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.

Result

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 232, which is nearly 4.3 billion IPv4 addresses for the world to use (4,294,967,296, to be exact).

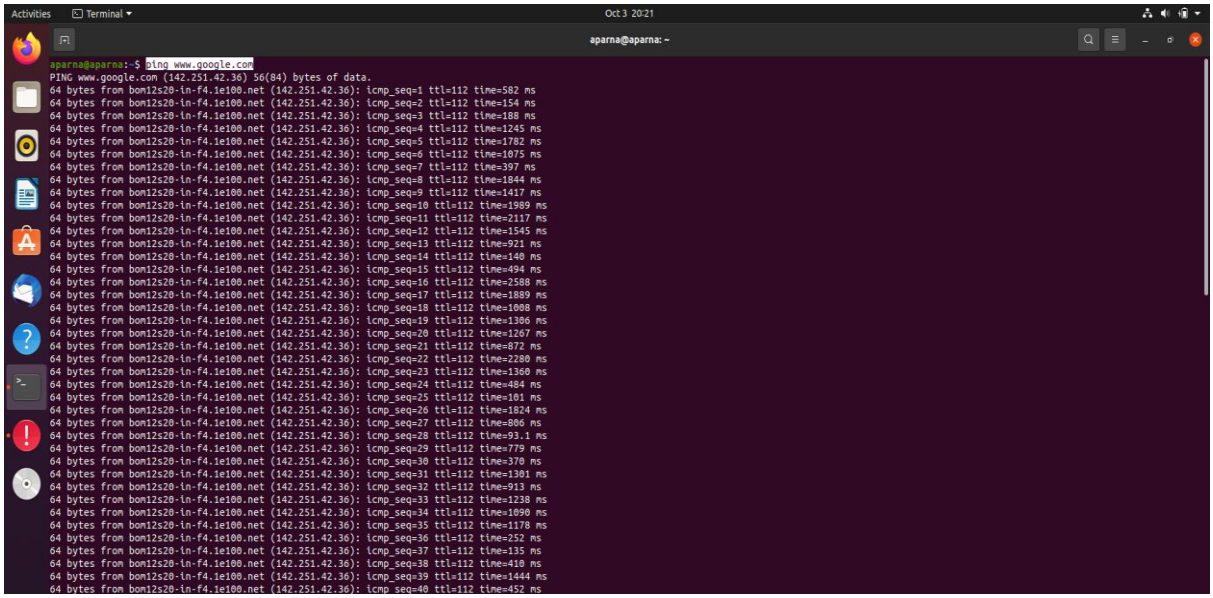
Network Commands

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 used to ping a website or hostname directly.



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:

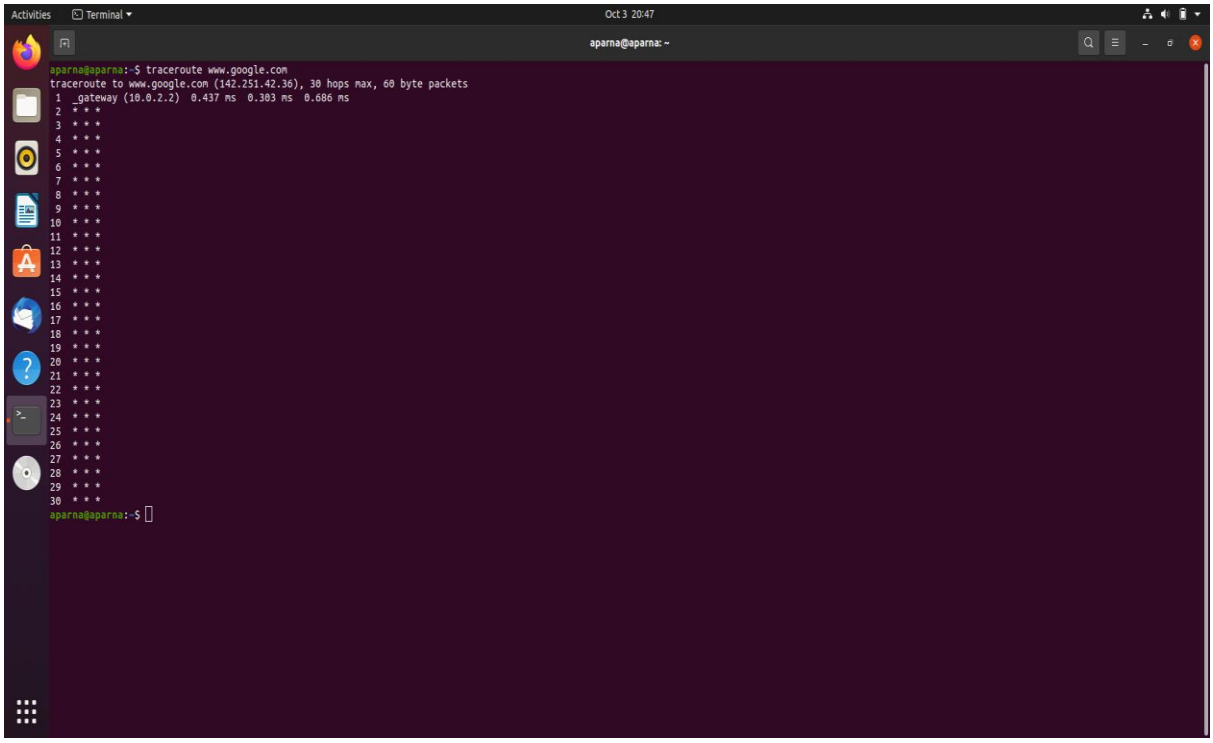
```
route [-f] [-p] [command [destination] [mask subnetmask]
[gateway] [metric costmetric]]
```

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.



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*

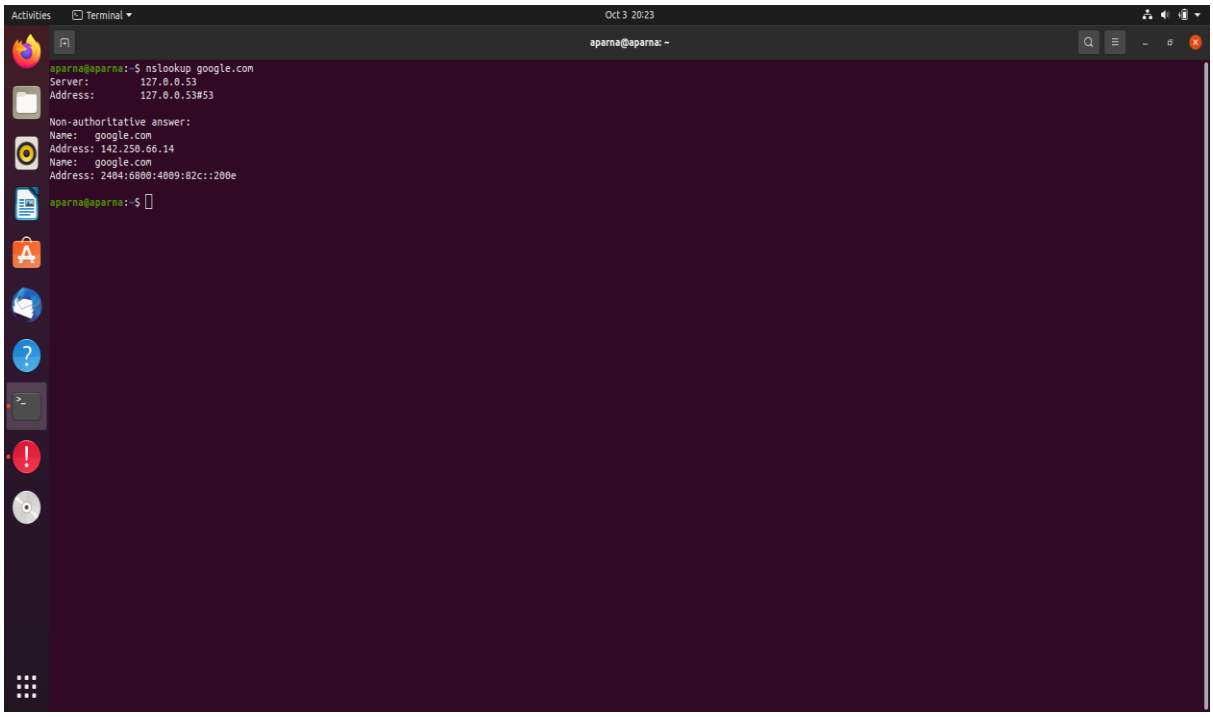
command sends three packets to the hop and each of the time refers to the time taken by the packet to reach the hop.

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.

