Network - Assignment

Task 1: Set up two focal VMs and assign IPs from the same network range and check if ping works.

Method 1: Static IP's Assigning

Step1: Install Two virtual Machines . In my case I have installed Linux Ubuntu Focal machines named VM1 and VM2 and set up a NAT adapter and an Internal Network adapter.

Step 2 : We need to Setup a Static IP address for the machines such that on reboot or on after any particular time. So , for that go into the `/etc/network/interfaces` file.

Use the command `sudo nano /etc/network/interfaces` or via netplan by using sudo nano /etc/netplan/<yaml file>

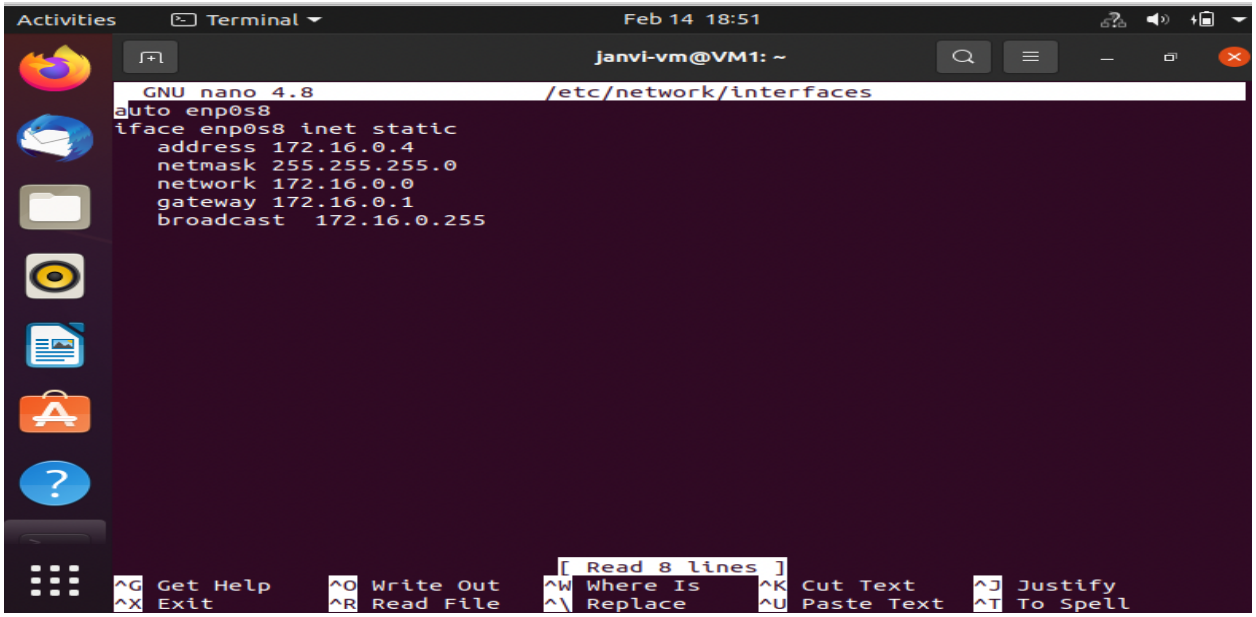
The netplan will have the configuration like

```
ethernets:
  enp0s3:
    addresses:
      -     172.16.0.1/24
    dhcp4: no
```

For network interface ,Write the configuration in that file

```
auto enp0s3
iface ep0s3 inet static
address <ip-address>
netmask <subnet-mask>
gateway <default-gateway>
broadcast <broadcast-point>
network <network>
```

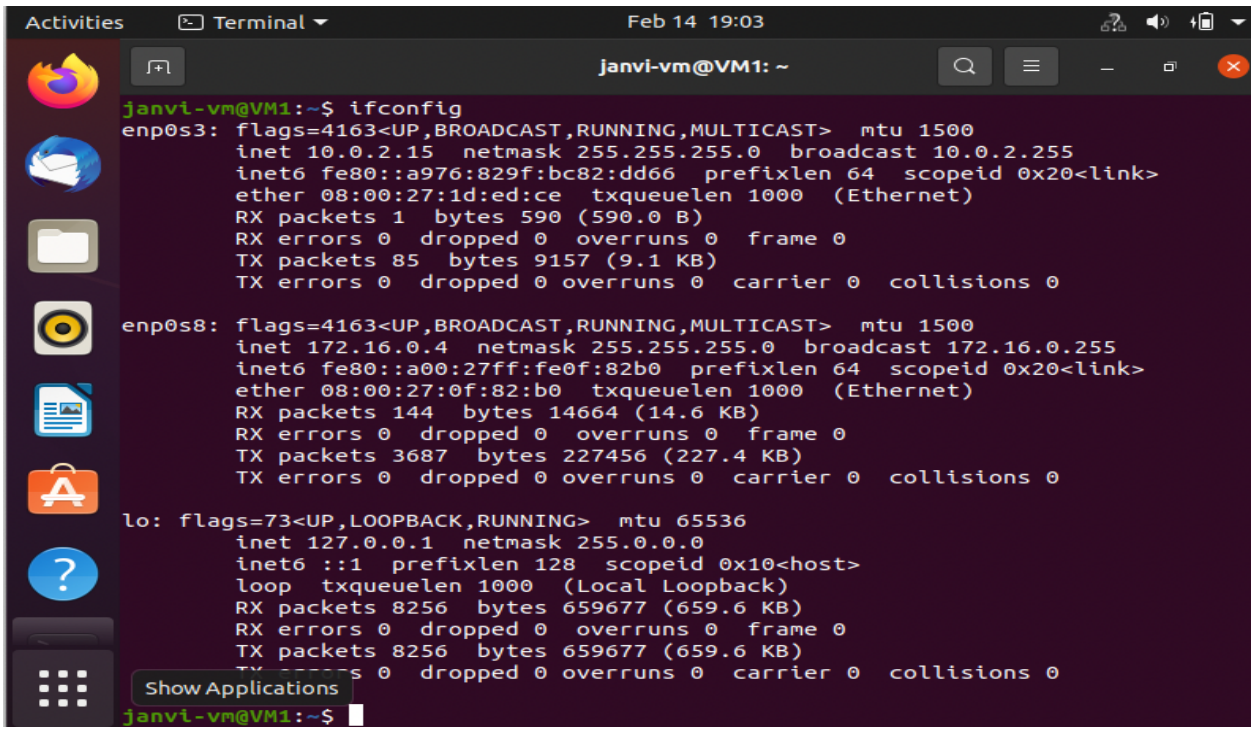
For me it looks like :



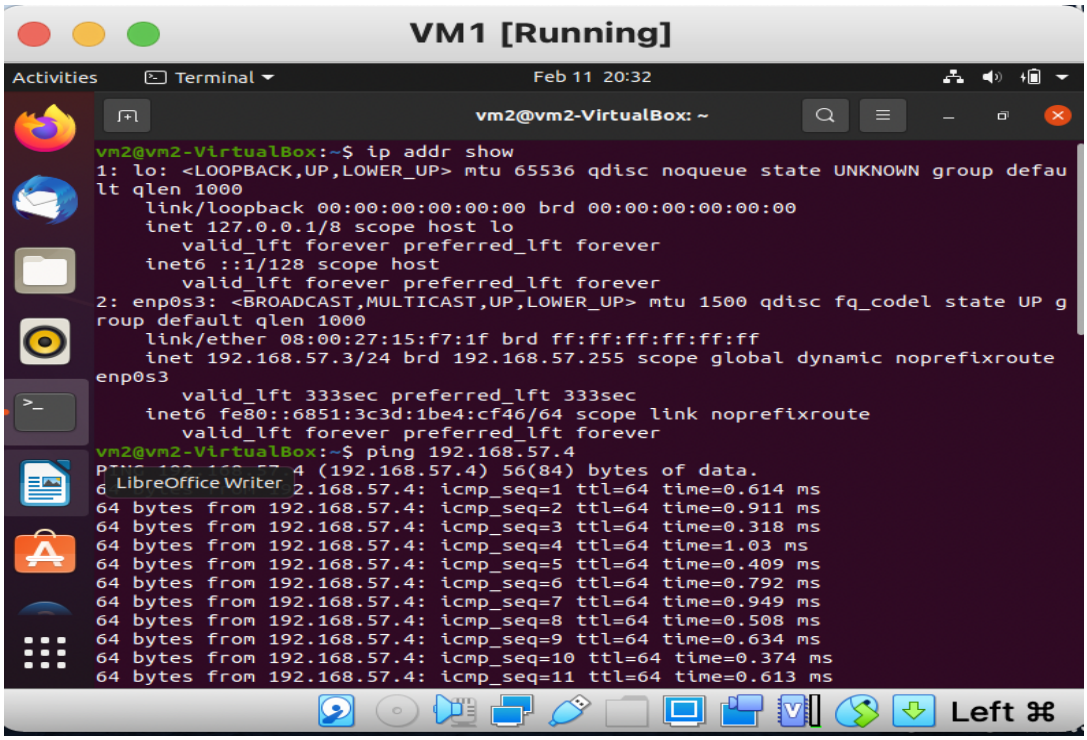
Save and exit this file .

