Experiment No : 8

Aim : Introduction to command line tools for networking IPv4 networking, network commands: ping route traceroute, nslookup, ip. Setting up static and dynamic IP addresses. Concept of Subnets, CIDR address schemes, Subnet masks, iptables, setting up a firewall for LAN, Application layer (L7) proxies.

Ipv4 Networking:

The operating system consists of various built-in, command-line networking utilities that are used for network troubleshooting.

IP is part of an internet protocol suite, which also includes the transmission control protocol. Together, these two are known as TCP/IP. The internet protocol suite governs rules for packetizing, addressing, transmitting, routing, and receiving data over networks.

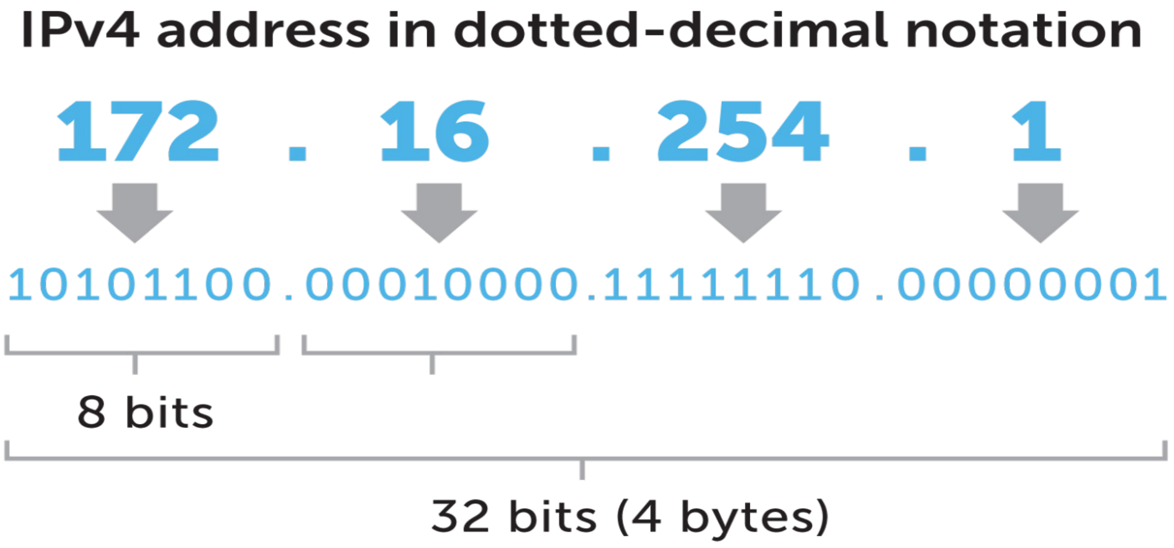
IP addressing is a logical means of assigning addresses to devices on a network. Each device connected to the internet requires a unique IP address.

Most networks that handle internet traffic are packet-switched. Small units of data, called packets, are routed through a network. A source host, like your computer, delivers these IP packets to a destination host, such as a server, based on IP addresses in packet headers. Packet-switching allows many users on a network to share the same data path.

An IP address has two parts—-one part identifies the host, such as a computer or other device. And the other part identifies the network it belongs to. TCP/IP uses a subnet mask to separate them.

* IP (version 4) addresses are 32-bit integers that can be expressed in hexadecimal notation. The more common format, known as dotted quad or dotted decimal, is x.x.x.x, where each x can be any value between 0 and 255. For example, 192.0.2.146 is a valid IPv4 address.

IPv4 still routes most of today’s internet traffic. A 32-bit address space limits the number of unique hosts to 232, which is nearly 4.3 billion IPv4 addresses for the world to use (4,294,967,296, to be exact).

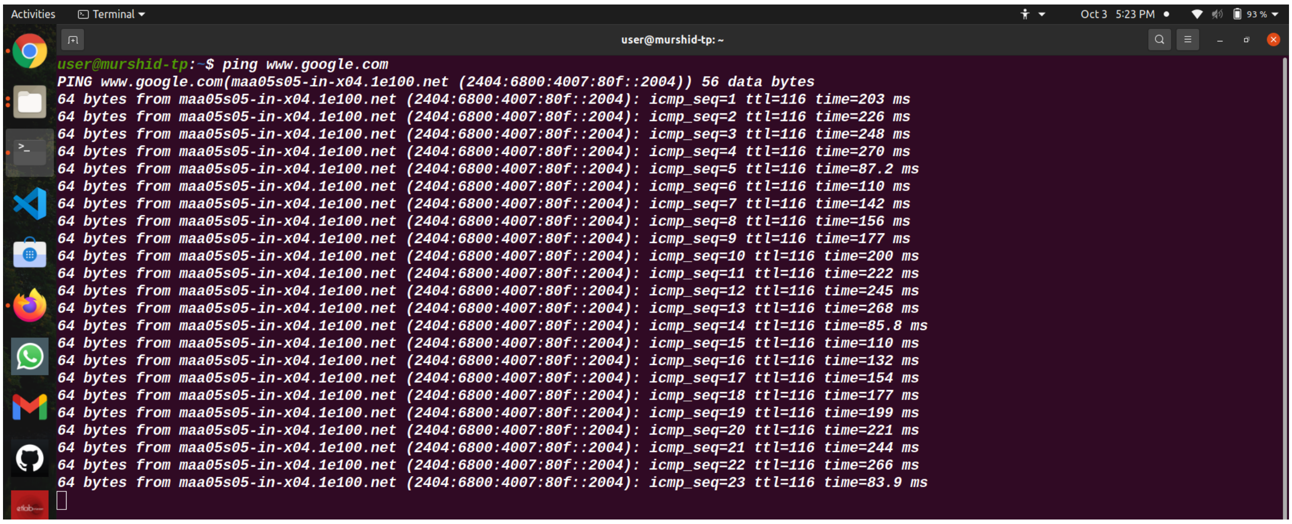


<u>**Network Commands:**</u>

ping: Ping command is typically used for checking the **network connectivity** from your system to an end device like a server or a printer and also of a website. This command is used while troubleshooting the entire network. So, when you enter a URL in your web browser, what you are actually doing is instructing your machine to connect to the website name. The website name is actually an alias for the IP address. So this command can be used in two ways:

1. It can be used to ping a network IP address.

2. It can used to ping a website or hostname directly.



```
user@murshid-tp:~$ ping www.google.com
PING www.google.com(maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004)) 56 data bytes
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=1 ttl=116 time=203 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=2 ttl=116 time=226 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=3 ttl=116 time=248 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=4 ttl=116 time=270 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=5 ttl=116 time=87.2 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=6 ttl=116 time=110 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=7 ttl=116 time=142 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=8 ttl=116 time=156 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=9 ttl=116 time=177 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=10 ttl=116 time=200 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=11 ttl=116 time=222 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=12 ttl=116 time=245 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=13 ttl=116 time=268 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=14 ttl=116 time=85.8 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=15 ttl=116 time=110 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=16 ttl=116 time=132 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=17 ttl=116 time=154 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=18 ttl=116 time=177 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=19 ttl=116 time=199 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=20 ttl=116 time=221 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=21 ttl=116 time=244 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=22 ttl=116 time=266 ms
64 bytes from maa05s05-in-x04.1e100.net (2404:6800:4007:80f::2004): icmp_seq=23 ttl=116 time=83.9 ms
```

Route: Using the route command displays or modifies the computer's routing table. For a typical computer that has a single network interface and is connected to a local area network (LAN) that has a router, the routing table is pretty simple and isn't often the source of network problems. Still, if you're having trouble accessing other computers or other networks, you can use the route command to make sure that a bad entry in the computer's routing table isn't the culprit.

For a computer with more than one interface and that's configured to work as a router, the routing table is often a major source of trouble. Setting up the routing table properly is a key part of configuring a router to work.

<u>Syntax:</u>

```
route [-f] [-p] [command [destination] [mask subnetmask] [gateway] [metric costmetric]]
```

This section explains each of the options that you can use with the route command.

The -f option clears the routing tables of all gateway entries. If you use the -f option in conjunction with one of the commands, the tables are cleared before you run the command.

By default, routes are not preserved when you restart the system. Use the -p option with the add command to make a route persistent. Use the -p option with the print command to view the list of registered persistent routes.

<u>Traceroute</u>

traceroute command in Linux prints the route that a packet takes to reach the host. This command is useful when you want to know about the route and about all the hops that a packet takes.

The first column corresponds to the hop count. The second column represents the address of that hop and after that, you see three space-separated time in milliseconds. *traceroute* command sends three packets to the hop and each of the time refers to the time taken by the packet to reach the hop.

Syntax:

```
traceroute [options]  host_Address [pathlength]
```

Options:

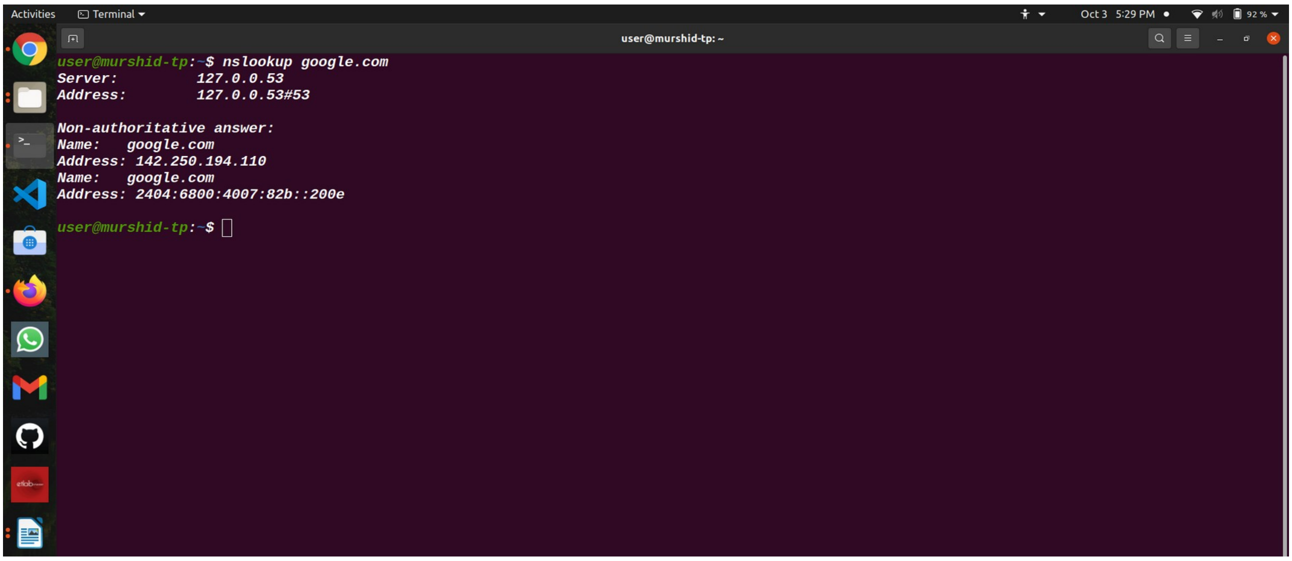
- -4 Option: Use ip version 4 i.e. use Ipv4
- -6 Option: Use ip version 6 i.e. use Ipv6
- -F Option: Do not fragment packet.

