**NETWORKING CONCEPT**

**1. IP Addressing**

IP (Internet Protocol) addressing is the foundation of networking. It provides unique identifiers for devices on a network and enables them to communicate.

* **IP Versions**:
  + **IPv4**: Uses 32-bit addresses (e.g., 192.168.1.1), which allows for approximately 4.3 billion unique addresses.
  + **IPv6**: Uses 128-bit addresses (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334), significantly increasing the number of unique addresses to accommodate the growing number of devices.
* **IP Address Components**:
  + **Network ID**: Identifies the network to which the device belongs.
  + **Host ID**: Identifies the specific device within that network.
* **Private and Public IPs**:
  + **Public IP**: Globally unique, used to identify devices on the internet.
  + **Private IP**: Used within local networks and not routable on the internet (e.g., 192.168.x.x, 10.x.x.x).
* **Subnetting**: Divides a large network into smaller sub-networks (subnets), optimizing IP address usage and improving security and performance.

**2. DNS (Domain Name System)**

DNS translates human-readable domain names (e.g., example.com) into IP addresses that computers use to communicate.

DNS is a critical component of networking, acting as the internet's phonebook. It translates domain names (like www.example.com) into IP addresses (like 192.168.1.1), allowing users to access websites using human-readable names instead of remembering complex IP addresses.

**Key DNS Concepts:**

**1.Domain Names**:

A domain name is the user-friendly address for a website or service (e.g., www.google.com).

It's organized hierarchically with:

**1.Top-Level Domain (TLD)**: The last part (e.g., .com, .org).

**2.Second-Level Domain (SLD)**: The main part of the domain (e.g., google in google.com).

**3.Subdomains**: Parts of a domain that serve different purposes (e.g., mail.google.com).

**2.DNS Resolution Process**: When you type www.example.com into a browser:

* **Step 1**: The browser checks its cache for the IP address associated with example.com.
* **Step 2**: If not cached, the request is sent to a recursive DNS server (usually provided by your ISP or set up manually, e.g., Google's DNS 8.8.8.8).
* **Step 3**: The recursive server checks if it has the IP address cached. If not, it queries authoritative DNS servers in the following order:
  + **Root DNS servers**: They point to the TLD DNS servers (e.g., .com).
  + **TLD DNS servers**: They point to the authoritative servers for the specific domain (e.g., example.com).
  + **Authoritative DNS servers**: They return the IP address for the domain.

**3.DNS in AWS (Route 53)**:

* **Amazon Route 53** is AWS's DNS service, which allows you to manage domain names and perform DNS routing for your applications.
* Route 53 provides advanced features like health checks, traffic routing (e.g., latency-based, geo-based), and DNS failover.

**4.DNS Commands in Linux**

1. **nslookup**:
   * Query DNS servers to find IP addresses for domain names or domain names for IP addresses.
   * Example: nslookup www.example.com
2. **dig**:
   * A powerful DNS lookup tool that provides detailed DNS information.
   * Example: dig example.com (returns detailed DNS information for the domain).
3. **host**:
   * A simple tool to perform DNS lookups.
   * Example: host example.com

These tools allow you to troubleshoot DNS issues or query DNS records directly from your Linux system. In AWS, you can also configure and test DNS settings for your EC2 instances or services using Route 53.

**Routing:**

Routing is the process of directing network traffic from one location to another across interconnected networks, ensuring that data packets travel efficiently to their destination. Routers, which operate at the network layer (Layer 3) of the OSI model, are responsible for determining the best path for data packets based on network topology and routing tables.

**Key Concepts in Routing:**

1. **Routing Table**:
   * A router (or any device performing routing functions) maintains a routing table, which contains a set of rules or routes that determine where packets should be forwarded.

**Types of Routes**:

* **Directly Connected Routes**: When a router is physically connected to a network, the route is automatically known (e.g., a router connected to 192.168.1.0/24 network).
* **Static Routes**: Manually configured routes by a network administrator. These are simple but less flexible as they don’t adapt to changes automatically.
* **Dynamic Routes**: Learned and updated by routers using dynamic routing protocols (e.g., OSPF, BGP, EIGRP), which adjust automatically based on network changes.

