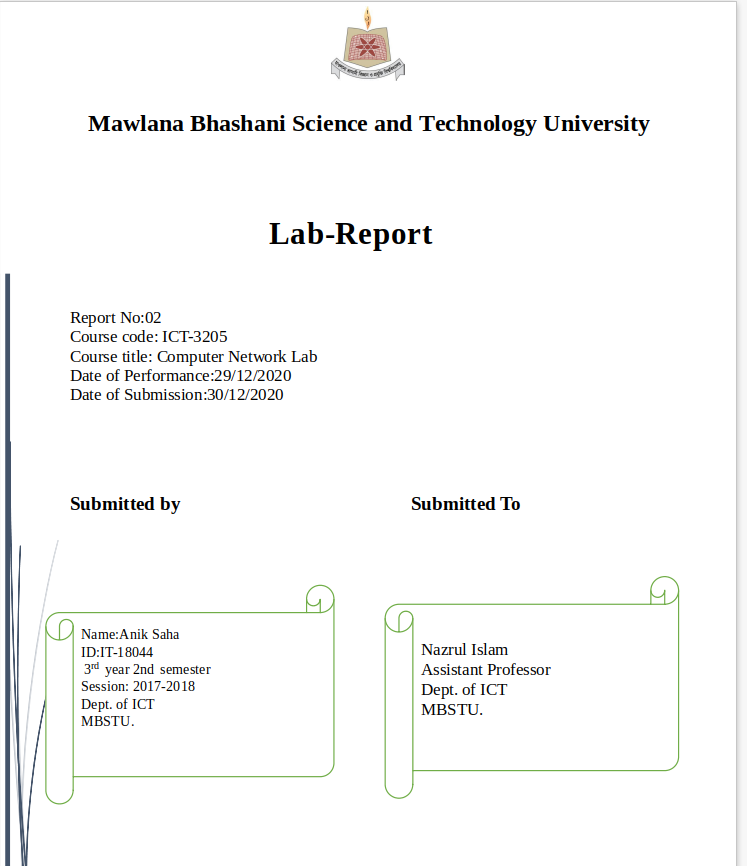
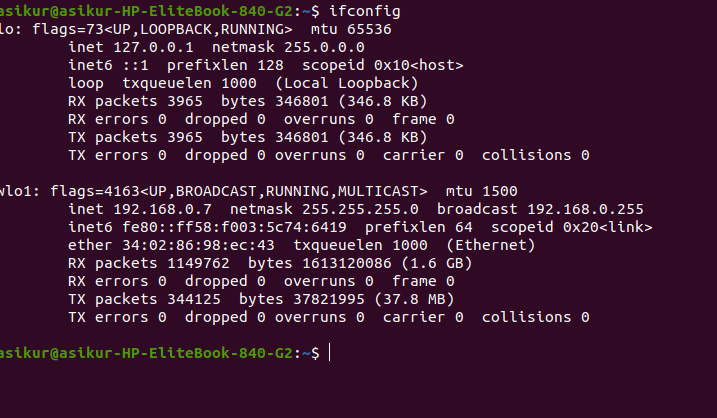
****

