

## EXPIRIMENT 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.

### Solution :-

The network infrastructure is a very complex structure of cables, routers, access points, data packets and a million other small components that together make the entire network work seamlessly. Any issue in any of these smaller components may lead to an overall collapse of the network infrastructure. This may lead to disruption of WiFi, cellular and wired(ethernet) infrastructure. This is the reason why it is very important to have an access to how the network is performing and know troubleshooting techniques.

The operating system acts as an intermediate platform between the user and the underlying network infrastructure. To use the below commands in Windows operating system, one needs to click on Start, go to Run and type cmd. This will open up the command prompt. In Mac OS, you can use the terminal application.

### 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.

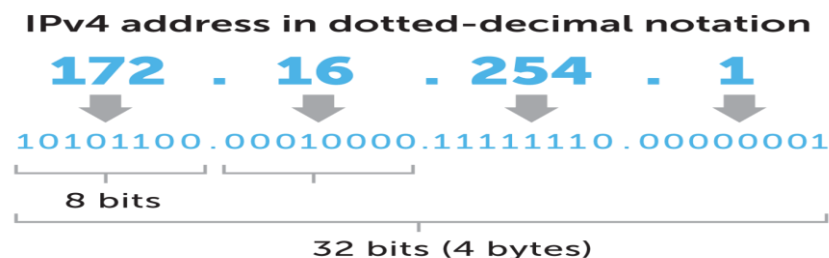
IP addressing is a logical means of assigning addresses to devices on a network. Each device connected to the internet requires a unique IP address.

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 2<sup>32</sup>, which is nearly 4.3 billion IPv4 addresses for the world to use (4,294,967,296, to be exact).

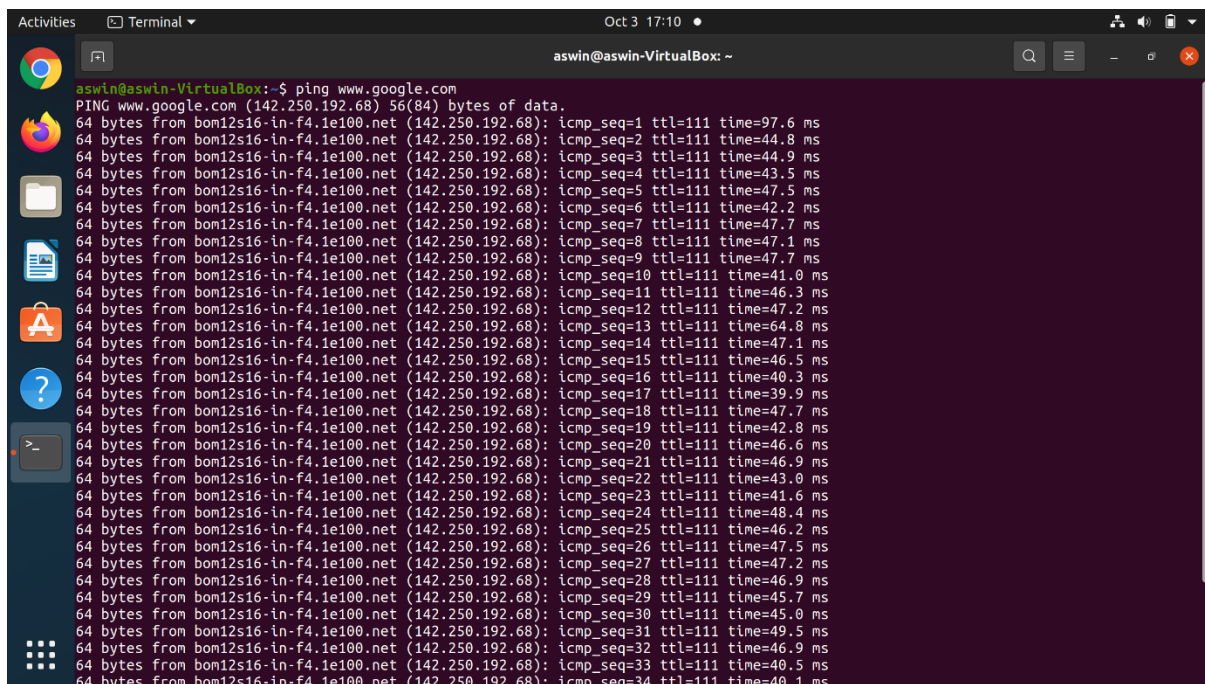


### Network Commands:

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 be used to ping a website or hostname directly.



```
aswin@aswin-VirtualBox: ~  
$ ping www.google.com  
PING www.google.com (142.250.192.68) 56(84) bytes of data:  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=1 ttl=111 time=97.6 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=2 ttl=111 time=44.8 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=3 ttl=111 time=44.9 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=4 ttl=111 time=43.5 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=5 ttl=111 time=47.5 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=6 ttl=111 time=42.2 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=7 ttl=111 time=47.7 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=8 ttl=111 time=47.1 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=9 ttl=111 time=47.7 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=10 ttl=111 time=41.0 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=11 ttl=111 time=46.3 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=12 ttl=111 time=47.2 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=13 ttl=111 time=64.8 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=14 ttl=111 time=47.1 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=15 ttl=111 time=46.5 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=16 ttl=111 time=40.3 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=17 ttl=111 time=39.9 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=18 ttl=111 time=47.7 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=19 ttl=111 time=42.8 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=20 ttl=111 time=46.6 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=21 ttl=111 time=46.9 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=22 ttl=111 time=43.0 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=23 ttl=111 time=41.6 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=24 ttl=111 time=48.4 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=25 ttl=111 time=46.2 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=26 ttl=111 time=47.5 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=27 ttl=111 time=47.2 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=28 ttl=111 time=46.9 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=29 ttl=111 time=45.7 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=30 ttl=111 time=45.0 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=31 ttl=111 time=49.5 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=32 ttl=111 time=46.9 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=33 ttl=111 time=40.5 ms  
64 bytes from bon12s16-in-f4.1e100.net (142.250.192.68): icmp_seq=34 ttl=111 time=40.1 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.

### Syntax:

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

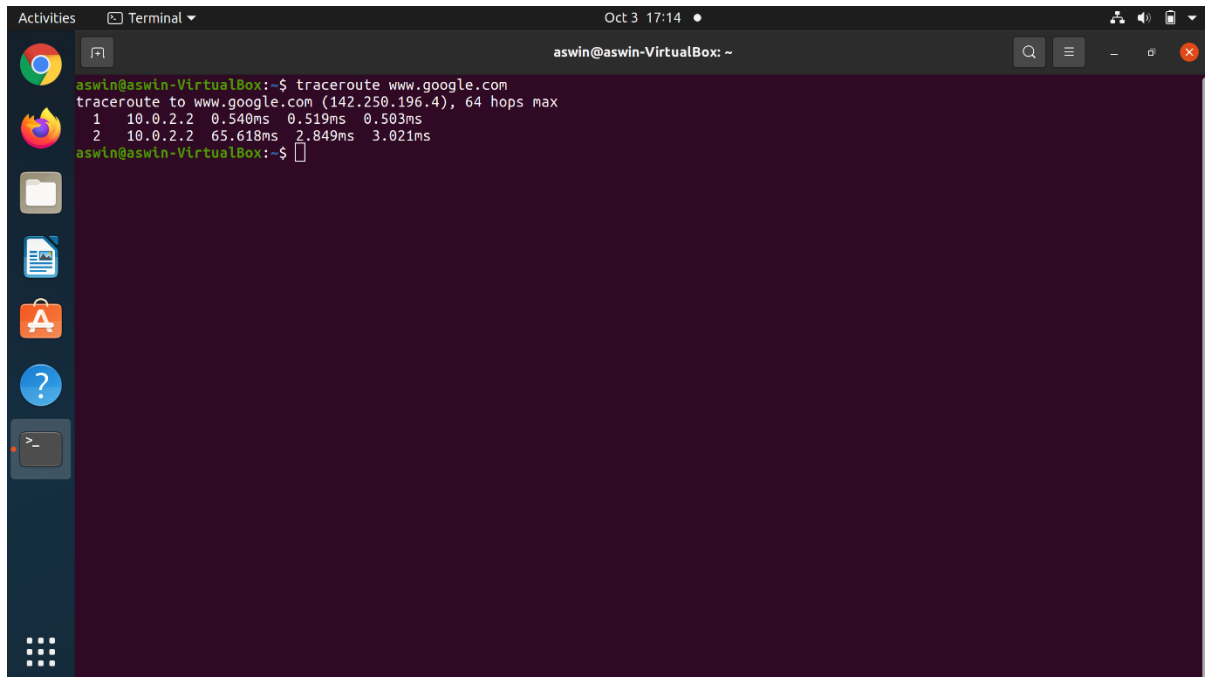
This section explains each of the options that you can use with the route command.

- ✓ 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.

## Traceroute

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.



```
aswin@aswin-VirtualBox:~$ traceroute www.google.com
traceroute to www.google.com (142.250.196.4), 64 hops max
 1  10.0.2.2  0.540ms  0.519ms  0.503ms
 2  10.0.2.2  65.618ms  2.849ms  3.021ms
aswin@aswin-VirtualBox:~$
```

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:

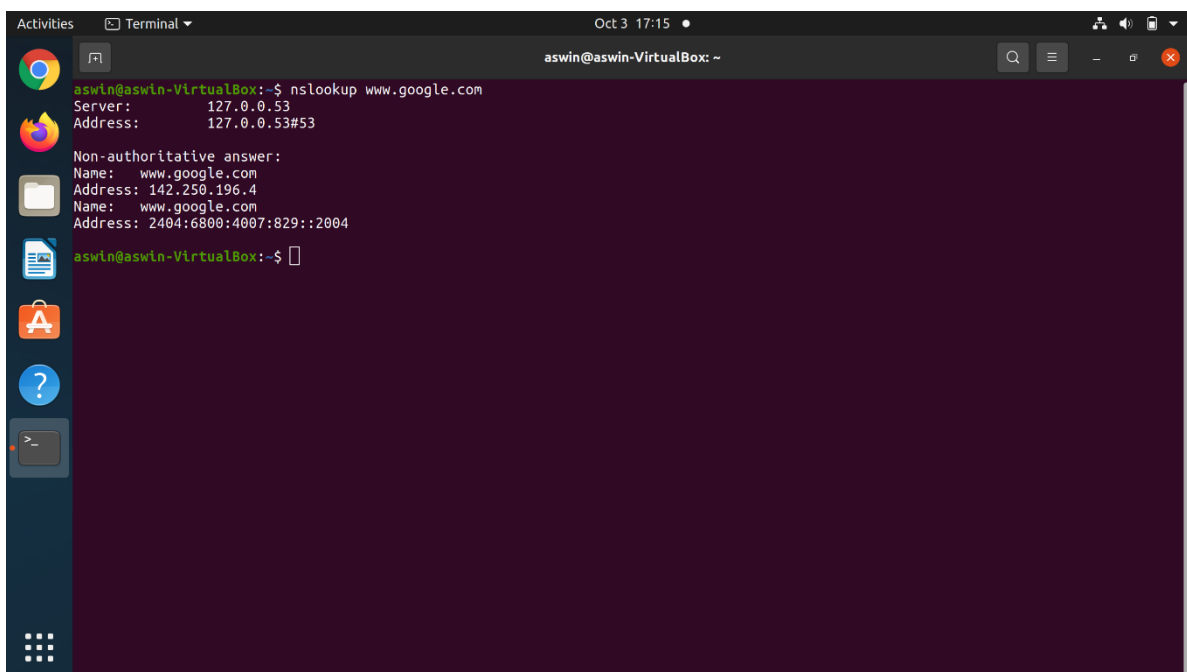
- -4 Option: Use ip version 4 i.e. use Ipv4
- -6 Option: Use ip version 6 i.e. use Ipv6
- -F Option: Do not fragment packet.

## 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:

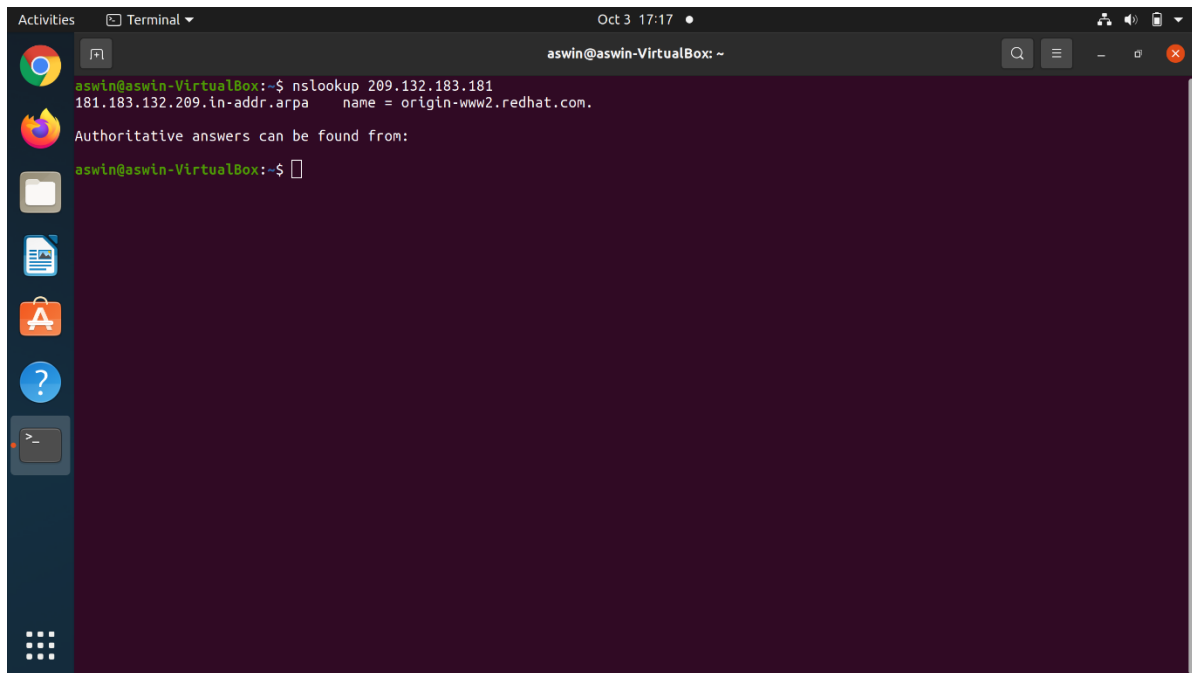
nslookup [option]



