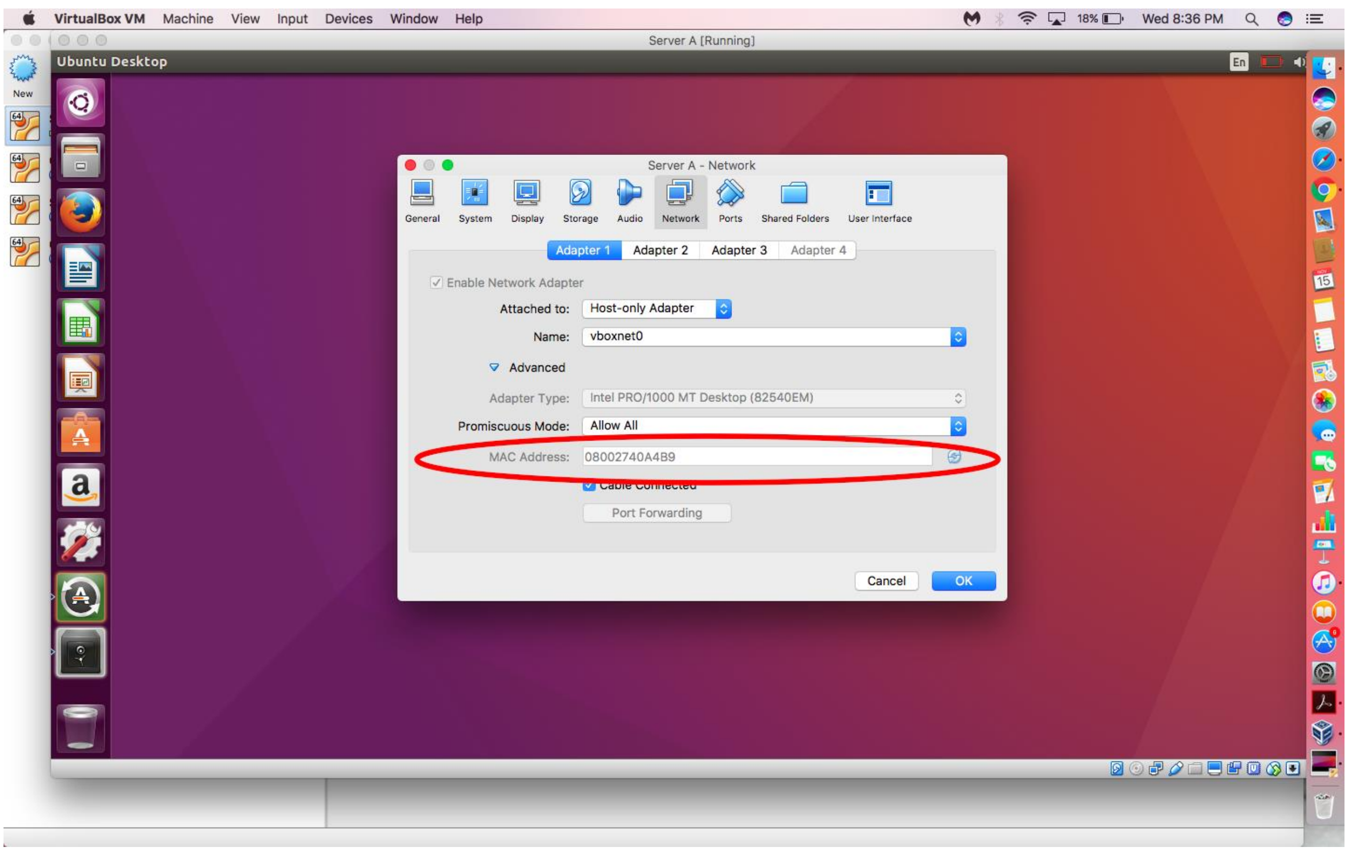
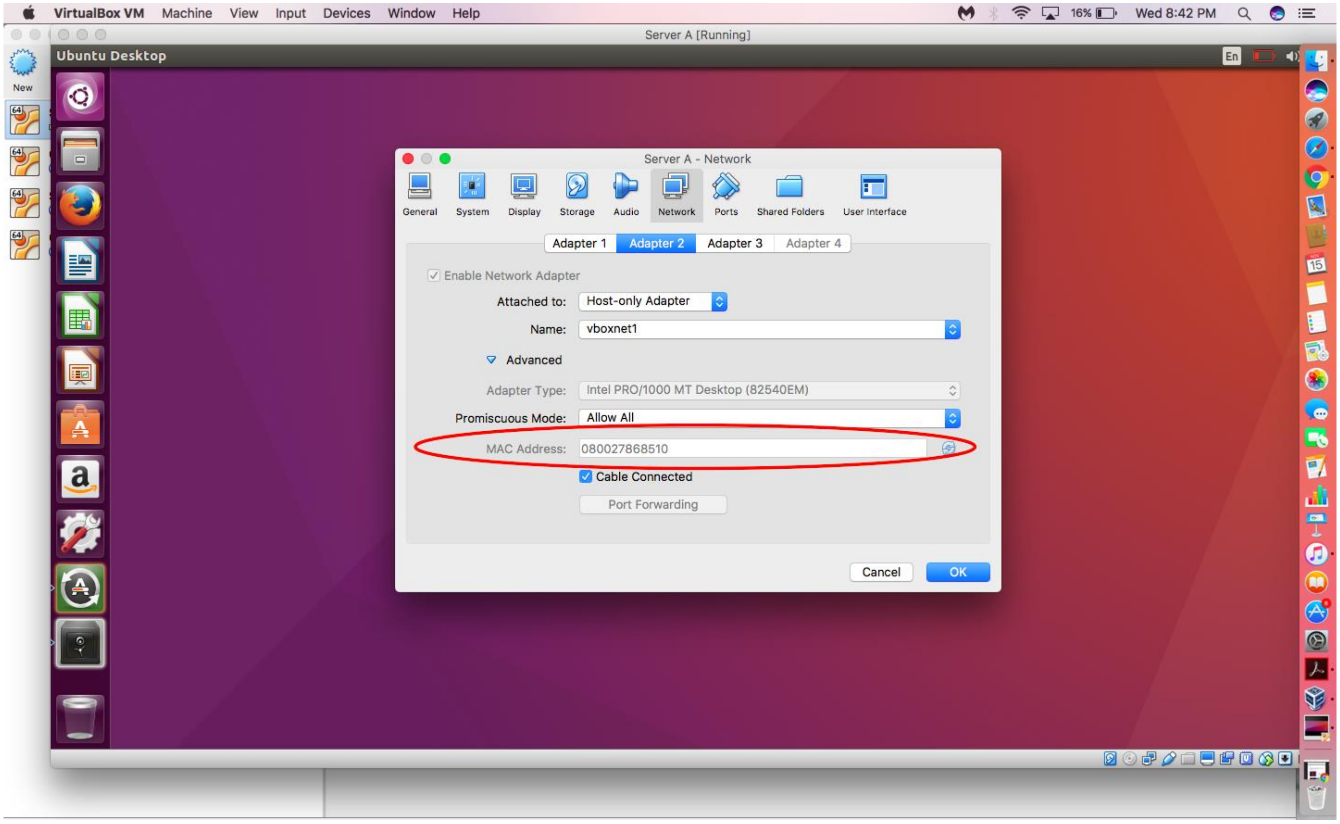
LAB-1 Networking and Firewalls

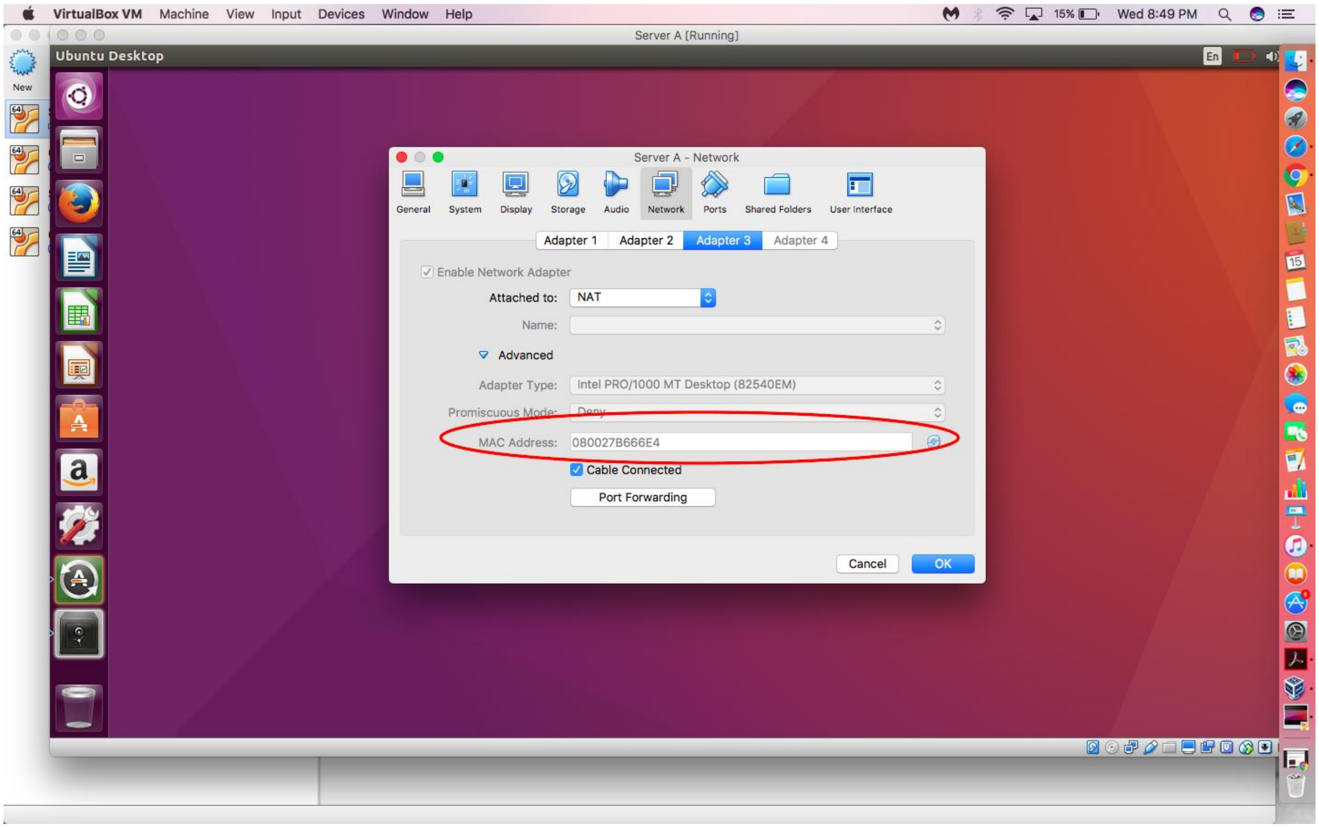
By
Monica Tamanampudi
mota17@student.bth.se
P No 960207-4180

Task 1:

To find out the MAC address of the configured adapters in the web server VM. The figures below show the mac address.



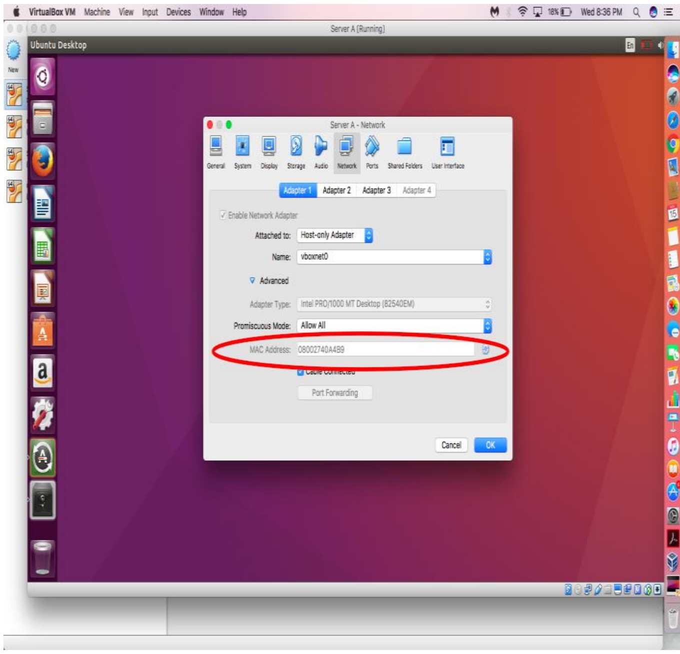

VirtualBox VM Machine View Input Devices Window Help
Wed 8:42 PM
Server A [Running]
Ubuntu Desktop
Server A - Network
General System Display Storage Audio Network Ports Shared Folders User Interface
Adapter 1 Adapter 2 Adapter 3 Adapter 4
Enable Network Adapter
Attached to: Host-only Adapter
Name: vboxnet1
Advanced
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)
Promiscuous Mode: Allow All
MAC Address: 080027868510
Cable Connected
Port Forwarding
Cancel
OK

