LAB – 1 Networking and Firewalls

By

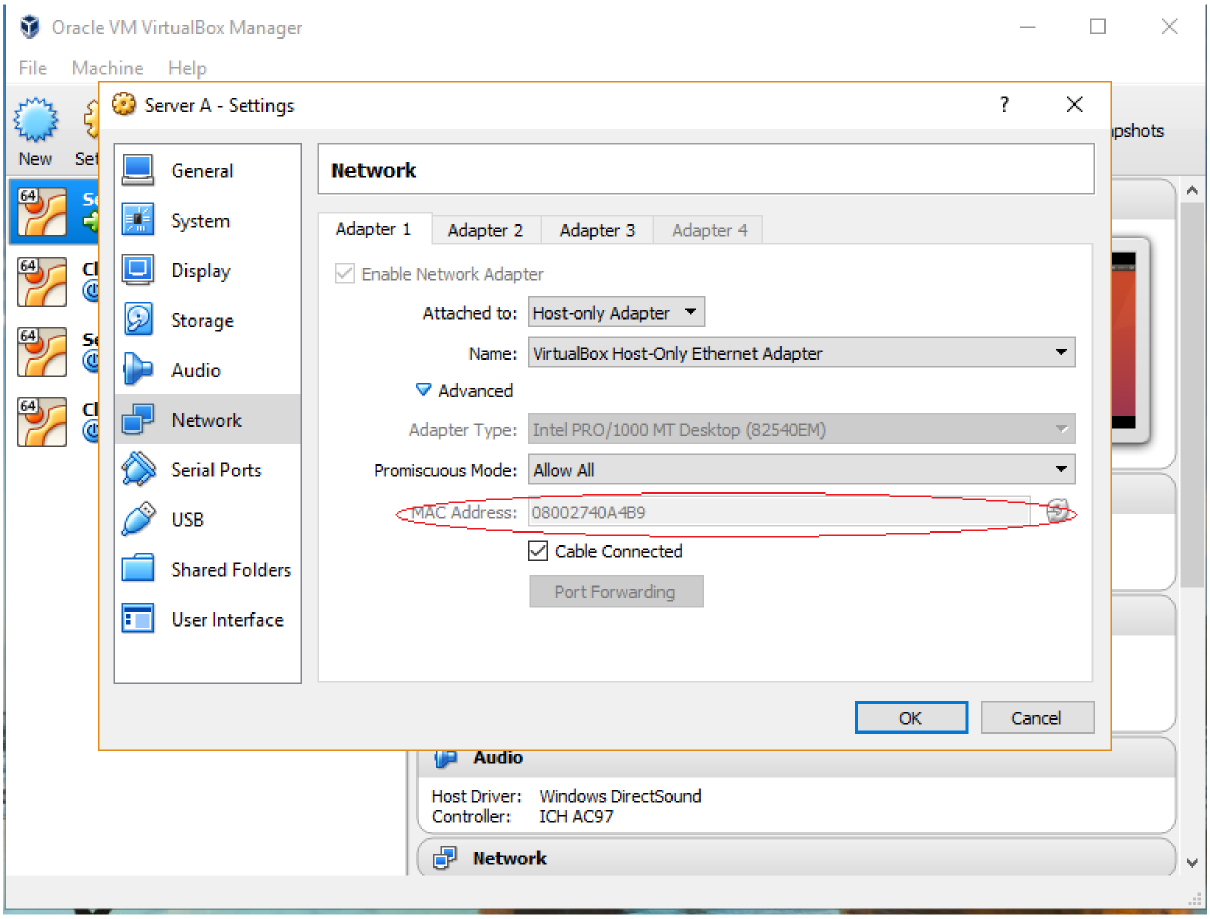
I V S K Chaitanya

Chaitanyaivvala1996@gmail.com

P No 960102-7775

Task 1:

To identify the MAC address of the configured adapters in the web server VM. Below figures shows the mac address.



☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

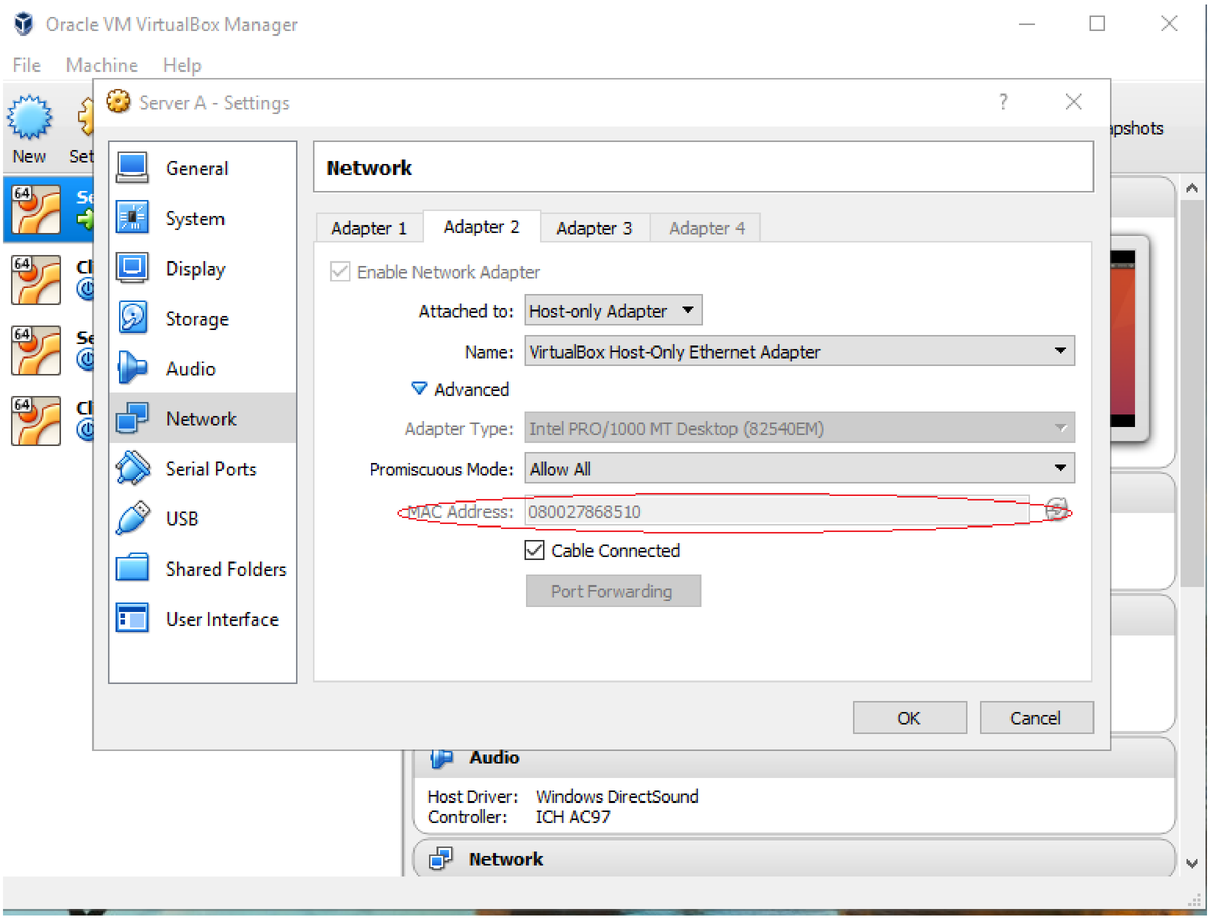
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 08002740A4B9

☑ Cable Connected

Host Driver: Windows DirectSound
Controller: ICH AC97



Network

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Advanced

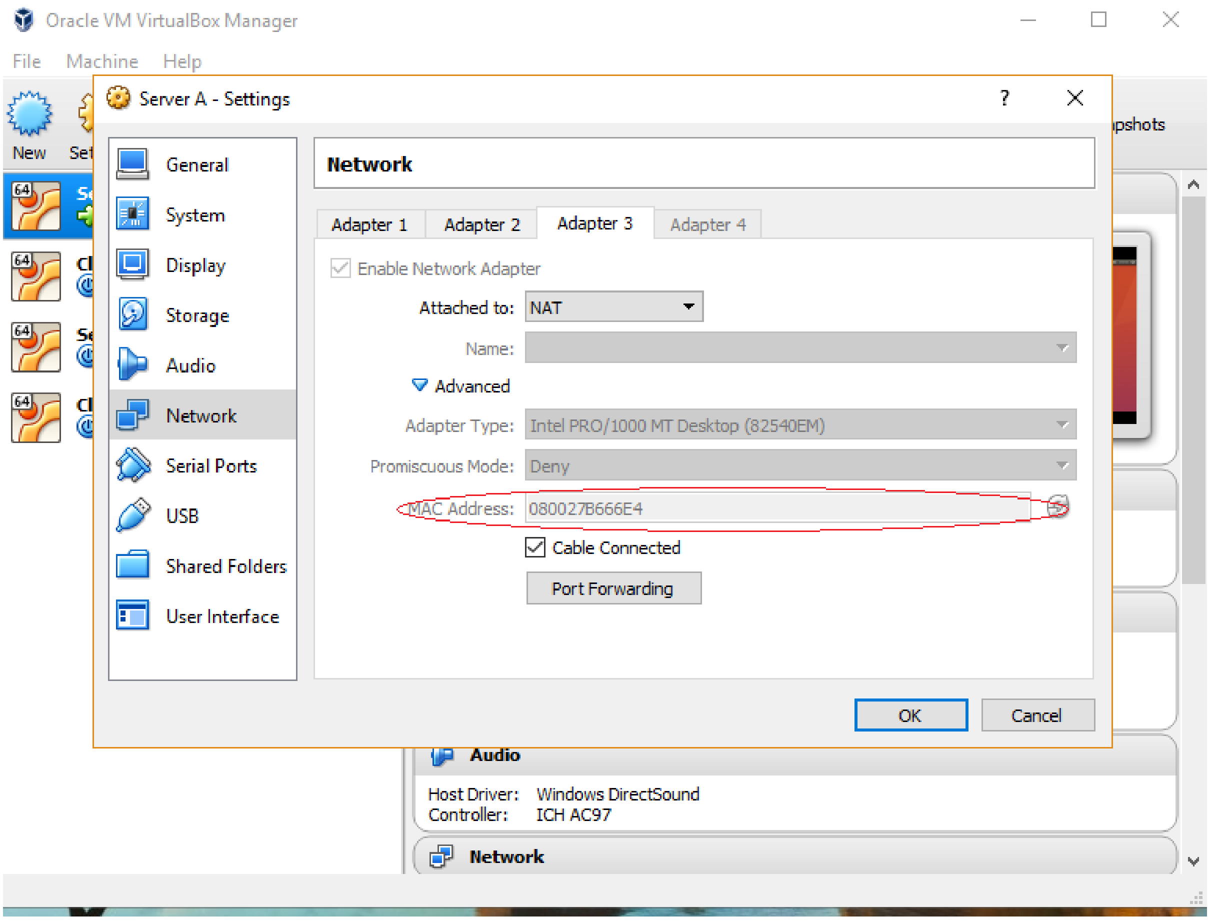
Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 080027868510

☑ Cable Connected

Host Driver: Windows DirectSound
Controller: ICH AC97



Network

☑ Enable Network Adapter

Attached to: NAT

Name:

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 080027B666E4

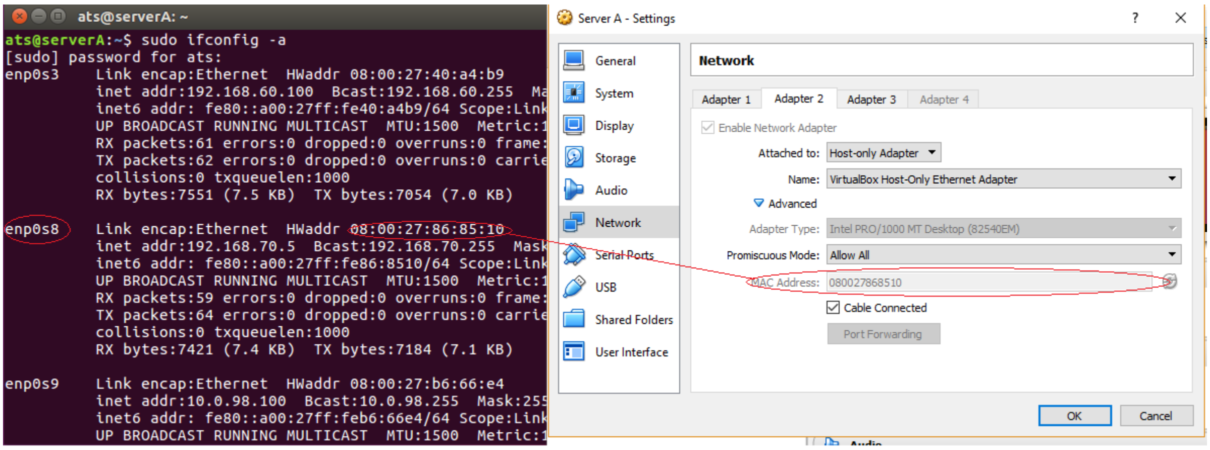
☑ Cable Connected

Host Driver: Windows DirectSound
Controller: ICH AC97

Task 2:

In this task, we need to identify what is the NAT interface and what are the host-only interfaces by using the MAC addresses. Enter the following command in the terminal of server A we can get the list of available interfaces.