```
aswin@aswin-VirtualBox:~$ nslookup www.google.com
Server:      127.0.0.53
Address:     127.0.0.53#53

Non-authoritative answer:
Name:   www.google.com
Address: 142.250.196.4
Name:   www.google.com
Address: 2404:6800:4007:829::2004

aswin@aswin-VirtualBox:~$
```

A screenshot of a Linux terminal window. The window title is "Terminal" and the date/time is "Oct 3 17:17". The prompt is "aswin@aswin-VirtualBox: ~". The user has entered the command "nslookup 209.132.183.181". The output shows "181.183.132.209.in-addr.arpa" and "name = origin-www2.redhat.com.". Below this, it says "Authoritative answers can be found from:". The prompt is now "aswin@aswin-VirtualBox:~\$".

```
aswin@aswin-VirtualBox:~$ nslookup 209.132.183.181
181.183.132.209.in-addr.arpa      name = origin-www2.redhat.com.

Authoritative answers can be found from:

aswin@aswin-VirtualBox:~$
```

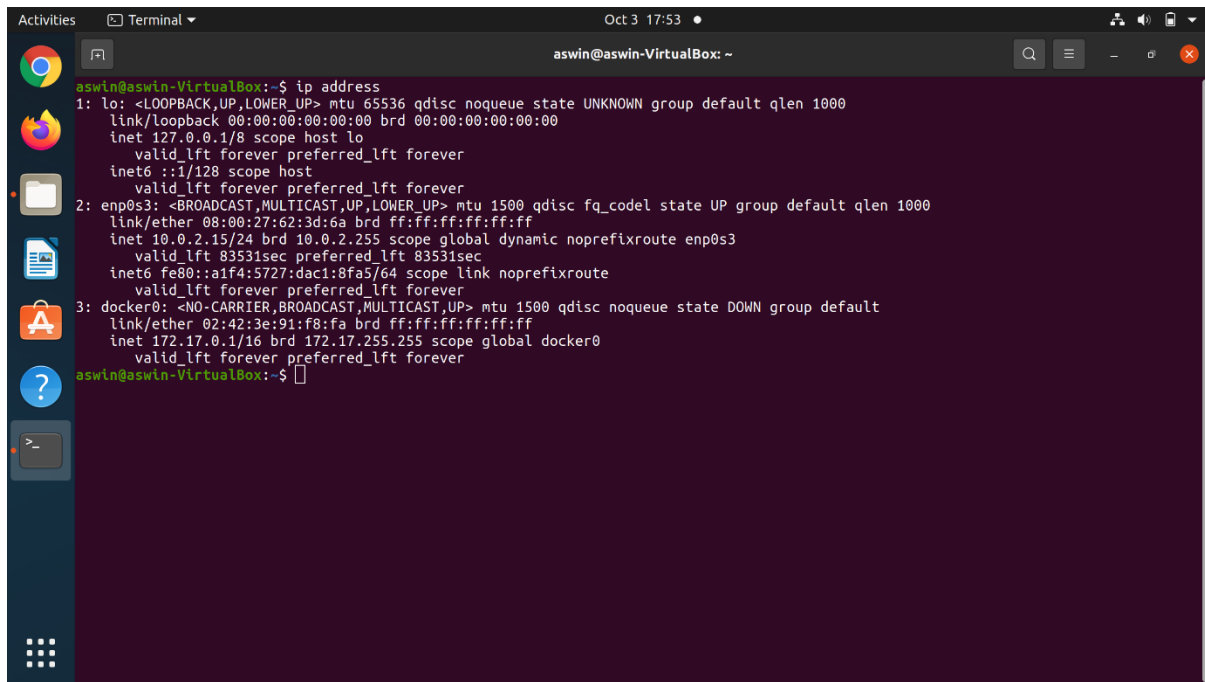
## ip:

ip command in Linux is present in the net-tools which is used for performing several network administration tasks. IP stands for Internet Protocol. This command is used to show or manipulate routing, devices, and tunnels. It is similar to *ifconfig* command but it is much more powerful with more functions and facilities attached to it. *ifconfig* is one of the deprecated commands in the net-tools of Linux that has not been maintained for many years. ip command is used to perform several tasks like assigning an address to a network interface or configuring network interface parameters.

It can perform several other tasks like configuring and modifying the default and static routing, setting up tunnel over IP, listing IP addresses and property information, modifying the status of the interface, assigning, deleting and setting up IP addresses and routes.

## Syntax:

```
ip [ OPTIONS ] OBJECT { COMMAND | help }
```



```
aswin@aswin-VirtualBox:~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:62:3d:6a brd ff:ff:ff:ff:ff:ff
   inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 83531sec preferred_lft 83531sec
   inet6 fe80::a1f4:5727:dac1:8fa5/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
   link/ether 02:42:3e:91:f8:fa brd ff:ff:ff:ff:ff:ff
   inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
       valid_lft forever preferred_lft forever
aswin@aswin-VirtualBox:~$
```

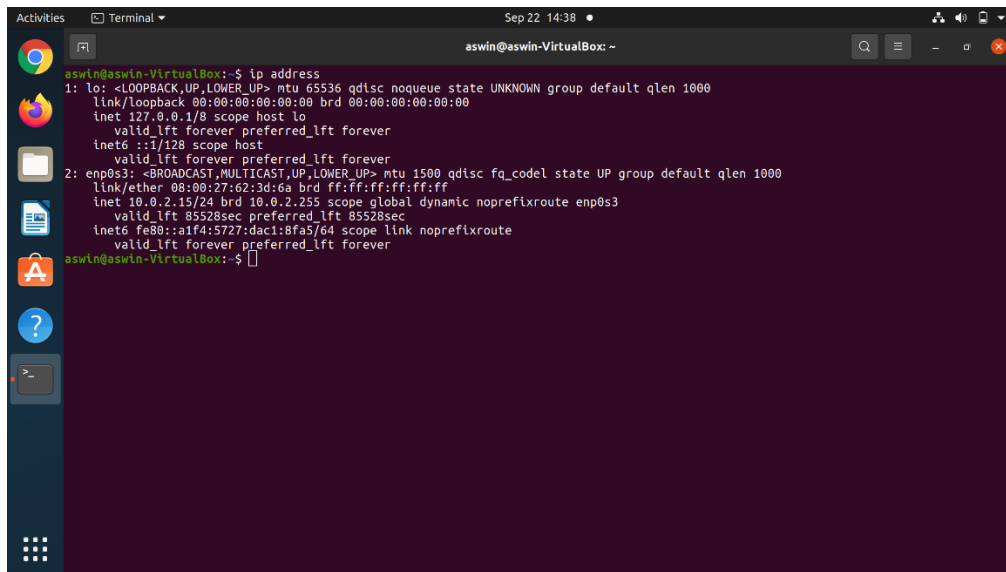
This will show the information related to all interfaces available on our system, but if we want to view the information of any particular interface, add the options show followed by the name of the particular network interface.

#### Options:

- -address: This option is used to show all IP addresses associated on all network devices.
- -link: It is used to display link layer information, it will fetch characteristics of the link layer devices currently available. Any networking device which has a driver loaded can be classified as an available device.

### Setting up static IP addresses

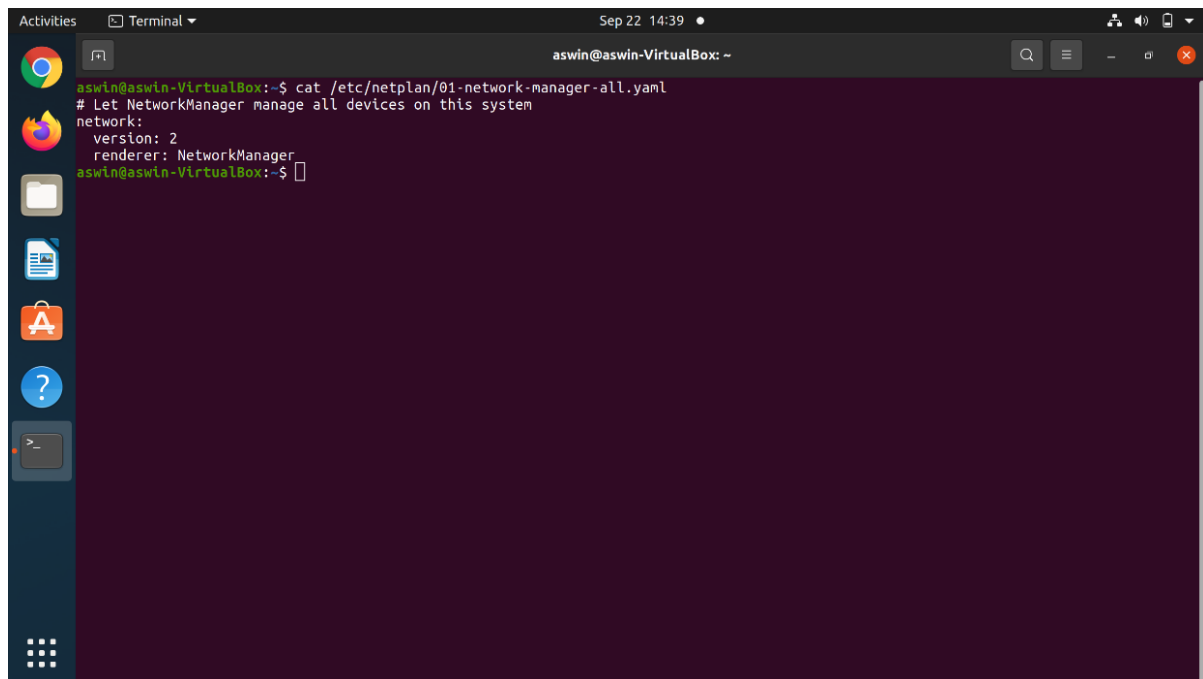
Step 1: List all the interfaces in the system. Use the ip address command to define a static IP address on an interface.



