LINUX OS INTERVIEW QUESTIONS

1. **What is “Filesystem” in linux?**

A **filesystem in Linux** is a method and data structure that an operating system uses to control how data is stored and retrieved on storage devices (like hard drives, SSDs, or USBs).

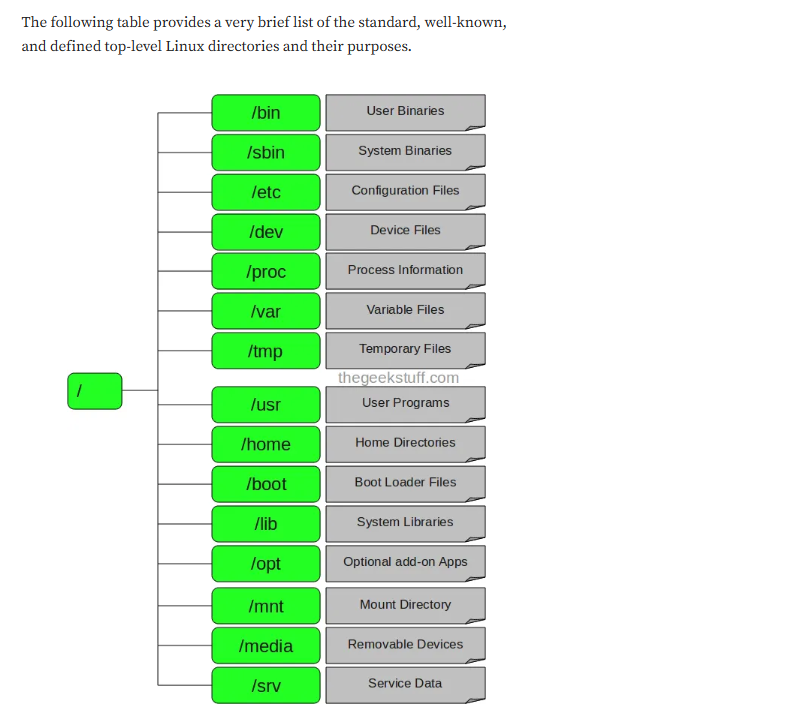
Key Concepts of Filesystem in Linux:

**Files & Directories**

* Everything in Linux is treated as a file (regular files, directories, devices, sockets, pipes).
* Directories are special files that contain references (links) to other files.

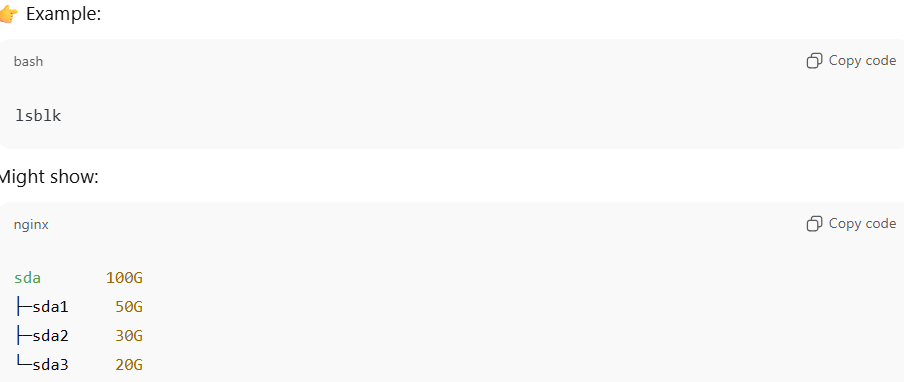
**Hierarchy (Tree Structure)**

* Linux uses a **single-rooted directory tree** starting from / (root directory).



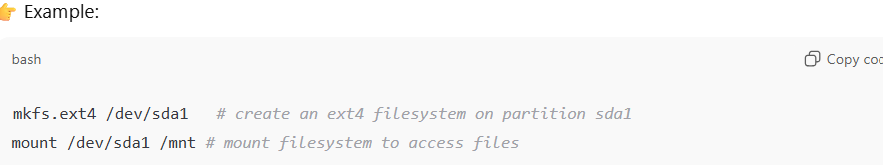
1. **What is the difference between Partition and Filesystem?**
   1. **Partition**

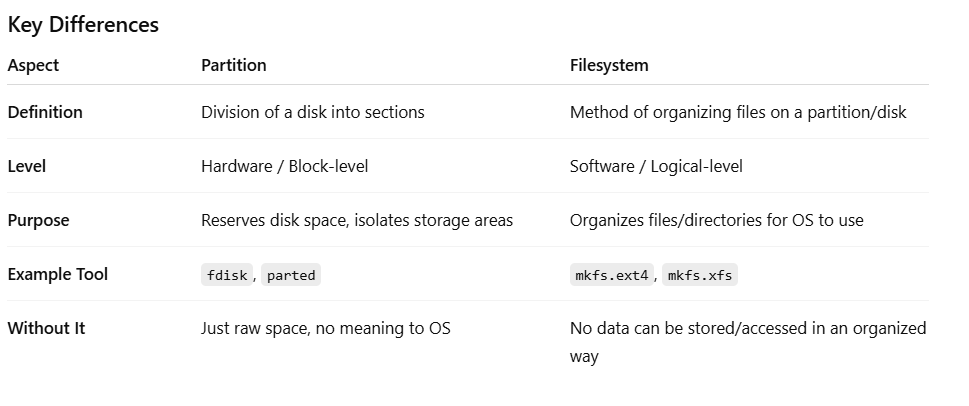
* A **partition** is a division of a physical storage device (like HDD/SSD/NVMe).
* It’s defined at the **block device level**.
* Think of it as carving a disk into slices (e.g., /dev/sda1, /dev/sda2).
* A partition itself only provides a **range of sectors/blocks** — it doesn’t define how data is organized inside.



2.2 A **filesystem** is a data structure (set of rules + metadata) that defines **how files and directories are stored, organized, and accessed** on a partition (or volume).

