Module 16

Linux server - Operate running systems

• Assignment Level Basic to Advance

1. What is PID ?

Ans:

PID stands for "Process Identifier." It is a unique numerical identifier assigned to each running process in an operating system. The PID is used by the operating system to track and manage processes.

When a program or application is executed, the operating system creates a process to run that program. This process is assigned a PID, which allows the operating system to keep track of various attributes and status of the process. These attributes include the parent process ID (PPID), memory usage, CPU time, state (running, sleeping, etc.), and more.

PIDs are important for several reasons:

Process Management: PIDs help the operating system manage and control processes efficiently. For instance, the OS can start, pause, resume, or terminate a process using its PID.

Resource Allocation: PIDs assist in managing the allocation of system resources (e.g., CPU time, memory) to different processes, preventing conflicts and ensuring efficient usage.

Error Handling: PIDs are useful for identifying processes in case of errors or issues within the system. Logs and error messages often include PIDs to help diagnose problems.

Interprocess Communication (IPC): PIDs are used for communication between processes, enabling processes to interact with each other and share data.

Each time a process is created, it is assigned a new, unique PID, which is typically a non-negative integer. PIDs are usually recycled once a system wraps around the maximum PID value, and old PIDs are reused for new processes. However, modern systems support large numbers of PIDs, reducing the likelihood of PID reuse during a system's uptime.

2. What is PPID?

Ans:PPID stands for "Parent Process Identifier." It is a unique numerical identifier assigned to the parent process of a particular running process in an operating system.

Every process in an operating system is created by another process, known as the "parent" process. The parent process spawns or creates a new process, which becomes its child. The child process inherits various attributes and properties from its parent, including the parent process ID (PPID). The PPID is used to identify the process that created or spawned the current process.

The PPID is essential for process management and control for several reasons:

Process Hierarchy: Understanding the parent-child relationship allows for the creation of process hierarchies, where processes can spawn other processes, forming a tree-like structure.

Parental Control: The parent process can manage and control its child processes, monitor their behavior, and even terminate them if necessary.

Process Termination: If a parent process terminates, its child processes may be adopted by the system or reassigned to a new parent process.

Resource Allocation: The parent process often determines the allocation of resources and permissions for its child processes.

By examining the PPID, one can trace the ancestry of a process back to its origin, revealing the process that initiated its creation. This information is valuable for troubleshooting, understanding system behavior, and managing processes effectively.

3. What is the use of “ ps “ command ?

Ans:The ps command in Unix and Unix-like operating systems (including Linux) is used to provide information about the currently running processes. It is a powerful utility that allows users to monitor and manage processes on the system. The term "ps" stands for "process status."

Here are some common uses and functionalities of the ps command:

Display Process Information:

The primary use of ps is to display information about the processes running on the system, including their process IDs (PIDs), CPU and memory usage, parent process IDs (PPIDs), status, and more.

List All Processes:

ps without any options lists the processes associated with the current terminal session.

Display Detailed Process Information:

Using various options (-f, -l, -e, etc.), you can get detailed information about processes, including the full command line, the user who started the process, start time, CPU and memory usage, and more.

Show All Users' Processes:

ps -e or ps aux displays all running processes on the system for all users.

Show Parent-Child Relationships:

ps -f displays the parent-child process relationships, showing the parent process ID (PPID) for each process.

Sort and Customize Output:

ps allows you to sort and customize the output to display the information in the desired format or order.

Monitor Specific Processes:

By using ps -p <PID>, you can monitor a specific process by providing its PID.

Continuous Monitoring:

The ps command can be used in combination with other tools (like watch) to continuously monitor processes and their statuses.

Filter and Search for Processes:

ps allows you to filter processes based on various criteria (user, command, etc.) using options like -U, -C, -G, etc.

Overall, the ps command is a fundamental and versatile tool for system administrators and users to monitor and manage processes effectively on a Unix-based operating system.

4. What is the use of “ps aux “ command ?

Ans:The ps aux command is a widely used Unix/Linux command that provides a detailed listing of all processes currently running on the system. Here's a breakdown of the ps aux command and its uses:

ps: This is the command itself, which stands for "process status." It is used to display information about active processes.

aux: These are options passed to the ps command to customize the output:

a: Lists processes of all users. By default, ps only shows processes associated with the current terminal.

u: Displays a detailed output format that includes the user, CPU usage, memory usage, start time, and command line.

x: Lists processes not associated with a terminal (daemons). This is useful for showing background processes.

Putting it all together, ps aux displays a comprehensive list of all processes running on the system, providing extensive information about each process, including the user, CPU and memory usage, start time, and more.

Usage and Examples:

General Process Listing: ps aux lists all processes on the system, including those running in the background (daemons).

Process Monitoring: Administrators often use ps aux to monitor processes and identify any abnormal CPU or memory usage.

Finding Specific Process: You can combine ps aux with tools like grep to filter for specific processes. For example, ps aux | grep "process\_name" will display processes containing "process\_name" in their command line.

Identifying Resource Usage: The ps aux output provides valuable information on CPU and memory usage, helping administrators identify resource-intensive processes that may need optimization or termination.

Overall, ps aux is a powerful tool for understanding the system's current process status, making it an essential part of system monitoring, debugging, and management.

5. What is the use of “ tops “ command ?

6. Which command is used to change priority value ?

Ans:The nice and renice commands are used to change the priority (niceness) of a process in Unix and Unix-like operating systems, including Linux. These commands allow you to adjust the scheduling priority of a process, influencing its CPU time allocation.

Here's an overview of both commands:

nice Command:

The nice command is used to launch a new process with an adjusted scheduling priority (niceness).

Lowering the niceness value gives the process a higher priority, making it use more CPU time.

Raising the niceness value makes the process less of a priority, allowing other processes to get more CPU time.

Usage:

nice -n <value> <command>

Example:

nice -n -10 ./my\_program # Starts "my\_program" with higher priority (lower niceness)

renice Command:

The renice command is used to change the priority (niceness) of an existing process.

Similar to nice, lower niceness values give the process higher priority, while higher niceness values reduce its priority.

Usage:

renice [-n] <niceness\_value> -p <PID>

Example:

renice -n -5 -p 12345 # Changes the niceness of the process with PID 12345 to -5

You can also specify the process by name using pgrep:

renice -n -10 $(pgrep my\_process) # Changes the niceness of the process named "my\_process" to -10

Both commands are useful for adjusting the priority of processes, especially in scenarios where you want to manage CPU time allocation and optimize system performance. The renice command is particularly valuable for adjusting the priority of existing processes.

7. What is the use of “jobs” command ?

Ans:The jobs command is used in Unix-like operating systems, including Linux, to display the status of background and suspended jobs associated with the current shell session. A "job" typically refers to a process or a group of processes that have been initiated from the same shell session.

Here's a summary of the uses of the jobs command:

Displaying Running and Suspended Jobs:

jobs displays a list of all background and suspended jobs initiated within the current shell session.

It provides information such as job number, status, and command associated with each job.

Monitoring Background Jobs:

When a command or process is executed in the background (e.g., using & at the end of a command), it runs independently, allowing the user to continue entering other commands in the foreground.

The jobs command helps monitor the progress and status of background jobs.

Managing Background Jobs:

The jobs command allows you to manage background jobs by providing job numbers that can be used with other commands like bg, fg, and kill to control the jobs.

Bringing Jobs to the Foreground (fg):

The fg command is used to bring a background job to the foreground and continue its execution in the foreground.

Putting Jobs in the Background (bg):

The bg command is used to put a stopped or suspended job in the background, allowing it to continue its execution without interfering with the current shell session.

Signaling Jobs (kill):

The kill command, used in combination with job numbers, allows you to send signals to jobs, enabling you to terminate, pause, or resume job execution.

Overall, the jobs command is a crucial tool for managing and monitoring background and suspended jobs, enabling efficient multitasking and control over running processes in a Unix-like environment.

8. What is the use of grep command ?

Ans:The grep command is a powerful utility in Unix-like operating systems, including Linux, that is used for searching and pattern matching within files or command output. "grep" stands for "Global Regular Expression Print."

Here are the key uses of the grep command:

Searching for a Pattern in Files:

grep allows you to search for a specified pattern (a regular expression or simple string) in one or more files. It prints lines that match the pattern.

