

1. Use the correct ifconfig command to display the current network configuration. Highlight your IP address, MAC address, and the network mask.

```
(antoinette-elam@Kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fd17:625c:f037:2:ef16:3a0e:7566:58f prefixlen 64 scopeid 0x0<
global>
    inet6 fd17:625c:f037:2:a00:27ff:fe28:662f prefixlen 64 scopeid 0x0<
global>
    inet6 fe80::a00:27ff:fe28:662f prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:28:66:2f txqueuelen 1000 (Ethernet)
    RX packets 30 bytes 11847 (11.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 67 bytes 14238 (13.9 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 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

2. Use the correct route command to display the current routing table.

```
(antoinette-elam@Kali)-[~]
└─$ route -n
Kernel IP routing table
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0          10.0.2.2      0.0.0.0         UG    100    0      0 eth0
10.0.2.0         0.0.0.0       255.255.255.0   U     100    0      0 eth0
```

3. Use the netstat command to list current TCP connections.

```
(antoinette-elam@Kali)-[~]
└─$ ss -t
State Recv-Q Send-Q Local Address:Port Peer Address:Port
```

4. Use the ping command to determine if the ubuntu.com system is accessible via the network.

```
(antoinette-elam@Kali)-[~]
$ ping -c 10 ubuntu.com
PING ubuntu.com (185.125.190.20) 56(84) bytes of data.
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=1 ttl=255 time=96.6 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=2 ttl=255 time=93.9 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=3 ttl=255 time=97.0 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=4 ttl=255 time=95.1 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=5 ttl=255 time=96.5 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=6 ttl=255 time=104 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=7 ttl=255 time=96.7 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=8 ttl=255 time=97.5 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=9 ttl=255 time=95.8 ms
64 bytes from website-content-cache-1.ps5.canonical.com (185.125.190.20): icmp
p_seq=10 ttl=255 time=94.9 ms

— ubuntu.com ping statistics —
10 packets transmitted, 10 received, 0% packet loss, time 9038ms
rtt min/avg/max/mdev = 93.936/96.795/104.022/2.621 ms
```

(Use the correct option to send 10 ping requests only.)

5. Use the host command to perform a DNS query on www.odu.edu

```
(antoinette-elam@Kali)-[~]
$ host www.odu.edu
www.odu.edu has address 35.170.140.174
```

6. Use the cat command to display the contents of the file that contains the system's hostname.

```
(antoinette-elam@Kali)-[~]
$ cat /etc/hostname
Kali
```

7. Use the cat command to display the contents of the file that contains the DNS servers for this

```
(antoinette-elam@Kali)-[~]
$ cat /etc/resolv.conf
# Generated by NetworkManager
nameserver 75.75.75.75
nameserver 75.75.76.76
nameserver fd17:625c:f037:2::3
```

system.

8. Edit the same file you display in the previous step, set the system's hostname to your MIDAS ID

permanently. Reboot system and repeat step 6.

```
(antoINETte-elam@Kali)-[~]
$ sudo nano /etc/hosts
GNU nano 8.4 /etc/hosts *
127.0.0.1 localhost
127.0.1.1 aelam010

# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

(antoINETte-elam@aELam010)-[~]
$ cat /etc/hostname
aelam010
```

### Task B – A Different Network Setting

1. Change the VM network connection from NAT to the bridge mode (you will lose your Internet connection if you are connected to the ODU campus Wi-Fi network, but it is okay).
2. Reboot your system, then repeat Steps 1 – 7 in Task A.

```
(antoINETte-elam@aELam010)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 2601:5cf:4781:b210:a00:27ff:fe28:662f prefixlen 64 scopeid 0x
0<global>
    inet6 fe80::a00:27ff:fe28:662f prefixlen 64 scopeid 0x20<link>
    inet6 2601:5cf:4781:b210:5cc5:f072:ad38:9c1d prefixlen 64 scopeid 0
x0<global>
    ether 08:00:27:28:66:2f txqueuelen 1000 (Ethernet)
    RX packets 235 bytes 44096 (43.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 259 bytes 37446 (36.5 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 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(antoINETte-elam@aELam010)-[~]
$ route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
```

```
(antoINETte-elam@aelam010)-[~]
$ netstat -t
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State

(antoINETte-elam@aelam010)-[~]
$ ping -c 10 ubuntu.com
ping: ubuntu.com: Temporary failure in name resolution

(antoINETte-elam@aelam010)-[~]
$ host www.odu.edu
;; communications error to 2001:558:feed::1#53: timed out
;; communications error to 2001:558:feed::1#53: timed out
;; communications error to 2001:558:feed::2#53: timed out
;; no servers could be reached

(antoINETte-elam@aelam010)-[~]
$ cat /etc/hostname
aelam010

(antoINETte-elam@aelam010)-[~]
$ cat /etc/resolv.conf
# Generated by NetworkManager
```

3. Highlight the differences at the end of each step and discuss what do you find.

Step 1:

My virtual machine had a virtual NAT IP address when it was in NAT mode. In Bridge mode, the VM attempted to receive an IP from the physical network, resulting in a different IP (or no IP if prohibited). Additionally, the subnet and gateway were altered.

Step 2:

In NAT mode, the default route referred to the VirtualBox NAT internal router. In Bridge mode, the default route was missing. This shows that the VM was trying to connect directly to the real network.

Step 3:

In Bridge mode, the TCP connections showed direct LAN communication rather than NAT-translated connections. Due to the virtual machine's limited or nonexistent internet access, several connections that appeared in NAT mode did not display in Bridge mode.

Step 4:

Ping was successful in NAT mode since VirtualBox NAT offers complete internet access. The VM was unable to connect to a DNS server on the ODU WiFi network, which is why the ping failed in Bridge mode. This demonstrates that college networks are blocking Bridge mode traffic.

Step 5:

The DNS query functioned normally in NAT mode. In Bridge mode, the query failed because the VM had no functional DNS server. This is in line with Step 4's ping test.

Step 6:

NAT and Bridge mode did not differ from one another. Network mode has no bearing on the hostname, which is a system parameter.

Step 7:

VirtualBox automatically sets up the DNS server in NAT mode. In Bridge mode, the DNS server information was missing, which explains why DNS resolution did not operate.