Name: Sai Nikhil Boyapati

Personal id: 19981120-9692

Email Id: sainikhil666@gmail.com

ET2595 Network and system security

Lab 1: Linux networking and firewalls

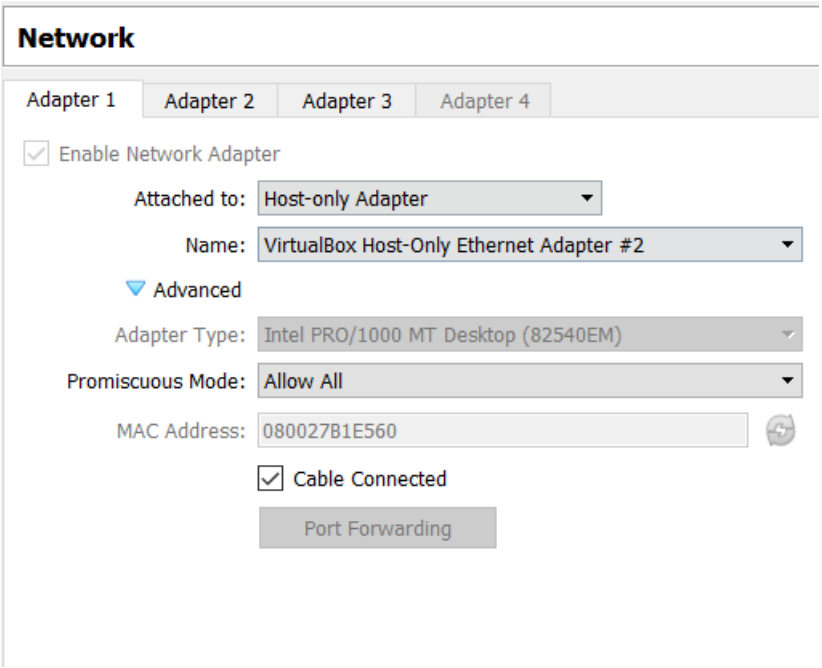
Task 1: MAC addresses:

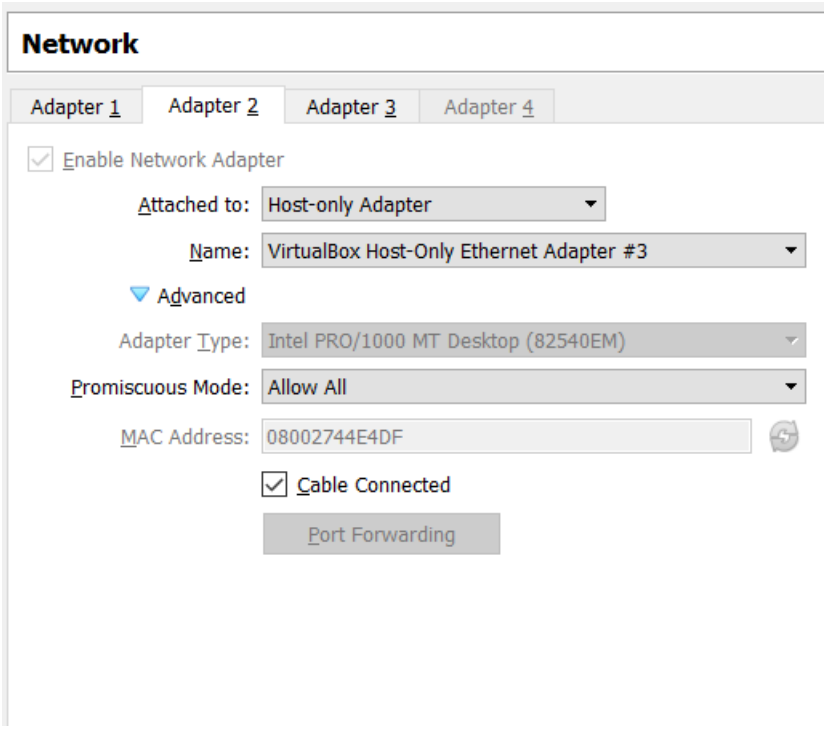
I have identified the MAC address of the configured adapters in the Web Server VM.

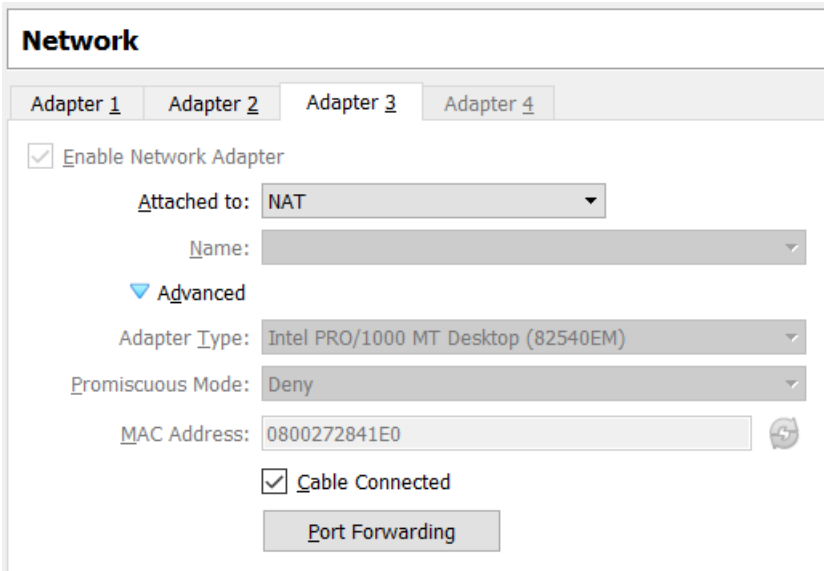
Adapter 1: 080027B1E560

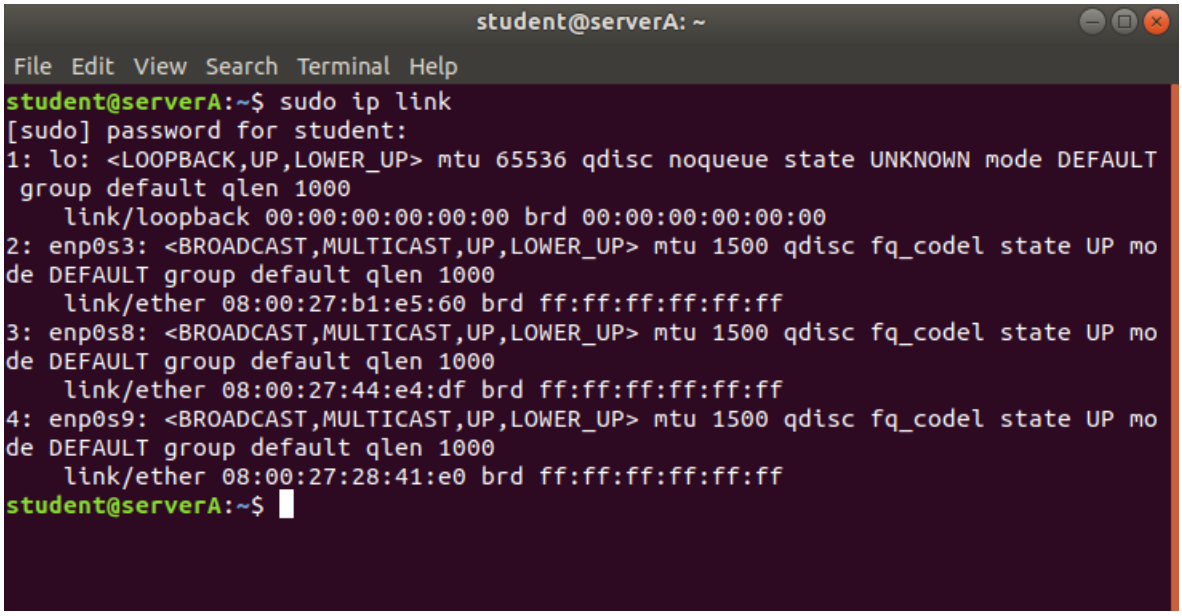
Adapter 2: 08002744E4DF

Adapter 3: 0800272841E0




Network
Adapter 1
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to:
Host-only Adapter
Name:
VirtualBox Host-Only Ethernet Adapter #3
Advanced
Adapter Type:
Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode:
Allow All
MAC Address:
08002744E4DF
Cable Connected
Port Forwarding


Network
Adapter 1
Adapter 2
Adapter 3
Adapter 4
Enable Network Adapter
Attached to:
NAT
Name:
Advanced
Adapter Type:
Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode:
Deny
MAC Address:
0800272841E0
Cable Connected
Port Forwarding



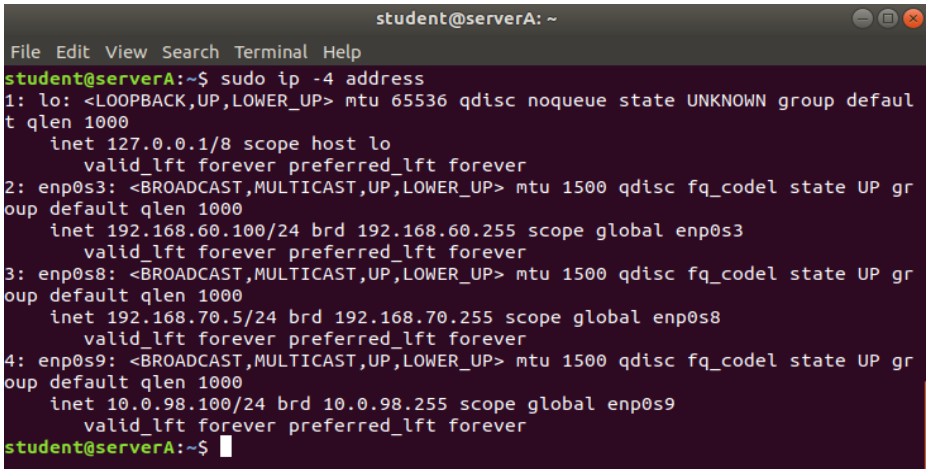
In the above figure I have used the command sudo ip link in which it displays all the interfaces.

Task 2: Network interfaces:

List of network interfaces with MAC addresses in serverA VM can be displayed by using sudo ip link command. The above figure displays the network interfaces.

Host-Only interfaces – enp0s3, enp0s8

NAT interface – enp0s9



TASK - 3: IP addresses, netmasks and subnet:

The command for displaying ip addresses netmasks and subnets is

sudo cat /etc/network/interfaces

The below figure displays IP addresses, netmasks and subnets of different interfaces.

Interface name	**Subnet mask**	**IPV4 address**	**Network address**	**Interface Type**
enp0s9	255.255.255.0	10.0.98.100	10.0.0.98/24	NAT
enp0s3	255.255.255.0	192.168.60.100	192.168.60.0/24	Host only
enp0s8	255.255.255.0	192.168.70.5	192.168.70.0/24	Host only

NAT:

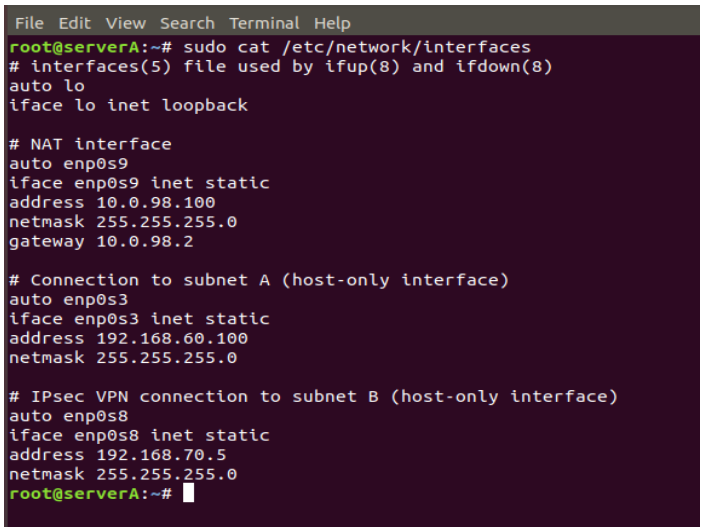
IP address	10.0.98.100	0000 1010 . 0000 0000 . 0110 0010 . 0110 0100
Netmask	255.255.255.0	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
Network address	10.0.98.0	0000 1010 . 0000 0000 . 0110 0010 . 0000 0000

enp0s3:

IP address	192.168.60.100	1100 0000 . 1010 1000 . 0011 1100 . 0110 0100
Netmask	255.255.255.0	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
Network address	192.168.60.0	1100 0000 . 1010 1000 . 0011 1100 . 0000 0000

enp0s9:

IP address	192.168.70.5	1100 0000 . 1010 1000 . 0100 0110 . 0000 0101
Netmask	255.255.255.0	1111 1111 . 1111 1111 . 1111 1111 . 0000 0000
Network address	192.168.70.0	1100 0000 . 1010 1000 . 0100 0110 . 0000 0000