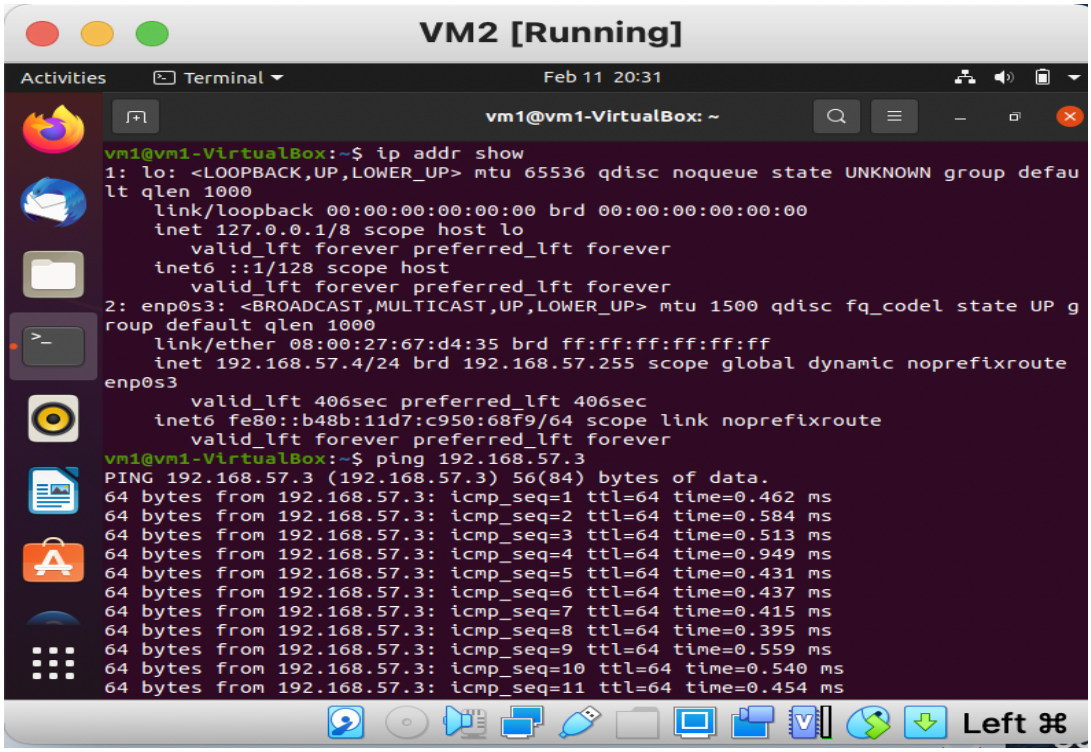
Step 3 : Reboot both the machines using sudo reboot.

Step 4 : After the machine is done with the reboot , Open the terminal and type in the command ifconfig or ip addr show . You'll see the IP address you have assigned to it.



You can reboot your machine as many times as you want now . This IP address is going to remain static.
Repeat the Process for The Virtual Machine 2.
In the similar way i have created two virtual machines having nat network , provided the static IP through the netplan or network interface method and tried to reboot he machines
Now after rebooting when you'll run the command id addr show , both the machines will be having the same Ip that we provided.
You can actually see it in the below screenshots. Also , they are connecting with each other via the ping command. So the ping command working looks fine.

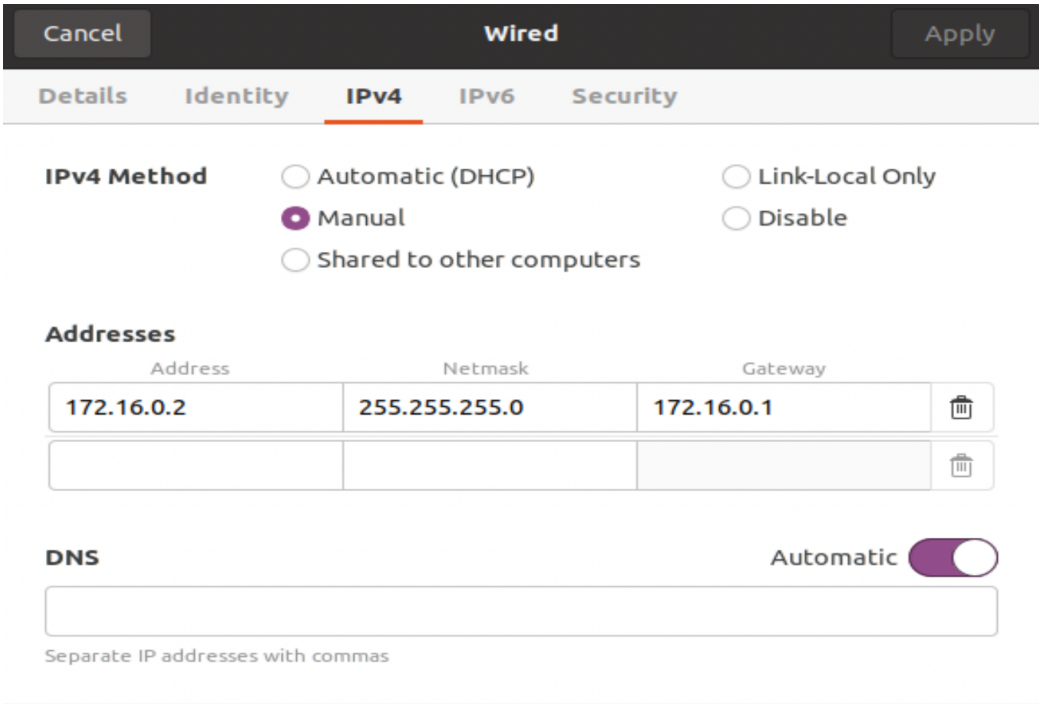




Method 3 : Manually Assigning the IP address.

In this method , Go to the Network settings of the virtual machines , go to the settings of the wired connection . There will be a connected network . Go to the settings icon of that connected network. Go to the IPV4 column there. The network is currently taking the automatic (DHCP) addresses.
You can select the manual option from there and you can assign the IP address , subnet mask and gateway to that machine.

The manual configuration page looks like this.



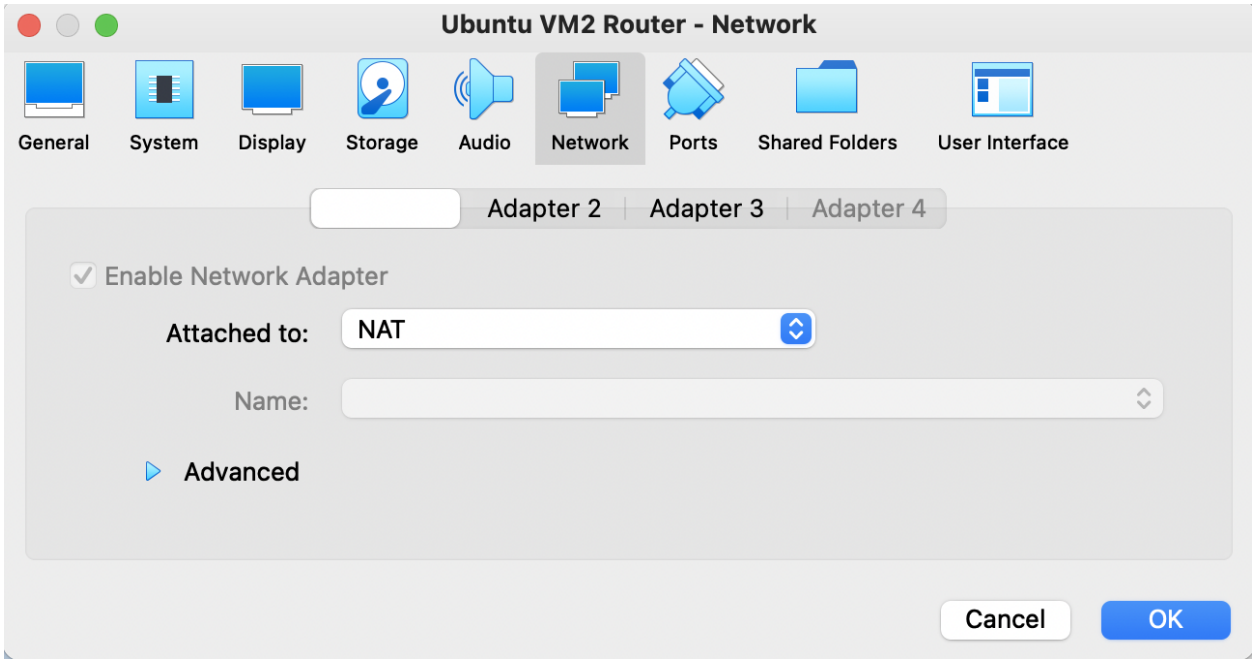
Just Keep in mind that the subnet mask and gateway will be the same. Only IP's will be changed.