Nslookup:

nslookup (stands for “Name Server Lookup”) is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot DNS related problems.

Syntax:

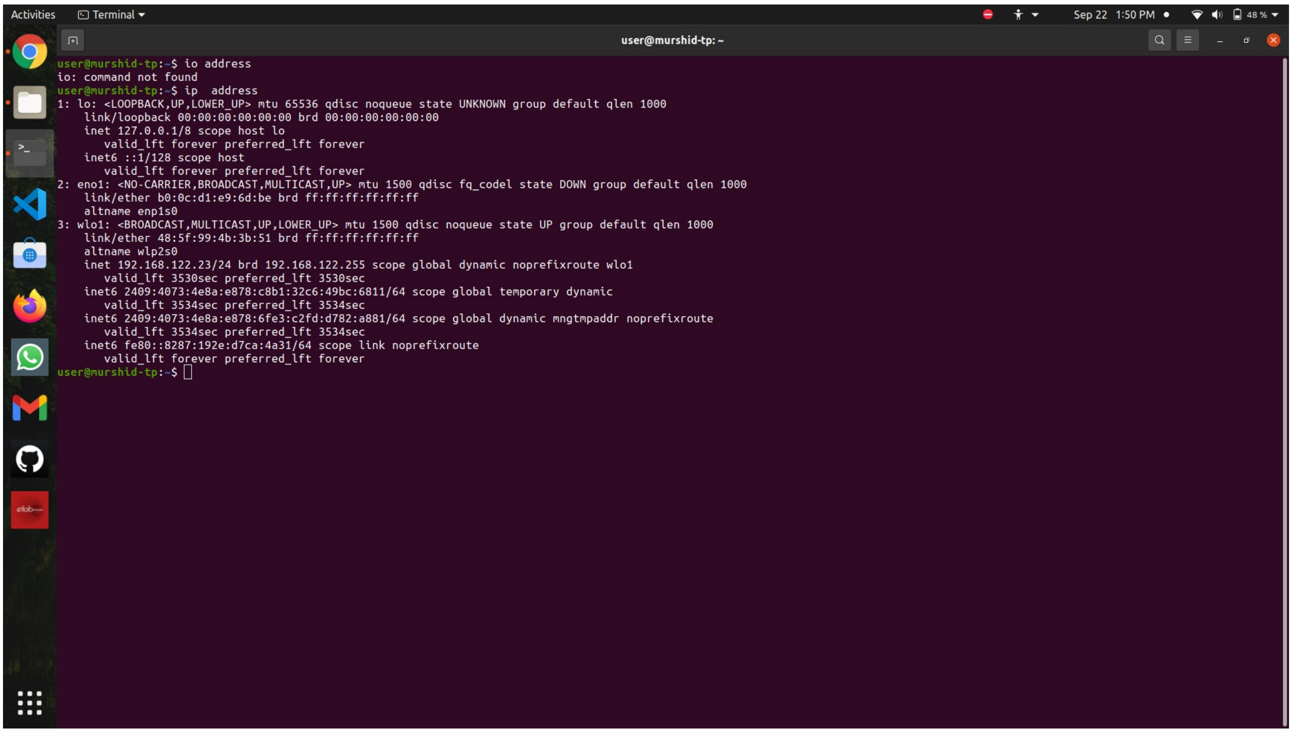
nslookup [option]



```
user@murshid-tp:~$ nslookup google.com
Server:         127.0.0.53
Address:        127.0.0.53#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.194.110
Name:   google.com
Address: 2404:6800:4007:82b::200e

user@murshid-tp:~$ 
```

Setting up static ip address

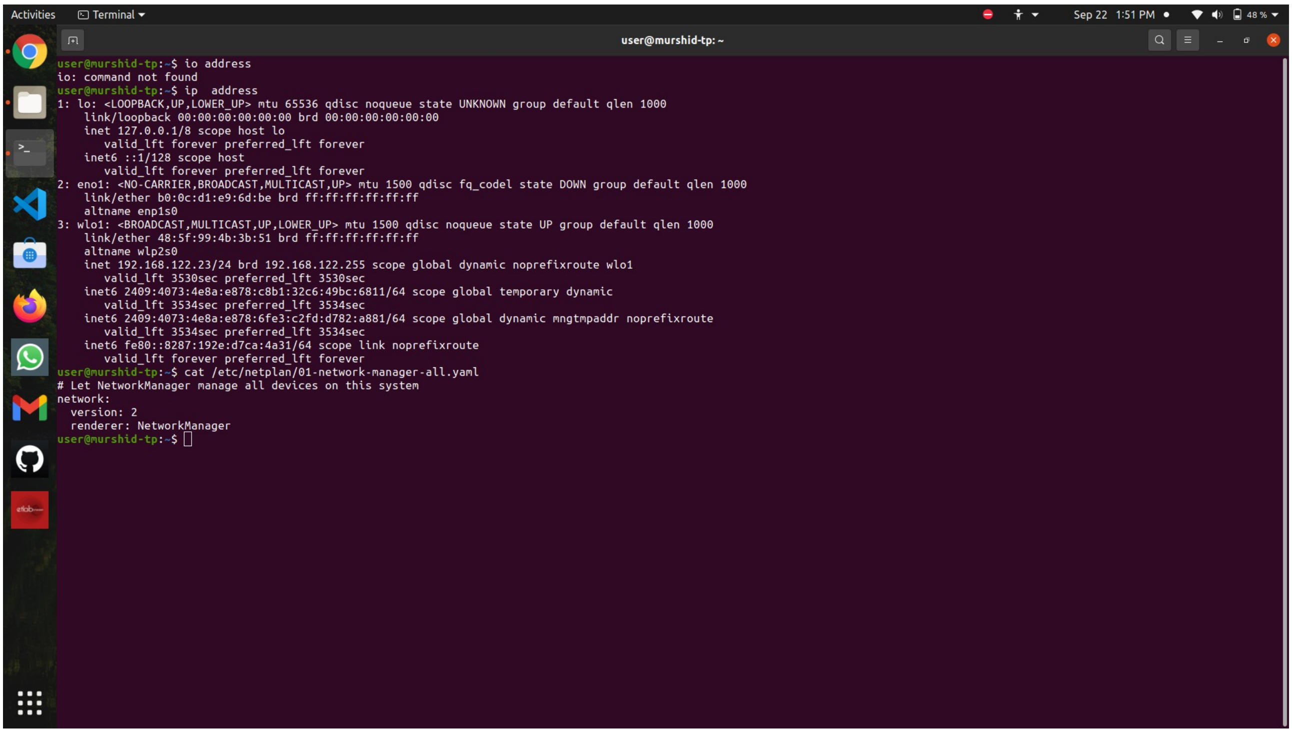
Step 1 : Step 1 : List all the interfaces in the system.Use the ip address command to define a static IP address on an interface.



```
user@murshid-tp:~$ io address
io: command not found
user@murshid-tp:~$ ip  address
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: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether b0:0c:d1:e9:6d:be brd ff:ff:ff:ff:ff:ff
    altname enp1s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 48:5f:99:4b:3b:51 brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.122.23/24 brd 192.168.122.255 scope global dynamic noprefixroute wlo1
       valid_lft 3530sec preferred_lft 3530sec
    inet6 2409:4073:4e8a:e878:c8b1:32c6:49bc:6811/64 scope global temporary dynamic
       valid_lft 3534sec preferred_lft 3534sec
    inet6 2409:4073:4e8a:e878:6fe3:c2fd:d782:a881/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3534sec preferred_lft 3534sec
    inet6 fe80::8287:192e:d7ca:4a31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user@murshid-tp:~$ 
```

