Assignment-1: Linux

Name: K M Chandrashekhar
Batch: DevOps Batch – 1
Date: May 2025

Q1. Install and configure Tomcat version 9 on your Linux machine with the Manager and Host Manager sections enabled

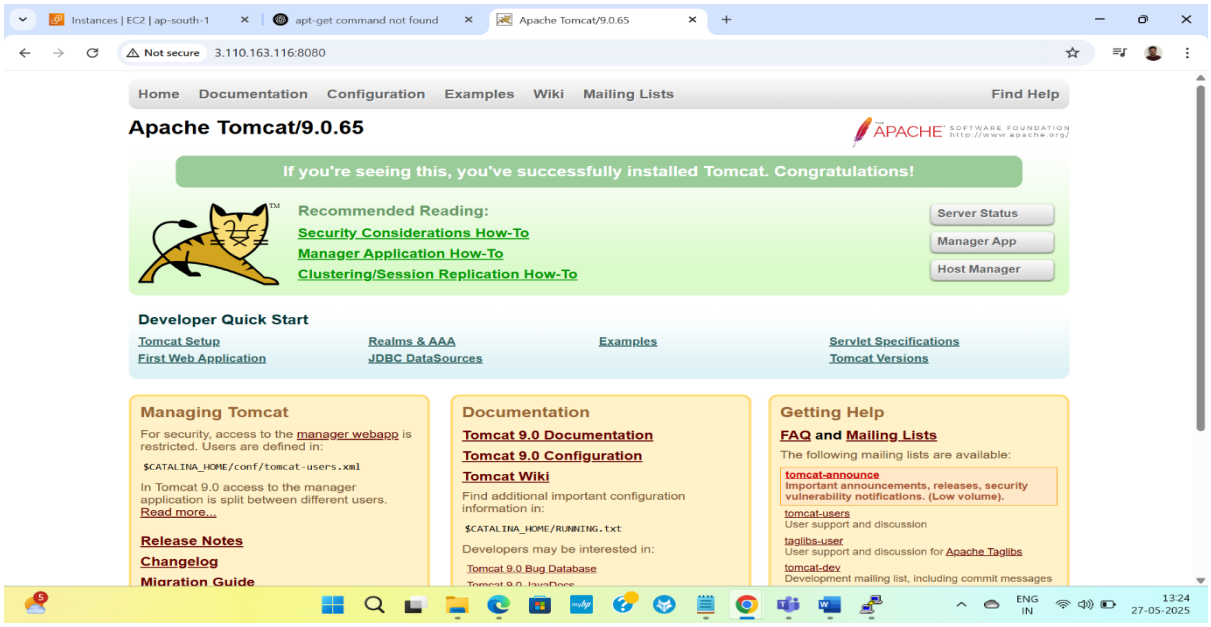
Ans: Apache Tomcat is an open-source web server and servlet container used for hosting Java-based web applications. It's a widely used tool for deploying and running applications built with Java technologies. Tomcat is not a normal server like Apache or Nginx, because its main goal is to provide a good web environment to run Java applications only unlike other normal web servers.

Prerequisites to install Tomcat

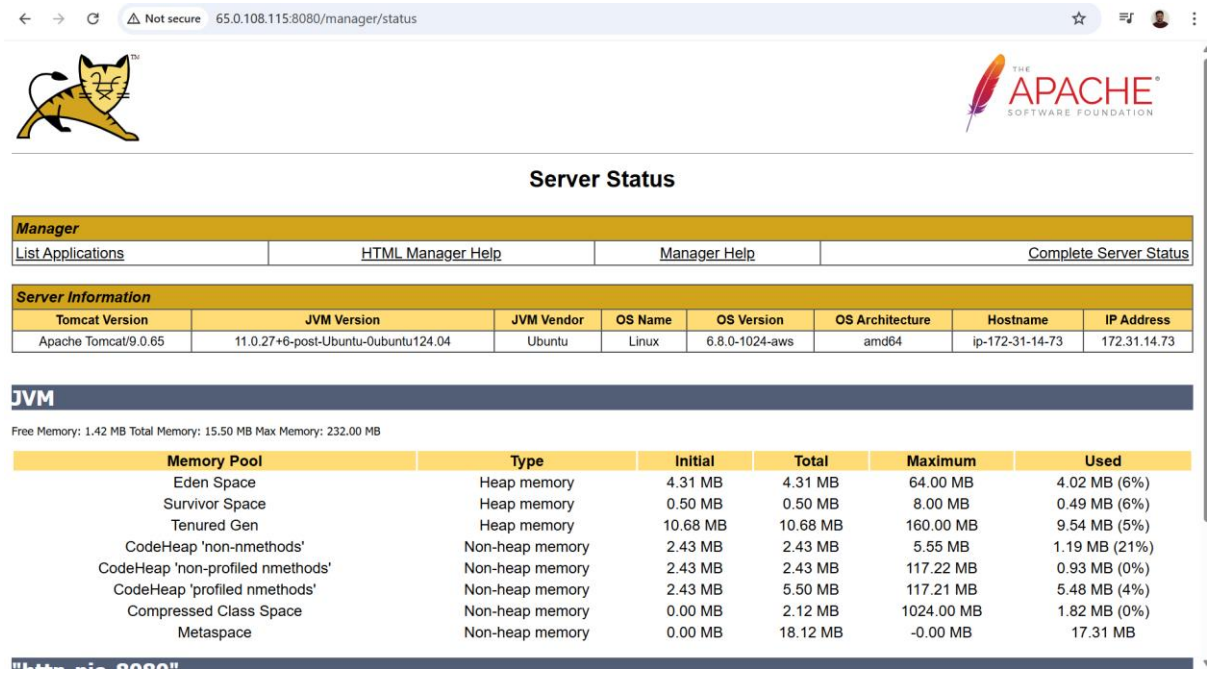
Steps:

1. **Launch a Ubuntu Linux server in AWS.**
2. **Sudo apt-get update**
3. **Sudo apt-get install openjdk -11-jdk**
4. **Next need to go to tomcat archive website & copy tomcat version in binary**
   eg: apache-tomcat-9.0.65.tar.gz
5. **Need check java version: java-version**
6. **Cd /opt**
7. **We need to copy the tomcat tar file:**
   wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.65/bin/apache-tomcat-9.0.65.tar.gz
8. **Next need to extract the tar file : sudo tar -xvf apache-tomcat-9.0.65.tar.gz**
9. **Then need to remove the tar file: sudo rm -rf apache-tomcat-9.0.65.tar.gz**
10. **Need switch as root: sudo su**
11. **Need to go conf directory: cd /opt/apache-tomcat-9.0.65/conf**
12. **Then need to configure the user**
13. **Sudo vi tomcat-users.xml**
    --add below line at the end (2nd Last Line)
    <user username=”admin” password=“admin1234” roles=“admin-gui, manager-gui”/>
14. **Need to link startup & shutdown files**
    Sudo ln -s /opt/apache-tomcat-9.0.65/bin/startup.sh /usr/bin/startTomcat
    Sudo ln -s /opt/apache-tomcat-9.0.65/bin/shutdown.sh /usr/bin/stopTomcat
15. **Sudo vi /opt/apache-tomcat-9.0.65/webapps/manager/META-INF/context.xml**
    Comment in the line: like <!- - Valve class name
    Allow= “127 - ->

