Chapter 7: Networking Fundamentals

1. Why Networking Is Important in Linux

Most Linux servers:

- Communicate over networks
- Host APIs, websites, databases
- Interact with other services and cloud components

If networking fails:

- Application becomes unreachable
- Monitoring alerts fire
- Production incidents occur

Interview insight:

Many “application issues” are actually networking issues.

---

2. TCP vs UDP (VERY IMPORTANT)

TCP (Transmission Control Protocol)

Characteristics

- Connection-oriented
- Reliable
- Ordered delivery
- Error checking and retransmission

Use Cases

- HTTP / HTTPS
- SSH
- FTP
- Database connections

Interview line:

TCP guarantees delivery, but with overhead.

UDP (User Datagram Protocol)

Characteristics

- Connectionless
- Faster
- No delivery guarantee
- No retransmission

Use Cases

- DNS
- Video streaming
- VoIP
- Online gaming

Interview line:

UDP is faster but unreliable.

TCP vs UDP Comparison

Feature	TCP	UDP
Reliability	High	Low
Speed	Slower	Faster
Connection	Yes	No
Use case	Data integrity	Real-time data

---

3. What Is an IP Address?

Definition

An IP address uniquely identifies a device on a network.

Types:

- IPv4 (e.g., 192.168.1.10)
- IPv6 (e.g., 2001:db8::1)

Interview insight:

IPv4 is still most common in production.

---

4. What Is a Port?

Definition

A port is a logical endpoint that allows multiple services to run on the same IP address.

Common Ports (Interview Must-Know)

Service	Port
SSH	22
HTTP	80
HTTPS	443
MySQL	3306
PostgreSQL	5432

Interview line:

IP identifies the server; port identifies the service.

---

5. What Is a Socket?

Definition

A socket is a combination of:

- IP address
- Port
- Protocol (TCP/UDP)

Example:

```
192.168.1.10:80 (TCP)
```

Interview explanation:

A socket uniquely identifies a network connection.

---

6. DNS Resolution in Linux (VERY IMPORTANT)

What Is DNS?

DNS converts a **domain name into an IP address**.

DNS Resolution Flow

1. Application requests domain
2. Check `/etc/hosts`
3. Query DNS server
4. Receive IP
5. Connect to server

Interview insight:

/etc/hosts is checked before DNS.

Useful Files

- **`/etc/hosts`**
- **`/etc/resolv.conf`**

---

7. /etc/hosts

Purpose

Maps hostnames to IP addresses locally.

Example:

```
127.0.0.1   localhost
10.0.0.5    internal-api
```

Interview use case:

Used for testing or overriding DNS temporarily.

---

8. What Is NAT (Network Address Translation)?

Definition

NAT allows multiple private IPs to access the internet using a single public IP.

Why NAT Exists

- Saves public IP addresses
- Improves security
- Used in cloud and home networks

Interview line:

NAT hides internal IP addresses from the public network.

---

9. iptables (Firewall Basics)

Definition

iptables is a firewall tool that controls incoming and outgoing traffic.

What iptables Can Do

- Allow traffic
- Block traffic
- Forward traffic
- Filter by IP, port, protocol

Interview insight:

iptables works at the packet level.

---

10. Checking Open Ports

Common Commands

```
ss -tulnp
netstat -tulnp
```

Interview note:

ss is faster and more modern than netstat.

---

11. netstat vs ss

Feature	netstat	ss
Speed	Slower	Faster
Status	Deprecated	Active
Output	Verbose	Cleaner

Interview line:

ss is preferred on modern Linux systems.

---

12. Basic Network Troubleshooting Flow (VERY IMPORTANT)

When a service is not reachable:

1. **Check IP address**

```
ip a
```

2. **Check routing**

```
ip route
```

3. **Check port listening**

```
ss -tulnp
```

4. **Check firewall**
5. **Check DNS resolution**

```
ping
nslookup
```

Interview insight:

Always troubleshoot from network → service → application.

---

13. Real-Life Production Scenarios

Scenario 1: Application Works Locally but Not Remotely

- Service bound to localhost
- Firewall blocking port
- Port not exposed

Scenario 2: Cannot SSH into Server

- SSH service down
- Port 22 blocked
- Wrong IP or DNS issue

Scenario 3: DNS Resolution Fails

- Incorrect resolv.conf
- DNS server unreachable
- Network misconfiguration

Chapter 7: Interview Takeaways

After this chapter, you should be able to:

- Explain TCP vs UDP confidently
- Understand IP, ports, and sockets
- Explain DNS resolution flow
- Use /etc/hosts correctly
- Check open ports
- Troubleshoot network issues logically
- Explain firewall basics
