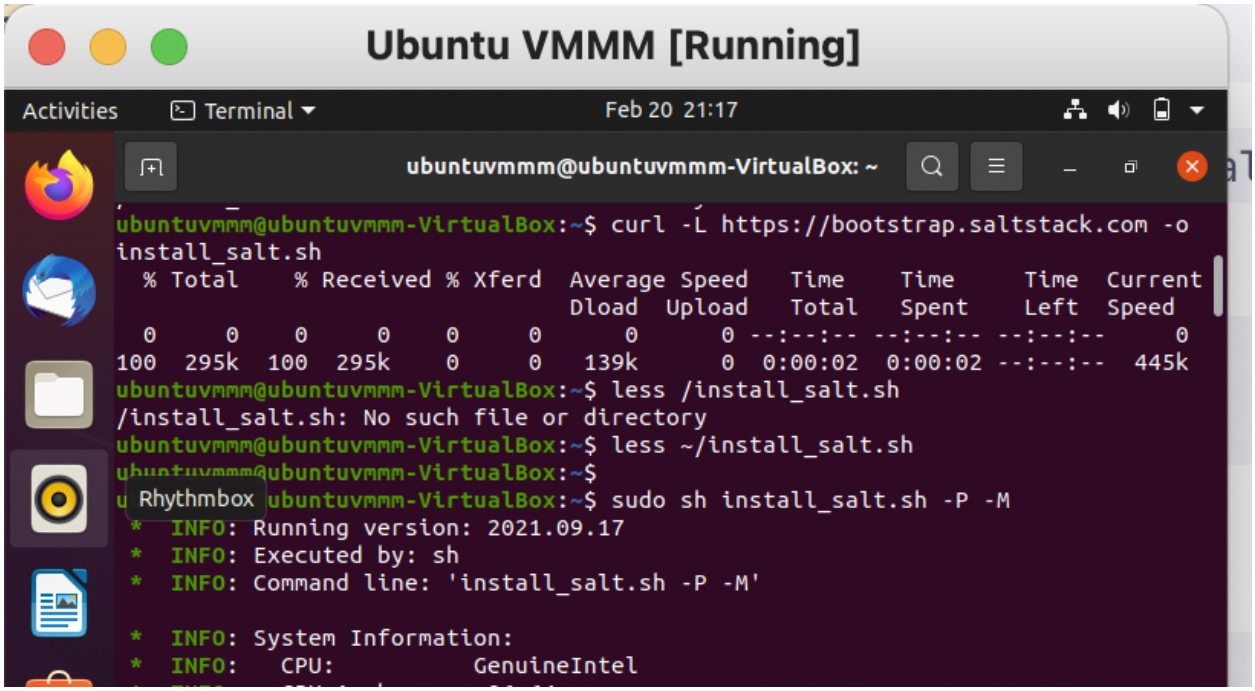
SALTSTACK ASSIGNMENT

Task: Create two linux VMs on local

- setup one host as a salt-master

To setup a virtual machine as the host master , we need to install saltstack on the virtual machine. To do so run the following commands:

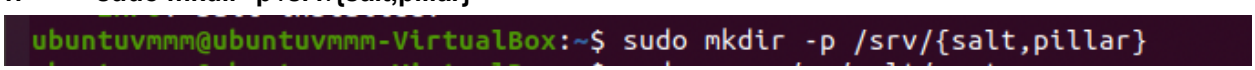
1. There are multiple ways to install this , I am using the salt bootstrap way.
   Use: **curl -L https://bootstrap.saltstack.com -o install_salt.sh**
   Use: **less ~/install_salt.sh**
   Run the Script: **sudo sh install_salt.sh -P -M**



Configure the saltstack as the salt master now:

Create the /srv directory with salt and pillar .