Displaying Matching Lines:

By default, grep displays all lines in a file that contain the specified pattern.

Using Regular Expressions:

Regular expressions (regex) allow for more complex pattern matching. grep supports various regex patterns to search for specific types of text.

Case-Insensitive Search:

You can use the -i option with grep to perform a case-insensitive search.

Showing Line Numbers:

Use the -n option to display line numbers along with the matching lines.

Displaying Inverse Matches:

The -v option inverts the match, displaying lines that do not match the pattern.

Displaying Count of Matches:

The -c option counts the number of lines that match the pattern rather than displaying the lines themselves.

Recursive Search:

With the -r or -R option, grep can perform a recursive search through directories and subdirectories.

Suppressing Error Messages:

The -s option suppresses error messages, which is useful in combination with other commands or scripts.

Usage Examples:

Basic Search:

grep "pattern" filename

Case-Insensitive Search:

grep -i "pattern" filename

Display Line Numbers:

grep -n "pattern" filename

Invert Match:

grep -v "pattern" filename

Count Matches:

grep -c "pattern" filename

Recursive Search:

grep -r "pattern" directory

The grep command is a versatile tool for finding specific patterns or text within files and is commonly used in various scripts, commands, and pipelines to process and analyze data efficiently.

9. What is system?

Ans:A "system" generally refers to a collection of interconnected or interdependent components or parts that work together to perform a specific function or purpose. In computing and technology, the term "system" can refer to various entities:

Computer System:

In the context of computing, a computer system typically refers to a combination of hardware, software, and user input/output devices working together to execute tasks or processes.

Operating System:

The operating system (OS) is a critical software component of a computer system that manages hardware resources, provides user interfaces, and enables the execution of applications and processes.

Software System:

A software system comprises multiple software components, modules, or programs that work together to achieve specific objectives. These components can include applications, libraries, and frameworks.

Information System:

An information system refers to a combination of hardware, software, data, people, and procedures that are organized to collect, process, store, and distribute information in an organization to support decision-making and control.

Ecosystem:

In a broader sense, "system" can refer to a complex and interrelated set of elements within a specific environment, such as an ecological system, economic system, or social system.

Each type of system has its own set of components, rules, interactions, and objectives. The study and understanding of systems are fundamental in various fields, including computer science, engineering, biology, sociology, economics, and more, to analyze and design efficient and effective solutions.

10.What is daemons?

11.I want to check the service status for” sshd”, which will help me?

Ans:To check the service status for "sshd" (SSH daemon) on a Linux system, you can use the systemctl command. systemctl is a powerful command-line utility used to control the systemd system and service manager.

Here's how you can check the status of the SSH daemon service:

Using systemctl status:

systemctl status sshd

This command will provide detailed information about the status of the SSH daemon service, including whether it's running, any errors, and other relevant details.

Using service (an alternative method, especially on older systems):

service sshd status

On older systems or systems not using systemd, you can use the service command to check the status of the SSH daemon service.

These commands will give you information about whether the SSH daemon service is active (running) or inactive (stopped), and other details like the process ID, memory usage, and more.

Make sure to run these commands with appropriate privileges (e.g., using sudo) if needed, especially if you're not logged in as the system administrator or a user with administrative privileges

12.How to stop and start services in terminal?

Ans:To stop and start services in a terminal on a Linux system, you can use the systemctl command. The systemctl command is used to control services and the systemd system and service manager. Here's how you can stop and start services:

Stopping a Service:

sudo systemctl stop service\_name

Replace service\_name with the actual name of the service you want to stop. This command will stop the specified service.

Starting a Service:

sudo systemctl start service\_name

Replace service\_name with the actual name of the service you want to start. This command will start the specified service.

Restarting a Service:

sudo systemctl restart service\_name

Replace service\_name with the actual name of the service you want to restart. This command will stop and then start the specified service.

Reloading a Service (Reloads Configuration):

sudo systemctl reload service\_name

Replace service\_name with the actual name of the service you want to reload. This command will reload the configuration of the specified service.

Enabling a Service (Starts on Boot):

sudo systemctl enable service\_name

Replace service\_name with the actual name of the service you want to enable. This command will configure the specified service to start automatically at boot.

Disabling a Service (Stops Starting on Boot):

sudo systemctl disable service\_name

Replace service\_name with the actual name of the service you want to disable. This command will prevent the specified service from starting automatically at boot.

Make sure to replace service\_name with the actual service name you want to manipulate (e.g., sshd for the SSH daemon). Additionally, use sudo to run the commands with superuser privileges, as managing services typically requires administrative access.

13.What is the use of openSSH ?

Ans:OpenSSH, which stands for Open Secure Shell, is a suite of network communication protocols and tools primarily used for secure remote access to systems over a network. It provides a secure way to log into and control a remote machine, as well as to transfer files securely between systems. OpenSSH is widely used and is considered the standard for secure remote access on Unix and Unix-like operating systems, including Linux.

Here are some key uses of OpenSSH:

Remote Shell (SSH):

OpenSSH allows users to securely access and manage remote machines over a network using the SSH protocol (SSH-2 by default). Users can log in to a remote server and execute commands just as if they were physically present at the machine.

Secure File Transfer (SFTP):

OpenSSH provides the SFTP (Secure File Transfer Protocol) subsystem, which enables secure file transfers between systems over an encrypted connection. SFTP is commonly used for uploading, downloading, and managing files on remote servers securely.

Secure Copy (SCP):

SCP is a command-line tool provided by OpenSSH that allows for secure copying of files between a local and a remote machine or between two remote machines. It encrypts the data during transfer, providing a secure way to copy files.

Tunneling and Port Forwarding:

OpenSSH can set up encrypted tunnels between machines, allowing secure transmission of data between them. This is useful for securely accessing services (e.g., web servers) on a remote machine that might not be directly accessible from the user's local machine due to network restrictions.

Public Key Authentication:

OpenSSH supports public key-based authentication, which is more secure than traditional password-based authentication. Users can generate key pairs (public and private keys) and securely authenticate themselves using these keys, eliminating the need for passwords.

Remote Command Execution:

OpenSSH allows users to execute commands remotely on a machine using the ssh command. This is useful for automating tasks, scripts, and system administration tasks on remote systems.

Secure Connection for X11:

OpenSSH can securely forward X11 sessions, allowing users to run graphical applications on a remote server and display them locally.

OpenSSH is a fundamental tool for system administrators, developers, and anyone needing secure remote access and file transfers. It's a critical component in securing network communication and is widely adopted due to its robust security features and versatility.

14.Which command is used to generate key in linux ?

Ans:To generate a cryptographic key pair (public and private keys) in Linux, you can use the ssh-keygen command. This command is used to create SSH keys for secure authentication, primarily using RSA or DSA algorithms. Here's a basic usage and example:

Basic Usage:

ssh-keygen -t <key\_type> -b <key\_length> -C "your\_comment"

<key\_type>: The type of key to create, typically rsa (for RSA algorithm) or dsa (for DSA algorithm).

<key\_length>: The length of the key in bits, usually 2048 for RSA keys.

"your\_comment": An optional comment to describe the key (e.g., your email or a meaningful identifier).

Example (RSA key with 2048 bits):

ssh-keygen -t rsa -b 2048 -C "your\_email@example.com"

This will generate a new RSA key pair (public and private keys) with a length of 2048 bits and the specified comment.

Generating DSA Key:

ssh-keygen -t dsa -b 1024 -C "your\_comment"

Replace dsa with rsa or ecdsa for other key types.

Custom Key Location:

By default, ssh-keygen will store the generated keys in ~/.ssh/. You can specify a custom path using the -f option:

ssh-keygen -t rsa -b 2048 -C "your\_email@example.com" -f /path/to/custom\_key

This will generate the keys with the custom path and file name.

The ssh-keygen command will prompt you for a passphrase to enhance the security of the key. The passphrase acts as an additional layer of protection, making it more secure but also requiring it each time you use the key.

After running ssh-keygen, you'll have a new SSH key pair (public and private keys) generated and stored in the specified location, ready to be used for secure authentication.

15.Which commannd is used to copy ssh key?