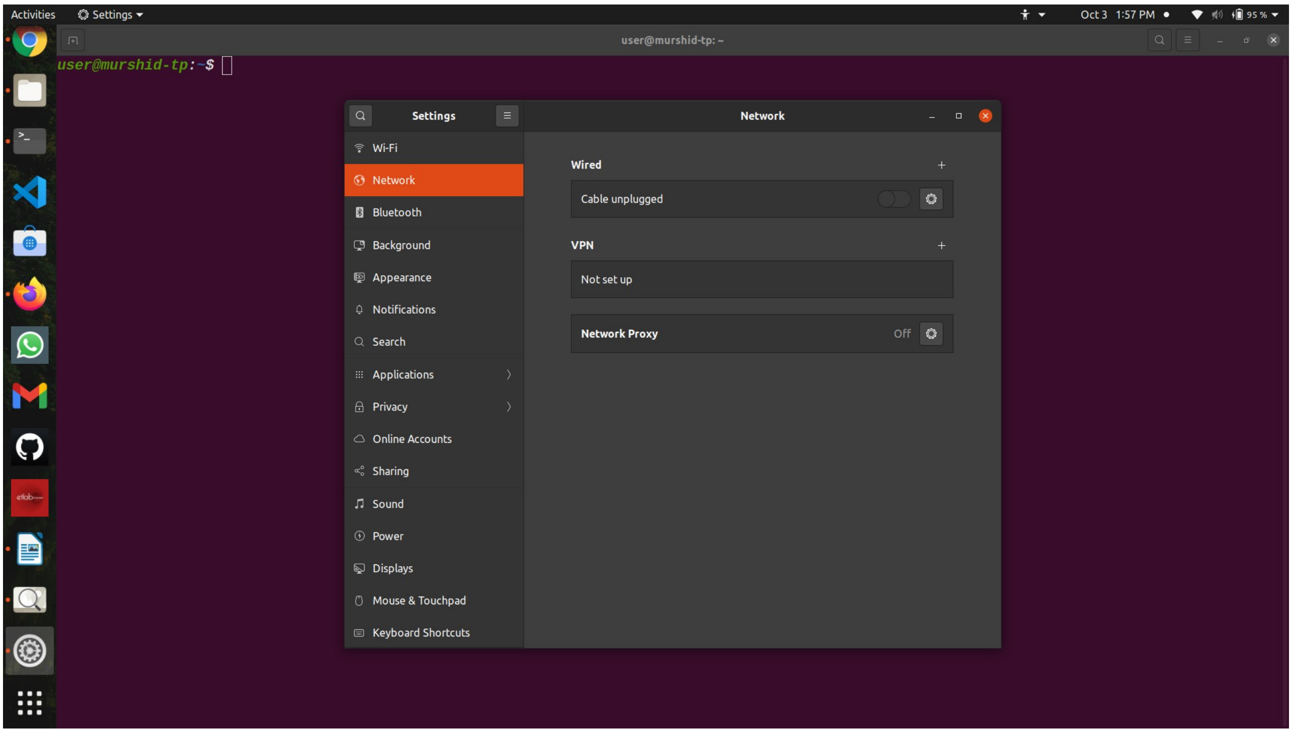
Step 2 : To view the content of Netplan network configuration file, run the following

command:
cat /etc/netplan/01-network-manager-all.yaml



```
user@murshid-tp:~$ io address
io: command not found
user@murshid-tp:~$ ip  address
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: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether b0:0c:d1:e9:6d:be brd ff:ff:ff:ff:ff:ff
    altname enp1s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 48:5f:99:4b:3b:51 brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.122.23/24 brd 192.168.122.255 scope global dynamic noprefixroute wlo1
       valid_lft 3530sec preferred_lft 3530sec
    inet6 2409:4073:4e8a:e878:c8b1:32c6:49bc:6811/64 scope global temporary dynamic
       valid_lft 3534sec preferred_lft 3534sec
    inet6 2409:4073:4e8a:e878:6fe3:c2fd:d782:a881/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3534sec preferred_lft 3534sec
    inet6 fe80::8287:192e:d7ca:4a31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user@murshid-tp:~$ cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
user@murshid-tp:~$ 
```

Step 3 : Click on the top right network icon and select settings of the network interface you wish to configure to use a static IP address on Ubuntu.



```
user@murshid-tp:~$ 
```

Wired

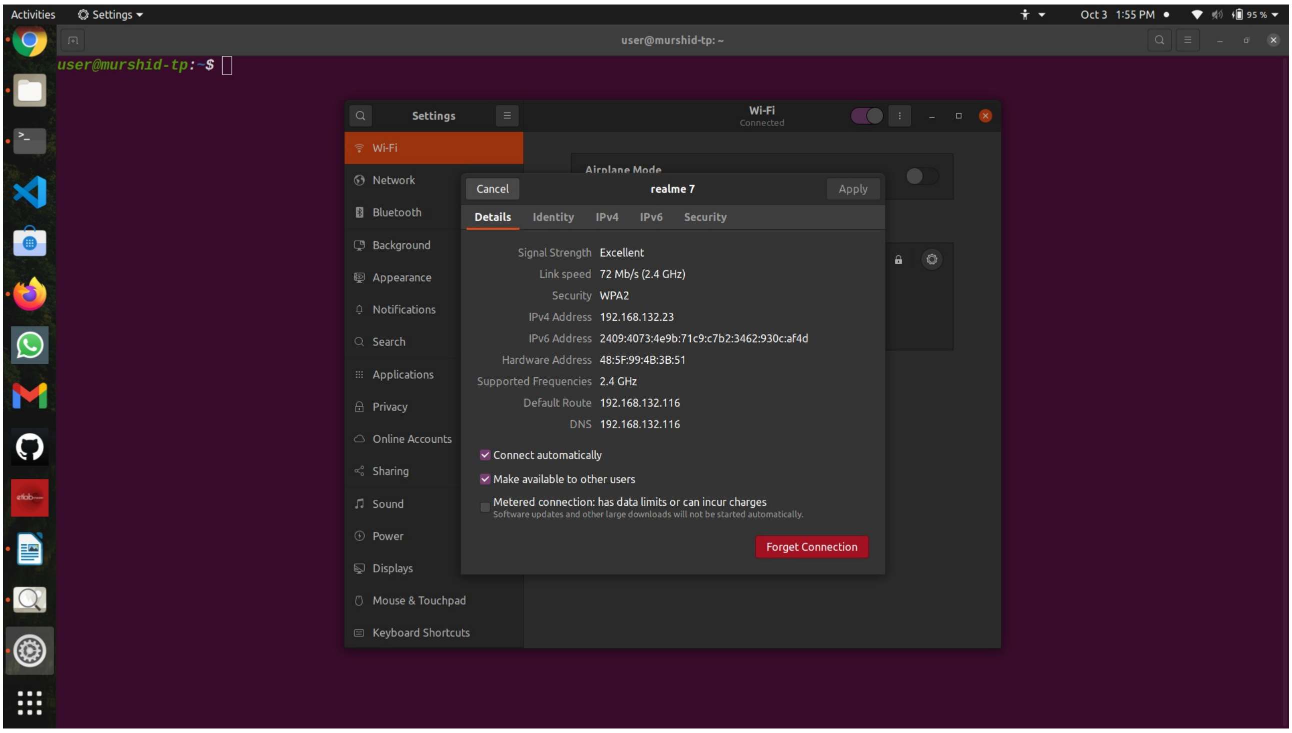
Cable unplugged

VPN

Not set up

Network Proxy Off

Click on the settings icon to start IP address configuration.

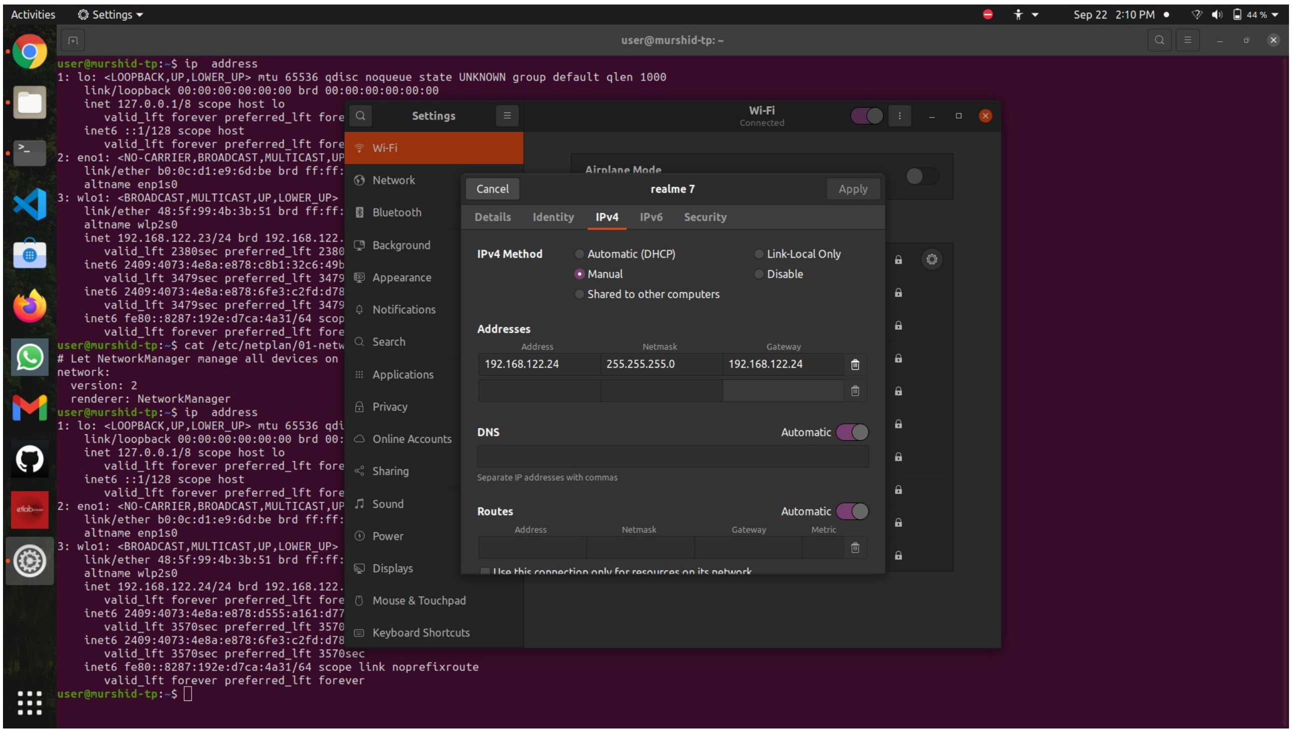


```
user@murshid-tp:~$ 
```

Signal Strength Excellent
Link speed 72 Mb/s (2.4 GHz)
Security WPA2
IPv4 Address 192.168.132.23
IPv6 Address 2409:4073:4e9b:71c9:c7b2:3462:930c:af4d
Hardware Address 48:5F:99:4B:3B:51
Supported Frequencies 2.4 GHz
Default Route 192.168.132.116
DNS 192.168.132.116

☑ Connect automatically
☑ Make available to other users
□ Metered connection: has data limits or can incur charges
Software updates and other large downloads will not be started automatically.