1. **sudo mkdir -p /srv/{salt,pillar}**

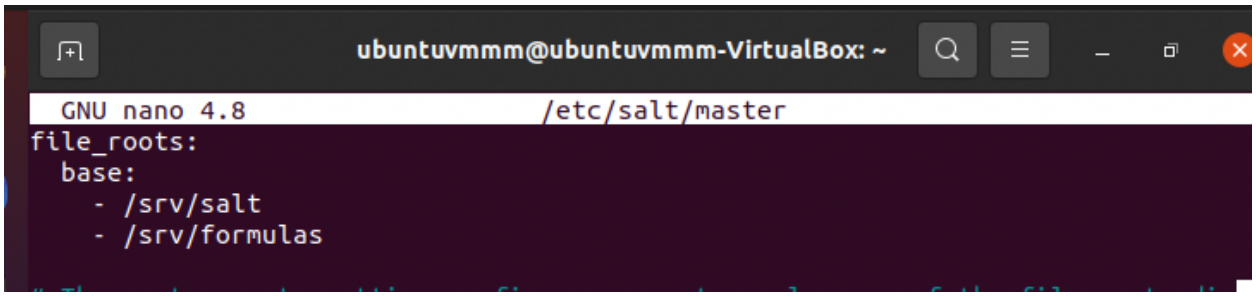


2. **sudo nano /etc/salt/master**

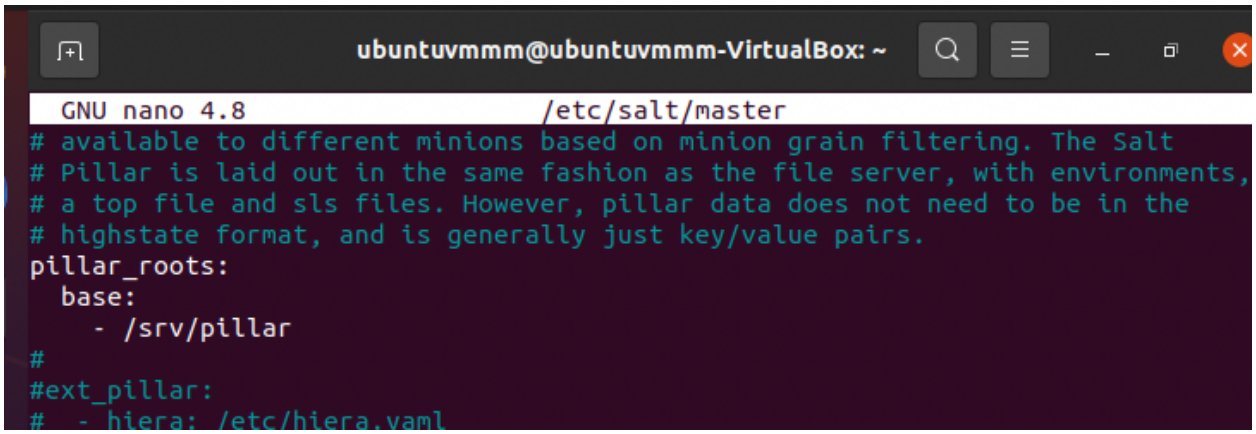
```
ubuntuvmmm@ubuntuvmmm-VirtualBox:~$ sudo nano /etc/salt/master
```

In the configuration file , remove the hash in front of these lines

file_roots:
base:
-/srv/salt
-/srv/formulas



pillar roots:
base:
-srv/pillar



3. Save and close the file now.

- ## Setup one host as a salt-minion

To setup a virtual machine as the host minion , we need to install saltstack-minion on the virtual machine . Run the following command to do so.

1. There are multiple ways to install this , I am using the salt bootstrap way.
   Use: **curl -L https://bootstrap.saltstack.com -o install_salt.sh**
   Use: **less ~/install_salt.sh**

Run the Script: **sudo sh install_salt.sh -P**

```
ubuntusm@ubuntusm-VirtualBox:~$ curl -L https://bootstrap.saltstack.com -o inst
all_salt.sh
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  295k  100  295k    0     0   133k      0  0:00:02  0:00:02 --:--:--  332k
ubuntusm@ubuntusm-VirtualBox:~$ sudo sh install_salt.sh -P
 *  INFO: Running version: 2021.09.17
 *  INFO: Executed by: sh
 *  INFO: Command line: 'install_salt.sh -P'

 *  INFO: System Information:
 *  INFO:   CPU:          GenuineIntel
 *  INFO:   CPU Arch:     x86_64
 *  INFO:   OS Name:      Linux
```

To configure the salt minion configuration:

1. Go to **sudo nano /etc/salt/minion**

```
ubuntusm@ubuntusm-VirtualBox:~$ sudo nano /etc/salt/minion
```

2. In the file : remove the hash in front of the master and assign the ip address of the salt master to it.
   master: ip_of_salt_master

   Now from the saltmaster virtual machine do:
3. **sudo salt-key -F master**
4. Take the master.pub key and use it in the minion configuration.

```
ubuntuvmmm@ubuntuvmmm-VirtualBox:~$ sudo salt-key -F master
Local Keys:
master.pem:  d0:f3:65:cd:0a:d6:09:e8:7b:1b:e6:b9:21:b2:7a:a9:ef:68:95:6a:b8:2a:
19:92:98:33:79:8a:92:d8:5c:d4
master.pub:  d5:75:0f:ac:58:c9:fe:1c:0b:38:de:94:b8:af:00:fb:15:51:04:9d:cb:6d:
3a:e5:22:a6:f3:9c:7d:7f:1c:70
```

   Go to the sudo minion virtual machine again and
5. **Sudo nano /etc/salt/minion**
6. Set the master_fingerprint here.
   master_finger: 'master_fingerprint_here'

```
# Set the location of the salt master server. If the master server cannot be
# resolved, then the minion will fail to start.
master: 10.0.2.15
master_finger: 'd5:75:0f:ac:58:c9:fe:1c:0b:38:de:94:b8:af:00:fb:15:51:04:9d:cb'
```

Restart the salt-minion and salt-master virtual machines.

1. Next Step is to Go to salt master and accept the key.

To do so do this:

2. Check the keys using **sudo salt-key –list all**

```
ubuntuvmmm@ubuntuvmmm-VirtualBox:~$ sudo systemctl restart salt-master
Help vmmm@ubuntuvmmm-VirtualBox:~$ sudo salt-key --list all
Accepted Keys:
Denied Keys:
Unaccepted Keys:
ubuntusm-VirtualBox
Rejected Keys:
```

3. Accept the key using **sudo salt-key -a <unaccepted_key_name>**

```
ubuntuvmmm@ubuntuvmmm-VirtualBox:~$ sudo salt-key -a ubuntusm-VirtualBox
The following keys are going to be accepted:
Unaccepted Keys:
ubuntusm-VirtualBox
Proceed? [n/Y] y
Key for minion ubuntusm-VirtualBox accepted.
ubuntuvmmm@ubuntuvmmm-VirtualBox:~$ sudo salt-key --list all
Accepted Keys:
ubuntusm-VirtualBox
Denied Keys:
Unaccepted Keys:
Rejected Keys:
```

4. Check that the connection is working.
5. **Sudo salt '*' test.ping**

```
ubuntuvmmm@ubuntuvmmm-VirtualBox:~$ sudo salt '*' test.ping
ubuntusm-VirtualBox:
    True
```

- write the salt states to setup the minion as an nginx web server sourcing the config from pillar.

To setup the salt states and configuration from pillar , we will have to configure the master file and set up the file roots and pillar roots.

To setup nginx as as web server in the minion.

Make the states and the pillar file at the location where the things have been setup in the master config for the file root and pillar root.