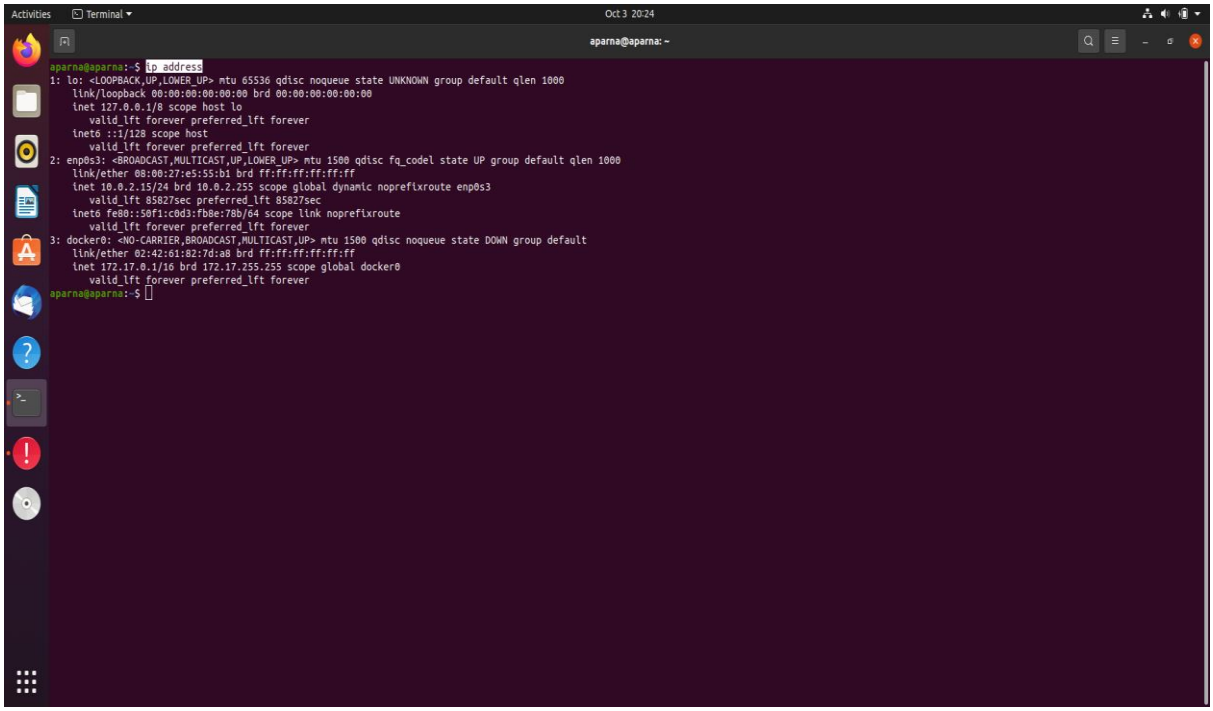
Nslookup: nslookup (stands for "Name Server Lookup") is a useful command for getting information from DNS server. It is a network administration tool for querying the Domain Name System (DNS) to obtain domain name or IP address mapping or any other specific DNS record. It is also used to troubleshoot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

Syntax: *nslookup [option]*



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 }

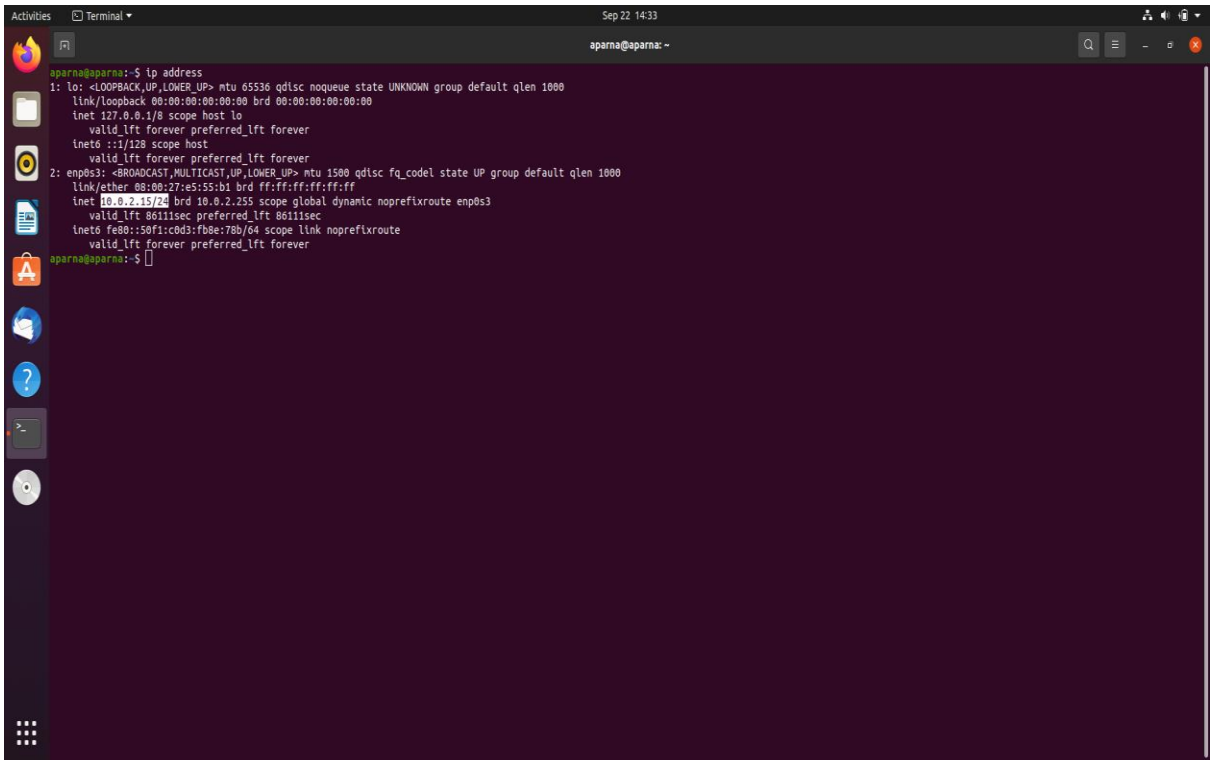


This will show the information related to all interfaces available on our system, but if we want to view the information of any particular interface, add the options show followed by the name of the particular network interface.

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.

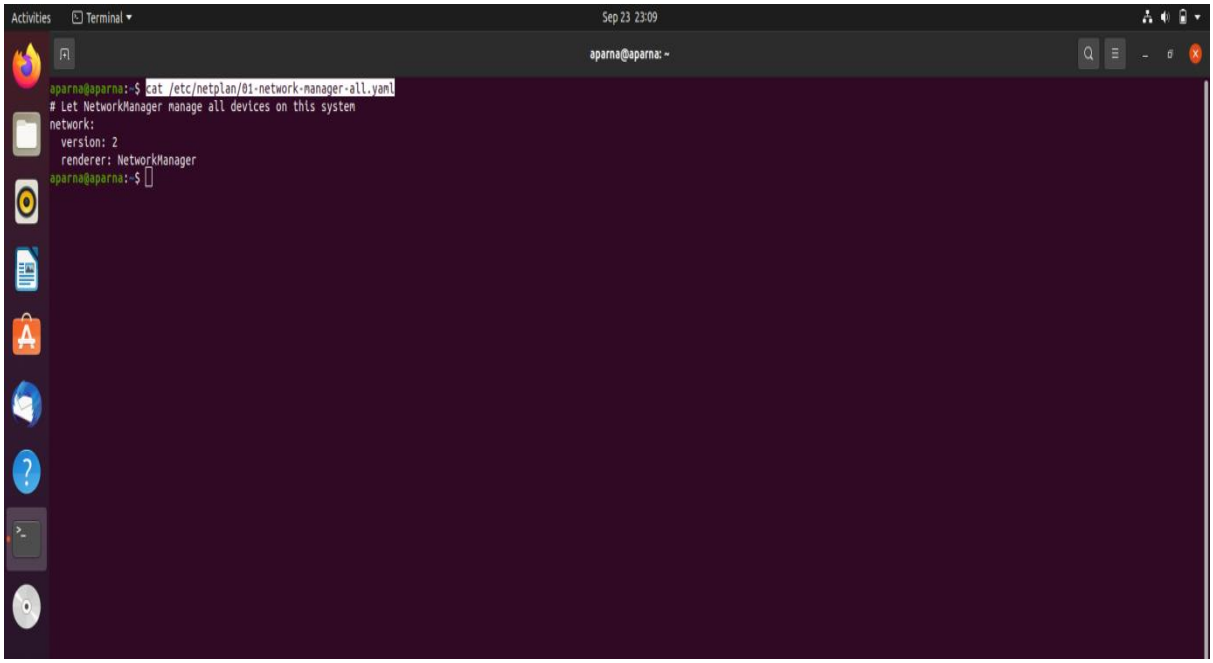
Setting up static IP addresses

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

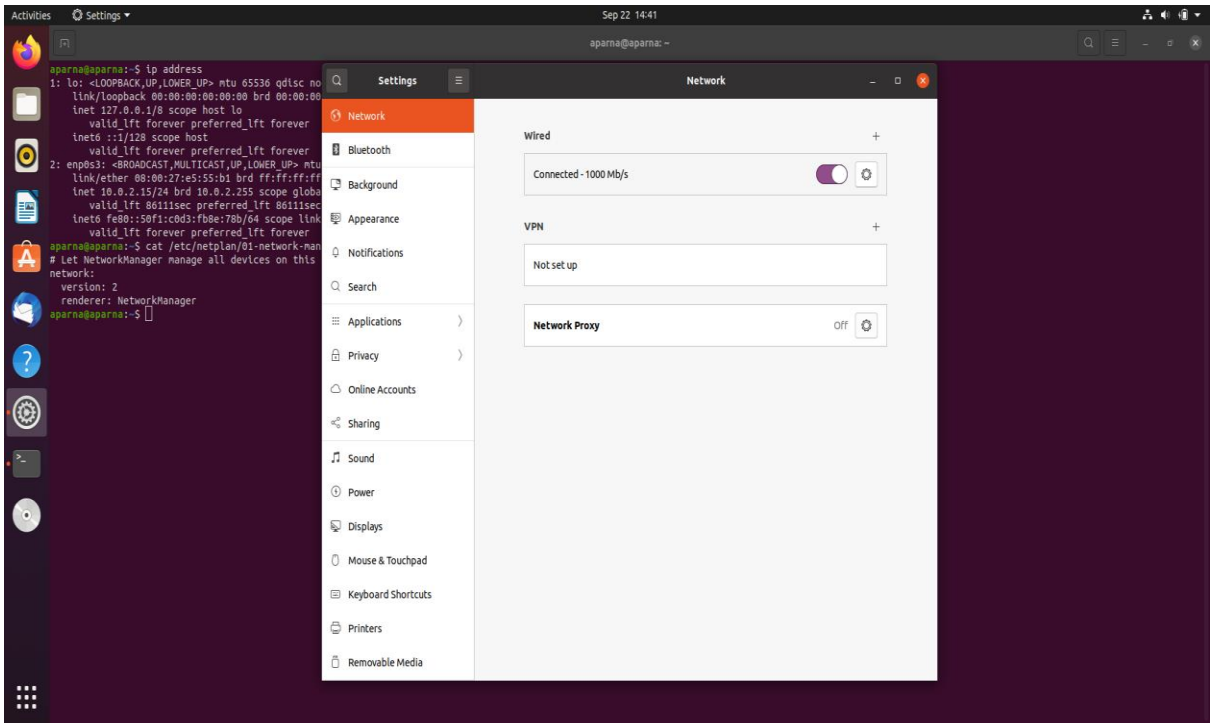


- Step 2 : To view the content of Netplan network configuration file, run the following command:

