QUESTIONS BANK

1. What is a server, in simple terms?
In simple terms, a server is a powerful computer that provides a service to other computers on a network.
It "Serves" Data: Think of it like a waiter in a restaurant. You (the customer) ask for a specific dish (data or service), and the waiter (the server) brings it to you.
Always On: It's designed to be running constantly, 24/7, so that it's always available to provide its service.
Central Location: It holds resources—like websites, files, or applications—that many other computers, called "clients," need to access.

2. What is the difference between a server and a regular desktop computer?
The main differences are in their purpose, hardware, and operating system.

Feature	Server	Regular Desktop Computer
1. Purpose & Role	Provides services to many other computers (clients) over a network.	Used by one person for tasks like browsing, gaming, and office work.
2. Hardware	More powerful: multiple processors, vast amounts of RAM, and many large hard drives (often in a RAID) for reliability and speed.	Standard, consumer-grade components designed for cost-effectiveness and single-user performance.
3. Reliability & OS	Built with redundant parts (like power supplies) to run 24/7 without failing. Runs a specialized server operating system (e.g., Windows Server, Linux).	Not designed for constant, critical operation. Runs a consumer OS (e.g., Windows 11, macOS).

3. What is the client-server model?
The client-server model is the basic structure of how most networks and internet services work.
1. Two Main Roles: It involves two types of computers:
The Server: The central computer that stores data and provides services (e.g., a web server hosting a website).
The Client: The computer that accesses the service (e.g., your laptop or phone using a browser to view that website).
2. Request-Response Cycle: The model works on a simple cycle: the client sends a request, and the server responds to that request.
3. Centralized Management: This model allows for easy management, security, and backup of data because everything is stored centrally on the server.

4. Can any computer be a server?
Yes, technically any computer can act as a server, but with major limitations.
1. Software: You can install server software (like a web server program) on a regular desktop or laptop.
2. Practical Limitations:

Performance: A regular computer isn't powerful enough to handle requests from many users at once. It would become very slow or crash.

Reliability: Desktops aren't built to run 24/7 and are more likely to fail, making the service unreliable.

Operating System: While possible, consumer operating systems aren't optimized for handling multiple, simultaneous network requests efficiently.

5. Explain some common types of servers?

Servers are specialized based on the service they provide. Here are five common types:

1. Web Server: Hosts websites and delivers web pages to your browser. (e.g., Apache, Nginx).
2. File Server: A central storage location for files that users on a network can access and share.
3. Mail Server: Handles the sending, receiving, and storing of email messages. (e.g., Microsoft Exchange, Gmail's servers).
4. Database Server: Provides database services to other computers, storing and managing large amounts of structured data. (e.g., MySQL, Microsoft SQL Server).
5. Game Server: The powerful computer that runs the online world of a multiplayer video game, allowing all players to interact in the same environment.

6. What is the difference between a physical server and a virtual server?

The difference lies in the form the server takes: a tangible machine versus a software-based simulation.

Feature	Physical Server	Virtual Server
1. What it is	A single, tangible, physical computer dedicated to being a server.	A software-based "simulation" of a server, running inside a physical server.
2. Relationship to Hardware	One-to-One: One physical machine runs one server operating system.	Many-to-One: One powerful physical server can host multiple, independent virtual servers.
3. Efficiency & Flexibility	Often underutilized, as it's dedicated to one task. inflexible to change.	Highly efficient, sharing hardware resources. Very flexible; new virtual servers can be created in minutes.

7. What is a "blade" server?

A blade server is a compact, modular server computer designed to be housed in a large enclosure with many other blades.

1. Modular Design: Unlike a bulky tower or rack server, a blade server is a thin, stripped-down circuit board ("blade") containing processors, memory, and network controllers.
2. The Chassis: Multiple blade servers slide into a shared enclosure called a chassis. This chassis provides the power, cooling, networking, and management for all the blades inside it.
3. Key Benefits: This design is highly space-efficient and energy-efficient, making it ideal for large data centers where many servers are needed in a small space.

8. What is an IP address and a port in the context of a server?

1. IP Address: This is the street address of the building. It uniquely identifies the server on the network (e.g., `192.168.1.5`). It tells your computer where to find the server.

2. Port: This is the specific apartment or office number inside the building. It's a number (like 80, 443, 21) that identifies the specific service or application you want to talk to on that server.
3. Working Together: The combination of an IP address and a port (e.g., `192.168.1.5:80`) ensures that your request is delivered to the correct service (e.g., the web server) on the correct computer.

9. What does "hosting" mean?
In simple terms, hosting means storing and maintaining something on a server to make it accessible to others.
1. The Service: A company (a "host") owns powerful, always-on servers in a data center.
2. The Offering: They rent out space, power, and resources on these servers to individuals or businesses who want to make their website, application, or files available on the internet.
3. The Result: Instead of you having to buy and manage your own server, you pay a hosting provider to do it for you, ensuring your site is online and available 24/7.

10. What is the difference between on-premise and cloud servers?
The core difference is who owns the hardware and where it is located.

Feature	On-Premise Server	Cloud Server
1. Location & Ownership	Located on your own property (e.g., in a back office). You buy and own the physical hardware.	Located in a provider's data center (e.g., Amazon AWS, Microsoft Azure). You rent the virtual server resources.
2. Cost & Maintenance	High upfront cost to buy the server. You are responsible for all maintenance, repairs, and security.	Pay-as-you-go model (like a utility bill). The cloud provider handles all maintenance and security of the physical hardware.
3. Scalability	Inflexible. To get more power, you must buy and install new physical hardware, which takes time.	Highly Scalable. You can increase or decrease server power (CPU, RAM) almost instantly with a few clicks.

11. What is a "rack" and a "data center"?
Data Center: This is the entire library building. It's a specialized facility designed to house many computer systems. It provides power, cooling, physical security, and high-speed internet connections.
Server Rack: This is a bookshelf within the library. It's a metal frame, typically 19 inches wide, designed to hold multiple pieces of hardware in standardized slots.
The Relationship: Inside a data center, you will find rows and rows of server racks. Each rack is filled with individual servers, switches, and other equipment, all neatly organized.

12. What is a reverse proxy server?
A reverse proxy is a server that sits in front of one or more web servers, acting as a gateway and shield.
1. Traffic Interception: It receives all requests from clients on the internet.

2. Request Forwarding: It forwards these requests to the appropriate backend server (which holds the actual website or application).
3. Key Benefits:

Load Balancing: Distributes incoming traffic across multiple servers to prevent any single one from being overwhelmed.

