PART - A

1. What is the difference between Linux and Unix ?

feature	Linux	Unix
Open source	yes	no
availability	Widely available	Typically only available on high -end servers and workstations
features	Similar to unix	Similar to Linux
examples	Ubuntu,Debian,fedora	AIX ,HP-UX, olaries macOS

- Overall, Linux is a more accessible and user-friendly operating system than Unix. It is also more widely available and supported. However, Unix is still a popular choice for high-end servers and workstations.

2. What is BASH?

BASH stands for Bourne Again Shell.

Bash is a command interpreter, which means that it takes user input and executes it. It can be used to perform a wide variety of tasks, such as:

- Navigating the file system
- Manipulating files and directories
- Starting and stopping processes
- Communicating with other computers

Here are some examples of bash commands

- ls : list the contents of the current directory.
- cd : change directory
- mkdir : create a new directory
- rmdir : remove a directory
- ps : list running processes

3. What is Shell ?

A shell is a user interface that provides a way for users to interact with an operating system. It is a program that interprets user commands and executes them. Shells are typically command-line interfaces (CLIs), but there are also graphical user interfaces (GUIs) that provide a more user-friendly way to interact with the shell .

Shells are used to perform a wide variety of tasks such as:

- Navigating the file system
- Manipulating files and directories
- Starting and stopping processes

Here are some examples of popular shells:

- Bash
- C shell
- Z shell
- Ksh

4. **What is a swap space?**

Swap space is a portion of a computer's hard disk that is used as virtual memory. Virtual memory is a memory management technique that allows the operating system to use more memory than is physically installed in the system.

Swap space is a useful feature for systems that have limited physical memory. It allows the system to run more programs than would be possible with just the physical memory that is installed in the system. However, using swap space can slow down the system

5. **Explain File Permission types in Linux?**

There are three types of file permissions in Linux:

- Read I: Allows the user to read the contents of the file.
- Write (w): Allows the user to modify the contents of the file.
- Execute (x): Allows the user to execute the file, if it is an executable file.

File permissions are applied to three groups of users:

- **Owner:** The user who owns the file.
- **Group:** The group to which the file belongs.
- **Other:** All other users on the system.

File permissions are represented by a three-character string, where each character represents a permission for the owner, group, and other users. For example, the file permission string rwxr-xr–means that the owner has read, write, and execute permissions, the group has read and execute permissions, and all other users have read permissions

6. **What are the symbolic links?**

Symbolic links, also known as symlinks or soft links, are special types of files that point to other files or directories. They are similar to shortcuts in Windows.

To create a symbolic link, you can use the ln command with the -s option. For example, to create a symbolic link to the file myfile.txt called mylink.txt, you would use the following command:

```
ln -s myfile.txt mylink.txt
```

Once the symbolic link is created, you can access the file myfile.txt through the symbolic link mylink.txt. Any changes you make to the file through the symbolic link will also be reflected in the original file.

7. **What are the hard links?**

A hard link is a type of file in Linux that points to the same underlying data as another file. This means that a hard link is essentially a copy of the original file, but it takes up no additional space on the disk.

To create a hard link, you can use the ln command without the -s option. For example, to create a hard link to the file myfile.txt called mylink.txt, you would use the following command:

```
ln myfile.txt mylink.txt
```

Once the hard link is created, you can access the file myfile.txt through the hard link mylink.txt. Any changes you make to the file through the hard link will also be reflected in the original file.

8. **What are inode and process id?**

An inode, or index node, is a data structure that stores information about a file or directory in a Linux filesystem. This information includes the file's size, permissions, and location on the disk. Each file and directory has a unique inode number, which is used to identify the file or directory to the operating system.

A process ID, or PID, is a unique 4aited4yer that Is assigned to each running process on a Linux system. PIDs are used by the operating system to manage processes and to track their resource usage.

Inodes and PIDs are two important concepts in Linux systems. Inodes are used to store information about files and directories, while PIDs are used to manage and track running processes.

9. **What are the Process states in Linux?**

The process states in linux are :

- Running: The process is currently executing on the CPU.
- Runnable: The process is ready to run, but it is waiting for the CPU to be available.
- Sleeping: The process is waiting for an event to occur, such as a disk I/O operation to complete.
- Stopped: The process has been stopped by a signal from the user or the system.

Processes can transition between states at any time.

For example,

a running process may transition to the runnable state if it is interrupted by another process that has a higher priority. A sleeping process may transition to the runnable state when the event it is waiting for occurs. A stopped process may be resumed by sending it a signal. A zombie process may be reaped by its parent process by calling the wait() or 4aited() system call.

10. **Explain File Permission groups in Linux ?**

File permission groups in Linux are a way to organize users and groups of users so that you can easily control who has access to your files and directories.

There are three types of file permission groups in Linux:

- Owner: The user who owns the file.
- Group: The group to which the file belongs.
- Other: All other users on the system.

You can change the file permission groups of a file or directory using the chmod command. For example, to change the group ownership of a file called myfile.txt to the group developers, you would use the following command:

```
chmod g=developers myfile.txt
```

11.What is Umask?

Umask stands for User Mask. It is a four-digit octal number that is used to determine the default permissions for newly created files and directories. It is applied to the permissions that are granted to the owner, group, and other users.

The umask value is subtracted from the default permissions, which are 777 for files and 755 for directories. For example, a umask of 022 would result in default permissions of 655 for files and 733 for directories.

The umask value can be set using the umask command. For example, to set the umask to 022, you would use the following command:

```
umask 022
```

You can also view the current umask value using the **umask** command without any arguments.

12. How do you kill a process in Linux?

There are two ways to kill a process in Linux:

- Using the kill command. The kill command sends a signal to a process to terminate it. The most common signal to send is SIGTERM, which terminates the process gracefully. To send the SIGTERM signal to a process with the PID 1234, you would use the following command:

```
kill 1234
```

- Using the pkill command. The pkill command kills processes based on their name or other criteria. For example, to kill all processes with the name firefox, you would use the following command:

```
pkill firefox
```

13. What is a zombie process and what could be the cause of it?

A zombie process is a process that has finished executing but still has an entry in the process table. This is because the parent process has not yet reaped its exit status.

There are a few things that can cause a zombie process:

- The parent process may exit before the child process.
- The parent process may be waiting for a different child process to exit.
- The parent process may have crashed.

To fix a zombie process, you can reap it using the **wait()** or **6aited()** system call.

14. Explain what [echo "1" > /proc/sys/net/ipv4/ip_forward] does.?

The command echo "1" > /proc/sys/net/ipv4/ip_forward enables IP forwarding on a Linux system. IP forwarding is a process by which a Linux system routes network packets to other networks. It is typically used on routers and gateways, but it can also be used on other Linux systems to create a transparent bridge between two networks.

To enable IP forwarding, the file /proc/sys/net/ipv4/ip_forward must contain a value of 1. This can be done using the echo command, as shown in the example comman

Here are some examples of situations where you might want to enable IP forwarding on a Linux system:

- You are using the system as a router or gateway.
- You are using the system to create a transparent bridge between two networks.
- You are using the system to host a VPN server.

15. How do you troubleshoot memory performance issues? Please explain the details?

To troubleshoot memory performance issues, you can follow these steps:

- Identify the problem. The first step is to identify the specific memory performance issue that you are experiencing. Are you seeing high memory usage? Are applications crashing due to out-of-memory errors? Once you have identified the problem, you can start to troubleshoot it.

- Check the system logs. The system logs can provide valuable information about memory usage and performance. You can use the dmesg command to view the kernel logs, and the syslog command to view the system logs.

- Use a system monitoring tool. A system monitoring tool can provide real-time information about memory usage and performance. Some popular system monitoring tools include top, htop, and vmstat.

16.Mention one command which shows disks partitions sizes and types?

One command which shows disks partitions sizes and types is **fdisk -l.** This command lists all of the partitions on all of the disks in your system, along with their sizes and types.

To use the **fdisk -l** command, simply open a terminal window and type the following command:

```
fdisk -l
```

17. **Command shows free inodes on mounted filesystems?**

The command **df -i** shows free inodes on mounted filesystems.

The **df** command is used to display information about the file systems on a system. The **-i** option tells the **df** command to display the number of free inodes on each mounted filesystem.
To use the **df -i** command, simply open a terminal window and type the following command:

```
df -i
```

18. **Command shows free space on mounted file systems?**

The command **df -h** shows free space on mounted file systems.

The df command is used to display information about the file systems on a system. The -h option tells the df command to display the file system sizes in human-readable format
(e.g., MB, GB, TB).

To use the **df -h** command, simply open a terminal window and type the following command:

```
df -h
```

19. **How do you limit memory usage for commands?To limit memory usage for commands, you can use the ulimit command or the cgroups subsystem.**

Using the ulimit command

- The ulimit command is a built-in Linux command that allows you to set limits on various resources, including memory usage. To limit the memory usage for a command using ulimit, you can use the -m option.

For example, to limit the memory usage for the command my_command to 1GB, you would use the following command:

```
ulimit -m 1024
```

- You can also use the ulimit command to set soft limits and hard limits on resource usage. Soft limits can be exceeded, but hard limits cannot. To set a hard limit on memory usage, use the -l option. For example, to set a hard limit on memory usage of 1GB, you would use the following command:

```
ulimit -l 1024
```

20.Command shows top disk users in the current dir?

There is no direct command to show the top disk users in the current directory. However, you can use the following commands to get the same information:

```
du -a | sort -n -r | head -n 5
```

This command will list all of the files and directories in the current directory, including their sizes.

The -a option tells the du command to include hidden files and directories.

21.Command shows files by size, biggest file will be displayed last?

To display files by size, with the biggest file displayed last, you can use the following command:

```
ls -lhS
```

22. command lists the open files associated with your application ?

To list the open files associated with your application in Linux, you can use the lsof command. The lsof command lists information about open files, including the process that has opened them.

To list the open files associated with your application, simply run the following command:

```
lsof -p <pid>
```

23. How do you check the permissions of each directory to a file?

To check the permissions of each directory to a file in Linux, you can use the following command:

```
getfacl <path/to/file>
```

For example, to check the permissions of each directory to the file /path/to/file.txt, you would use the following command:

```
getfacl /path/to/file.txt
```

24. **How do you find who is logged in?**

There are a few ways to find who is logged in on a Linux system:

Use the who command. The who command displays a list of all users who are currently logged in to the system. To use the who command, simply open a terminal window and type the following command:

```
Who
```

25. How do you get the full path of a file in Linux?

There are a few ways to get the full path of a file in Linux:

Use the pwd command.

