**Week 1 Assignment**

**Filesystem Management**

1. Create a directory named project\_files



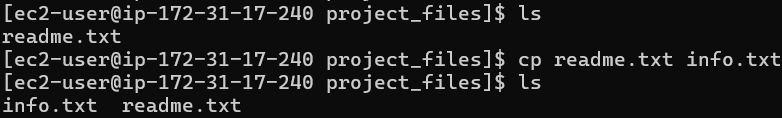
1. Navigate the directory using cd.



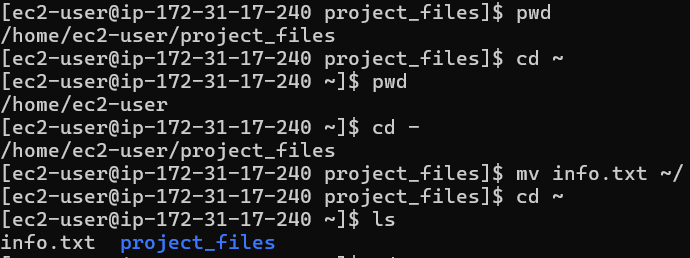
1. Create an empty file named readme.txt



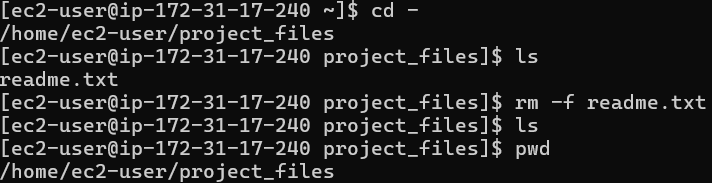
1. Copy readme.txt to a new file info.txt



1. Move info.txt to your home directory



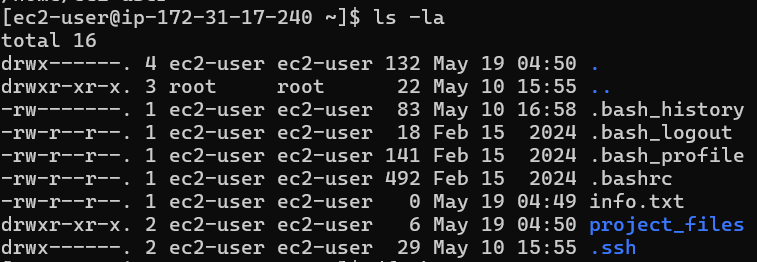
1. Delete readme.txt



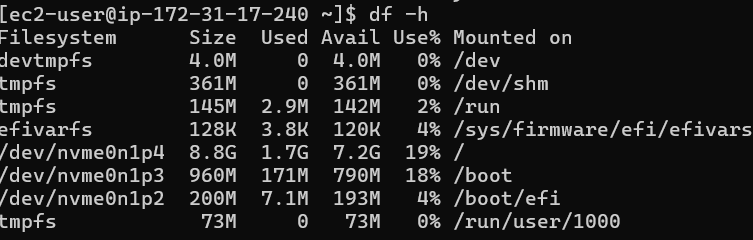
1. Display the current directory path using pwd



1. View the list of files including hidden ones using ls -la



1. Check disk usage with df -h

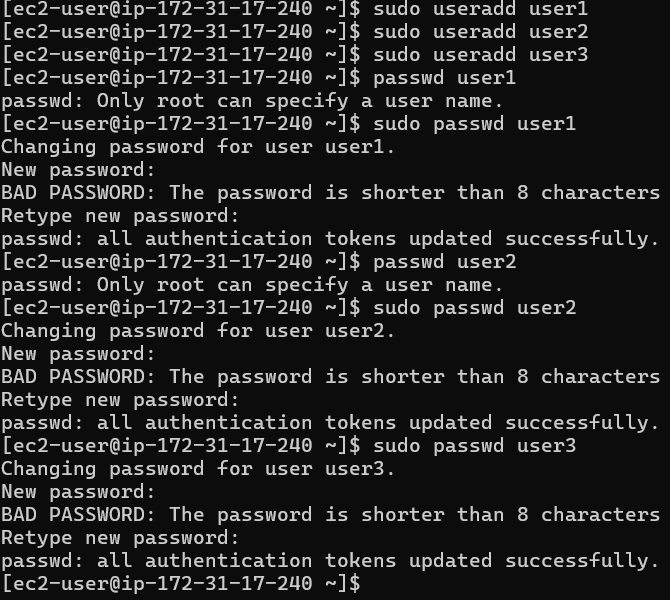


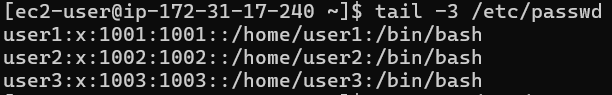
1. Check the file space used by the project\_files directory using du -sh project\_files



**User and group administration**

1. Create users user1,user2 and user3



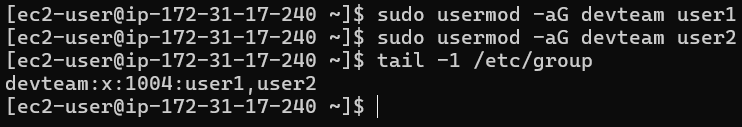


1. Create a group named devteam

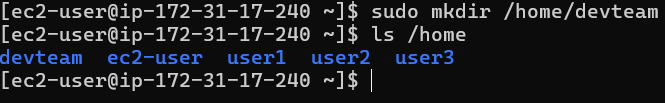




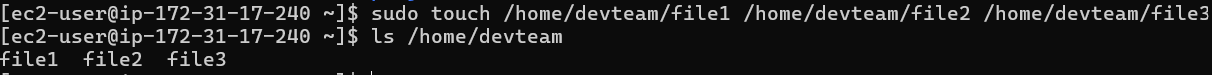
1. Add user1 and user2 to devteam



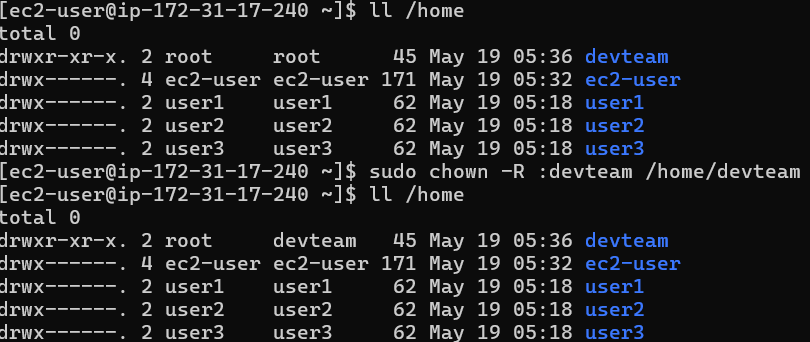
1. Create a folder /home/devteam



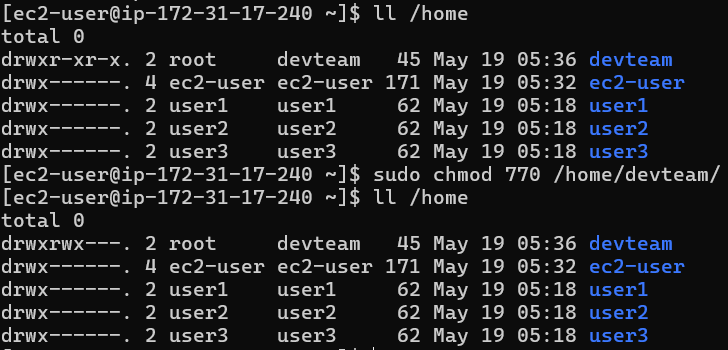
1. Create 2-3 files into this folder



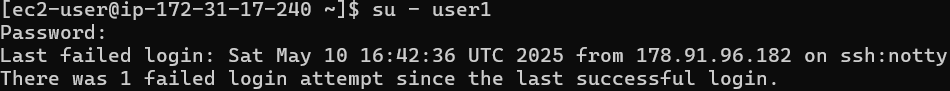
1. Change group ownership of /home/devteam to devteam