```
aswin@aswin-VirtualBox:~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:62:3d:6a brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
        valid_lft 85528sec preferred_lft 85528sec
    inet6 fe80::a1f4:5727:dac1:8fa5/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
aswin@aswin-VirtualBox:~$
```

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

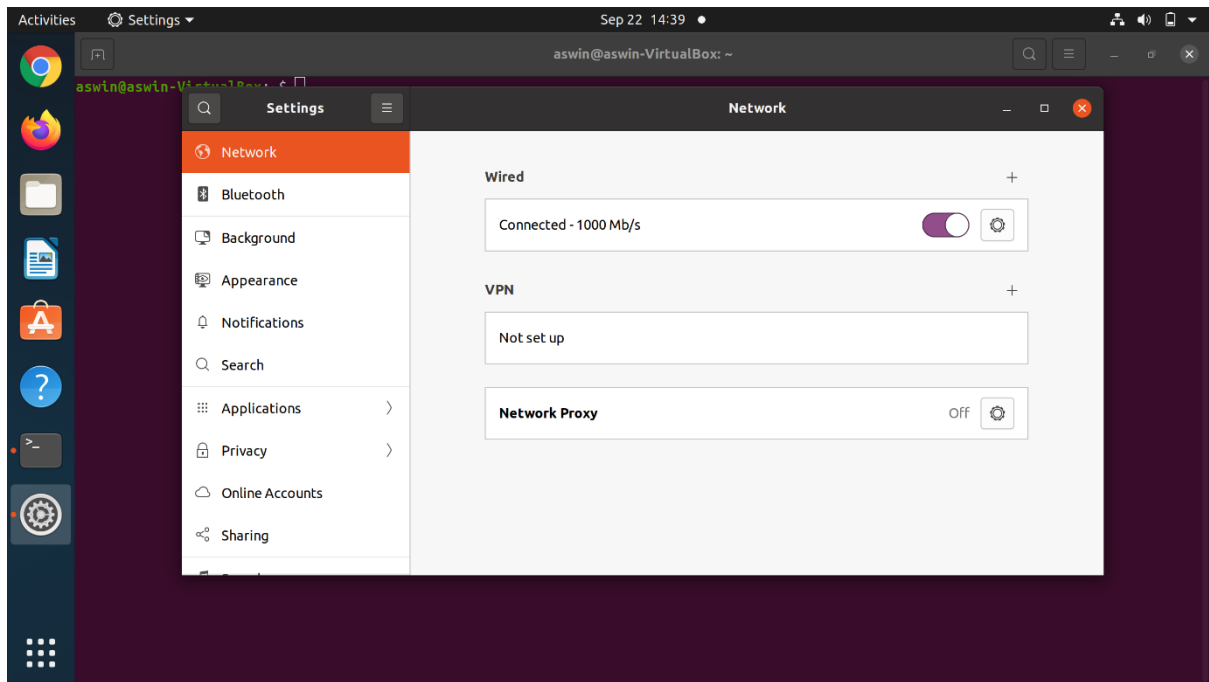
```
cat /etc/netplan/01-network-manager-all.yaml
```



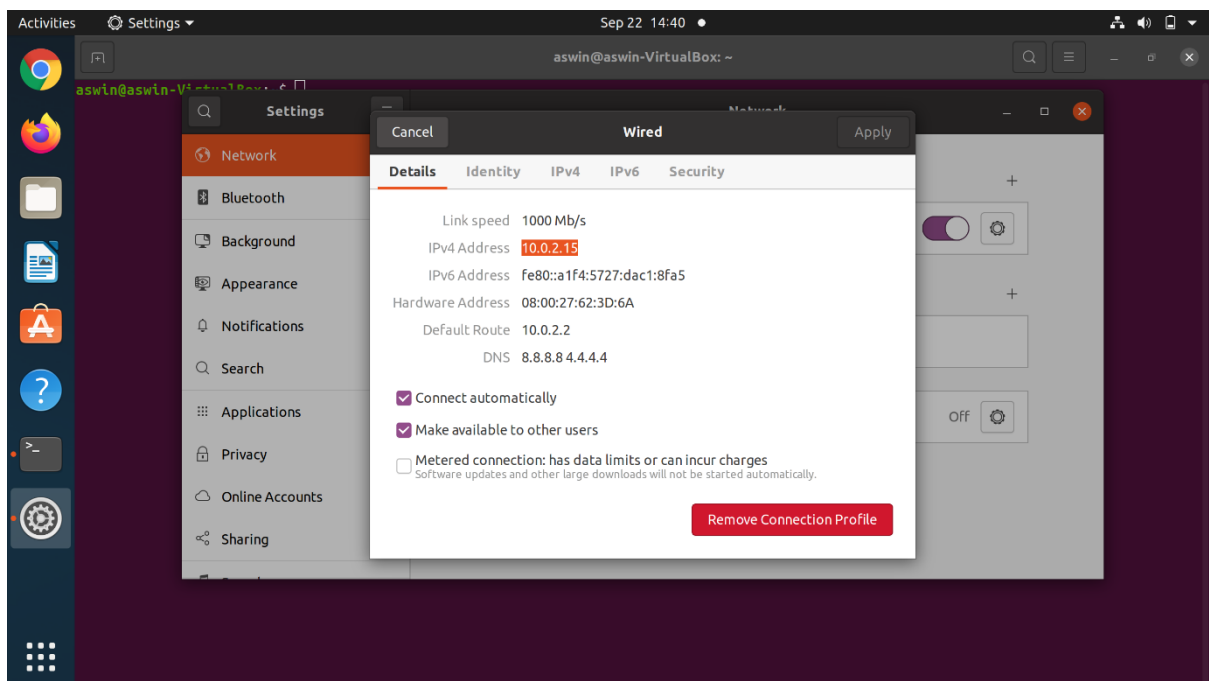
```
aswin@aswin-VirtualBox:~$ cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
aswin@aswin-VirtualBox:~$
```

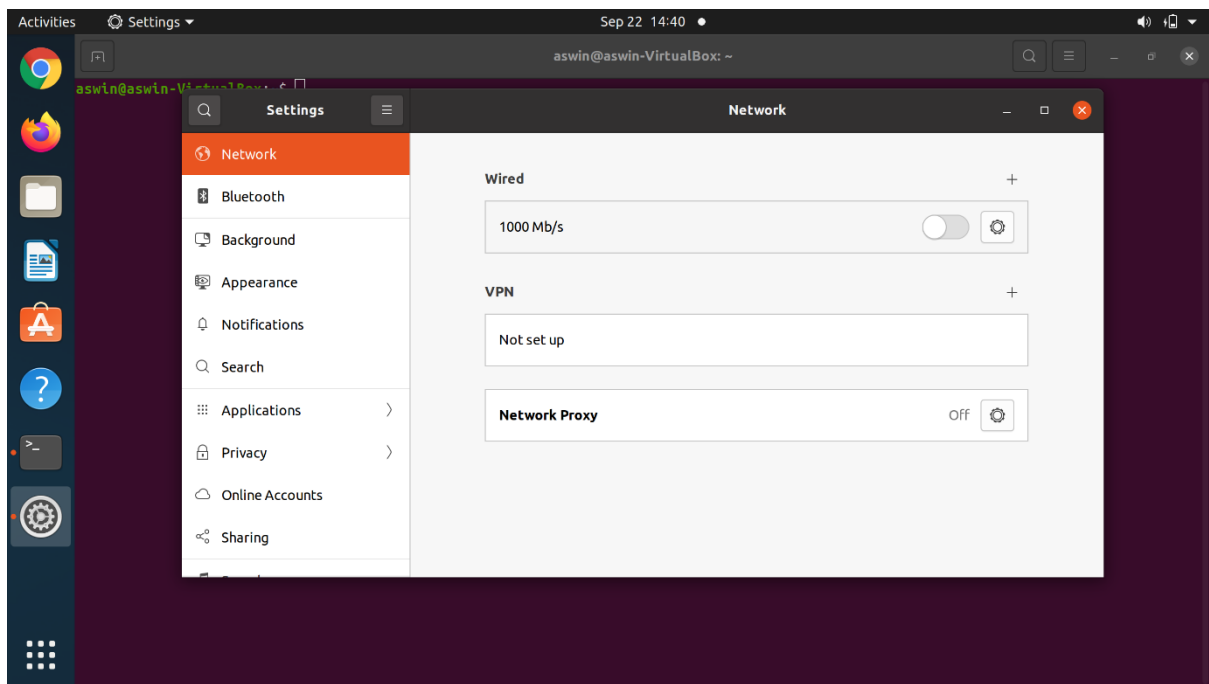
Step 3: Click on the top right network icon and select settings of the network interface you wish to configure to use a static IP address on Ubuntu.



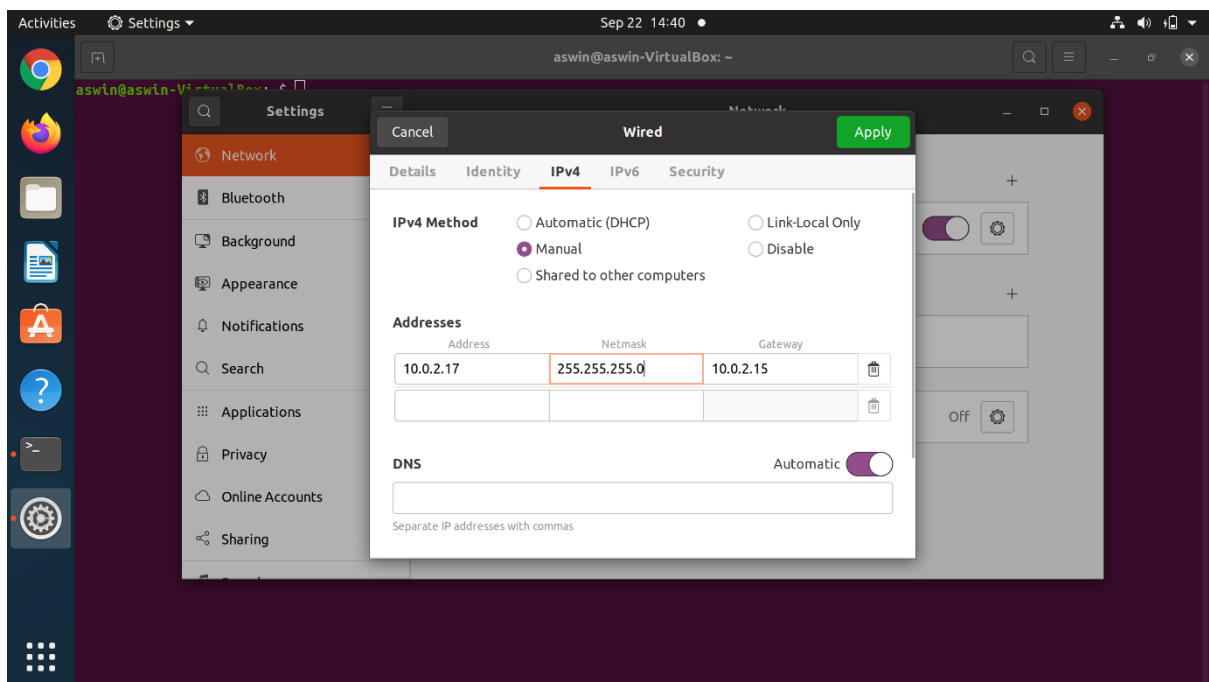


Click on the settings icon to start IP address configuration.

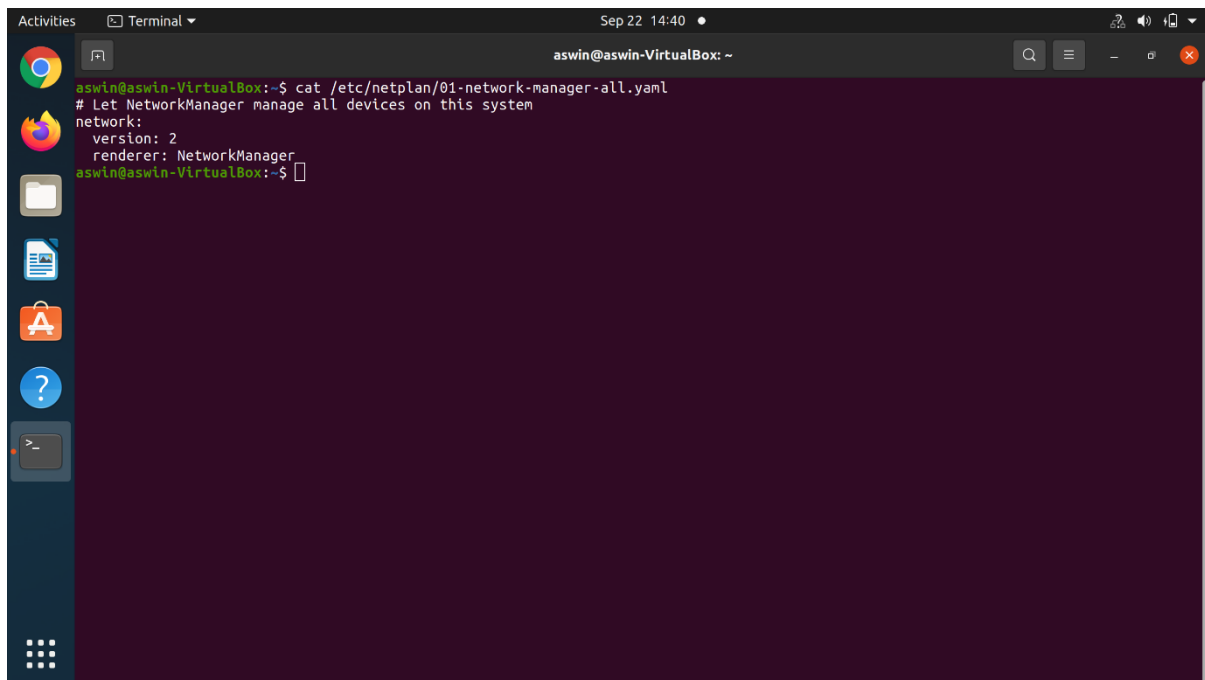




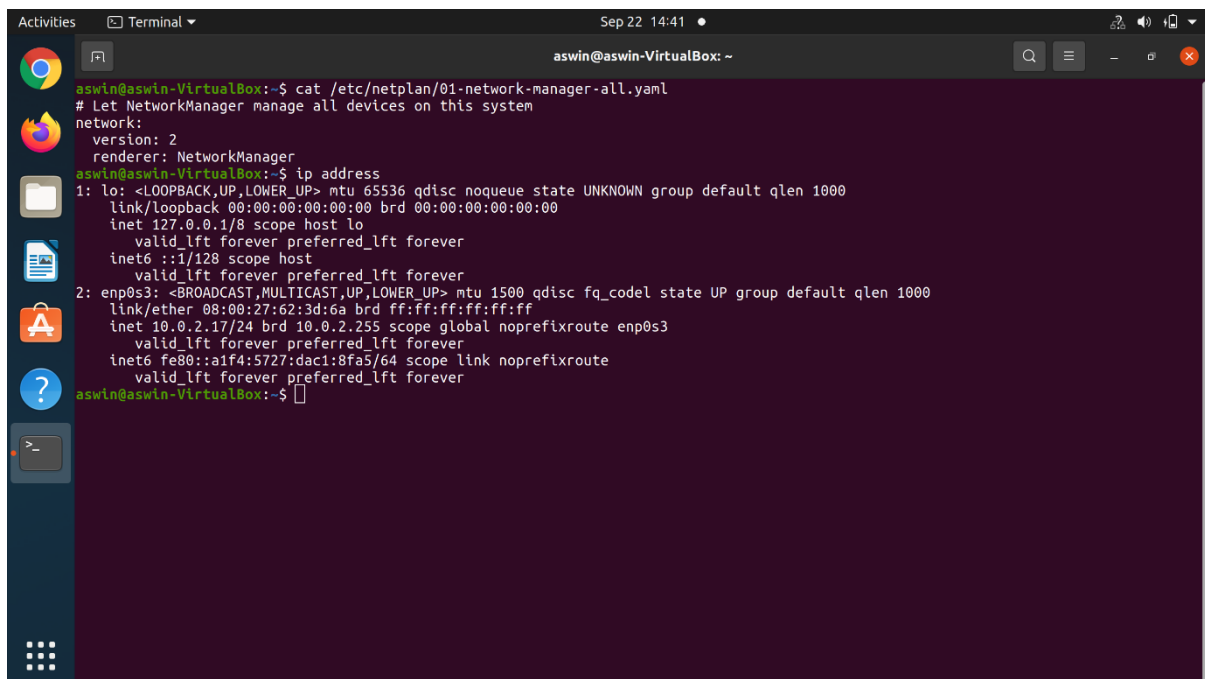
Turn OFF and ON switch to apply your new network static IP configuration settings.



Step 5: Run the command `ip address` and click on the network settings icon once again to confirm your new static IP address settings.