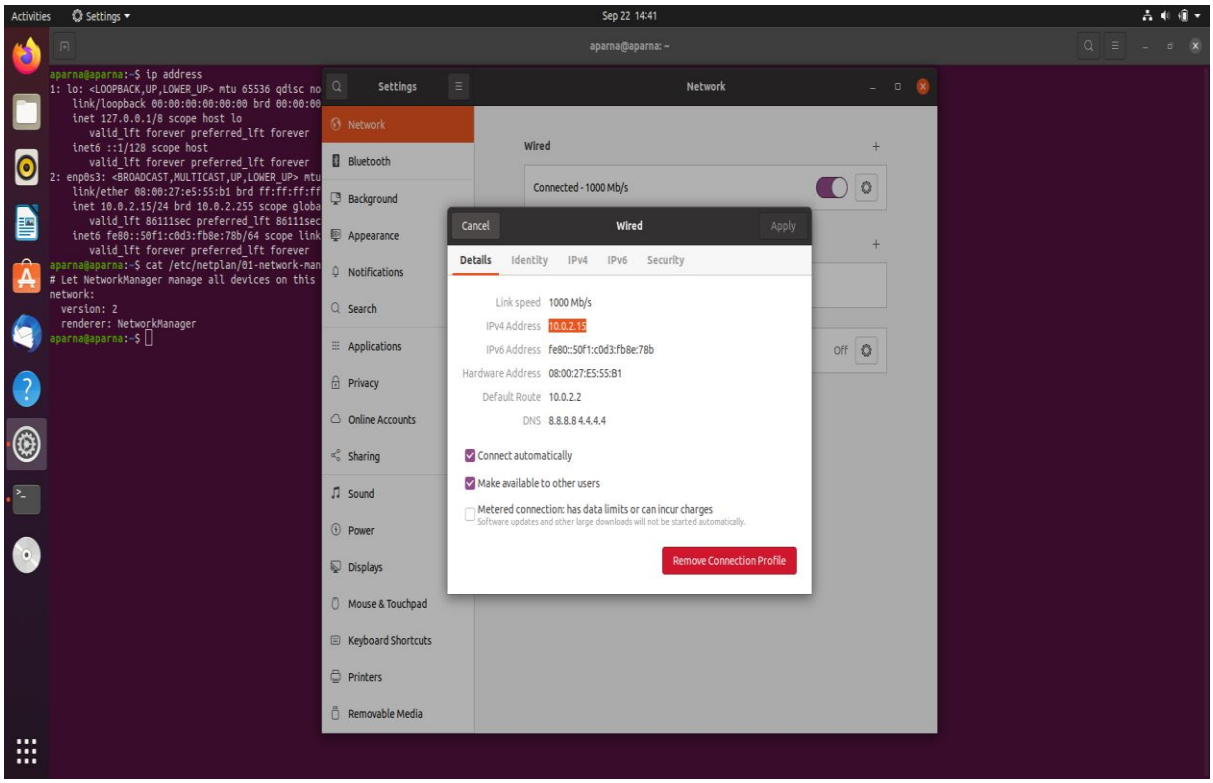
  cat /etc/netplan/01-network-manager-all.yaml



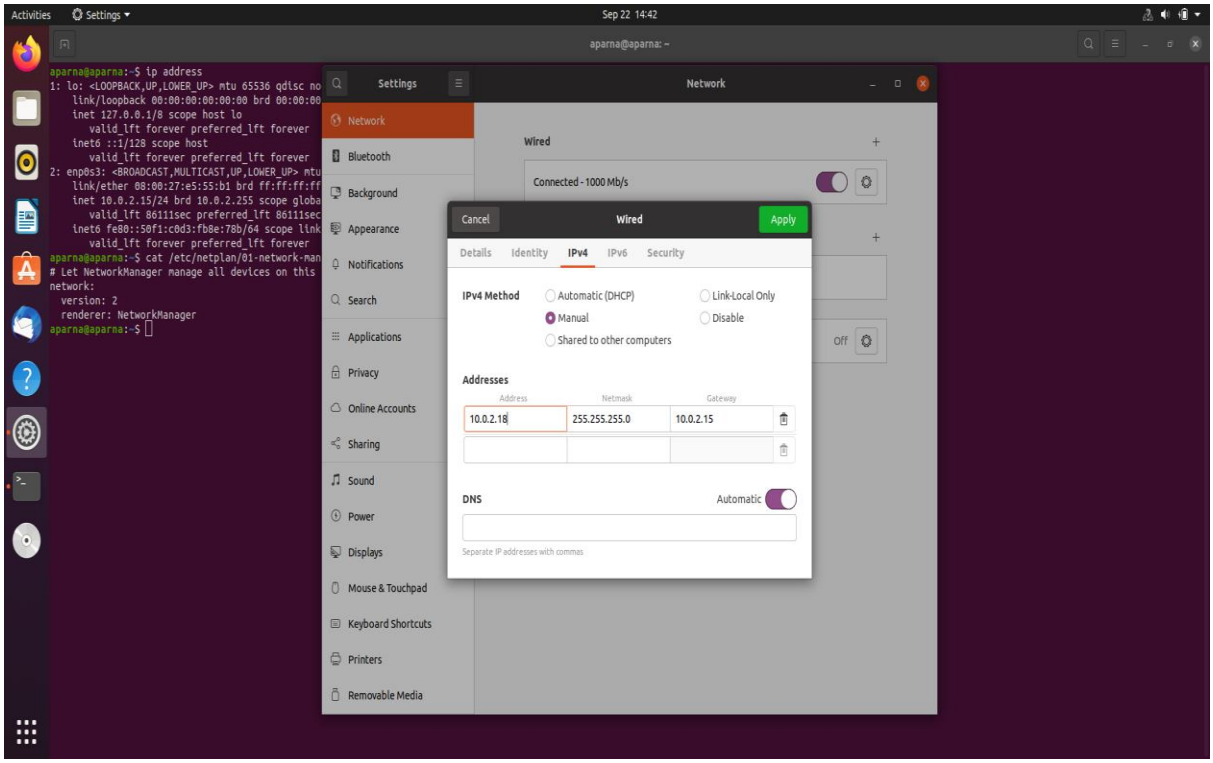
- Step 3 : Click on the top right network icon and select settings of the network interface you wish to configure to use a static IP address on Ubuntu.



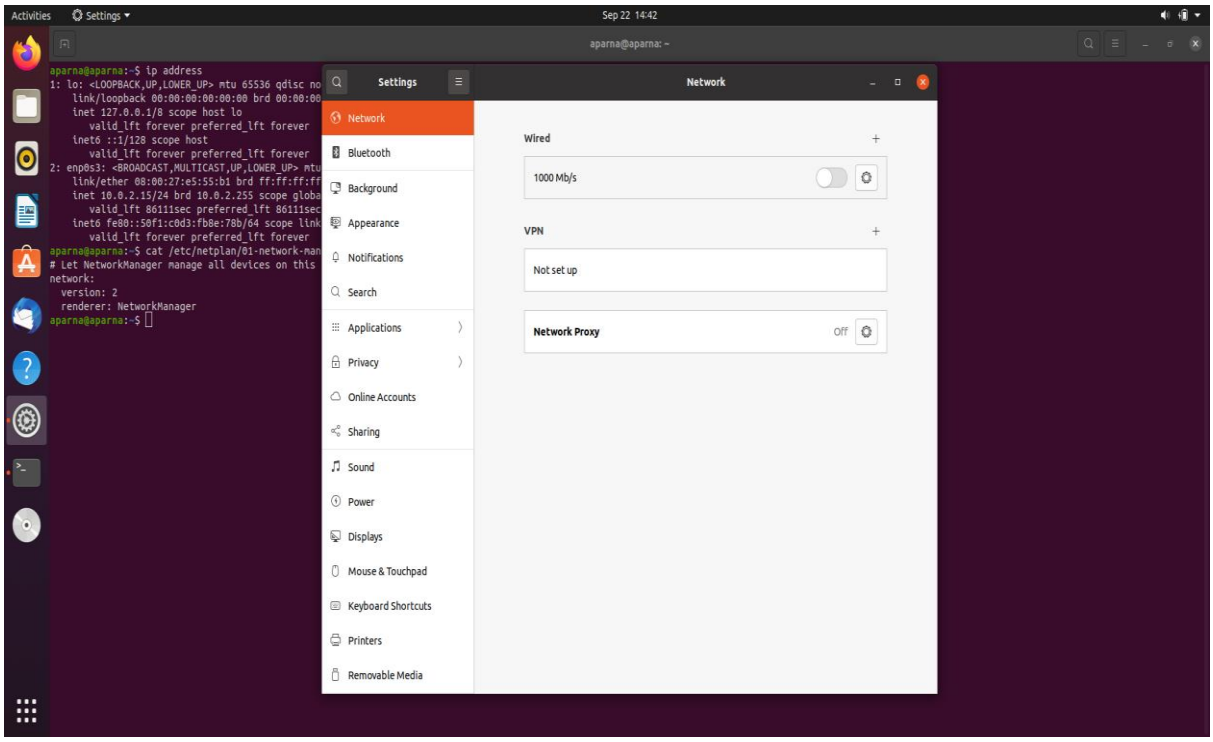
Click on the settings icon to start IP address configuration.



