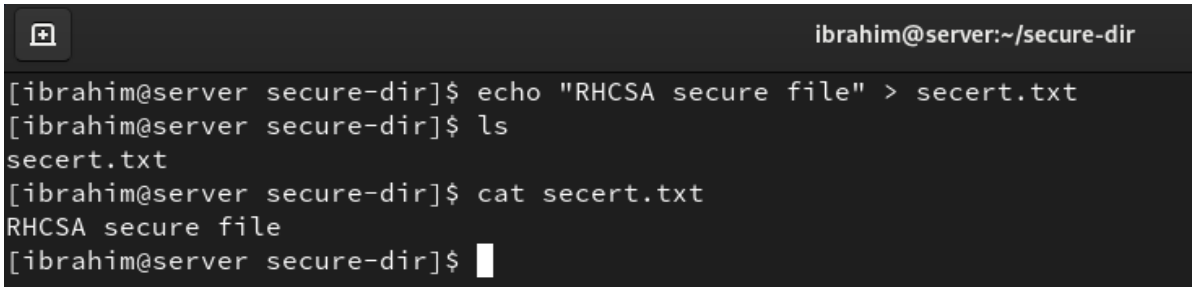
Task 1: Managing File Permissions (Chapter 7 - Controlling Access to Files)

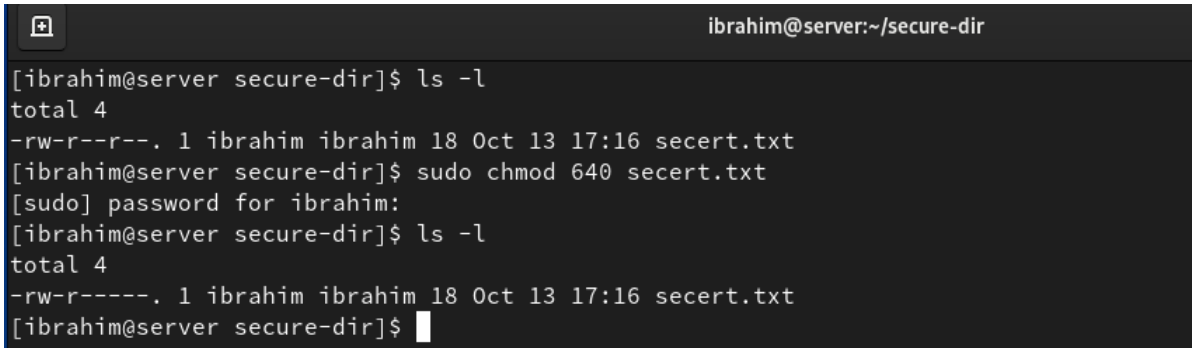
1- Set File Permissions:

a- Create a directory called secure-dir inside your home directory

b-Inside secure-dir, create a file named secret.txt and add the text "RHCSA Secure File" to it.



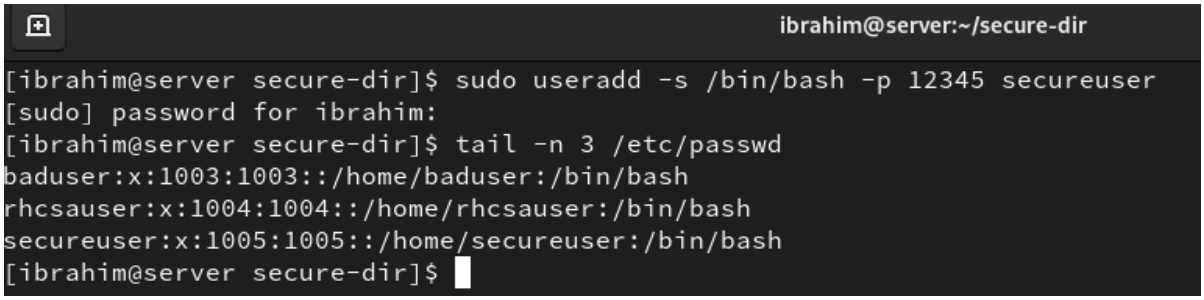
c- Set the permissions of secret.txt so that: The owner has read and write access. The group has read-only access. Others have no access.