“Sudo ifconfig –a “



```
ats@serverA: ~
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  Ma
          inet6 addr: fe80::a00:27ff:fe40:a4b9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61 errors:0 dropped:0 overruns:0 frame:
          TX packets:62 errors:0 dropped:0 overruns:0 carrie
          collisions:0 txqueuelen:1000
          RX bytes:7551 (7.5 KB)  TX bytes:7054 (7.0 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:86:85:10
          inet addr:192.168.70.5  Bcast:192.168.70.255  Mask
          inet6 addr: fe80::a00:27ff:fe86:8510/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:59 errors:0 dropped:0 overruns:0 frame:
          TX packets:64 errors:0 dropped:0 overruns:0 carrie
          collisions:0 txqueuelen:1000
          RX bytes:7421 (7.4 KB)  TX bytes:7184 (7.1 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
          inet6 addr: fe80::a00:27ff:feb6:66e4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

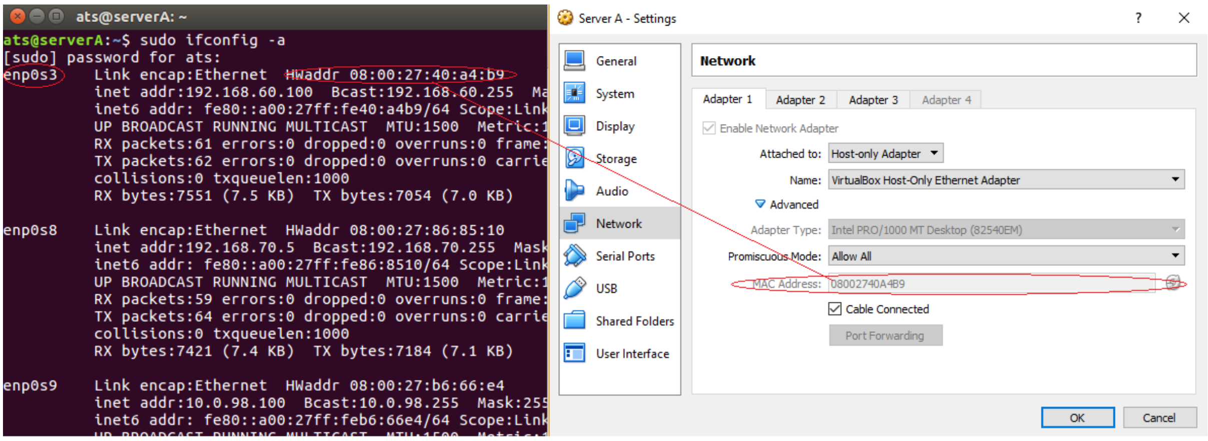
Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Allow All

MAC Address: 080027868510

☑ Cable Connected



```
ats@serverA: ~
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  Ma
          inet6 addr: fe80::a00:27ff:fe40:a4b9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61 errors:0 dropped:0 overruns:0 frame:
          TX packets:62 errors:0 dropped:0 overruns:0 carrie
          collisions:0 txqueuelen:1000
          RX bytes:7551 (7.5 KB)  TX bytes:7054 (7.0 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:86:85:10
          inet addr:192.168.70.5  Bcast:192.168.70.255  Mask
          inet6 addr: fe80::a00:27ff:fe86:8510/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:59 errors:0 dropped:0 overruns:0 frame:
          TX packets:64 errors:0 dropped:0 overruns:0 carrie
          collisions:0 txqueuelen:1000
          RX bytes:7421 (7.4 KB)  TX bytes:7184 (7.1 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
          inet6 addr: fe80::a00:27ff:feb6:66e4/64 Scope:Link
```

Network

Adapter 1 | Adapter 2 | Adapter 3 | Adapter 4

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

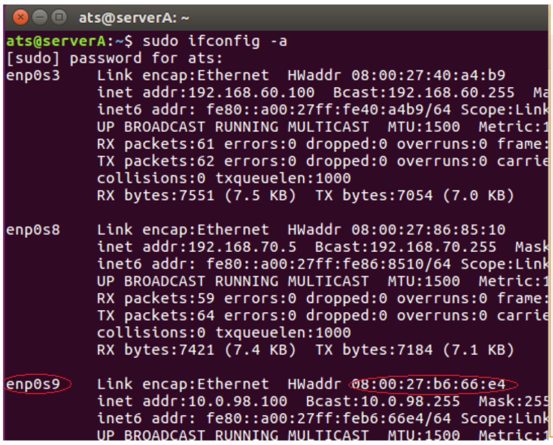
Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

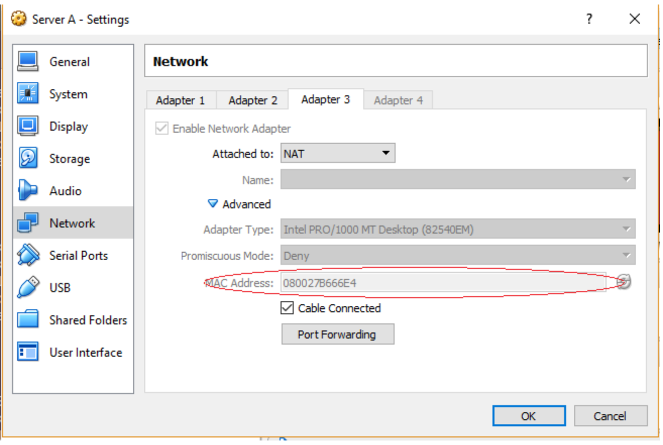
Promiscuous Mode: Allow All

MAC Address: 08002740A4B9

☑ Cable Connected



```
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  Ma
          inet6 addr: fe80::a00:27ff:fe40:a4b9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61 errors:0 dropped:0 overruns:0 frame:
          TX packets:62 errors:0 dropped:0 overruns:0 carrie
          collisions:0 txqueuelen:1000
          RX bytes:7551 (7.5 KB)  TX bytes:7054 (7.0 KB)

enp0s8    Link encap:Ethernet  HWaddr 08:00:27:86:85:10
          inet addr:192.168.70.5  Bcast:192.168.70.255  Mask
          inet6 addr: fe80::a00:27ff:fe86:8510/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:59 errors:0 dropped:0 overruns:0 frame:
          TX packets:64 errors:0 dropped:0 overruns:0 carrie
          collisions:0 txqueuelen:1000
          RX bytes:7421 (7.4 KB)  TX bytes:7184 (7.1 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
          inet6 addr: fe80::a00:27ff:feb6:66e4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```



Server A - Settings

Network

Adapter 3

☑ Enable Network Adapter

Attached to: NAT

Name:

Advanced

Adapter Type: Intel PRO/1000 MT Desktop (82540EM)

Promiscuous Mode: Deny

MAC Address: 080027B666E4

☑ Cable Connected

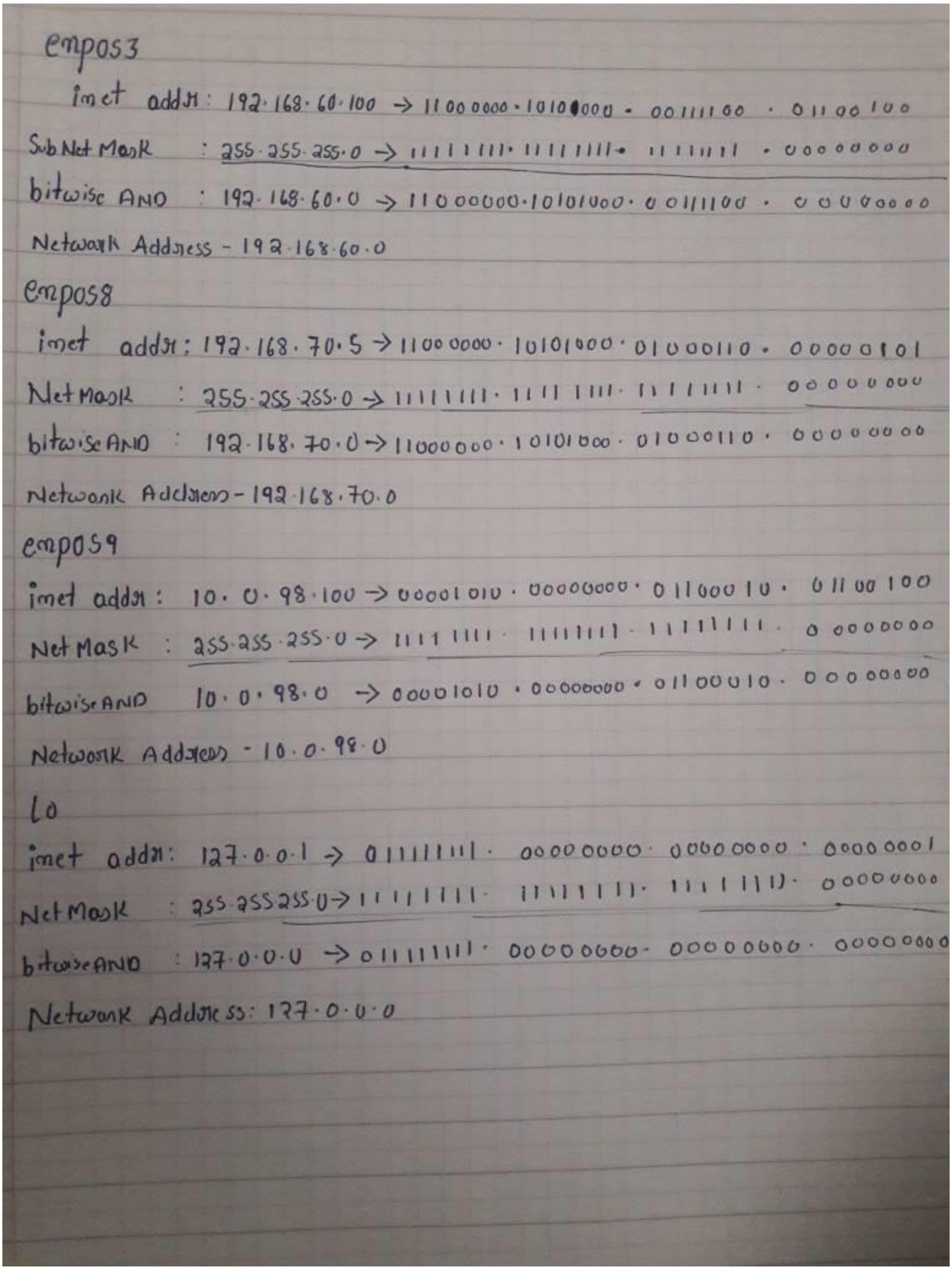
From the above figure we can conlud that

“enp0s9” is the NAT interface

“enp0s3” and “enp0s8” are the host-only interfaces.

Task 3:

To find the network address of each interface associated with their ip address and their netmask. Below figure shows the network address corresponding ip address.



emp0s3

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

emp0s8

inet addr : 192·168·70·5 → 11000000· 10101000· 01000110 · 00000101

NetMask : 255·255·255·0 → 11111111· 11111111· 11111111 · 00000000

bitwiseAND : 192·168·70·0 → 11000000· 10101000· 01000110 · 00000000

Network Address - 192·168·70·0

emp0s9

inet addr : 10· 0· 98·100 → 00001010 · 00000000· 01100010 · 01100100

Net Mask : 255·255·255·0 → 11111111 · 11111111 · 11111111 · 00000000

bitwiseAND 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

NetMask : 255·255·255·0 → 11111111 · 11111111 · 11111111 · 00000000

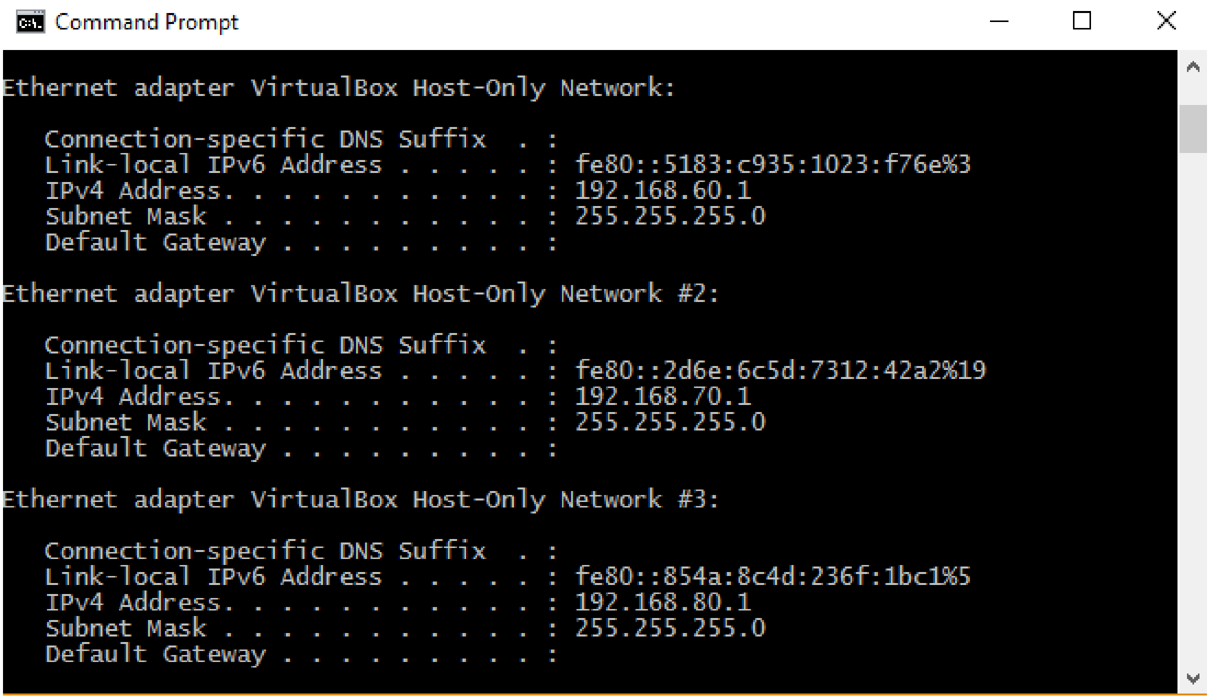
bitwiseAND : 127·0·0·0 → 01111111 · 00000000 · 00000000 · 00000000

Network Address: 127·0·0·0

Task 4:

To identify 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 will identify the host-only interfaces (Ethernet adapter virtual box host only network, Ethernet adapter virtual box host only network #2, Ethernet adapter virtual box host only network #3)

Ipconfig /all



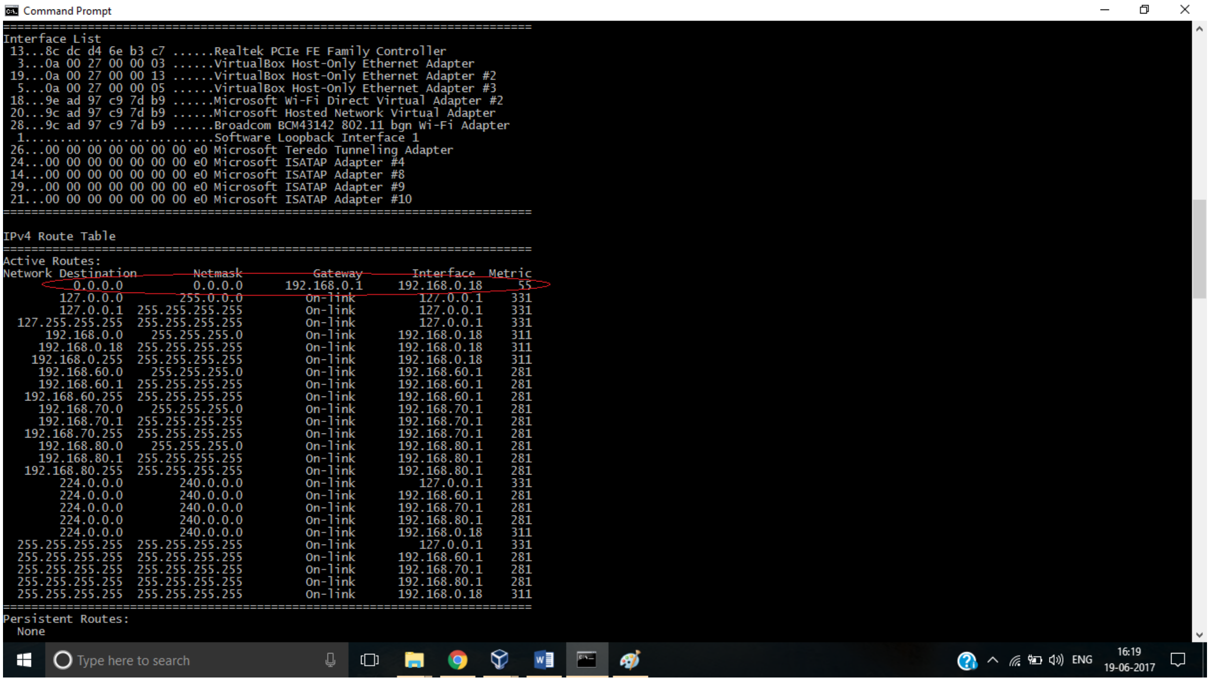
Command Prompt

```
Ethernet adapter VirtualBox Host-Only Network:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5183:c935:1023:f76e%3
   IPv4 Address. . . . . . . . . . . : 192.168.60.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VirtualBox Host-Only Network #2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::2d6e:6c5d:7312:42a2%19
   IPv4 Address. . . . . . . . . . . : 192.168.70.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Ethernet adapter VirtualBox Host-Only Network #3:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::854a:8c4d:236f:1bc1%5
   IPv4 Address. . . . . . . . . . . : 192.168.80.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :
```

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.