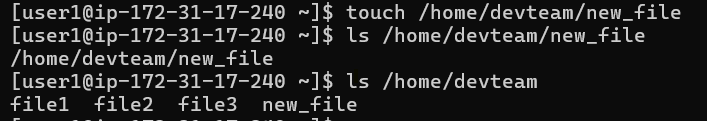
1. Set permissions so only group members can read or write: chmod 770



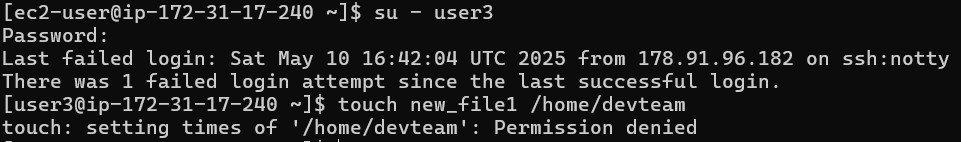
1. Login as user1 and create a file inside the folder



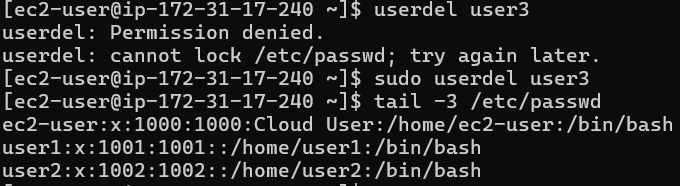




1. Login as user3(not in group) and try accessing the folder (should be denied)

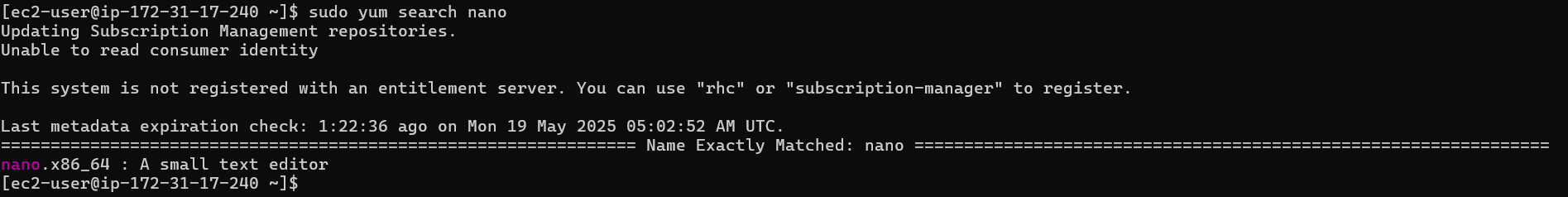


1. Delete the user3



**Package Management with yum**

1. Search for the nano text editor using yum search nano



1. Install nano





1. Remove nano

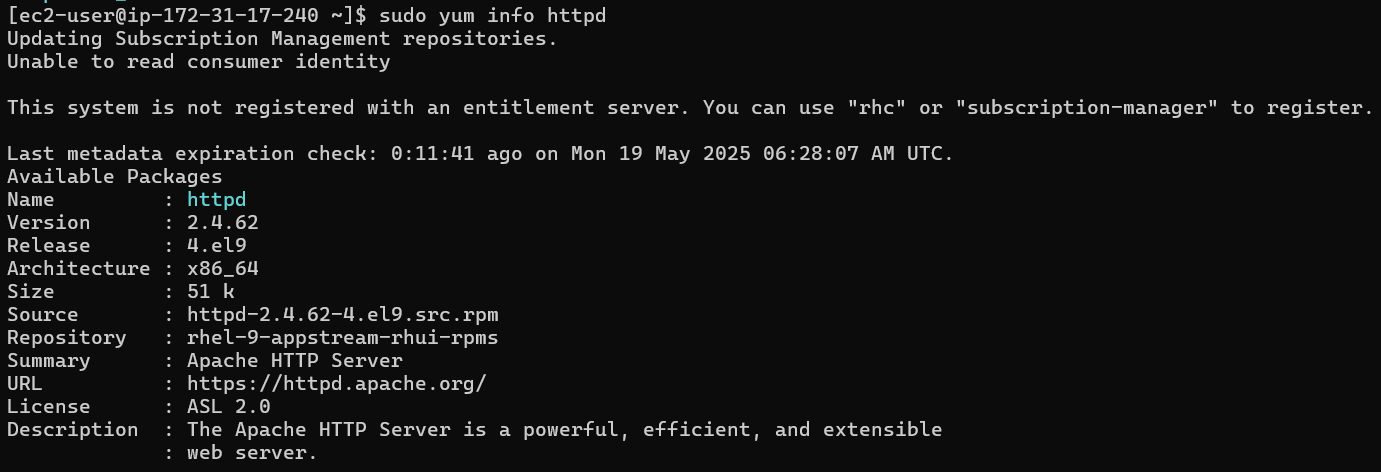