The pwd command prints the current working directory. To get the full path of a file, simply navigate to the directory where the file is located and usen the cd/path/to/directory and enter pwd to get the full path

PART – B

1. Launch two servers and establish password less authentication between them.

Step1: Launch two linux server in aws account server A(master) and server B(slave) and connect to both the servers

Step2: create a new user in server(slave) by using these commands

Sudo adduser <username>

Sudo passwd <username>

Enter the password and confirm it

Step3: change the directory to the created username

Su - <username>

Step4: Give some privileged premissions using below command

Sudo visudo

Go inside the file and add this line

<username> ALL=(ALL) NOPASSWD ALL

Step5: Do configuration changes in both the **server A** and **server B**

Sudo nano/etc/ssh/sshd_config

Inside the configuration file change the following settings

#**Enable authentication password** yes

# Pub authentication key yes

Step6: After the changes we must restart the service using below command

Restart it using sudo systemctl restart sshd

Step7: Remove password in master server and connect to instance without password(main server)

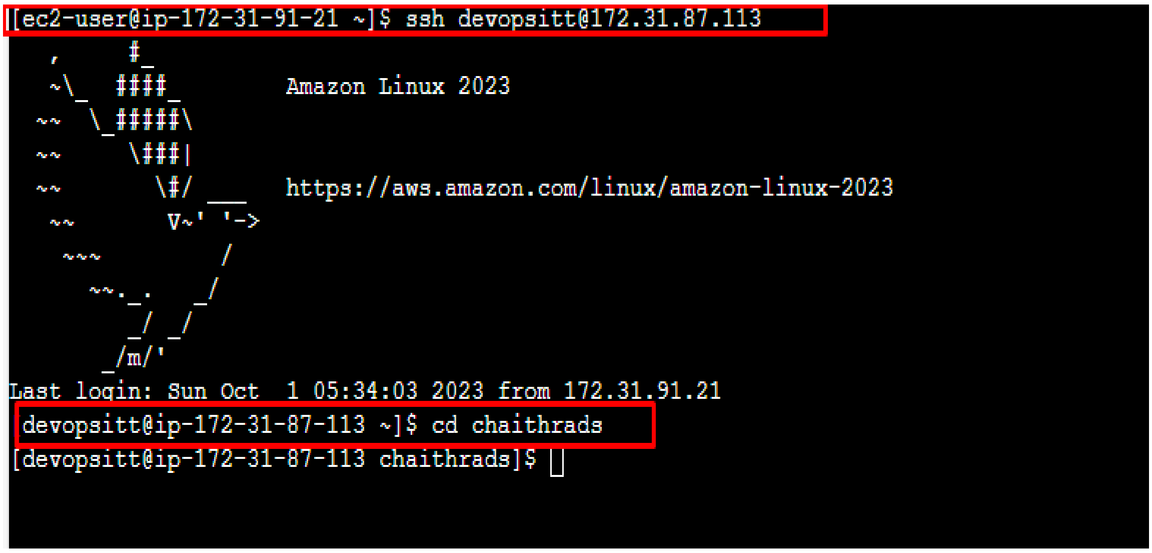
ssh-keygen # Generate an SSH key pair

Cd .ssh #change the directory

Ssh-copy-id-devopsitt@public # copy the public key

Step8: we can now do password-less authentication using below command

SSh username@IP



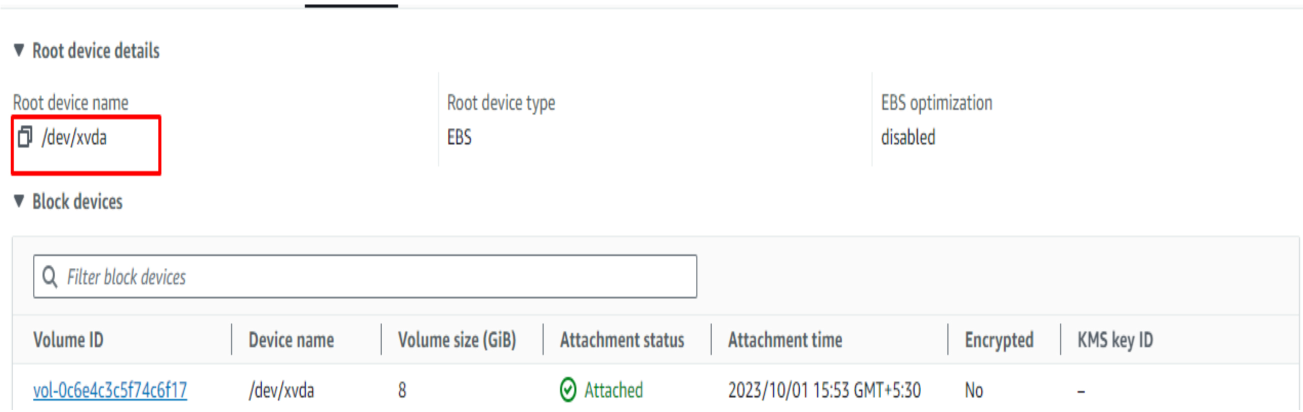
```
[ec2-user@ip-172-31-91-21 ~]$ ssh devopsitt@172.31.87.113
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
        _/m/'
Last login: Sun Oct  1 05:34:03 2023 from 172.31.91.21
 devopsitt@ip-172-31-87-113 ~]$ cd chaithrads
[devopsitt@ip-172-31-87-113 chaithrads]$ 
```

2.If the pem key of the server is lost, how will you retrieve the key? Launch the instance delete the pem key and again retrieve the key.

Step1: Launch a new instance with same configuration has your old instance

Step2: Stop your old instance and detach the volume of the old instance

Step3: Now you need to detach the volume from the new web First EC2 instance and attach this volume to the My backup server (second EC2 instance) and note down the root device name in the image.



▼ Root device details

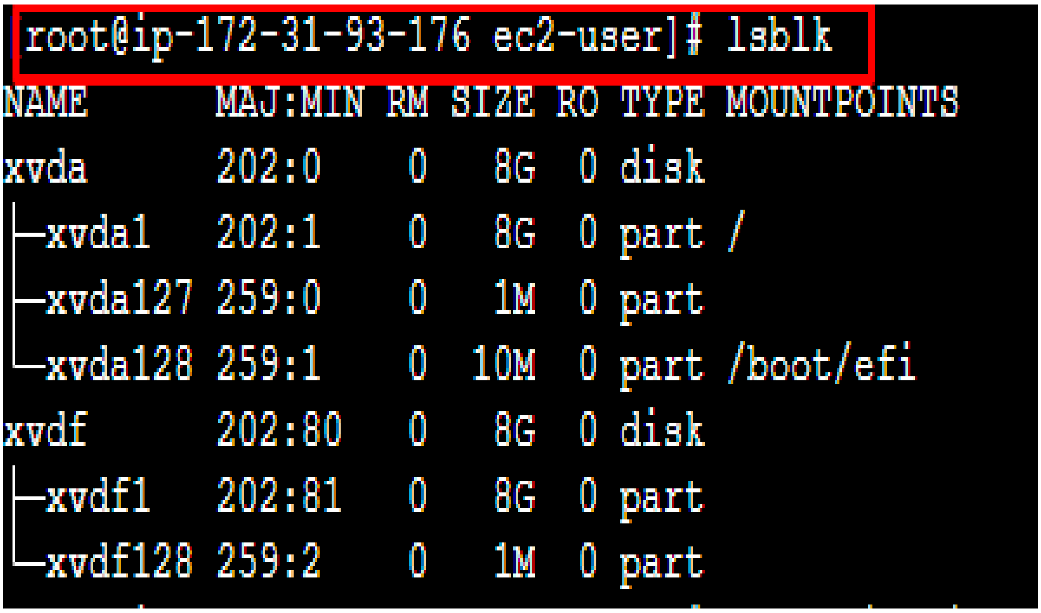
Root device name	Root device type	EBS optimization
/dev/xvda	EBS	disabled

▼ Block devices

Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time	Encrypted	KMS key ID
vol-0c6e4c3c5f74c6f17	/dev/xvda	8	Attached	2023/10/01 15:53 GMT+5:30	No	-

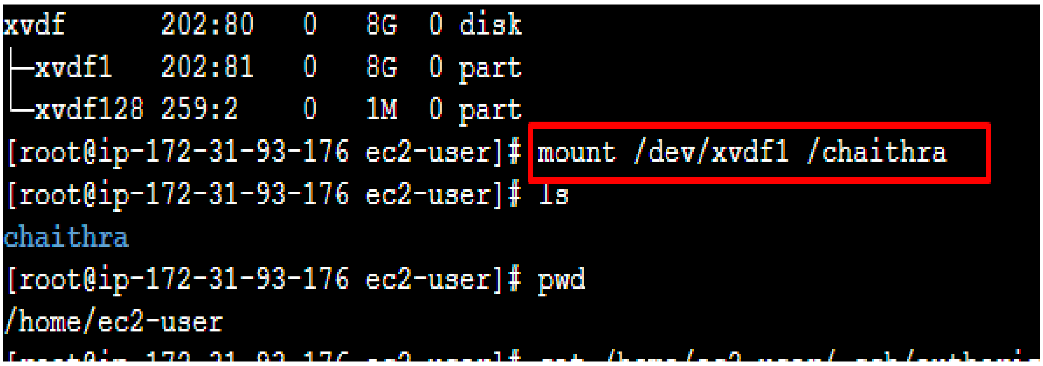
Step4: We must SSH to the new instance (My backup server) and attach the Volume of the First EC2 instance to this instance

Step5: listing the volume attached to this EC2 instance by using the command “**lsblk**”



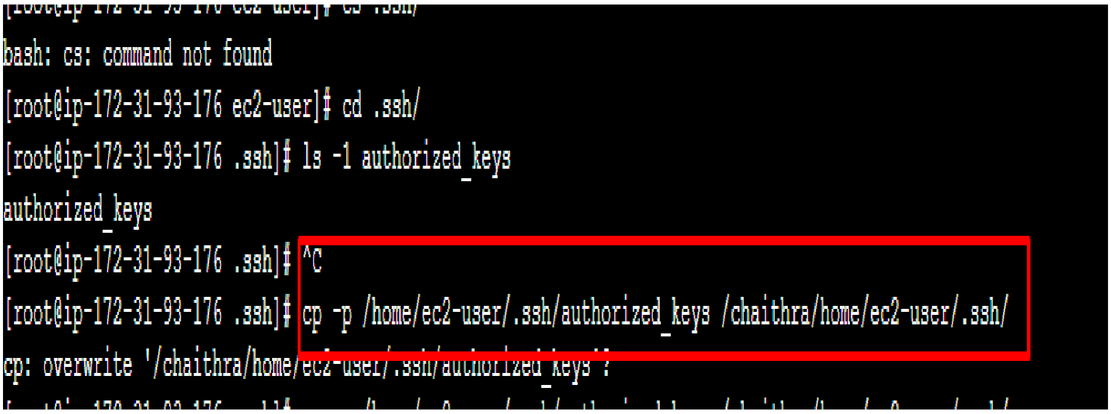
```
root@ip-172-31-93-176 ec2-user]# lsblk
NAME      MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda      202:0    0   8G  0 disk
├─xvda1   202:1    0   8G  0 part /
├─xvda127 259:0    0   1M  0 part
└─xvda128 259:1    0  10M  0 part /boot/efi
xvdf      202:80   0   8G  0 disk
├─xvdf1   202:81   0   8G  0 part
└─xvdf128 259:2    0   1M  0 part
```

Step6: To mount this disk, First, create the directory (in my case the directory name is data). To mount the disk **/dev/xvdf1** on the data directory using the mount command.



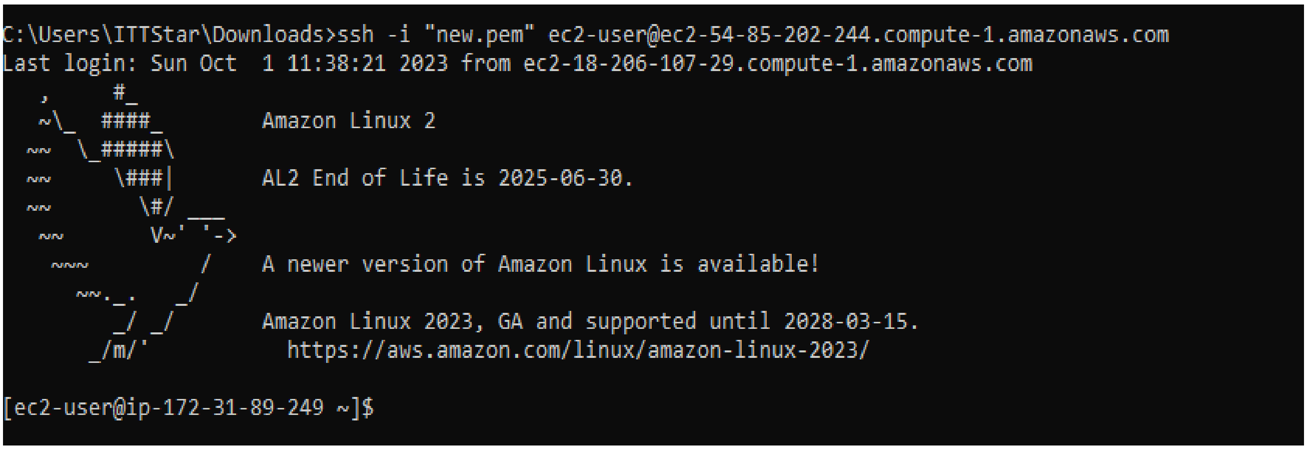
```
xvdf        202:80   0   8G  0 disk
├─xvdf1     202:81   0   8G  0 part
└─xvdf128 259:2    0   1M  0 part
[root@ip-172-31-93-176 ec2-user]# mount /dev/xvdf1 /chaithra
[root@ip-172-31-93-176 ec2-user]# ls
chaithra
[root@ip-172-31-93-176 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-93-176 ec2-user]# cat /home/ec2-user/.ssh/authori
```