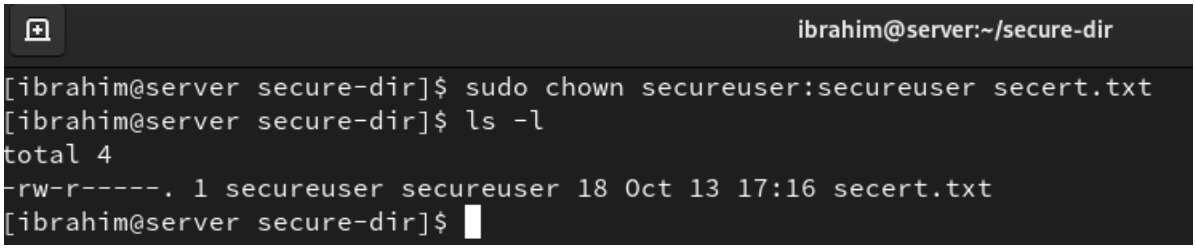
>> the above steps showing the verification you asked for.

2- Changing Ownership:

a- Create a new user named secureuser



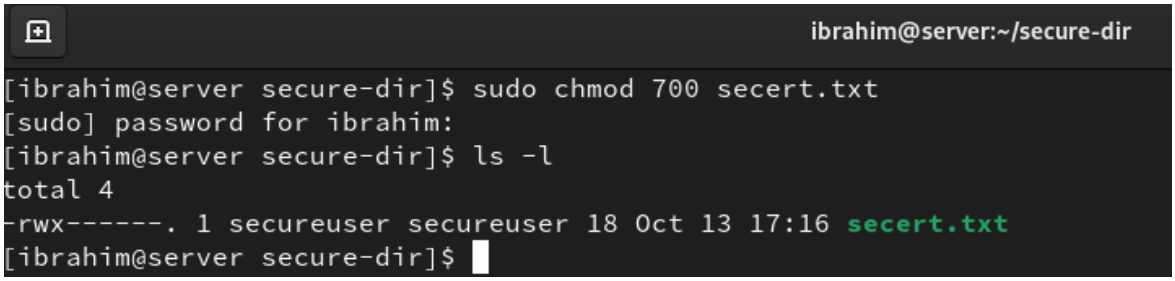
b- Change the ownership of secret.txt so that secureuser is the owner, and the group is set to securegroup (create this group if necessary).



>> the above steps showing the verification you asked for

3- Modifying Permissions:

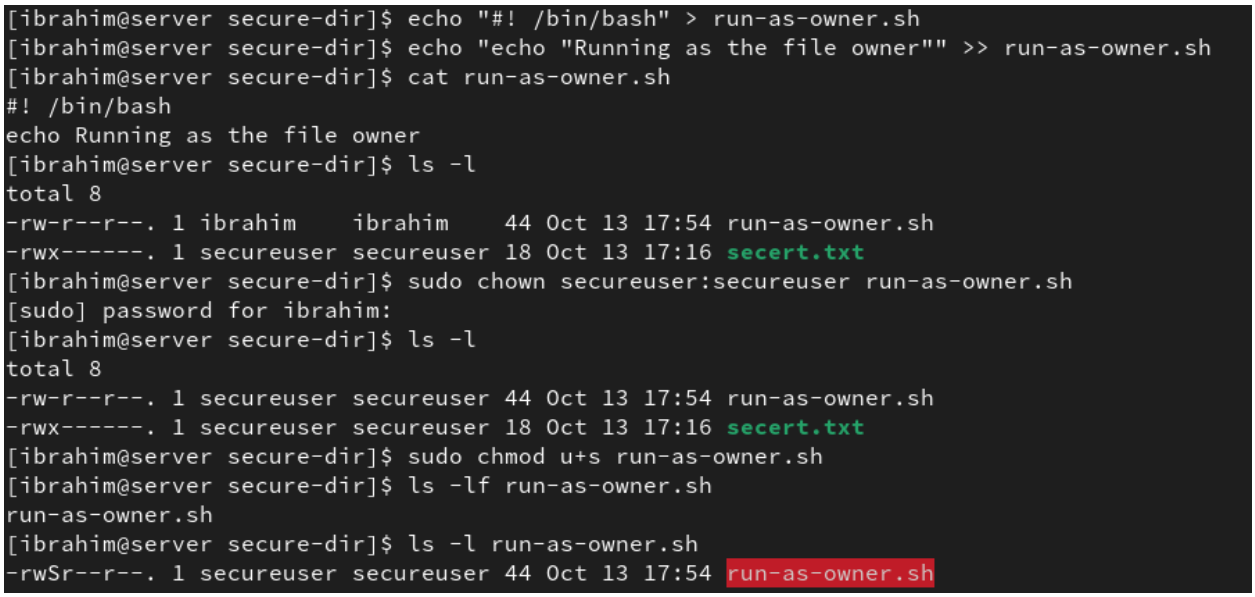
a- Use the chmod command to: Add execute permission for the owner of secret.txt. Remove all permissions for others.



>> the above steps showing the verification you asked for

4- Special Permissions:

a- Apply the SetUID permission to a script called run-as-owner.sh that you create in secure-dir. Ensure that the owner of the script is secureuser and it prints "Running as the file owner".



b- Verify that the script runs with the permissions of the file owner.

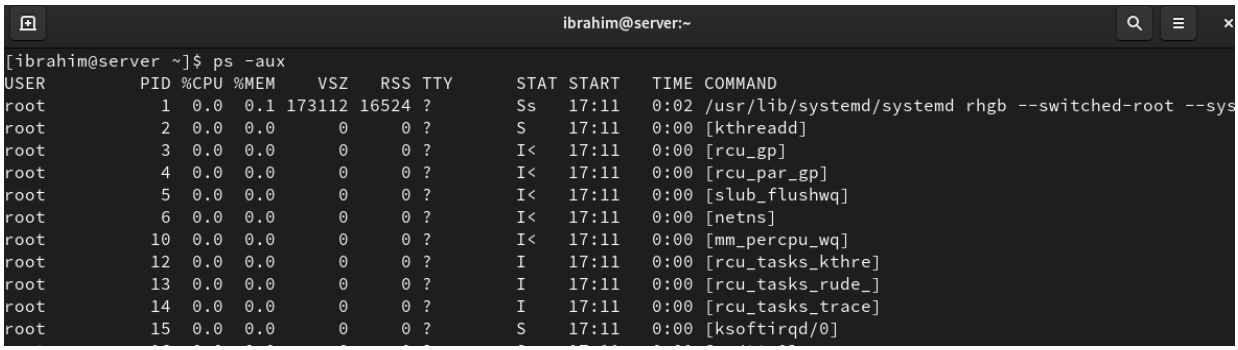
>> To verify that you need make this file executable.

