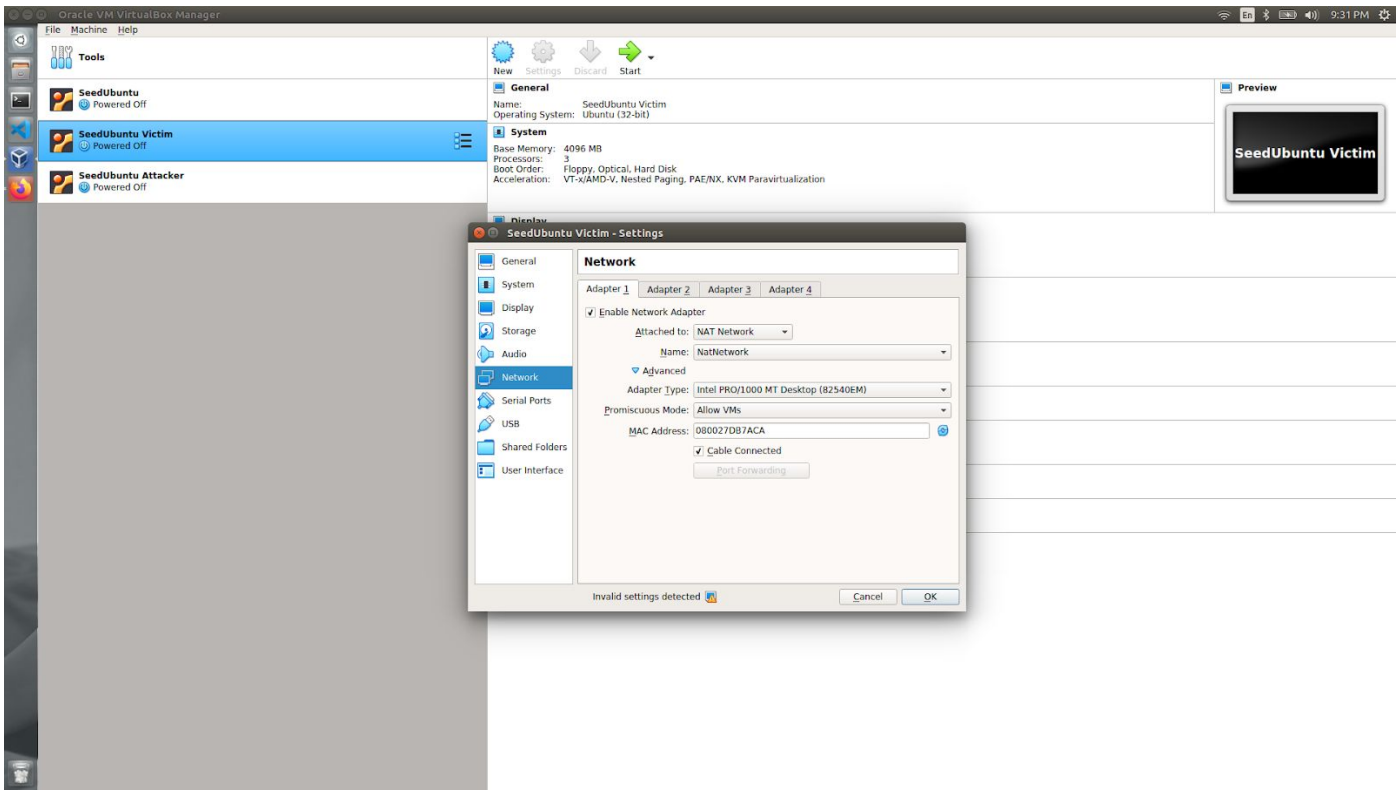
Task 1 : VM Setup:

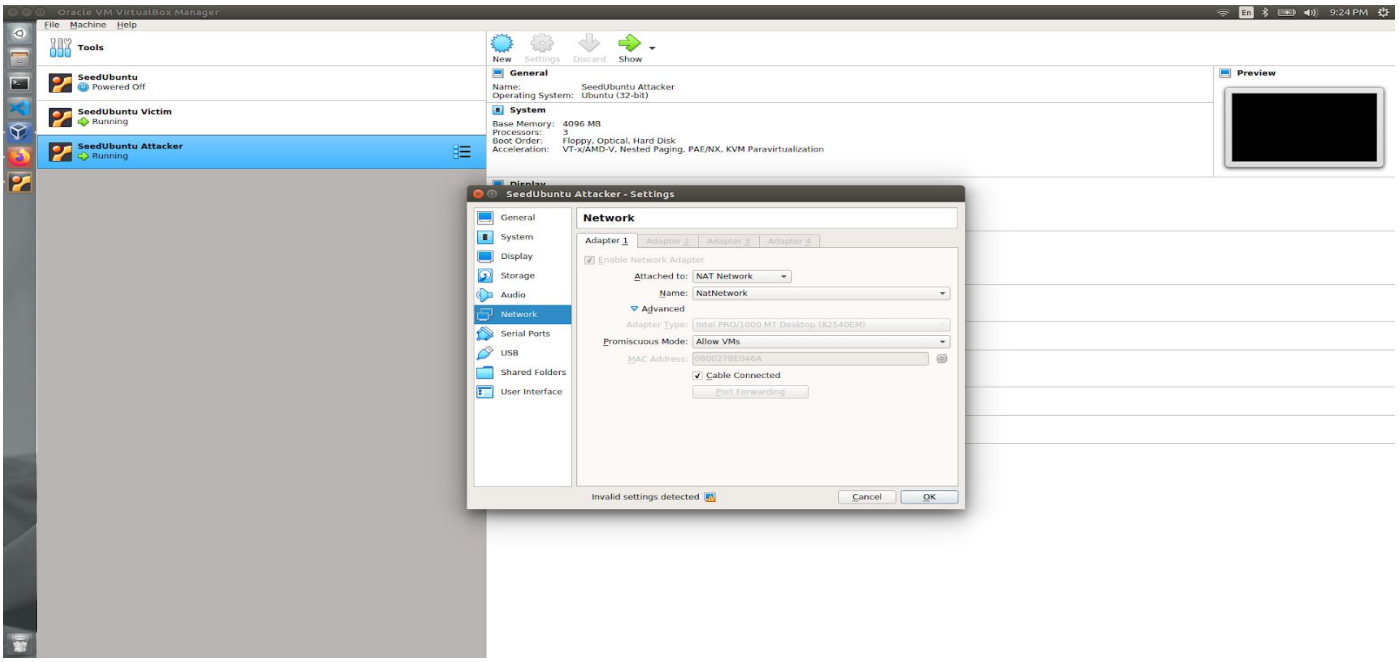
For all the tasks following in this lab, we need a minimum of 2 VM's… VM1 and VM2. These two machines are connected via the internet through the Routers. Here we use a LAN to emulate the internet Connection. Here we connect the two VM's to a LAN using the NAT Network Adapter. The screenshot below shows the setup we've done for these lab tasks.