```
aswin@aswin-VirtualBox: ~  
$ cat /etc/netplan/01-network-manager-all.yaml  
# Let NetworkManager manage all devices on this system  
network:  
  version: 2  
  renderer: NetworkManager  
aswin@aswin-VirtualBox: ~$
```



```
aswin@aswin-VirtualBox: ~  
$ cat /etc/netplan/01-network-manager-all.yaml  
# Let NetworkManager manage all devices on this system  
network:  
  version: 2  
  renderer: NetworkManager  
aswin@aswin-VirtualBox: ~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
   link/ether 08:00:27:62:3d:6a brd ff:ff:ff:ff:ff:ff  
   inet 10.0.2.17/24 brd 10.0.2.255 scope global noprefixroute enp0s3  
     valid_lft forever preferred_lft forever  
   inet6 fe80::a1f4:5727:dac1:8fa5/64 scope link noprefixroute  
     valid_lft forever preferred_lft forever  
aswin@aswin-VirtualBox: ~$
```

## 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

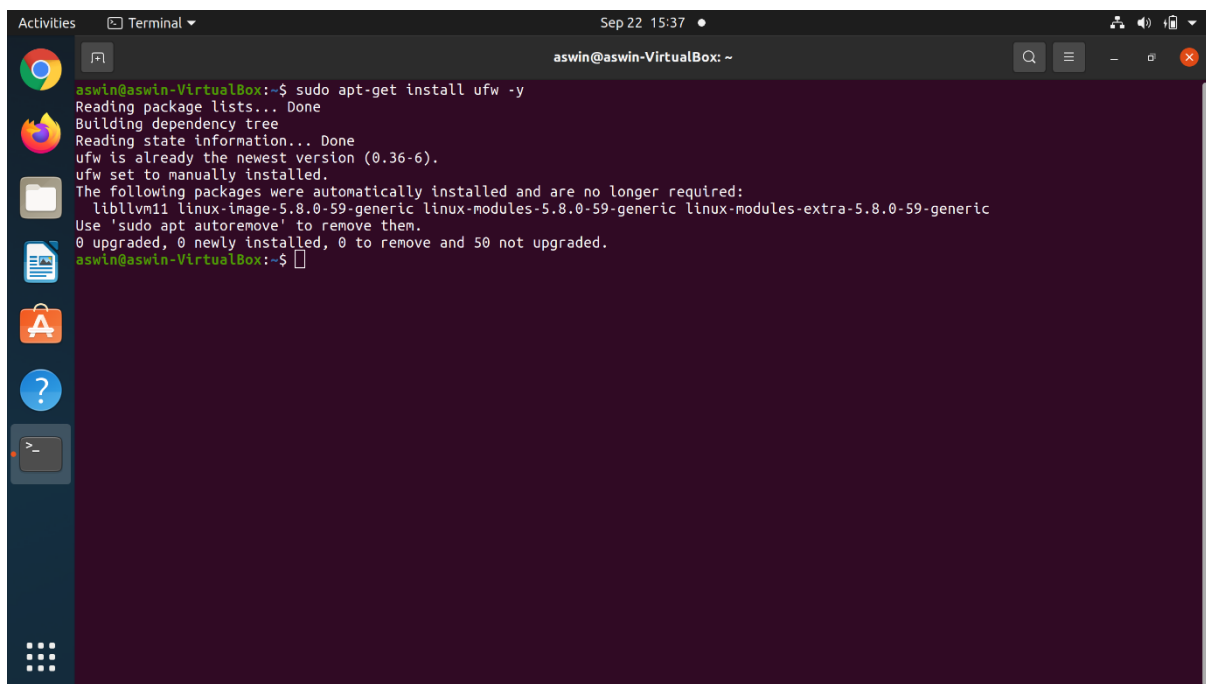
beginner to learns and configure the firewall rules where we will secure the network from unknown users are machines. UFW works on the policies we configure as rules.

- 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
```

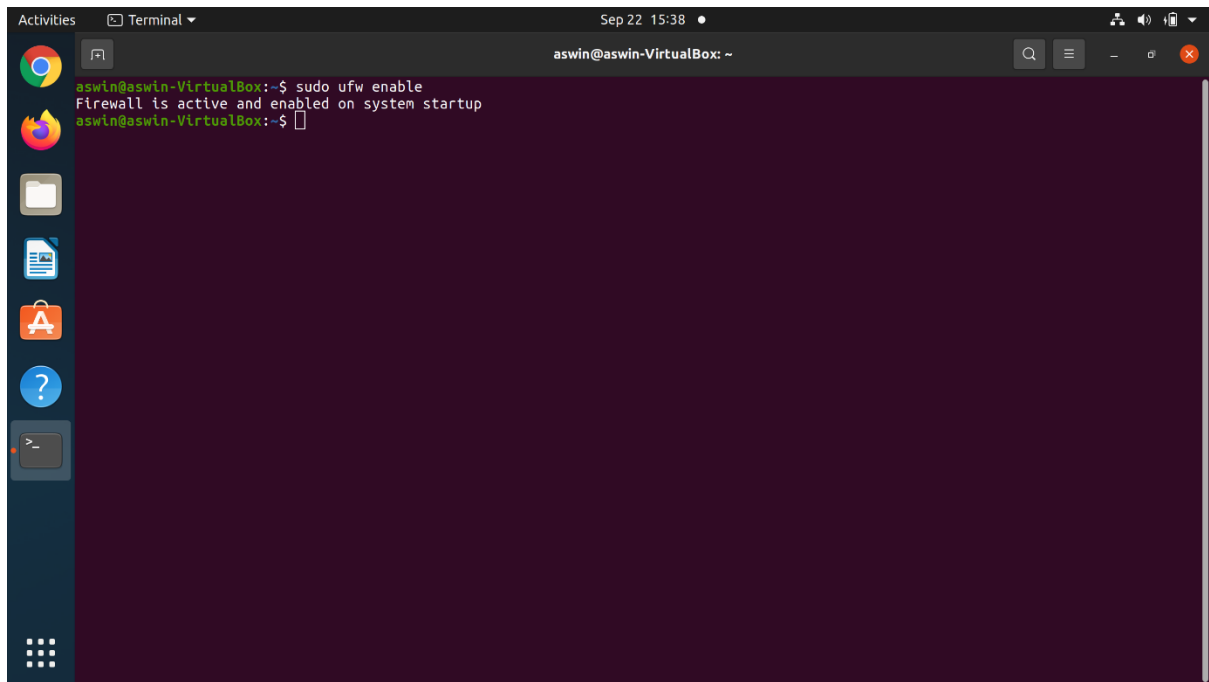
A screenshot of a terminal window titled "aswin@aswin-VirtualBox: ~". The terminal shows the command "sudo apt-get install ufw -y" being executed. The output indicates that the package lists are read, the dependency tree is built, and the state information is read. It confirms that ufw is already the newest version (0.36-6) and is set to manually installed. A list of packages that are no longer required is shown, including libllvm11, linux-image-5.8.0-59-generic, linux-modules-5.8.0-59-generic, and linux-modules-extra-5.8.0-59-generic. The terminal also shows the command "sudo apt autoremove" being used to remove these packages. The final output shows that 0 packages were upgraded, 0 were newly installed, 0 were to be removed, and 50 were not upgraded. The terminal prompt returns to "aswin@aswin-VirtualBox:~\$".

```
aswin@aswin-VirtualBox:~$ sudo apt-get install ufw -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
ufw is already the newest version (0.36-6).
ufw set to manually installed.
The following packages were automatically installed and are no longer required:
  libllvm11 linux-image-5.8.0-59-generic linux-modules-5.8.0-59-generic linux-modules-extra-5.8.0-59-generic
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 50 not upgraded.
aswin@aswin-VirtualBox:~$
```