Set up the salt states in the state file like this:

Structure of the directories and file:

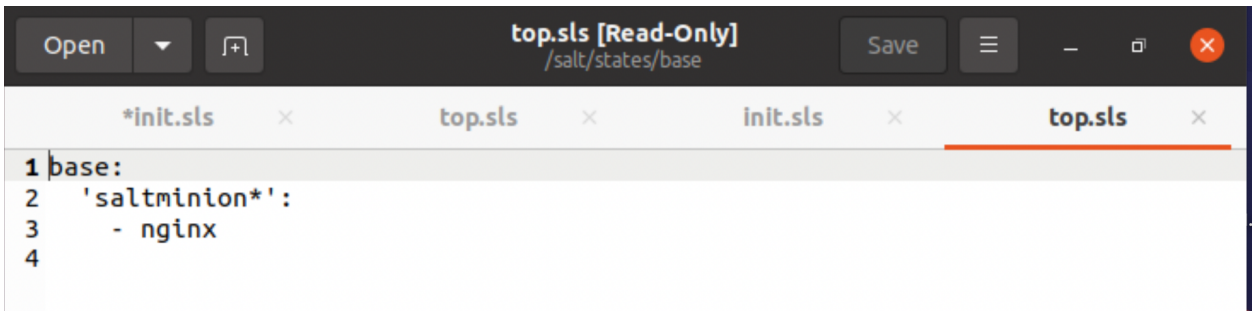
```
saltmaster@saltmaster-VirtualBox:~$ cd /salt
saltmaster@saltmaster-VirtualBox:/salt$ ls -R
.:
pillars  states

./pillars:
base

./pillars/base:
pkg  top.sls

./pillars/base/pkg:
init.sls

./states:
base

./states/base:
nginx  top.sls

./states/base/nginx:
init.sls
```

Salt states file:

/salt/states/base/nginx/init.sls

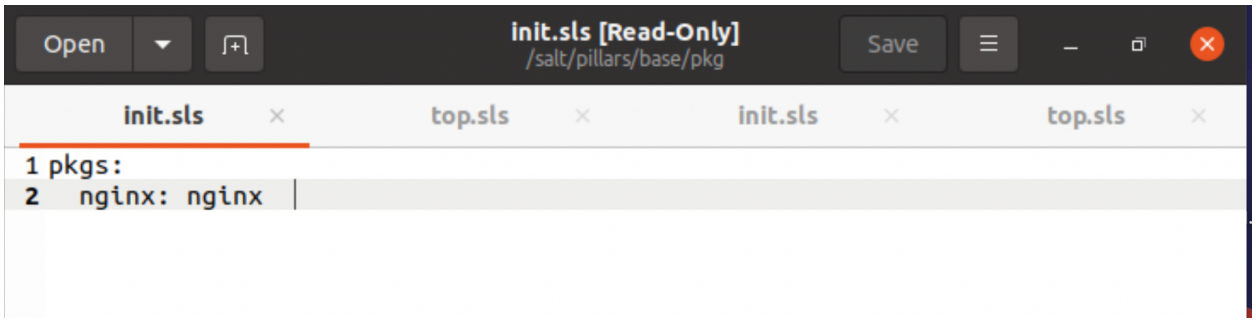
```
Open   init.sls [Read-Only]   Save
       /salt/states/base/nginx

*init.sls | top.sls | init.sls | top.sls

 1 nginx:
 2   pkg.installed:
 3     - name: {{ pillar['pkgs']['nginx'] }}
 4   service.running:
 5     - name: {{ pillar['pkgs']['nginx'] }}
 6     - watch:
 7       - pkg: nginx
 8       - file: /etc/nginx/nginx.conf
 9
10 /etc/nginx/nginx.conf:
11   file.managed:
12     - user: root
13     - group: root
14     - mode: 644
```

Top.sls file of /salt/states/base

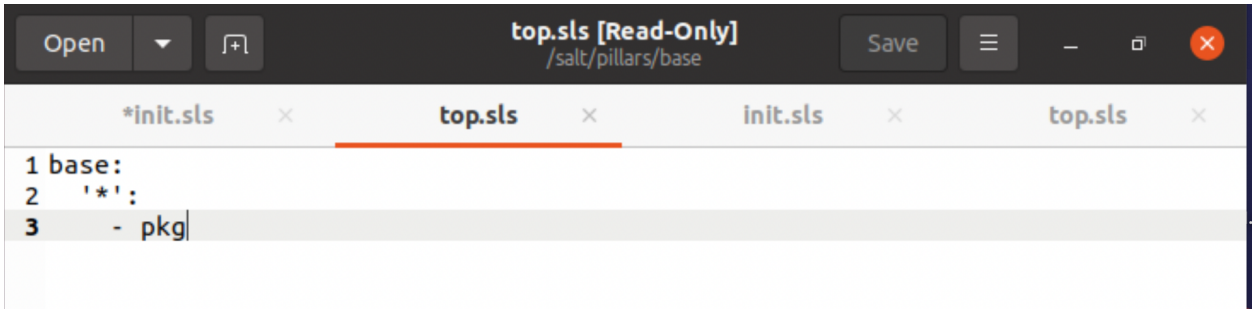


Configuration with the pillars include the data or package that needs to be send over there to be configured in the root pillars.