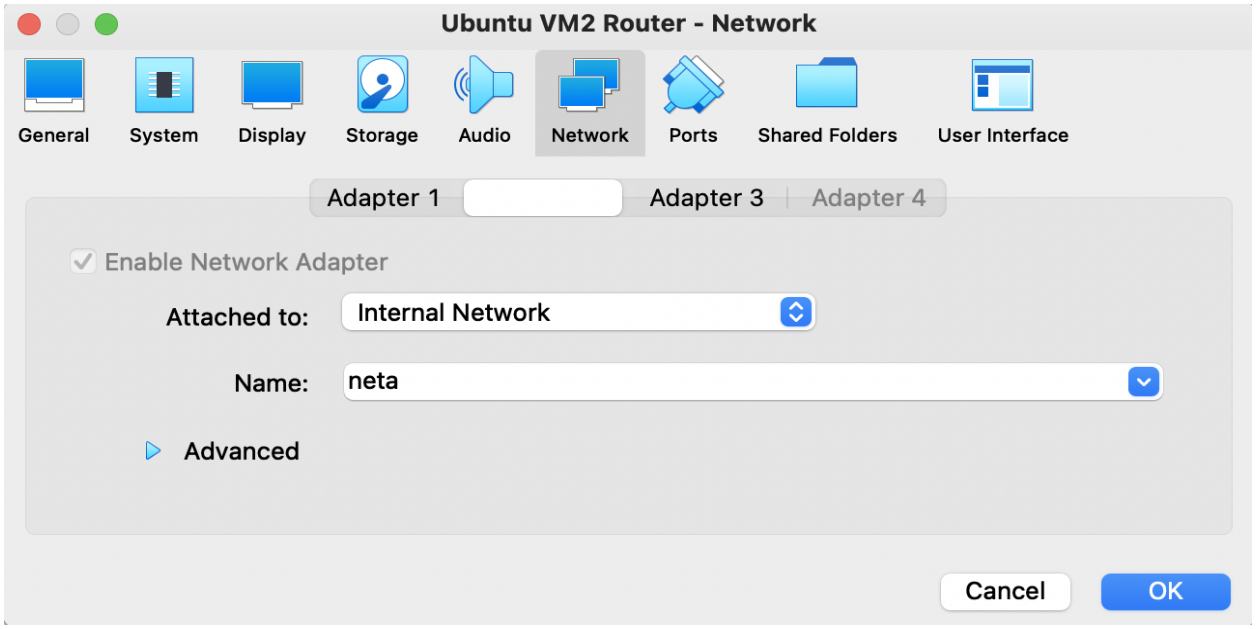
Task 2 : Add another focal VM to the setup and assign connectivity in the following order where VM2 acts like a router. VM1 <-> VM2 <-> VM3. VM1 should be from 172.16.0.0 /24 and VM3 should be from 192.168.0.0 /24. Should be done using static routing.

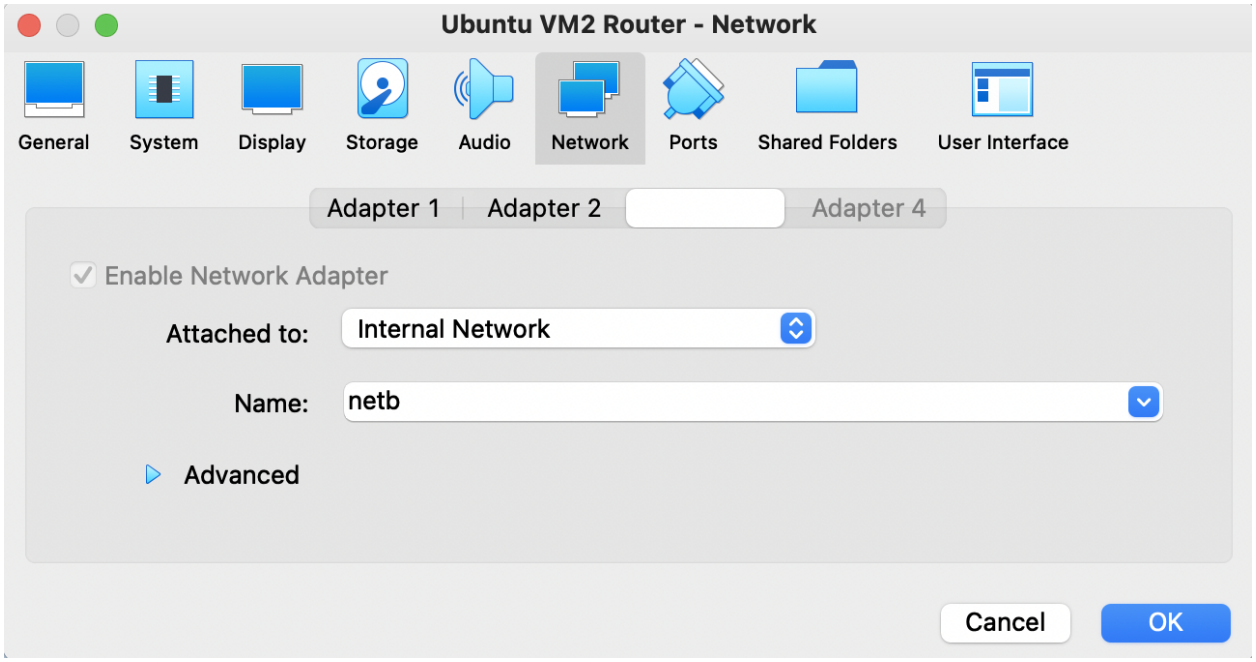
Step 1: Install three focal Virtual Machine's .

Step 2: To enable the connectivity in the order of VM1<-> VM2 <-> VM3 . We need to do this :
-> The VM1 consists of two network adapters , one is NAT for the internet acces and other using an internal network . let it be named neta.
-> The VM2 has to act like a router so it consists of three network adapters , one is NAT for the internet access and other two are the Internal networks that you have connected with VM1 and VM3.

-> The VM3 consists of two network adapters , one is NAt for the internet access and other using an internal network. Let it be named netb.



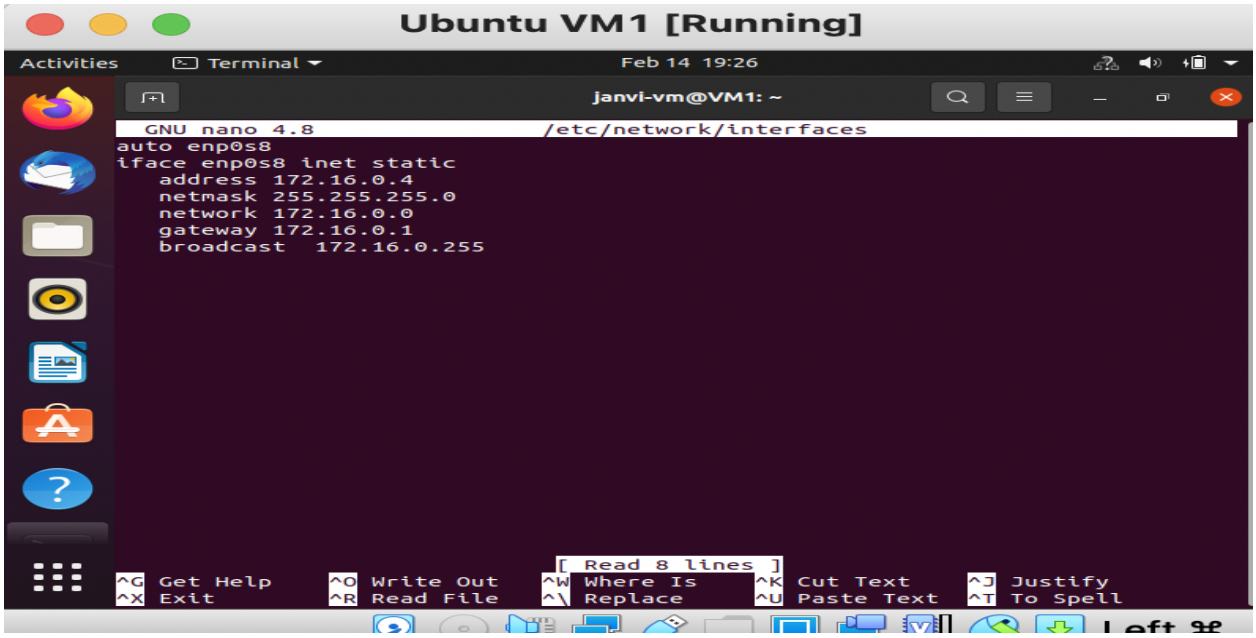




I have shows just for the VM2 above .

Step 3: Configure the Interfaces now.

-> In the VM1 Add the following to the /etc/network/interfaces file.