### Enabling the UFW (Firewall)

Below is the command to enable the UFW –

```
sudo ufw enable
```

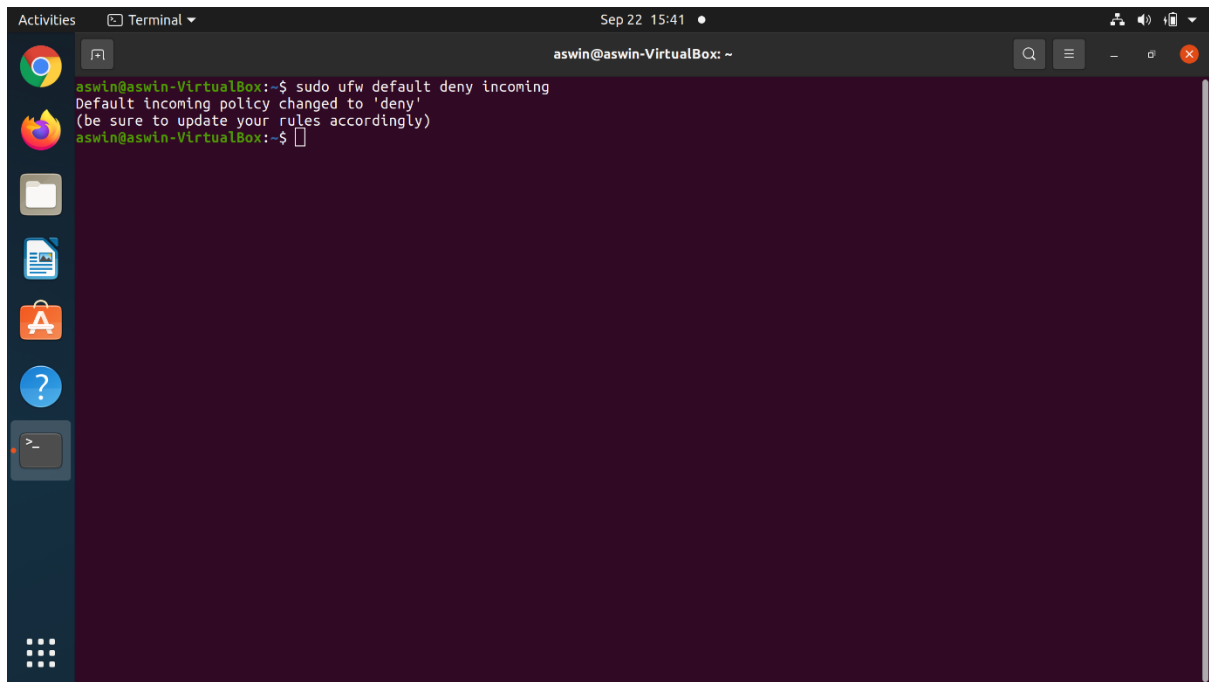


### Enabling the Default Policies

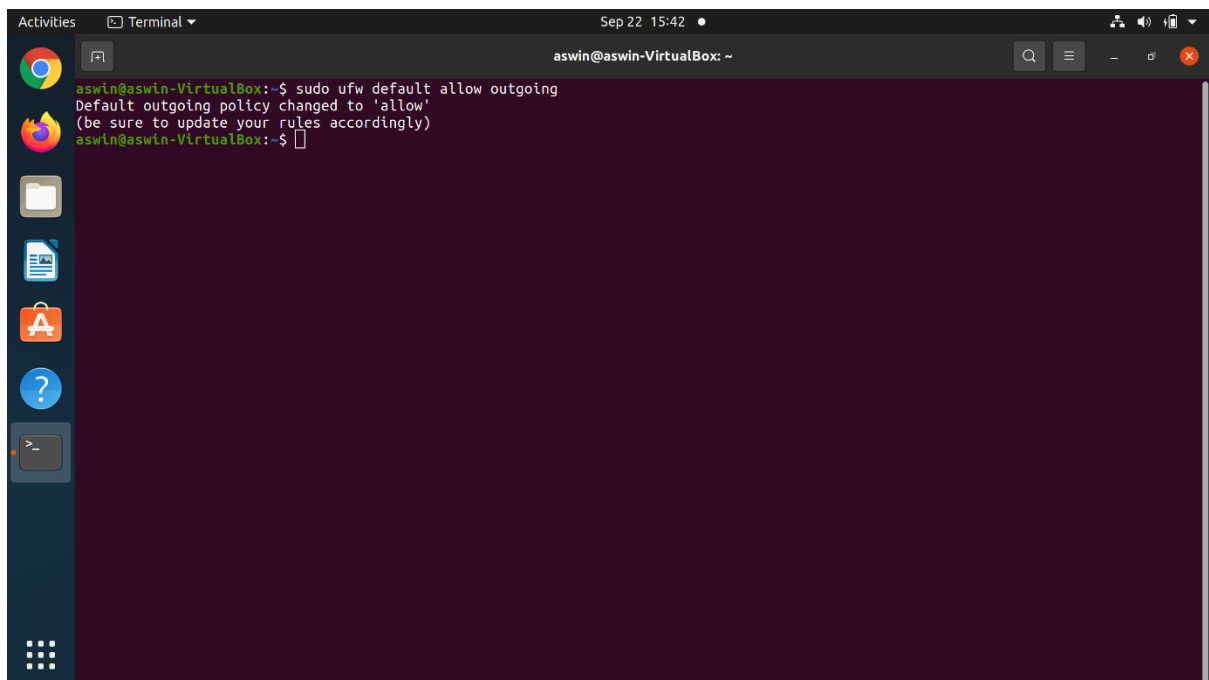
As the beginner, we will first configure default policies, which control and handles the traffic which will not match the other rules. By default, the rules will deny all incoming connections and allow all outgoing connections will be allowed which stops someone trying to reach the machine from the internet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

```
sudo ufw default deny incoming
```

```
sudo ufw default allow outgoing
```



```
aswin@aswin-VirtualBox:~$ sudo ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
aswin@aswin-VirtualBox:~$
```

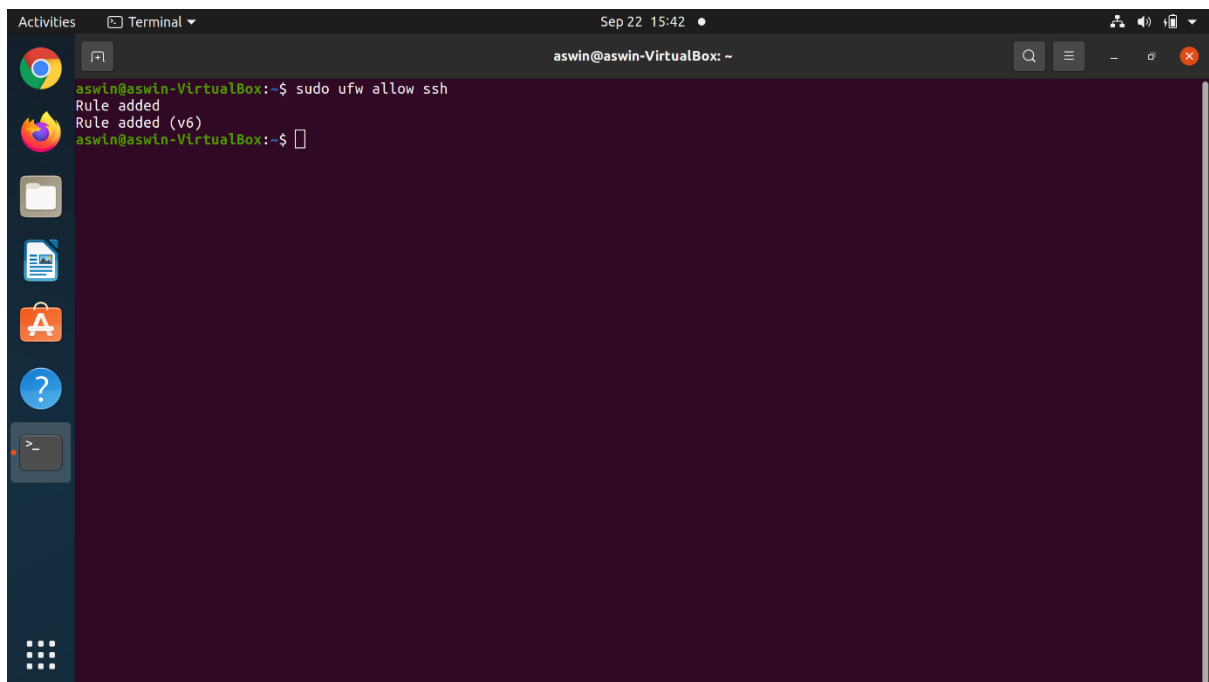


```
aswin@aswin-VirtualBox:~$ sudo ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
aswin@aswin-VirtualBox:~$
```

### 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
```

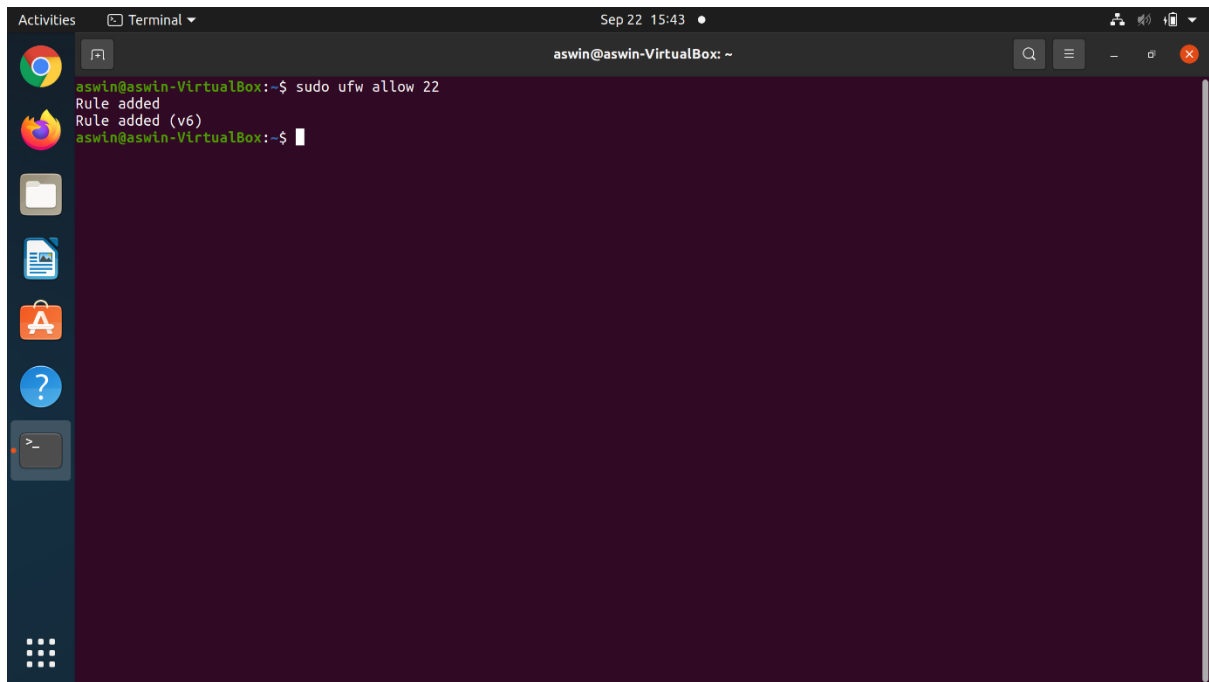
A screenshot of a Linux terminal window. The window title is "aswin@aswin-VirtualBox: ~". The terminal shows the command "sudo ufw allow ssh" being entered. The output is "Rule added" followed by "Rule added (v6)". The prompt "aswin@aswin-VirtualBox:~\$" is visible at the bottom. The terminal has a dark purple background. On the left side of the window, there is a vertical dock with several application icons: a web browser, a file manager, a terminal, and others. The top of the window shows the system clock as "Sep 22 15:42".

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.

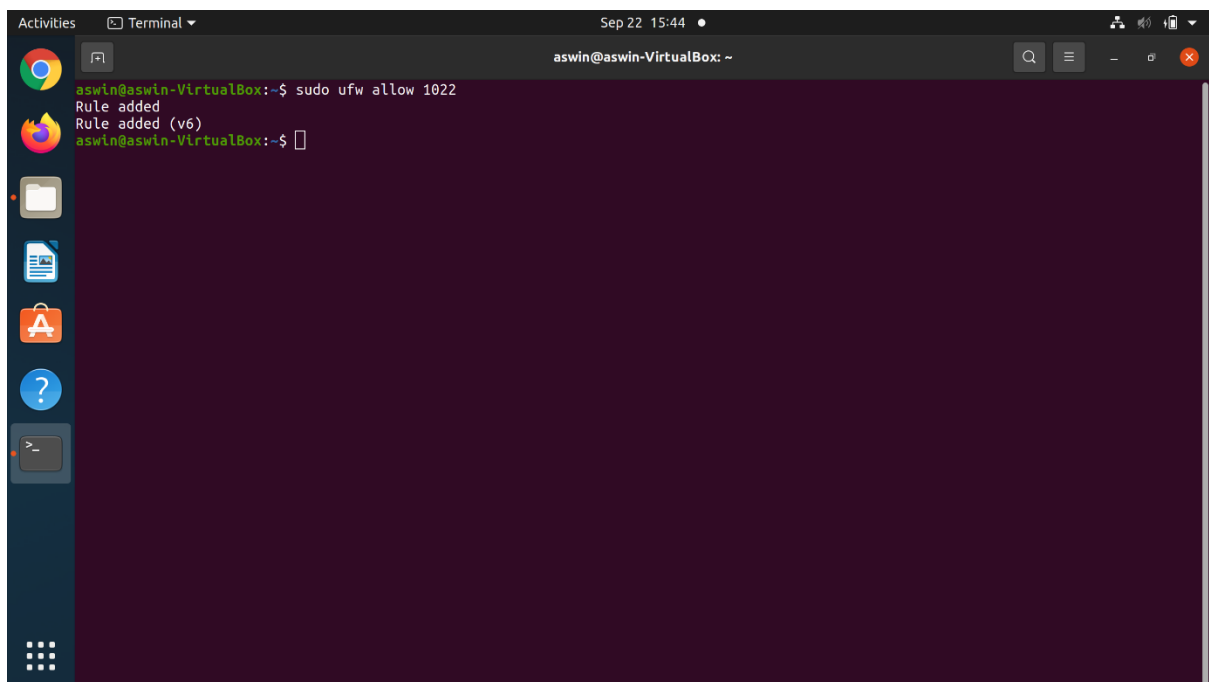
```
sudo ufw allow 22
```

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
```



```
aswin@aswin-VirtualBox:~$ sudo ufw allow 22
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```



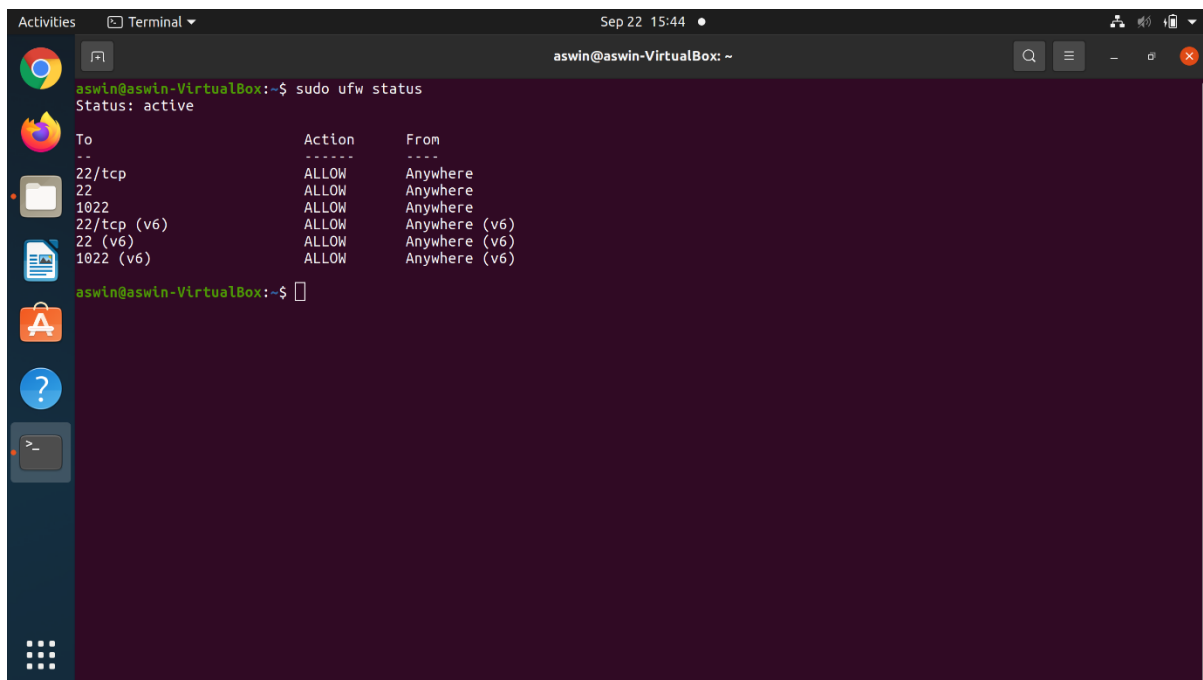
```
aswin@aswin-VirtualBox:~$ sudo ufw allow 1022
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```

