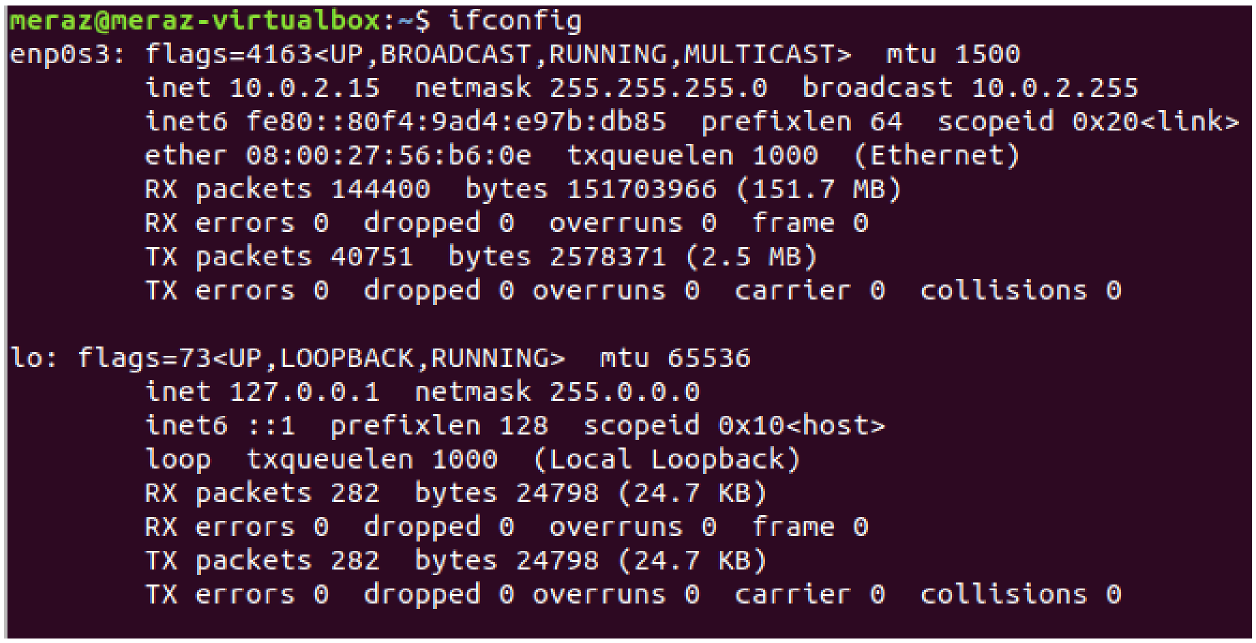
Name:Meraz Ahmed

ID:IT-18005

Linux Assignment -02

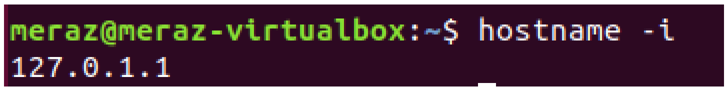
1.Find IP & MAC Find out about network and hardware information for the computer you are currently using.

Whether your connection is wireless or wired, you can also find this information by opening the Apple menu, and then heading to System Preferences > Network. Select your network connection, and then click “Advanced.” You'll find IP address information on the “TCP/IP” tab and the MAC address on the “Hardware” tab.



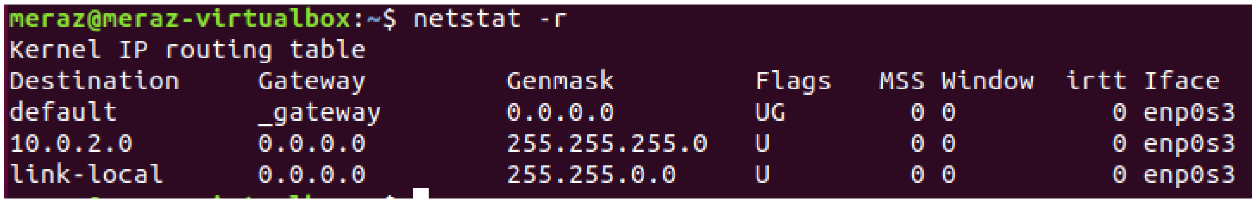
```
meraz@meraz-virtualbox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::80f4:9ad4:e97b:db85  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:56:b6:0e  txqueuelen 1000  (Ethernet)
        RX packets 144400  bytes 151703966 (151.7 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 40751  bytes 2578371 (2.5 MB)
        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 282  bytes 24798 (24.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 282  bytes 24798 (24.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Command :hostname – i