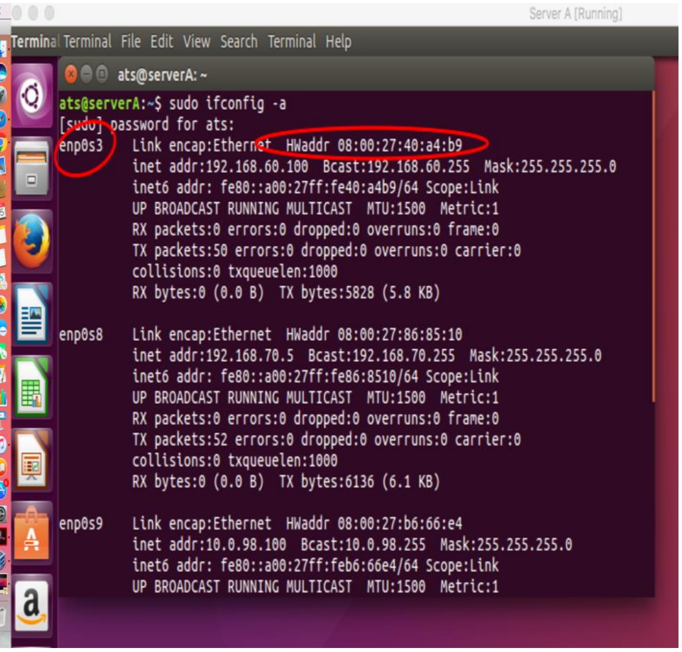


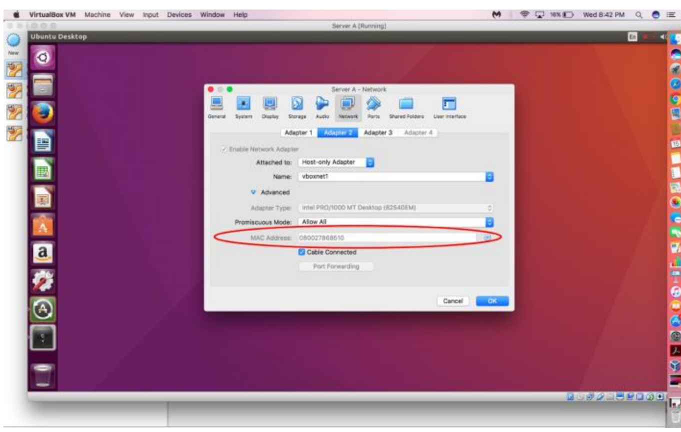
Task 2:

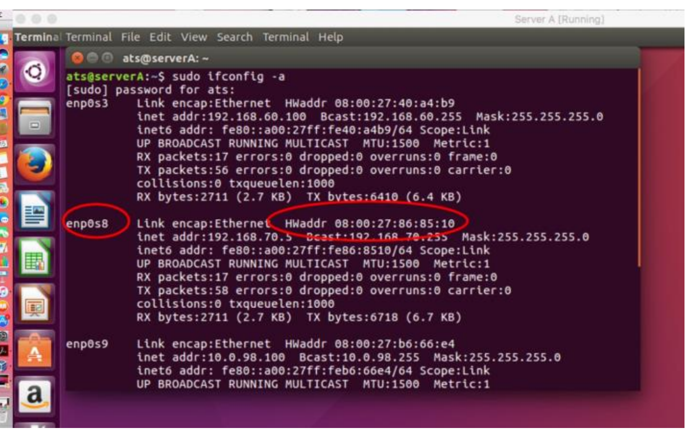
We need to identify which is the NAT interface and which are the host-only interfaces by using the MAC addresses. Enter the following command in the terminal of server A to get the list of available interfaces.

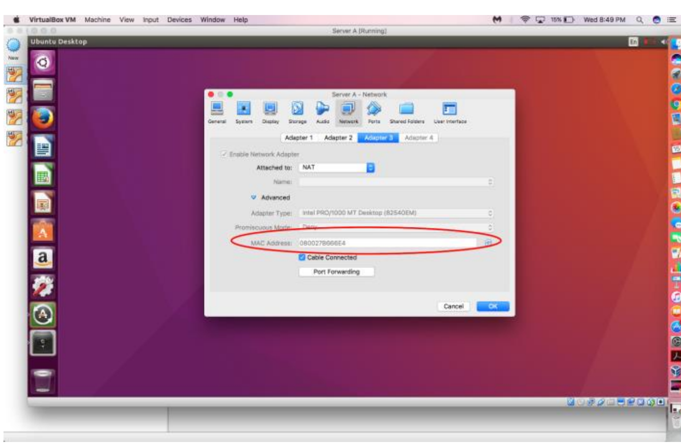
“Sudo ifconfig –a “

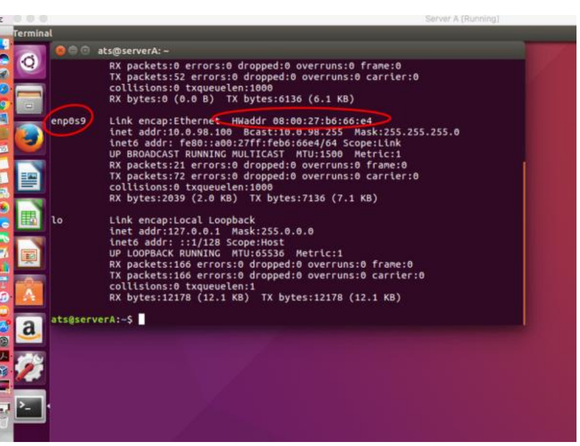



ats@serverA:~$ sudo ifconfig -a
[sudo] password for ats:
enp0s3 Link encap:Ethernet HWaddr 08:00:27:40:a4:b9
inet addr:192.168.60.100 Bcast:192.168.60.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe40:a4b9/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:50 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:5828 (5.8 KB)
enp0s8 Link encap:Ethernet HWaddr 08:00:27:86:85:10
inet addr:192.168.70.5 Bcast:192.168.70.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe86:8510/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:6136 (6.1 KB)
enp0s9 Link encap:Ethernet HWaddr 08:00:27:b6:66:e4
inet addr:10.0.98.100 Bcast:10.0.98.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:feb6:66e4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1




ats@serverA:~$ sudo ifconfig -a
[sudo] password for ats:
enp0s3 Link encap:Ethernet HWaddr 08:00:27:40:a4:b9
inet addr:192.168.60.100 Bcast:192.168.60.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe40:a4b9/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:17 errors:0 dropped:0 overruns:0 frame:0
TX packets:56 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2711 (2.7 KB) TX bytes:6410 (6.4 KB)
enp0s8 Link encap:Ethernet HWaddr 08:00:27:86:85:10
inet addr:192.168.70.5 Bcast:192.168.70.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:fe86:8510/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:17 errors:0 dropped:0 overruns:0 frame:0
TX packets:58 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2711 (2.7 KB) TX bytes:6718 (6.7 KB)
enp0s9 Link encap:Ethernet HWaddr 08:00:27:b6:66:e4
inet addr:10.0.98.100 Bcast:10.0.98.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:feb6:66e4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1




RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:52 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 B) TX bytes:6136 (6.1 KB)
enp0s9 Link encap:Ethernet HWaddr 08:00:27:b6:66:e4
inet addr:10.0.98.100 Bcast:10.0.98.255 Mask:255.255.255.0
inet6 addr: fe80::a00:27ff:feb6:66e4/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:21 errors:0 dropped:0 overruns:0 frame:0
TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:2039 (2.0 KB) TX bytes:7136 (7.1 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:166 errors:0 dropped:0 overruns:0 frame:0
TX packets:166 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:12178 (12.1 KB) TX bytes:12178 (12.1 KB)
ats@serverA:~$

Task 3:

The picture below shows the network address corresponding to their IP address in order to find out the network address of each interface associated with their IP address and their netmask

enp0s3

inet addr : 192.168.60.100 → 11000000 . 10101000 . 00111100 . 01100100

subnet mask : 255.255.255.0 → 11111111 . 11111111 . 11111111 . 00000000

Bitwise AND : 192.168.60.0 → 11000000 . 10101000 . 00111100 . 00000000

Network address — 192.168.60.0

enp0s8

inet addr : 192.168.70.5 → 11000000 . 10101000 . 01000110 . 00000101

subnet mask : 255.255.255.0 → 11111111 . 11111111 . 11111111 . 00000000

bitwise AND : 192.168.70.0 → 11000000 . 10101000 . 01000110 . 00000000

Network Address — 192.168.70.0

enp0s9

inet addr : 10.0.98.100 → 00001010 . 00000000 . 01100010 . 01100100

net mask : 255.255.255.0 → 11111111 . 11111111 . 11111111 . 00000000

bitwise AND : 10.0.98.0 → 00001010 . 00000000 . 01100010 . 00000000

Network Address — 10.0.98.0

lo

inet addr : 127.0.0.1 → 01111111 . 00000000 . 00000000 . 00000001

net mask : 255.255.255.0 → 11111111 . 11111111 . 11111111 . 00000000

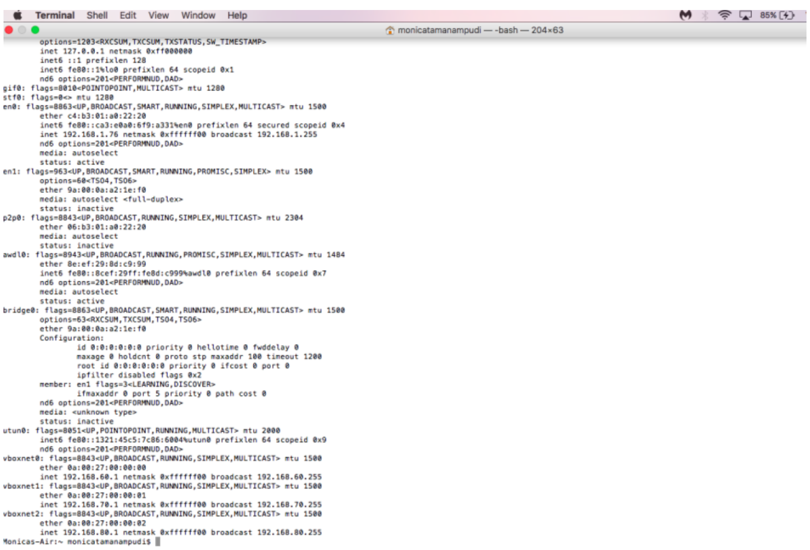
bitwise AND : 127.0.0.0 → 01111111 . 00000000 . 00000000 . 00000000

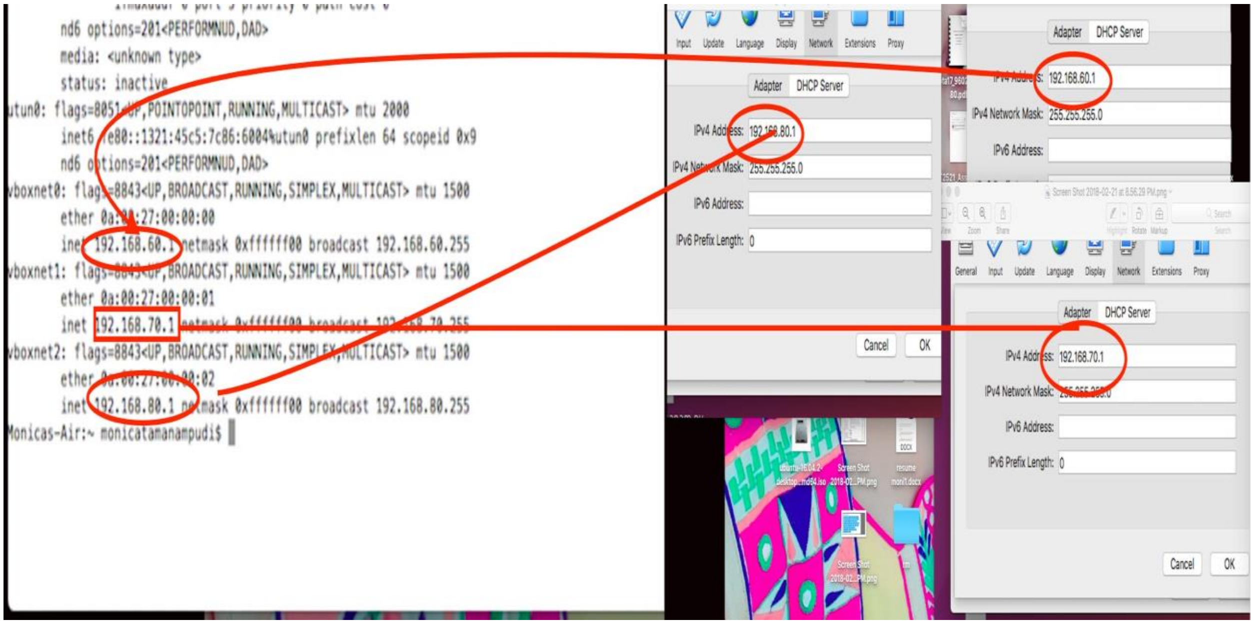
Network Address , 127.0.0.0

Task 4:

To find out the host-only interfaces in the HOST OS. By entering the following command in the terminal of HOST OS we can know the list of available interfaces and from the available interfaces we can find out the host-only interfaces (vboxnet0, vboxnet1, vboxnet 2)

ifconfig -a

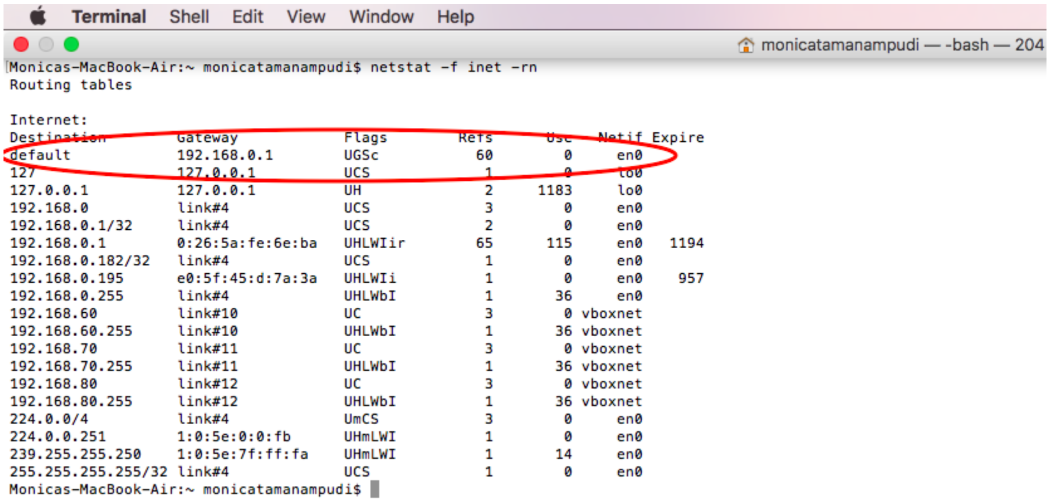




Task 5:

