Lab-02B and Lab-03

Name: Harsh Gajjar

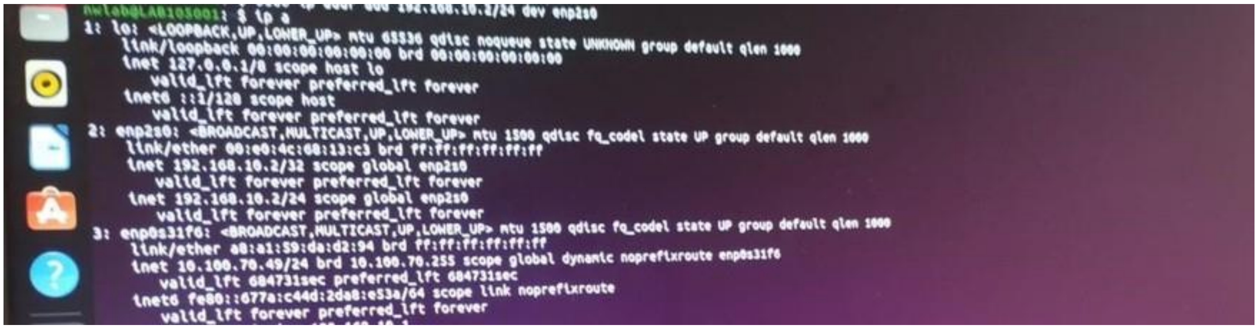
202201140

Connect Both PCs on Same Network

Procedure:

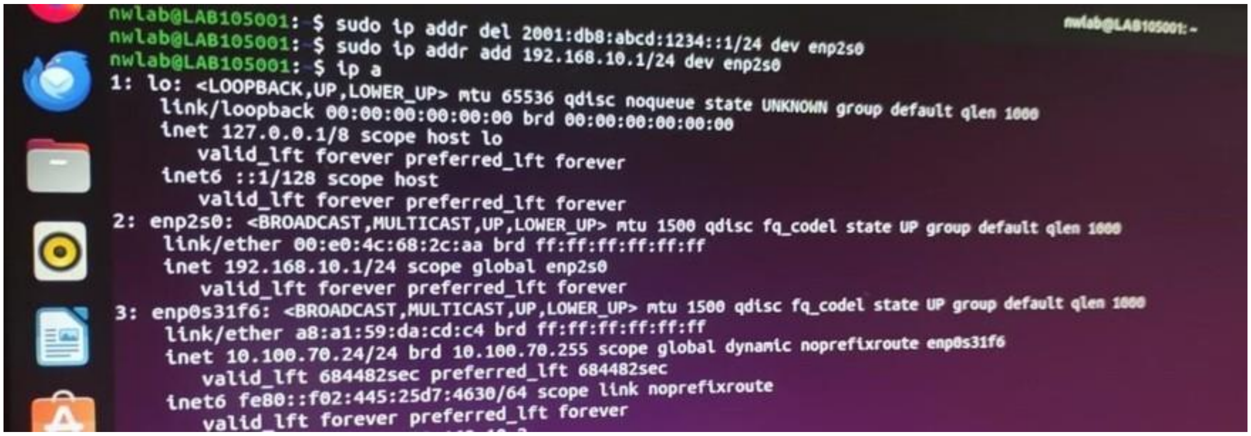
Step-01)Begin by physically connecting both PCs to the switch using straight-through Ethernet cables. Insert one end of each cable into the network port on the PC, and connect the other end to an available port on the switch.

Step-02) Next, open the terminal on both PCs and enter the ip a command.



```
                              192.168.10.2/24 dev enp2s0
nwlab@LAB105001: $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:13:c3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.2/32 scope global enp2s0
       valid_lft forever preferred_lft forever
    inet 192.168.10.2/24 scope global enp2s0
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:d2:94 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.49/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 684731sec preferred_lft 684731sec
    inet6 fe80::677a:c44d:2da8:e53a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

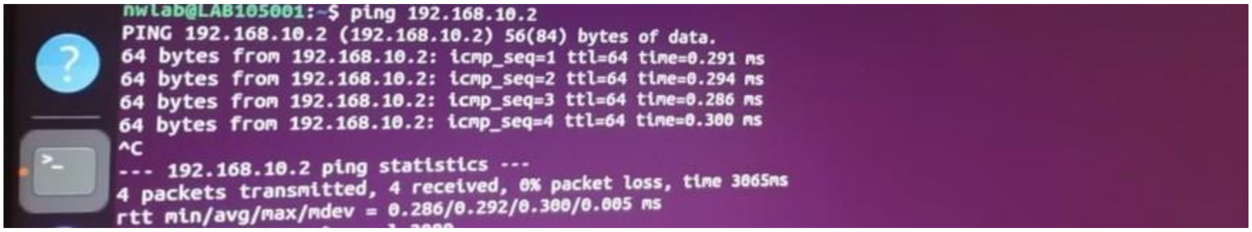
Step-03)The 'ip a' command is used to display all the network interfaces and their associated IP addresses. It provides detailed information about each interface, including the IP addresses, the state of the interface (up or down), the MAC address, and more.



```
nwlab@LAB105001:~$ sudo ip addr del 2001:db8:abcd:1234::1/24 dev enp2s0
nwlab@LAB105001:~$ sudo ip addr add 192.168.10.1/24 dev enp2s0
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:2c:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 scope global enp2s0
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:cd:c4 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.24/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 684482sec preferred_lft 684482sec
    inet6 fe80::f02:445:25d7:4630/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

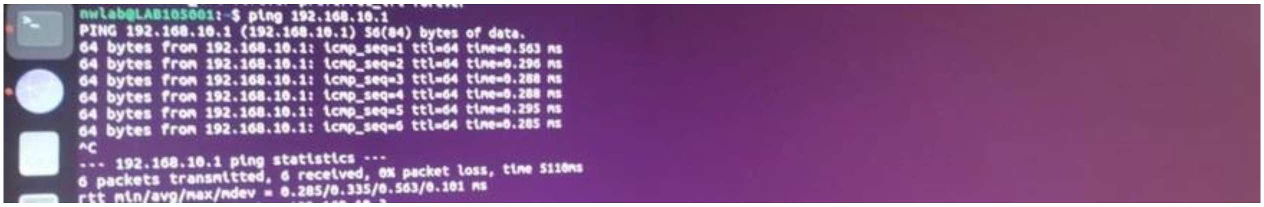
Step-04)Use the command sudo ip addr add 192.168.10.1/24 dev {ethernet name} to assign a static IPv4 address to a particular network interface, such as an Ethernet interface. For this temporary scenario, we used enp2s0. Then, run the ip a command again to verify the assigned IPv4 address.

Step-05) Apply similar changes to the other connected PC, but this time, change the IP address to 192.168.10.2/24 to ensure a connection between the two PCs.



```
nwlab@LAB105001:~$ ping 192.168.10.2
PING 192.168.10.2 (192.168.10.2) 56(84) bytes of data.
64 bytes from 192.168.10.2: icmp_seq=1 ttl=64 time=0.291 ms
64 bytes from 192.168.10.2: icmp_seq=2 ttl=64 time=0.294 ms
64 bytes from 192.168.10.2: icmp_seq=3 ttl=64 time=0.286 ms
64 bytes from 192.168.10.2: icmp_seq=4 ttl=64 time=0.300 ms
^C
--- 192.168.10.2 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3065ms
rtt min/avg/max/mdev = 0.286/0.292/0.300/0.005 ms
```

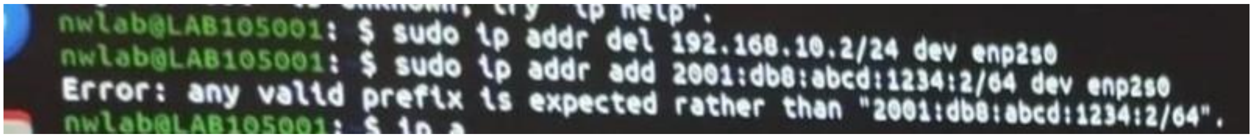
Step-06) Use the ping command to test network connectivity between the devices. By running ping IP_ADDRESS_OF_PC2, ICMP echo request packets are sent to the specified IP address (in this case, the IP address of PC2) and await a response. This allows you to check if the devices can communicate over the network. If the PCs are not connected, a "Request timeout" or "Destination Host Unreachable" error message will appear.



```
nwlab@LAB105001:~$ ping 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.563 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.296 ms
64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.288 ms
64 bytes from 192.168.10.1: icmp_seq=4 ttl=64 time=0.288 ms
64 bytes from 192.168.10.1: icmp_seq=5 ttl=64 time=0.295 ms
64 bytes from 192.168.10.1: icmp_seq=6 ttl=64 time=0.285 ms
^C
--- 192.168.10.1 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5110ms
rtt min/avg/max/mdev = 0.285/0.335/0.563/0.101 ms
```

Step- 07) The “ping” command for PC2.

- → Now, perform steps 4 to 7 using the IPv6 address. However, you must first delete the existing IPv4 address.

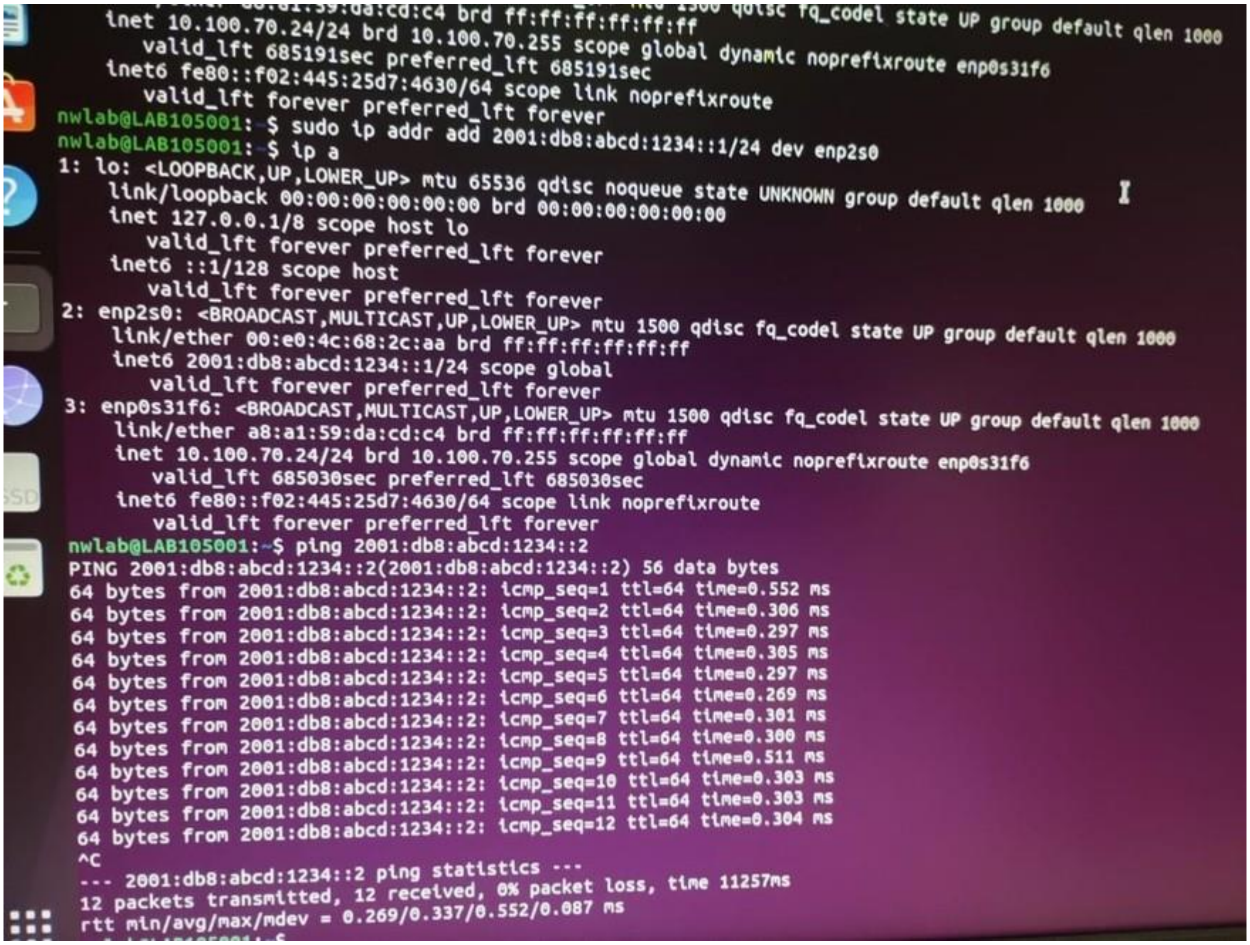


```
try "ip help".
nwlab@LAB105001: $ sudo ip addr del 192.168.10.2/24 dev enp2s0
nwlab@LAB105001: $ sudo ip addr add 2001:db8:abcd:1234:2/64 dev enp2s0
Error: any valid prefix is expected rather than "2001:db8:abcd:1234:2/64".
nwlab@LAB105001: $ ip a
```

Step-08: The command sudo ip addr del 192.168.1.50/24 dev {ethernet name} is used to remove a specific IP address from a network interface in Linux.

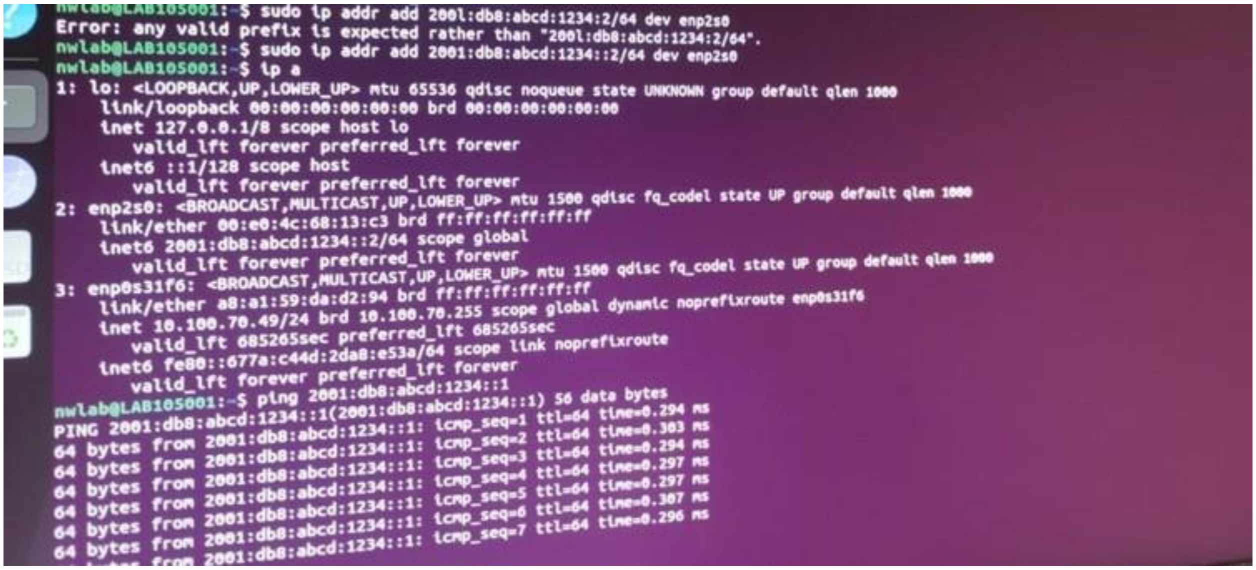
Lab-02B and Lab-03

For PC1:



```
    link/ether a8:a1:59:da:cd:c4 brd ff:ff:ff:ff:ff:ff     qdisc fq_codel state UP group default qlen 1000
    inet 10.100.70.24/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 685191sec preferred_lft 685191sec
    inet6 fe80::f02:445:25d7:4630/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
nwlab@LAB105001:~$ sudo ip addr add 2001:db8:abcd:1234::1/24 dev enp2s0
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:2c:aa brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:abcd:1234::1/24 scope global
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:cd:c4 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.24/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 685030sec preferred_lft 685030sec
    inet6 fe80::f02:445:25d7:4630/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
nwlab@LAB105001:~$ ping 2001:db8:abcd:1234::2
PING 2001:db8:abcd:1234::2(2001:db8:abcd:1234::2) 56 data bytes
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=1 ttl=64 time=0.552 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=2 ttl=64 time=0.306 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=3 ttl=64 time=0.297 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=4 ttl=64 time=0.305 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=5 ttl=64 time=0.297 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=6 ttl=64 time=0.269 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=7 ttl=64 time=0.301 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=8 ttl=64 time=0.300 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=9 ttl=64 time=0.511 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=10 ttl=64 time=0.303 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=11 ttl=64 time=0.303 ms
64 bytes from 2001:db8:abcd:1234::2: icmp_seq=12 ttl=64 time=0.304 ms
^C
--- 2001:db8:abcd:1234::2 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11257ms
rtt min/avg/max/mdev = 0.269/0.337/0.552/0.087 ms
nwlab@LAB105001:~$
```

For PC2:



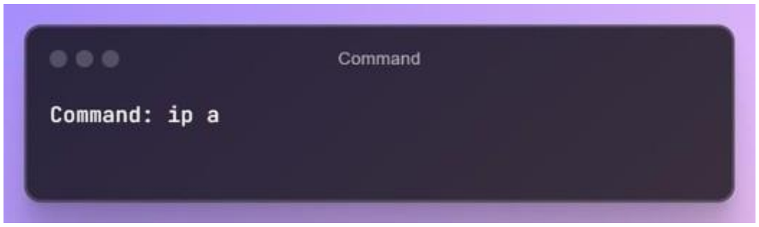
```
nwlab@LAB105001:~$ sudo ip addr add 200l:db8:abcd:1234:2/64 dev enp2s0
Error: any valid prefix is expected rather than "200l:db8:abcd:1234:2/64".
nwlab@LAB105001:~$ sudo ip addr add 2001:db8:abcd:1234::2/64 dev enp2s0
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:13:c3 brd ff:ff:ff:ff:ff:ff
    inet6 2001:db8:abcd:1234::2/64 scope global
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:d2:94 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.49/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 685265sec preferred_lft 685265sec
    inet6 fe80::677a:c44d:2da8:e53a/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
nwlab@LAB105001:~$ ping 2001:db8:abcd:1234::1
PING 2001:db8:abcd:1234::1(2001:db8:abcd:1234::1) 56 data bytes
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=1 ttl=64 time=0.294 ms
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=2 ttl=64 time=0.303 ms
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=3 ttl=64 time=0.294 ms
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=4 ttl=64 time=0.297 ms
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=5 ttl=64 time=0.297 ms
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=6 ttl=64 time=0.307 ms
64 bytes from 2001:db8:abcd:1234::1: icmp_seq=7 ttl=64 time=0.296 ms
64 bytes from 2001:db8:abcd:1234::1:
```