Security: Hides the identities and characteristics of the backend servers, protecting them from direct attack.

Caching: Can store (cache) copies of frequently requested content, speeding up delivery to users.

13. What is the difference between scaling up and scaling out?
This is about how you add power to your server infrastructure.

Scaling Type	Also Known As	How It Works	Simple Analogy
Scaling Up	Vertical Scaling	Adding more power (CPU, RAM) to an existing single server.	Replacing your car's 4-cylinder engine with a more powerful 8-cylinder engine.
Scaling Out	Horizontal Scaling	Adding more servers to a pool and distributing the load among them.	Adding more cars to a train to carry more passengers, rather than making one car gigantic.

Key Difference: Scaling up has a hard limit (you can only make one server so powerful), while scaling out is more flexible and fault-tolerant (if one server fails, the others can take over).

14. What is meant by "serverless" computing?
Serverless computing is a cloud model where the developer writes and deploys code without ever having to think about the underlying servers.
1. No Server Management: The cloud provider (e.g., AWS, Azure) automatically manages the server infrastructure—provisioning, scaling, and maintenance.
2. Event-Driven & Scalable: Code is executed in response to specific events (e.g., an image upload, a click on a website). It scales automatically and perfectly with the number of requests.
3. Pay-Per-Use: You only pay for the computation time your code actually uses while it runs, not for idle server time.

15. What is the difference between a stateful and a stateless server?
The difference is whether the server "remembers" you between requests.

Feature	Stateful Server	Stateless Server
1. Session Data	Remembers client-specific data (state) from one request to the next.	Does not remember any client data between requests.

Feature	Stateful Server	Stateless Server
2. How it Works	Stores session information (like your shopping cart) in its own memory.	Each request must contain all the information needed to process it (like a "remember me" token).
3. Example	A traditional FTP server or a server managing a live chat session.	A REST API or a standard web server that uses cookies/tokens for persistence.
4. Scalability	Harder to scale because a user must be directed back to the same server that has their session data.	Easier to scale. Any server in a pool can handle any request, as they don't store local state.

16. What is a "headless" server?

A headless server is a computer that operates without a direct graphical user interface (GUI), monitor, keyboard, or mouse.

1. No Peripherals: It is administered and controlled entirely remotely over a network connection, typically using command-line tools.
2. Efficiency: By eliminating the GUI, it uses less processing power, memory, and storage, dedicating all its resources to its core server tasks. This makes it more efficient and secure.
3. Common Use: This is the standard configuration for almost all servers in data centers and the cloud.

17. What is server clustering and load balancing?

These are two related techniques for creating highly available and scalable services.

Server Clustering: This is when a group of servers (nodes) work together as a single system. The goal is to provide high availability (minimizing downtime) and, in some cases, increased processing power. If one server in the cluster fails, another one automatically takes over its workload.

Load Balancing: This is the technology that distributes network traffic across the servers in a cluster. It sits in front of a group of servers and acts as a "traffic cop," directing incoming client requests to the server that is best able to handle it at that moment.

Relationship: Load balancing is often a key component that makes a server cluster effective. The cluster provides the redundant servers, and the load balancer efficiently uses them.

18. What is the role of an API in server communication?

An API (Application Programming Interface) is a set of defined rules and protocols that allows different software applications to talk to each other.

In server communication, its role is to be a standardized "waiter" or "contract" between a client and a server.

1. Structured Requests: It defines exactly how a client (e.g., a mobile app) must ask for data or a service from a server.
2. Standardized Data Format: It specifies the format (like JSON or XML) in which the server will respond with the data.
3. Abstraction: It allows the client to use the server's functionality without needing to know the internal details of how the server works or how its data is stored.

Example: When a weather app shows you the forecast, it uses a weather service's API to request the data. The app doesn't need to know where the server gets its data; it just follows the API's rules to ask for it and display it.

19. What is containerization (e.g., Docker) and how is it different from a virtual server?

Feature	Virtual Server (VM)	Container (e.g., Docker)
What it virtualizes	An entire machine, including the guest operating system (OS).	Just the application and its dependencies.
Architecture	Each VM runs its own full OS on top of a hypervisor.	All containers on a host share the same underlying host OS kernel.
Key Difference	Heavyweight: More resource-intensive due to multiple OS copies.	Lightweight: Much more efficient, allowing you to run many more instances on the same hardware.
Boot Time	Slow (minutes) as it must boot a full OS.	Near-instantaneous (seconds).

20. What is "server hardening"?
Server hardening is the process of securing a server by reducing its surface area for attack.
1. Minimization: This involves removing any unnecessary software, user accounts, and services. If a program isn't running, it can't be hacked.
2. Configuration: Changing default settings to be more secure, such as changing default passwords, enforcing strong password policies, and configuring firewalls to block all ports except the essential ones.
3. Patching & Updates: Applying the latest security patches to the operating system and all installed software to fix known vulnerabilities.
The goal of server hardening is to create a server that performs only its intended function and is protected against as many known threats as possible.

21. What might cause a server to go down?
1. Hardware Failure: Physical component failure (e.g., hard drive, power supply, memory, CPU, or network interface).
2. Software/System Issues: Operating system crashes, buggy application updates, software conflicts, or resource exhaustion (CPU, RAM, disk space).
3. External Factors: Power outages, network connectivity loss, distributed denial-of-service (DDoS) attacks, or natural disasters affecting the data center.

22. What is server monitoring and why is it important?
1. Definition: It is the continuous observation of a server's performance, health, and availability using specialized tools to track metrics like CPU load, memory usage, disk I/O, network traffic, and application status.
2. Proactive Issue Resolution: It enables the early detection of problems (e.g., rising resource usage, service failures) before they cause major downtime, allowing for proactive intervention.
3. Ensuring Reliability & Planning: It ensures service level agreements (SLAs) are met, provides data for capacity planning and security auditing, and helps maintain overall system reliability and user satisfaction.

23. What is the difference between a dedicated server and a shared hosting plan?
    1. Resources & Performance: A dedicated server provides exclusive use of all physical resources (CPU, RAM, storage) to one client, ensuring consistent high performance. Shared hosting partitions a single server's resources among many websites, leading to variable performance that can be affected by "noisy neighbors."
    2. Control & Configuration: Dedicated servers offer full root/administrative access, allowing complete control over the OS, software, and security settings. Shared hosting provides limited control, typically via a control panel (e.g., cPanel), with configurations managed by the host.
    3. Cost & Use Case: Dedicated servers are significantly more expensive and are used for high-traffic websites, critical applications, or unique software needs. Shared hosting is low-cost and suitable for small websites, blogs, or businesses with minimal technical requirements.