Step 4 : Select IPv4 tab.Select manual and enter your desired IP address, netmask, gateway and DNS settings. Once ready click Apply button.



```
user@murshid-tp:~$ ip  address
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 fore
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft fore
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP
    link/ether b0:0c:d1:e9:6d:be brd ff:ff:
    altname enp1s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP>
    link/ether 48:5f:99:4b:3b:51 brd ff:ff:
    altname wlp2s0
    inet 192.168.122.23/24 brd 192.168.122.
       valid_lft 2380sec preferred_lft 2380
    inet6 2409:4073:4e8a:e878:c8b1:32c6:49b
       valid_lft 3479sec preferred_lft 3479
    inet6 2409:4073:4e8a:e878:6fe3:c2fd:d78
       valid_lft 3479sec preferred_lft 3479
    inet6 fe80::8287:192e:d7ca:4a31/64 scop
       valid_lft forever preferred_lft fore
user@murshid-tp:~$ cat /etc/netplan/01-netw
# Let NetworkManager manage all devices on
network:
  version: 2
  renderer: NetworkManager
user@murshid-tp:~$ ip  address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdi
    link/loopback 00:00:00:00:00:00 brd 00:
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft fore
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft fore
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP
    link/ether b0:0c:d1:e9:6d:be brd ff:ff:
    altname enp1s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP>
    link/ether 48:5f:99:4b:3b:51 brd ff:ff:
    altname wlp2s0
    inet 192.168.122.24/24 brd 192.168.122.
       valid_lft forever preferred_lft fore
    inet6 2409:4073:4e8a:e878:d555:a161:d77
       valid_lft 3570sec preferred_lft 3570
    inet6 2409:4073:4e8a:e878:6fe3:c2fd:d78
       valid_lft 3570sec preferred_lft 3570sec
    inet6 fe80::8287:192e:d7ca:4a31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user@murshid-tp:~$ 
```

IPv4 Method ○ Automatic (DHCP) ○ Link-Local Only
● Manual ○ Disable
○ Shared to other computers

Addresses

Address	Netmask	Gateway
192.168.122.24	255.255.255.0	192.168.122.24

DNS Automatic

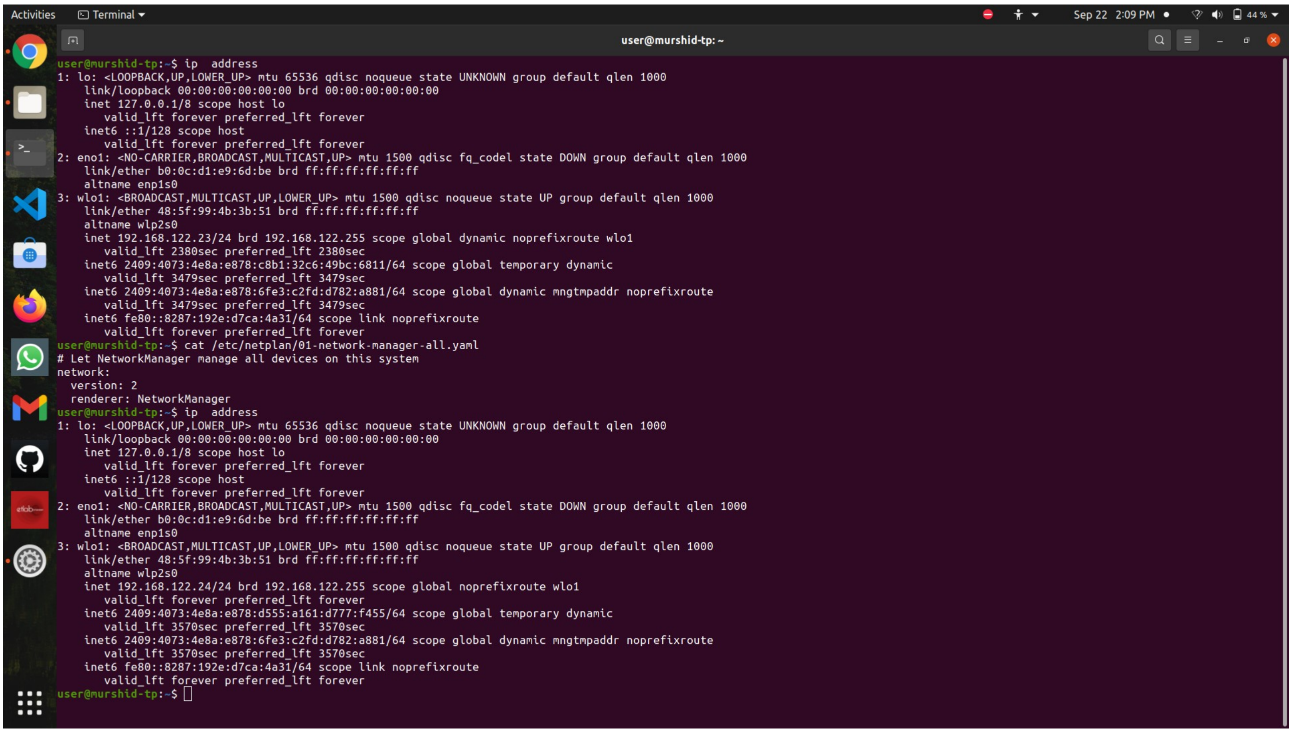
Separate IP addresses with commas

Routes Automatic

Address	Netmask	Gateway	Metric

□ Use this connection only for resources on its network

Step 5 : Turn OFF and ON switch to apply your new network static IP configuration settings.Run the command ip address and click on the network settings icon once again to confirm your new static IP address settings.



```
user@murshid-tp:~$ ip  address
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: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether b0:0c:d1:e9:6d:be brd ff:ff:ff:ff:ff:ff
    altname enp1s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 48:5f:99:4b:3b:51 brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.122.23/24 brd 192.168.122.255 scope global dynamic noprefixroute wlo1
       valid_lft 2380sec preferred_lft 2380sec
    inet6 2409:4073:4e8a:e878:c8b1:32c6:49bc:6811/64 scope global temporary dynamic
       valid_lft 3479sec preferred_lft 3479sec
    inet6 2409:4073:4e8a:e878:6fe3:c2fd:d782:a881/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3479sec preferred_lft 3479sec
    inet6 fe80::8287:192e:d7ca:4a31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user@murshid-tp:~$ cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
user@murshid-tp:~$ ip  address
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: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether b0:0c:d1:e9:6d:be brd ff:ff:ff:ff:ff:ff
    altname enp1s0
3: wlo1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 48:5f:99:4b:3b:51 brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.122.24/24 brd 192.168.122.255 scope global noprefixroute wlo1
       valid_lft forever preferred_lft forever
    inet6 2409:4073:4e8a:e878:d555:a161:d777:f455/64 scope global temporary dynamic
       valid_lft 3570sec preferred_lft 3570sec
    inet6 2409:4073:4e8a:e878:6fe3:c2fd:d782:a881/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 3570sec preferred_lft 3570sec
    inet6 fe80::8287:192e:d7ca:4a31/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
user@murshid-tp:~$ 
```

Static IP Addresses

A static IP address is an IP address that always stays the same. If you have a web server, FTP server, or other Internet resource that must have an address that cannot change, you can get a static IP address from your ISP. A static IP address is usually more expensive than a dynamic IP address, and some ISPs do not supply static IP addresses. You must configure a static IP address manually.

Dynamic IP Addresses

A dynamic IP address is an IP address that an ISP lets you use temporarily. If a dynamic address is not in use, it can be automatically assigned to a different device. Dynamic IP addresses are assigned using either DHCP or PPPoE.

Subnet

A subnet, or subnetwork, is a segmented piece of a larger network. More specifically, subnets are a logical partition of an IP network into multiple, smaller network segments. The Internet Protocol (IP) is the method for sending data from one computer to another over the internet. Each computer, or host, on the internet has at least one IP address as a unique identifier.

CIDR

CIDR stands for **Classless Inter-Domain Routing**. It is an IP address assigning method that improves the efficiency of address distribution. It is also known as supernetting that replaces the older system based on classes A, B, and C networks. By using a single CIDR IP address many unique IP addresses can be designated. CIDR IP address is the same as the normal IP address except that it ends with a slash followed by a number.
172.200.0.0/16 It is called IP network prefix.

Characteristics of CIDR

It dynamically allocates the IP addresses by using CIDR blocks on the requirement of the user based on certain rules. The assignment of the CIDR block is handled by the Internet Assigned Number Authority (IANA). CIDR block consists of IP addresses and it consists of some rules:

- All IP addresses which are allocated to host must be continuous.
- The block size must be of power 2 and equal to the total number of IP addresses.
- The size of the block must be divisible by the first IP address of the block.

Subnet Mask

A subnet mask is a 32 bits address used to distinguish between a network address and a host address in IP address. A subnet mask identifies which part of an IP address is the network address and the host address. They are not shown inside the data packets traversing the Internet. They carry the destination IP address, which a router will match with a subnet.

iptables

iptables is a user-space utility program that allows a system administrator to configure the IP packet filter rules of the Linux kernel firewall, implemented as different Netfilter modules. The filters are organized in different tables, which contain chains of rules for how to treat network traffic packets. Different kernel modules and programs are currently used for different protocols; iptables applies to IPv4, ip6tables to IPv6, arptables to ARP, and ebtables to Ethernet frames.