1. List all installed packages

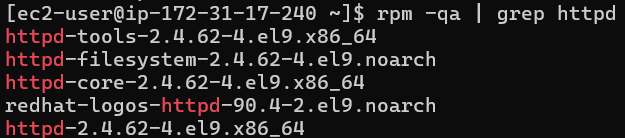


1. Get details about httpd package

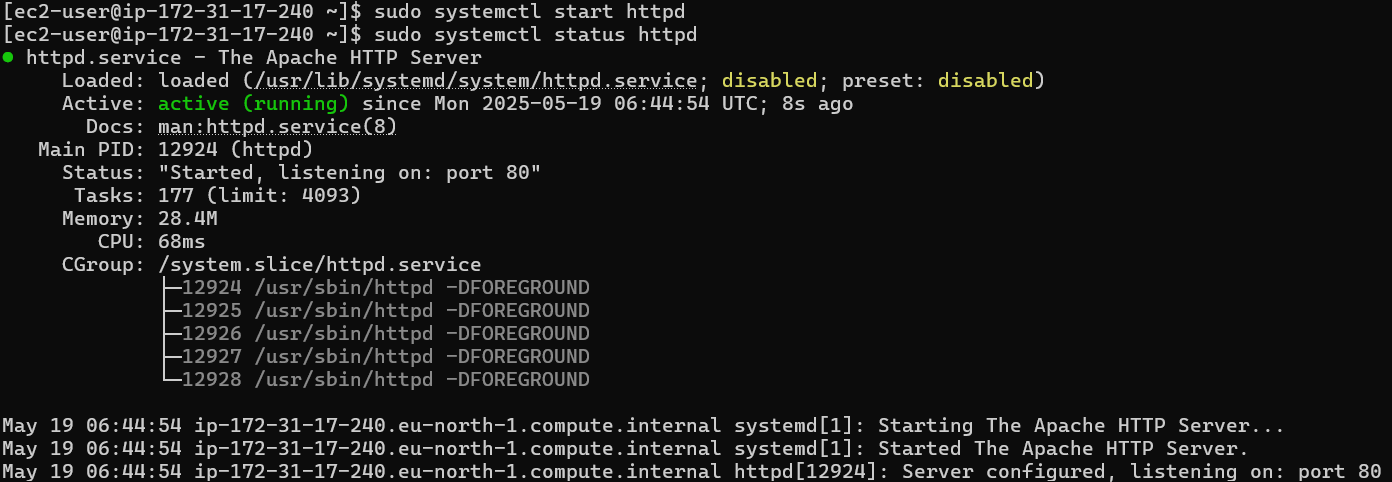


1. Install apache(httpd)

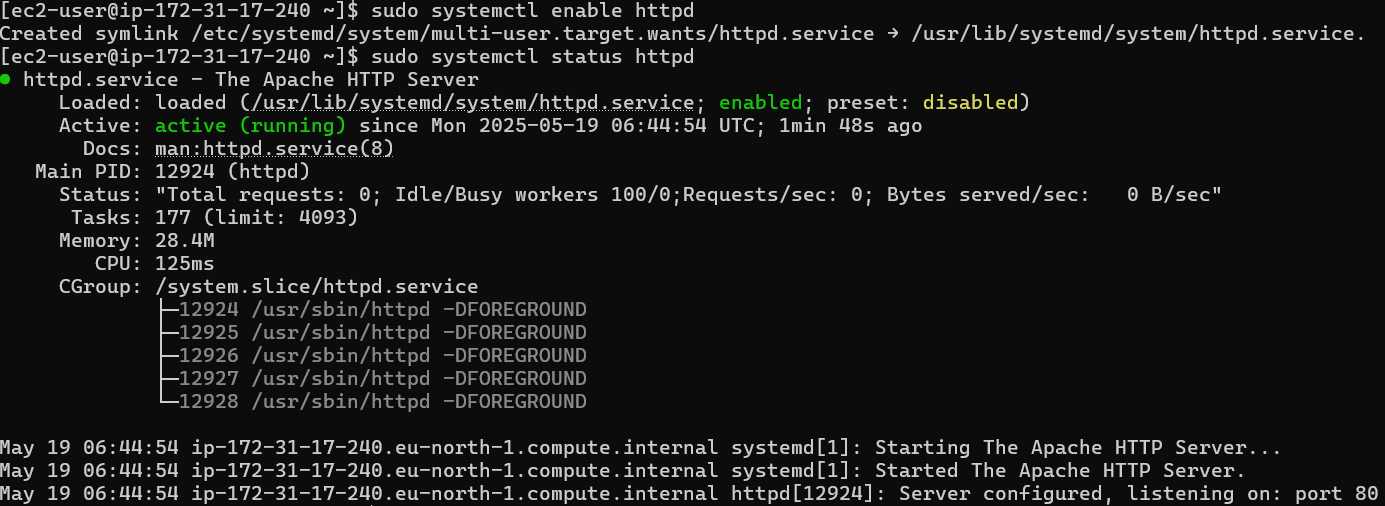




1. Start the apache service: systemctl start httpd

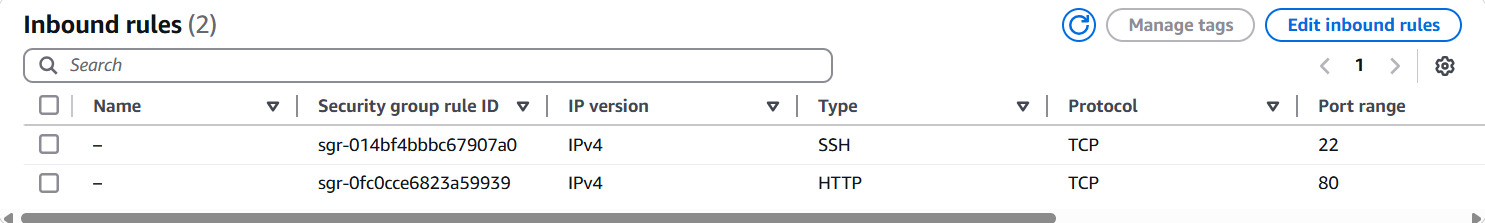


1. Enable it at boot: systemctl enable httpd

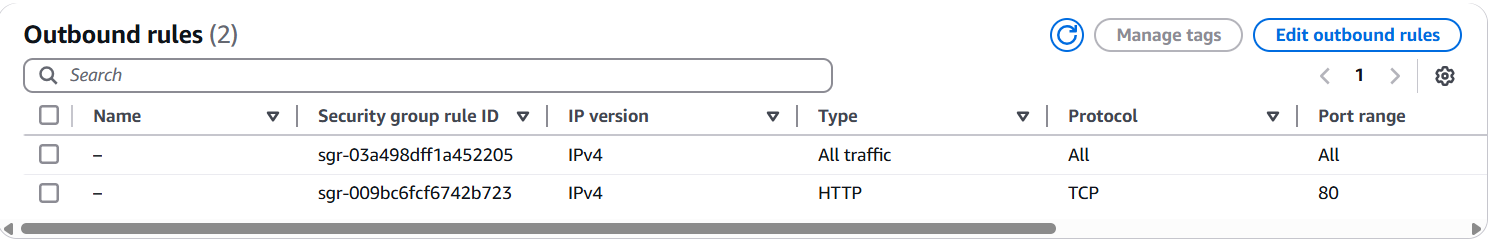


1. Visit http:/localhost to verify

* Allow http in inbound and outbound rules in aws
* Go to EC2🡪Security groups
* Edit inbound rules
* Add rule and allow http
* Save rules

