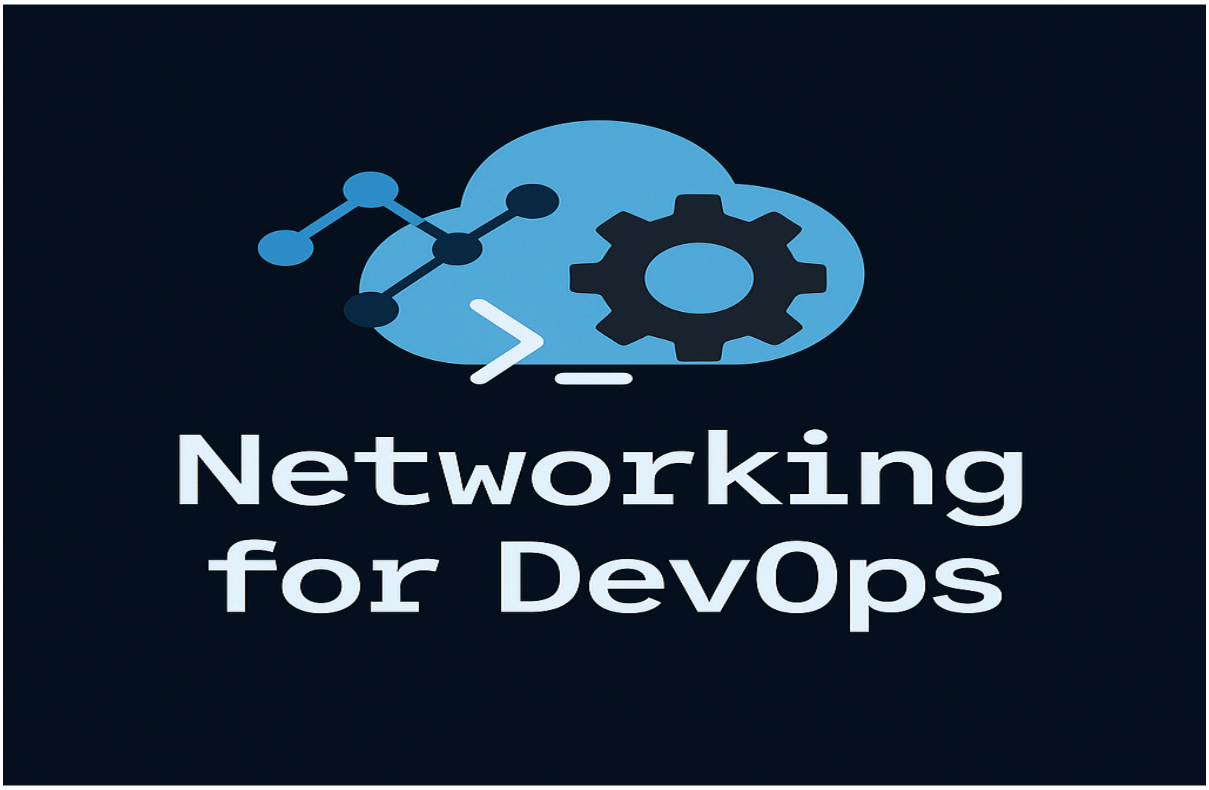


Networking basics are essential for DevOps engineers because they provide a foundational understanding of how different components of a system communicate. This knowledge is crucial for troubleshooting issues, securing the infrastructure, implementing automation, and optimizing performance. DevOps involves collaboration between development and operations, and a grasp of networking principles enables effective communication and coordination across distributed applications.

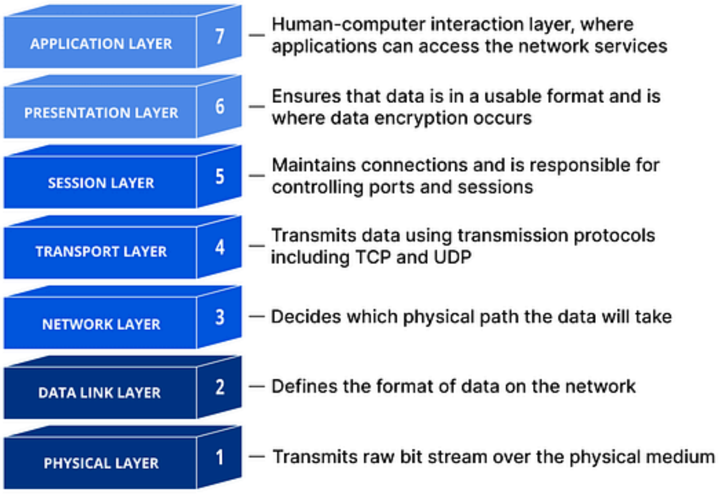
Networking concepts should know:

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

1.OSI Model

The OSI (Open Systems Interconnection) model is a conceptual framework that standardizes the functions of a telecommunication or computing system into seven abstraction layers. Each layer has a specific role in managing aspects like hardware, addressing, routing, and application-level interactions. The OSI model is used to understand and design network architectures, and it helps in troubleshooting network communication issues.

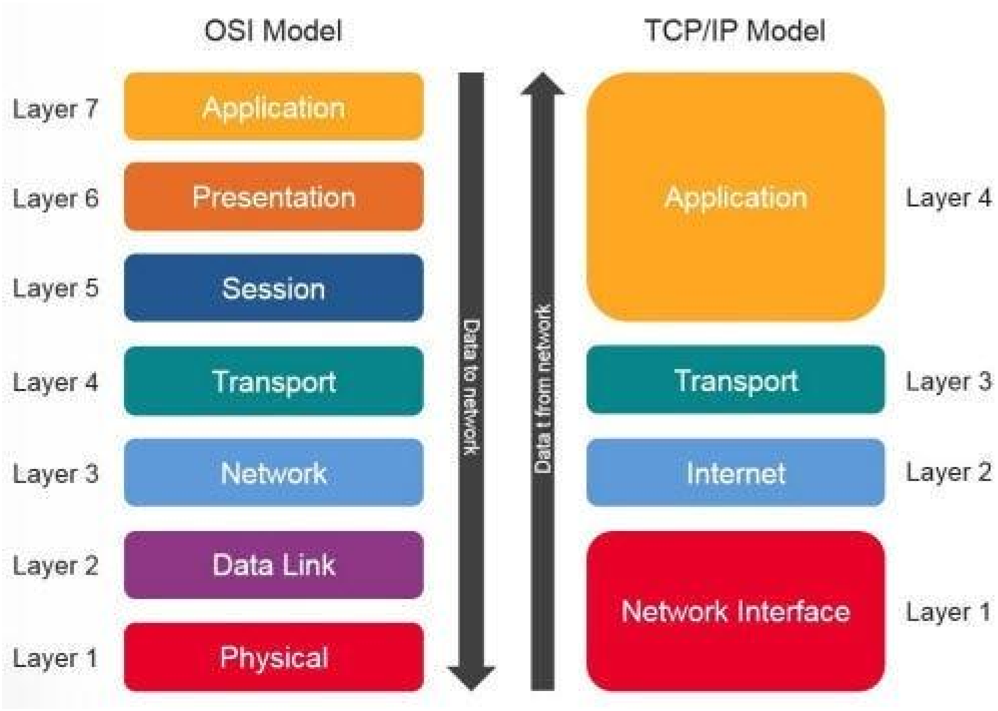
Zoom image will be displayed


APPLICATION LAYER
7
Human-computer interaction layer, where applications can access the network services
PRESENTATION LAYER
6
Ensures that data is in a usable format and is where data encryption occurs
SESSION LAYER
5
Maintains connections and is responsible for controlling ports and sessions
TRANSPORT LAYER
4
Transmits data using transmission protocols including TCP and UDP
NETWORK LAYER
3
Decides which physical path the data will take
DATA LINK LAYER
2
Defines the format of data on the network
PHYSICAL LAYER
1
Transmits raw bit stream over the physical medium

Learning Resource:

https://www.geeksforgeeks.org/open-systems-interconnectionmodel-osi/

Osi model is great understanding of network but it is very challenging to use in practice. this is why we using TCP/IP model.



Top three layers are combined into one layer and bottommost two layers are combined into one layer. so instead of having seven layer we have four layers model and it is quiet easy to practice. Because we are using this.