Configure and Set Up a Firewall on Ubuntu

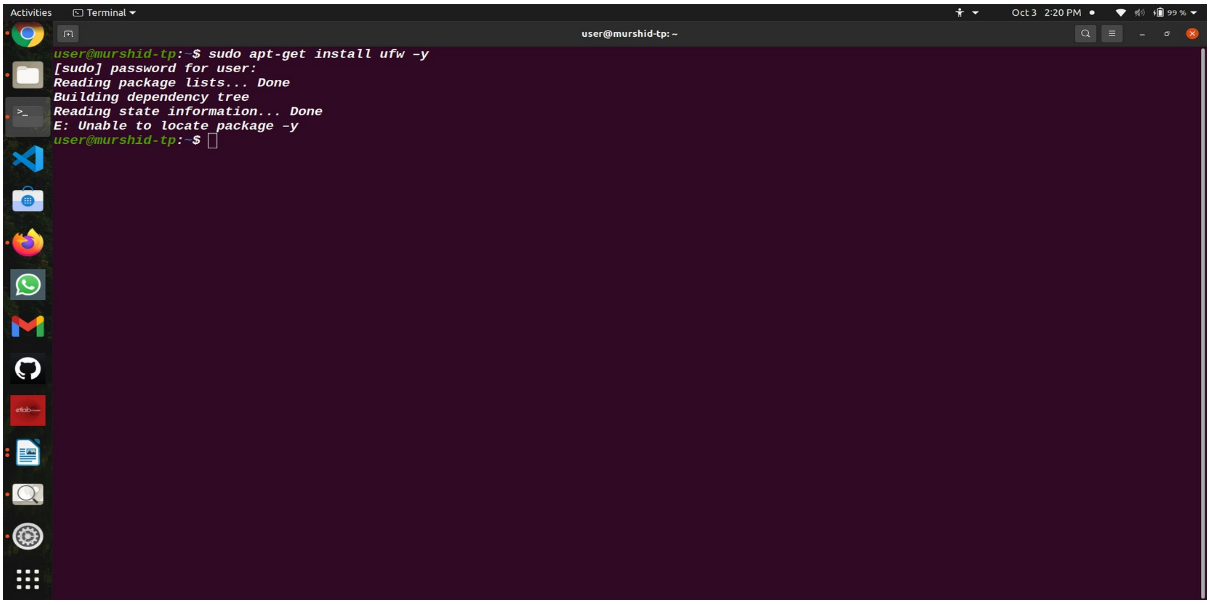
UFW stands for Uncomplicated Firewall which acts as an interface to IPTABLES that simplifies the process of the configuration of firewalls it will be a very hard for a beginners to learns and configure the firewall rules where we will secure the network from unknown users are machines. UFW works on the policies we configure as rules.

- For this, we needed a non-root user with root permission on the machine.

Installing the UFW (Firewall)

UFW is installed by default with Ubuntu, if not installed then we will install them using the below command –

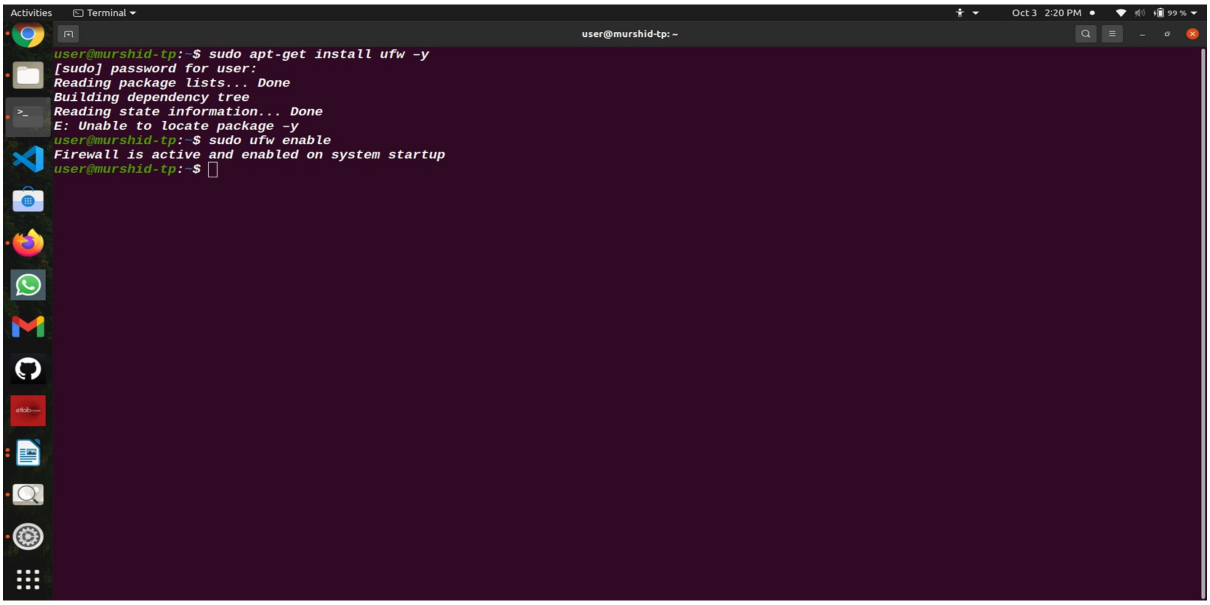
sudo apt-get install ufw –y



```
user@murshid-tp:~$ sudo apt-get install ufw –y
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package –y
user@murshid-tp:~$
```

Enabling the UFW (Firewall)

Below is the command to enable the UFW –
sudo ufw enable

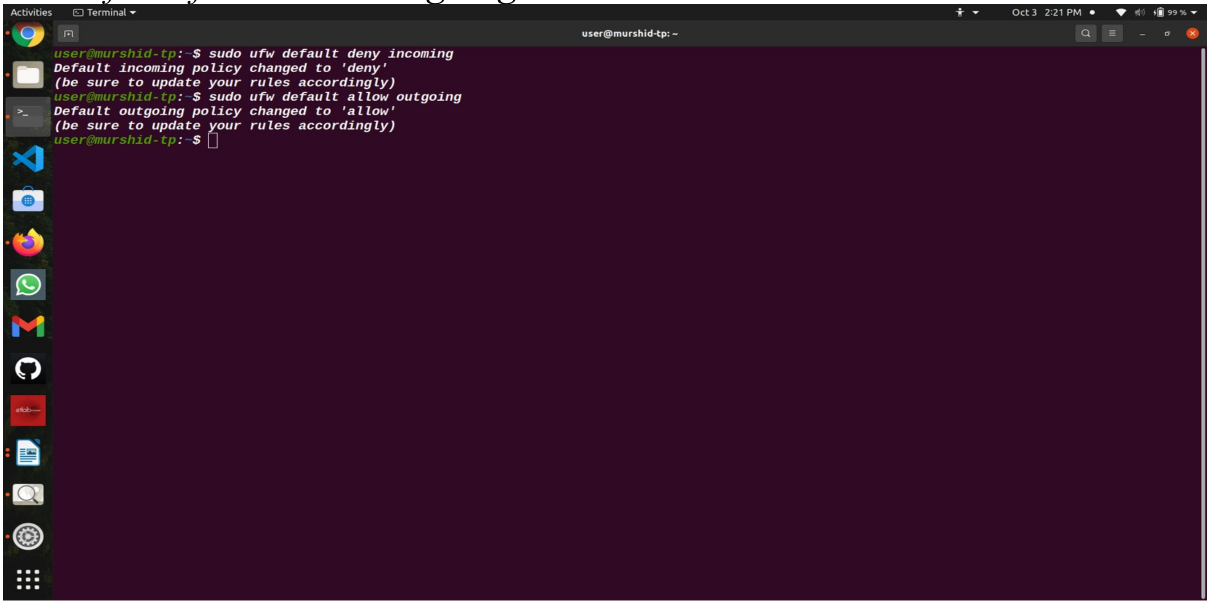


```
user@murshid-tp:~$ sudo apt-get install ufw –y
[sudo] password for user:
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package –y
user@murshid-tp:~$ sudo ufw enable
Firewall is active and enabled on system startup
user@murshid-tp:~$ 
```

Enabling the Default Policies

As the beginner, we will first configure default policies, which control and handles the traffic which will not match the other rules. By default, the rules will deny all incoming connections and allow all outgoing connections will be allowed which stops someone trying to reach the machine from the internet world.

sudo ufw default deny incoming
sudo ufw default allow outgoing

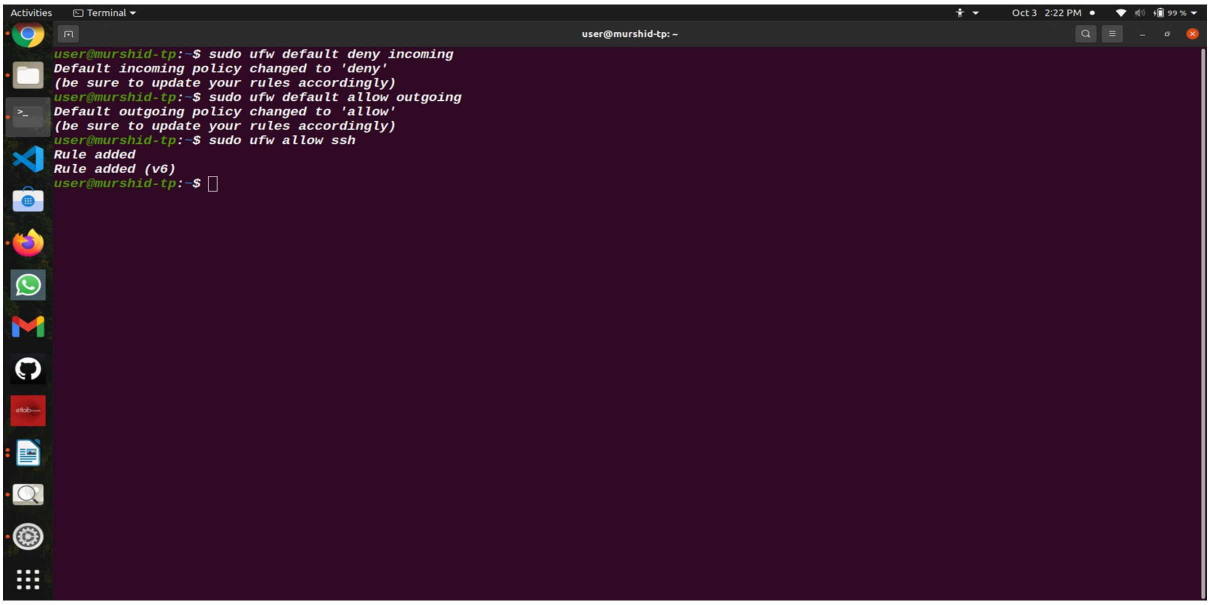


```
user@murshid-tp:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
user@murshid-tp:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
user@murshid-tp:~$ 
```

Enabling SSH Connections

Using the above commands, we have disabled all the incoming connections, it will deny all the incoming connections, we needed to create a rule which will explicitly allow the SSH incoming connection.Below is the command to enable the incoming connection for SSH.