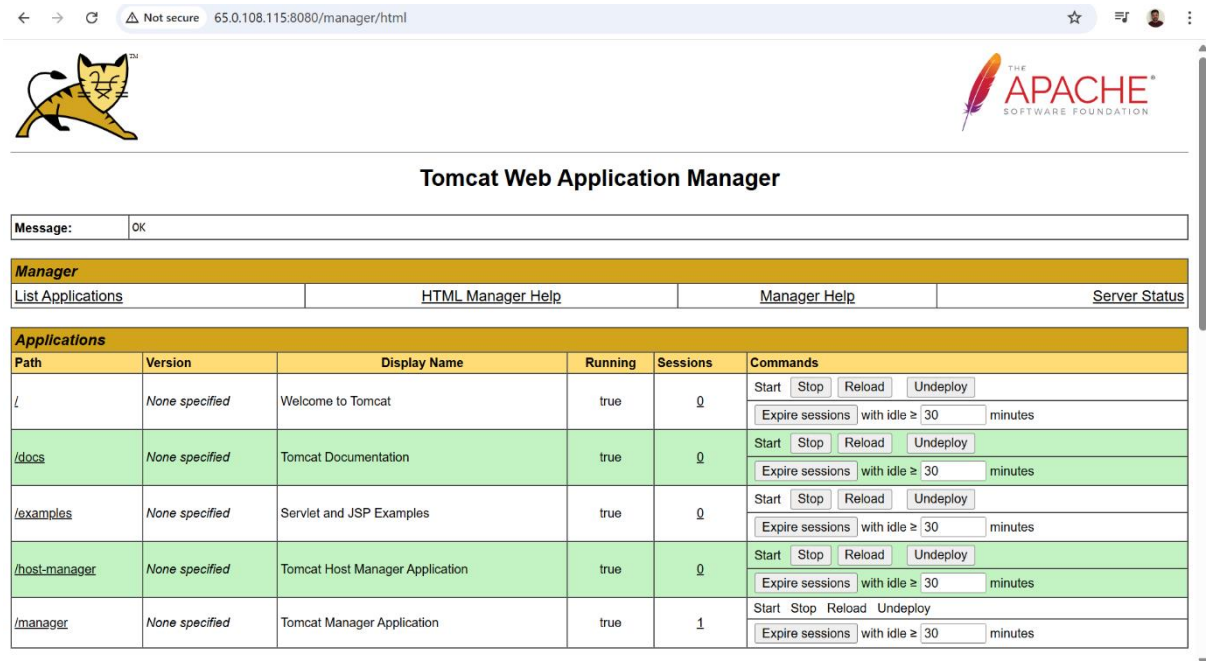
16. **Sudo vi /opt/apache-tomcat-9.0.65/webapps/host-manager/META-INF/context.xml**
    Comment in the line: like <!- - Valve class name
    Allow= “127 - ->
17. **Sudo stopTomcat**
18. **Sudo startTomcat**
19. **Then copy ip address in URL with :8080 tomcat server will open**
20. **Need to define in security groups that allow custom tcp 8080 my Ip**