Here is the init.sls created in /salt/pillars/base/pkg

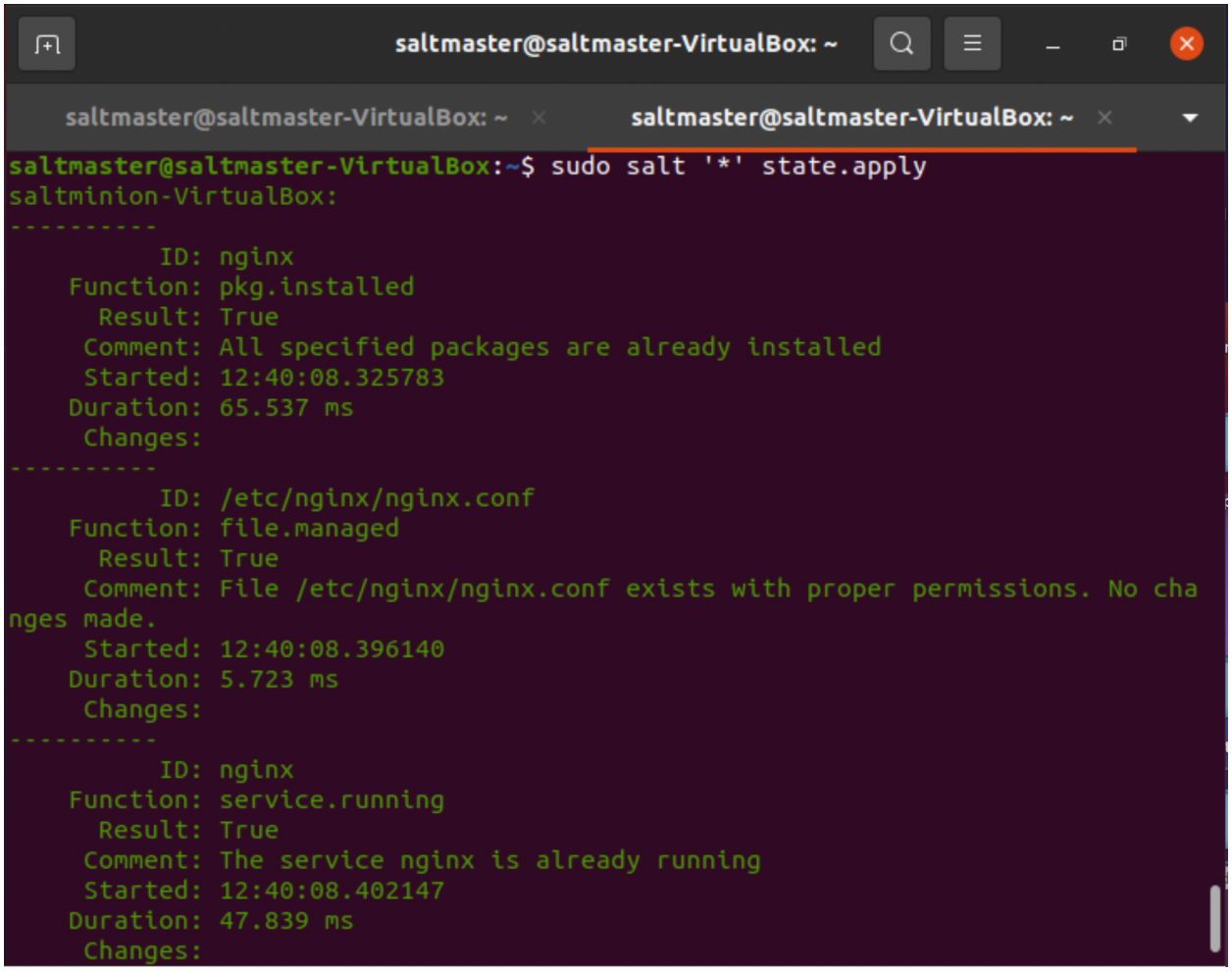


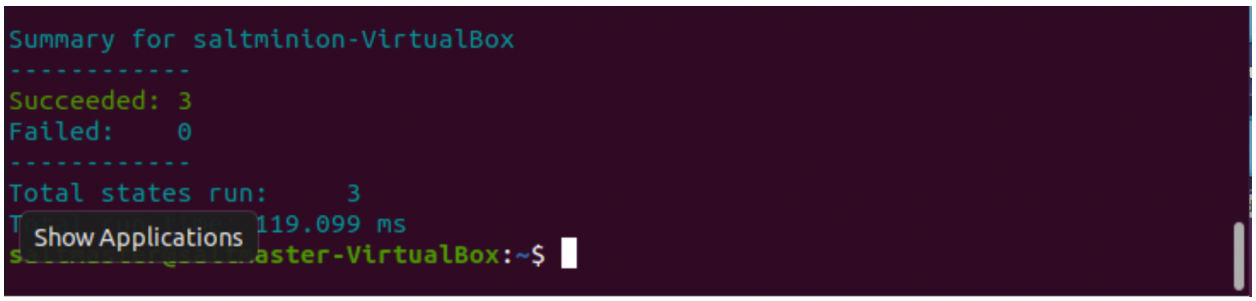
This is the top.sls file of the root pillars.
/salt/pillars/base/top.sls



Run the command

1. Start the service.
2. Refresh the pillars. Use : sudo salt '*' saltutil.refresh_pillar
3. Apply the salt states: Use: sudo salt '*' state.apply





- set one of the above VM as a recursive dns server using any of the tools (powerdns or dnscache) as mentioned in the session today and you should be able to resolv records from that dns server onto the other one.
    1. Install mariadb database server on salt-master .

       sudo apt update
       sudo apt install mariadb-server -y

2. Create a database user and assign the privileges by logging into the mariadb shell with sudo mysql -u root

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE pdns;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> GRANT ALL on pdns.* TO 'pdns'@'localhost' IDENTIFIED BY 'root
';
Query OK, 0 rows affected (0.004 sec)

MariaDB [(none)]> use pdns;
Database changed
MariaDB [pdns]> CREATE TABLE domains (
    -> id    INT AUTO_INCREMENT,
    -> name VARCHAR(255) NOT NULL,
    -> master VARCHAR(128) DEFAULT NULL,
    -> last_check INT DEFAULT NULL,
    -> type  VARCHAR(6) NOT NULL,
    -> notified_serial  INT DEFAULT NULL,
    -> account  VARCHAR(40) DEFAULT NULL,
    -> PRIMARY KEY (id)
    -> ) Engine = InnoDB;
Query OK, 0 rows affected (0.008 sec)