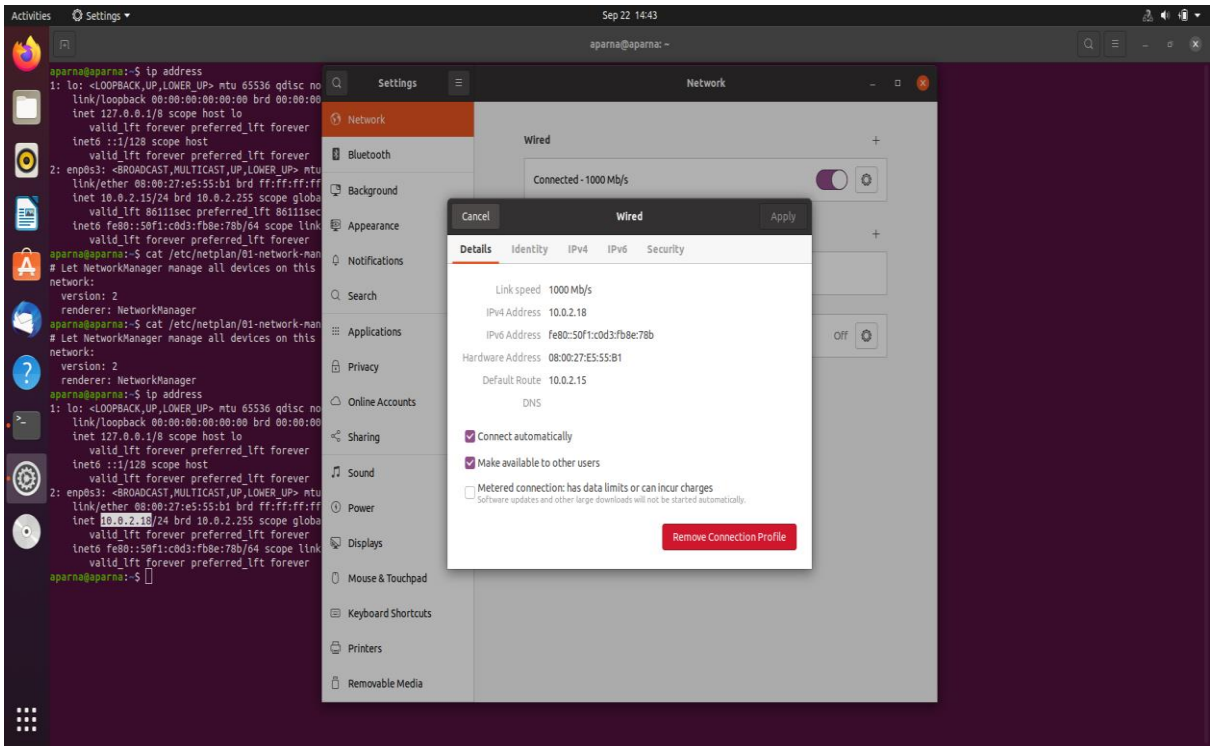
- Step 4 : Select IPv4 tab.Select manual and enter your desired IP address, netmask, gateway and DNS settings. Once ready click Apply button.



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.



Subnet:

A subnet, or subnetwork, is a network inside a network. Subnets make networks more efficient. Through subnetting, network traffic can travel a shorter distance without passing through unnecessary routers to reach its destination.Organizations will use a subnet to subdivide large networks into smaller, more efficient subnetworks. One goal of a subnet is to split a large network into a grouping of smaller, interconnected networks to help minimize traffic. This way, traffic doesn't have to flow through unnecessary routs, increasing network speeds.

CIDR address scheme:

CIDR, which stands for Classless Inter-Domain Routing, is an IP addressing scheme that improves the allocation of IP addresses. It replaces the old system based on classes A, B, and C. This scheme also helped greatly extend the life of IPv4 as well as slow the growth of routing tables.

CIDR is based on variable-length subnet masking (VLSM). This allows it to define prefixes of arbitrary lengths making it much more efficient than the old system. CIDR IP addresses are composed of two sets of numbers. The network address is written as a prefix, like you would see a normal IP address (e.g. 192.255.255.255). The second part is the suffix which indicates how many bits are in the entire address (e.g. /12). Putting it together, a CIDR IP address would look like the following:

192.255.255.255/12

The network prefix is also specified as part of the IP address. This varies depending upon the number of bits required. Therefore, taking the example above, we can say that the first 12 bits are the network part of the address while the last 20 bits are for host addresses.

Subnet mask:

A subnet mask is like an IP address, but for only internal usage within a network. Routers use subnet masks to route data packets to the right place. Subnet masks are not indicated within data packets traversing the Internet — those packets only indicate 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.

•Tables is the name for a set of chains.

•Chain is a collection of rules.

•Rule is condition used to match packet.

•Target is action taken when a possible rule matches. Examples of the target areACCEPT, DROP, QUEUE.

•Policy is the default action taken in case of no match with the inbuilt chains and can be ACCEPT or DROP.

Syntax:

```
iptables --table TABLE -A/-C/-D... CHAIN rule --jump Target
```

TABLE

There are five possible tables:

•**filter:** Default used table for packet filtering. It includes chains like INPUT, OUTPUT and FORWARD.

•**nat :** Related to Network Address Translation. It includes PREROUTING and POSTROUTING chains.

•**mangle :** For specialised packet alteration. Inbuilt chains include PREROUTING and OUTPUT.

•**raw :** Configures exemptions from connection tracking. Built-in chains are PREROUTING and OUTPUT.

•**security :** Used for Mandatory Access Control CHAINS

There are few built-in chains that are included in tables. They are:

•**INPUT :**set of rules for packets destined to localhost sockets.

•**FORWARD :**for packets routed through the device.

•**OUTPUT :**for locally generated packets, meant to be transmitted outside.

•**PREROUTING :**for modifying packets as they arrive.

•**POSTROUTING :**for modifying packets as they are leaving.

Configure and Set Up a Firewall on Ubuntu

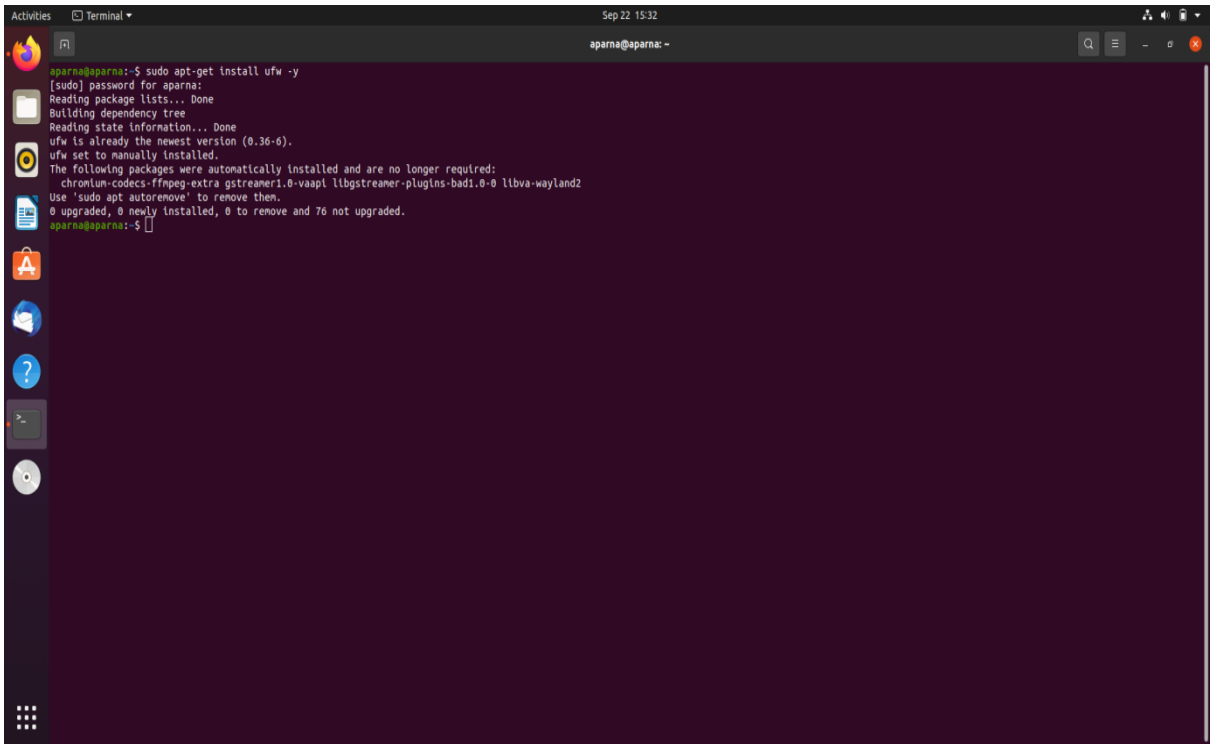
UFW stands for Uncomplicated Firewall which acts as an interface to IPTABLES that simplifies the process of the configuration of firewalls it will be a very hard for a beginners to learns and configure the firewall rules where we will secure the network from unknown users are machines. UFW works on the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