2. Protocols : TCP/UDP/IP

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

i. TCP (Transmission Control Protocol):

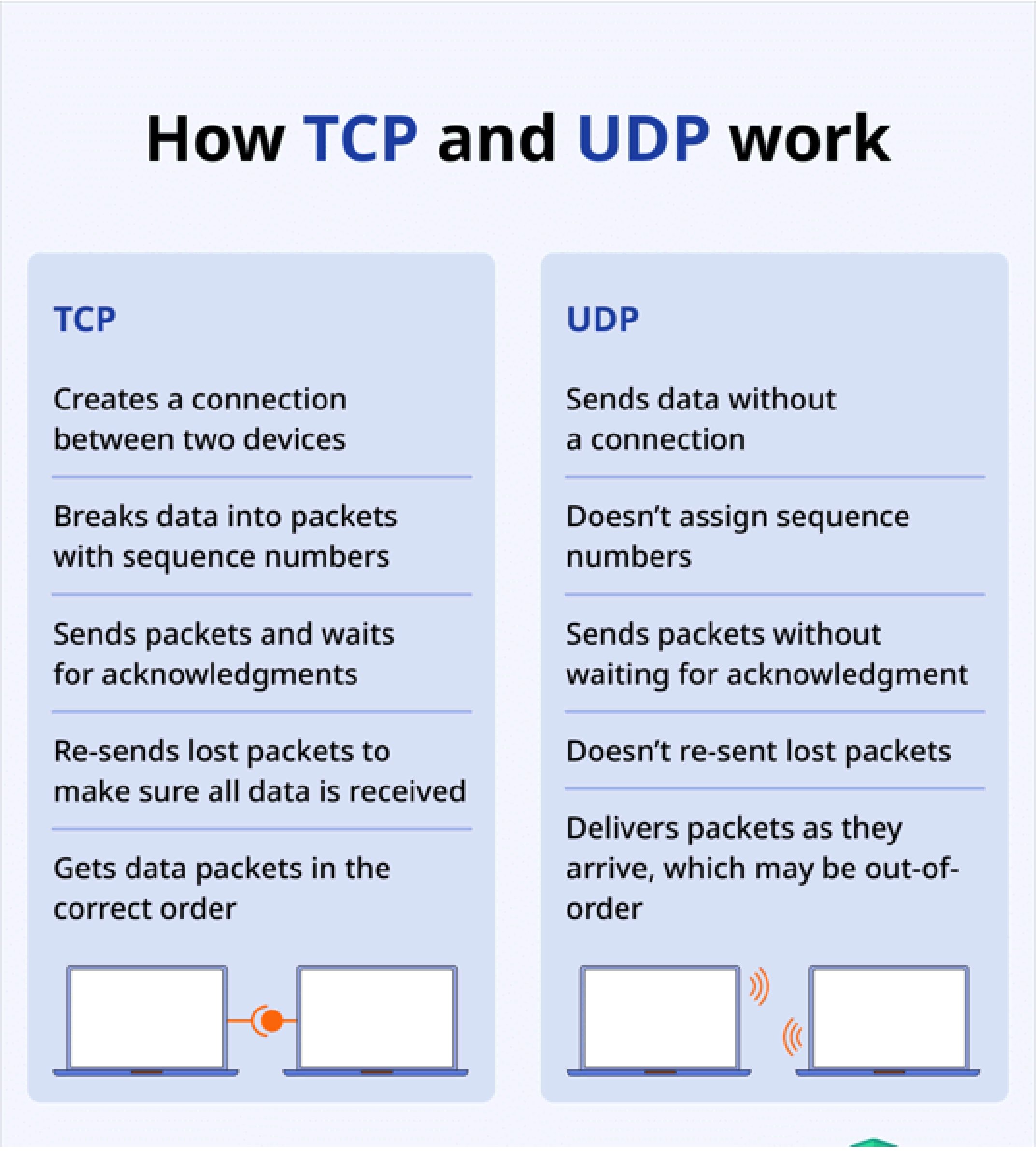
Description: TCP operates at the transport layer of the OSI model. It establishes a connection between two devices before data exchange, ensuring reliable and ordered delivery of information.

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.

ii. 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.