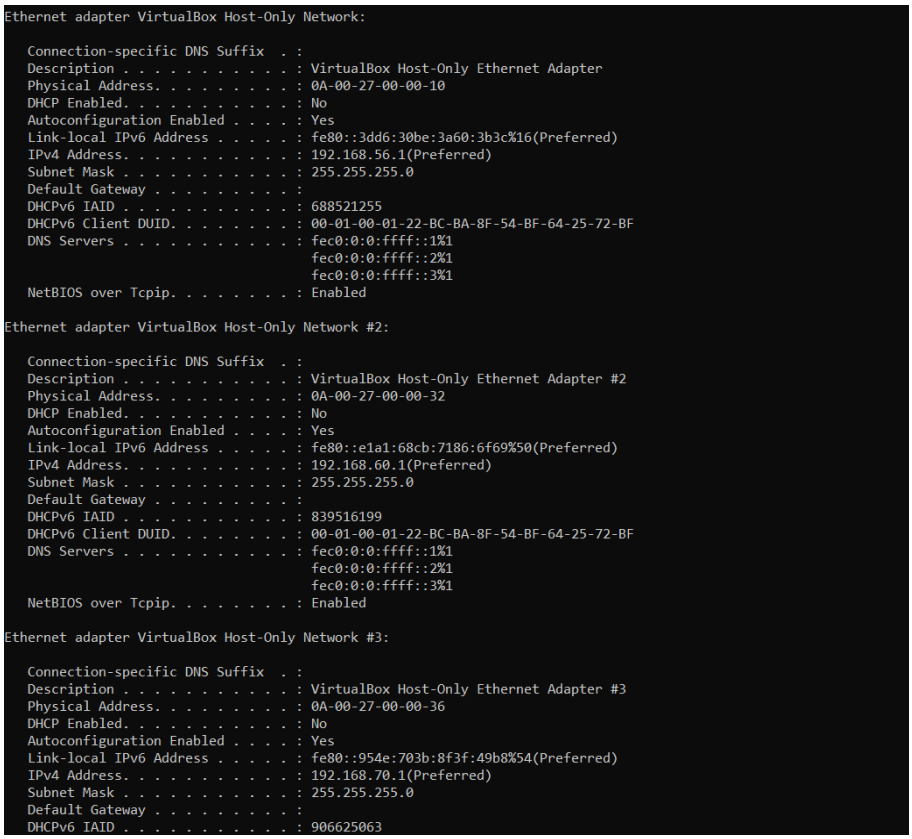
Task 4: Host-only interfaces:

Since my host OS is windows, I have used command ipconfig /all in command prompt. The subnet mask for following interfaces is 255.255.255.0.

VirtualBox Host-Only Ethernet Adapter #2 - 192.168.60.1

VirtualBox Host-Only Ethernet Adapter #3 - 192.168.70.1

The above host only interfaces ethernet adapter 2 and ethernet adapter 3 are connected to enp0s3 and enp0s8 in guest OS (VM) through 192.168.60.1 and 192.168.70.1 addresses.



Task 5: Routing tables in the host OS:

Since my host OS is windows, I have used command “route -4 PRINT” in command prompt.

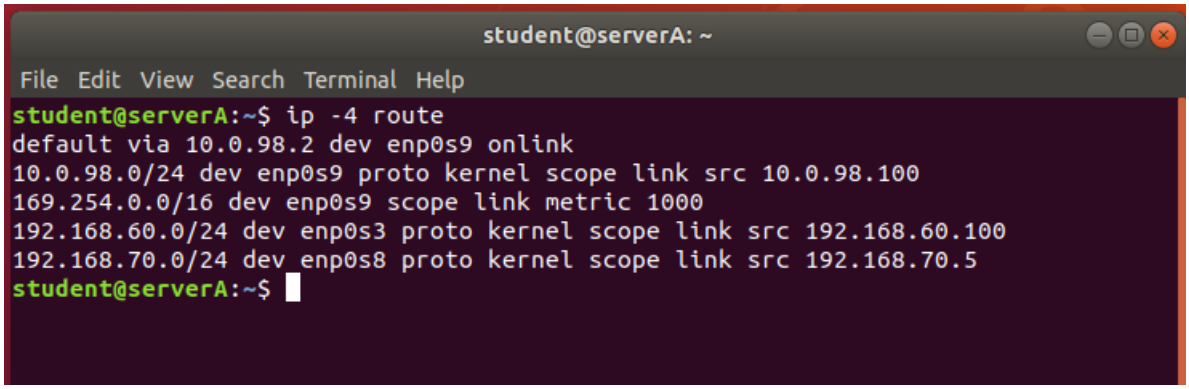
Host OS is directly connected to guest OS without gateway. In the below figure we can see in the gateway column there is onlink which means it is directly connected.

```
C:\Users\saini>route -4 PRINT
===========================================================================
Interface List
 11...54 bf 64 25 72 bf ......Killer E2400 Gigabit Ethernet Controller
 16...0a 00 27 00 00 10 ......VirtualBox Host-Only Ethernet Adapter
 50...0a 00 27 00 00 32 ......VirtualBox Host-Only Ethernet Adapter #2
 54...0a 00 27 00 00 36 ......VirtualBox Host-Only Ethernet Adapter #3
 58...0a 00 27 00 00 3a ......VirtualBox Host-Only Ethernet Adapter #4
 17...c0 b6 f9 73 06 41 ......Microsoft Wi-Fi Direct Virtual Adapter
  4...c2 b6 f9 73 06 40 ......Microsoft Wi-Fi Direct Virtual Adapter #2
  3...c0 b6 f9 73 06 40 ......Intel(R) Wireless-AC 9560 160MHz
  1...........................Software Loopback Interface 1
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0     192.168.29.1   192.168.29.133     35
        127.0.0.0        255.0.0.0         On-link         127.0.0.1    331
        127.0.0.1  255.255.255.255         On-link         127.0.0.1    331
  127.255.255.255  255.255.255.255         On-link         127.0.0.1    331
     192.168.29.0    255.255.255.0         On-link    192.168.29.133    291
   192.168.29.133  255.255.255.255         On-link    192.168.29.133    291
   192.168.29.255  255.255.255.255         On-link    192.168.29.133    291
     192.168.56.0    255.255.255.0         On-link      192.168.56.1    281
     192.168.56.1  255.255.255.255         On-link      192.168.56.1    281
   192.168.56.255  255.255.255.255         On-link      192.168.56.1    281
     192.168.60.0    255.255.255.0         On-link      192.168.60.1    281
     192.168.60.1  255.255.255.255         On-link      192.168.60.1    281
   192.168.60.255  255.255.255.255         On-link      192.168.60.1    281
     192.168.70.0    255.255.255.0         On-link      192.168.70.1    281
     192.168.70.1  255.255.255.255         On-link      192.168.70.1    281
   192.168.70.255  255.255.255.255         On-link      192.168.70.1    281
     192.168.80.0    255.255.255.0         On-link      192.168.80.1    281
     192.168.80.1  255.255.255.255         On-link      192.168.80.1    281
   192.168.80.255  255.255.255.255         On-link      192.168.80.1    281
    192.168.137.0    255.255.255.0         On-link     192.168.137.1    281
    192.168.137.1  255.255.255.255         On-link     192.168.137.1    281
  192.168.137.255  255.255.255.255         On-link     192.168.137.1    281
        224.0.0.0        240.0.0.0         On-link         127.0.0.1    331
        224.0.0.0        240.0.0.0         On-link      192.168.56.1    281
        224.0.0.0        240.0.0.0         On-link    192.168.29.133    291
        224.0.0.0        240.0.0.0         On-link     192.168.137.1    281
        224.0.0.0        240.0.0.0         On-link      192.168.60.1    281
        224.0.0.0        240.0.0.0         On-link      192.168.70.1    281
        224.0.0.0        240.0.0.0         On-link      192.168.80.1    281
  255.255.255.255  255.255.255.255         On-link         127.0.0.1    331
```

Task 6: Routing tables in the guest OS:

I have used the command ip -4 route in guest OS. In the below figure we can clearly see the gateway onlink in which guest OS is directly to host OS. We can see the interface is enp0s9 which it is a NAT interface.

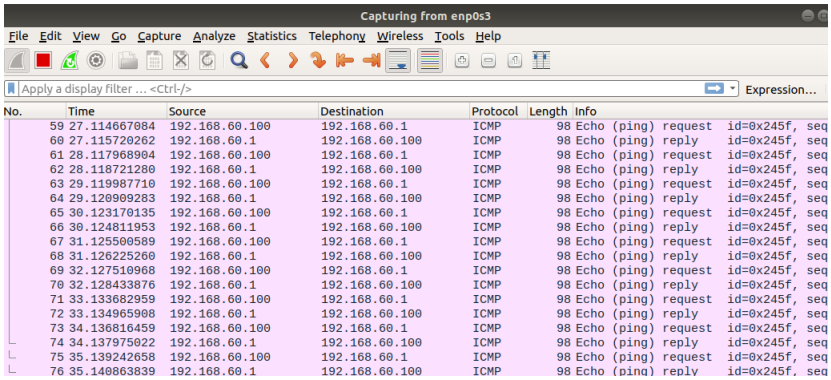


Task 7: Ping the host-based host-only interface:

In the terminal in the guest OS ping the IP address corresponding to the host-only interface

in the host OS using command “ping 192.168.60.1”. By observing both figures, ICMP traffic from the two Wireshark instances are identical.

For guest os:



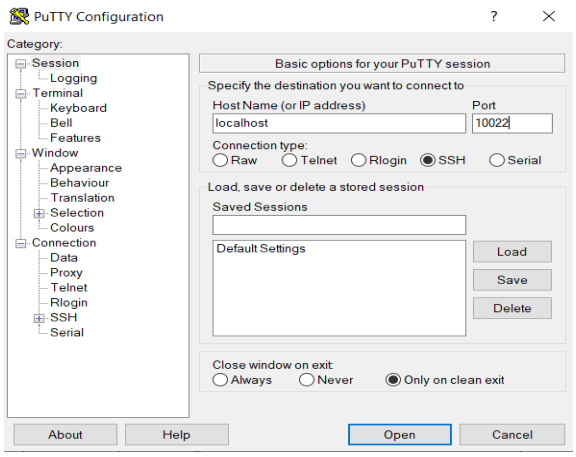
No.	Time	Source	Destination	Protocol	Length	Info
59	27.114667084	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
60	27.115720262	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
61	28.117968904	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
62	28.118721280	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
63	29.119987710	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
64	29.120909283	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
65	30.123170135	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
66	30.124811953	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
67	31.125500589	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
68	31.126225260	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
69	32.127510968	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
70	32.128433876	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
71	33.133682959	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
72	33.134965908	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
73	34.136816459	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
74	34.137975022	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq
75	35.139242658	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq
76	35.140863839	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq

For host os:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=73/18688, ttl=64 (reply in 2)
2	0.000119	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=73/18688, ttl=64 (request in 1)
3	1.001379	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=74/18944, ttl=64 (reply in 4)
4	1.001502	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=74/18944, ttl=64 (request in 3)
5	2.021775	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=75/19200, ttl=64 (reply in 6)
6	2.022014	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=75/19200, ttl=64 (request in 5)
7	3.023202	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=76/19456, ttl=64 (reply in 8)
8	3.023334	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=76/19456, ttl=64 (request in 7)
9	4.024584	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=77/19712, ttl=64 (reply in 10)
10	4.024837	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=77/19712, ttl=64 (request in 9)
11	5.027254	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=78/19968, ttl=64 (reply in 12)
12	5.027411	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=78/19968, ttl=64 (request in 11)
13	6.029480	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=79/20224, ttl=64 (reply in 14)
14	6.029668	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=79/20224, ttl=64 (request in 13)
15	7.030507	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=80/20480, ttl=64 (reply in 16)
16	7.030636	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=80/20480, ttl=64 (request in 15)
17	8.032679	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=81/20736, ttl=64 (reply in 18)
18	8.032796	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=81/20736, ttl=64 (request in 17)
19	9.033875	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) request id=0x245f, seq=82/20992, ttl=64 (reply in 20)
20	9.034089	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply id=0x245f, seq=82/20992, ttl=64 (request in 19)