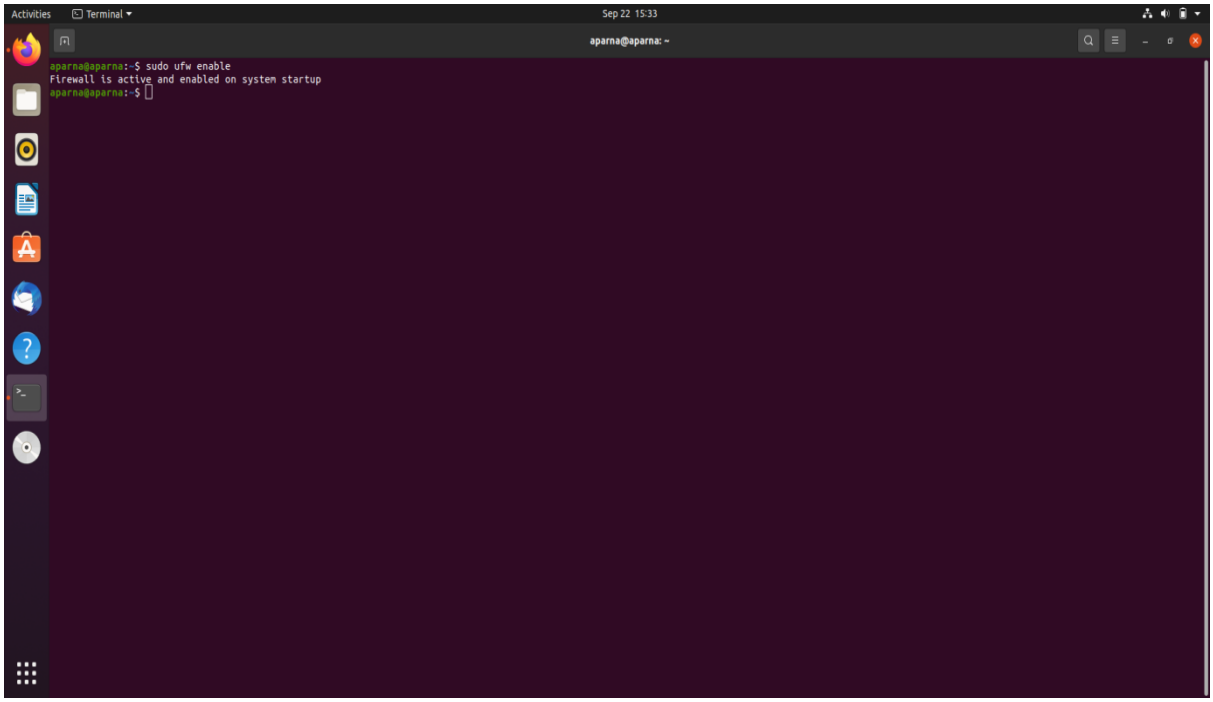
```
sudo apt-get install ufw -y
```



Enabling the UFW (Firewall)

Below is the command to enable the UFW –

```
sudo ufw enable
```

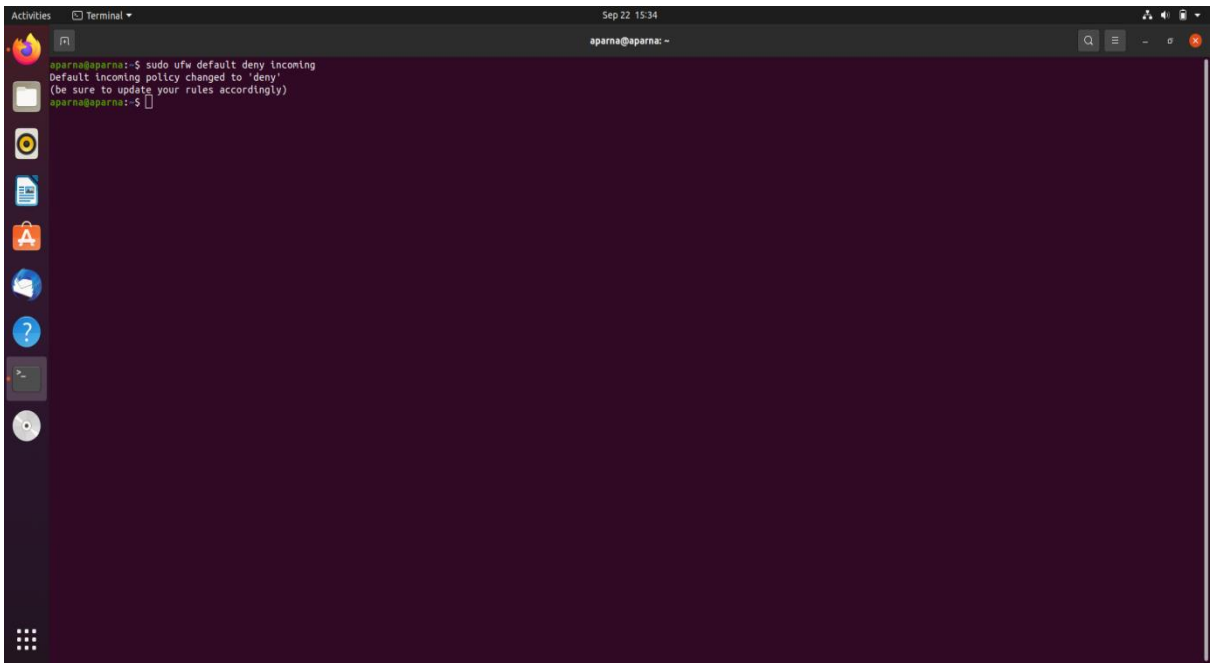


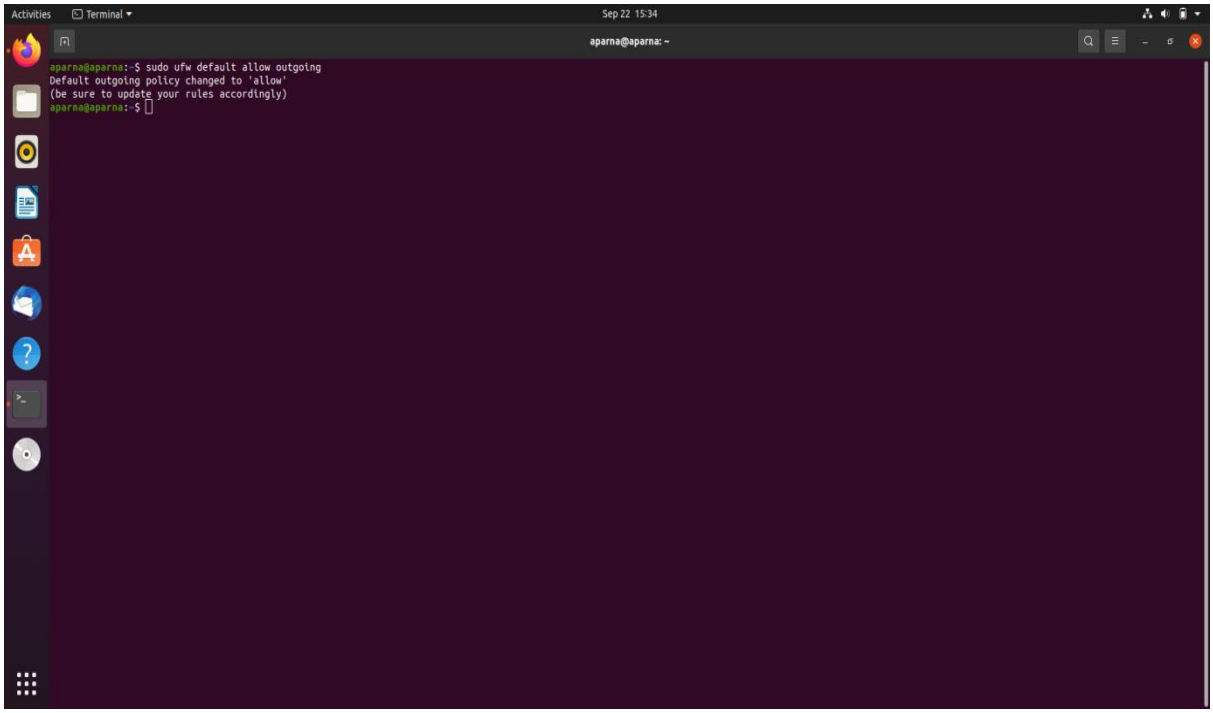
Enabling the Default Policies

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
```

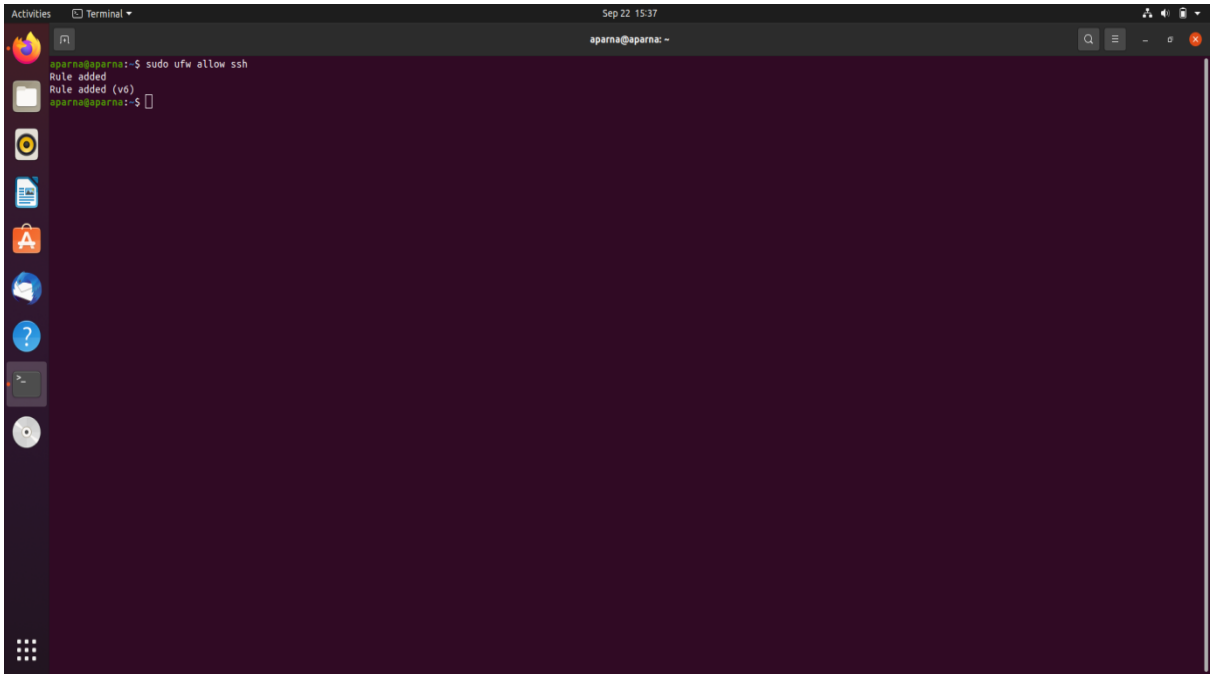




Enabling SSH Connections

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
```