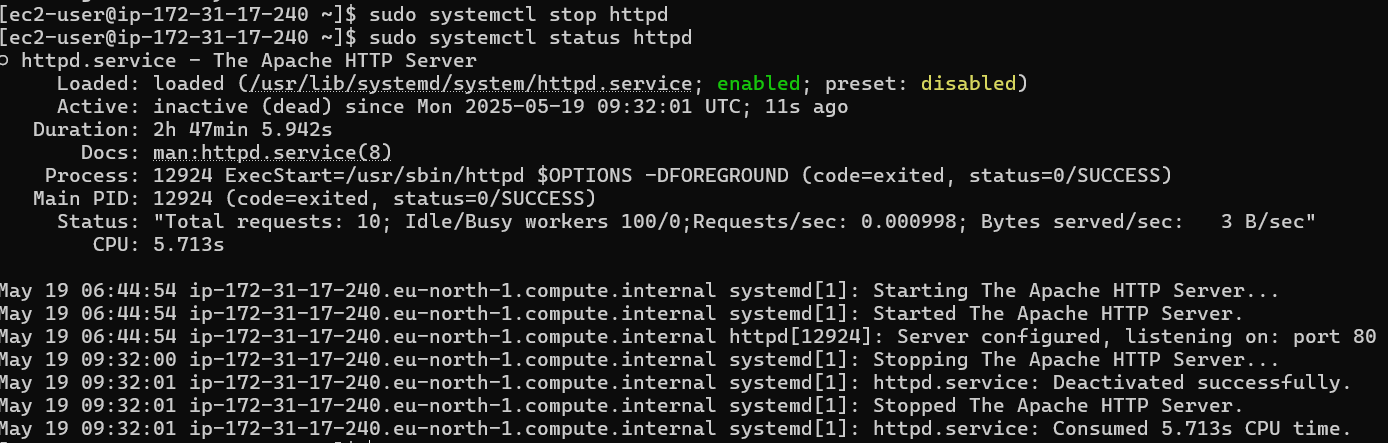


* Edit outbound rules also
* Add rule and allow http
* Save rules



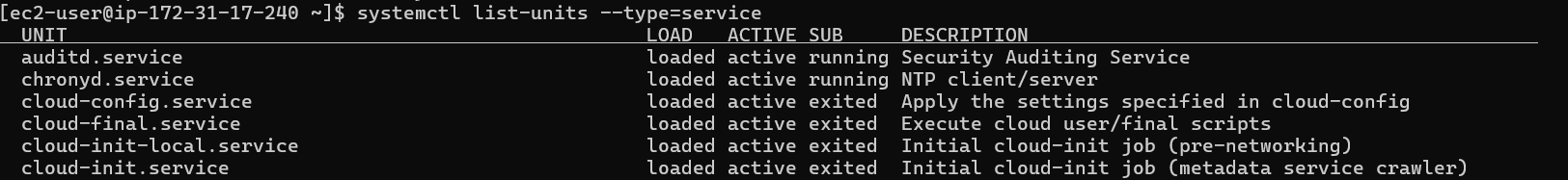


1. Stop httpd service

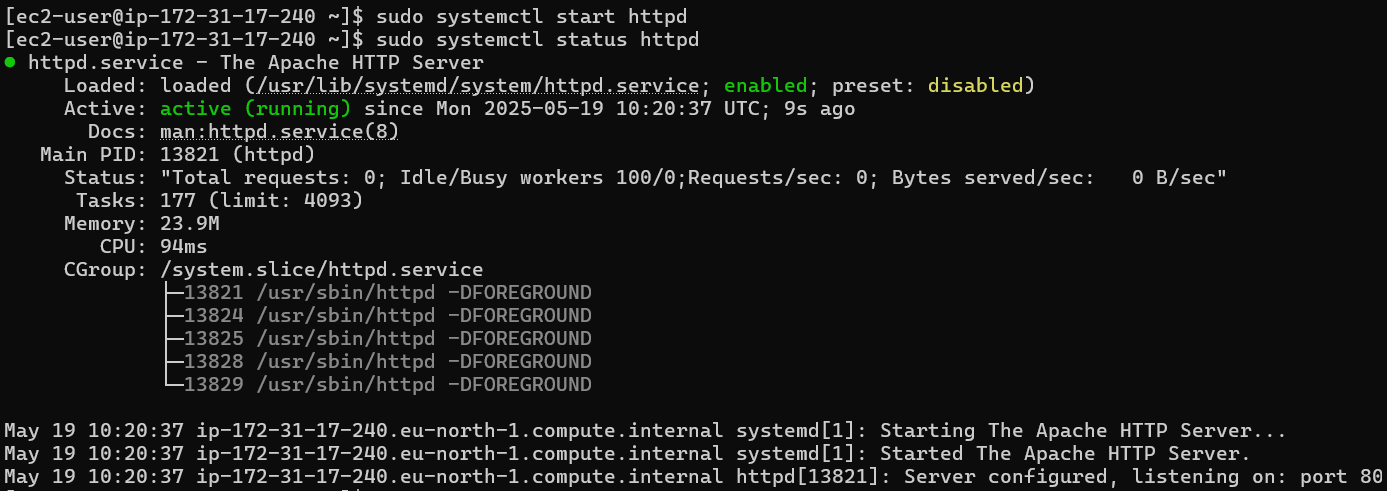


**System services and systems**

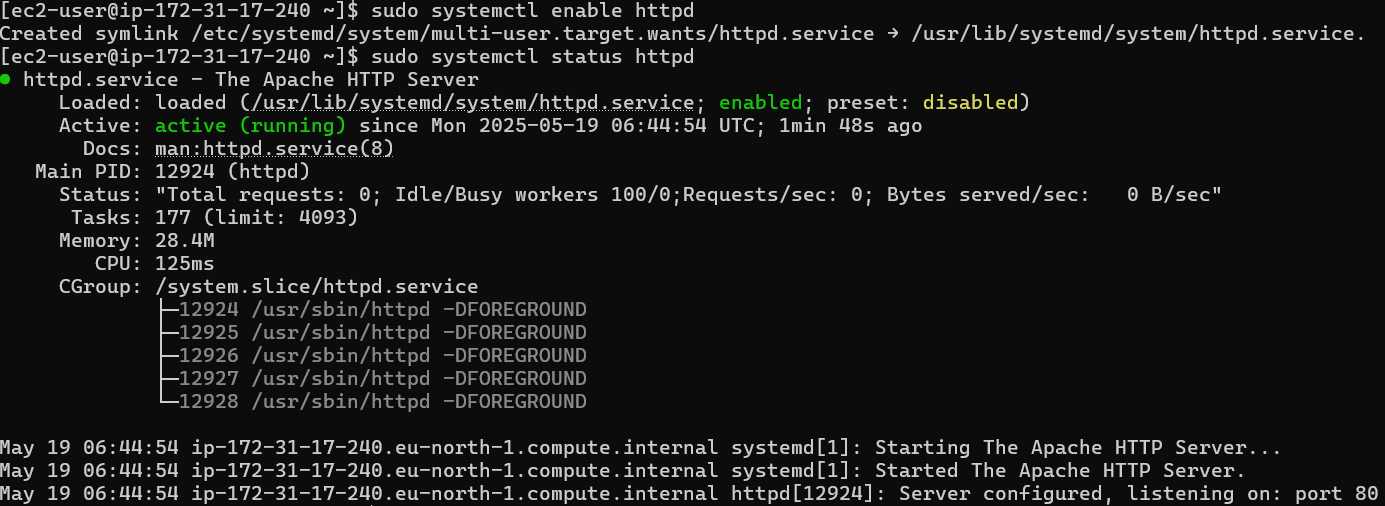
1. View all active services using systemctl list-units –type=service