24. What is meant by "edge server" or "edge computing"?
    1. Definition: It refers to deploying computing resources (servers, data processing) geographically closer to the source of data or end-users, at the "edge" of the network, rather than in a centralized data center.
    2. Primary Goal: To reduce latency (delay) and bandwidth usage by processing data locally. This is critical for real-time applications like video streaming, IoT, online gaming, and autonomous vehicles.
    3. Key Benefit: It enhances performance and user experience for geographically distributed users and enables faster, more efficient processing for applications requiring immediate response or generating large data volumes.

25. How does a server typically handle multiple simultaneous connections?
    1. Concurrency Models: It uses techniques like multithreading (handling different connections on separate threads within a process) or multiprocessing (using multiple processes, e.g., worker processes in web servers like Nginx/Apache).
    2. Event-Driven Architecture: Employed by servers like Nginx and Node.js, this model uses a single-threaded event loop to efficiently manage thousands of connections by reacting to events (e.g., incoming data) without blocking, instead of dedicating a thread to each connection.
    3. Underlying OS Support: The operating system's kernel provides the socket API and uses mechanisms like I/O multiplexing (e.g., select, poll, epoll on Linux) to allow the server to monitor and manage multiple network sockets simultaneously from a single or a limited number of threads.

26. How do you approach server patch management and updates securely?
    1. Staged Testing & Deployment: I follow a phased approach, first applying patches to a non-production test environment to identify conflicts or issues. Then, after validation, patches are rolled out to production servers in a staged manner, minimizing widespread risk.
    2. Scheduled Maintenance Windows & Categorization: I implement updates during pre-approved, scheduled maintenance windows to minimize user impact. Patches are categorized by criticality (e.g., security, critical, non-critical) and prioritized accordingly, with security patches expedited.
    3. Pre-Update Preparation & Rollback Plan: Before applying any update, I verify the integrity of backups and document the current system state. Crucially, I always have a clear and tested rollback plan to quickly revert changes in case of a failure or unforeseen consequences.

27. What role does documentation play in your server management practices?

1. Operational Guide & Consistency: It acts as a single source of truth for configurations, procedures, and network details, ensuring tasks are performed correctly and consistently by anyone on the team, reducing human error.
2. Efficient Troubleshooting & Onboarding: Comprehensive documentation (runbooks, network diagrams, change logs) is critical for rapid problem diagnosis and resolution during incidents. It also significantly speeds up the onboarding process for new team members.
3. Compliance & Knowledge Preservation: It provides an audit trail for compliance requirements (e.g., SOX, PCI-DSS) and ensures that institutional knowledge is retained and not lost when personnel change, protecting against operational risks.

28. How do you monitor server health and performance metrics?
    1. Deployment of Centralized Monitoring Tools: I use agent-based or agentless monitoring tools (e.g., Zabbix, Nagios, Prometheus/Grafana, Datadog) to centrally collect metrics from all servers, providing a unified view of system health.
    2. Tracking of Key Performance Indicators (KPIs): I monitor core KPIs including CPU, memory, and disk utilization, network I/O and latency, disk I/O wait times, and critical application/service availability to identify bottlenecks and predict capacity needs.
    3. Proactive Alerting & Visualization: I configure intelligent alert thresholds (not just static limits) to generate notifications (via email, SMS, Slack) for anomalies or impending issues. Data is presented on customized dashboards for real-time visibility and historical trend analysis.

29. What are the first three steps you would take to secure a freshly installed Windows Server?
    1. Immediate Patching & Updates: I would first disable or restrict network access, then run Windows Update to install all critical security patches and service packs before connecting it fully to the production network.
    2. Harden the Local Security Policy: I would disable the default Administrator account, rename the built-in administrator account, and create a new, complex-named administrative account. I would then enforce a strong password policy and configure the local firewall to block all inbound traffic except explicitly required management ports.
    3. Disable Unnecessary Services & Features: I would review the Server Manager roles and features, and disable or uninstall any services not required for the server's specific function (e.g., Print Spooler on a web server, Internet Explorer Enhanced Security Configuration for admins). I would also disable or restrict the Guest account.

30. If a server cannot connect to the internet, what is the first physical thing you should check?
    1. The Network Cable and Physical Link: The first physical check is to inspect the Ethernet cable connections at both the server's network interface card (NIC) and the switch/router. I would look for a secure, fully-seated connection and verify that the link lights on the NIC and switch port are illuminated correctly (typically a solid/green light for link and a blinking/amber light for activity).
    2. Purpose: This step rules out the most common and simplest physical layer failures—a loose, unplugged, or damaged cable—before proceeding to more complex logical troubleshooting like checking IP configurations, switch port settings, or firewall rules.

31. What is the practical use of the ping command?
    1. Basic Connectivity Testing: Its primary practical use is to verify network-level connectivity between two hosts. It sends an ICMP echo request to a target IP address or hostname to test if it is reachable and responsive on the network.

2. Troubleshooting & Latency Measurement: It is a fundamental first step in network troubleshooting to isolate where a connection problem exists. It also provides round-trip time (latency) statistics, helping gauge network performance and identify delays.
3. DNS Verification & Simple Monitoring: The command can help verify if DNS resolution is working by pinging a hostname. It is also used for basic availability monitoring in scripts to check if a critical server or gateway is up.

32. If a server is running slowly, which basic tool would you use first to identify which process is consuming the most resources?
1. Linux/Unix Systems: The first basic tool is the top command (or its more user-friendly variant, htop). It provides a real-time, dynamic view of all running processes, sorted by default by CPU usage.
2. Windows Systems: The equivalent first tool is the Task Manager (taskmgr.exe), specifically the "Processes" or "Details" tab, which shows CPU, memory, disk, and network usage per process.
3. Purpose: Both tools allow you to quickly identify the specific process (PID) consuming excessive CPU or memory, which is the critical first step in diagnosing performance bottlenecks before moving to more detailed profiling tools.

33. Why is it important to check server logs regularly?
1. Proactive Issue Detection & Security: Regular log review enables the early detection of errors, warnings, and performance degradation patterns before they cause major failures. It is also critical for identifying security breaches or suspicious activity (e.g., failed login attempts, unauthorized access).
2. Audit Trail & Compliance: Logs provide an immutable record of all system events, user actions, and configuration changes, which is essential for forensic analysis during incidents and for meeting regulatory compliance requirements (e.g., SOX, HIPAA, PCI-DSS).
3. Troubleshooting & Performance Analysis: When a problem occurs, logs are the primary source of diagnostic information to understand the root cause. They also contain data useful for capacity planning and performance trend analysis.

