Lab Project - 4

Objective: Neetworking with Linux

DURATION: 2 - 3 Hourse

PRE-REQUISITES:

Oracle VirtualBox or VMWare, Ubuntu installed.

Lab 1: Basic Network Configuration and Testing

Objective:

• Understand how to configure and test basic network settings on a Linux system.

Tasks:

1. Check Network Interfaces:

   o Use ip or ifconfig to list all available network interfaces on the system.

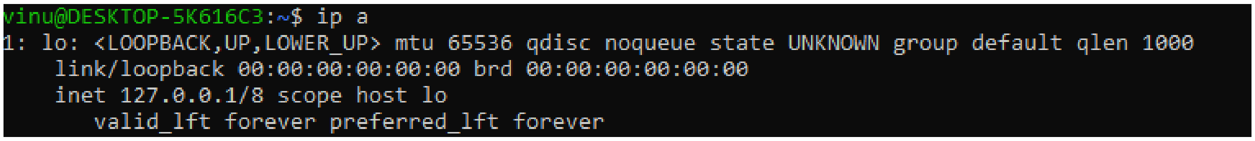
bash

Copy code

ip a

#or

ifconfig



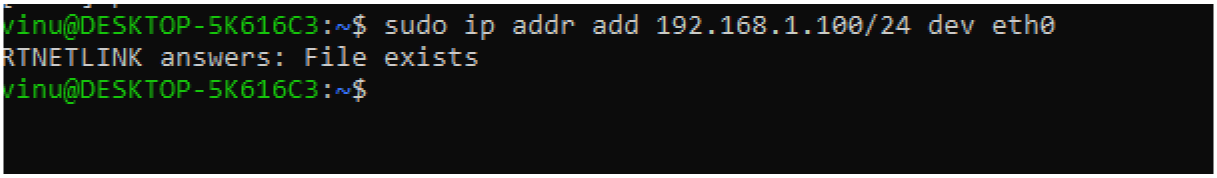
```
vinu@DESKTOP-5K616C3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
```

2. Configure IP Address Manually:

O Use the ip command to assign a static IP address to an interface.
bash

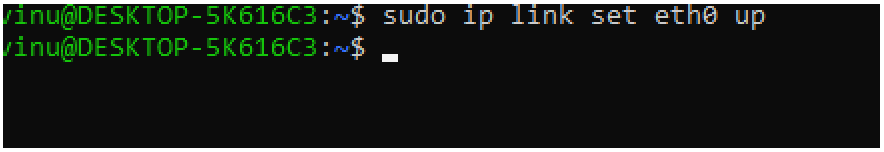
Copy code

sudo ip addr add 192.168.1.100/24 dev eth0

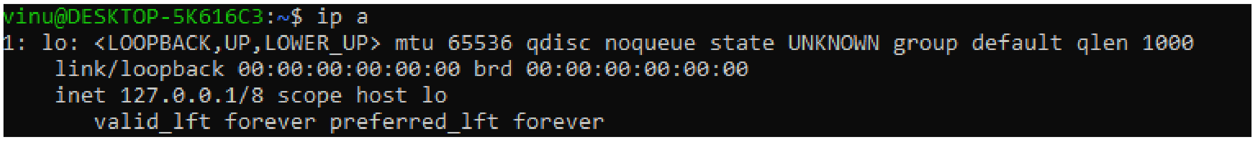


```
vinu@DESKTOP-5K616C3:~$ sudo ip addr add 192.168.1.100/24 dev eth0
RTNETLINK answers: File exists
vinu@DESKTOP-5K616C3:~$
```

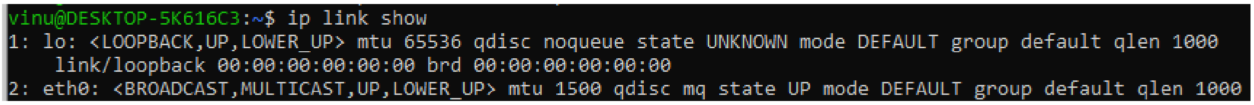
sudo ip link set eth0 up



```
vinu@DESKTOP-5K616C3:~$ sudo ip link set eth0 up
vinu@DESKTOP-5K616C3:~$
```



```
vinu@DESKTOP-5K616C3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
```



```
vinu@DESKTOP-5K616C3:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
```

1. Check Existing Network Interfaces

Run the following command to list all available network interfaces:

```bash
ip addr show
```

or

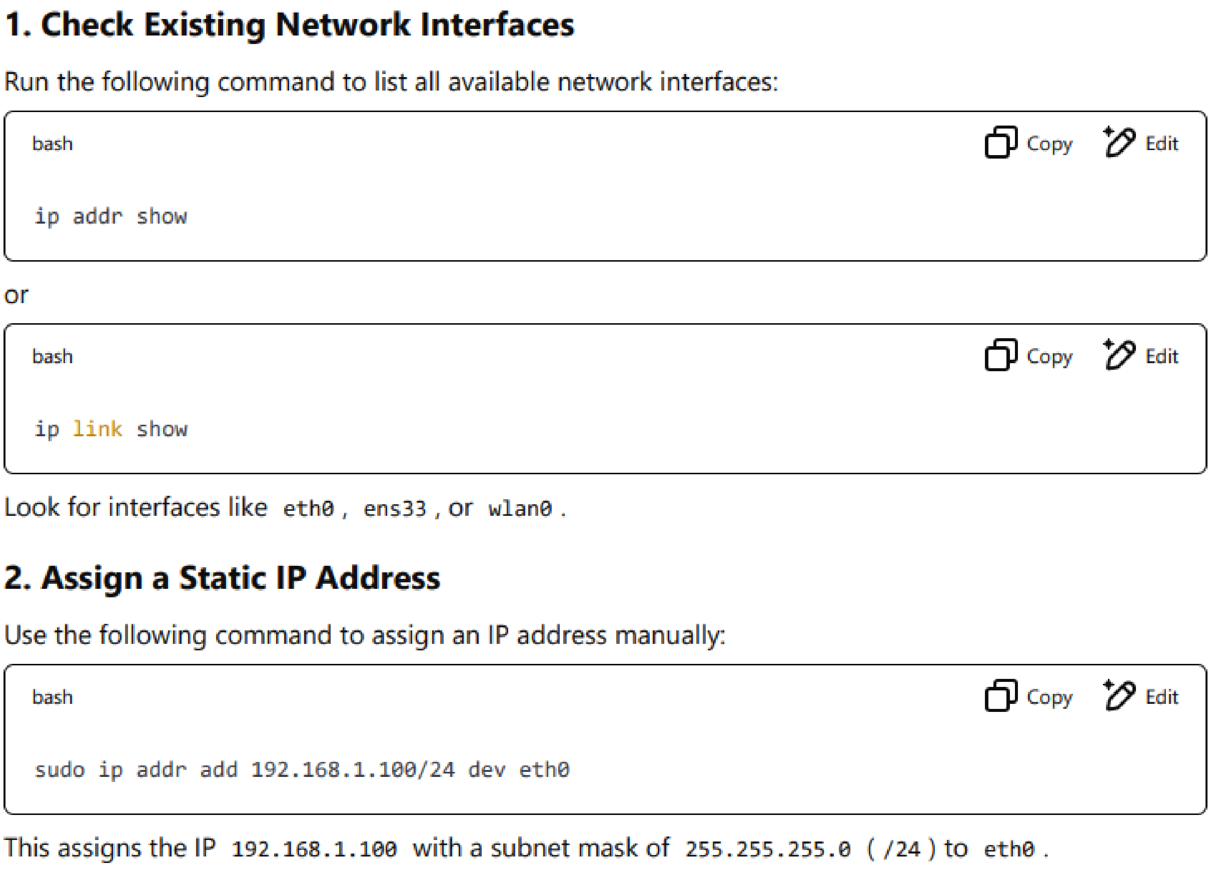
```bash
ip link show
```

Look for interfaces like `eth0` , `ens33` , or `wlan0` .

2. Assign a Static IP Address

Use the following command to assign an IP address manually:

```bash
sudo ip addr add 192.168.1.100/24 dev eth0
```



1. Check Existing Network Interfaces

Run the following command to list all available network interfaces:

```bash
ip addr show
```

or

```bash
ip link show
```

Look for interfaces like `eth0`, `ens33`, or `wlan0`.

2. Assign a Static IP Address

Use the following command to assign an IP address manually:

```bash
sudo ip addr add 192.168.1.100/24 dev eth0
```

This assigns the IP `192.168.1.100` with a subnet mask of `255.255.255.0` (`/24`) to `eth0`.

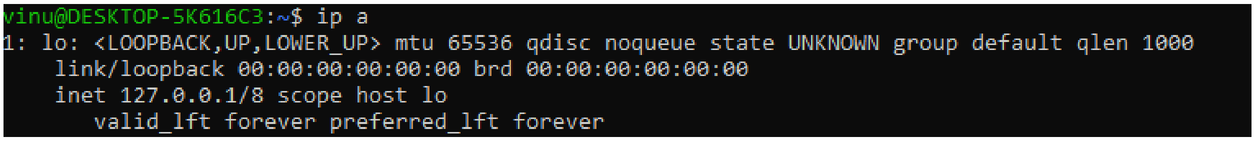
3. Verify the Configuration:

   - Verify the IP address configuration using ip or ifconfig.

bash

Copy code

ip a



```
vinu@DESKTOP-5K616C3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
```

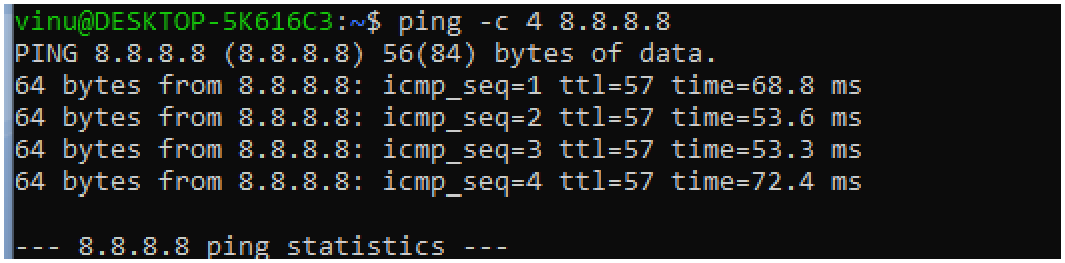
4. Test the Network Connectivity:

o Use ping to test the network connectivity between the local machine and a remote host.

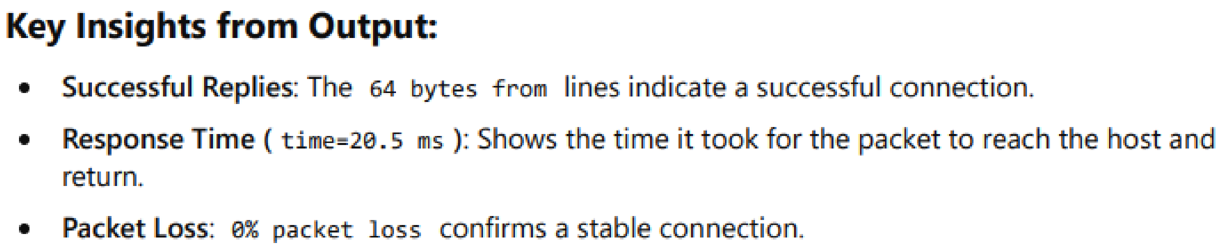
bash

Copy code

ping -c 4 8.8.8.8



```
vinu@DESKTOP-5K616C3:~$ ping -c 4 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=57 time=68.8 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=57 time=53.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=57 time=53.3 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=57 time=72.4 ms

--- 8.8.8.8 ping statistics ---
```



Key Insights from Output:

- **Successful Replies**: The `64 bytes from` lines indicate a successful connection.
- **Response Time** (`time=20.5 ms`): Shows the time it took for the packet to reach the host and return.
- **Packet Loss**: `0% packet loss` confirms a stable connection.