AnS: To copy an SSH public key from your local machine to a remote server, you can use the ssh-copy-id command. This command securely installs your public key in the ~/.ssh/authorized\_keys file on the remote server, allowing you to authenticate to the remote server using your private key.

Here's the basic usage of ssh-copy-id:

ssh-copy-id user@remote\_server

Replace user with the username you want to use to log in to the remote server, and replace remote\_server with the IP address or hostname of the remote server.

The ssh-copy-id command will prompt you for the password of the user on the remote server. Once you provide the password, it will copy your public key to the appropriate location on the remote server and set the correct permissions.

Example:

ssh-copy-id john@example.com

This will copy the SSH public key of the current user to the user john on the remote server example.com.

Make sure you have generated your SSH key pair (public and private keys) using ssh-keygen before using ssh-copy-id. The ssh-keygen command generates the keys, and ssh-copy-id copies the public key to the remote server for authentication.

16.How do we prohibit the root user from logging in using ssh?

Ans:

17.How do we prohibit password authentication using ssh?

Ans:To prohibit password authentication and enforce the use of SSH key-based authentication in SSH, you need to modify the SSH daemon configuration file (sshd\_config) on the remote server. Here are the steps to achieve this:

Open the sshd\_config file for editing:

sudo nano /etc/ssh/sshd\_config

Replace nano with your preferred text editor if you have one.

Find the Password Authentication setting:

Look for the line that begins with PasswordAuthentication. It will usually be set to yes.

Change Password Authentication setting:

Change PasswordAuthentication to no to disable password authentication.

PasswordAuthentication no

Save and close the file:

In nano, press Ctrl + O to save the file, then Ctrl + X to exit.

Restart SSH service:

Restart the SSH service for the changes to take effect.

sudo systemctl restart sshd

Now, SSH will only allow key-based authentication, and password authentication will be disabled.

Please ensure that you have already set up SSH key-based authentication and tested it successfully before disabling password authentication to avoid being locked out of the server. Also, it's important to keep your private keys secure and back them up in a safe location.

18.Where we find general logs ?

Ans:In Linux-based systems, general system logs are typically stored in the /var/log directory. This is where various log files are located, each serving a specific purpose and containing information related to system operations, processes, applications, errors, and more.

Here are some common log files found in the /var/log directory:

auth.log or secure:

Contains authentication-related messages, including login attempts, su (switch user) activity, and more.

Location: /var/log/auth.log or /var/log/secure (location can vary based on the distribution).

syslog:

General system messages and logging from various system components.

Location: /var/log/syslog.

kern.log:

Kernel-related messages, including hardware-related events and errors.

Location: /var/log/kern.log.

messages:

General system messages, often containing information from daemons and services.

Location: /var/log/messages.

dmesg:

Kernel ring buffer, providing detailed information about the system's hardware and drivers.

Can be viewed using the dmesg command.

auth.log (older Debian-based systems) or secure (older Red Hat-based systems):

Deprecated, but still found in some older systems, similar to the current auth.log and secure.

cron:

Contains logs for cron jobs (scheduled tasks).

Location: /var/log/cron.

mail.log:

Logs related to email servers and mail delivery.

Location: /var/log/mail.log.

boot.log:

Logs related to the system boot process.

Location: /var/log/boot.log.

These logs can be viewed using various commands like cat, tail, less, or grep in the terminal. For example:

cat /var/log/syslog

tail -f /var/log/auth.log

less /var/log/messages

Additionally, some applications and services may have their own specific log files stored in the /var/log directory or subdirectories. It's important to check the documentation of the specific application or service for details on where their logs are located.

19.Where we find secure logs ?

Ans:The "secure" log file, which typically contains authentication-related messages such as login attempts and authentication failures, is often found on systems that use the systemd logging system. However, the specific location of the "secure" log file can vary based on the Linux distribution. Here are some common locations for the "secure" log file:

Ubuntu, Debian, and related distributions:

Location: /var/log/auth.log

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/secure

To view the contents of the "secure" log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/auth.log # For Ubuntu, Debian, and related distributions

tail -f /var/log/secure # For CentOS, Fedora, Red Hat, and related distributions

less /var/log/auth.log # To view the log contents using less

Replace the path with the appropriate path for your specific distribution. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to authentication and security.

20.Where we find mail log ?

Ans:The mail log, which contains logs related to email servers and mail delivery, is often located in the /var/log directory on Unix-like systems. The specific name of the mail log file and its location can vary based on the mail server software being used and the Linux distribution. Here are some common locations for the mail log file:

Postfix (a popular mail server):

Location: /var/log/mail.log (or /var/log/maillog on some systems)

Sendmail:

Location: /var/log/maillog

Exim:

Location: /var/log/exim4/mainlog (on Debian-based systems, like Ubuntu)

Location: /var/log/exim/mainlog (on Red Hat-based systems, like CentOS)

Dovecot (a popular IMAP and POP3 server):

Location: /var/log/mail.log (or /var/log/maillog on some systems)

To view the contents of the mail log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/mail.log # For Postfix or Dovecot

tail -f /var/log/maillog # For Sendmail

less /var/log/exim4/mainlog # For Exim (on Debian-based systems)

Replace the path with the appropriate path for your specific mail server software and Linux distribution. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to mail or mail server software.

21.Where we find scheduling logs?

Ans:Scheduling logs, typically related to cron jobs (scheduled tasks) and at jobs, are usually stored in specific log files in the /var/log directory or its subdirectories on Unix-like systems. The specific log files can vary based on the system and the scheduling mechanisms being used.

Here are common log files related to scheduling:

Cron Jobs:

Ubuntu, Debian, and related distributions:

Location: /var/log/syslog

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/cron

At Jobs:

Ubuntu, Debian, and related distributions:

Location: /var/log/syslog

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/secure

To view the contents of the scheduling log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/syslog # For scheduling logs on Ubuntu, Debian, and related distributions

tail -f /var/log/cron # For cron logs on CentOS, Fedora, Red Hat, and related distributions

less /var/log/syslog # To view the log contents using less

Replace the path with the appropriate path for your specific Linux distribution and scheduling mechanism. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to scheduling or cron jobs

22.Where we find booting logs?

Ans:Booting logs, which contain information about the system boot process and events related to system startup, are often stored in specific log files in the /var/log directory or its subdirectories on Unix-like systems. These logs can provide valuable information for troubleshooting boot issues and understanding the system's startup sequence.

The common log file related to the boot process is:

Boot Logs:

Ubuntu, Debian, and related distributions:

Location: /var/log/boot.log

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/boot.log

To view the contents of the boot log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/boot.log # For boot logs on Ubuntu, Debian, and related distributions

less /var/log/boot.log # To view the log contents using less

Replace the path with the appropriate path for your specific Linux distribution. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to the boot process or startup. Additionally, some distributions may use dmesg to display kernel ring buffer messages, which often include boot-related information. Use dmesg for viewing kernel logs related to the boot process:

dmesg | less

23.What is the use of “lastb” command ?

Ans:The lastb command is used to display a list of failed login attempts recorded in the system's wtmp or btmp files. It stands for "last bad login."

Key uses and features of the lastb command include:

Display Failed Login Attempts:

The primary purpose of lastb is to show a list of failed login attempts on the system.

Show Source IP and User:

It displays the source IP address or hostname and the username associated with each failed login attempt.

Display Time and Date:

lastb provides information about the time and date of each failed login attempt.

Access Security Information:

It's a valuable tool for system administrators to monitor and analyze potential security breaches or unauthorized login attempts.

Useful for Security Auditing:

The information displayed by lastb can be helpful for security auditing, allowing administrators to detect and respond to suspicious login activities.

Usage:

lastb

By default, lastb reads data from the /var/log/btmp file, which logs failed login attempts. The information displayed includes the user attempting to log in, the source IP address, and the date and time of the attempt.

It's important to note that lastb requires superuser privileges to access the /var/log/btmp file and read the login failure records. Typically, you would run this command with sudo or as the root user.

For successful login history, you can use the last command without the b option (last instead of lastb).

24.Where we find general logs ?

Ans:General system logs, which include a wide range of system and application-related messages, are typically stored in various log files across the system. On Unix-like systems such as Linux, these logs are often found in the /var/log directory or its subdirectories. Here are some common log files that contain general system logs:

syslog:

Contains general system messages and logging from various system components.