To identify over what interface, we can reach the default gateway for host. enter the following command in the terminal of the HOST OS we can view the routing table.

netstat –f inet -rn

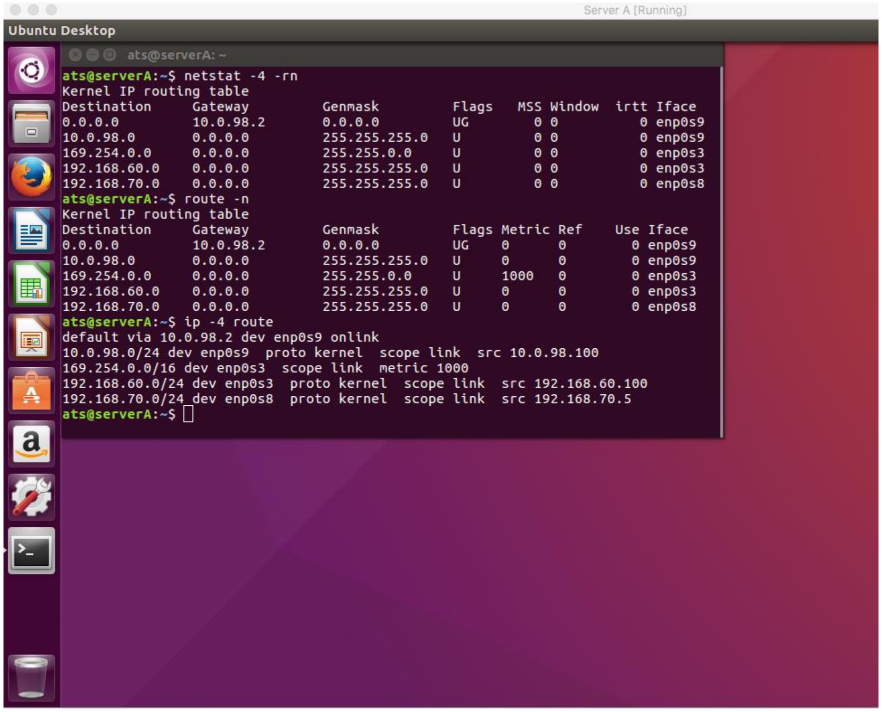


Task 6: To identify the interface through which we can reach the default gateway for my host. By entering the following command in the terminal of guest OS we can view the routing table. The default gateway is 10.0.98.2.

"Netstat -4 -rn"

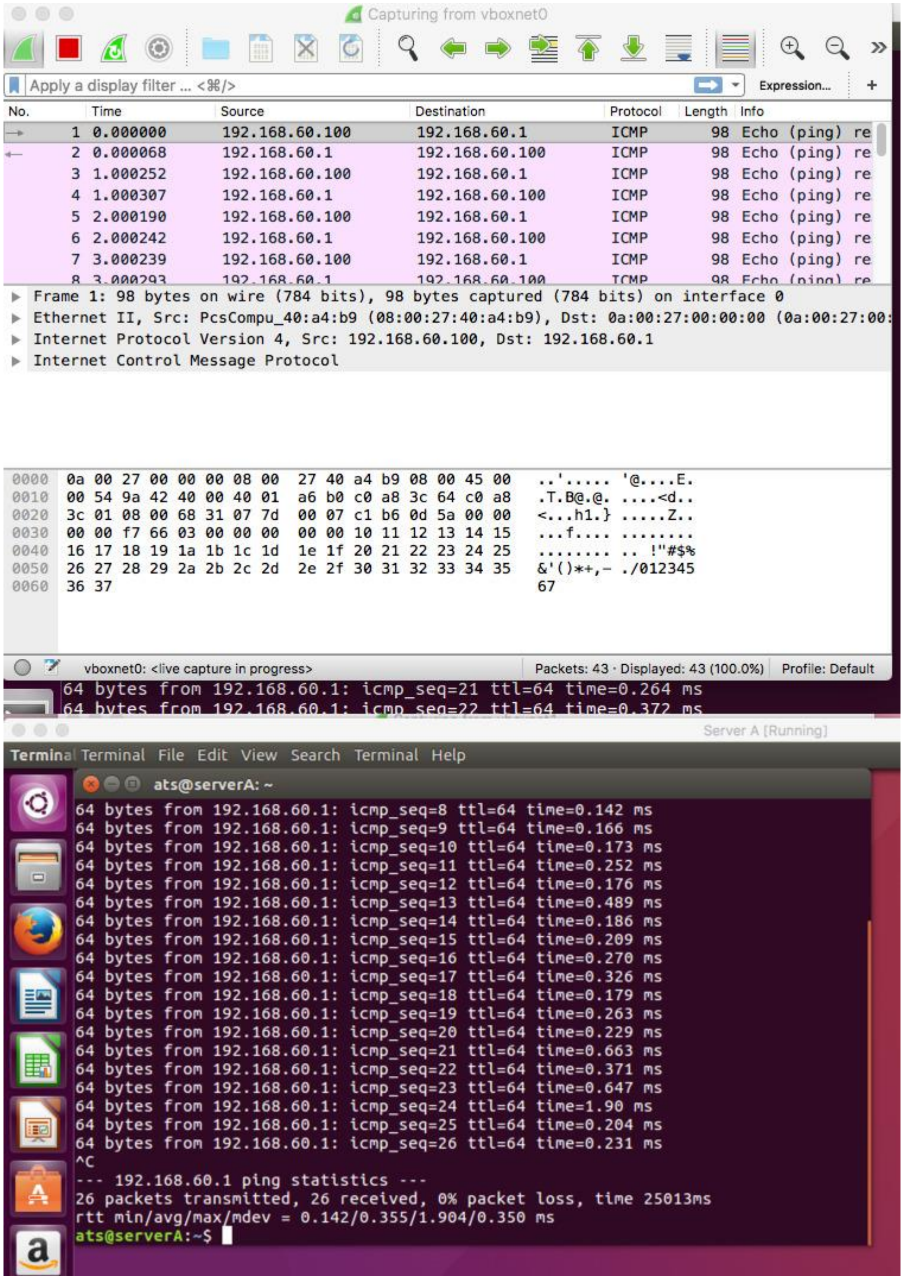
"Route -n"

"Ip -4 route"



Task 7:

To ping the IP address corresponding to the host-only interface in the host OS and capture the packets in the Wireshark. To examine the icmp traffic

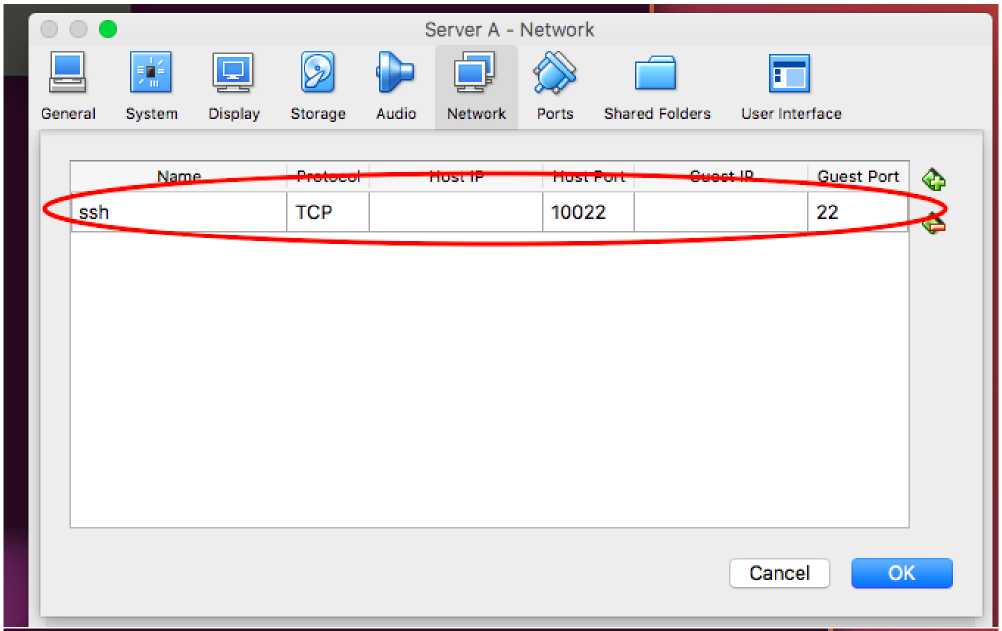


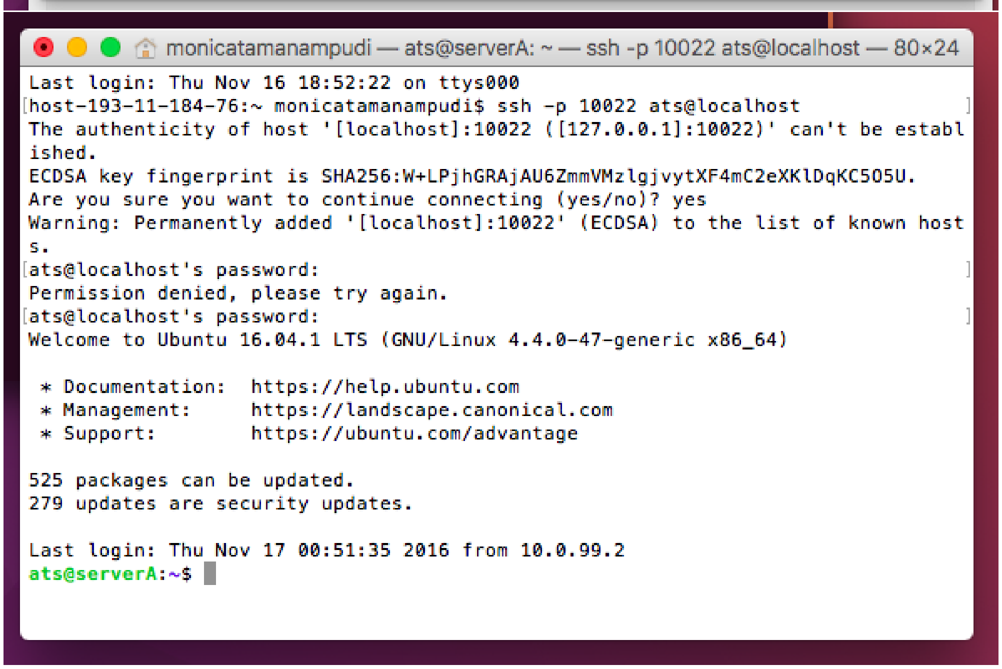
From this we can conclude that the ping and the Wireshark capturing of the icmp traffic are identical

Task 8:

To ssh into VM via localhost from the HOST OS. Enter the following command on the host