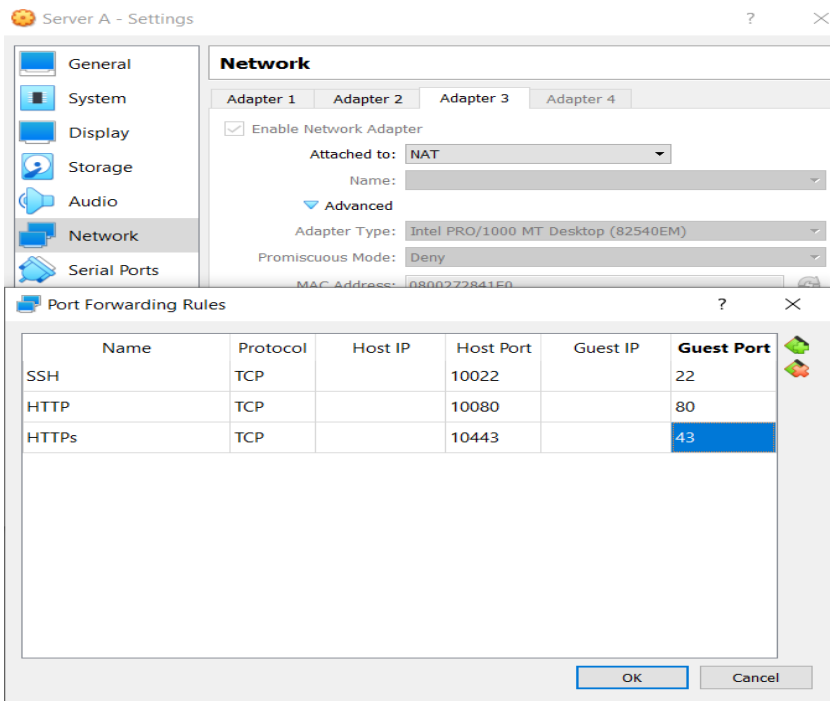
Task 8: SSH into VM via localhost:

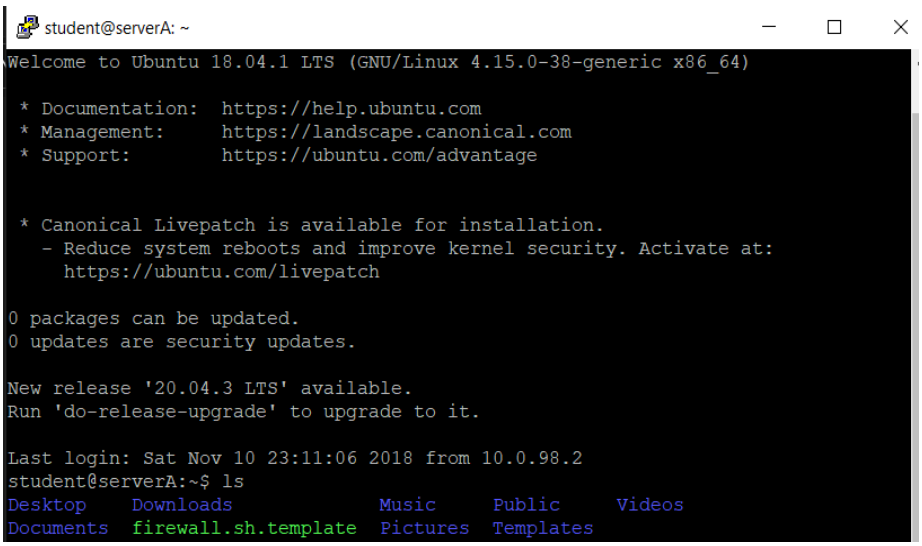
Since hostOS is windows in putty I have given hostname as localhost and port number as 10022.



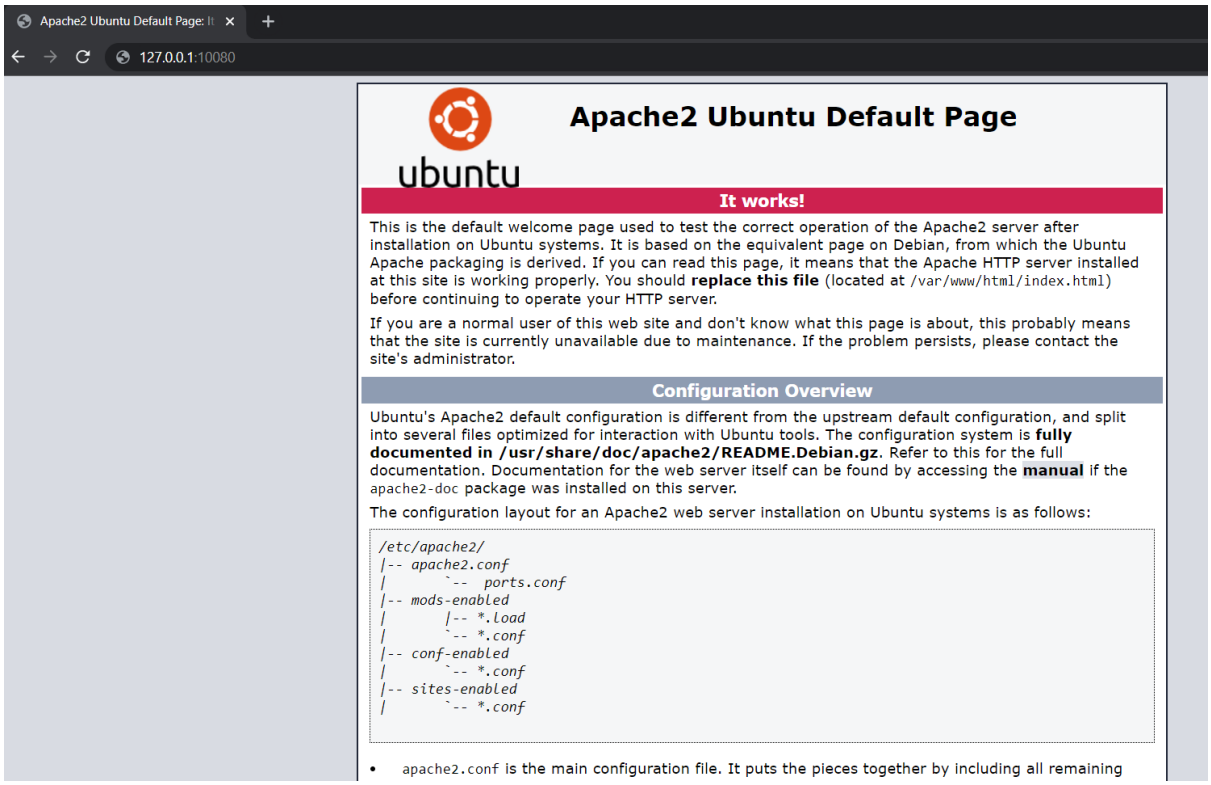
After the terminal is opened, I have listed the folders using ls command. The figure below shows different types of folders.

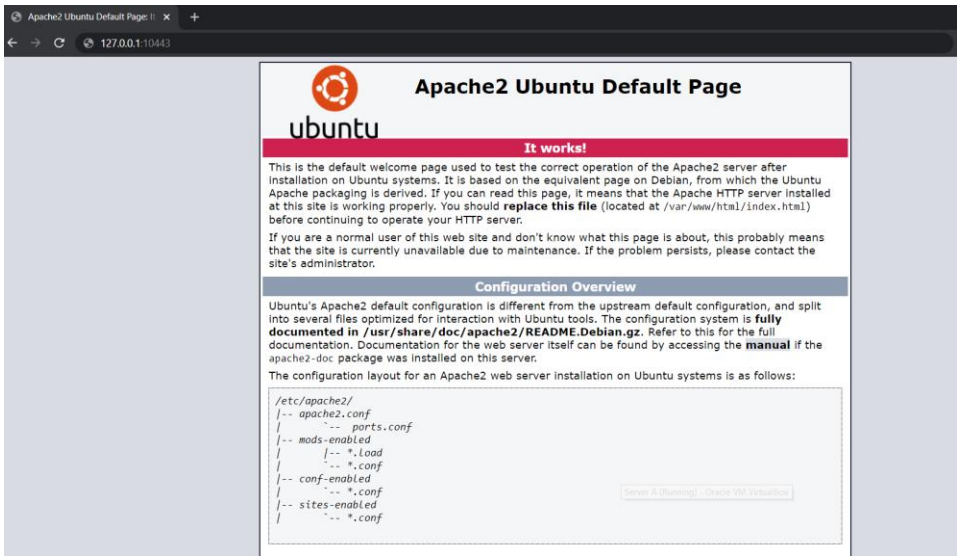
Task 9: Add forwarding rules for HTTP and HTTPS in VirtualBox:





I have used 127.0.0.1:10080 for http and 127.0.0.1:10443 for https in the browser. Apache2 server has worked well for both ports. The below image shows apache2 server for both ports.



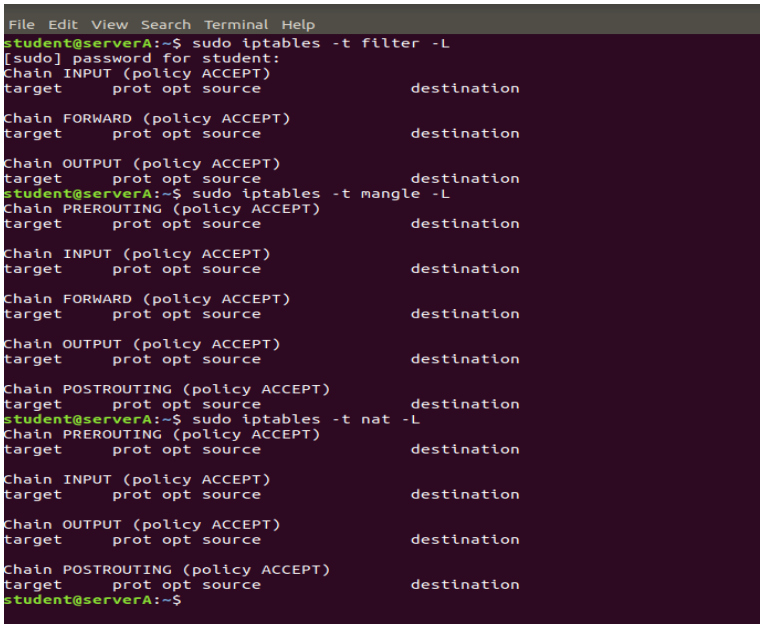


TASK - 10: Default firewall policy and rules:

I have used following commands to see the default policy in the tables. From the commands t indicates table and L indicates list.

sudo iptables -t filter -L
sudo iptables -t mangle -L

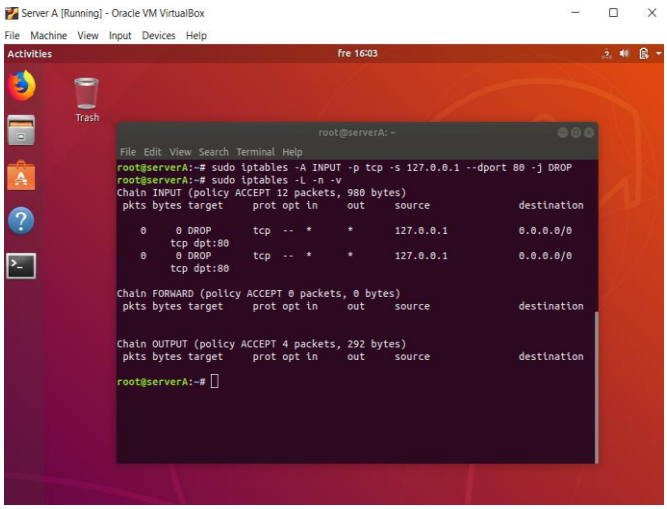
sudo iptables -t nat -L



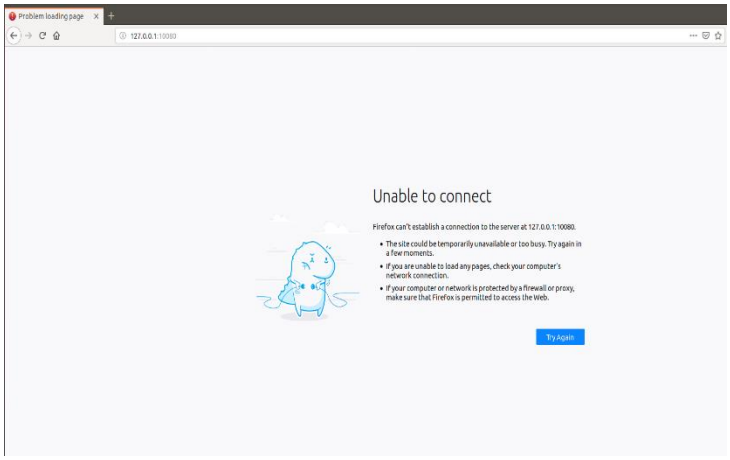
TASK - 11: Block HTTP-browsing in the guest OS:

I have used following command to block HTTP browsing from guest OS but I can be able to browse HTTPs.