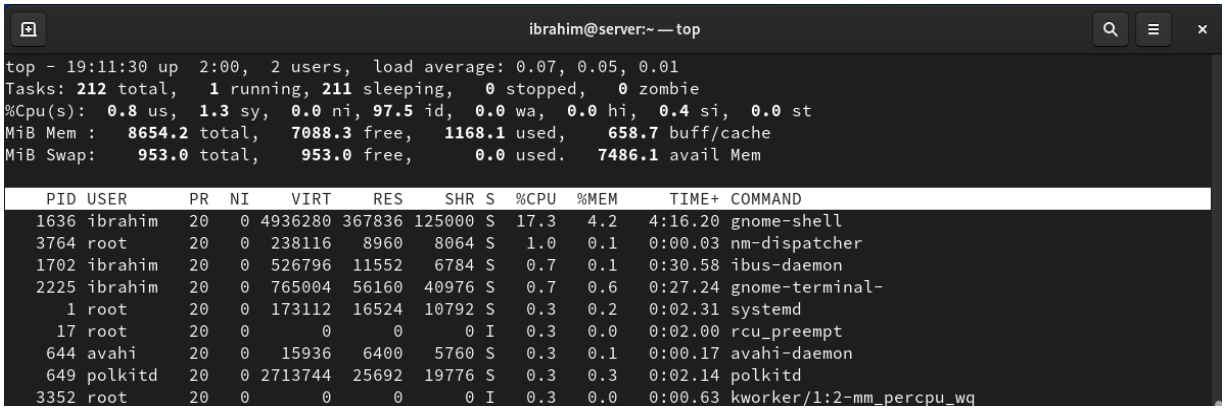
```
[ibrahim@server secure-dir]$ ls -l run-as-owner.sh
-rwsr-xr-x. 1 secureuser secureuser 47 Oct 13 21:23 run-as-owner.sh
[ibrahim@server secure-dir]$ ./run-as-owner.sh
Running as the file owner
[ibrahim@server secure-dir]$
```

Task 2: Monitoring and Managing Processes (Chapter 8 - Monitoring and Managing Linux Processes

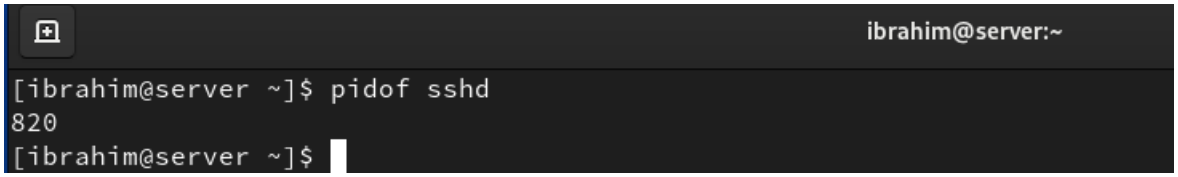
1- List Running Processes:

a- Use the ps and top commands to list all processes currently running on the system





b- Identify the PID (Process ID) of the sshd service.



c- Capture and explain the output of top and ps aux commands

>>the capture of the two commands are provided at point (a) while the explanation of the two outputs is that the ps command gives a static information of the system at a specific time this information includes(the process id , cpu utilization,memory utilization for each procees ,and the user who run the procces)

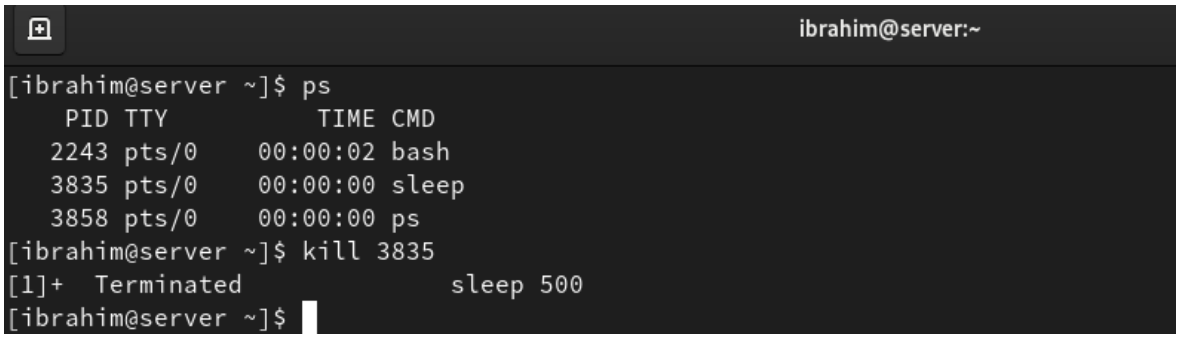
While the top command gives a dynamic information about the system this informations are change each 3 seconds by default (provide a live info) this information includes(process id, the priority of the process , cpu and memory utilization of each process , the user who

runs the process) in addition to these information the top command can also display the total utilization of the memory and cpu by using the customization options and more.