### Checking the UFW (Firewall) Status

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

```
sudo ufw status
```



A terminal window titled 'aswin@aswin-VirtualBox: ~' showing the output of the 'sudo ufw status' command. The output indicates that UFW is active and lists several rules allowing traffic on ports 22, 1022, and their IPv6 equivalents from anywhere.

```
aswin@aswin-VirtualBox:~$ 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)
```

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

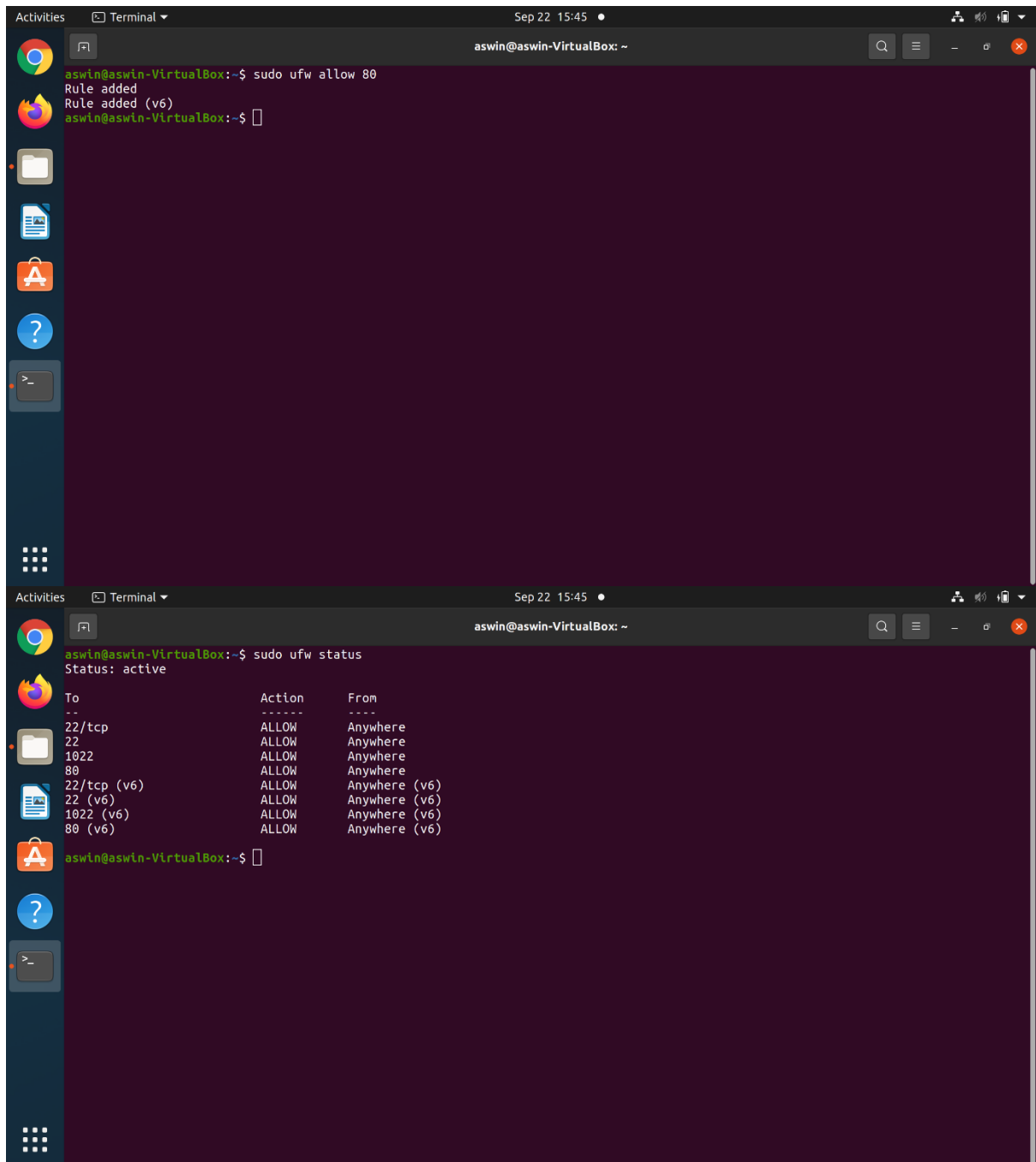
At this point, we will allow others to connect to the server for the regular ports like HTTP, HTTPS, and FTP ports respectively.

#### HTTP port 80

```
sudo ufw allow 80
```

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

```
sudo ufw status
```



```
aswin@aswin-VirtualBox:~$ sudo ufw allow 80
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```

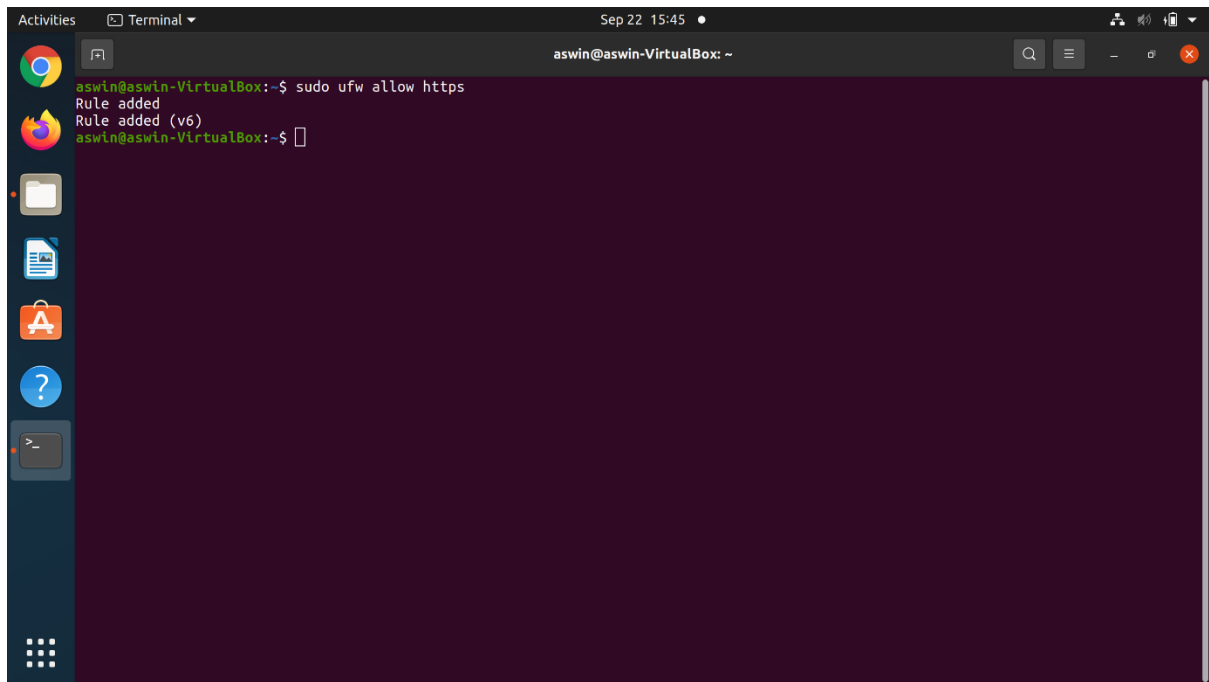
```
aswin@aswin-VirtualBox:~$ sudo ufw status
Status: active

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)
```

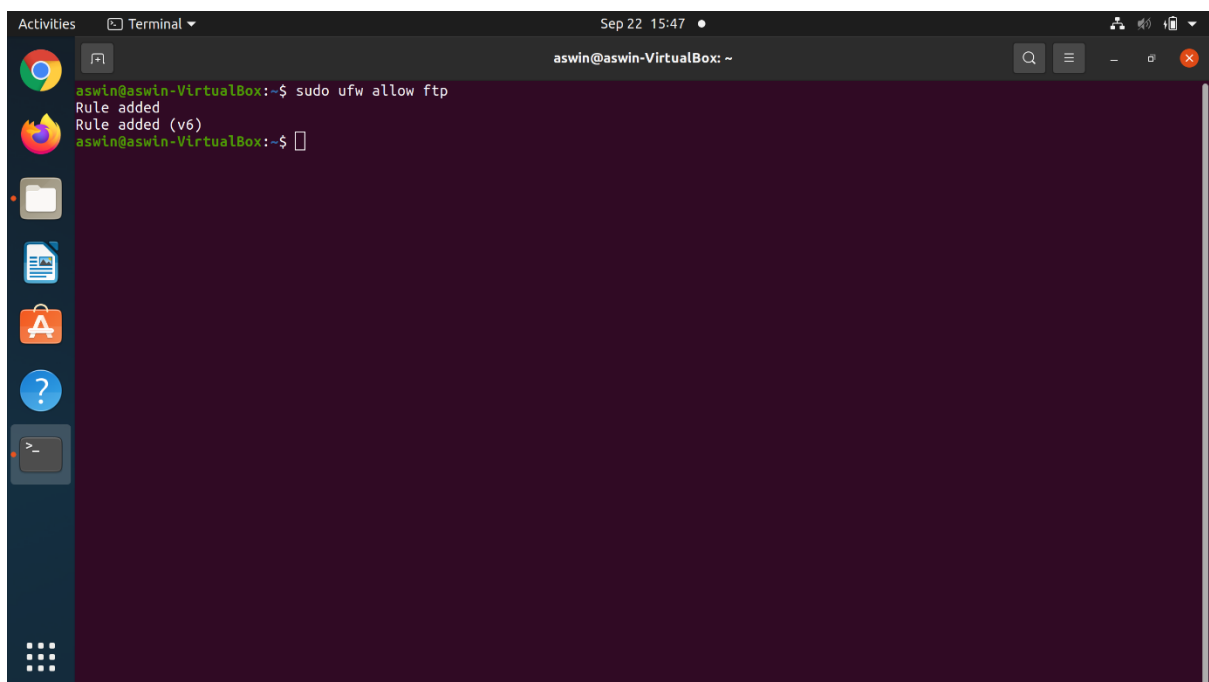
Like that will use the below command to enable HTTPs and FTP ports (443 and 21) respectively.