Step7: Now switch to the directory /data/home/ec2-user/.ssh. Now, you have to replace **authorized_keys** of the New web server (First EC2 instance) with My Backup server (Second EC2 instance) using cp command



```
[root@ip-172-31-93-176 ec2-user]# cs .ssh/
bash: cs: command not found
[root@ip-172-31-93-176 ec2-user]# cd .ssh/
[root@ip-172-31-93-176 .ssh]# ls -1 authorized_keys
authorized_keys
[root@ip-172-31-93-176 .ssh]# ^C
[root@ip-172-31-93-176 .ssh]# cp -p /home/ec2-user/.ssh/authorized_keys /chaithra/home/ec2-user/.ssh/
cp: overwrite '/chaithra/home/ec2-user/.ssh/authorized_keys'?
[root@ip-172-31-93-176 .ssh]# cp -p /home/ec2-user/.ssh/authorized_keys /chaithra/home/ec2-user/.ssh/
```

Step8: Now detach the volume from this instance and again re-attach this volume with name noted down previously to the first instance and using the new pem file we can access the instance.



```
C:\Users\ITTStar\Downloads>ssh -i "new.pem" ec2-user@ec2-54-85-202-244.compute-1.amazonaws.com
Last login: Sun Oct  1 11:38:21 2023 from ec2-18-206-107-29.compute-1.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2
  ~~  \_#####\
  ~~     \###|       AL2 End of Life is 2025-06-30.
  ~~       \#/ ___
   ~~       V~' '->
    ~~~         /    A newer version of Amazon Linux is available!
      ~~._.   _/
         _/ _/       Amazon Linux 2023, GA and supported until 2028-03-15.
       _/m/'           https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-89-249 ~]$
```

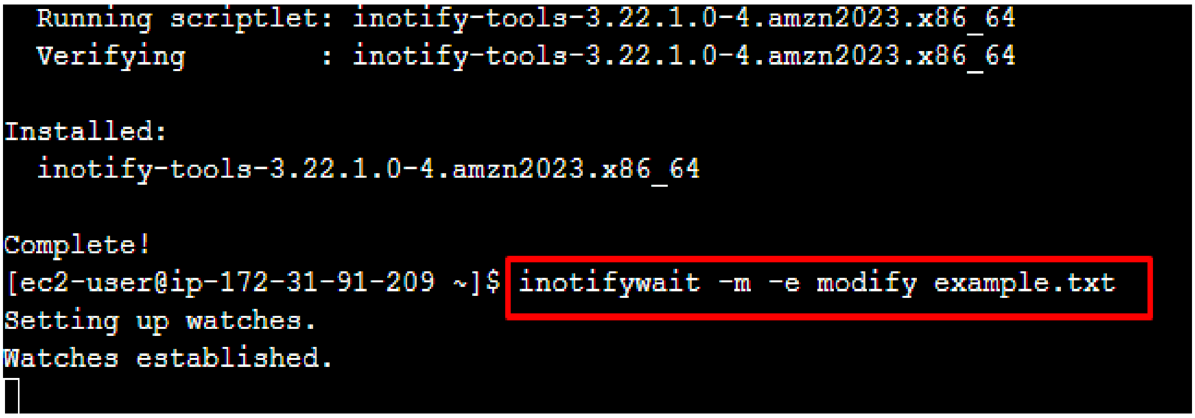
3.) How do you run the command every time a file is modified?

Step1: We can use the **inotify** tool to run a command every time a file is modified on a Linux server.so first we need to install the inotify in the server by using the below command

sudo yum install inotify-tools

Step2: we can use the inotifywait command to monitor a specific file or directory for modifications.To setup on the specific file we can use below command.

inotifywait -m -e modify example.txt



```
  Running scriptlet: inotify-tools-3.22.1.0-4.amzn2023.x86_64
  Verifying        : inotify-tools-3.22.1.0-4.amzn2023.x86_64

Installed:
  inotify-tools-3.22.1.0-4.amzn2023.x86_64

Complete!
[ec2-user@ip-172-31-91-209 ~]$ inotifywait -m -e modify example.txt
Setting up watches.
Watches established.
```

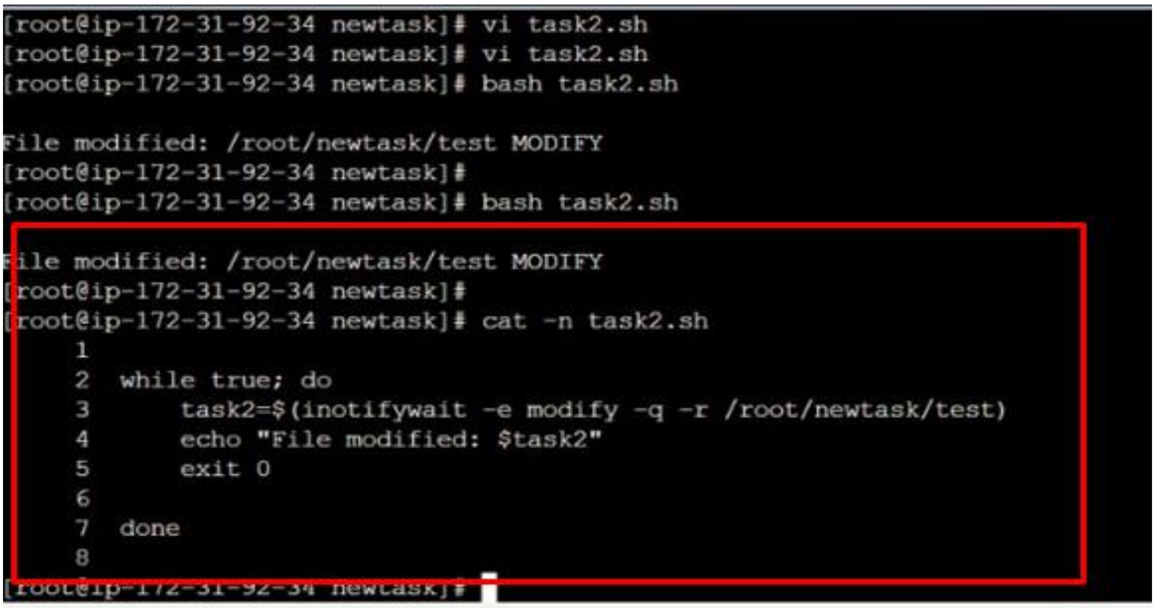
Step3: create a script and with the **inotifywait** command to monitor a file for modifications and execute your desired command when a change occurs.

Here is an example script

#!/bin/bash

```
# Define the file to monitor and the command to execute
            file_to_monitor="example.txt"
 command_to_execute="echo 'File modified: $file_to_monitor'
            # Start monitoring the file
            while true; do
               inotifywait -e modify "$file_to_monitor"
               eval "$command_to_execute"
            Done
```

Step4:save the script and run the script in background using **chmod +x watch_file.sh** and everytime there is change in the file it will start running the command.



```
[root@ip-172-31-92-34 newtask]# vi task2.sh
[root@ip-172-31-92-34 newtask]# vi task2.sh
[root@ip-172-31-92-34 newtask]# bash task2.sh

File modified: /root/newtask/test MODIFY
[root@ip-172-31-92-34 newtask]#
[root@ip-172-31-92-34 newtask]# bash task2.sh

File modified: /root/newtask/test MODIFY
[root@ip-172-31-92-34 newtask]#
[root@ip-172-31-92-34 newtask]# cat -n task2.sh
     1
     2  while true; do
     3      task2=$(inotifywait -e modify -q -r /root/newtask/test)
     4      echo "File modified: $task2"
     5      exit 0
     6
     7  done
     8
[root@ip-172-31-92-34 newtask]#
```

4.) User root has created a file "TestFile" which must not be opened by anyone except root and another user "Kiran", how can this be done?

Step1: Set ownership for the file which you want to restrict by using following command and Restrict Access Set the permissions on "TestFile" to allow only the root user to read, write, and execute the file, and deny all access to others

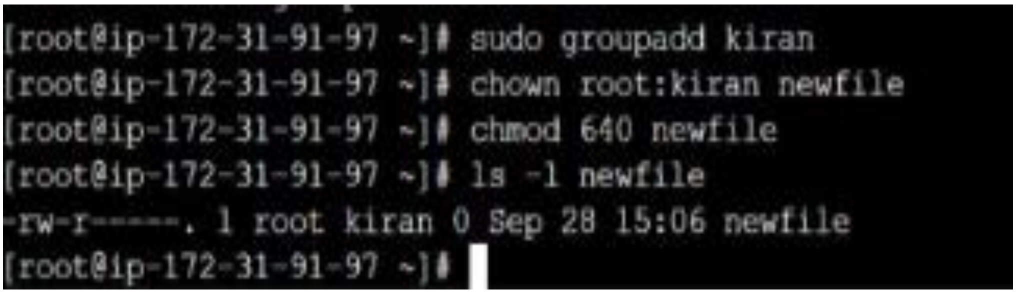
sudo chown root:root <TestFile>

sudo chmod 700 TestFile

Step2: Grant Access to "username" To allow the "Kiran" user to access the file, you need to add "Kiran" to a group, and then grant that group access to the file. Here, we'll create a group named "kiran_group" and add "Kiran" to it using below commands

sudo groupadd kiran_group

sudo usermod -aG kiran_group

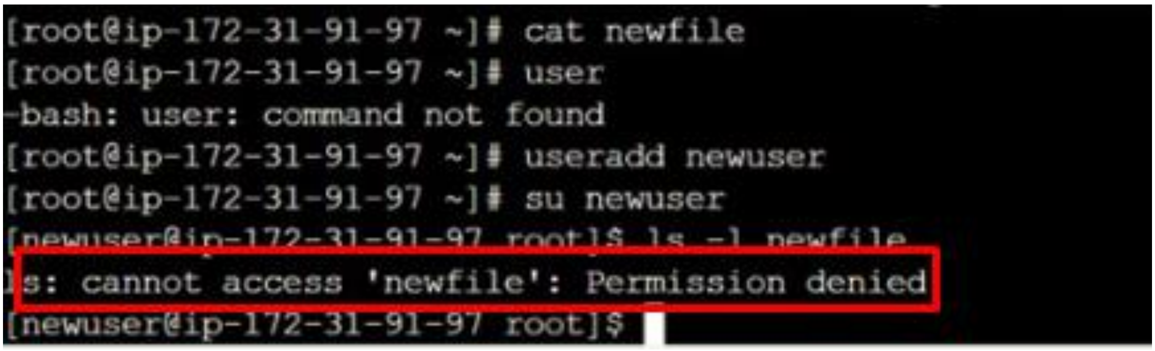


```
[root@ip-172-31-91-97 ~]# sudo groupadd kiran
[root@ip-172-31-91-97 ~]# chown root:kiran newfile
[root@ip-172-31-91-97 ~]# chmod 640 newfile
[root@ip-172-31-91-97 ~]# ls -l newfile
-rw-r-----. 1 root kiran 0 Sep 28 15:06 newfile
[root@ip-172-31-91-97 ~]#
```

Step3: Change the group ownership of "TestFile" to the newly created "kiran_group" by using the below command

sudo chown :kiran_group TestFile

Step4: Allow the group "kiran_group" to read and write the file and this file will only be accessed by “rootuser” and user “kiran”



```
[root@ip-172-31-91-97 ~]# cat newfile
[root@ip-172-31-91-97 ~]# user
-bash: user: command not found
[root@ip-172-31-91-97 ~]# useradd newuser
[root@ip-172-31-91-97 ~]# su newuser
[newuser@ip-172-31-91-97 root]$ ls -l newfile
ls: cannot access 'newfile': Permission denied
[newuser@ip-172-31-91-97 root]$
```

5.)Cannot SSH as root/user to the server how do you trouble shoot this issue?