MariaDB [pdns]> CREATE UNIQUE INDEX name_index ON domains(name);
Query OK, 0 rows affected (0.008 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

3. Create the tables and input some data in that table.

```
MariaDB [pdns]> CREATE TABLE records ( id     BIGINT AUTO_INCREMENT, domain_id
 INT DEFAULT NULL, name   VARCHAR(255) DEFAULT NULL, type   VARCHAR(10) DEFAULT
 NULL, content VARCHAR(4000) DEFAULT NULL, ttl     INT DEFAULT NULL, prio    IN
T DEFAULT NULL, change_date INT DEFAULT NULL, disabled TINYINT(1) DEFAULT 0, or
dername VARCHAR(255) BINARY DEFAULT NULL, auth TINYINT(1) DEFAULT 1, PRIMARY KE
Y(id) ) Engine=InnoDB;
Query OK, 0 rows affected (0.012 sec)

MariaDB [pdns]> CREATE INDEX nametype_index ON records(name,type);
Query OK, 0 rows affected (0.008 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [pdns]> CREATE INDEX domain_id ON records(domain_id);
Query OK, 0 rows affected (0.009 sec)
Records: 0  Duplicates: 0  Warnings: 0

Terminal
M[illegible]pdns]> CREATE INDEX recordorder ON records (domain_id,ordername);
Query OK, 0 rows affected (0.009 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [pdns]> CREATE TABLE supermasters( ip                VARCHAR(64) NOT NUL
L, nameserver       VARCHAR(255) NOT NULL, account           VARCHAR(40) NOT NUL
L, PRIMARY KEY(ip,nameserver) ) Engine=InnoDB;
Query OK, 0 rows affected (0.008 sec)

MariaDB [pdns]> CREATE TABLE comments (
    -> id  INT AUTO_INCREMENT,
    -> domain_id  INT NOT NULL,
    -> name  VARCHAR(255) NOT NULL,
    -> type  VARCHAR(10) NOT NULL,
    -> modified_at  INT NOT NULL,
    -> account VARCHAR(40) NOT NULL,
    -> comment  VARCHAR(4000) NOT NULL,
    -> PRIMARY KEY(id)
    -> ) Engine=InnoDB;
Query OK, 0 rows affected (0.009 sec)

MariaDB [pdns]> CREATE INDEX comments_domain_id_idx ON comments(domain_id
    -> );
Query OK, 0 rows affected (0.008 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [pdns]> CREATE INDEX comments_name_type_idx ON comments(name,type);
Query OK, 0 rows affected (0.011 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [pdns]> CREATE INDEX comments_order_idx ON comments(domain_id,modified_
at);
Query OK, 0 rows affected (0.014 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [pdns]> CREATE TABLE domainmetadata(
    -> id    INT AUTO_INCREMENT,
    -> domain_id  INT NOT NULL,
    -> kind    VARCHAR(32),
    -> content  TEXT,
    -> PRIMARY KEY(id)
    -> ) Engine=InnoDB;
Query OK, 0 rows affected (0.009 sec)

MariaDB [pdns]> CREATE INDEX domainmetadata_idx ON domainmetadata(domain_id,kin
d);
Query OK, 0 rows affected (0.009 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [pdns]> CREATE TABLE cryptokeys(
    -> id INT AUTO_INCREMENT,
    -> domain_id   INT NOT NULL,
    -> flag  INT NOT NULL,
    -> active BOOL,
    -> content TEXT,
    -> PRIMARY KEY(id)
    -> ) Engine=InnoDB;
Query OK, 0 rows affected (0.009 sec)

MariaDB [pdns]> CREATE INDEX domainidindex ON cryptokeys(domain_id);
Query OK, 0 rows affected (0.007 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [pdns]> CREATE TABLE tsigkeys(
    -> id INT AUTO_INCREMENT
    -> ,
    -> name  VARCHAR(255),
    -> algorithm VARCHAR(50),
    -> secret  VARCHAR(255),
    -> PRIMARY KEY(id)
    -> ) Engine=InnoDB;
Query OK, 0 rows affected (0.016 sec)

MariaDB [pdns]> CREATE UNIQUE INDEX namealgoindex ON tsigkeys(name,algorithm);
Query OK, 0 rows affected (0.010 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
MariaDB [pdns]> INSERT INTO domains(name,type) values ('example.com','NATIVE');
Query OK, 1 row affected (0.003 sec)

MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
ES (1,'example.com','localhost admin.example.com 1 10380 3600 604800 3600','SOA
',86400,NULL);
Query OK, 1 row affected (0.002 sec)

MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
ES (1,'example.com','dns-us1.powerdns.net','NS',86400,NULL);
Query OK, 1 row affected (0.003 sec)

MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
ES (1,'example.com','dns-eu1.powerdns.net','NS',86400,NULL);
Query OK, 1 row affected (0.003 sec)

MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
ES (1,'www.example.com','192.0.2.10','A',120,NULL);
Query OK, 1 row affected (0.004 sec)

MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
ES (1,'mail.example.com','192.0.2.12','A',120,NULL);
Query OK, 1 row affected (0.003 sec)

MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
E Show Applications .example.com','127.0.0.1','A',120,NULL);
Query OK, 1 row affected (0.003 sec)
```

```
MariaDB [pdns]> INSERT INTO records (domain_id,name,content,type,ttl,prio) VALU
ES (1,'example.com','mail.example.com','MX',120,25);
Q                                  affected (0.003 sec)
Show Applications
```