* It provides the logic to interpret raw blocks into meaningful files (/home/user/file.txt).
* Different filesystems exist with different features:
* **ext4** (common Linux FS)
* **xfs**
* **btrfs**
* **vfat / ntfs** (Windows compatibility)





1. Explain the difference between hard links and soft (symbolic) links.

* **Hard Link:**
* Points directly to the **inode** of the file. Hard links share the same inode number as the original file.
* If the original file is deleted, the data remains accessible as long as at least one hard link to that inode still exists. The file data is only truly removed when the last hard link is deleted.
* Cannot span across filesystems.
* Hard links cannot be created for directories.
* **Soft Link (symlink):**
* A pointer to the **path** of the file. Soft links have their own unique inode number, distinct from the target file or directory.
* If the original is deleted, symlink becomes broken.
* Soft links can point to files or directories located on different file systems.

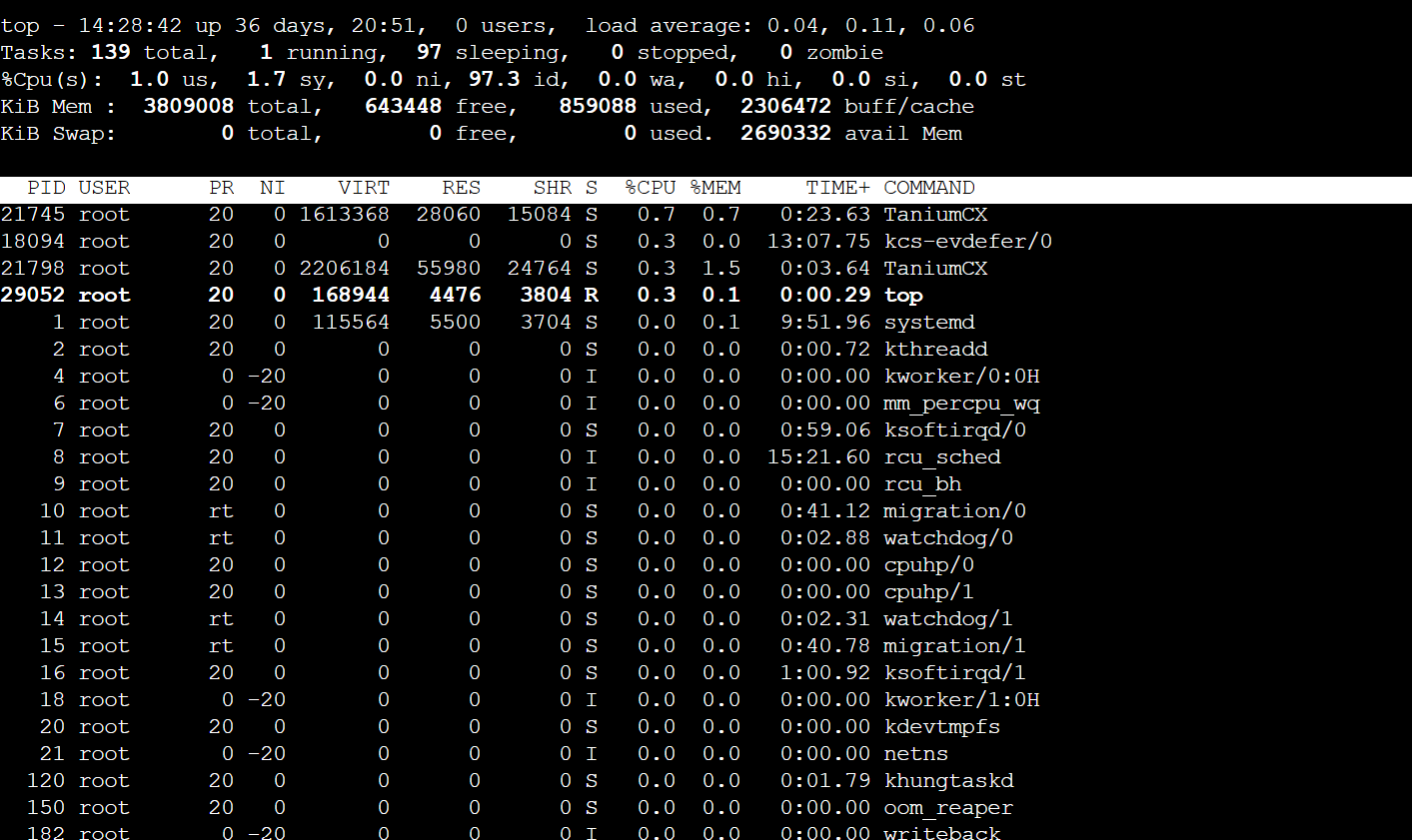
A screenshot of a computer

AI-generated content may be incorrect.

1. What’s the difference between ext4, xfs?

* **ext4**
* **Default** filesystem in most Linux distributions (Ubuntu, Debian, etc.).
* **Stable and reliable** — evolved from ext2/ext3, very mature.
* Good for **general-purpose workloads** (desktops, laptops, smaller servers).
* **Performance**:
* Good for small to medium files.
* Handles many small files efficiently.
* **Features**:
* Journaling (fast recovery after crashes).
* Supports files up to **16 TB**, filesystem size up to **1 EB**.
* Extents (to reduce fragmentation).
* Delayed allocation (improves performance).
* **Limitations**:
* Metadata handling is slower compared to XFS.
* Less efficient for very large filesystems (>100TB) or large concurrent writes.
* **xfs**
* **High-performance filesystem** designed for scalability.
* **Best for**: large servers, databases, big files (media, logs, VMs).
* **Performance**:
* Excels in **parallel I/O** (multi-threaded workloads).
* Great for handling **large files** (e.g., video, database dumps).
* **Features**:
* Metadata journaling (fast crash recovery).
* Supports filesystem sizes up to **8 EB**.
* Dynamic inode allocation (not fixed at format time like ext4).
* Advanced allocation groups → scales well with multiple CPUs.
* **Limitations**:
* **Poor at handling many small files** (higher metadata overhead).
* **Resize**: Can grow online, but shrinking is **not supported**.