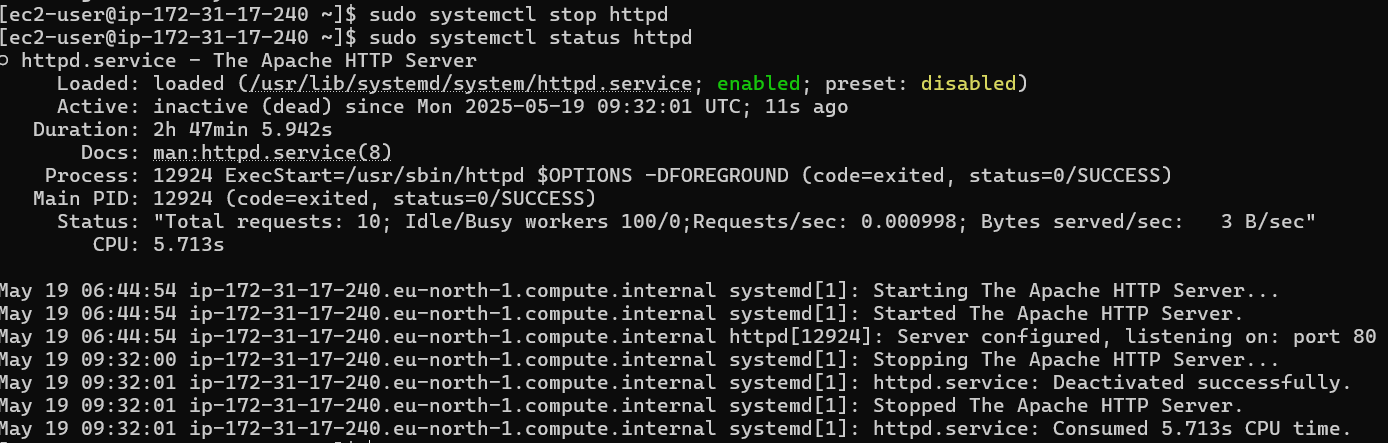
1. Start the httpd service



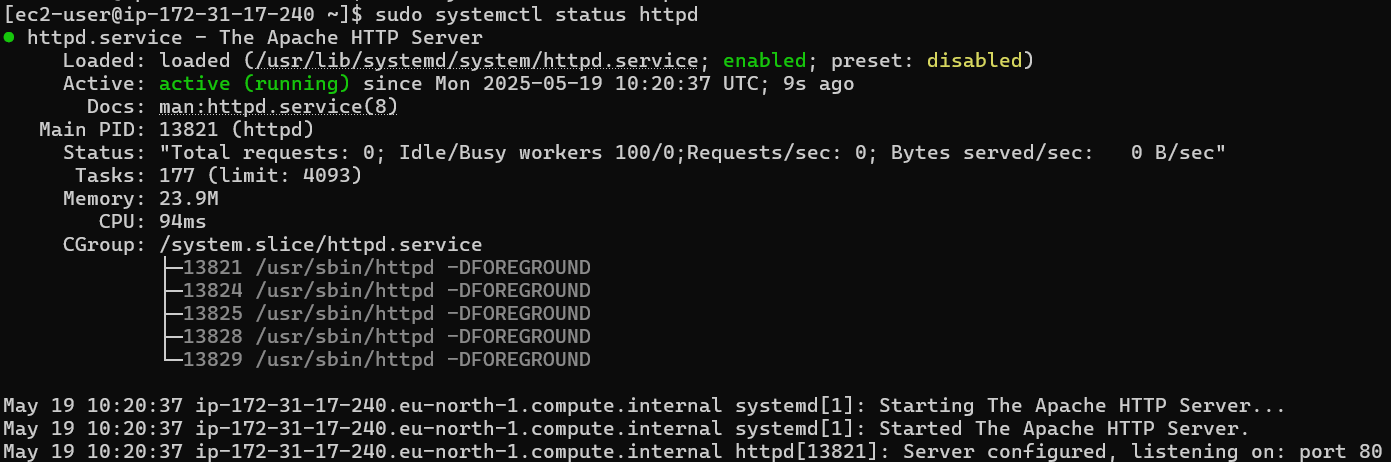
1. Enable httpd at boot



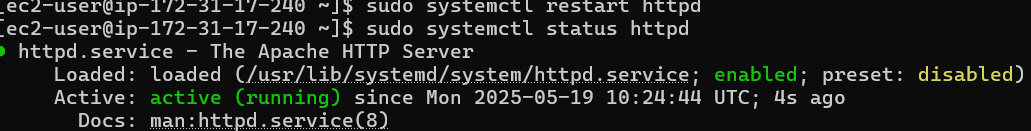
1. Stop the httpd service



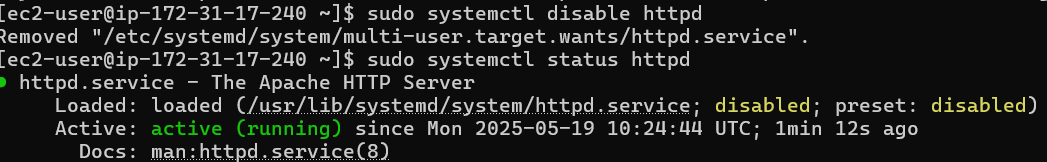
1. View the status of httpd service



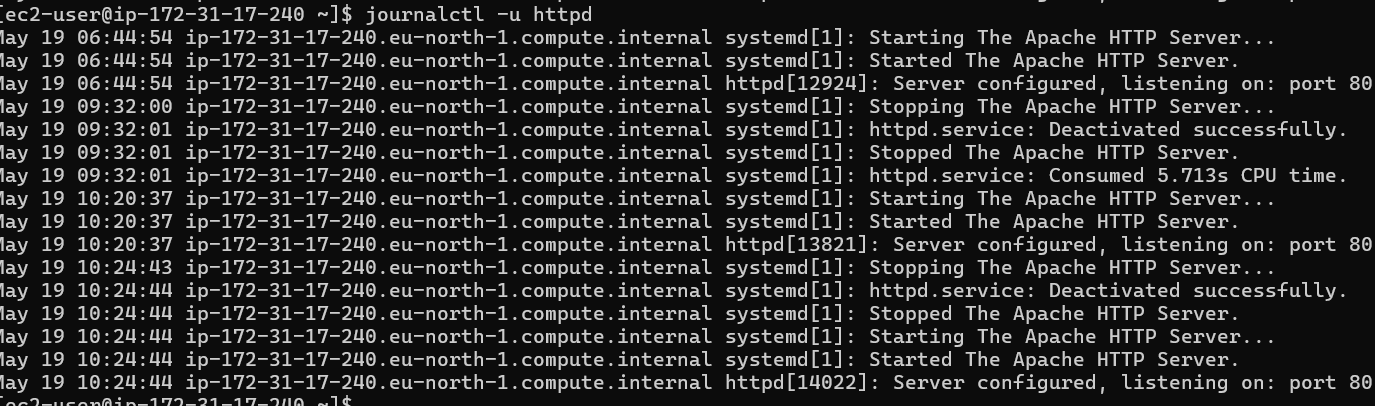
1. Restart the httpd service



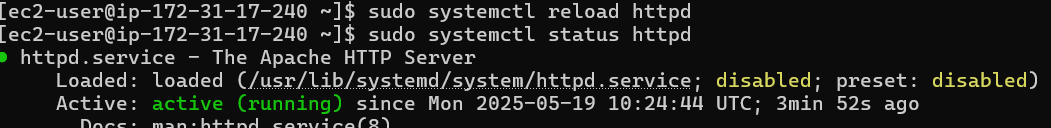
1. Disable httpd service



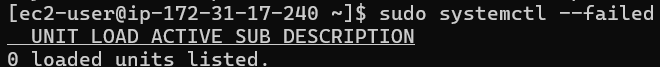
1. Use journalctl -u httpd to view logs



1. Reload a running service systemctl reload httpd

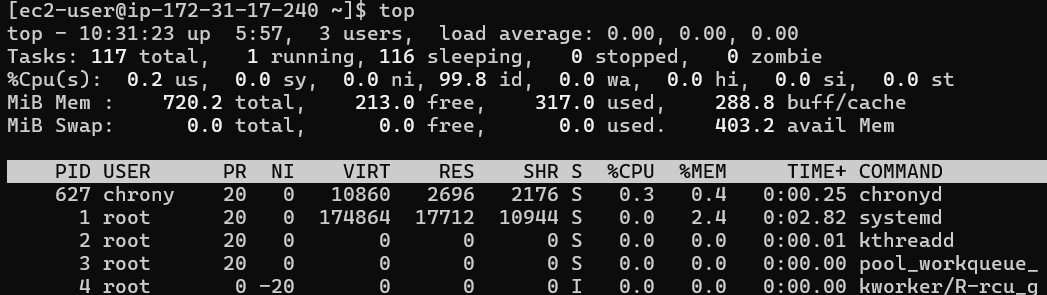


1. Check which services failed using systemctl –failed



**System performance monitoring + system info**

1. Run top to monitor system processes



1. Use htop for a user friendly view(install if needed)

Htop command not found and try to install it through yum install htop then no matches found. Then I ran the command yum update. Even though no match htop.

