LINUX NETWORKING COMMANDS

Linux provides a wide range of networking commands for **troubleshooting, monitoring, and configuring network connections**. These commands help system administrators manage IP addresses, check network connectivity, inspect routing tables, and secure the network.

---

1 Checking Network Configuration

`ip` – View and Configure Network Interfaces

The `ip` command is used to manage IP addresses, routes, and interfaces.

- **Display active network interfaces and IP addresses:**

```
bash
CopyEdit
ip addr show
```

- **Assign a new IP address to an interface:**

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

- **Remove an assigned IP address:**

```
bash
CopyEdit
sudo ip addr del 192.168.1.100/24 dev eth0
```

- **Check routing table:**

```
bash
CopyEdit
ip route show
```

`ifconfig` – Legacy Network Configuration Tool

(Deprecated in modern distributions, replaced by `ip`)

- **Show network interfaces and assigned IPs:**

  ```
  bash
  CopyEdit
  ifconfig
  ```

- **Enable or disable a network interface:**

  ```
  bash
  CopyEdit
  sudo ifconfig eth0 up   # Enable
  sudo ifconfig eth0 down # Disable
  ```

---

2 Testing Network Connectivity

`ping` – Check if a Host is Reachable

Used to test connectivity between two devices using ICMP packets.

```
bash
CopyEdit
ping google.com
```

- Add `-c` to specify the number of packets:

  ```
  bash
  CopyEdit
  ping -c 4 8.8.8.8
  ```

`traceroute` – Track the Path of Network Packets

Displays all the hops a packet takes to reach its destination.

```
bash
CopyEdit
traceroute google.com
```

- Alternative (`mtr`) with real-time updates:

  ```
  bash
  CopyEdit
  mtr google.com
  ```

`nslookup` & `dig` – Query DNS Records

- **Find the IP address of a domain using `nslookup`:**

  ```
  bash
  ```

```
CopyEdit
nslookup google.com
```

- **Get detailed DNS information using `dig`:**

```
bash
CopyEdit
dig google.com
```

---

3 Managing Network Connections

`netstat` – Display Network Statistics (Deprecated, Use `ss`)

- **Show active network connections:**

```
bash
CopyEdit
netstat -tulnp
```

`ss` – Display Detailed Network Socket Information

More efficient than `netstat` for checking open ports and connections.

- **Show all listening ports:**

```
bash
CopyEdit
ss -tuln
```

- **View connections by protocol:**

```
bash
CopyEdit
ss -tan  # TCP connections
ss -uan  # UDP connections
```

---

4 Network Monitoring & Packet Analysis

`tcpdump` – Capture Network Packets

- **Capture all packets on an interface:**

```
bash
CopyEdit
sudo tcpdump -i eth0
```

- **Filter by protocol (e.g., HTTP traffic):**

```
bash
CopyEdit
sudo tcpdump -i eth0 port 80
```

`nmap` – Scan for Open Ports & Hosts

- **Basic network scan:**

```
bash
CopyEdit
nmap 192.168.1.1
```

- **Scan for open ports:**

```
bash
CopyEdit
nmap -p 22,80,443 192.168.1.1
```

- **Scan an entire subnet:**

```
bash
CopyEdit
nmap -sn 192.168.1.0/24
```

---

5 Configuring & Managing Network Services

`systemctl` – Manage Network Services

- **Restart the networking service:**

```
bash
CopyEdit
sudo systemctl restart networking
```

- **Enable automatic startup:**

```
bash
CopyEdit
sudo systemctl enable networking
```

`firewalld` & `iptables` – Manage Firewall Rules

- **Check firewall status:**

```
bash
CopyEdit
```

```
sudo systemctl status firewalld
```

- **Allow a specific port (e.g., 80 for HTTP):**

```
bash
CopyEdit
sudo firewall-cmd --add-port=80/tcp --permanent
sudo firewall-cmd --reload
```

- **View current `iptables` rules:**

```
bash
CopyEdit
sudo iptables -L -v
```

---

6 Managing SSH Connections

`ssh` – Securely Connect to a Remote Server

- **Basic SSH connection:**

```
bash
CopyEdit
ssh user@remote-server-ip
```

- **Specify a port (default is 22):**

```
bash
CopyEdit
ssh -p 2222 user@remote-server-ip
```

- **Copy files between local and remote servers using `scp`:**

```
bash
CopyEdit
scp file.txt user@remote-server-ip:/home/user/
```

---

7 Checking Network Logs & Debugging Issues

View system logs related to networking:

```
bash
CopyEdit
sudo journalctl -u networking --since "1 hour ago"
```

Check active connections and open ports:

```
bash
CopyEdit
sudo lsof -i -P -n
```

Restart network services (for debugging network issues):

```
bash
CopyEdit
sudo systemctl restart NetworkManager
```

Conclusion

Linux provides a **powerful set of networking commands** that help in configuring, troubleshooting, and securing network connections. Mastering these commands will improve your ability to diagnose connectivity issues, monitor network traffic, and manage firewalls effectively.