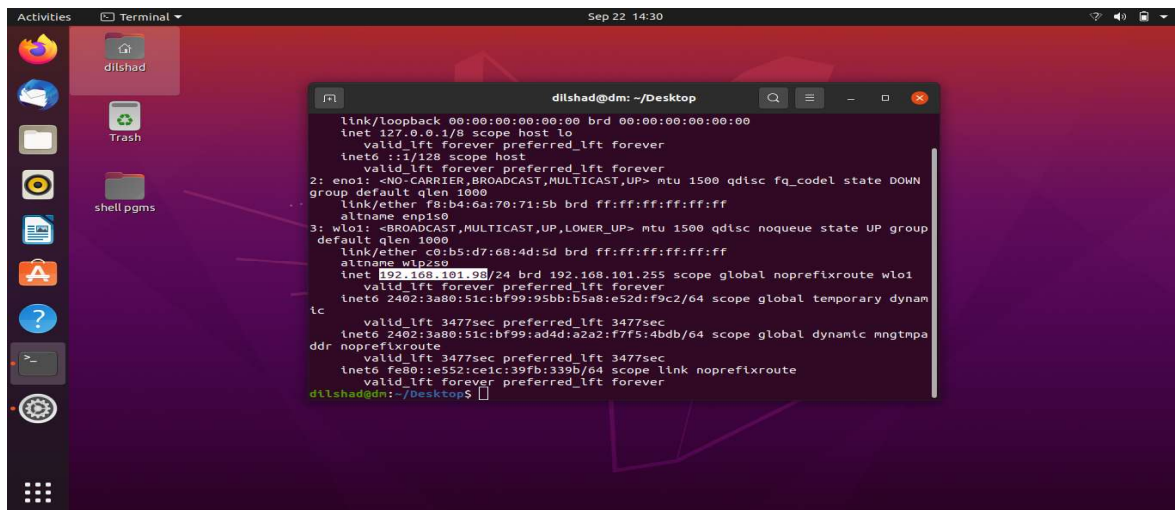


# Experiment 8

## Ipv4 –networking

Step 1 :