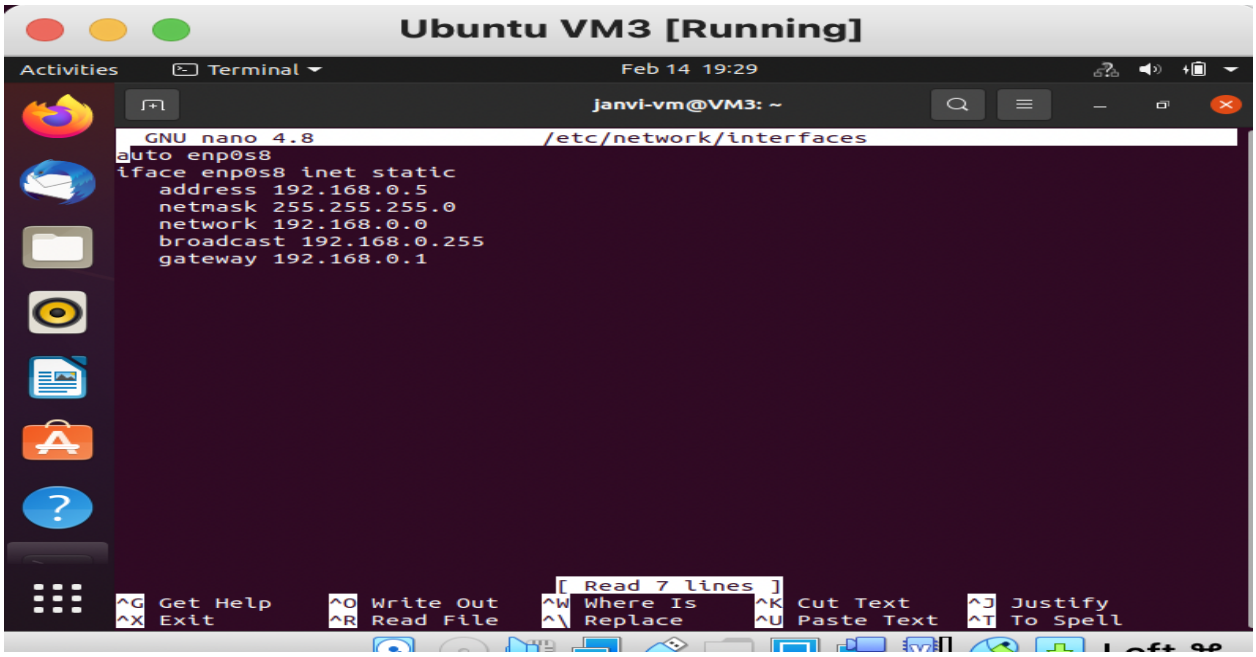


-> In the VM2 Add the following to the /etc/network/interfaces :

```
  GNU nano 4.8                    /etc/network/interfaces

#the internal interface for neta
auto enp0s8
iface enp0s8 inet static
   address 172.16.0.5
   netmask 255.255.255.0
   network 172.16.0.0
   broadcast 172.16.0.255
#the internal interface for netb
auto enp0s9
iface enp0s9 inet static
   address 192.168.0.4
   netmask 255.255.255.0
   network 192.168.0.0
   broadcast 192.168.0.255
```

-> In VM3 , Add the following in /etc/network/interfaces

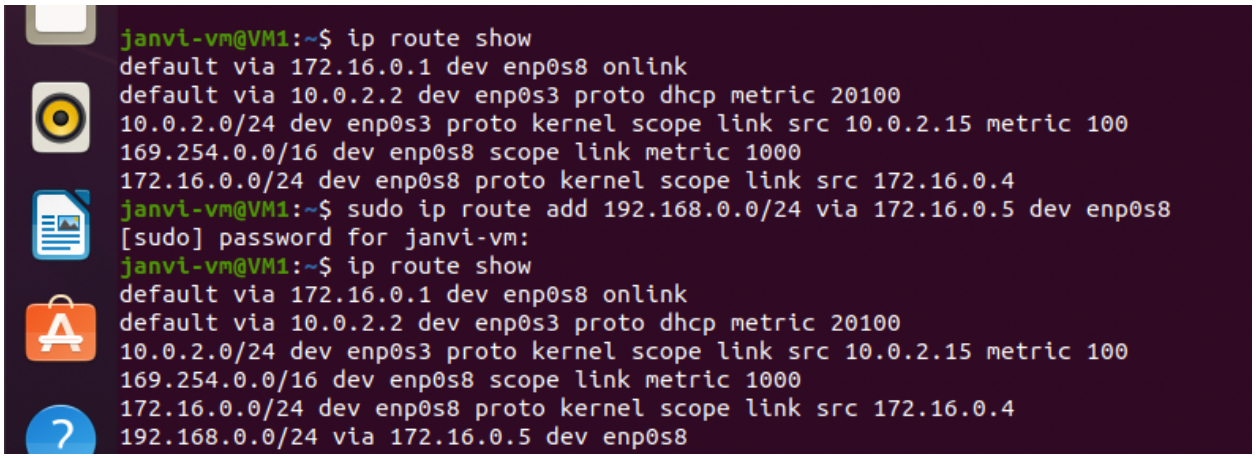


Step 4: Enable forwarding in the /etc/sysctl.conf file
You just have to uncomment the hask in the line referring to ip_forward to be :
net.ipv4.ip_forward = 1

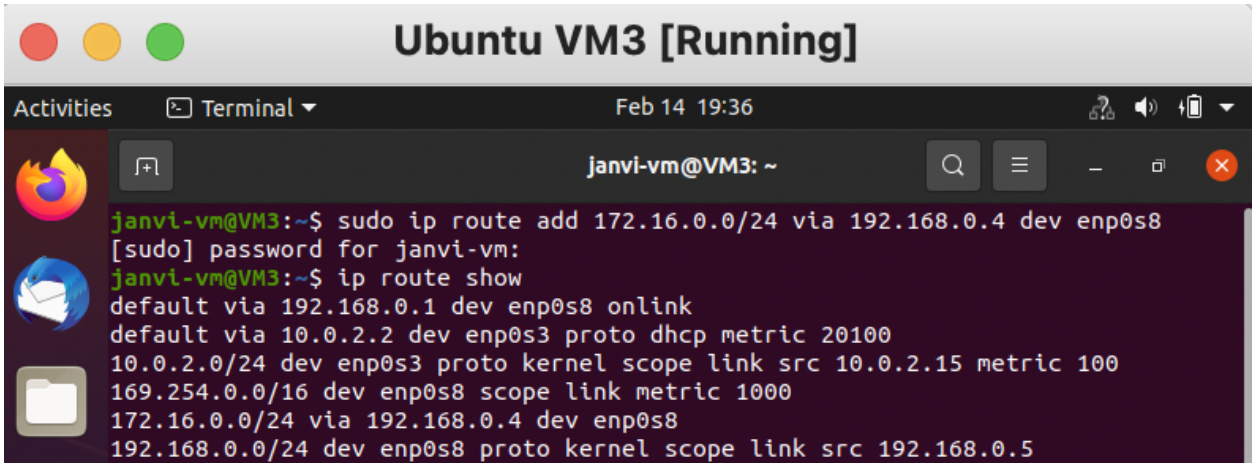
Step 5: Reboot the VMs. Using sudo reboot

Step 5 : Setup Routing . Check the routing table through the ip route show in each of the VM. In order to set up the connectivity between these VM's. Now add static routing using the below commands in VM1 and VM3.

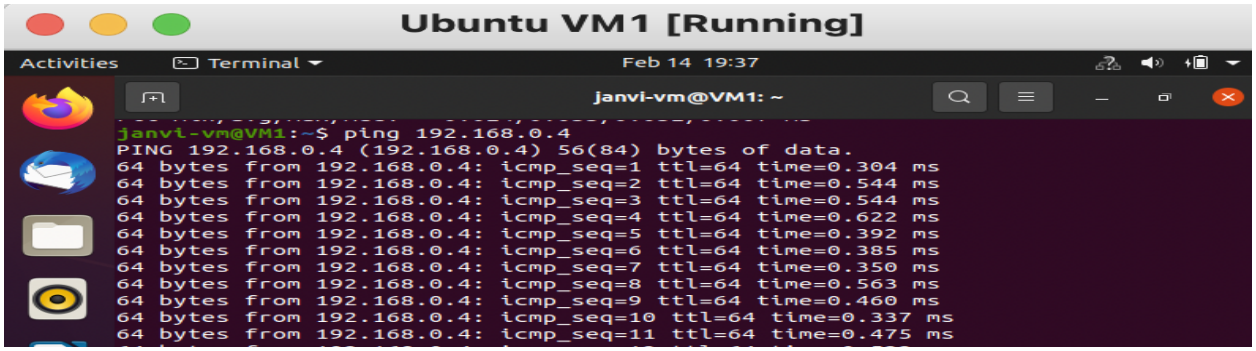
For VM1:

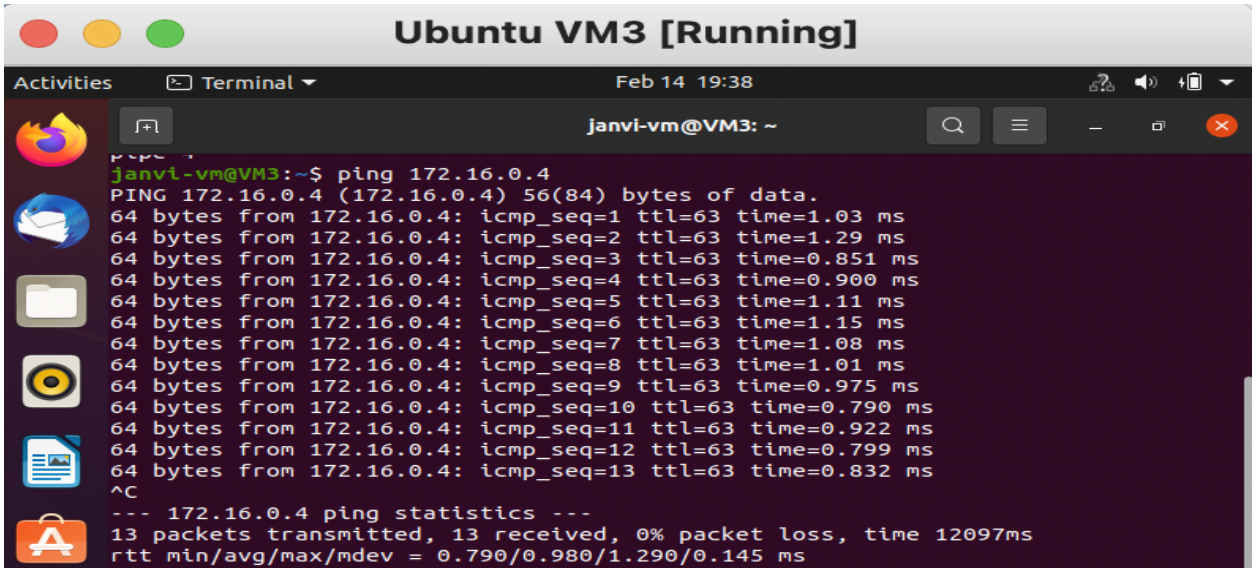


For VM3:



Step 6 : Ping in and check the connectivity





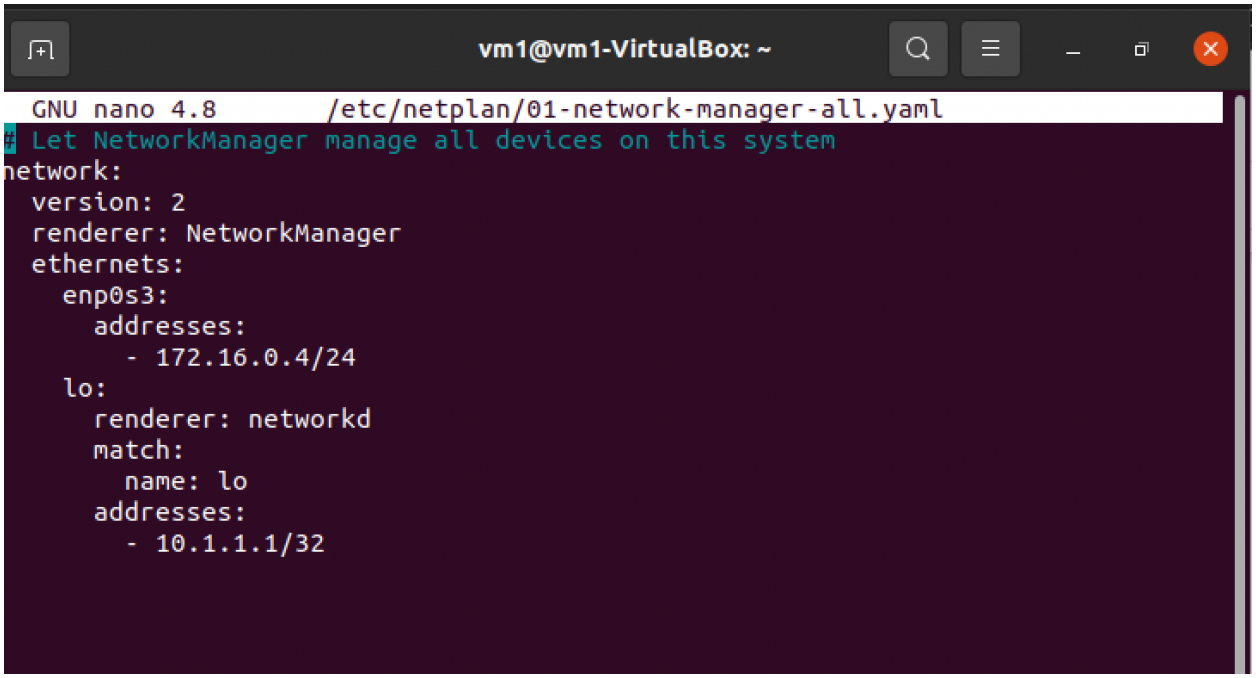
You are done. The connectivity is set up between the machines.

Task 3 : Assign IPs to loopback and recreate #1 and #2 via BGP via FRR. VM1 lo: 10.1.1.1 /32 , VM2 lo:10.1.1.2 /32 , VM3 lo : 10.1.1.3 /32 , ping should work between the loopback addresses. The network range for BGP IPs is 172.16.0.0 /24 for both the scenarios.

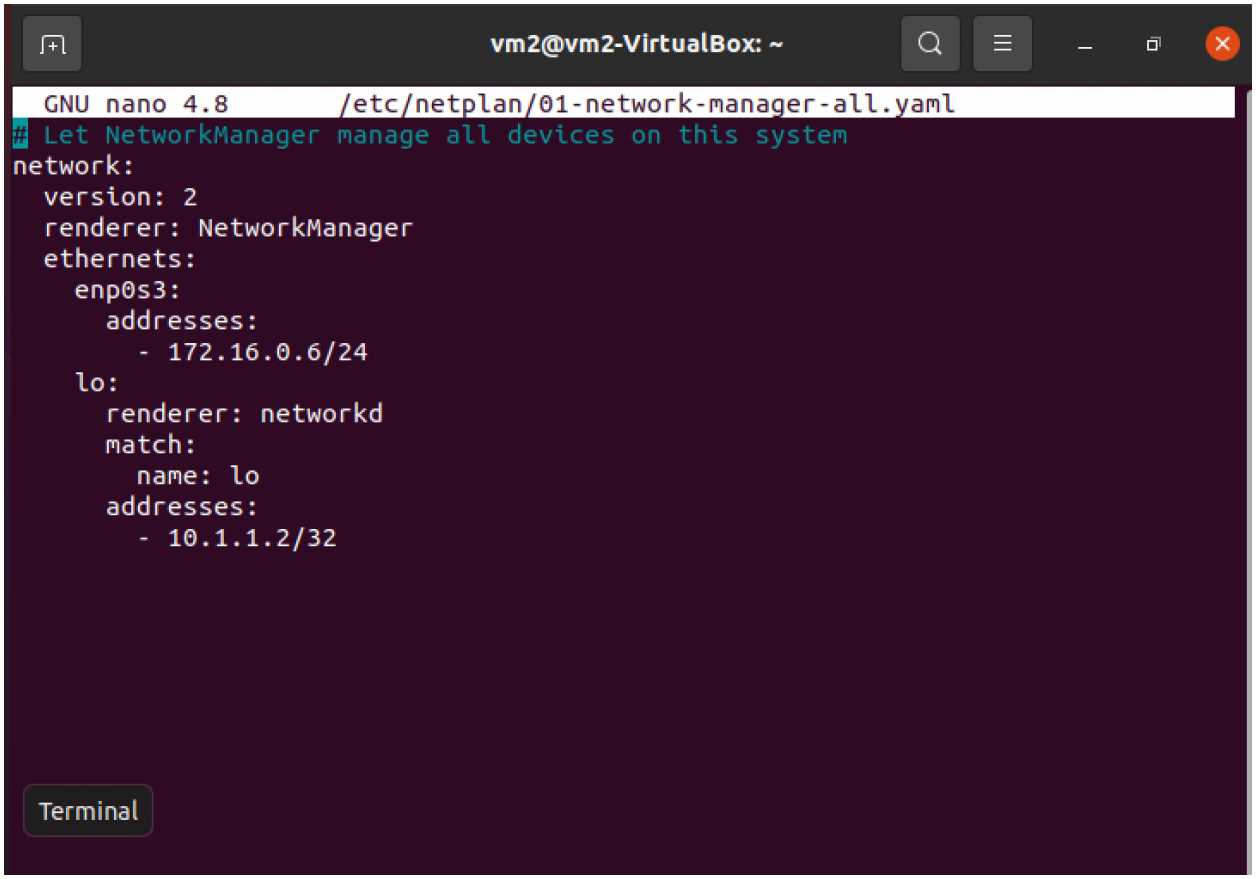
Recreating one:

1. Setup two virtual machines and connect them to a NAT network.
2. Open the VM's and assign the IP addresses to them .

VM1:


vm1@vm1-VirtualBox: ~
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
Let NetworkManager manage all devices on this system
network:
version: 2
renderer: NetworkManager
ethernets:
enp0s3:
addresses:
- 172.16.0.4/24
lo:
renderer: networkd
match:
name: lo
addresses:
- 10.1.1.1/32

VM2:


vm2@vm2-VirtualBox: ~
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
Let NetworkManager manage all devices on this system
network:
version: 2
renderer: NetworkManager
ethernets:
enp0s3:
addresses:
- 172.16.0.6/24
lo:
renderer: networkd
match:
name: lo
addresses:
- 10.1.1.2/32
Terminal