How TCP and UDP work

TCP

- Creates a connection between two devices
- Breaks data into packets with sequence numbers
- Sends packets and waits for acknowledgments
- Re-sends lost packets to make sure all data is received
- Gets data packets in the correct order

UDP

- Sends data without a connection
- Doesn’t assign sequence numbers
- Sends packets without waiting for acknowledgment
- Doesn’t re-sent lost packets
- Delivers packets as they arrive, which may be out-of-order

iii. 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.

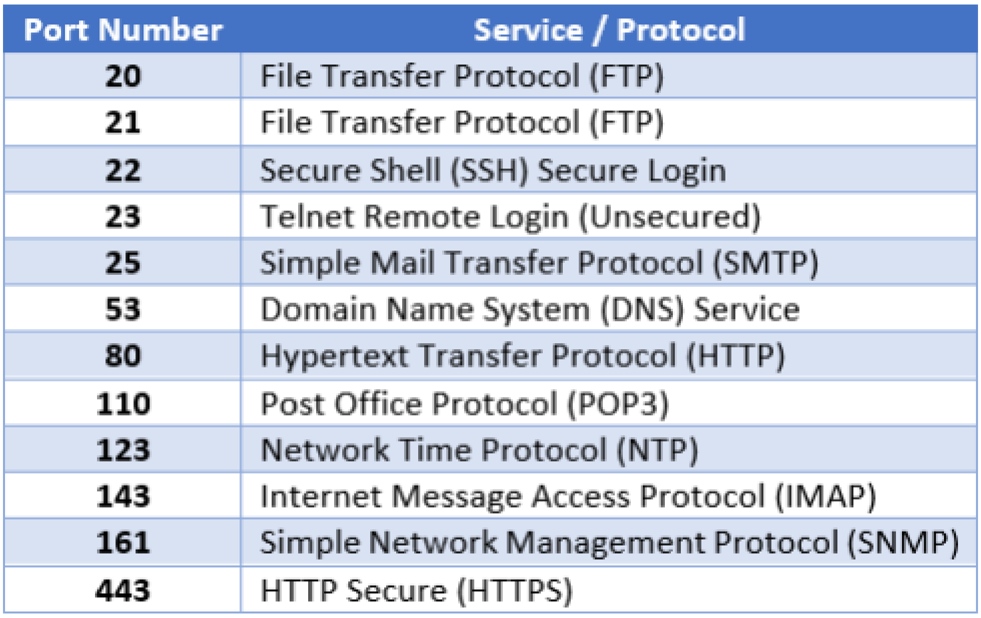
Functionality: IP assigns unique IP addresses to devices and uses routing tables to direct data across networks. It’s responsible for the logical connection between different devices on the Internet.

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.

3. Ports

Ports are essential for directing network traffic to specific applications or services on devices.

DevOps engineers need a solid understanding of ports to manage the networking aspects of application deployment, configuration, and maintenance. This knowledge is crucial for building robust, scalable, and secure infrastructures in a DevOps environment.



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)

4. Subnetting

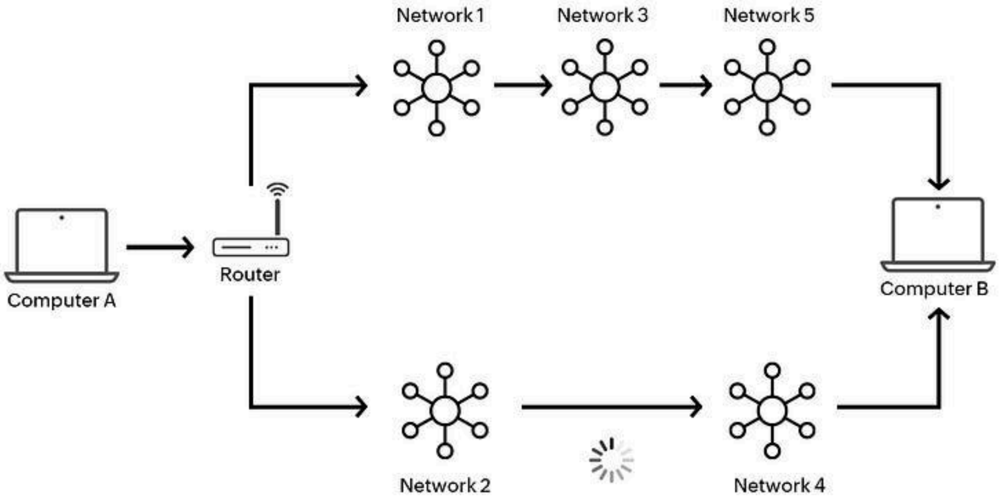
For a DevOps engineer, understanding subnetting is important for several reasons related to networking and infrastructure management. This knowledge is essential for effective IP address management, secure network configurations, cloud networking, and troubleshooting in a DevOps environment.