Lab-02B and Lab-03

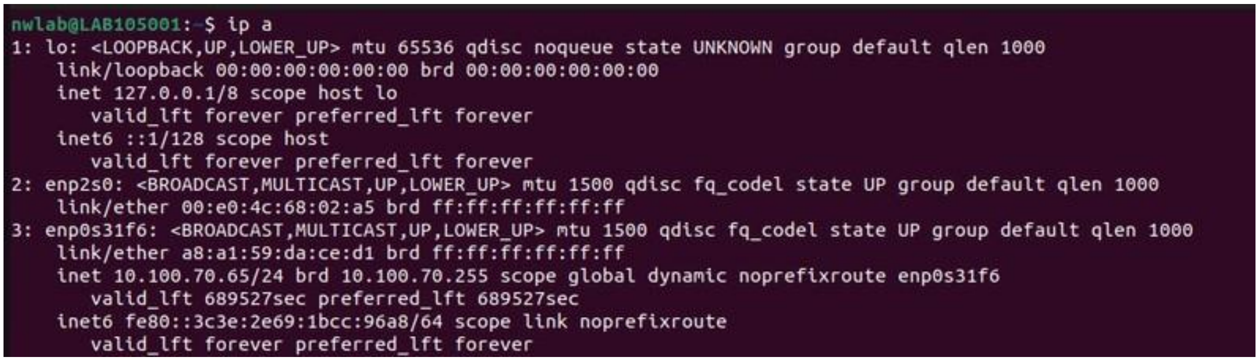
Setting the IP Address of PC (Permanent Change) using NetPlan

Procedure:

Step-01)Identify the current Network Interface



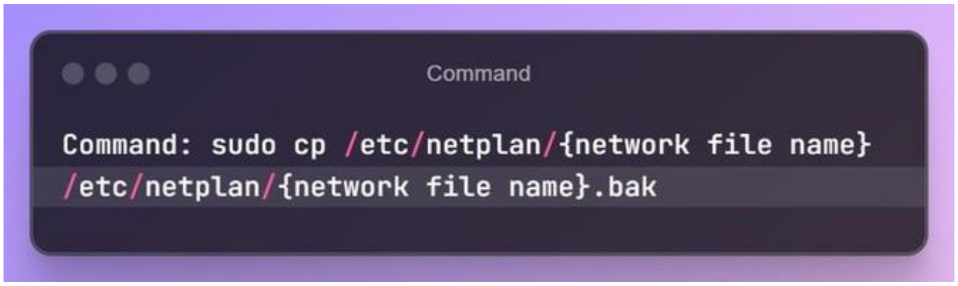
```
Command: ip a
```



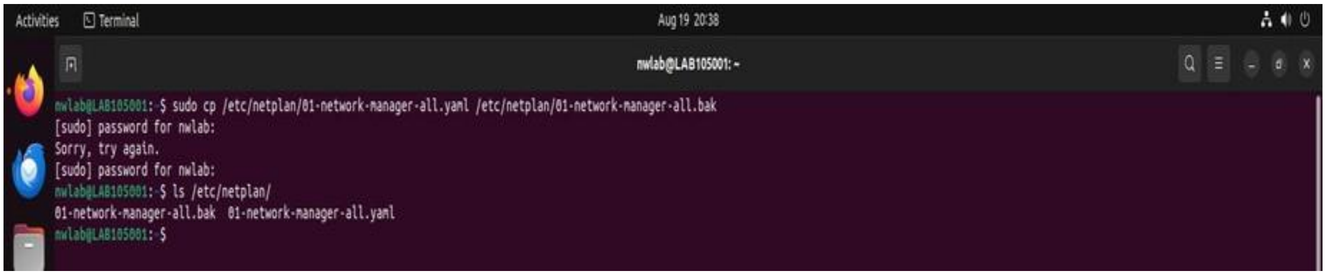
```
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:02:a5 brd ff:ff:ff:ff:ff:ff
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.65/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 689527sec preferred_lft 689527sec
    inet6 fe80::3c3e:2e69:1bcc:96a8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Step-02)Create a Backup of the current Netplan Configuration file.

- ➔ This step involves creating a backup of a network configuration file managed by Netplan, a utility used for configuring network settings on Ubuntu and other Linux distributions.
- ➔ It is best practice to create a backup of the network configuration file before making any changes. This allows you to restore the original settings if something goes wrong during the configuration process.



```
Command: sudo cp /etc/netplan/{network file name}
/etc/netplan/{network file name}.bak
```

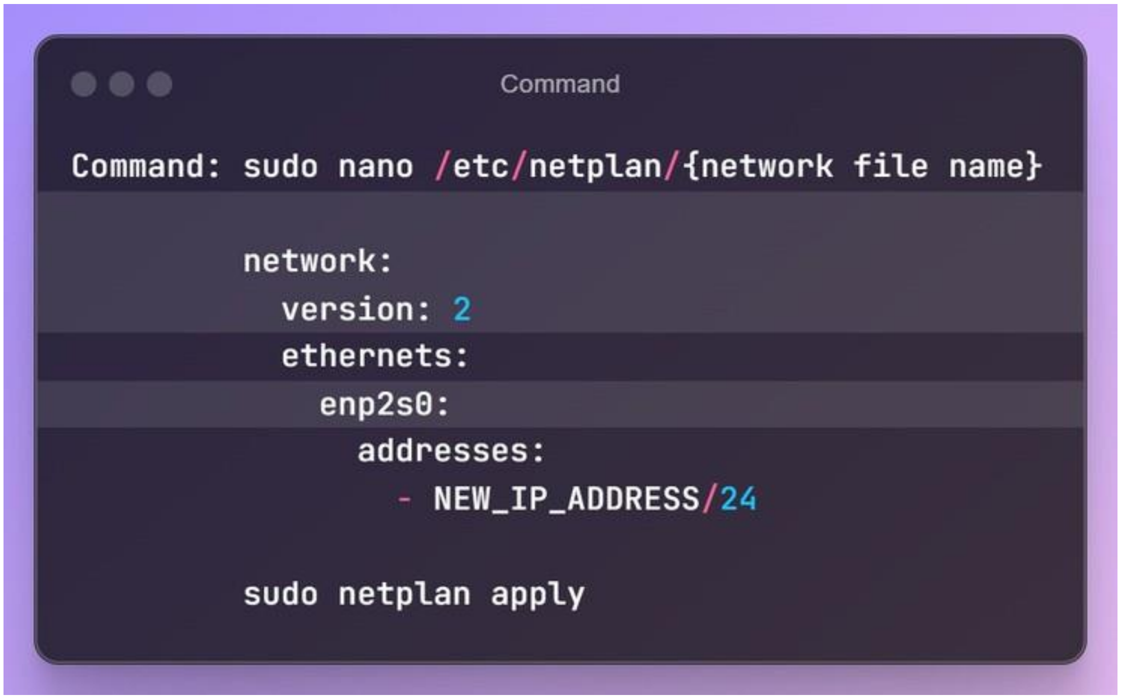


```
nwlab@LAB105001:~$ sudo cp /etc/netplan/01-network-manager-all.yaml /etc/netplan/01-network-manager-all.bak
[sudo] password for nwlab:
Sorry, try again.
[sudo] password for nwlab:
nwlab@LAB105001:~$ ls /etc/netplan/
01-network-manager-all.bak  01-network-manager-all.yaml
nwlab@LAB105001:~$
```

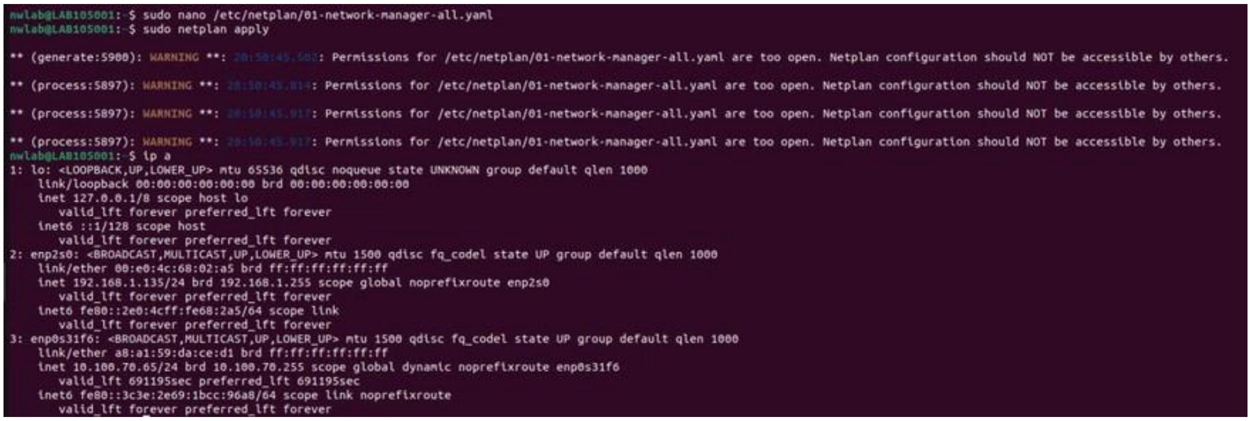
Step-03) Edit the Netplan configuration file.

Lab-02B and Lab-03

➔ To set up the network settings on your system, you need to modify the relevant Netplan configuration file. This can be done using the nano text editor.



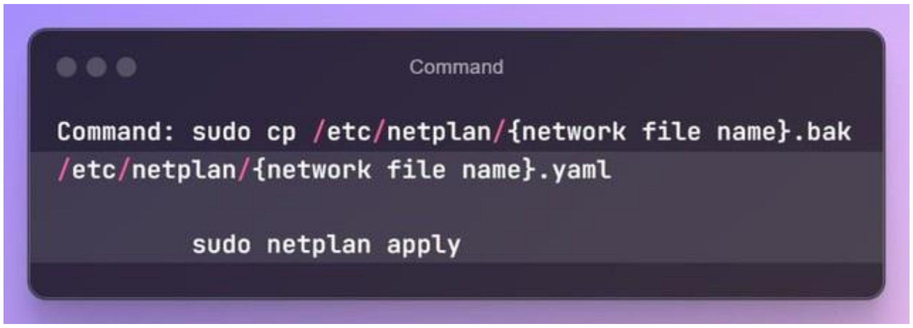
```
Command: sudo nano /etc/netplan/{network file name}

          network:
            version: 2
            ethernets:
              enp2s0:
                addresses:
                  - NEW_IP_ADDRESS/24

          sudo netplan apply
```



```
nwlab@LAB105001:~$ sudo nano /etc/netplan/01-network-manager-all.yaml
nwlab@LAB105001:~$ sudo netplan apply

** (generate:5900): WARNING **: 20:50:45.502: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:5897): WARNING **: 20:50:45.814: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:5897): WARNING **: 20:50:45.917: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:5897): WARNING **: 20:50:45.917: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:02:a5 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.135/24 brd 192.168.1.255 scope global noprefixroute enp2s0
       valid_lft forever preferred_lft forever
    inet6 fe80::2e0:4cff:fe68:2a5/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.65/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 691195sec preferred_lft 691195sec
    inet6 fe80::3c3e:2e69:1bcc:96a8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

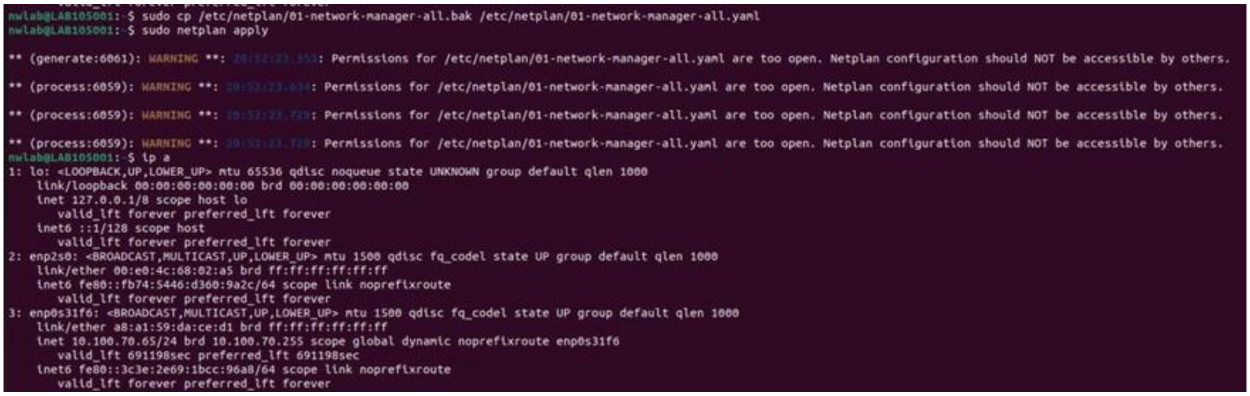
Step-04)Revert to the original settings.

➔ If you want to revert to the original configuration file, load the backup file into the original one and apply the changes.



```
Command: sudo cp /etc/netplan/{network file name}.bak
/etc/netplan/{network file name}.yaml

         sudo netplan apply
```

Lab-02B and Lab-03



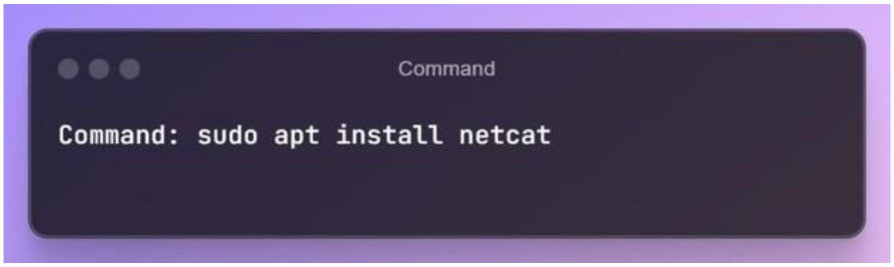
```
      valid_lft forever preferred_lft forever
nwlab@LAB105001:~$ sudo cp /etc/netplan/01-network-manager-all.bak /etc/netplan/01-network-manager-all.yaml
nwlab@LAB105001:~$ sudo netplan apply

** (generate:6061): WARNING **: 20:52:23.353: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:6059): WARNING **: 20:52:23.634: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:6059): WARNING **: 20:52:23.720: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.

** (process:6059): WARNING **: 20:52:23.720: Permissions for /etc/netplan/01-network-manager-all.yaml are too open. Netplan configuration should NOT be accessible by others.
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:02:a5 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::fb74:5446:d360:9a2c/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:d1 brd ff:ff:ff:ff:ff:ff
    inet 10.100.70.65/24 brd 10.100.70.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 691198sec preferred_lft 691198sec
    inet6 fe80::3c3e:2e69:1bcc:96a8/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Sending Messages from One Machine to Another Using Netcat

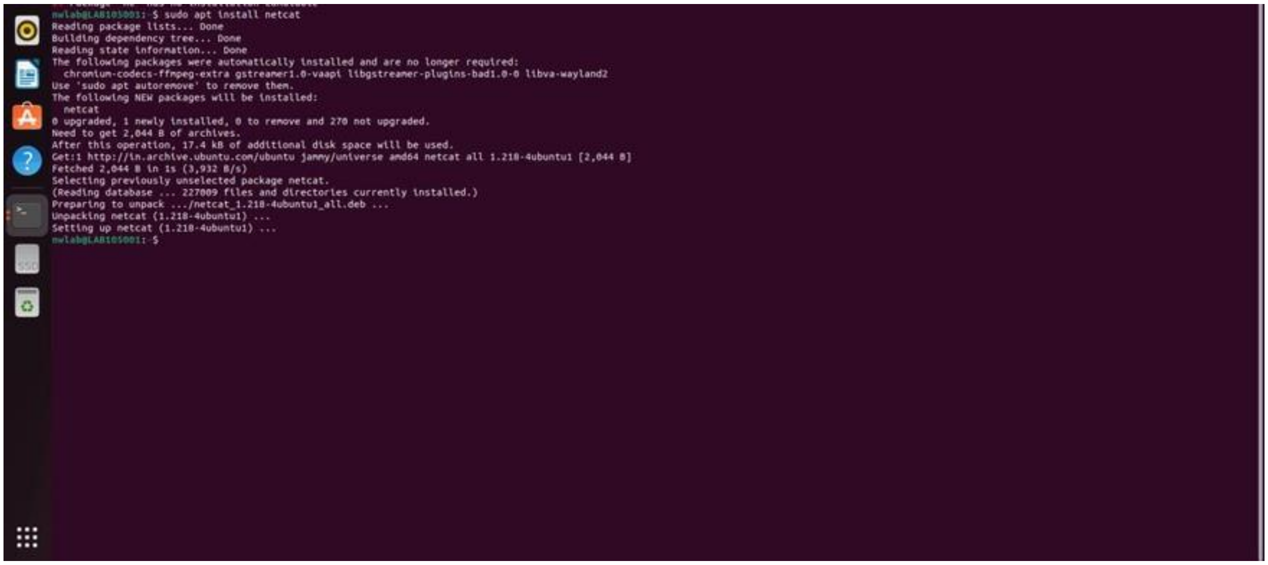
Lab-02B and Lab-03

Procedure:

Step-01) First, connect two PCs within the same LAN and install Netcat.



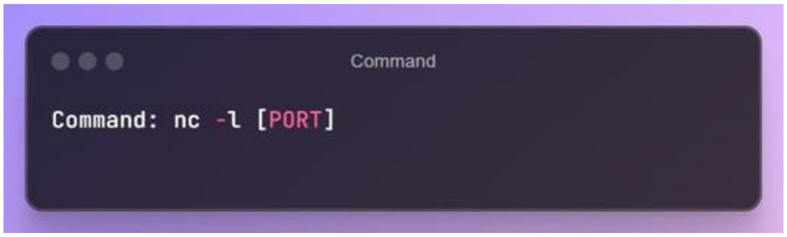
```
Command: sudo apt install netcat
```



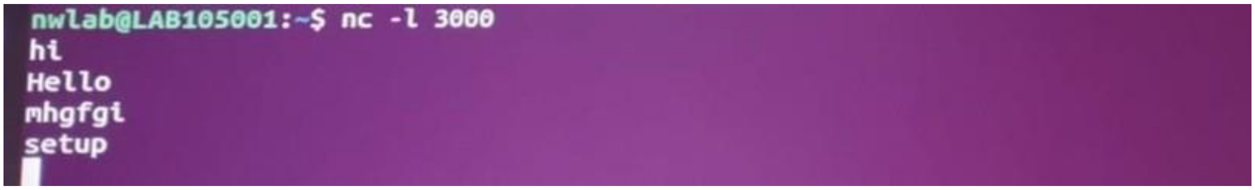
```
E: Package 'nc' has no installation candidate
nwlab@LAB105001:~$ sudo apt install netcat
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  netcat
0 upgraded, 1 newly installed, 0 to remove and 270 not upgraded.
Need to get 2,044 B of archives.
After this operation, 17.4 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 netcat all 1.218-4ubuntu1 [2,044 B]
Fetched 2,044 B in 1s (3,932 B/s)
Selecting previously unselected package netcat.
(Reading database ... 227009 files and directories currently installed.)
Preparing to unpack .../netcat_1.218-4ubuntu1_all.deb ...
Unpacking netcat (1.218-4ubuntu1) ...
Setting up netcat (1.218-4ubuntu1) ...
nwlab@LAB105001:~$
```

Step-02) Set up the listening PC (Receiver):

- Open the terminal on PC1, which will receive the messages.
- Start the Netcat command in listening mode on the receiver PC.