Here are some steps we can take to troubleshoot why we are unable to ssh as root/user to the server:

- check that we are using correct credentials.
- make sure that we are using correct username and password for the root/user account.
- we can try resetting the password if we have forgotten it.
- Check that the SSH service is running. we can use the following command to check if the SSH service is running service **ssh status**
- If the SSH service is not running, you can start it using the following command:service **ssh start**

Check that the SSH port is open. The default SSH port is 22. You can use the following command to check if the SSH port is open:

netstat -an | grep 22

If the SSH port is not open, you can open it using the following command:

firewall-cmd --permanent --zone=public --add-port=22/tcp

Check that the SSH configuration file is correct. The SSH configuration file is located at **/etc/ssh/sshd_config.** You can open the file using a text editor such as nano or vim. Make sure that the following lines are uncommented:

PermitRootLogin :yes

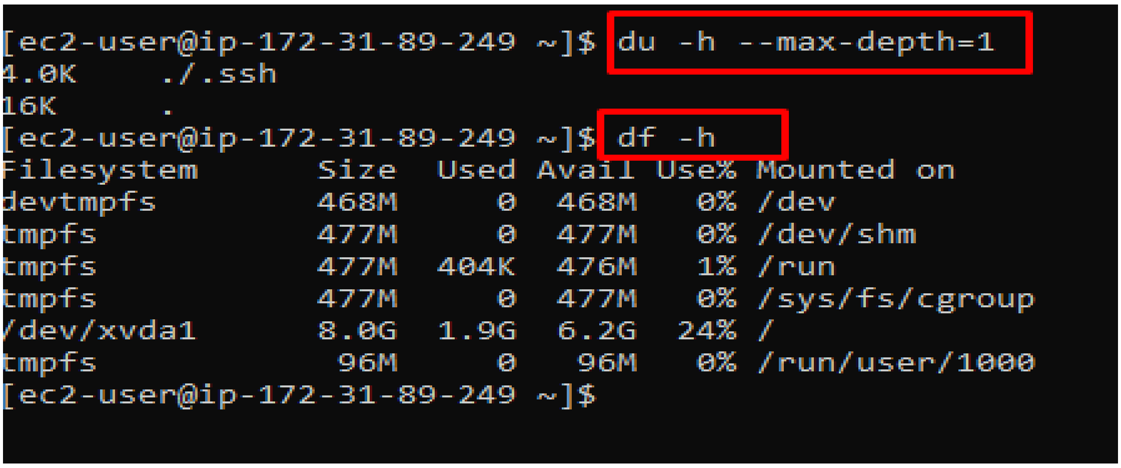
PasswordAuthentication:yes

Try restarting the server. If you have tried all of the above steps and you are still unable to SSH as root/user to the server, you can try restarting the server. This may resolve the issue.

If you are still having problems, you can contact your server provider for assistance.

6)Disk Space is full and cannot add/extend the disk space how do you resolve the issue?

1.Identify Disk Usage: Begin by identifying which directories or files are consuming the most space. You can use commands like **du** (disk usage) and **df** (disk free) to get an overview of your disk usage.



```
[ec2-user@ip-172-31-89-249 ~]$ du -h --max-depth=1
4.0K    ./.ssh
16K     .
[ec2-user@ip-172-31-89-249 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        468M     0  468M   0% /dev
tmpfs           477M     0  477M   0% /dev/shm
tmpfs           477M  404K  476M   1% /run
tmpfs           477M     0  477M   0% /sys/fs/cgroup
/dev/xvda1      8.0G  1.9G  6.2G  24% /
tmpfs            96M     0   96M   0% /run/user/1000
[ec2-user@ip-172-31-89-249 ~]$
```

2. Delete Unnecessary Files and Directories Identify and delete files and directories that are no longer needed. Use the rm command to remove files and the **rmdir** or **rm -r** command to remove directories.

3. Clear Log Files: Log files can sometimes take up a significant amount of disk space. You can use the **truncate** or **echo** command to clear the contents of log files without deleting them.

truncate -s 0 /var/log/some-log-file.log

4. Compress and Archive Files: Compressing and archiving files can save space. You can use tools like **tar** and **zip** to create compressed archives of files and directories.

```
tar -czvf archive.tar.gz /path/to/files #tar file

zip -r archive.zip /path/to/files #zip file
```

5. Cleanup Temporary Files: Temporary files, cache files, and old backups can accumulate over time. Clean up these files using commands like **find** and **rm**.

6.Check for Unneeded Software or Packages: Review the software and packages installed on your system. If you have unused or redundant software, consider uninstalling it to free up space. Use package management tools like **apt, yum,** or **dnf** to uninstall packages.

7. Monitor and Automate Cleanup: To prevent future disk space issues, set up monitoring and automation scripts to periodically clean up or archive files and directories that are no longer needed.

7) Create two servers and send files from one server to another server.

Step1: Create two servers and send files from one server to another server using the commands.

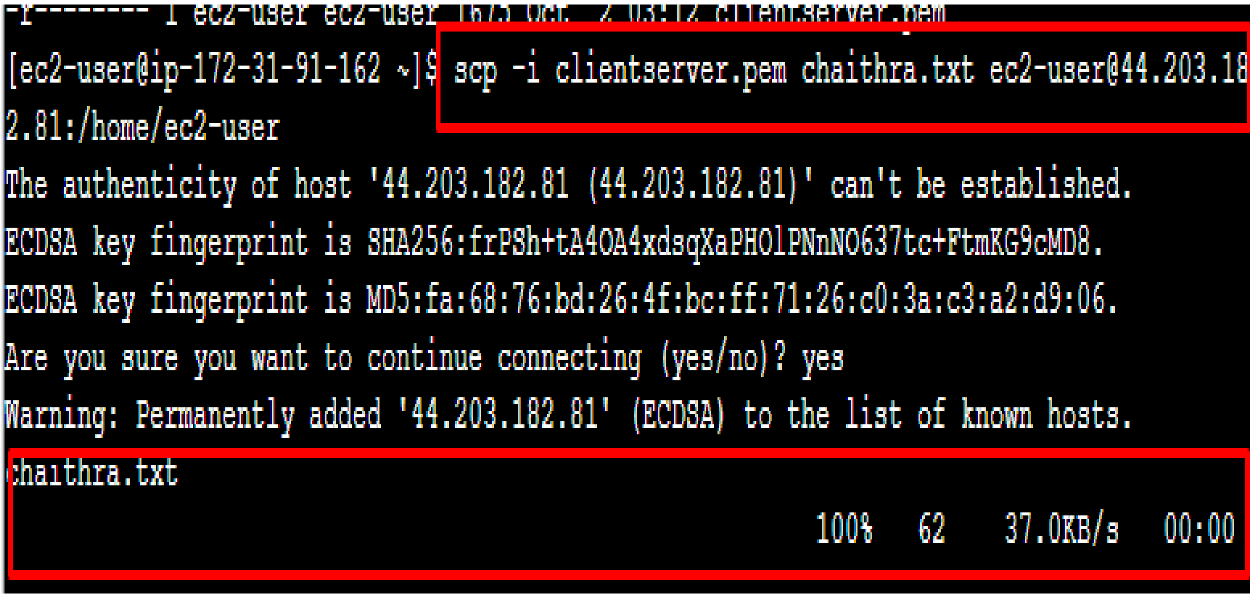
Scp command: It is a straightforward way to copy files from one Linux server to another.

```
 scp –i /path/to/source/file keyfile.pem
user@destination_server:/path/to/destination/
```

Rsync command: This command will initiate the rsync operation directly between the two remote servers using SSH as the transport mechanism.

```
rsync -avz -e 'ssh -i /path/to/your/ssh/key' user@source_server:/path/to/source/
user@destination_server:/path/to/destination/
```

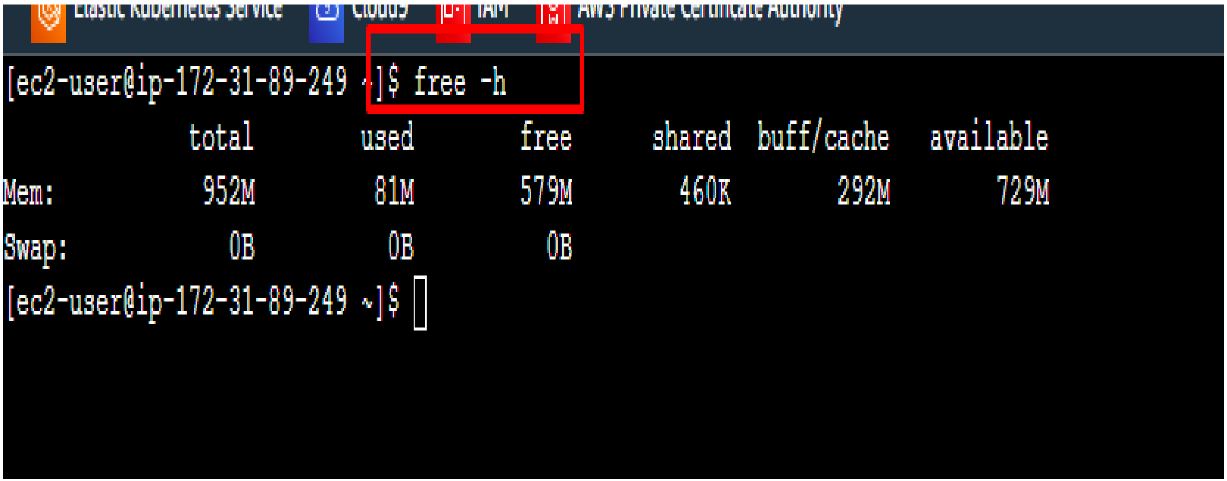
Output:



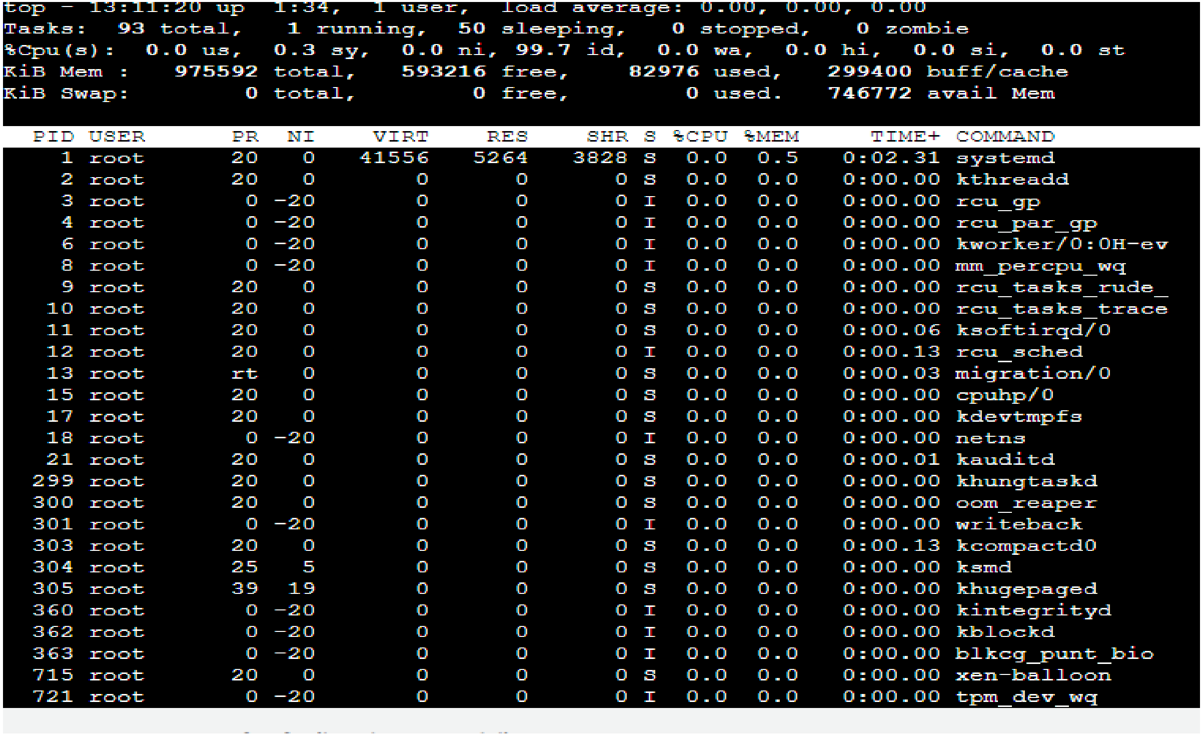
```
-r-------- 1 ec2-user ec2-user 1675 Oct  2 03:12 clientserver.pem
[ec2-user@ip-172-31-91-162 ~]$ scp -i clientserver.pem chaithra.txt ec2-user@44.203.18
2.81:/home/ec2-user
The authenticity of host '44.203.182.81 (44.203.182.81)' can't be established.
ECDSA key fingerprint is SHA256:frPSh+tA4OA4xdsqXaPHOlPNnNO637tc+FtmKG9cMD8.
ECDSA key fingerprint is MD5:fa:68:76:bd:26:4f:bc:ff:71:26:c0:3a:c3:a2:d9:06.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '44.203.182.81' (ECDSA) to the list of known hosts.
chaithra.txt
                                                        100%   62    37.0KB/s   00:00
```