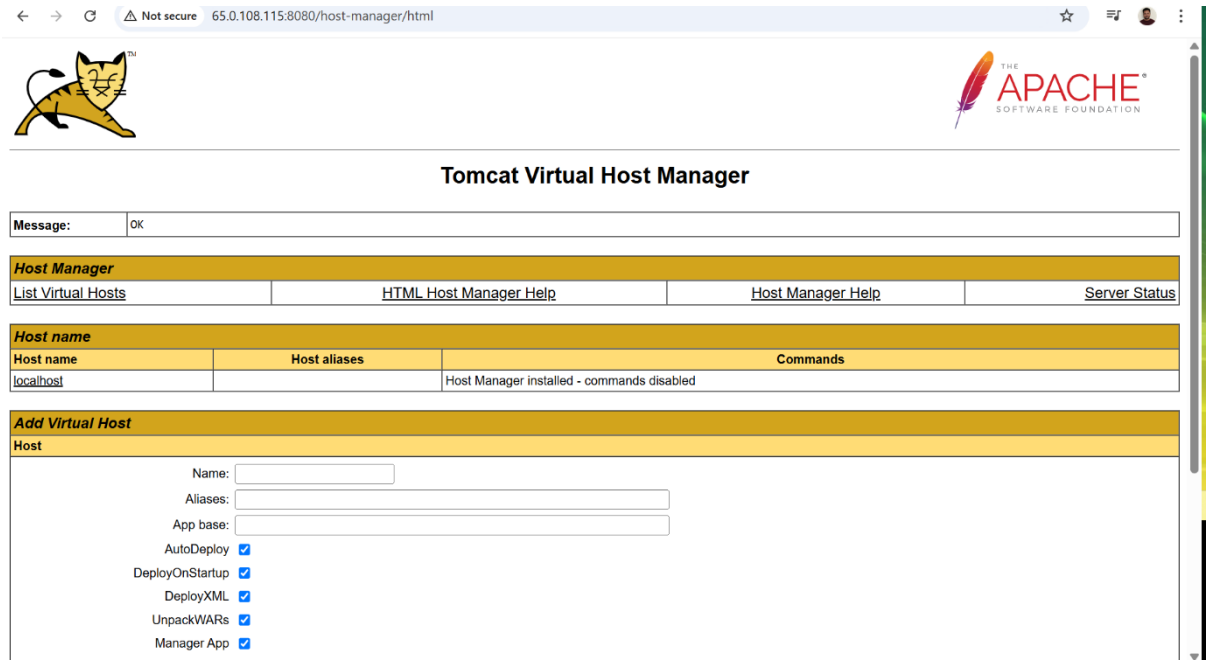
Server Status



Application Manager



Virtual Host Manager

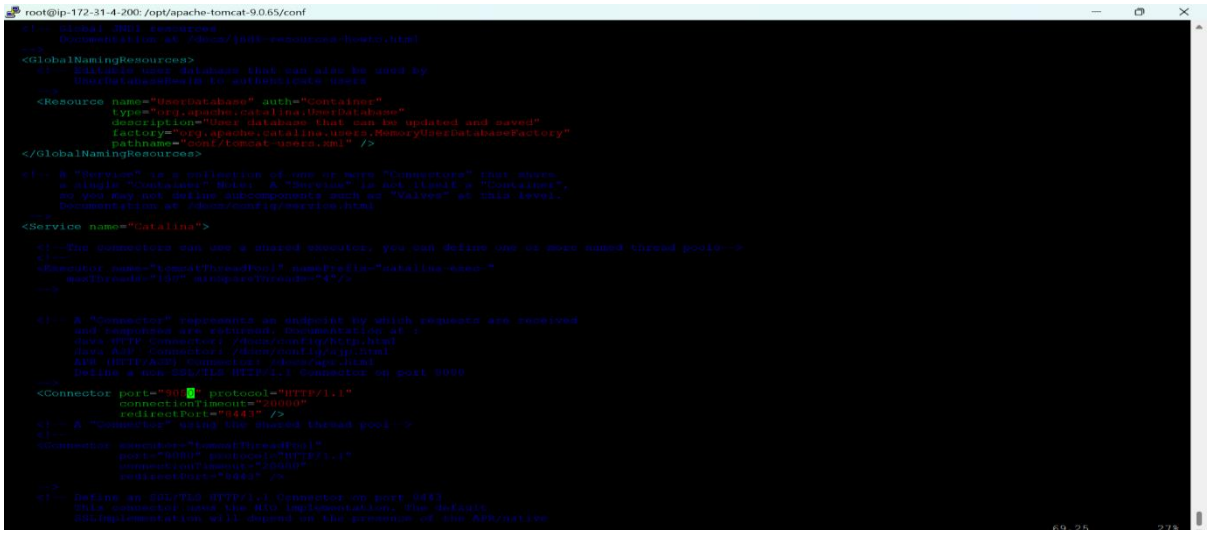


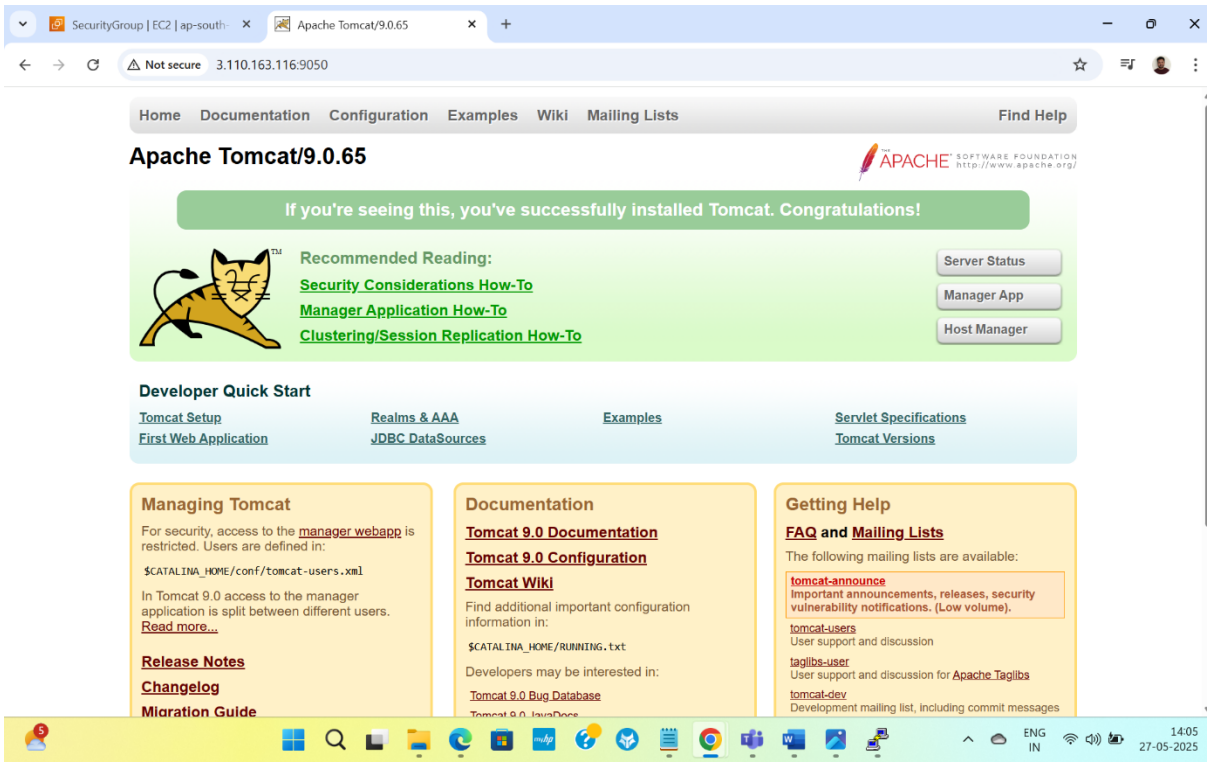
2. Change the default Tomcat port to 9050.

Ans : To change the default port number from 8080 to 9050

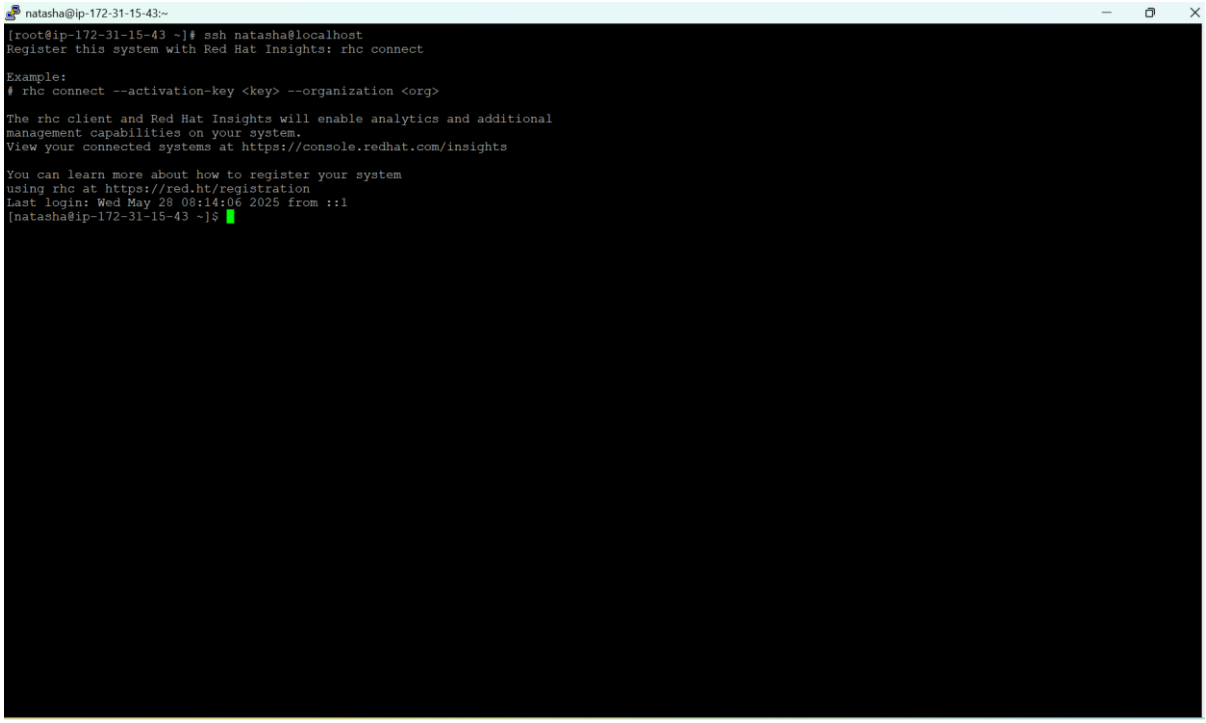
Steps:

1. Should be root user if not then sudo su
2. Then Cd /opt/apache-tomcat-9.0.65/conf
3. Vi server.xml
4. In connecter line need to change the port number from 8080 to 9050
5. Then need to add inbound security rules allow 9050
6. Then need to stop tomcat by using command sudo stopTomcat
7. Then sudo startTomcat
8. If we use ip address in URL with :9050 tomcat server will open



