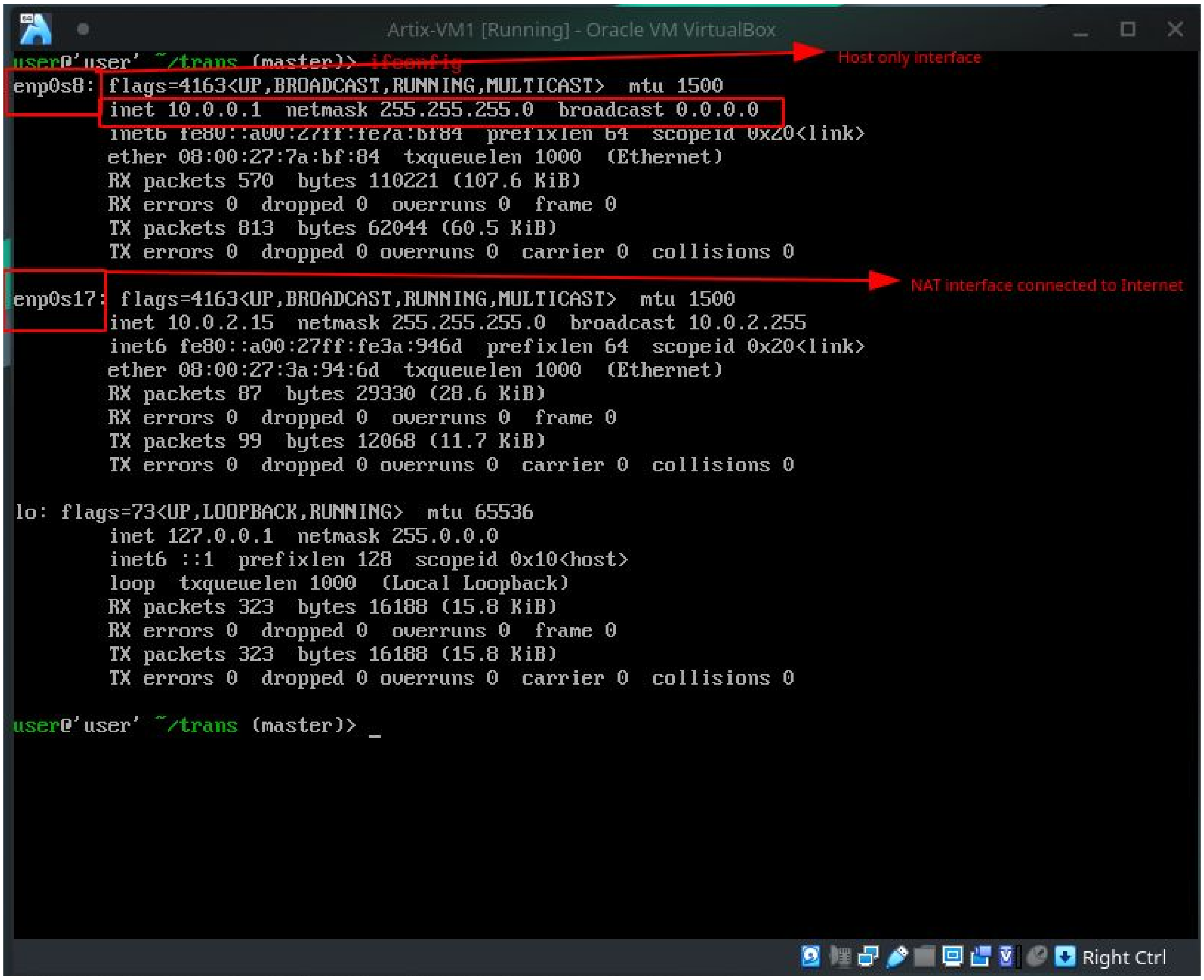
NSSII Exercise 1

Manavjeet Singh, 2018295

Procedure Details

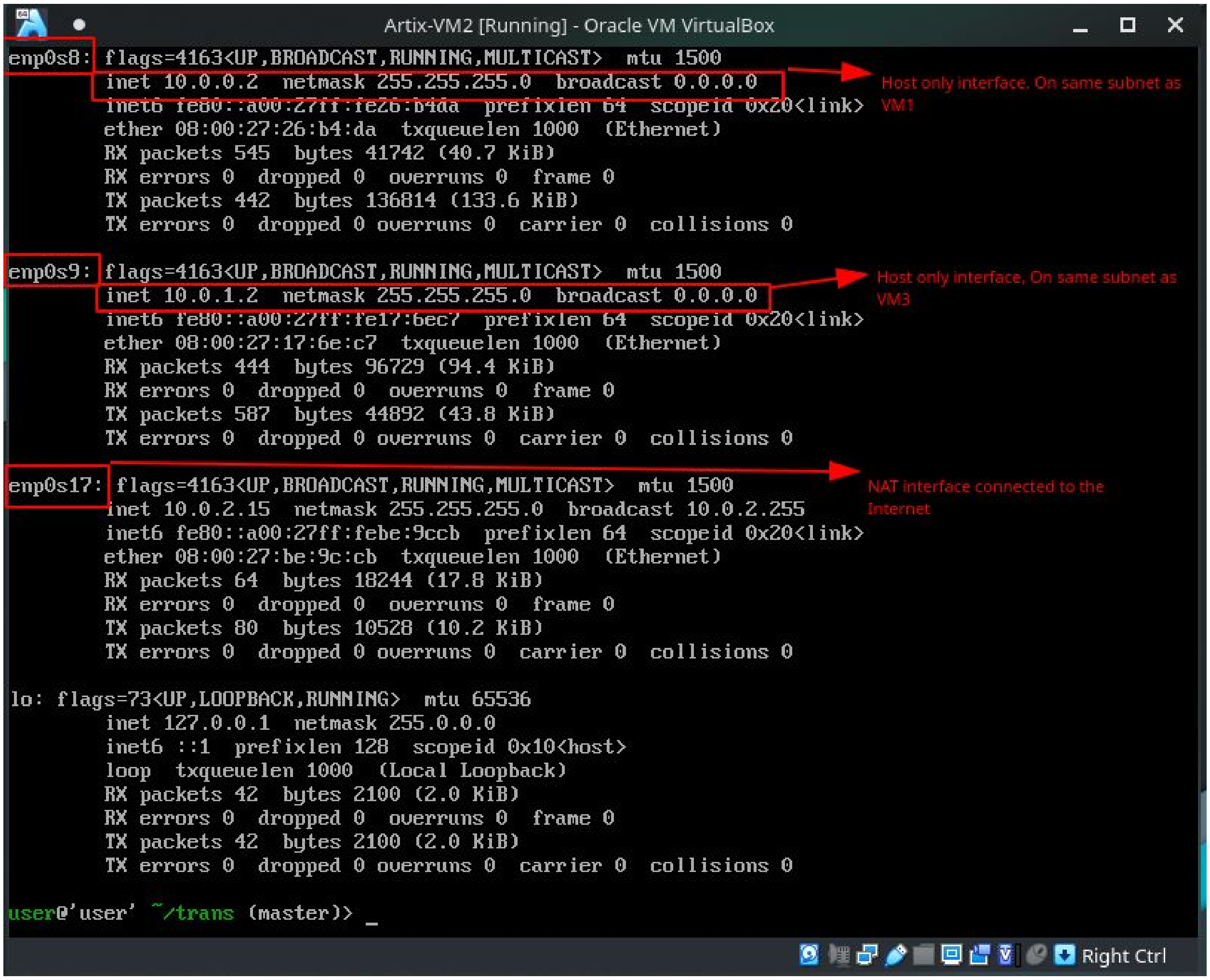
Three VMs on Virtual Box, running Artix OS were used. The network configurations are as follows:

VM1


Artix-VM1 [Running] - Oracle VM VirtualBox
Host only interface
NAT interface connected to Internet
Right Ctrl

```
user@'user' ~/trans (master)> ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.1  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::a00:27ff:fe7a:bf84  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7a:bf:84  txqueuelen 1000  (Ethernet)
        RX packets 570  bytes 110221 (107.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 813  bytes 62044 (60.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s17: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe3a:946d  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:3a:94:6d  txqueuelen 1000  (Ethernet)
        RX packets 87  bytes 29330 (28.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 99  bytes 12068 (11.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 323  bytes 16188 (15.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 323  bytes 16188 (15.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

user@'user' ~/trans (master)> _
```

VM2



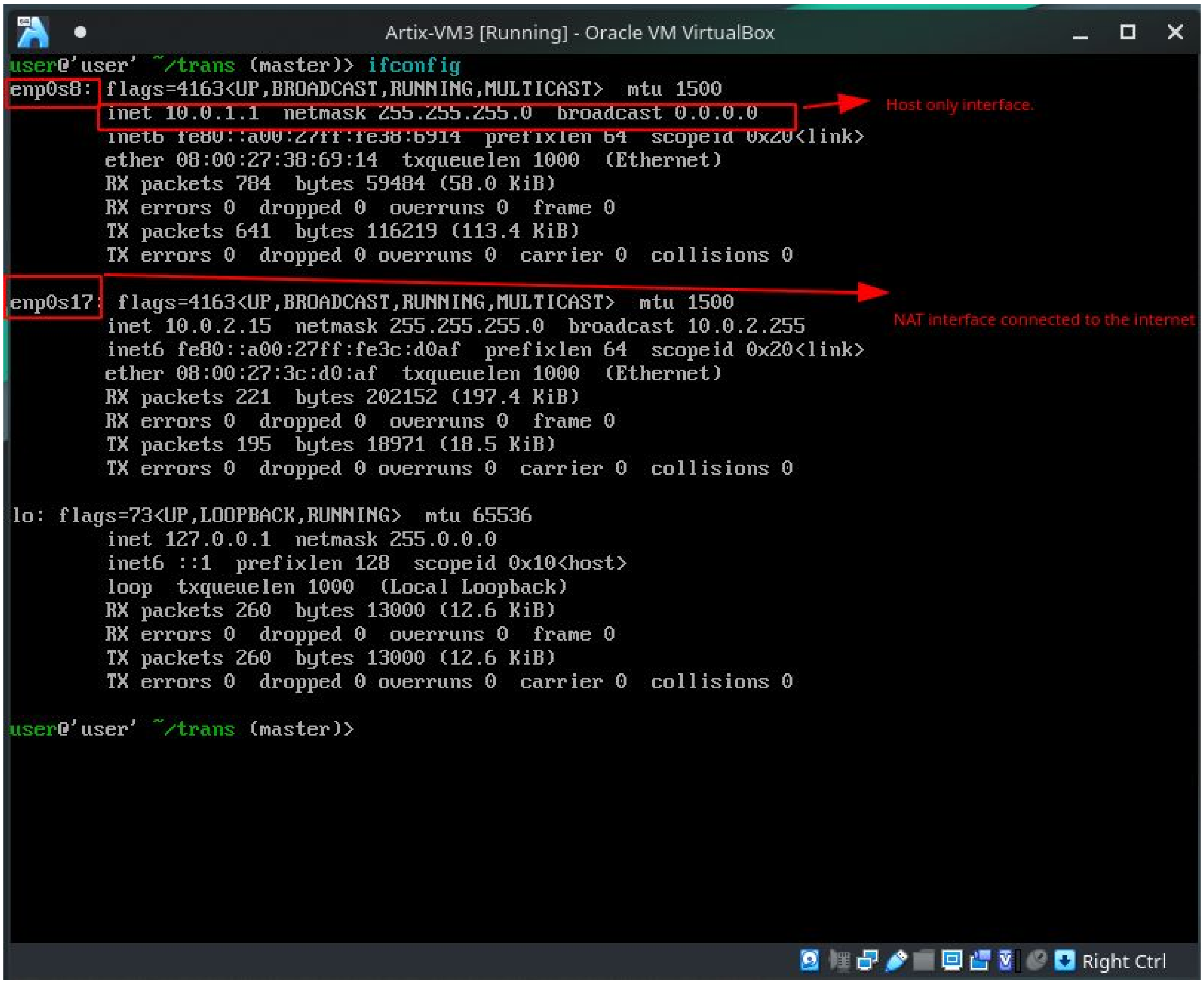
```
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.0.2  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::a00:27ff:fe26:b4da  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:26:b4:da  txqueuelen 1000  (Ethernet)
        RX packets 545  bytes 41742 (40.7 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 442  bytes 136814 (133.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.2  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::a00:27ff:fe17:6ec7  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:17:6e:c7  txqueuelen 1000  (Ethernet)
        RX packets 444  bytes 96729 (94.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 587  bytes 44892 (43.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s17: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:febe:9ccb  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:be:9c:cb  txqueuelen 1000  (Ethernet)
        RX packets 64  bytes 18244 (17.8 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 80  bytes 10528 (10.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 42  bytes 2100 (2.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 42  bytes 2100 (2.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

user@'user' ~/trans (master)> _
```

Host only interface. On same subnet as VM1

Host only interface, On same subnet as VM3

NAT interface connected to the Internet

VM3



```
user@'user' ~/trans (master)> ifconfig
enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.1.1  netmask 255.255.255.0  broadcast 0.0.0.0
        inet6 fe80::a00:27ff:fe38:6914  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:38:69:14  txqueuelen 1000  (Ethernet)
        RX packets 784  bytes 59484 (58.0 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 641  bytes 116219 (113.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s17: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe3c:d0af  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:3c:d0:af  txqueuelen 1000  (Ethernet)
        RX packets 221  bytes 202152 (197.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 195  bytes 18971 (18.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 260  bytes 13000 (12.6 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 260  bytes 13000 (12.6 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

user@'user' ~/trans (master)>
```

Host only interface.

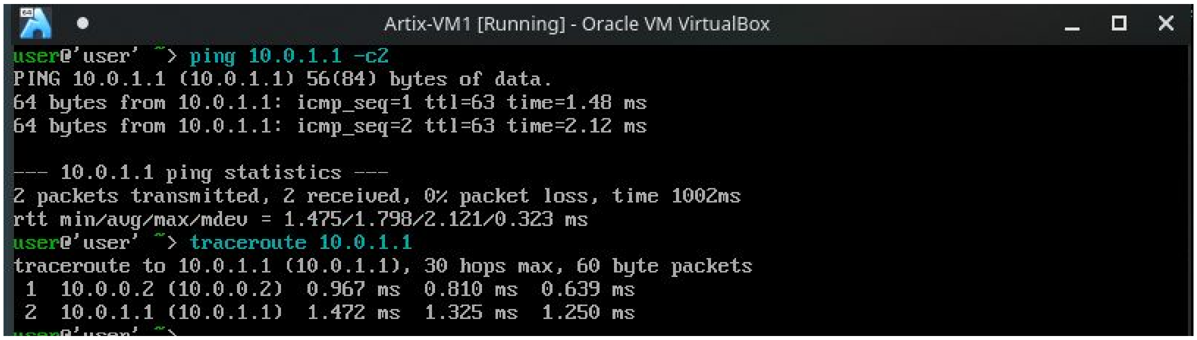
NAT interface connected to the internet

As evident in the above screen shots, VM1 and VM2(iface 1) are on subnet 10.0.0.0/24 and VM3 and VM2(iface 2) are on subnet 10.0.1.0/24.