The configuration of the VM1 which is named **SeedUbuntu Victim** is shown below:



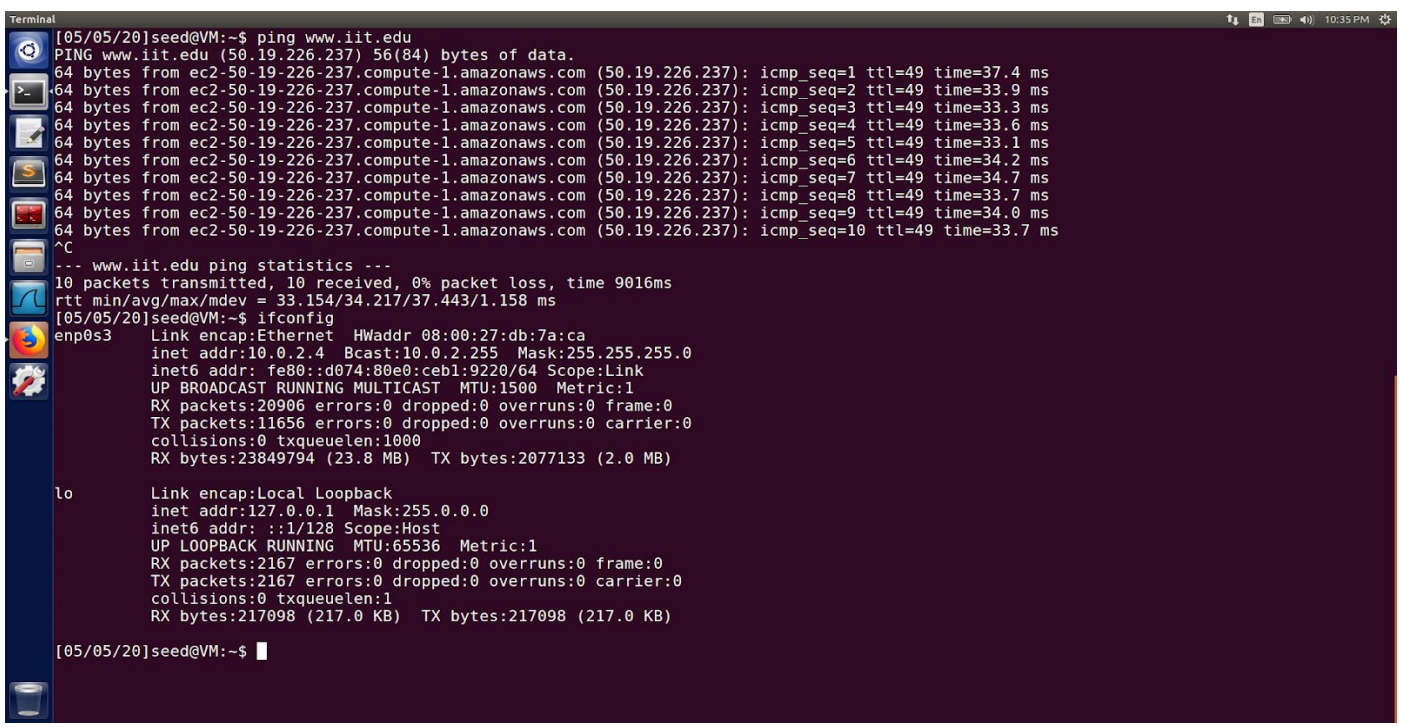
A similar configuration is done to the VM2 which is named as **SeedUbuntu Attacker.** Having NAT adapter set, makes the 2 VM's a part of a LAN.