```
Command: nc -l [PORT]
```



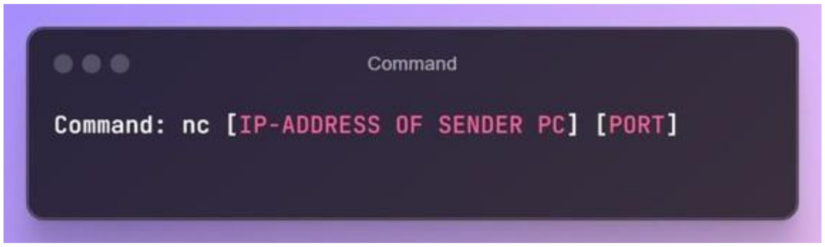
```
nwlab@LAB105001:~$ nc -l 3000
hi
Hello
mhgfgi
setup
```

Lab-02B and Lab-03

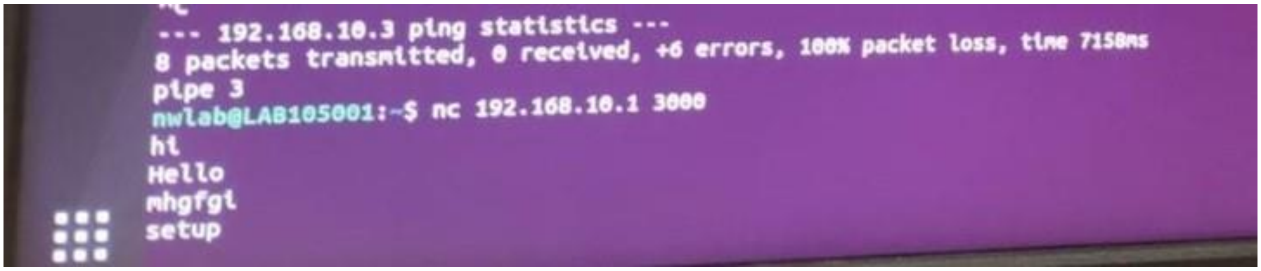
- The command nc -l 3000 initiates Netcat in listening mode on port 3000, meaning it will wait for incoming connections on that port. Once connected, data can be sent and received.

Step-03)Set up the sending PC (Sender):

- Open the terminal on PC2, which will send the messages to the receiver PC.
- Start the Netcat command on the sender PC.



```
Command: nc [IP-ADDRESS OF SENDER PC] [PORT]
```



```
^C
--- 192.168.10.3 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7158ms
pipe 3
nwlab@LAB105001:~$ nc 192.168.10.1 3000
hi
Hello
mhgfgt
setup
```

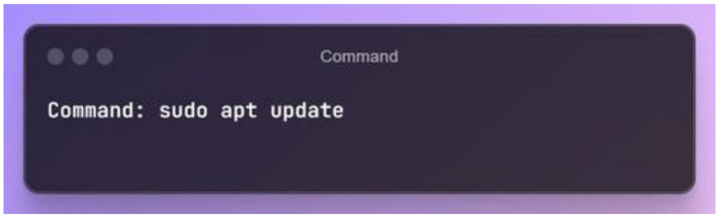
- The command nc <IP_ADDRESS_OF_PC1> 3000 connects to the IP address of the receiver PC (PC1) on port 3000 using Netcat.
- Now, both PCs are connected, and messages can be sent and received between them.

Transferring a File from Server to Client using FTP

Lab-02B and Lab-03

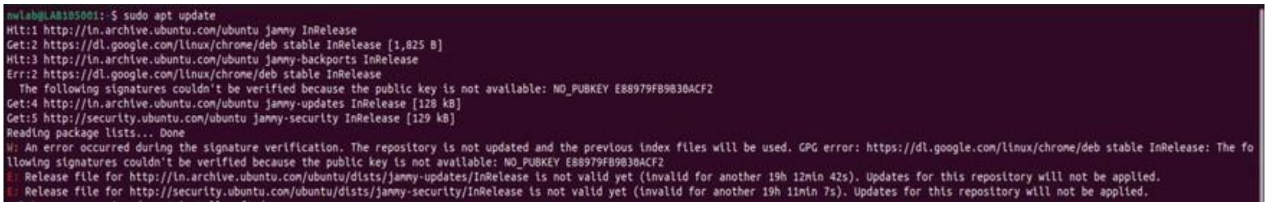
Procedure:

Step-01)Begin by updating all the packages on the computer.



```
Command: sudo apt update
```

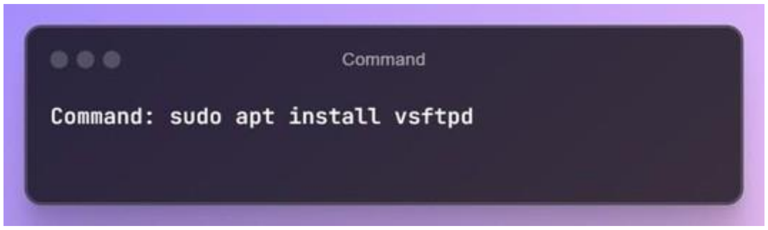
→ This command lists available packages and their versions, ensuring that you have the latest information before installing or upgrading software.



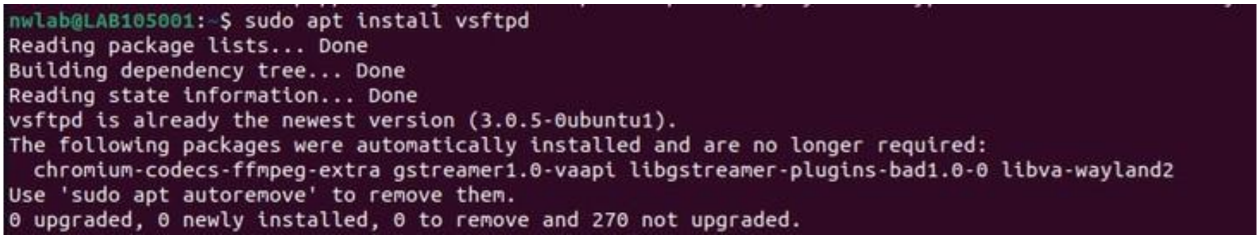
```
nwlab@LAB105001:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Err:2 https://dl.google.com/linux/chrome/deb stable InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY E88979FB9B30ACF2
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Reading package lists... Done
W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: https://dl.google.com/linux/chrome/deb stable InRelease: The fo
llowing signatures couldn't be verified because the public key is not available: NO_PUBKEY E88979FB9B30ACF2
E: Release file for http://in.archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease is not valid yet (invalid for another 19h 12min 42s). Updates for this repository will not be applied.
E: Release file for http://security.ubuntu.com/ubuntu/dists/jammy-security/InRelease is not valid yet (invalid for another 19h 11min 7s). Updates for this repository will not be applied.
```

Step-02)Install the vsftpd library.

→ vsftpd (Very Secure FTP Daemon) is a widely used FTP server software for Unix-based systems like Linux. It is known for its security, stability, and performance. Many Linux distributions use vsftpd as their default FTP server due to its strong security features.



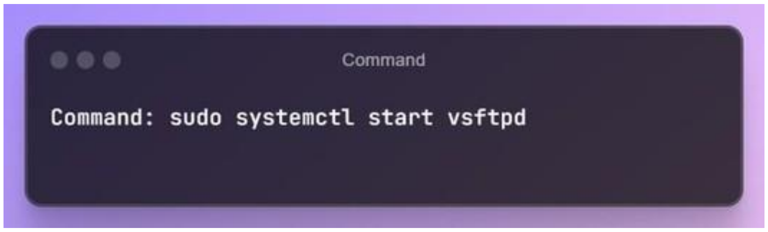
```
Command: sudo apt install vsftpd
```



```
nwlab@LAB105001:~$ sudo apt install vsftpd
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
vsftpd is already the newest version (3.0.5-0ubuntu1).
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 270 not upgraded.
```

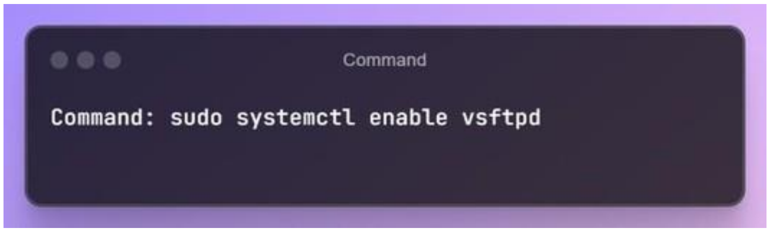
Step-03)Start the vsftpd service.

Lab-02B and Lab-03



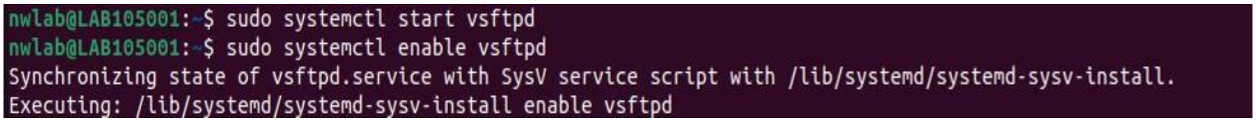
```
Command: sudo systemctl start vsftpd
```

➔ After configuring vsftpd, use this command to start the FTP server so it begins accepting connections.



```
Command: sudo systemctl enable vsftpd
```

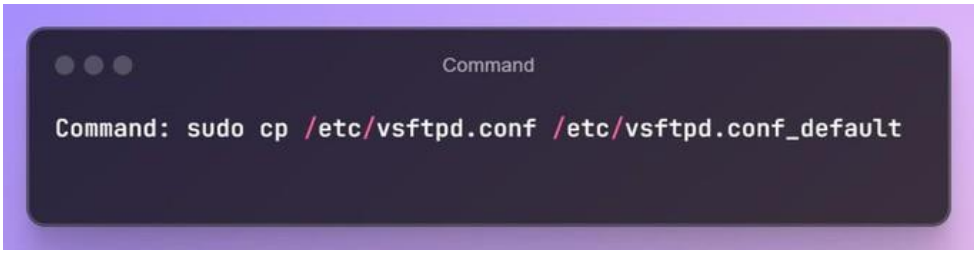
➔ enabling the vsftpd service after starting it ensures the FTP server will automatically start every time the system reboots.



```
nwlab@LAB105001:~$ sudo systemctl start vsftpd
nwlab@LAB105001:~$ sudo systemctl enable vsftpd
Synchronizing state of vsftpd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable vsftpd
```

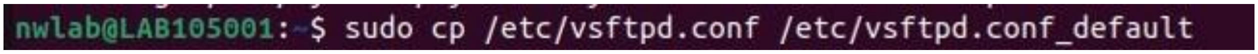
Step-04) Take a backup of the configuration file.

➔ Before modifying the vsftpd configuration files, it is good practice to create a backup of the original settings so you can easily revert to the original configuration in case of any errors.



```
Command: sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
```

➔ The command copies the current configuration file, /etc/vsftpd.conf, to a new file named /etc/vsftpd.conf_default. This backup allows you to restore the default settings if needed.

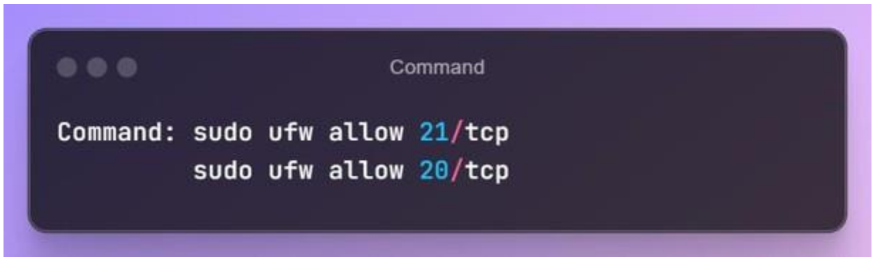


```
nwlab@LAB105001:~$ sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default
```

Step-05) Configure the firewall settings to allow traffic while transferring the files.

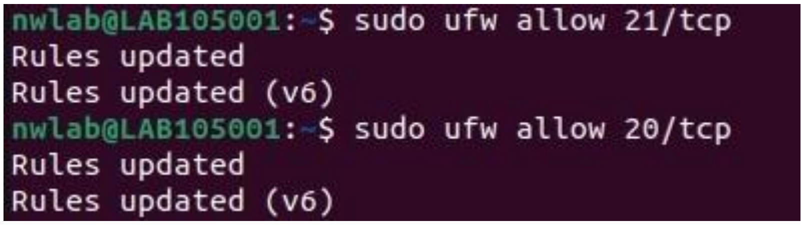
Lab-02B and Lab-03

- ➔ To ensure proper communication between your FTP server and clients, you need to configure the firewall to permit FTP traffic. This requires opening TCP ports 21 and 20, which are used for FTP control and data transfer, respectively.
- ➔ Failing to execute this step could result in the firewall blocking file transfers due to high traffic.



```
Command: sudo ufw allow 21/tcp
         sudo ufw allow 20/tcp
```

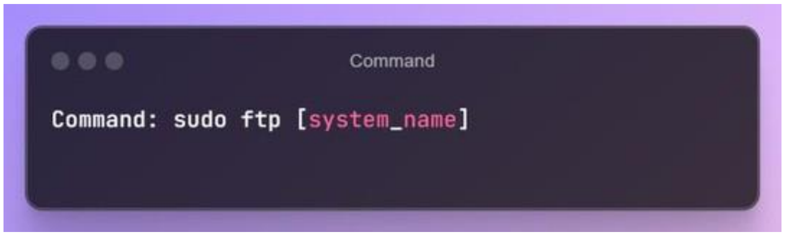
- ➔ "ufw" stands for "Uncomplicated Firewall," which is a tool for managing firewall rules on Ubuntu and other Debian-based distributions.
- ➔ TCP port 21 is utilized for FTP control commands (e.g., initiating connections).
- ➔ TCP port 20 is used for data transfer in active FTP mode.



```
nwlab@LAB105001:~$ sudo ufw allow 21/tcp
Rules updated
Rules updated (v6)
nwlab@LAB105001:~$ sudo ufw allow 20/tcp
Rules updated
Rules updated (v6)
```

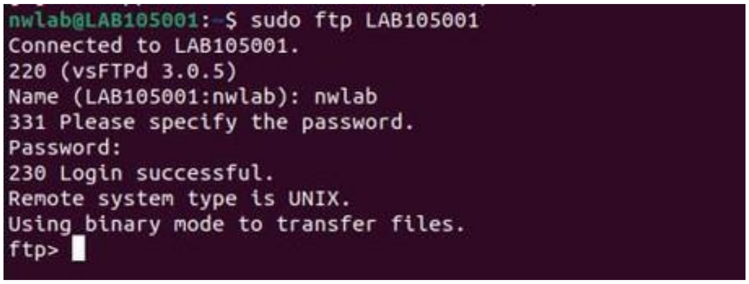
Step-06) Connect to the FTP Server.

- ➔ Initiate an FTP session from the client machine to the FTP server.



```
Command: sudo ftp [system_name]
```

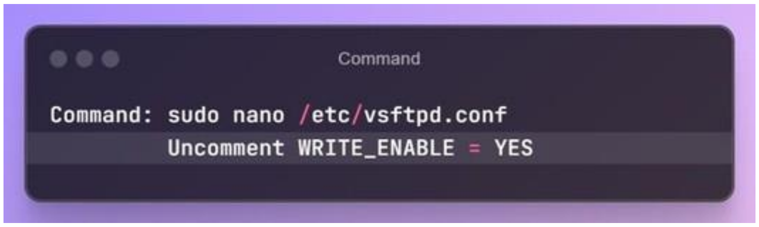
- ➔ Enter the system name of the server you wish to connect to.



```
nwlab@LAB105001:~$ sudo ftp LAB105001
Connected to LAB105001.
220 (vsFTPd 3.0.5)
Name (LAB105001:nwlab): nwlab
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

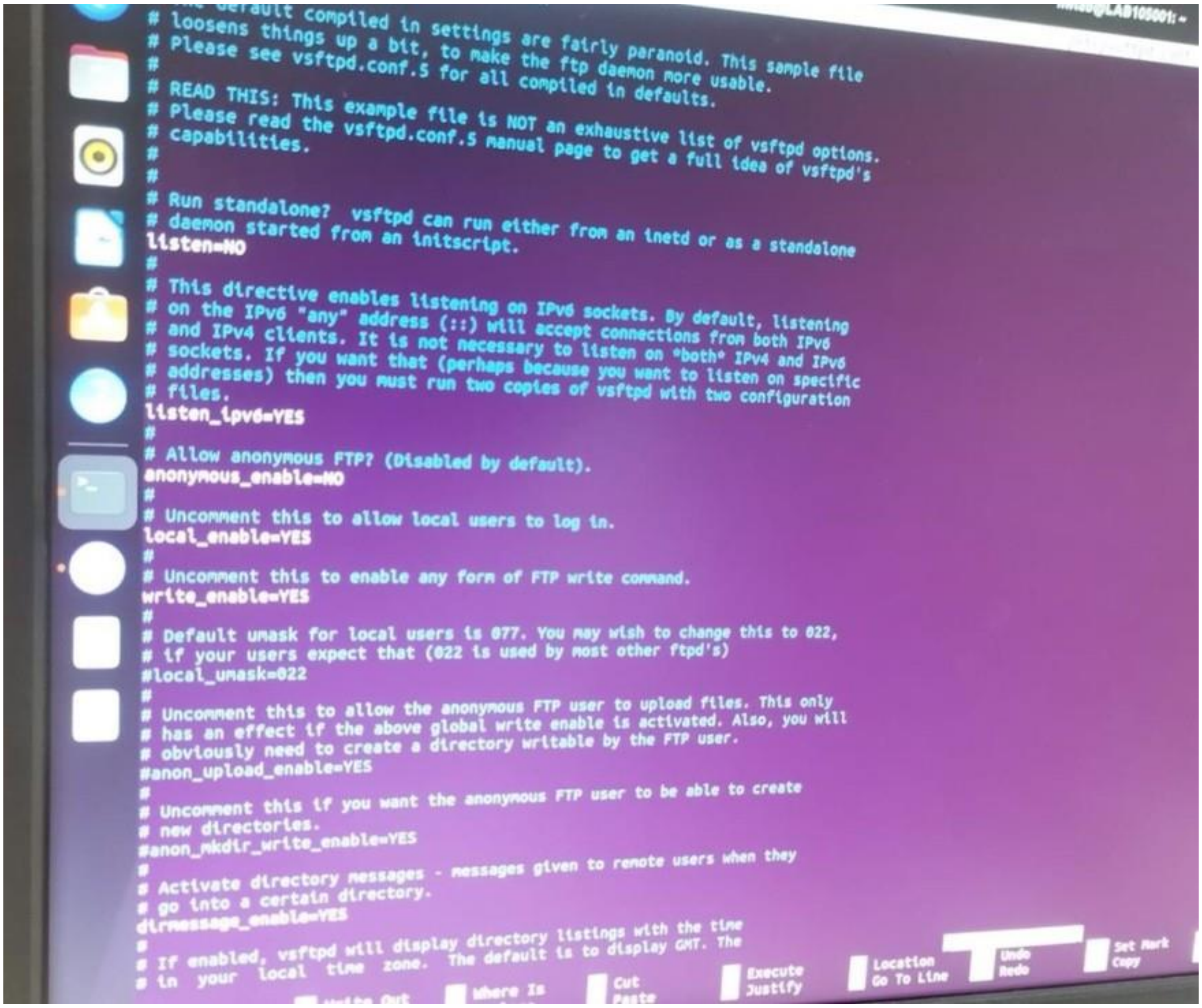
Lab-02B and Lab-03

Step-07) Modify Configuration Files to Allow Write Permissions.



```
Command: sudo nano /etc/vsftpd.conf
        Uncomment WRITE_ENABLE = YES
```

➔ In the vsftpd.conf file, you need to set the configuration option to allow FTP users to upload files to the server. By default, vsftpd might not permit file uploads, but setting write_enable=YES will enable this capability.

