



# Cloud DevOps LokeshTech

## Linux Interview Question & Answers



Lokeshkumar



### LINUX COMMANDS

#### 1. What is Linux?

- Linux is an open-source operating system / kernel that forms the basis for various Linux distributions.

#### 2. What are the main components of a Linux system?

- The main components of a Linux system are the kernel, shell, and file system.

#### 3. What is the role of the Linux kernel?

The Linux kernel is the core component of the operating system that manages system resources and provides services to applications.

#### 4. What is a shell in Linux?

- The shell is a command-line interpreter that allows users to interact with the operating system.

- It accepts and executes commands.



**5. What are some popular Linux distributions?**

- Popular Linux distributions include Ubuntu, Debian, Fedora, CentOS, and Red Hat Enterprise Linux.

**6. How do you change file permissions in Linux?**

- The "chmod" command is used to change file permissions in Linux. For example, "chmod 755 filename" sets read, write, and execute permissions for the owner and read and execute permissions for others.

**7. What is the purpose of the "grep" command?**

- The "grep" command is used to search for specific patterns within files. It is often used for text searching and filtering.

**8. How do you find files in Linux?**

- The "find" command is used to search for files and directories in Linux based on various criteria like name, size, and permissions.

**9. What is the purpose of the "top" command?**

- The "top" command is used to monitor system processes and resource usage in real-time.

**10. How do you check the disk usage in Linux?**

- The "df" command is used to display disk space usage of file systems.

**11. What is a symbolic link in Linux?**

- A symbolic link, also known as a soft link, is a special type of file that points to another file or directory.

**12. What is the purpose of the "tar" command?**

- The "tar" command is used to create and manipulate archive files, often used for bundling multiple files into a single file.

**13. How do you start and stop services in Linux?**

- Service management varies among distributions. In systemd-based systems, you can use commands like "systemctl start service\_name" and "systemctl stop service\_name" to start and stop services.

-



**14. What is the purpose of the "ping" command?**

- The "ping" command is used to check the connectivity between a source and a destination using Internet Control Message Protocol (ICMP) echo requests and replies.

**15. How do you check the network configuration in Linux?**

- The "ifconfig" command is used to display the network configuration of a Linux system. However, in newer distributions, it has been replaced by the "ip" command.

**16. What is SSH and how does it work?**

- SSH (Secure Shell) is a cryptographic network protocol used for secure remote login, command execution, and file transfer between computers. It encrypts the communication between the client and server.

**17. How do you kill a process in Linux?**

The "kill" command is used to terminate a process. You can use the process ID (PID) or the "killall" command to kill processes by name.

**18. What is the purpose of the "rsync" command?**

- The "rsync" command is used for efficient file synchronization and transfer between systems.

**19. How do you check system hardware information in Linux?**

- The "lshw" command can be used to obtain detailed information about the hardware of a Linux system.

**20. What is a firewall in Linux?**

- A firewall is a security mechanism that controls incoming and outgoing network traffic based on a set of rules. It helps protect the system from unauthorized access.

**21. How do you check the system's IP address in Linux?**

- The "ip addr" command is used to display the IP addresses assigned to network interfaces on a Linux system.



**22. What is the purpose of the "cron" daemon?**

- The "cron" daemon is used for scheduling and automating recurring tasks in Linux.

**23. How do you mount a filesystem in Linux?**

- The "mount" command is used to attach a filesystem to the directory tree.

**24. What is the purpose of the "chroot" command?**

- The "chroot" command is used to change the root directory for a process, creating a separate environment with its own root directory.

**25. How do you compress and decompress files in Linux?**

- The "gzip" and "gunzip" commands are used to compress and decompress files using the gzip compression algorithm.

**26. What is the purpose of the "iptables" command?**

- The "iptables" command is used for configuring the Linux kernel firewall, which filters network traffic based on user-defined rules.

**27. How do you check the CPU usage in Linux?**

- The "top" command or tools like "htop" and "mpstat" can be used to monitor CPU usage in Linux.

**28. What is the purpose of the "useradd" command?**

- The "useradd" command is used to create user accounts in Linux.

**29. How do you search for a string within files in a directory?**

- The "grep" command with the "-r" option can be used to search for a string recursively within files in a directory.

**30. How do you check the available memory in Linux?**

- The "free" command displays the amount of free and used memory in a Linux system.



