1. The reboot command is used to restart or reboot a computer or system, it initiates the process of shutting down the OS and then automatically starts it up again. Only tape reboot
2. The shutdown command is used to gracefully shut down or schedule the shutdown of a system. It allows you to specify when to shut down the system and can also send notification to logged-in users. To shut down the system immediately: sudo shutdown now; to provide a notification message to the users logged into the system before shutting down: sudo shutdown -k “notification message”;
3. To cancel a scheduled system shut down using the shutdown command: shutdown -c. note that you need to cancel the shutdown before the scheduled time.
4. The best practices when using the reboot and shutdown command are: save your work and close all open applications; - inform users using system notifications, messages or communication channels to minimize disruptions and allow them to save their work; - specify shutdown/reboot time to allows users to see a warning message.
5. The difference between using the reboot and shutdown command to restart the system is that: the reboot command is used for an immediate system restsrt, while the shutdown command allows you to schedule a shutdown or restart in the future to allow users to save their work and exit applications gracefully before the system restart.
6. To schedule a shutdown for a specific time: sudo shutdown -h hh:mm
7. The free command is used to display information about system’s memory usage, including both physical RAM and swap space.
8. To display information about system memory usage: free -h
9. The df command is used to display information about the disk space usage on your system
10. To check disk space usage and available space on file system: df -h
11. The top command ia a powerful utility that provides real-time monitoring and management of system resources, it displays an interactive, dynamic view of system processes, CPU usage, memory usage, and other system statistics.
12. To monitor system processes, resource usage and CPU usage : top
13. The kill command is used to send signals to processes, allows to manage and control running processes. To terminate processes gracefully, you can identify the process ID(PID) of the process: ps aux ! grep example-process; then execute: kill PID. Some signal are SGTERM (15) terminate the process gracefully; SIGKILL (9) and SIGINT (2) to interrupt the process
14. The difference between kill -9 and kill -15 is that: kill -9 PID (forcelly terminate the process immediately without allowing cleanup) and kill -15 PID (to terminate the process gracefully)
15. The du command is used to estimate disk usage for files and directories, it provides information about the size of files and directories, helping ypu identify where disk space is being utilized. Options -h (displays the sizes in human-readable format); -s (display only the total size of the specified directory or file); -c (display the gran total size of all directories or files); -a (display sizes for all files and directories including hidden ones); -d depth> (limits the depth of the directory tree to be display).
16. To estimate disk usage of files and directory: du -h -d 1 /directory
17. To troubleshoot a slow server:

* Check server resources: use commands top, htop or free to monitor resource usage in real-time
* Review system logs: to verify if will have any error messages or warnings related to performance issues
* Identify CPU-intensive processes: use top or htop
* Check disk I/O: monitor disk I/O using iotop or iostat
* Network analysis: use netstat, iftop or tcpdump to analyze network activity
* Analyze memory usage: use free or top
* Review application-specific logs: if the slow server is running specific applications, review the respective application logs for any errors
* Check for security issues
* Evaluate whether the server’s hardware meets the requirements for the workload it’s handling
* Optimize server configuration
* Implement monitoring tools or solutions to continuously monitor server performance.