5. Configure Default Gateway:

o Use ip to add a default gateway for routing.

bash

Copy code

sudo ip route add default via 192.168.1.1

1. Check Current Routing Table

Before adding a default gateway, check the current routing table to see existing r

bash

```
ip route show
```

2. Add a Default Gateway

Use the following command to set the default gateway, replacing `<GATEWAY_IP>`
address of your gateway:

bash

```
ip route add default via <GATEWAY_IP>
```

For example, if your gateway is `192.168.1.1` , the command would be:

bash

```
ip route add default via 192.168.1.1
```

3. Specify the Network Interface (Optional)

If you need to specify a particular interface, add the `dev` option:

```bash
ip route add default via 192.168.1.1 dev eth0
```

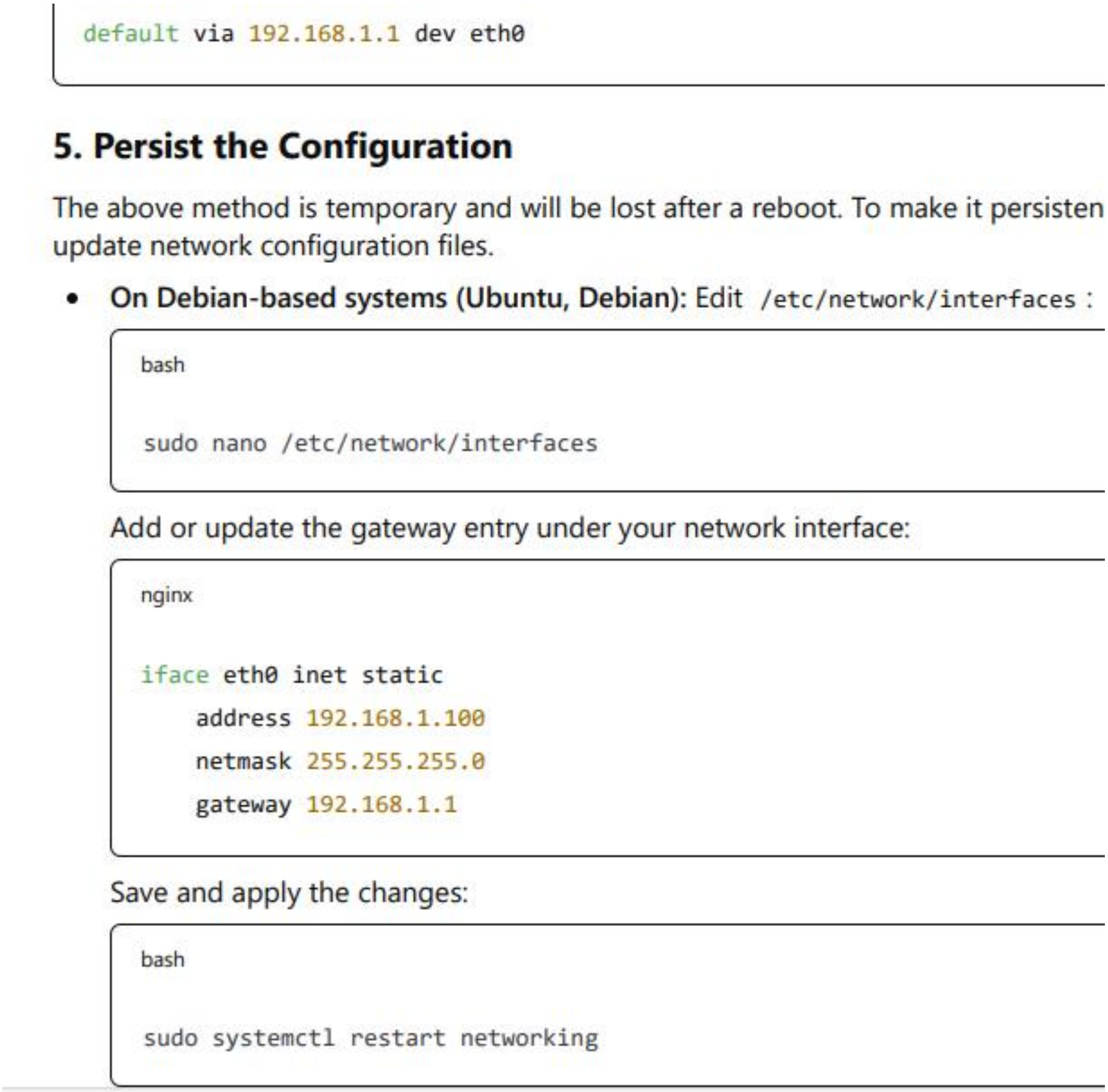
4. Verify the New Route

After adding the route, confirm that it has been added successfully:

```bash
ip route show
```

You should see a line like:

```nginx
```



```
default via 192.168.1.1 dev eth0
```

5. Persist the Configuration

The above method is temporary and will be lost after a reboot. To make it persisten update network configuration files.

- **On Debian-based systems (Ubuntu, Debian):** Edit `/etc/network/interfaces` :

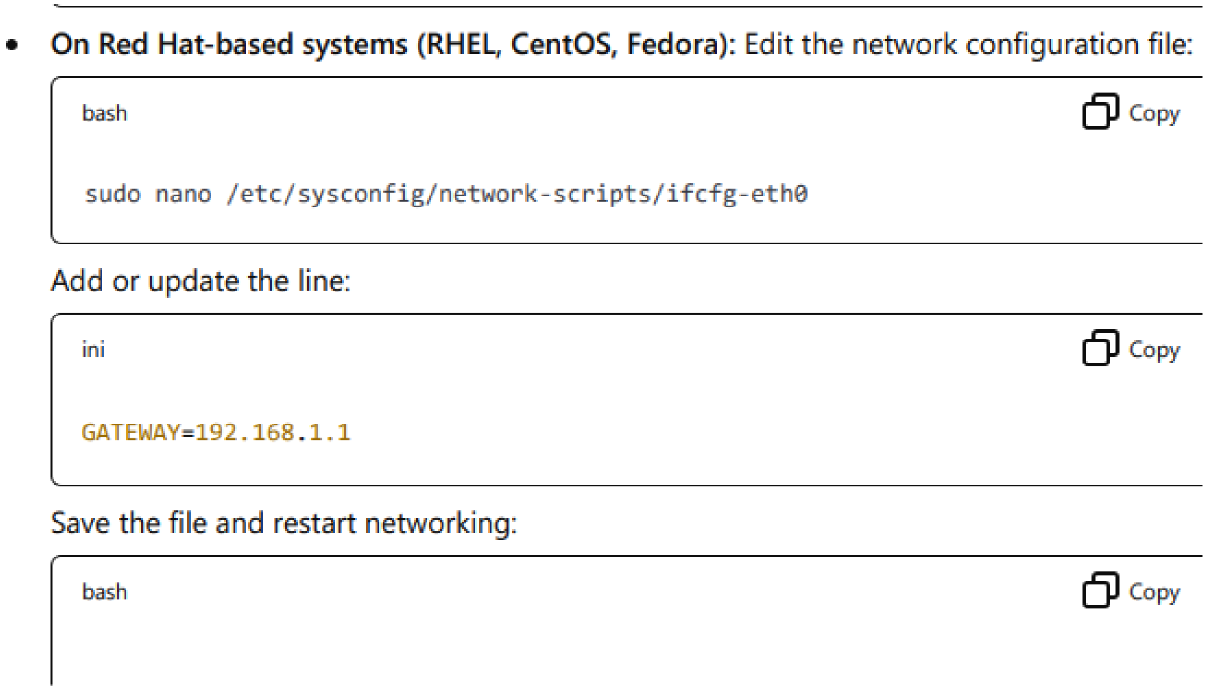
  ```bash
  sudo nano /etc/network/interfaces
  ```

  Add or update the gateway entry under your network interface:

  ```nginx
  iface eth0 inet static
      address 192.168.1.100
      netmask 255.255.255.0
      gateway 192.168.1.1
  ```

  Save and apply the changes:

  ```bash
  sudo systemctl restart networking
  ```



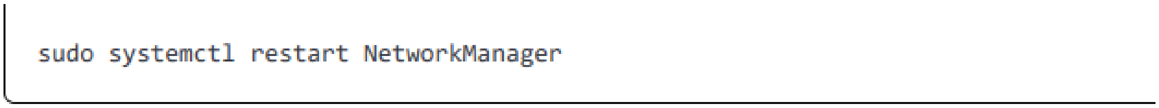
- **On Red Hat-based systems (RHEL, CentOS, Fedora):** Edit the network configuration file:

```bash
sudo nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

Add or update the line:

```ini
GATEWAY=192.168.1.1
```

Save the file and restart networking:



```bash
sudo systemctl restart NetworkManager
```

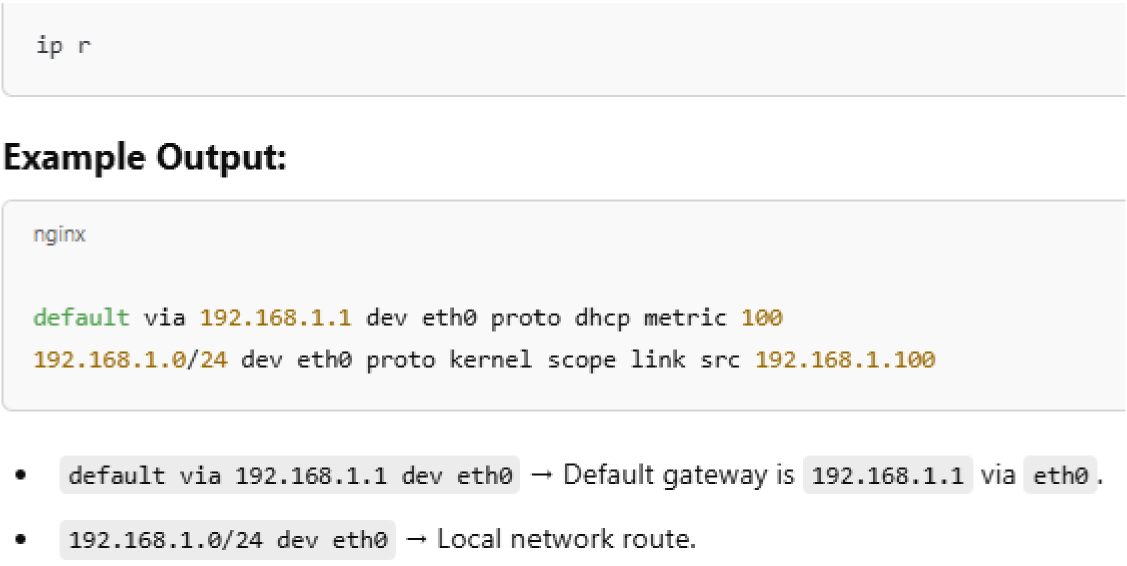
6.Verify Routing Table:

o Check the routing table to ensure that the default gateway is correctly configured.

bash

Copy code

ip route



```
ip r
```

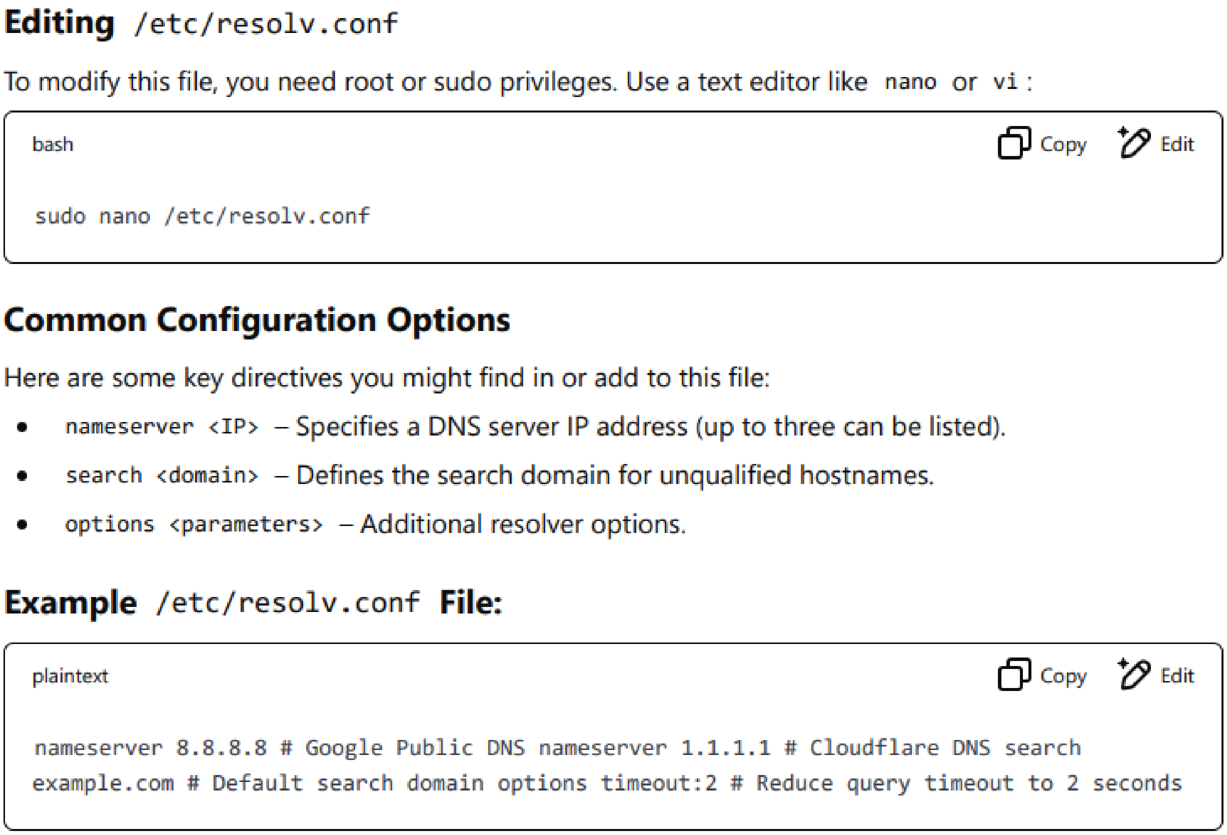
Example Output:

```
default via 192.168.1.1 dev eth0 proto dhcp metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.100
```

- `default via 192.168.1.1 dev eth0` → Default gateway is `192.168.1.1` via `eth0`.
- `192.168.1.0/24 dev eth0` → Local network route.

7.DNS Configuration:

o Edit /etc/resolv.conf to configure DNS servers:



Editing `/etc/resolv.conf`

To modify this file, you need root or sudo privileges. Use a text editor like `nano` or `vi`:

```
sudo nano /etc/resolv.conf
```

Common Configuration Options

Here are some key directives you might find in or add to this file:

- `nameserver <IP>` – Specifies a DNS server IP address (up to three can be listed).
- `search <domain>` – Defines the search domain for unqualified hostnames.
- `options <parameters>` – Additional resolver options.

Example `/etc/resolv.conf` File:

```
nameserver 8.8.8.8 # Google Public DNS nameserver 1.1.1.1 # Cloudflare DNS search
example.com # Default search domain options timeout:2 # Reduce query timeout to 2 seconds
```

Example `/etc/resolv.conf` File:

```plaintext
nameserver 8.8.8.8 # Google Public DNS nameserver 1.1.1.1 # Cloudflare DNS search
example.com # Default search domain options timeout:2 # Reduce query timeout to 2 seconds
```

Persisting Changes

Many modern Linux distributions use `systemd-resolved` or `NetworkManager`, which may overwrite manual changes. To ensure persistence:

1. **Disable automatic overwrites (if needed)**:
   - If using `systemd-resolved`:

     ```bash
     sudo systemctl disable --now systemd-resolved
     ```

     Then modify `/etc/resolv.conf` and set immutable mode:

     ```bash
     sudo chattr +i /etc/resolv.conf
     ```

```bash
/etc/NetworkManager/NetworkManager.conf
```

or use:

```bash
nmcli device modify eth0 ipv4.dns "8.8.8.8 1.1.1.1" nmcli connection reload
```

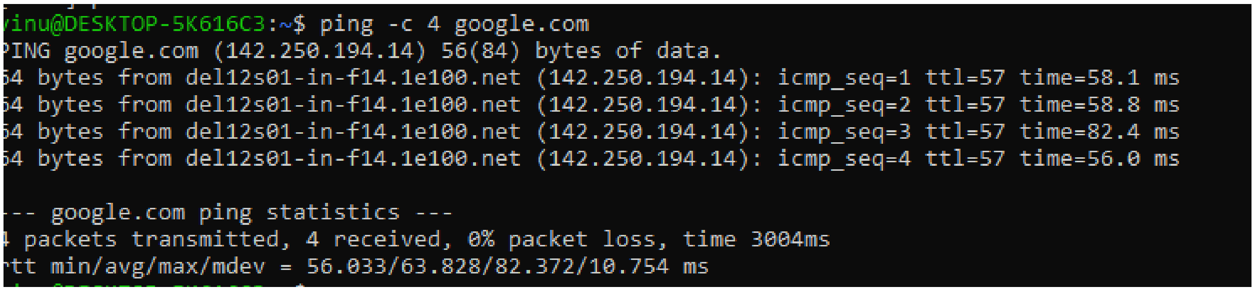
8.   Test Name Resolution:

o Test the DNS configuration by pinging a domain.

bash

Copy code

ping -c 4 google.com



```
vinu@DESKTOP-5K616C3:~$ ping -c 4 google.com
PING google.com (142.250.194.14) 56(84) bytes of data.
64 bytes from del12s01-in-f14.1e100.net (142.250.194.14): icmp_seq=1 ttl=57 time=58.1 ms
64 bytes from del12s01-in-f14.1e100.net (142.250.194.14): icmp_seq=2 ttl=57 time=58.8 ms
64 bytes from del12s01-in-f14.1e100.net (142.250.194.14): icmp_seq=3 ttl=57 time=82.4 ms
64 bytes from del12s01-in-f14.1e100.net (142.250.194.14): icmp_seq=4 ttl=57 time=56.0 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 56.033/63.828/82.372/10.754 ms
```

Lab 2: Dynamic IP Address Configuration using DHCP

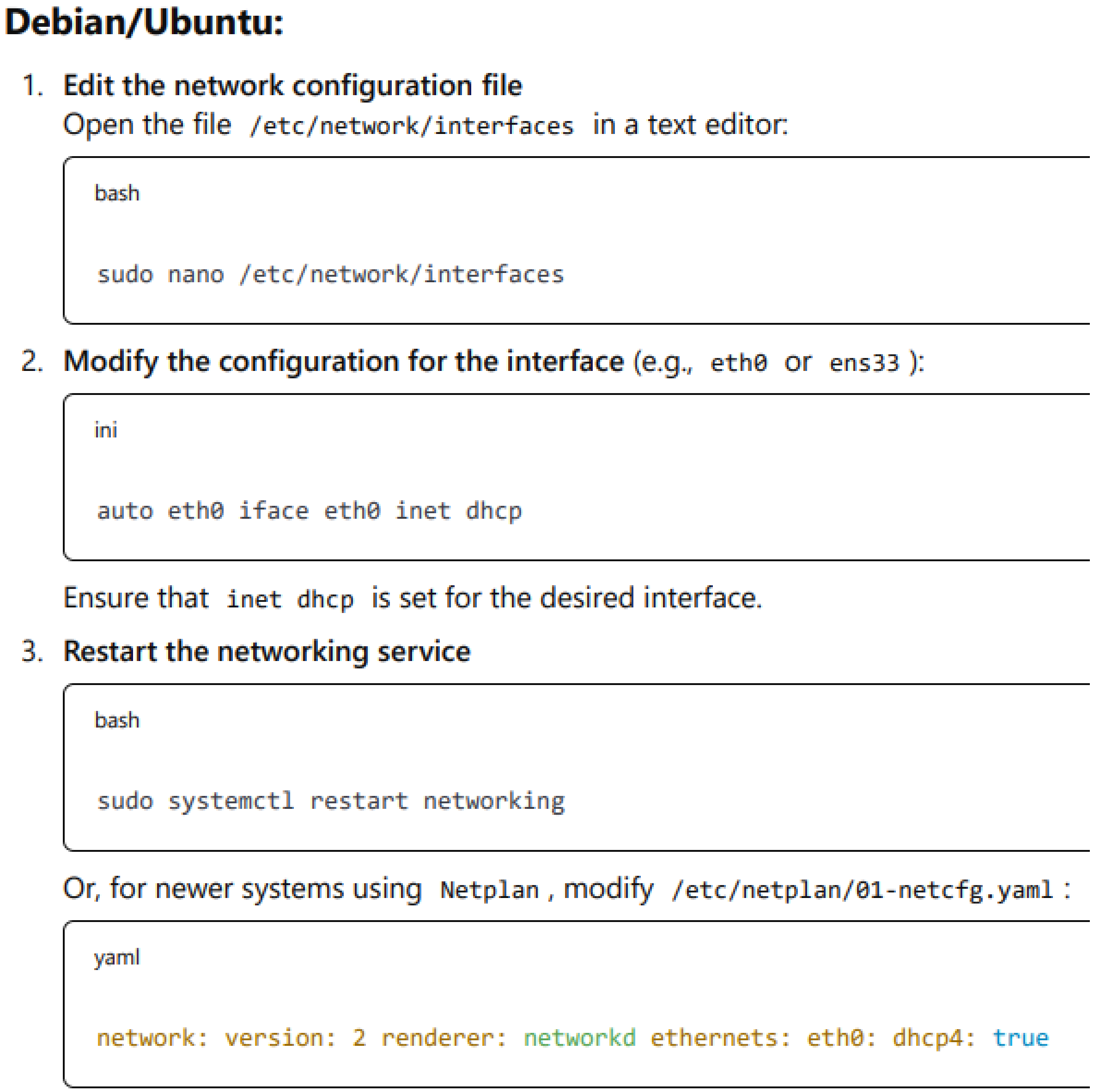
Objective:

• Learn how to configure and test dynamic IP address assignment using DHCP.

Tasks:

1.Configure DHCP Client:

O Ensure that the system is set to obtain an IP address automatically from a DHCP server. Modify the network interface configuration file, usually located at /etc/network/interfaces (Debian/Ubuntu) or /etc/sysconfig/network-scripts/ifcfg-eth0 (CentOS/RHEL):



Debian/Ubuntu:

1. **Edit the network configuration file**
   Open the file `/etc/network/interfaces` in a text editor:

```bash
sudo nano /etc/network/interfaces
```

2. **Modify the configuration for the interface** (e.g., `eth0` or `ens33`):

```ini
auto eth0 iface eth0 inet dhcp
```

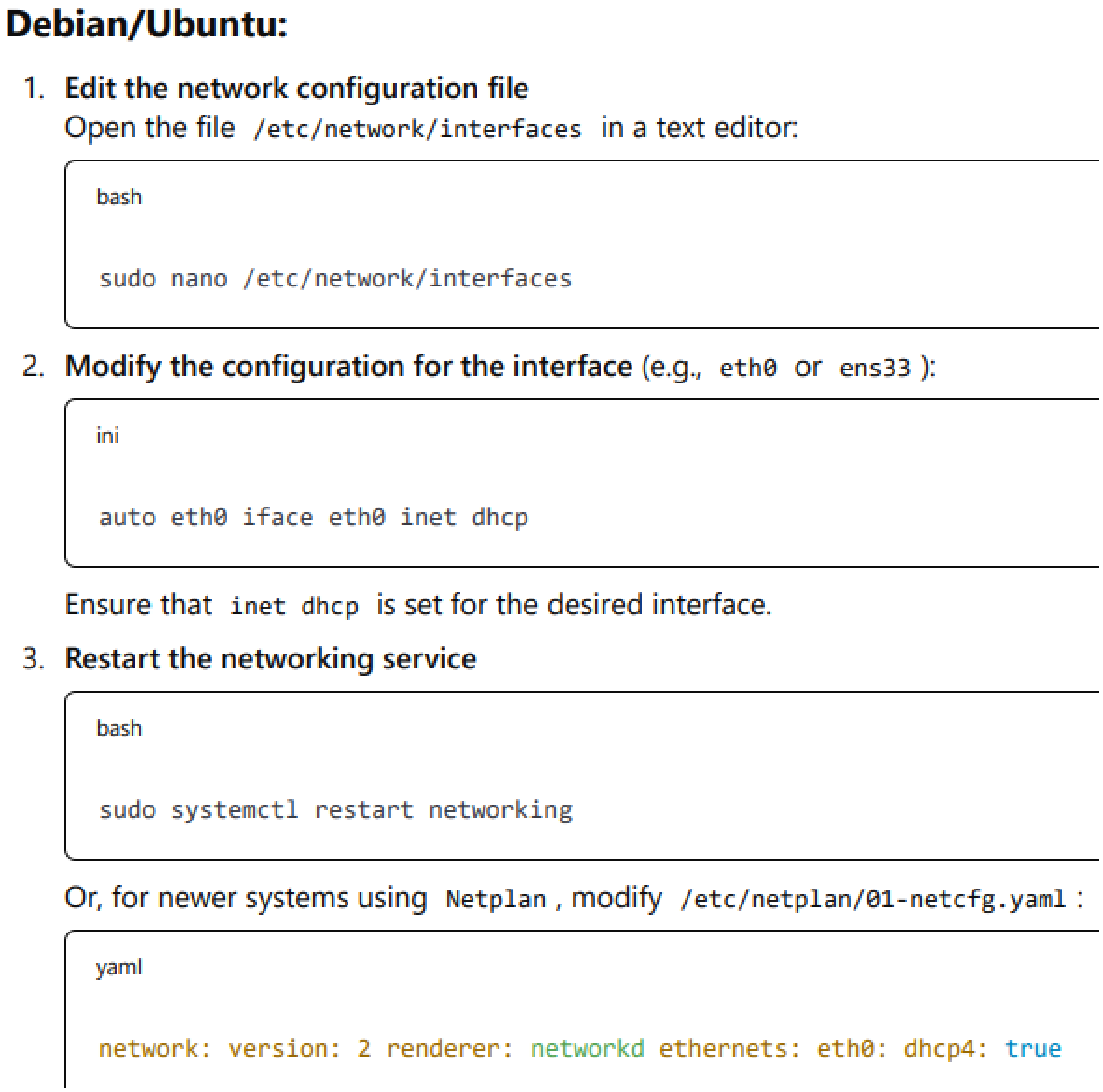
   Ensure that `inet dhcp` is set for the desired interface.

3. **Restart the networking service**

