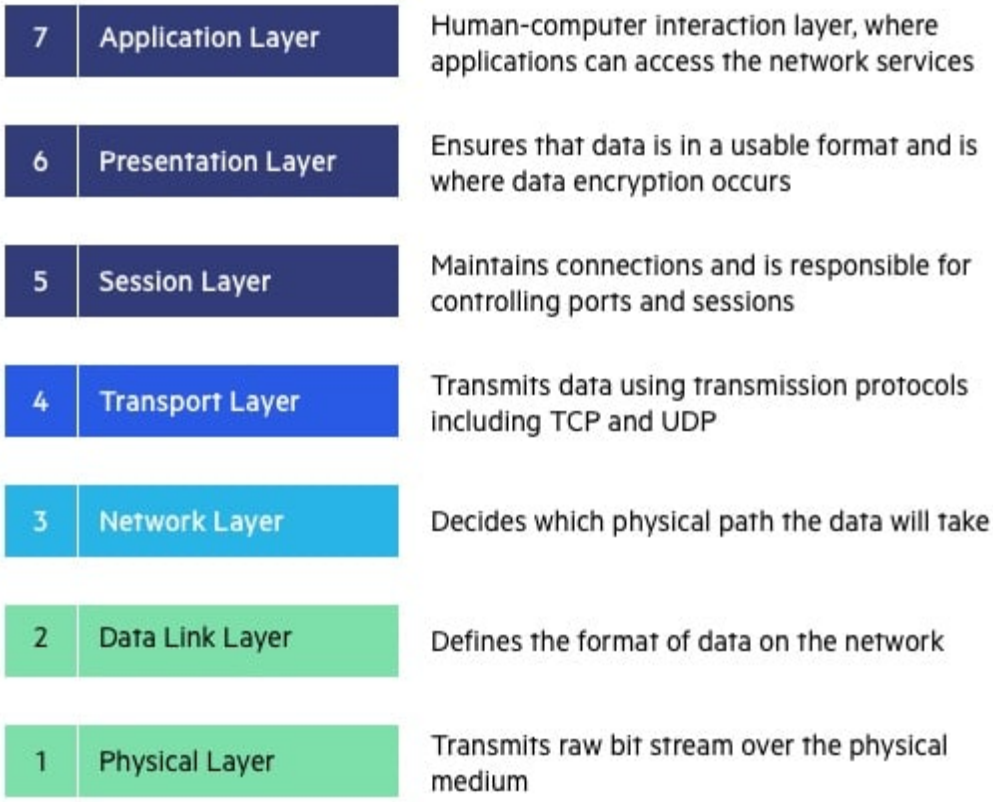
# Networking for DevOps Engineers!

## Networking concepts every devops engineer should know:

1. OSI Model
2. Protocols : TCP/UDP/IP
3. Ports
4. Subnetting
5. Routing
6. DNS
7. VPN (Virtual Private Network)
8. Networking tools

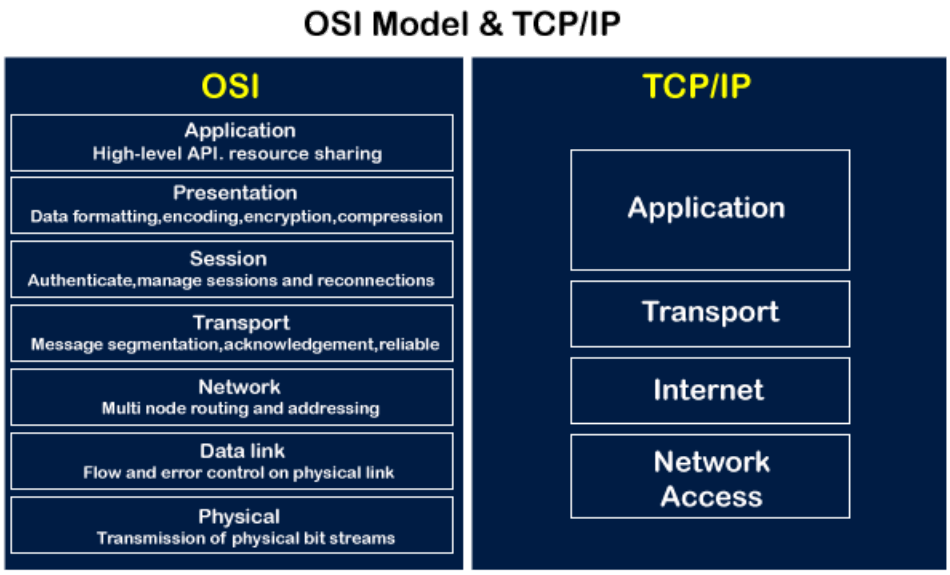
## OSI Model:

The OSI (Open Systems Interconnection) model is a framework with seven layers that standardizes how different computer systems communicate. From the physical connection (Layer 1) to end-user services (Layer 7), each layer has a specific role in managing aspects like hardware, addressing, routing, and application-level interactions. It simplifies understanding and troubleshooting network processes.




The model helps understand and troubleshoot network processes by breaking them down into distinct layers.

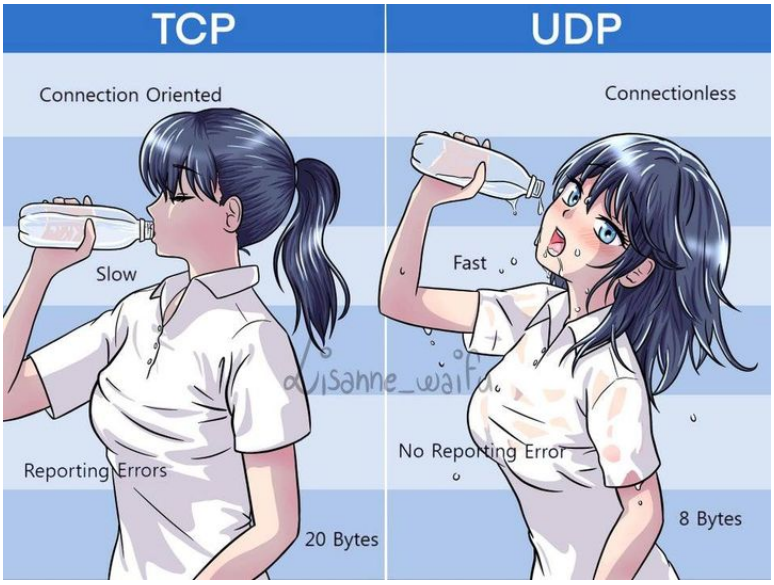
Learning Resource: https://www.geeksforgeeks.org/open-systems-interconnection-model-osi/




# Protocols :

A protocol is a set of rules that defines how data is transmitted and received between devices in a network. It ensures standardized communication, allowing different systems to understand and interact with each other. Examples include TCP/IP, HTTP, and SMTP.

1. **TCP (Transmission Control Protocol):**
    - **Description:** TCP operates at the transport layer of the OSI model. It establishes a connection between two devices before data exchange, ensuring reliable and ordered delivery of information.
    - **Functionality:** It breaks data into packets, assigns sequence numbers, and uses acknowledgment messages to guarantee delivery. It's connection-oriented, meaning it sets up, maintains, and terminates a connection for data exchange.
2. **UDP (User Datagram Protocol):**
    - **Description:** Also operating at the transport layer, UDP is a connectionless protocol that offers minimal services. It's like a 'fire and forget' approach for data transmission.
    - **Functionality:** It sends data without establishing a connection, providing low-latency communication. However, it doesn't guarantee delivery or order, making it suitable for real-time applications like video streaming or online gaming.