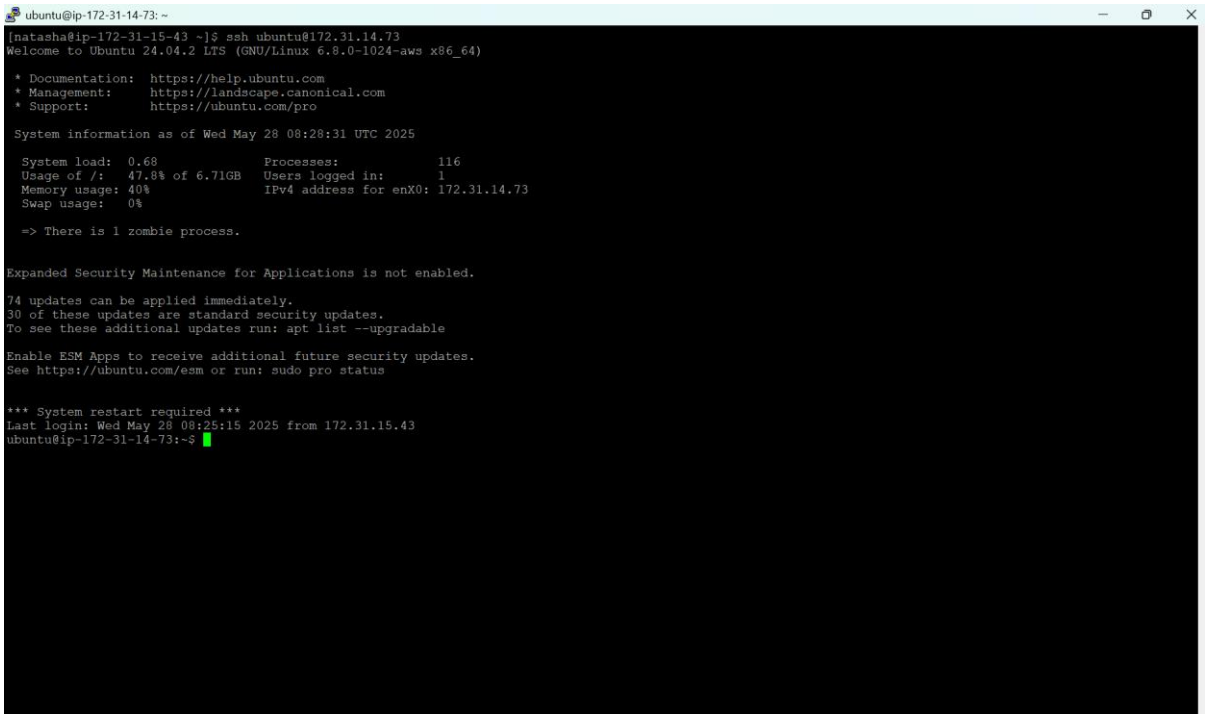


Q3: Create a new user and configure the necessary settings for secure login via Putty or other
remote access tools:
1. Create a new user
2. Add new users under /etc/sudoers
3. Generate SSH key pair (as current user) - ssh-keygen -t rsa
4. This will create: ~/.ssh/id_rsa (private key)
~/.ssh/id_rsa.pub (public key)
5. Switch to the new user
6. Create .ssh directory for the new user:
mkdir -p ~/.ssh
chmod 700 ~/.ssh
7. Create or edit the authorized_keys file
8. From root user cat ~/.ssh/id_rsa.pub | sudo tee -a /home/natasha/.ssh/authorized_keys
9. Fix permission
sudo chown -R natasha:natasha /home/natasha/.ssh
sudo chmod 700 /home/natasha/.ssh
sudo chmod 600 /home/natasha/.ssh/authorized_keys
10. ssh username@localhost



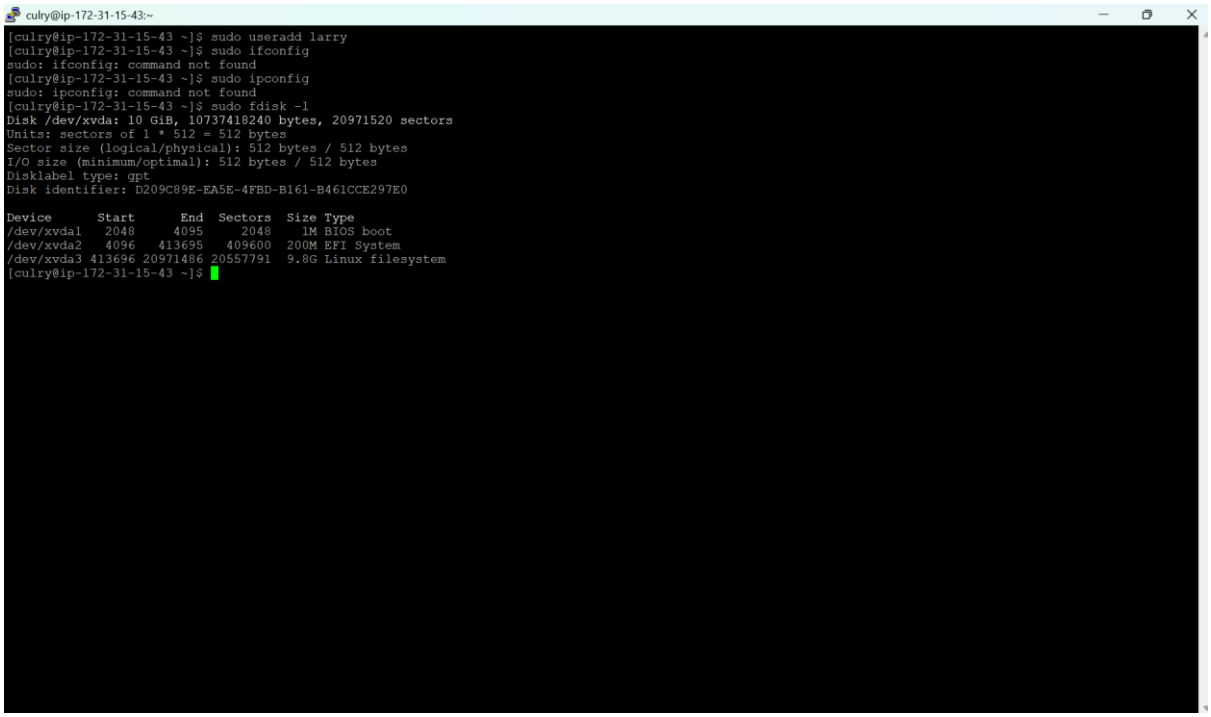
Q4. Establish passwordless connection between two Linux server

1. Server1: ssh-keygen
2. cd .ssh
3. copy the content in id_rsa.pub
4. go to server2: cd .ssh
5. vi authorised keys: append the content copied from server1(do not redirect/delete just have to
add/append)
6. on server1: ssh <server2-username>@private ip of server2
7. successfully established passwordless connection between two servers



Q5. Create a new user account on your Linux machine and grant them sudo privileges.