sudo ufw allow ssh



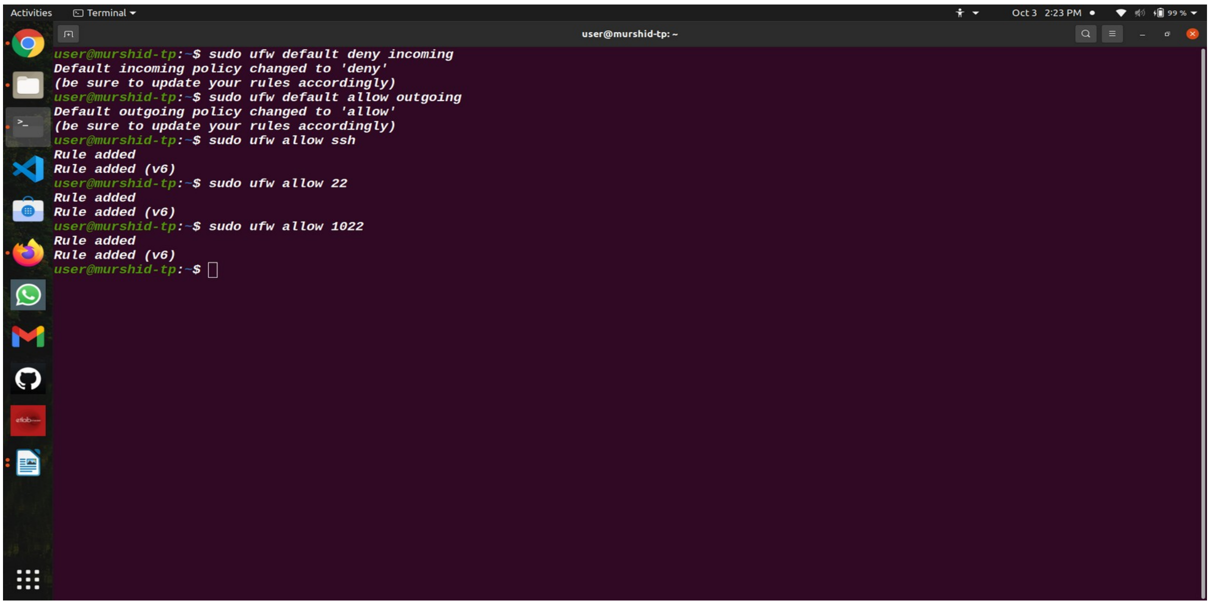
```
user@murshid-tp:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
user@murshid-tp:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
user@murshid-tp:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
user@murshid-tp:~$ 
```

With the above command, the port 22 will be allowed for incoming connections. We can use the below command directly using the port no 22 to allow the SSH connections.

sudo ufw allow 22

However, if we have configured the SSH daemon to use a different port like 2022 or 1022, then we can use the below command –

sudo ufw allow 1022

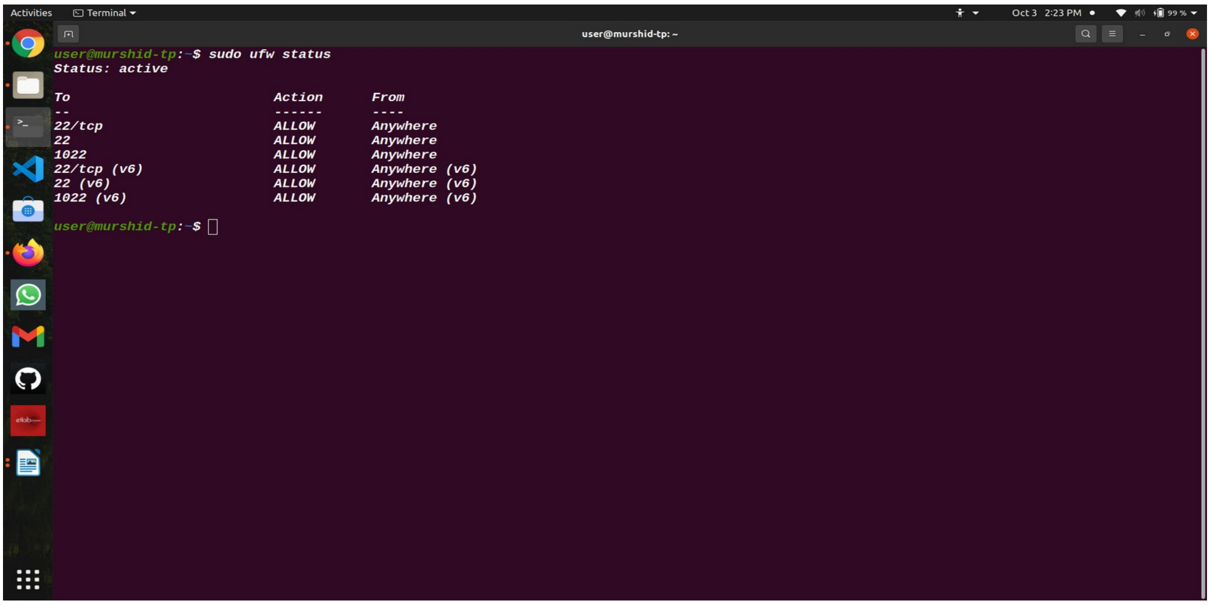


```
user@murshid-tp:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
user@murshid-tp:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
user@murshid-tp:~$ sudo ufw allow ssh
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw allow 22
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw allow 1022
Rule added
Rule added (v6)
user@murshid-tp:~$ 
```

Checking the UFW (Firewall) Status

Below is the command to check the current status of the firewall rules.

sudo ufw status



```
user@murshid-tp:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22                         ALLOW       Anywhere
1022                       ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
1022 (v6)                  ALLOW       Anywhere (v6)

user@murshid-tp:~$
```

Enabling the UFW for regular port like (HTTP, HTTPS & FTP)

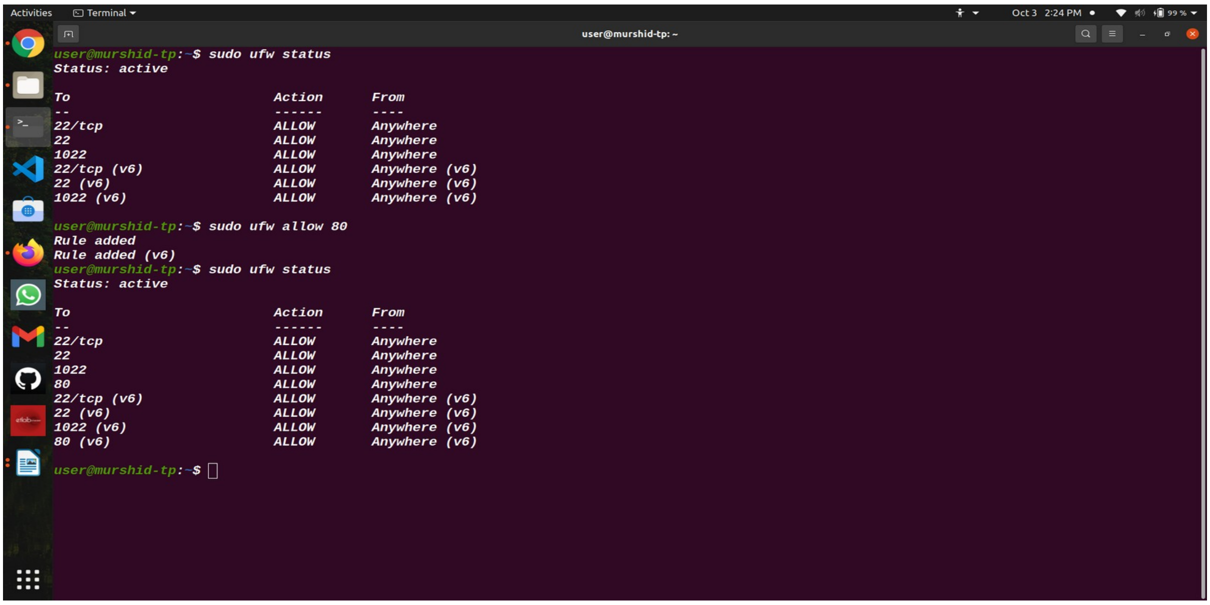
At this point, we will allow others to connect to the server for the regular ports like HTPP, HTTPS, and FTP ports respectively.

HTTP port 80

sudo ufw allow 80

We can check the UFW (Firewall) status using the below command

sudo ufw status

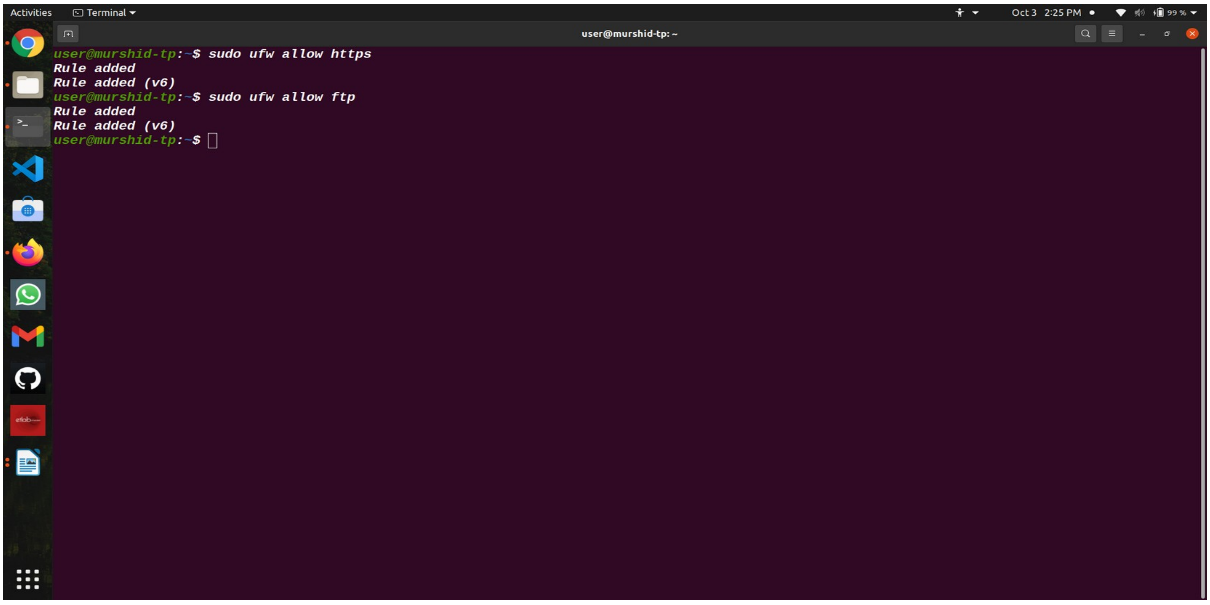


```
user@murshid-tp:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22                         ALLOW       Anywhere
1022                       ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
1022 (v6)                  ALLOW       Anywhere (v6)

user@murshid-tp:~$ sudo ufw allow 80
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW       Anywhere
22                         ALLOW       Anywhere
1022                       ALLOW       Anywhere
80                         ALLOW       Anywhere
22/tcp (v6)                ALLOW       Anywhere (v6)
22 (v6)                    ALLOW       Anywhere (v6)
1022 (v6)                  ALLOW       Anywhere (v6)
80 (v6)                    ALLOW       Anywhere (v6)

user@murshid-tp:~$
```