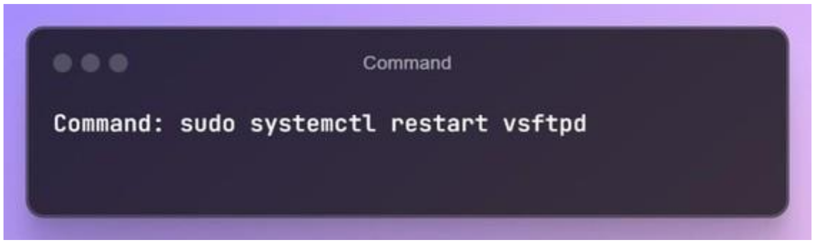


```
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
#
# Run standalone?  vsftpd can run either from an inetd or as a standalone
# daemon started from an initscript.
listen=NO
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
listen_ipv6=YES
#
# Allow anonymous FTP? (Disabled by default).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# If enabled, vsftpd will display directory listings with the time
# in  your  local  time  zone.  The default is to display GMT. The
```

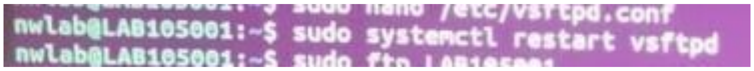
Step-08) Restart the FTP Server.

Lab-02B and Lab-03

- After making changes to the vsftpd configuration file (e.g., enabling write permissions), restart the service to apply these changes.



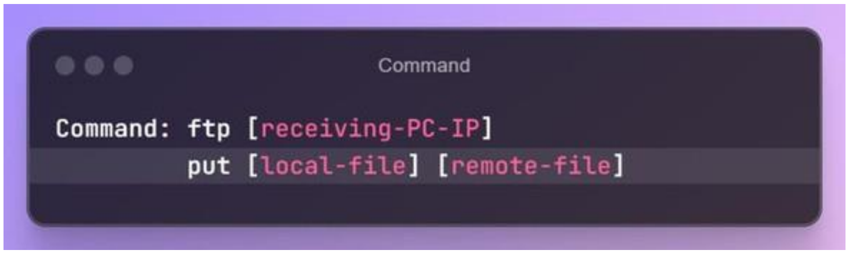
```
Command: sudo systemctl restart vsftpd
```



```
nwlab@LAB105001:~$ sudo nano /etc/vsftpd.conf
nwlab@LAB105001:~$ sudo systemctl restart vsftpd
nwlab@LAB105001:~$ sudo ftp LAB105001
```

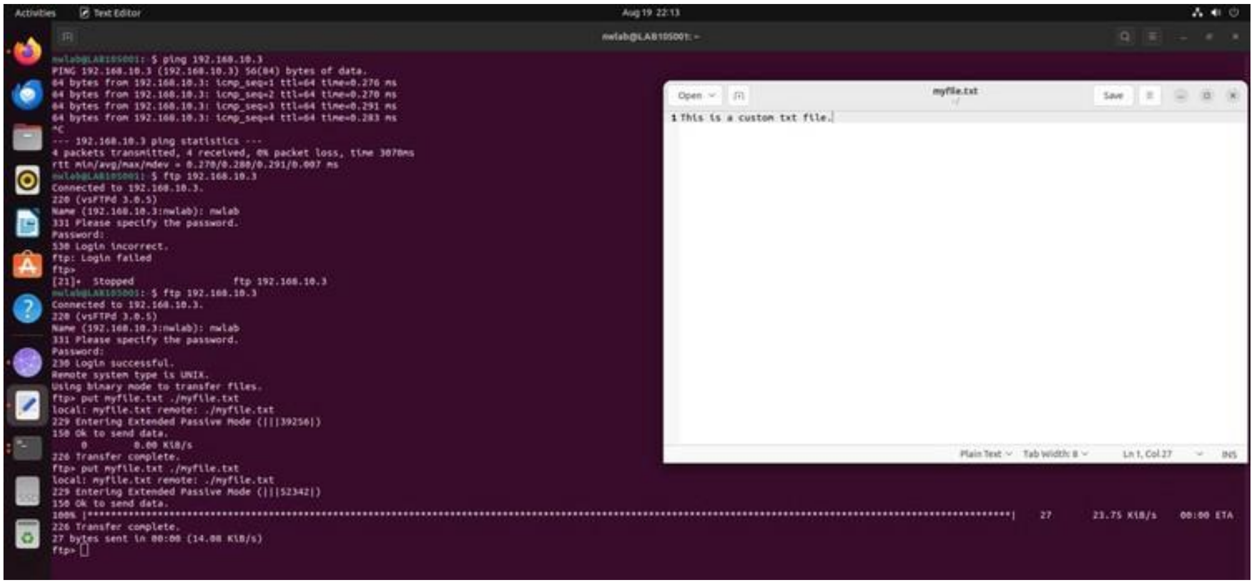
Step-09) Prepare to Send Files.

- First, establish a connection from the PC sending the files to the PC receiving the files.
- Once the connection is successful, use the PUT command to transfer the desired file to the receiver.



```
Command: ftp [receiving-PC-IP]
         put [local-file] [remote-file]
```

- The command uploads a file from the local machine (client) to the remote server.
- [local-file]: The name of the file on your local machine you wish to upload.
- [remote-file]: The name the file will have on the remote server after the upload. It can be the same as or different from the local file name.

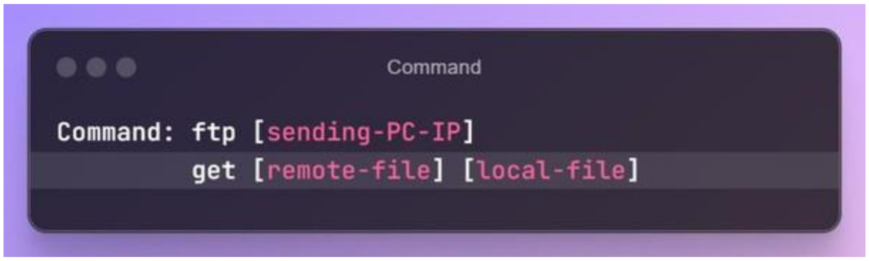


```
nwlab@LAB105001:~$ ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=64 time=0.276 ms
64 bytes from 192.168.10.3: icmp_seq=2 ttl=64 time=0.270 ms
64 bytes from 192.168.10.3: icmp_seq=3 ttl=64 time=0.291 ms
64 bytes from 192.168.10.3: icmp_seq=4 ttl=64 time=0.283 ms
^C
--- 192.168.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.270/0.280/0.291/0.007 ms
nwlab@LAB105001:~$ ftp 192.168.10.3
Connected to 192.168.10.3.
220 (vsFTPd 3.0.5)
Name (192.168.10.3:nwlab): nwlab
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed
ftp>
[21]+  Stopped                 ftp 192.168.10.3
nwlab@LAB105001:~$ ftp 192.168.10.3
Connected to 192.168.10.3.
220 (vsFTPd 3.0.5)
Name (192.168.10.3:nwlab): nwlab
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put myfile.txt ./myfile.txt
local: myfile.txt remote: ./myfile.txt
229 Entering Extended Passive Mode (|||39256|)
150 Ok to send data.
     0        0.00 KiB/s
226 Transfer complete.
ftp> put myfile.txt ./myfile.txt
local: myfile.txt remote: ./myfile.txt
229 Entering Extended Passive Mode (|||52342|)
150 Ok to send data.
100% |***********************************************************************************************************************|    27       23.75 KiB/s    00:00 ETA
226 Transfer complete.
27 bytes sent in 00:00 (14.08 KiB/s)
ftp> 
```

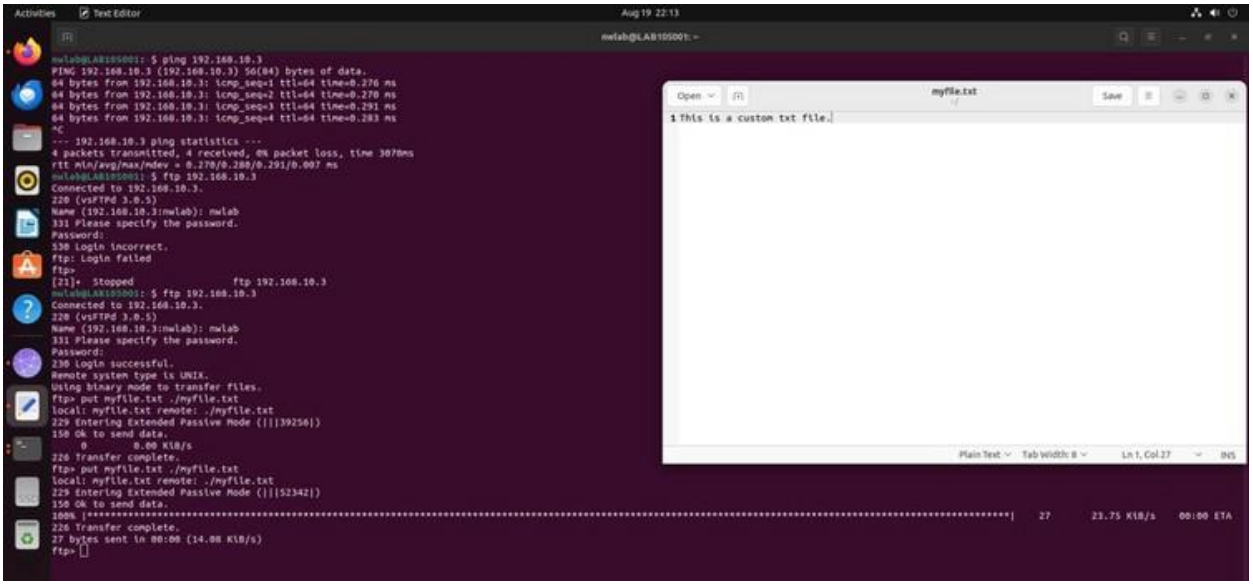
```
1 This is a custom txt file.
```

- On the server side, connect to the client’s IP address and then use the GET command to retrieve the file uploaded by the client.

Lab-02B and Lab-03



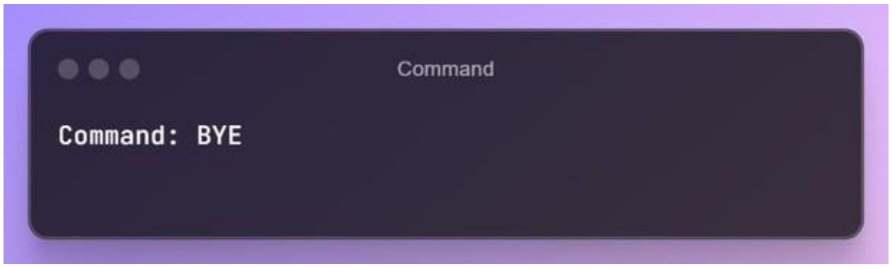
```
Command: ftp [sending-PC-IP]
         get [remote-file] [local-file]
```



```
nwlab@LAB105001:~$ ping 192.168.10.3
PING 192.168.10.3 (192.168.10.3) 56(84) bytes of data.
64 bytes from 192.168.10.3: icmp_seq=1 ttl=64 time=0.276 ms
64 bytes from 192.168.10.3: icmp_seq=2 ttl=64 time=0.270 ms
64 bytes from 192.168.10.3: icmp_seq=3 ttl=64 time=0.291 ms
64 bytes from 192.168.10.3: icmp_seq=4 ttl=64 time=0.283 ms
^C
--- 192.168.10.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.270/0.280/0.291/0.007 ms
nwlab@LAB105001:~$ ftp 192.168.10.3
Connected to 192.168.10.3.
220 (vsFTPd 3.0.5)
Name (192.168.10.3:nwlab): nwlab
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed
ftp>
[21]+  Stopped                 ftp 192.168.10.3
nwlab@LAB105001:~$ ftp 192.168.10.3
Connected to 192.168.10.3.
220 (vsFTPd 3.0.5)
Name (192.168.10.3:nwlab): nwlab
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put myfile.txt ./myfile.txt
local: myfile.txt remote: ./myfile.txt
229 Entering Extended Passive Mode (|||39256|)
150 Ok to send data.
     0        0.00 KiB/s
226 Transfer complete.
ftp> put myfile.txt ./myfile.txt
local: myfile.txt remote: ./myfile.txt
229 Entering Extended Passive Mode (|||52342|)
150 Ok to send data.
100% |*************************************************************************************************************************|    27       23.75 KiB/s    00:00 ETA
226 Transfer complete.
27 bytes sent in 00:00 (14.08 KiB/s)
ftp> 
```

```
1 This is a custom txt file.
```

Step-10) Exit the FTP Session After transferring the files, close the FTP session by executing the appropriate command.



```
Command: BYE
```

Sample Questions

Q1) What is a MAC address? Describe its structure and the OSI layer where it is utilized.

Solution: A MAC (Media Access Control) address is a distinctive identifier assigned to the network interface card (NIC) of a device. It plays a crucial role in ensuring that data packets are directed to the correct device within a local area network (LAN).

➔ **Structure of a MAC Address:**

- **Length:** A MAC address is 48 bits in length, which translates to 6 bytes.
- **Format:**
  - **Hexadecimal Format:** It is commonly displayed in hexadecimal notation.
- **Components:**
  - The first 24 bits (3 bytes) denote the manufacturer of the NIC.
  - The remaining 24 bits (3 bytes) are specific to the individual device, assigned by the manufacturer.

➔ **OSI Model Layer:**

1. **Layer:** Operates at the Data Link Layer (Layer 2) of the OSI model.
2. **Function:**

- **Local Communication:** Utilized for addressing and routing data frames within the same local network segment.
- **Switching:** Network switches utilize MAC addresses to forward data to the appropriate port based on the destination MAC address.

Q2) Why are IP addresses necessary despite the existence of MAC addresses?

Solution: MAC addresses and IP addresses serve distinct roles and function at different layers of the OSI model, making IP addresses indispensable for several reasons:

1. **Layer Differentiation:**

- **MAC Address (Layer 2):**
  - Used within the Data Link Layer (Layer 2) for addressing devices within the same local network (e.g., Ethernet, Wi-Fi).
  - Ensures data frames are correctly delivered to devices within the local network.
  - Each device has a unique MAC address within its local network.
- **IP Address (Layer 3):**
  - Operates at the Network Layer (Layer 3) for routing data across diverse networks, such as the internet.
  - Provides logical addressing to facilitate communication between devices across various networks.

    - IP addresses contain hierarchical information about both network and host for efficient routing.

2. **Network Routing:**

- **MAC Address Limitation:** MAC addresses are not routable beyond local networks; they are used solely within a single network segment.
- **IP Address Necessity:** IP addresses are essential for inter-network communication. Routers use IP addresses to determine the optimal path for data packets to travel across networks and reach their destination.

3. **Communication Scope:**

- **Local Communication:** Within a local network, devices communicate using MAC addresses. Switches, for instance, use MAC addresses to direct frames to the correct device on the same network.
- **Global Communication:** For communication between different networks or over the internet, IP addresses are required. Routers use IP addresses to forward packets to the appropriate network.

Q3) Are IP addresses truly unique?

Solution:

1. **Public IP Addresses:**
    - **Global Uniqueness:** Public IP addresses are unique worldwide, assigned by IANA or regional internet registries (RIRs).
    - **Usage:** Identifies devices on the global internet.
2. **Private IP Addresses:**
    - **Local Uniqueness:** Private IP addresses are unique within a specific local network.
    - **Reusability:** The same private IP ranges can be used in different private networks.
    - **Usage:** Employed within private networks like home or office environments.
3. **IPv4 vs. IPv6:**
    - **IPv4:**
        - Limited to around 4.3 billion addresses.
        - Address exhaustion is a concern due to the increasing number of devices.
    - **IPv6:**
        - Provides a substantially larger address space (approximately 340 undecillion addresses).
        - Designed to ensure global uniqueness with a vastly larger address pool.
4. **Static vs. Dynamic IP Addresses:**
    - **Static IP Address:**
        - Remains constant unless manually changed.
        - Guarantees consistent identification with the same IP address.

- **Dynamic IP Address:**
    - Assigned by a DHCP server and may change over time.
    - The IP address can be reassigned to different devices as leases expire.

Q4) What are classful and classless addressing?

Solution:

Classful Addressing:

1. **Fixed Classes:** IP addresses are divided into predefined classes (A, B, C, D, E) based on the initial bits.
2. **Default Subnet Masks:** Each class has a specific default subnet mask (e.g., Class A: /8, Class B: /16).
3. **Inefficiency:** Can result in inefficient use of IP addresses due to rigid class boundaries.
4. **Examples:**
    - Class A: Ranges from 1.0.0.0 to 126.0.0.0
    - Class B: Ranges from 128.0.0.0 to 191.255.0.0

Classless Addressing (CIDR):

1. **Flexible Addressing:** Removes fixed class boundaries for greater flexibility.
2. **Variable Subnet Masks:** Utilizes variable-length subnet masks (VLSM) for more efficient IP address allocation.
3. **Efficiency and Scalability:** Minimizes IP address wastage and allows for more precise network sizing.
4. **CIDR Notation:** Represents networks in formats like 192.168.1.0/24, where /24 indicates the size of the network portion.

Q5) How is Ping related to ICMP?

Solution: Ping relies on ICMP (Internet Control Message Protocol) to test network connectivity. When a Ping command is executed, it sends an ICMP Echo Request to the target device. If the device is reachable, it replies with an ICMP Echo Reply. Ping measures the round-trip time of these ICMP messages and checks for packet loss. Essentially, Ping uses ICMP to evaluate network connectivity and performance.

Q6) What is a subnet mask? What are the default subnet masks for Class A, B, and C?

Solution: A subnet mask is a 32-bit number that divides an IP address into network and host segments. It helps determine which portion of the IP address identifies the network and which part identifies the host within that network.