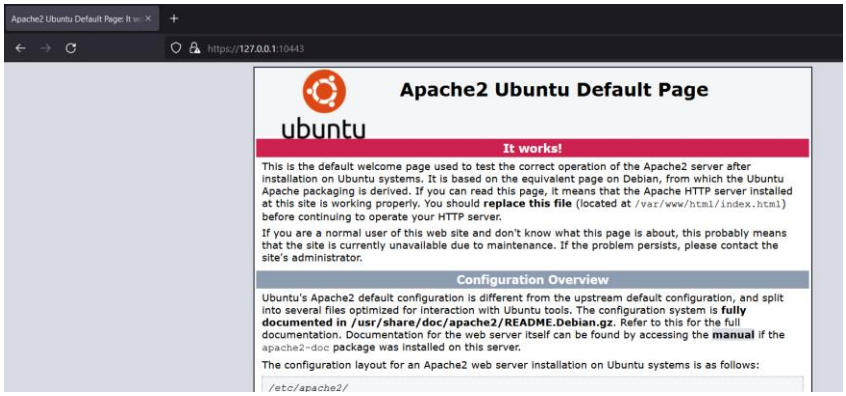
sudo iptables -A INPUT -p tcp -s 127.0.0.1 --dport 80 -j DROP

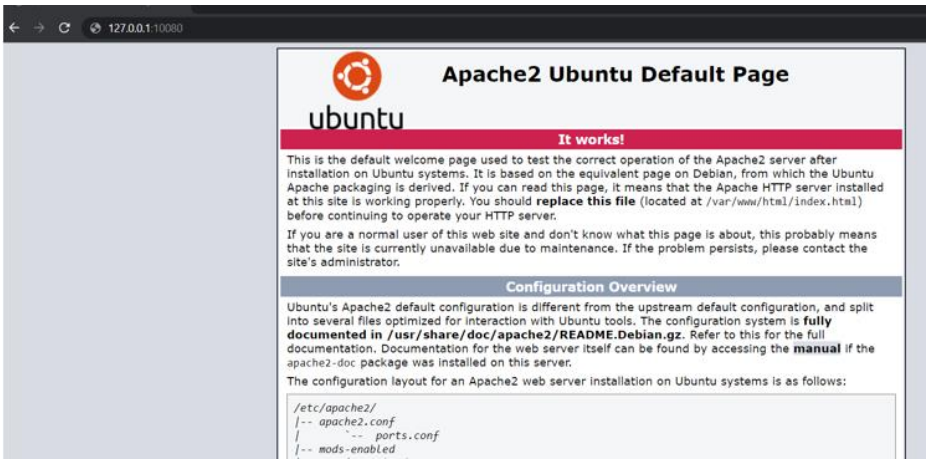


After using the above the command now we tried to browse http from guest os below figure shows we are not able to view web page.



The figure below shows content served over https we can able to see apache2 page for ports 80 and 443 in host OS. Below shows that we are able to browse https from host os.

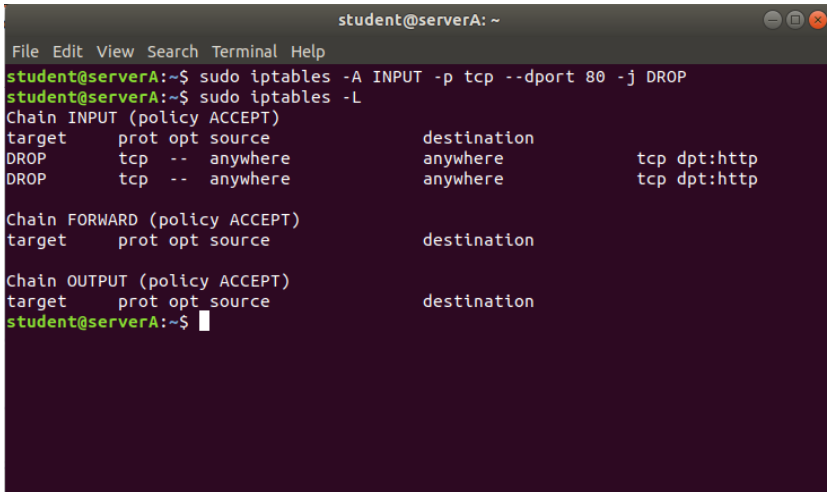


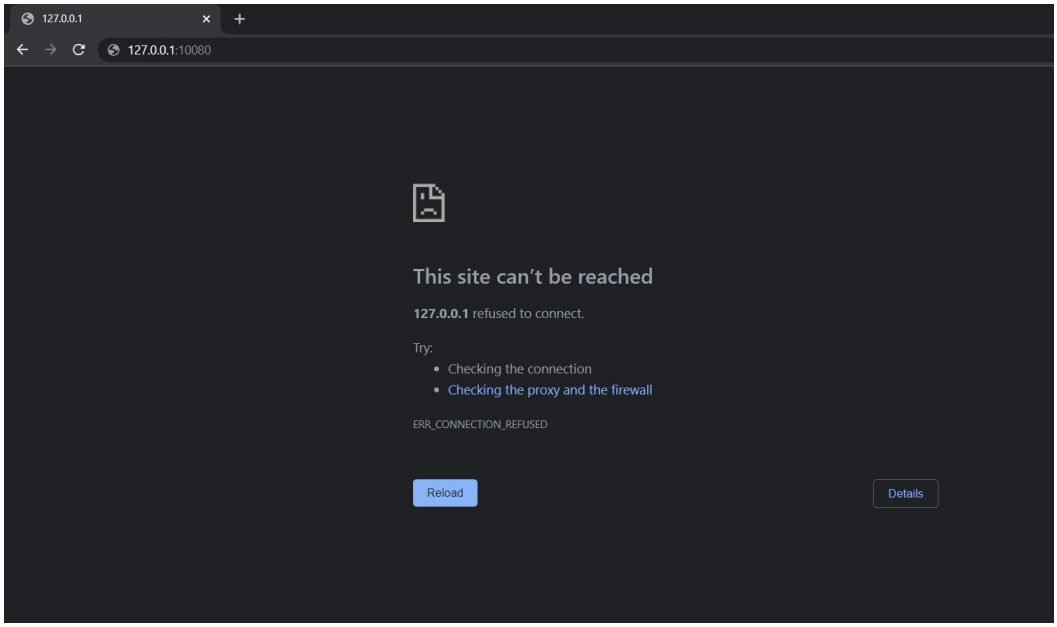


Task 12: Block Apache web server from serving content over HTTP:

I have used following command to block apche2 webserver from serving content over HTTP.

sudo iptables -A INPUT -p tcp –dport 80 -j DROP

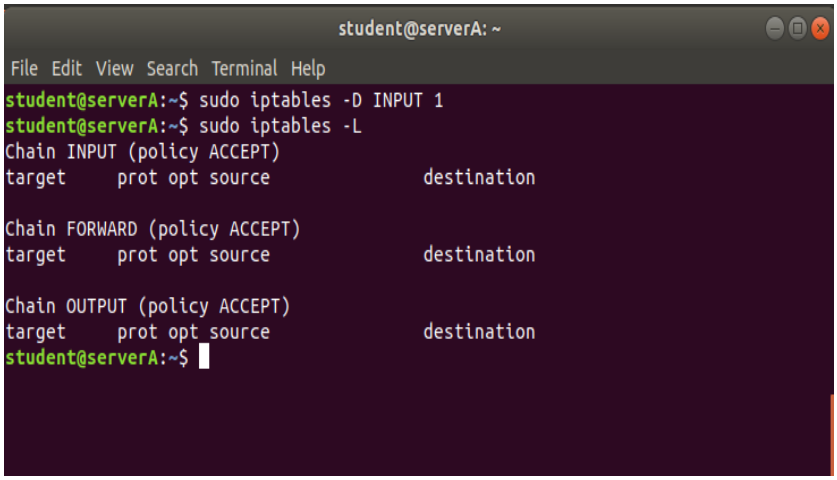




Task 13: Unblock HTTP-browsing in the guest OS:

I have used following command to unblock HTTP browsing in the guest OS.

sudo iptables -D INPUT 1

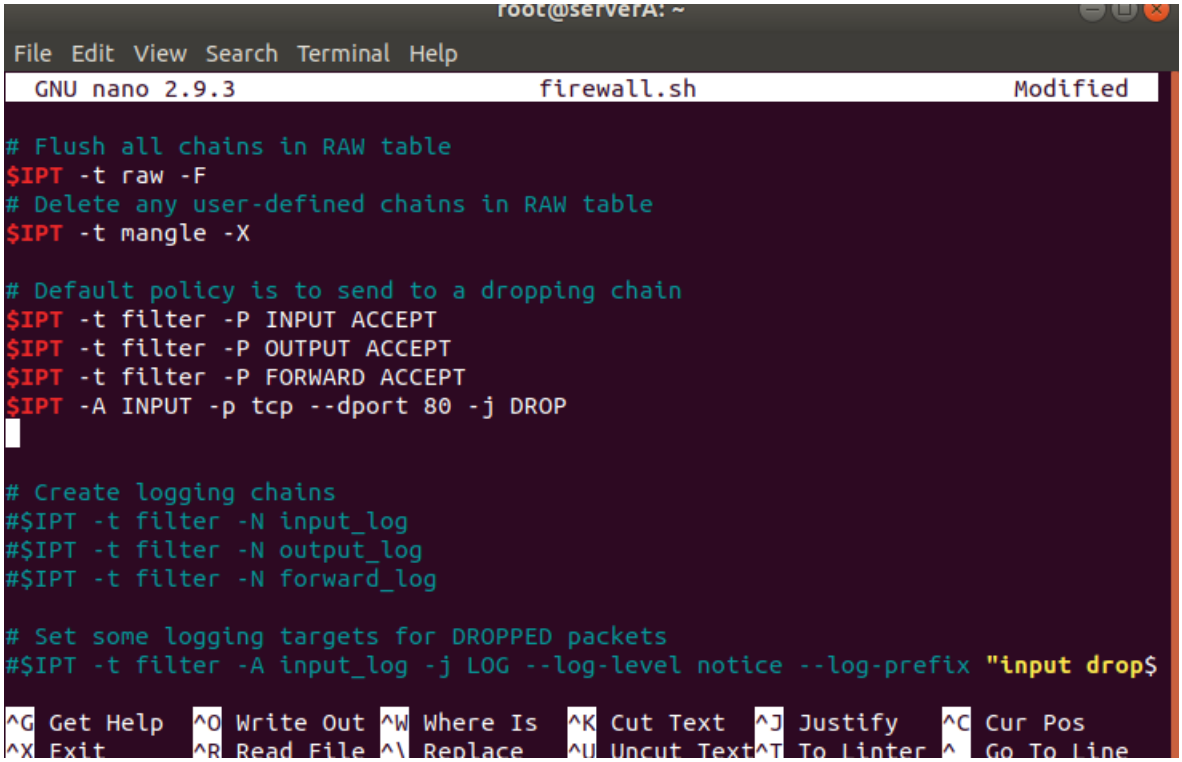


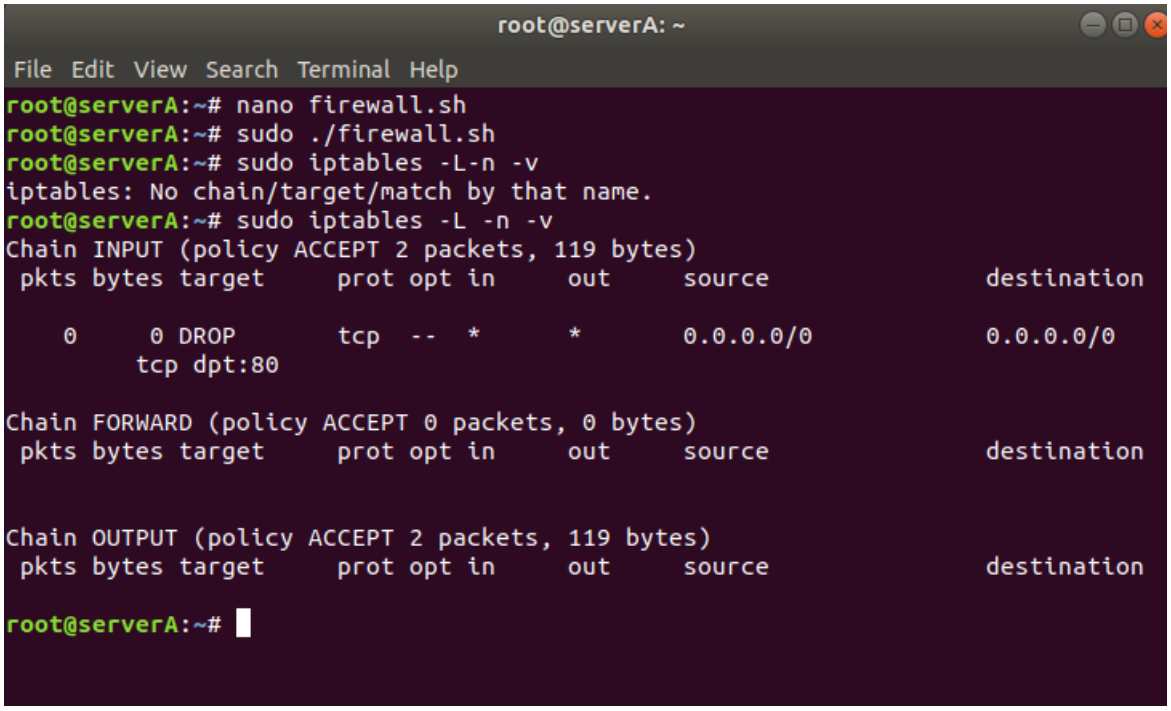
Task 14: Use firewall.sh to configure the firewall:

I have copied the firewall.sh template file into firewall.sh by using the following command.

cp firewall.sh.template firewall.sh

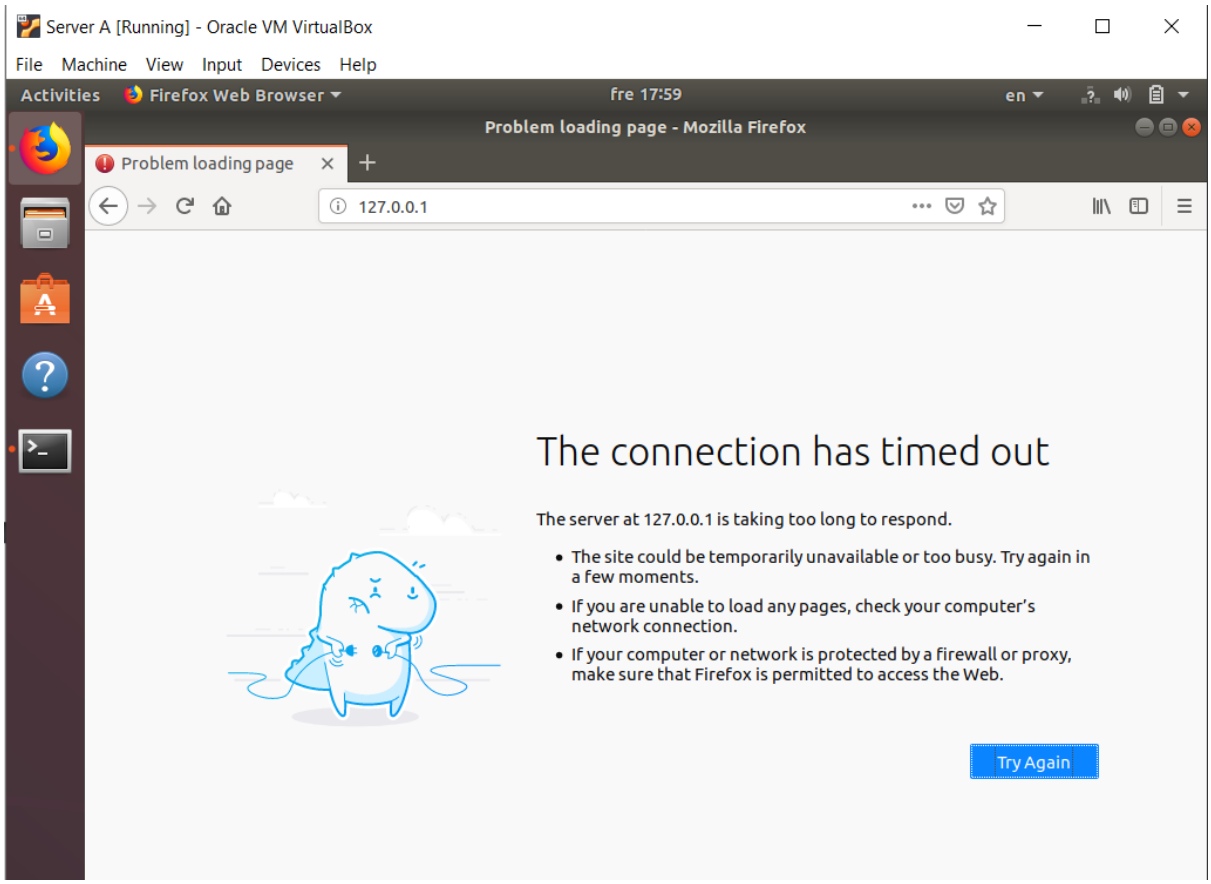
I have used the command $IPT -A INPUT -p tcp --dport 80 -j DROP to block host from viewing HTTP.





After adding rules in firewall.sh and then I have executed it by using command

sudo ./firewall.sh. In the below figure we can observe that apache2 server is blocked in guestOS through HTTP.



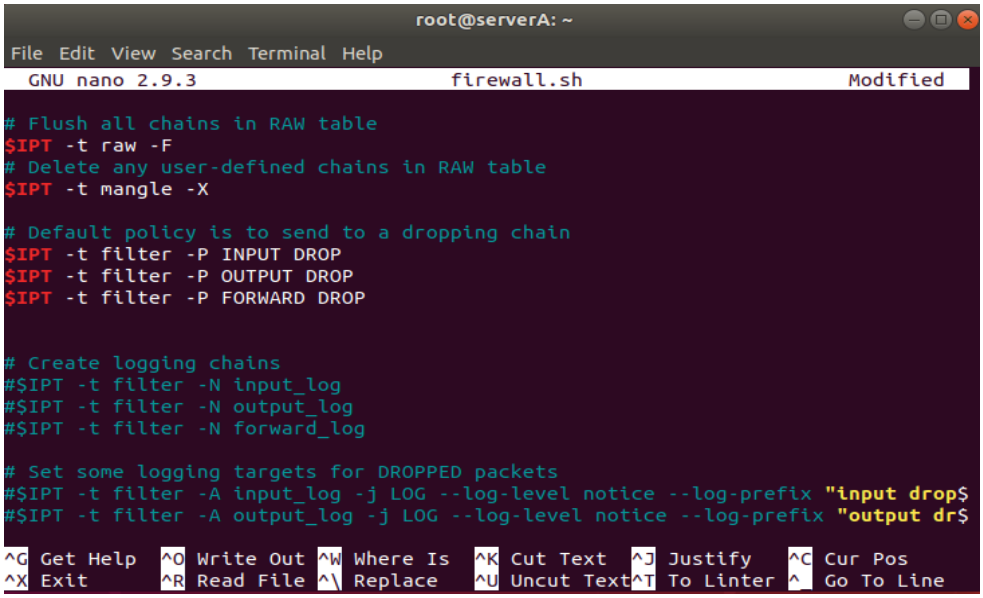
Task 15: Change default firewall policy to DROP:

Initially in firewall.sh script default firewall policy is in ACCEPT. I have changed default firewall policy to DROP.

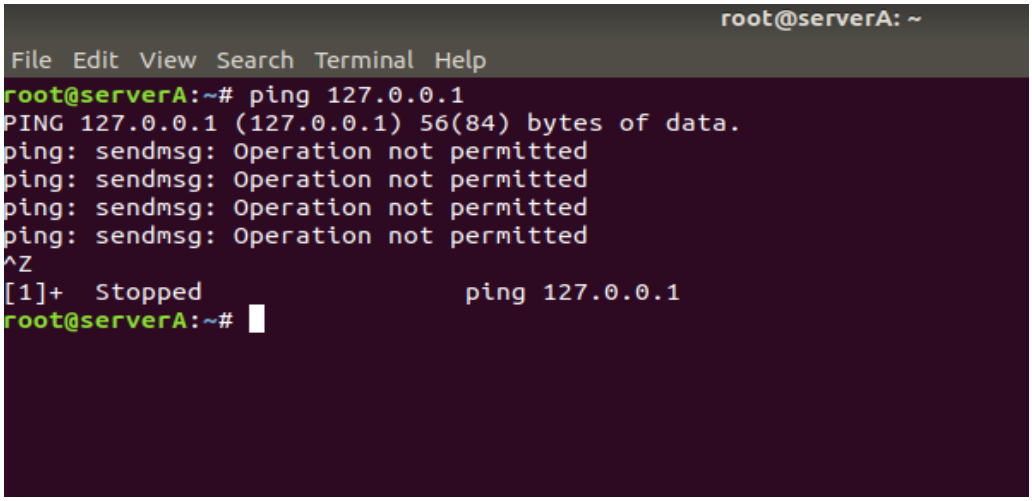
$IPT -t filter -p INPUT DROP

$IPT -t filter -p OUTPUT DROP

$IPT -t filter -p FORWARD DROP

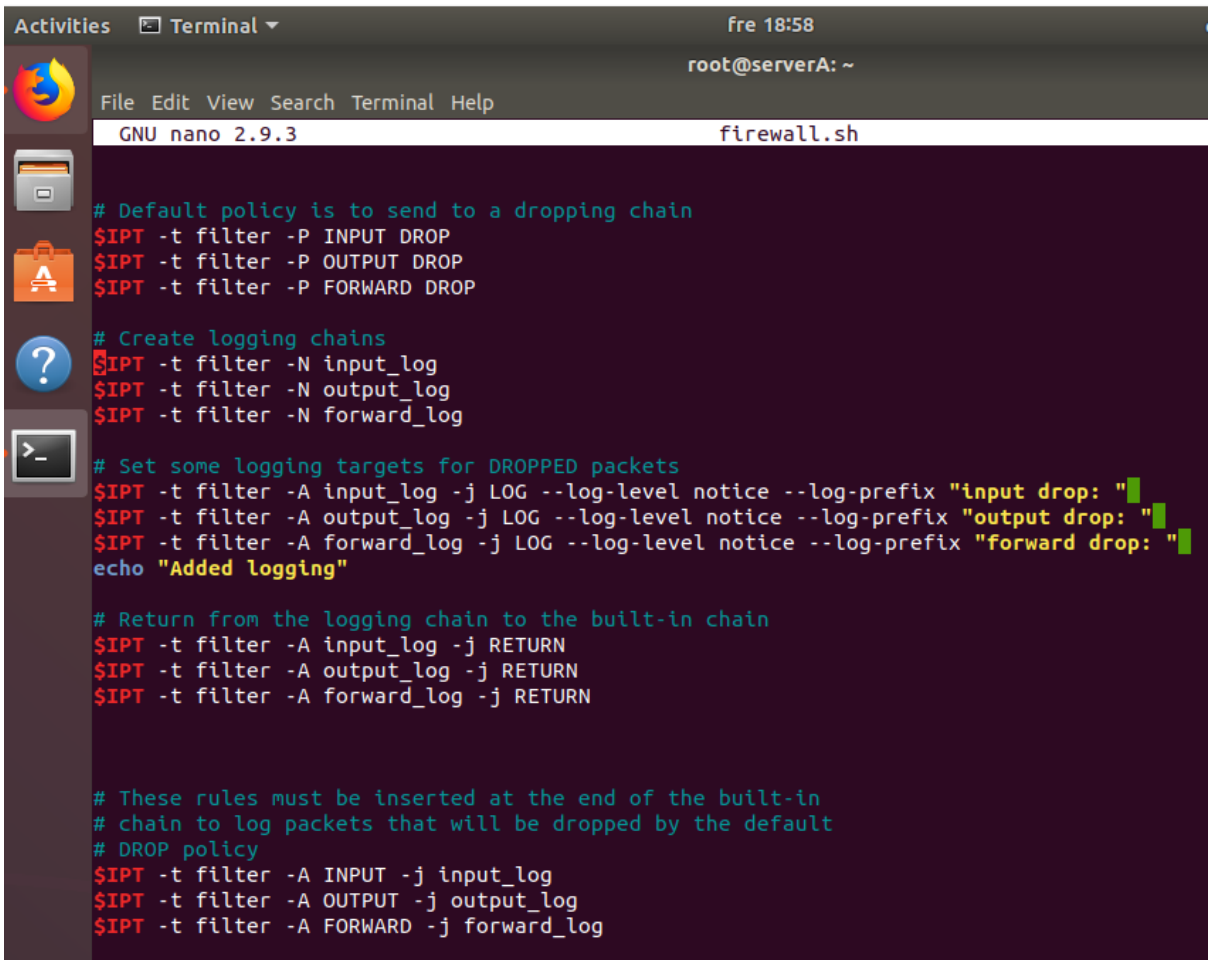


My expectation is correct. The ping is not permitted when trying to ping loopback interface. We can clearly see in the below figure.