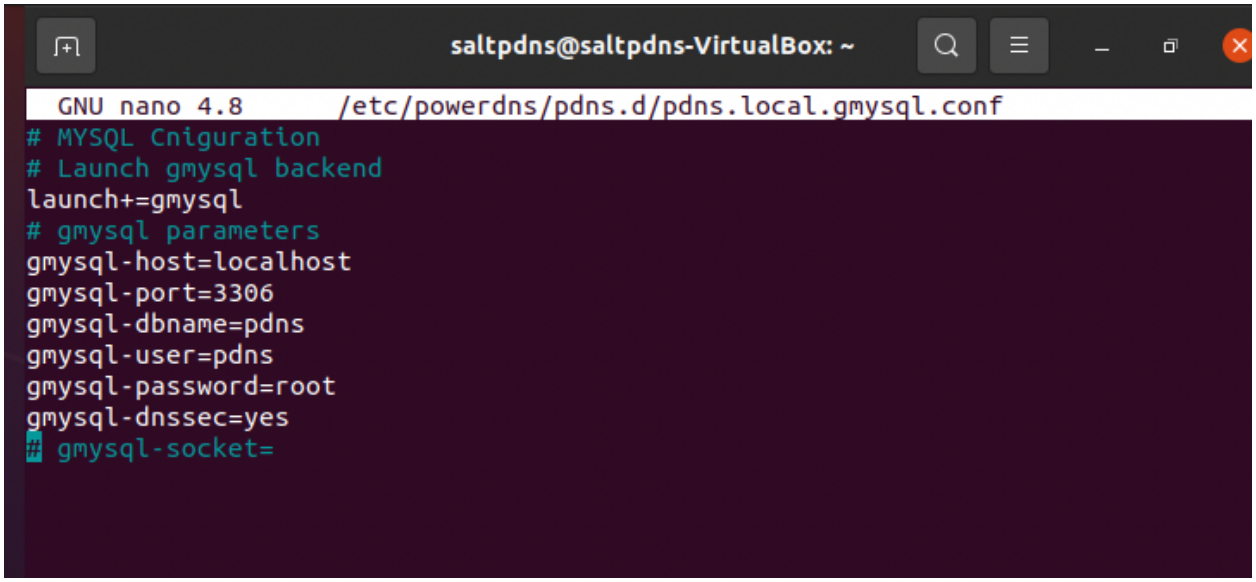
4. Disable the systemd-resolve to avoid the conflict with Powerdns ports

```
saltpdns@saltpdns-VirtualBox:~$ sudo systemctl disable systemd-resolved
[sudo] password for saltpdns:
Removed /etc/systemd/system/dbus-org.freedesktop.resolve1.service.
Removed /etc/systemd/system/multi-user.target.wants/systemd-resolved.service.
saltpdns@saltpdns-VirtualBox:~$ sudo systemctl stop systemd-resolved
saltpdns@saltpdns-VirtualBox:~$ ls -lh /etc/resolve.conf
ls: cannot access '/etc/resolve.conf': No such file or directory
saltpdns@saltpdns-VirtualBox:~$ ls -lh /etc/resolv.conf
lrwxrwxrwx 1 root root 39 Mar  6 21:32 /etc/resolv.conf -> ../run/systemd/resol
ve/stub-resolv.conf
saltpdns@saltpdns-VirtualBox:~$ sudo unlink /etc/resolv.conf
saltpdns@saltpdns-VirtualBox:~$ echo "nmeserver 8.8.8.8" | sudo tee /etc/resolv
.conf
nmeserver 8.8.8.8
saltpdns@saltpdns-VirtualBox:~$ sudo apt update
```

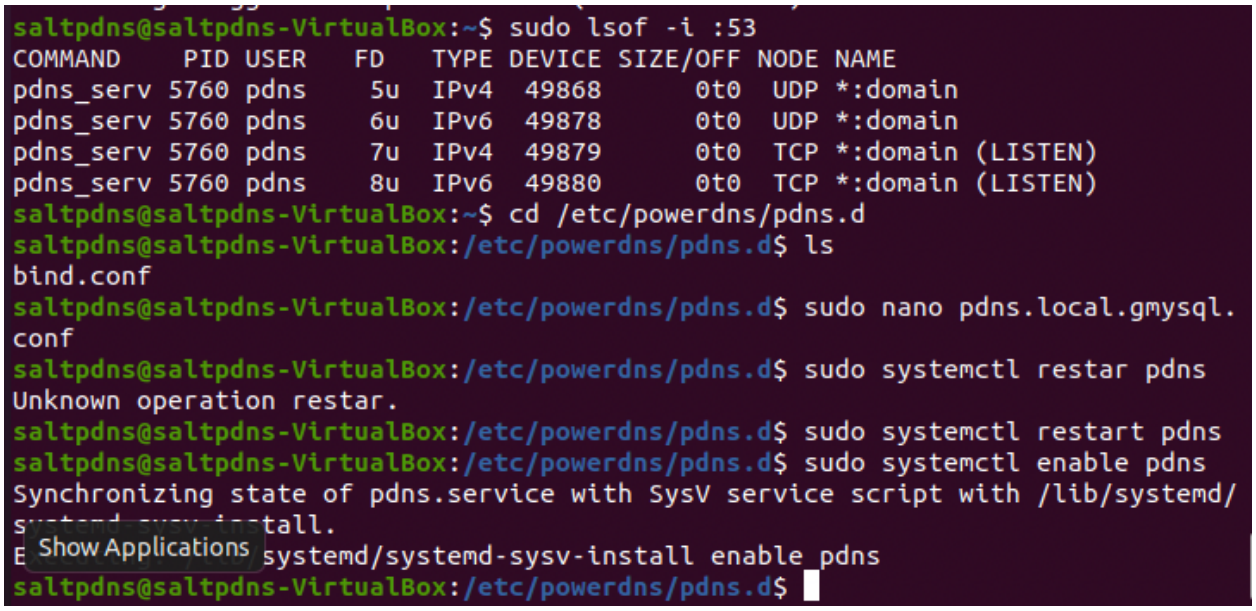
5. Install powerDNS :
   sudo apt update
   Sudo apt install pdns-server pdns-backend-mysql

```
saltpdns@saltpdns-VirtualBox:~$ sudo apt update
Get:1 http://in.archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Hit:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu focal-security InRelease
Hit:5 http://archive.ubuntu.com/ubuntu focal InRelease
Fetched 265 kB in 1s (305 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
All packages are up to date.
saltpdns@saltpdns-VirtualBox:~$ sudo apt install pdns-server pdns-backend-mysql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libboost-program-options1.71.0 pdns-backend-bind
Suggested packages:
  default-mysql-server
The following NEW packages will be installed:
  libboost-program-options1.71.0 pdns-backend-bind pdns-backend-mysql
  pdns-server
0 upgraded, 4 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,648 kB of archives.
After this operation, 14.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu focal/main amd64 libboost-program-opt
ions1.71.0 amd64 1.71.0-6ubuntu6 [342 kB]
```