Like that will use the below command to enable HTTPs and FTP ports (443 and 21) respectively.

sudo ufw allow https

sudo ufw allow ftp

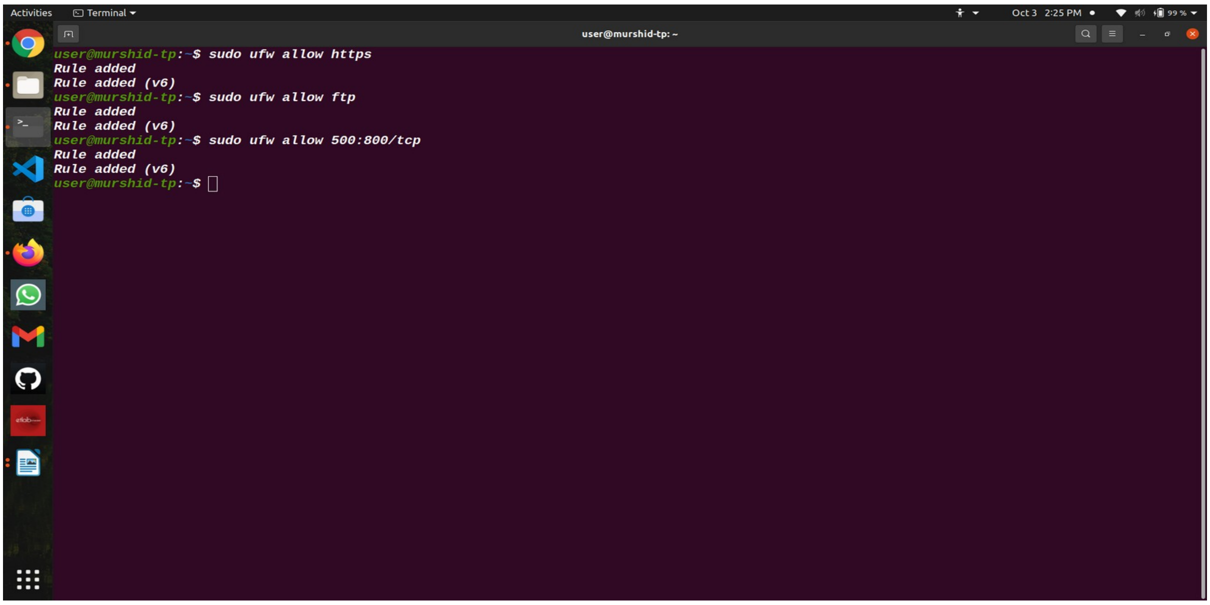


```
user@murshid-tp:~$ sudo ufw allow https
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw allow ftp
Rule added
Rule added (v6)
user@murshid-tp:~$
```

Enabling to Allow Specific Range of Ports

We can also allow or deny particular ranges of ports with UFW to allow the multiple ports instead of allowing single ports.Below is the command to enable a specific range of ports.

sudo ufw allow 500:800/tcp



```
user@murshid-tp:~$ sudo ufw allow https
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw allow ftp
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw allow 500:800/tcp
Rule added
Rule added (v6)
user@murshid-tp:~$
```

Enable to Allow specific IP Addresses

If we want to allow a particular machine to allow for all the ports. We can use the below command.

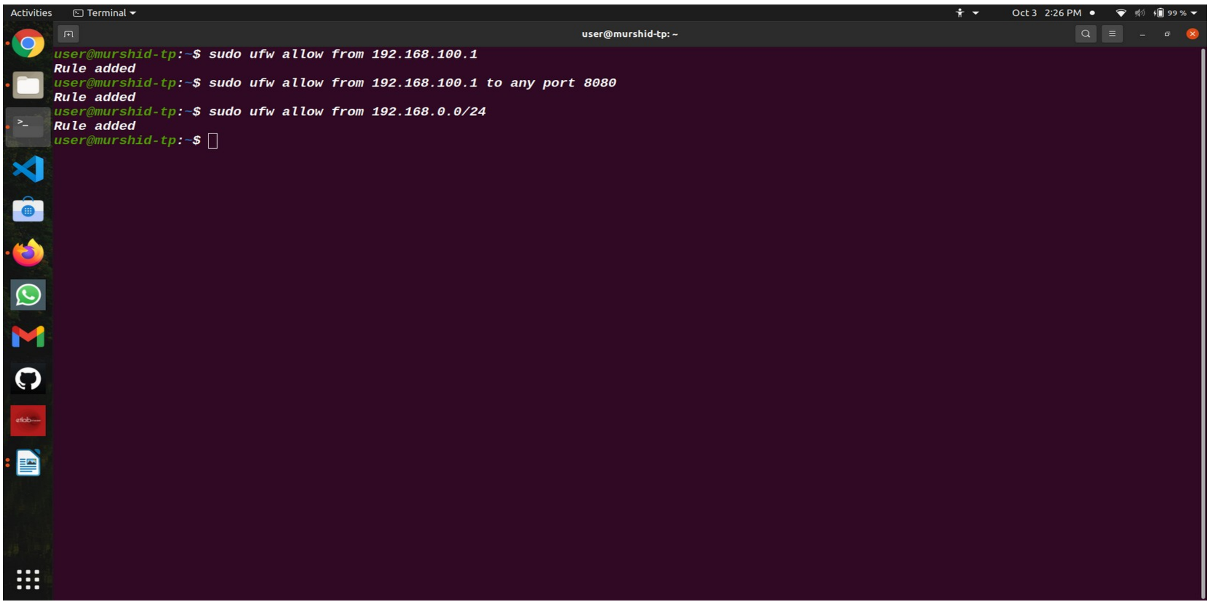
sudo ufw allow from 192.168.100.1

If we want to allow for only specific port we can use the below command.

sudo ufw allow from 192.168.100.1 to any port 8080

If we want to enable the specific subnets like we want to enable for office networks we can use the below command.

sudo ufw allow from 192.168.0.0/24



```
user@murshid-tp:~$ sudo ufw allow from 192.168.100.1
Rule added
user@murshid-tp:~$ sudo ufw allow from 192.168.100.1 to any port 8080
Rule added
user@murshid-tp:~$ sudo ufw allow from 192.168.0.0/24
Rule added
user@murshid-tp:~$ 
```

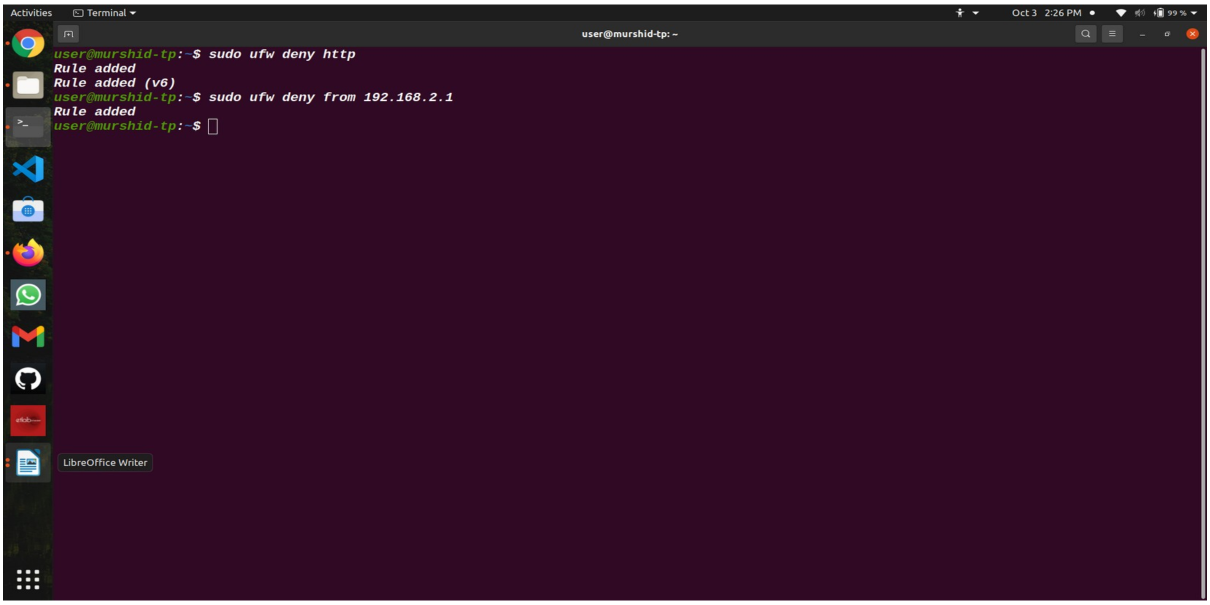
Deny the Connections or Rules

If we want to deny any ports or network we can use the below commands to deny the connections.