8. Running Out of Memory? How do you check the memory and how do you resolve this issue?

Check Current Memory Usage: We can use **free** and **top** commands to check the current memory usage of your Linux server



```
[ec2-user@ip-172-31-89-249 ~]$ free -h
              total        used        free      shared  buff/cache   available
Mem:           952M         81M        579M        460K        292M        729M
Swap:            0B          0B          0B
[ec2-user@ip-172-31-89-249 ~]$ 
```



```
top - 13:11:20 up  1:34,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  93 total,   1 running,  50 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   975592 total,   593216 free,    82976 used,   299400 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   746772 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
    1 root      20   0   41556   5264   3828 S  0.0  0.5   0:02.31 systemd
    2 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kworker/0:0H-ev
    8 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_tasks_rude_
   10 root      20   0       0      0      0 S  0.0  0.0   0:00.00 rcu_tasks_trace
   11 root      20   0       0      0      0 S  0.0  0.0   0:00.06 ksoftirqd/0
   12 root      20   0       0      0      0 I  0.0  0.0   0:00.13 rcu_sched
   13 root      rt   0       0      0      0 S  0.0  0.0   0:00.03 migration/0
   15 root      20   0       0      0      0 S  0.0  0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S  0.0  0.0   0:00.00 kdevtmpfs
   18 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 netns
   21 root      20   0       0      0      0 S  0.0  0.0   0:00.01 kauditd
  299 root      20   0       0      0      0 S  0.0  0.0   0:00.00 khungtaskd
  300 root      20   0       0      0      0 S  0.0  0.0   0:00.00 oom_reaper
  301 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 writeback
  303 root      20   0       0      0      0 S  0.0  0.0   0:00.13 kcompactd0
  304 root      25   5       0      0      0 S  0.0  0.0   0:00.00 ksmd
  305 root      39  19       0      0      0 S  0.0  0.0   0:00.00 khugepaged
  360 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kintegrityd
  362 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 kblockd
  363 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 blkcg_punt_bio
  715 root      20   0       0      0      0 S  0.0  0.0   0:00.00 xen-balloon
  721 root       0 -20       0      0      0 I  0.0  0.0   0:00.00 tpm_dev_wq
```

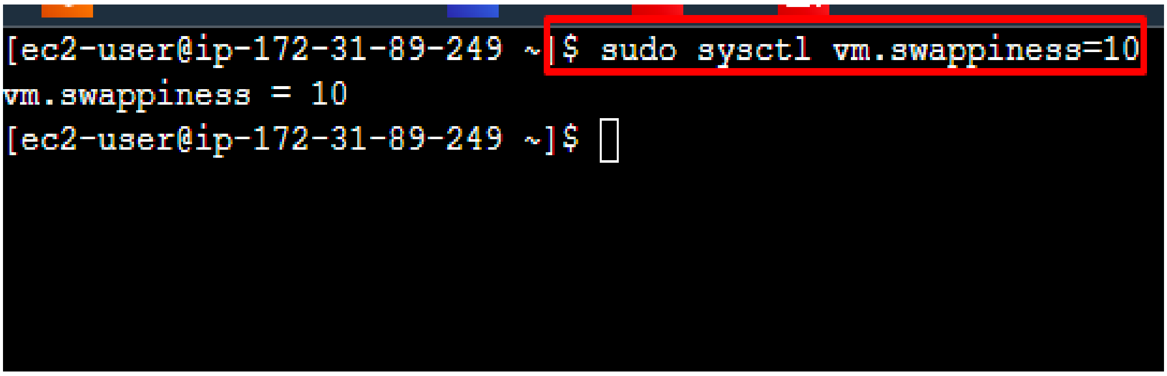
2. **View System Logs**: Check system logs (**/var/log/messages**, **/var/log/syslog**) for any memory-related error messages or issues

Resolving Memory Issues:

Identify Memory-Hungry Use the **top** or **htop** command to identify processes that are consuming a significant amount of memory. Look for any processes that are using more memory than expected.

Restart or Stop Problematic Processes: If you identify processes that are using excessive memory and are not critical, consider restarting or stopping them. You can use the kill or **pkill** command to terminate processes. Be cautious when killing processes, as it may affect system stability.

Adjust Swappiness: Swappiness determines how aggressively the Linux kernel uses swap space. You can adjust the swappiness value to control how often the system swaps data to disk.



```
[ec2-user@ip-172-31-89-249 ~]$ sudo sysctl vm.swappiness=10
vm.swappiness = 10
[ec2-user@ip-172-31-89-249 ~]$ 
```

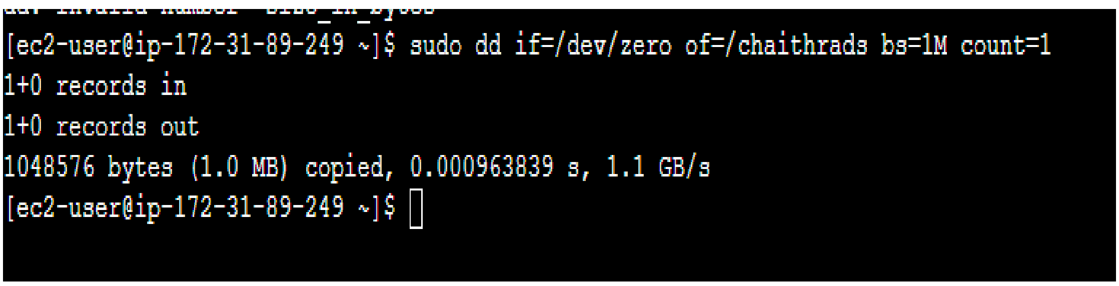
Add More RAM: If your server consistently runs out of memory, and you've optimized processes and swap space, consider adding more RAM to your server. This is the most effective way to improve memory performance.

Monitoring and Alerts:
Set up monitoring tools and alerts to notify you when memory usage exceeds certain thresholds. This allows you to take action before severe memory issues occur.

9. How do you add/Extend the Swap space?

1)To add a swap file we can use the below command

sudo dd if=/dev/zero of=/path/to/new/swapfile bs=1M count=size_in_bytes



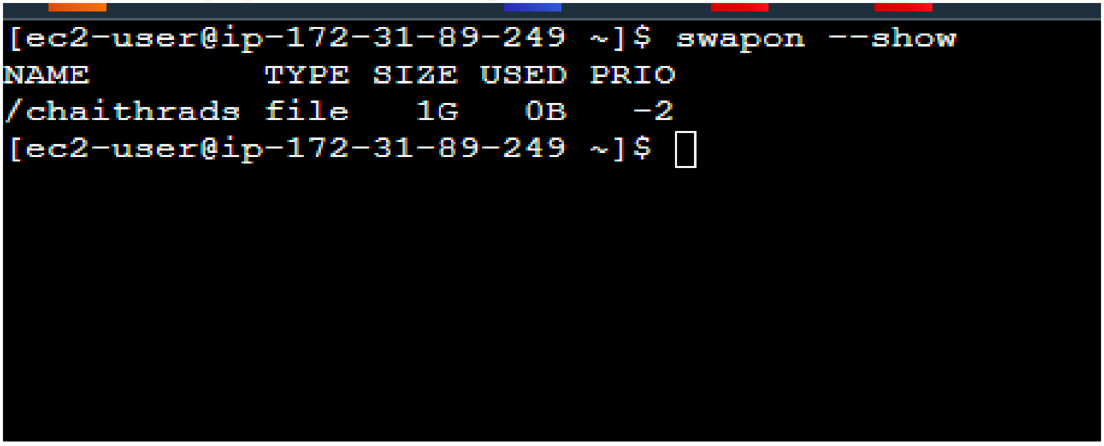
```
dd: invalid number 'size_in_bytes'
[ec2-user@ip-172-31-89-249 ~]$ sudo dd if=/dev/zero of=/chaithrads bs=1M count=1
1+0 records in
1+0 records out
1048576 bytes (1.0 MB) copied, 0.000963839 s, 1.1 GB/s
[ec2-user@ip-172-31-89-249 ~]$
```

2) **Set Appropriate Permissions by using the below command**

sudo chmod 600 /path/to/new/swapfile

3) **Initialize the Swap File and enable the swap file and swap file is created**

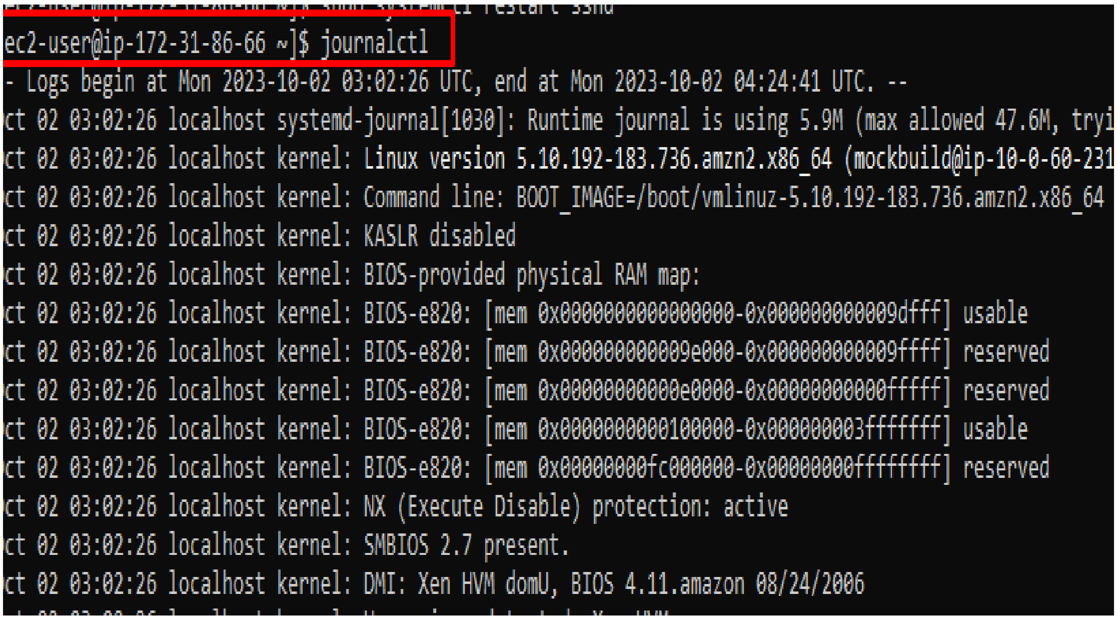
sudo mkswap /path/to/new/swapfile



```
[ec2-user@ip-172-31-89-249 ~]$ swapon --show
NAME        TYPE SIZE USED PRIO
/chaithrads file   1G   0B   -2
[ec2-user@ip-172-31-89-249 ~]$
```

10.) System unexpectedly reboot and process restart? How do you troubleshoot this issue.