```
sudo ufw allow https
```

```
sudo ufw allow ftp
```



```
aswin@aswin-VirtualBox:~$ sudo ufw allow https
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```

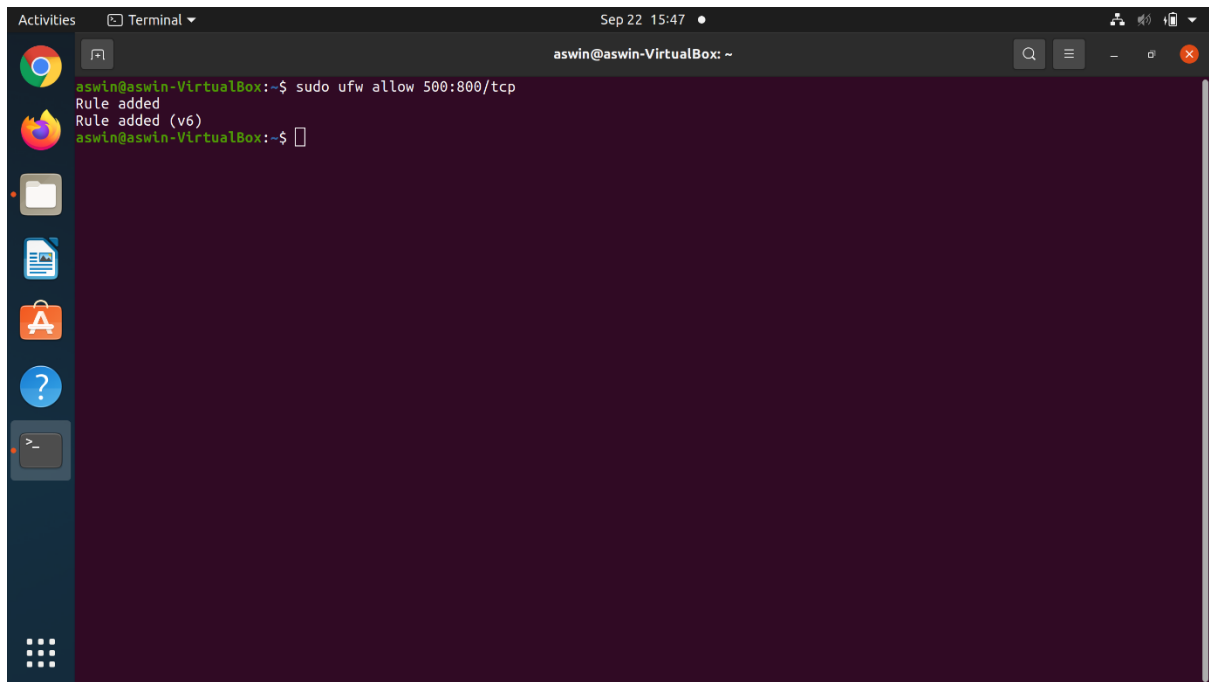


```
aswin@aswin-VirtualBox:~$ sudo ufw allow ftp
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```

### 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
```

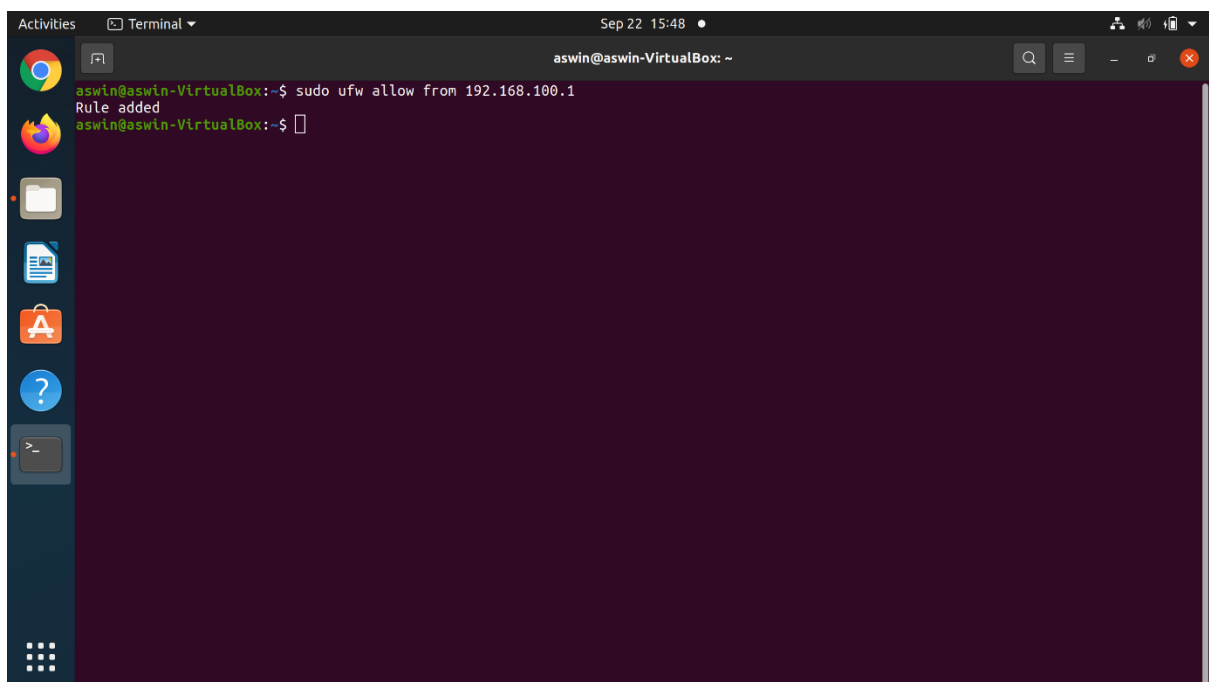
A terminal window titled 'aswin@aswin-VirtualBox: ~' showing the execution of the command 'sudo ufw allow 500:800/tcp'. The output shows 'Rule added' and 'Rule added (v6)'. The terminal is part of a desktop environment with a sidebar containing icons for various applications.

```
aswin@aswin-VirtualBox:~$ sudo ufw allow 500:800/tcp
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```

### 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.

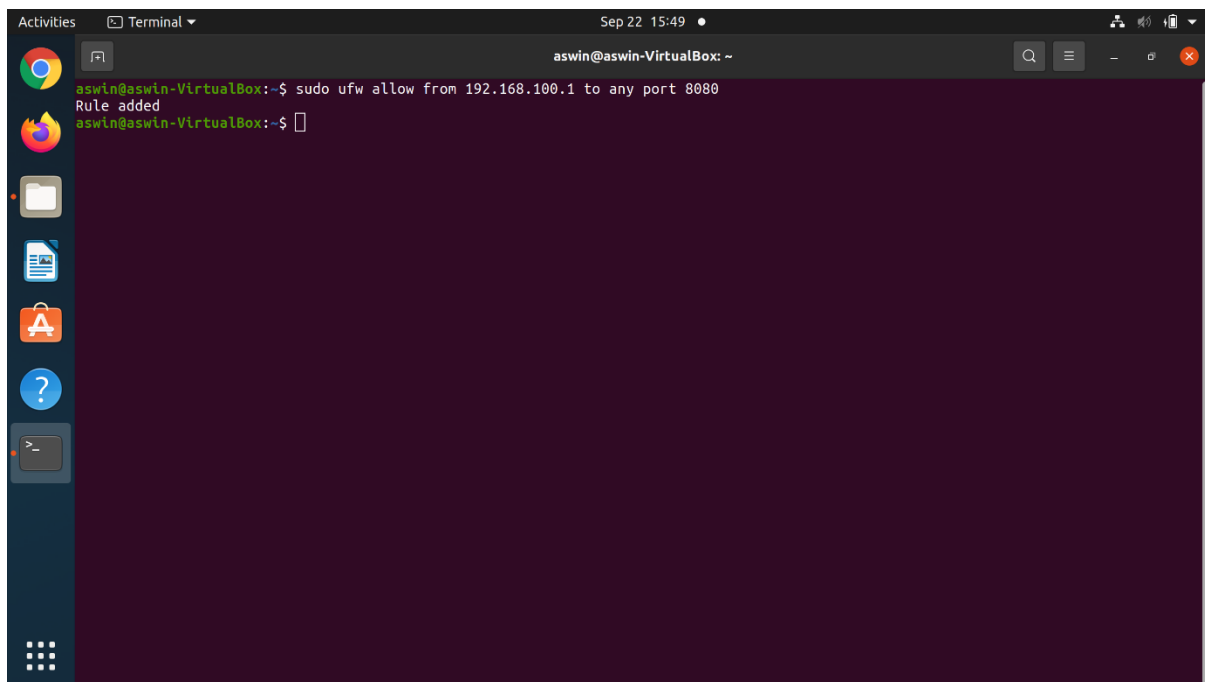
`sudo ufw allow from 192.168.100.1`

A terminal window titled 'aswin@aswin-VirtualBox: ~' showing the execution of the command 'sudo ufw allow from 192.168.100.1'. The output shows 'Rule added'. The terminal is part of a desktop environment with a sidebar containing icons for various applications.