Task 1

- The following commands were used to assign IP addresses to the interfaces.
    - Connecting to NAT interface for internet access
        - `ip link set dev enp0s17 up` **# To "up" the network interface"**
        - `dhcpcd enp0s17` **# To automatically assign the ip address and connect to the host machine for internet access.**
    - Configuring up Host-only interfaces
        - **VM1**
            - `ip link set dev enp0s8 up` **# To "up" the network interface"**

    - `ip addr add 10.0.0.1/24 dev enp0s8` **# Assigning IP address with subnet mask to the interface**
    - `route add -net 10.0.1.0 netmask 255.255.255.0 gw 10.0.0.2` **#Adding route for the given subnet in the routing table manually**
  - **VM2**
    - `ip link set dev enp0s8 up` **# To “up” the network interface”**
    - `ip addr add 10.0.0.2/24 dev enp0s8` **# Assigning IP address with subnet mask to the interface**
    - `route add -net 10.0.0.0 netmask 255.255.255.0 gw 10.0.0.2` **#Adding route for the given subnet in the routing table manually**
    - `ip link set dev enp0s9 up` **# To “up” the network interface”**
    - `ip addr add 10.0.1.2/24 dev enp0s9` **# Assigning IP address with subnet mask to the interface**
    - `route add -net 10.0.1.0 netmask 255.255.255.0 gw 10.0.1.2` **#Adding route for the given subnet in the routing table manually**
  - **VM3**
    - `ip link set dev enp0s8 up` **# To “up” the network interface”**
    - `ip addr add 10.0.1.1/24 dev enp0s8` **# Assigning IP address with subnet mask to the interface**
    - `route add -net 10.0.0.0 netmask 255.255.255.0 gw 10.0.1.2` **#Adding route for the given subnet in the routing table manually**
- To enable IP forwarding, `net.ipv4.ip_forward=1` was added to /etc/sysctl.conf file and command `sysctl -p` was issued to load the settings from the file. The following screenshot shows that VM1(10.0.0.1) was able to ping VM3(10.0.1.1) and also the route was followed through VM2(10.0.0.2)



```
user@'user' ~> ping 10.0.1.1 -c2
PING 10.0.1.1 (10.0.1.1) 56(84) bytes of data.
64 bytes from 10.0.1.1: icmp_seq=1 ttl=63 time=1.48 ms
64 bytes from 10.0.1.1: icmp_seq=2 ttl=63 time=2.12 ms

--- 10.0.1.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 1.475/1.798/2.121/0.323 ms
user@'user' ~> traceroute 10.0.1.1
traceroute to 10.0.1.1 (10.0.1.1), 30 hops max, 60 byte packets
 1  10.0.0.2 (10.0.0.2)  0.967 ms  0.810 ms  0.639 ms
 2  10.0.1.1 (10.0.1.1)  1.472 ms  1.325 ms  1.250 ms
user@'user' ~>
```

- Commands to configure Bi-directional NAT for allowing only TCP connections from :
    - `sudo iptables -A OUTPUT -s 10.0.1.2/24 -j` **DROP #Dropping all packets generated from VM2**
    - **#-----For TCP requests on Port 80---**
    - `sudo iptables -A FORWARD -i enp0s8 -o enp0s9 -p tcp --syn --dport 80 -m conntrack --ctstate NEW -j ACCEPT` **#Forwarding TCP SYN packets from VM2(Iface1) towards VM2(Iface2) on port 80 and track connection for stateful firewall in using conntrack extension.**
    - `sudo iptables -A FORWARD -i enp0s8 -o enp0s9 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT` **#Forwarding all the packets from VM2(Iface1) towards VM2(Iface2) related to already established TCP connections.**
    - `sudo iptables -A FORWARD -o enp0s8 -i enp0s9 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT` **#Forwarding all the packets from VM2(Iface2) towards VM2(Iface1) related to already established TCP connections**
    - `sudo iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 80 -j DNAT --to-destination 10.0.1.1` **# Destination NAT the packets, for packets coming from VM1, change destination IP address to that of VM3**
    - `sudo iptables -t nat -A POSTROUTING -o enp0s9 -p tcp --dport 80 -d 10.0.1.1 -j SNAT --to-source 10.0.1.2` **# Source NAT the packets, for packets going towards VM3 to that of IP address of VM2(iface2)**
    - `sudo iptables -t nat -A PREROUTING -i enp0s9 -p tcp  -j DNAT --to-destination 10.0.0.1` **# Destination NAT the packets, for packets coming from VM3, change destination IP address to that of VM1**
    - `sudo iptables -t nat -A POSTROUTING -o enp0s8 -p tcp  -d 10.0.0.1 -j SNAT --to-source 10.0.0.2` **# Source NAT the packets, for packets going towards VM1 to that of IP address of VM2(iface1)**

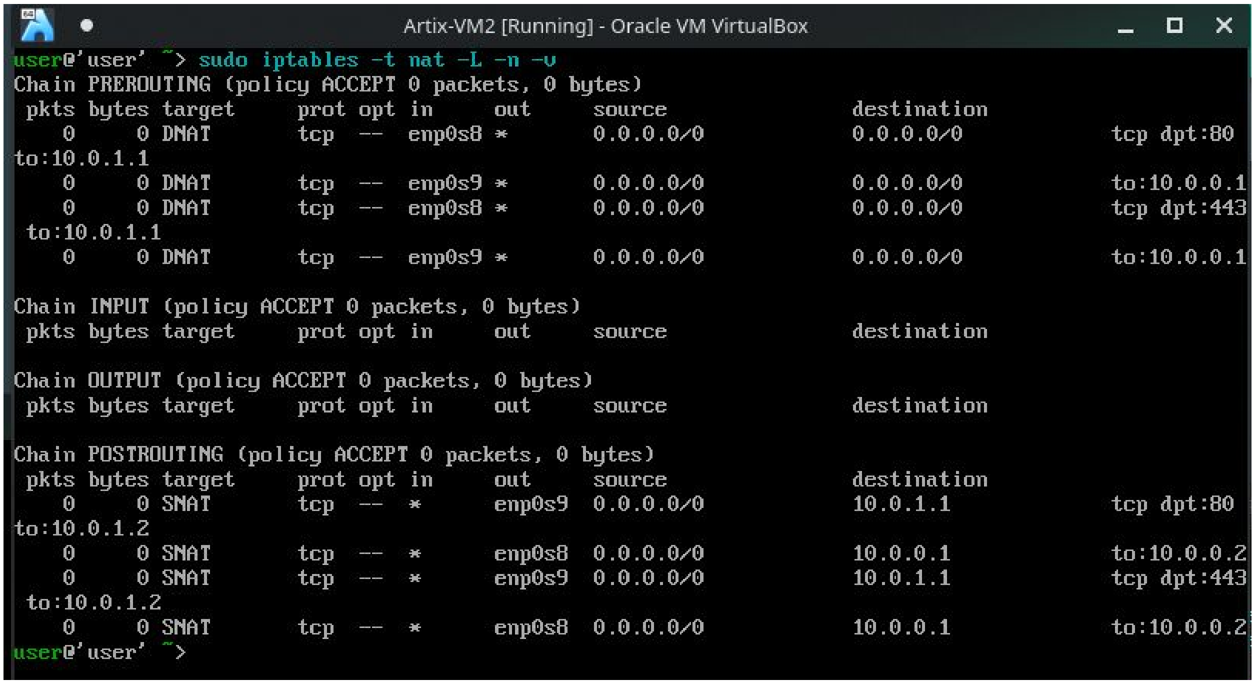
    - **#Similarly for port 443, just replacing port from 80 to 443**