```
meraz@meraz-virtualbox:~$ hostname -i
127.0.1.1
```

2.Routing Table basics:
The route -n command lists the routing table; the -n option displays the results as IP addresses only and does not attempt to perform a DNS lookup which would replace the IP address with hostnames if they are available. The netstat -rn command produces very similar results.

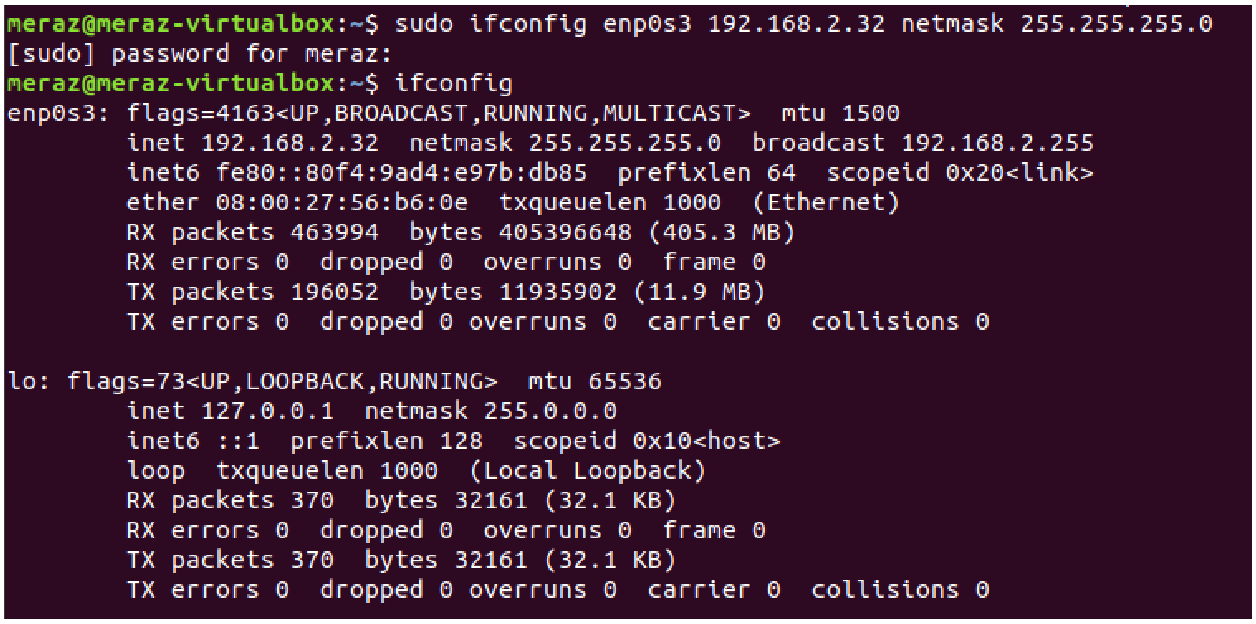