- **Function:** The subnet mask, when applied to an IP address using a bitwise AND operation, isolates the network address used for routing traffic.

Default Subnet Masks:

- **Class A:**
  - **Default Subnet Mask:** 255.0.0.0
  - **CIDR Notation:** /8
  - **Network/Host Division:** The first 8 bits (1 byte) are for the network, and the remaining 24 bits (3 bytes) are for hosts.
- **Class B:**
  - **Default Subnet Mask:** 255.255.0.0
  - **CIDR Notation:** /16
  - **Network/Host Division:** The first 16 bits (2 bytes) are for the network, and the remaining 16 bits (2 bytes) are for hosts.
- **Class C:**
  - **Default Subnet Mask:** 255.255.255.0
  - **CIDR Notation:** /24
  - **Network/Host Division:** The first 24 bits (3 bytes) are for the network, and the remaining 8 bits (1 byte) are for hosts.

Q7) What factors contribute to variations in RTT values across multiple Ping commands?

Solution: Variations in Round-Trip Time (RTT) across Ping command outputs can be attributed to several factors:

1. **Network Congestion:**
   - **Explanation:** High network traffic can cause delays in packet transmission and processing.
   - **Impact:** Increased RTT values due to congestion, causing delays as packets queue up or experience routing delays.
2. **Load Balancing:**
   - **Explanation:** Traffic may be distributed across multiple paths or servers.
   - **Impact:** Different paths may have different latencies, leading to fluctuating RTT values.
3. **Routing Changes:**
   - **Explanation:** Dynamic routing protocols might alter packet paths based on current network conditions.

   - **Impact:** Changes in routing paths can result in varying RTT if packets travel through different nodes.

4. **Server Load:**
   - **Explanation:** The server being pinged may have varying loads.
   - **Impact:** High server load can cause delays in processing and responding to pings, resulting in increased RTT.
5. **Network Equipment Performance:**
   - **Explanation:** Performance of network equipment such as routers and switches can vary.
   - **Impact:** Variability in equipment performance can cause fluctuations in RTT.
6. **Packet Loss and Retransmissions:**
   - **Explanation:** Packet loss leads to retransmissions, which can increase RTT.
   - **Impact:** Even minor packet loss can cause significant RTT increases due to retransmission delays.
7. **Network Interference:**
   - **Explanation:** Wireless networks can experience interference from other devices.
   - **Impact:** Interference can lead to increased RTT due to retransmissions or transmission delays.
8. **System Overhead:**
   - **Explanation:** Both sending and receiving systems have processing overhead.
   - **Impact:** Variability in system performance or concurrent processes can affect RTT.

Example Analysis:

- **Initial Ping RTT:** 30 ms
- **Subsequent RTTs:** 50 ms, 45 ms, 60 ms

Potential Causes:

1. Increased Network Traffic: Higher traffic during periods of higher RTTs.
2. Routing Changes: Possible adjustments in routing paths.
3. Server Load Fluctuations: Variations in server load during pings.

Q.8) Analyze the difference between time effects of Ping Command, results from Hub and Switch?

Solution: Here’s a comparative analysis of the time effects of the Ping command when using a hub versus a switch, presented in tabular form:

Aspect	Hub	Switch
Traffic Handling	Broadcasts all traffic to all connected devices.	Sends traffic only to the intended recipient based on MAC address.
Collision Domain	All devices share the same collision domain, leading to potential collisions.	Each port is a separate collision domain, reducing collisions.
Network Latency	Higher latency due to the potential for collisions and the need to process and broadcast packets to all devices.	Lower latency due to reduced collisions and direct forwarding of packets.
Ping Command Response Time	May have increased response time due to network congestion and collisions.	Typically has faster response time due to efficient traffic handling and reduced congestion.
Efficiency	Less efficient due to the broadcast nature and potential for network collisions.	More efficient due to selective forwarding and reduced collisions.
Scalability	Less scalable as more devices increase the likelihood of collisions and network congestion.	More scalable as it can handle more devices with less impact on performance due to reduced collisions.

- **Switches** effectively manage traffic and minimise accidents, resulting in decreased RTT and improved performance.
- **Hubs** typically cause higher RTT and higher latency because of network collisions and congestion.

To create Sub-LAN Networks

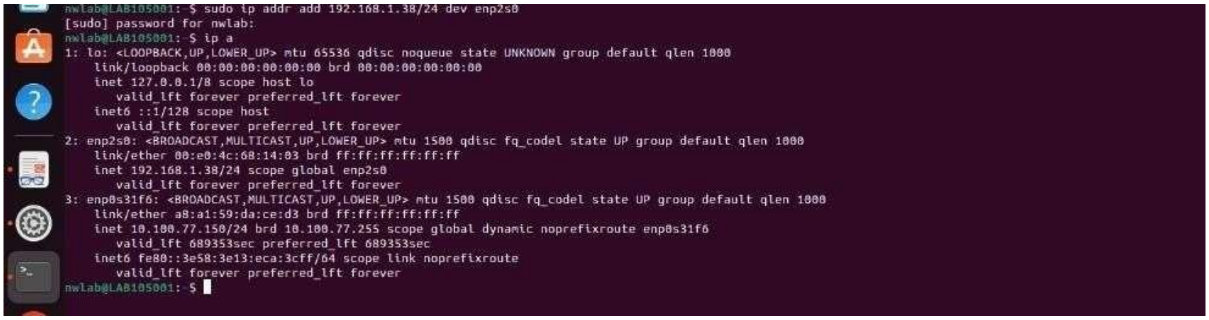
Goal: Attach three PCs to a switch, setting up two of them in one sub-LAN and the other in a separate sub-LAN.
Method:

Step 1) Use straight through ethernet cables to physically connect both PCs to the switch. Connect one end of each cable to the PC's network port and the other end to a switch port that is open.

Step 2) Type the command "ip a" to examine the PCs' network interface.

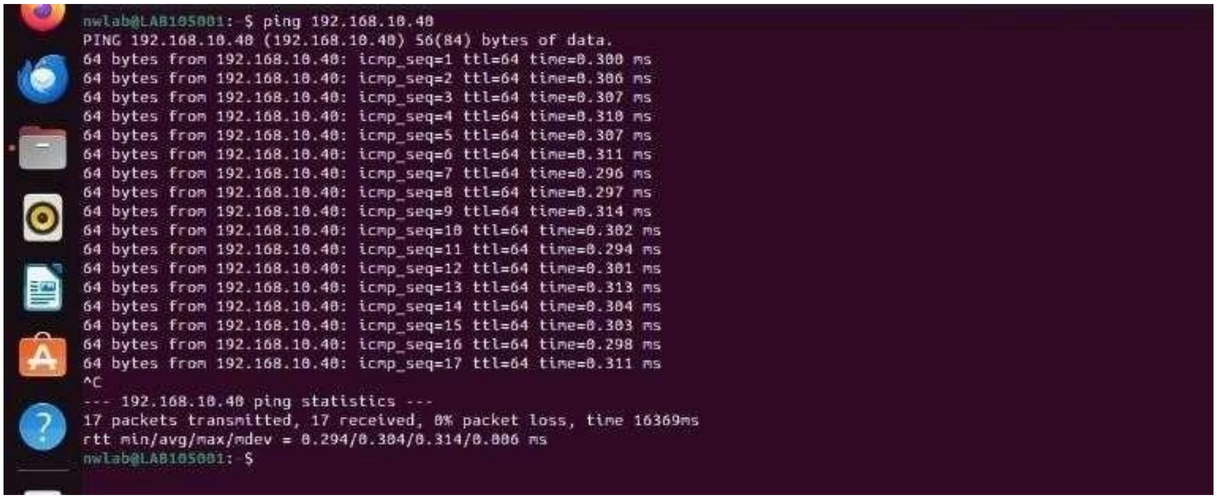
A. How to create two PCs on the same subnet:

Step 3) First, give each PC on the same network a static IP address (for example, PC1's address is 192.168.10.38 and PC2's address is 192.168.10.40).



```
nwlab@LAB105001:~$ sudo ip addr add 192.168.1.38/24 dev enp2s0
[sudo] password for nwlab:
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:14:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.38/24 scope global enp2s0
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.100.77.150/24 brd 10.100.77.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 689353sec preferred_lft 689353sec
    inet6 fe80::3e58:3e13:eca:3cff/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
nwlab@LAB105001:~$
```

Step-4) Now, use PC1's "ping command" to examine and confirm PC1 and PC2 connections.

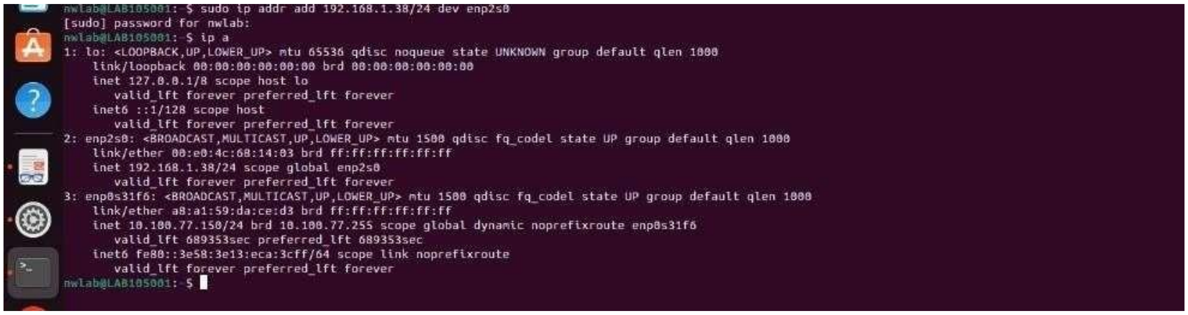


```
nwlab@LAB105001:~$ ping 192.168.10.40
PING 192.168.10.40 (192.168.10.40) 56(84) bytes of data.
64 bytes from 192.168.10.40: icmp_seq=1 ttl=64 time=0.300 ms
64 bytes from 192.168.10.40: icmp_seq=2 ttl=64 time=0.306 ms
64 bytes from 192.168.10.40: icmp_seq=3 ttl=64 time=0.307 ms
64 bytes from 192.168.10.40: icmp_seq=4 ttl=64 time=0.310 ms
64 bytes from 192.168.10.40: icmp_seq=5 ttl=64 time=0.307 ms
64 bytes from 192.168.10.40: icmp_seq=6 ttl=64 time=0.311 ms
64 bytes from 192.168.10.40: icmp_seq=7 ttl=64 time=0.296 ms
64 bytes from 192.168.10.40: icmp_seq=8 ttl=64 time=0.297 ms
64 bytes from 192.168.10.40: icmp_seq=9 ttl=64 time=0.314 ms
64 bytes from 192.168.10.40: icmp_seq=10 ttl=64 time=0.302 ms
64 bytes from 192.168.10.40: icmp_seq=11 ttl=64 time=0.294 ms
64 bytes from 192.168.10.40: icmp_seq=12 ttl=64 time=0.301 ms
64 bytes from 192.168.10.40: icmp_seq=13 ttl=64 time=0.313 ms
64 bytes from 192.168.10.40: icmp_seq=14 ttl=64 time=0.304 ms
64 bytes from 192.168.10.40: icmp_seq=15 ttl=64 time=0.303 ms
64 bytes from 192.168.10.40: icmp_seq=16 ttl=64 time=0.298 ms
64 bytes from 192.168.10.40: icmp_seq=17 ttl=64 time=0.311 ms
^C
--- 192.168.10.40 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 16369ms
rtt min/avg/max/mdev = 0.294/0.304/0.314/0.006 ms
nwlab@LAB105001:~$
```

Similarly, when we use PC2's "ping command" to verify PC1's connection, we receive the same results.

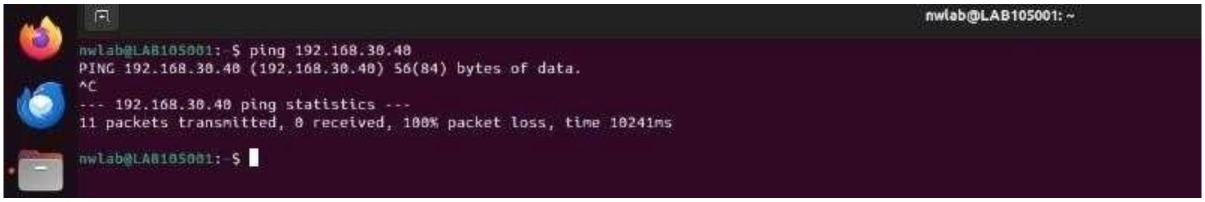
A. For making Two PCs on different subnet:

Step-3) First, give each PC a static IP address and ensure that the subnets are selected differently. (i.e. 192.168.10.38 for PC1 and 192.168.30.40 for PC2).



```
nwlab@LAB105001:~$ sudo ip addr add 192.168.1.38/24 dev enp2s0
[sudo] password for nwlab:
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:14:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.38/24 scope global enp2s0
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.100.77.150/24 brd 10.100.77.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 689353sec preferred_lft 689353sec
    inet6 fe80::3e58:3e13:eca:3cff/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
nwlab@LAB105001:~$
```

Step-4) Now, use PC2's "ping command" to examine and confirm PC1 and PC2 connections.



```
nwlab@LAB105001:~$ ping 192.168.30.40
PING 192.168.30.40 (192.168.30.40) 56(84) bytes of data.
^C
--- 192.168.30.40 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10241ms

nwlab@LAB105001:~$ 
```

Results:

9. **For two PCs connected to the same subnet:** The PCs have successfully established a connection. The ping command has been used to confirm that both PCs are connected.
10. **Regarding the Connection of Two PCs on Different Subnets:** Because the PCs are on different subnets, the connection between them cannot be properly created. As a result, even when packets are transmitted, none are received.

❖ Sample Questions:- (Part-A)

Q1. Explain the differences between a LAN and a sub-LAN. Why would you use sub-LANs within a LAN?

A1) A **LAN (Local Area Network)** is a network that connects devices within a limited geographic area, such as a home, office, or building. All devices on a LAN can typically communicate with each other directly. The key characteristics of a LAN are its limited geographic scope, high-speed connections, and typically, a single shared network infrastructure.

A **sub-LAN**, often referred to as a subnet (subnetwork), is a smaller, logically divided section of a larger LAN. Subnets are created by subdividing an IP network address into multiple smaller network segments. This is done using subnetting, where a network's IP address is split into a network portion and a host portion, allowing for multiple distinct sub-LANs within a single LAN.

Differences Between a LAN and a Sub-LAN:

1. Scope:

- LAN: Encompasses the entire local network within a geographic area.

- Sub-LAN: Represents a segment within the larger LAN, often created for logical separation.

2. Addressing:

- LAN: Uses a single IP address range for all devices.

- Sub-LAN: Uses distinct IP address ranges for different subnets within the LAN.

3. Broadcast Domain:

- LAN: Typically, all devices are part of the same broadcast domain, meaning broadcast messages are received by all devices.

- Sub-LAN: Each subnet has its own broadcast domain, reducing broadcast traffic across the entire network.

4. Routing:

- LAN: Devices communicate directly without the need for a router when on the same network.

- Sub-LAN: Communication between devices on different subnets requires routing, often through a router or Layer 3 switch.

5. Security and Management:

- LAN: Security policies and network management are typically applied to the entire network.

- Sub-LAN: Allows for more granular control, where different security policies or network configurations can be applied to different subnets.

Why Use Sub-LANs Within a LAN?

1. Improved Network Performance: By dividing a LAN into subnets, you reduce the size of broadcast domains, which can decrease unnecessary traffic and improve overall network performance.

2. Enhanced Security: Subnets allow for the segmentation of sensitive data or departments, reducing the risk of unauthorized access or breaches within the entire LAN.

3. Efficient IP Address Management: Subnetting helps manage IP addresses more effectively, particularly in larger networks, by dividing a large address space into smaller, more manageable segments.

4. Simplified Network Management: By logically separating different areas of the network, administrators can more easily apply policies, troubleshoot issues, and manage the network.

5. Scalability: As a network grows, subnetting allows for easier expansion without the need for major reconfigurations of the existing network.

Using sub-LANs or subnets within a LAN is a common practice to optimize network performance, enhance security, and manage resources more efficiently.

Q-2) Explain the role of subnet masks in dividing networks. How does the subnet mask in this experiment determine which PCs are in the same or different sub- LANs?

A2) Role of Subnet Masks in Dividing Networks:

- ➔ Subnet masks divide a larger network into smaller sub-networks (subnets), which helps in efficient network management.
- ➔ The subnet mask determines which portion of an IP address identifies the network and which part identifies the host within that network.
- ➔ Subnetting reduces the size of routing tables and enhances routing efficiency by limiting the broadcast domains.
- ➔ By isolating different subnets, a subnet mask can enhance network security, restricting communication between different subnets.
- ➔ Subnets can be used to allocate IP addresses more efficiently, ensuring that IP addresses are not wasted.

Determining PCs in the Same or Different Sub-LANs:

- ➔ The subnet mask is applied to the IP addresses of PCs to determine which portion of the address is the network part.
  - o If two PCs have IP addresses that, when masked, result in the same network address, they are in the same sub-LAN.
  - o If the network portion of the IP addresses (after applying the subnet mask) differs, the PCs are in different sub-LANs.
- ➔ PCs within the same subnet can communicate directly (without routing), while communication between different subnets requires a router.
- ➔ The subnet mask defines the range of IP addresses that belong to a particular subnet, determining the possible devices within that sub-LAN.

Q-3) Given the configuration above, if PC3 needed to communicate with PC1 and PC2, what additional network configuration would be necessary?

A3)

1. Router: Add a router to your network. This router will facilitate communication between the different subnets where PC1, PC2, and PC3 are located.

2. Set Default Gateway: Set the default gateway (router’s IP address) on PC1 and PC2 to ensure they can communicate with devices outside their subnet. Set the default gateway on

PC3 to point to the router’s IP address in its subnet, allowing it to route traffic to other subnets.

3. Assign IP Addresses: Ensure PC1 and PC2 have IP addresses within the same subnet (e.g., 192.168.10.x). Assign an IP address to PC3 in a different subnet (e.g., 192.168.20.x) to ensure it is logically separated.

4. Routing: Configure the router to route traffic between the subnets. This configuration might involve setting up static routes or relying on the router's automatic routing capabilities to manage communication between PC1, PC2, and PC3.

Q-4) What challenges might you face if you were to add more devices to each sub- LAN?

A4)

1. IP Address Exhaustion:

○ Each sub-LAN has a limited number of available IP addresses based on its subnet mask. Adding more devices could exhaust the available IP addresses, requiring a reconfiguration of the subnet mask or the creation of additional sub-LANs.

2. Increased Network Traffic:

○ As more devices are added, the amount of broadcast and multicast traffic within each sub-LAN may increase, potentially leading to network congestion and reduced performance.

3. Complexity in Network Management:

○ Managing and maintaining a larger number of devices across multiple sub-LANs can become more complex, requiring careful planning of IP address allocation, routing, and security policies.