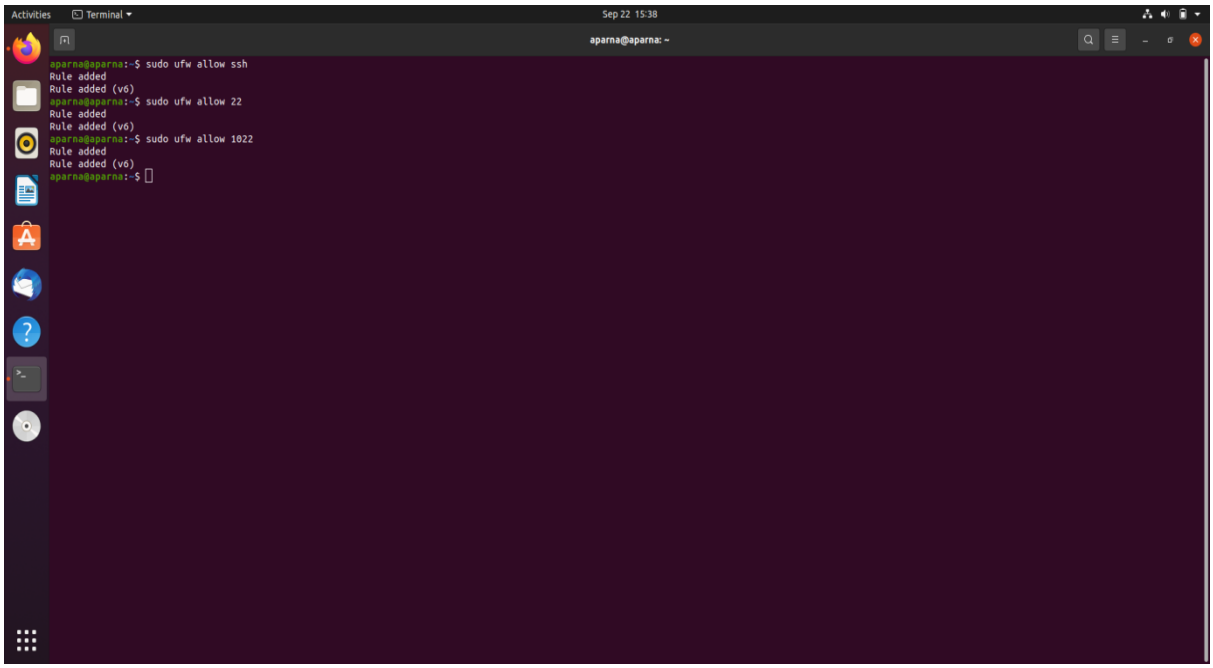


With the above command, the port 22 will be allowed for incoming connections. We can use the below command directly using the port no 22 to allow the SSH connections.

```
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 –

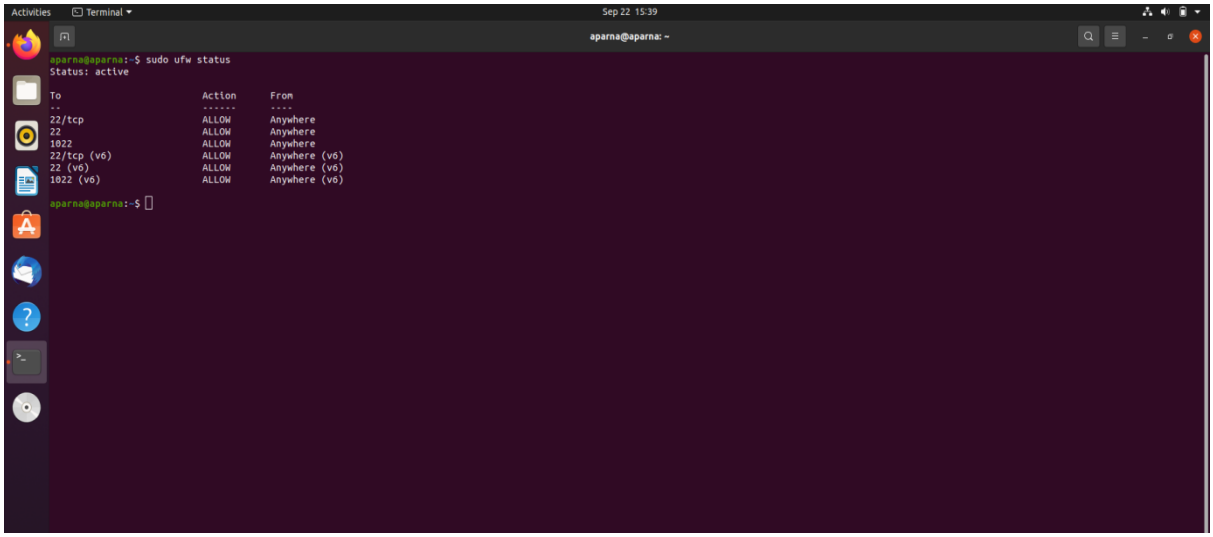
```
sudo ufw allow 1022
```



Checking the UFW (Firewall) Status

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

```
sudo ufw status
```



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

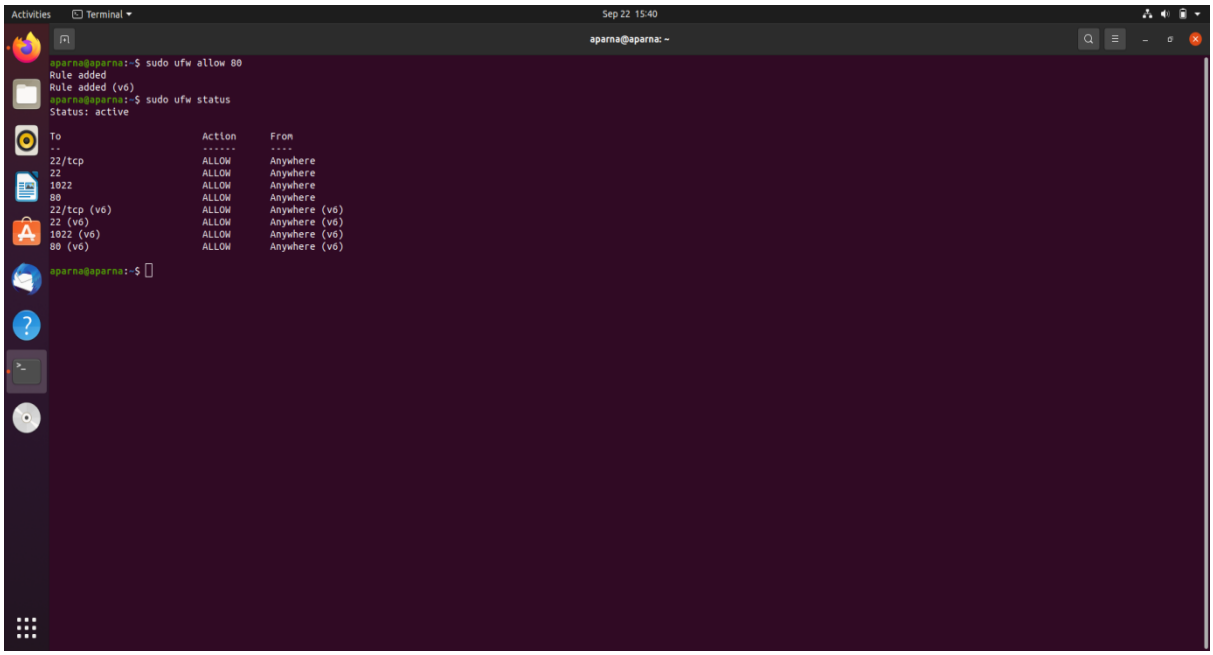
At this point, we will allow others to connect to the server for the regular ports like HTPP, HTTPS, and FTP ports respectively.

HTTP port 80

```
sudo ufw allow 80
```

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

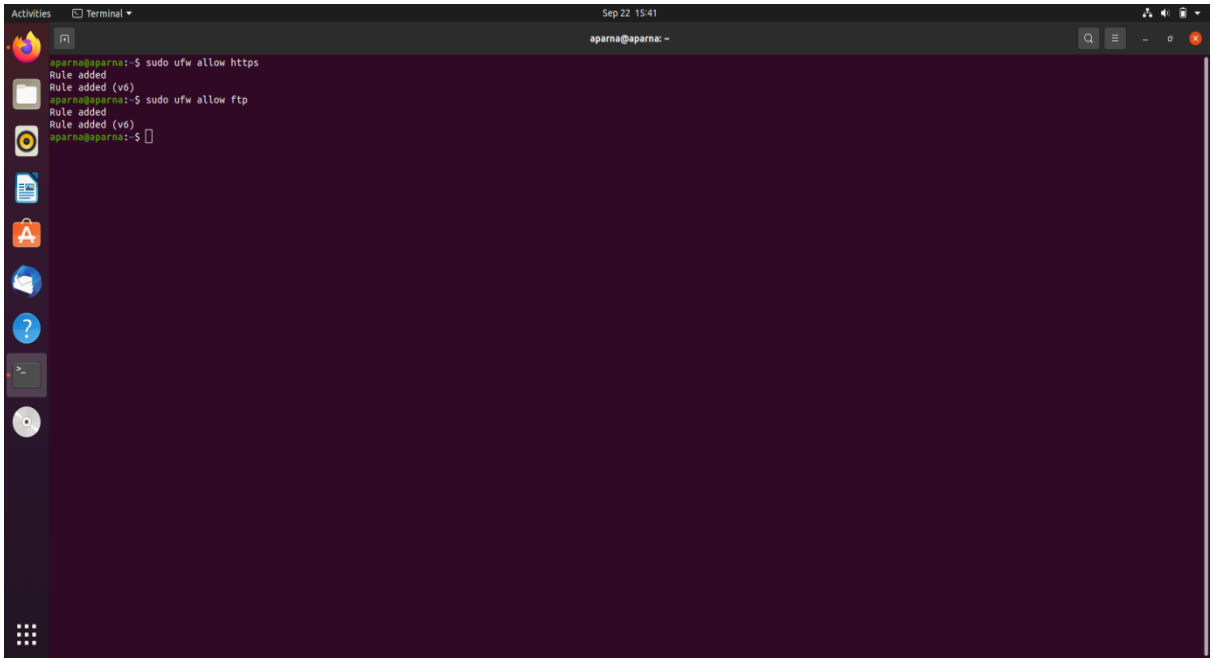
```
sudo ufw status
```



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