Location: /var/log/syslog.

messages:

Contains general system messages, often including logs from daemons and services.

Location: /var/log/messages.

kern.log:

Contains kernel-related messages, including hardware-related events and errors.

Location: /var/log/kern.log.

auth.log or secure:

Contains authentication-related messages, including login attempts, su (switch user) activity, and more.

Ubuntu, Debian, and related distributions: /var/log/auth.log.

CentOS, Fedora, Red Hat, and related distributions: /var/log/secure.

auth.log (older Debian-based systems) or secure (older Red Hat-based systems):

Deprecated, but still found in some older systems, similar to the current auth.log and secure.

boot.log:

Contains logs related to the system boot process.

Location: /var/log/boot.log.

debug:

Contains debug messages from various applications and components.

Location: /var/log/debug.

To view the contents of these log files, you can use commands such as cat, tail, less, or grep. For example:

cat /var/log/syslog # View syslog contents

tail -f /var/log/messages # View messages in real-time

less /var/log/kern.log # View kernel logs using less

Each log file provides specific information about the system's activities, errors, and events, which can be invaluable for troubleshooting, monitoring, and maintaining the system. The appropriate log file to consult will depend on the type of information you are seeking.

25.Where we find secure logs ?

Ans:The "secure" log file, which typically contains authentication-related messages such as login attempts and authentication failures, is often found on systems that use the systemd logging system. However, the specific location of the "secure" log file can vary based on the Linux distribution. Here are some common locations for the "secure" log file:

Ubuntu, Debian, and related distributions:

Location: /var/log/auth.log

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/secure

To view the contents of the "secure" log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/auth.log # For Ubuntu, Debian, and related distributions

tail -f /var/log/secure # For CentOS, Fedora, Red Hat, and related distributions

less /var/log/auth.log # To view the log contents using less

Replace the path with the appropriate path for your specific distribution. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to authentication and security.

26.Where we find mail log ?

Ans:The mail log, which contains logs related to email servers and mail delivery, is often located in the /var/log directory on Unix-like systems. The specific name of the mail log file and its location can vary based on the mail server software being used and the Linux distribution. Here are some common locations for the mail log file:

Postfix (a popular mail server):

Location: /var/log/mail.log (or /var/log/maillog on some systems)

Sendmail:

Location: /var/log/maillog

Exim:

Location: /var/log/exim4/mainlog (on Debian-based systems, like Ubuntu)

Location: /var/log/exim/mainlog (on Red Hat-based systems, like CentOS)

Dovecot (a popular IMAP and POP3 server):

Location: /var/log/mail.log (or /var/log/maillog on some systems)

To view the contents of the mail log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/mail.log # For Postfix or Dovecot

tail -f /var/log/maillog # For Sendmail

less /var/log/exim4/mainlog # For Exim (on Debian-based systems)

Replace the path with the appropriate path for your specific mail server software and Linux distribution. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to mail or mail server software.

27.Where we find scheduling logs?

Ans:Scheduling logs, typically related to cron jobs (scheduled tasks) and at jobs, are usually stored in specific log files in the /var/log directory or its subdirectories on Unix-like systems. The specific log files can vary based on the system and the scheduling mechanisms being used.

Here are common log files related to scheduling:

Cron Jobs:

Ubuntu, Debian, and related distributions:

Location: /var/log/syslog

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/cron

At Jobs:

Ubuntu, Debian, and related distributions:

Location: /var/log/syslog

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/secure

To view the contents of the scheduling log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/syslog # For scheduling logs on Ubuntu, Debian, and related distributions

tail -f /var/log/cron # For cron logs on CentOS, Fedora, Red Hat, and related distributions

less /var/log/syslog # To view the log contents using less

Replace the path with the appropriate path for your specific Linux distribution and scheduling mechanism. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to scheduling or cron jobs.

28.Where we find booting logs?

Ans:Booting logs, which contain information about the system boot process and events related to system startup, are often stored in specific log files in the /var/log directory or its subdirectories on Unix-like systems. These logs can provide valuable information for troubleshooting boot issues and understanding the system's startup sequence.

The common log file related to the boot process is:

Boot Logs:

Ubuntu, Debian, and related distributions:

Location: /var/log/boot.log

CentOS, Fedora, Red Hat, and related distributions:

Location: /var/log/boot.log

To view the contents of the boot log file, you can use the cat, tail, less, or grep command in the terminal. For example:

cat /var/log/boot.log # For boot logs on Ubuntu, Debian, and related distributions

less /var/log/boot.log # To view the log contents using less

Replace the path with the appropriate path for your specific Linux distribution. If you're unsure of the location, you can use the ls command to list the contents of the /var/log directory and look for files related to the boot process or startup. Additionally, some distributions may use dmesg to display kernel ring buffer messages, which often include boot-related information. Use dmesg for viewing kernel logs related to the boot process:

dmesg | less

29.What is the use of “lastb” command ?

Ans:The lastb command is used to display a list of failed login attempts recorded in the system's wtmp or btmp files. It stands for "last bad login."

Key uses and features of the lastb command include:

Display Failed Login Attempts:

The primary purpose of lastb is to show a list of failed login attempts on the system.

Show Source IP and User:

It displays the source IP address or hostname and the username associated with each failed login attempt.

Display Time and Date:

lastb provides information about the time and date of each failed login attempt.

Access Security Information:

It's a valuable tool for system administrators to monitor and analyze potential security breaches or unauthorized login attempts.

Useful for Security Auditing:

The information displayed by lastb can be helpful for security auditing, allowing administrators to detect and respond to suspicious login activities.

Usage:

bash

Copy code

lastb

By default, lastb reads data from the /var/log/btmp file, which logs failed login attempts. The information displayed includes the user attempting to log in, the source IP address, and the date and time of the attempt.

It's important to note that lastb requires superuser privileges to access the /var/log/btmp file and read the login failure records. Typically, you would run this command with sudo or as the root user.

For successful login history, you can use the last command without the b option (last instead of lastb).

• Assignment Level Intermediate

1. Remote host is “NADIAD”, Remote user is “KAMAL,

how to access remote user via ssh? [ wright down the command]

Ans:To access a remote host named "NADIAD" with the user "kamal" via SSH, you would typically use the following command:

ssh kamal@NADIAD

Replace "NADIAD" with the actual hostname or IP address of the remote host.

If you need to specify a custom SSH port (other than the default port 22), you can do so using the -p option followed by the port number. For example, if the SSH port is 2222:

ssh -p 2222 kamal@NADIAD

This command will attempt to establish an SSH connection to the remote host "NADIAD" with the username "kamal". It will prompt for the user's password unless SSH key-based authentication is configured for the user "kamal" on the remote host.

2. What is the use of “w -f “command ?

Ans:The w command in Unix-like operating systems is used to display information about currently logged-in users and what they are doing. The -f option for the w command provides additional details related to the user's FROM address and login time.

Here's the specific use of w -f:

Display Detailed User Information:

The w command alone displays general information about logged-in users, including their username, terminal, remote host (FROM address), login time, idle time, and current activity.

With -f Option:

When you add the -f option (w -f), it enhances the output by providing additional details such as the FROM address (the remote host or terminal from which the user is logged in) and the login time.

Usage:

bash

Copy code

w -f

Output example:

l

15:40:50 up 10 days, 5:20, 3 users, load average: 0.00, 0.01, 0.05

USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT

user1 tty1 :0 15:38 10:00 0.05s 0.05s bash

user2 pts/0 :0 14:40 0.00s 0.03s 0.01s w -f

user3 pts/1 192.168.1.10 13:20 2:19 0.02s 0.02s -bash

In this example, the w -f command provides details about logged-in users, including the FROM address (remote host or terminal) and the time of login.

USER: Username of the logged-in user.

TTY: Terminal or device the user is connected to.

FROM: Remote host or address from which the user is connected.

LOGIN@: Time of login.

IDLE: Idle time since the last activity.

JCPU: Total CPU time used by all processes attached to the tty.

PCPU: CPU time used by the current process (the w -f command in this case).

WHAT: Current activity or command being executed by the user.

Using w -f provides a more detailed view of the logged-in users and their activities.