ssh -p 10022 ats@localhost

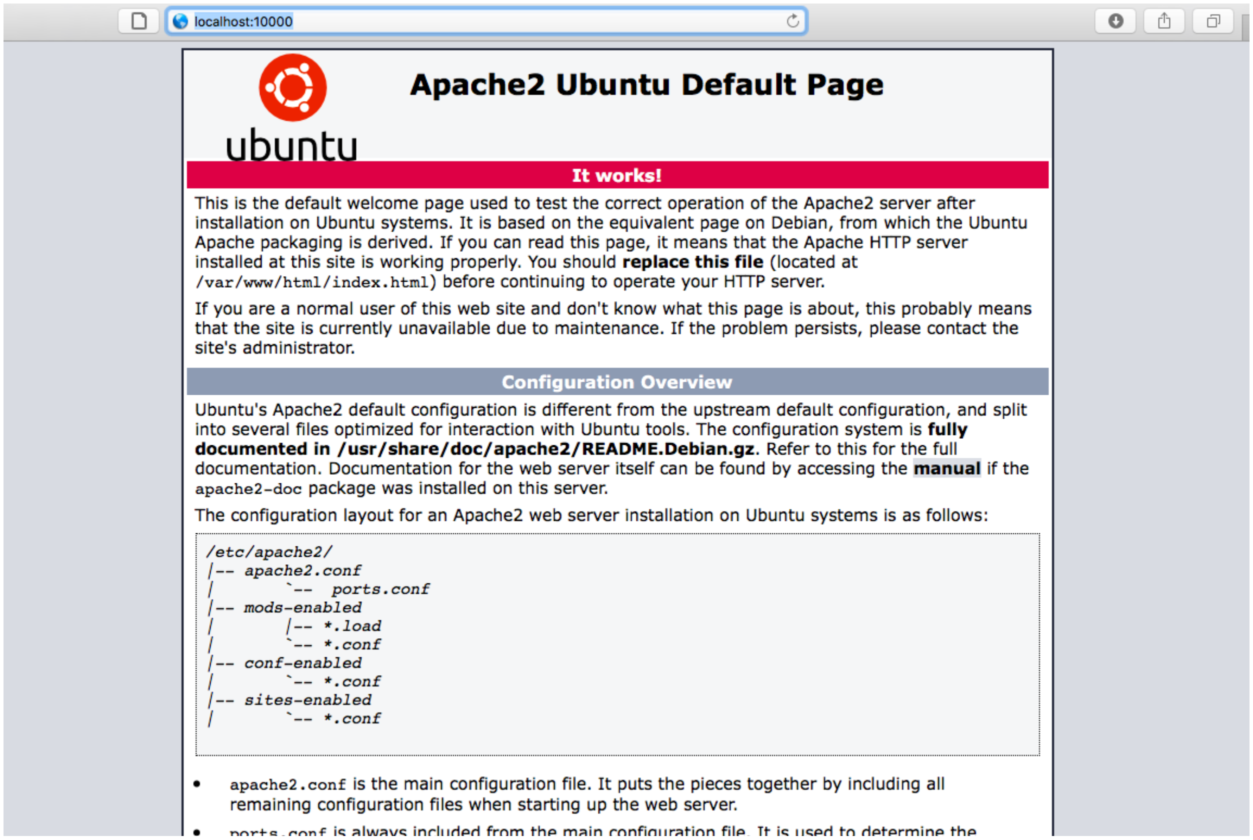


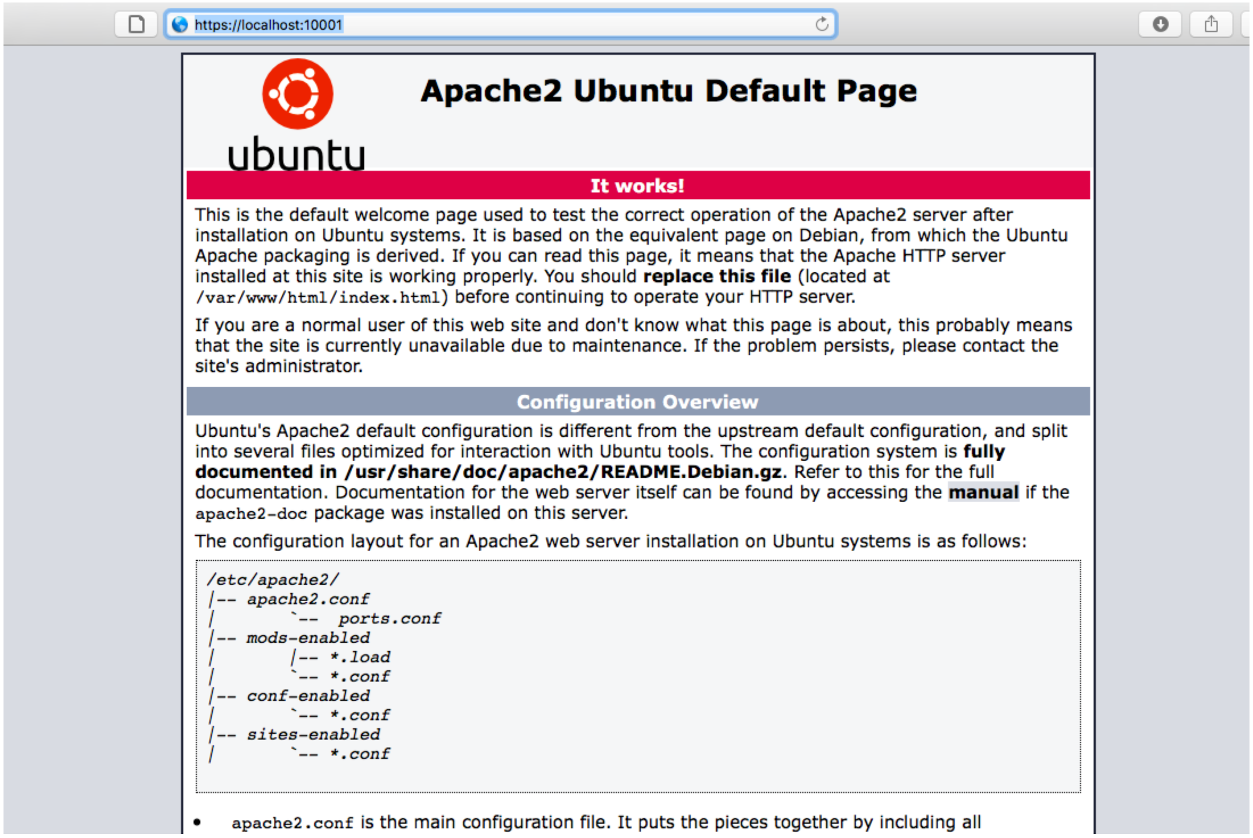


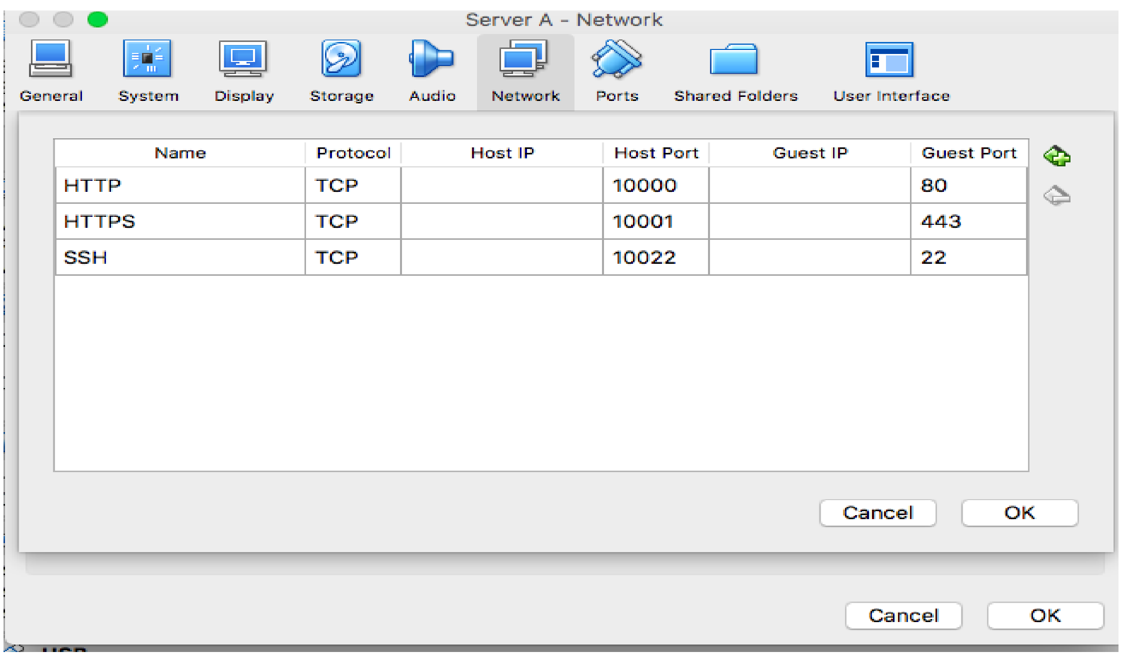
Task 9:

To add the forwarding rules for HTTP and HTTPS in Virtual Box, so that the host user can view the HTTP and HTTPS content of the apache2 server in the guest OS. In order to do this we are using port 10000 for HTTP and 10001 for HTTPS in the host OS and forwarding these ports to the official ports for

HTTP (80) and HTTPS (443) of the guest OS. The images below show that the host OS can view the HTTP and HTTPS content of the apache2 server in the host OS.




https://localhost:10001
ubuntu
Apache2 Ubuntu Default Page
It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all


Server A - Network
General
System
Display
Storage
Audio
Network
Ports
Shared Folders
User Interface
Name
Protocol
Host IP
Host Port
Guest IP
Guest Port
HTTP
TCP
10000
80
HTTPS
TCP
10001
443
SSH
TCP
10022
22
Cancel
OK
Cancel
OK

Task 10:

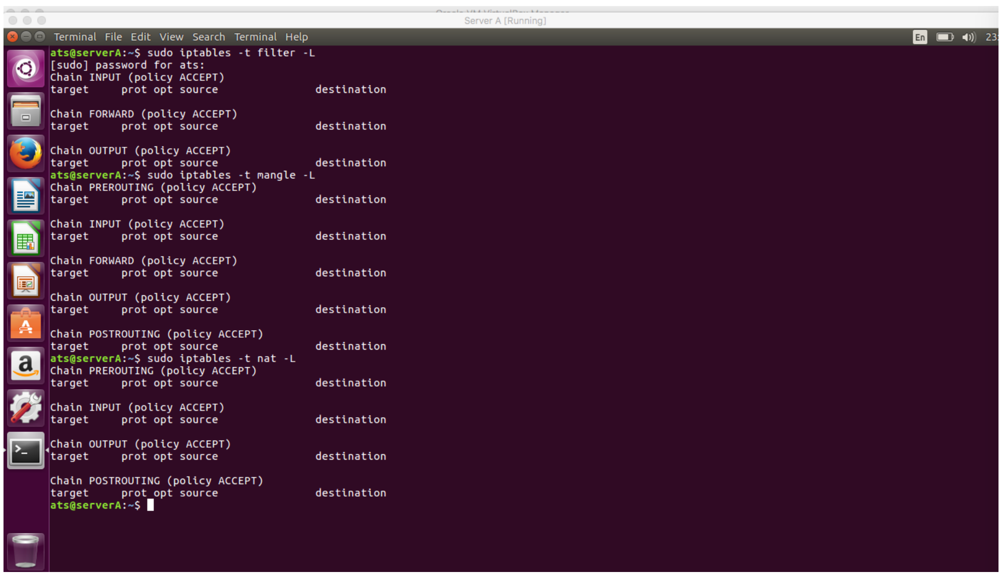
The commands to view the default rules are.

“sudo iptables –t filter -L”

“sudo iptables –t mangle -L”

“sudo iptables –t nat -L”

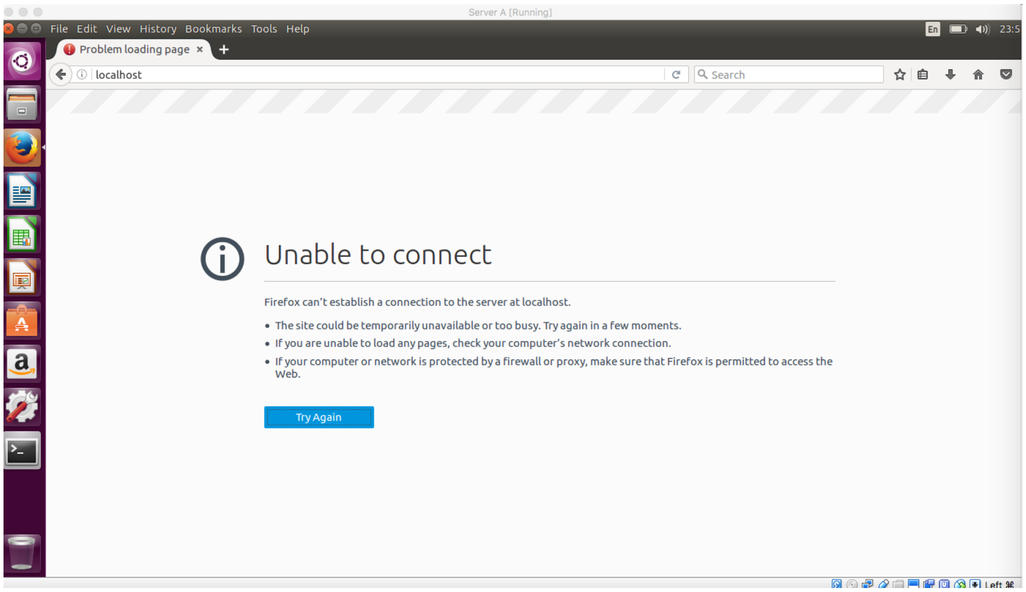
In the picture below the default policy and rules of the tables are shown (filter, mangle, nat)



Task 11:

To block the HTTP-browsing in the guest OS. To block the HTTP browsing we need to block the INPUT chain of the filter table for port number 80. The following command will block the HTTP browsing in the guest OS.

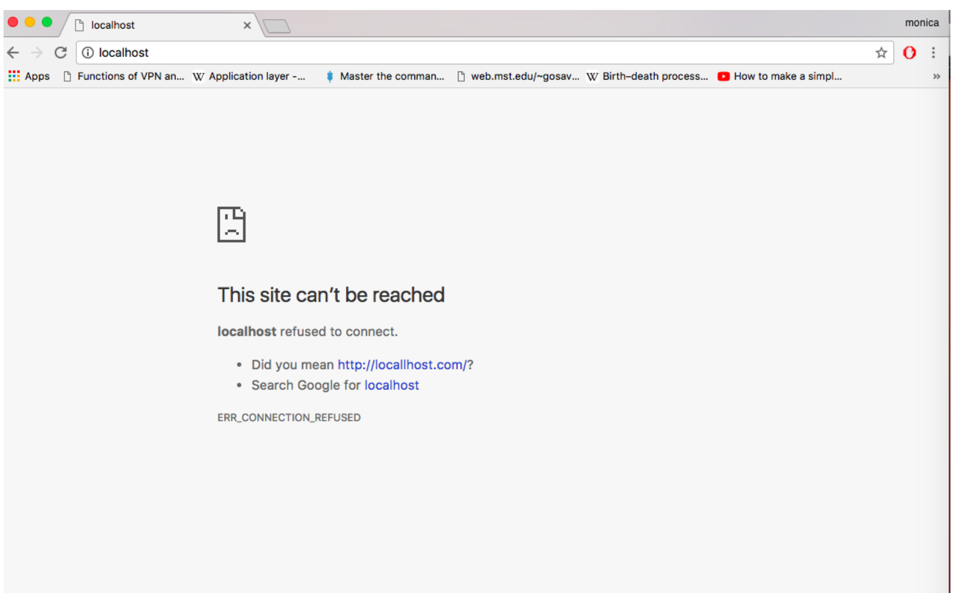
“sudo iptables –A INPUT –p tcp --dport 80 –j REJECT”



Task 12:

To block the HTTP-browsing in the host OS. To block the HTTP browsing we need to block the OUTPUT chain of the filter table for port number 80. The following command will block the HTTP browsing in the guest OS.