1. “TOP” usage for getting server statistics.



* top - 14:30:39 up 36 days, 20:53, 0 users, load average: 0.12, 0.10, 0.06
* Current time, system uptime, logged-in users.
* Load averages (1 min, 5 min, 15 min) → number of processes waiting for CPU.
* Rule of thumb: Load should be ≤ number of CPU cores.
* Tasks
* Total processes: running, sleeping, stopped, zombie.
* **%Cpu(s):**
* us = user space CPU usage
* sy = system/kernel usage
* ni = processes with changed priority (nice)
* id = idle CPU
* wa = I/O wait (waiting for disk/network)
* hi = hardware interrupts
* si = software interrupts
* st = stolen time (VMs only, CPU taken by hypervisor)
* **Mem (Memory usage)**
* total = total RAM
* free = unused RAM
* used = currently used
* buff/cache = used by Linux buffers & cache (can be freed if needed)
* **Swap**
* Swap total, free, used, and available memory.

Process List (lower table)

PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND

**Field Explanation:**

* **PID** → Process ID.
* **USER** → User who owns the process.
* **PR** → Priority (lower = higher priority).
* **NI** → "Nice" value (affects scheduling priority).
* **VIRT** → Virtual memory used (includes code, data, libraries, swap).
* **RES** → Resident memory (RAM actually used, no swap).
* **SHR** → Shared memory with other processes.
* **S** → Process status:
  + R = Running
  + S = Sleeping
  + Z = Zombie
  + T = Stopped
* **%CPU** → CPU usage percentage per process.
* **%MEM** → RAM usage percentage per process.
* **TIME+** → Total CPU time used by process since start.
* **COMMAND** → Process name/command that started it.

1. What is the usage of SWAP space ?

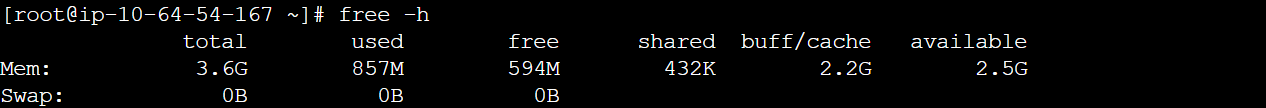
Swap space is a portion of the hard disk (or SSD) that the Linux kernel uses when **RAM is full**. It acts as **virtual memory** by extending the physical memory.

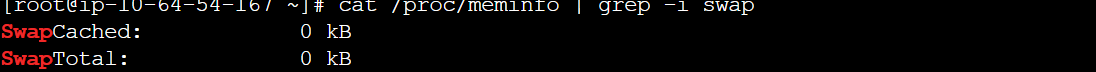
**🔹 Usage of Swap Space**

1. **Extend RAM capacity**
   * When RAM is exhausted, inactive pages (less frequently used data) are moved from RAM to Swap.
   * This frees up RAM for active processes.
2. **Handle memory spikes**
   * Useful for sudden increases in memory usage (e.g., compiling software, running heavy queries).
   * Prevents out-of-memory (OOM) crashes.
3. **System stability**

* Without Swap, if RAM runs out → kernel may invoke **OOM-killer** (kills processes to free memory).
* With Swap, the system degrades gracefully instead of crashing.

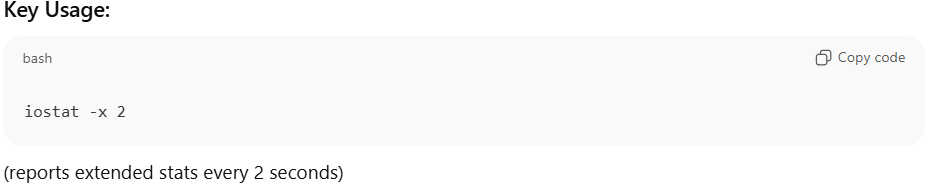
1. how to check current swap usage on a Linux server?





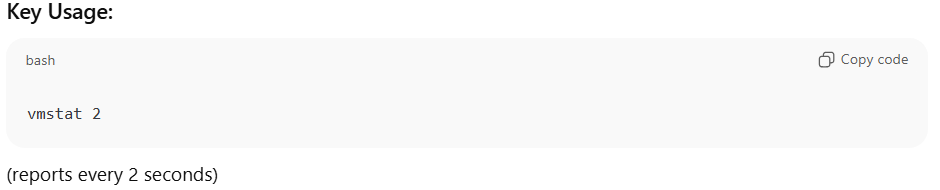
1. What is the difference between “iostat” vs “vmstat”

* IOSTAT
  + Focus: **CPU utilization + I/O statistics (disk devices, partitions, NFS)**.
  + Used to identify **I/O bottlenecks**.



**Important Fields:**

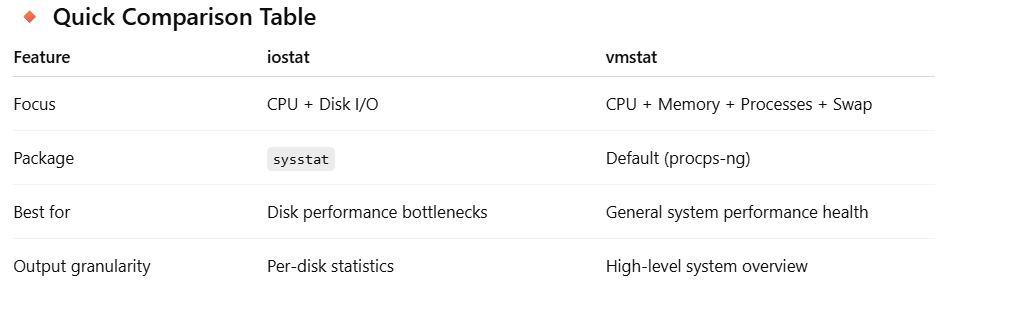
* **%user, %system, %iowait, %idle** → CPU usage breakdown.
* **r/s, w/s** → Reads/writes per second.
* **rkB/s, wkB/s** → Data read/written per second.
* **await** → Average wait time for I/O requests (higher = slower).
* **%util** → How busy the device is (100% = saturated).
* 👉 **Best for:** Checking **disk performance** (e.g., if app is slow because of disk I/O).
* VMSTAT
* Focus: **CPU, memory, swap, process, system activity**.
* Gives a **holistic system view** (not just disks).