```bash
sudo systemctl restart networking
```

   Or, for newer systems using `Netplan`, modify `/etc/netplan/01-netcfg.yaml`:

```yaml
network: version: 2 renderer: networkd ethernets: eth0: dhcp4: true
```



Debian/Ubuntu:

1. **Edit the network configuration file**
   Open the file `/etc/network/interfaces` in a text editor:

   bash

   ```
   sudo nano /etc/network/interfaces
   ```

2. **Modify the configuration for the interface** (e.g., `eth0` or `ens33`):

   ini

   ```
   auto eth0 iface eth0 inet dhcp
   ```

   Ensure that `inet dhcp` is set for the desired interface.

3. **Restart the networking service**

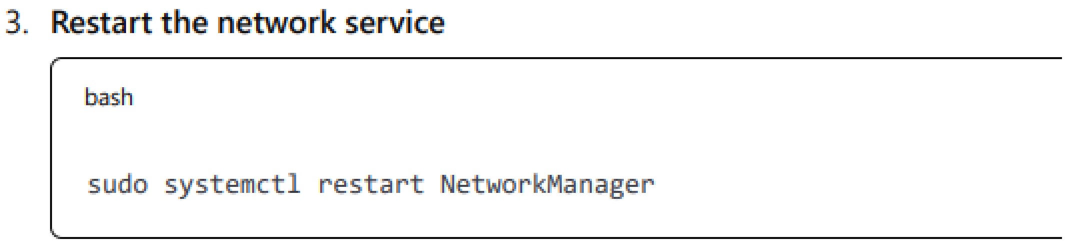
   bash

   ```
   sudo systemctl restart networking
   ```

   Or, for newer systems using `Netplan`, modify `/etc/netplan/01-netcfg.yaml`:

   yaml

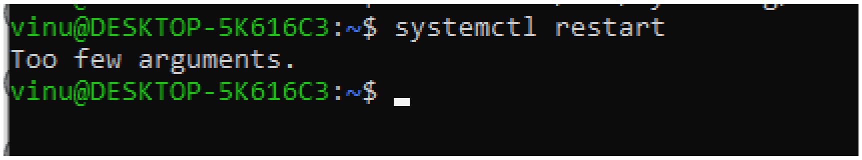
   ```
   network: version: 2 renderer: networkd ethernets: eth0: dhcp4: true
   ```



3. **Restart the network service**

   bash

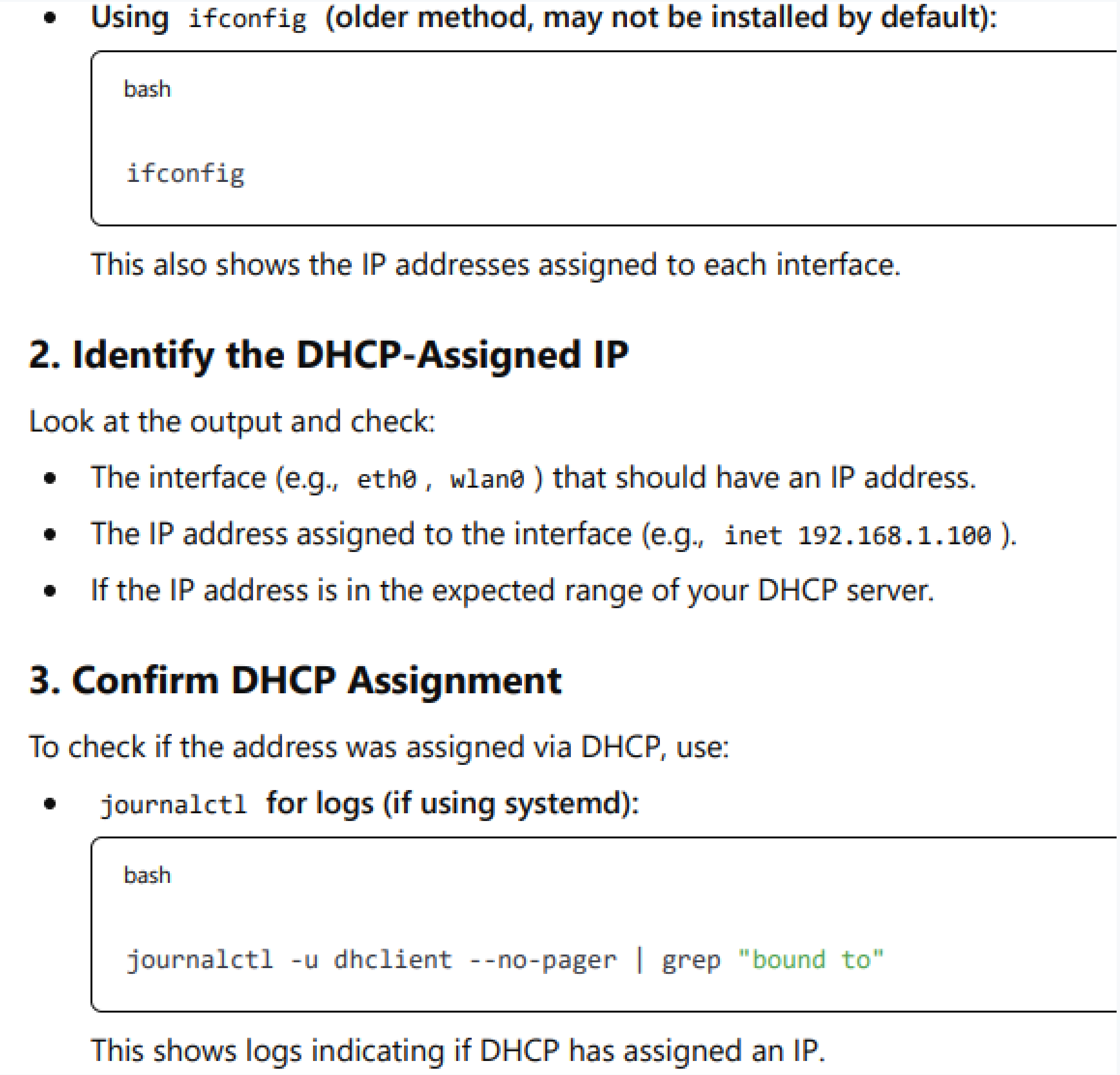
   ```
   sudo systemctl restart NetworkManager
   ```



```
vinu@DESKTOP-5K616C3:~$ systemctl restart
Too few arguments.
vinu@DESKTOP-5K616C3:~$
```

3. Verify DHCP Assignment:

o Use ip a or ifconfig to check if the IP address has been assigned by the DHCP server.



- **Using `ifconfig` (older method, may not be installed by default):**

  ```bash
  ifconfig
  ```

  This also shows the IP addresses assigned to each interface.

2. Identify the DHCP-Assigned IP

Look at the output and check:

- The interface (e.g., `eth0`, `wlan0`) that should have an IP address.
- The IP address assigned to the interface (e.g., `inet 192.168.1.100`).
- If the IP address is in the expected range of your DHCP server.

3. Confirm DHCP Assignment

To check if the address was assigned via DHCP, use:

- **`journalctl` for logs (if using systemd):**

  ```bash
  journalctl -u dhclient --no-pager | grep "bound to"
  ```

  This shows logs indicating if DHCP has assigned an IP.

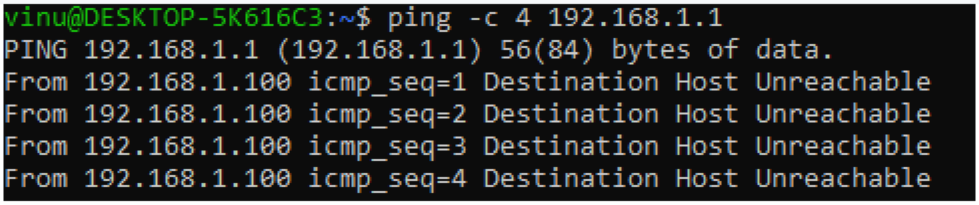
1. Ping Test:

o Use the ping command to check the network connectivity to another system.

bash

Copy code

ping -c 4 192.168.1.1



```
vinu@DESKTOP-5K616C3:~$ ping -c 4 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
From 192.168.1.100 icmp_seq=1 Destination Host Unreachable
From 192.168.1.100 icmp_seq=2 Destination Host Unreachable
From 192.168.1.100 icmp_seq=3 Destination Host Unreachable
From 192.168.1.100 icmp_seq=4 Destination Host Unreachable
```

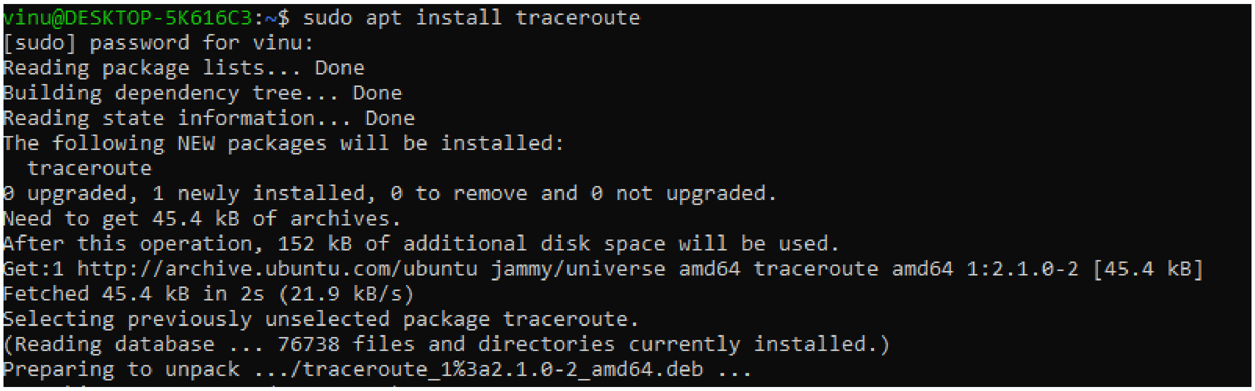
2.Traceroute:

o Use traceroute to track the route that packets take to reach a destination.

bash

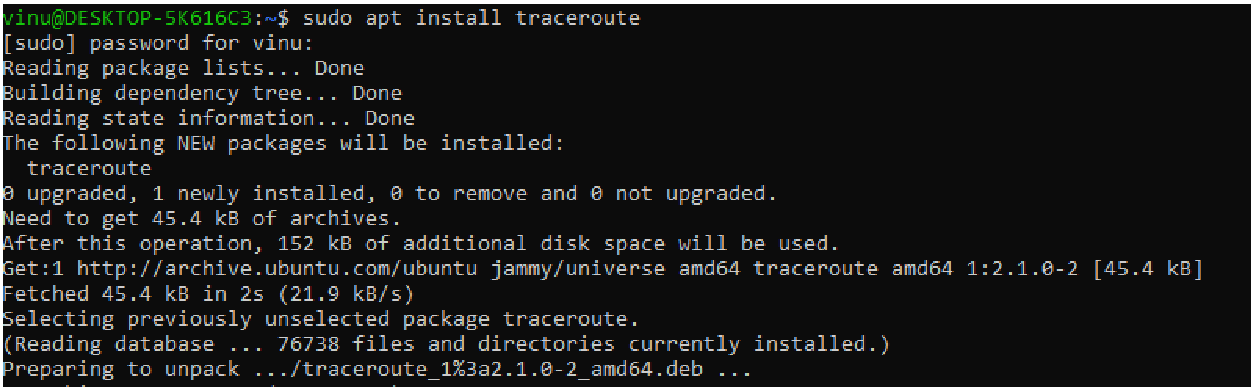
Copy code

sudo apt install traceroute # Ubuntu/Debian



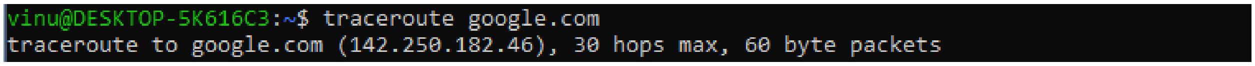
```
vinu@DESKTOP-5K616C3:~$ sudo apt install traceroute
[sudo] password for vinu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  traceroute
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 45.4 kB of archives.
After this operation, 152 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 traceroute amd64 1:2.1.0-2 [45.4 kB]
Fetched 45.4 kB in 2s (21.9 kB/s)
Selecting previously unselected package traceroute.
(Reading database ... 76738 files and directories currently installed.)
Preparing to unpack .../traceroute_1%3a2.1.0-2_amd64.deb ...
```

sudo yum install traceroute # CentOS/RHEL