The configuration of the VM1 which is named **SeedUbuntu Attacker** is shown below:



Task 2 : Set up Firewall:

In this task, we're setting up the firewall rules on VM1 to block the access of a target website. The IP of the target website, www.iit.edu we'll be using is **50.19.226.237.** And the IP of the VM1 that we'll be using is **10.0.2.4.** We can see that from the screenshot below.



```
Terminal
[05/05/20]seed@VM:~$ ping www.iit.edu
PING www.iit.edu (50.19.226.237) 56(84) bytes of data.
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=1 ttl=49 time=37.4 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=2 ttl=49 time=33.9 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=3 ttl=49 time=33.3 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=4 ttl=49 time=33.6 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=5 ttl=49 time=33.1 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=6 ttl=49 time=34.2 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=7 ttl=49 time=34.7 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=8 ttl=49 time=33.7 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=9 ttl=49 time=34.0 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=10 ttl=49 time=33.7 ms
^C
--- www.iit.edu ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9016ms
rtt min/avg/max/mdev = 33.154/34.217/37.443/1.158 ms
[05/05/20]seed@VM:~$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:db:7a:ca
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::d074:80e0:ceb1:9220/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:20906 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11656 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:23849794 (23.8 MB)  TX bytes:2077133 (2.0 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:2167 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2167 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:217098 (217.0 KB)  TX bytes:217098 (217.0 KB)

[05/05/20]seed@VM:~$
```

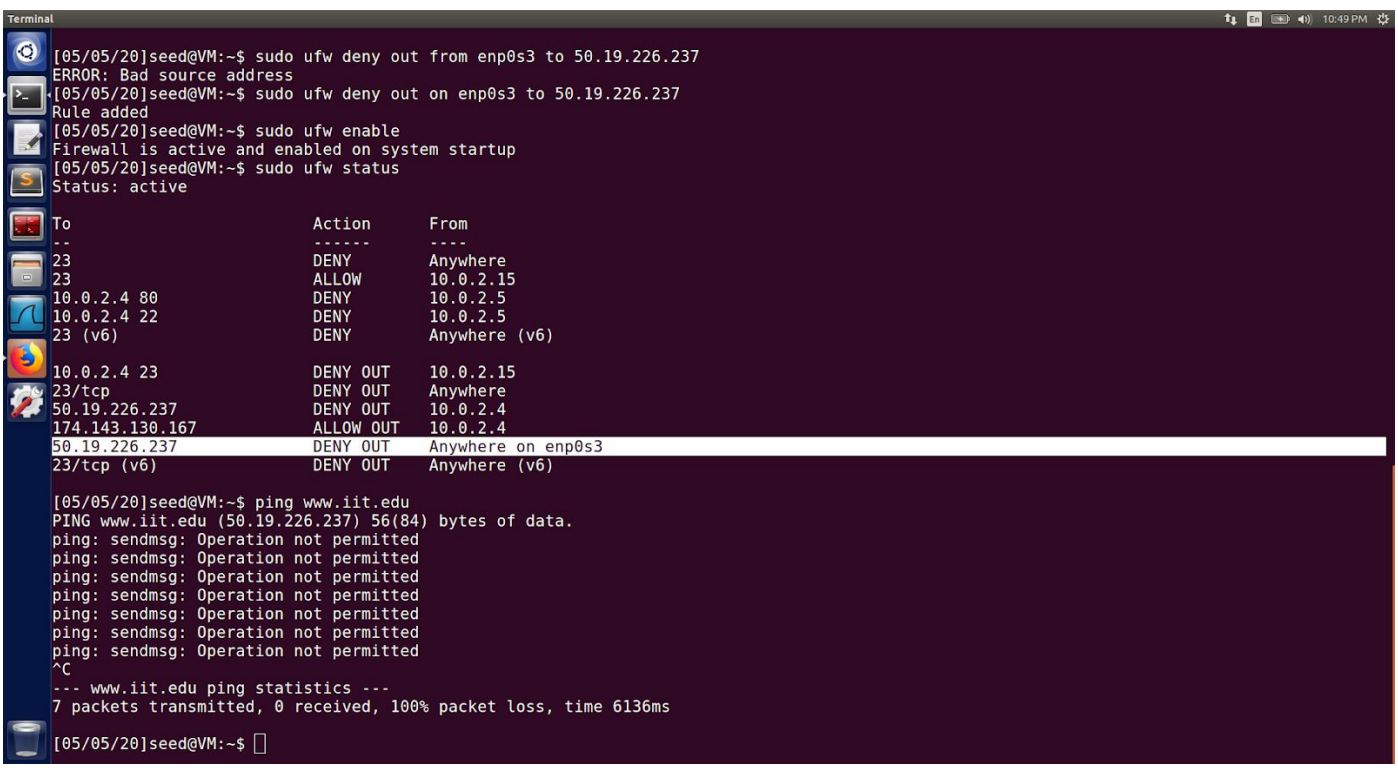
In this task, taking the IP of www.iit.edu we got from the ping command above as **50.19.226.237,** we update the firewall rule by executing the ufw command as shown below.

sudo ufw deny out from 10.0.2.4 to 50.19.226.237

Once the above command is executed, the above firewall rule is updated. But this updated rule has to be activated. To activate the updated firewall rules, we execute the following command.