34. What does the term "uptime" mean in a server context, and why is high uptime critical for organizations?
1. Definition: In a server context, uptime refers to the amount or percentage of time a server (or service) is operational, accessible, and performing its intended function. It is the opposite of "downtime."
2. Business Continuity & Revenue: High uptime is critical for ensuring continuous business operations. For e-commerce, SaaS, or online services, downtime directly translates to lost revenue, lost productivity, and missed transactions.
3. Reputation & User Trust: Consistent availability builds customer trust and brand reputation. Frequent or prolonged downtime leads to user frustration, erosion of customer confidence, and potential loss of clients to competitors.

35. What are the primary responsibilities of a Server Operating System?
1. Hardware Abstraction & Resource Management: Its core responsibility is to act as an intermediary between software applications and the physical hardware, efficiently managing and allocating CPU, memory, disk I/O, and network resources among all processes and users.
2. Providing Core Server Services: It provides essential network and application services such as a web server (IIS, Apache), file sharing (SMB, NFS), directory services (Active Directory, OpenLDAP), and database hosting platforms.
3. Security & Access Control: It is responsible for system security, including user authentication, authorization, file permissions, and network-level firewalling to ensure data integrity and controlled access in a multi-user environment.

36. Name one practical reason to choose a Linux server over a Windows server for a specific task.

1. Reason: To host a high-performance, low-cost web or database server requiring maximum stability and resource efficiency.
2. Justification: Linux servers are renowned for their stability, security, and efficient handling of high concurrent connections (e.g., using Nginx/PHP/MySQL stack). They typically have lower licensing costs and can run for extended periods without reboots, making them ideal for cost-sensitive, high-uptime web infrastructure.
3. (Alternative reason: For a task requiring deep customization, scripting automation, or access to a vast repository of free, open-source software.)

37. What is the primary goal of using RAID (Redundant Array of Independent Disks) in a server?

1. Primary Goal: The primary goal is to improve data reliability and availability by combining multiple physical disk drives into a single logical unit to provide data redundancy (fault tolerance) and/or increased performance.
2. Key Benefit - Redundancy: In common RAID levels like 1, 5, 6, or 10, data is written across disks so that if one (or more) disk fails, no data is lost and the system can continue operating, often while the failed disk is replaced.
3. Key Benefit - Performance: Some RAID levels (like 0 or 10) also provide improved read/write speeds by striping data across multiple disks, allowing for parallel I/O operations.

38. Name three hardware components that differentiate a server from a standard desktop computer.

1. Multiple Processors / High Core Count: Servers often have multiple physical CPUs (sockets) or CPUs with a very high number of cores to handle parallel processing for numerous simultaneous requests.
2. Error-Correcting Code (ECC) Memory: Server's use ECC RAM, which can detect and correct common types of internal data corruption, preventing crashes and data errors that are critical in a 24/7 environment.
3. Redundant Power Supplies (PSUs): Servers are typically equipped with dual or more hot-swappable power supplies connected to separate power sources to ensure continuous operation if one PSU or power feed fails.
   (Other examples: Hardware RAID controllers, hot-swappable drive bays, out-of-band management (ILO/iDRAC), and more robust cooling.)

39. In a server environment, why would you assign a static IP address to a server instead of a dynamic one?

1. Consistent Network Identity: A static IP ensures the server has a permanent, predictable network address. This is essential so that clients, applications, and DNS records can reliably find and connect to it without the address changing.
2. Prerequisite for Hosting Services: Critical network services like DNS servers, domain controllers, web servers, and database servers must have unchanging IPs for their roles to function correctly and for firewall rules to be applied accurately.
3. Security & Manageability: Static IPs simplify network security policies, access control lists (ACLs), and monitoring tools, as they can be configured to allow or block traffic based on a known, fixed address.

40. Differentiate between a public IP address and a private IP address.

1. Scope & Routability: A public IP address is globally unique and routable on the public Internet. A private IP address is used only within a private local area network (LAN) and is not routable over the public Internet.

2. Assignment & Purpose: Public IPs are assigned by an ISP or regional registry and are used to identify a device uniquely on the Internet. Private IPs are freely assigned by a network administrator from reserved ranges (e.g., 10.x.x.x, 172.16.x.x-172.31.x.x, 192.168.x.x) for internal network communication.
3. Cost & Translation: Public IPs are a finite resource and often have an associated cost. Private IPs are free to use. Devices with private IPs access the Internet via Network Address Translation (NAT), where a router translates their private address to a single public IP.

41. A user name and password are used to determine what?
They are used to determine user identity and access rights.
1. The username identifies who the user is.
2. The password authenticates the user.
3. Together, they determine what resources and permissions the user is allowed to access.

42. Why is climate control important? What impact does it have on performance or availability?
Climate control is important to maintain proper temperature and humidity.
1. Prevents overheating, which can reduce system performance.
2. Reduces risk of hardware failure and downtime.
3. Ensures systems remain available and reliable.

43. What is performance monitoring?
Performance monitoring is the process of tracking system resource usage.
1. Monitors CPU, memory, disk, and network usage.
2. Helps identify bottlenecks and failures.
3. Ensures systems operate efficiently and reliably.

44. What three methods are used to identify how pin block connectors should be plugged in?
1. Keying (physical shape prevents incorrect insertion).
2. Color coding.
3. Labels or markings (numbers, arrows, or text).

45. You are installing a new server and you don't have any new circuits available to you, so you have to use an existing circuit. What should you do before plugging in the new server?
1. Calculate the electrical load on the circuit.
2. Ensure the circuit can handle the additional power draw.
3. Verify proper grounding and circuit protection.

46. You are evaluating three different products with similar features and similar costs. You're concerned about making the right decision. How should you determine which is the best product?
1. Review vendor reputation and support.
2. Check reliability, reviews, and warranty.
3. Consider compatibility and future scalability.

47. You are getting ready to add a new server. What method might you use to choose the operating system to run on that new server?
1. Determine the server’s role and application requirements.
2. Check hardware and software compatibility.
3.Consider security, support, and administrator expertise.

48. When should firmware updates be performed?
1. When updates fix bugs, security issues, or stability problems.
2. During scheduled maintenance windows.
3. After testing to ensure compatibility.