Type ip address in the terminal and find out current ip address of network

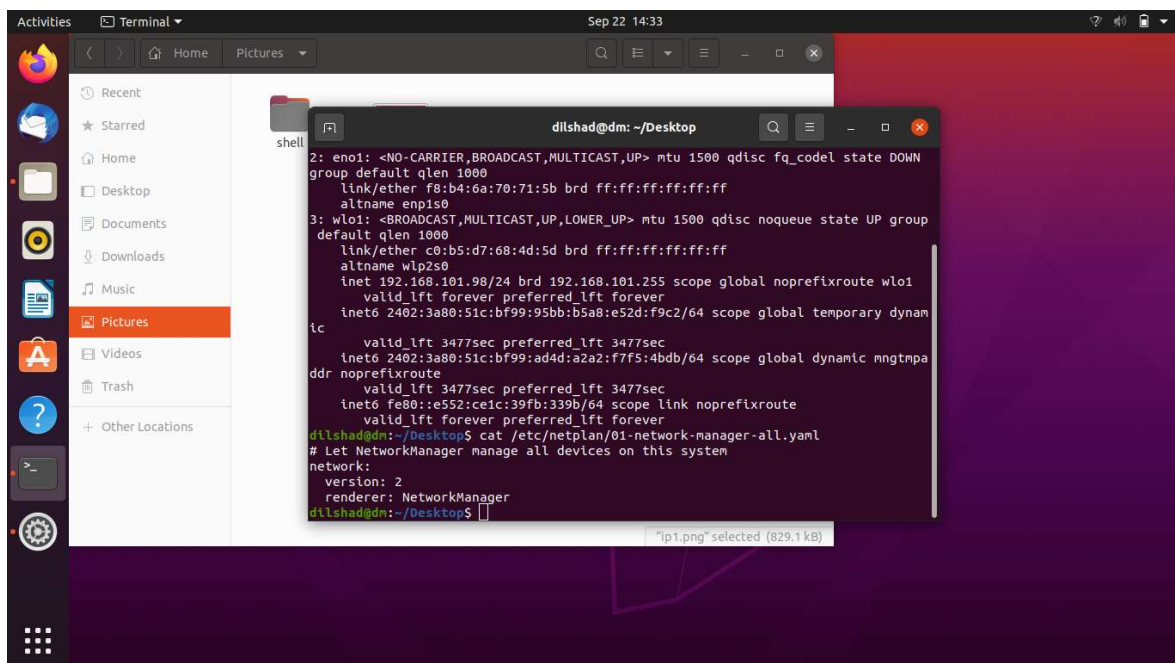


The screenshot shows a Linux desktop with a terminal window open. The terminal displays the output of the 'ip' command, showing network configuration details for the 'lo' and 'enp1s0' interfaces. The 'lo' interface is configured with IP address 127.0.0.1 and netmask 255.0.0.0. The 'enp1s0' interface is configured with IP address 192.168.101.255 and netmask 255.255.255.0. The terminal also shows the status of the 'wlp2s0' interface, which is currently down.

```
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 f8:b4:6a:70:71:5b 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 c0:b5:d7:68:4d:5d brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.101.255/24 brd 192.168.101.255 scope global noprefixroute wlo1
        valid_lft forever preferred_lft forever
    inet6 2402:3a80:51c:bf99:95bb:b5a8:e52d:f9c2/64 scope global temporary dynam
tc
    valid_lft 3477sec preferred_lft 3477sec
    inet6 2402:3a80:51c:bf99:ad4d:a2a2:f7f5:4bdb/64 scope global dynamic mngmtppa
ddr noprefixroute
    valid_lft 3477sec preferred_lft 3477sec
    inet6 fe80::e552:ce1c:39fb:339b/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
dilshad@dm:~/Desktop$
```

Step 2 :