    - `sudo iptables -A FORWARD -i enp0s8 -o enp0s9 -p tcp --syn --dport 443 -m conntrack --ctstate NEW -j ACCEPT`
    - `sudo iptables -A FORWARD -i enp0s8 -o enp0s9 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`
    - `sudo iptables -A FORWARD -o enp0s8 -i enp0s9 -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT`
    - `sudo iptables -t nat -A PREROUTING -i enp0s8 -p tcp --dport 443 -j DNAT --to-destination 10.0.1.1`
    - `sudo iptables -t nat -A POSTROUTING -o enp0s9 -p tcp --dport`

443 -d 10.0.1.1 -j SNAT --to-source 10.0.1.2

  - sudo iptables -t nat -A PREROUTING -i enp0s9 -p tcp -j DNAT --to-destination 10.0.0.1
  - sudo iptables -t nat -A POSTROUTING -o enp0s8 -p tcp -d 10.0.0.1 -j SNAT --to-source 10.0.0.2
  - #----
  - sudo iptables -P FORWARD DROP **#Dropping all the forward packets that are not matched in the rules above.**
  - Reference: https://www.digitalocean.com/community/tutorials/how-to-forward-ports-through-a-linux-gateway-with-iptables

- NAT table:

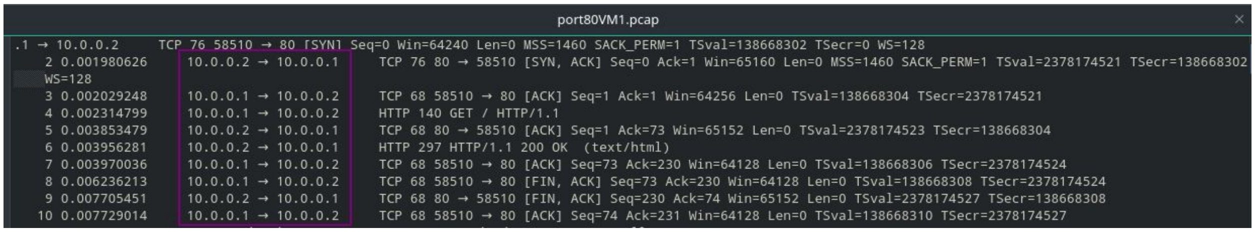


Artix-VM2 [Running] - Oracle VM VirtualBox

```
user@'user' ~> sudo iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DNAT       tcp  --  enp0s8 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:80
to:10.0.1.1
    0     0 DNAT       tcp  --  enp0s9 *       0.0.0.0/0            0.0.0.0/0            to:10.0.0.1
    0     0 DNAT       tcp  --  enp0s8 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:443
 to:10.0.1.1
    0     0 DNAT       tcp  --  enp0s9 *       0.0.0.0/0            0.0.0.0/0            to:10.0.0.1

Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       tcp  --  *      enp0s9  0.0.0.0/0            10.0.1.1             tcp dpt:80
to:10.0.1.2
    0     0 SNAT       tcp  --  *      enp0s8  0.0.0.0/0            10.0.0.1             to:10.0.0.2
    0     0 SNAT       tcp  --  *      enp0s9  0.0.0.0/0            10.0.1.1             tcp dpt:443
 to:10.0.1.2
    0     0 SNAT       tcp  --  *      enp0s8  0.0.0.0/0            10.0.0.1             to:10.0.0.2
user@'user' ~>
```

- The packets on the three machines were captured using tshark and saved in utf-8 format as shown in the screenshots below. The command used was: sudo `tshark -i any > file`.

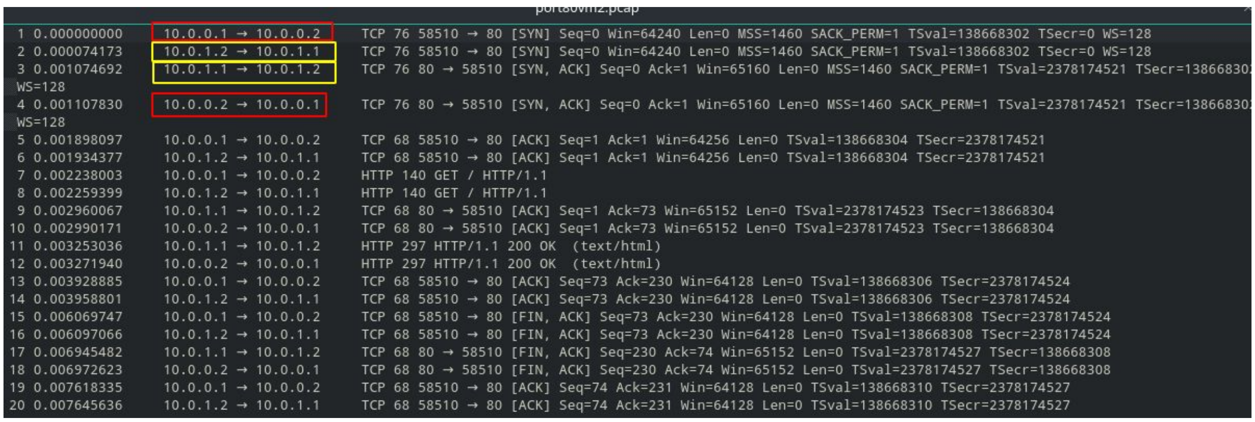
  On VM1



port80VM1.pcap

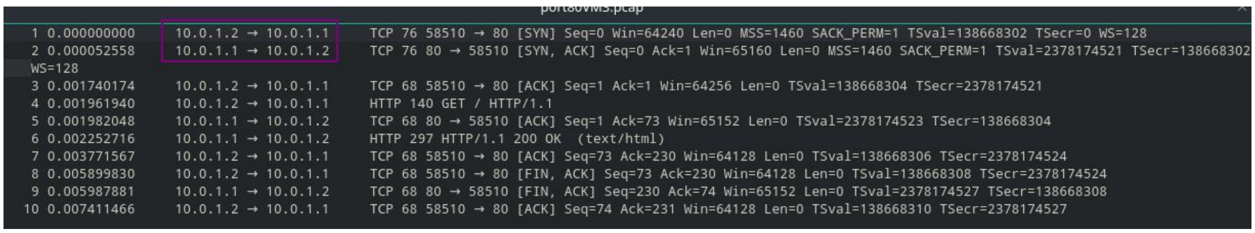
```
.1 → 10.0.0.2     TCP 76 58510 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=138668302 TSecr=0 WS=128
    2 0.001980626    10.0.0.2 → 10.0.0.1    TCP 76 80 → 58510 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2378174521 TSecr=138668302
    WS=128
    3 0.002029248    10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=138668304 TSecr=2378174521
    4 0.002314799    10.0.0.1 → 10.0.0.2    HTTP 140 GET / HTTP/1.1
    5 0.003853479    10.0.0.2 → 10.0.0.1    TCP 68 80 → 58510 [ACK] Seq=1 Ack=73 Win=65152 Len=0 TSval=2378174523 TSecr=138668304
    6 0.003956281    10.0.0.2 → 10.0.0.1    HTTP 297 HTTP/1.1 200 OK  (text/html)
    7 0.003970036    10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668306 TSecr=2378174524
    8 0.006236213    10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [FIN, ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668308 TSecr=2378174524
    9 0.007705451    10.0.0.2 → 10.0.0.1    TCP 68 80 → 58510 [FIN, ACK] Seq=230 Ack=74 Win=65152 Len=0 TSval=2378174527 TSecr=138668308
   10 0.007729014    10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [ACK] Seq=74 Ack=231 Win=64128 Len=0 TSval=138668310 TSecr=2378174527
```