**Default Route**:

* A special route that directs packets destined for unknown networks (i.e., networks not listed in the routing table) to a default next-hop address (typically the router's gateway to the internet).
* Example: 0.0.0.0/0 is used to denote the default route, often pointing to the router or gateway connected to the internet.

**Routing in AWS**

In AWS, routing typically involves setting up and managing routes within **Virtual Private Clouds (VPCs)** and configuring:

1. **Route Tables**:
   * Each subnet in a VPC is associated with a route table, which defines how traffic is directed within the VPC and to external networks (such as the internet).
   * Example:
     + A route for traffic within the VPC: 10.0.0.0/16 → local (meaning within the VPC).
     + A default route to the internet: 0.0.0.0/0 → Internet Gateway (IGW).
2. **Internet Gateway (IGW)**:
   * A horizontally scaled, redundant, and highly available VPC component that allows communication between instances in the VPC and the internet.
3. **NAT Gateway**:
   * Used to allow private subnets to access the internet while keeping the instances themselves unreachable from the outside.
4. **VPC Peering**:
   * Allows routing of traffic between VPCs in the same or different AWS regions.

**Routing Commands in Linux**

1. **route**:
   * Displays or manipulates the routing table. Useful for checking or setting static routes.
   * Example: route -n (shows the routing table without resolving IP addresses to names).
2. **ip route**:
   * A more modern and versatile tool for manipulating the routing table.
   * Example: ip route show (displays the current routing table).
3. **traceroute**:
   * Traces the route that packets take from your machine to a destination.
   * Example: traceroute www.example.com (displays the path taken by packets to reach www.example.com).

These tools allow you to view and troubleshoot routing configurations on your AWS EC2 instances or any Linux-based systems.

To check network configurations in Linux (specifically for an Amazon EC2 instance), you can use several commands to gather information about network interfaces, IP addresses, routing tables, and network status.

**Common Commands for Checking Network Configuration:**

**1.ifconfig** (deprecated but still useful)

* + Displays information about all active network interfaces.
  + It shows the IP address, netmask, broadcast address, MAC address, and other interface details.

🡪ifconfig

You’ll see details such as:

* inet addr: The IP address assigned to the interface.
* Mask: The subnet mask.
* HWaddr: The MAC address of the interface.

2. **ip addr** (replacement for ifconfig)

* Provides more detailed information about network interfaces and their IP addresses

🡪ip addr show

3. **Output will show**:

* IP addresses (IPv4 and IPv6).
* Interface status (up or down).
* MAC address.

**ip link**

* Shows all the available network interfaces and their states (whether they are up or down)
* route -n : Command used to View or modify Routing tab;e

A **routing table** is a data structure stored in a router or a host computer that lists the routes to particular network destinations. It contains information about the paths available to send network traffic to its destination. Each entry in the routing table typically contains:

**Key Components of a Routing Table:**

1. **Destination**: The IP address of the destination network or host.
2. **Gateway (Next Hop)**: The IP address of the next device (usually a router) to which the packet should be forwarded to reach its final destination.
3. **Genmask (Netmask)**: The subnet mask that defines the network portion of an IP address.
4. **Interface**: The outgoing network interface (e.g., eth0) through which the packet should be sent.
5. **Metric**: A value that defines the "cost" or preference for using a particular route; lower metrics typically indicate more preferred routes.
6. **Flags:**
7. **U**: The route is up.
8. **G**: The route uses a gateway.
9. **H**: The destination is a host (specific IP).
10. **!**: A reject route, which means traffic for this destination is discarded.
11. **Routing Table Use Cases:**
12. When a host sends a packet, it uses the routing table to determine which network interface (or next hop router) to send the packet through.
13. A router uses its routing table to forward packets between different networks.

**Network Troubleshooting**

**1.ping**  
ping tests network connectivity to a specific host and measures the round-trip time for messages.  
**🡪**ping browser.com

This sends ICMP packets to Google's server and checks if it responds. And also gives the time taken to travel from instance to server and back, and bytes shows how much data was sent

The **rtt min/avg/max/mdev** values show the minimum, average, maximum round-trip times, and the mean deviation.