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

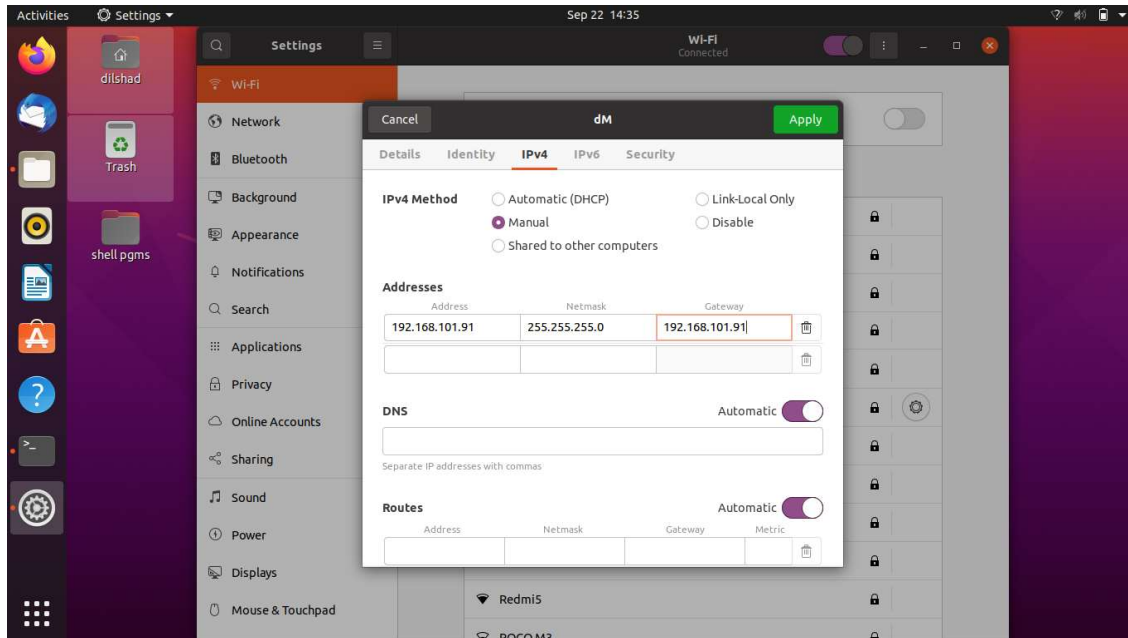


The screenshot shows a Linux desktop with a terminal window and a file manager window. The terminal displays the output of the 'cat /etc/netplan/01-network-manager-all.yaml' command, showing the network configuration for the system. The file manager window shows the contents of the '/etc/netplan/' directory, with the file '01-network-manager-all.yaml' selected.

```
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN
group default qlen 1000
    link/ether f8:b4:6a:70:71:5b 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 c0:b5:d7:68:4d:5d brd ff:ff:ff:ff:ff:ff
    altname wlp2s0
    inet 192.168.101.98/24 brd 192.168.101.255 scope global noprefixroute wlo1
        valid_lft forever preferred_lft forever
    inet6 2402:3a80:51c:bf99:95bb:b5a8:e52d:f9c2/64 scope global temporary dynam
tc
    valid_lft 3477sec preferred_lft 3477sec
    inet6 2402:3a80:51c:bf99:ad4d:a2a2:f7f5:4bdb/64 scope global dynamic mngmtppa
ddr noprefixroute
    valid_lft 3477sec preferred_lft 3477sec
    inet6 fe80::e552:ce1c:39fb:339b/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
dilshad@dm:~/Desktop$ cat /etc/netplan/01-network-manager-all.yaml
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
dilshad@dm:~/Desktop$
```

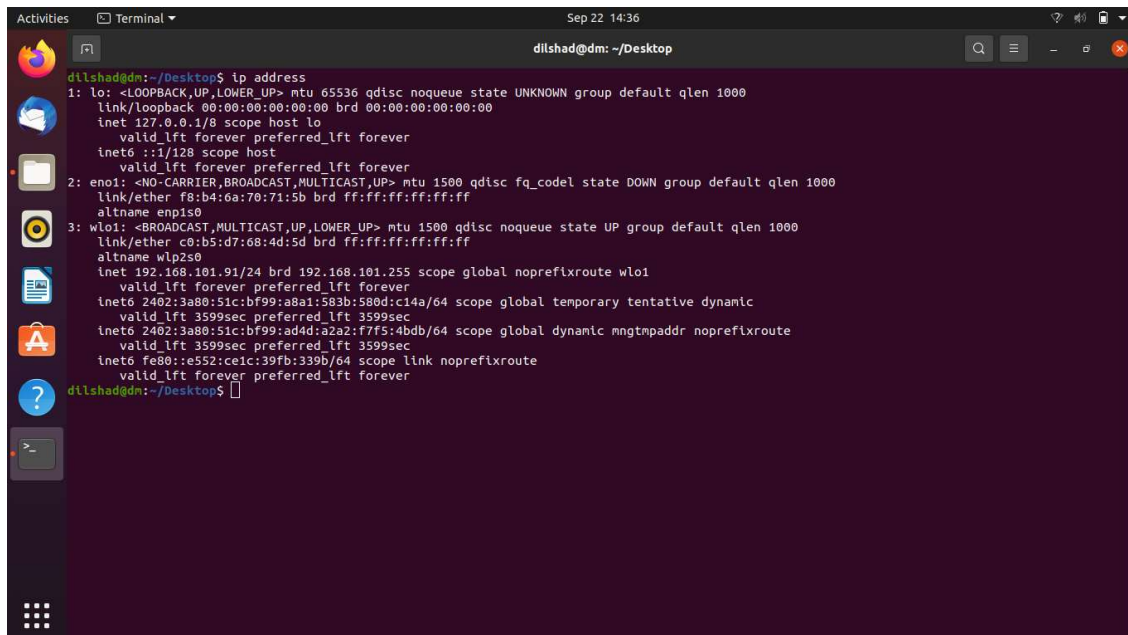
Step 3 :

Open wifi connection settings and click on ipv4 and change to manual from automatic. Change the current ip address and click apply.



Step 4:

Ip address get changed.



## **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 learn and configure the firewall rules where we will secure the network from unknown users and 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 handle 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 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
```

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
```