```
aswin@aswin-VirtualBox:~$ sudo ufw allow from 192.168.100.1
Rule added
aswin@aswin-VirtualBox:~$
```

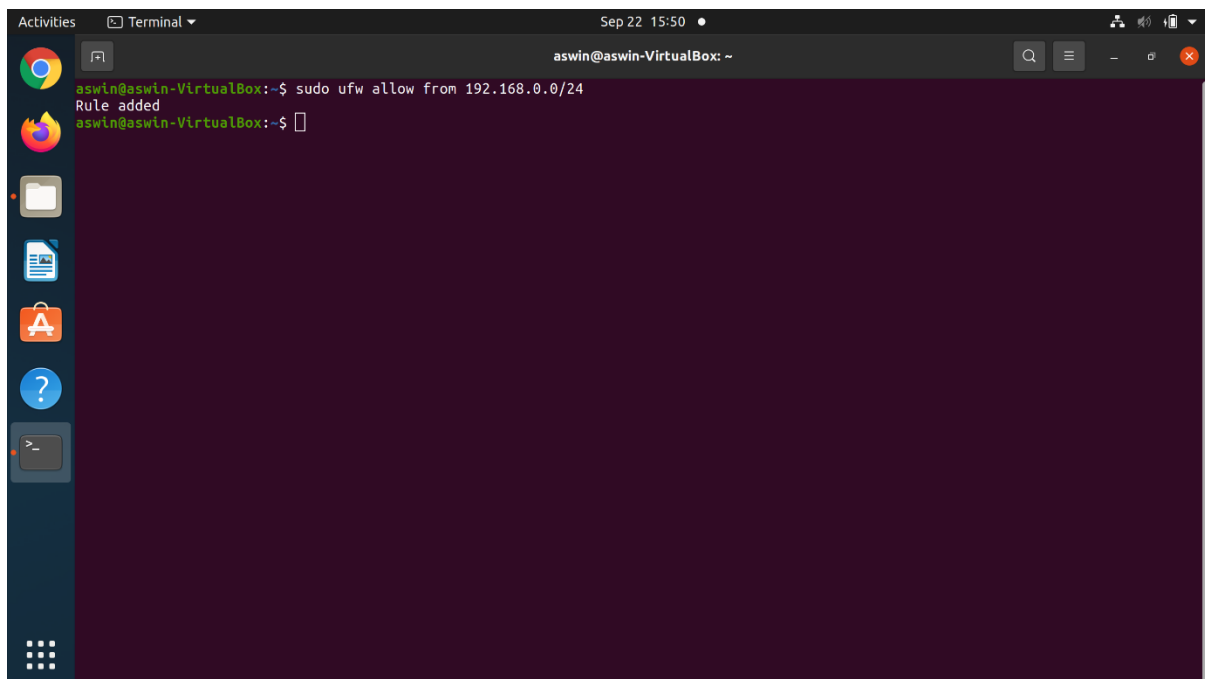
If we want to allow for only specific port we can use the below command.

`sudo ufw allow from 192.168.100.1 to any port 8080`

A screenshot of a Linux terminal window. The window title is "aswin@aswin-VirtualBox: ~". The terminal shows the command `aswin@aswin-VirtualBox:~$ sudo ufw allow from 192.168.100.1 to any port 8080` being entered. The output is `Rule added`. The prompt `aswin@aswin-VirtualBox:~$` is shown again. The terminal has a dark purple background and a light blue prompt. The window's top bar shows "Activities", "Terminal", and the date "Sep 22 15:49". The left sidebar shows various application icons.

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

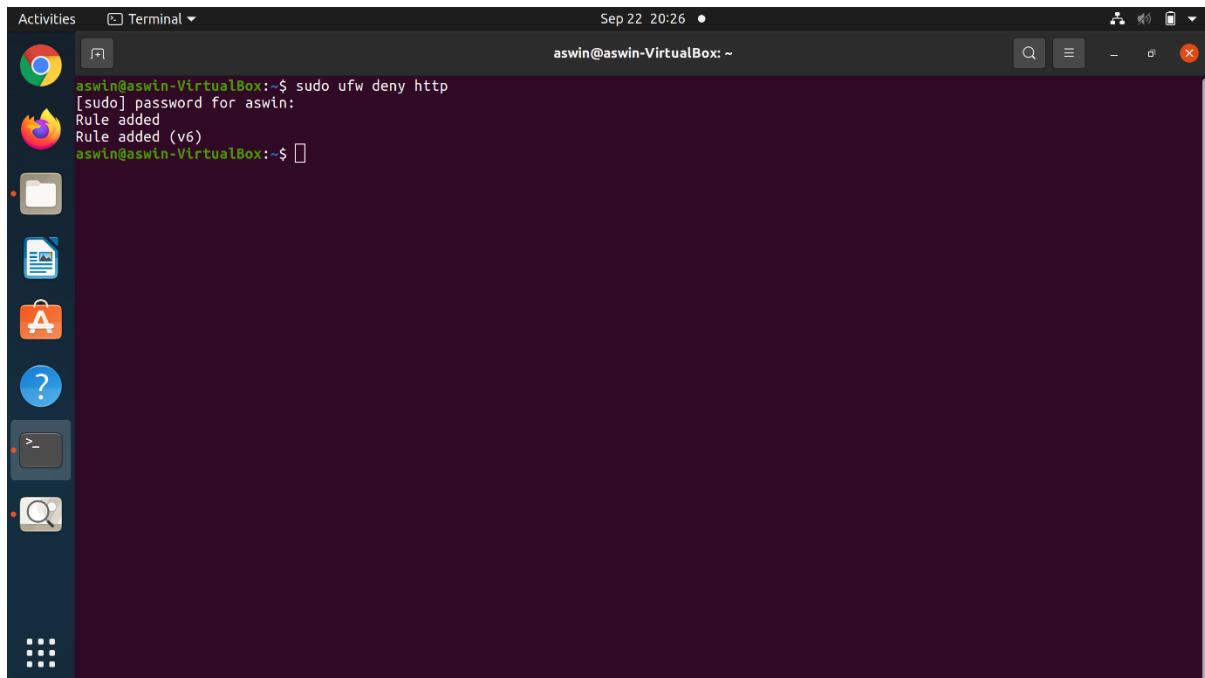
`sudo ufw allow from 192.168.0.0/24`

A screenshot of a Linux terminal window. The window title is "aswin@aswin-VirtualBox: ~". The terminal shows the command `aswin@aswin-VirtualBox:~$ sudo ufw allow from 192.168.0.0/24` being entered. The output is `Rule added`. The prompt `aswin@aswin-VirtualBox:~$` is shown again. The terminal has a dark purple background and a light blue prompt. The window's top bar shows "Activities", "Terminal", and the date "Sep 22 15:50". The left sidebar shows various application icons.

## Deny the Connections or Rules

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

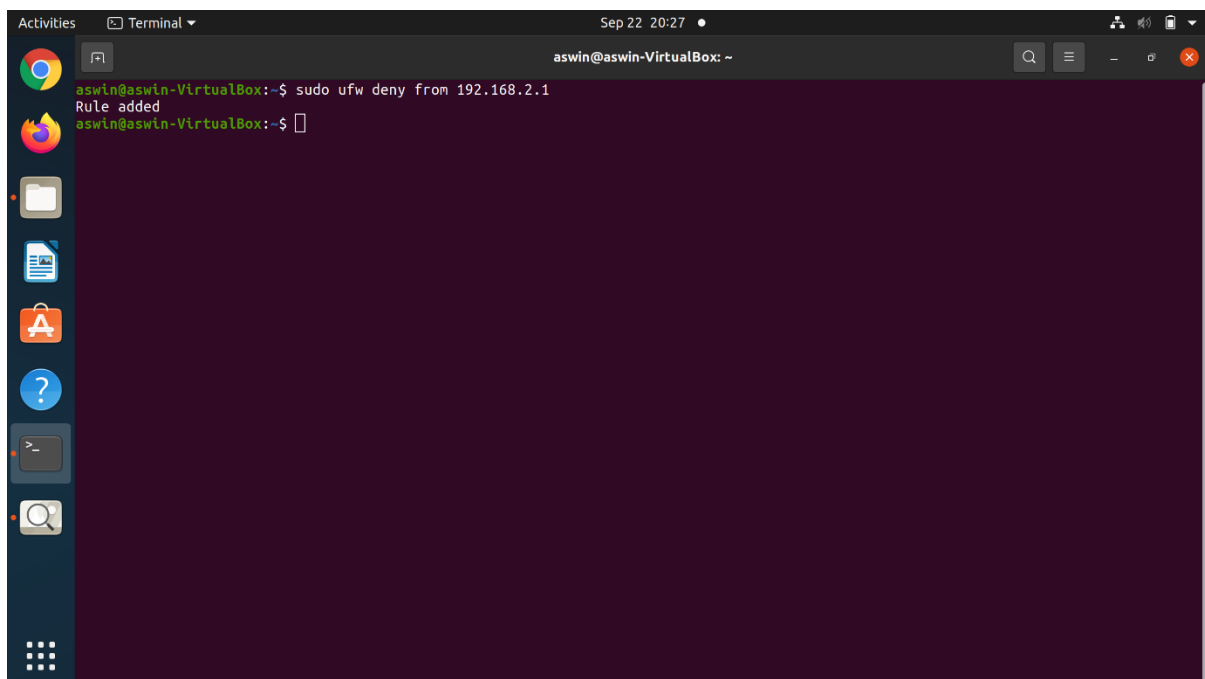
`sudo ufw deny http`

A terminal window titled 'aswin@aswin-VirtualBox: ~' showing the execution of the command 'sudo ufw deny http'. The output shows '[sudo] password for aswin:', 'Rule added', and 'Rule added (v6)'. The prompt returns to 'aswin@aswin-VirtualBox:~\$'.

```
aswin@aswin-VirtualBox:~$ sudo ufw deny http
[sudo] password for aswin:
Rule added
Rule added (v6)
aswin@aswin-VirtualBox:~$
```

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`

A terminal window titled 'aswin@aswin-VirtualBox: ~' showing the execution of the command 'sudo ufw deny from 192.168.2.1'. The output shows 'Rule added'. The prompt returns to 'aswin@aswin-VirtualBox:~\$'.

```
aswin@aswin-VirtualBox:~$ sudo ufw deny from 192.168.2.1
Rule added
aswin@aswin-VirtualBox:~$
```

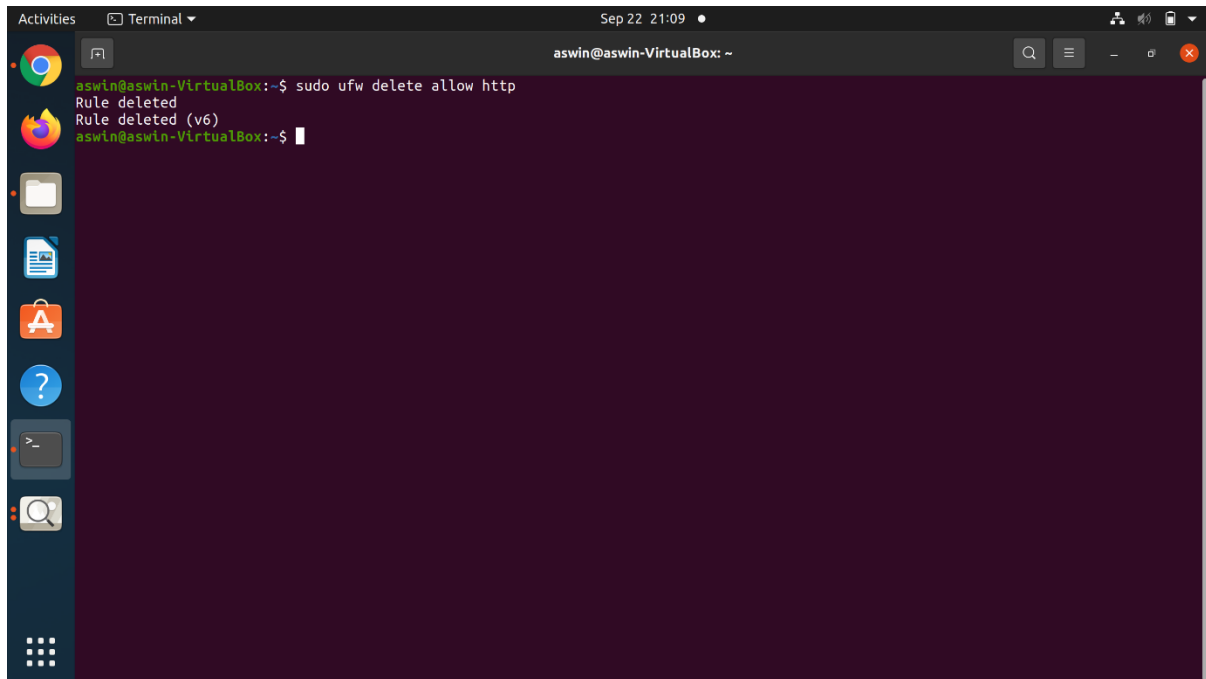
### Deleting the Rules

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 delete allow http`

A screenshot of a Linux terminal window. The window title is 'aswin@aswin-VirtualBox: ~'. The terminal shows the command 'sudo ufw delete allow http' being entered and executed. The output of the command is 'Rule deleted' followed by 'Rule deleted (v6)'. The terminal has a dark purple background and a light-colored cursor. The window is part of a desktop environment with a sidebar on the left containing various application icons.

`sudo ufw delete allow http`

### 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`

```
Activities Terminal Sep 22 21:11 aswin@aswin-VirtualBox: ~
aswin@aswin-VirtualBox:~$ 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)

aswin@aswin-VirtualBox:~$
```

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`

```
Activities Terminal Sep 22 21:11 aswin@aswin-VirtualBox: ~
aswin@aswin-VirtualBox:~$ sudo ufw delete 14
Deleting:
allow 1022
Proceed with operation (y/n)? y
Rule deleted (v6)
aswin@aswin-VirtualBox:~$
```