```
vinu@DESKTOP-5K616C3:~$ sudo apt install traceroute
[sudo] password for vinu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  traceroute
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 45.4 kB of archives.
After this operation, 152 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 traceroute amd64 1:2.1.0-2 [45.4 kB]
Fetched 45.4 kB in 2s (21.9 kB/s)
Selecting previously unselected package traceroute.
(Reading database ... 76738 files and directories currently installed.)
Preparing to unpack .../traceroute_1%3a2.1.0-2_amd64.deb ...
```

traceroute google.com



```
vinu@DESKTOP-5K616C3:~$ traceroute google.com
traceroute to google.com (142.250.182.46), 30 hops max, 60 byte packets
```

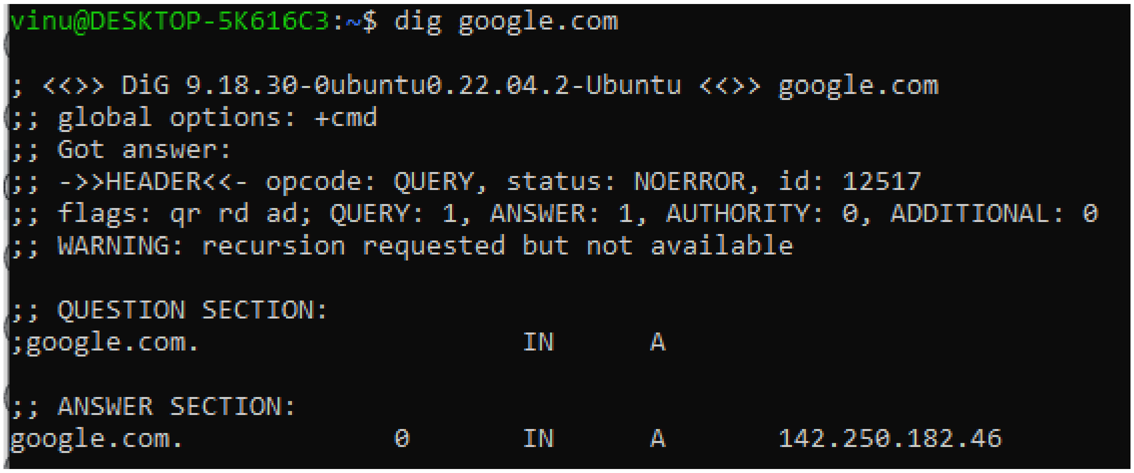
3.Check DNS Resolution:

O Use dig or nslookup to check DNS resolution for a domain.

bash

Copy code

dig google.com



```
vinu@DESKTOP-5K616C3:~$ dig google.com

; <<>> DiG 9.18.30-0ubuntu0.22.04.2-Ubuntu <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 12517
;; flags: qr rd ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;google.com.                    IN      A

;; ANSWER SECTION:
google.com.             0       IN      A       142.250.182.46
```

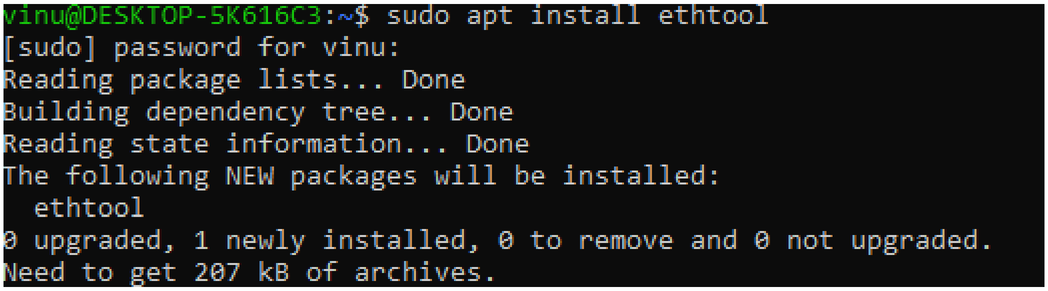
4.Network Interface Status:

o Use ethtool to check the status of the network interface (whether it's up, down, speed, etc.).

bash

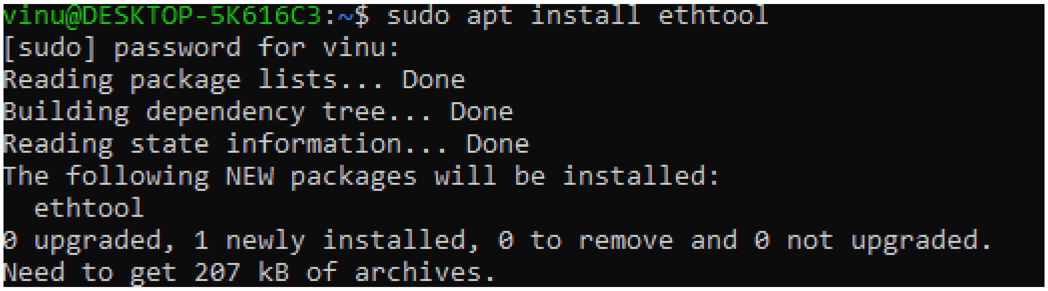
Copy code

sudo apt install ethtool # Ubuntu/Debian



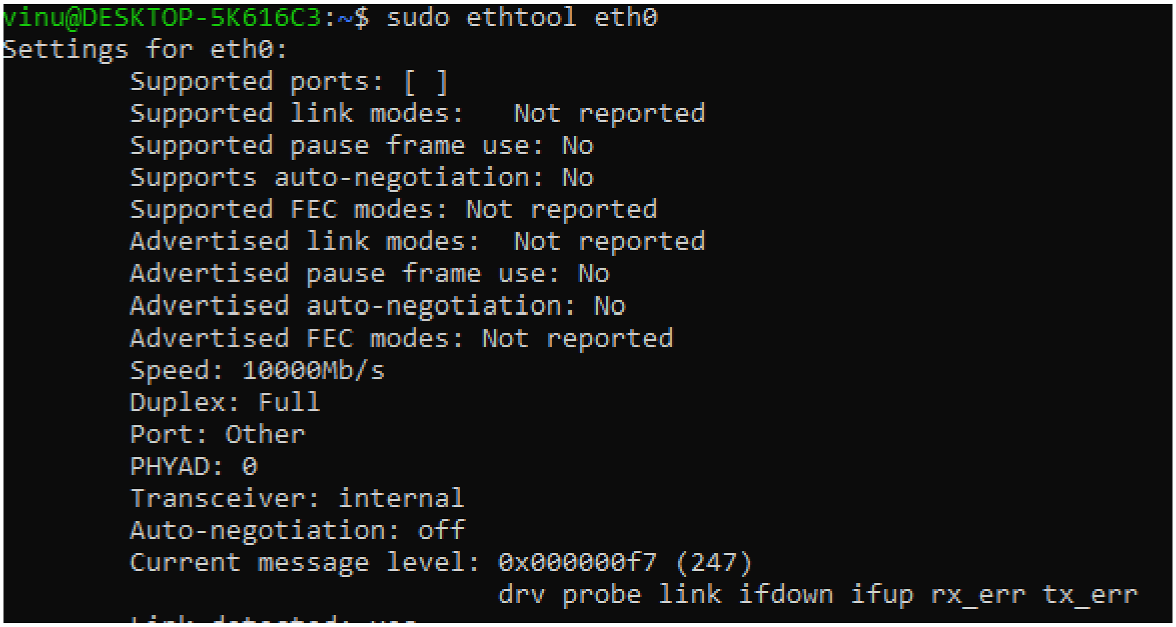
```
vinu@DESKTOP-5K616C3:~$ sudo apt install ethtool
[sudo] password for vinu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ethtool
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 207 kB of archives.
```

sudo yum install ethtool # CentOS/RHEL



```
vinu@DESKTOP-5K616C3:~$ sudo apt install ethtool
[sudo] password for vinu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ethtool
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 207 kB of archives.
```

sudo ethtool eth0



```
vinu@DESKTOP-5K616C3:~$ sudo ethtool eth0
Settings for eth0:
        Supported ports: [ ]
        Supported link modes:   Not reported
        Supported pause frame use: No
        Supports auto-negotiation: No
        Supported FEC modes: Not reported
        Advertised link modes:  Not reported
        Advertised pause frame use: No
        Advertised auto-negotiation: No
        Advertised FEC modes: Not reported
        Speed: 10000Mb/s
        Duplex: Full
        Port: Other
        PHYAD: 0
        Transceiver: internal
        Auto-negotiation: off
        Current message level: 0x000000f7 (247)
                               drv probe link ifdown ifup rx_err tx_err
```

5. View Routing Table:

o Use ip route or netstat -r to view the current routing table.

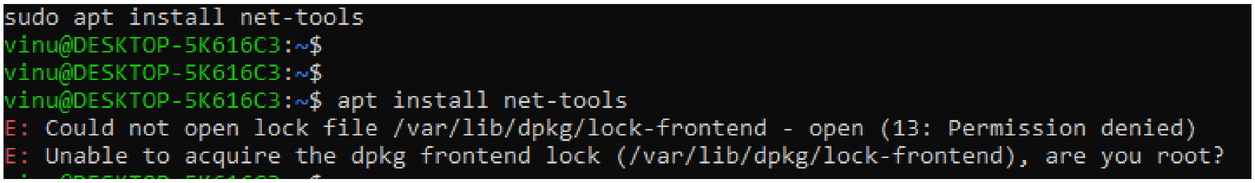
bash

Copy code

ip route

#or

netstat –r



```
sudo apt install net-tools
vinu@DESKTOP-5K616C3:~$
vinu@DESKTOP-5K616C3:~$
vinu@DESKTOP-5K616C3:~$ apt install net-tools
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?
```

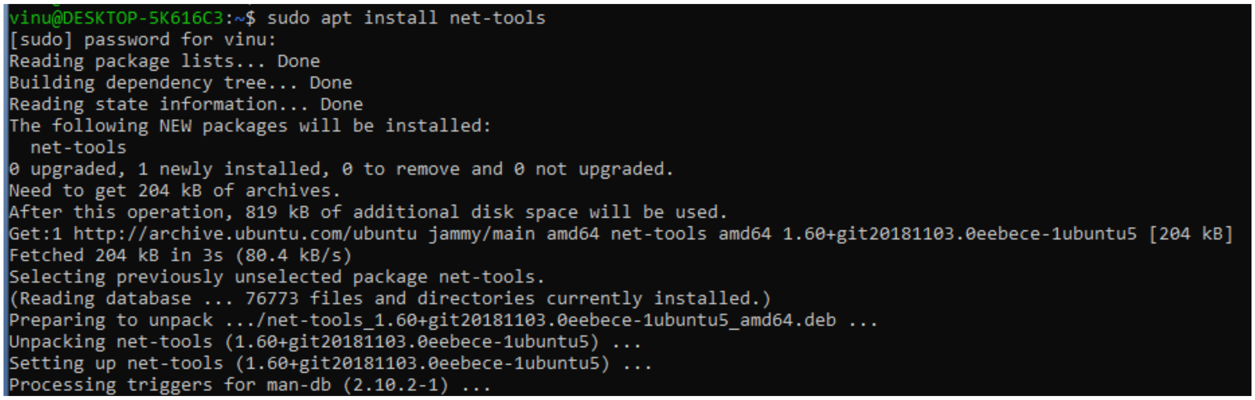
6.Check Active Connections:

o Use netstat or ss to view active network connections on the system.

bash

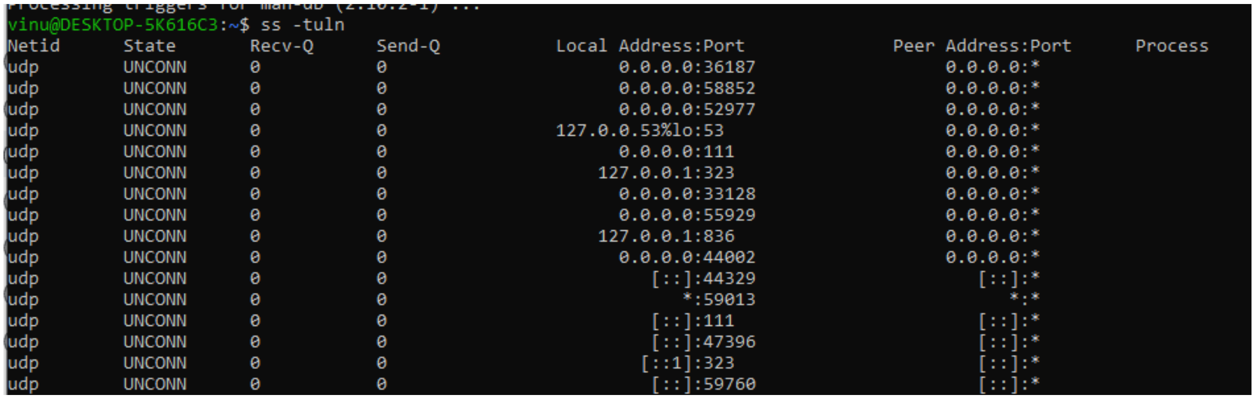
Copy code

netstat –tuln



