Networking Basics

(with real outage lessons)

Day 5 of learning DevOps

OSI Model

where the problem lives matters

Outage:

Website was down.
Teams restarted the app.

Reality:

DNS was failing.

Wrong layer → wasted hours.
Right layer → quick fix.

IP Address

identity, not connectivity

Outage:

New server had an IP.
Still unreachable.

Reality:

Firewall rules blocked traffic.

Server existed.
Connection didn’t

Subnet

who can talk directly

Outage:

Two services worked in staging,
failed in prod.

Reality:

They were in different subnets
with no routing.

Same code.
Different network boundaries.

CIDR

how big the network really is

Outage:

A subnet ran out of IPs.
New pods stopped starting.

Reality:

CIDR range was too small.

Capacity planning is networking too.

Routes

decisions, not guesses

Outage:

Private API suddenly stopped responding.

Reality:

A route table entry was removed during cleanup.

Packets had nowhere to go.
No errors. Just silence.

Router

bridge between networks

Outage:

Services in different networks couldn’t talk.

Reality:

No router / forwarding path existed.

Different networks don’t “auto-connect”.

DNS

name to number translation

Outage:

Load balancer was replaced.

Reality:

DNS TTL was high.

Some clients still used old IPs.

Random failures = DNS classic.

Silent Failures

Networking failures are quiet

No stack trace.

No crash logs.

Only timeouts.

That’s why networking issues feel scary.

DevOps Realisation

Most outages are not code bugs.

They are:

- missing routes
- wrong subnets
- blocked traffic
- broken DNS

Mental Model

IP →

Subnet →

Route →

Router →

Return Path →

Firewall →

DNS

Break one → system breaks.

Networking Commands (80/20 for DevOps)

These few commands solve most networking issues:

- **ip addr → Who am I on the network?**
- **ip route → Where does my traffic go?**
- **ping → Can packets travel and return?**
- **traceroute → Where is the packet getting stuck?**
- **nslookup → What IP does DNS return?**

If these make sense,
networking stops feeling random.

Commands help you fix.
Concepts help you think.

Understand the concept
behind the command —
not just the command itself.
That feels like progress.

Learning networking changed how I debug systems.

BY: Samyak Jain (Aspiring DevOps Engineer)