“Route -4 PRINT”



Command Prompt

```
===========================================================================
Interface List
 13...8c dc d4 6e b3 c7 ......Realtek PCIe FE Family Controller
  3...0a 00 27 00 00 03 ......VirtualBox Host-Only Ethernet Adapter
 19...0a 00 27 00 00 13 ......VirtualBox Host-Only Ethernet Adapter #2
  5...0a 00 27 00 00 05 ......VirtualBox Host-Only Ethernet Adapter #3
 18...9e ad 97 c9 7d b9 ......Microsoft Wi-Fi Direct Virtual Adapter #2
 20...9c ad 97 c9 7d b9 ......Microsoft Hosted Network Virtual Adapter
 28...9c ad 97 c9 7d b9 ......Broadcom BCM43142 802.11 bgn Wi-Fi Adapter
  1...........................Software Loopback Interface 1
 26...00 00 00 00 00 00 00 e0 Microsoft Teredo Tunneling Adapter
 24...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #4
 14...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #8
 29...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #9
 21...00 00 00 00 00 00 00 e0 Microsoft ISATAP Adapter #10
===========================================================================

IPv4 Route Table
===========================================================================
Active Routes:
Network Destination        Netmask          Gateway       Interface  Metric
          0.0.0.0          0.0.0.0      192.168.0.1    192.168.0.18     55
        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
      192.168.0.0    255.255.255.0         On-link     192.168.0.18    311
     192.168.0.18  255.255.255.255         On-link     192.168.0.18    311
    192.168.0.255  255.255.255.255         On-link     192.168.0.18    311
     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
        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     192.168.60.1    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
        224.0.0.0        240.0.0.0         On-link     192.168.0.18    311
  255.255.255.255  255.255.255.255         On-link        127.0.0.1    331
  255.255.255.255  255.255.255.255         On-link     192.168.60.1    281
  255.255.255.255  255.255.255.255         On-link     192.168.70.1    281
  255.255.255.255  255.255.255.255         On-link     192.168.80.1    281
  255.255.255.255  255.255.255.255         On-link     192.168.0.18    311
===========================================================================
Persistent Routes:
  None
```

from the above figure, we can clearly identify that default gateway for the HOST OS (Windows) is 192.168.0.1 and interface is 192.168.0.18

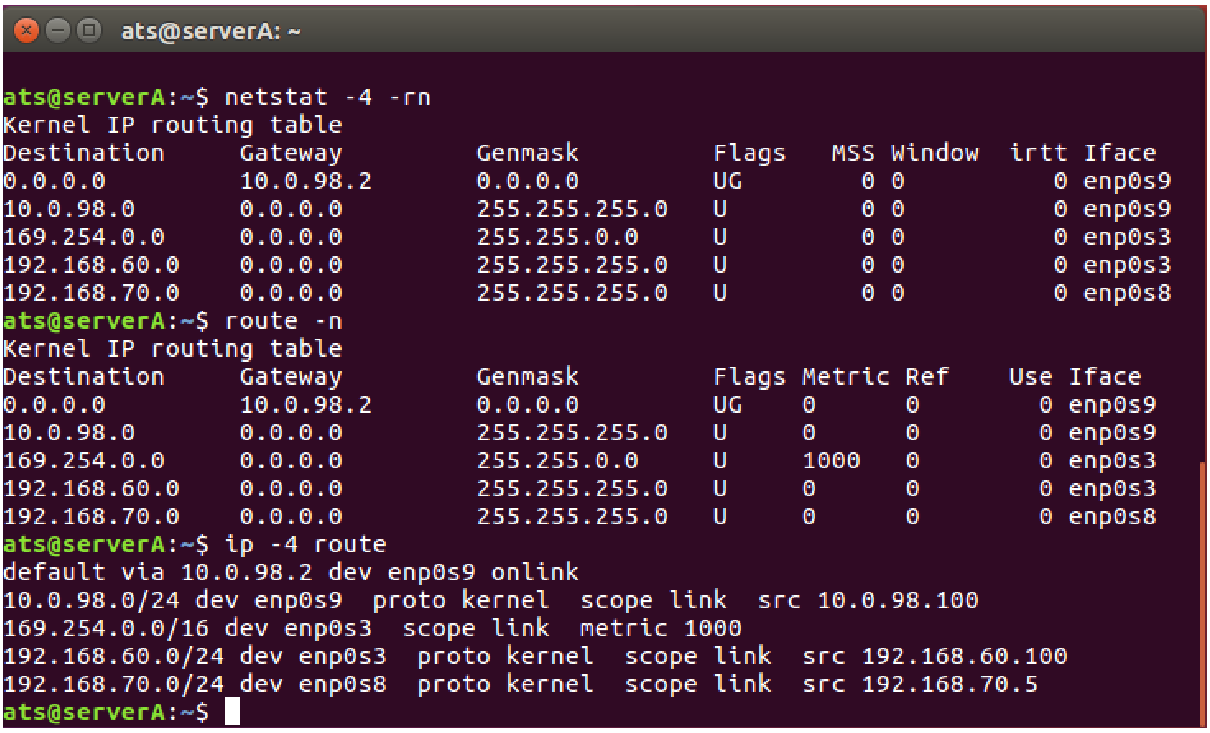
Task 6:

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.

“Netstat -4 -rn”

“Route -n”

“Ip -4 route”



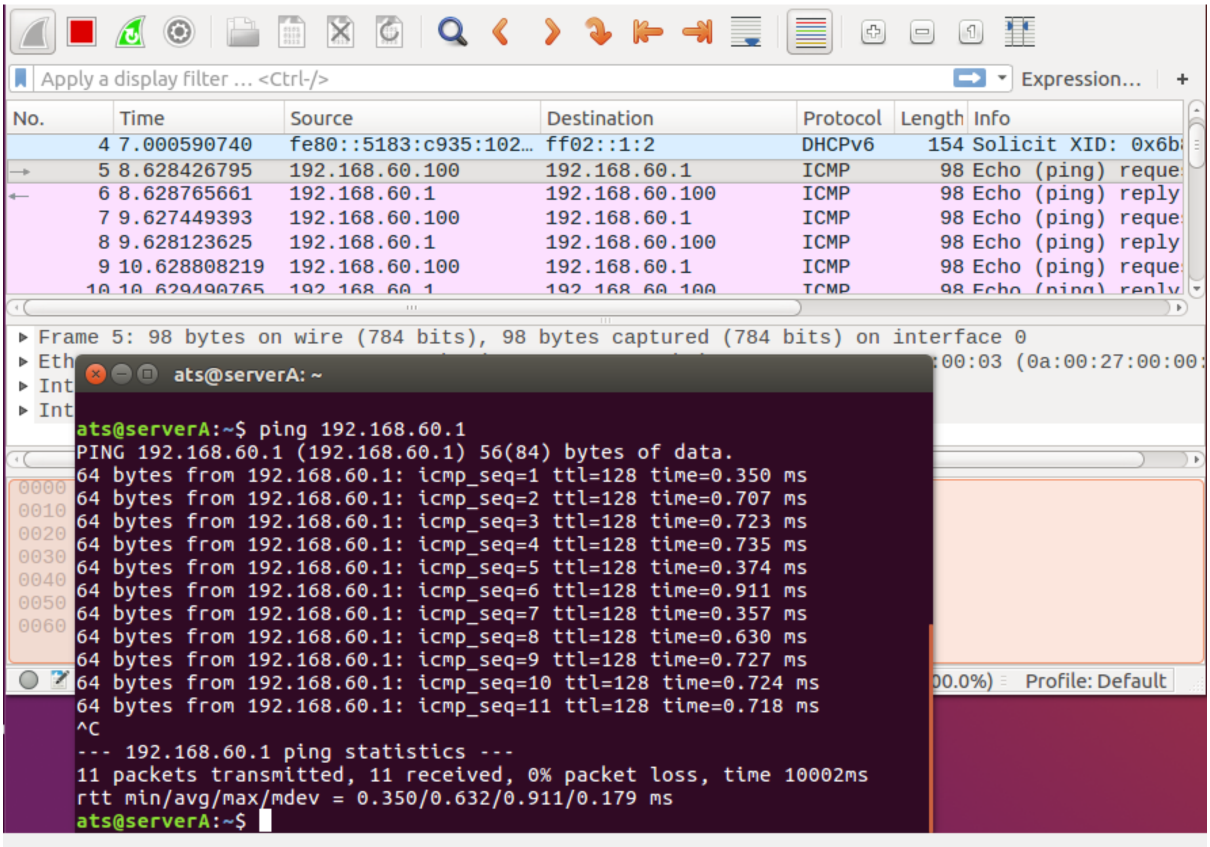
ats@serverA: ~

```
ats@serverA:~$ netstat -4 -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0         10.0.98.2       0.0.0.0         UG        0 0          0 enp0s9
10.0.98.0       0.0.0.0         255.255.255.0   U         0 0          0 enp0s9
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
192.168.70.0    0.0.0.0         255.255.255.0   U         0 0          0 enp0s8
ats@serverA:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.98.2       0.0.0.0         UG    0      0        0 enp0s9
10.0.98.0       0.0.0.0         255.255.255.0   U     0      0        0 enp0s9
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 enp0s3
192.168.60.0    0.0.0.0         255.255.255.0   U     0      0        0 enp0s3
192.168.70.0    0.0.0.0         255.255.255.0   U     0      0        0 enp0s8
ats@serverA:~$ ip -4 route
default via 10.0.98.2 dev enp0s9 onlink
10.0.98.0/24 dev enp0s9  proto kernel  scope link  src 10.0.98.100
169.254.0.0/16 dev enp0s3  scope link  metric 1000
192.168.60.0/24 dev enp0s3  proto kernel  scope link  src 192.168.60.100
192.168.70.0/24 dev enp0s8  proto kernel  scope link  src 192.168.70.5
ats@serverA:~$
```

We can conclude form the above figure that through host-only interface the guest OS can reach the default gateway for host OS.

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.



No.	Time	Source	Destination	Protocol	Length	Info
4	7.000590740	fe80::5183:c935:102…	ff02::1:2	DHCPv6	154	Solicit XID: 0x6b
5	8.628426795	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) reque
6	8.628765661	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply
7	9.627449393	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) reque
8	9.628123625	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply
9	10.628808219	192.168.60.100	192.168.60.1	ICMP	98	Echo (ping) reque
10	10.629490765	192.168.60.1	192.168.60.100	ICMP	98	Echo (ping) reply

```
▸ Frame 5: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
▸ Eth                                                      :00:03 (0a:00:27:00:00:
▸ Int
▸ Int
```

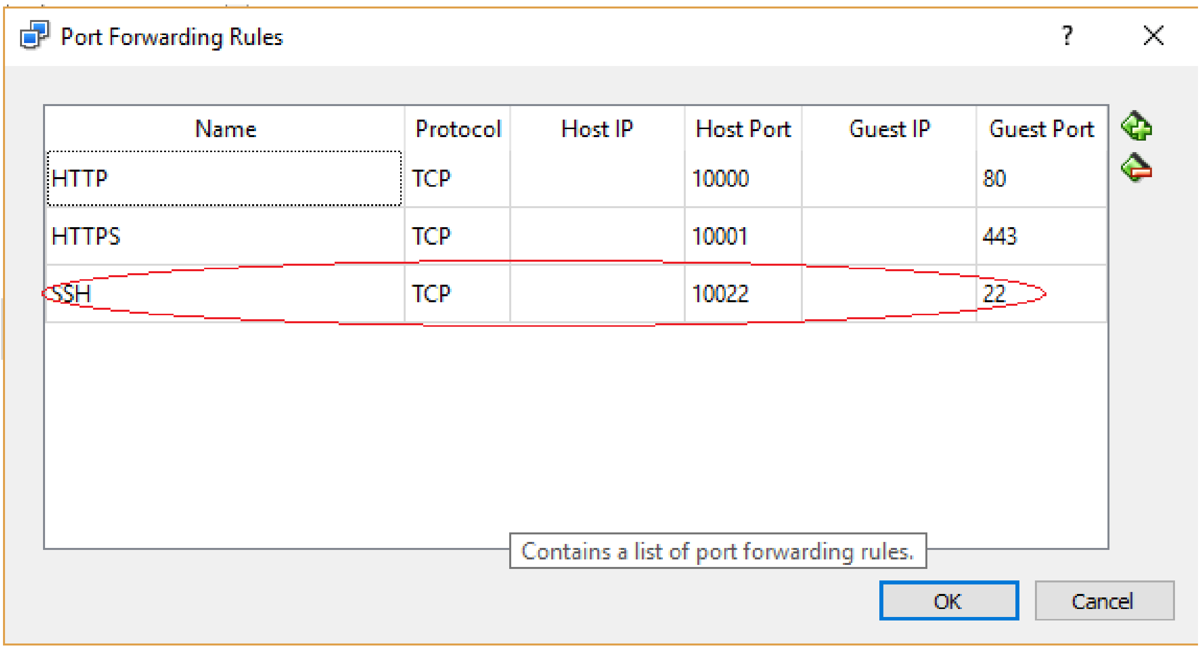
```
0000
0010
0020
0030
0040
0050
0060
```

```
ats@serverA:~$ ping 192.168.60.1
PING 192.168.60.1 (192.168.60.1) 56(84) bytes of data.
64 bytes from 192.168.60.1: icmp_seq=1 ttl=128 time=0.350 ms
64 bytes from 192.168.60.1: icmp_seq=2 ttl=128 time=0.707 ms
64 bytes from 192.168.60.1: icmp_seq=3 ttl=128 time=0.723 ms
64 bytes from 192.168.60.1: icmp_seq=4 ttl=128 time=0.735 ms
64 bytes from 192.168.60.1: icmp_seq=5 ttl=128 time=0.374 ms
64 bytes from 192.168.60.1: icmp_seq=6 ttl=128 time=0.911 ms
64 bytes from 192.168.60.1: icmp_seq=7 ttl=128 time=0.357 ms
64 bytes from 192.168.60.1: icmp_seq=8 ttl=128 time=0.630 ms
64 bytes from 192.168.60.1: icmp_seq=9 ttl=128 time=0.727 ms
64 bytes from 192.168.60.1: icmp_seq=10 ttl=128 time=0.724 ms
64 bytes from 192.168.60.1: icmp_seq=11 ttl=128 time=0.718 ms
^C
--- 192.168.60.1 ping statistics ---
11 packets transmitted, 11 received, 0% packet loss, time 10002ms
rtt min/avg/max/mdev = 0.350/0.632/0.911/0.179 ms
ats@serverA:~$
```