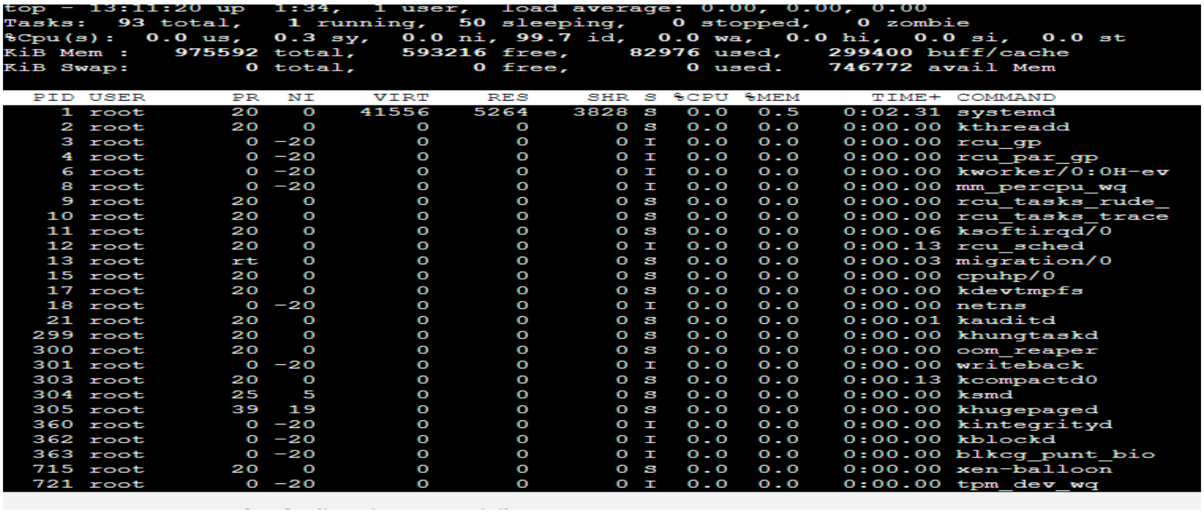
1) **Check System Logs**: Review system logs for any events leading up to the reboot. Common log files include **/var/log/syslog**, **/var/log/messages**, Using the **journalctl** command.



```
ec2-user@ip-172-31-86-66 ~]$ journalctl
- Logs begin at Mon 2023-10-02 03:02:26 UTC, end at Mon 2023-10-02 04:24:41 UTC. --
ct 02 03:02:26 localhost systemd-journal[1030]: Runtime journal is using 5.9M (max allowed 47.6M, tryi
ct 02 03:02:26 localhost kernel: Linux version 5.10.192-183.736.amzn2.x86_64 (mockbuild@ip-10-0-60-231
ct 02 03:02:26 localhost kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-5.10.192-183.736.amzn2.x86_64
ct 02 03:02:26 localhost kernel: KASLR disabled
ct 02 03:02:26 localhost kernel: BIOS-provided physical RAM map:
ct 02 03:02:26 localhost kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009dfff] usable
ct 02 03:02:26 localhost kernel: BIOS-e820: [mem 0x000000000009e000-0x000000000009ffff] reserved
ct 02 03:02:26 localhost kernel: BIOS-e820: [mem 0x00000000000e0000-0x00000000000fffff] reserved
ct 02 03:02:26 localhost kernel: BIOS-e820: [mem 0x0000000000100000-0x000000003fffffff] usable
ct 02 03:02:26 localhost kernel: BIOS-e820: [mem 0x00000000fc000000-0x00000000ffffffff] reserved
ct 02 03:02:26 localhost kernel: NX (Execute Disable) protection: active
ct 02 03:02:26 localhost kernel: SMBIOS 2.7 present.
ct 02 03:02:26 localhost kernel: DMI: Xen HVM domU, BIOS 4.11.amazon 08/24/2006
```

2) Identify the processes that are restarting: Once you have identified the processes that are restarting, you can try to determine why they are restarting. You can use the **ps aux** command to view a list of all running processes. To view the details of a specific process, you can use the **top** command.

3) **Review System Metrics:** Examine system metrics like CPU, memory, and disk usage using commands like **top, htop,** or monitoring tools like **sar.** High resource utilization could lead to system instability.



```
top - 13:11:20 up  1:34,  1 user,  load average: 0.00, 0.00, 0.00
Tasks:  93 total,   1 running,  50 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.3 sy,  0.0 ni, 99.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :   975592 total,   593216 free,    82976 used,   299400 buff/cache
KiB Swap:        0 total,        0 free,        0 used.   746772 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
    1 root      20   0   41556   5264   3828 S   0.0  0.5   0:02.31 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kthreadd
    3 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_gp
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 rcu_par_gp
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0H-ev
    8 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_wq
    9 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_tasks_rude_
   10 root      20   0       0      0      0 S   0.0  0.0   0:00.00 rcu_tasks_trace
   11 root      20   0       0      0      0 S   0.0  0.0   0:00.06 ksoftirqd/0
   12 root      20   0       0      0      0 I   0.0  0.0   0:00.13 rcu_sched
   13 root      rt   0       0      0      0 S   0.0  0.0   0:00.03 migration/0
   15 root      20   0       0      0      0 S   0.0  0.0   0:00.00 cpuhp/0
   17 root      20   0       0      0      0 S   0.0  0.0   0:00.00 kdevtmpfs
   18 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 netns
   21 root      20   0       0      0      0 S   0.0  0.0   0:00.01 kauditd
  299 root      20   0       0      0      0 S   0.0  0.0   0:00.00 khungtaskd
  300 root      20   0       0      0      0 S   0.0  0.0   0:00.00 oom_reaper
  301 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 writeback
  303 root      20   0       0      0      0 S   0.0  0.0   0:00.13 kcompactd0
  304 root      25   5       0      0      0 S   0.0  0.0   0:00.00 ksmd
  305 root      39  19       0      0      0 S   0.0  0.0   0:00.00 khugepaged
  360 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kintegrityd
  362 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kblockd
  363 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 blkcg_punt_bio
  715 root      20   0       0      0      0 S   0.0  0.0   0:00.00 xen-balloon
  721 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 tpm_dev_wq
```

4) **Examine Kernel Logs:** Check the kernel logs, which might provide information about a kernel panic. Use the **dmesg** command to view the kernel ring buffer

5) **Check CloudTrail Logs and CloudWatch metrics (AWS Server):** AWS CloudTrail logs can provide valuable audit and event history. Review

CloudTrail logs to see if any unusual activities or API calls might have triggered the reboot and can see the utilization in metrics.

6) **Check for third-party application updates:** Third-party applications that are installed on your AWS Linux server may also release updates. It is important to keep these applications up to date as well.

7) **Examine Application Logs:** Inspect application-specific logs for errors or issues that may have caused the processes to restart. Application logs can provide valuable insights into the root cause.

11) HTTP error 403: forbidden yum occurs when we try to install a package using yum?How do you troubleshoot the issue?

- **Check the yum configuration file:** The yum configuration file is located at /etc/yum.conf. Check this file to make sure that the baseurl and mirrorlist entries are correct. You can also try disabling any third-party repositories that you are not using.

- **Check the yum cache:** Sometimes a corrupted yum cache can cause this error. To clear the yum cache, run the following command:

```
yum clean all
```

- **Check the proxy settings:** If you are using a proxy server, make sure that the proxy settings are configured correctly in the yum configuration file.

- **Check the yum version:** Make sure that you are using the latest version of yum. To update yum, run the following command:

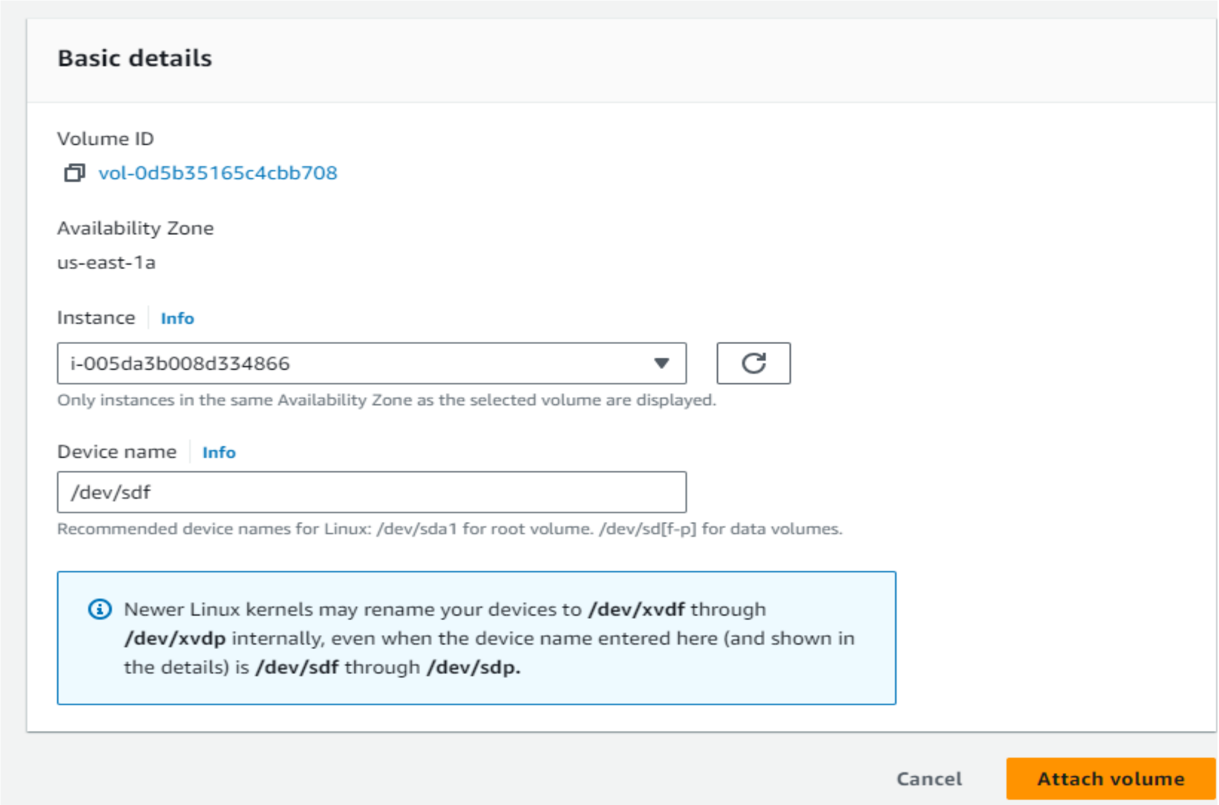
```
yum update yum
```

- **Check the repository permissions:** Make sure that the yum repository files and directories have the correct permissions. You can run the following command to check the permissions:

```
ls -l /etc/yum.repos.d/*
```

12). Attach a new volume to the server and mount it to a specific path of a file system.

Step1: Create a new ebs volume in the aws management console and attach the volume to your running ec2 instance.



Basic details

Volume ID

vol-0d5b35165c4cbb708

Availability Zone

us-east-1a

Instance Info

i-005da3b008d334866

Only instances in the same Availability Zone as the selected volume are displayed.

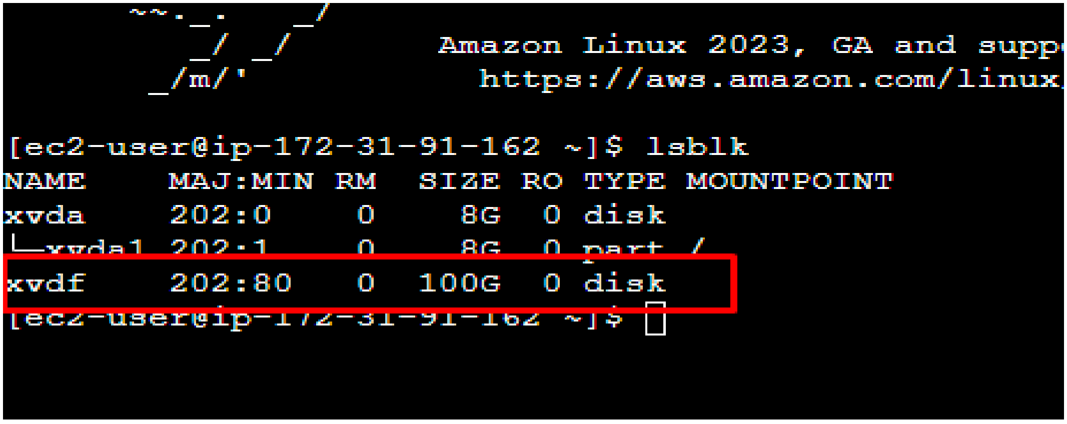
Device name Info

/dev/sdf

Recommended device names for Linux: /dev/sda1 for root volume. /dev/sd[f-p] for data volumes.