3. What is “SSHS host keys “?

Ans:It seems like there might be a typo or misunderstanding in your question. "SSHS" is not a standard acronym or term related to SSH (Secure Shell). However, I'll provide information about SSH host keys in the context of SSH (Secure Shell).

In SSH, host keys are used for secure communication and verification of the authenticity of the SSH server. When you connect to a remote SSH server, your SSH client verifies the server's identity using its host key. The host key is essentially a cryptographic fingerprint unique to that server.

There are three main types of SSH host keys:

RSA (Rivest-Shamir-Adleman): An older and widely used type of SSH host key based on the RSA algorithm.

DSA (Digital Signature Algorithm): Another older type of SSH host key, not as commonly used today due to potential security concerns.

ECDSA (Elliptic Curve Digital Signature Algorithm): A modern and widely used SSH host key type that offers strong security with smaller key sizes compared to RSA and DSA.

Ed25519: A modern and highly secure SSH host key type based on the EdDSA (Edwards-curve Digital Signature Algorithm).

When you connect to a new SSH server, your SSH client checks if the server's host key is already stored in your known\_hosts file (usually located in ~/.ssh/known\_hosts). If not, it will prompt you to accept the new host key. Once you accept, the host key is stored, and future connections to that server will use this key for verification.

The host key is essential for establishing a secure and encrypted communication channel between your client and the SSH server, ensuring that the server you are connecting to is the correct and trusted one. If the host key changes unexpectedly (e.g., due to a potential security issue), your SSH client will alert you about the change to prevent potential man-in-the-middle attacks.

4. What is the default location for server’s public key in client side?

Ans:The default location for storing a server's public key on the client side is typically within the ~/.ssh/known\_hosts file, where ~ represents the user's home directory.

The known\_hosts file is a text file that stores the public keys of hosts (servers) you have connected to and accepted the host keys for. This file is used by the SSH client to verify the authenticity of the server when connecting.

Each line in the known\_hosts file typically contains the following information:

hostname ssh-algorithm public-key

hostname: The hostname or IP address of the SSH server.

ssh-algorithm: The encryption algorithm used by the SSH server.

public-key: The server's public key for the specified algorithm.

For example:

example.com ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC...

If the known\_hosts file doesn't exist, SSH will create it the first time you connect to a remote SSH server and accept its public key.

Keep in mind that if the server's host key changes (for instance, due to a reinstallation of the server), SSH will display a warning about a possible man-in-the-middle attack. In this case, it's important to verify the server's identity and, if appropriate, update the known\_hosts file with the new host key.

5. I want to fire “ls -l /etc” command on remote host “desktop”

[ wright down the command ]

Ans:To execute the ls -l /etc command on a remote host named "desktop" using SSH, you can use the ssh command. Here's the command:

ssh username@desktop "ls -l /etc"

Replace "username" with your actual username on the remote host, and replace "desktop" with the actual hostname or IP address of the remote host.

This command will SSH into the "desktop" host, execute the ls -l /etc command, and display the detailed listing of files in the /etc directory.

6. What is the use of this command “ #journalctl --since today “

Ans:The journalctl --since today command is used to display system journal entries (logs) since the beginning of the current day. This command is particularly useful for viewing system logs recorded on the same day when troubleshooting or monitoring system activities.

Here's what each part of the command does:

journalctl: This command is used to query and display logs from the systemd journal, a centralized logging system used by modern Linux distributions.

--since today: This option filters and displays logs recorded since the beginning of the current day, allowing you to focus on recent events and activities.

Usage:

journalctl --since today

Output example:

-- Logs begin at Wed 2021-10-06 09:45:23 UTC, end at Wed 2021-10-06 14:30:01 UTC. --

Oct 06 09:45:23 mymachine systemd[1]: Starting The Apache HTTP Server...

Oct 06 09:45:23 mymachine apachectl[12345]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. Set the 'ServerName' directive globally to suppress this message

Oct 06 09:45:23 mymachine systemd[1]: Started The Apache HTTP Server.

...

In this example, the journalctl --since today command displays logs that start from the beginning of the current day until the present time. It provides details about various system events, services, and processes recorded in the system journal for the specified time frame.

7. What is “ chronyd “?

Ans:The term "chroynd" does not appear to be a standard or recognized term in the context of computer science, software, or common technical terminology up to my knowledge cutoff date in September 2021. It's possible that it's a typo, a misspelling, or a specific term related to a particular field or domain that is not widely recognized or documented.

If you have additional context or details about where you encountered this term or how it's being used, please provide more information so I can attempt to assist you further.

8. Full form of NTP

Ans:NTP stands for Network Time Protocol. NTP is a networking protocol used for clock synchronization between computer systems over packet-switched, variable-latency data networks. It's essential for maintaining accurate and synchronized time across devices and systems in a network, ensuring consistent time for applications, security, and other processes that rely on precise timing. NTP helps in managing time and coordinating time-based activities in a networked environment.

9. Port number for NTP is…

Ans:The default port number for NTP (Network Time Protocol) is 123, both for UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) transport. NTP primarily uses UDP for time synchronization.

When configuring NTP servers or clients, port 123 is the standard port used for communication between them. It's important to ensure that this port is open and accessible for NTP traffic to flow between servers and clients, allowing for accurate time synchronization across the network.

10.I want to check timzone, which command will help me ?

Ans:To check the timezone on a Linux system, you can use the timedatectl command, which is a command-line utility for configuring the system clock, time, and date settings, including the timezone.

Here's how you can use timedatectl to check the timezone:

timedatectl

The output will display detailed information about the current system time, date, and timezone, including the timezone name, if set.

Example output:

Local time: Fri 2023-10-06 16:45:23 UTC

Universal time: Fri 2023-10-06 16:45:23 UTC

RTC time: Fri 2023-10-06 16:45:24

Time zone: UTC (UTC, +0000)

In this example, the timezone is set to UTC.

If you're interested in listing all available timezones, you can use the timedatectl command with the list-timezones option:

timedatectl list-timezones

This command will display a long list of available timezones that you can set on your system.

11.How to set timezone? Give a comman….

Ans:To check the timezone on a Linux system, you can use the timedatectl command, which is a command-line utility for configuring the system clock, time, and date settings, including the timezone.

Here's how you can use timedatectl to check the timezone:

timedatectl

The output will display detailed information about the current system time, date, and timezone, including the timezone name, if set.

Example output:

Local time: Fri 2023-10-06 16:45:23 UTC

Universal time: Fri 2023-10-06 16:45:23 UTC

RTC time: Fri 2023-10-06 16:45:24

Time zone: UTC (UTC, +0000)

In this example, the timezone is set to UTC.

If you're interested in listing all available timezones, you can use the timedatectl command with the list-timezones option:

timedatectl list-timezones

This command will display a long list of available timezones that you can set on your system.

User

how to set timezone? give a comman..

ChatGPT

To set the timezone on a Linux system using the timedatectl command, you can use the following command:

sudo timedatectl set-timezone Your\_Timezone

Replace Your\_Timezone with the specific timezone you want to set. For example, to set the timezone to "America/New\_York":

sudo timedatectl set-timezone America/New\_York

Make sure to use the correct timezone name from the list of available timezones. You can get a list of available timezones using the timedatectl list-timezones command.

• Task :1

1. Display all processes on display

Ans:To display all processes on a Linux system, you can use the ps command. To list all processes currently running in a terminal, you can use the following command:

ps -ef

This will display a list of all processes along with detailed information about each process, including the process ID (PID), the user running the process, the command that started the process, and more.

If you want a more user-friendly display with a tree-like structure showing the relationships between processes, you can use the pstree command:

pstree

This will display a process tree, illustrating the parent-child relationships between processes.

Keep in mind that the exact options and behavior of the ps and pstree commands may vary slightly depending on your Linux distribution and system configuration.

2. In terminal, Determine the number of logical CPUs

Ans:To determine the number of logical CPUs (threads) on a Linux system using the terminal, you can use the nproc command. Here's how:

nproc

The nproc command will output the number of processing units available to the current process, which is typically the number of logical CPUs or threads on the system.

If you want to also display the number of physical CPUs (cores), you can use the lscpu command:

lscpu

This command will provide detailed information about the CPU, including the number of sockets, cores, threads, and other relevant information.