3. Install frr on both the VM's now.

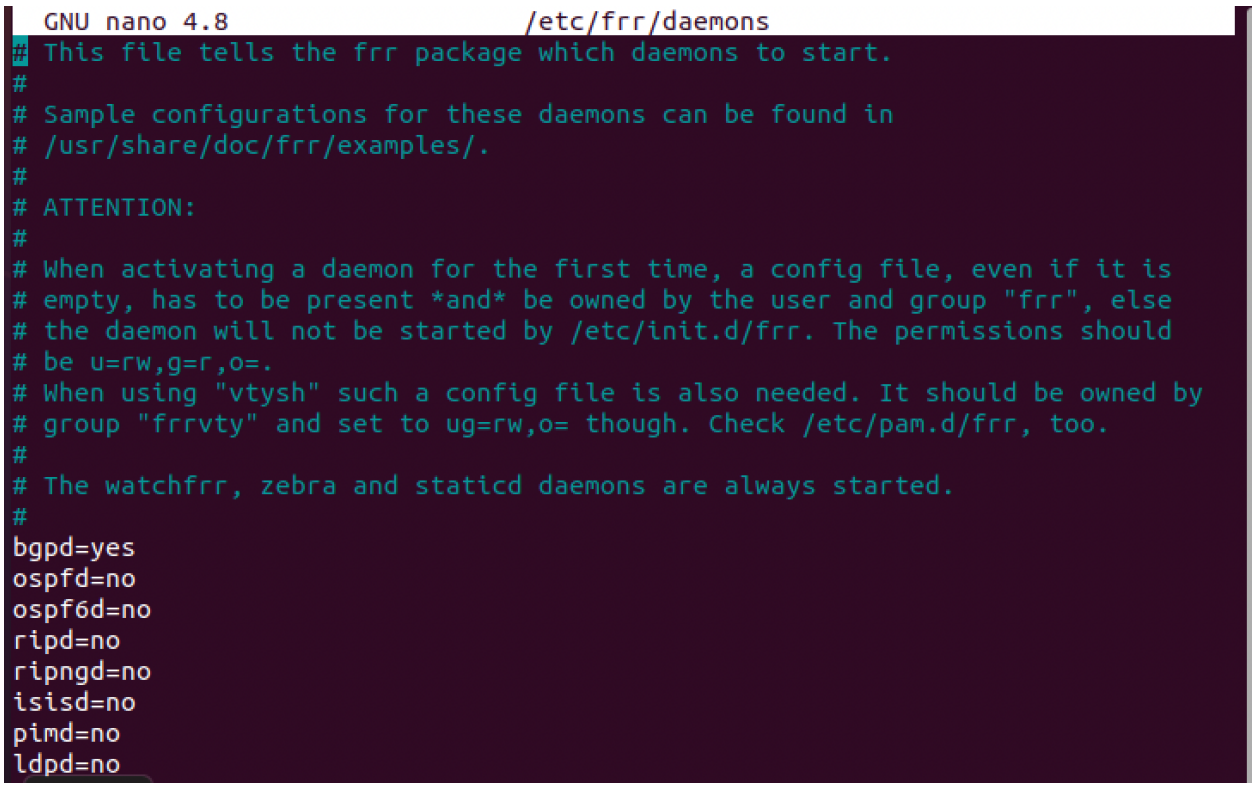
   Curl -s https://deb.frrouting.org/frr/keys.asc | sudo apt-key add -

   FRRVER = "frr-stable"

   Echo deb https://deb.frrouting.org/frr $(lsb_release -s -c) $FRRVER | sudo tee -a /etc/apt/sources.list.d/frr.list

   sudo apt update && sudo apt install frr frr-pythontools

4. Enable bgp in the /etc/frr/daemons and restart the systemctl
   sudo nano /etc/frr/daemons



5. Enter the vtysh shell using sudo vtysh command.

6. Do the FRR configuration.

```
vm1@vm1-VirtualBox:~$ sudo systemctl restart frr
vm1@vm1-VirtualBox:~$ sudo vtysh

Hello, this is FRRouting (version 8.1).
Copyright 1996-2005 Kunihiro Ishiguro, et al.

vm1-VirtualBox# conf t
vm1-VirtualBox(config)# router bgp 100
vm1-VirtualBox(config-router)# neighbor 172.16.0.6 remote-as 200
vm1-VirtualBox(config-router)# network 10.1.1.1/32
vm1-VirtualBox(config-router)# neighbor 172.16.0.6 prefix-list machines in
vm1-VirtualBox(config-router)# neighbor 172.16.0.6 prefix-list machines out
vm1-VirtualBox(config-router)# exit
vm1-VirtualBox(config)# ip prefix-list machines permit 0.0.0.0/0 ge 32 le 32
vm1-VirtualBox(config)# write memory
% Unknown command: write memory
vm1-VirtualBox(config)# exit
vm1-VirtualBox# write memory
Note: this version of vtysh never writes vtysh.conf
Building Configuration...
Integrated configuration saved to /etc/frr/frr.conf
[OK]
vm1-VirtualBox# sh run
Building configuration...
```

7. Do the same steps for VM2

```
  GNU nano 4.8                    /etc/frr/frr.conf
frr version 8.1
frr defaults traditional
hostname vm2-VirtualBox
log syslog informational
no ip forwarding
no ipv6 forwarding
service integrated-vtysh-config
!
router bgp 200
 neighbor 172.16.0.4 remote-as 100
 !
 address-family ipv4 unicast
  network 10.1.1.2/32
  neighbor 172.16.0.4 prefix-list machines in
  neighbor 172.16.0.4 prefix-list machines out
 exit-address-family
exit
!
ip prefix-list machines seq 5 permit 0.0.0.0/0 ge 32 le 32
!
```

8. Try to ping over the loopback address on both the machine now.

```
vm1@vm1-VirtualBox:~$ ip route show
10.1.1.2 nhid 10 via 172.16.0.6 dev enp0s3 proto bgp metric 20
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.0.0/24 dev enp0s3 proto kernel scope link src 172.16.0.4 metric 100
```

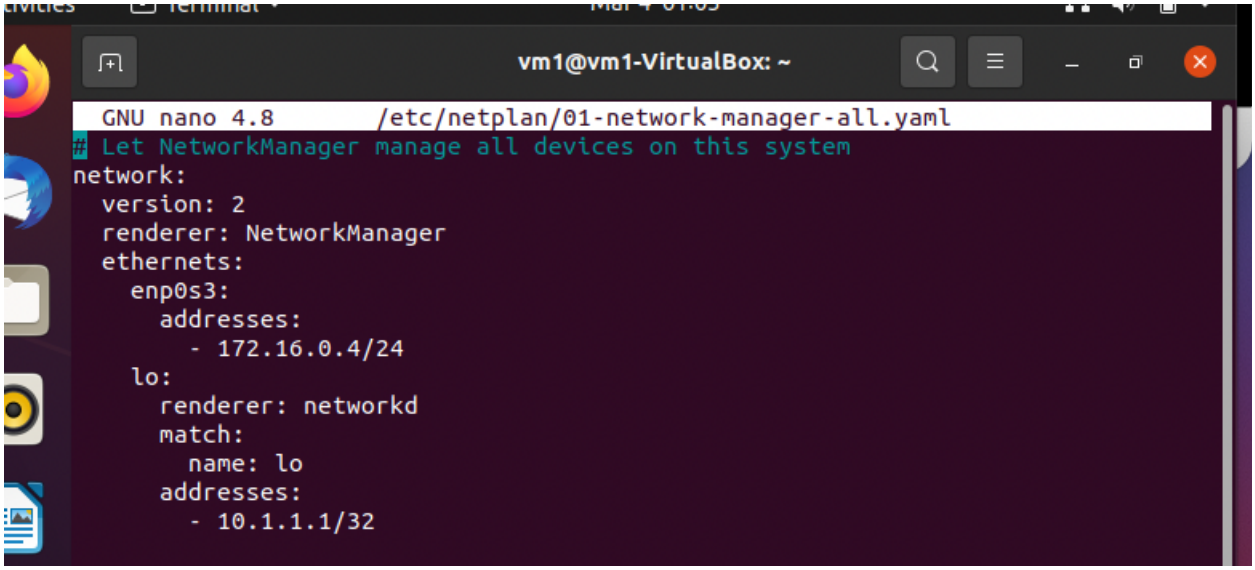
```
vm2@vm2-VirtualBox:~$ ip route show
10.1.1.1 nhid 8 via 172.16.0.4 dev enp0s3 proto bgp metric 20
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.0.0/24 dev enp0s3 proto kernel scope link src 172.16.0.6 metric 100
```

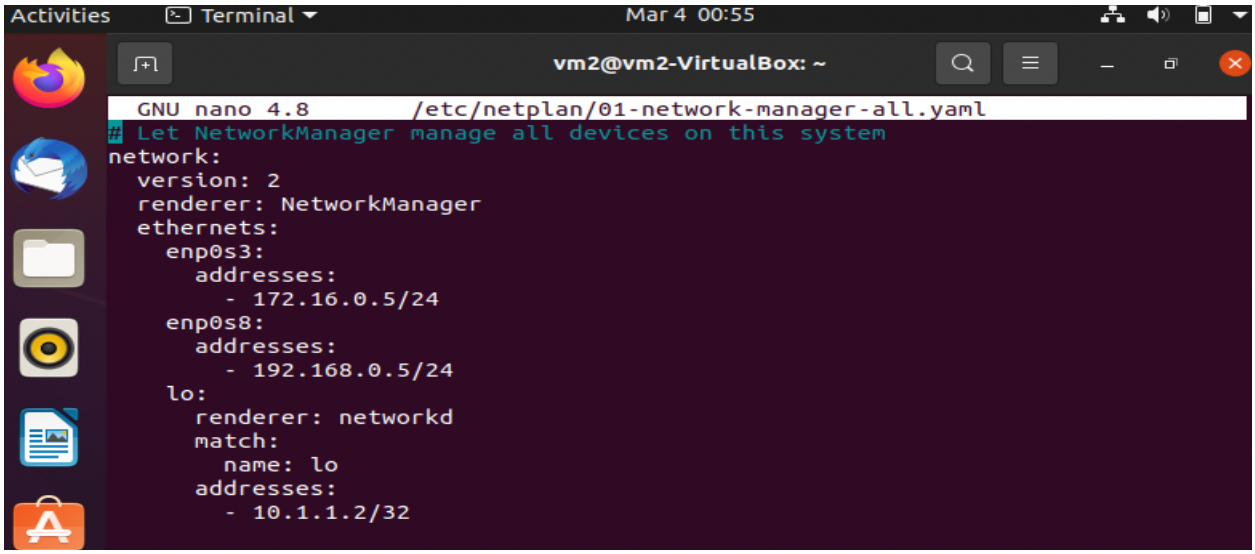
```
vm1@vm1-VirtualBox:~$ ping 10.1.1.2
PING 10.1.1.2 (10.1.1.2) 56(84) bytes of data.
64 bytes from 10.1.1.2: icmp_seq=1 ttl=64 time=0.338 ms
64 bytes from 10.1.1.2: icmp_seq=2 ttl=64 time=0.430 ms
64 bytes from 10.1.1.2: icmp_seq=3 ttl=64 time=0.416 ms
^X64 bytes from 10.1.1.2: icmp_seq=4 ttl=64 time=0.480 ms
^Z
[1]+  Stopped                 ping 10.1.1.2
```