To install epel-release need red hat subscription but I don’t have.

So, I install epel-release manually by the following command.

sudo dnf install -y \

<https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm>

but the instance stucked because I did yum update.

After that I rebooted the instance and ran the command again. But no dnf command run successfully and also it shows killed at the end. After some searching for the error I got the reason that the instance out of memory. Then I ran some of the below commands toclear the cache and extend the memory with swap.

sudo dnf clean all

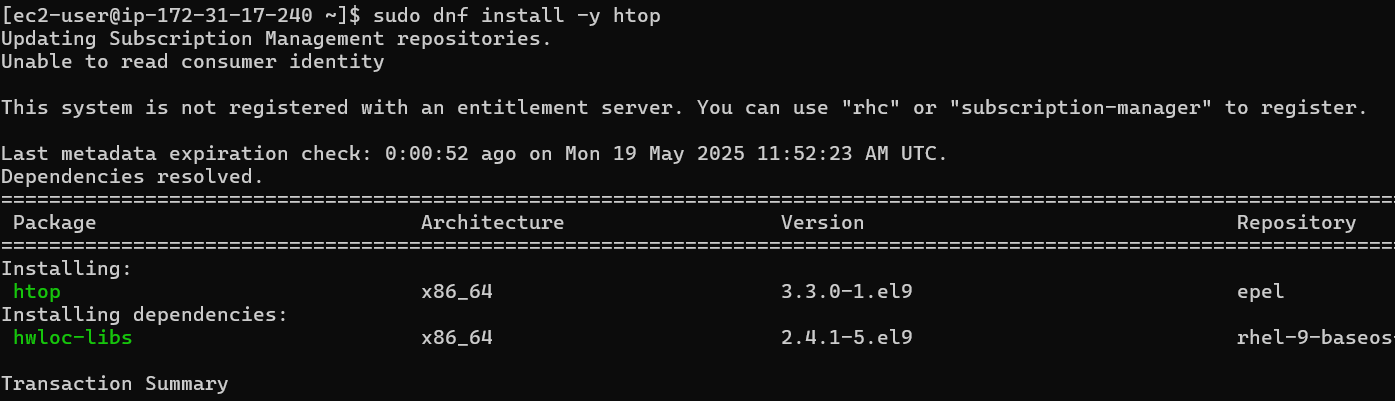
sudo dnf makecache --setopt=metadata\_timer\_cache=0

sudo dd if=/dev/zero of=/swapfile bs=1M count=1024

sudo chmod 600 /swapfile

sudo mkswap /swapfile

sudo swapon /swapfile

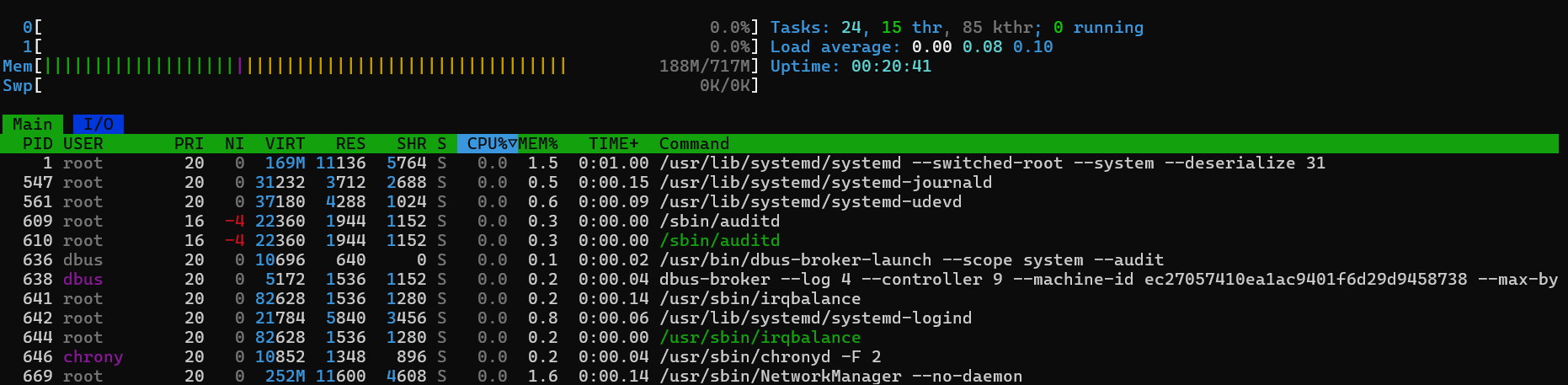


After that disable swap:

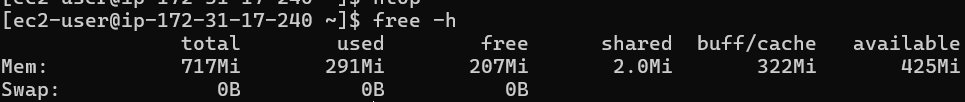
sudo swapoff /swapfile

sudo rm /swapfile

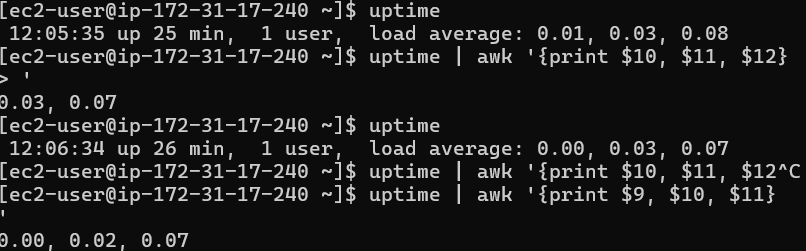
htop command output:



1. Use free -h to view ram and swap



1. Run uptime and just extract the load average using
2. Uptime | awk ‘{print $9, $10, $11}



1. Get architecture info with uname -m



1. Display kernel version using uname -r



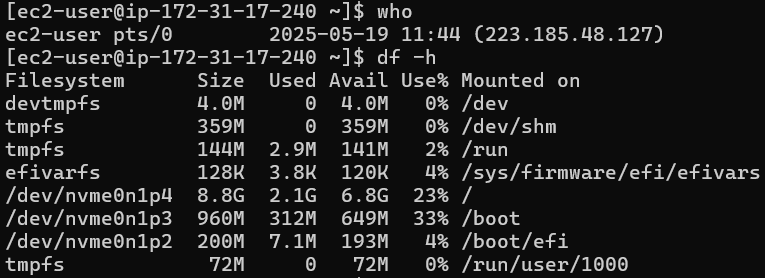
1. Show os type with uname -o



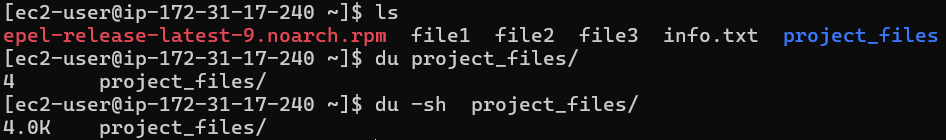
1. View logged in users with who



1. Show disk usage with df -h

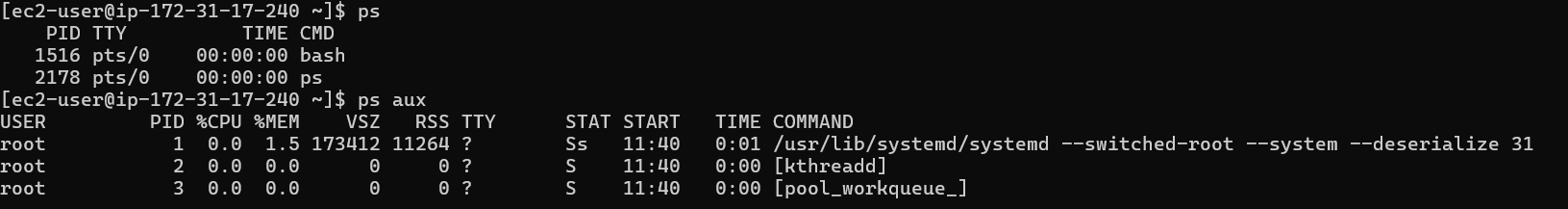


1. Use du to check any folder size



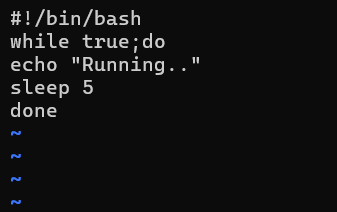
**Process management**

1. Run ps and ps aux to list all currently running processes



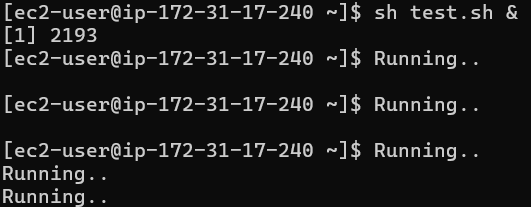
1. Create a shell script(test.sh) that runs an infinite loop





1. Start the script in the background

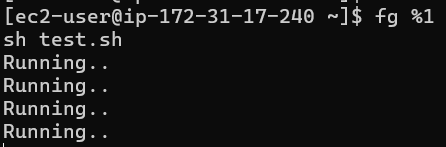
sh test.sh &



1. View background jobs using jobs



1. Bring the script to the foreground



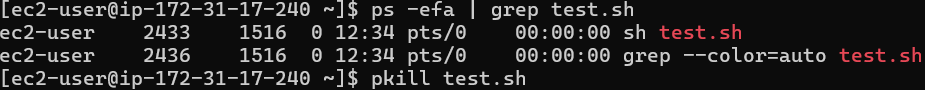
1. Suspend the running script with ctrl+z then send it back to background



1. Kill the background process using its pid

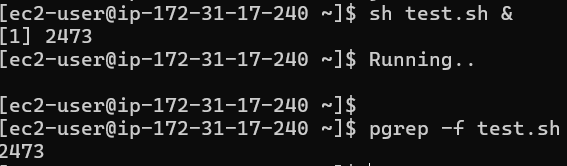


1. Kill the script by name using pkill test.sh





1. Find the script’s pid using pgrep -f test.sh



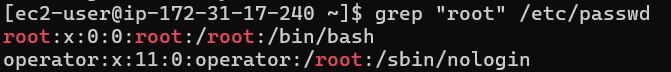
1. Run the script in the background and make it immune to terminal closure

Nohup sh test.sh &

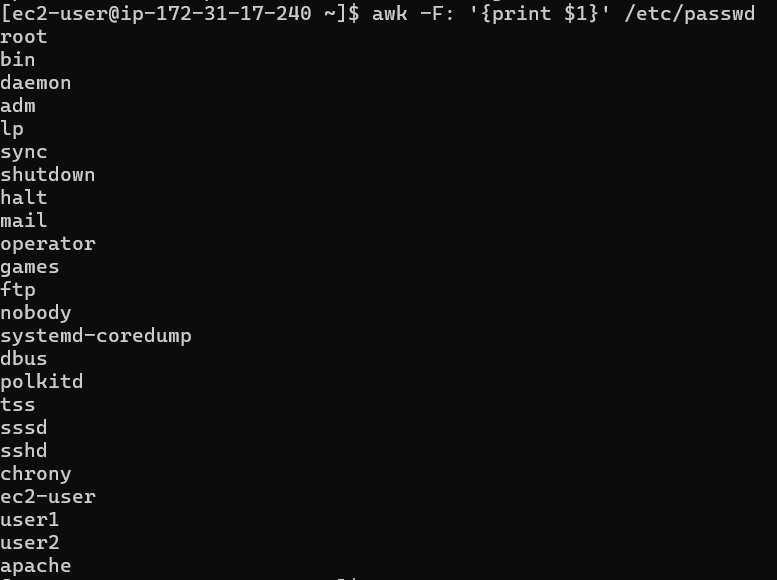


**Text processing(grep, awk, sed, find)**

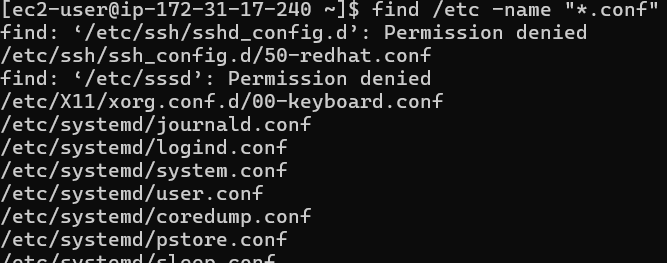
1. Use grep “root” /etc/passwd to find lines with root



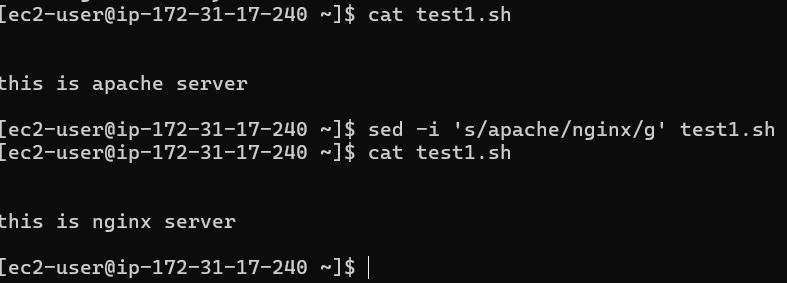
1. Use awk -F: ‘{print $1} /etc/passwd to list usernames



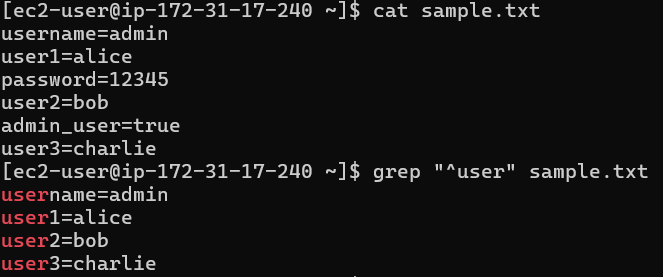
1. Find all .conf files in /etc using find /etc -name “\*.conf”



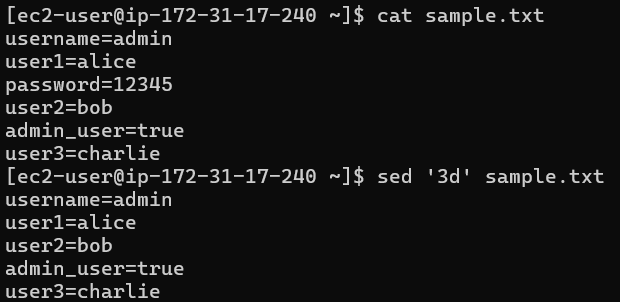
1. Replace “apache” with “nginx” in a test file using sed