1. Create user
2. Grant sudo permission under /etc/passwd
3. ALL=(ALL) NOPASSWD:ALL
4. Ex: run sudo useradd ram – sudo command has to work on new user account

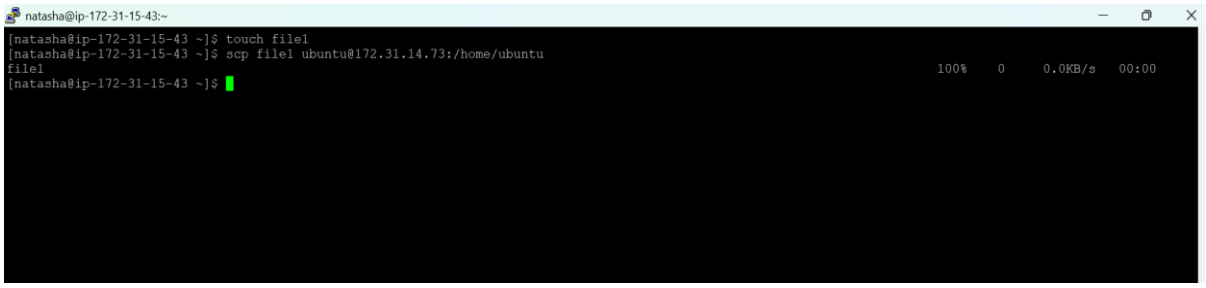


Q6. Use the 'scp' command to securely copy files and directories between Linux machines without password

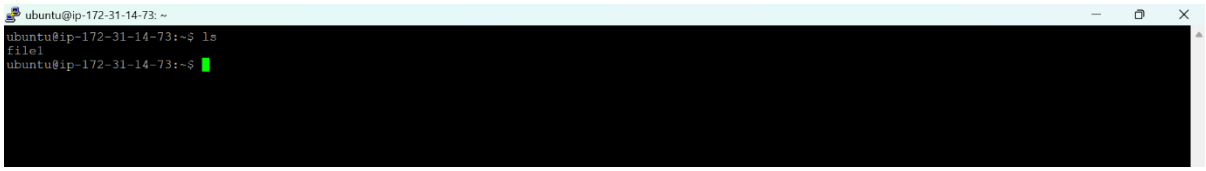
Ans: Continued from q4 question Number

1. Server1:ec2 server2: ubuntu

2. On sever1: scp filename server2-username@private-ip-server2: /home/ubuntu



3. On server 2: ls



7. How can you list the contents of a tar.gz file and extract a particular file

Ans: To list and extract specific files from a tar.gz archive in Linux, use the tar command with appropriate options.

Command to create tar file with gz method

Eg : tar -cvfz etc.tar.gz /etc

Which will compress /etc directory by gz method

```
[root@ip-172-31-15-43 ~]# ls
[root@ip-172-31-15-43 ~]# tar -cvzf etc.tar.gz /etc
tar: Removing leading `/' from member names
/etc/
/etc/grub.d/
/etc/grub.d/00_header
/etc/grub.d/01_users
/etc/grub.d/08_fallback_counting
/etc/grub.d/10_linux
/etc/grub.d/10_reset_boot_success
/etc/grub.d/12_menu_auto_hide
/etc/grub.d/14_menu_show_once
/etc/grub.d/20_linux_xen
/etc/grub.d/20_ppc_terminfo
/etc/grub.d/25_bli
/etc/grub.d/30_os-prober
/etc/grub.d/30_uefi-firmware
/etc/grub.d/40_custom
/etc/grub.d/41_custom
/etc/grub.d/README
/etc/grub.d/35_fwupd
/etc/grub.d/00_tuned
/etc/depmod.d/
/etc/depmod.d/dist.conf
/etc/rhsm/
/etc/rhsm/rhsm.conf.cloud_save
/etc/rhsm/rhsm.conf
/etc/rhsm/ca/
/etc/rhsm/ca/redhat-entitlement-authority.pem
/etc/rhsm/ca/redhat-uep.pem
/etc/rhsm/facts/
/etc/rhsm/pluginconf.d/
/etc/rhsm/syspurpose/
/etc/rhsm/syspurpose/valid_fields.json
/etc/modprobe.d/
/etc/modprobe.d/tuned.conf
/etc/credstore/
/etc/NetworkManager/
/etc/NetworkManager/NetworkManager.conf
/etc/NetworkManager/conf.d/
/etc/NetworkManager/conf.d/30-cloud-init-ip6-addr-gen-mode.conf
```

List Contents of a tar.gz File

Eg: tar -tzvf etc.tar.gz

```
root@ip-172-31-15-43:~
/etc/.updated
/etc/hostname
/etc/resolv.conf
/etc/subuid-
/etc/subgid-
[root@ip-172-31-15-43 ~]# ls
etc.tar.gz
[root@ip-172-31-15-43 ~]#
[root@ip-172-31-15-43 ~]# ls
etc.tar.gz
[root@ip-172-31-15-43 ~]# du -sh /etc
23M     /etc
[root@ip-172-31-15-43 ~]# du -sh etc.tar.gz
5.4M    etc.tar.gz
[root@ip-172-31-15-43 ~]#
[root@ip-172-31-15-43 ~]# tar -tzvf etc.tar.gz
drwxr-xr-x root/root         0 2025-05-28 08:37 etc/
drwx------ root/root         0 2025-04-23 14:16 etc/grub.d/
-rwxr-xr-x root/root      9380 2025-03-25 00:00 etc/grub.d/00_header
-rwxr-xr-x root/root       236 2025-03-25 00:00 etc/grub.d/01_users
-rwxr-xr-x root/root       835 2025-03-25 00:00 etc/grub.d/08_fallback_counting
-rwxr-xr-x root/root     20334 2025-03-25 00:00 etc/grub.d/10_linux
-rwxr-xr-x root/root       833 2025-03-25 00:00 etc/grub.d/10_reset_boot_success
-rwxr-xr-x root/root       892 2025-03-25 00:00 etc/grub.d/12_menu_auto_hide
-rwxr-xr-x root/root       410 2025-03-25 00:00 etc/grub.d/14_menu_show_once
-rwxr-xr-x root/root     14627 2025-03-25 00:00 etc/grub.d/20_linux_xen
-rwxr-xr-x root/root      2562 2025-03-25 00:00 etc/grub.d/20_ppc_terminfo
-rwxr-xr-x root/root       869 2025-03-25 00:00 etc/grub.d/25_bli
-rwxr-xr-x root/root     11006 2025-03-25 00:00 etc/grub.d/30_os-prober
-rwxr-xr-x root/root      1166 2025-03-25 00:00 etc/grub.d/30_uefi-firmware
-rwxr-xr-x root/root       218 2025-03-25 00:00 etc/grub.d/40_custom
-rwxr-xr-x root/root       219 2025-03-25 00:00 etc/grub.d/41_custom
-rw-r--r-- root/root       483 2025-03-25 00:00 etc/grub.d/README
-rwxr-xr-x root/root       725 2024-12-16 00:00 etc/grub.d/35_fwupd
-rwxr-xr-x root/root      1100 2025-02-03 19:17 etc/grub.d/00_tuned
drwxr-xr-x root/root         0 2025-04-23 14:16 etc/depmod.d/
-rw-r--r-- root/root       116 2025-01-14 00:00 etc/depmod.d/dist.conf
drwxr-xr-x root/root         0 2025-04-23 14:16 etc/rhsm/
-rw-r--r-- root/root      3151 2025-04-23 14:16 etc/rhsm/rhsm.conf.cloud_save
-rw-r--r-- root/root      3049 2025-04-23 14:16 etc/rhsm/rhsm.conf
drwxr-xr-x root/root         0 2025-04-23 14:15 etc/rhsm/ca/
-rw-r--r-- root/root      2305 2022-06-23 10:36 etc/rhsm/ca/redhat-entitlement-authority.pem
-rw-r--r-- root/root      7411 2022-06-23 10:36 etc/rhsm/ca/redhat-uep.pem
drwxr-xr-x root/root         0 2025-03-06 00:00 etc/rhsm/facts/
drwxr-xr-x root/root         0 2025-03-06 00:00 etc/rhsm/pluginconf.d/
drwxr-xr-x root/root         0 2025-04-23 14:16 etc/rhsm/syspurpose/
-rw-r--r-- root/root       347 2025-03-06 00:00 etc/rhsm/syspurpose/valid_fields.json
drwxr-xr-x root/root         0 2025-04-23 14:16 etc/modprobe.d/
-rw-r--r-- root/root       674 2025-02-03 19:17 etc/modprobe.d/tuned.conf
```