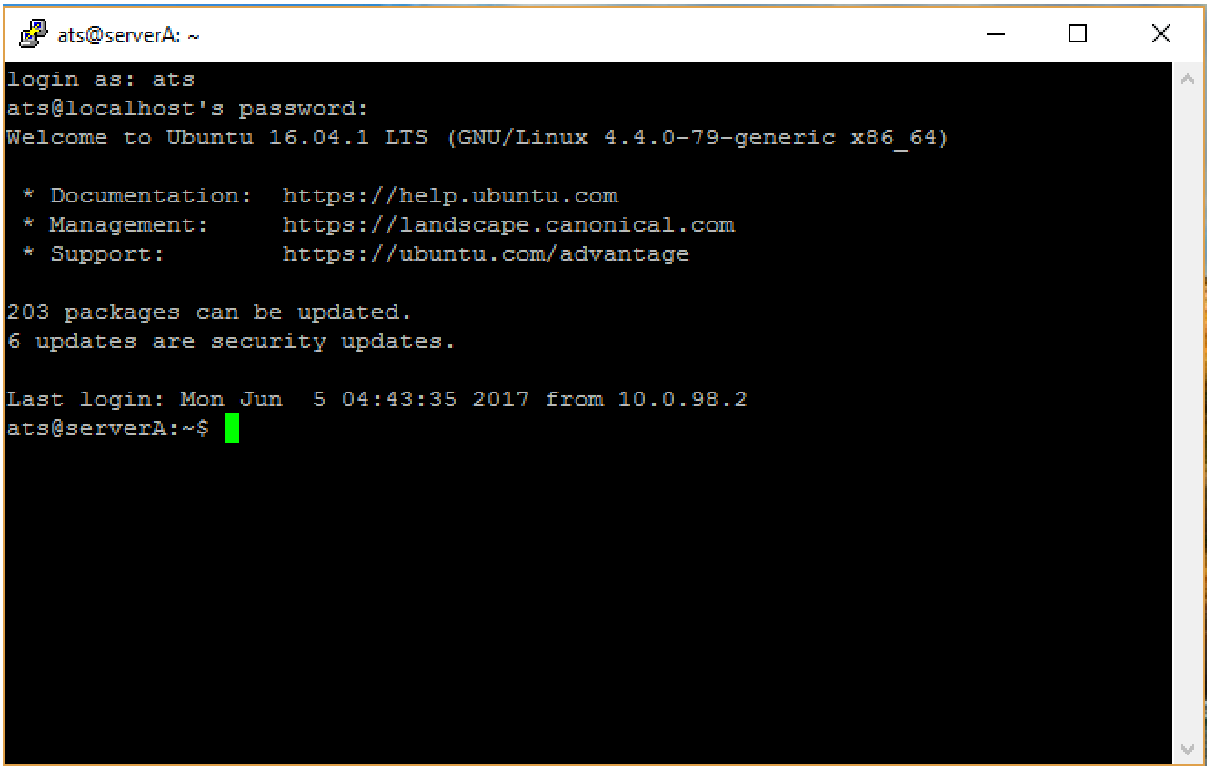
From the above figure, we can conclude that the ping and the Wireshark capturing of the icmp traffic are identical.

Task 8:

To ssh into VM via localhost from the HOST OS. It can be done by starting Putty and enter localhost in Host name (or IP address) field and 10022 in the Port field. Then click the Open button.



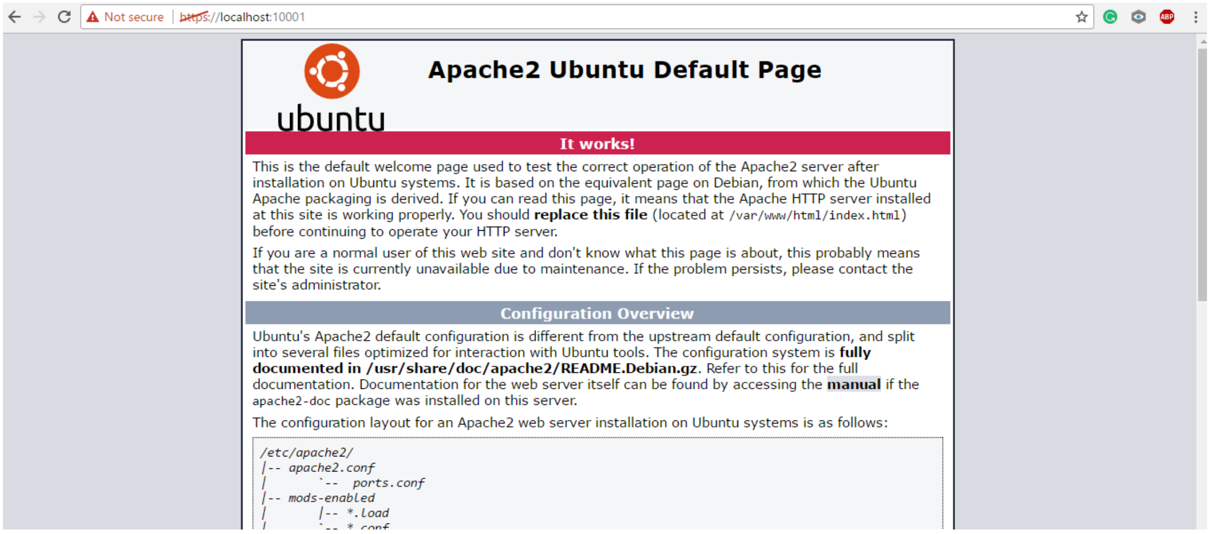
Name	Protocol	Host IP	Host Port	Guest IP	Guest Port
HTTP	TCP		10000		80
HTTPS	TCP		10001		443
SSH	TCP		10022		22



```
login as: ats
ats@localhost's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

203 packages can be updated.
6 updates are security updates.

Last login: Mon Jun  5 04:43:35 2017 from 10.0.98.2
ats@serverA:~$
```

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. For doing 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. Below images shows that the host OS can view the HTTP and HTTPS content of the apache2 server in the host OS.



Not secure | ~~https~~://localhost:10001

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.

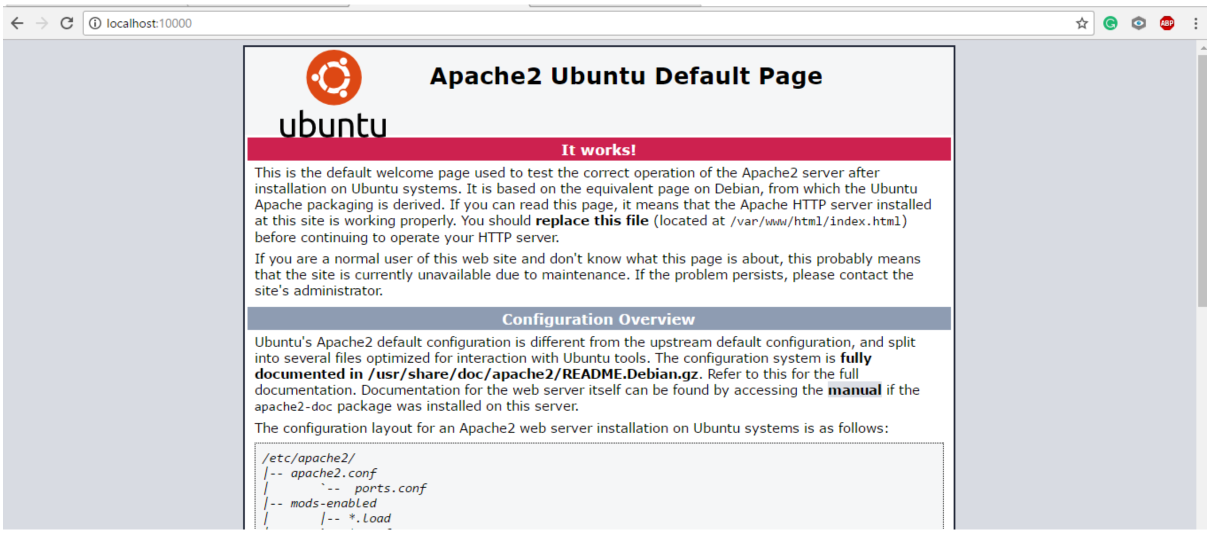
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

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
```



localhost:10000

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

Task 10:

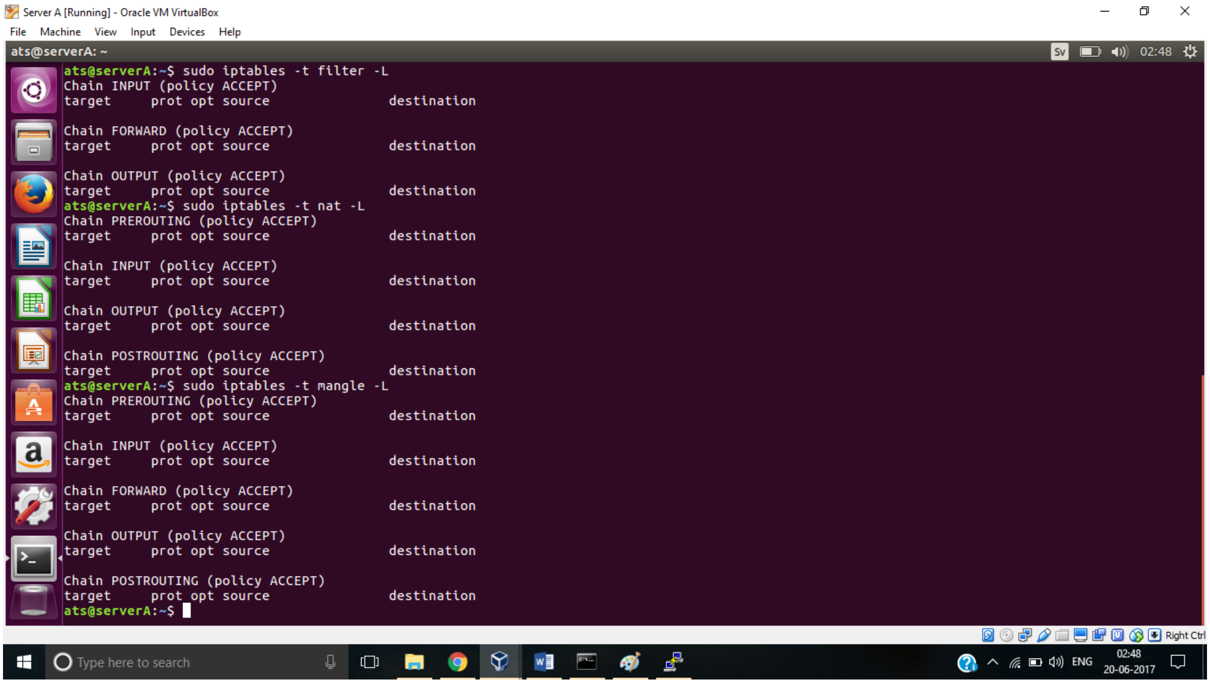
Following commands to view the default rules .

“sudo iptables –t filter -L”

“sudo iptables –t mangle -L”

“sudo iptables –t nat -L”

Below shown are the default policy and rules of the tables (filter, mangle, nat)

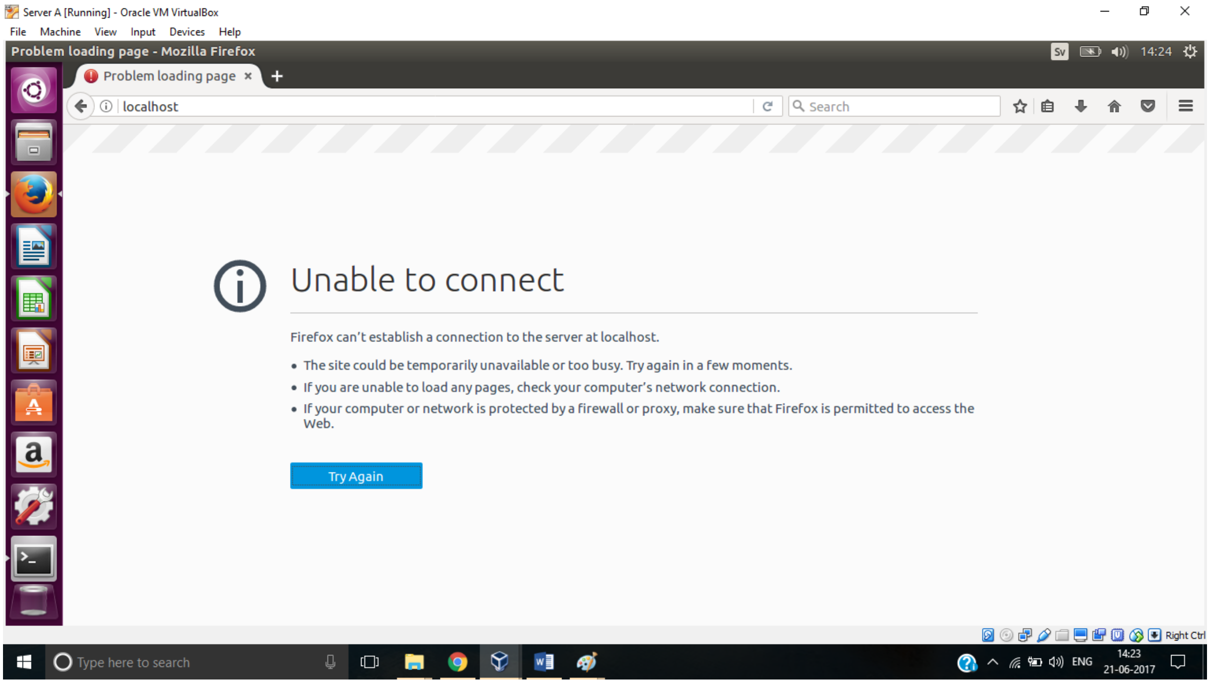

Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
ats@serverA: ~
02:48
Right Ctrl
Type here to search
ENG
02:48
20-06-2017