The number of logical CPUs is often what people refer to as the "number of CPUs" in a system, especially in multi-core/multi-threaded environments.

3. Start and check any new job

Ans:To start a new job in a Linux terminal, you typically run a command or a script in the background. You can use the & operator to run a command in the background. For example:

your\_command &

Replace your\_command with the command or script you want to run in the background.

To check the status of the background job, you can use the jobs command:

jobs

This will display a list of all the jobs running in the current shell session, along with their job numbers.

If you want to bring a background job to the foreground, you can use the fg command followed by the job number. For example, if the job number is 1:

fg %1

This will bring job number 1 to the foreground.

If you want to send a job to the background or resume a suspended job, you can use the bg command followed by the job number:

bg %1

Replace 1 with the appropriate job number.

Keep in mind that if you start a new terminal session or close the current one, the background jobs running in that session will be terminated. If you want a job to persist even after you close the terminal, consider using tools like nohup or tmux.

4. Start any new job in background

Ans:To start a new job in the background, let's use a simple example of running the sleep command for 60 seconds. Here's how you can start this job in the background:

bash

Copy code

sleep 60 &

In this example, sleep 60 will make the terminal "sleep" (do nothing) for 60 seconds. The & operator at the end runs this command in the background.

You can replace sleep 60 with any other command or script you want to run in the background.

To check the status of the background job, you can use the jobs command:

jobs

This will display the job number and its status.

If you want to bring a background job to the foreground, you can use the fg command followed by the job number. For example:

fg %1

Replace 1 with the appropriate job number.

To send a job to the background or resume a suspended job, you can use the bg command followed by the job number:

bg %1

Replace 1 with the appropriate job number.

Remember, you can replace sleep 60 with any other command or script you want to run in the background.

5. Start any background job on foreground

Ans:To start a new background job and then bring it to the foreground, you'll first start a job in the background using the & operator and then use the fg command to bring it to the foreground. Let's use the sleep command as an example:

Start a background job using sleep:

sleep 30 &

This will make the sleep command run in the background for 30 seconds.

Use jobs to see the job number. It will be something like [1] for the first job:

jobs

Bring the background job to the foreground using its job number (replace [1] with the appropriate job number):

fg %1

The sleep command will now be in the foreground.

You can replace sleep 30 with any other command or script you want to run in the background and then bring to the foreground.

6. Check running process

Ans:To start a new background job and then bring it to the foreground, you'll first start a job in the background using the & operator and then use the fg command to bring it to the foreground. Let's use the sleep command as an example:

Start a background job using sleep:

bash

Copy code

sleep 30 &

This will make the sleep command run in the background for 30 seconds.

Use jobs to see the job number. It will be something like [1] for the first job:

bash

Copy code

jobs

Bring the background job to the foreground using its job number (replace [1] with the appropriate job number):

bash

Copy code

fg %1

The sleep command will now be in the foreground.

You can replace sleep 30 with any other command or script you want to run in the background and then bring to the foreground.

User

check running process

ChatGPT

To check the currently running processes on a Linux system, you can use the ps command. Here are several variations of the ps command that you can use to display information about running processes:

Display a List of User's Processes:

ps

Display Detailed Information for All Processes:

ps -ef

Display Process Tree:

pstree

Display All Processes in a Tree-Like Structure:

ps auxf

Display Processes in a Process Group:

ps -e -o pid,ppid,cmd,pgid,sid,tty,stat

This command displays the process ID, parent process ID, command, process group ID, session ID, terminal, and process status for each process.

These commands will provide different levels of detail about the running processes on your system. Choose the one that best fits your needs based on the information you are looking for.

7. Check all running process under user

Ans:To check all running processes for a specific user, you can use the ps command with the -u option followed by the username. Here's the command to achieve this:ps -u username

Replace username with the actual username for which you want to check the running processes.

For example, to check all running processes for the user "john":

ps -u john

This command will display a list of all running processes associated with the specified username.

8. Kill any process via it’s PID

Ans:To terminate (kill) a process by its Process ID (PID), you can use the kill command in the terminal. Here's how you can do it:

First, find out the PID of the process you want to terminate. You can use the ps command to list the processes and find the PID associated with the process you want to terminate. For example:

ps aux | grep process\_name

Replace process\_name with the name or part of the name of the process you are looking for.

Once you have the PID of the process, use the kill command followed by the PID to terminate the process. For example, to kill a process with PID 12345:

kill 12345

If you need to force the termination of the process, you can use the -9 (SIGKILL) option with kill:

kill -9 12345

Replace 12345 with the actual PID of the process you want to terminate.

Keep in mind that using kill without -9 sends a termination signal (SIGTERM) to the process, allowing it to perform cleanup actions before exiting. Using -9 (SIGKILL) forcefully terminates the process without allowing it to clean up. It's usually recommended to try the regular kill first and resort to -9 if the process does not respond to the regular termination signal.

9. Change nice values for any new process

Ans:To set the nice value for a new process, you can use the nice command or its equivalent renice. The nice value determines the scheduling priority of a process, with lower values indicating higher priority.

Using nice for a New Process:

To start a new process with a specific nice value, use the nice command followed by the desired nice value and the command you want to run. For example:

nice -n 10 your\_command

Replace 10 with the desired nice value (values range from -20 to 19, with lower values having higher priority), and replace your\_command with the actual command you want to run.

Using renice to Change Nice Value of an Existing Process:

To change the nice value of an existing process, you can use the renice command followed by the nice value and the process ID (PID). For example:

renice 10 -p PID

Replace 10 with the desired nice value, and replace PID with the process ID of the process you want to adjust.

Keep in mind that adjusting the nice value requires appropriate permissions. If you don't have the necessary privileges, you may need to use sudo before the command to run it with superuser privileges.

10.Change nice value for any running user

Ans:To change the nice value (scheduling priority) of a running process owned by a specific user, you'll use the renice command. Here's how you can do it:

First, identify the process ID (PID) of the running process you want to modify. You can use the ps command to list processes and find the PID of the process owned by the user. For example, to find the PID of processes owned by the user "john":

ps -u john

Note down the PID of the process you want to modify.

Use the renice command to change the nice value for the process. You'll need superuser privileges (root or sudo) to adjust the nice value of a process not owned by your user. For example, to change the nice value to 10 for a process with PID 12345:

sudo renice 10 -p 12345

Replace 10 with the desired nice value (values range from -20 to 19, with lower values having higher priority), and replace 12345 with the actual PID of the process you want to adjust.

Keep in mind that changing the nice value requires appropriate permissions. Using sudo before the renice command ensures that you have the necessary privileges to modify the nice value of a process.

11.Check the status of sshd.service

Ans:

To check the status of the SSH (sshd) service on a Linux system, you can use the systemctl command, which is the standard tool for controlling and managing services on a systemd-based system. Here's how to check the status of the SSH service:

systemctl status sshd.service

This command will display detailed information about the current status of the SSH service, including whether it's running, any errors, and other relevant information.

If you're not running systemd or your system uses a different init system, the command to check the SSH service status may vary. However, on most modern Linux distributions, systemd is the default init system.

12.Stop the servive of sshd.service

Ans:To stop the SSH (sshd) service on a Linux system using systemctl, you can use the following command:

sudo systemctl stop sshd.service

This command will stop the SSH service (sshd.service) immediately. Keep in mind that you typically need superuser privileges (hence sudo) to start or stop services.

If you're using a different init system or older versions of SSH that use init scripts, the command to stop the service may differ. However, for modern Linux distributions using systemd, the systemctl command is the standard way to manage services, including starting and stopping them.

13.Start the service of sshd.service

Ans:To start the SSH (sshd) service on a Linux system using systemctl, you can use the following command:

sudo systemctl start sshd.service

This command will start the SSH service (sshd.service). Make sure you have the necessary privileges (hence sudo) to start services.

If you're using a different init system or older versions of SSH that use init scripts, the command to start the service may differ. However, for modern Linux distributions using systemd, the systemctl command is the standard way to manage services, including starting and stopping them.

14.Login in server vm

Ans:To log in to a server virtual machine (VM), you'll typically use SSH (Secure Shell) if it's a remote server. Here's a step-by-step process to log in using SSH:

Open a Terminal (local machine):

Open a terminal on your local machine. This could be the terminal application on your computer if you're using a desktop OS, or it could be a terminal emulator within a virtualization application.

