Interview Tips Day 1: Linux for DevOps - Some Common Questions with Basic Commands

1. What is Linux?

Linux is like the heartbeat of various operating systems, providing customization, stability, security, and cost-effectiveness. It's an open-source operating system kernel that forms the backbone of diverse distributions.

2. Why should I use Linux for DevOps?

Linux is a powerhouse for DevOps because it offers:

- **Open Source and Flexibility:** Customization and adaptability to meet specific needs.
- **Powerful Command-Line Interface (CLI):** Enables quick automation and execution of tasks.
- **Rich Toolset:** Comprehensive tools for development and DevOps readily available.
- **Stability and Reliability:** Known for reliability, crucial for CI/CD and infrastructure management.
- **Containerization and Orchestration:** Dominant platform for Docker and Kubernetes, ensuring scalability.
- **Community Support and Documentation:** Extensive resources and support from an active community.
- **Compatibility with Cloud Services:** Aligns seamlessly with cloud platforms for integration and management.
- **Security and Performance:** Renowned for security features and optimal performance.
- **Cost-Effectiveness:** Often free to use, reducing infrastructure costs for businesses.

3 What are the commands used in Linux for DevOps?

Time to embark on your command-line journey! Here are your allies:

1. **ls:** Lists contents of a directory.
   - Example: `ls`, `ls -l`, `ls /path/to/directory`
2. **cd:** Changes the current directory.
   - Example: `cd directory_name`, `cd /path/to/directory`
3. **pwd:** Prints the current working directory.
   - Example: `pwd`
4. **mkdir:** Creates a new directory.

- Example: `mkdir new_directory`

5. **rm:** Removes files or directories.
   - Example: `rm file_name`, `rm -r directory_name`
6. **cp:** Copies files or directories.
   - Example: `cp file1.txt file2.txt`, `cp -r directory1 directory2`
7. **mv:** Moves or renames files or directories.
   - Example: `mv file1.txt file2.txt`, `mv file1.txt /path/to/new/location`
8. **grep:** Searches for patterns within files.
   - Example: `grep "pattern" file.txt`, `grep -r "pattern" /path/to/directory`
9. **find:** Searches for files and directories.
   - Example: `find /path/to/start -name "filename"`
10. **chmod:** Changes file permissions.
    - Example: `chmod 644 file.txt`, `chmod +x script.sh`
11. **chown:** Changes file ownership.
    - Example: `chown user:group file.txt`
12. **tar:** Archives files.
    - Example: `tar -czvf archive.tar.gz directory`
13. **gzip/gunzip:** Compresses/decompresses files.
    - Example: `gzip file.txt`, `gunzip file.txt.gz`
14. **ssh:** Securely connects to remote servers.
    - Example: `ssh username@hostname`
15. **scp:** Securely copies files between systems.
    - Example: `scp file.txt username@hostname:/remote/directory`
16. **rsync:** Synchronizes files and directories between systems.
    - Example: `rsync -avz source_directory/ target_directory`
17. **df:** Displays disk space usage.

- Example: `df -h`

18. **du:** Shows directory space usage.
    - Example: `du -sh /path/to/directory`
19. **top/htop:** Displays system processes and resource usage.
    - Example: `top`, `htop`
20. **ps:** Shows current running processes.
    - Example: `ps aux`
21. **kill:** Terminates processes.
    - Example: `kill PID`, `killall process_name`
22. **wget/curl:** Downloads files from the internet.
    - Example: `wget URL`, `curl -O URL`
23. **sed:** Streams editor for text manipulation.
    - Example: `sed 's/old_text/new_text/g' file.txt`
24. **awk:** Powerful scripting language for text processing.
    - Example: `awk '{print $1}' file.txt`
25. **journalctl:** Views system logs and journal entries.
    - Example: `journalctl -u service_name`
26. **systemctl:** Controls systemd services.
    - Example: `systemctl start service_name`, `systemctl status service_name`
27. **ifconfig/ip:** Network configuration and information.
    - Example: `ifconfig`, `ip addr show`
28. **ping:** Checks connectivity to a host.
    - Example: `ping domain.com`
29. **traceroute/tracert:** Traces the path packets take to a destination.
    - Example: `traceroute domain.com`
30. **nc/netcat:** Networking utility for reading and writing data across network connections.
    - Example: `nc -l -p port_number`, `nc remote_host port_number`

4 Explain the difference between Linux distributions.

- Understanding various distributions like Ubuntu, CentOS, Debian, and their specific use cases.

5 How do you navigate the Linux file system?

- Demonstrating usage of basic commands like ls, cd, pwd, mkdir, rm to navigate, view contents, create, and remove directories/files.

6 What is a shell?

- Definition of a shell, common shells (Bash, Zsh), and familiarity with basic shell scripting concepts.

7 Explain file permissions in Linux.

- Understanding permissions (chmod) - read, write, execute for owner/group/others and the numeric representation of permissions.

8 How do you find files in Linux?

- Usage of find and grep commands to search for files and patterns within files.

9 What is SSH, and how is it used?

- Definition of Secure Shell (SSH), its usage for secure remote access, file transfer (scp), and executing commands on remote servers.

10 What is a package manager in Linux?

- Understanding package management systems like apt-get, yum, or dnf, and their role in installing, updating, and removing software packages.

11 How do you check system resources in Linux?

- Usage of commands like top, htop, df, du to monitor system resource utilization, disk space, and directory usage.

12 Explain Linux networking commands/tools.

- Familiarity with commands like ifconfig, ip, ping, traceroute, netstat, and their usage in network configuration, testing connectivity, and troubleshooting.