sudo ufw enable

Once the firewall rules are activated, I now execute the ping command with www.iit.edu as an argument to the ping command. Since the firewall is updated, we can see the ping to www.iit.edu fails as an access to www.iit.edu is blocked from VM1. We can see all this happening from the screenshot below.



```
[05/05/20]seed@VM:~$ sudo ufw deny out from enp0s3 to 50.19.226.237
ERROR: Bad source address
[05/05/20]seed@VM:~$ sudo ufw deny out on enp0s3 to 50.19.226.237
Rule added
[05/05/20]seed@VM:~$ sudo ufw enable
Firewall is active and enabled on system startup
[05/05/20]seed@VM:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
23                         DENY        Anywhere
23                         ALLOW       10.0.2.15
10.0.2.4 80                DENY        10.0.2.5
10.0.2.4 22                DENY        10.0.2.5
23 (v6)                    DENY        Anywhere (v6)

10.0.2.4 23                DENY OUT    10.0.2.15
23/tcp                     DENY OUT    Anywhere
50.19.226.237              DENY OUT    10.0.2.4
174.143.130.167            ALLOW OUT   10.0.2.4
50.19.226.237              DENY OUT    Anywhere on enp0s3
23/tcp (v6)                DENY OUT    Anywhere (v6)

[05/05/20]seed@VM:~$ ping www.iit.edu
PING www.iit.edu (50.19.226.237) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
^C
--- www.iit.edu ping statistics ---
7 packets transmitted, 0 received, 100% packet loss, time 6136ms

[05/05/20]seed@VM:~$
```

From the highlighted section from the screenshot we can see that access to **50.19.226.237** is denied from anywhere on the **enp0s3** interface. And the ping to www.iit.edu fails as in the output we can see the operation not being permitted.

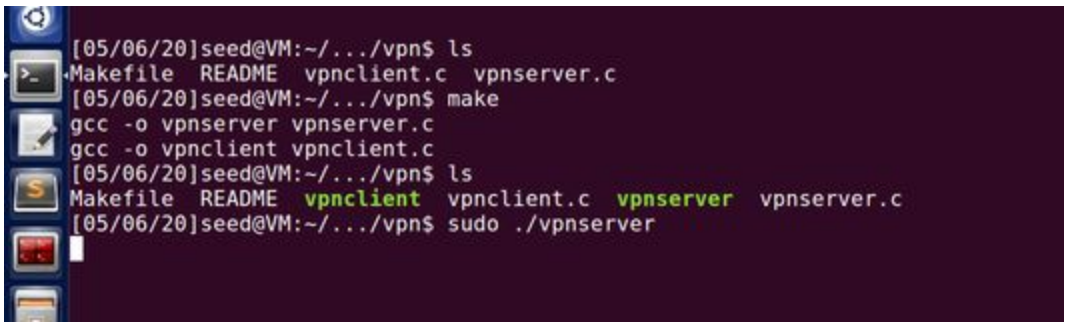
Task 3: Bypassing Firewall using VPN

For this task, we make use of a VPN tunnel by executing the client and server programs downloaded from the lab website.

Step 1: Run VPN Server.

Here for this task, I chose VM2 as a server and ran vpnserver program. After we execute the make command, it generates the 2 executables vpnclient and vpnserver.

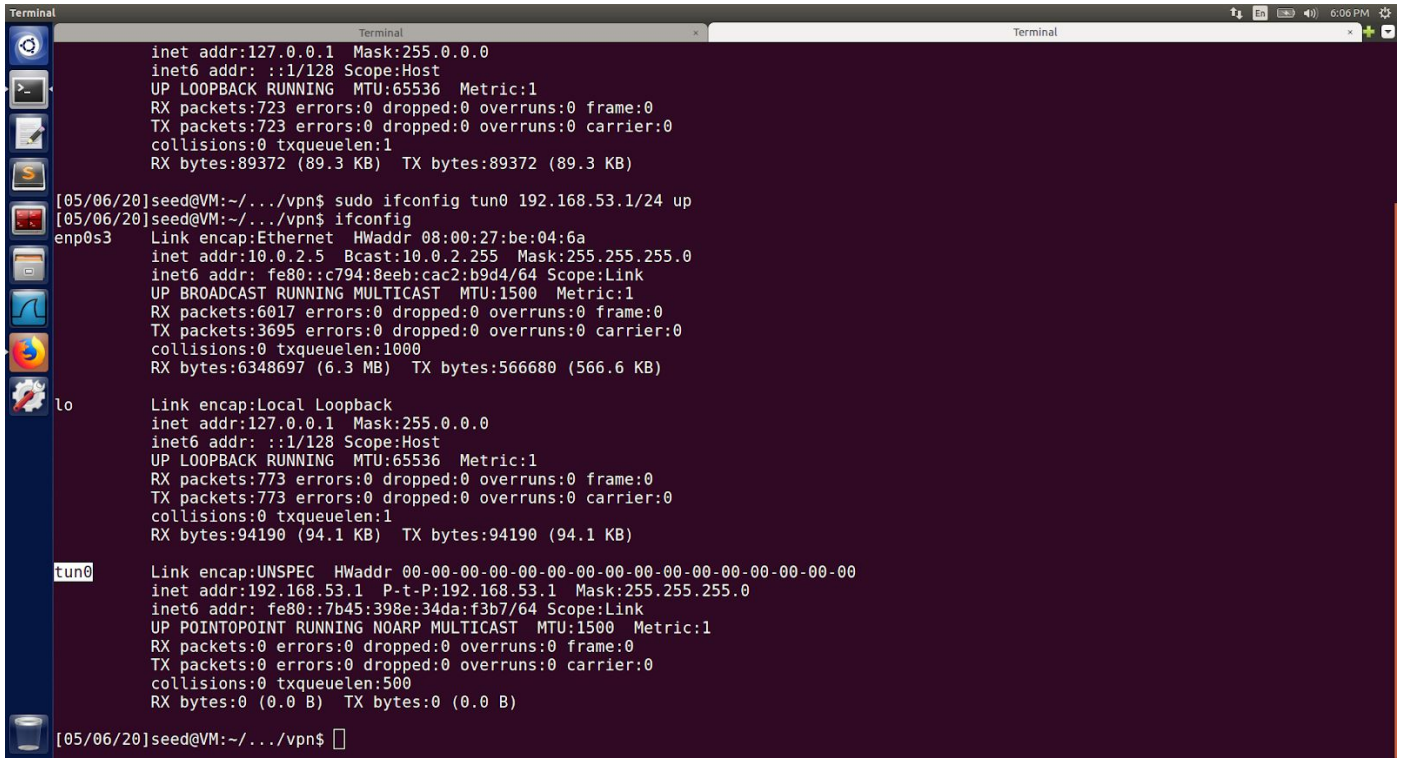
On the VM2 I ran vpnserver executable with sudo



```
[05/06/20]seed@VM:~/.../vpn$ ls
Makefile  README  vpnclient.c  vpnserver.c
[05/06/20]seed@VM:~/.../vpn$ make
gcc -o vpnserver vpnserver.c
gcc -o vpnclient vpnclient.c
[05/06/20]seed@VM:~/.../vpn$ ls
Makefile  README  vpnclient  vpnclient.c  vpnserver  vpnserver.c
[05/06/20]seed@VM:~/.../vpn$ sudo ./vpnserver
```