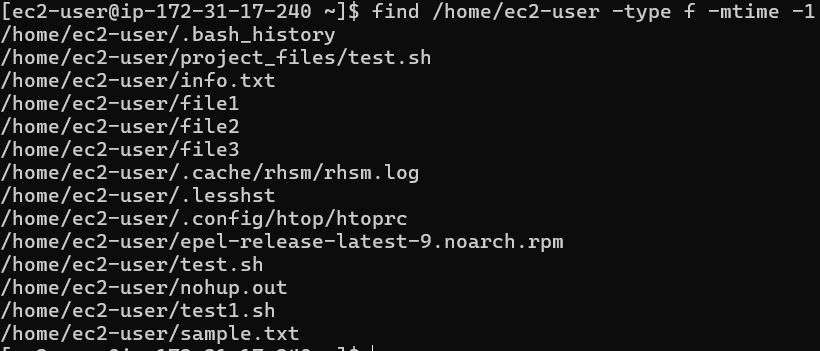
1. Use grep “^user” to find lines starting with user



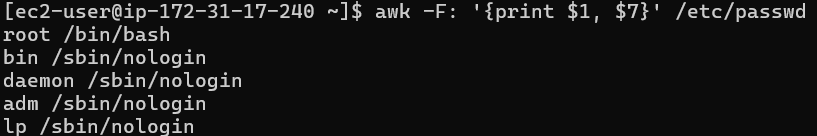
1. Delete line 3 of a file using sed ‘3d’ filename



1. Use find /home/ec2-user -type f -mtime -1 to locate recent files



1. Extract usernames and shells from /etc/passwd using awk -F: ‘{print $1, $7}’

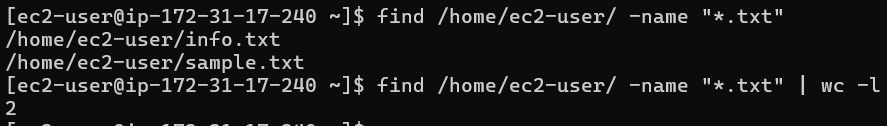


1. Use grep -I “error” /var/log/messages to locate recent files



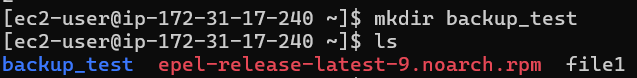
1. Count .txt in shared folder:

Find /home/ec2-user -name “\*.txt” | wc -l

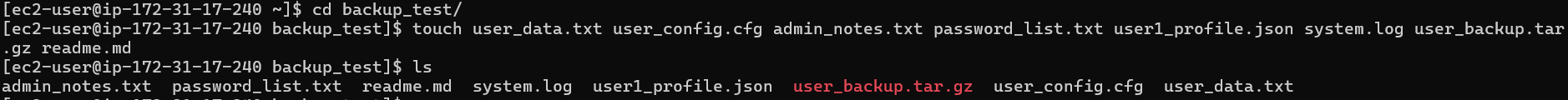


**Backup and Restore**

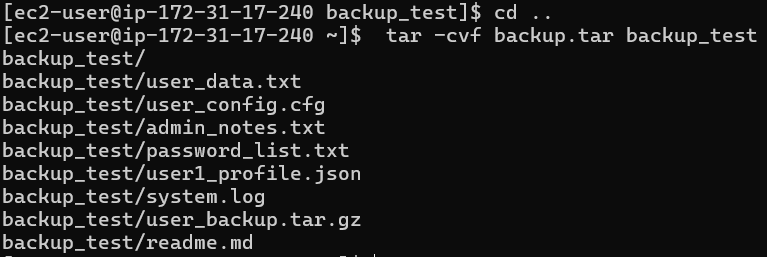
1. Create a directory backup\_test



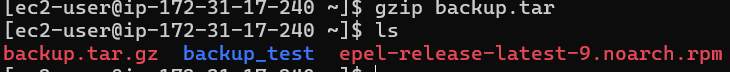
1. Add a few files to the directory



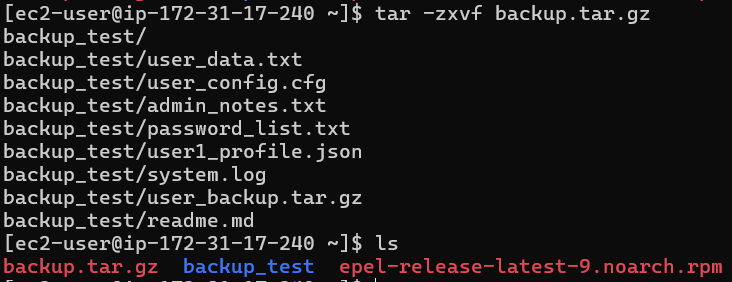
1. Create a backup using tar -cvf backup.tar backup\_test



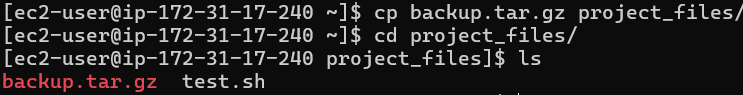
1. Compress the backup using gzip



1. Extract the backup using tar -zxvf



1. Copy the backup to another location.



1. Create a daily backup script

#!/bin/bash

# === Configuration ===

SOURCE\_DIR="/home/ec2-user/backup\_test" # Directory to back up

BACKUP\_DIR="/home/ec2-user/daily-backup" # Where backups are stored

DATE=$(date +%Y-%m-%d) # Current date for filename

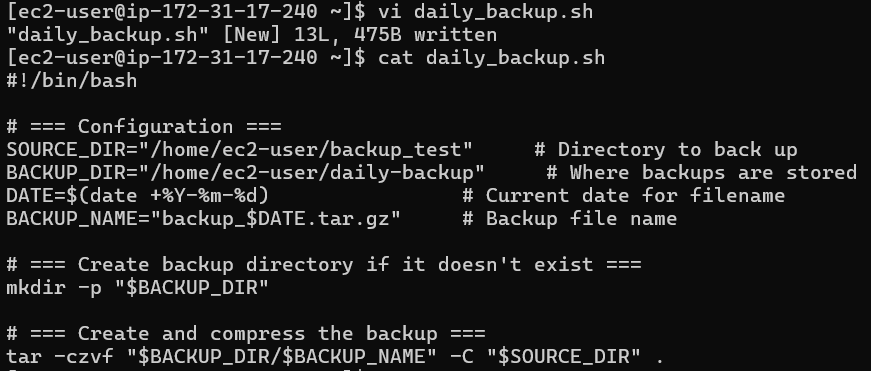
BACKUP\_NAME="backup\_$DATE.tar.gz" # Backup file name

# === Create backup directory if it doesn't exist ===

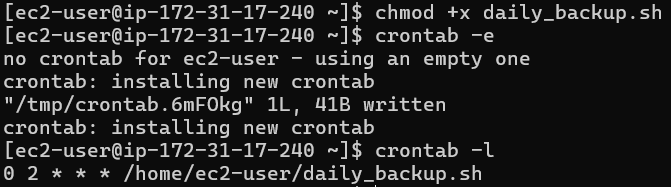
mkdir -p "$BACKUP\_DIR"

# === Create and compress the backup ===

tar -czvf "$BACKUP\_DIR/$BACKUP\_NAME" -C "$SOURCE\_DIR"



1. Schedule the script with crontab



1. View current crontab jobs



1. Test restoring single file from the archive