## Linux Commands



Lokeshkumar



1. **ls:** List files and directories in the current directory.

Example: `'ls'`

2. **cd:** Change directory.

Example: `'cd /path/to/directory'`

3. **pwd:** Print the current working directory.

Example: `'pwd'`

4. **mkdir:** Create a new directory.

Example: `'mkdir new_directory'`

5. **rm:** Remove files and directories.

Example: `'rm file.txt'`

6. **cp:** Copy files and directories.

Example: `'cp file.txt /path/to/destination'`

7. **mv:** Move or rename files and directories.

Example: `'mv file.txt /path/to/destination'`



8. **touch:** Create an empty file.  
Example: `'touch file.txt'`
9. **cat:** Display the contents of a file.  
Example: `'cat file.txt'`
10. **grep:** Search for a pattern in a file.  
Example: `'grep "pattern" file.txt'`
11. **head:** Display the first few lines of a file.  
Example: `'head file.txt'`
12. **tail:** Display the last few lines of a file.  
Example: `'tail file.txt'`
13. **chmod:** Change the permissions of a file or directory.  
Example: `'chmod 755 file.txt'`
14. **chown:** Change the ownership of a file or directory.  
Example: `'chown user:group file.txt'`
15. **ln:** Create a symbolic link to a file.  
Example: `'ln -s /path/to/file link'`
16. **find:** Search for files and directories.  
Example: `'find /path/to/search -name "pattern"'`
17. **tar:** Archive files and directories.  
Example: `'tar -czvf archive.tar.gz files/'`
18. **unzip:** Extract files from a zip archive.  
Example: `'unzip archive.zip'`
19. **man:** Display the manual page/ Help of a command.  
Example: `'man ls'`
20. **history:** View command history.  
Example: `'history'`





- 21. ps:** Display currently running processes.  
Example: `'ps aux'`
- 22. kill:** Terminate a process.  
Example: `'kill PID'`
- 23. df:** Display disk space usage.  
Example: `'df -h'`
- 24. du:** Estimate file and directory sizes.  
Example: `'du -sh directory'`
- 25. scp:** Copy files between hosts securely.  
Example: `'scp file.txt user@remote:/path/to/destination'`
- 26. ssh:** Connect to a remote host securely.  
Example: `'ssh user@host'`
- 27. ping:** Send ICMP echo requests to a host.  
Example: `'ping google.com'`
- 28. ifconfig:** Display network interface information.  
Example: `'ifconfig'`
- 29. wget:** Download files from the web.  
Example: `'wget https://example.com/file.txt'`
- 30. curl:** Transfer data from or to a server.  
Example: `'curl https://example.com'`
- 31. top:** Display system resource usage and running processes.  
Example: `'top'`
- 32. apt-get:** Package manager for Debian-based distributions.  
Example: `'apt-get install package'`
- 33. yum:** Package manager for Red Hat-based distributions.  
Example: `'yum install package'`



- 34. systemctl:** Control system services.  
Example: ``systemctl start service``
- 35. journalctl:** Query and display system logs.  
Example: ``journalctl -u service``
- 36. grep:** Search for a pattern in files.  
Example: ``grep "pattern" file.txt``
- 37. sed:** Stream editor for text manipulation.  
Example: ``sed 's/old/new/' file.txt``
- 38. awk:** Text processing and pattern scanning.  
Example: ``awk '{print $1}' file.txt``
- 39. sort:** Sort lines of text files.  
Example: ``sort file.txt``
- 40. uniq:** Report or omit repeated lines.  
Example: ``uniq file.txt``
- 41. tar:** Manipulate tape archives.  
Example: ``tar -xvf archive.tar``
- 42. gzip:** Compress files.  
Example: ``gzip file.txt``
- 43. gunzip:** Decompress files.  
Example: ``gunzip file.txt.gz``
- 44. ssh-keygen:** Generate SSH keys.  
Example: ``ssh-keygen -t rsa``
- 45. ssh-copy-id:** Copy SSH public key to a remote host.  
Example: ``ssh-copy-id user@host``
- 46. mount:** Mount a file system.  
Example: ``mount /dev/sdb1 /mnt``