Now, in another tab, I configure the new interface tun0 and gave it an IP 192.168.53.1.

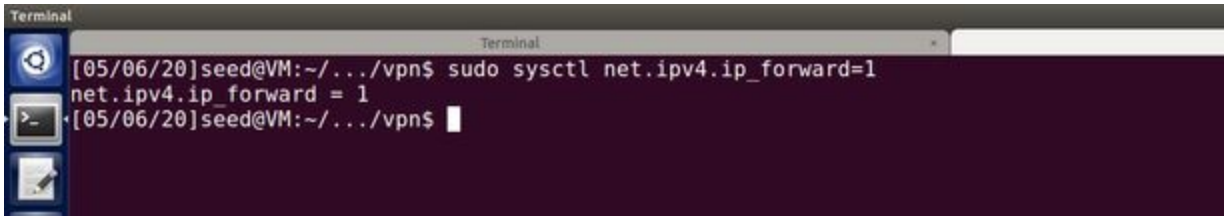
sudo ifconfig tun0 192.168.53.1/24 up is executed and here is how it looks.



```
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:723 errors:0 dropped:0 overruns:0 frame:0
          TX packets:723 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:89372 (89.3 KB)  TX bytes:89372 (89.3 KB)

[05/06/20]seed@VM:~/.../vpn$ sudo ifconfig tun0 192.168.53.1/24 up
[05/06/20]seed@VM:~/.../vpn$ ifconfig
enp0s3    Link encap:Ethernet  HWaddr 08:00:27:be:04:6a
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::c794:8eeb:cac2:b9d4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:6017 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3695 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:6348697 (6.3 MB)  TX bytes:566680 (566.6 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:773 errors:0 dropped:0 overruns:0 frame:0
          TX packets:773 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:94190 (94.1 KB)  TX bytes:94190 (94.1 KB)

tun0      Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:192.168.53.1  P-t-P:192.168.53.1  Mask:255.255.255.0
          inet6 addr: fe80::7b45:398e:34da:f3b7/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

[05/06/20]seed@VM:~/.../vpn$
```

Now that the interface is configured, unless specified this computer will only act as a host. We need to configure it in such a way that it has to act as a gateway so that it can forward packets to other destinations. We need to enable IP forwarding to make it behave like a gateway. We run this command in order to make it a gateway:

sudo sysctl net.ipv4.ip_forward=1

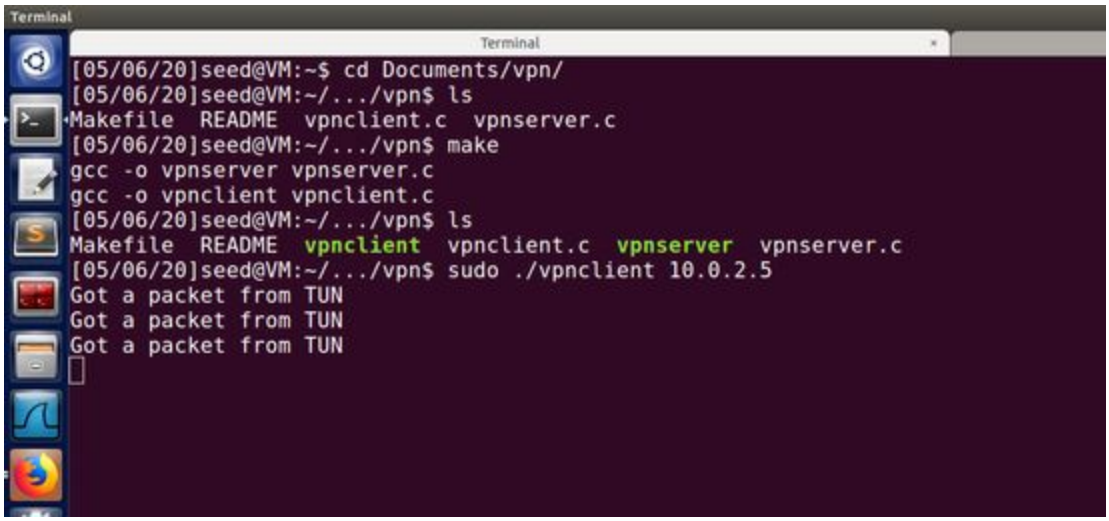


<u>Step 2: Run VPN client.</u>

For this, I chose VM1 as a client and ran a VPN client program on the VM2. We execute the following command

