OSI Model (Open Systems Interconnection) - Explained for DevOps

----

The OSI model is a framework that describes how data moves through a network in 7 layers.

Layer 7 - Application Layer:

- Closest to the user.
- Protocols: HTTP, HTTPS, SSH, FTP, DNS
- DevOps Examples: curl, SSH, AWS CLI, Terraform

Layer 6 - Presentation Layer:

- Translates data formats and handles encryption.
- Protocols: SSL, TLS, JSON, XML
- DevOps Examples: TLS in HTTPS, API JSON response formatting

Layer 5 - Session Layer:

- Manages sessions (start, maintain, end).
- DevOps Examples: SSH sessions, web login sessions

Layer 4 - Transport Layer:

- Ensures data is delivered correctly.
- Protocols: TCP, UDP
- DevOps Examples: Ports 22 (SSH), 80 (HTTP), 443 (HTTPS)

Layer 3 - Network Layer:

- Handles IP addresses and routing.

- Protocols: IP (IPv4, IPv6)

- DevOps Examples: CIDR, VPC routing, IP assignment, traceroute

Layer 2 - Data Link Layer:

- Works with MAC addresses and switches.

- DevOps Examples: Ethernet troubleshooting in hybrid cloud

Layer 1 - Physical Layer:

- Actual physical hardware (cables, NICs, signals).

- DevOps Examples: Data center cables, Wi-Fi signal

DevOps Focus: Layers 7, 6, 4, 3 (most used in cloud & tool setup)

Understanding Ports in DevOps

What is a Port?

- A port is a logical number (0 to 65535) used to identify a specific service on a server.

- Example: SSH uses port 22, HTTP uses port 80.

Why Needed?

- A server runs multiple services; ports help distinguish them.

- IP = house address, Port = room number

Port Ranges:

- 0-1023: Well-known ports (HTTP, SSH, FTP, DNS)
- 1024-49151: Registered ports
- 49152-65535: Dynamic/private ports

Types:

- Public Port: Open to the internet (e.g., 80 for web server)
- Private Port: Used internally inside VPC or local system

DevOps Use Cases:

- AWS EC2: Opening ports in security groups
- Docker: Mapping container port to host port
- Kubernetes: Exposing services via NodePort
- Firewall rules: Allowing/blocking specific ports

Common DevOps Ports:

- SSH: 22
- HTTP: 80
- HTTPS: 443
- MySQL: 3306
- PostgreSQL: 5432
- Jenkins: 8080
- Docker API: 2375
- Kubernetes API: 6443

Summary:

- Port = identifies service on a server

- Combine IP + Port to reach specific app (e.g., 192.168.1.10:8080)
- DevOps regularly deals with public/private ports, opening ports in cloud security groups, containers, and firewalls.