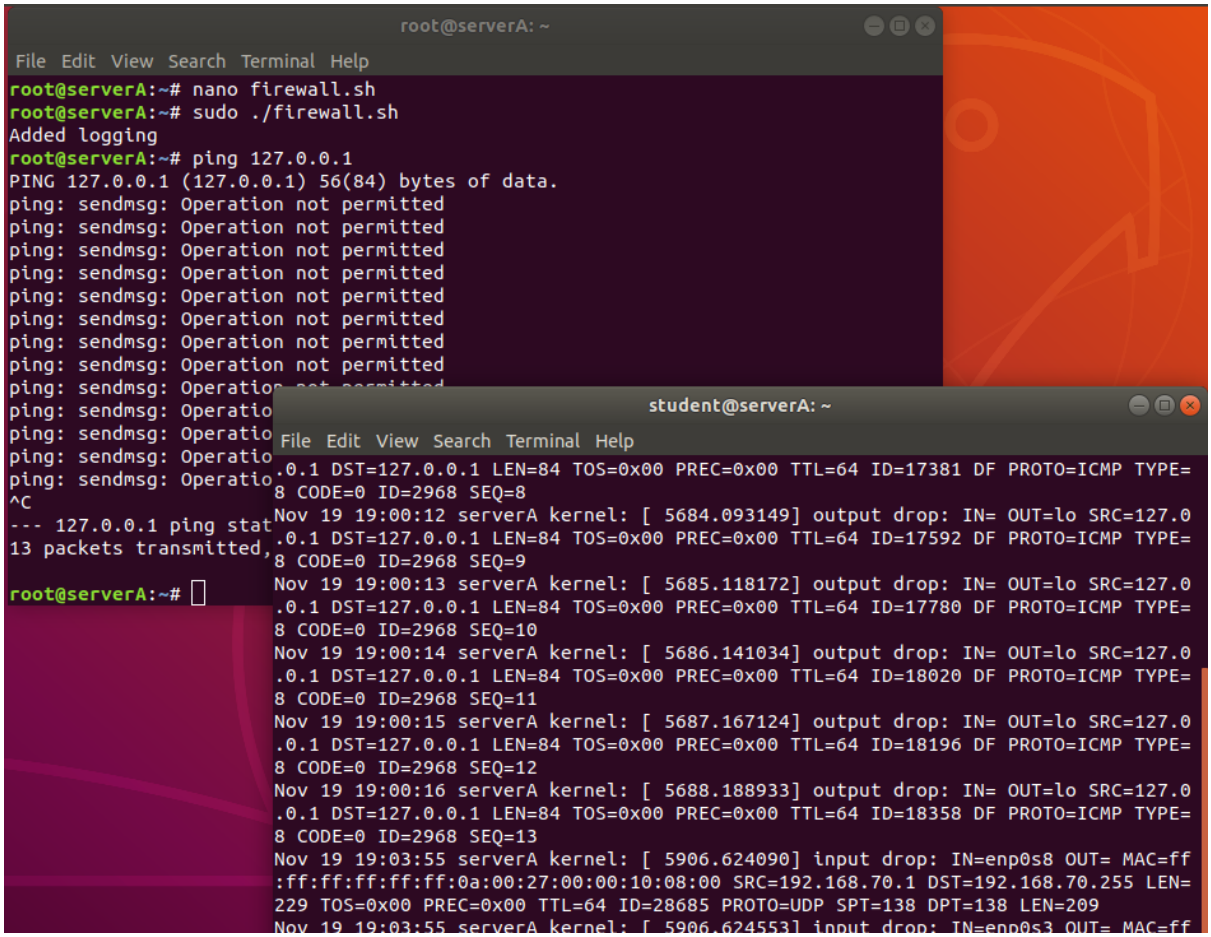


Task 16: Logging DROPPED packets:

I have enabled all logging rules in firewall.sh file. In the below we can clearly observe these rules.



After enabling the rules, I executed firewall.sh file then tried to tried to ping the loopback interface using ping 127.0.0.1 it displays as operation not permitted. I opened other terminal in the new terminal I entered the command sudo tail -f /var/log/kern.log I can be able to see dropped packets from it. We can clearly see in the below figure.

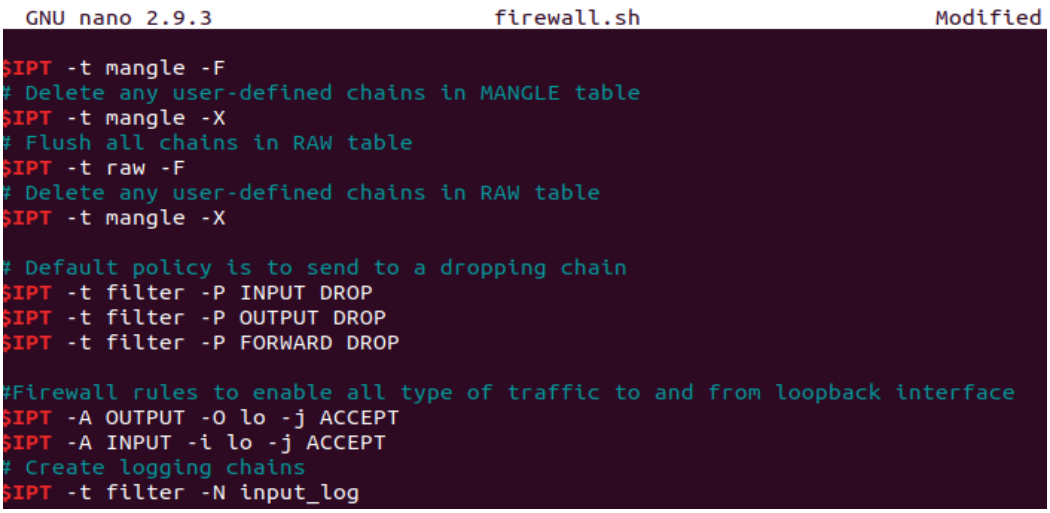


Task 17: Enable traffic from loopback interface:

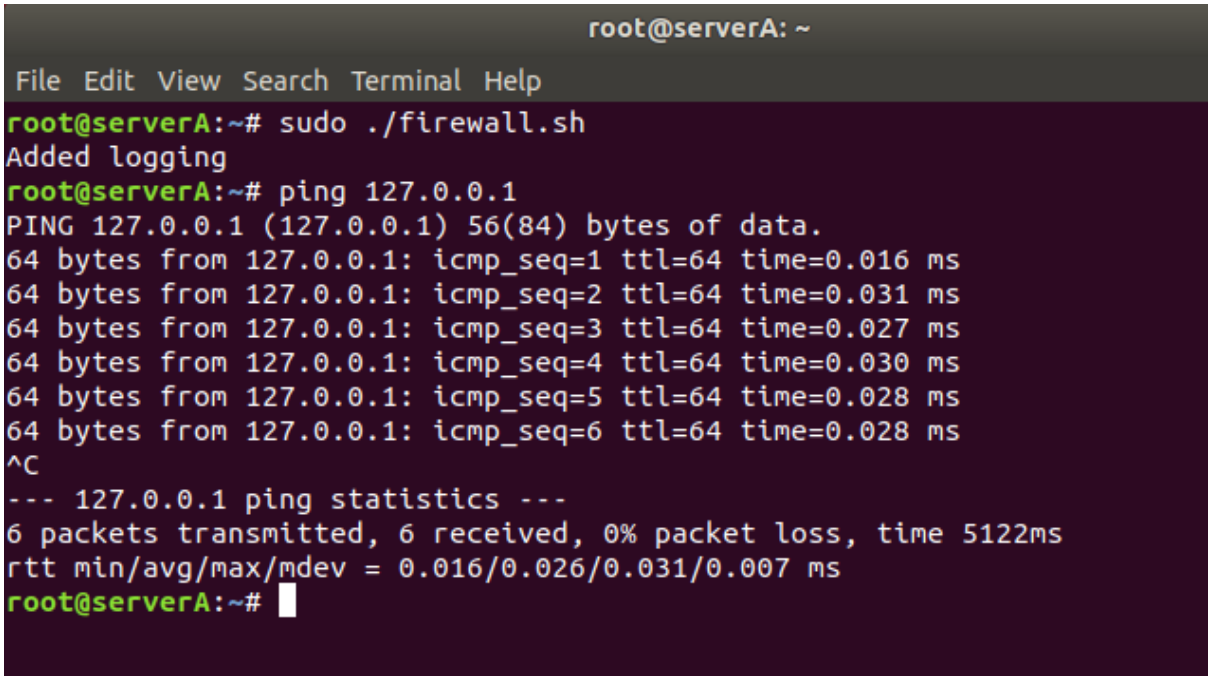
I have added firewall rules in firewall.sh to enable all type of traffic. We can observe the figure below.

$IPT -A OUTPUT -o lo -j ACCEPT

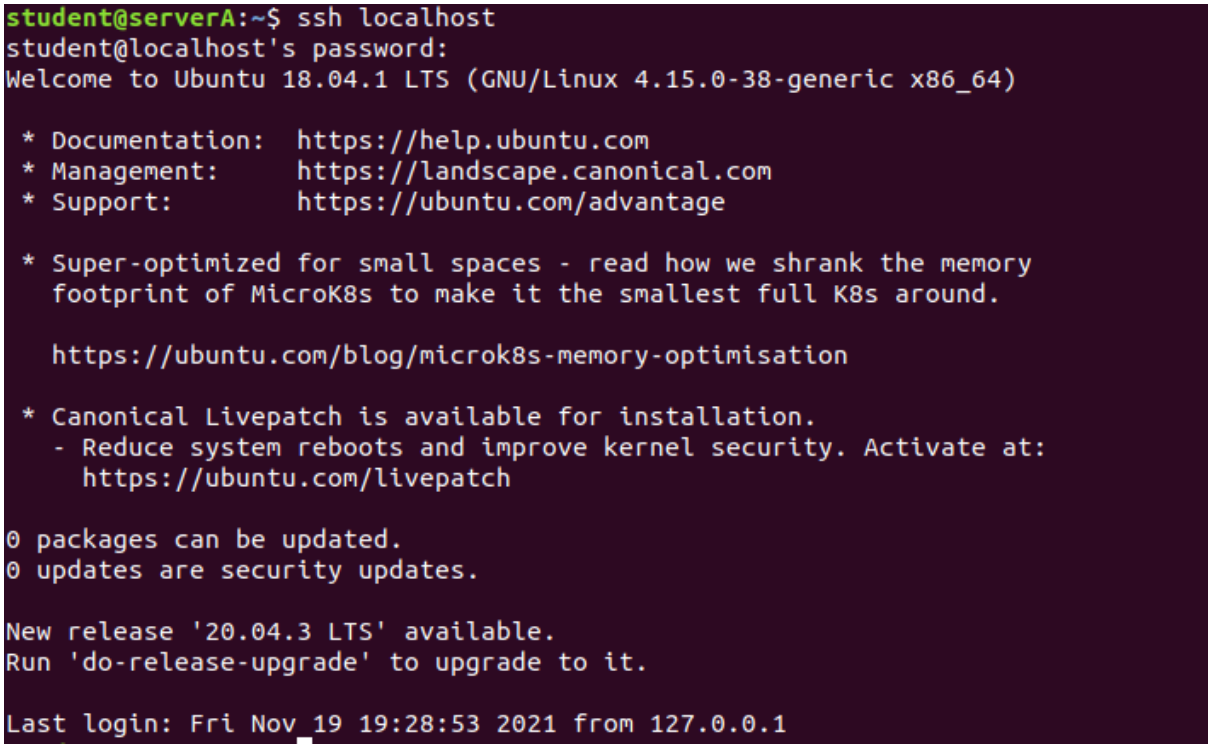
$IPT -A INPUT -I lo -j ACCEPT



After executing firewall.sh I entered ping 127.0.0.1 it worked properly as we can see in figure below.



I am now able to SSH into localhost via loopback interface. We can see in figure below.



Task 18: Allow Server A to ping the other interfaces:

I have added firewall rules to allow ping traffic initiated from Server A.

$IPT -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT

$IPT -A INPUT -p icmp --icmp-type 0 -j ACCEPT

```
#Firewall rules to enable all type of traffic to and from loopback interface
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A INPUT -i lo -j ACCEPT

#Allowing serverA to ping the other interfaces
$IPT -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type 0 -j ACCEPT

# Create logging chains
$IPT -t filter -N input_log
$IPT -t filter -N output_log
$IPT -t filter -N forward_log

# Set some logging targets for DROPPED packets
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop:$
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dro$
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward d$
```

I tried to ping google server by using ping 8.8.8.8, icmp traffic will be able accept as we can see in figure below.

```
root@serverA: ~
File  Edit  View  Search  Terminal  Help
root@serverA:~# sudo ./firewall.sh
Added logging
root@serverA:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=23.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=20.8 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=20.4 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=22.3 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=52 time=20.8 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=52 time=23.1 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=52 time=23.8 ms
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6034ms
rtt min/avg/max/mdev = 20.401/22.070/23.875/1.270 ms
root@serverA:~#
```

Task 19: Allow Server A to ping all hosts:

I have added the following rules in firewall.sh file to allow server A to ping all hosts.