sudo ./vpnclient 10.0.2.5

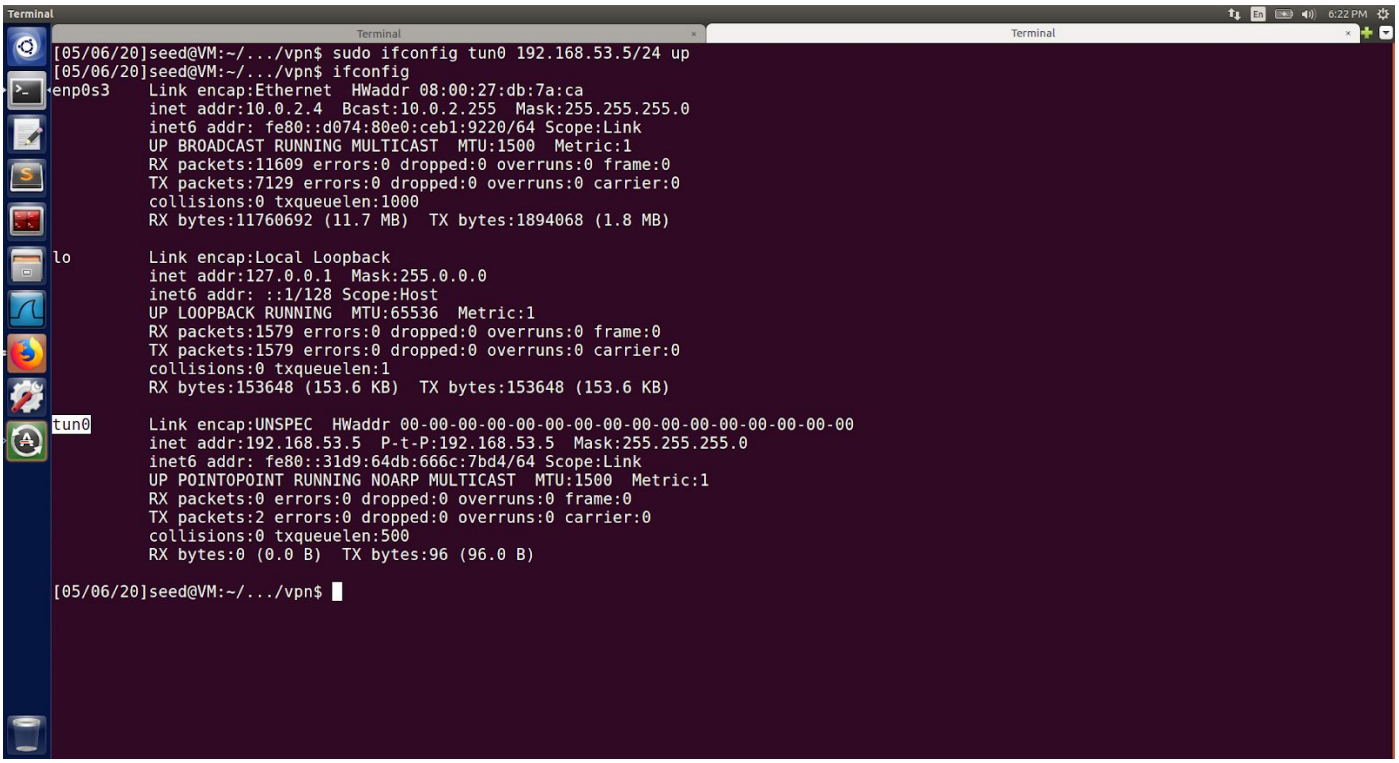
Here 10.0.2.5 is the ip of the machine on which the Server program is running.



Now in another tab, I configured the IP to the tun0 interface by executing the following command:

sudo ifconfig tun0 192.168.53.5/24 up

Once we've configured the IP to this interface, here is how it looks.