**Important Fields:**

* **procs**
  + r = runnable processes (waiting for CPU).
  + b = blocked processes (waiting for I/O).
* **memory**
  + swpd = swap used.
  + free = free memory.
  + buff/cache = memory used for buffers/cache.
* **swap**
  + si/so = swap in/swap out.
* **io**
  + bi/bo = blocks received/sent to block devices.
* **system**
  + in = interrupts.
  + cs = context switches.
* **cpu**
  + us = user time, sy = system time, id = idle, wa = waiting for I/O.

👉 **Best for:** Checking **overall system health** (CPU, memory, swap pressure).



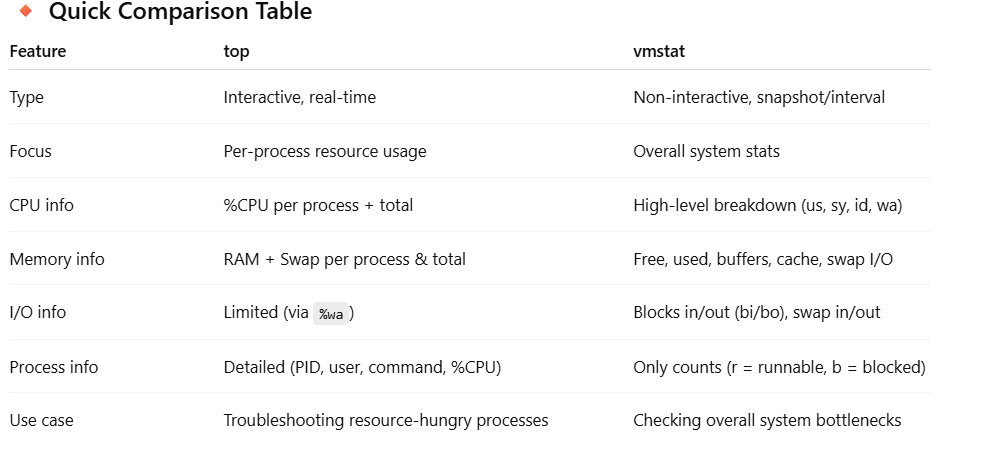
1. what is the difference between top and vmstat?

**🔹 1. top**

* **Interactive, real-time monitoring tool**.
* Shows **system summary + per-process details**.
* Lets you **sort, filter, and kill processes** directly

**🔹 2. vmstat**

* **Non-interactive snapshot tool** (not process-level).
* Reports **CPU, memory, swap, I/O, processes, interrupts** in a concise tabular format.
* Doesn’t show *which* process is consuming resources.



1. What is the difference between nice, renice, and ionice?

* nice: Starts a process with a given scheduling priority. Lower priority → less CPU.
* renice: Changes priority of an already running process.
* ionice: Controls **I/O scheduling priority** (useful for disk-heavy processes).

A close-up of a computer screen

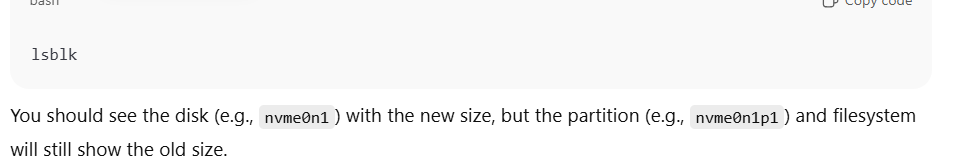
AI-generated content may be incorrect.

1. **How do you recover a system if /etc/fstab has incorrect entries and system fails to boot?**

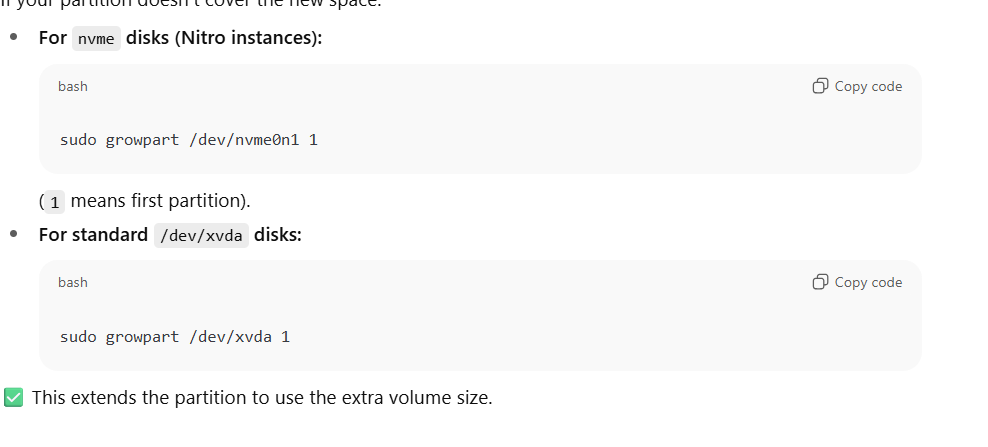
* Boot into **rescue mode** or use a **live CD/ISO**.
* Mount root partition manually:
* mount -o remount,rw /
* Edit /etc/fstab and comment out problematic entries.
* Reboot.

1. How to increase partition/filesystem space on linux after expanding volume size from AWS ?

🔹 Step 1: Verify New Disk Size



🔹 Step 2: Grow the Partition (if needed)



🔹 Step 3: Resize the Filesystem

A screenshot of a computer

AI-generated content may be incorrect.