```
vinu@DESKTOP-5K616C3:~$ sudo apt install net-tools
[sudo] password for vinu:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  net-tools
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 204 kB of archives.
After this operation, 819 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Fetched 204 kB in 3s (80.4 kB/s)
Selecting previously unselected package net-tools.
(Reading database ... 76773 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Processing triggers for man-db (2.10.2-1) ...
```

ss -tuln



```
Processing triggers for man-db (2.10.2-1) ...
vinu@DESKTOP-5K616C3:~$ ss -tuln
Netid      State       Recv-Q      Send-Q            Local Address:Port               Peer Address:Port      Process
udp        UNCONN      0           0                       0.0.0.0:36187                   0.0.0.0:*
udp        UNCONN      0           0                       0.0.0.0:58852                   0.0.0.0:*
udp        UNCONN      0           0                       0.0.0.0:52977                   0.0.0.0:*
udp        UNCONN      0           0                 127.0.0.53%lo:53                      0.0.0.0:*
udp        UNCONN      0           0                       0.0.0.0:111                     0.0.0.0:*
udp        UNCONN      0           0                     127.0.0.1:323                     0.0.0.0:*
udp        UNCONN      0           0                       0.0.0.0:33128                   0.0.0.0:*
udp        UNCONN      0           0                       0.0.0.0:55929                   0.0.0.0:*
udp        UNCONN      0           0                     127.0.0.1:836                     0.0.0.0:*
udp        UNCONN      0           0                       0.0.0.0:44002                   0.0.0.0:*
udp        UNCONN      0           0                          [::]:44329                      [::]:*
udp        UNCONN      0           0                             *:59013                         *:*
udp        UNCONN      0           0                          [::]:111                        [::]:*
udp        UNCONN      0           0                          [::]:47396                      [::]:*
udp        UNCONN      0           0                         [::1]:323                        [::]:*
udp        UNCONN      0           0                          [::]:59760                      [::]:*
```

7.Check Network Configuration with ifconfig or ip:

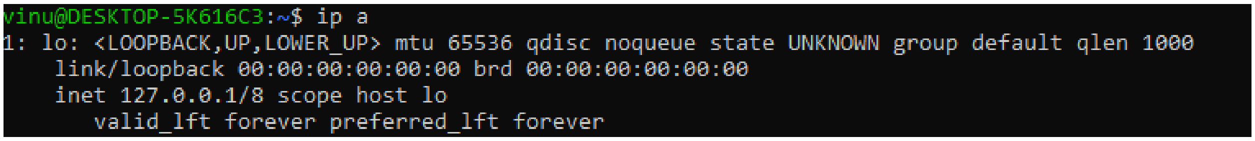
o Verify network interface configuration using ifconfig or ip.
bash

Copy code

ifconfig

#or

ip a



```
vinu@DESKTOP-5K616C3:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
```

Lab 4: Configuring Advanced Network Settings (Static Routes, VLANs, etc.)

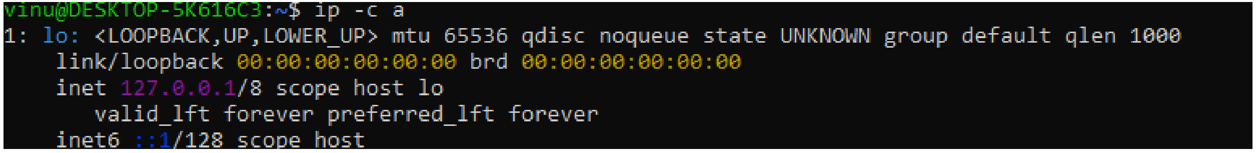
Objective:

• Learn how to configure advanced network settings such as static routes and VLANs on a Linux system.

Tasks:

1.Add a Static Route:

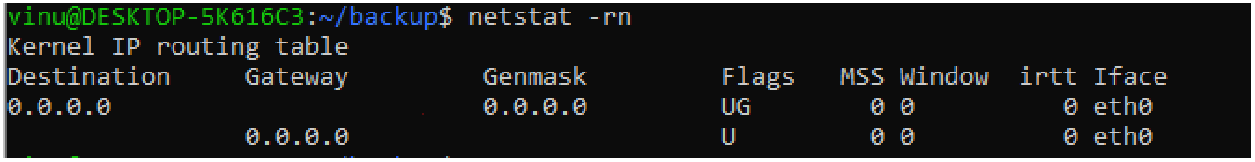
o Use ip to add a static route.
For example, to route traffic destined for 192.168.2.0/24 via a gateway 192.168.1.1:



```
vinu@DESKTOP-5K616C3:~$ ip -c a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
```

2.View Routing Table:

o View the routing table to ensure the static route has been added:

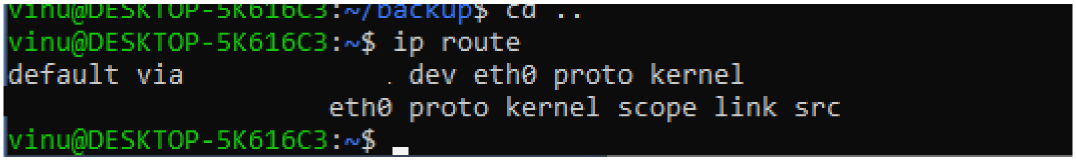


```
vinu@DESKTOP-5K616C3:~/backup$ netstat -rn
Kernel IP routing table
Destination     Gateway         Genmask         Flags   MSS Window  irtt Iface
0.0.0.0                         0.0.0.0         UG        0 0          0 eth0
                0.0.0.0                         U         0 0          0 eth0
```

bash

Copy code

ip route



```
vinu@DESKTOP-5K616C3:~/backup$ cd ..
vinu@DESKTOP-5K616C3:~$ ip route
default via            . dev eth0 proto kernel
                    eth0 proto kernel scope link src
vinu@DESKTOP-5K616C3:~$
```

```
3.Configure a VLAN Interface:

 o Create a VLAN interface using vconfig or ip commands. For
example, to create VLAN 10 on interface eth0:
```

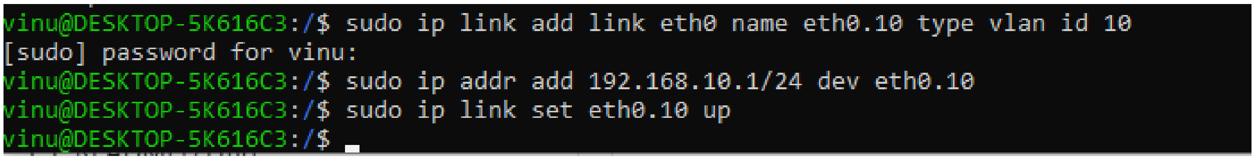
bash

Copy code

sudo ip link add link eth0 name eth0.10 type vlan id 10

sudo ip addr add 192.168.10.1/24 dev eth0.10

sudo ip link set eth0.10 up



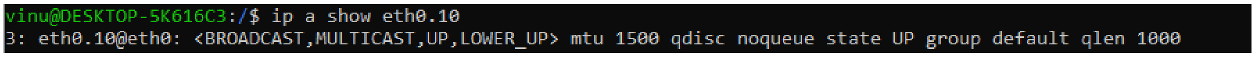
```
vinu@DESKTOP-5K616C3:/$ sudo ip link add link eth0 name eth0.10 type vlan id 10
[sudo] password for vinu:
vinu@DESKTOP-5K616C3:/$ sudo ip addr add 192.168.10.1/24 dev eth0.10
vinu@DESKTOP-5K616C3:/$ sudo ip link set eth0.10 up
vinu@DESKTOP-5K616C3:/$
```

4.Verify VLAN Configuration:

o Verify the VLAN interface is up and has the correct IP address: bash

Copy code

ip a show eth0.10



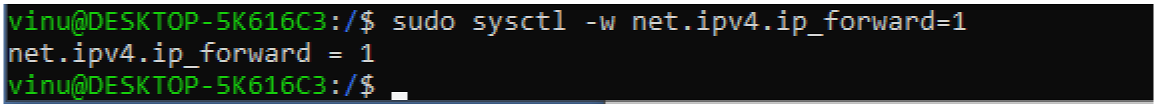
```
vinu@DESKTOP-5K616C3:/$ ip a show eth0.10
3: eth0.10@eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
```

5.Enable IP Forwarding (for Routing Between Networks):

oEnable IP forwarding to allow routing between different subnets: bash

Copy code

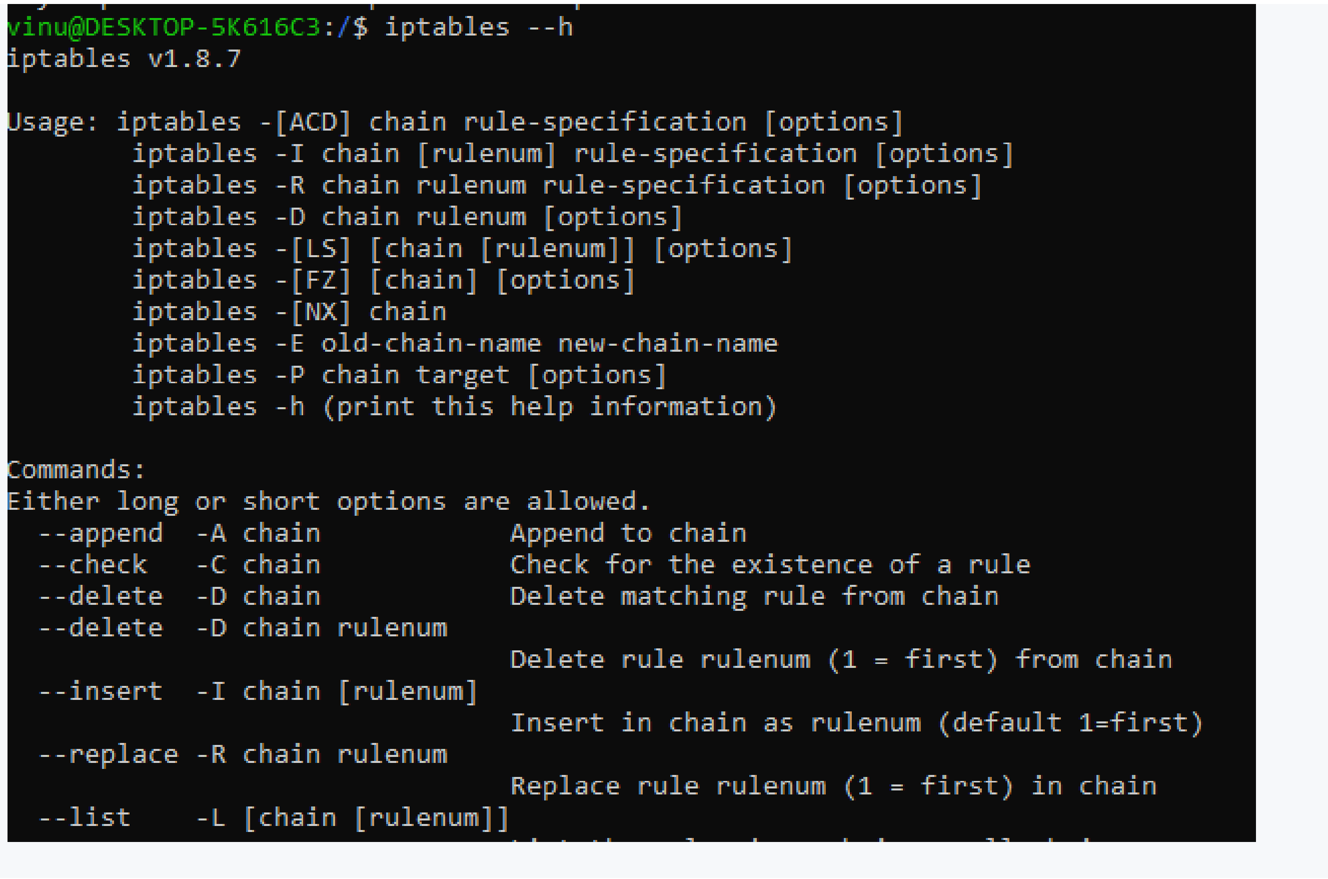
sudo sysctl -w net.ipv4.ip_forward=1



