**1. What could be the issue if you can ping a server by IP but not by hostname?**

* **Potential Issues:**
  1. **DNS Issue**: The Domain Name System (DNS) is not resolving the hostname to the correct IP address.
  2. **Host File Misconfiguration**: The /etc/hosts file (on Linux/Unix) or C:\Windows\System32\drivers\etc\hosts (on Windows) may lack the proper mapping.
  3. **DNS Server Unreachable**: The DNS server may be down or unreachable from the client.
  4. **Cache Issue**: The DNS cache on the client may be stale or corrupted.
* **Troubleshooting Steps:**
  1. Test DNS resolution using nslookup or dig.
  2. Check and update the local hosts file.
  3. Flush the DNS cache (e.g., sudo systemd-resolve --flush-caches on Linux or ipconfig /flushdns on Windows).
  4. Verify that the DNS server is functioning and reachable.
  5. Check for typos in the hostname.

**2. You get a call that a user cannot reach a server. How would you troubleshoot?**

* **Steps to Troubleshoot:**
  1. **Verify Connectivity**:
     + Ping the server's IP address and hostname.
     + Use traceroute (Linux) or tracert (Windows) to identify network path issues.
  2. **Check Server Status**:
     + Use SSH or RDP to log in to the server.
     + Check if the server is powered on and responsive.
  3. **Firewall and Security Rules**:
     + Verify that firewall rules or security groups allow traffic from the user's IP.
  4. **Logs and Monitoring**:
     + Review server logs (e.g., /var/log/messages, /var/log/syslog) for errors.
  5. **Reproduce the Issue**:
     + Attempt to replicate the user's steps to narrow down the problem.

**3. How to troubleshoot if a website running on your server is down?**

* **Troubleshooting Steps**:
  1. **Check Website Availability**:
     + Use curl or a browser to check if the website is accessible.
  2. **Verify Web Server Status**:
     + Restart the web server (e.g., systemctl restart apache2 or nginx).
     + Check the server status with systemctl status apache2 or nginx.
  3. **Check Application Logs**:
     + Inspect application-specific logs for errors (e.g., /var/log/nginx/error.log or /var/log/httpd/error\_log).
  4. **DNS Issues**:
     + Confirm that the DNS resolves the website correctly.
  5. **Firewall and Network**:
     + Ensure the required ports (e.g., 80/443) are open in the firewall.
  6. **Resource Availability**:
     + Verify there is sufficient CPU, memory, and disk space using tools like top, htop, or df.

**4. You are notified that a server is down. How to troubleshoot?**

* **Troubleshooting Steps**:
  1. **Check Server Accessibility**:
     + Ping the server and attempt SSH/RDP login.
  2. **Power and Hardware**:
     + Verify that the server is powered on.
     + Check for hardware alerts if using a physical server.
  3. **Application and Services**:
     + Confirm if critical services are running (systemctl list-units or ps).
  4. **Disk and Resources**:
     + Check if the disk is full (df -h) or if memory/CPU is exhausted (free -m, top).
  5. **Logs and Alerts**:
     + Review logs in /var/log or equivalent.
     + Check monitoring systems like Nagios or Prometheus for insights.

**5. How to handle a situation when your DBA is asking for more memory/disk frequently?**

* **Steps to Address the Issue:**
  1. **Understand Requirements**:
     + Engage with the DBA to understand why additional resources are needed.
  2. **Optimize Resources**:
     + Review database queries and optimize for efficiency.
     + Purge or archive old data.
  3. **Scale Resources**:
     + If justified, provision more memory or disk using cloud scaling or virtualization tools.
  4. **Monitoring**:
     + Set up monitoring tools like CloudWatch or Grafana to proactively identify trends.

**6. What is your experience with shell scripting? Provide examples.**

* **Experience**:
  + Automating backups using rsync and cron.
  + Writing deployment scripts for CI/CD pipelines.
  + Parsing logs and generating reports with awk, sed, and grep.
* **Examples**:
  + **Disk Space Monitoring**:

bash

Copy code

#!/bin/bash

df -h | awk '$5 > 80 {print $0}' > /tmp/disk\_alert.txt

mail -s "Disk Usage Alert" admin@example.com < /tmp/disk\_alert.txt

* + **Service Health Check**:

bash

Copy code

#!/bin/bash

if ! systemctl is-active --quiet apache2; then

systemctl restart apache2

echo "Apache restarted on $(date)" >> /var/log/apache\_monitor.log

fi

**7. What are the different types of run-levels?**

* **Linux Run Levels**:
  1. **Runlevel 0**: Halt (shutdown).
  2. **Runlevel 1**: Single-user mode.
  3. **Runlevel 2**: Multi-user mode without networking.
  4. **Runlevel 3**: Multi-user mode with networking.
  5. **Runlevel 4**: Undefined/customizable.
  6. **Runlevel 5**: Multi-user mode with GUI.
  7. **Runlevel 6**: Reboot.
* Modern systems use systemd targets (e.g., graphical.target or multi-user.target).