Newer Linux kernels may rename your devices to **/dev/xvdf** through **/dev/xvdp** internally, even when the device name entered here (and shown in the details) is **/dev/sdf** through **/dev/sdp.**

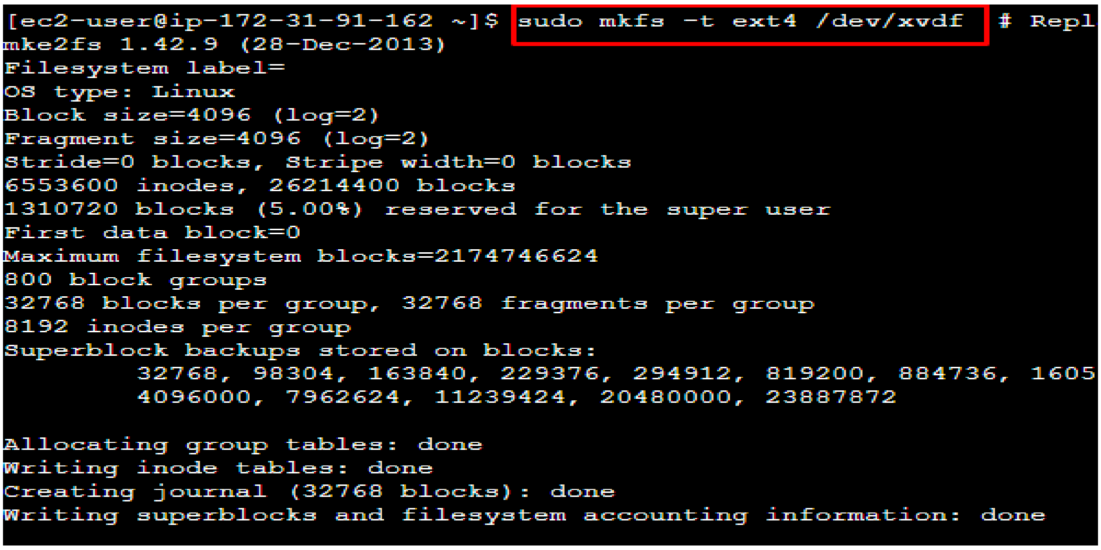
Step2: ssh into your instance and list all available disks using "lsblk" command and the new volume will typically appear as /dev/xvd*



```
       ~~._.   _/
          _/ _/        Amazon Linux 2023, GA and supp
        _/m/'            https://aws.amazon.com/linux

[ec2-user@ip-172-31-91-162 ~]$ lsblk
NAME     MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda     202:0    0    8G  0 disk
└─xvda1  202:1    0    8G  0 part /
xvdf     202:80   0  100G  0 disk
[ec2-user@ip-172-31-91-162 ~]$
```

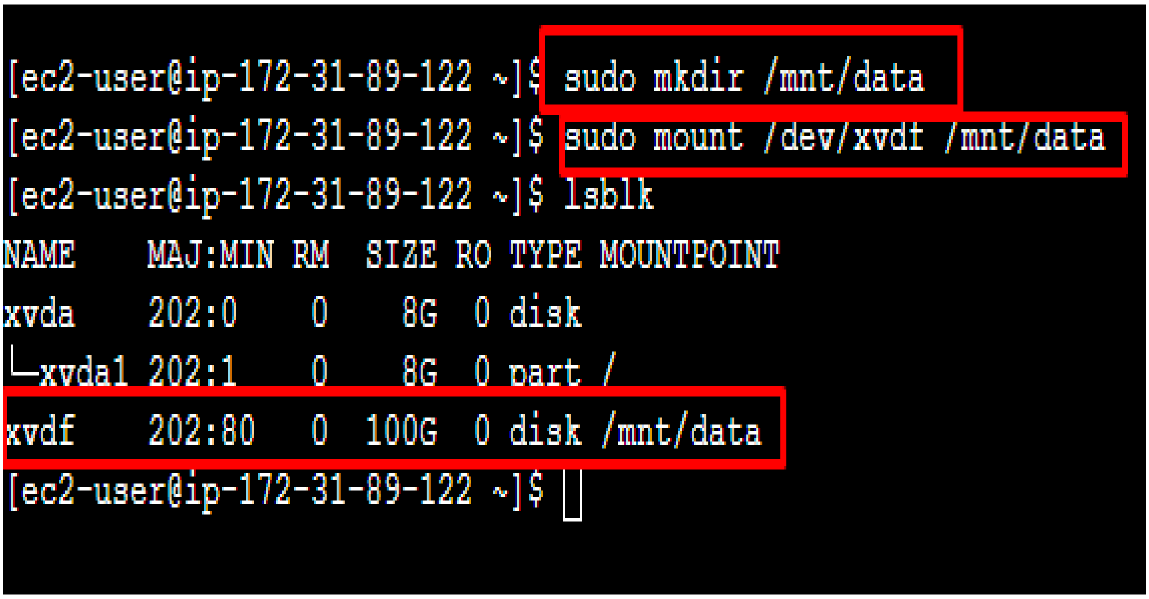
Step3: If the volume is new and hasn't been formatted yet, we need to create a filesystem on it. You can use **mkfs** for this purpose.



```
[ec2-user@ip-172-31-91-162 ~]$ sudo mkfs -t ext4 /dev/xvdf  # Repl
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
6553600 inodes, 26214400 blocks
1310720 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2174746624
800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done
```

Step4:create a mount point and mount the volume to the specific file path Using **sudo mkdir /mnt/data** and **sudo mount /dev/xvdf /mnt/data**



```
[ec2-user@ip-172-31-89-122 ~]$ sudo mkdir /mnt/data
[ec2-user@ip-172-31-89-122 ~]$ sudo mount /dev/xvdf /mnt/data
[ec2-user@ip-172-31-89-122 ~]$ lsblk
NAME    MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
xvda    202:0    0    8G  0 disk
└─xvda1 202:1    0    8G  0 part /
xvdf    202:80   0  100G  0 disk /mnt/data
[ec2-user@ip-172-31-89-122 ~]$
```

13. Unable to Run Certain Commands? How do you resolve this.

- **Check User Permissions:** Ensure that you are running the commands with the appropriate permissions. Some commands require superuser privileges (root) or elevated permissions using **sudo**

- **Command Not Found:** If you receive a "command not found" error, it might indicate that the command is not in the system's PATH or is not installed. Verify that the command is installed on your system or specify the full path to the command.

- **Check Command Syntax:** Ensure that you are using the correct syntax for the command. Refer to the command's documentation or help message (man or --help) for guidance on how to use it.

- **Check for Missing Dependencies:** Some commands rely on external libraries or dependencies. Make sure that these dependencies are installed on your system.

- **Inspect Configuration Files:** Some commands may be disabled or restricted through configuration files, such as **/etc/sudoers.** Review these configuration files to ensure that the command is not restricted.

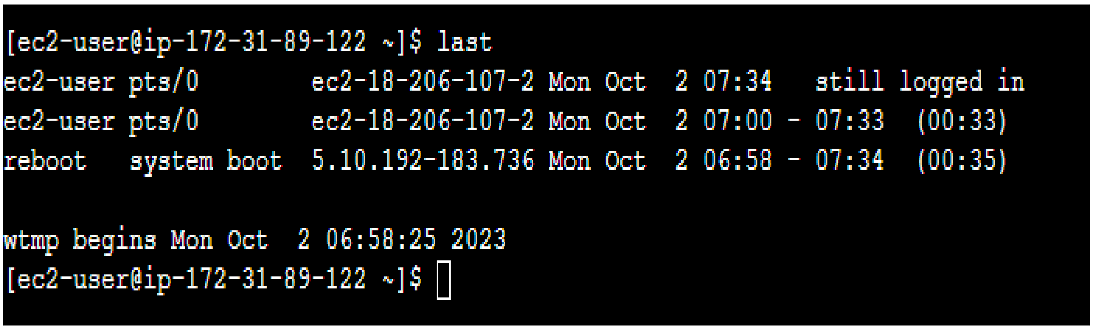
- **Check for Disk or Filesystem Errors:** Disk or filesystem errors can lead to issues with command execution. Run filesystem checks (e.g., fsck) and address any errors found.

- **User and Group Permissions:** Verify that the user executing the command has the appropriate permissions to access the required files and directories.

- **Consult Logs:** Check system logs (e.g., /var/log/syslog, /var/log/messages, or specific application logs) for any error messages or clues related to the commands you are trying to run.

14.)How do you List User Last Login on server and where this information is stored? And also find the last login by dates? And also find the bad login attempts?

Step1:To List Last Logins for All Users: we can use **"last"** command and for specific user we can use **"last username"** and the information is stored in **/var/log/wtmp** or **/var/log/utmp files.**

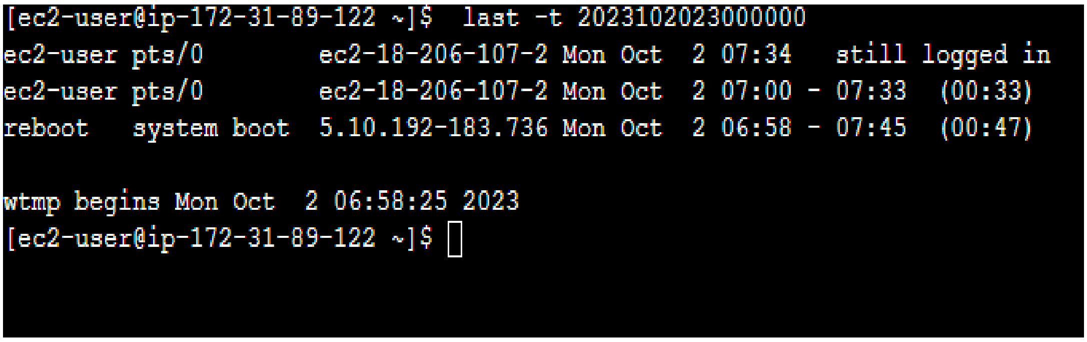


```
[ec2-user@ip-172-31-89-122 ~]$ last
ec2-user pts/0        ec2-18-206-107-2 Mon Oct  2 07:34   still logged in
ec2-user pts/0        ec2-18-206-107-2 Mon Oct  2 07:00 - 07:33  (00:33)
reboot   system boot  5.10.192-183.736 Mon Oct  2 06:58 - 07:34  (00:35)

wtmp begins Mon Oct  2 06:58:25 2023
[ec2-user@ip-172-31-89-122 ~]$ 
```

Step2: Filter Login Records by Dates: we can filter login records by specifying a date range using the **-s (since)** and **-t (until)** options with the last command.

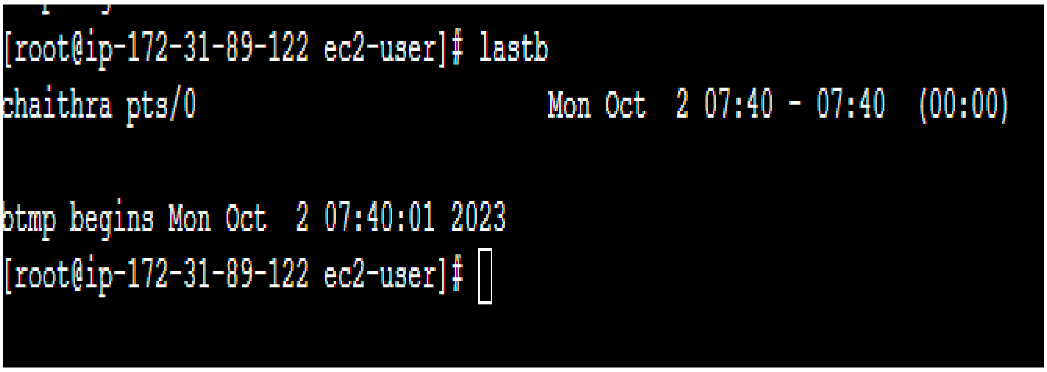
last -t YYYYMMDDHHMMSS



```
[ec2-user@ip-172-31-89-122 ~]$  last -t 20231020230000000
ec2-user pts/0        ec2-18-206-107-2 Mon Oct  2 07:34   still logged in
ec2-user pts/0        ec2-18-206-107-2 Mon Oct  2 07:00 - 07:33  (00:33)
reboot   system boot  5.10.192-183.736 Mon Oct  2 06:58 - 07:45  (00:47)

wtmp begins Mon Oct  2 06:58:25 2023
[ec2-user@ip-172-31-89-122 ~]$ 
```

Step3: Find bad login attemps: you can simply run the lastb command without any additional arguments. This command will display a list of failed login attempts from the /var/log/btmp file.



```
[root@ip-172-31-89-122 ec2-user]# lastb
chaithra pts/0                    Mon Oct  2 07:40 - 07:40  (00:00)

btmp begins Mon Oct  2 07:40:01 2023
[root@ip-172-31-89-122 ec2-user]# 
```

15.) What causes high I/O wait time? How to diagnose I/O wait time in Linux? How do you identify which process causing high I/O wait time

Causes of High I/O Wait Time:

- **Disk Activity:** One common cause is heavy disk activity, such as reading/writing large files or using swap space excessively.