```
vinu@DESKTOP-5K616C3:/$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
vinu@DESKTOP-5K616C3:/$ _
```

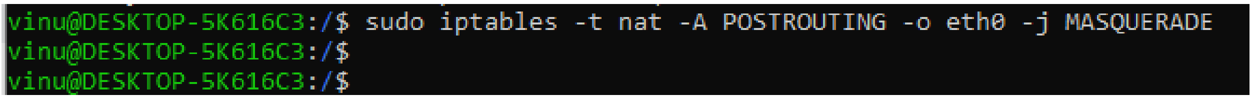
6.Configure NAT for Internet Sharing:

o Configure Network Address Translation (NAT) using iptables to share the internet connection with a local network:



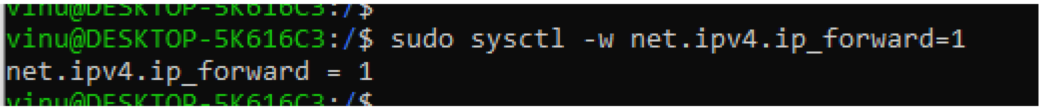
```
vinu@DESKTOP-5K616C3:/$ iptables --h
iptables v1.8.7

Usage: iptables -[ACD] chain rule-specification [options]
        iptables -I chain [rulenum] rule-specification [options]
        iptables -R chain rulenum rule-specification [options]
        iptables -D chain rulenum [options]
        iptables -[LS] [chain [rulenum]] [options]
        iptables -[FZ] [chain] [options]
        iptables -[NX] chain
        iptables -E old-chain-name new-chain-name
        iptables -P chain target [options]
        iptables -h (print this help information)

Commands:
Either long or short options are allowed.
  --append  -A chain            Append to chain
  --check   -C chain            Check for the existence of a rule
  --delete  -D chain            Delete matching rule from chain
  --delete  -D chain rulenum
                                Delete rule rulenum (1 = first) from chain
  --insert  -I chain [rulenum]
                                Insert in chain as rulenum (default 1=first)
  --replace -R chain rulenum
                                Replace rule rulenum (1 = first) in chain
  --list    -L [chain [rulenum]]
```

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE



```
vinu@DESKTOP-5K616C3:/$ sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
vinu@DESKTOP-5K616C3:/$
vinu@DESKTOP-5K616C3:/$
```

sudo sysctl -w net.ipv4.ip_forward=1



```
vinu@DESKTOP-5K616C3:/$
vinu@DESKTOP-5K616C3:/$ sudo sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
vinu@DESKTOP-5K616C3:/$
```

Lab 5: Securing Linux Network Services

Objective:

• Learn how to secure network services on a Linux system by configuring firewalls and using SSH for secure communication.

Tasks:

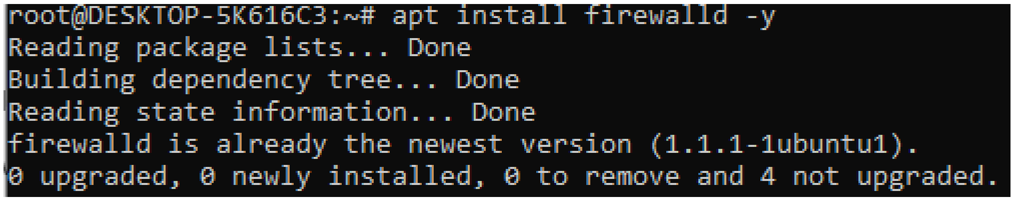
1.Configure UFW (Uncomplicated Firewall) on Ubuntu/Debian:

o Install and configure UFW to allow only certain services (e.g., SSH, HTTP):

bash

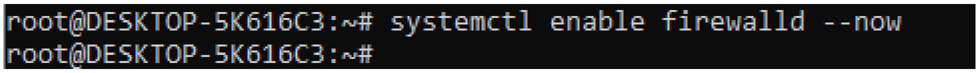
Copy code

Apt install firewalld –y



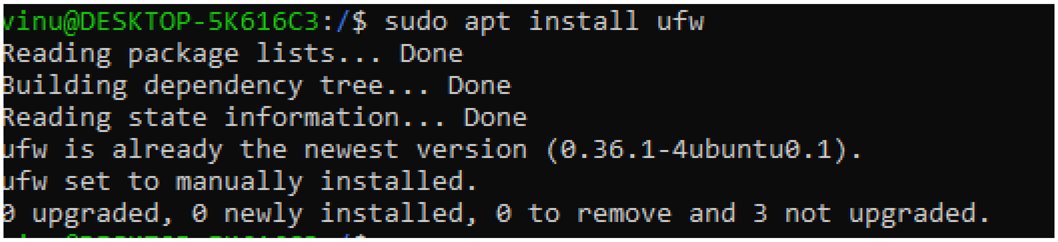
```
root@DESKTOP-5K616C3:~# apt install firewalld -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
firewalld is already the newest version (1.1.1-1ubuntu1).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

systemctl enable firewalld –now



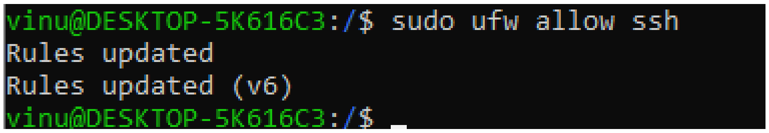
```
root@DESKTOP-5K616C3:~# systemctl enable firewalld --now
root@DESKTOP-5K616C3:~#
```

sudo apt install ufw



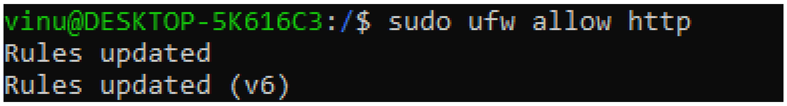
```
vinu@DESKTOP-5K616C3:/$ sudo apt install ufw
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ufw is already the newest version (0.36.1-4ubuntu0.1).
ufw set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

sudo ufw allow ssh



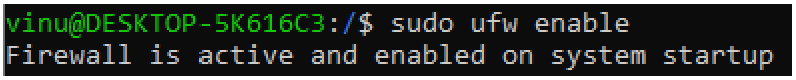
```
vinu@DESKTOP-5K616C3:/$ sudo ufw allow ssh
Rules updated
Rules updated (v6)
vinu@DESKTOP-5K616C3:/$ _
```

sudo ufw allow http



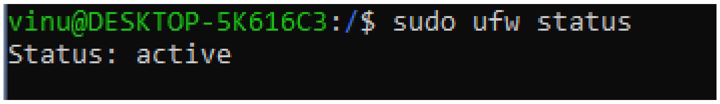
```
vinu@DESKTOP-5K616C3:/$ sudo ufw allow http
Rules updated
Rules updated (v6)
```

sudo ufw enable



```
vinu@DESKTOP-5K616C3:/$ sudo ufw enable
Firewall is active and enabled on system startup
```

sudo ufw status 2. Configure FirewallD on CentOS/RHEL:



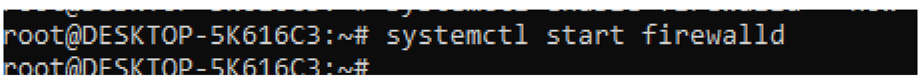
```
vinu@DESKTOP-5K616C3:/$ sudo ufw status
Status: active
```

```
o Install and configure firewalld to allow only certain services:
```

bash

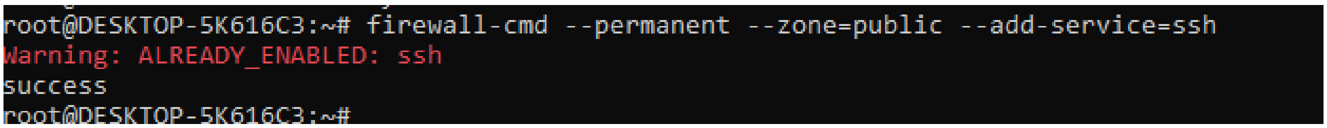
Copy code

sudo systemctl start firewalld



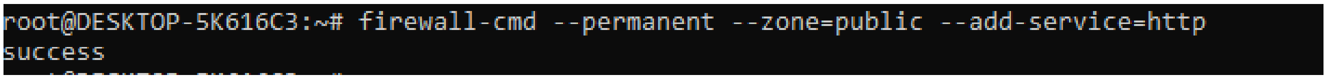
```
root@DESKTOP-5K616C3:~# systemctl start firewalld
root@DESKTOP-5K616C3:~#
```

sudo firewall-cmd --permanent --zone=public --add-service=ssh



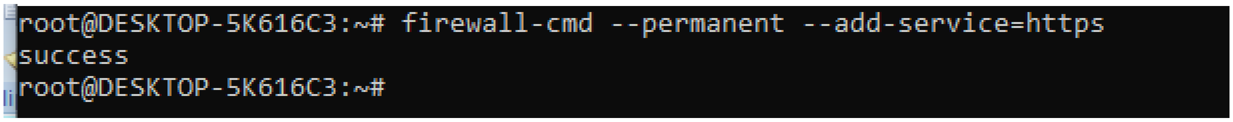
```
root@DESKTOP-5K616C3:~# firewall-cmd --permanent --zone=public --add-service=ssh
Warning: ALREADY_ENABLED: ssh
success
root@DESKTOP-5K616C3:~#
```

sudo firewall-cmd --permanent --zone=public --add-service=http



```
root@DESKTOP-5K616C3:~# firewall-cmd --permanent --zone=public --add-service=http
success
```

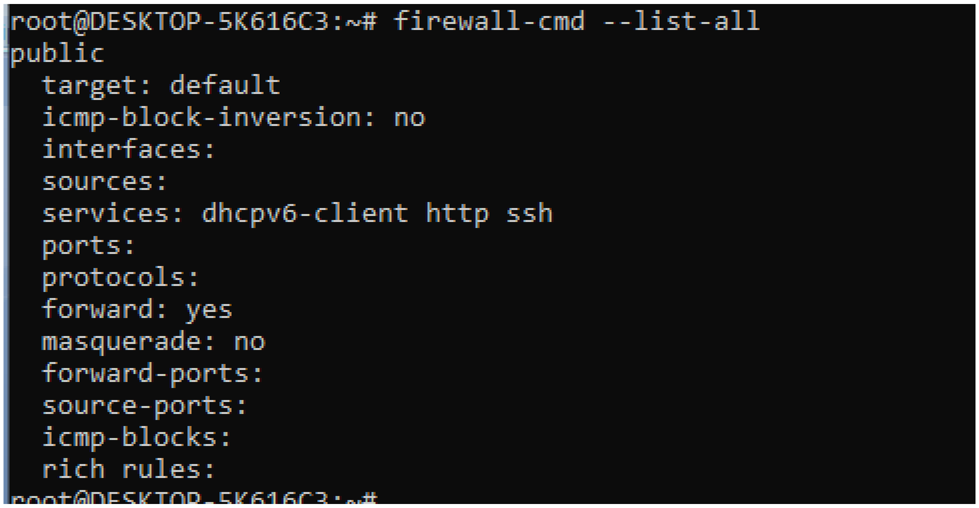
firewall-cmd --permanent --add-service=https



```
root@DESKTOP-5K616C3:~# firewall-cmd --permanent --add-service=https
success
root@DESKTOP-5K616C3:~#
```

sudo firewall-cmd --reload

sudo firewall-cmd --list-all