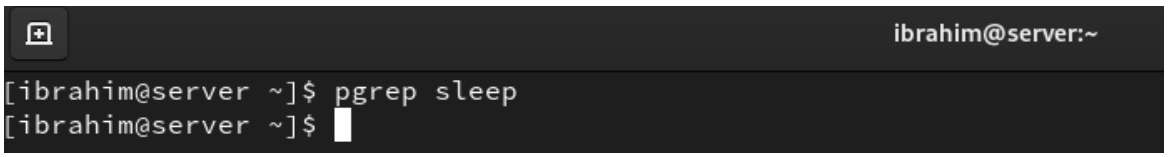
2- Killing Processes:

a- Start a long-running process using the sleep 500 command in the background.

b- Use ps to find the PID of the sleep process and terminate it using the kill command



c- Verify that the process has been terminated.



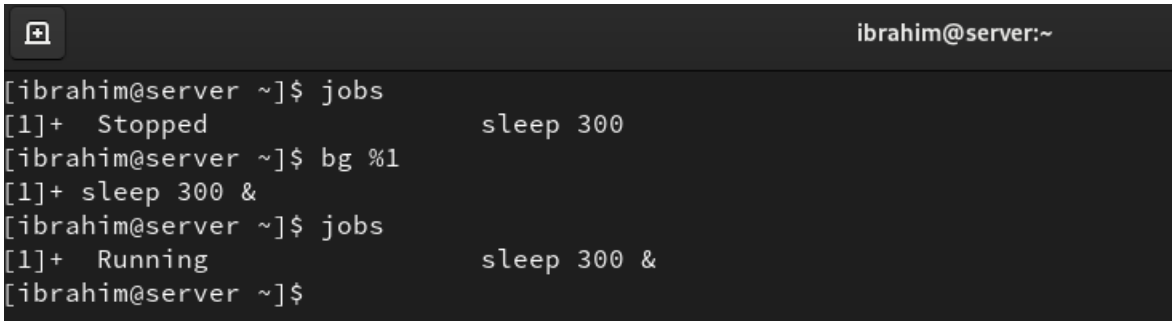
>> you will note that there is no output (you can also use the job command and you will find no output also).

3- Job Control:

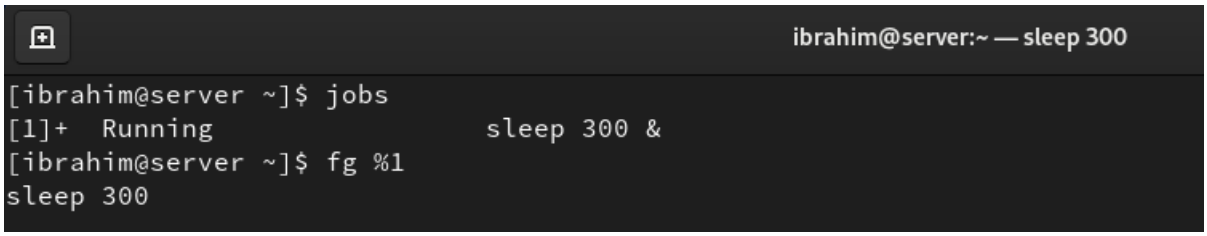
a- Start the sleep 300 command in the foreground.

b- Suspend the process using Ctrl + Z, and verify it with jobs.

c- Resume the process in the background using bg.