  On VM2



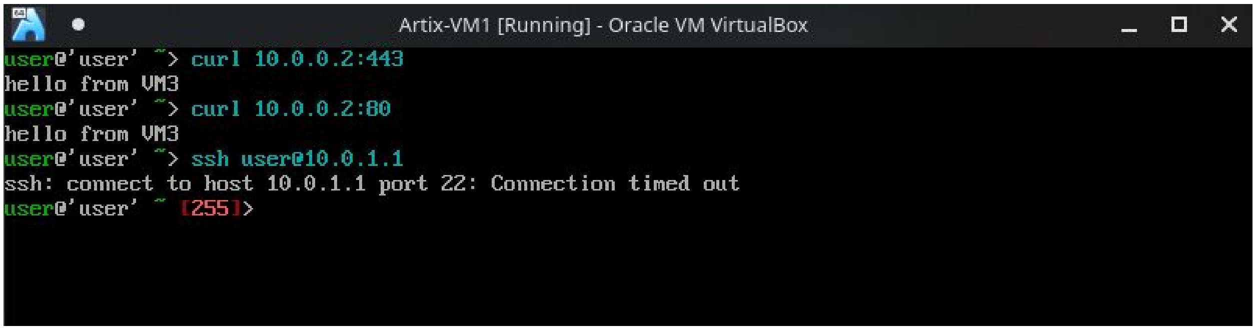
```
 1 0.000000000   10.0.0.1 → 10.0.0.2    TCP 76 58510 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=138668302 TSecr=0 WS=128
 2 0.000074173   10.0.1.2 → 10.0.1.1    TCP 76 58510 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=138668302 TSecr=0 WS=128
 3 0.001074692   10.0.1.1 → 10.0.1.2    TCP 76 80 → 58510 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2378174521 TSecr=13866830
WS=128
 4 0.001107830   10.0.0.2 → 10.0.0.1    TCP 76 80 → 58510 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2378174521 TSecr=13866830
WS=128
 5 0.001898097   10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=138668304 TSecr=2378174521
 6 0.001934377   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=138668304 TSecr=2378174521
 7 0.002238003   10.0.0.1 → 10.0.0.2    HTTP 140 GET / HTTP/1.1
 8 0.002259399   10.0.1.2 → 10.0.1.1    HTTP 140 GET / HTTP/1.1
 9 0.002960067   10.0.1.1 → 10.0.1.2    TCP 68 80 → 58510 [ACK] Seq=1 Ack=73 Win=65152 Len=0 TSval=2378174523 TSecr=138668304
10 0.002990171   10.0.0.2 → 10.0.0.1    TCP 68 80 → 58510 [ACK] Seq=1 Ack=73 Win=65152 Len=0 TSval=2378174523 TSecr=138668304
11 0.003253036   10.0.1.1 → 10.0.1.2    HTTP 297 HTTP/1.1 200 OK  (text/html)
12 0.003271940   10.0.0.2 → 10.0.0.1    HTTP 297 HTTP/1.1 200 OK  (text/html)
13 0.003928885   10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668306 TSecr=2378174524
14 0.003958801   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668306 TSecr=2378174524
15 0.006069747   10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [FIN, ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668308 TSecr=2378174524
16 0.006097066   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [FIN, ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668308 TSecr=2378174524
17 0.006945482   10.0.1.1 → 10.0.1.2    TCP 68 80 → 58510 [FIN, ACK] Seq=230 Ack=74 Win=65152 Len=0 TSval=2378174527 TSecr=138668308
18 0.006972623   10.0.0.2 → 10.0.0.1    TCP 68 80 → 58510 [FIN, ACK] Seq=230 Ack=74 Win=65152 Len=0 TSval=2378174527 TSecr=138668308
19 0.007618335   10.0.0.1 → 10.0.0.2    TCP 68 58510 → 80 [ACK] Seq=74 Ack=231 Win=64128 Len=0 TSval=138668310 TSecr=2378174527
20 0.007645636   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [ACK] Seq=74 Ack=231 Win=64128 Len=0 TSval=138668310 TSecr=2378174527
```

On VM3



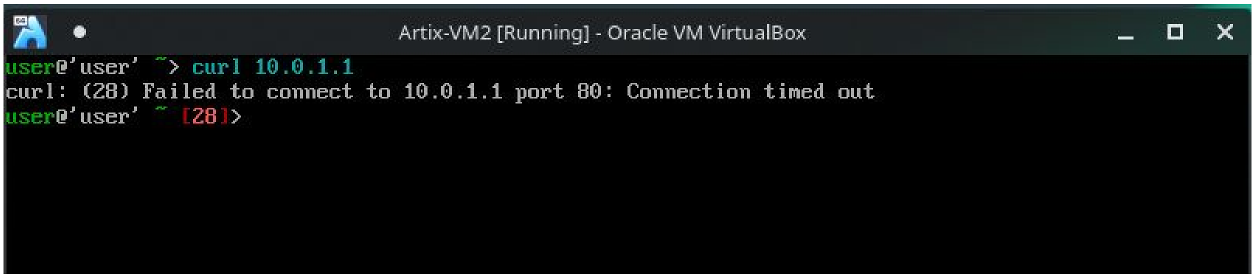
```
  1 0.000000000   10.0.1.2 → 10.0.1.1    TCP 76 58510 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=138668302 TSecr=0 WS=128
  2 0.000052558   10.0.1.1 → 10.0.1.2    TCP 76 80 → 58510 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2378174521 TSecr=138668302
  WS=128
  3 0.001740174   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=138668304 TSecr=2378174521
  4 0.001961940   10.0.1.2 → 10.0.1.1    HTTP 140 GET / HTTP/1.1
  5 0.001982048   10.0.1.1 → 10.0.1.2    TCP 68 80 → 58510 [ACK] Seq=1 Ack=73 Win=65152 Len=0 TSval=2378174523 TSecr=138668304
  6 0.002252716   10.0.1.1 → 10.0.1.2    HTTP 297 HTTP/1.1 200 OK  (text/html)
  7 0.003771567   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668306 TSecr=2378174524
  8 0.005899830   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [FIN, ACK] Seq=73 Ack=230 Win=64128 Len=0 TSval=138668308 TSecr=2378174524
  9 0.005987881   10.0.1.1 → 10.0.1.2    TCP 68 80 → 58510 [FIN, ACK] Seq=230 Ack=74 Win=65152 Len=0 TSval=2378174527 TSecr=138668308
 10 0.007411466   10.0.1.2 → 10.0.1.1    TCP 68 58510 → 80 [ACK] Seq=74 Ack=231 Win=64128 Len=0 TSval=138668310 TSecr=2378174527
```

- Following screen shot shows that HTTP request is working from VM1 to VM3 and SSH gives a timeout error.



```
user@'user' ~> curl 10.0.0.2:443
hello from VM3
user@'user' ~> curl 10.0.0.2:80
hello from VM3
user@'user' ~> ssh user@10.0.1.1
ssh: connect to host 10.0.1.1 port 22: Connection timed out
user@'user' ~ [255]>
```