**2.Test Network Connectivity**

**🡪**ping

* **Purpose:** Tests network connectivity by sending ICMP echo requests**.**

**3. Trace Route to Remote Host**

**🡪** traceroute

* **Purpose:** Shows the path packets take to reach a destination, useful for identifying network hops and delays.

**4. Check Routing Table**

**🡪** route -n **or** ip route

* **Purpose:** Displays the routing table and shows how traffic is being directed**.**

**5. List Network Connections**

**🡪**netstat  **or** ss -antp

* **Purpose:** Lists all active connections, their protocol, local and foreign addresses, and states**.**

**6. Monitor Network Activity**

**🡪** iftop

* **Purpose:** Provides real-time network bandwidth usage by interface**.**

sudo yum install iftop -y

sudo iftop 🡪 shows Interface, IP address and Mac address

7.**Check Open Ports**

**🡪**netstat -tuln or ss -tuln

* **Purpose**: Lists all listening ports and their associated services.

**8.Test Port Connectivity**

**🡪**telnet **or** nc (Netcat)

* **Purpose:** Tests if a specific port on a remote server is open and reachable.

9. **DNS Lookup**

**🡪**dig or nslookup

* **Purpose**: Checks DNS resolution and shows detailed information about DNS queries.

10. **Monitor System Logs**

🡪tail -f /var/log/messages or journalctl -f

**Purpose**: Monitors system logs in real-time, useful for spotting issues during troubleshooting.

**SYSTEM MONITERIONG**

**System monitoring** for disk usage can be efficiently managed using commands like df and du in Linux.

**🡪df (Disk Free)**

The df command provides an overview of disk space usage on your filesystem. It shows how much space is used and available on all mounted filesystems.

🡪**df -h** Shows disk usage in a human-readable format (e.g., GB, MB).

🡪**df -T I**ncludes the type of each filesystem in the output.

🡪 **df /path/to/directory** Shows the disk usage for the filesystem containing the specified directory.

**du (Disk Usage)**

The du command provides information about disk space usage for files and directories. It’s useful for finding which directories or files are taking up the most space.

🡪**du** Shows disk usage for the current directory and its subdirectories in default units.

🡪**du -h** Displays disk usage in a human-readable format.

🡪 **du -sh /path/to/directory** Provides a summary of the total disk usage of the specified directory.

🡪**du -ah** Shows disk usage for all files and directories recursively in a human-readable format. Display All Subdirectories

🡪 **du -ah | sort -rh | head -n 10** Lists the top 10 largest files and directories in the current directory.

**Practical Use Cases**

* **Monitoring Disk Usage:** Use df -h to check overall disk space usage and ensure you have sufficient space on your filesystems.
* **Finding Large Files and Directories:** Use du -ah combined with sort to identify large files and directories that may be consuming excessive disk space.
* **Checking Specific Directory Usage:** Use du -sh /path/to/directory to get a summary of disk usage for a particular directory, which helps in managing space effectively.

**Checking system logs**

🡪 **dmesg (Display Message or Driver Message)**

The dmesg command displays messages from the kernel ring buffer. This includes system messages related to hardware, drivers, and kernel events.

🡪**dmesg -T** Converts timestamps to human-readable format.

🡪**dmesg | grep keyword** Searches for specific messages related to a keyword.

🡪 **dmesg --level=err** Displays only messages with error severity.

**journalctl (Query and Display Messages from the Journal)**

journalctl is used to query and display logs from systemd's journal. It provides detailed and structured logging information

🡪**journalctl** Displays all logs recorded by systemd since the last boot.

🡪 **journalctl -r** Shows logs in reverse order, with the most recent logs first.

🡪 **journalctl -u service\_name** Displays logs for a specific systemd service.

**🡪 journalctl -b** Displays logs from the current boot session.

🡪 **journalctl --since "YYYY-MM-DD HH:MM:SS" --until "YYYY-MM-DD HH:MM:SS"** Shows logs within a specified time range.

🡪 **journalctl -k** Displays kernel logs.

**/var/log Directory**

The /var/log directory contains log files for various system services and applications. Logs are often written to specific files based on the service or application.