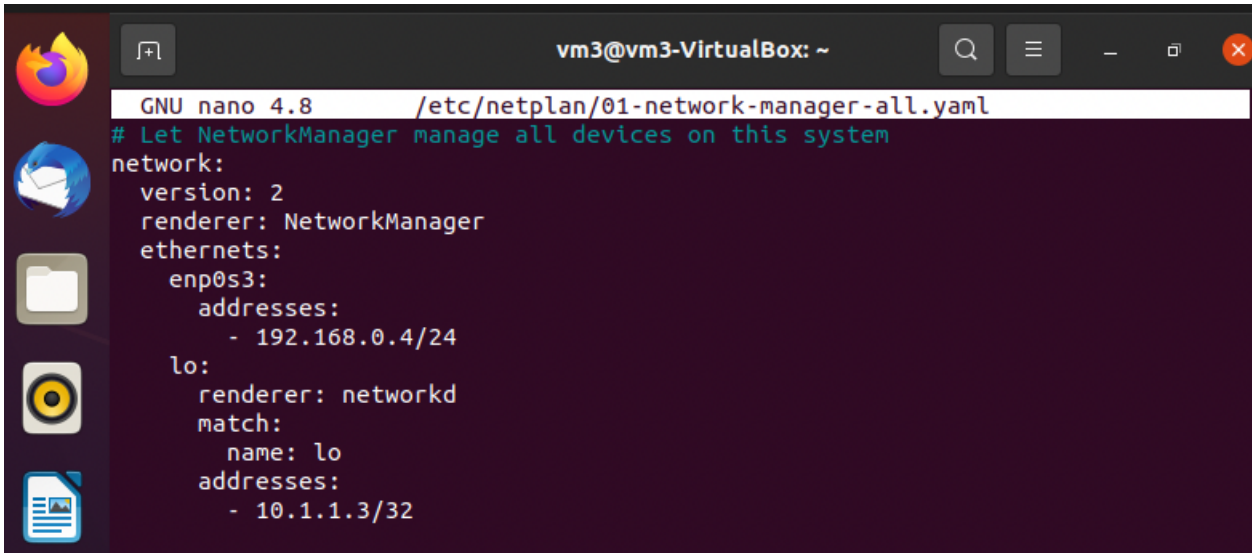
```
vm2@vm2-VirtualBox:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=64 time=0.575 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=64 time=0.393 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=64 time=0.386 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=64 time=0.645 ms
^Z
[ Show Applications           ping 10.1.1.1
```

Recreating 2:

1. Set up 3 virtual machines.
2. Install FRR and enable bgp on all the 3 virtual machines as shown above.
3. Add loopback addresses and Bgp ip’s to the machines.

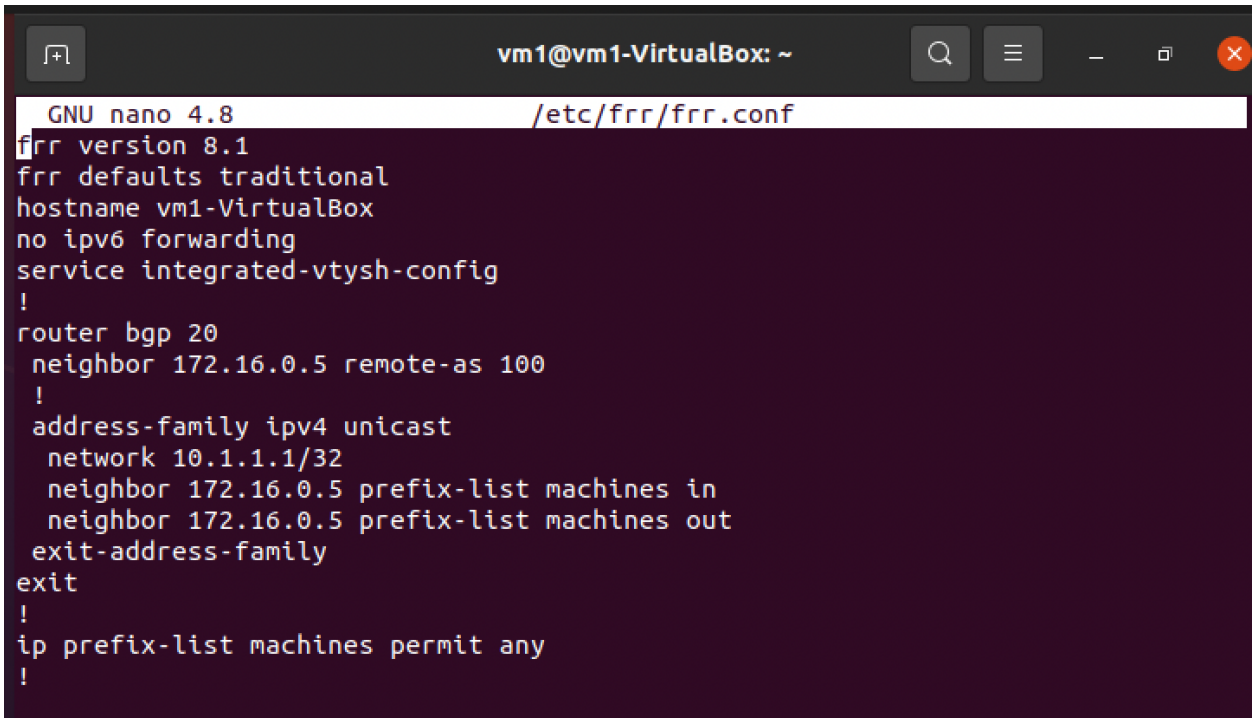

vm1@vm1-VirtualBox: ~
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
Let NetworkManager manage all devices on this system
network:
version: 2
renderer: NetworkManager
ethernets:
enp0s3:
addresses:
- 172.16.0.4/24
lo:
renderer: networkd
match:
name: lo
addresses:
- 10.1.1.1/32


Activities
Terminal
Mar 4 00:55
vm2@vm2-VirtualBox: ~
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
Let NetworkManager manage all devices on this system
network:
version: 2
renderer: NetworkManager
ethernets:
enp0s3:
addresses:
- 172.16.0.5/24
enp0s8:
addresses:
- 192.168.0.5/24
lo:
renderer: networkd
match:
name: lo
addresses:
- 10.1.1.2/32

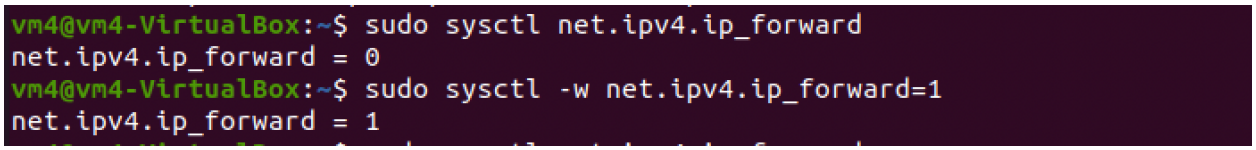

vm3@vm3-VirtualBox: ~
GNU nano 4.8 /etc/netplan/01-network-manager-all.yaml
Let NetworkManager manage all devices on this system
network:
version: 2
renderer: NetworkManager
ethernets:
enp0s3:
addresses:
- 192.168.0.4/24
lo:
renderer: networkd
match:
name: lo
addresses:
- 10.1.1.3/32