**8. You get a call that a server is running slow. How would you troubleshoot?**

* **Steps to Troubleshoot**:
  1. **Identify Symptoms**:
     + Check for slowness in CPU (top), memory (free -m), or I/O (iostat).
  2. **Processes**:
     + Identify resource-hungry processes (top or ps aux).
  3. **Network Latency**:
     + Test with ping, traceroute, and netstat.
  4. **Disk Usage**:
     + Check disk utilization with df -h and du.
  5. **Logs**:
     + Investigate logs for errors or anomalies.

**9. Walk me through the process of recovering root password.**

* **Recovery Steps**:
  1. **Reboot the System**:
     + Restart the system and access the GRUB menu.
  2. **Edit GRUB**:
     + Press e to edit the boot entry.
     + Add init=/bin/bash or single at the end of the kernel line.
  3. **Boot into Single-User Mode**:
     + Press Ctrl+X or F10 to boot.
  4. **Mount Filesystem**:
     + Run mount -o remount,rw / to mount the filesystem in read-write mode.
  5. **Reset Password**:
     + Use passwd root to reset the password.
  6. **Reboot Normally**:
     + Reboot the system with reboot or shutdown -r now.

**Show all lines except any lines starting with the character # in a file**

* **Question**: How to display all lines from a file except those starting with the character #?
* **Answer**:
  1. Use the grep command with the -v option to invert the match.
  2. Execute the following:

bash

Copy code

grep -v '^#' filename

* 1. Verify that all lines starting with # are excluded.

**2. How can you continuously monitor log files for errors?**

* **Question**: How do you monitor a log file for real-time errors?
* **Answer**:
  1. Use tail -f to follow the log file updates.
  2. Combine with grep to filter for errors:

bash

Copy code

tail -f /var/log/syslog | grep -i error

* 1. Verify the output for live error messages.

**3. Find and remove files older than 7 days through a cron job**

* **Question**: How do you automate the deletion of files older than 7 days using a cron job?
* **Answer**:
  1. Use the find command with -mtime:

bash

Copy code

find /path/to/directory -type f -mtime +7 -exec rm -f {} \;

* 1. Test the command to ensure it identifies files correctly.
  2. Add the command to the cron job:

bash

Copy code

crontab -e

* 1. Schedule it (e.g., daily at 3:00 AM):

bash

Copy code

0 3 \* \* \* find /path/to/directory -type f -mtime +7 -exec rm -f {} \;

* 1. Save and verify using crontab -l.

**4. Find, print, users on the system, and redirect to a file**

* **Question**: How do you list all users on a system and save the output to a file?
* **Answer**:
  1. Extract the first column (username) from /etc/passwd:

bash

Copy code

cut -d: -f1 /etc/passwd

* 1. Redirect the output to a file:

bash

Copy code

cut -d: -f1 /etc/passwd > users.txt

* 1. Open the file to verify the results:

bash

Copy code

cat users.txt

**5. How would you get only the 2nd column from a file?**

* **Question**: How can you extract the second column from a file?
* **Answer**:
  1. Use the awk command:

bash

Copy code

awk '{print $2}' filename

* 1. Verify the extracted column by reviewing the output.
  2. Save the output to a file if needed:

bash

Copy code

awk '{print $2}' filename > column2.txt

**6. Broadcast a message to users**

* **Question**: How do you send a broadcast message to all logged-in users?
* **Answer**:
  1. Use the wall command:

bash

Copy code

wall "Server maintenance in 15 minutes. Please save your work."

* 1. Ensure the message is received by all users.

**7. User with no login access**

* **Question**: How do you create a user with no login access?
* **Answer**:
  1. Add a user with the nologin shell:

bash

Copy code

useradd -s /sbin/nologin username

* 1. Verify the user's shell in /etc/passwd:

bash

Copy code

grep username /etc/passwd

* 1. Confirm that the user cannot log in.

**8. Schedule a server reboot in 15 minutes**

* **Question**: How do you schedule a reboot in 15 minutes?
* **Answer**:
  1. Use the shutdown command:

bash

Copy code

shutdown -r +15 "Server rebooting in 15 minutes."

* 1. Confirm the scheduled reboot with:

bash

Copy code

shutdown -c

(Use -c to cancel if necessary.)