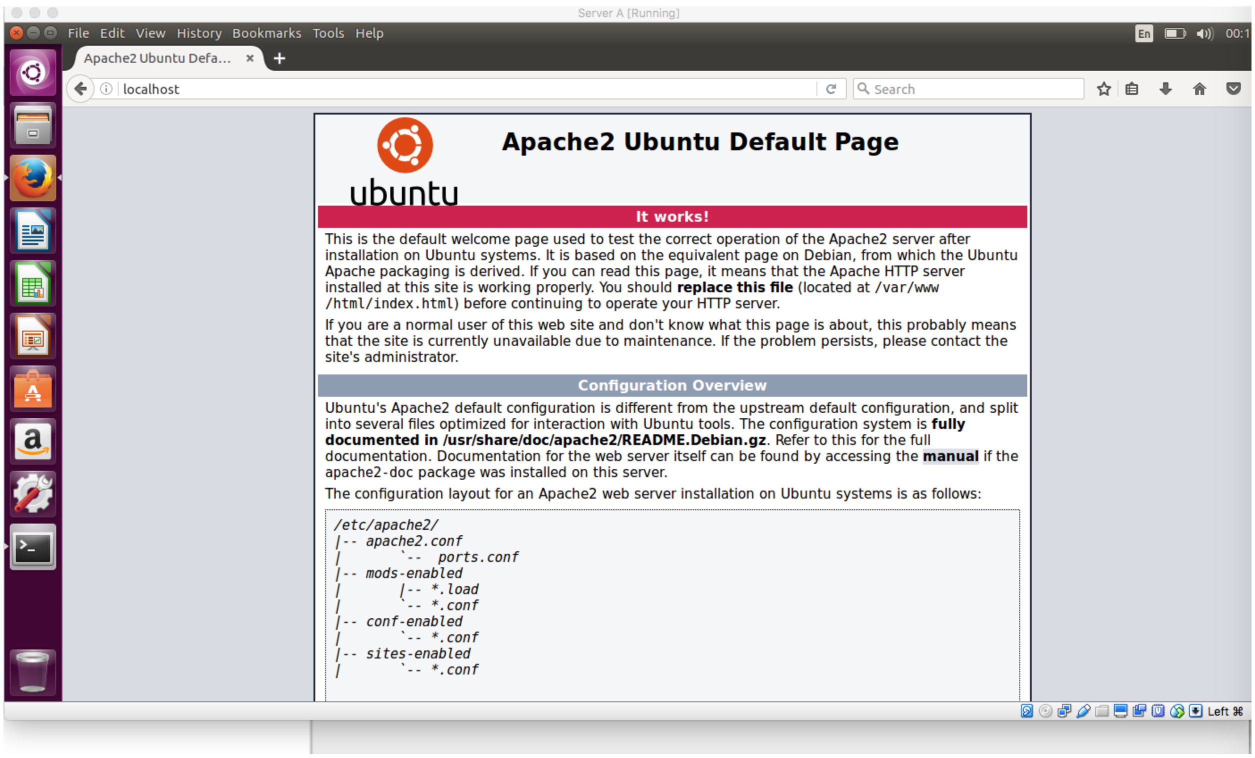
"sudo iptables –A OUTPUT –p tcp --dport 80 –j REJECT"



Task 13:

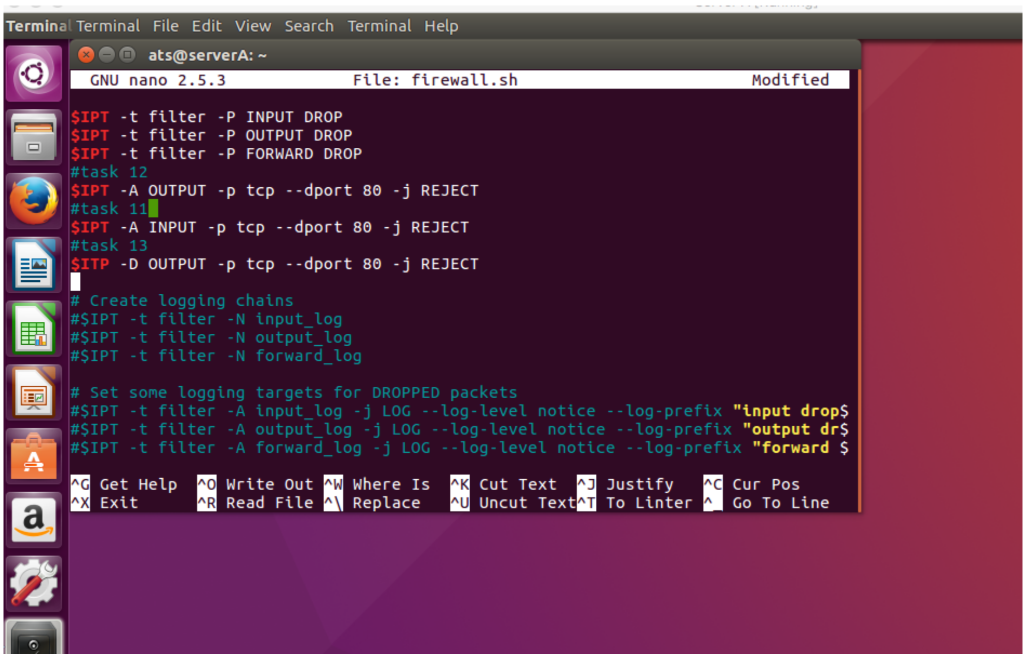
To unblock the HTTP-browsing in the guest OS. To block the HTTP browsing we need to block the INPUT chain of the filter table for port number 80. The following command will unblock the HTTP browsing in the guest OS.

“sudo iptables –D INPUT –p tcp --dport 80 –j REJECT”



Task 14:

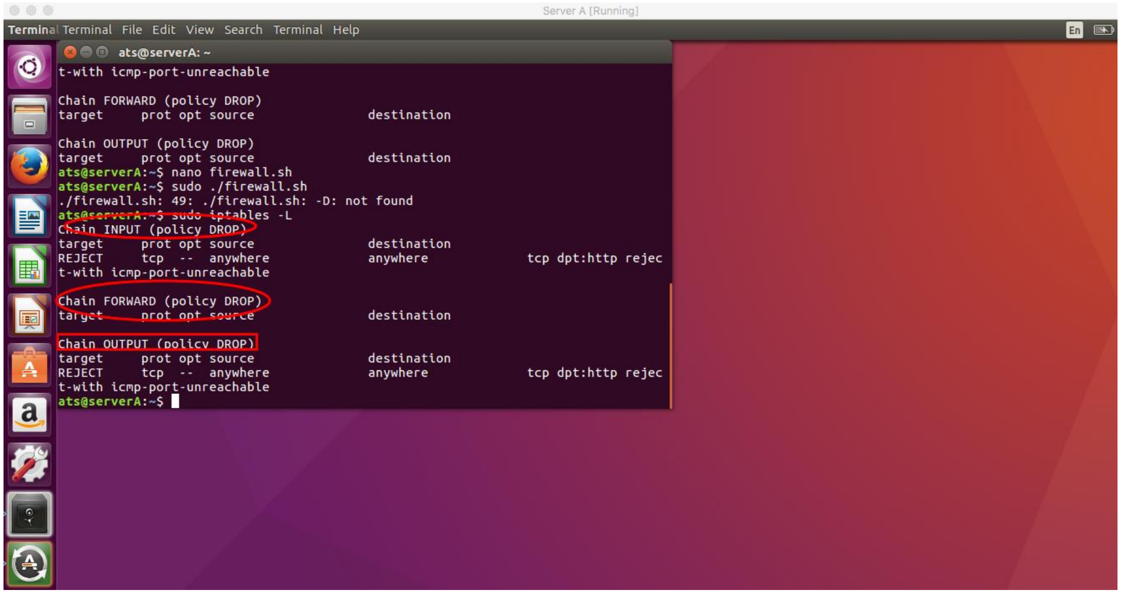
To modify the script firewall.sh to bring this server A firewall to the state we had in task 13 guest OS can view HTTP and HTTPS pages, but apache2 server is blocked from serving the HTTP content



Task 15:

To change the default firewall policy to DROP. Add the following commands in the firewall.sh script and executing the script, the firewall policy will be changed to DROP.

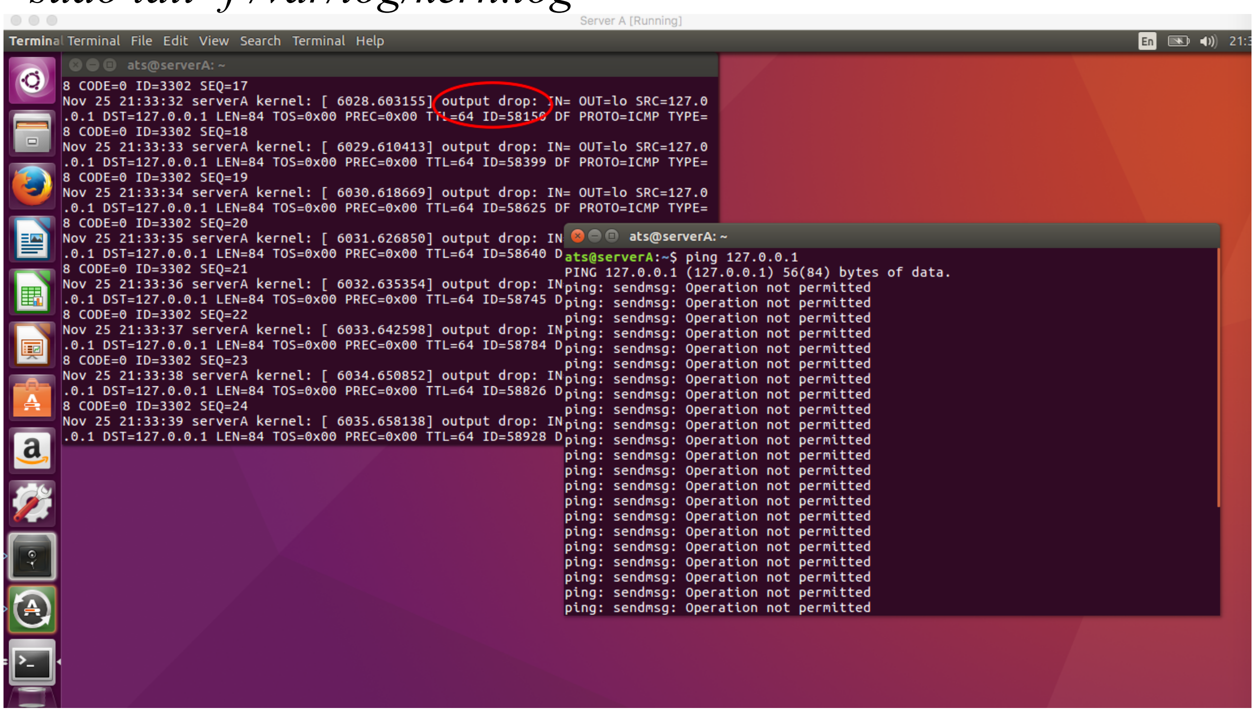
"$IPT –P INPUT DROP"
"$IPT –P OUTPUT DROP"
"$IPT –P FORWARD DROP"



Task 16:

To see the live logs of linux kernel by entering the following command. In task 15 since we have set the default firewall policy to drop and so the packets are being dropped.

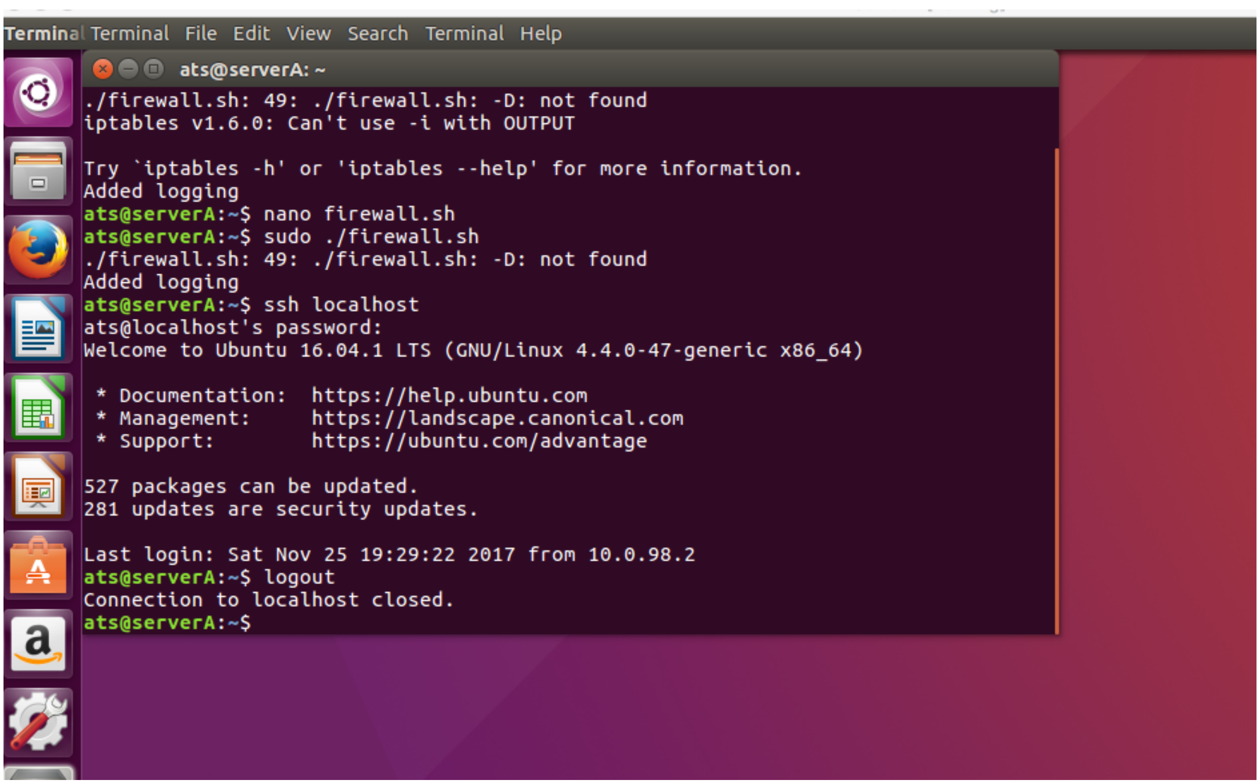
"sudo tail -f /var/log/kern.log"



Task 17:

To fix the firewall rules such that all type of traffic to and from loopback interface is enabled. By writing the following commands in the firewall script and executing it, we can enable the traffic for loopback interfaces.