CIDR Notation: Understanding Classless Inter-Domain Routing (CIDR) notation is essential for expressing IP address ranges and subnet masks in a concise and standardized format. DevOps engineers commonly encounter CIDR notation in network configurations.

5.Routing

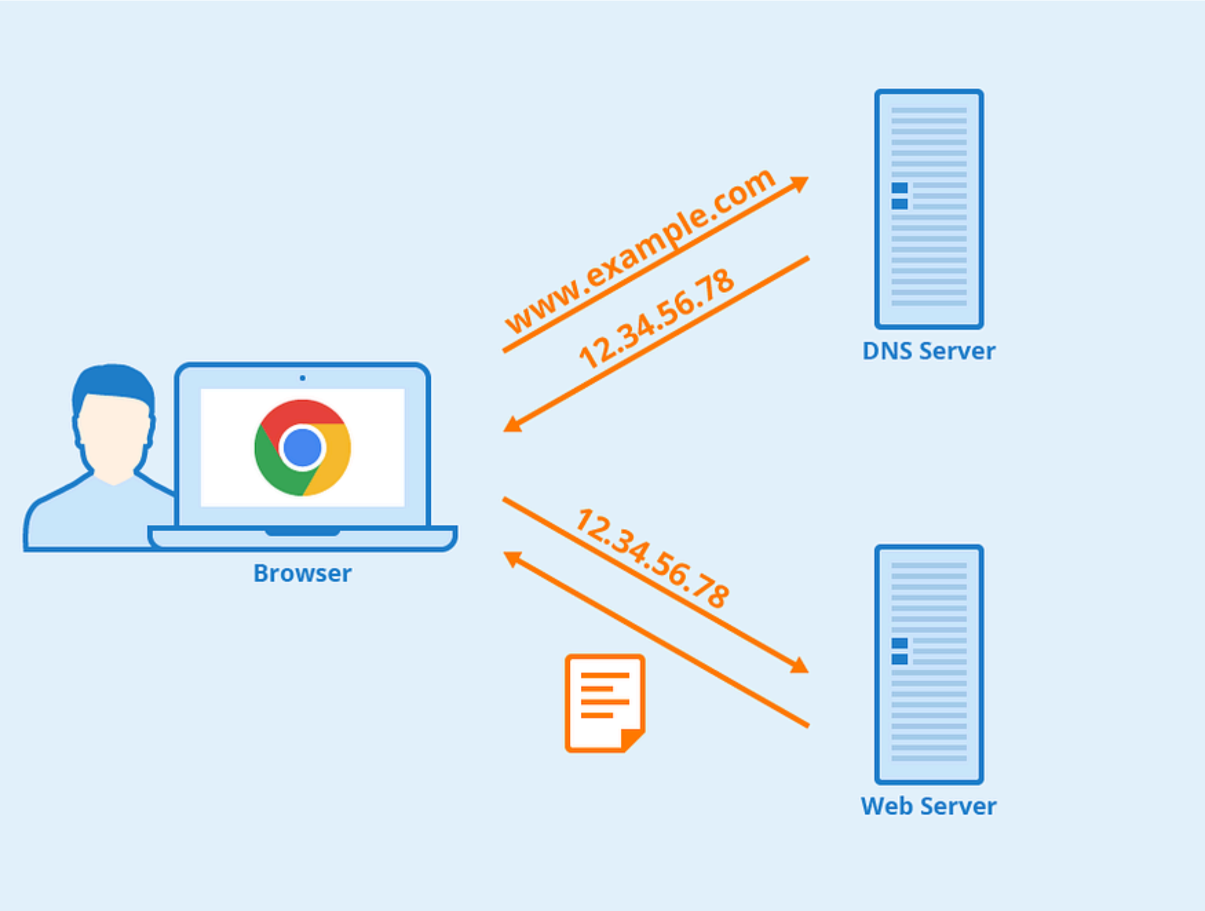
Routing is the process of directing data packets from a source to a destination across a network. Routers use routing tables and protocols to decide the path for data transmission, ensuring efficient and reliable communication between devices.