1. **IP (Internet Protocol):**
    - **Description:** IP functions at the network layer and is a fundamental part of the TCP/IP protocol suite. It handles addressing and routing to ensure data packets reach their intended destinations.
    - **Functionality:** IP assigns unique IP addresses to devices and uses routing tables to direct data across networks. It's responsible for the logical connection between different devices on the Internet.

In short, TCP ensures reliable and ordered communication with a connection-oriented approach, UDP prioritizes speed and is connectionless, and IP manages the addressing and routing for data packets across networks. Together, they form the backbone of internet communication.

Learning resource: https://www.pluralsight.com/blog/it-ops/networking-basics-tcp-udp-tcpip-osi-models

# Ports:

Ports are **communication endpoints** that allow different services on a device to send and receive data.

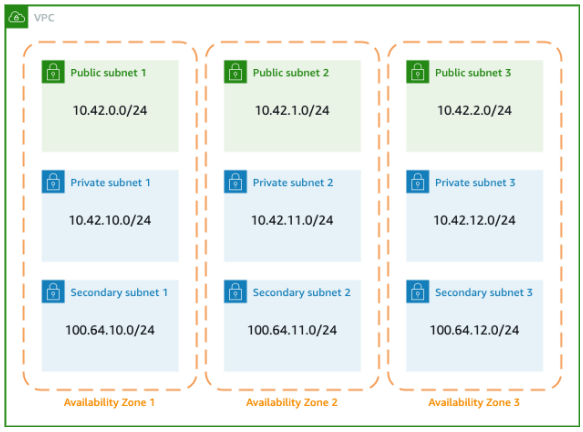
A DevOps engineer should know about ports because they are important in configuring networking settings, defining rules for firewalls, managing container communication, orchestrating services, and troubleshooting network issues.

| Port Number | Service / Protocol |
|---|---|
| 20 | File Transfer Protocol (FTP) |
| 21 | File Transfer Protocol (FTP) |
| 22 | Secure Shell (SSH) Secure Login |
| 23 | Telnet Remote Login (Unsecured) |
| 25 | Simple Mail Transfer Protocol (SMTP) |
| 53 | Domain Name System (DNS) Service |
| 80 | Hypertext Transfer Protocol (HTTP) |
| 110 | Post Office Protocol (POP3) |
| 123 | Network Time Protocol (NTP) |
| 143 | Internet Message Access Protocol (IMAP) |
| 161 | Simple Network Management Protocol (SNMP) |
| 443 | HTTP Secure (HTTPS) |

Learning resource: https://www.cloudflare.com/en-in/learning/network-layer/what-is-a-computer-port/

# Subnetting (CIDR)

DevOps engineers should understand IP addressing, subnetting, and CIDR notation. This knowledge is crucial for designing and managing IP address spaces effectively.



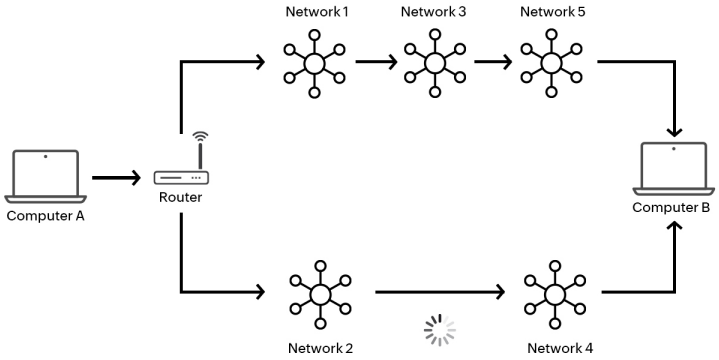

Subnetting is a technique used in computer networking to divide a larger IP network into smaller, more manageable sub-networks or subnets.