```
ats@serverA:~$ sudo iptables -t filter -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ats@serverA:~$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
ats@serverA:~$ sudo iptables -t mangle -L
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
ats@serverA:~$ 
```

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 "


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Problem loading page - Mozilla Firefox
14:24
Problem loading page
localhost
Search
Try Again
Right Ctrl
Type here to search
ENG
14:23
21-06-2017

Unable to connect

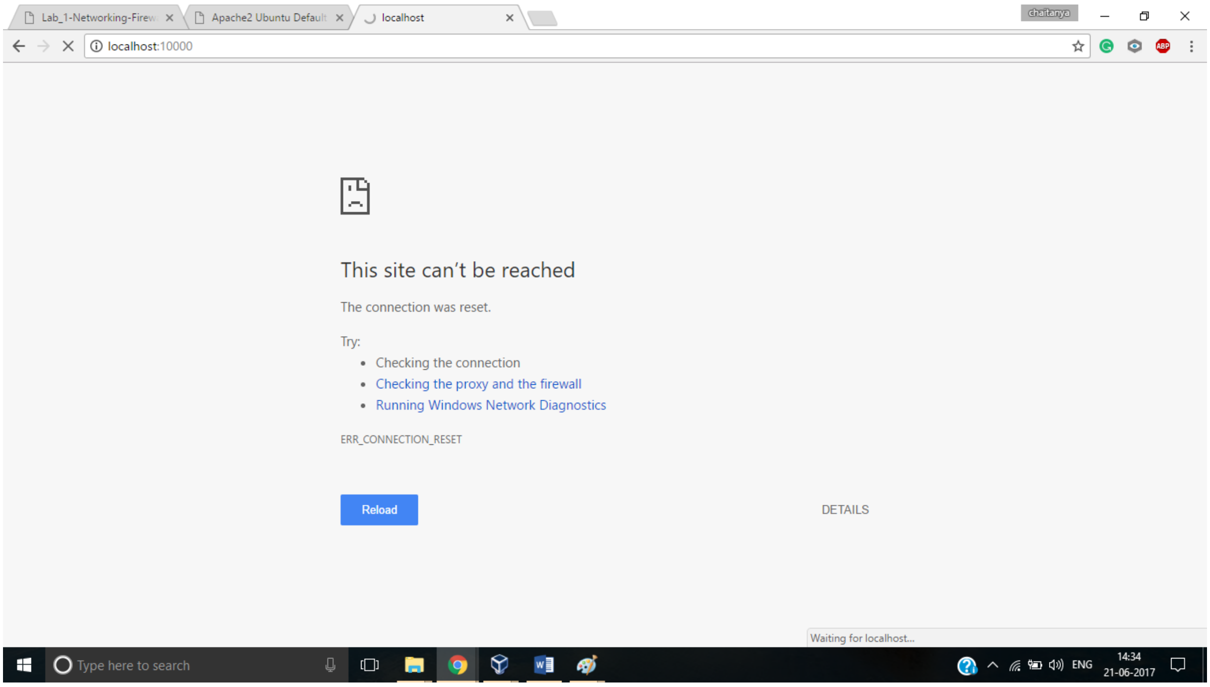
Firefox can't establish a connection to the server at localhost.

- The site could be temporarily unavailable or too busy. Try again in a few moments.
- If you are unable to load any pages, check your computer's network connection.
- If your computer or network is protected by a firewall or proxy, make sure that Firefox is permitted to access the Web.

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”



This site can’t be reached

The connection was reset.

Try:

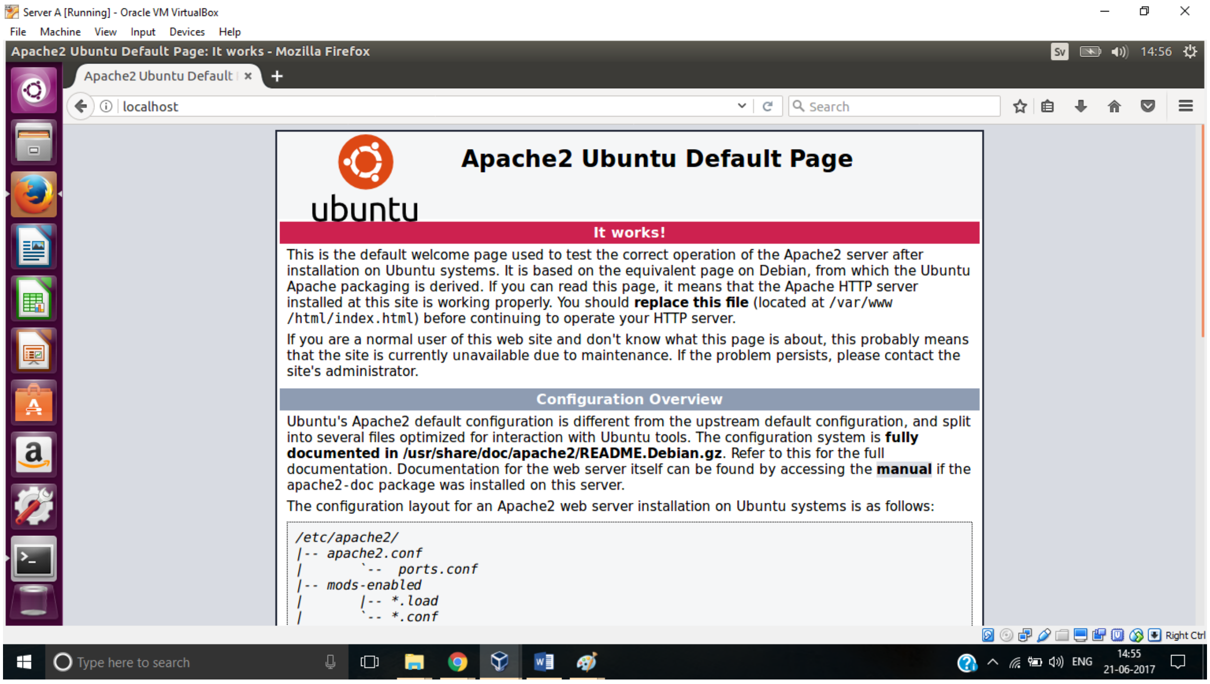
- Checking the connection
- Checking the proxy and the firewall
- Running Windows Network Diagnostics

ERR_CONNECTION_RESET

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 ”


Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Apache2 Ubuntu Default Page: It works - Mozilla Firefox
14:56
Apache2 Ubuntu Default
localhost
Search
Right Ctrl
Type here to search
ENG
14:55
21-06-2017

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

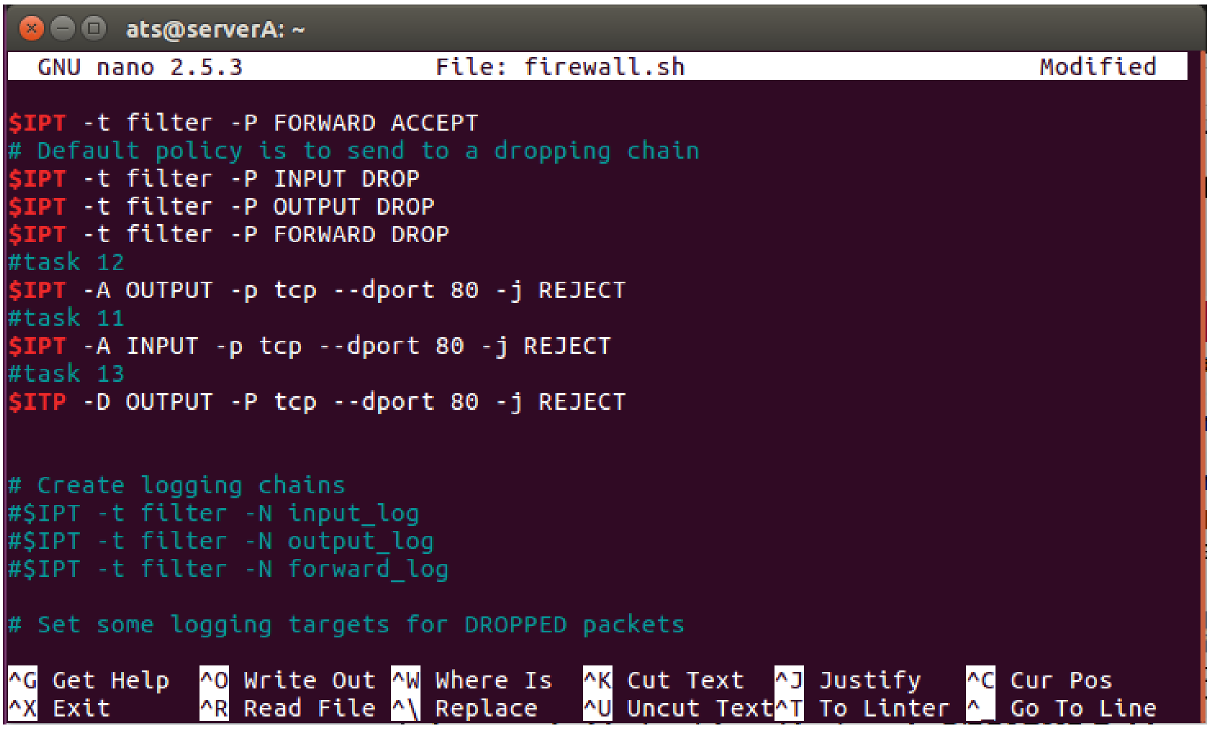
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. 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
```

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.



```
  GNU nano 2.5.3              File: firewall.sh                       Modified

$IPT -t filter -P FORWARD ACCEPT
# Default policy is to send to a dropping chain
$IPT -t filter -P INPUT DROP
$IPT -t filter -P OUTPUT DROP
$IPT -t filter -P FORWARD DROP
#task 12
$IPT -A OUTPUT -p tcp --dport 80 -j REJECT
#task 11
$IPT -A INPUT -p tcp --dport 80 -j REJECT
#task 13
$ITP -D OUTPUT -P tcp --dport 80 -j REJECT


# 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

^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 Linter ^_ Go To Line
```

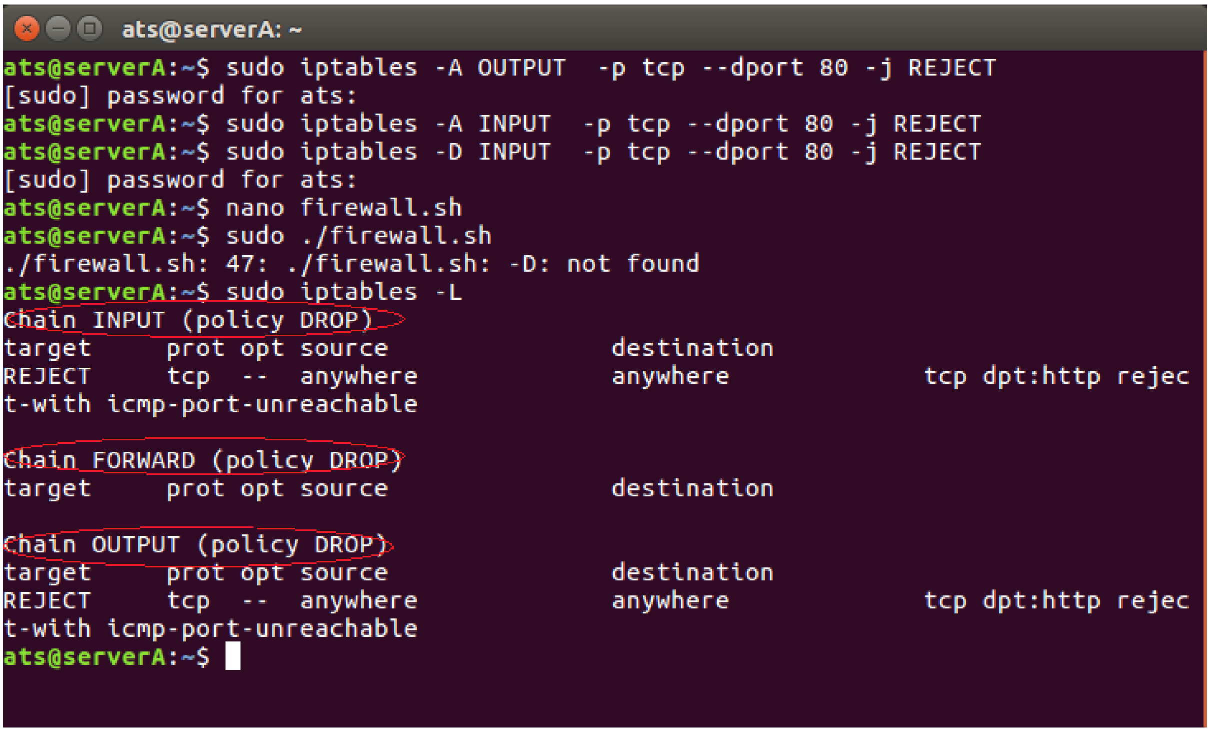
Task 15:

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”