4. Potential for IP Address Conflicts:

○ With more devices, the risk of IP address conflicts increases, particularly if devices are manually configured or if DHCP (Dynamic Host Configuration Protocol) is not properly managed.

5. Performance Degradation:

○ Adding many devices to a sub-LAN can strain the network’s resources, leading to slower response times, higher latency, and potential bottlenecks, especially if the network hardware (like switches or routers) is not designed to handle the increased load.

6. Security Concerns:

○ More devices in each sub-LAN can increase the attack surface, making the network more vulnerable to security breaches. Ensuring proper security measures, such as firewalls, access control lists (ACLs), and network segmentation, becomes more critical.

7. Difficulty in Troubleshooting:
○ As the number of devices increases, identifying and resolving network issues can become more difficult. Pinpointing the source of a problem within a crowded network may require advanced monitoring tools and techniques.

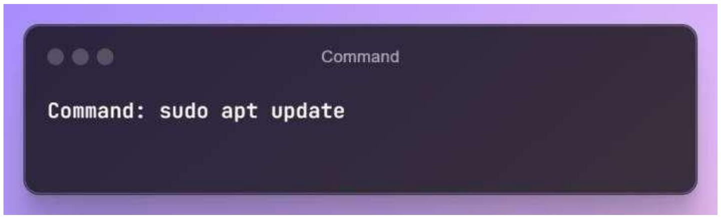
8. Scalability Limitations:
○ The existing network infrastructure may not be scalable enough to support a significant increase in devices. This may necessitate upgrades to network hardware, reconfiguration of subnets, or even redesigning the network architecture.

To set up and configure DHCP Server

Goal: To set up and configure DHCP

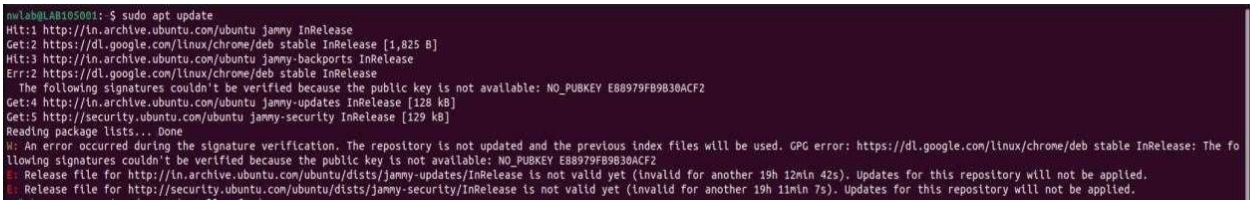
Procedure for Server Side:

Step-01) Firstly Update all the packages on the computer.



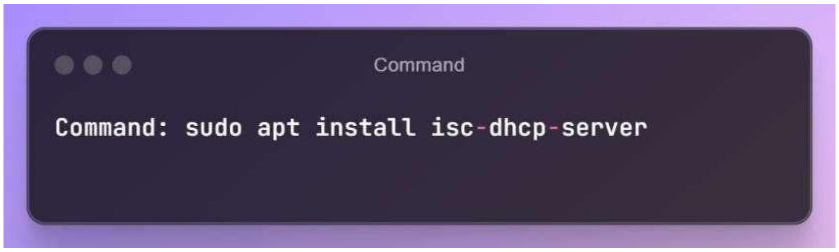
```
Command: sudo apt update
```

➔ Using this command will ensure that, prior to installing or upgrading software, you get the most recent information available regarding packages and their versions.



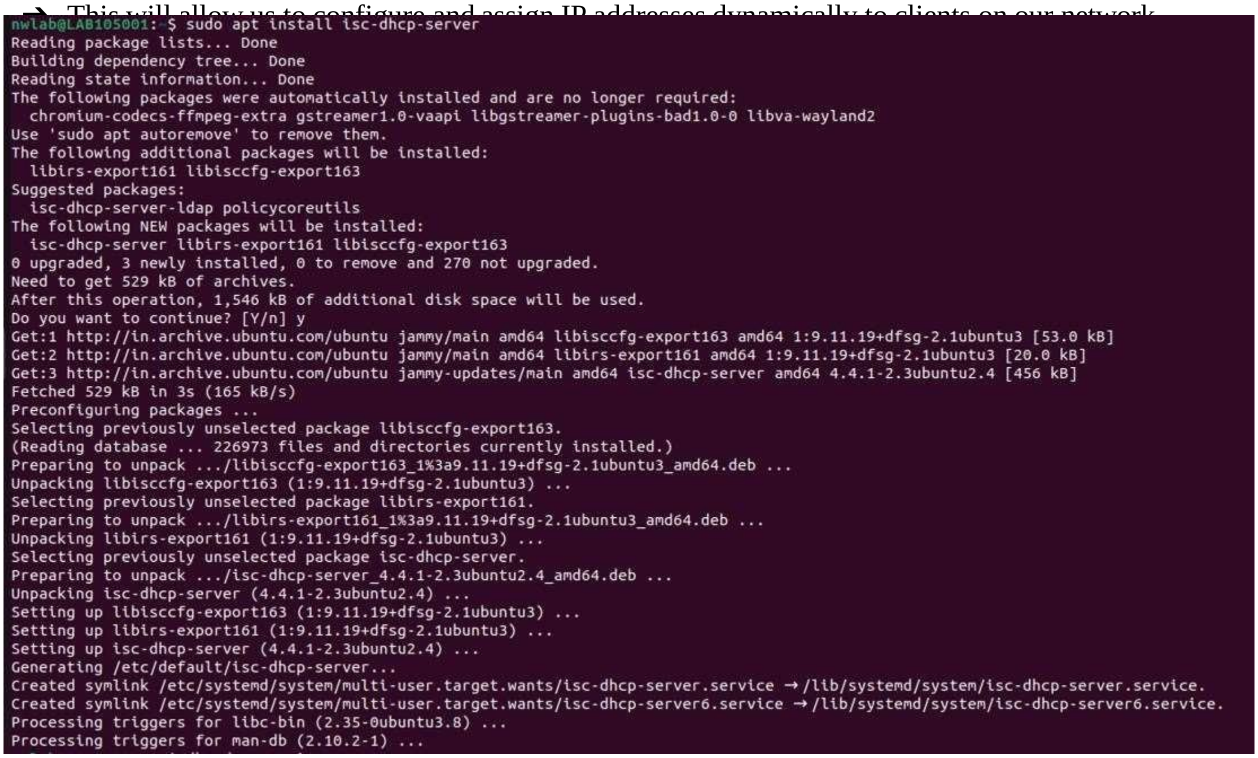
```
nwlab@LAB105001:~$ sudo apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 https://dl.google.com/linux/chrome/deb stable InRelease [1,825 B]
Hit:3 http://in.archive.ubuntu.com/ubuntu jammy-backports InRelease
Err:2 https://dl.google.com/linux/chrome/deb stable InRelease
  The following signatures couldn't be verified because the public key is not available: NO_PUBKEY E88979FB9B30ACF2
Get:4 http://in.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Reading package lists... Done
W: An error occurred during the signature verification. The repository is not updated and the previous index files will be used. GPG error: https://dl.google.com/linux/chrome/deb stable InRelease: The fo
llowing signatures couldn't be verified because the public key is not available: NO_PUBKEY E88979FB9B30ACF2
E: Release file for http://in.archive.ubuntu.com/ubuntu/dists/jammy-updates/InRelease is not valid yet (invalid for another 19h 12min 42s). Updates for this repository will not be applied.
E: Release file for http://security.ubuntu.com/ubuntu/dists/jammy-security/InRelease is not valid yet (invalid for another 19h 11min 7s). Updates for this repository will not be applied.
```

Step-02) Installing isc-dhcp-server Package



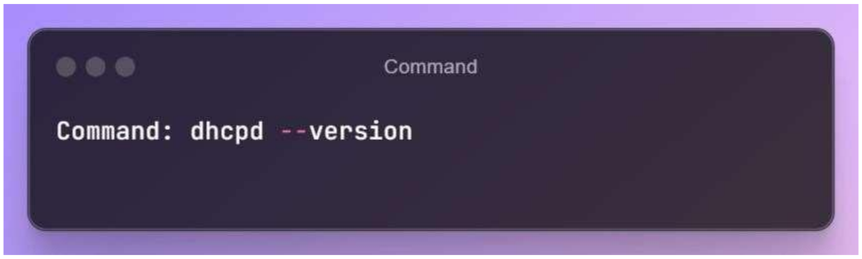
```
Command: sudo apt install isc-dhcp-server
```

- ➔ This command is used to install the ISC DHCP server on a Linux system.
- ➔ This will allow us to configure and assign IP addresses dynamically to clients on our network.



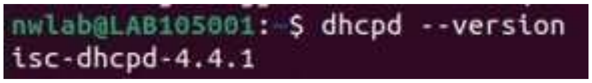
```
nwlab@LAB105001:~$ sudo apt install isc-dhcp-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  chromium-codecs-ffmpeg-extra gstreamer1.0-vaapi libgstreamer-plugins-bad1.0-0 libva-wayland2
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libirs-export161 libisccfg-export163
Suggested packages:
  isc-dhcp-server-ldap policycoreutils
The following NEW packages will be installed:
  isc-dhcp-server libirs-export161 libisccfg-export163
0 upgraded, 3 newly installed, 0 to remove and 270 not upgraded.
Need to get 529 kB of archives.
After this operation, 1,546 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libisccfg-export163 amd64 1:9.11.19+dfsg-2.1ubuntu3 [53.0 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 libirs-export161 amd64 1:9.11.19+dfsg-2.1ubuntu3 [20.0 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy-updates/main amd64 isc-dhcp-server amd64 4.4.1-2.3ubuntu2.4 [456 kB]
Fetched 529 kB in 3s (165 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libisccfg-export163.
(Reading database ... 226973 files and directories currently installed.)
Preparing to unpack .../libisccfg-export163_1%3a9.11.19+dfsg-2.1ubuntu3_amd64.deb ...
Unpacking libisccfg-export163 (1:9.11.19+dfsg-2.1ubuntu3) ...
Selecting previously unselected package libirs-export161.
Preparing to unpack .../libirs-export161_1%3a9.11.19+dfsg-2.1ubuntu3_amd64.deb ...
Unpacking libirs-export161 (1:9.11.19+dfsg-2.1ubuntu3) ...
Selecting previously unselected package isc-dhcp-server.
Preparing to unpack .../isc-dhcp-server_4.4.1-2.3ubuntu2.4_amd64.deb ...
Unpacking isc-dhcp-server (4.4.1-2.3ubuntu2.4) ...
Setting up libisccfg-export163 (1:9.11.19+dfsg-2.1ubuntu3) ...
Setting up libirs-export161 (1:9.11.19+dfsg-2.1ubuntu3) ...
Setting up isc-dhcp-server (4.4.1-2.3ubuntu2.4) ...
Generating /etc/default/isc-dhcp-server...
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server.service → /lib/systemd/system/isc-dhcp-server.service.
Created symlink /etc/systemd/system/multi-user.target.wants/isc-dhcp-server6.service → /lib/systemd/system/isc-dhcp-server6.service.
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
Processing triggers for man-db (2.10.2-1) ...
```

Step-03) Verify if the dhcp-server is properly installed or not.



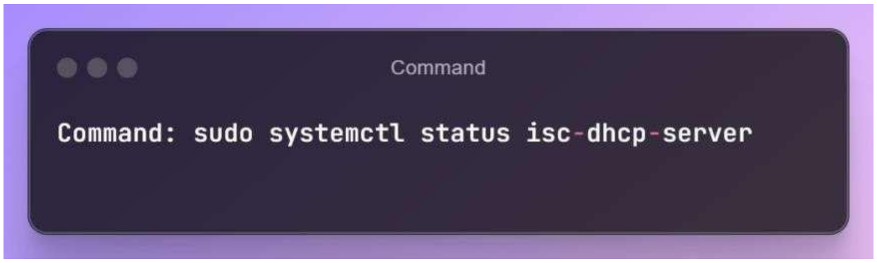
```
Command: dhcpd --version
```

- ➔ This will show the version of isc-dhcp-server.
- ➔ If it shows the version, it means that the package is successfully installed.



```
nwlab@LAB105001:~$ dhcpd --version
isc-dhcpd-4.4.1
```

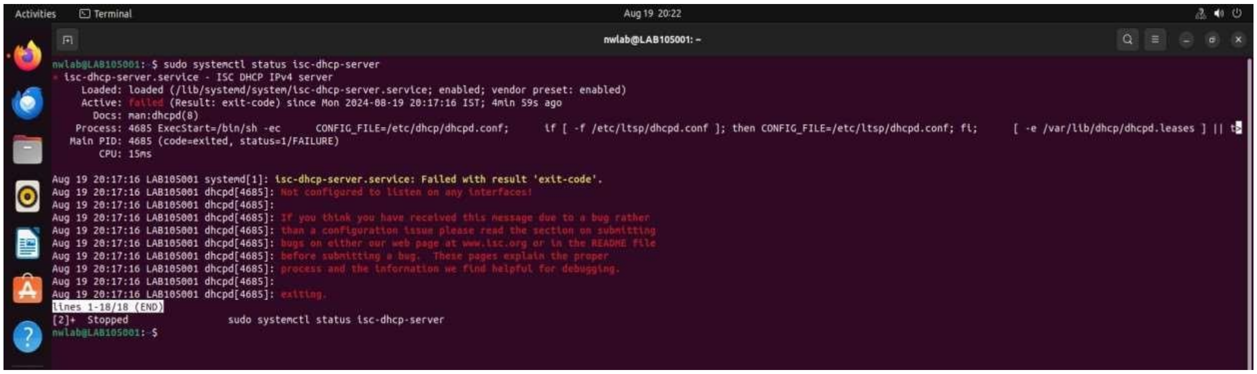
Step-04) Checking the status of the dhcp server.



```
Command: sudo systemctl status isc-dhcp-server
```

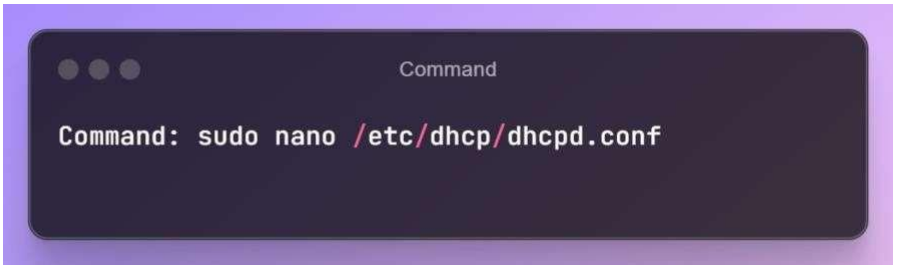
- ➔ If the service is running correctly, you'll see something like “active” in the output and If there's an issue, it might show “Inactive”.

➔ In case of inactivity, we need to change the config file.



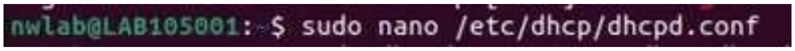
```
nwlab@LAB105001:~$ sudo systemctl status isc-dhcp-server
× isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
     Active: failed (Result: exit-code) since Mon 2024-08-19 20:17:16 IST; 4min 59s ago
       Docs: man:dhcpd(8)
    Process: 4685 ExecStart=/bin/sh -ec      CONFIG_FILE=/etc/dhcp/dhcpd.conf;      if [ -f /etc/ltsp/dhcpd.conf ]; then CONFIG_FILE=/etc/ltsp/dhcpd.conf; fi;      [ -e /var/lib/dhcp/dhcpd.leases ] || t>
   Main PID: 4685 (code=exited, status=1/FAILURE)
        CPU: 15ms

Aug 19 20:17:16 LAB105001 systemd[1]: isc-dhcp-server.service: Failed with result 'exit-code'.
Aug 19 20:17:16 LAB105001 dhcpd[4685]: Not configured to listen on any interfaces!
Aug 19 20:17:16 LAB105001 dhcpd[4685]:
Aug 19 20:17:16 LAB105001 dhcpd[4685]: If you think you have received this message due to a bug rather
Aug 19 20:17:16 LAB105001 dhcpd[4685]: than a configuration issue please read the section on submitting
Aug 19 20:17:16 LAB105001 dhcpd[4685]: bugs on either our web page at www.isc.org or in the README file
Aug 19 20:17:16 LAB105001 dhcpd[4685]: before submitting a bug.  These pages explain the proper
Aug 19 20:17:16 LAB105001 dhcpd[4685]: process and the information we find helpful for debugging.
Aug 19 20:17:16 LAB105001 dhcpd[4685]:
Aug 19 20:17:16 LAB105001 dhcpd[4685]: exiting.
lines 1-18/18 (END)
[2]+  Stopped                 sudo systemctl status isc-dhcp-server
nwlab@LAB105001:~$
```

<u>Step-05)</u> Configuring the DHCP config file.



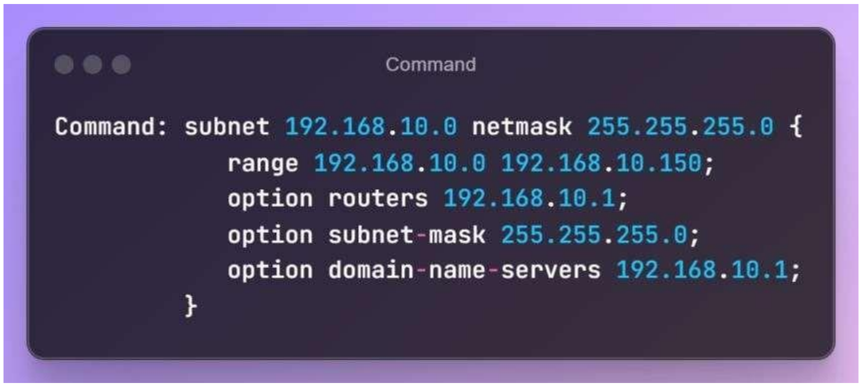
```
Command: sudo nano /etc/dhcp/dhcpd.conf
```

➔ This command will open the DHCP config file in the nano text editor, where we will change some of the configurations.



```
nwlab@LAB105001:~$ sudo nano /etc/dhcp/dhcpd.conf
```

<u>Step-06)</u> Inserting the changes into the config file.



```
Command: subnet 192.168.10.0 netmask 255.255.255.0 {
             range 192.168.10.0 192.168.10.150;
             option routers 192.168.10.1;
             option subnet-mask 255.255.255.0;
             option domain-name-servers 192.168.10.1;
         }
```

➔ This command needs to be inserted into the dhcp config file using nano text editor.