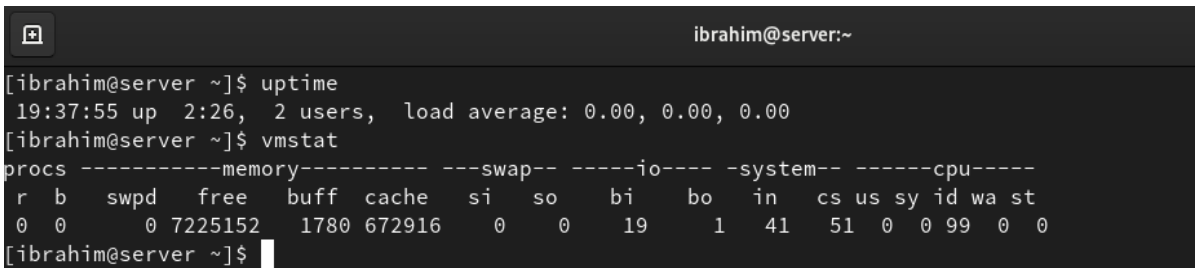


d- Bring the process back to the foreground using fg.



4- Monitoring System Activity:

a- Use the uptime and vmstat commands to display system load and memory usage.



b- Capture the output of both commands and explain the meaning of load averages, free memory, and swap usage.

>> the capture is provided above. The meaning of the load average is represents the average number of processes that are either in a runnable state (using CPU) or waiting for I/O (like disk or network).

A load average of 1 means one CPU is fully utilized.

If your system has 4 CPUs, a load average of 4.00 means the system is fully utilized but not overloaded.

If the load average exceeds the number of CPUs (e.g., 8.00 on a 4-core machine), the system may be overloaded.

>> in the output above the values (0.00,0.00,0.00) indicate the load avg of the system last 1,5,15 minutes which indicates that at that time there is no load on the system.

While the free memory indicates how much RAM is available. If free memory is very low, it might indicate that the system is under memory pressure.

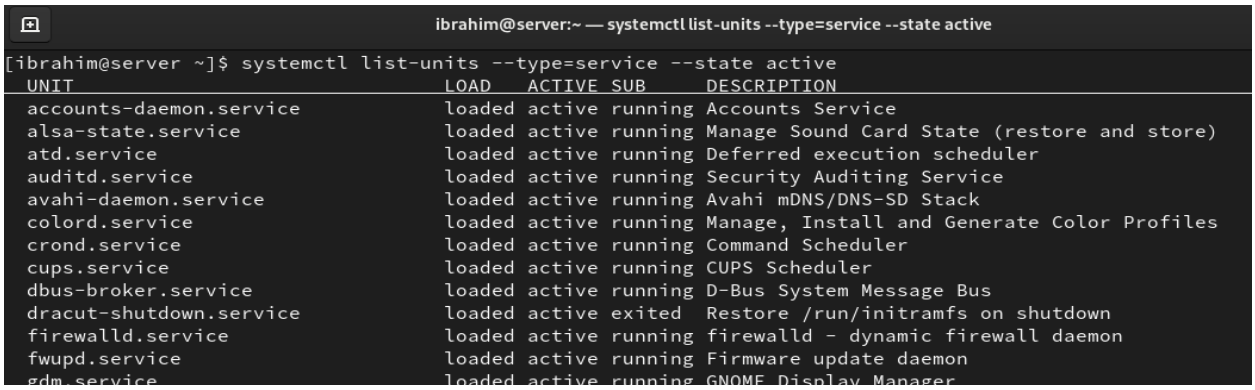
The swap usage indicates how much memory from ram is used by or swapped to the disk.

If swpd is non-zero, it means some memory is being swapped to disk. A high swap usage could indicate memory shortages and can slow down system performance.