**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.**

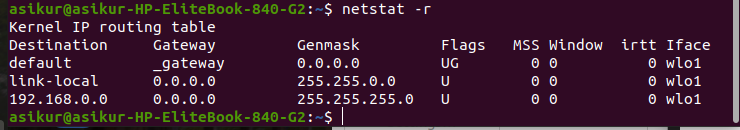
****

**Command :hostname - i**

****

**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.**

****

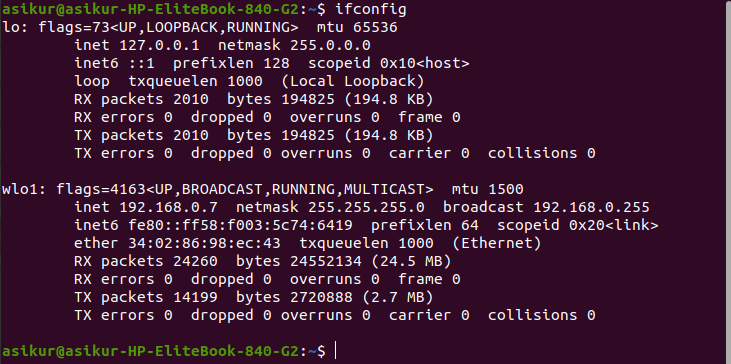
**3.Virtual Interface:**

**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**

****

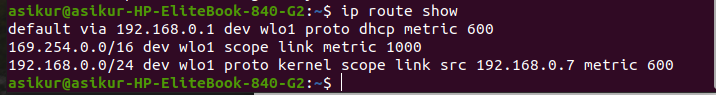
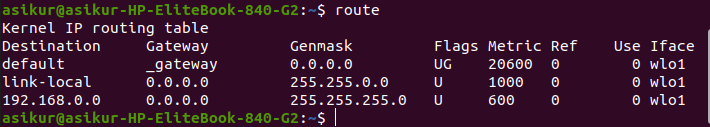
****

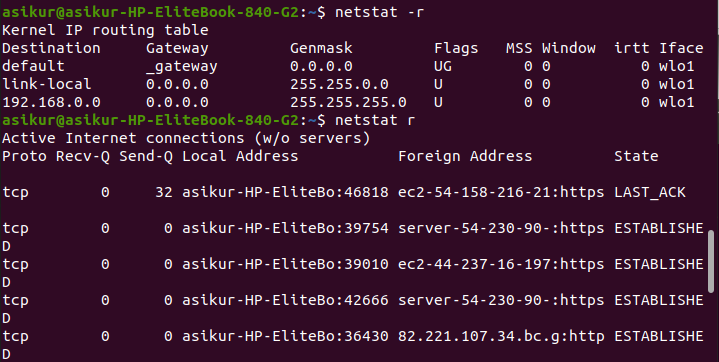
****

**b) Virtual network routing**

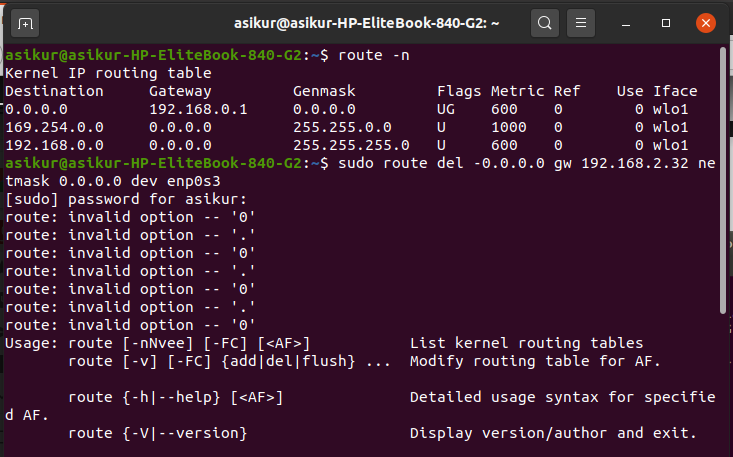
****

****

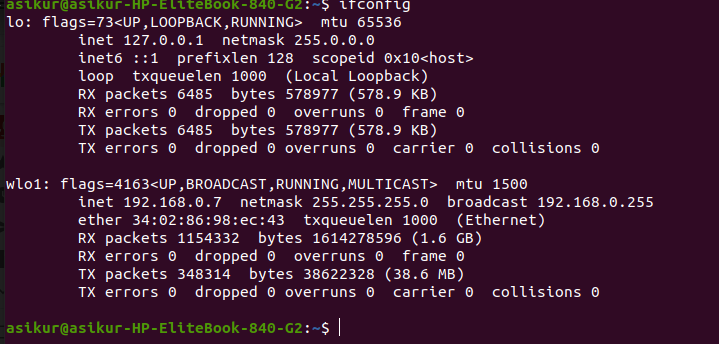
****

****

**c) Next remove the route for this interface**



**d) Then remove the interface completely.**

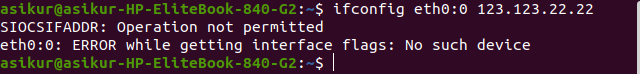
****

**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:

****

**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.**