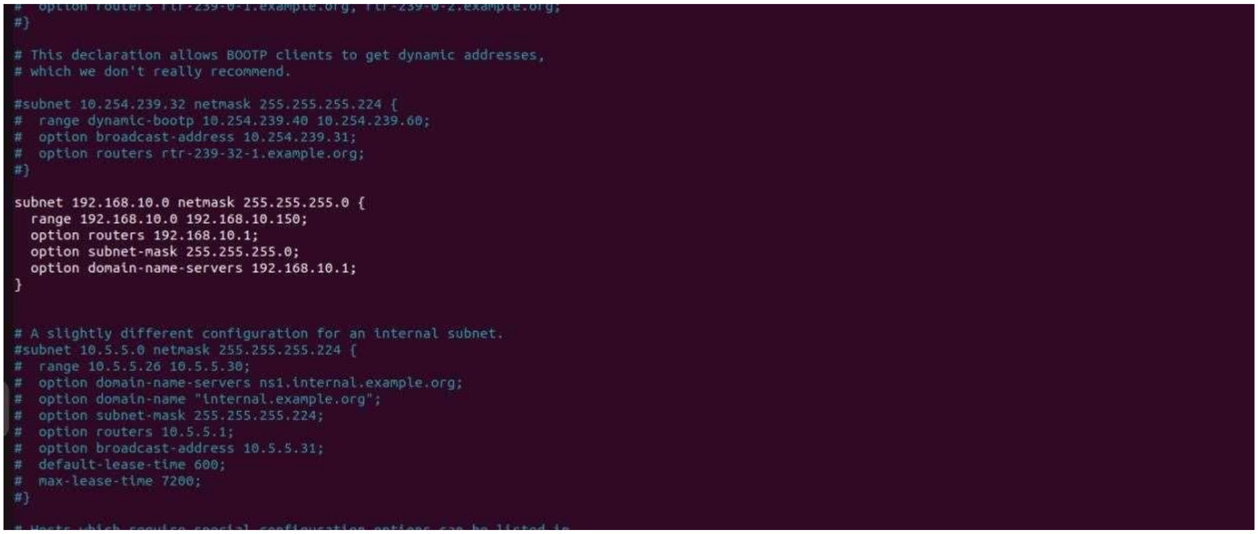
➔ Here, we are defining the subnet 192.168.10.0 and its netmask 255.255.255.0, which means the network will have the IP address range from 192.168.10.0 to 192.168.10.255.

➔ Then, we are assigning the range of IP addresses to the server, from which the server will assign the dynamic IP to the clients. In this case, the server can assign the IP’s from 192.168.10.1 to 192.168.10.150. Here 192.168.10.0 is not assignable to hosts.

➔ After that, option router 192.168.10.1 means that the default router/gateway for the clients will be

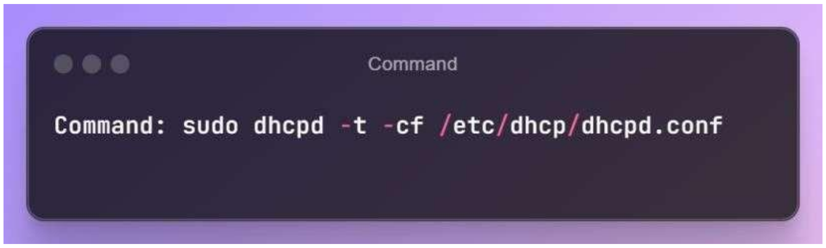
192.168.10.1.

- ➔ The, the clients must use their subnet mask as 255.255.255.0.
- ➔ At last, we are defining the DNS to be used by client, here it is 192.168.10.1.



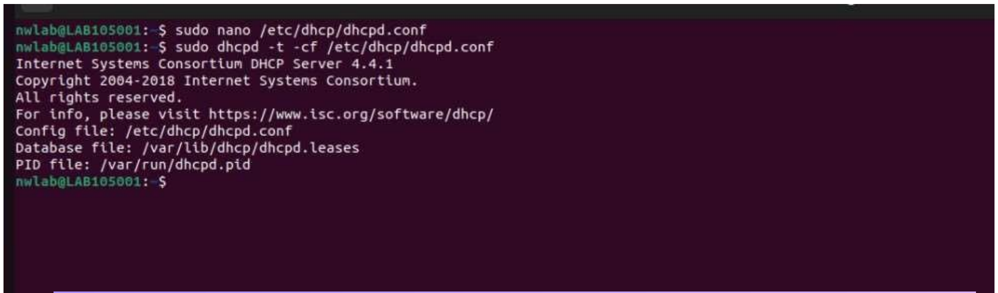
```
#  option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}

# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

subnet 192.168.10.0 netmask 255.255.255.0 {
  range 192.168.10.0 192.168.10.150;
  option routers 192.168.10.1;
  option subnet-mask 255.255.255.0;
  option domain-name-servers 192.168.10.1;
}


# A slightly different configuration for an internal subnet.
#subnet 10.5.5.0 netmask 255.255.255.224 {
#  range 10.5.5.26 10.5.5.30;
#  option domain-name-servers ns1.internal.example.org;
#  option domain-name "internal.example.org";
#  option subnet-mask 255.255.255.224;
#  option routers 10.5.5.1;
#  option broadcast-address 10.5.5.31;
#  default-lease-time 600;
#  max-lease-time 7200;
#}

# Hosts which require special configuration options can be listed in
```

Step-07) Testing the syntax of the DHCP config file.

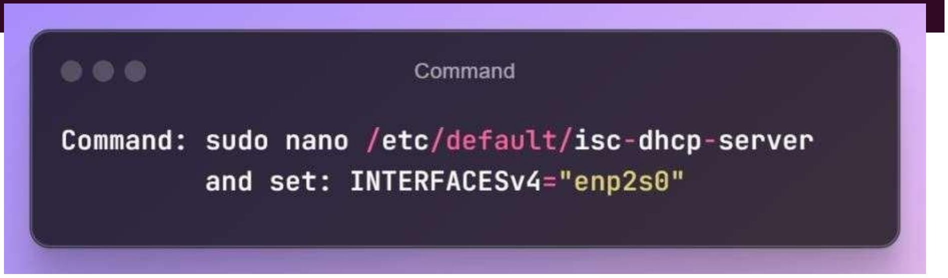


```
Command: sudo dhcpd -t -cf /etc/dhcp/dhcpd.conf
```

- ➔ This command is used to test the syntax of the DHCP server configuration file before starting or restarting the DHCP service.



```
nwlab@LAB105001:~$ sudo nano /etc/dhcp/dhcpd.conf
nwlab@LAB105001:~$ sudo dhcpd -t -cf /etc/dhcp/dhcpd.conf
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpd.conf
Database file: /var/lib/dhcp/dhcpd.leases
PID file: /var/run/dhcpd.pid
nwlab@LAB105001:~$
```

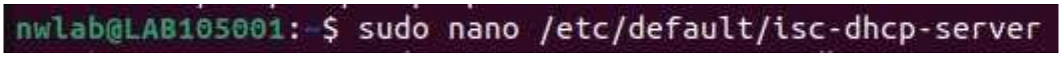


```
Command: sudo nano /etc/default/isc-dhcp-server
         and set: INTERFACESv4="enp2s0"
```

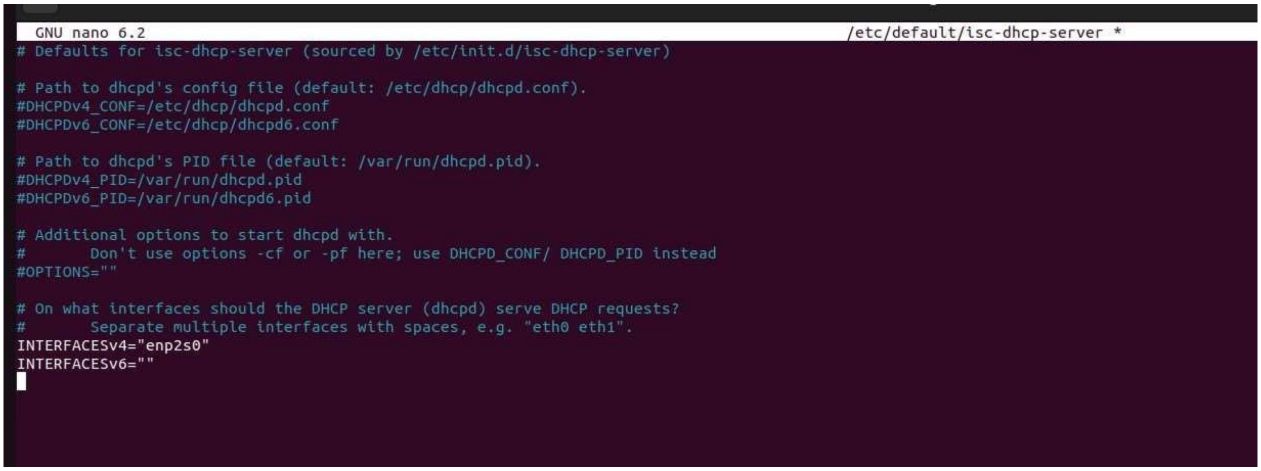
Step-08) Defining the NIC on which the DHCP server should listen.

- ➔ Specifying the interface ensures that the DHCP server only listens for and serves DHCP requests on that particular network interface.

➔ This is especially important on systems with multiple network interfaces.

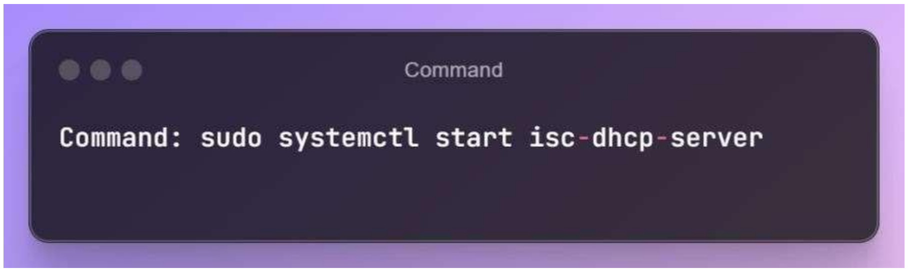


```
nwlab@LAB105001:~$ sudo nano /etc/default/isc-dhcp-server
```



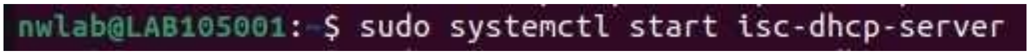
```
  GNU nano 6.2                                                              /etc/default/isc-dhcp-server *
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp2s0"
INTERFACESv6=""
```

<u>Step-09)</u> Starting the DHCP-server service.



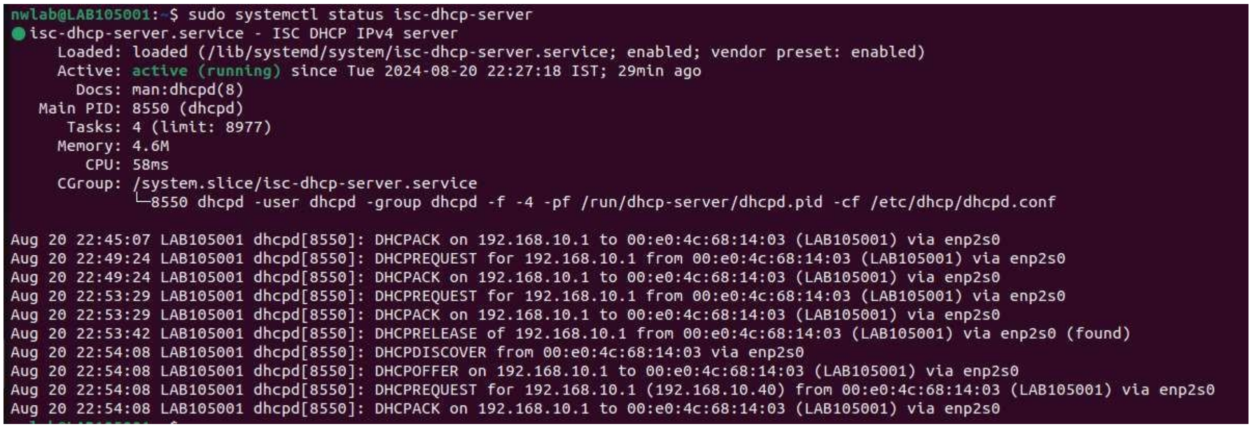
```
Command: sudo systemctl start isc-dhcp-server
```

➔ The ISC DHCP server will start, and it will begin listening on the specified network interface(s) (as defined in the dhcp config file).



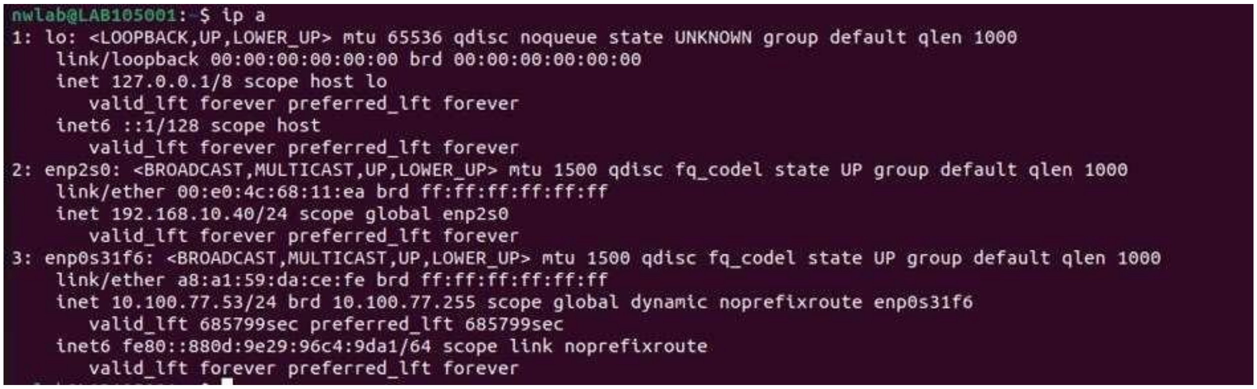
```
nwlab@LAB105001:~$ sudo systemctl start isc-dhcp-server
```

<u>Step-10)</u> Again check the status of the DHCP server as specified in Step-03.



```
nwlab@LAB105001:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
     Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-08-20 22:27:18 IST; 29min ago
       Docs: man:dhcpd(8)
   Main PID: 8550 (dhcpd)
      Tasks: 4 (limit: 8977)
     Memory: 4.6M
        CPU: 58ms
     CGroup: /system.slice/isc-dhcp-server.service
             └─8550 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf

Aug 20 22:45:07 LAB105001 dhcpd[8550]: DHCPACK on 192.168.10.1 to 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:49:24 LAB105001 dhcpd[8550]: DHCPREQUEST for 192.168.10.1 from 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:49:24 LAB105001 dhcpd[8550]: DHCPACK on 192.168.10.1 to 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:53:29 LAB105001 dhcpd[8550]: DHCPREQUEST for 192.168.10.1 from 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:53:29 LAB105001 dhcpd[8550]: DHCPACK on 192.168.10.1 to 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:53:42 LAB105001 dhcpd[8550]: DHCPRELEASE of 192.168.10.1 from 00:e0:4c:68:14:03 (LAB105001) via enp2s0 (found)
Aug 20 22:54:08 LAB105001 dhcpd[8550]: DHCPDISCOVER from 00:e0:4c:68:14:03 via enp2s0
Aug 20 22:54:08 LAB105001 dhcpd[8550]: DHCPOFFER on 192.168.10.1 to 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:54:08 LAB105001 dhcpd[8550]: DHCPREQUEST for 192.168.10.1 (192.168.10.40) from 00:e0:4c:68:14:03 (LAB105001) via enp2s0
Aug 20 22:54:08 LAB105001 dhcpd[8550]: DHCPACK on 192.168.10.1 to 00:e0:4c:68:14:03 (LAB105001) via enp2s0
```

<u>Step-11)</u> Check the IP address for the NIC enp2s0.

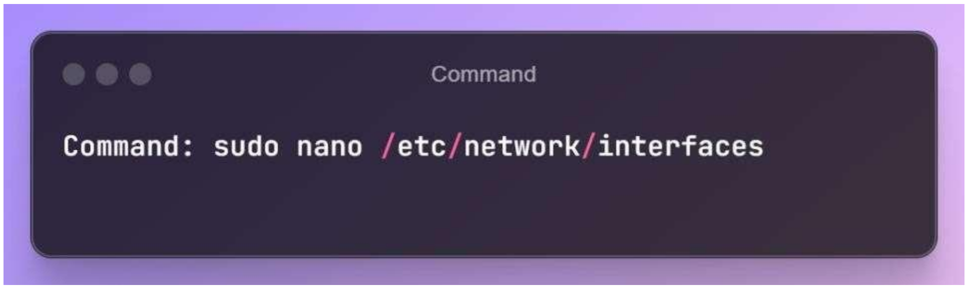


```
nwlab@LAB105001:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:11:ea brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.40/24 scope global enp2s0
       valid_lft forever preferred_lft forever
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:fe brd ff:ff:ff:ff:ff:ff
    inet 10.100.77.53/24 brd 10.100.77.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 685799sec preferred_lft 685799sec
    inet6 fe80::880d:9e29:96c4:9da1/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

➔ Here, an IP will be assigned onto the enp2s0 NIC card automatically by the server.

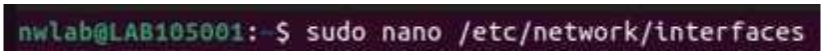
<u>Procedure for Client Side:</u>

<u>Step-01)</u> Configuring the Client side file.



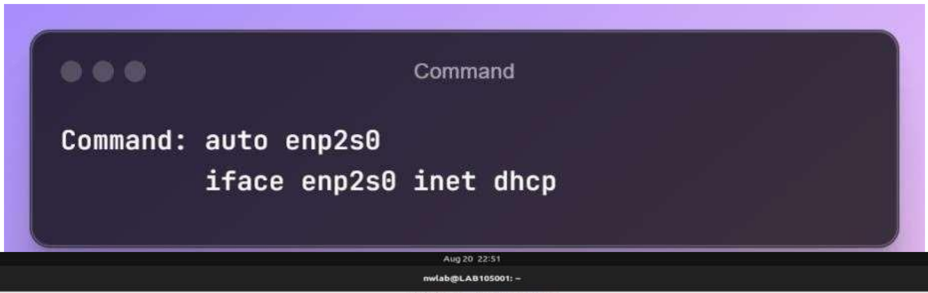
```
Command: sudo nano /etc/network/interfaces
```

➔ This file defines the network settings for your system. For a DHCP client, we need to configure it to obtain an IP address automatically from a DHCP server.

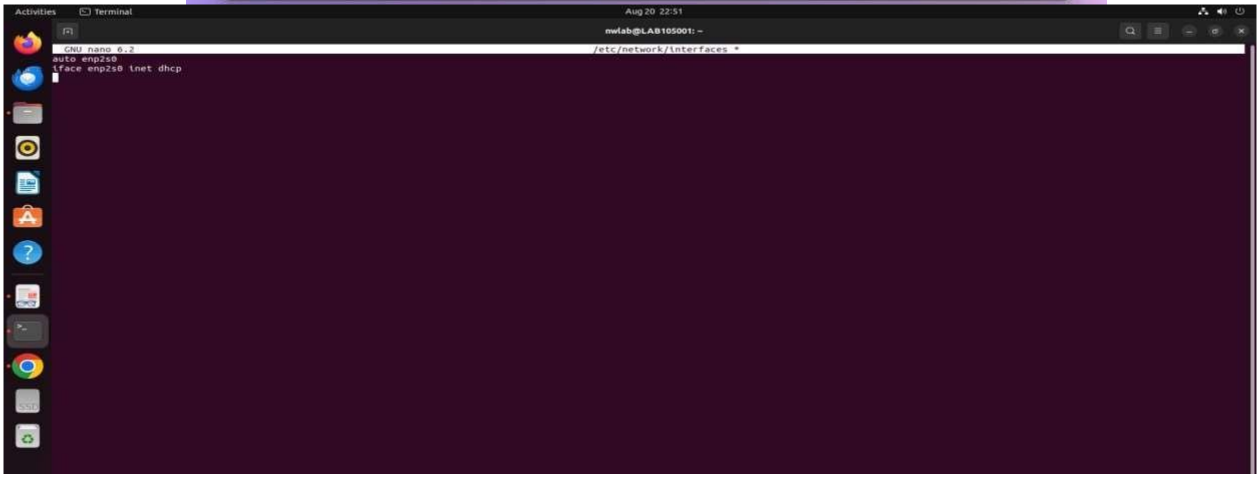