**9. Disk usage**

* **Question**: How do you check disk usage on a system?
* **Answer**:
  1. Use the df command:

bash

Copy code

df -h

* 1. Analyze disk usage for each partition.

**10. Sticky Bit**

* **Question**: What is a sticky bit, and how do you set it?
* **Answer**:
  1. A sticky bit restricts deletion of files to their owners.
  2. Set the sticky bit on a directory:

bash

Copy code

chmod +t /directory

* 1. Verify with:

bash

Copy code

ls -ld /directory

* 1. The directory permissions should display t (e.g., drwxrwxrwt).

**11. Print the 10th line of a file only**

* **Question**: How do you print only the 10th line of a file?
* **Answer**:
  1. Use sed:

bash

Copy code

sed -n '10p' filename

* 1. Verify the output matches the 10th line.

**12. Restore a hacked server**

* **Question**: How do you restore a hacked server?
* **Answer**:
  1. Disconnect the server from the network.
  2. Identify and isolate malicious processes:

bash

Copy code

ps aux | grep suspicious

* 1. Analyze logs in /var/log for anomalies.
  2. Backup important data to a secure location.
  3. Reinstall the operating system to ensure security.
  4. Harden the server post-restoration:
     + Apply patches and updates.
     + Set up a firewall.
     + Enable intrusion detection systems.

**13. Server security**

* **Question**: How do you secure a server?
* **Answer**:
  1. Regularly update all packages:

bash

Copy code

apt update && apt upgrade -y

* 1. Configure a firewall (e.g., ufw or iptables).
  2. Enforce SSH security:
     + Disable root login (PermitRootLogin no).
     + Use key-based authentication.
  3. Monitor logs and configure alert systems (e.g., fail2ban).

**14. Log messages**

* **Question**: How do you monitor log messages?
* **Answer**:
  1. Use tail for real-time log monitoring:

bash

Copy code

tail -f /var/log/syslog

* 1. Filter specific messages with grep:

bash

Copy code

grep 'error' /var/log/syslog

**15. ICMP Disable**

* **Question**: How do you disable ICMP (ping) requests?
* **Answer**:
  1. Temporarily disable ICMP:

bash

Copy code

echo 1 > /proc/sys/net/ipv4/icmp\_echo\_ignore\_all

* 1. Make it permanent by adding to /etc/sysctl.conf:

bash

Copy code

net.ipv4.icmp\_echo\_ignore\_all = 1

* 1. Apply changes:

bash

Copy code

sysctl -p

**16. Explain fields in /etc/passwd file**

* **Question**: What are the fields in /etc/passwd?
* **Answer**:
  + **Format**: username:x:UID:GID:comment:home\_directory:shell
  + **Fields**:
    1. Username.
    2. Placeholder for password (x refers to /etc/shadow).
    3. User ID (UID).
    4. Group ID (GID).
    5. User info (e.g., full name).
    6. Home directory.
    7. Login shell.

**17. System uptime**

* **Question**: How do you check system uptime?
* **Answer**:
  1. Use the uptime command:

bash

Copy code

uptime

* 1. Review the output for uptime, users, and load averages.

**18. Listening Ports**

* **Question**: How do you find listening ports?
* **Answer**:
  1. Use the netstat or ss command:

bash

Copy code

netstat -tuln

or

bash

Copy code

ss -tuln

* 1. Analyze the output for listening ports and processes.

**19. Server down troubleshooting**

* **Question**: How do you troubleshoot a server that is down?
* **Answer**:
  1. **Ping** the server to check network connectivity.
  2. Attempt SSH or RDP access.
  3. Check if the server is powered on.
  4. Review logs in /var/log.
  5. Verify disk usage (df -h) and resource availability (top or free -m).

**20. Find files with specific permissions**

* **Question**: How do you locate files with specific permissions?
* **Answer**:
  1. Use the find command:

bash

Copy code

find /path/to/directory -type f -perm 0644

* 1. Adjust 0644 to the desired permission value.

**1. Find files that are over 10MB in size**

* **Question**: How do you find files larger than 10MB?
* **Answer**:
  1. Use the find command:

bash

Copy code

find /path/to/search -type f -size +10M

* 1. Explanation:
     + /path/to/search: Specify the directory to search.
     + -type f: Only find files.
     + -size +10M: Finds files larger than 10MB.
  2. To list details of the files:

bash

Copy code

find /path/to/search -type f -size +10M -exec ls -lh {} \;

**2. Show lines only containing #**

* **Question**: How do you find lines that contain only # in a file?
* **Answer**:
  1. Use grep:

bash

Copy code

grep '^#' filename

* 1. Explanation:
     + ^#: Matches lines beginning with #.
     + It excludes lines where # is in the middle or end.