📌 Note:

* resize2fs works on the partition device (/dev/nvme0n1p1).
* xfs\_growfs needs the **mount point** (e.g., / or /data).

🔹 Step 4: Verify

A white background with black text

AI-generated content may be incorrect.

1. What is DHCP and how does it work?

DHCP stands for **Dynamic Host Configuration Protocol**. It is a network protocol used to **automatically assign IP addresses and other network configuration details** to devices on a network so they can communicate with other devices and access the internet without manual setup.

**Purpose**

* Assigns **IP addresses dynamically** to devices (computers, phones, servers, etc.).
* Provides additional network settings such as:
  + Subnet mask
  + Default gateway
  + DNS servers
  + Lease time (how long the IP is valid)

**How DHCP Works (4-step process called DORA)**

1. **Discover** – The client device sends a broadcast request to find a DHCP server.
2. **Offer** – DHCP server responds with an available IP address and network info.
3. **Request** – The client requests to use the offered IP address.
4. **Acknowledge** – DHCP server confirms and assigns the IP, completing the setup.

**Example**

If you connect your laptop to a Wi-Fi network, the router (acting as a DHCP server) assigns your laptop an IP like 192.168.1.10 automatically, along with the gateway and DNS info.

1. Network Interface Not Getting IP via DHCP

**Scenario:** Server boots but doesn’t get an IP.

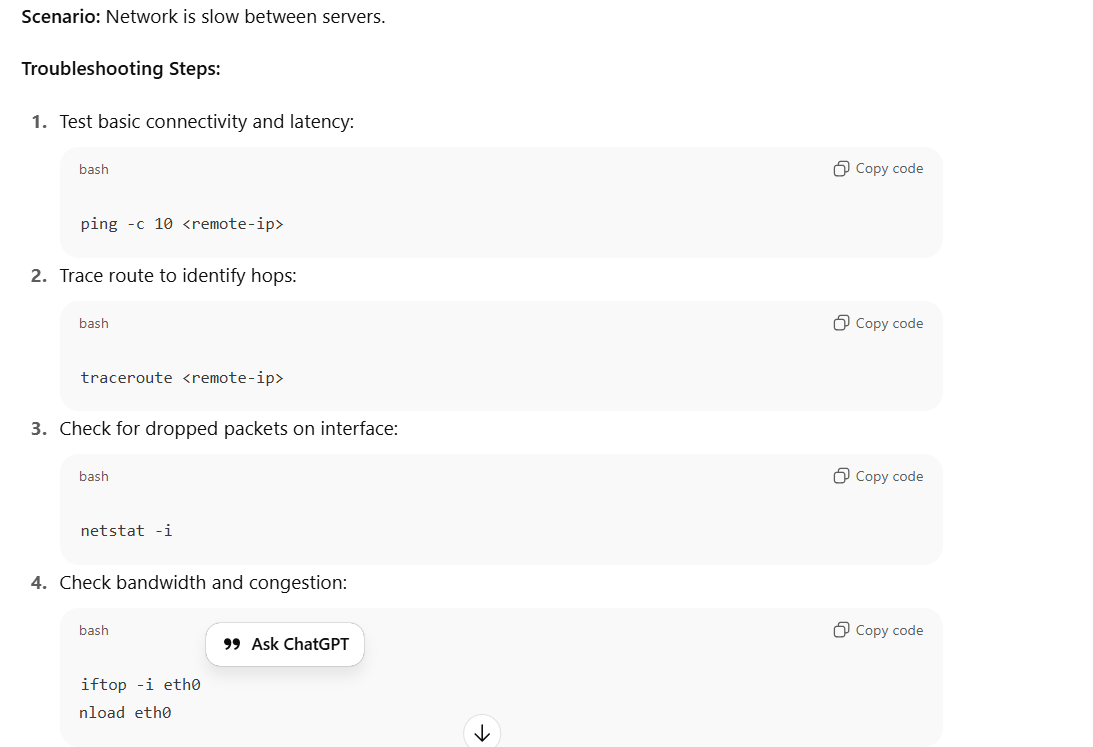
**Troubleshooting Steps:**

1. Check interface status:
2. ip addr show
3. Restart networking / DHCP client:
4. sudo systemctl restart network
5. sudo dhclient -v eth0
6. Check DHCP logs:
7. journalctl -u NetworkManager
8. tail -f /var/log/syslog # Ubuntu
9. Verify connectivity to DHCP server:
10. ping <dhcp-server-ip>

**Explanation:**

* Could be **DHCP server unreachable**, **wrong interface config**, or **MAC filtering**.
* dhclient -v provides debug info about DHCP handshake.

1. High Network Latency or Packet Loss



**Explanation:**

* Latency could be caused by **network congestion, misconfigured MTU, or routing loops**.
* High packet loss often points to **NIC errors, duplex mismatch, or firewall drop**

1. NETSTAT USAGE

netstat is a classic networking utility that reports on network connections, listening ports, routing tables, interface statistics and protocol usage