| IP Addresses | Bits | Prefix | Subnet Mask |
|---|---|---|---|
| 1 | 0 | /32 | 255.255.255.255 |
| 2 | 1 | /31 | 255.255.255.254 |
| 4 | 2 | /30 | 255.255.255.252 |
| 8 | 3 | /29 | 255.255.255.248 |
| 16 | 4 | /28 | 255.255.255.240 |
| 32 | 5 | /27 | 255.255.255.224 |
| 64 | 6 | /26 | 255.255.255.192 |
| 128 | 7 | /25 | 255.255.255.128 |
| 256 | 8 | /24 | 255.255.255.0 |
| 512 | 9 | /23 | 255.255.254.0 |
| 1 K | 10 | /22 | 255.255.252.0 |
| 2 K | 11 | /21 | 255.255.248.0 |
| 4 K | 12 | /20 | 255.255.240.0 |
| 8 K | 13 | /19 | 255.255.224.0 |
| 16 K | 14 | /18 | 255.255.192.0 |
| 32 K | 15 | /17 | 255.255.128.0 |

Learning resource: https://youtu.be/I_LXaIg6mkM?si=H71TnHbE8oOVfMzp

# Routing

Routing is the process of directing data packets from a source to a destination across a network. Routers use routing tables and protocols to decide the path for data transmission, ensuring efficient and reliable communication between devices.

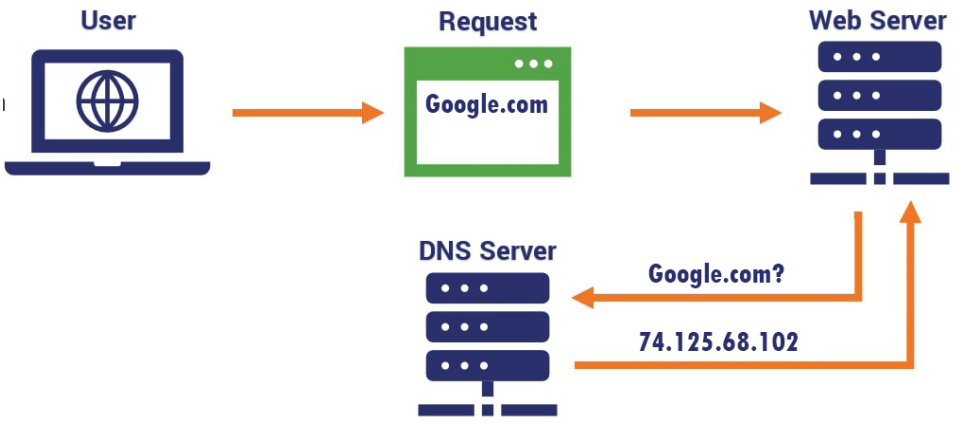



Learning resource: https://aws.amazon.com/what-is/routing/#:~:text=Routing is the process of,place through many different paths.

# DNS

DNS, or Domain Name System, translates easy-to-remember domain names to computer-friendly IP addresses, helps find mail servers, balances web traffic among servers, redirects requests, performs reverse lookups, and speeds up responses through caching. It's a vital system that ensures efficient and reliable communication on the internet.