49. You are installing a new server in a new rack, complete with a very heavy rack-mount UPS. Your boss likes the fact that the UPS has LEDs on it indicating load and battery availability and wants the UPS mounted at the top of the rack where everyone will be able to see these LEDs. Why is this a bad idea?
1. It raises the center of gravity, making the rack unstable.
2. Increases risk of tipping and injury.
3. Heavy equipment should be mounted at the bottom for safety.

50. What problems can a mustimeter help you diagnose?
1. Voltage, current, and resistance issues.
2. Power supply and grounding problems.
3. Continuity and faulty cables or circuits.

51. What role do servers typically play in IT monitoring?
Servers are central to IT monitoring in the following ways:
1. Data Collection & Aggregation – They run monitoring agents or act as collectors (e.g., Nagios server, Zabbix server) that gather metrics, logs, and status information from network devices, applications, and other servers.
2. Alerting & Notification – Monitoring servers process collected data, compare it against predefined thresholds, and trigger alerts (email, SMS, dashboards) when issues are detected.
3. Performance & Availability Reporting – They store historical data, generate reports, and provide visibility into system health, capacity trends, and uptime through dashboards (e.g., Grafana, PRTG).

52. What IT automation tools have you used with servers?
Common server automation tools include:
1. Configuration Management – Ansible (agentless, YAML-based), Puppet (model-driven), Chef (Ruby-based), for consistent server setup and compliance.
2. Orchestration & Deployment – Terraform for infrastructure-as-code provisioning, Jenkins for CI/CD pipelines to automate deployment and testing.
3. Scripting & Task Automation – PowerShell (Windows), Bash scripting (Linux), and Python with libraries like Paramiko or Fabric for ad-hoc automation tasks.

53. How do domain name servers' function?
Domain Name Servers (DNS) translate human-readable domain names into IP addresses:
1. Hierarchical Resolution Process – When a user enters a URL, the DNS resolver queries the root DNS servers, then TLD (Top-Level Domain) servers (.com, .org), then authoritative name servers for the specific domain to obtain the IP address.
2. Caching for Efficiency – DNS resolvers (often at ISPs or within organizations) cache responses temporarily to speed up future requests and reduce load on authoritative servers.
3. Record Types & Responses – DNS servers store different record types (A, AAAA, MX, CNAME, etc.) and respond to queries with the appropriate record, directing traffic to the correct server.

54. What are the benefits of using subnets?
Subnetting (dividing a network into smaller segments) provides:
1. Improved Network Performance & Reduced Congestion – By limiting broadcast traffic to smaller segments, subnetting decreases collisions and improves data transfer efficiency.
2. Enhanced Security & Isolation – Subnets allow network administrators to implement access controls (ACLs, firewalls) between segments, containing breaches and restricting lateral movement.
3. Efficient IP Address Management – Subnetting helps organize IP addresses geographically, departmentally, or by function, making allocation simpler and reducing wastage of addresses.

55. How do you handle a server that won't boot?

A systematic approach to troubleshooting a non-booting server involves:

1. Initial Diagnosis & Physical Checks – Verify power supply (cables, PSU, UPS), check indicator lights, listen for abnormal beep codes from the motherboard, and ensure hardware components (RAM, drives, cables) are seated properly. Check for overheating or visible damage.
2. Access Console/Diagnostic Tools – Use out-of-band management tools like iDRAC (Dell), iLO (HPE), or IPMI to view console output and BIOS/POST messages remotely. This helps identify the boot stage failure (e.g., disk error, missing OS, memory failure, corrupted bootloader).
3. Boot Process Troubleshooting – Based on the error:
    - Disk/Boot Issues: Boot from rescue media (USB/CD), check disk health (fsck, chkdsk), repair bootloader (GRUB, Windows Boot Manager), or restore from backup.
    - Driver/Kernel Issues: Use safe mode or recovery mode to roll back updates or disable problematic drivers.
    - Hardware Failure: Test with minimal hardware configuration (one RAM stick, onboard graphics) and replace faulty components (disk, memory, PSU) as indicated by diagnostics.

56. What's the difference between a firewall and antivirus software?

The primary differences are:

1. Purpose & Protection Layer – A firewall is a network security system (hardware or software) that monitors and controls incoming/outgoing traffic based on predetermined rules, acting as a barrier between trusted and untrusted networks. Antivirus software is an endpoint security application that scans, detects, and removes malicious software (malware, viruses, trojans) from files and systems.
2. Operation & Scope – Firewalls filter traffic at the network/transport layer (IP addresses, ports, protocols) to prevent unauthorized access. Antivirus works at the file/application level, using signature-based and heuristic analysis to identify and quarantine malicious code.
3. Proactive vs. Reactive – Firewalls are primarily preventive, blocking threats before they reach the system. Antivirus is often reactive (though modern versions include real-time protection), detecting and removing malware already present or attempting to execute.

57. How would you recover lost files from a system infected by a virus?

1. Isolate & Clean the System – Disconnect the infected system from the network to prevent spread. Use a reputable, updated antivirus or anti-malware tool (bootable USB if OS is compromised) to scan and remove the virus. Tools like Malwarebytes, Kaspersky Rescue Disk, or Windows Defender Offline can be used.
2. Recover Files from Backups – If regular backups exist (cloud, external drive, network share), restore clean versions of the lost files after ensuring the backup itself is not infected (scan backup media).
3. Use Data Recovery Tools – If no backup exists, employ data recovery software (e.g., Recuva, EaseUS, TestDisk) to attempt recovery of deleted/corrupted files. For critical data, consider professional recovery services. Always recover files to an external drive to avoid overwriting data.

58. What are proxy servers and why are they important?

Proxy servers act as intermediaries between client devices and the internet:

1. Function & Operation – A proxy receives client requests, forwards them to the destination server (often masking the client’s IP), and returns the response. It can cache content, filter requests, and log activity.

2. Key Benefits/Importance:
    - Security & Privacy: Hides internal IP addresses, providing an additional layer of anonymity and protection.
    - Access Control & Filtering: Enforces organizational policies by blocking unwanted websites and monitoring user traffic.
    - Performance: Caches frequently accessed web content, reducing bandwidth usage and speeding up response times for users.

59. What is Windows Server?
Windows Server is a line of server operating systems from Microsoft designed for enterprise-level management, data storage, applications, and networking.

1. Core Purpose – It provides a platform for running critical network services such as Active Directory (user/device management), DNS/DHCP, file/print sharing, web servers (IIS), and virtualization (Hyper-V).
2. Key Features – Includes centralized management tools, enhanced security (e.g., Windows Defender, BitLocker), high availability features (failover clustering), and support for large-scale workloads and cloud integration (Azure Arc).
3. Common Versions – Examples include Windows Server 2016, 2019, and 2022, each offering Long-Term Servicing Channel (LTSC) releases for stable, long-term support.