```
meraz@meraz-virtualbox:~$ netstat -r
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
default         _gateway        0.0.0.0         UG        0 0          0 enp0s3
10.0.2.0        0.0.0.0         255.255.255.0   U         0 0          0 enp0s3
link-local      0.0.0.0         255.255.0.0     U         0 0          0 enp0s3
```

3.Virtual Interface:

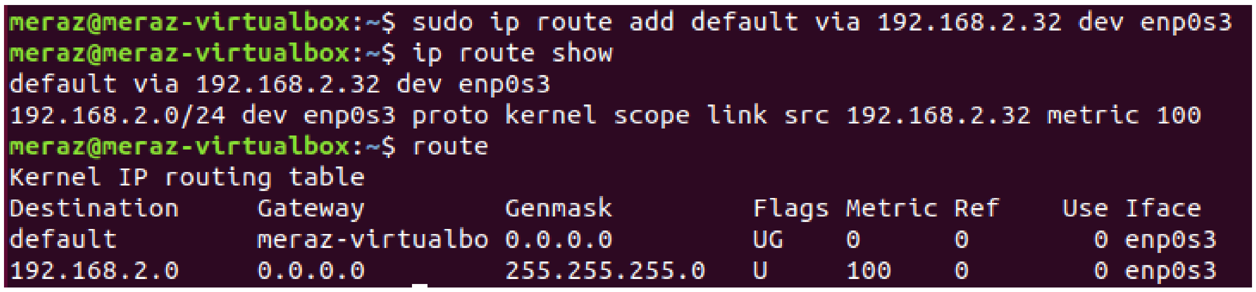
The process of creating a virtual network interface in Linux is a quite simple matter. It involves a single execution of the ifconfig command.

a) Creating a virtual network:



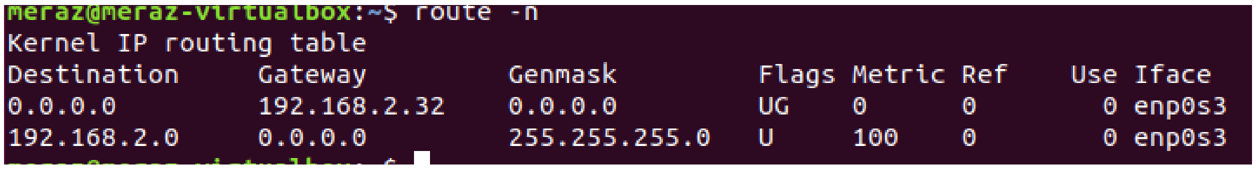
```
meraz@meraz-virtualbox:~$ sudo ifconfig enp0s3 192.168.2.32 netmask 255.255.255.0
[sudo] password for meraz:
meraz@meraz-virtualbox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.2.32  netmask 255.255.255.0  broadcast 192.168.2.255
        inet6 fe80::80f4:9ad4:e97b:db85  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:56:b6:0e  txqueuelen 1000  (Ethernet)
        RX packets 463994  bytes 405396648 (405.3 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 196052  bytes 11935902 (11.9 MB)
        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 370  bytes 32161 (32.1 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 370  bytes 32161 (32.1 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

b) Virtual network routing:



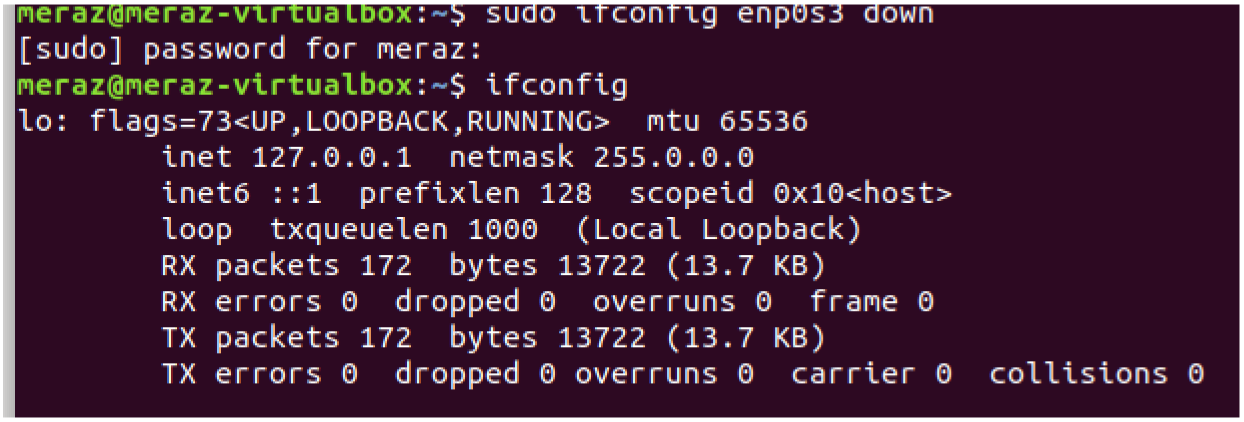
```
meraz@meraz-virtualbox:~$ sudo ip route add default via 192.168.2.32 dev enp0s3
meraz@meraz-virtualbox:~$ ip route show
default via 192.168.2.32 dev enp0s3
192.168.2.0/24 dev enp0s3 proto kernel scope link src 192.168.2.32 metric 100
meraz@meraz-virtualbox:~$ route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
default         meraz-virtualbo 0.0.0.0         UG    0      0        0 enp0s3
192.168.2.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
```