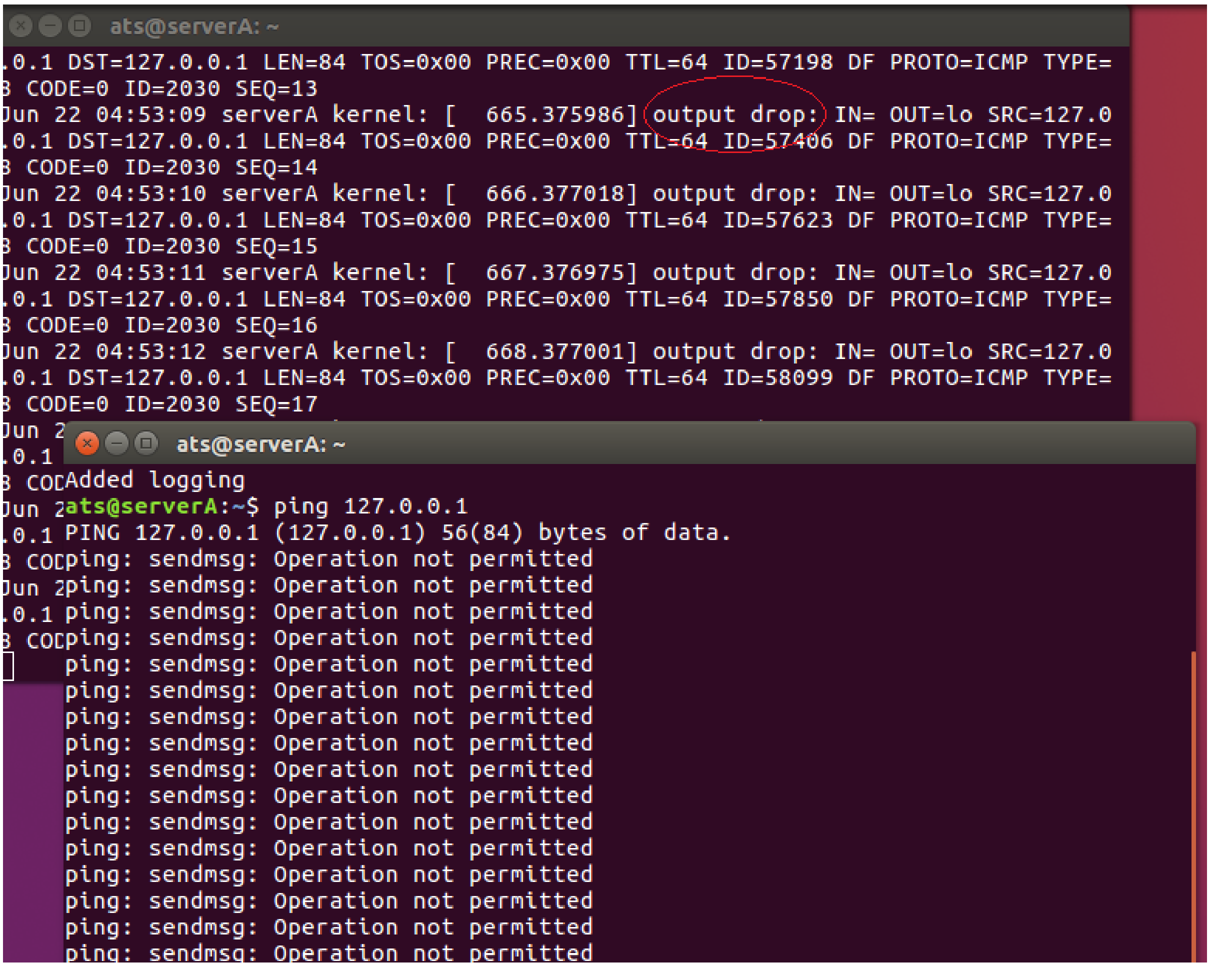


```
ats@serverA:~$ sudo iptables -A OUTPUT  -p tcp --dport 80 -j REJECT
[sudo] password for ats:
ats@serverA:~$ sudo iptables -A INPUT  -p tcp --dport 80 -j REJECT
ats@serverA:~$ sudo iptables -D INPUT  -p tcp --dport 80 -j REJECT
[sudo] password for ats:
ats@serverA:~$ nano firewall.sh
ats@serverA:~$ sudo ./firewall.sh
./firewall.sh: 47: ./firewall.sh: -D: not found
ats@serverA:~$ sudo iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             tcp dpt:http rejec
t-with icmp-port-unreachable

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
REJECT     tcp  --  anywhere             anywhere             tcp dpt:http rejec
t-with icmp-port-unreachable
ats@serverA:~$
```

Task 16:

To see the live logs of linux kernel by entering the following command.

“sudo tail -f /var/log/kern.log”



```
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=57198 DF PROTO=ICMP TYPE=
8 CODE=0 ID=2030 SEQ=13
Jun 22 04:53:09 serverA kernel: [  665.375986] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=57406 DF PROTO=ICMP TYPE=
8 CODE=0 ID=2030 SEQ=14
Jun 22 04:53:10 serverA kernel: [  666.377018] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=57623 DF PROTO=ICMP TYPE=
8 CODE=0 ID=2030 SEQ=15
Jun 22 04:53:11 serverA kernel: [  667.376975] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=57850 DF PROTO=ICMP TYPE=
8 CODE=0 ID=2030 SEQ=16
Jun 22 04:53:12 serverA kernel: [  668.377001] output drop: IN= OUT=lo SRC=127.0
.0.1 DST=127.0.0.1 LEN=84 TOS=0x00 PREC=0x00 TTL=64 ID=58099 DF PROTO=ICMP TYPE=
8 CODE=0 ID=2030 SEQ=17
Jun 2
.0.1
8 COD
Jun 2
.0.1
8 COD
Jun 2
.0.1
8 COD
```

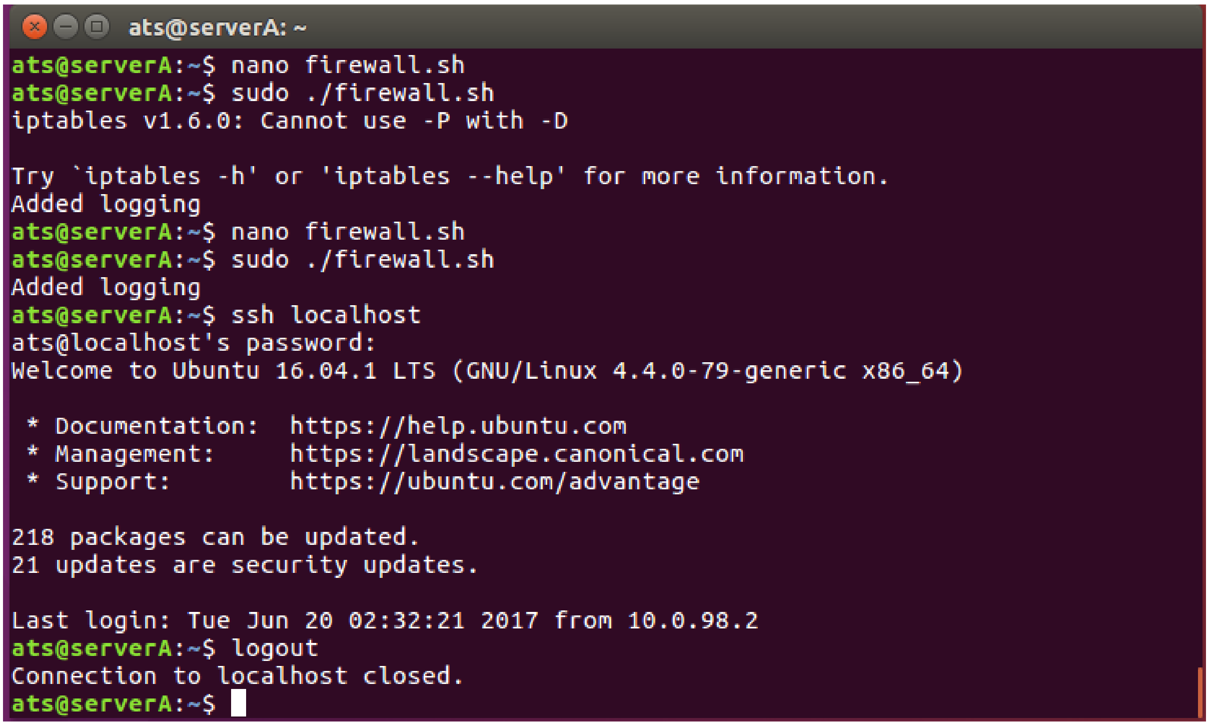
```
Added logging
ats@serverA:~$ ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
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
ping: sendmsg: Operation not permitted
ping: sendmsg: Operation not permitted
```

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”



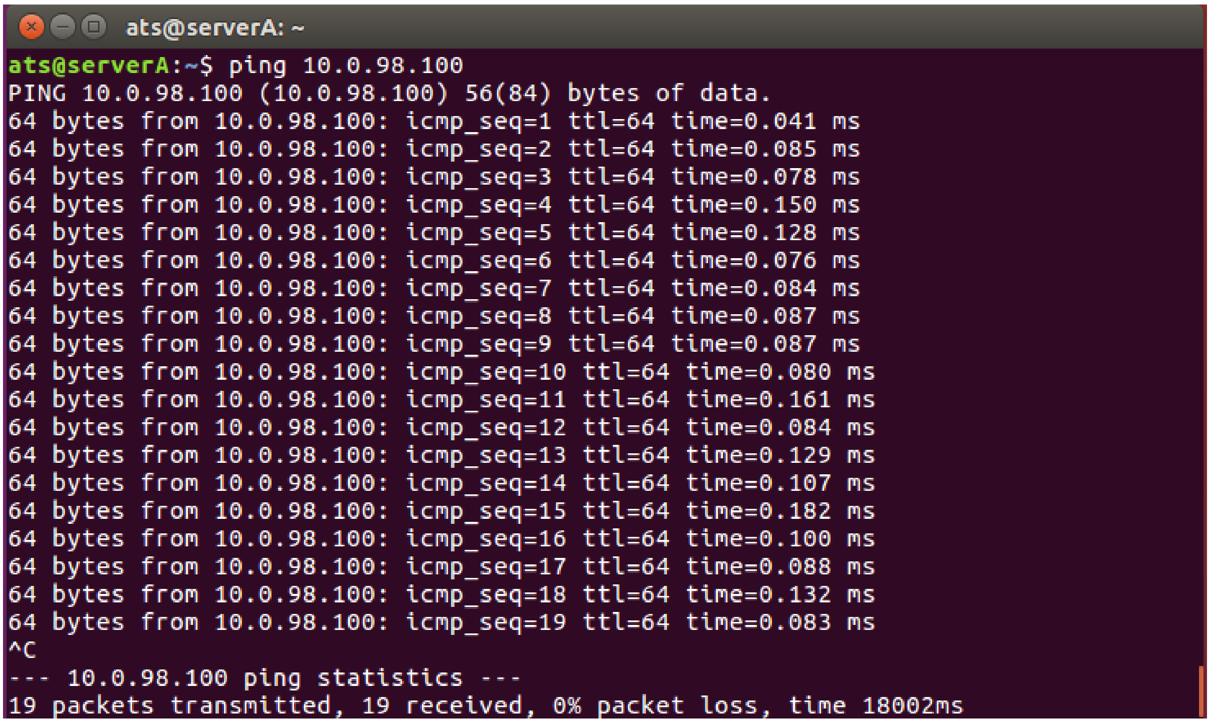
```
ats@serverA:~$ nano firewall.sh
ats@serverA:~$ sudo ./firewall.sh
iptables v1.6.0: Cannot use -P with -D

Try `iptables -h' or 'iptables --help' for more information.
Added logging
ats@serverA:~$ nano firewall.sh
ats@serverA:~$ sudo ./firewall.sh
Added logging
ats@serverA:~$ ssh localhost
ats@localhost's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

218 packages can be updated.
21 updates are security updates.

Last login: Tue Jun 20 02:32:21 2017 from 10.0.98.2
ats@serverA:~$ logout
Connection to localhost closed.
ats@serverA:~$ 
```

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”



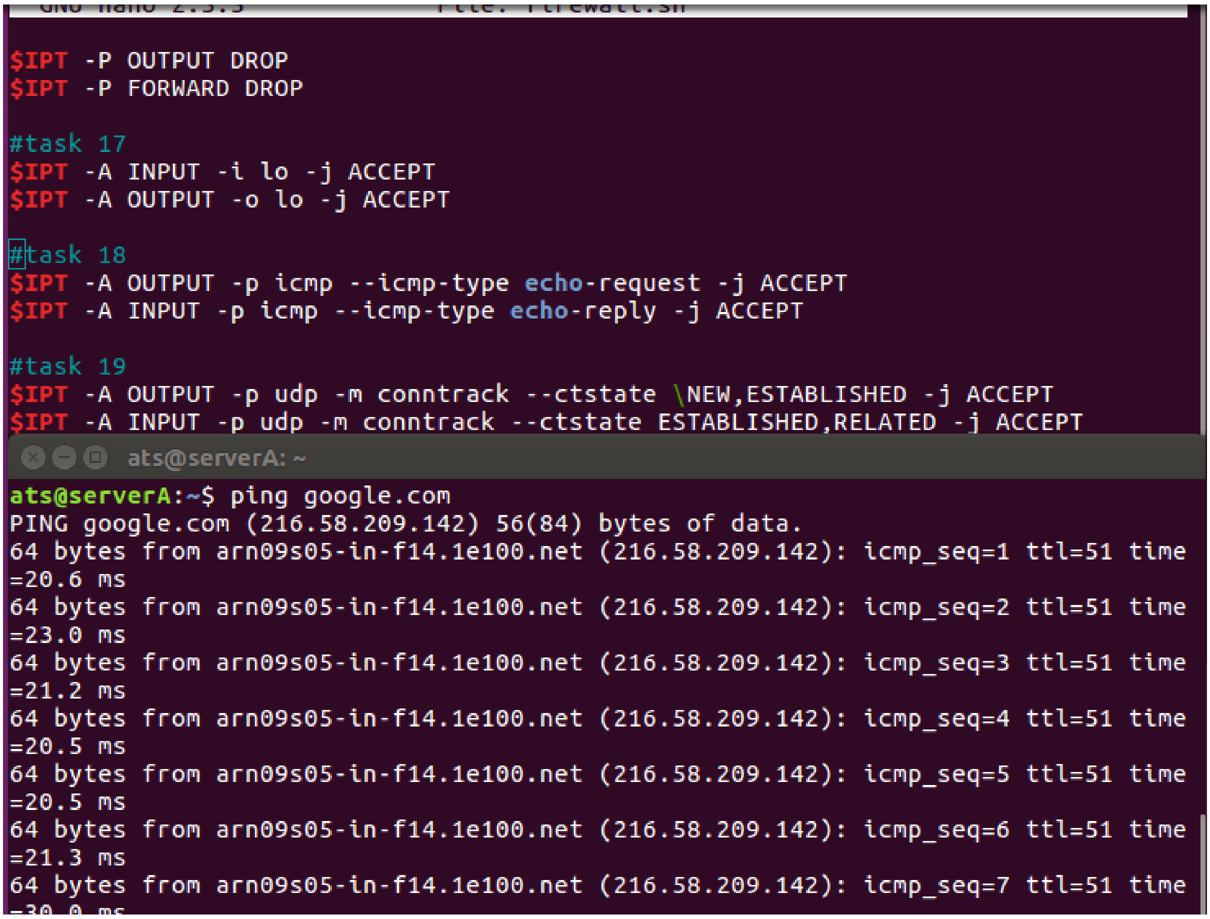
```
ats@serverA:~$ ping 10.0.98.100
PING 10.0.98.100 (10.0.98.100) 56(84) bytes of data.
64 bytes from 10.0.98.100: icmp_seq=1 ttl=64 time=0.041 ms
64 bytes from 10.0.98.100: icmp_seq=2 ttl=64 time=0.085 ms
64 bytes from 10.0.98.100: icmp_seq=3 ttl=64 time=0.078 ms
64 bytes from 10.0.98.100: icmp_seq=4 ttl=64 time=0.150 ms
64 bytes from 10.0.98.100: icmp_seq=5 ttl=64 time=0.128 ms
64 bytes from 10.0.98.100: icmp_seq=6 ttl=64 time=0.076 ms
64 bytes from 10.0.98.100: icmp_seq=7 ttl=64 time=0.084 ms
64 bytes from 10.0.98.100: icmp_seq=8 ttl=64 time=0.087 ms
64 bytes from 10.0.98.100: icmp_seq=9 ttl=64 time=0.087 ms
64 bytes from 10.0.98.100: icmp_seq=10 ttl=64 time=0.080 ms
64 bytes from 10.0.98.100: icmp_seq=11 ttl=64 time=0.161 ms
64 bytes from 10.0.98.100: icmp_seq=12 ttl=64 time=0.084 ms
64 bytes from 10.0.98.100: icmp_seq=13 ttl=64 time=0.129 ms
64 bytes from 10.0.98.100: icmp_seq=14 ttl=64 time=0.107 ms
64 bytes from 10.0.98.100: icmp_seq=15 ttl=64 time=0.182 ms
64 bytes from 10.0.98.100: icmp_seq=16 ttl=64 time=0.100 ms
64 bytes from 10.0.98.100: icmp_seq=17 ttl=64 time=0.088 ms
64 bytes from 10.0.98.100: icmp_seq=18 ttl=64 time=0.132 ms
64 bytes from 10.0.98.100: icmp_seq=19 ttl=64 time=0.083 ms
^C
--- 10.0.98.100 ping statistics ---
19 packets transmitted, 19 received, 0% packet loss, time 18002ms
```