**3. File lines with numbers**

* **Question**: How do you print file lines along with their line numbers?
* **Answer**:
  1. Use nl command:

bash

Copy code

nl filename

* 1. Or use cat with the -n option:

bash

Copy code

cat -n filename

* 1. Explanation:
     + nl: Numbers the lines in the output.
     + cat -n: Displays file content with line numbers.

**4. Server version and release**

* **Question**: How do you find the server's version and release?
* **Answer**:
  1. Use the uname command:

bash

Copy code

uname -a

* 1. Check the OS version:

bash

Copy code

cat /etc/os-release

* 1. For kernel version:

bash

Copy code

uname -r

* 1. Explanation:
     + uname -a: Displays kernel and system information.
     + /etc/os-release: Contains OS-specific details.

**5. Network Interface Files**

* **Question**: Where are the network interface configuration files located?
* **Answer**:
  1. For most Linux distributions:
     + Configuration files are typically found in /etc/network/interfaces or /etc/sysconfig/network-scripts/.
  2. View network interfaces using:

bash

Copy code

ip a

* 1. Explanation:
     + Debian-based systems use /etc/network/interfaces.
     + Red Hat-based systems use /etc/sysconfig/network-scripts/ifcfg-\*.

**6. Installed Software Packages**

* **Question**: How do you list installed software packages?
* **Answer**:
  1. On Debian-based systems:

bash

Copy code

dpkg -l

* 1. On Red Hat-based systems:

bash

Copy code

rpm -qa

* 1. To list with details on any system:

bash

Copy code

apt list --installed # Debian/Ubuntu

yum list installed # RHEL/CentOS

* 1. Explanation:
     + These commands display all installed software packages and their versions.

**7. Line Count**

* **Question**: How do you count the number of lines in a file?
* **Answer**:
  1. Use wc:

bash

Copy code

wc -l filename

* 1. Explanation:
     + wc -l: Counts the number of lines in the file.

**8. Find Disk Usage by the Largest Directories**

* **Question**: How do you find which directories consume the most disk space?
* **Answer**:
  1. Use du:

bash

Copy code

du -h --max-depth=1 /path/to/directory | sort -h

* 1. Explanation:
     + du -h: Displays disk usage in a human-readable format.
     + --max-depth=1: Limits the depth of directory traversal.
     + sort -h: Sorts the output by size.

**9. Find all Directories Named .conf Under Root**

* **Question**: How do you find directories named .conf under the root directory?
* **Answer**:
  1. Use the find command:

bash

Copy code

find / -type d -name "\*.conf"

* 1. Explanation:
     + /: Starts the search from the root directory.
     + -type d: Looks only for directories.
     + -name "\*.conf": Matches directories with .conf in their name.

**10. Find Files Not Accessed in Over 3 Days**

* **Question**: How do you find files that haven’t been accessed in over 3 days?
* **Answer**:
  1. Use find:

bash

Copy code

find /path/to/search -type f -atime +3

* 1. Explanation:
     + -type f: Only files are considered.
     + -atime +3: Finds files last accessed more than 3 days ago.

**11. View Differences Between Two Files**

* **Question**: How do you compare two files to see their differences?
* **Answer**:
  1. Use diff:

bash

Copy code

diff file1 file2

* 1. For side-by-side comparison:

bash

Copy code

diff -y file1 file2

* 1. Explanation:
     + diff: Shows the differences between files line by line.
     + -y: Displays a side-by-side comparison.

**12. System Configuration Files to Back Up Regularly**

* **Question**: Which system configuration files should be backed up regularly?
* **Answer**:
  1. **Essential Configuration Files**:
     + /etc/passwd: User accounts.
     + /etc/group: Group accounts.
     + /etc/shadow: Passwords.
     + /etc/fstab: Mount points.
     + /etc/network/interfaces or /etc/sysconfig/network-scripts/: Network configuration.
     + /etc/hosts: Hostname and IP mappings.
  2. **Backup Command**:

bash

Copy code

tar -cvzf config\_backup.tar.gz /etc

* 1. Explanation:
     + Always back up critical configuration files to ensure recovery in case of failure.

**13. View All Currently Logged-In Users**

* **Question**: How do you view all currently logged-in users?
* **Answer**:
  1. Use the who command:

bash

Copy code

who

* 1. For more details:

bash

Copy code

w

* 1. Explanation:
     + who: Lists users and their terminal sessions.
     + w: Displays users along with additional details like active processes and system load.