Explanation:

- t: list contents
- z: handle gzip compression
- v: verbose (show file names)
- f: archive file to use
-

Extract a Specific File from .tar.gz

Eg: tar -xvf etc.tar.gz

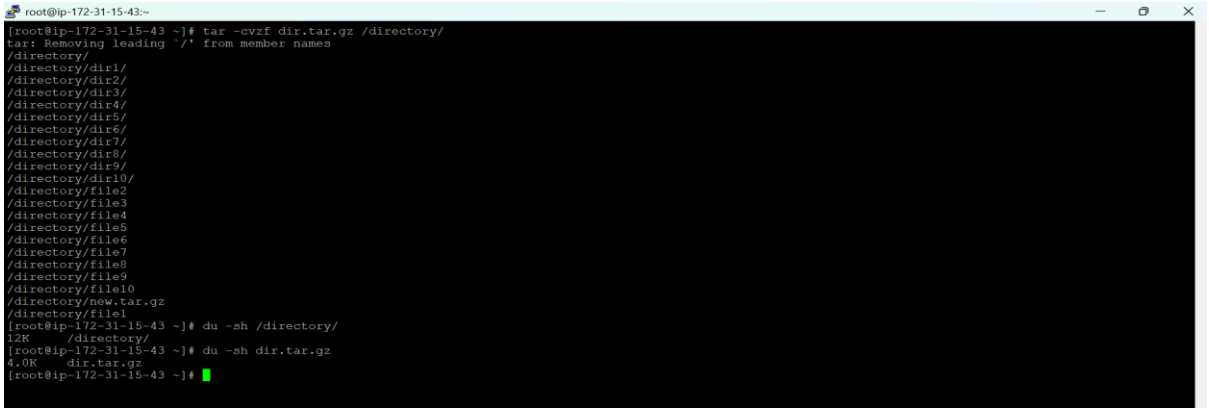
It will extract to the current directory

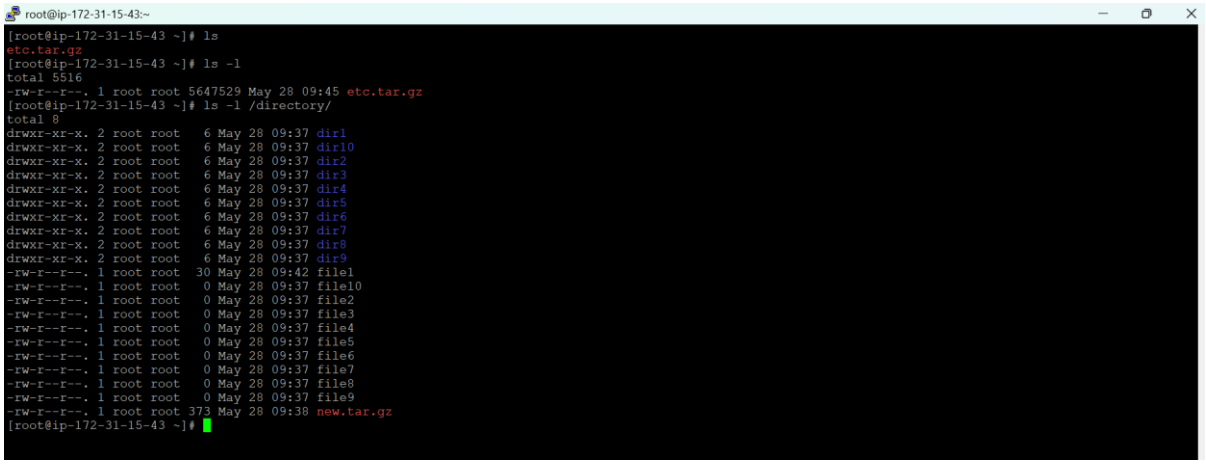
8. How can you compress a directory or file into a tar archive on Linux

Ans: To compress a directory or file into a .tar archive on Linux, you use the tar command.

Command : tar -czvf archive_name.tar.gz /path/to/directory_or_file

Eg tar -cvzf new.tar.gz /directory





Option	Meaning
c	**Create** a new archive
z	**Compress** using gzip
v	**Verbose** output (shows progress)
f	**Filename** of the archive

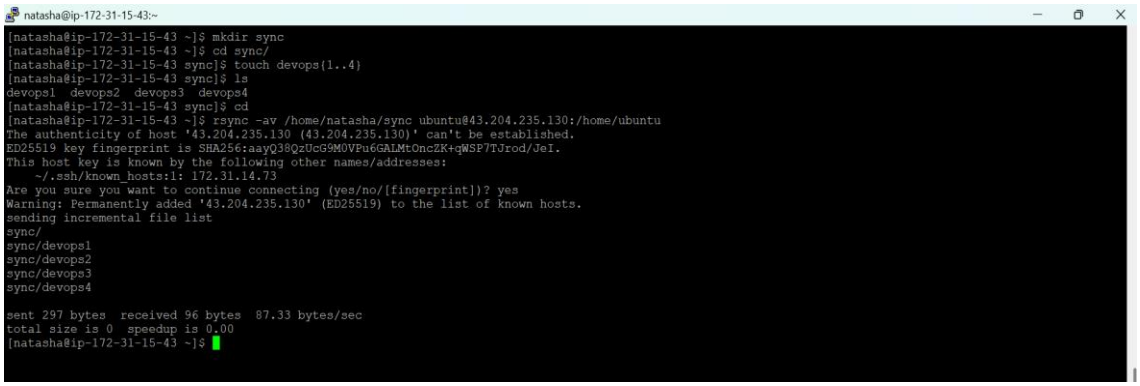
Other Compression options:

Format	Command Example	Compression Tool
.tar.gz	tar -czvf archive.tar.gz folder/	gzip
.tar.bz2	tar -cjvf archive.tar.bz2 folder/	bzip2
.tar.xz	tar -cJvf archive.tar.xz folder/	xz