```
sudo ufw allow https
```

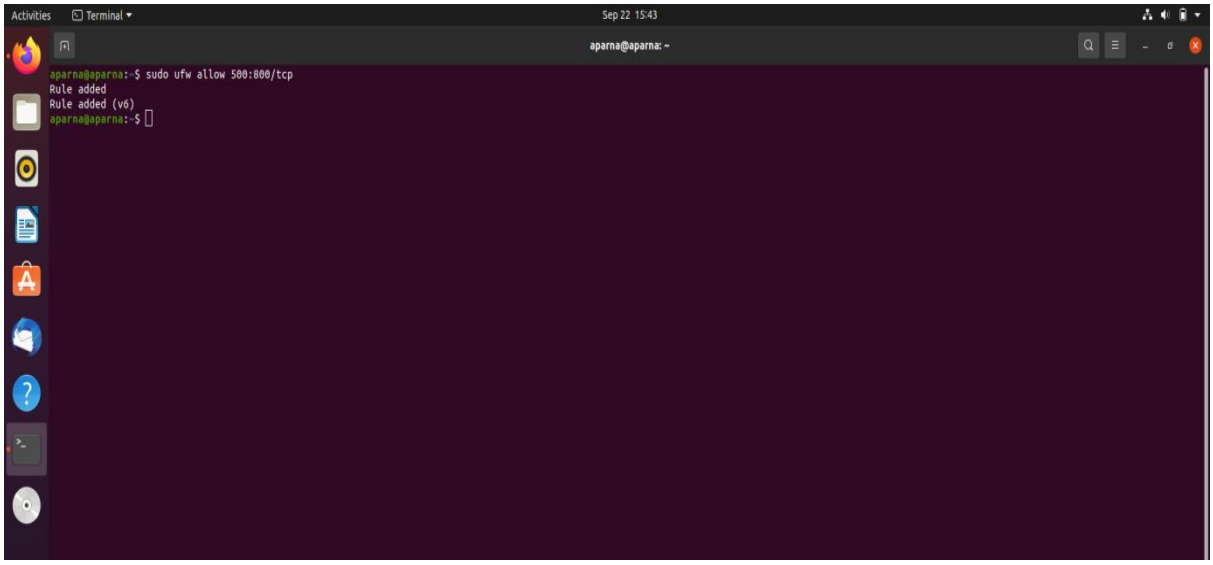
```
sudo ufw allow ftp
```



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
```



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.

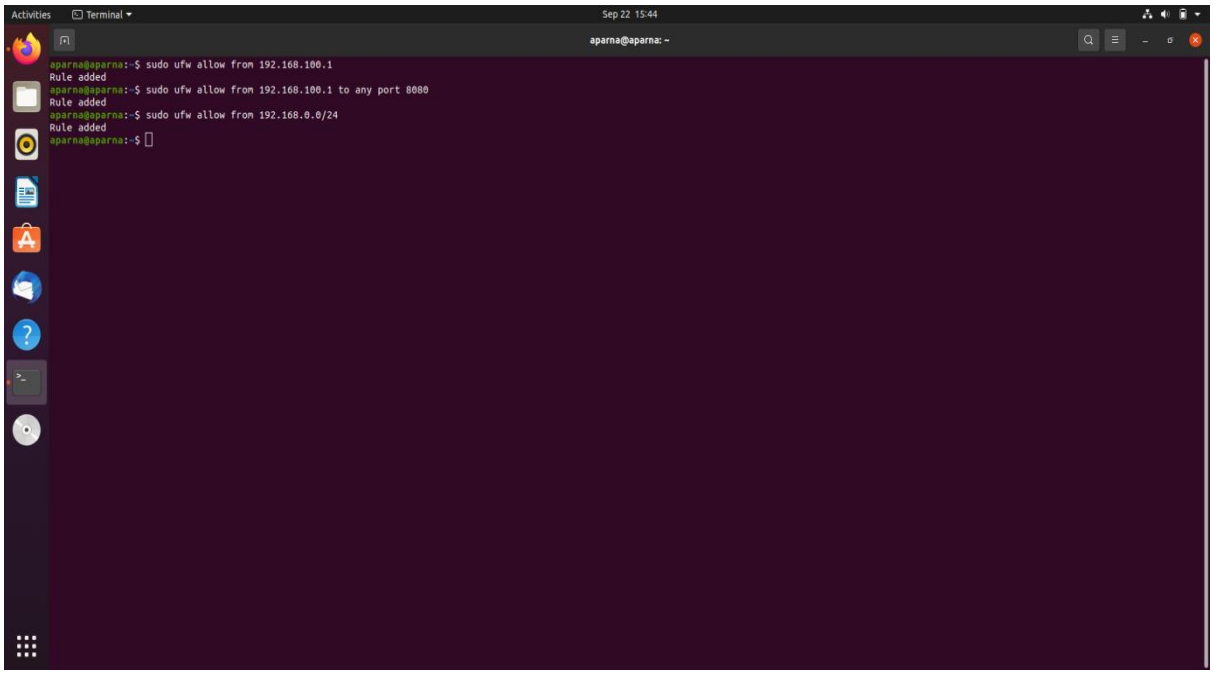
```
sudo ufw allow from 192.168.100.1
```