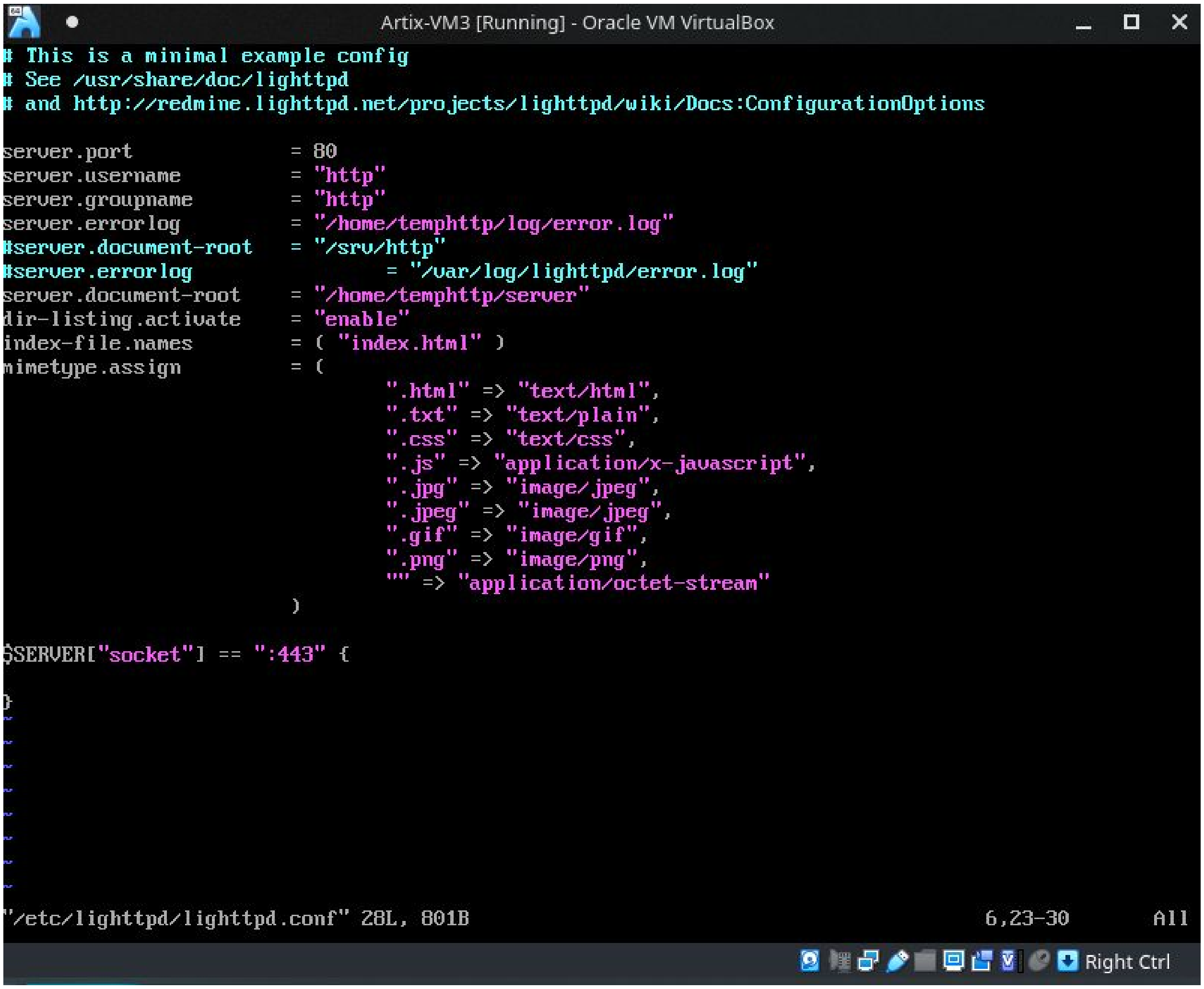
- Curl request from VM2 to VM3 results in a timeout error.



```
user@'user' ~> curl 10.0.1.1
curl: (28) Failed to connect to 10.0.1.1 port 80: Connection timed out
user@'user' ~ [28]>
```

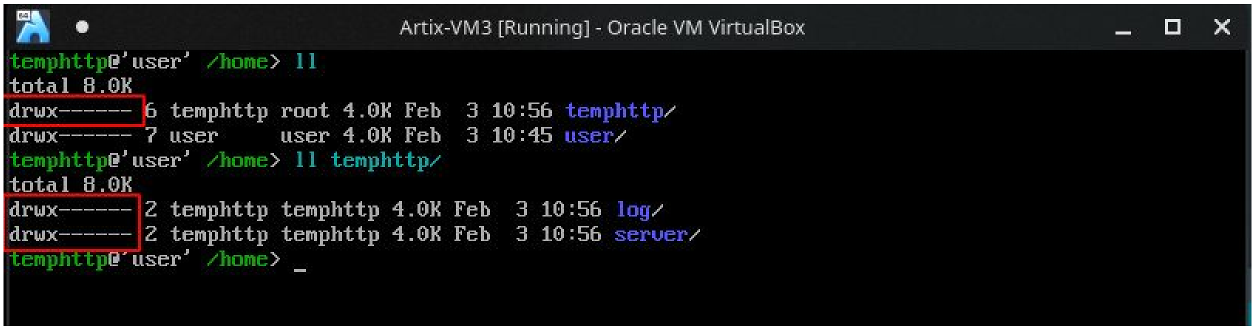
Task 2

- Changed the source root directory and error log file for lighttpd server to a directory in home folder of temphttp user.



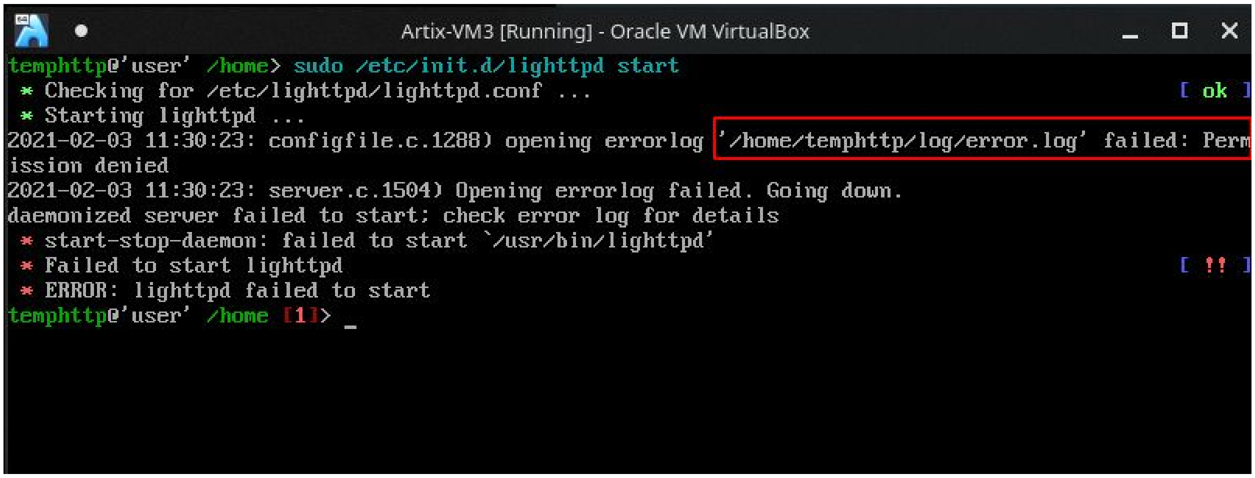
```
# This is a minimal example config
# See /usr/share/doc/lighttpd
# and http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs:ConfigurationOptions

server.port             = 80
server.username         = "http"
server.groupname        = "http"
server.errorlog         = "/home/temphttp/log/error.log"
#server.document-root   = "/srv/http"
#server.errorlog                = "/var/log/lighttpd/error.log"
server.document-root    = "/home/temphttp/server"
dir-listing.activate    = "enable"
index-file.names        = ( "index.html" )
mimetype.assign         = (
                                ".html" => "text/html",
                                ".txt" => "text/plain",
                                ".css" => "text/css",
                                ".js" => "application/x-javascript",
                                ".jpg" => "image/jpeg",
                                ".jpeg" => "image/jpeg",
                                ".gif" => "image/gif",
                                ".png" => "image/png",
                                "" => "application/octet-stream"
                        )

$SERVER["socket"] == ":443" {

}
~
~
~
~
~
~
~
~
"/etc/lighttpd/lighttpd.conf" 28L, 801B                                6,23-30       All
```

- The home /home/temphttp dir is owned by temphttp user and no r,w or x permissions are given to the group or other user.