60. What is DNS and why is it critical?
DNS (Domain Name System) is a hierarchical, decentralized naming system that translates human-readable domain names (e.g., www.example.com) into machine-readable IP addresses (e.g., 192.0.2.1).
Why it is critical:

1. Enables User-Friendly Navigation – Without DNS, users would need to memorize numerical IP addresses for every website or service, making the internet impractical for everyday use.
2. Supports Core Network Services – DNS underpins email delivery (MX records), service discovery (SRV records), load balancing, and content delivery networks (CDNs).
3. Ensures Redundancy and Reliability – Distributed DNS architecture provides fault tolerance and prevents single points of failure, ensuring continuous network accessibility.

61. What is the role of a Hypervisor in server virtualization?
    The hypervisor is the software layer that sits between the physical hardware and the virtual machines (VMs).
    1. Abstraction: It decouples the VM's operating system and applications from the underlying physical hardware.
    2. Resource Management: It allocates the physical server's resources (CPU, RAM, storage, network) to multiple VMs, ensuring they don't interfere with each other.
    3. Isolation: It provides a secure, isolated environment for each VM, so a crash or security breach in one VM doesn't affect the others. (Examples: VMware ESXi, Microsoft Hyper-V, KVM).

62. Explain the concept of "Infrastructure as Code" (IaC).
    IaC is the practice of managing and provisioning infrastructure (servers, networks, load balancers) through machine-readable definition files, rather than manual hardware configuration or interactive configuration tools.
    1. Automation & Consistency: It allows you to automate the entire server setup process, ensuring that environments are built identically every time, eliminating configuration drift.

2. Version Control: The configuration files can be stored in a version control system (like Git), providing a history of changes, facilitating collaboration, and enabling easy rollbacks.
3. Efficiency: It makes infrastructure management faster, more repeatable, and less prone to human error. (Examples: Terraform, AWS CloudFormation).

63. What is a Content Delivery Network (CDN) and how does it relate to servers?
A CDN is a geographically distributed network of proxy servers and their data centers.
    1. Distributed Caching: Its goal is to distribute service spatially relative to end-users to provide high availability and high performance. It caches static content (images, CSS, JavaScript) from an origin server.
    2. Reduced Latency: When a user requests content, the CDN serves it from the closest edge server (Point of Presence), drastically reducing latency and load times.
    3. Offloading Origin Servers: By handling a large portion of content delivery, a CDN reduces the traffic and processing load on the main origin servers, allowing them to focus on dynamic content and core application logic.

64. What is the difference between HTTP and HTTPS?
The primary difference is security.
    1. HTTP (Hypertext Transfer Protocol): Transfers data between a web server and a browser in plain text. This means any intercepted communication (like passwords or credit card numbers) can be read.
    2. HTTPS (HTTP Secure): Encrypts the data using SSL/TLS (Secure Sockets Layer/Transport Layer Security). This ensures that all data exchanged is private and integral, and it authenticates the website the user is communicating with.
    3. Importance for Servers: Any server handling sensitive data must use HTTPS. It's also a ranking factor for search engines and a requirement for many modern web features.

65. What is a Bootloader?
A bootloader is a small, critical piece of software that runs when a computer or server is first turned on (after the POST).
    1. Initialization: Its primary job is to load the operating system kernel into memory and then start it.
    2. Location: It is typically stored on the master boot record (MBR) or a GUID Partition Table (GPT) of the boot drive.
    3. Examples: Common bootloaders include GRUB (Grand Unified Bootloader) for Linux and the Windows Boot Manager for Windows Server.

66. What is the purpose of a DMZ (Demilitarized Zone) network?
A DMZ is a physical or logical subnetwork that separates an internal local area network (LAN) from other untrusted networks, usually the public internet.
    1. Security Buffer: It acts as a buffer zone. Servers that need to be accessible from the internet (like web, email, and DNS servers) are placed in the DMZ.
    2. Isolation: If an attacker compromises a server in the DMZ, they are still isolated from the internal corporate network by a firewall. This prevents them from easily accessing sensitive internal data.
    3. Layered Defense: It's a key component of a defense-in-depth security strategy.

67. What is the "three-tier architecture" in relation to servers?
It's a well-established software architecture pattern that organizes applications into three logical layers, each often running on its own server or set of servers.

1. Presentation Tier: The user interface (e.g., a web browser or mobile app). It interacts with the user and sends requests to the logic tier.
2. Application (Logic) Tier: The core of the application, containing the business logic. It processes user requests, performs calculations, and manages data flow between the presentation and data tiers.
3. Data Tier: The database server(s) that stores and manages the application's data. It is typically isolated and secured.

68. What is a "bare-metal" server?
    A bare-metal server is a physical server dedicated to a single tenant or user.
    1. Dedicated Hardware: Unlike virtual servers that share hardware, a bare-metal server gives you exclusive access to all its physical resources (CPU, RAM, storage).
    2. High Performance & Control: This makes it ideal for resource-intensive, high-performance computing (HPC) workloads, or applications that require specific hardware configurations or hypervisor-level access that isn't possible in a virtualized environment.
    3. Cloud Offering: Many cloud providers now offer bare-metal instances as a service, combining the performance of dedicated hardware with the on-demand provisioning of the cloud.

69. What is the difference between TCP and UDP?
    Both are core transport protocols for sending data over a network, but they have different characteristics.
    1. TCP (Transmission Control Protocol): Is connection-oriented and reliable. It establishes a connection, ensures all data packets are received in the correct order, and retransmits lost packets. It's like a phone call. Used for web (HTTP), email (SMTP), and file transfers (FTP).
    2. UDP (User Datagram Protocol): Is connectionless and unreliable. It sends data packets (datagrams) without establishing a connection, with no guarantee of delivery or order. It's like sending a letter through the mail. It's faster but less reliable, ideal for live streaming, online gaming, and DNS queries where speed is critical and occasional packet loss is acceptable.

70. What is server latency?
    Server latency is the time delay it takes for data to travel from a client to a server and back again (round-trip time).
    1. Measurement: It is typically measured in milliseconds (ms).
    2. Causes: High latency can be caused by physical distance, network congestion, overloaded server resources, or inefficient application code.
    3. Impact: High latency negatively impacts user experience, making websites and applications feel slow and unresponsive. It is a critical metric for real-time applications like video conferencing and online gaming.