- **Storage Hardware:** Slow or failing storage devices, including hard drives and SSDs, can lead to high I/O wait times.

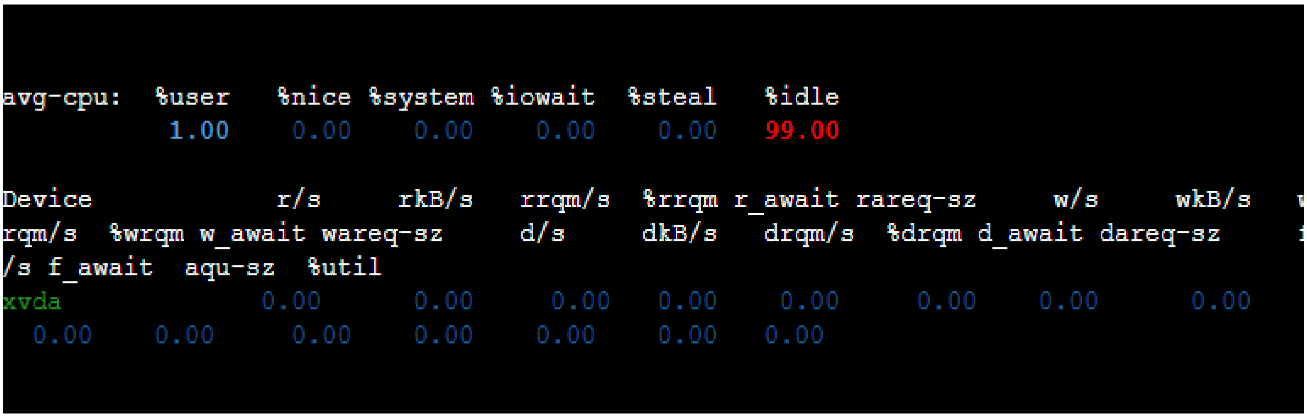
- **Overloaded File Systems:** File systems that are nearing their capacity or experiencing fragmentation can slow down I/O operations.

- **Excessive I/O Requests:** Many concurrent processes making I/O requests canlead to contention and high I/O wait times.

Diagnosing I/O Wait Time:

- **Use top or htop:** Open a terminal and run the top or htop command. Look at the "%wa" (percentage of time the CPU is waiting for I/O) field. If this value is consistently high, it indicates high I/O wait time.

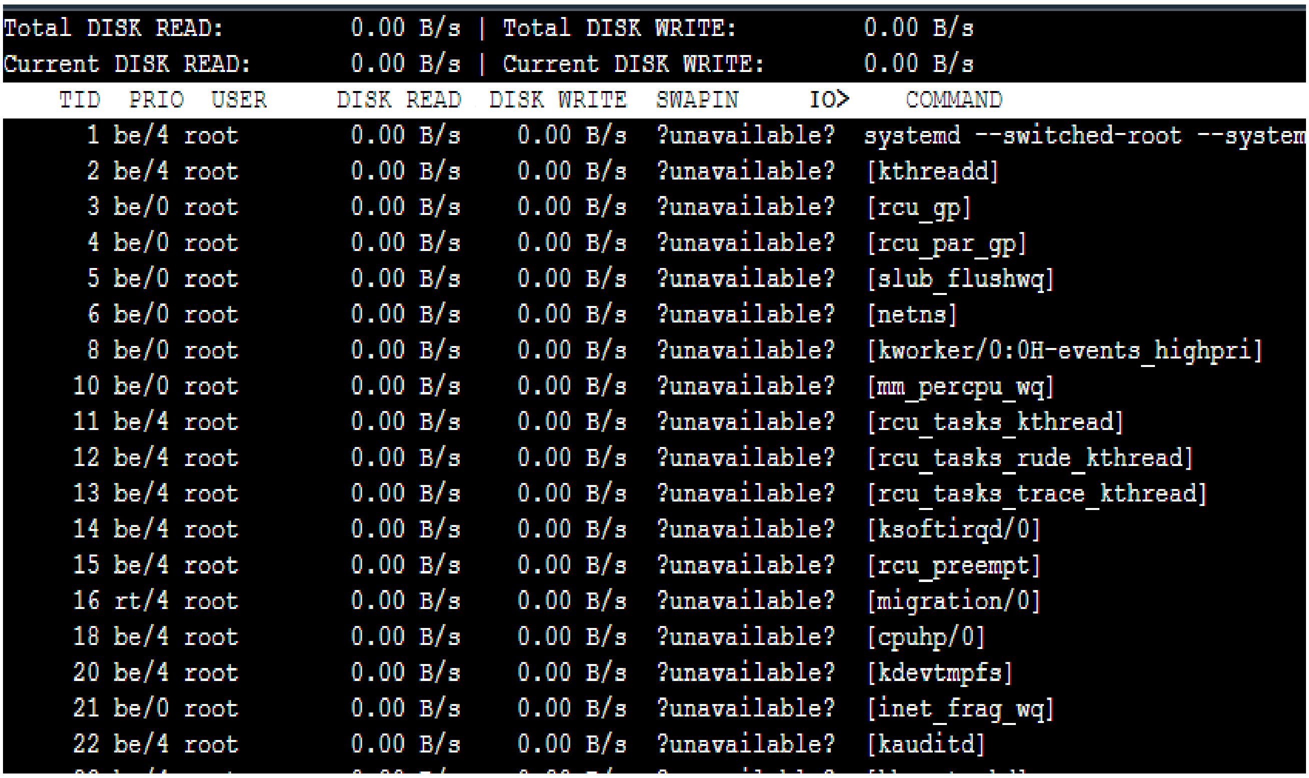
- **Use iostat:** The iostat command provides detailed I/O statistics. Run it with the -x flag to display extended statistics:



```
avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           1.00    0.00    0.00    0.00    0.00   99.00

Device            r/s     rkB/s   rrqm/s  %rrqm r_await rareq-sz     w/s     wkB/s   w
rqm/s  %wrqm w_await wareq-sz     d/s     dkB/s   drqm/s  %drqm d_await dareq-sz     f
/s f_await  aqu-sz  %util
xvda             0.00      0.00     0.00   0.00    0.00     0.00    0.00      0.00
  0.00    0.00    0.00    0.00    0.00    0.00   0.00
```

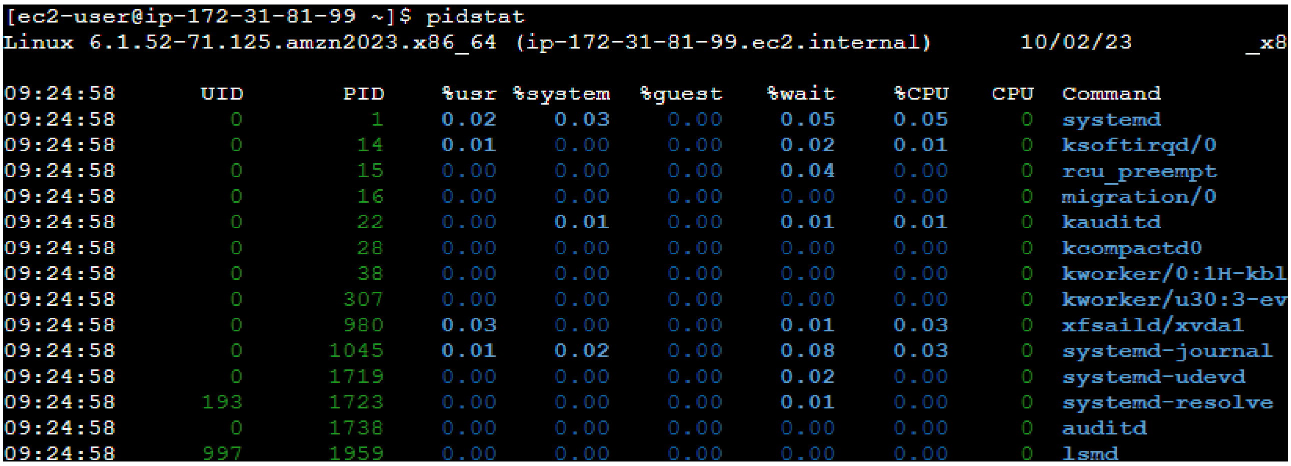
- **Check Disk Activity:** Use tools like **iotop** to identify processes with high I/O usage and terminate unwanted resources



```
Total DISK READ:         0.00 B/s | Total DISK WRITE:         0.00 B/s
Current DISK READ:       0.00 B/s | Current DISK WRITE:       0.00 B/s
    TID  PRIO  USER     DISK READ  DISK WRITE  SWAPIN     IO>    COMMAND
      1 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  systemd --switched-root --system
      2 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [kthreadd]
      3 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [rcu_gp]
      4 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [rcu_par_gp]
      5 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [slub_flushwq]
      6 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [netns]
      8 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [kworker/0:0H-events_highpri]
     10 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [mm_percpu_wq]
     11 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [rcu_tasks_kthread]
     12 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [rcu_tasks_rude_kthread]
     13 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [rcu_tasks_trace_kthread]
     14 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [ksoftirqd/0]
     15 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [rcu_preempt]
     16 rt/4 root        0.00 B/s    0.00 B/s ?unavailable?  [migration/0]
     18 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [cpuhp/0]
     20 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [kdevtmpfs]
     21 be/0 root        0.00 B/s    0.00 B/s ?unavailable?  [inet_frag_wq]
     22 be/4 root        0.00 B/s    0.00 B/s ?unavailable?  [kauditd]
```

Identifying Processes Causing High I/O Wait Time:

- pidstat: we can use the pidstat command to monitor individual process I/O statistics: **pidstat -d 1**



```
[ec2-user@ip-172-31-81-99 ~]$ pidstat
Linux 6.1.52-71.125.amzn2023.x86_64 (ip-172-31-81-99.ec2.internal)      10/02/23        _x8

09:24:58      UID       PID    %usr %system  %guest   %wait    %CPU   CPU  Command
09:24:58        0         1    0.02    0.03    0.00    0.05    0.05     0  systemd
09:24:58        0        14    0.01    0.00    0.00    0.02    0.01     0  ksoftirqd/0
09:24:58        0        15    0.00    0.00    0.00    0.04    0.00     0  rcu_preempt
09:24:58        0        16    0.00    0.00    0.00    0.00    0.00     0  migration/0
09:24:58        0        22    0.00    0.01    0.00    0.01    0.01     0  kauditd
09:24:58        0        28    0.00    0.00    0.00    0.00    0.00     0  kcompactd0
09:24:58        0        38    0.00    0.00    0.00    0.00    0.00     0  kworker/0:1H-kbl
09:24:58        0       307    0.00    0.00    0.00    0.00    0.00     0  kworker/u30:3-ev
09:24:58        0       980    0.03    0.00    0.00    0.01    0.03     0  xfsaild/xvda1
09:24:58        0      1045    0.01    0.02    0.00    0.08    0.03     0  systemd-journal
09:24:58        0      1719    0.00    0.00    0.00    0.02    0.00     0  systemd-udevd
09:24:58      193      1723    0.00    0.00    0.00    0.01    0.00     0  systemd-resolve
09:24:58        0      1738    0.00    0.00    0.00    0.00    0.00     0  auditd
09:24:58      997      1959    0.00    0.00    0.00    0.00    0.00     0  lsmd
```

Check System Logs: Review system logs, such as **/var/log/messages,** for any error messages related to I/O issues, disk errors, or storage problems.