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



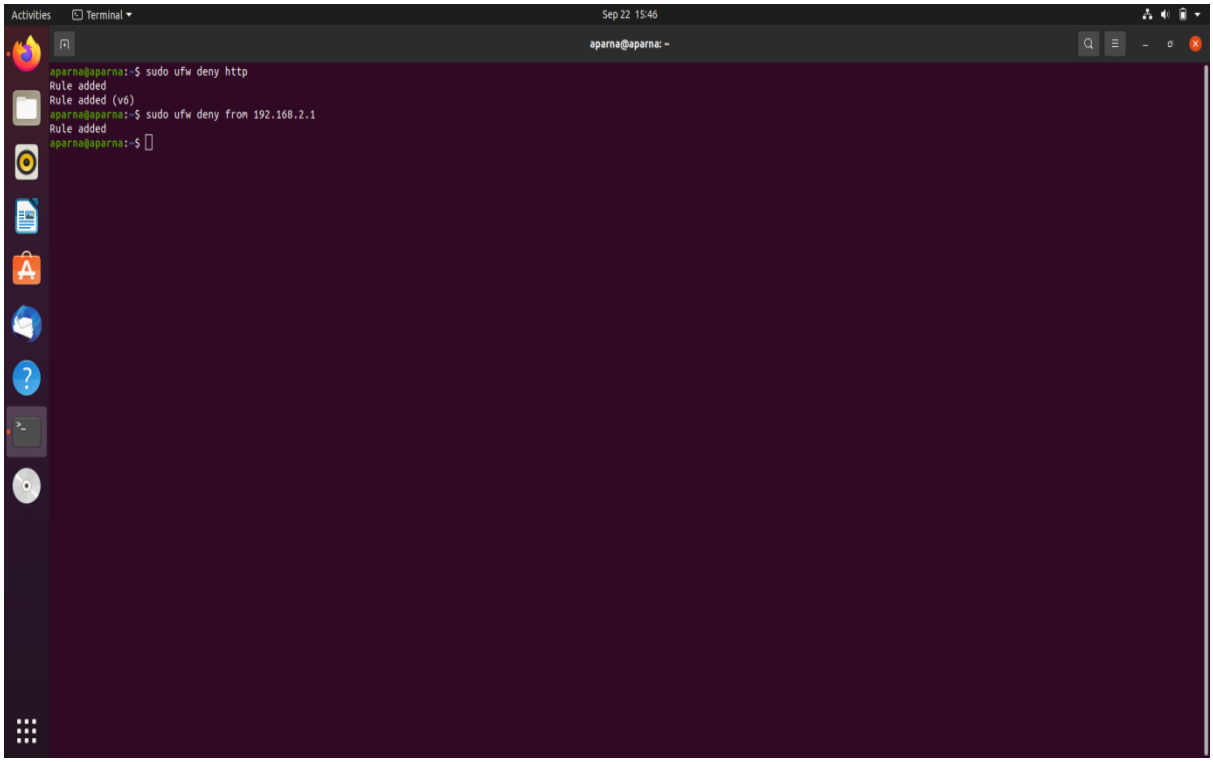
Deny the Connections or Rules

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



Deleting the Rules

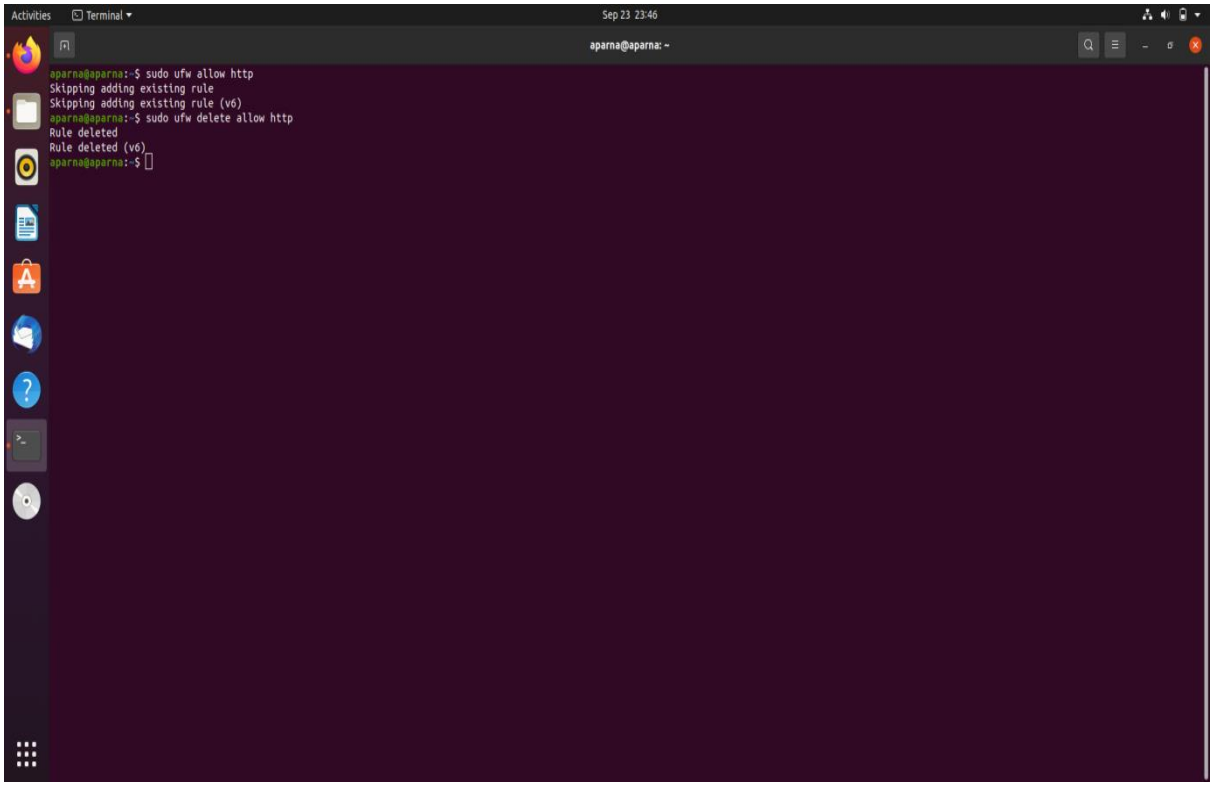
We can delete the rules in two ways one with the actual rules and other with the rules numbers.

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
```

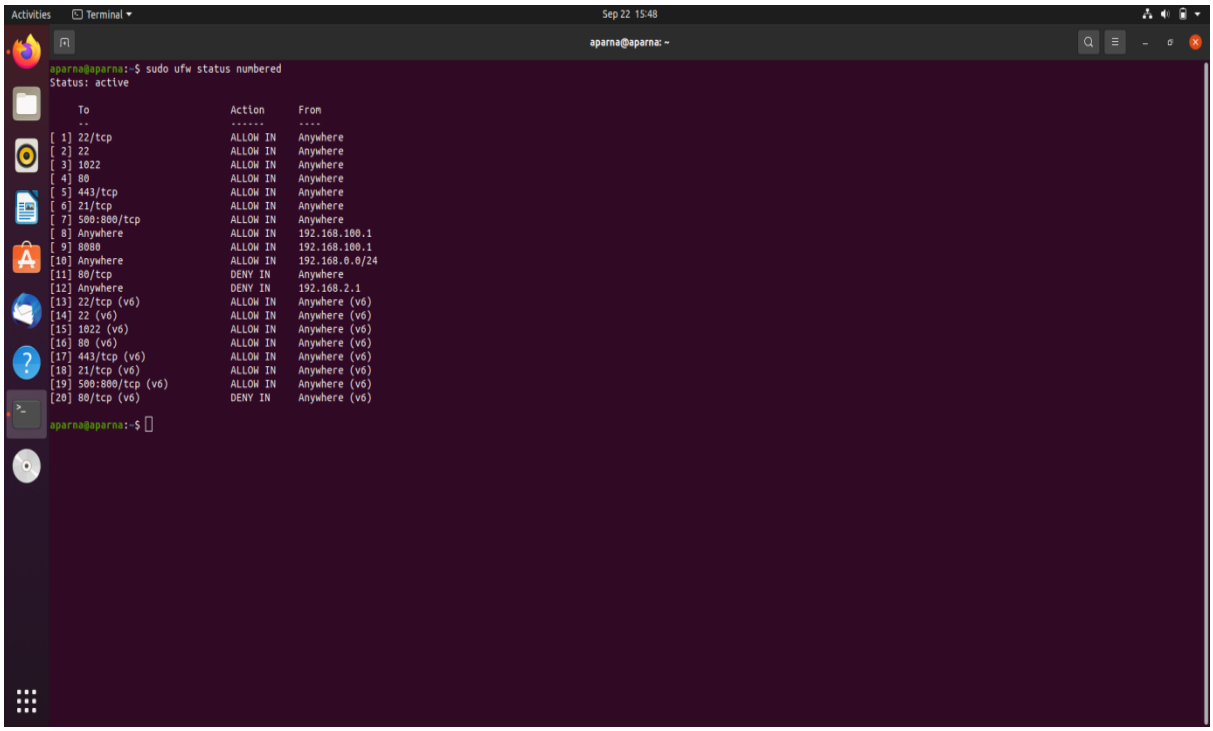
```
sudo ufw delete allow http
```



Rules Number

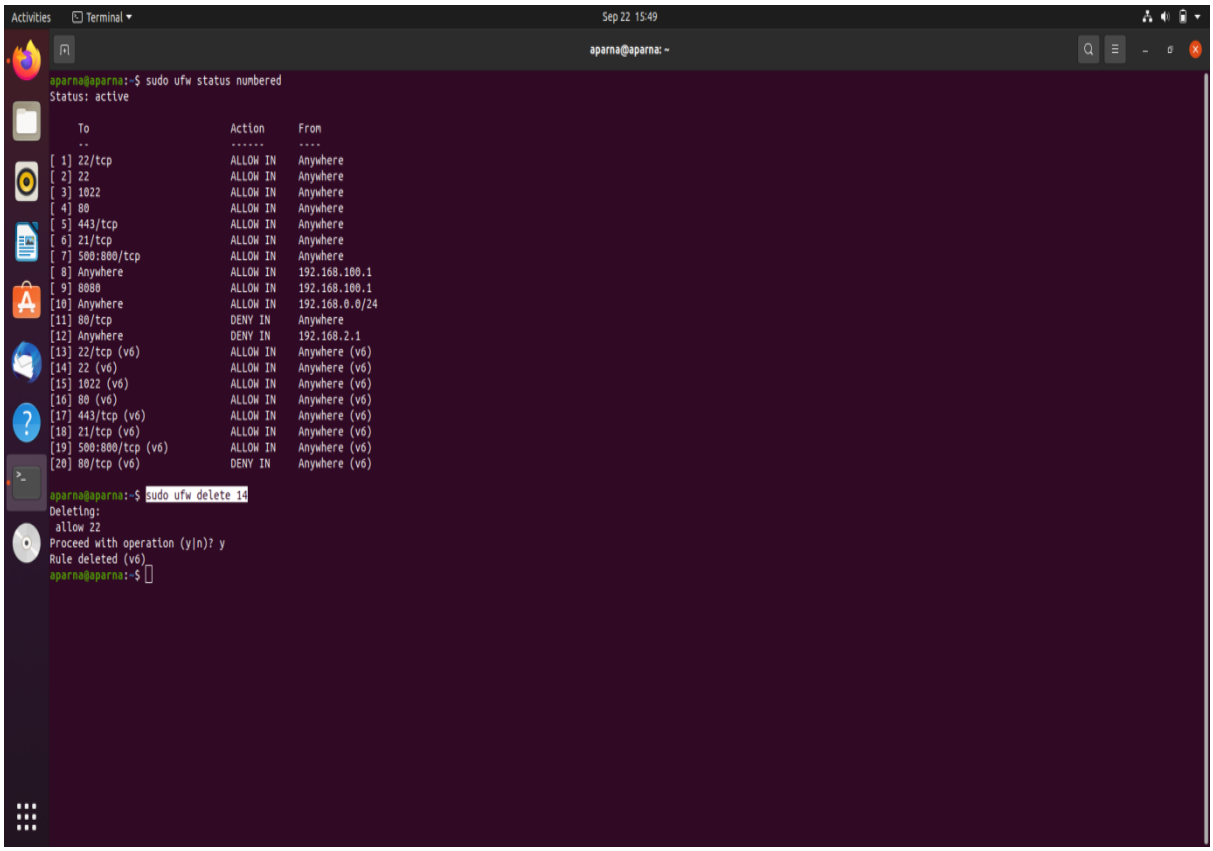
We can use the Rules numbers to delete the firewall rules, we can get the list of firewall rules with the below command.

sudo ufw status numbered



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
```

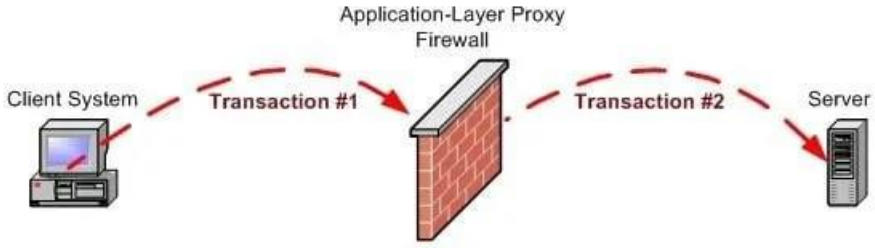


Application layer Proxies

Application Layer Proxy is any service or server that acts as a proxy for client computer requests at the application's protocols. For example, in Microsoft Proxy Server, the Web Proxy Service is an application layer proxy for the Hypertext Transfer Protocol (HTTP), Secure Hypertext Transfer Protocol (S-HTTP), File Transfer Protocol (FTP), and Gopher protocols. Application layer proxies provide security by hiding internal network addresses from the outside world.

Application layer proxies provide more support for the additional capabilities of each protocol than do circuit layer proxies. For example, application layer proxies can support virus scanning. Application layer proxies are also client-neutral and require no special software components or operating system on the client computer to enable the client to communicate with servers on the Internet using the proxy server.

Microsoft Proxy Server can grant users access to selected application layer protocols and can restrict access to remote Web sites by domain name, IP address, and subnet mask.



Transaction is split in two: to client, firewall appears to be the server (transaction #1); to server, firewall appears to be the client (transaction #2)