71. What is a cron job?
    A cron job is a scheduled task that runs automatically on Unix/Linux servers at specified times or intervals. It is used to automate repetitive work like backups, log rotation, certificate renewal, or script execution. You define it using a cron expression (minute, hour, day of month, month, day of week) followed by the command to run. Cron jobs are managed with the `crontab` command.
72. What is the difference between a process and a service on a server?
    A process is any running instance of a program or application on the server. It has a process ID (PID) and can run in the foreground or background. A service (also called a daemon) is a special type of process that runs continuously in the background, waiting for requests or performing system functions, such as `sshd`, `nginx`, or `mysql`.

Services are typically managed by an init system like systemd and can be started, stopped, or enabled to run at boot. Regular processes are often temporary or user-initiated.

73. What is a jump server (or jump box)?
    A jump server is a hardened, intermediary server used to access other servers within a secure network zone. It acts as a single entry point, requiring administrators to connect to the jump server first before accessing internal resources. This reduces the attack surface, enforces stronger authentication (like multi-factor), and provides a centralized audit trail. Jump servers are often deployed in a DMZ or as a bastion host.
74. What is the purpose of a heartbeat in server clustering?
    A heartbeat is a periodic signal sent between servers in a cluster to indicate that they are operational. If a cluster node stops sending heartbeats, other nodes assume it has failed and begin failover procedures. Heartbeats typically run over a dedicated network interface or separate VLAN to avoid false positives due to network congestion. They also help maintain cluster quorum and prevent split-brain situations where multiple nodes act as the active master simultaneously.
75. What is out-of-band management (OOB)?
    Out-of-band management is a method of managing servers using a dedicated management channel that is separate from the main network connection. It allows administrators to access the server's console, power cycle it, mount virtual media, and view hardware health even when the primary operating system is down or the main network is offline. Common examples include Dell iDRAC, HP iLO, and generic IPMI. This provides "lights-out" management and reduces the need for physical presence in a data center.
76. What is a sticky session (session affinity) in load balancing?
    Sticky sessions ensure that all requests from a specific client are sent to the same backend server for the duration of their session. This is required when a server is stateful and stores session data locally, because the client's state would be lost if requests were distributed to different servers. The load balancer tracks the client, often via a cookie or by hashing the source IP address. The trade-off is that sticky sessions reduce load-balancing flexibility and can lead to uneven load distribution.
77. What is the three-second rule in server response time?
    The three-second rule is a guideline that users expect a webpage or service to respond within three seconds. Longer response times significantly increase bounce rates and user frustration. For server administrators, this means optimizing performance through caching, faster databases, content delivery networks (CDNs), and load balancing. Modern expectations can be even stricter, with sub-second responses ideal for APIs and interactive applications.
78. What is the difference between a cold, warm, and hot standby server?
    A cold standby is a backup server that is powered off. It requires manual intervention and operating system boot before taking over, resulting in significant downtime. A warm standby is powered on and has services installed but is not actively processing traffic; failover is partly automated but still takes minutes. A hot standby is an identical server actively running in parallel, ready to take over instantly (often in milliseconds) when a failure is detected. Hot standbys are used in active-passive or active-active clusters.
79. What is vendor lock-in in the context of servers?
    Vendor lock-in occurs when a customer becomes dependent on a single vendor's proprietary hardware, software, or application programming interfaces (APIs), making it costly or difficult to switch to another provider. Examples include using proprietary server management tools like VMware ESXi without open alternatives, cloud-specific services like AWS Lambda, or specialized storage arrays. The risks include reduced bargaining power, difficulty migrating, and potential price increases. Mitigation strategies include adopting open standards, using Infrastructure as Code (IaC) with portability in mind, and preferring multi-cloud or hybrid architectures.

80. What is the purpose of a watchdog timer on a server?
    A watchdog timer is a hardware or software mechanism that automatically resets a server if it becomes unresponsive. The server's operating system must periodically "kick" or "pet" the watchdog. If the server freezes due to a kernel panic or other crash, the timer expires and triggers a hardware reset. This is critical for embedded systems, remote servers, and appliances where manual reboot is impractical. Watchdog timers are available in most server motherboards (hardware watchdog) or via software like the `watchdog` daemon on Linux.
81. What is server sprawl?
    Server sprawl is the uncontrolled proliferation of servers (physical or virtual) within an organization. It is caused by easy provisioning of virtual machines and containers, lack of governance, and no retirement process for unused servers. The consequences include wasted resources (power, cooling, licenses), increased management overhead, security vulnerabilities from orphaned and unpatched servers, and higher costs. Prevention strategies include implementing lifecycle management, regular audits, tagging policies, and automated decommissioning.
82. What is configuration drift in server management?
    Configuration drift occurs when servers that were initially identical become different over time due to manual changes, updates, or untracked modifications. This leads to inconsistent behavior, hard-to-debug failures, security gaps where one server misses a patch, and compliance violations. Prevention involves using Infrastructure as Code (IaC) and configuration management tools like Ansible, Puppet, or Chef to enforce a desired state. Detection can be done through periodic audits, drift detection scripts, and immutable infrastructure patterns where servers are replaced instead of modified.
83. What is the difference between scalability and elasticity in server infrastructure?
    Scalability is the ability of a system to handle growing workloads by adding resources, either by scaling up (vertical) or scaling out (horizontal). It is often planned and gradual. Elasticity is the ability to automatically add or remove resources in real time based on current demand. It is dynamic and on-demand, typical of cloud environments. For example, a scalable website can handle 1 million users by design, while an elastic website automatically spins up extra servers during a traffic spike and removes them when traffic drops.
84. What is server decommissioning?
    Server decommissioning is the formal process of retiring a server from active service. The steps include identifying the server, migrating or archiving data, wiping storage media using secure erase or physical destruction, removing the server from monitoring and backup systems, disconnecting hardware, and updating documentation. This process is important because it prevents orphaned servers that pose security risks, reclaims resources, and reduces licensing and maintenance costs. Many regulations like HIPAA and PCI-DSS require certified data sanitization before disposal.
85. What is the role of systemd on a modern Linux server?
    systemd is the init system and service manager used by most modern Linux distributions. It starts, stops, and monitors system services (daemons) using unit files with a `.service` extension. systemd significantly reduces boot time by starting services concurrently. It also provides logging through journald, timers as an alternative to cron, socket activation, and dependency management between services.
86. What is swap space on a server?
    Swap space is a portion of the hard disk used as an extension of physical RAM when memory becomes full. The operating system moves inactive memory pages to swap, freeing RAM for active processes. Swap is much slower than RAM, often by a factor of 1000 or more, so heavy swapping indicates memory pressure and leads to severe performance degradation. On modern servers with ample RAM, a small amount of swap (2-4 GB) is still recommended for crash dumps and handling rare spikes, but it should not be used as a substitute for sufficient memory.