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 ”



```
  GNU nano 2.5.3              File: firewall.sh

$IPT -P OUTPUT DROP
$IPT -P FORWARD DROP

#task 17
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

#task 18
$IPT -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
$IPT -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

#task 19
$IPT -A OUTPUT -p udp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
```

```
ats@serverA:~$ ping google.com
PING google.com (216.58.209.142) 56(84) bytes of data.
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=1 ttl=51 time
=20.6 ms
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=2 ttl=51 time
=23.0 ms
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=3 ttl=51 time
=21.2 ms
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=4 ttl=51 time
=20.5 ms
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=5 ttl=51 time
=20.5 ms
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=6 ttl=51 time
=21.3 ms
64 bytes from arn09s05-in-f14.1e100.net (216.58.209.142): icmp_seq=7 ttl=51 time
=30.0 ms
```

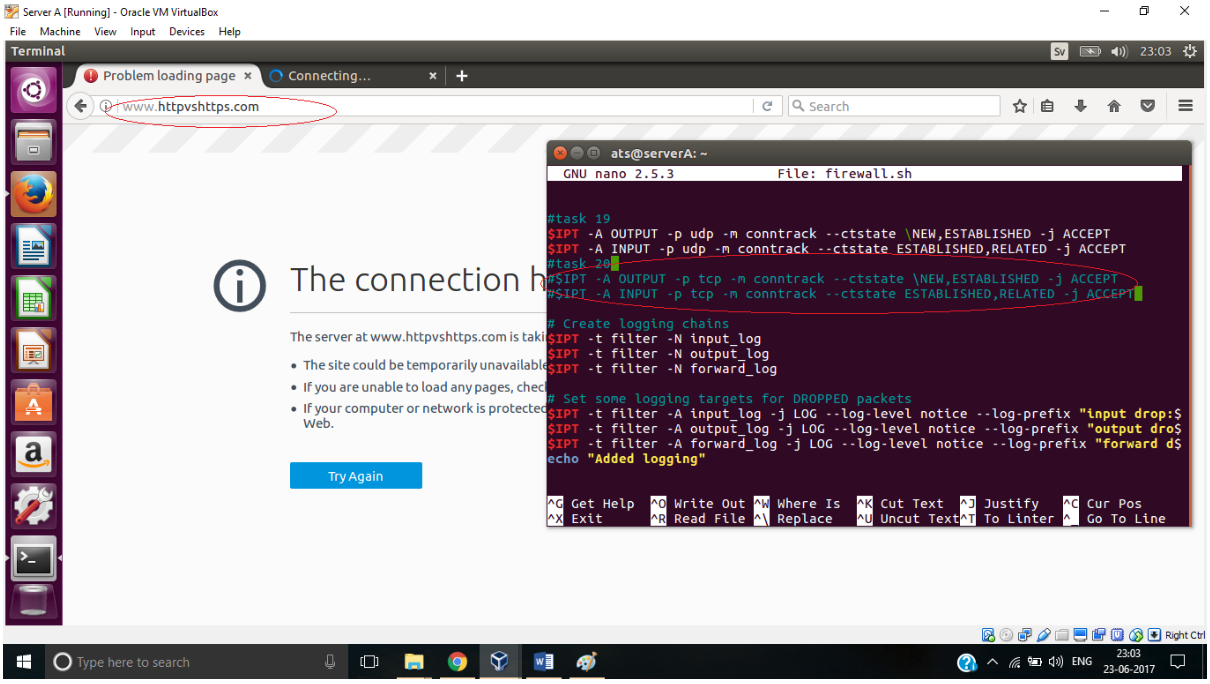
Task 20:

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 ”

Below image shows us that the tcp connection is not establish for the website by commenting the above rules. (http://www.httpvshttps.com/)



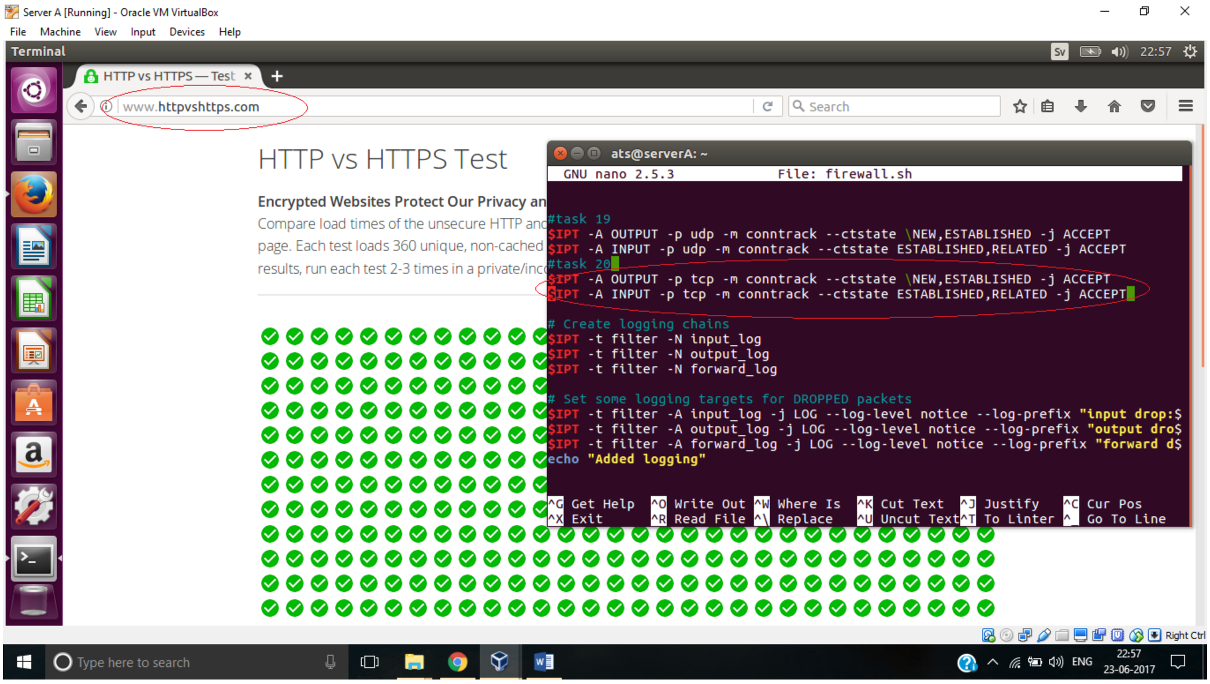
The connection h

The server at www.httpvshttps.com is taki

- The site could be temporarily unavailable
- If you are unable to load any pages, chec
- If your computer or network is protected Web.

```
  GNU nano 2.5.3              File: firewall.sh


#task 19
$IPT -A OUTPUT -p udp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p udp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT
#task 20
#$IPT -A OUTPUT -p tcp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT
#$IPT -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -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
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop:$
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dro$
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward d$
echo "Added logging"


^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 Linter ^_ Go To Line
```

Below image shows us that the tcp connection is establish for the website by adding the above rules. (http://www.httpvshttps.com/)



HTTP vs HTTPS Test

Encrypted Websites Protect Our Privacy an

Compare load times of the unsecure HTTP and page. Each test loads 360 unique, non-cached results, run each test 2-3 times in a private/inc

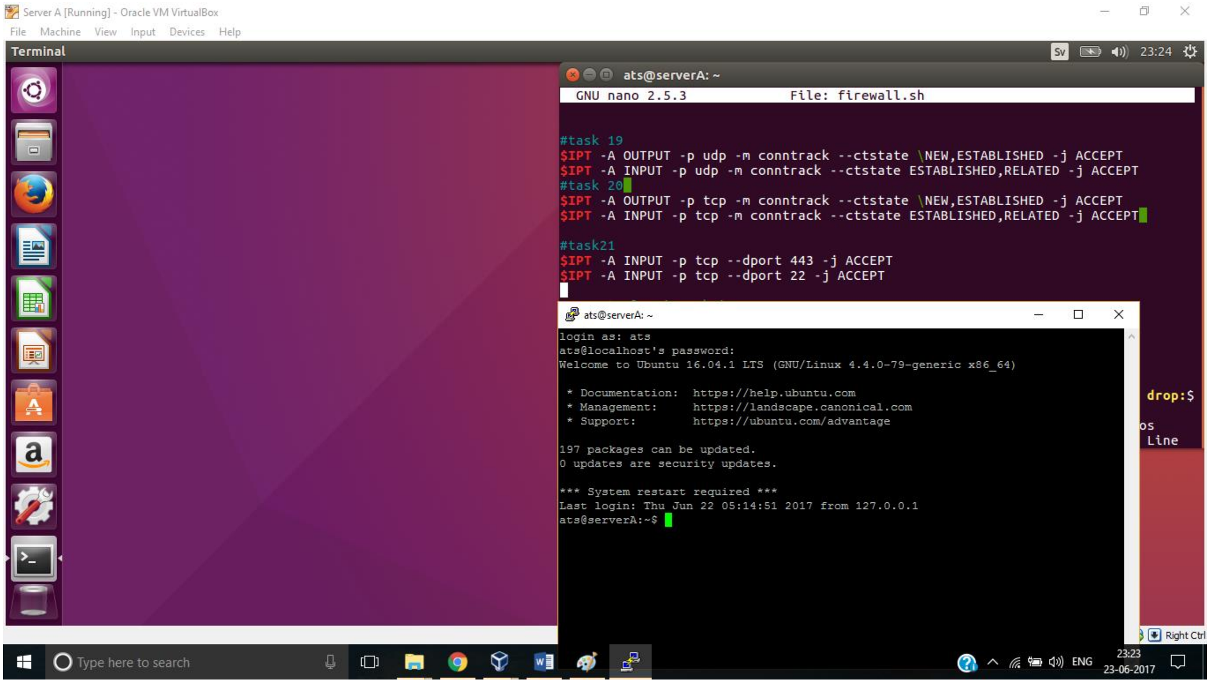
```
  GNU nano 2.5.3              File: firewall.sh


#task 19
$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
$IPT -A OUTPUT -p tcp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -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
$IPT -t filter -A input_log -j LOG --log-level notice --log-prefix "input drop:$
$IPT -t filter -A output_log -j LOG --log-level notice --log-prefix "output dro$
$IPT -t filter -A forward_log -j LOG --log-level notice --log-prefix "forward d$
echo "Added logging"


^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 Linter ^_ Go To Line
```

Task 21:

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

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

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



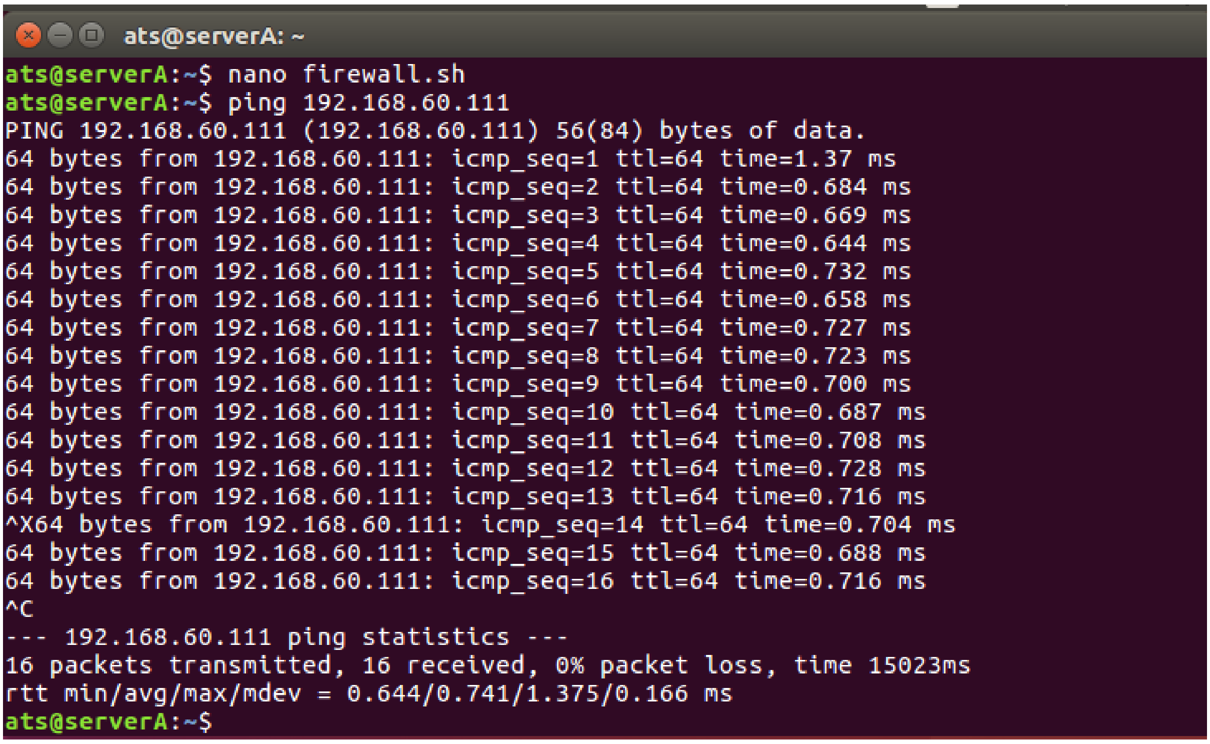
Server A [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Sv
23:24
ats@serverA: ~

```
  GNU nano 2.5.3              File: firewall.sh


#task 19
$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
$IPT -A OUTPUT -p tcp -m conntrack --ctstate \NEW,ESTABLISHED -j ACCEPT
$IPT -A INPUT -p tcp -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

#task21
$IPT -A INPUT -p tcp --dport 443 -j ACCEPT
$IPT -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
login as: ats
ats@localhost's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

197 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Thu Jun 22 05:14:51 2017 from 127.0.0.1
ats@serverA:~$
```