9. Write a command to synchronise files and directories between 2 local folders

Ans: To synchronize files and directories between two **local folders** in Linux, use the rsync command.
Command : rsync -av /source/directory/ /destination/directory/
rsync: the tool for syncing files
• -a: archive mode (preserves permissions, timestamps, etc.)
• -v: verbose (shows progress)
This will:

- Copy all files and folders **from source directory/ to destination directory/**
- Only copy **new or changed files** (efficient sync)



10. What is load average and swap memory in Linux

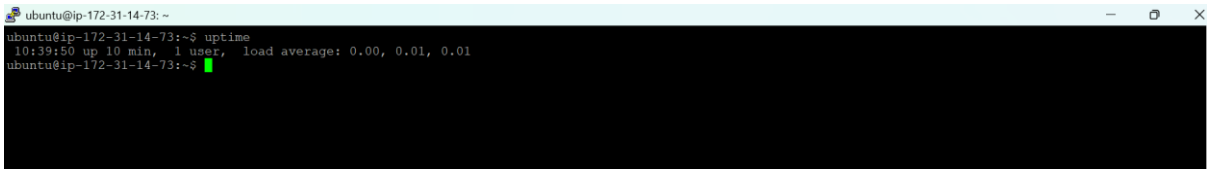
Ans: **Load average** shows the average number of processes waiting to be run on the CPU(s) over time.
Commands : top, uptime, or cat /proc/loadavg.
Eg: load average: 0.45, 0.60, 0.70

This shows load over the last:

- 1 minute, 5 minutes, 15 minutes

How to Interpret It:

- If you have 1 CPU core:
  - Load 1.00 means 100% usage (fully utilized).
  - Load >1.00 means CPU is overloaded.

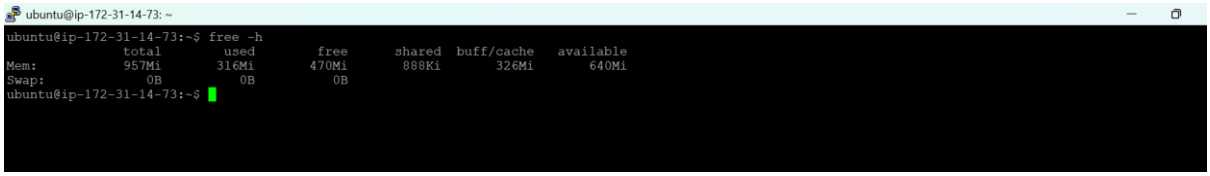


Swap is disk space used as virtual memory when physical RAM is full.

- It helps avoid out-of-memory crashes.
- It is much slower than RAM.

Command : free -h, top, or vmstat.
Eg: free -h

	total	used	free	shared	buff/cache	available
Mem:	8.0G	7.5G	0.2G	0.1G	0.3G	0.4G
Swap:	2.0G	1.0G	1.0G			



If swap is in use, it usually means:

- Your system ran out of RAM.
- Processes are being pushed to slower disk-based memory.

Excessive swap usage = performance drop.

11. What are the important system directories in linux and what are they genereally used for [Ex: var, home etc]

Ans: Important Linux System Directories

Directory	Purpose
`/`	**Root directory** – Top-level of the filesystem hierarchy. Everything starts here.
`/bin`	Essential **user binaries** (e.g., `ls`, `cp`, `rm`) needed for basic system use.
`/sbin`	Essential **system binaries** for system administration (e.g., `reboot`, `ifconfig`).
`/etc`	**Configuration files** for the system and services (e.g., `/etc/ssh/ssh_config`).
`/home`	**User home directories** (e.g., `/home/alice`, `/home/bob`). User data lives here.
`/root`	**Home directory of the root user**. Different from `/home`.
`/var`	Variable data – **logs**, **databases**, **mail**, **print spool**, etc.
`/usr`	User-related programs and data; contains subdirs like `/usr/bin`, `/usr/lib`.
`/lib`	Essential **shared libraries** needed by binaries in `/bin` and `/sbin`.

/tmp	Temporary files – cleared on reboot or periodically. Accessible by all users.
/opt	Optional software or **third-party applications** (e.g., `/opt/google/chrome`).
/dev	Represents **device files** (e.g., `/dev/sda`, `/dev/tty`). Interfaces to hardware.
/proc	Virtual directory containing **process and kernel information** (e.g., `/proc/cpuinfo`).
/sys	Virtual filesystem for **kernel and hardware status** (related to `/proc`).
/boot	Files needed for **booting the system**, including the kernel and GRUB files.
/media	Mount point for **removable media** (USB drives, CDs, etc.).
/mnt	General **mount point** for temporarily mounting filesystems.
/run	Holds **runtime data** like PID files and sockets (usually cleared on boot).