87. What is a mail transfer agent (MTA)?
    An MTA is software responsible for routing, delivering, and receiving email messages between servers. It uses the Simple Mail Transfer Protocol (SMTP) to relay emails from a mail client or another MTA to the destination mail server. Common examples include Postfix, Sendmail, Exim, and Microsoft Exchange (which includes MTA functionality). Proper configuration involves DNS records such as MX, SPF, and DKIM for correct delivery and spam prevention.
88. What is time synchronization and why is it critical for servers?
    Time synchronization ensures all servers have the same accurate time, typically using the Network Time Protocol (NTP). It is critical because authentication protocols like Kerberos and SSL/TLS rely on timestamps; clock skew can break logins and certificate validation. Logs from multiple servers are only useful for debugging if timestamps align. Distributed systems like databases and clusters require consistent time to avoid data corruption. Best practice is for all servers to sync to a trusted internal or public NTP server such as pool.ntp.org.
89. What is the purpose of resource limits (ulimit/cgroups) on a server?
    Resource limits prevent a single process or user from consuming all available CPU, memory, or disk I/O. This ensures fairness so that all users and applications get their share of resources, and it stops a runaway process from crashing the entire server. On Linux and Unix systems, ulimit provides per-shell and per-process limits. Control groups (cgroups) are used by container runtimes like Docker and Kubernetes, as well as systemd, for fine-grained resource isolation.
90. What is the difference between a patch and an update?
    A patch is a small, targeted fix for a specific issue, often a security vulnerability or critical bug. Patches are usually applied urgently. An update is a broader collection of fixes, improvements, and sometimes new features. Updates are typically larger and installed less frequently, such as service packs or cumulative updates. In practice, the terms are often used interchangeably, but patches are a subset of updates with higher priority, especially for security.
91. What is remediation in server security?
    Remediation is the process of fixing a confirmed security vulnerability or misconfiguration. The steps include identifying the root cause, applying the necessary fix such as a patch, configuration change, firewall rule, or removal of malicious files, and then verifying that the fix is effective. Tools like Ansible, Chef, or vulnerability scanners such as Tenable and Qualys can automate remediation tasks. Remediation follows detection, for example from a vulnerability scan or audit, and is part of the security lifecycle.
92. What is the principle of least privilege for servers?
    The principle of least privilege states that a user, process, or service should be granted only the minimum permissions necessary to perform its function. In practice, users should not have administrator or root access unless required. Services should run under dedicated, low-privileged accounts. This limits the damage from a compromised account or process. Implementation methods include role-based access control (RBAC), sudo rules, file permissions, and removing unnecessary privileges.
93. What is a key pair in server authentication?
    A key pair consists of a private key, which is kept secret, and a public key, which is shared. It is used for cryptographic authentication. For SSH authentication, the user stores the private key on their client machine, and the public key is added to the server's ~/.ssh/authorized_keys file. The server verifies possession of the private key without transmitting it. Key pairs are more secure than passwords because they resist brute-force attacks, enable automation without interactive login, and can be protected with a passphrase.
94. What is the purpose of a bastion host?
    A bastion host is a special-purpose server that is specifically hardened to withstand attacks and is positioned as the sole entry point to a private network. It is the only

server exposed directly to the internet or an untrusted network; all internal servers are inaccessible from outside. Hardening measures include minimal installed packages, strict firewall rules, intensive logging, intrusion detection, and multi-factor authentication. Administrators first SSH into the bastion host, and from there connect to internal resources.

95. What is rate limiting on a server?
    Rate limiting controls the number of requests a client can make to a server within a specific time window. Its purpose is to prevent abuse, mitigate DDoS attacks, ensure fair resource usage, and protect backend services from being overwhelmed. Implementation can be done at the web server level using Nginx `limit_req` or Apache `mod_ratelimit`, at API gateways, or on load balancers. When a client exceeds the limit, the server typically returns an HTTP `429 Too Many Requests` response or simply drops the connection.
96. What is the difference between synchronous and asynchronous replication for server data?
    Synchronous replication writes data to both the primary server and the replica(s) before the write operation is acknowledged as successful. This provides strong consistency but introduces higher latency and reduced availability if a replica fails. Asynchronous replication acknowledges the write immediately on the primary and replicates data to replicas later. This results in lower latency but carries the risk of data loss if the primary fails before replication completes. Synchronous replication is used for critical financial systems, while asynchronous replication is used for geographic redundancy or non-critical workloads.
97. What is the role of a reverse proxy cache (for example, Varnish)?
    A reverse proxy cache stores copies of frequently requested web content, such as HTML pages, images, or API responses, in memory or on disk. It serves subsequent requests directly from the cache without hitting the backend application server. This drastically reduces response time and backend load. The cache respects HTTP cache headers like `Cache-Control` and `Expires`, and can bypass caching for dynamic or authenticated requests. Popular examples include Varnish Cache and Nginx with caching enabled.
98. What is a socket in server networking?
    A socket is a software endpoint that enables bidirectional communication between two processes over a network or within the same machine. It consists of an IP address and a port number combined, for example `192.168.1.10:443`. There are stream sockets using TCP for reliable, ordered connections, and datagram sockets using UDP for connectionless, fast transmission. Servers create listening sockets that accept incoming client connections, and each accepted connection gets a new dedicated socket for that communication session.
99. What is a jumbo frame and why would a server use it?
    A jumbo frame is an Ethernet frame with a payload larger than the standard 1500 bytes, typically up to 9000 bytes. The benefit is reduced overhead by sending fewer, larger packets, which improves throughput and reduces CPU utilization for bulk data transfers such as NFS, iSCSI, or backups. The requirement is that all devices on the same network segment, including switches, routers, and network interface cards, must support and be configured with the same maximum transmission unit (MTU). The downside is incompatibility with standard internet traffic, so jumbo frames are typically used only in isolated storage or data center networks.
100. What is a post-mortem in server management?
    A post-mortem is a formal, blameless analysis conducted after a significant server outage or incident. The goal is to understand what happened, why it happened, and what actions will prevent recurrence. Key elements include a timeline of events, root cause analysis, impact assessment, and action items with owners and deadlines. High-reliability teams share post-mortems openly to learn collectively rather than to assign

blame. The document is often stored in a wiki or runbook system as a reference for future troubleshooting.