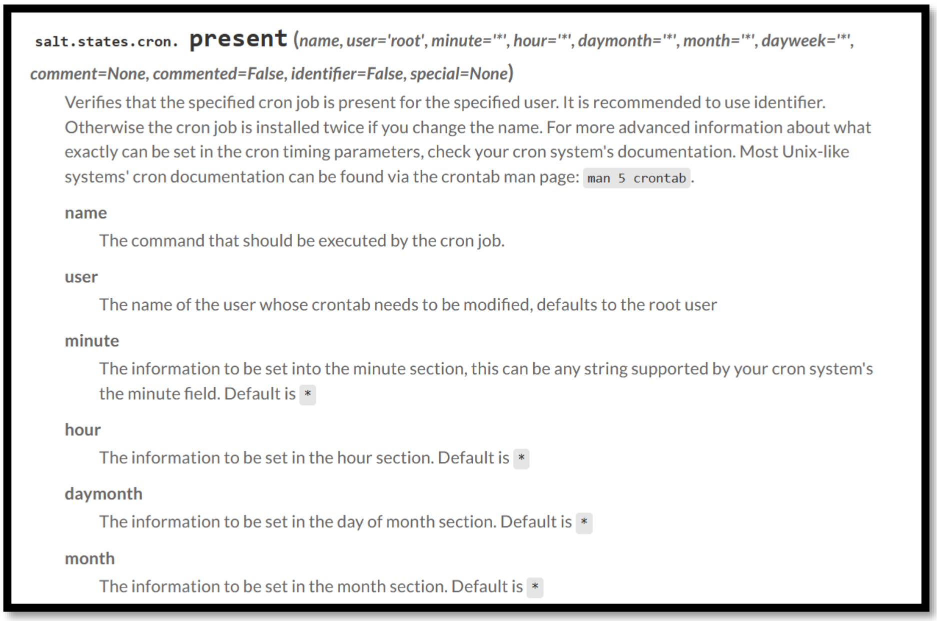
host

The network/host that the grant should apply to

grant_option

Adds the WITH GRANT OPTION to the defined grant. Default is `False`

- Prepare a cron that will index the MySQL database dump every night at 02:00. `/backup`



`salt.states.cron.` **`present`** (*name, user='root', minute='*', hour='*', daymonth='*', month='*', dayweek='*', comment=None, commented=False, identifier=False, special=None*)

Verifies that the specified cron job is present for the specified user. It is recommended to use identifier. Otherwise the cron job is installed twice if you change the name. For more advanced information about what exactly can be set in the cron timing parameters, check your cron system's documentation. Most Unix-like systems' cron documentation can be found via the crontab man page: `man 5 crontab`.

name

The command that should be executed by the cron job.

user

The name of the user whose crontab needs to be modified, defaults to the root user

minute

The information to be set into the minute section, this can be any string supported by your cron system's the minute field. Default is `*`

hour

The information to be set in the hour section. Default is `*`

daymonth

The information to be set in the day of month section. Default is `*`

month

The information to be set in the month section. Default is `*`

❖ Testing:

NOTE : The whole code file is uploaded on my github.

I did testing and 80% of my code is working, rest 20% is issue in last part i.e. MySql modules.