4. Restart the frr service
5. Now configure the connection via bgp using frr with the help of vtysh shell.

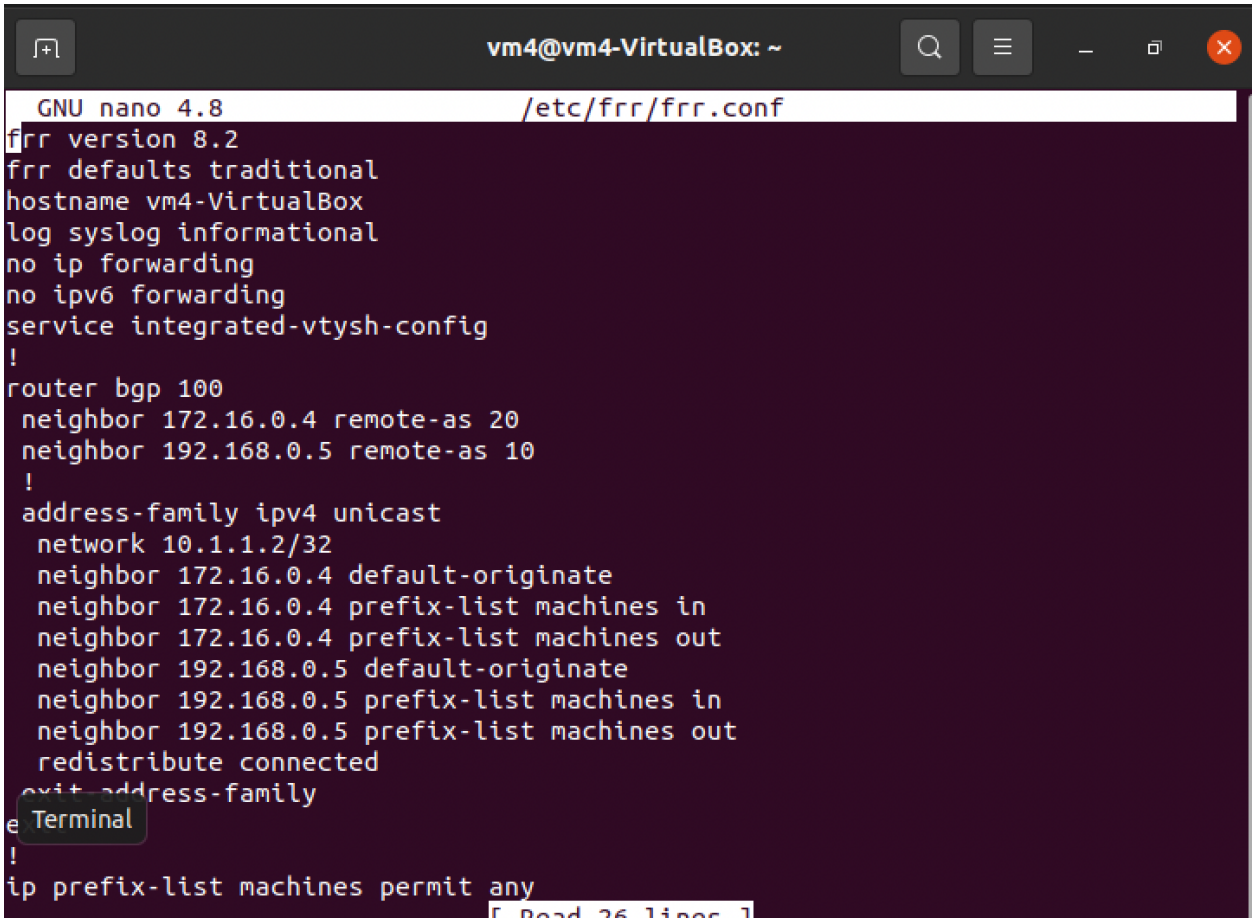
VM1:



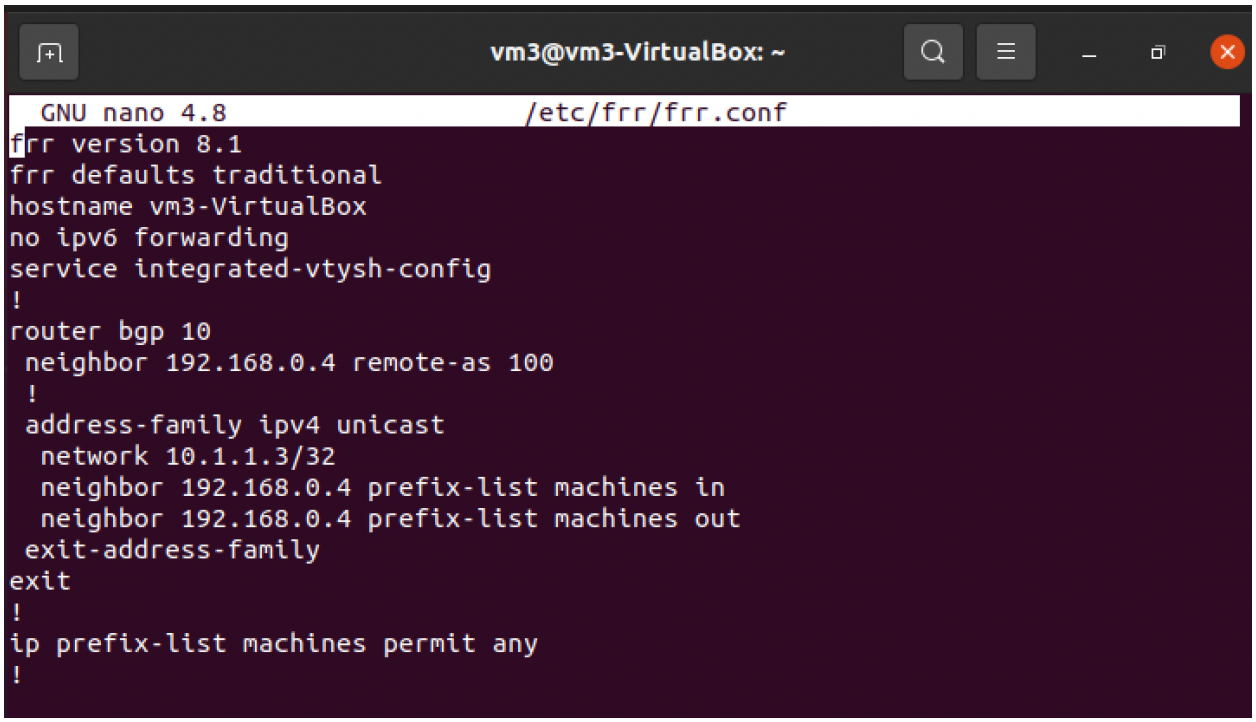
VM2:



Enable Ipv4 on the routing machine using these commands.


vm4@vm4-VirtualBox: ~
GNU nano 4.8 /etc/frr/frr.conf
frr version 8.2
frr defaults traditional
hostname vm4-VirtualBox
log syslog informational
no ip forwarding
no ipv6 forwarding
service integrated-vtysh-config
!
router bgp 100
neighbor 172.16.0.4 remote-as 20
neighbor 192.168.0.5 remote-as 10
!
address-family ipv4 unicast
network 10.1.1.2/32
neighbor 172.16.0.4 default-originate
neighbor 172.16.0.4 prefix-list machines in
neighbor 172.16.0.4 prefix-list machines out
neighbor 192.168.0.5 default-originate
neighbor 192.168.0.5 prefix-list machines in
neighbor 192.168.0.5 prefix-list machines out
redistribute connected
exit-address-family
Terminal
!
ip prefix-list machines permit any

VM3:


vm3@vm3-VirtualBox: ~
GNU nano 4.8 /etc/frr/frr.conf
frr version 8.1
frr defaults traditional
hostname vm3-VirtualBox
no ipv6 forwarding
service integrated-vtysh-config
!
router bgp 10
neighbor 192.168.0.4 remote-as 100
!
address-family ipv4 unicast
network 10.1.1.3/32
neighbor 192.168.0.4 prefix-list machines in
neighbor 192.168.0.4 prefix-list machines out
exit-address-family
exit
!
ip prefix-list machines permit any
!

6. Check the configuration that if the loopback address is advertised via bgp or not using ip route show.
7. Try pinging the Vm's.

```
vm1@vm1-VirtualBox:~$ sudo systemctl restart frr
vm1@vm1-VirtualBox:~$ ip r s
default nhid 10 via 172.16.0.5 dev enp0s3 proto bgp metric 20
10.1.1.2 nhid 10 via 172.16.0.5 dev enp0s3 proto bgp metric 20
10.1.1.3 nhid 10 via 172.16.0.5 dev enp0s3 proto bgp metric 20
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.0.0/24 dev enp0s3 proto kernel scope link src 172.16.0.4 metric 100
192.168.0.0/24 via 172.16.0.5 dev enp0s3
```

```
vm4@vm4-VirtualBox:~$ ip r s
10.1.1.1 nhid 104 via 172.16.0.4 dev enp0s3 proto bgp metric 20
10.1.1.3 nhid 102 via 192.168.0.5 dev enp0s8 proto bgp metric 20
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.0.0/24 dev enp0s3 proto kernel scope link src 172.16.0.5 metric 100
192.168.0.0/24 dev enp0s8 proto kernel scope link src 192.168.0.4 metric 101
```

```
vm3@vm3-VirtualBox:~$ sudo systemctl restart frr
vm3@vm3-VirtualBox:~$ ip r s
default nhid 10 via 192.168.0.4 dev enp0s3 proto bgp metric 20
10.1.1.1 nhid 10 via 192.168.0.4 dev enp0s3 proto bgp metric 20
10.1.1.2 nhid 10 via 192.168.0.4 dev enp0s3 proto bgp metric 20
169.254.0.0/16 dev enp0s3 scope link metric 1000
172.16.0.0/24 via 192.168.0.4 dev enp0s3
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.5 metric 100
```

```
vm1@vm1-VirtualBox:~$ ping 10.1.1.3
PING 10.1.1.3 (10.1.1.3) 56(84) bytes of data.
64 bytes from 10.1.1.3: icmp_seq=1 ttl=63 time=0.582 ms
64 bytes from 10.1.1.3: icmp_seq=2 ttl=63 time=0.608 ms
64 bytes from 10.1.1.3: icmp_seq=3 ttl=63 time=0.826 ms
64 bytes from 10.1.1.3: icmp_seq=4 ttl=63 time=0.800 ms
64 bytes from 10.1.1.3: icmp_seq=5 ttl=63 time=0.549 ms
^Z
[4]+  Stopped                 ping 10.1.1.3
```

```
vm3@vm3-VirtualBox:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=63 time=0.578 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=63 time=0.790 ms
^Z
[5]+  Stopped                 ping 10.1.1.1
vm3@vm3-VirtualBox:~$
```