Use SSH to Connect:

Use the ssh command to connect to the server VM. Replace username with your username and server\_ip with the IP address or hostname of the server VM:

ssh username@server\_ip

If using a specific port (other than the default SSH port 22), you can specify it with the -p option:

ssh -p PORT username@server\_ip

Enter Password (if required):

If it's your first time connecting to the server or if SSH key authentication is not set up, you'll be prompted to enter the password associated with the username.

Authenticate with SSH Key (optional):

If you've set up SSH key authentication, you won't need to enter a password. Instead, your SSH key will be used to authenticate you.

ssh -i /path/to/private\_key username@server\_ip

Replace /path/to/private\_key with the actual path to your private SSH key.

You're Now Logged In:

Once authenticated, you will be logged in to the server VM and can use the terminal on your local machine to interact with the server.

Remember to replace username and server\_ip with your actual username and the IP address or hostname of the server VM you want to connect to.

15.Display the status of “chronyd”

Ans:To display the status of the "chronyd" service, you can use the systemctl command in a Linux terminal. Here's how you can check the status:

sudo systemctl status chronyd

This command will provide detailed information about the status of the "chronyd" service, including whether it's running, any errors, and other relevant information.

If the "chronyd" service is active and running, you'll see output indicating its status, when it was started, and other relevant details. If it's inactive or not running, the output will reflect that status as well.

16.Restart “sshd.service”

Ans:To restart the SSH (sshd) service on a Linux system using systemctl, you can use the following command:

sudo systemctl restart sshd.service

This command will stop and then start the SSH service (sshd.service), effectively restarting it. Make sure you have the necessary privileges (hence sudo) to restart services.

After running this command, the SSH service will be restarted, and any configuration changes or updates will take effect.

• TASK: 3

1. Start desktop machine

Ans:Starting a desktop machine can be done physically by pressing the power button on the computer or virtually if you're working with a virtual machine. Since you mentioned "desktop machine," I'll assume you're referring to a physical desktop.

To start a physical desktop machine:

Make sure the desktop machine is connected to a power source and the power cable is plugged in securely.

Locate the power button on the desktop machine. It's usually on the front or top of the computer case.

Press the power button to start the machine. The computer should begin its boot process, and you should see the manufacturer's logo or a startup screen.

Wait for the operating system to load. Depending on the speed of your hardware and the operating system, this may take a few moments.

Once the operating system is loaded, you can log in and start using the desktop machine.

If you're referring to starting a virtual machine, the process would depend on the virtualization software you're using (e.g., VirtualBox, VMware, Hyper-V). Let me know if you need instructions for starting a virtual machine.

2. Get remote access of server machine

Ans:To get remote access to a server machine, you'll typically use SSH (Secure Shell), which provides a secure way to access the server over a network. Here's a step-by-step guide to establish a remote SSH connection to a server:

Get Server Details:

Obtain the following information about the server you want to connect to:

IP address or hostname of the server.

Username to log in as.

Open a Terminal (Local Machine):

Open a terminal or command prompt on your local machine.

Connect using SSH:

Use the ssh command to connect to the server. Replace username with your actual username and server\_ip with the IP address or hostname of the server.

ssh username@server\_ip

If you're using a non-standard SSH port (other than the default port 22), you can specify it with the -p option:

ssh -p PORT username@server\_ip

Enter Password or SSH Key Passphrase:

If it's your first time connecting to the server or if SSH key authentication is not set up, you'll be prompted to enter the password associated with the username. If using SSH key authentication, you may need to enter the passphrase for your SSH key.

You're Now Connected to the Server:

Once authenticated, you will be connected to the server remotely via SSH, and you can use the terminal on your local machine to interact with the server.

Make sure you have the necessary credentials and permissions to access the server. If you encounter any issues or need further assistance, feel free to ask.

3. Create new user in server name “user1”

Ans:To create a new user named "user1" on a Linux server, you can use the adduser or useradd command, depending on your distribution. Here's a general guide:

Using adduser (interactive and user-friendly):

sudo adduser user1

Follow the prompts to set a password and provide additional user information as needed.

Using useradd (command-line):

sudo useradd user1

This command will create the user, but you'll need to set a password separately using the passwd command:

sudo passwd user1

Follow the prompts to set the password.

Grant sudo (optional):

If you want to grant administrative privileges to the user (add them to the sudo group), you can use the usermod command:

sudo usermod -aG sudo user1

Replace sudo with the appropriate group name based on your distribution (e.g., wheel on some systems).

The user "user1" should now be created and, if desired, have sudo privileges. Adjust the steps slightly if you need to customize the user's home directory, shell, or other settings during user creation.

4. From desktop machine,login “user1” of server user

Ans:To log in as "user1" on the server from your desktop machine, you'll use SSH to establish a remote connection. Here are the steps to log in as "user1" on the server from your desktop:

Open a Terminal (Desktop Machine):

Open a terminal or command prompt on your desktop machine.

Use SSH to Connect to the Server:

Use the ssh command to connect to the server as "user1". Replace server\_ip with the actual IP address or hostname of the server.

ssh user1@server\_ip

If you're using a non-standard SSH port, you can specify it with the -p option:

ssh -p PORT user1@server\_ip

Enter Password or SSH Key Passphrase:

If it's your first time connecting or if SSH key authentication is not set up, you'll be prompted to enter the password associated with "user1" or the SSH key passphrase.

You're Now Logged In as "user1":

Once authenticated, you'll be logged in to the server as "user1" and can use the terminal on your local machine to interact with the server.

Ensure you have the correct credentials and permissions to log in as "user1". If you encounter any issues or need further assistance, feel free to ask.

5. Execute single command “hostname” , on remote host

(server), and as a remote user “user1”

Ans:To execute the hostname command on the remote server as the remote user "user1," you can use SSH. Here's the command to achieve this:

ssh user1@server\_ip 'hostname'

Replace server\_ip with the actual IP address or hostname of the server. When you run this command, it will connect to the server as "user1" and execute the hostname command remotely, displaying the server's hostname.

6. Display a list of currently logged into the computer

Ans:

To display a list of currently logged-in users on a Linux server, you can use the who or w command. Here's how:

Using who command:

who

This will display a list of users currently logged in, along with information such as username, terminal, date, and time of login.

Using w command:

w

The w command provides more detailed information, including the username, terminal, from where the user logged in, idle time, JCPU (total CPU time used by the job's processes), and more.

These commands will provide information about currently logged-in users on the system. If you need additional details or have specific requirements, feel free to ask.

7. Generate private-public ssh key with password

Ans:To generate a SSH key pair with a passphrase (password) for added security, you'll use the ssh-keygen command. Here are the steps to generate a private-public SSH key pair with a passphrase:

Open a Terminal:

Open a terminal on your local machine.

Generate SSH Key Pair:

Use the ssh-keygen command to generate the SSH key pair. You'll be prompted to enter a file to save the key and the passphrase.

bash

Copy code

ssh-keygen -t rsa -b 2048

Replace -t rsa with the desired key type (rsa, dsa, ecdsa, ed25519) and -b 2048 with the desired key length (2048 bits is commonly used).

Enter File Location and Passphrase:

You'll be prompted to enter the file to save the key. Press Enter to accept the default location, usually ~/.ssh/id\_rsa for the private key.

You'll then be prompted to enter a passphrase. Enter a strong passphrase, which acts as an extra layer of security.

Confirm Passphrase:

Re-enter the passphrase to confirm.

SSH Key Pair Generated:

Your SSH key pair (private and public keys) has been generated.

The private key will be saved in ~/.ssh/id\_rsa (or the location you specified) and the public key in ~/.ssh/id\_rsa.pub. The private key is protected by the passphrase you entered.

You can now use this SSH key pair to authenticate to remote servers. When connecting to a server, you'll be prompted to enter the passphrase for the private key.

8. Import this key on remote host side

Ans:To import your SSH key (public key) to the remote host and allow authentication using this key, you need to copy the public key to the ~/.ssh/authorized\_keys file on the remote host. Here are the steps to achieve this:

Copy the Public Key (Local Machine):

First, copy the content of the public key file on your local machine. You can use the cat command to display the content, and then copy it.

bash

Copy code

cat ~/.ssh/id\_rsa.pub

Copy the entire content displayed.