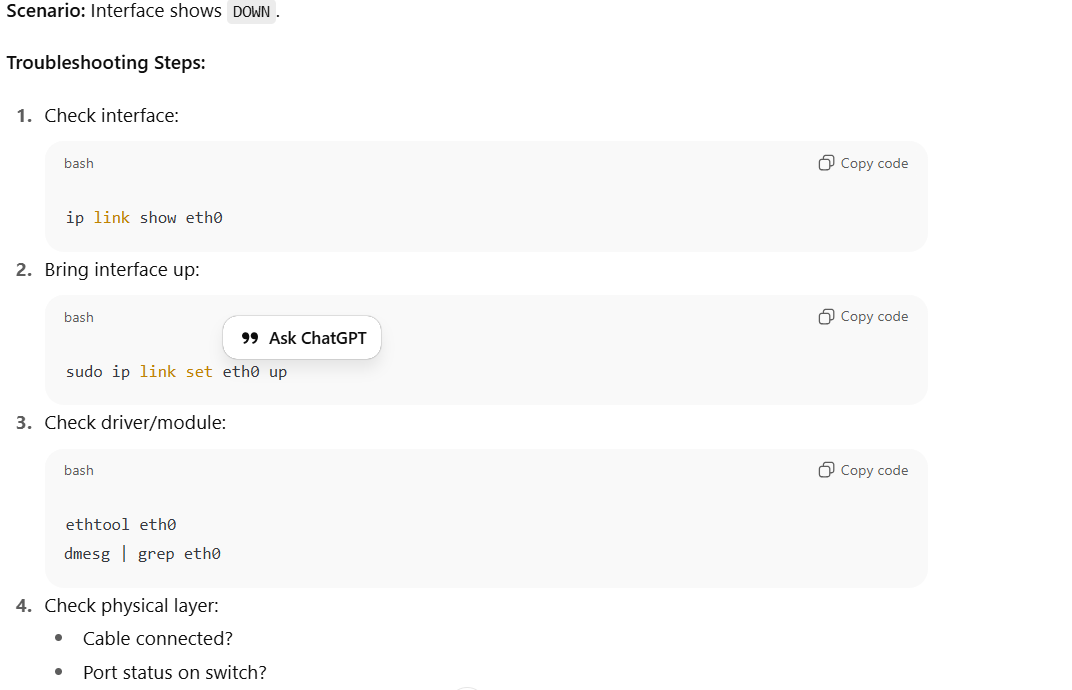
**What netstat does (high level)**

* Lists active network connections (TCP, UDP, RAW, etc.).
* Shows listening sockets (server ports).
* Prints the kernel routing table.
* Displays interface statistics (packets, errors).
* Shows per-protocol statistics (counts of TCP/UDP events).
* Optionally shows PID/program owning each socket (requires root).

**Common options (GNU/Linux netstat)**

* -a, --all  
  Show all sockets (both listening and non-listening).
* -t, --tcp  
  Show TCP sockets only.
* -u, --udp  
  Show UDP sockets only.
* -l, --listening  
  Show only listening sockets.
* -n, --numeric  
  Do not resolve names (show numeric IPs and ports) — faster and clearer.
* -p, --program  
  Show PID/program name for each socket (must be root to see processes you don’t own).
* -r, --route  
  Show kernel routing table (same as route -n).
* -i, --interfaces  
  Show network interfaces and statistics (like ifconfig -a view of stats).
* -s, --statistics  
  Show per-protocol statistics (TCP/UDP/ICMP counts, errors, etc).
* -e, --extend  
  Show extended information (extra columns in various listings).
* -c, --continuous  
  Continuously display updated output (refresh repeatedly).
* -g, --groups  
  Show multicast group membership information.
* --numeric-ports / --numeric-hosts  
  More granular numeric options (if supported).
* --tcp / --udp / --raw  
  Explicit protocol filters.

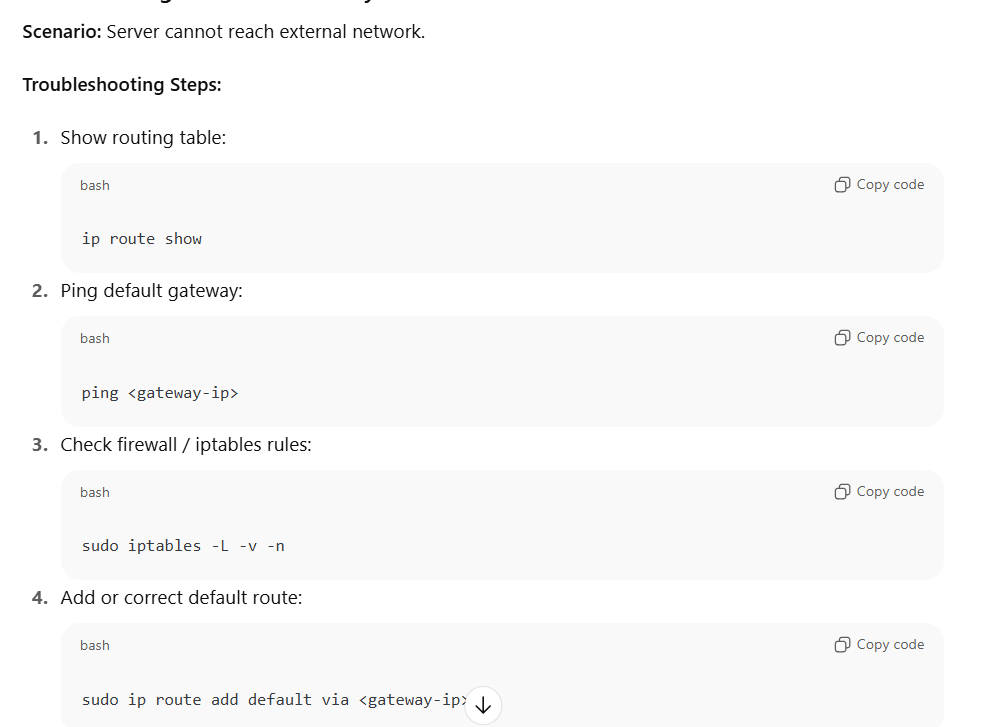
1. Network Interface Down / Not Responding



**Explanation:**

* Could be **driver missing, hardware issue, or config problem**.
* ethtool shows link speed, duplex, and errors.