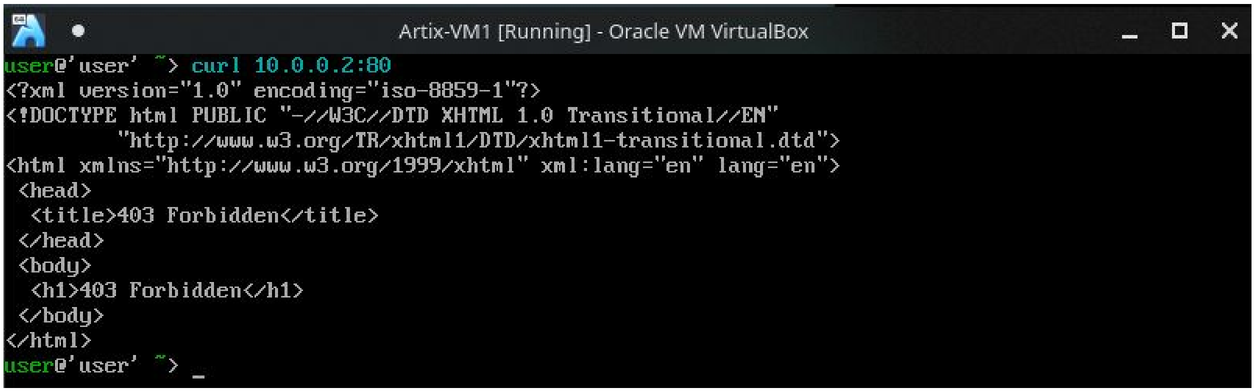


```
temphttp@'user' /home> ll
total 8.0K
drwx------ 6 temphttp root 4.0K Feb  3 10:56 temphttp/
drwx------ 7 user     user 4.0K Feb  3 10:45 user/
temphttp@'user' /home> ll temphttp/
total 8.0K
drwx------ 2 temphttp temphttp 4.0K Feb  3 10:56 log/
drwx------ 2 temphttp temphttp 4.0K Feb  3 10:56 server/
temphttp@'user' /home> _
```

- This way the server is not able to write into the log file thus giving error on startup as shown below. Even if the log file is accessible and the server is running, it gives a 403 forbidden error in VM1. This is because a folder should have a executable permission to open it and here we can see only temphttp has that permission, whereas the server runs as another user called “http”.

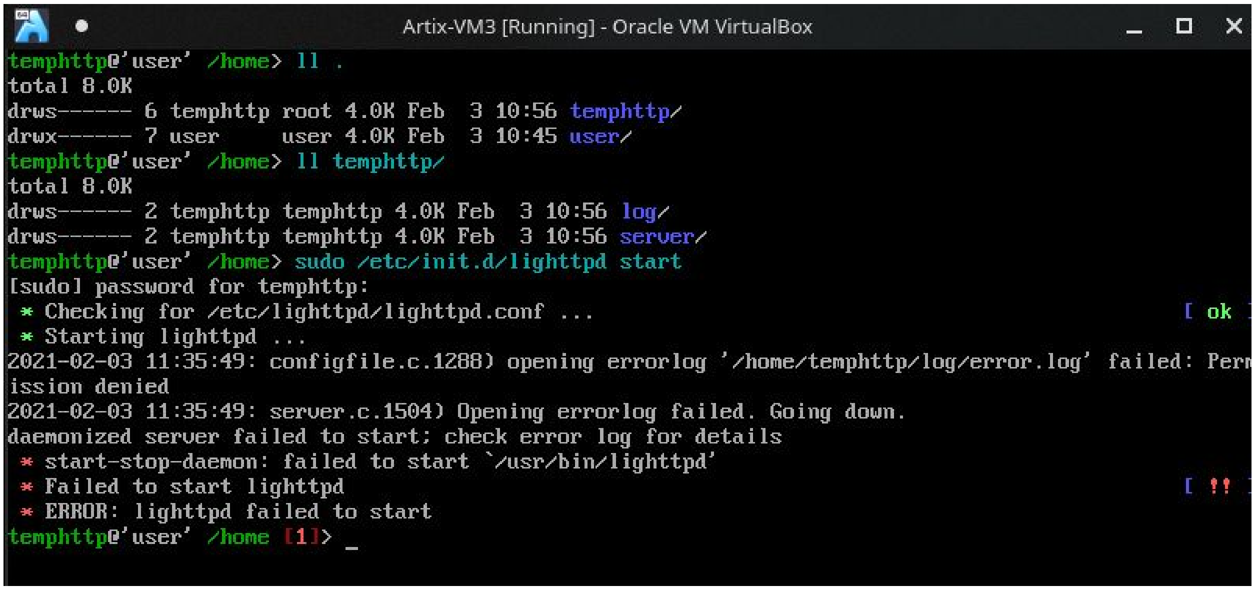


```
temphttp@'user' /home> sudo /etc/init.d/lighttpd start
 * Checking for /etc/lighttpd/lighttpd.conf ...                                      [ ok ]
 * Starting lighttpd ...
2021-02-03 11:30:23: configfile.c.1288) opening errorlog '/home/temphttp/log/error.log' failed: Perm
ission denied
2021-02-03 11:30:23: server.c.1504) Opening errorlog failed. Going down.
daemonized server failed to start; check error log for details
 * start-stop-daemon: failed to start `/usr/bin/lighttpd'
 * Failed to start lighttpd                                                          [ !! ]
 * ERROR: lighttpd failed to start
temphttp@'user' /home [1]> _
```



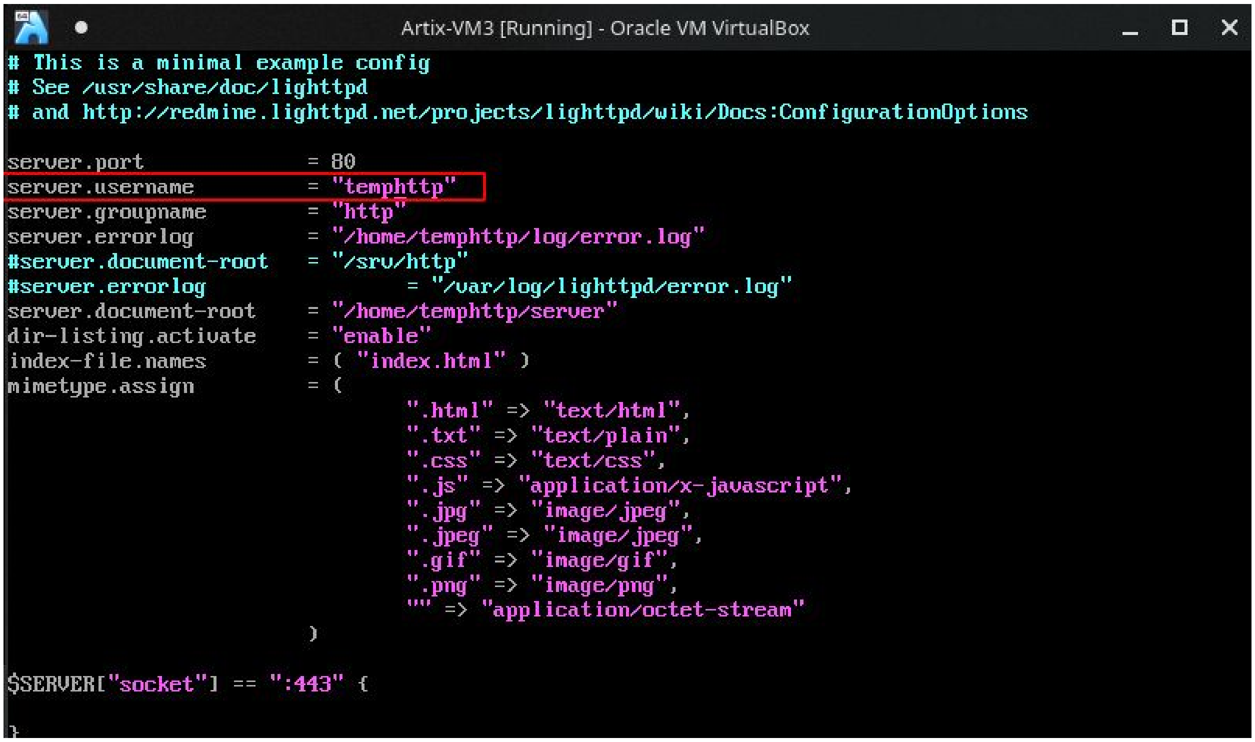
```
user@'user' ~> curl 10.0.0.2:80
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
         "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
 <head>
  <title>403 Forbidden</title>
 </head>
 <body>
  <h1>403 Forbidden</h1>
 </body>
</html>
user@'user' ~> _
```