SSH to the Remote Host:

Log in to the remote host using SSH.

bash

Copy code

ssh user@remote\_host\_ip

Replace user with your username and remote\_host\_ip with the IP address or hostname of the remote host.

Edit the authorized\_keys File:

Once logged in to the remote host, use a text editor (like nano or vi) to edit the authorized\_keys file.

bash

Copy code

nano ~/.ssh/authorized\_keys

If the authorized\_keys file does not exist, you can create it.

Paste the Public Key (Remote Host):

Paste the previously copied public key at the end of the authorized\_keys file.

Save and Exit (Remote Host):

Save the file and exit the text editor.

For nano, press Ctrl + X, then press Y to confirm changes, and press Enter to exit.

For vi, type :wq and press Enter.

Set Correct Permissions (Remote Host):

Ensure the correct permissions for the ~/.ssh directory and authorized\_keys file on the remote host.

bash

Copy code

chmod 700 ~/.ssh

chmod 600 ~/.ssh/authorized\_keys

These commands set the appropriate permissions: 700 for the ~/.ssh directory and 600 for the authorized\_keys file.

You have now imported your SSH public key to the remote host. You can use your private key to authenticate to the remote host without entering a password, provided you have entered a passphrase when generating the key pair.

• Task: 4

1. Open general logs

Ans:In a Linux-based system, "general logs" typically refer to system logs, which contain a record of system events, processes, and other important information. These logs are usually stored in the /var/log directory.

To view the general system logs, you can use various commands and tools depending on the specific log you want to access. Here are some common logs you might want to check and how to view them:

System Log (syslog or messages):

Use the cat or less command to view the system log:

cat /var/log/syslog

or

less /var/log/syslog

You might also find logs in /var/log/messages on some systems.

Kernel Log (kern.log):

Use the same approach as above to view the kernel log:

cat /var/log/kern.log

or

less /var/log/kern.log

Authentication Logs (auth.log or secure):

View logs related to authentication:

cat /var/log/auth.log

or

less /var/log/auth.log

On some systems, you might find this information in /var/log/secure.

Application Logs:

Application-specific logs are often located in /var/log. For example, Apache logs might be in /var/log/apache2/.

ls /var/log

Use cd to navigate to a specific application's log directory, and then use cat or less to view the logs.

Remember, viewing system logs typically requires superuser privileges. You can use sudo before these commands to elevate your privileges if needed.

2. Open secure message logs

Ans:The location and name of the secure message logs can vary slightly depending on the specific Linux distribution you are using. However, a common location for authentication-related logs is /var/log/auth.log on Debian-based distributions and /var/log/secure on Red Hat-based distributions.

To view the secure message logs, you can use the cat or less command, depending on whether you want to display the entire content at once or page through the content.

Here are the commands to view the secure message logs for both Debian-based and Red Hat-based distributions:

For Debian-based distributions (e.g., Ubuntu):

cat /var/log/auth.log

For Red Hat-based distributions (e.g., CentOS):

cat /var/log/secure

Replace cat with less if you prefer to view the logs one page at a time:

less /var/log/auth.log # For Debian-based distributions

less /var/log/secure # For Red Hat-based distributions

Remember that viewing logs often requires superuser privileges, so you may need to use sudo before the commands if you are not logged in as the root user.

3. Open only mail logs

AnS:Mail logs are often stored in /var/log/mail.log on Debian-based distributions and /var/log/maillog on Red Hat-based distributions. Let's open these logs using the appropriate command based on the distribution you're using:

For Debian-based distributions (e.g., Ubuntu), to view mail logs:

sudo less /var/log/mail.log

For Red Hat-based distributions (e.g., CentOS), to view mail logs:

sudo less /var/log/maillog

Replace less with cat if you prefer to display the entire content at once rather than paging through it.

Please note that you might need superuser privileges (sudo) to view these logs. Adjust the commands accordingly based on your specific distribution and permissions.

4. Check scheduling logs

Ans:Scheduling-related logs are typically found in /var/log/cron or /var/log/syslog depending on the distribution. The cron daemon is responsible for scheduling tasks on a Linux system.

To check scheduling logs, you can use the cat or less command based on your preference:

For viewing scheduling logs on Debian-based distributions (e.g., Ubuntu), you can use:

sudo cat /var/log/syslog

or

sudo less /var/log/syslog

For viewing scheduling logs on Red Hat-based distributions (e.g., CentOS), you can use:

sudo cat /var/log/cron

or

sudo less /var/log/cron

Replace cat with less if you prefer to page through the logs.

Remember to use sudo to ensure you have the necessary permissions to view the logs. Adjust the commands based on your specific distribution and permissions.

5. Check booting logs

Ans:Booting-related logs are typically found in /var/log/boot.log or /var/log/syslog. These logs contain information related to the boot process of the system.

To check booting logs, you can use the cat or less command based on your preference:

For viewing booting logs on Debian-based distributions (e.g., Ubuntu), you can use:

sudo cat /var/log/boot.log

or

sudo less /var/log/boot.log

For viewing booting logs on Red Hat-based distributions (e.g., CentOS), you can use:

sudo cat /var/log/boot.log

or

sudo less /var/log/boot.log

Replace cat with less if you prefer to page through the logs.

Remember to use sudo to ensure you have the necessary permissions to view the logs. Adjust the commands based on your specific distribution and permissions.

6. See the info about “bad logging”

Ans:To check for information related to "bad logging," we'll typically look into the general system logs or authentication logs, as "bad logging" could refer to failed login attempts or other security-related events.

Let's check the general system logs and authentication logs for any relevant information:

General System Logs (syslog or messages) for "bad logging":

bash

Copy code

sudo grep "bad logging" /var/log/syslog

or

sudo grep "bad logging" /var/log/messages

Authentication Logs (auth.log or secure) for failed login attempts or "bad logging":

sudo grep "bad logging" /var/log/auth.log

or

sudo grep "bad logging" /var/log/secure

Replace grep with cat if you want to see the entire content instead of filtering specific entries.

If "bad logging" refers to a specific error message, replace "bad logging" with the appropriate keyword or error message you are looking for.

These commands will search the specified logs for the term "bad logging" and display any matching entries. Adjust the commands based on your specific distribution and the exact information you're looking for.

7. Check today’s temporary logs

Ans:Temporary logs in a Linux system are usually stored in the /var/log directory or its subdirectories. The specific log files and locations may vary based on your Linux distribution and configuration.

To view the logs for today, you can use the grep and awk commands to filter logs for the current date. Here's a general approach:

grep "$(date +'%b %d')" /var/log/\* | less

This command uses grep to search for logs that contain the current month and day (e.g., "Oct 08" for October 8) in the /var/log directory and its subdirectories. The output is then piped to less for easier viewing.

Keep in mind that logs are usually rotated and archived, so the current day's logs might be stored in an archived file (e.g., syslog.1, syslog.2.gz, etc.). If you want to include archived logs, you can use tools like zcat or zgrep to read compressed logs.

zcat /var/log/syslog.\*.gz | grep "$(date +'%b %d')" | less

Replace syslog with the specific log file you are interested in.

Adjust the commands based on your specific distribution and log file names. If you have a particular log file or directory in mind, please provide more details for a more accurate command.

8. Set new time zone

Ans:To set a new time zone on a Linux system, you'll use the timedatectl command, which is available on systems that use systemd. Here's how you can change the time zone:

Check Current Time Zone:

Start by checking the current time zone using the timedatectl command:

bash

Copy code

timedatectl

The output will display the current time zone, including its abbreviation and other details.

Set a New Time Zone:

To set a new time zone, use the timedatectl set-timezone command followed by the desired time zone. For example, to set the time zone to "America/New\_York":

sudo timedatectl set-timezone America/New\_York

Replace "America/New\_York" with the appropriate time zone according to the IANA time zone database.

Verify the Change:

Verify that the time zone has been updated by running timedatectl again:

timedatectl

Check the Time zone field in the output to confirm the change.

The time zone has now been set to the new value, and the system will use this time zone for timekeeping.

Keep in mind that you might need superuser privileges (hence sudo) to change the time zone. Also, note that this procedure may vary slightly depending on your specific Linux distribution.