**47. umount:** Unmount a file system.

Example: ``umount /mnt``

**48. lsblk:** List block devices.

Example: ``lsblk``

**49. fdisk:** Partition table manipulator for disks.

Example: ``fdisk /dev/sdb``

**50. date:** Display the current date and time.

Example: ``date``

**51. echo:** Print text or variables.

Example: ``echo "Hello, World!"``

**52. tee:** Read from standard input and write to standard output and files.

Example: ``echo "Hello" | tee file.txt``

**53. nc:** Read and write data across network connections.

Example: ``echo "Hello" | nc host port``

**54. basename:** Strip directory and suffix from filenames.

Example: ``basename /path/to/file.txt``

**55. dirname:** Strip last component from file name.

Example: ``dirname /path/to/file.txt``

**56. uptime:** Display system uptime.

Example: ``uptime``

**57. whoami:** Print the current username.

Example: ``whoami``

**58. su:** Switch user.

Example: ``su username``



**59. sudo:** Execute a command as a superuser.

Example: `'sudo command'`

**60. useradd:** Create a new user account.

Example: `'useradd username'`

**61. passwd:** Change user password.

Example: `'passwd username'`

**62. groupadd:** Create a new group.

Example: `'groupadd groupname'`

**63. usermod:** Modify user account.

Example: `'usermod -aG groupname username'`

**64. groups:** Display group membership for a user.

Example: `'groups username'`

**65. chgrp:** Change group ownership of a file or directory.

Example: `'chgrp groupname file.txt'`

**66. wc:** Count lines, words, and characters in files.

Example: `'wc file.txt'`

**67. ln:** Create hard links to files.

Example: `'ln /path/to/file link'`

**68. free:** Display free and used memory in the system.

Example: `'free -h'`

**69. echo:** Print text or variables.

Example: `'echo "Hello, World!"'`

**70. uptime:** Display system uptime.

Example: `'uptime'`

**71. whereis:** Locate the binary, source, and manual page files for a command.

Example: `'whereis ls'`



- 72. cmp:** Compare two files byte by byte.  
Example: ``cmp file1.txt file2.txt``
- 73. diff:** Compare files line by line.  
Example: ``diff file1.txt file2.txt``
- 74. curl:** Transfer data from or to a server.  
Example: ``curl https://example.com``
- 75. sort:** Sort lines of text files.  
Example: ``sort file.txt``
- 76. find:** Search for files and directories.  
Example: ``find /path/to/search -name "pattern"``
- 77. tar:** Archive files and directories.  
Example: ``tar -czvf archive.tar.gz files/``
- 78. grep:** Search for a pattern in a file.  
Example: ``grep "pattern" file.txt``
- 79. sed:** Stream editor for text manipulation.  
Example: ``sed 's/old/new/' file.txt``
- 80. awk:** Text processing and pattern scanning.  
Example: ``awk '{print $1}' file.txt``
- 81. wc:** Count lines, words, and characters in files.  
Example: ``wc file.txt``
- 82. chmod:** Change the permissions of a file or directory.  
Example: ``chmod 755 file.txt``
- 83. chown:** Change the ownership of a file or directory.  
Example: ``chown user:group file.txt``
- 84. head:** Display the first few lines of a file.



Example: `'head file.txt'`

**85. tail:** Display the last few lines of a file.

Example: `'tail file.txt'`



**86. top:** Display system resource usage and running processes.

Example: `'top'`

**87. ps:** Display currently running processes.

Example: `'ps aux'`

**88. kill:** Terminate a process.

Example: `'kill PID'`

**89. shutdown:** Shutdown or restart the system.

Example: `'shutdown -h now'`

**90. reboot:** Reboot the system.

Example: `'reboot'`

**91. ifconfig:** Display network interface information.

Example: `'ifconfig'`

**92. ip:** Show or manipulate routing, devices, policy routing, and tunnels.

Example: `'ip addr show'`

**93. ping:** Send ICMP echo requests to a host.

Example: `'ping google.com'`

**94. traceroute:** Print the route packets take to a network host.

Example: `'traceroute google.com'`

**95. netstat:** Print network connections, routing tables, and interface statistics.

Example: `'netstat -tuln'`

**96. iptables:** Administration tool for IPv4/IPv6 packet filtering and NAT.

Example: `'iptables -L'`

**97. hostname:** Show or set the system's host name.

Example: `'hostname'`



**98. uname:** Print system information.

Example: ``uname -a``

**99. history:** View command history.

Example: ``history``

**100. exit:** Exit the current shell or terminal.

Example: ``exit``

**Thank you**

**CLOUD DEVOPS LOKESHTECH**

=====END=====