$IPT -A INPUT -p tcp -s 10.0.98.3 --sport 53 -j ACCEPT

$IPT -A OUTPUT -p tcp -d 10.0.98.3 --dport 53 -j ACCEPT

$IPT -A INPUT -p udp -s 10.0.98.3 --sport 53 -j ACCEPT

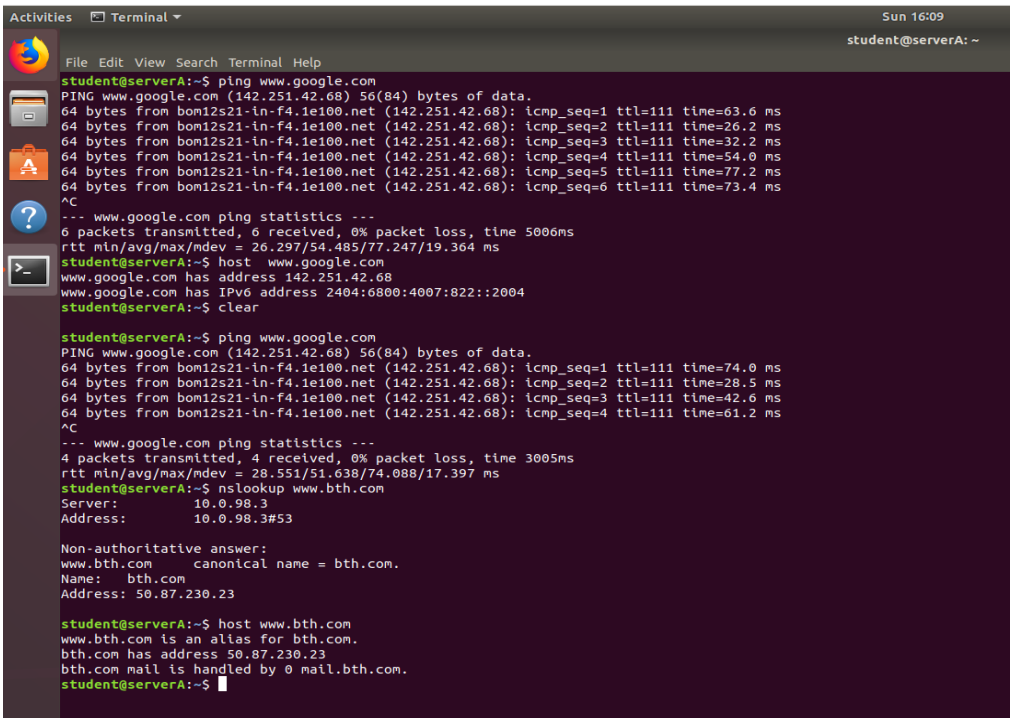
$IPT -A OUTPUT -p udp -d 10.0.98.3 --dport 53 -j ACCEPT

```
#Firewall rules to enable all type of traffic to and from loopback interface
$IPT -A OUTPUT -o lo -j ACCEPT
$IPT -A INPUT -i lo -j ACCEPT

#Allowing serverA to ping the other interfaces
$IPT -A OUTPUT -p icmp --icmp-type 8 -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type 0 -j ACCEPT

#Allowing serverA to ping all hosts
$IPT -A INPUT -p tcp -s 10.0.98.3 --sport 53 -j ACCEPT
$IPT -A OUTPUT -p tcp -d 10.0.98.3 --dport 53 -j ACCEPT
$IPT -A INPUT -p udp -s 10.0.98.3 --sport 53 -j ACCEPT
$IPT -A INPUT -p udp -d 10.0.98.3 --dport 53 -j ACCEPT

# Create logging chains
$IPT -t filter -N input_log
$IPT -t filter -N output_log
$IPT -t filter -N forward_log
```



Task 20: Enable stateful firewall:

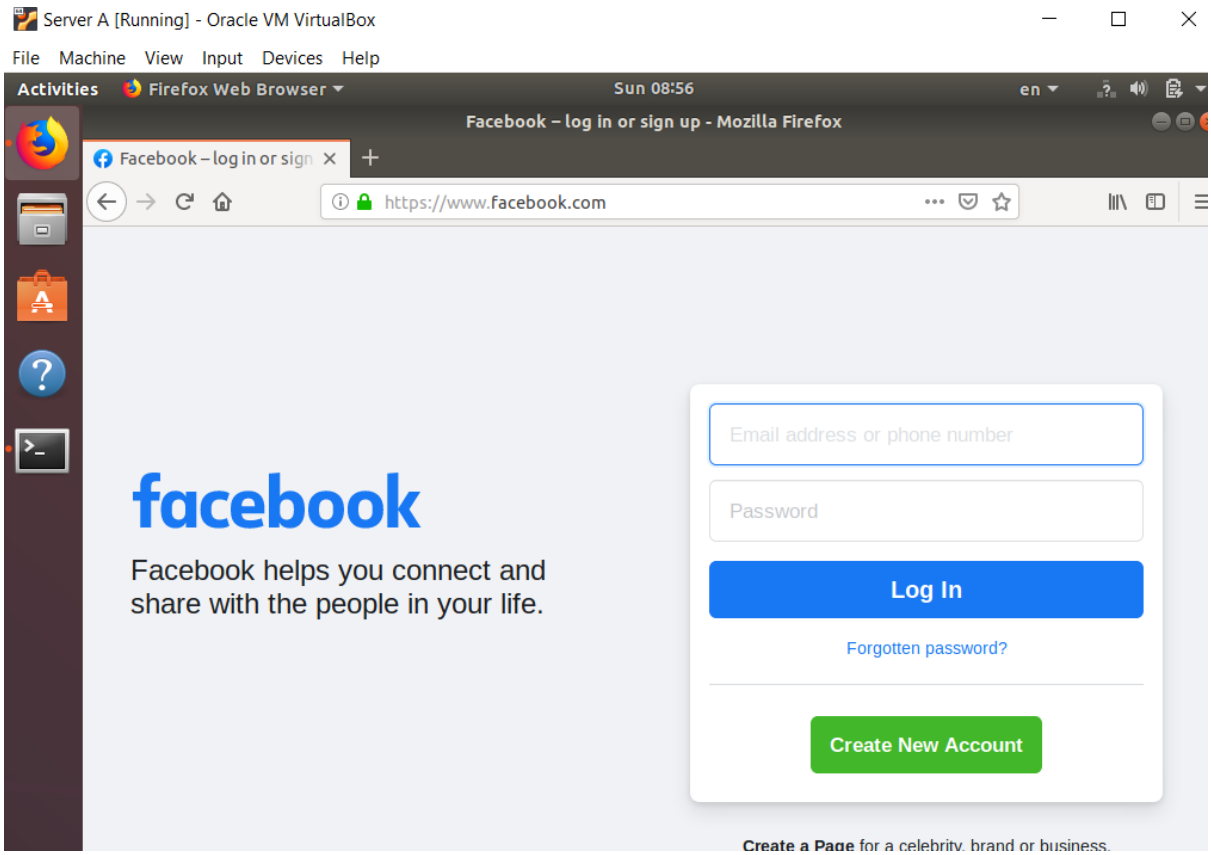
I have used following commands in firewall.sh file for allowing outgoing traffic and also we can able to browse web sites with the Firefox browser from Server A.

$IPT -t filter -A INPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT

$IPT -t filter -A OUTPUT -p tcp -m multiport --sports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT

$IPT -t filter -A INPUT -p tcp -m multiport --sports 80,443 -m conntrack --ctstate ESTABLISHED -j ACCEPT

$IPT -t filter -A OUTPUT -p tcp -m multiport --dports 80,443 -m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT



Task 21: Enable SSH and HTTPS content from apache2 server for web browser on host:

I have used following commands in firewall.sh file to enable SSH and HTTPS content from apache2 server.

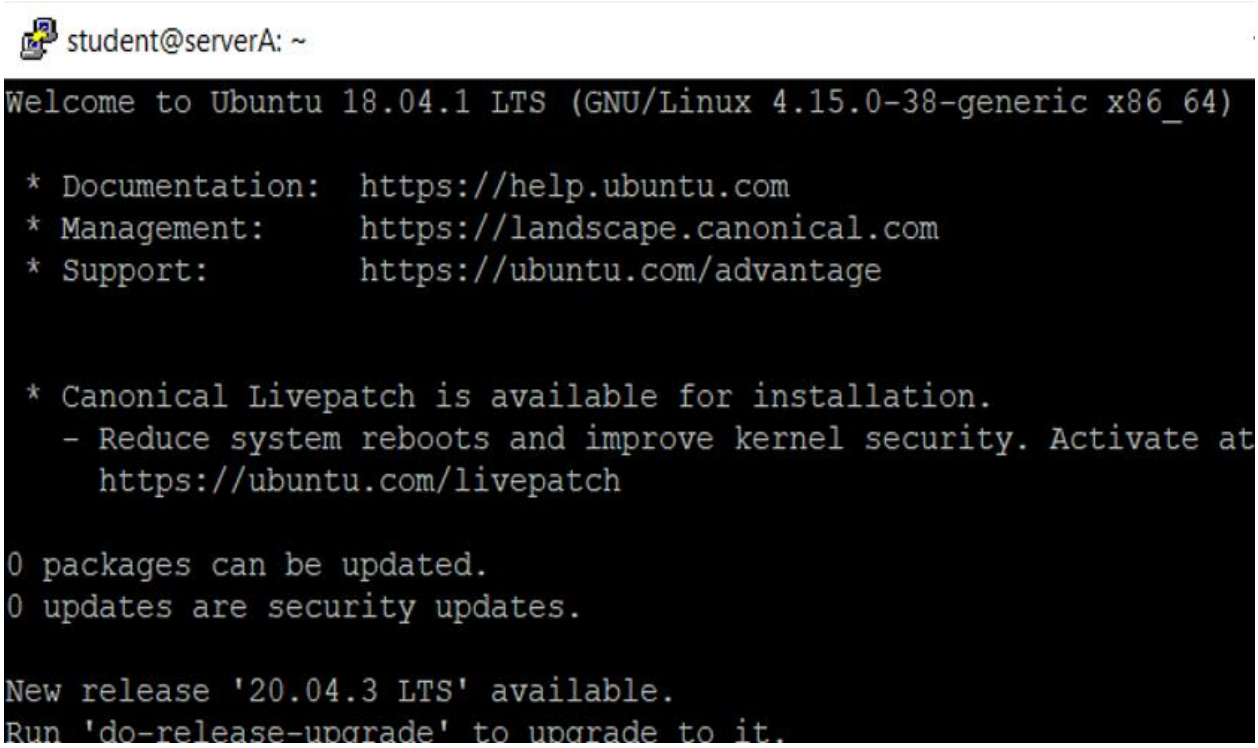
For enabling SSH from apache2 server,

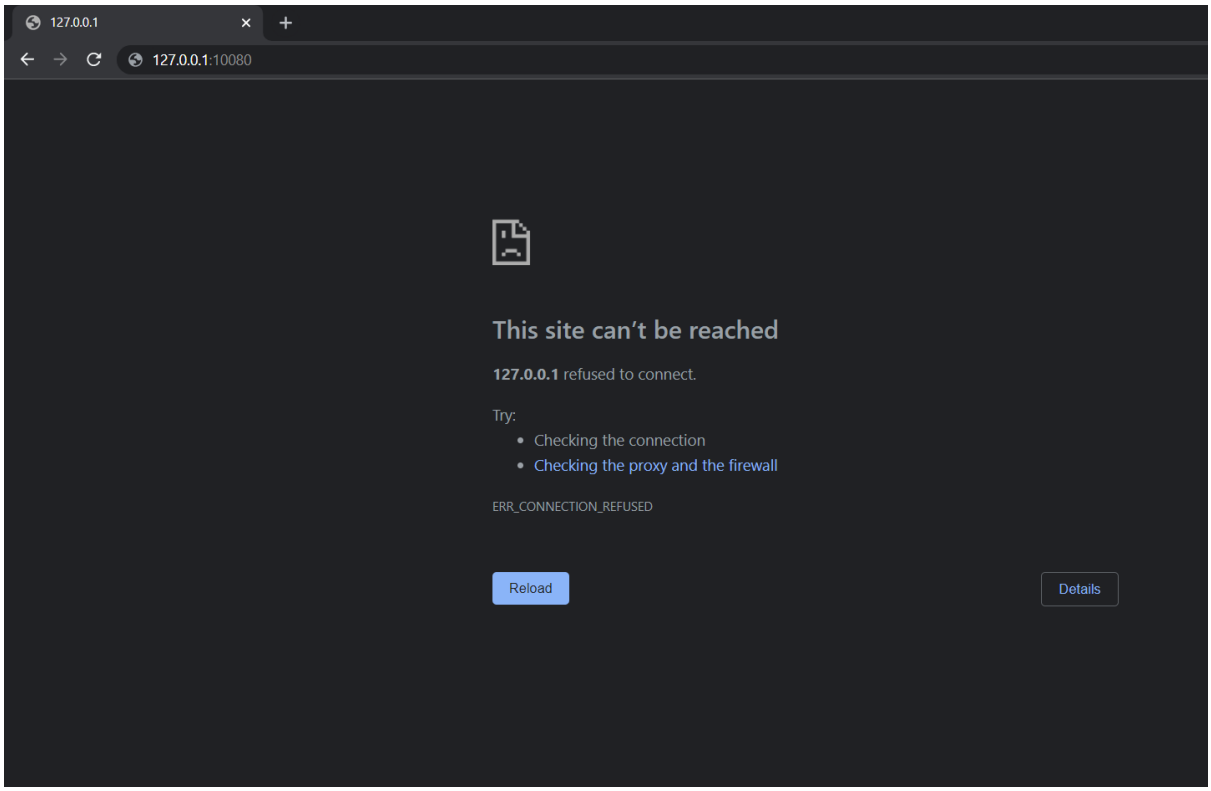
$IPT -A INPUT -p tcp -s 10.0.98.2 --sport 22 -j ACCEPT