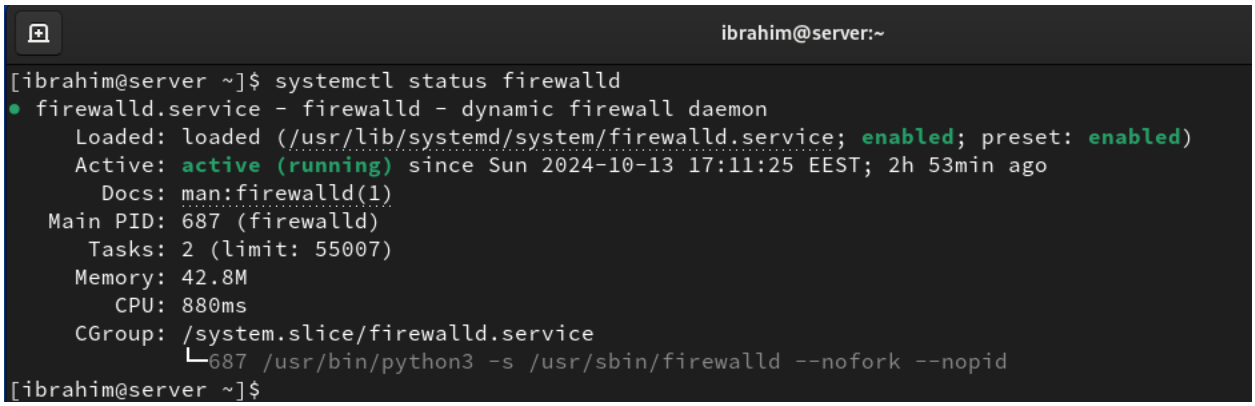
Task 3: Controlling Services and Daemons (Chapter 9 - Controlling Services and Daemons)

1- Managing Systemd Services:

a- List all active systemd services using systemctl.



b- Find the status of the firewalld service. If it's not running, start it and enable it to start at boot.

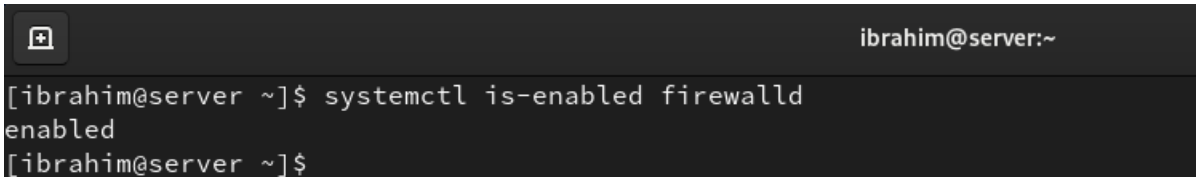


>>The service is up and running so no need to start or enable it but you can use the >>systemctl enable firewalld command to make it start the service during the boot time.

c- Verify that the service is now active and will start on boot

>> you can check that the service is running by the systemctl status firewalld command

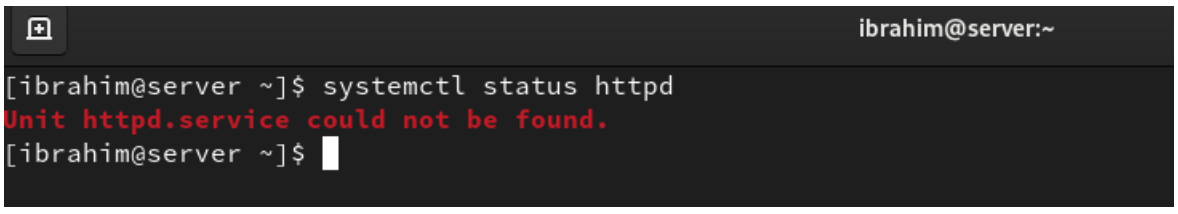
>>You can see below how to check if the service is enabled or not