sudo ufw deny http

If we want to deny all the connects from a specific network we can use the below command.

sudo ufw deny from 192.168.2.1



```
user@murshid-tp:~$ sudo ufw deny http
Rule added
Rule added (v6)
user@murshid-tp:~$ sudo ufw deny from 192.168.2.1
Rule added
user@murshid-tp:~$ 
```

Deleting the Rules

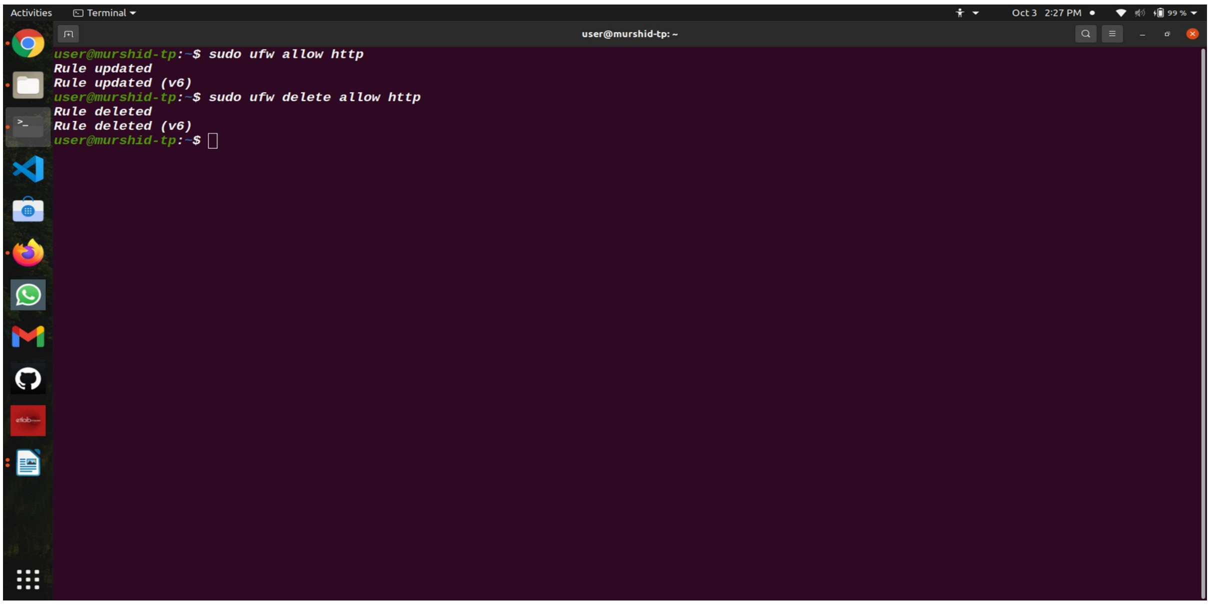
We can delete the rules in two ways one with the actual rules and other with the rules numbers.

Actual Rules

The rules can be deleted using the actual rule which we allowed using the allow command. Below is the command to delete the HTTP rules from UFW.

sudo ufw allow http

sudo ufw delete allow http

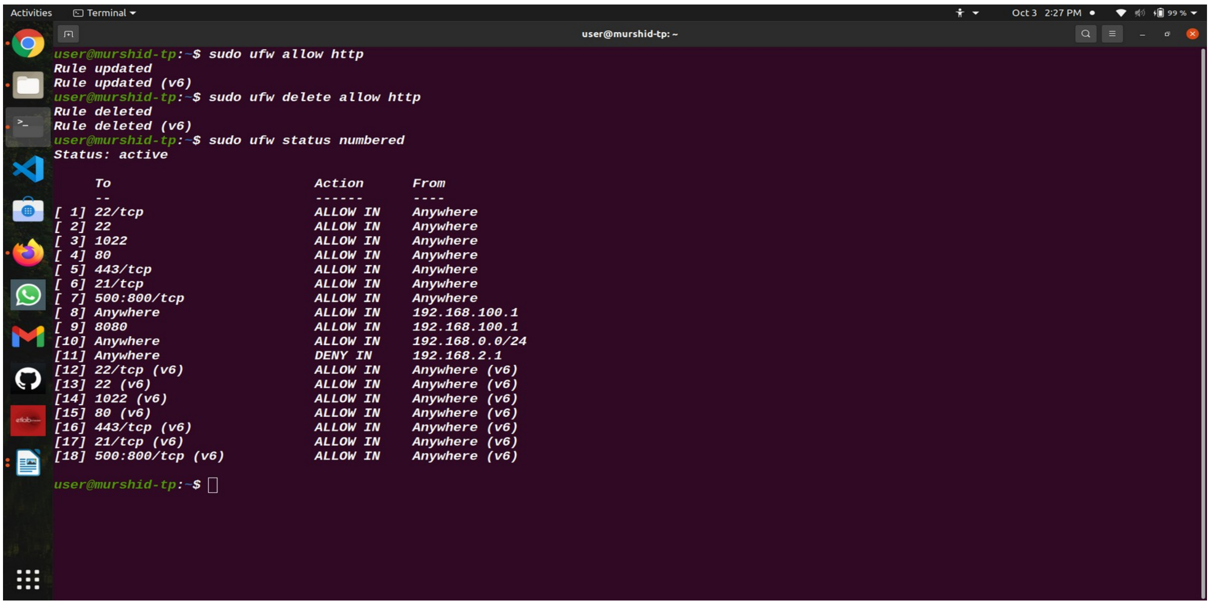


```
user@murshid-tp:~$ sudo ufw allow http
Rule updated
Rule updated (v6)
user@murshid-tp:~$ sudo ufw delete allow http
Rule deleted
Rule deleted (v6)
user@murshid-tp:~$ 
```

Rules Number

We can use the Rules numbers to delete the firewall rules, we can get the list of firewall rules with the below command.

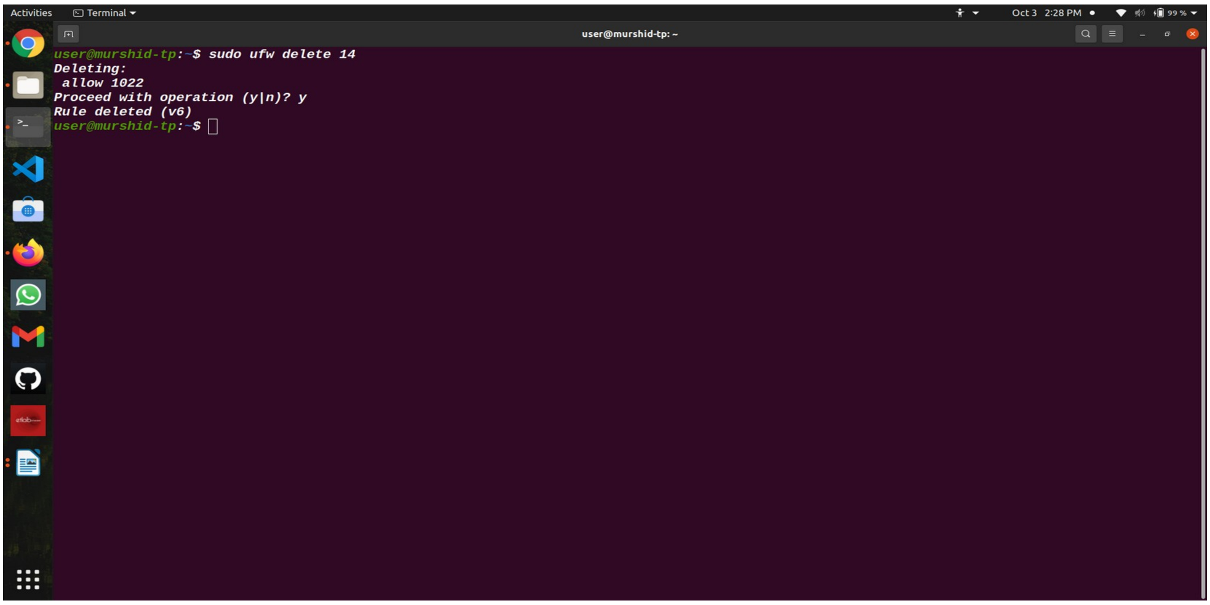
sudo ufw status numbered



```
user@murshid-tp:~$ sudo ufw allow http
Rule updated
Rule updated (v6)
user@murshid-tp:~$ sudo ufw delete allow http
Rule deleted
Rule deleted (v6)
user@murshid-tp:~$ sudo ufw status numbered
Status: active

     To                         Action      From
     --                         ------      ----
[ 1] 22/tcp                     ALLOW IN    Anywhere
[ 2] 22                         ALLOW IN    Anywhere
[ 3] 1022                       ALLOW IN    Anywhere
[ 4] 80                         ALLOW IN    Anywhere
[ 5] 443/tcp                    ALLOW IN    Anywhere
[ 6] 21/tcp                     ALLOW IN    Anywhere
[ 7] 500:800/tcp                ALLOW IN    Anywhere
[ 8] Anywhere                   ALLOW IN    192.168.100.1
[ 9] 8080                       ALLOW IN    192.168.100.1
[10] Anywhere                   ALLOW IN    192.168.0.0/24
[11] Anywhere                   DENY IN     192.168.2.1
[12] 22/tcp (v6)                ALLOW IN    Anywhere (v6)
[13] 22 (v6)                    ALLOW IN    Anywhere (v6)
[14] 1022 (v6)                  ALLOW IN    Anywhere (v6)
[15] 80 (v6)                    ALLOW IN    Anywhere (v6)
[16] 443/tcp (v6)               ALLOW IN    Anywhere (v6)
[17] 21/tcp (v6)                ALLOW IN    Anywhere (v6)
[18] 500:800/tcp (v6)           ALLOW IN    Anywhere (v6)

user@murshid-tp:~$
```

If we want to delete the rule 14, then we can use the below command to delete the rules with the below command.

sudo ufw delete 14



```
user@murshid-tp:~$ sudo ufw delete 14
Deleting:
 allow 1022
Proceed with operation (y|n)? y
Rule deleted (v6)
user@murshid-tp:~$
```