6. Configure powerdns to use mysql backend by doing configuration in this file.



7. Check if the service is running on port. Restart powerdns and enable it.



8. Check the status of powerDNS.

```
saltpdns@saltpdns-VirtualBox:/etc/powerdns/pdns.d$ sudo systemctl status pdns
● pdns.service - PowerDNS Authoritative Server
     Loaded: loaded (/lib/systemd/system/pdns.service; enabled; vendor preset:>
     Active: active (running) since Sun 2022-03-06 23:58:33 IST; 24s ago
       Docs: man:pdns_server(1)
             man:pdns_control(1)
             https://doc.powerdns.com
   Main PID: 8840 (pdns_server)
      Tasks: 8 (limit: 1087)
     Memory: 6.0M
     CGroup: /system.slice/pdns.service
             └─8840 /usr/sbin/pdns_server --guardian=no --daemon=no --disable->

Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: TCPv6 server bound to [>
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: PowerDNS Authoritative >
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: Using 64-bits mode. Bui>
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: PowerDNS comes with ABS>
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: Creating backend connec>
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: [bindbackend] Parsing 0>
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: [bindbackend] Done pars>
Mar 06 23:58:33 saltpdns-VirtualBox systemd[1]: Started PowerDNS Authoritative>
Mar 06 23:58:33 saltpdns-VirtualBox pdns_server[8840]: About to create 3 backe>
Mar 06 23:58:34 saltpdns-VirtualBox pdns_server[8840]: Done launching threads,>
lines 1-22/22 (END)
```

Saltminionpdns:
Namespace 10.0.2.4(salt_master address)

```
  GNU nano 4.8                    /etc/resolv.conf                    Modified
# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "resolvectl status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.
nameserver 10.0.2.
nameserver 127.0.0.53
options edns0 trust-ad
```

```
saltminionpdns@saltminionpdns-VirtualBox:~$ sudo nano /etc/resolv.conf
saltminionpdns@saltminionpdns-VirtualBox:~$ sudo nano /etc/resolv.conf
saltminionpdns@saltminionpdns-VirtualBox:~$ dig +short www.example.com @10.0.2.
4
192.0.2.10
saltminionpdns@saltminionpdns-VirtualBox:~$
```

- write a daemon which monitors the health of machine - uptime, ram, cpu and disk usage

Create a script that monitors the health of a machine.

```
  GNU nano 4.8                          monitor.sh
#! /bin/bash

unset tecreset loadaverage tecuptime

tecreset=$(tput sgr0)

# Check RAM and SWAP usages
echo "-------------------------------------------------------------------------->
free -h | grep -v + > /tmp/ramcache
echo "Ram Usages:" $tecreset
cat /tmp/ramcache | grep -v "Swap"
echo "-------------------------------------------------------------------------->
echo "Swap Usages:" $tecreset
cat /tmp/ramcache | grep -v "Mem"

echo "-------------------------------------------------------------------------->
# Check Disk Usages
df -h | grep 'Filesystem\|/dev/sda*' > /tmp/diskusage
echo "Disk Usages :" $tecreset
cat /tmp/diskusage

echo "-------------------------------------------------------------------------->
# Check Load Average
loadaverage=$(top -n 1 -b | grep "load average:" | awk '{print $10 $11 $12}')
echo "Load Average:" $tecreset $loadaverage
```

```
echo "-------------------------------------------------------------------------->
#Check System Uptime
tecuptime=$(uptime | awk '{print $3,$4}' | cut -f1 -d,)
echo "System Uptime Days/(HH:MM) :" $tecreset $tecuptime

unset tecreset loadaverage tecuptime

rm /tmp/ramcache /tmp/diskusage
```

Output:

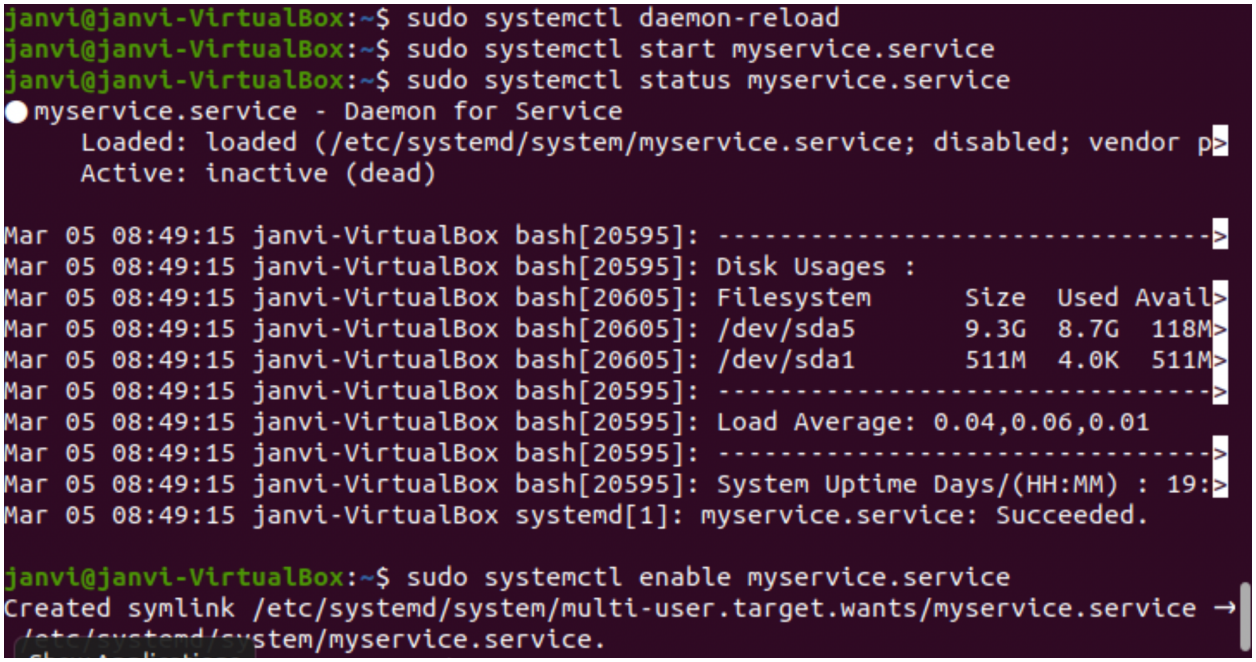
```
janvi@janvi-VirtualBox:~$ sudo nano monitor.sh
janvi@janvi-VirtualBox:~$ sh monitor.sh
-------------------------------------------------------------------------
Ram Usages:
              total        used        free      shared  buff/cache   available
Mem:          2.7Gi       1.7Gi       152Mi        13Mi       809Mi       783Mi
-------------------------------------------------------------------------
Swap Usages:
              total        used        free      shared  buff/cache   available
Swap:         448Mi       433Mi        15Mi
-------------------------------------------------------------------------
Disk Usages :
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda5       9.3G  8.7G  118M  99% /
/dev/sda1       511M  4.0K  511M   1% /boot/efi
-------------------------------------------------------------------------
Load Average:  0.03,0.03,0.00
-------------------------------------------------------------------------
System Uptime Days/(HH:MM) :  19:02
janvi@janvi-VirtualBox:~$
```

Run this as a daemon:

```
janvi@janvi-VirtualBox:~$ sudo cp monitor.sh /usr/bin/test_service.sh
janvi@janvi-VirtualBox:~$ sudo chmod +x /usr/bin/test_service.sh
janvi@janvi-VirtualBox:~$ cd /etc/systemcd/system
bash: cd: /etc/systemcd/system: No such file or directory
janvi@janvi-VirtualBox:~$ cd /etc/systemd/system
janvi@janvi-VirtualBox:/etc/systemd/system$ touch myservice.service
touch: cannot touch 'myservice.service': Permission denied
janvi@janvi-VirtualBox:/etc/systemd/system$ sudo touch myservice.service
```

```
janvi@janvi-VirtualBox:/etc/systemd/system$ sudo chmod 644 /etc/systemd/system/
myservice.service
janvi@janvi-VirtualBox:/etc/systemd/system$ ls -ls myservice.service
0 -rw-r--r-- 1 root root 0 Mar  5 08:43 myservice.service
janvi@janvi-VirtualBox:/etc/systemd/system$ sudo nano myservice.service
```

```
  GNU nano 4.8                    myservice.service                    M
[Unit]
Description=Daemon for Service

[Service]
Type=simple
ExecStart=/bin/bash /usr/bin/test_service.sh

[Install]
WantedBy=multi-user.target
```


janvi@janvi-VirtualBox:~$ sudo systemctl daemon-reload
janvi@janvi-VirtualBox:~$ sudo systemctl start myservice.service
janvi@janvi-VirtualBox:~$ sudo systemctl status myservice.service
myservice.service - Daemon for Service
Loaded: loaded (/etc/systemd/system/myservice.service; disabled; vendor p
Active: inactive (dead)
Mar 05 08:49:15 janvi-VirtualBox bash[20595]:
Mar 05 08:49:15 janvi-VirtualBox bash[20595]: Disk Usages :
Mar 05 08:49:15 janvi-VirtualBox bash[20605]: Filesystem Size Used Avail
Mar 05 08:49:15 janvi-VirtualBox bash[20605]: /dev/sda5 9.3G 8.7G 118M
Mar 05 08:49:15 janvi-VirtualBox bash[20605]: /dev/sda1 511M 4.0K 511M
Mar 05 08:49:15 janvi-VirtualBox bash[20595]:
Mar 05 08:49:15 janvi-VirtualBox bash[20595]: Load Average: 0.04,0.06,0.01
Mar 05 08:49:15 janvi-VirtualBox bash[20595]:
Mar 05 08:49:15 janvi-VirtualBox bash[20595]: System Uptime Days/(HH:MM) : 19:
Mar 05 08:49:15 janvi-VirtualBox systemd[1]: myservice.service: Succeeded.
janvi@janvi-VirtualBox:~$ sudo systemctl enable myservice.service
Created symlink /etc/systemd/system/multi-user.target.wants/myservice.service →
stem/myservice.service.