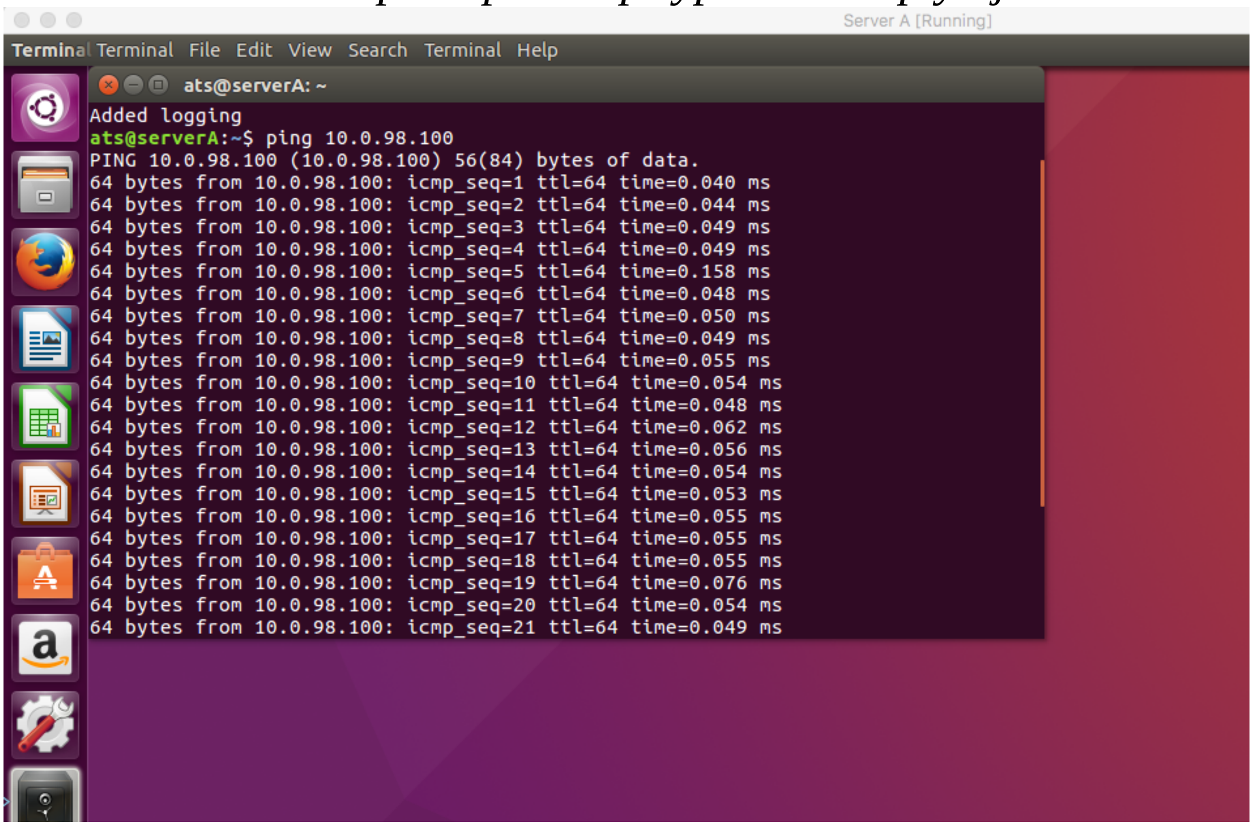
“$IPT –A INPUT –i lo -j ACCEPT”
“$IPT –A OUTPUT –o lo –j ACCEPT”



Task 18:

To allow ping traffic initiated from Server A. For ping traffic, we need to allow outgoing ICMP Echo Request and incoming ICMP Echo Reply messages. Enter the following commands.

“$IPT –A OUTPUT –p icmp --icmp-type echo-request –j ACCEPT”
“$IPT –A INPUT –p icmp --icmp-type echo-reply –j ACCEPT”

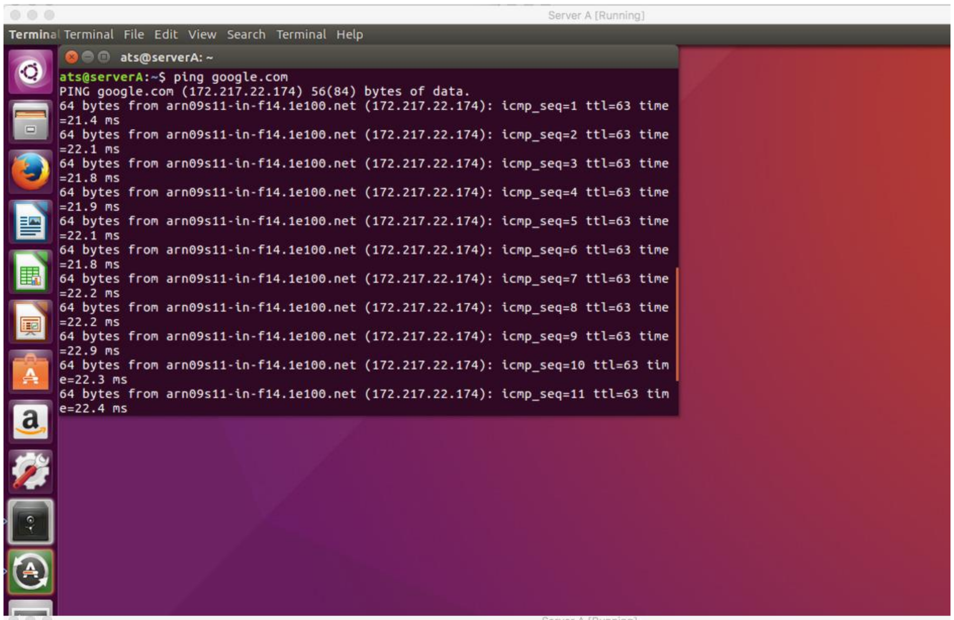


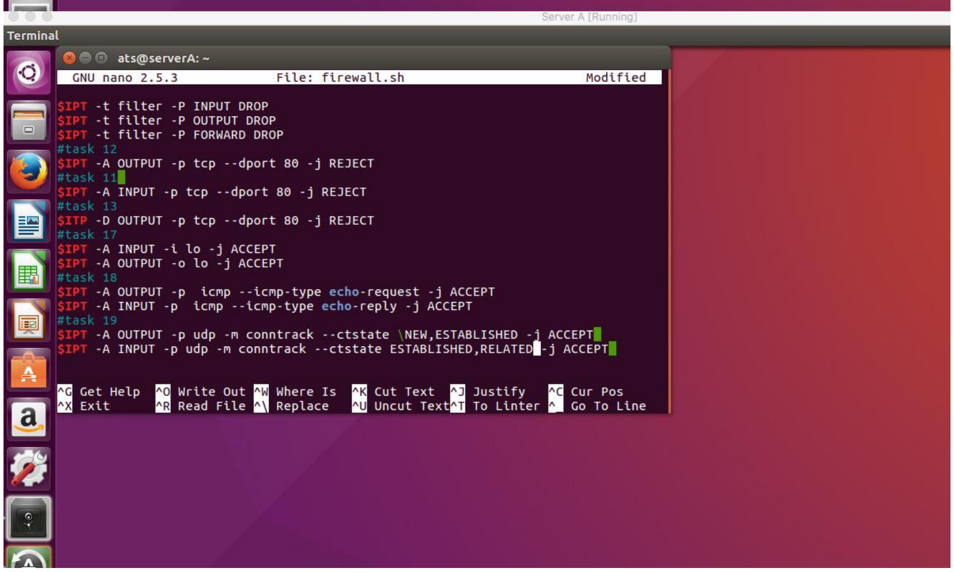
Task 19:

To allow the server to ping all hosts. By adding the following rules to the firewall.sh script and executing it, we are allowing the firewall to accept the outgoing ICMP traffic to any server and corresponding ICMP replies.

“$IPT –A OUTPUT –p udp –m conntrack --ctstate \NEW,ESTABLISHED –j ACCEPT ”

“$IPT –A INPUT –p udp –m conntrack --ctstate ESTABLISHED,RELATED –j ACCEPT ”



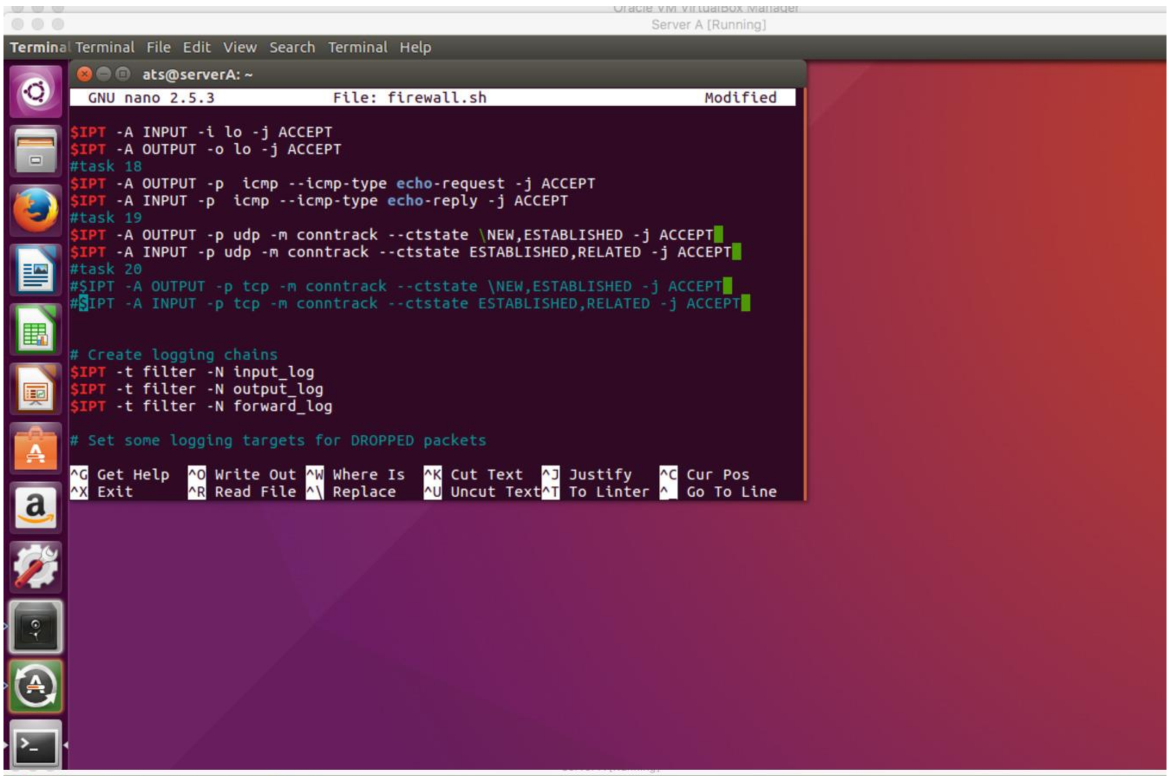


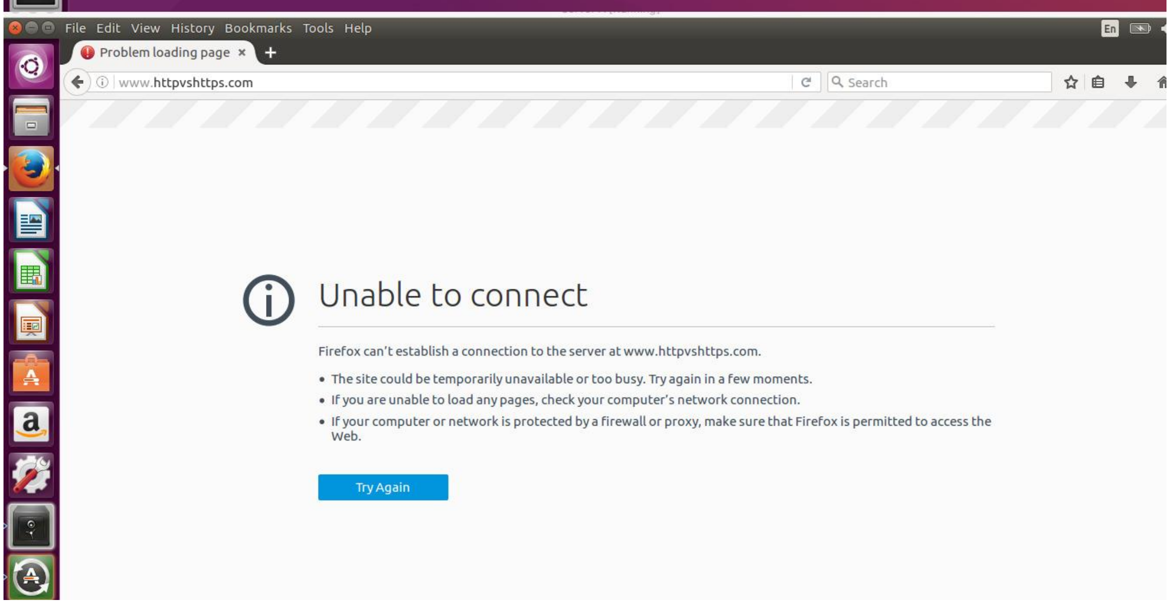
Task 20:

To add the following rules to the firewall.sh script and executing it and thus enable TCP connections to be established to any destination, so we can able to browse websites with the Firefox browser from Server A.