1. Routing / Default Gateway Issues

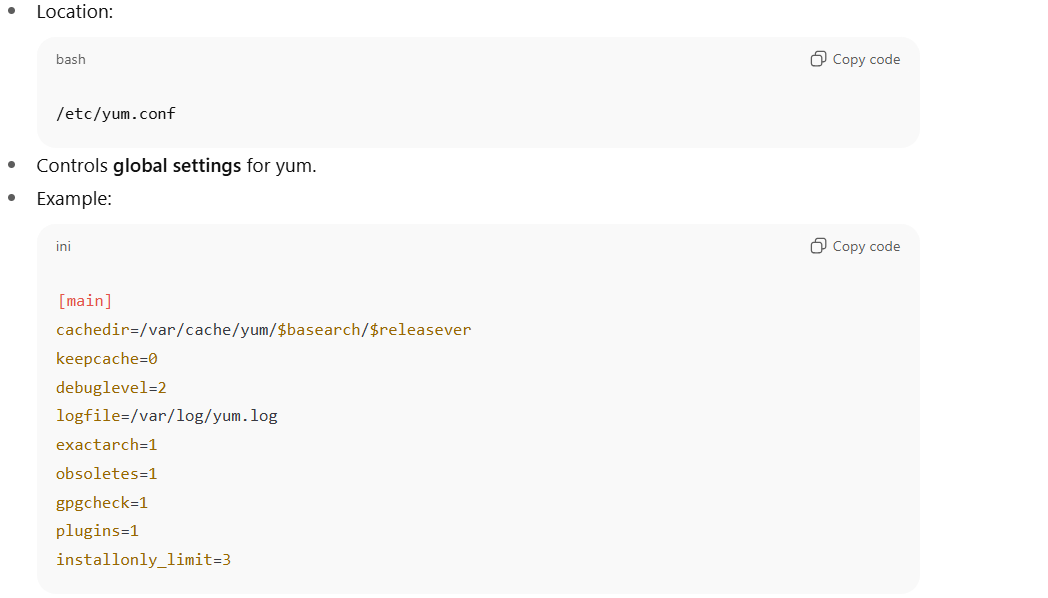


**Explanation:**

* Missing or wrong default gateway causes **all external traffic to fail**.
* Often seen after network reconfigurations or cloud instance migration.

1. How YUM -package manager is configured?

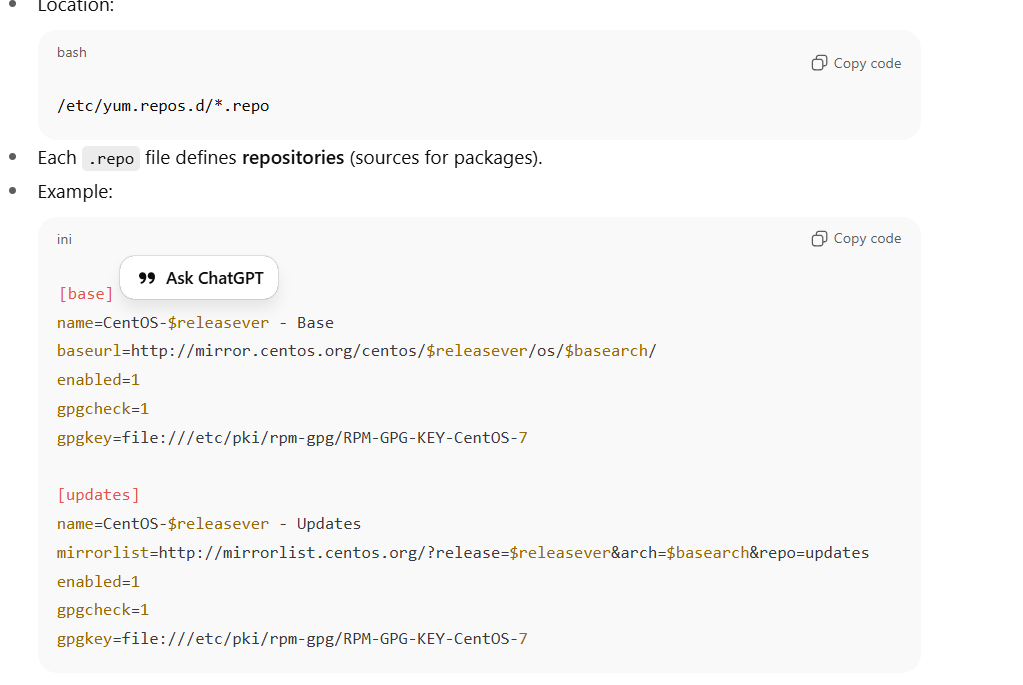
🔹 1. YUM Main Configuration File



**Key settings:**

* cachedir → Where yum stores cache.
* keepcache → Whether to keep downloaded packages.
* gpgcheck → Whether to verify package signatures.
* plugins → Enable/disable yum plugins.

🔹 2. YUM Repository Configuration



**Sections & fields:**

* [base], [updates] → repo IDs.
* name → descriptive name.
* baseurl → direct URL of repo.
* mirrorlist → dynamic mirror URL.
* enabled=1 → turn repo on.
* gpgcheck=1 → enforce signature verification.
* gpgkey → location of public key.

🔹 3. Yum Variables

Yum expands variables in repo configs:

* $releasever → current OS version.
* $basearch → architecture (x86\_64, aarch64, etc).
* $uuid, $infra, etc.

🔹 4. Yum Plugins

A white background with black text

AI-generated content may be incorrect.

🔹 5. Verification

A white box with black text

AI-generated content may be incorrect.

1. Difference between RPM and YUM.

**🔹 1. RPM (Red Hat Package Manager)**

* **Low-level package manager**
* Works directly with .rpm files
* Does not handle dependencies automatically

A white background with black text

AI-generated content may be incorrect.