```
root@DESKTOP-5K616C3:~# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client http ssh
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
root@DESKTOP-5K616C3:~#
```

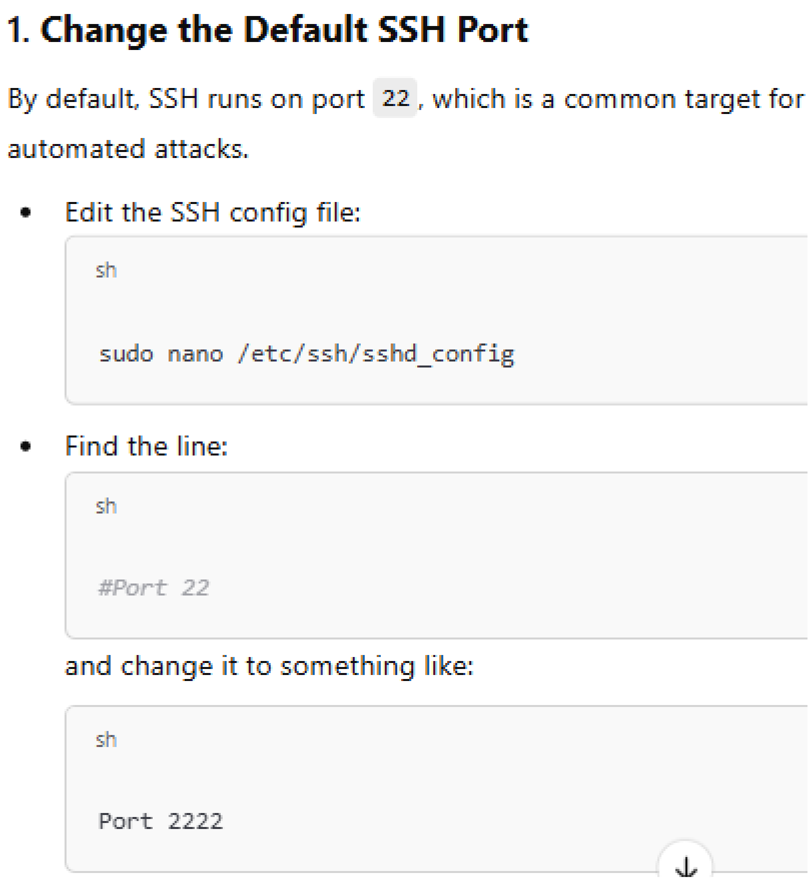
3.Secure SSH Access:

o Disable root login and change the SSH port by editing /etc/ssh/sshd_config:

bash

Copy code

PermitRootLogin no



1. Change the Default SSH Port

By default, SSH runs on port `22`, which is a common target for automated attacks.

- Edit the SSH config file:

  ```sh
  sudo nano /etc/ssh/sshd_config
  ```

- Find the line:

  ```sh
  #Port 22
  ```

  and change it to something like:

  ```sh
  Port 2222
  ```

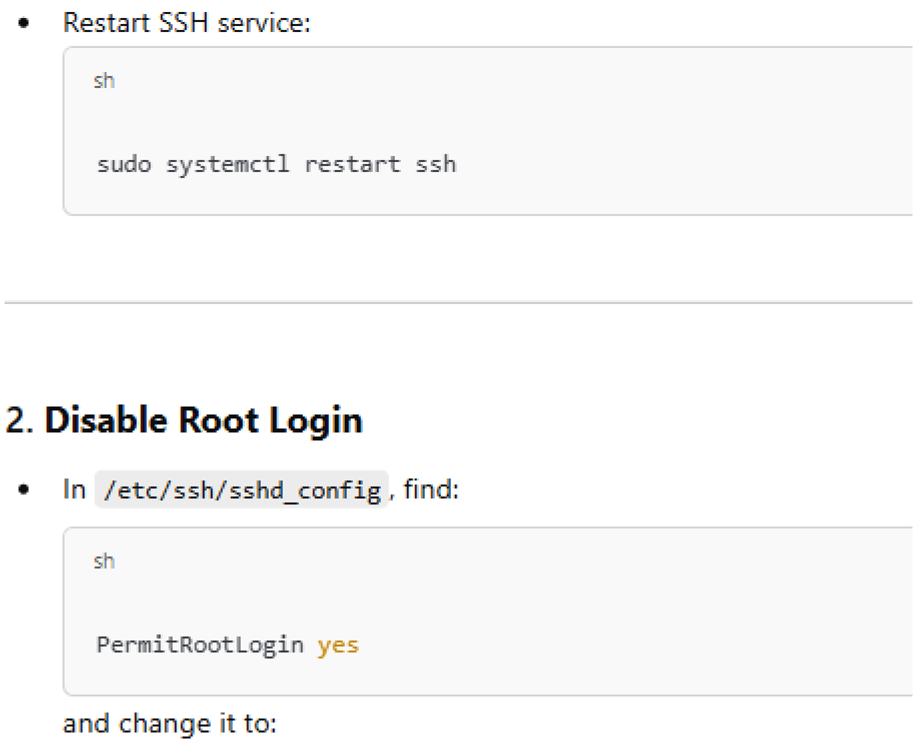
Port 2222

```
o     Restart the SSH service:
```

bash

Copy code

sudo systemctl restart sshd



- Restart SSH service:

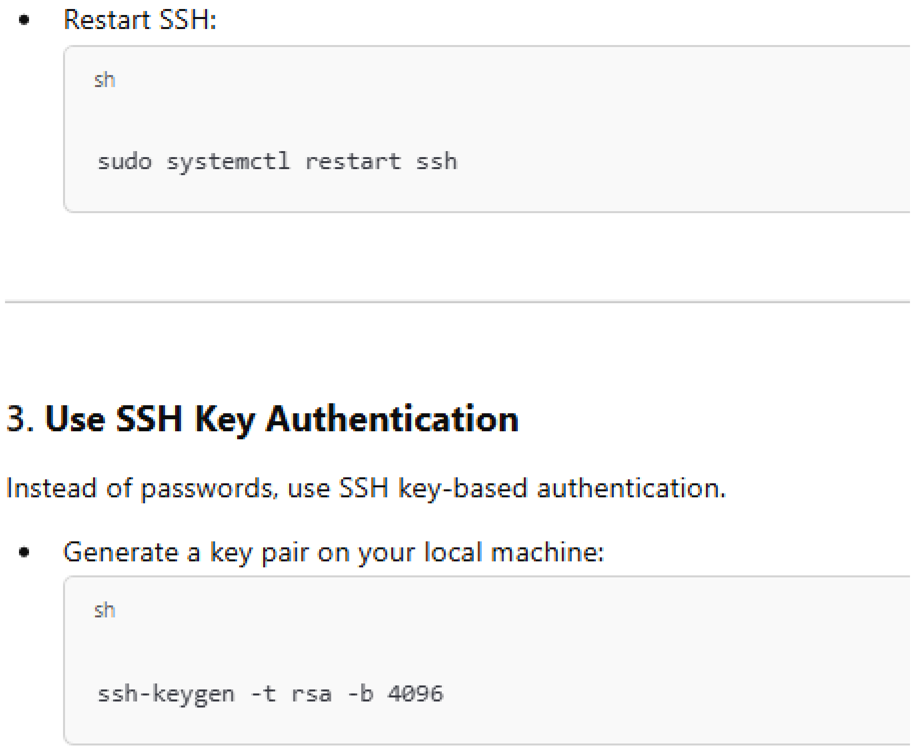
```sh
sudo systemctl restart ssh
```

2. Disable Root Login

- In `/etc/ssh/sshd_config`, find:

```sh
PermitRootLogin yes
```

and change it to:



- Restart SSH:

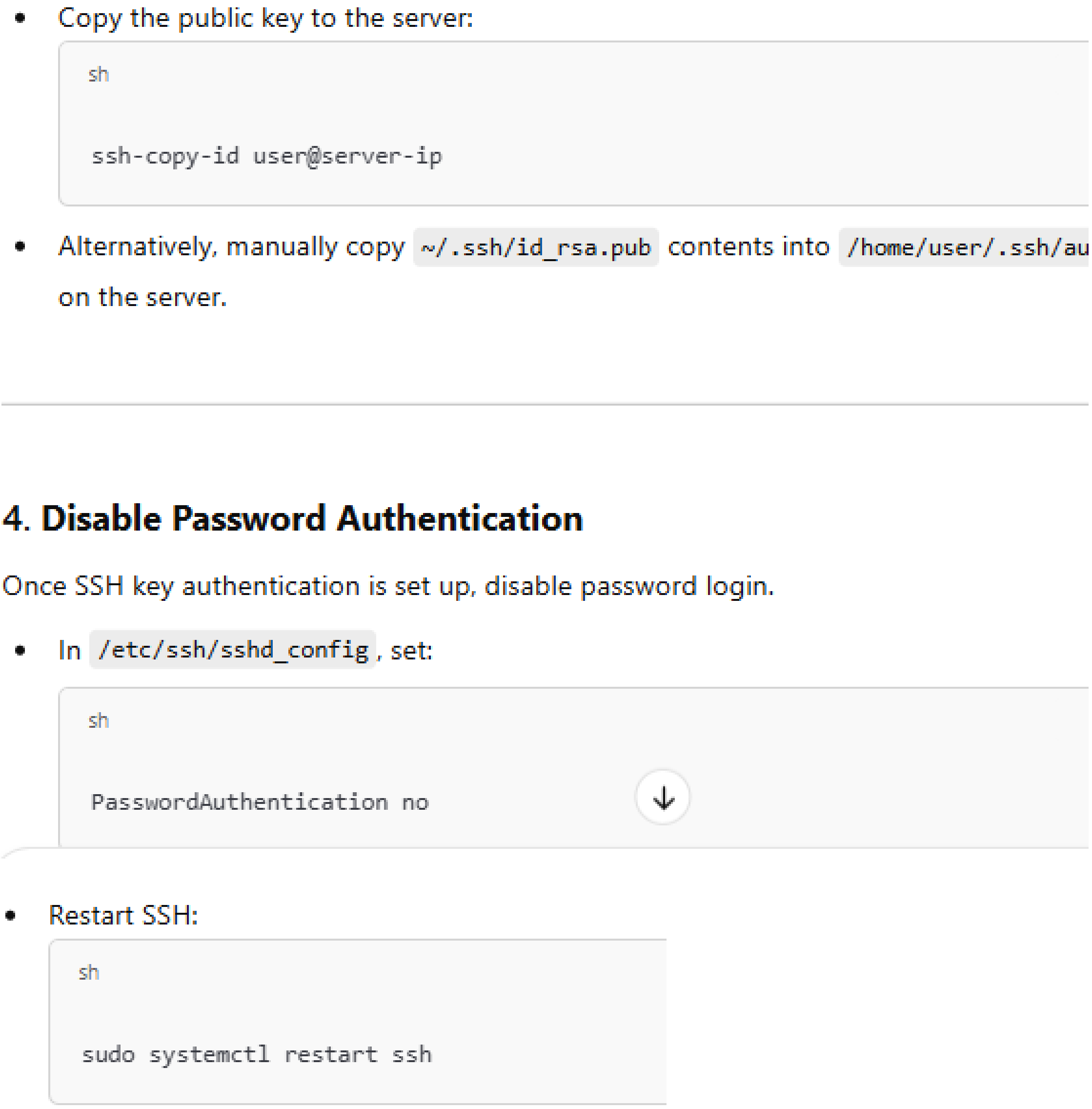
```sh
sudo systemctl restart ssh
```

3. Use SSH Key Authentication

Instead of passwords, use SSH key-based authentication.

- Generate a key pair on your local machine:

```sh
ssh-keygen -t rsa -b 4096
```



- Copy the public key to the server:

```sh
ssh-copy-id user@server-ip
```

- Alternatively, manually copy `~/.ssh/id_rsa.pub` contents into `/home/user/.ssh/au` on the server.

---

4. Disable Password Authentication

Once SSH key authentication is set up, disable password login.

- In `/etc/ssh/sshd_config`, set:

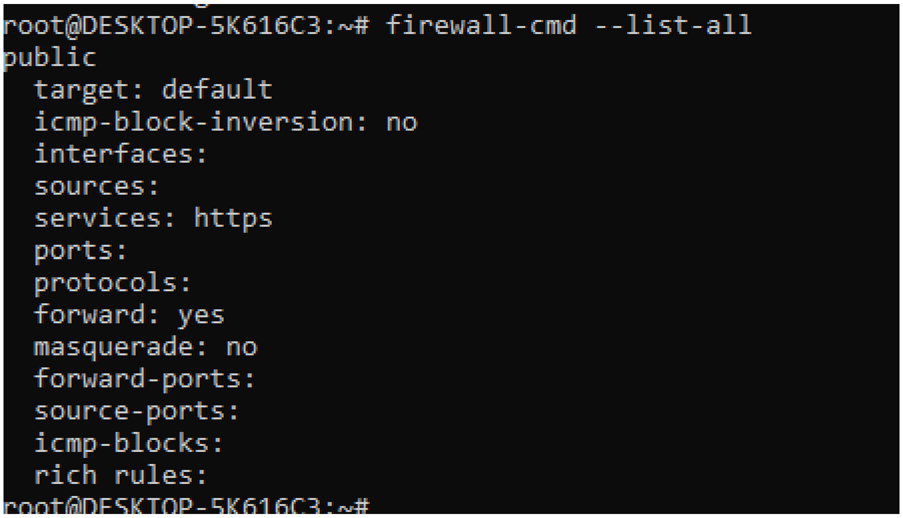
```sh
PasswordAuthentication no
```

- Restart SSH:

```sh
sudo systemctl restart ssh
```

4. Verify Firewall Configuration:

o Use ufw status or firewall-cmd --list-all to verify that only the required services are accessible.



```
root@DESKTOP-5K616C3:~# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: https
  ports:
  protocols:
  forward: yes
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
root@DESKTOP-5K616C3:~#
```