Task 22:

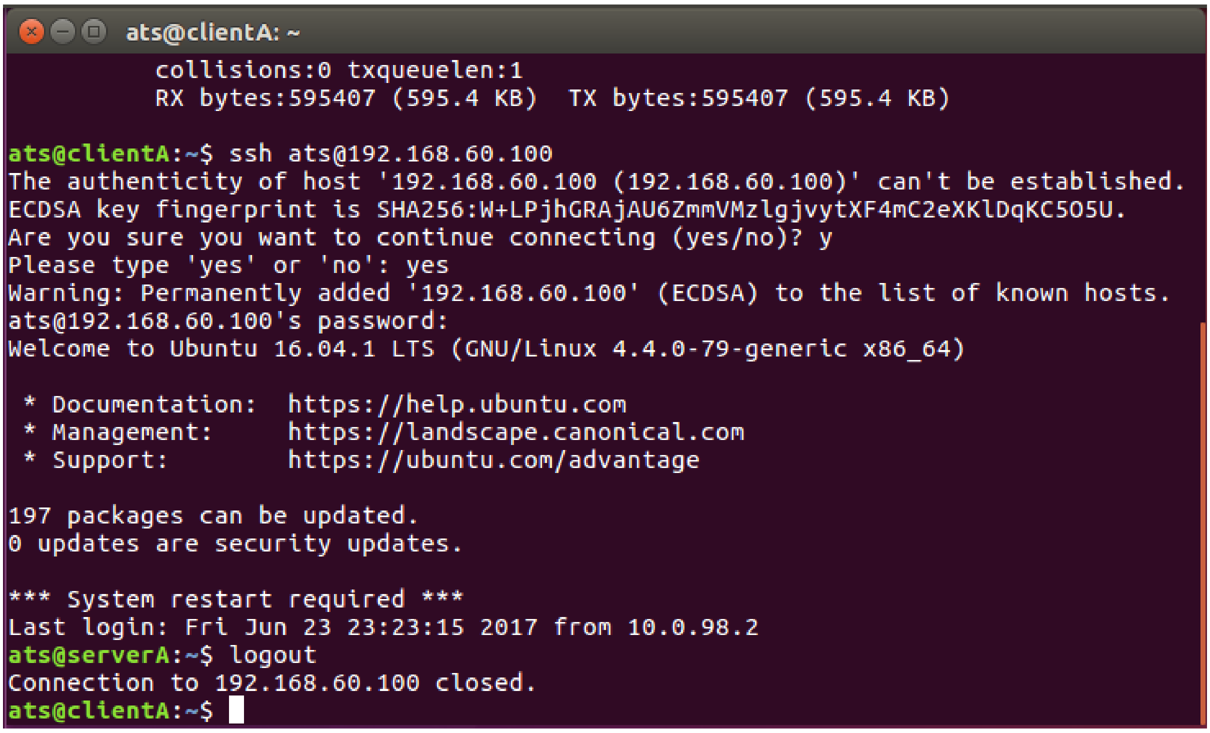
To add the firewall rules to ping server A from client A. add the following rules in firewall.sh



```
ats@serverA:~$ nano firewall.sh
ats@serverA:~$ ping 192.168.60.111
PING 192.168.60.111 (192.168.60.111) 56(84) bytes of data.
64 bytes from 192.168.60.111: icmp_seq=1 ttl=64 time=1.37 ms
64 bytes from 192.168.60.111: icmp_seq=2 ttl=64 time=0.684 ms
64 bytes from 192.168.60.111: icmp_seq=3 ttl=64 time=0.669 ms
64 bytes from 192.168.60.111: icmp_seq=4 ttl=64 time=0.644 ms
64 bytes from 192.168.60.111: icmp_seq=5 ttl=64 time=0.732 ms
64 bytes from 192.168.60.111: icmp_seq=6 ttl=64 time=0.658 ms
64 bytes from 192.168.60.111: icmp_seq=7 ttl=64 time=0.727 ms
64 bytes from 192.168.60.111: icmp_seq=8 ttl=64 time=0.723 ms
64 bytes from 192.168.60.111: icmp_seq=9 ttl=64 time=0.700 ms
64 bytes from 192.168.60.111: icmp_seq=10 ttl=64 time=0.687 ms
64 bytes from 192.168.60.111: icmp_seq=11 ttl=64 time=0.708 ms
64 bytes from 192.168.60.111: icmp_seq=12 ttl=64 time=0.728 ms
64 bytes from 192.168.60.111: icmp_seq=13 ttl=64 time=0.716 ms
^X64 bytes from 192.168.60.111: icmp_seq=14 ttl=64 time=0.704 ms
64 bytes from 192.168.60.111: icmp_seq=15 ttl=64 time=0.688 ms
64 bytes from 192.168.60.111: icmp_seq=16 ttl=64 time=0.716 ms
^C
--- 192.168.60.111 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 15023ms
rtt min/avg/max/mdev = 0.644/0.741/1.375/0.166 ms
ats@serverA:~$
```

Task 23:

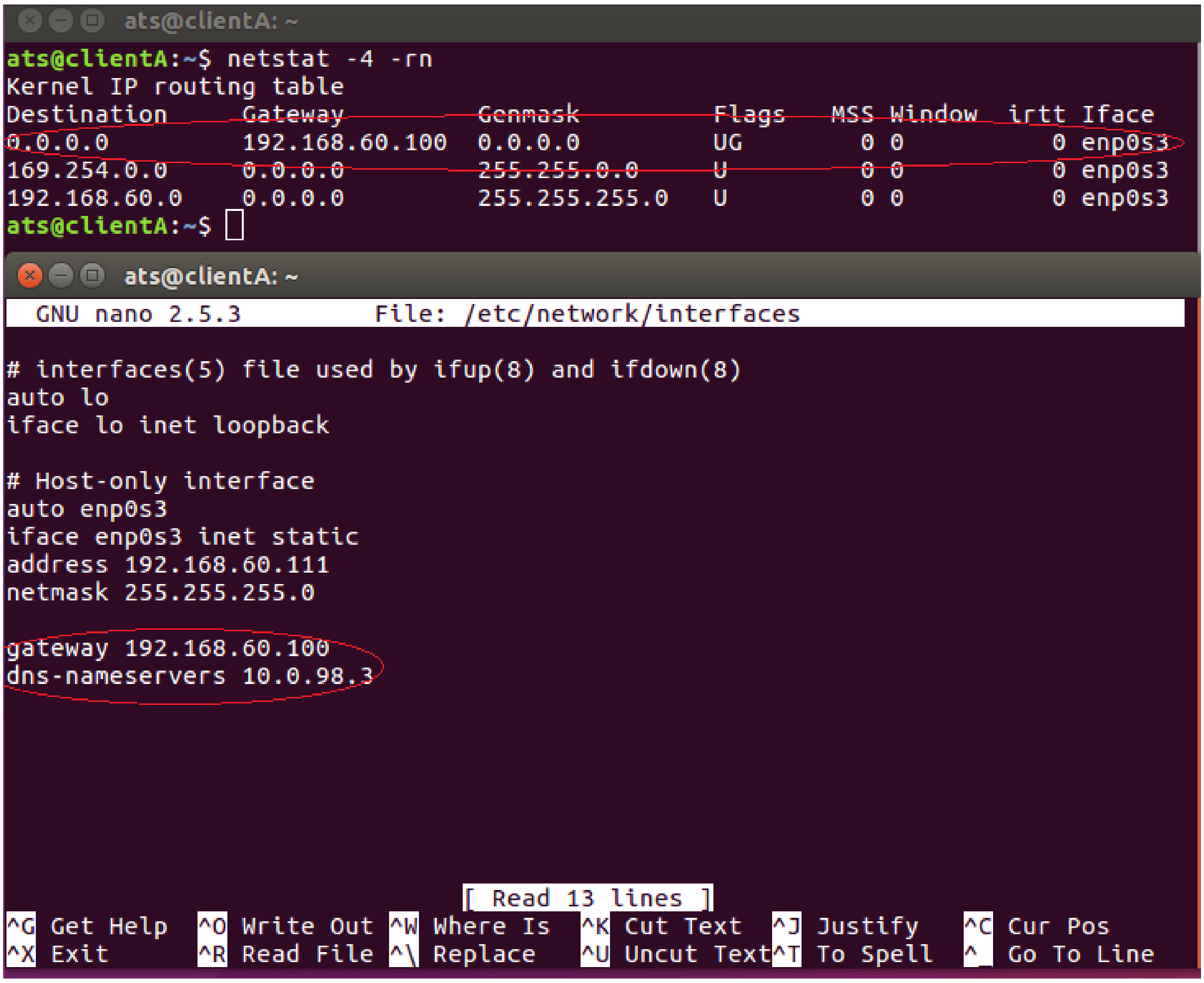
To fix the firewall rules such that we can SSH from Client A to Server A. Add the following rules to the firewall.



```
          collisions:0 txqueuelen:1
          RX bytes:595407 (595.4 KB)  TX bytes:595407 (595.4 KB)

ats@clientA:~$ ssh ats@192.168.60.100
The authenticity of host '192.168.60.100 (192.168.60.100)' can't be established.
ECDSA key fingerprint is SHA256:W+LPjhGRAjAU6ZmmVMzlgjvytXF4mC2eXKlDqKC5O5U.
Are you sure you want to continue connecting (yes/no)? y
Please type 'yes' or 'no': yes
Warning: Permanently added '192.168.60.100' (ECDSA) to the list of known hosts.
ats@192.168.60.100's password:
Welcome to Ubuntu 16.04.1 LTS (GNU/Linux 4.4.0-79-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

197 packages can be updated.
0 updates are security updates.

*** System restart required ***
Last login: Fri Jun 23 23:23:15 2017 from 10.0.98.2
ats@serverA:~$ logout
Connection to 192.168.60.100 closed.
ats@clientA:~$ 
```

Task 24:

In this task and the gateway and dns-nameserver in /etc/network/interface file in client A, so that we can able to add gateway and dns-servername to client A.



```
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:~$
```

```
  GNU nano 2.5.3           File: /etc/network/interfaces

# 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







                              [ Read 13 lines ]
^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:

To execute 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:

To 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 “

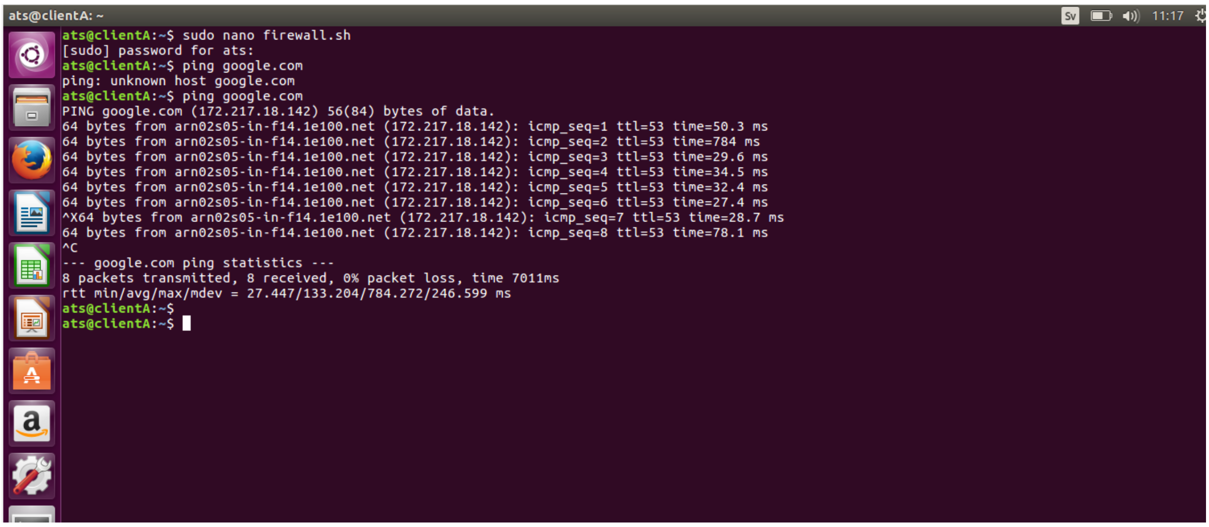
“$IPT -t filter -A FORWARD -i $NIF -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT”

After changing these rules the packets are forwarded to the NAT interface. But here 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 tell Server A to use NAT (more specifically Source NAT - SNAT). In order to do this we need to enable the SNAT on Server A.

Task 27:

To fix the problem outlined above you need to tell Server A to do SNAT on the NAT interface. You must add the following iptables rule.

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



```
ats@clientA:~$ sudo nano firewall.sh
[sudo] password for ats:
ats@clientA:~$ ping google.com
ping: unknown host google.com
ats@clientA:~$ ping google.com
PING google.com (172.217.18.142) 56(84) bytes of data.
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=1 ttl=53 time=50.3 ms
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=2 ttl=53 time=784 ms
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=3 ttl=53 time=29.6 ms
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=4 ttl=53 time=34.5 ms
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=5 ttl=53 time=32.4 ms
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=6 ttl=53 time=27.4 ms
^X64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=7 ttl=53 time=28.7 ms
64 bytes from arn02s05-in-f14.1e100.net (172.217.18.142): icmp_seq=8 ttl=53 time=78.1 ms
^C
--- google.com ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7011ms
rtt min/avg/max/mdev = 27.447/133.204/784.272/246.599 ms
ats@clientA:~$
ats@clientA:~$ 
```