**🔹 2. YUM (Yellowdog Updater, Modified)**

* **High-level package manager** (built on top of RPM)
* Resolves dependencies automatically by pulling packages from repositories
* Uses config files in /etc/yum.repos.d/\*.repo



1. Why DNS is required ?

**1.Human-Friendly Names vs. Machine IPs**

* **Computers talk using IP addresses (e.g., 142.250.72.14).**
* **Humans prefer names (e.g., google.com).**
* **DNS bridges this gap by translating names to IPs automatically.**

**2. Dynamic IPs and Scalability**

* **Servers often change IPs (cloud auto-scaling, load balancers, migrations).**
* **If users had to remember raw IPs, every change would break things.**
* **With DNS, you just update the record — users still use the same domain name.**

1. **Load Balancing and High Availability**
   * A domain (e.g., example.com) can map to **multiple IPs**.
   * DNS can distribute traffic across multiple servers globally.
   * Helps with redundancy, performance (CDN, GSLB).
2. **Service Discovery**

* DNS provides **different record types** (MX for email, SRV for services, TXT for verification).
* Without DNS, every service would need hardcoded IPs.

1. **Decoupling Users from Infrastructure**

* If a website changes hosting provider or cloud region, you only update the DNS record.
* Users don’t need to know or care — they keep using the same name.

1. Difference between AD and DNS?

**🔹 1. What is Active Directory (AD)?**

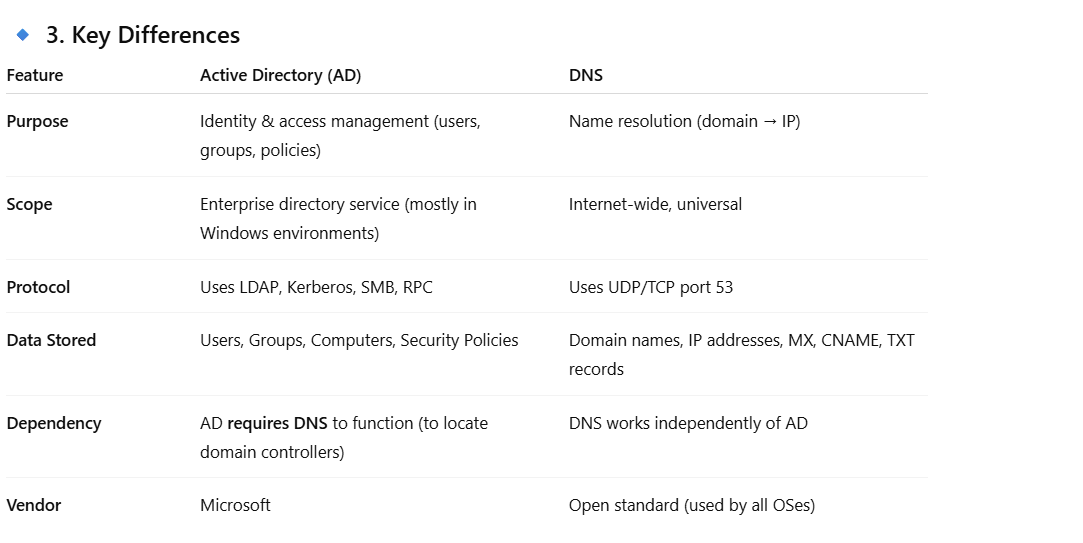
* **Directory service** developed by Microsoft.
* Runs mainly on **Windows Server**.
* Stores and manages **objects** in a network:
  + Users, Groups, Computers, Printers, Applications, Policies.
* Provides **authentication and authorization** (via Kerberos & LDAP).
* Centralizes management of users and resources in a domain/network.

👉 Example:  
When you log in to your company’s Windows laptop, AD checks **“Is this user valid? What permissions does he have?”**

**🔹 2. What is DNS?**

* **Domain Name System** (open standard, not Microsoft-only).
* Translates **domain names into IP addresses**.
* Hierarchical and distributed (Root → TLD → Authoritative).
* Used for **network communication**, not authentication.

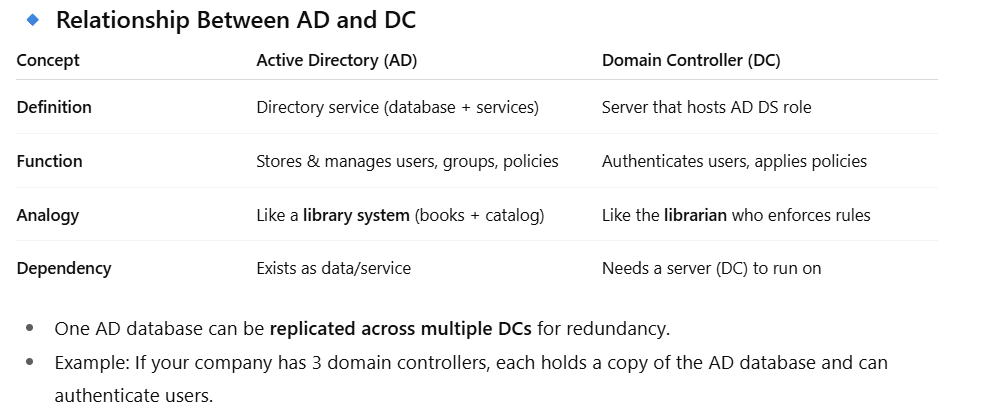
👉 Example:  
When you type intranet.company.com, DNS resolves it to the server’s IP before you connect.



**🔹 4. Relationship Between AD and DNS**

* AD is **built on top of DNS**.
* When you join a computer to a Windows domain:
  + The computer queries DNS to find the **Domain Controller (DC)**.
* AD uses **special DNS records** (SRV records like \_ldap.\_tcp.dc.\_msdcs.domain.com) to locate authentication services.

👉 Without DNS, Active Directory **cannot function properly**, because clients wouldn’t know where to find the domain controllers.



1. Difference Between LDAP and Kerberos?

* **LDAP (Lightweight Directory Access Protocol)** is a protocol used to **query and manage directory services** (like Active Directory, OpenLDAP).

👉 Think of LDAP as a **way to talk to a phonebook-like database** of users, groups, and policies.

* **Kerberos** is a **network authentication protocol** based on secret keys and tickets.
* It avoids sending plain passwords over the network.
* Very secure and widely used (especially in Windows AD).

A screenshot of a computer

AI-generated content may be incorrect.