c) Next remove the route for this interface:

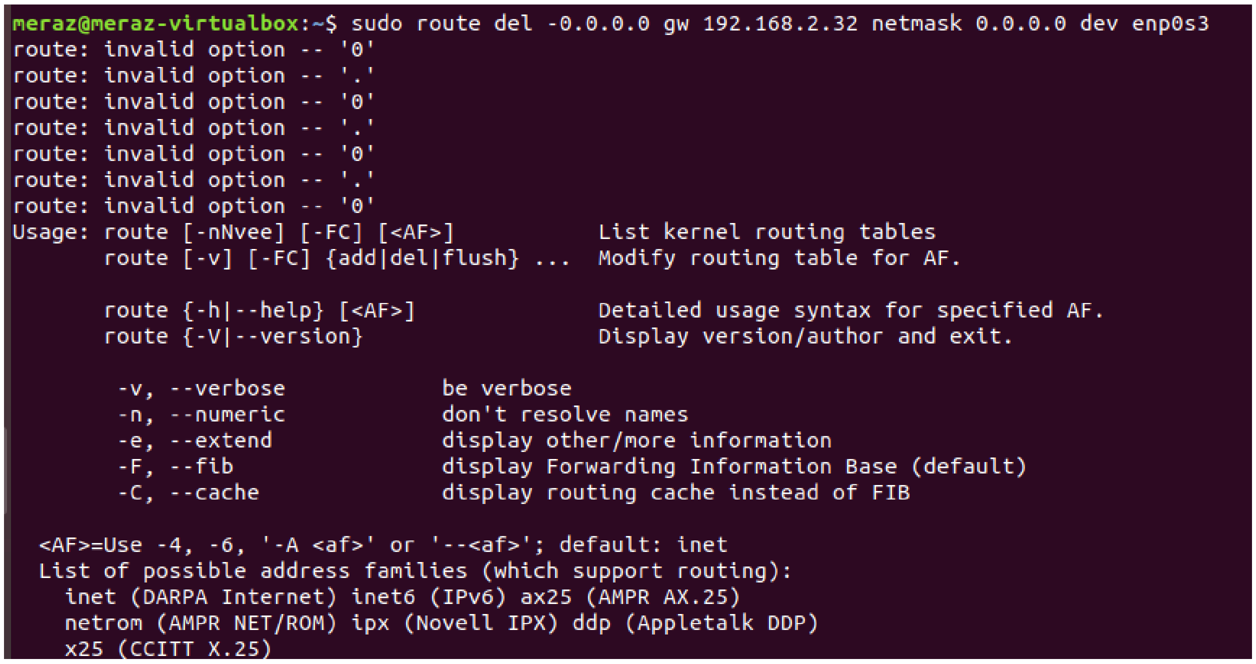


```
meraz@meraz-virtualbox:~$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.2.32    0.0.0.0         UG    0      0        0 enp0s3
192.168.2.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
```