- Even adding the setuid bit does not help because as per https://linuxconfig.org/how-to-use-special-permissions-the-setuid-setgid-and-sticky-bits, the setuid bit has no effect on directories. Setuid bit is activated using commandchmod u+s <folder name>



```
temphttp@'user' /home> ll .
total 8.0K
drws------ 6 temphttp root 4.0K Feb  3 10:56 temphttp/
drwx------ 7 user     user 4.0K Feb  3 10:45 user/
temphttp@'user' /home> ll temphttp/
total 8.0K
drws------ 2 temphttp temphttp 4.0K Feb  3 10:56 log/
drws------ 2 temphttp temphttp 4.0K Feb  3 10:56 server/
temphttp@'user' /home> sudo /etc/init.d/lighttpd start
[sudo] password for temphttp:
 * Checking for /etc/lighttpd/lighttpd.conf ...                                [ ok
 * Starting lighttpd ...
2021-02-03 11:35:49: configfile.c.1288) opening errorlog '/home/temphttp/log/error.log' failed: Perm
ission denied
2021-02-03 11:35:49: server.c.1504) Opening errorlog failed. Going down.
daemonized server failed to start; check error log for details
 * start-stop-daemon: failed to start `/usr/bin/lighttpd'
 * Failed to start lighttpd                                                    [ !!
 * ERROR: lighttpd failed to start
temphttp@'user' /home [1]> _
```

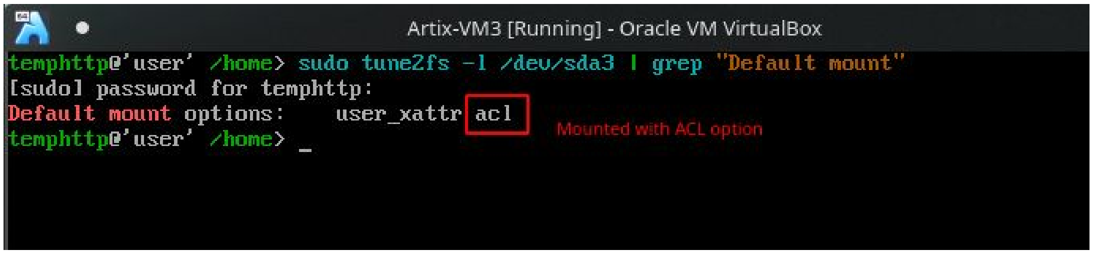
- One way to run the server without giving permission is to change the user name to temphttp in the server config file.



```
# This is a minimal example config
# See /usr/share/doc/lighttpd
# and http://redmine.lighttpd.net/projects/lighttpd/wiki/Docs:ConfigurationOptions

server.port             = 80
server.username         = "temphttp"
server.groupname        = "http"
server.errorlog         = "/home/temphttp/log/error.log"
#server.document-root   = "/srv/http"
#server.errorlog                = "/var/log/lighttpd/error.log"
server.document-root    = "/home/temphttp/server"
dir-listing.activate    = "enable"
index-file.names        = ( "index.html" )
mimetype.assign         = (
                                ".html" => "text/html",
                                ".txt" => "text/plain",
                                ".css" => "text/css",
                                ".js" => "application/x-javascript",
                                ".jpg" => "image/jpeg",
                                ".jpeg" => "image/jpeg",
                                ".gif" => "image/gif",
                                ".png" => "image/png",
                                "" => "application/octet-stream"
                        )

$SERVER["socket"] == ":443" {

}
```

- To make temphttp server directory accessible to server using ACLs.

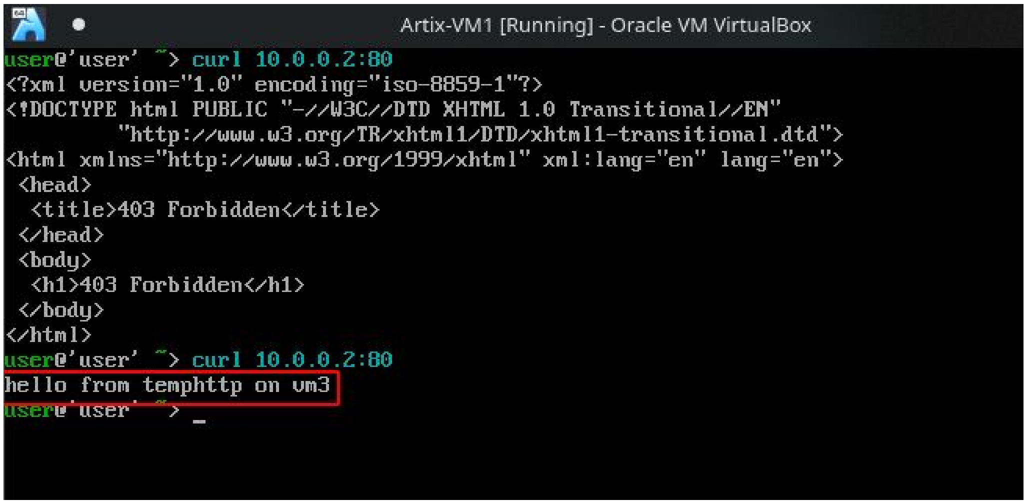
- Making sure that drive is mounted with ACL enabled, using `tune2fs -l /dev/sda3 | grep “Default mount”`

  ```
  temphttp@'user' /home> sudo tune2fs -l /dev/sda3 | grep "Default mount"
  [sudo] password for temphttp:
  Default mount options:    user_xattr acl
  temphttp@'user' /home> _
  ```

  Mounted with ACL option

- To give user **http** execute and read permissions to the home directory of temphttp, the server directory and log directory. (-m is short of --modify and rx means read and execute. Format to give permission- `u:<username>:<permissions>`)
  - `cd /home`
  - `setfacl -m u:http:rx temphttp/ temphttp/server /temphttp/log`
  - Setting the mask so that permission for user http is correct using `setfacl -m mask:rx temphttp/ temphttp/server /temphttp/log`
  - Now checking if server is accessible from VM1

    ```
    user@'user' ~> curl 10.0.0.2:80
    <?xml version="1.0" encoding="iso-8859-1"?>
    <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
             "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
    <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
     <head>
      <title>403 Forbidden</title>
     </head>
     <body>
      <h1>403 Forbidden</h1>
     </body>
    </html>
    user@'user' ~> curl 10.0.0.2:80
    hello from temphttp on vm3
    user@'user' ~> _
    ```

References

- https://www.digitalocean.com/community/tutorials/how-to-forward-ports-through-a-linux-gateway-with-iptables
- https://wiki.archlinux.org/index.php/Lighttpd
- https://linuxconfig.org/how-to-use-special-permissions-the-setuid-setgid-and-sticky-bits
- https://linuxconfig.org/how-to-manage-acls-on-linux