Google IP address: 74.125.68.102

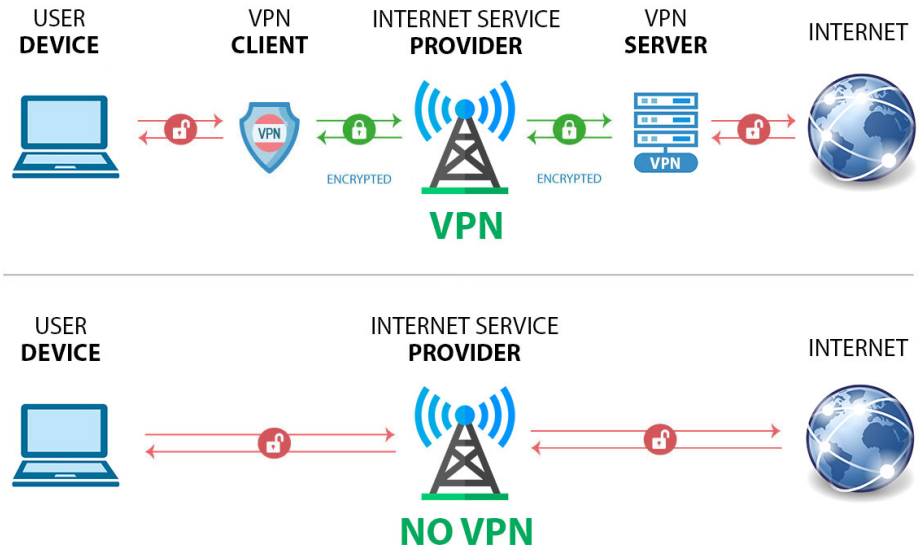



| Record Type | Purpose |
|---|---|
| **Addresses** | |
| A | Maps a fully qualified domain name (FQDN) to an IPv4 address |
| AAAA | Maps a FQDN to an IPv6 address |
| **Aliases** | |
| CNAME | Maps a FQDN to another FQDN |
| DNAME | Maps all subdomains of a FQDN to another FQDN |
| **Servers** | |
| NS | Maps a subdomain to a FQDN of a name server |
| MX | Maps an email domain to a FQDN of a mail server |

Learning Resource: https://aws.amazon.com/route53/what-is-dns/

# VPN

A VPN, or Virtual Private Network, is like a secure tunnel for your internet connection. It encrypts your data and routes it through a server, making your online activities more private and secure.




VPNs allow professionals to securely connect to remote servers, access cloud resources, and perform maintenance tasks without compromising data security. It ensures a private and encrypted connection, crucial when dealing with sensitive configurations, deployments, or infrastructure management tasks.

Learning Resource: https://www.devopsschool.com/blog/5-important-reasons-why-devops-need-to-use-a-vpn/#:~:text=So%2C there you have it,on DevOps%2C you need it.

# Networking tools

Important networking tools for DevOps Engineer:

1. **Ping:**

- **Purpose:** To check the reachability of a host on an Internet Protocol (IP) network.
- **Usage Example:** `ping google.com`

2. **Traceroute (or traceroute6):**
   - **Purpose:** To display the route and measure transit delays of packets across an Internet Protocol network.
   - **Usage Example:** `traceroute google.com`
3. **Netstat:**
   - **Purpose:** To display active network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.
   - **Usage Example:** `netstat -a`
4. **Nmap:**
   - **Purpose:** To discover hosts and services on a computer network, creating a "map" of the network.
   - **Usage Example:** `nmap -p 1-1000 target`
5. **Tcpdump:**
   - **Purpose:** To capture and analyze network traffic.
   - **Usage Example:** `tcpdump -i eth0`
6. **Ipconfig (Windows) / ifconfig (Linux):**
   - **Purpose:** To display the configuration of network interfaces.
   - **Usage Example (Linux):** `ifconfig`
7. **Dig (Domain Information Groper):**
   - **Purpose:** To query DNS name servers for information about host addresses, mail exchanges, name servers, and related information.
   - **Usage Example:** `dig google.com`
8. **Nslookup (Windows) / host (Linux):**
   - **Purpose:** To query DNS servers for domain information.

- **Usage Example (Linux):** `host google.com`

9. **Wireshark:**
    - **Purpose:** A network protocol analyzer for troubleshooting and analysis of the interactions between network components.
    - **Usage Example:** Capture and analyze packets on a specific network interface.
10. **Iperf:**
    - **Purpose:** To measure the TCP and UDP performance of a network.
    - **Usage Example:** `iperf -s` (server) and `iperf -c <server-ip>` (client).

These tools are invaluable for diagnosing network issues, understanding network performance, and ensuring the proper functioning of network connections. They remain essential for both network administrators and DevOps engineers in their day-to-day tasks.