d) Then remove the interface completely.



```
meraz@meraz-virtualbox:~$ sudo ifconfig enp0s3 down
[sudo] password for meraz:
meraz@meraz-virtualbox:~$ ifconfig
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 172  bytes 13722 (13.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 172  bytes 13722 (13.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



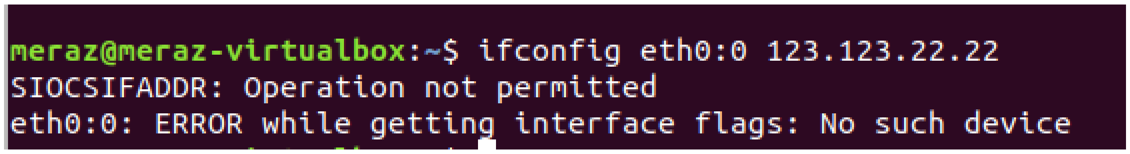
```
meraz@meraz-virtualbox:~$ sudo route del -0.0.0.0 gw 192.168.2.32 netmask 0.0.0.0 dev enp0s3
route: invalid option -- '0'
route: invalid option -- '.'
route: invalid option -- '0'
route: invalid option -- '.'
route: invalid option -- '0'
route: invalid option -- '.'
route: invalid option -- '0'
Usage: route [-nNvee] [-FC] [<AF>]           List kernel routing tables
       route [-v] [-FC] {add|del|flush} ...  Modify routing table for AF.

       route {-h|--help} [<AF>]              Detailed usage syntax for specified AF.
       route {-V|--version}                  Display version/author and exit.

        -v, --verbose            be verbose
        -n, --numeric            don't resolve names
        -e, --extend             display other/more information
        -F, --fib                display Forwarding Information Base (default)
        -C, --cache              display routing cache instead of FIB

  <AF>=Use -4, -6, '-A <af>' or '--<af>'; default: inet
  List of possible address families (which support routing):
    inet (DARPA Internet) inet6 (IPv6) ax25 (AMPR AX.25)
    netrom (AMPR NET/ROM) ipx (Novell IPX) ddp (Appletalk DDP)
    x25 (CCITT X.25)
```

4. Add a New Network

The process of creating a virtual network interface in Linux is a quite simple matter. It involves a single execution of the ifconfig command.

```
ifconfig eth0:0 123.123.22.22
```

The above command will create a new virtual network interface based on original eth0 physical interface. The only most important condition for creating the virtual network interface is the physical network interface, as in our case eth0 must exists. The whole example is shown below:



```
meraz@meraz-virtualbox:~$ ifconfig eth0:0 123.123.22.22
SIOCSIFADDR: Operation not permitted
eth0:0: ERROR while getting interface flags: No such device
```

There was a problem in my linux platform .There was no such device while I was getting interface flags.So, the operation was not permitted.

5. Multi network scenario configuration:

This is generally to improve the high availability of the network. When a network card fails, the second fastest network card is used. Although it sounds exaggerated, routers like Cisco will also be equipped with backup power or CPU (not the CPU of our computer, but the router)

The first step is to add SLAVE = yes to the two network card configuration files to turn it into a slave. Then set MASTER = bond0 and tell it that your master is bond0. But note here that each network card must set BOOTPROTO = none. Example: DEVICE = eth0 ONBOOT = yes BOOTPROTO = none MASTER = bond0 SLAVE = yes 2. Create their master bond0 (ifcfg-bond0). Example: DEVICE = bond0 BOOTPROTO = static IPADDR = 10.1.3.210 NETMASK = 255.255.255.0 GATEWAY = 10.1.3.254 ONBOOT = yes 3. Modify/etc/modprobe.d/dist. Add the following content to the conf: alias bond0 bonding options bond0 miimon = 100 mode = 1 mode = 0: indicates that load balancing (round-robin) is a load balancing method, and both network cards are working. mode = 1: indicates that fault-tolerance (active-backup) provides redundancy. The working mode is

the active and standby working mode. One of the network cards is working (if eth0 is broken), it will automatically switch to another network card (eth1 does Backup). Finally, service network restart to verify it.