1. The pipe symbol in linux command line operations is used to connect the output of one command to the input of another command. Example cat data.dat ! sort ! uniq -c ! sort -nr (this command reads the contents of data.dat, sorts the lines, counts the occurrences of each unique line using uniq -c and then sorts the result in descending order of frequency
2. The grep command is used for searching and filtering text based on patterns, it allows you to search for specific strings or patterns within files or the output of other commands. Options: -I (ignore case distinctions in the search); -r (search files recursively in directories and subdirectories); -l (only display filenames containing matches; -n (display line numbers along with matching lines); -v (display lines that don’t match the pattern); -w (match only whole words, not partial matches).
3. The cut command is used to extract specific portions, such as columns or fields, from lines of text or files, it allows you to specify a delimiter and select the desired portions. Options: -d=delim (specify the delimiter used to separate fields in the input); -f=list (select specific fields or columns to extract. Example
4. cut -d ‘ ‘ -f 2 data.txt (extract the second column of data.txt); cut -d ‘ ‘ -f 1,3 data.txt (extract the first and third columns of data.txt); cut -d ‘ ‘ -f 2-4 data.txt (extract 2, 3, and 4 columns for data.txt)
5. grep “pattern” data.txt ! cut -d ‘ ‘ -f 2 (this command display the second field of each line that matches the specified pattern
6. the awk command is a powerful text processing tool that allows you to manipulate and analyze structured text data. Example awk `{print $2}` data.txt (select the second field from each line)
7. to combine the grep and akw commands using a pipe: grep “armand” data.txt ! akw `{print $1, $2 “:”}`
8. example to use grep, cut and akw command: cat data.txt ! grep “armand” ! cut -d ‘ ‘ -f 3 ! awk ‘{ print $1 \* 2}’
9. see 20), 19) and 23) to see the flags of cut, grep and awk commands
10. the set command is used to modify the shell’s environment variables, command-line arguments, and other shell options, it allows you to view and change various settings related to the shell’s behavior and configuration.
11. Set -o (displays a list of the current shell options and their status); -p (turned on whenever the real and effective user ids don’t match); -t (exit after reading and executing one command)
12. The find command is a powerful utility that allows you to search for files and directories in a directory hierarchy based on various criteria; syntax find path expresion
13. Some options or flags of the find command: -name or -iname (find directly by nme); -type (searching by type, d for directory and f for file); combine search criteria: -a (and), -o (or), ! (not); applying action: -exec to executing, -delete to deleting file, -print to printing file information; controlling search depth: -maxdepth and -mindepth
14. To find a file in a specific directory and subdirectories: find . -name “example.txt” (find the file named example.txt in the current directory and its subdirectories)
15. Find /home -name “exam\*.txt” -type f -delete (find all the files start by exam with the extension .txt within the /home directory and its subdirectories and delete its)
16. The crontab command is used to create, edit, and manage cron jobs
17. To view the current user’s cron jobs: run crontab -l (each line represents a separate cron job entry)
18. Commonly options of crontab command are: -l to display the current user’s cron jobs; -e to edit the current user’s cron jobs; -r to remove all cron job
19. A cron job entry in the crontab file follow a specific format: \* \* \* \* \* command

* 1st \*: represents the minute of hour when the job should run, valid values range from 0 to 59
* 2nd \*: specified the hour of the day when the job should run, valid values range from 0 to 23
* 3rd \*: the day of the month when the job should run, valid values range from 1 to 31
* 4e \*: the month of the year when the job should run, valid values range from 1 to 12
* 5e \*: the day of the week (0-7) when the job should run, 1=Monday
* Command: specifies the command or script to be executed as part of the cron job

1. To edit an existing cron job using the crontab command:

* Open the cron table for editing: crontab -e
* Locate the cron job you want to edit
* Make the necessary changes to the cron job entry using the editor
* Save and exit the editor
* Verify the changes: crontab -l

1. To remove a cron job using the crontab command:

* Open the cron table for editing: crontab -e
* Locate the cron job you want to remove
* Remove the entire line containing the cron job entry that you want to delete
* Save and exit the editor

1. To schedule a task to run every week:

* Crontab -e
* \* \* \* \* 7 command (a run the command all hour Saturday)
* Save and exit

1. To reboot or shutdown a server automatically at 9h00 am every Sunday:

* Crontab -e
* 0 9 \* \* 0 /sbin/shutdown -r now
* Save and exit

1. The tree command provide a hierarchical view of the directory structure on your system. Options: -L limits the depth of the tree to the specified level (tree -L 3); -a shows all files, including hidden files that start with a dot; -d displayd only directories omitting files; -f prints the full path prefix for each file; -I <pattern> excludes files and directories that match the specified pattern (tree -I “\*.txt” will exclude all files with the .txt extension)