```
nwlab@LAB105001:~$ sudo nano /etc/network/interfaces
```

<u>Step-02)</u> Inserting the commands into the config file.



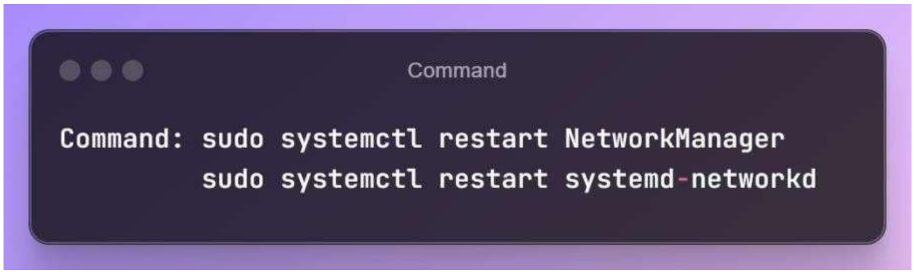
```
Command: auto enp2s0
         iface enp2s0 inet dhcp
```



```
  GNU nano 6.2                                        /etc/network/interfaces *
auto enp2s0
iface enp2s0 inet dhcp
```

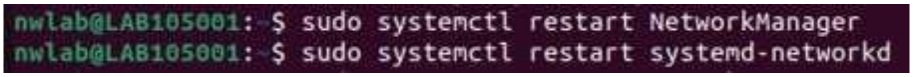
- ➔ The configuration you provided is used to set up a network interface to obtain its IP address dynamically using DHCP.
- ➔ Here, “auto enp2s0” tells the system to automatically bring up the enp2s0 network interface when the system boots.
- ➔ The last line configures the enp2s0 interface to use the DHCP protocol to obtain an IP address automatically from a DHCP server on the network.

Step-03) Restarting Network manager and systemd-networkd services to apply the changes.



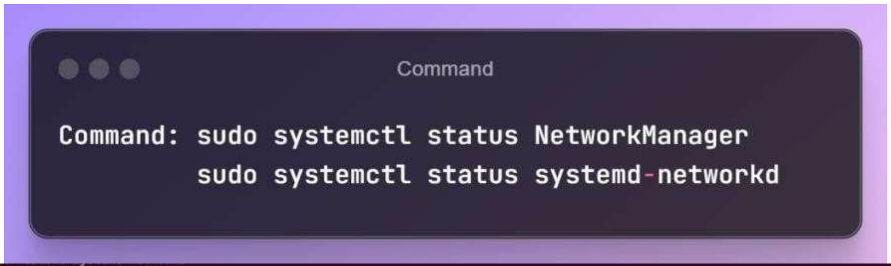
```
Command: sudo systemctl restart NetworkManager
         sudo systemctl restart systemd-networkd
```

- ➔ Restarting Network-Manager will reinitialize the network connections managed by it. This is often done after making changes to network configurations or when troubleshooting connectivity issues.
- ➔ Restarting systemd-networkd will reapply all network configurations managed by this service.

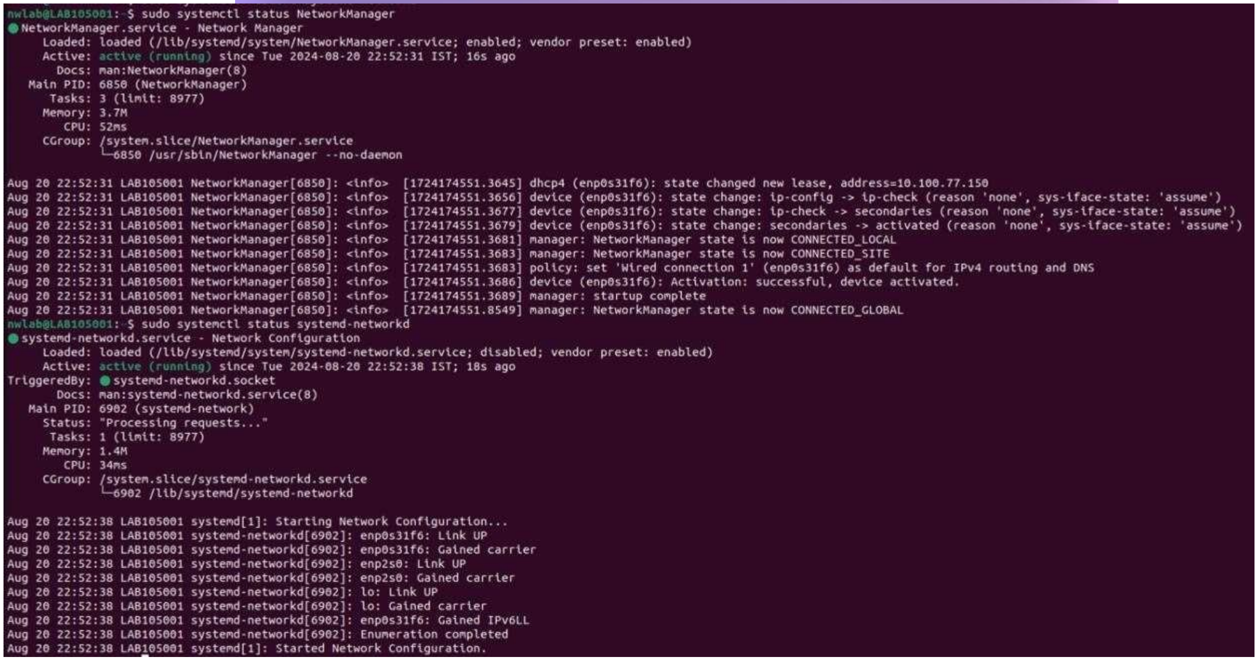


```
nwlab@LAB105001:~$ sudo systemctl restart NetworkManager
nwlab@LAB105001:~$ sudo systemctl restart systemd-networkd
```

Step-04) Checking the Status of Network manager and systemd-networkd services.

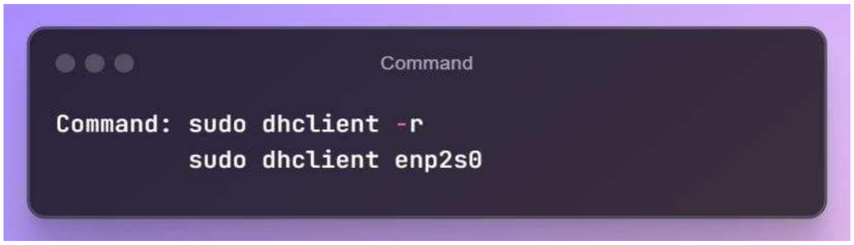


```
Command: sudo systemctl status NetworkManager
         sudo systemctl status systemd-networkd
```



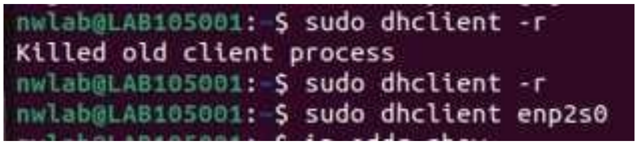
```
nwlab@LAB105001:~$ sudo systemctl status NetworkManager
● NetworkManager.service - Network Manager
     Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-08-20 22:52:31 IST; 16s ago
       Docs: man:NetworkManager(8)
   Main PID: 6850 (NetworkManager)
      Tasks: 3 (limit: 8977)
     Memory: 3.7M
        CPU: 52ms
     CGroup: /system.slice/NetworkManager.service
             └─6850 /usr/sbin/NetworkManager --no-daemon

Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3645] dhcp4 (enp0s31f6): state changed new lease, address=10.100.77.150
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3656] device (enp0s31f6): state change: ip-config -> ip-check (reason 'none', sys-iface-state: 'assume')
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3677] device (enp0s31f6): state change: ip-check -> secondaries (reason 'none', sys-iface-state: 'assume')
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3679] device (enp0s31f6): state change: secondaries -> activated (reason 'none', sys-iface-state: 'assume')
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3681] manager: NetworkManager state is now CONNECTED_LOCAL
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3683] manager: NetworkManager state is now CONNECTED_SITE
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3683] policy: set 'Wired connection 1' (enp0s31f6) as default for IPv4 routing and DNS
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3686] device (enp0s31f6): Activation: successful, device activated.
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.3689] manager: startup complete
Aug 20 22:52:31 LAB105001 NetworkManager[6850]: <info>  [1724174551.8549] manager: NetworkManager state is now CONNECTED_GLOBAL
nwlab@LAB105001:~$ sudo systemctl status systemd-networkd
● systemd-networkd.service - Network Configuration
     Loaded: loaded (/lib/systemd/system/systemd-networkd.service; disabled; vendor preset: enabled)
     Active: active (running) since Tue 2024-08-20 22:52:38 IST; 18s ago
TriggeredBy: ● systemd-networkd.socket
       Docs: man:systemd-networkd.service(8)
   Main PID: 6902 (systemd-network)
     Status: "Processing requests..."
      Tasks: 1 (limit: 8977)
     Memory: 1.4M
        CPU: 34ms
     CGroup: /system.slice/systemd-networkd.service
             └─6902 /lib/systemd/systemd-networkd

Aug 20 22:52:38 LAB105001 systemd[1]: Starting Network Configuration...
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: enp0s31f6: Link UP
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: enp0s31f6: Gained carrier
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: enp2s0: Link UP
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: enp2s0: Gained carrier
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: lo: Link UP
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: lo: Gained carrier
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: enp0s31f6: Gained IPv6LL
Aug 20 22:52:38 LAB105001 systemd-networkd[6902]: Enumeration completed
Aug 20 22:52:38 LAB105001 systemd[1]: Started Network Configuration.
```

Step-05) Manually obtaining the IP address.



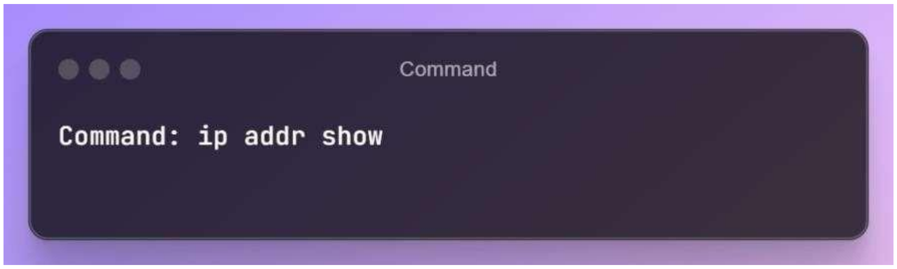
```
Command: sudo dhclient -r
         sudo dhclient enp2s0
```

- ➔ The first command releases the current IP address associated with the network interface, meaning the interface will no longer have an IP address until a new lease is requested.
- ➔ The second command requests a new IP address for the enp2s0 interface from the DHCP server.



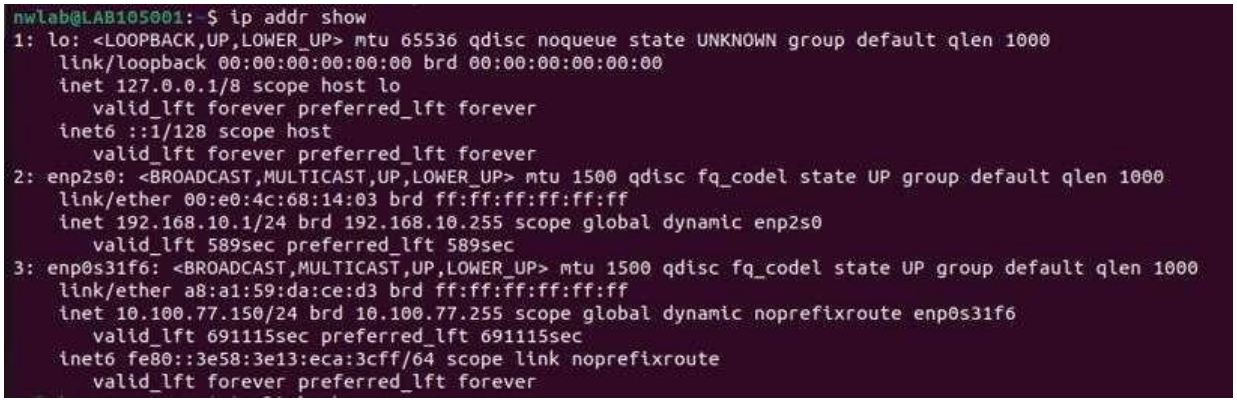
```
nwlab@LAB105001:~$ sudo dhclient -r
Killed old client process
nwlab@LAB105001:~$ sudo dhclient -r
nwlab@LAB105001:~$ sudo dhclient enp2s0
```

Step-06) Verifying the address obtained from DHCP Server.



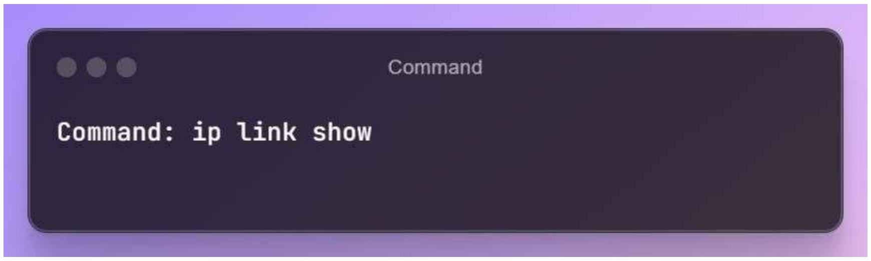
```
Command: ip addr show
```

- ➔ This command is used to display information about the network interfaces on a Linux system, including their IP addresses, MAC addresses, and other related details.
- ➔ Here, the server assigns the IP addresses in sequential manner, starting from 192.168.10.1 to 192.168.10.255.



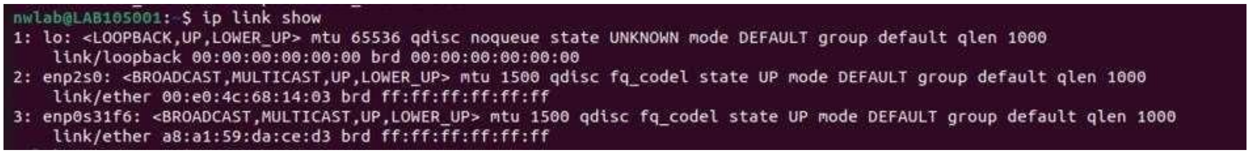
```
nwlab@LAB105001:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:e0:4c:68:14:03 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global dynamic enp2s0
       valid_lft 589sec preferred_lft 589sec
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a8:a1:59:da:ce:d3 brd ff:ff:ff:ff:ff:ff
    inet 10.100.77.150/24 brd 10.100.77.255 scope global dynamic noprefixroute enp0s31f6
       valid_lft 691115sec preferred_lft 691115sec
    inet6 fe80::3e58:3e13:eca:3cff/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Step-07) Checking the Network Interface status.



```
Command: ip link show
```

➔ This command shows the status of the network interface, which can help determine if the interface is up and running. It’s working depends on the network connectivity.



```
nwlab@LAB105001:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: enp2s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 00:e0:4c:68:14:03 brd ff:ff:ff:ff:ff:ff
3: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether a8:a1:59:da:ce:d3 brd ff:ff:ff:ff:ff:ff
```

❖ Sample Questions:- (Part-B)

Q1. What is the difference between static IP address and dynamic IP address?

A1)

Static IP Address:

- **Permanence**: A static IP address is manually assigned to a device and remains constant over time. It does not change unless manually reconfigured by the network administrator.
- **Assignment**: Typically assigned by a network administrator or configured directly on the device.
- **Usage**: Commonly used for devices that require a consistent IP address, such as servers, network printers, and routers, where a constant address is necessary for proper operation and remote access.
- **Reliability**: Offers consistent connectivity and makes it easier to manage services like web hosting, email servers, and remote access because the IP address remains unchanged.
- **Cost**: Often more expensive, especially for external (public) IP addresses, as many Internet Service Providers (ISPs) charge extra for static IPs.

Dynamic IP Address:

- **Permanence**: A dynamic IP address is temporarily assigned to a device from a pool of available addresses and can change each time the device connects to the network.

- **Assignment**: Automatically assigned by a DHCP (Dynamic Host Configuration Protocol) server on the network, making it easier to manage large numbers of devices.
- **Usage**: Commonly used for most client devices like laptops, smartphones, and home networks, where constant connectivity to a specific address is not required.
- **Flexibility**: Easier to manage in environments with many devices, as the DHCP server handles IP address assignment, reducing the likelihood of conflicts.
- **Cost**: Generally included in standard ISP packages, making it less expensive and more commonly used for residential and typical business purposes.

Q2. What do you mean by lease time of dynamically assigned IP address? How is its value governed?

A2) The lease time of a dynamically assigned IP address is the period a device can use the IP before needing to renew or release it.

Governance of Lease Time:

1. **Set by DHCP Server**: Configured by the DHCP server and determines how long an IP address is valid.
2. **Renewal**: The device attempts to renew the lease before it expires.
3. **Impact**: Shorter lease times allow more frequent IP reassignments, while longer times provide stability for devices that stay connected.

Q3. How can one bind an IP address to a MAC address?

A3) Binding an IP address to a MAC address involves creating a static mapping between the two to ensure that a device always receives the same IP address based on its MAC address. This is commonly done in two ways:

1. DHCP Reservation:

- **Access the DHCP Server Configuration**: Log in to the DHCP server's management interface.
- **Add a Reservation**: Create a reservation by entering the device's MAC address and the desired IP address.
- **Save Configuration**: Save the settings. The DHCP server will now always assign the specified IP address to the device with that MAC address.

2. Static ARP Entry:

- **Access Router or Switch**: Log in to the router or switch where you want to set the static binding.
- **Add ARP Entry**: Manually add a static ARP (Address Resolution Protocol) entry mapping the IP address to the MAC address.
- **Save Configuration**: Ensure the static ARP entry is saved and applied.

Both methods ensure that a specific IP address is always associated with a particular MAC address, which can be useful for network management and troubleshooting.