Zoom image will be displayed



6. 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.



DNS Record Types:

- A Record (Address Record): Maps a domain to an IPv4 address.
- AAAA Record: Maps a domain to an IPv6 address.
- CNAME Record (Canonical Name): Alias of one domain to another.
- MX Record (Mail Exchange): Specifies mail servers for the domain.
- NS Record (Name Server): Identifies authoritative DNS servers for the domain.

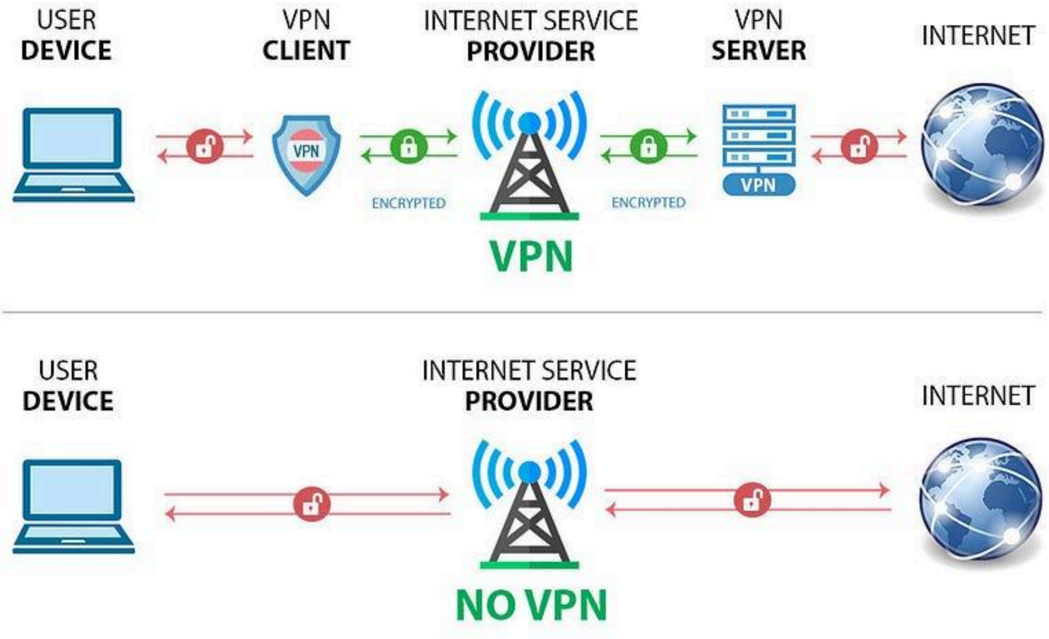
Google IP address: 74.125.68.102

Learning Resource: What is DNS? — Introduction to DNS — AWS (amazon.com)

7. 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

Zoom image will be displayed



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.

8. **Networking tools:**

Important networking tools for DevOps Engineer:

i. Ping:

Purpose: *To check the reachability of a host on an Internet Protocol (IP) network.*

Usage Example: *ping google.com*

ii. Traceroute (or traceroute6):

Purpose: To display the route and measure transit delays of packets across an Internet Protocol network.

Usage Example: traceroute google.com

iii. Netstat :

Purpose: To display active network connections, routing tables, interface statistics, masquerade connections, and multicast memberships.

Usage Example: netstat -a

iv. Nmap :

Purpose: To discover hosts and services on a computer network, creating a “map” of the network.

Usage Example: nmap -p 1–1000 target

v. Tcpdump :

Purpose: To capture and analyze network traffic.

Usage Example: tcpdump -i eth0

vi. Ipconfig (Windows) / ifconfig (Linux):

Purpose: *To display the configuration of network interfaces.*

Usage Example (Linux): *ifconfig*

vii. 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*

viii. Nslookup (Windows) / host (Linux):

Purpose: *To query DNS servers for domain information.*

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

ix. 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.*

x. Iperf :

Purpose: *To measure the TCP and UDP performance of a network.*

Usage Example: *iperf -s (server) and iperf -c (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