12. What are the package installers for Redhat, Ubuntu, Debian, CentOS, Alpine

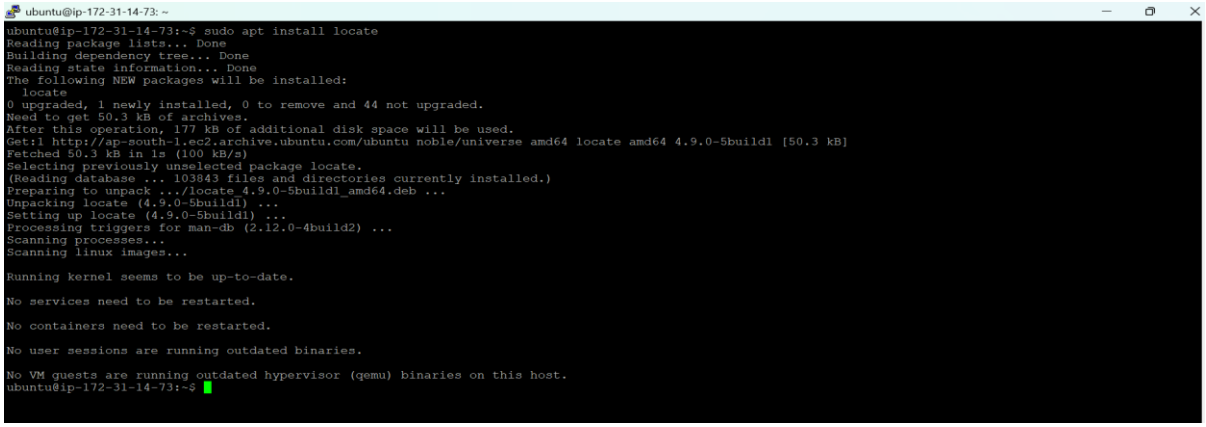
Ans: Redhat (RHEL) : YUM / DNF,

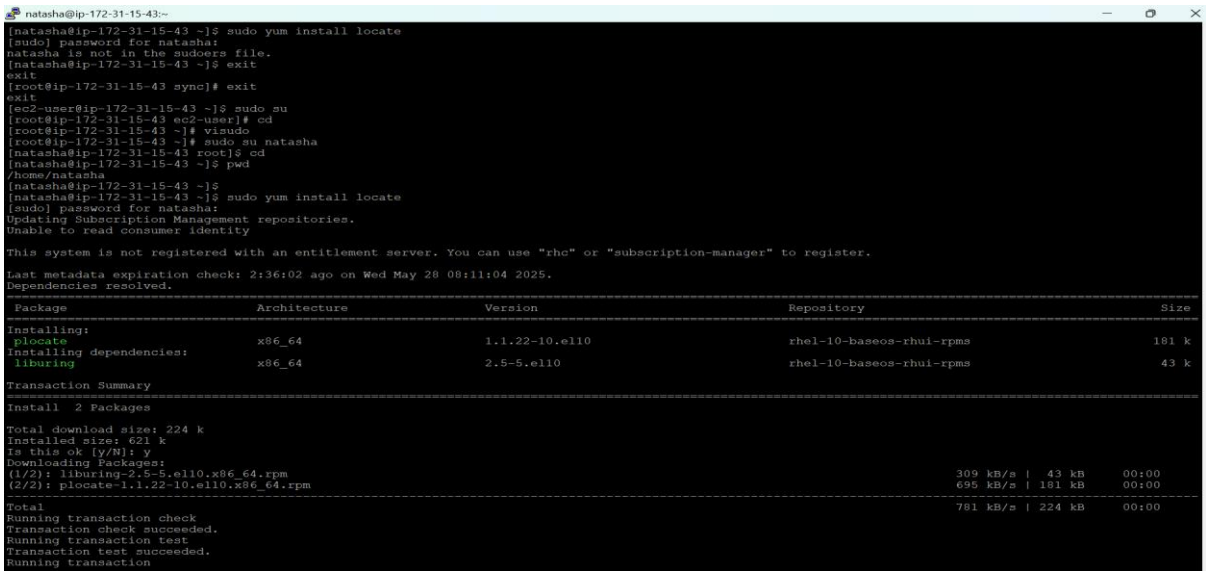
CentOS : YUM/DNF

Ubuntu : APT

Debian : APT

Alpine : APK





Optional Assignment: Configure Gmail on your Linux machine using Postfix/Sendmail to enable sending emails from your Gmail Account through the command line

Ans: RHEL Machine Setup

Step 1: Yum install postfix to check wheather installed the application or not by using command rpm -qa | grep postfix

Step 2: Yum install mailx

Step 3: then need to enter into postfix configuration file i.e., /etc/postfix/main.cf

Cd /etc/postfix

Step 4: need to take a backup if we are changing any configuration file it is better practice if anything happens after changing the configuration file we can roll back

Step 5: vi main.cf Add the following lines in bottom of file

your_hostname (enter your hostname)

step 6: at the bottom need to add the below commands

#Location of sasl_passwd we saved

smtp_sasl_password_maps = hash:/etc/postfix/sasl/sasl_passwd

#Enables SASL authentication for postfix

smtp_sasl_auth_enable = yes

smtp_tls_security_level = encrypt

#Disallow methods that allow anonymous authentication

smtp_sasl_security_options = noanonymous

step 7: mkdir sasl under postfix directory

touch sasl_passwd

step 8: In our mail need to change the settings

enable two factor authentication

in search bar type apps there you will get app password of 16 digit there you can create name: smtp generate

step 9: vi sasl_passwd

[smtp.gmail.com]:587 kmchandrashekhar.iitkgp@gmail.com:password (16 digit password to paste continuously)

Step 10: enter command postmap sasl_passwd under sasl directory

Change the permission for both the files sasl files by using command

Chmod 600 *(* for both the files permission)

Step 11: systemctl start postfix.service

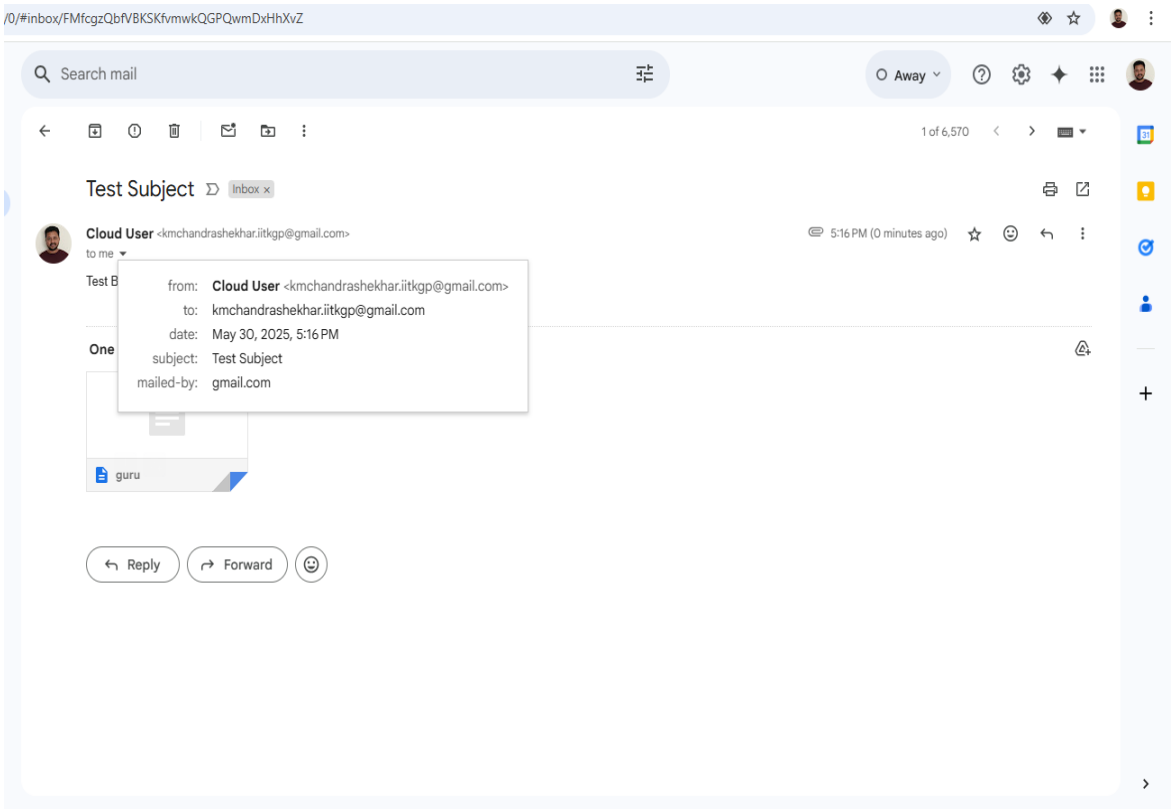
Systemctl status postfix.service

Step 12: echo "Test Mail" | mail -s "Postfix TEST" kmchandrashekhar.iitkgp@gmail.com

Step 13: how to attach a file

echo "Test Mail" | mail -s "Postfix TEST" -a filename kmchandrashekhar.iitkgp@gmail.com

step 14: if mail error then we can check in /var/log/maillog


/0/#inbox/FMfcgzQbfVBKSKfvmwkQGPQwmDxHhXvZ
Search mail
Away
1 of 6,570
Test Subject
Inbox ×
Cloud User <kmchandrashekhar.iitkgp@gmail.com>
to me
5:16 PM (0 minutes ago)
from: Cloud User <kmchandrashekhar.iitkgp@gmail.com>
to: kmchandrashekhar.iitkgp@gmail.com
date: May 30, 2025, 5:16 PM
subject: Test Subject
mailed-by: gmail.com
guru
Reply
Forward