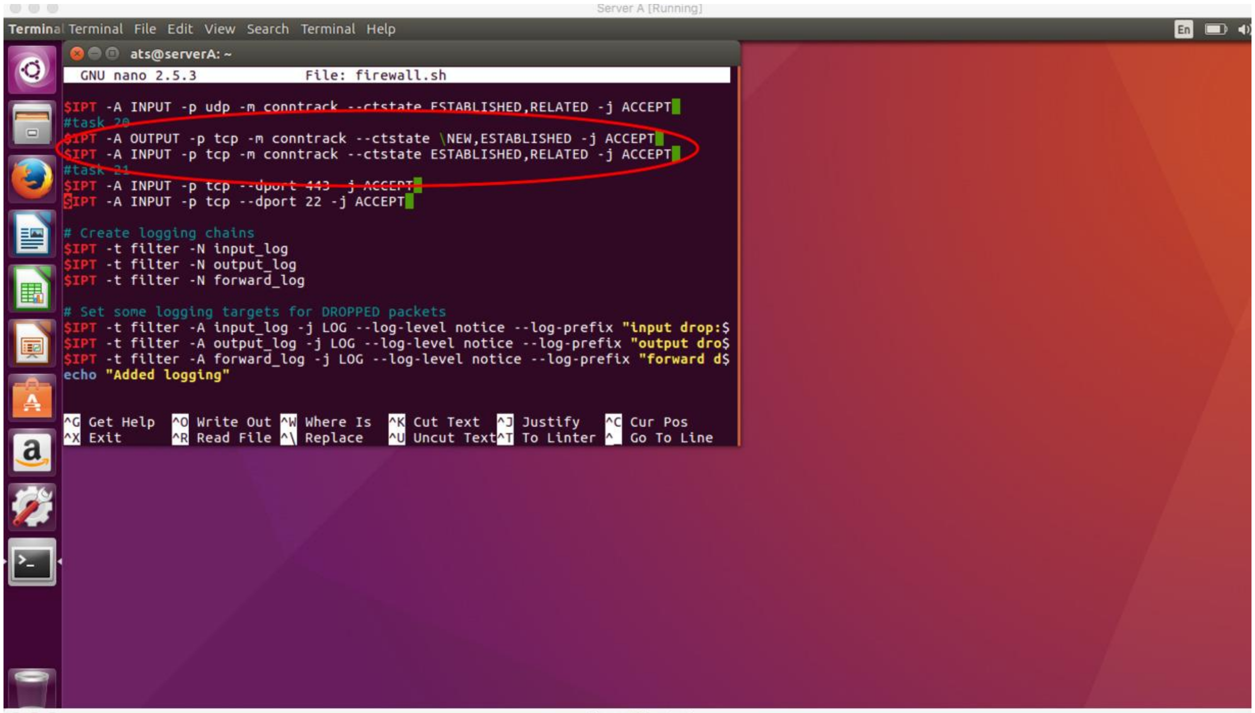
"$IPT –A OUTPUT –p tcp –m conntrack --ctstate \NEW,ESTABLISHED –j ACCEPT "
"$IPT –A INPUT –p tcp –m conntrack --ctstate ESTABLISHED,RELATED –j ACCEPT

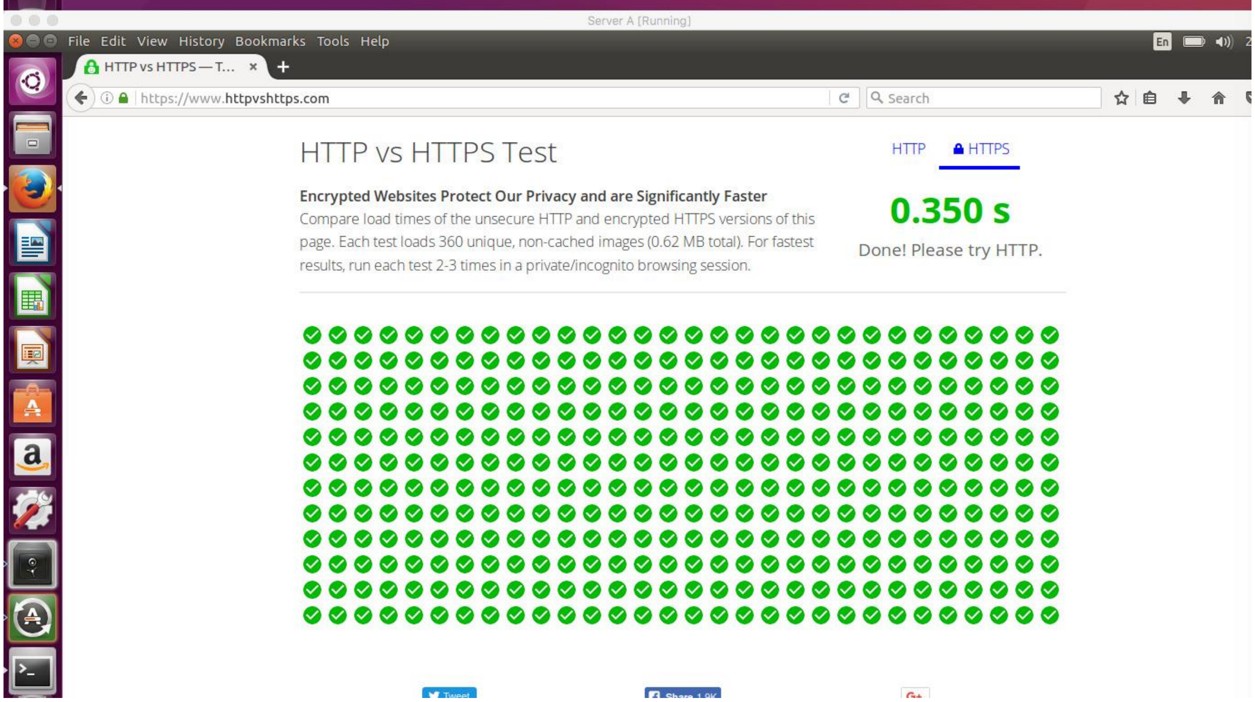
The images below show that the tcp connection is not established for the website by commenting the rules above (http://www.httpvshttps.com/)





The images below show that the tcp connection is established by adding the rules above.

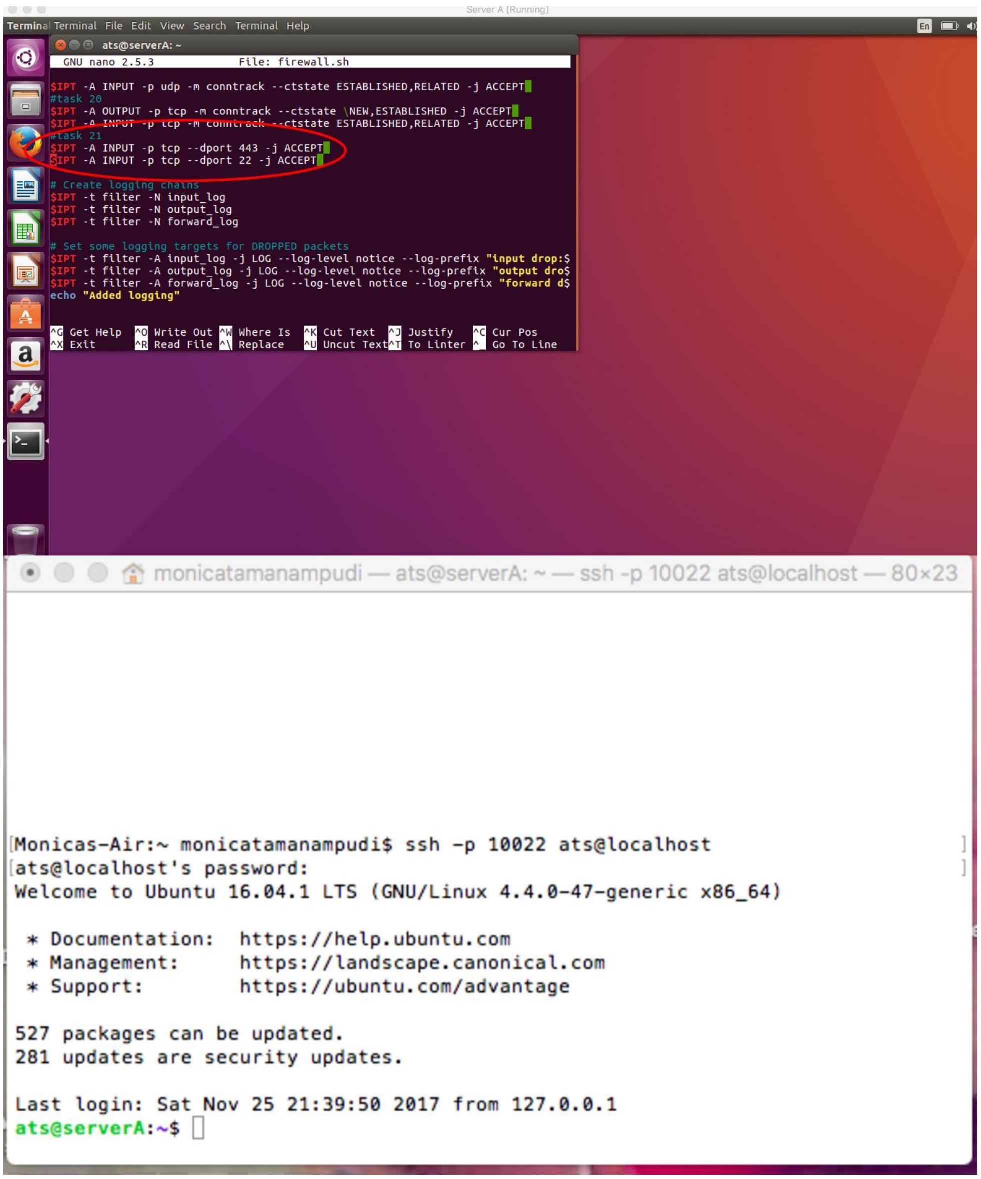




Task 21:

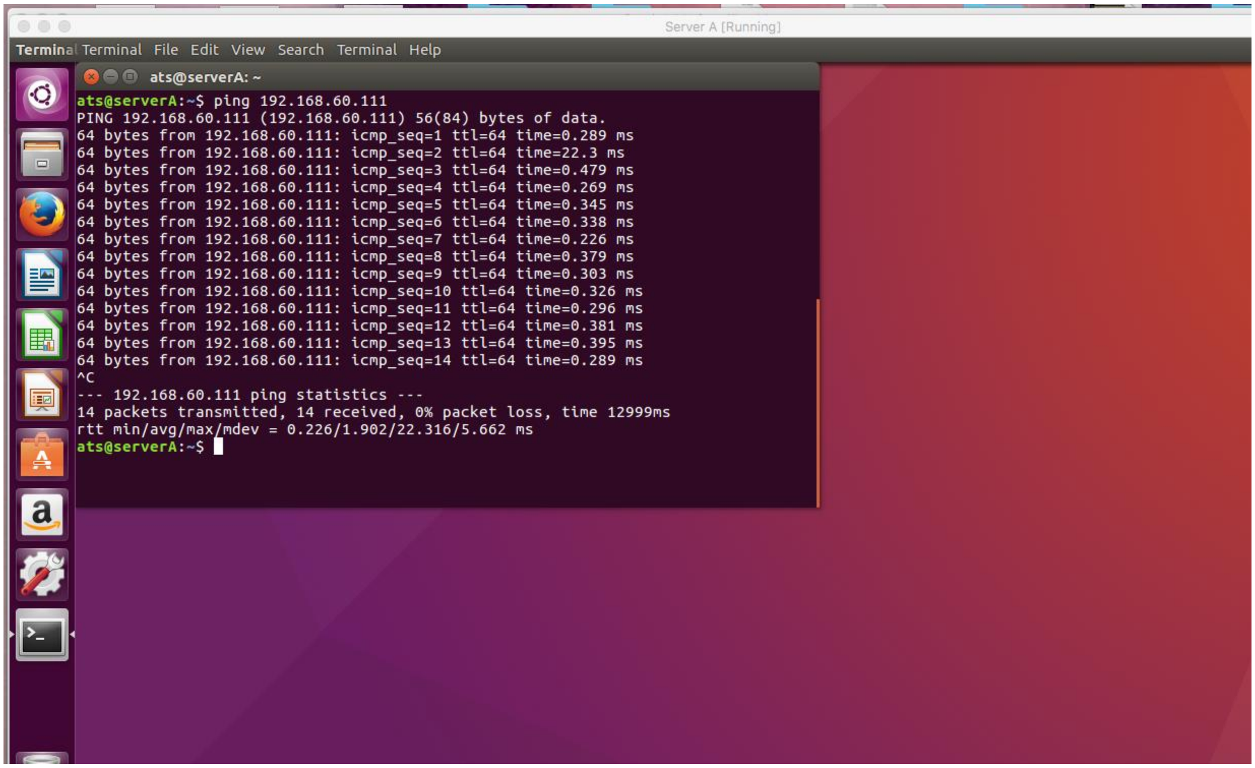
To Enable SSH and HTTPS content from apache2 server for web browser on HOST, add the commands below.