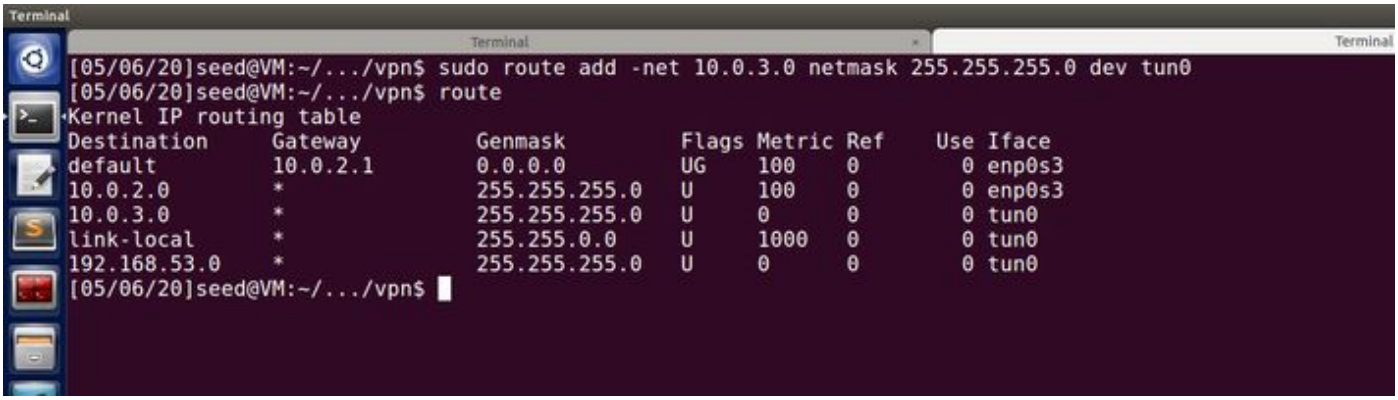


<u>Step 3 : Setup Routing on Client and Server VMs</u>

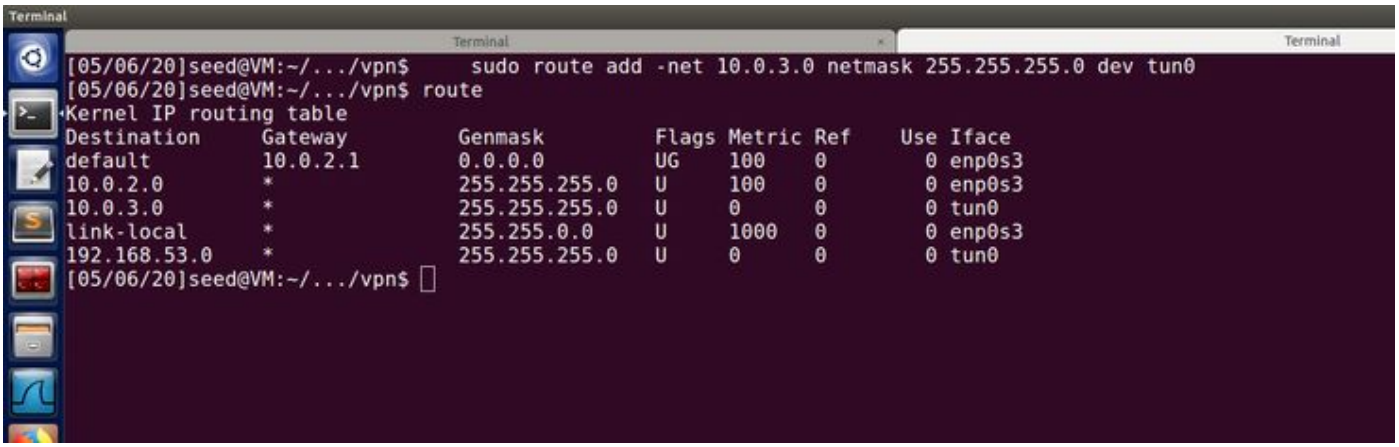
On completing the above 2 steps, a tunnel is not established between the Client and the Server. We now need to set up the routing paths between the client and the server to direct the intended traffic through the tunnel. We make use of the Route command to add a routing entry. **sudo route add -net 10.0.3.0 netmask 255.255.255.0 dev tun0**

The same command is run on Server side as well. We can see from the screenshots below.

On Client Side:

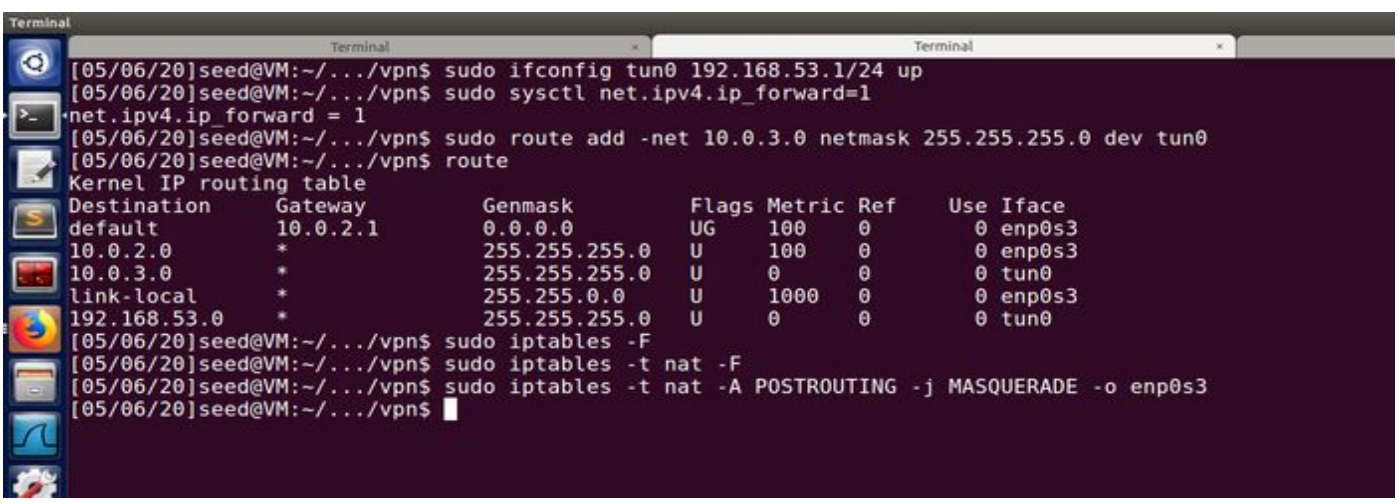


On Server Side:



<u>Step 4: Set Up NAT on Server VM.</u>

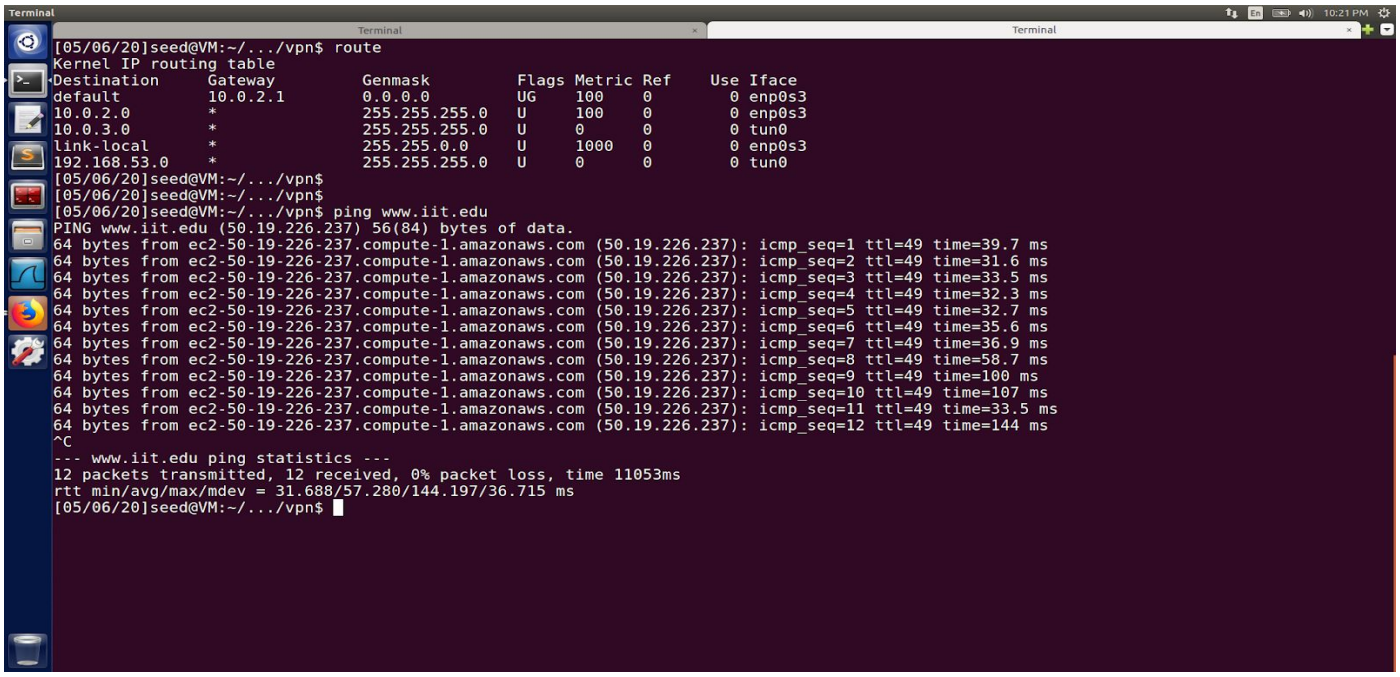
When the final destination sends packets back to users, the packet will be sent to the VPN Server first. **This is because the, since the firewall on the client that makes the calls to the www.iit.edu is configured in such a way that it blocks, all the connections from the Client are redirected to the Server through the tunnel and the call to www.iit.edu will be made from the server on behalf of the Client. And conventionally, the response from the destination hits the source which in our case is Server program on behalf of the Client.** We'll now be executing the set of commands shown in the below screenshot will setup IP forwarding and then clears the iptables rules and then adds a rule on postrouting position to the natnetwork adapter connected to the VPN server.



In our case the interface that we're using is enp0s3.

Doing this the packets received at the server end is not meant to be at the server. They must be forwarded. This is why we set up IP forwarding. We then explore the limitation of NAT by adding a rule in the POSTROUTING position to the NAT network adapter connected to the VPN server.

Now that the tunnel is set, configured and the route is also configured, we now ping www.iit.edu from the client machine. We can see from the screenshot below how the ping is successful.



```
[05/06/20]seed@VM:~/.../vpn$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         10.0.2.1        0.0.0.0         UG    100    0        0 enp0s3
10.0.2.0        *               255.255.255.0   U     100    0        0 enp0s3
10.0.3.0        *               255.255.255.0   U     0      0        0 tun0
link-local      *               255.255.0.0     U     1000   0        0 enp0s3
192.168.53.0    *               255.255.255.0   U     0      0        0 tun0
[05/06/20]seed@VM:~/.../vpn$
[05/06/20]seed@VM:~/.../vpn$
[05/06/20]seed@VM:~/.../vpn$ ping www.iit.edu
PING www.iit.edu (50.19.226.237) 56(84) bytes of data.
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=1 ttl=49 time=39.7 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=2 ttl=49 time=31.6 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=3 ttl=49 time=33.5 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=4 ttl=49 time=32.3 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=5 ttl=49 time=32.7 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=6 ttl=49 time=35.6 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=7 ttl=49 time=36.9 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=8 ttl=49 time=58.7 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=9 ttl=49 time=100 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=10 ttl=49 time=107 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=11 ttl=49 time=33.5 ms
64 bytes from ec2-50-19-226-237.compute-1.amazonaws.com (50.19.226.237): icmp_seq=12 ttl=49 time=144 ms
^C
--- www.iit.edu ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11053ms
rtt min/avg/max/mdev = 31.688/57.280/144.197/36.715 ms
[05/06/20]seed@VM:~/.../vpn$
```

When we ping www.iit.edu the ping to the IP **50.19.226.237** is now successful and we get the response to the Server's IP. The response is sent to the client through the tunnel. The wireshark capture when the ping is executed is shown below.

The ping's ICMP packets to the IP **50.19.226.237** will go through the tun0 interface to the client side of the tunnel. The packets are sent through the tunnel to the server side. The packets are then sent to the destination from the VPN server. Here the packets are actually originated from the Client but then sent by the server on behalf of the server.