**Common Log Files:**

1. **System Logs:**
   * /var/log/syslog or /var/log/messages: General system logs.
   * /var/log/boot.log: Boot process logs.
2. **Authentication Logs:**
   * /var/log/auth.log or /var/log/secure: Authentication and security-related logs.
3. **Application Logs:**
   * /var/log/httpd/ or /var/log/apache2/: Apache web server logs.
   * /var/log/nginx/: Nginx web server logs.
   * /var/log/mysql/: MySQL database logs.
4. **Kernel Logs:**
   * /var/log/kern.log: Kernel messages (if enabled).

🡪 **cat /var/log/syslog** Displays the contents of a specific log file.

🡪 **tail -f /var/log/syslog** Continuously displays new log entries as they are added.

🡪 **grep "keyword" /var/log/syslog** Searches for specific entries in the log file.

**Practical Use Cases**

* **Troubleshooting Hardware Issues:** Use dmesg to identify kernel messages related to hardware errors or driver issues.
* **Monitoring System Services:** Use journalctl to view logs from specific systemd services and identify issues with service startup or operation.
* **Checking System Logs for Errors:** Inspect /var/log/syslog or /var/log/messages for general system errors and events.
* **Real-Time Monitoring:** Use tail -f to monitor logs for real-time information, which is useful during troubleshooting.

**🡪vmstat**

**Purpose:**

* Reports virtual memory statistics, including processes, memory, paging, block IO, traps, and CPU activity.

🡪**vmstat 5 10** Displays system statistics every 5 seconds for 10 intervals.

**Practical Use Cases**

* **System Uptime:** Use uptime to check how long your system has been running and to understand the current load averages. This can help in assessing if the system is under high load or if it has been recently rebooted.
* **Memory Usage:** Use free to check memory utilization and ensure that there’s enough free memory available for processes. High memory usage with low free memory may indicate a need for additional resources or optimization.
* **System Performance Over Time:** Use vmstat to monitor system performance trends over time, including memory, CPU, and I/O statistics. This can help diagnose performance issues and optimize system resources.

**Introduction to Network Analysis Tools**

**1.ss (Socket Statictics)**

ss is a utility to investigate sockets on a Linux system. It is a modern replacement for the older **netstat** command, providing more detailed and faster information about network connections.

🡪**ss -t:** List all TCP connections

🡪**ss -u:** List all UDP connections

🡪**ss -l:** Display all listening sockets

🡪**ss -tnlp** Show detailed information

-t: TCP sockets

-n: Show numerical addresses

-l: Show listening sockets

-p: Show process using the socket

**2. tcpdump**

tcpdump is a network packet analyzer. It captures packets from network interfaces and displays them in a human-readable format. It is widely used for network troubleshooting and analysis.\

🡪 **tcpdump -i eth0** Capture packets on a specific interface

🡪 **tcpdump -i eth0 'tcp port 80'** Capture packets with a specific filter (e.g., HTTP traffic)

**🡪 tcpdump -i eth0 -w capture.pcap** Save captured packets to a file

🡪 **tcpdump -r capture.pcap** Read packets from a file

**3. Wireshark**

Wireshark is a powerful graphical network protocol analyzer. It provides detailed analysis of network traffic and is widely used for network troubleshooting, analysis, and development.

**Common Usage:**

* **Capture live network traffic:** Open Wireshark, select the network interface, and start capturing packets.
* **Analyze captured traffic:** Use the graphical interface to filter, inspect, and analyze packet data.
* **Filter packets:** Use display filters to focus on specific types of traffic, such as http, tcp, or ip.addr == 192.168.1.1.

**Typical Analysis Tasks:**

* **Inspect packet details:** Check protocols, payloads, and headers.
* **Analyze performance:** Look for latency, packet loss, or retransmissions.
* **Detect anomalies:** Identify unusual traffic patterns or potential security issues.

**Summary**

* **ss**: Efficiently lists socket statistics and provides detailed connection information.
* **tcpdump**: Captures and displays network packets from the command line.
* **Wireshark**: Provides a comprehensive graphical interface for deep packet inspection and analysis.

🡪 http Display HTTP traffic

🡪 ip.addr == 192.168.1.5 Display packet from a Specific IP address