“$IPT –A INPUT –p tcp --dport 443 –j ACCEPT”
“$IPT –A INPUT –p tcp --dport 22 –j ACCEPT



Task 22:

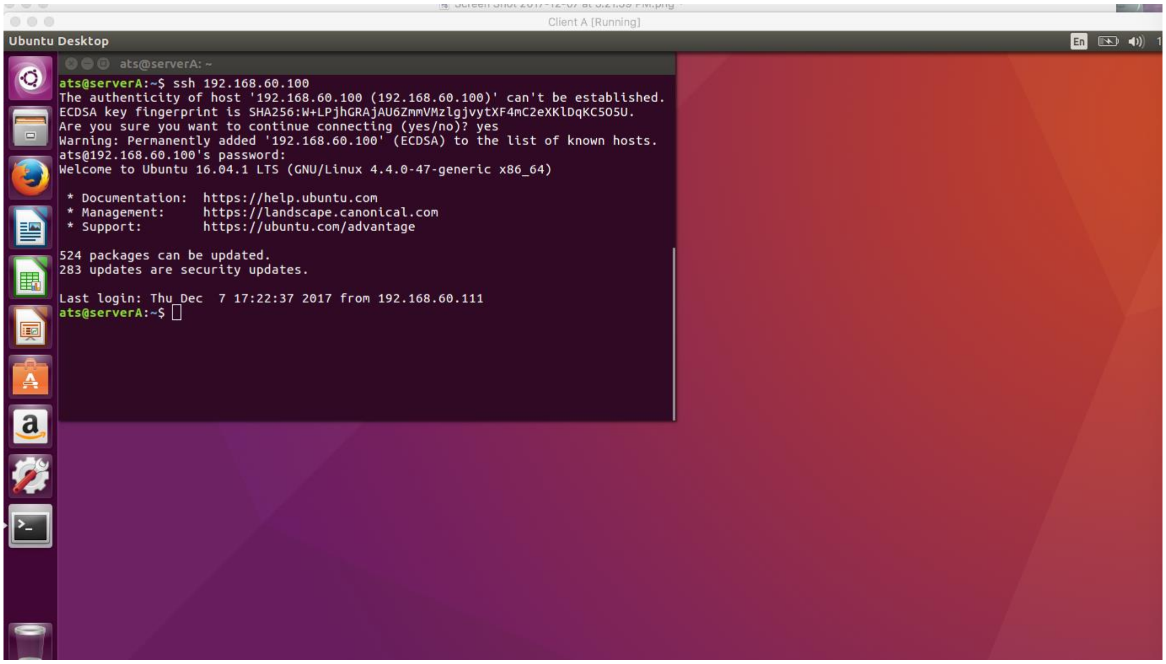
To ping server A from client A add rules in firewall.sh



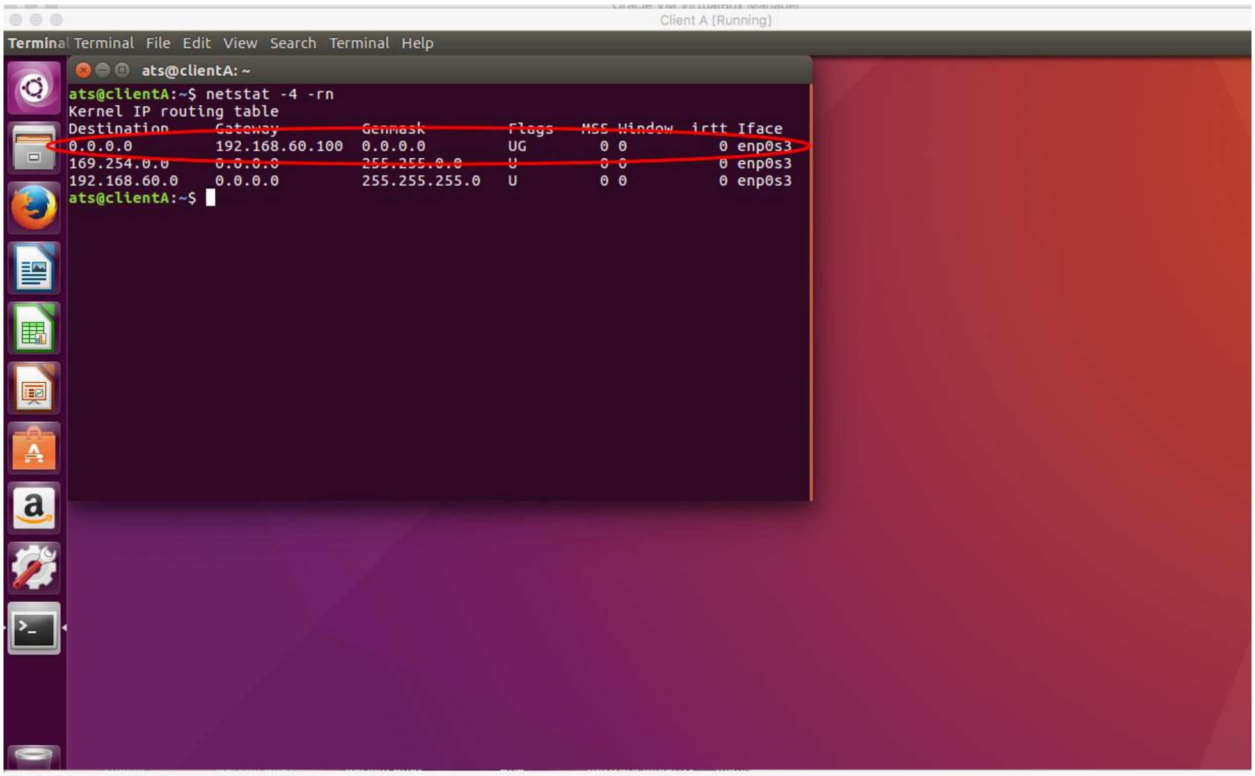
Task 23:

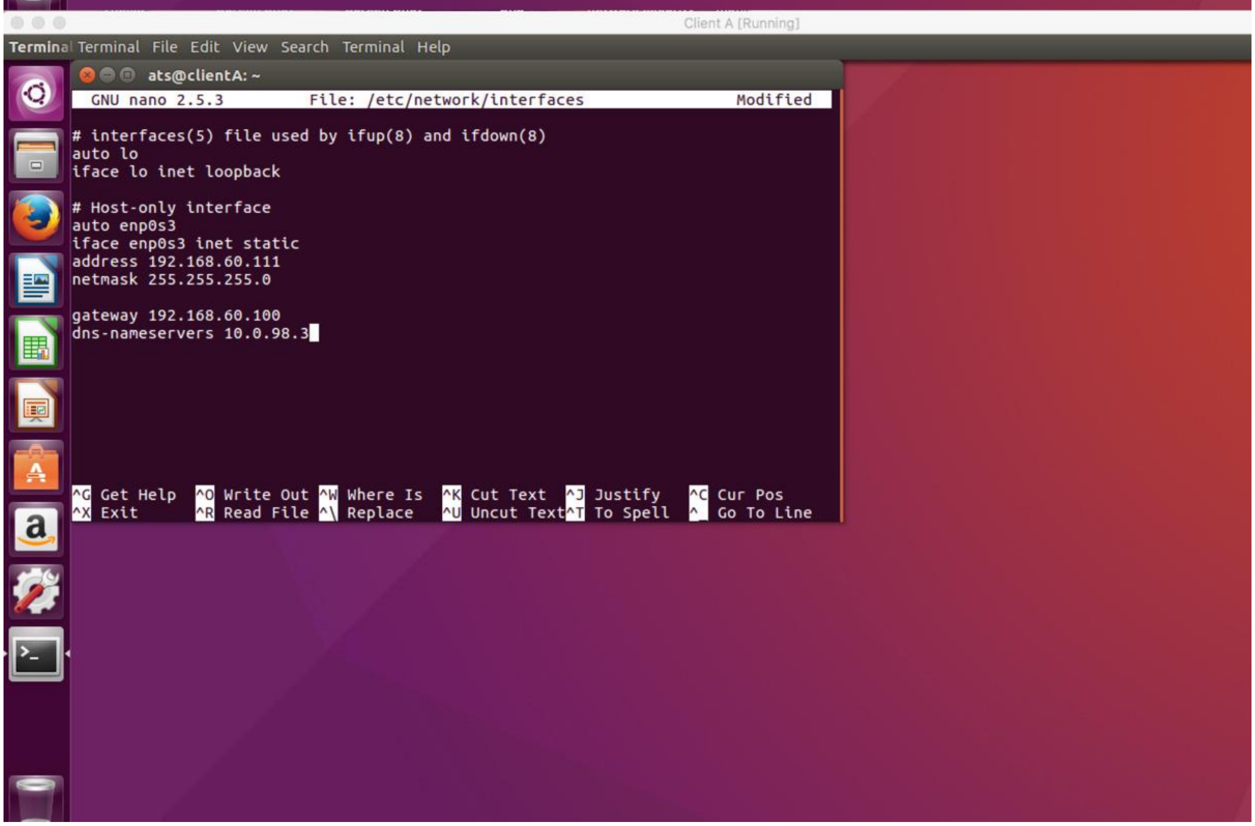
To SSH form server A to client A add the following rules

$IPT -A INPUT –p tcp –dport 22 -j ACCEPT



Task 24:
Add the gateway and dns-name server in /etc/network/interfaces file in client A.


Client A [Running]
Terminal Terminal File Edit View Search Terminal Help
ats@clientA: ~
ats@clientA:~$ netstat -4 -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 192.168.60.100 0.0.0.0 UG 0 0 0 enp0s3
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 enp0s3
192.168.60.0 0.0.0.0 255.255.255.0 U 0 0 0 enp0s3
ats@clientA:~$


Client A [Running]
Terminal Terminal File Edit View Search Terminal Help
ats@clientA: ~
GNU nano 2.5.3 File: /etc/network/interfaces Modified
interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
Host-only interface
auto enp0s3
iface enp0s3 inet static
address 192.168.60.111
netmask 255.255.255.0
gateway 192.168.60.100
dns-nameservers 10.0.98.3
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

Task 25:

Execute the following command in the terminal of Server A so that IP forwarding is enabled on the Server A. This will forward the packets from enp0s3 to enp0s9

“sudo sysctl -w net.ipv4.ip_forward=1”

“sudo sysctl –p”

Task 26:

Change the iptables rules to forward packets. add the following rules to forward packets from enp0s3 to enp0s9.

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

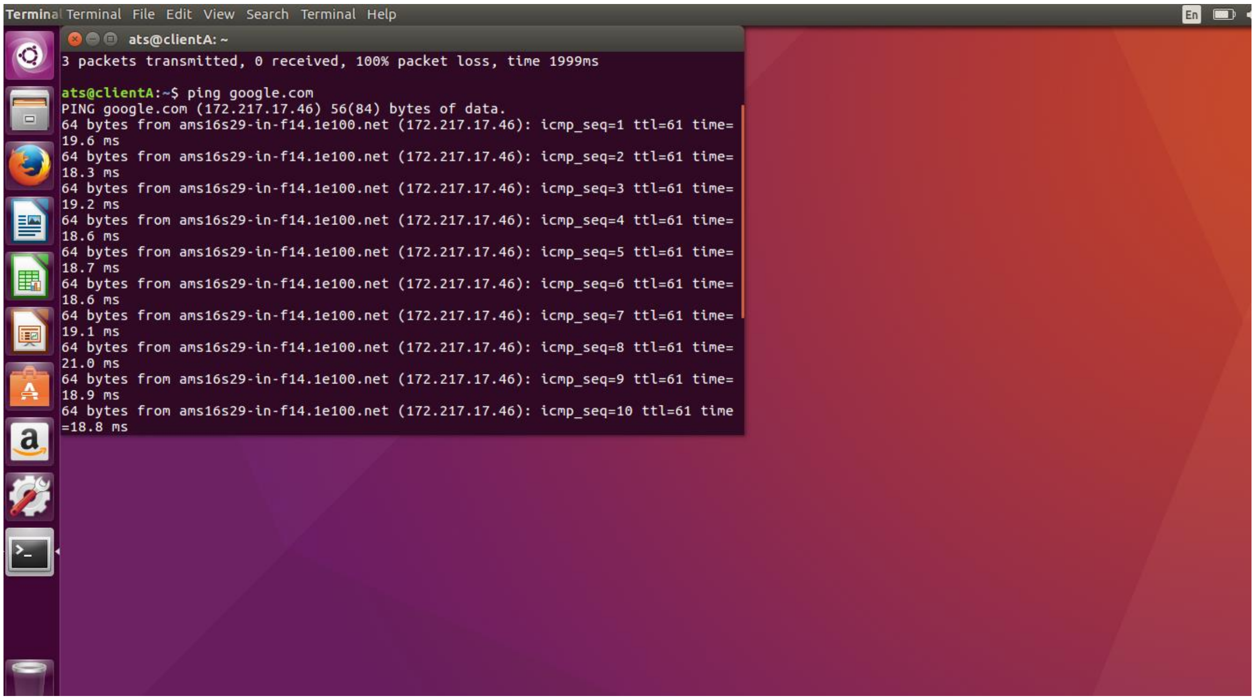
“$IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT”

And then the packets are forwarded to the NAT interface. But the problem is that Client A uses private address (192.168.60.111) and all routers will have a basic default rule to drop packets coming from the private addreses. So we need to inform Server A to use NAT (more specifically Source NAT - SNAT). To do this we need to enable the SNAT on Server A.

Task 27:

To fix the problem in the above task you need to inform Server A to do SNAT on the NAT interface and add the following iptables rule.

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


Terminal Terminal File Edit View Search Terminal Help
ats@clientA: ~
3 packets transmitted, 0 received, 100% packet loss, time 1999ms
ats@clientA:~$ ping google.com
PING google.com (172.217.17.46) 56(84) bytes of data.
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=1 ttl=61 time=19.6 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=2 ttl=61 time=18.3 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=3 ttl=61 time=19.2 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=4 ttl=61 time=18.6 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=5 ttl=61 time=18.7 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=6 ttl=61 time=18.6 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=7 ttl=61 time=19.1 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=8 ttl=61 time=21.0 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=9 ttl=61 time=18.9 ms
64 bytes from ams16s29-in-f14.1e100.net (172.217.17.46): icmp_seq=10 ttl=61 time=18.8 ms