$IPT -A OUTPUT -p tcp -d 10.0.98.2 --dport 22 -j ACCEPT

For enabling HTTPS from apache2 server,

$IPT -I INPUT -p tcp -d 10.0.98.100 --dport 80 -j DROP



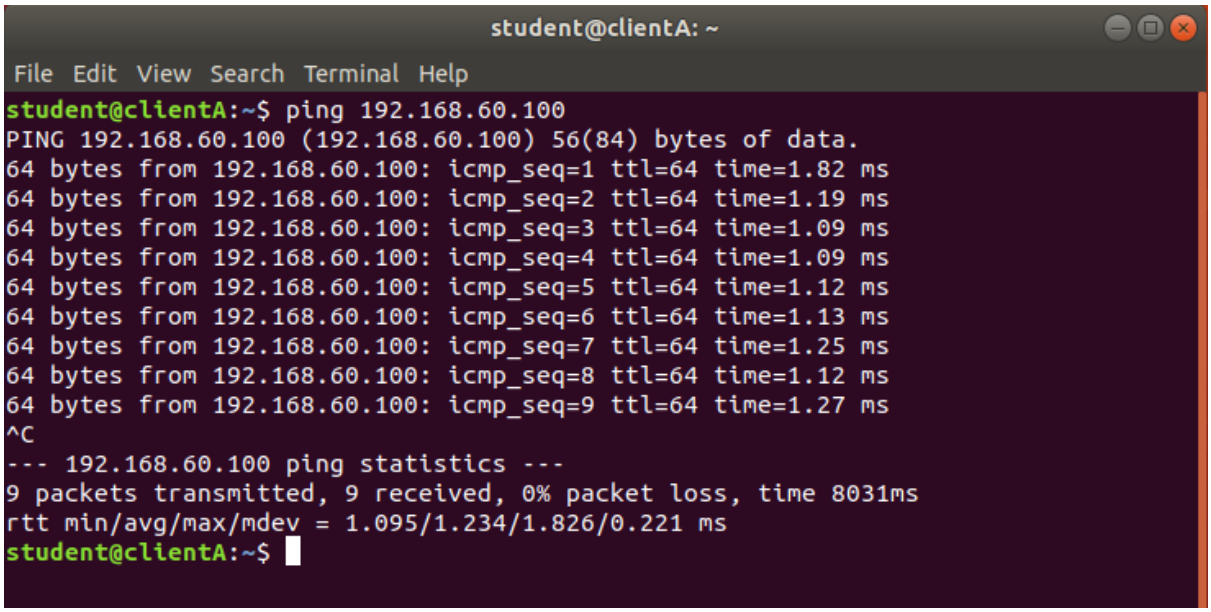


Task 22: Ping Server A from Client A:

I have used following commands to enable ping between server A and client A,

sudo iptables -A INPUT -p icmp --icmp-type 8 -s 192.168.60.111 -j ACCEPT

sudo iptables -A OUTPUT -p icmp --icmp-type 0 -d 192.168.60.111 -j ACCEPT

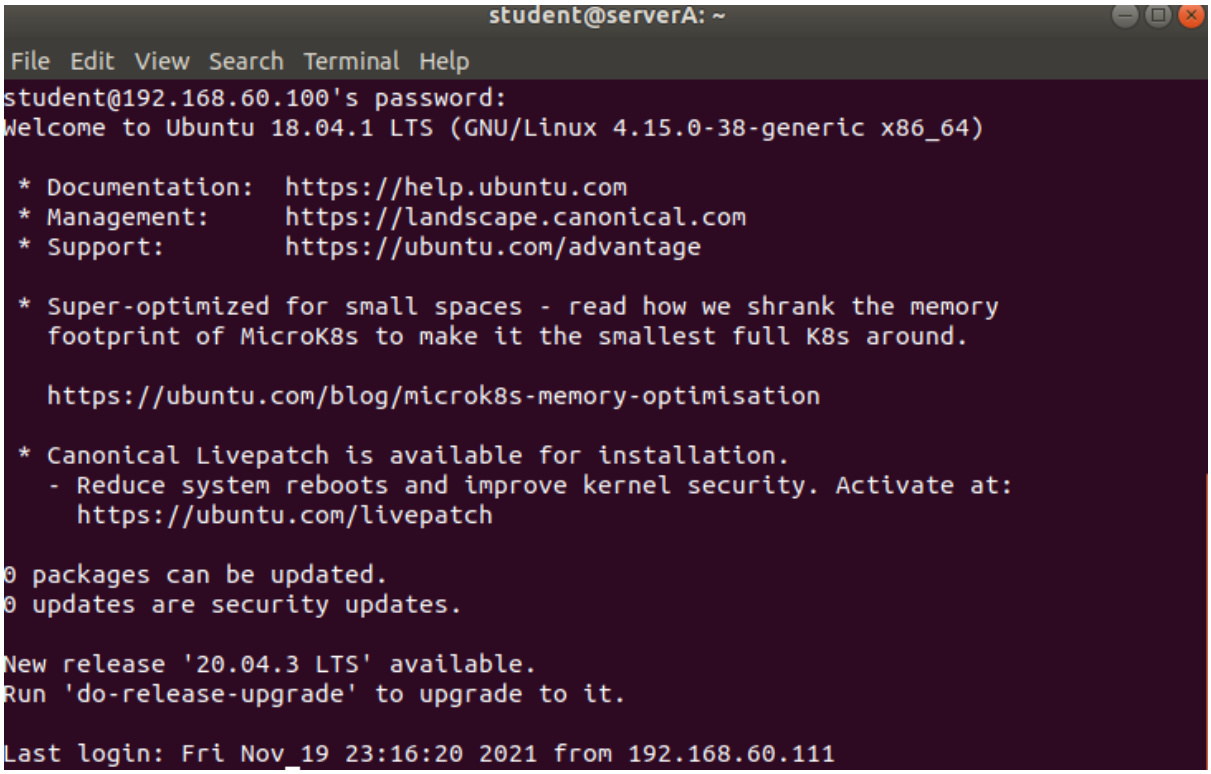


Task 23: SSH from Client A to Server A:

I have used following commands to enable SSH from client A to server A.

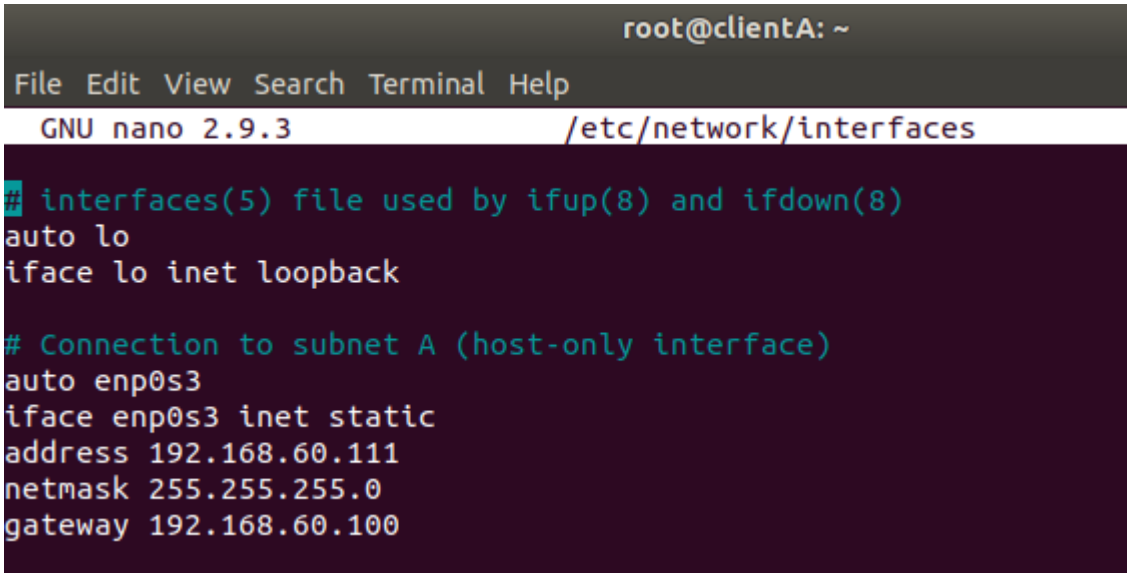
sudo iptables -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT

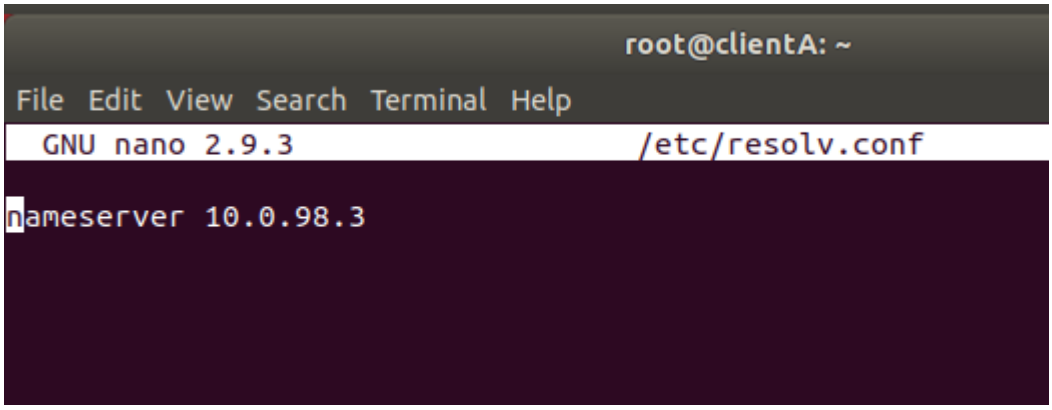
sudo iptables -A OUTPUT -p tcp -d 192.168.60.111 --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT



Task 24: Add gateway and DNS server to Client A:

In /etc/network/interfaces on Client A I added **gateway 192.168.60.100** and in /etc/resolv.conf on Client A I verified the 10.0.98.3 is listed as DNS server. In the below images we can clearly see these lines.





Task 25: Enable IP forwarding on Server A:

For enabling IP forwarding on server A I have used following commands in firewall.sh.

sudo sysctl -w net.ipv4.ip_forward=1

sudo sysctl -p

```
$IPT -A INPUT -p tcp -s 10.0.98.2 --sport 22 -j ACCEPT
$IPT -A OUTPUT -p tcp -d 10.0.98.2 --dport 22 -j ACCEPT

$IPT -I INPUT -p tcp -d 10.0.98.100 --dport 80 -j DROP


$IPT -A INPUT -p icmp --icmp-type 8 -s 192.168.60.111 -j ACCEPT
$IPT -A OUTPUT -p icmp --icmp-type 0 -d 192.168.60.111 -j ACCEPT

$IPT -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -p tcp -d 192.168.60.111 --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT

sysctl -w net.ipv4.ip_forward=1
sysctl -p
```

Task 26: Change iptables to forward packets:

I have added following rules in firewall.sh to change iptables to forward packets.

$IPT -t filter -A FORWARD -i $HIF -j ACCEPT

$IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

```
$IPT -A INPUT -p icmp --icmp-type 8 -s 192.168.60.111 -j ACCEPT
$IPT -A OUTPUT -p icmp --icmp-type 0 -d 192.168.60.111 -j ACCEPT

$IPT -A INPUT -p tcp -s 192.168.60.111 --dport 22 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPT -A OUTPUT -p tcp -d 192.168.60.111 --sport 22 -m conntrack --ctstate ESTABLISHED -j ACCEPT

sysctl -w net.ipv4.ip_forward=1
sysctl -p

$IPT -t filter -A FORWARD -i $HIF -j ACCEPT
$IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

Task 27:Enable SNAT on Server A:

I have added following rule in firewall.sh to enable SNAT on server A. I can access internet from client A.

$IPT -t nat -A POSTROUTING -j SNAT -o $NIF --to $NIP