2- Enable and Disable Services:

a- Disable the httpd service

>>service is not exist … so I need to download it



>>download it

```
[ibrahim@server ~]$ sudo yum install -y httpd
[sudo] password for ibrahim:
Updating Subscription Management repositories.
Waiting for process with pid 4757 to finish.
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)                    2.0 MB/s |  41 MB     00:20
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)                       2.0 MB/s |  32 MB     00:15
Dependencies resolved.
================================================================================================================
 Package                  Architecture   Version                  Repository                              Size
================================================================================================================
Installing:
 httpd                    x86_64         2.4.57-11.el9_4.1        rhel-9-for-x86_64-appstream-rpms         51 k
Installing dependencies:
 apr                      x86_64         1.7.0-12.el9_3           rhel-9-for-x86_64-appstream-rpms        126 k
 apr-util                 x86_64         1.6.1-23.el9             rhel-9-for-x86_64-appstream-rpms         97 k
 apr-util-bdb             x86_64         1.6.1-23.el9             rhel-9-for-x86_64-appstream-rpms         14 k
 httpd-core               x86_64         2.4.57-11.el9_4.1        rhel-9-for-x86_64-appstream-rpms        1.5 M
 httpd-filesystem         noarch         2.4.57-11.el9_4.1        rhel-9-for-x86_64-appstream-rpms         14 k
 httpd-tools              x86_64         2.4.57-11.el9_4.1        rhel-9-for-x86_64-appstream-rpms         86 k
```

>>check its status again: it is dead

```
ibrahim@server:~
[ibrahim@server ~]$ systemctl status httpd
○ httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: inactive (dead)
       Docs: man:httpd.service(8)
[ibrahim@server ~]$
```

b- Now, re-enable the httpd service and start it again. Verify it is running.

```
[ibrahim@server ~]$ sudo systemctl start httpd
[sudo] password for ibrahim:
[ibrahim@server ~]$ systemctl status httpd
● httpd.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
     Active: active (running) since Sun 2024-10-13 20:31:05 EEST; 9s ago
       Docs: man:httpd.service(8)
   Main PID: 36155 (httpd)
     Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec:   0 B/sec"
      Tasks: 177 (limit: 55007)
     Memory: 50.3M
        CPU: 168ms
     CGroup: /system.slice/httpd.service
             ├─36155 /usr/sbin/httpd -DFOREGROUND
             ├─36156 /usr/sbin/httpd -DFOREGROUND
             ├─36157 /usr/sbin/httpd -DFOREGROUND
             ├─36158 /usr/sbin/httpd -DFOREGROUND
             └─36159 /usr/sbin/httpd -DFOREGROUND
[ibrahim@server ~]$
```

>> Enable the service

```
ibrahim@server:~
[ibrahim@server ~]$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/system
[ibrahim@server ~]$ systemctl is-enabled httpd
enabled
```

3- Automated Start of Services:

a- Set up the crond service to start automatically at boot if it isn't already.
Verify its status and ensure it's enabled with systemctl is-enabled crond
(the service is already started and enabled)

```
[ibrahim@server ~]$ systemctl status crond
● crond.service - Command Scheduler
     Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-10-13 17:11:26 EEST; 3h 24min ago
   Main PID: 841 (crond)
      Tasks: 1 (limit: 55007)
     Memory: 1.2M
        CPU: 233ms
     CGroup: /system.slice/crond.service
             └─841 /usr/sbin/crond -n
[ibrahim@server ~]$ systemctl is-enabled crond
enabled
[ibrahim@server ~]$
```

4- Analyzing Logs for Services:

a- Use the journalctl command to view the logs of the sshd service.

```
ibrahim@server:~
[ibrahim@server ~]$ sudo journalctl -u sshd
[sudo] password for ibrahim:
Oct 13 17:11:25 server.com systemd[1]: Starting OpenSSH server daemon...
Oct 13 17:11:26 server.com sshd[820]: Server listening on 0.0.0.0 port 22.
Oct 13 17:11:26 server.com sshd[820]: Server listening on :: port 22.
Oct 13 17:11:26 server.com systemd[1]: Started OpenSSH server daemon.
[ibrahim@server ~]$
```

b- Identify the last time the sshd service was restarted
Save the output of this log to a file named sshd-log.txt

```
ibrahim@server:~
[ibrahim@server ~]$ sudo journalctl -u sshd | grep started > sshd-logs.txt
[ibrahim@server ~]$
```

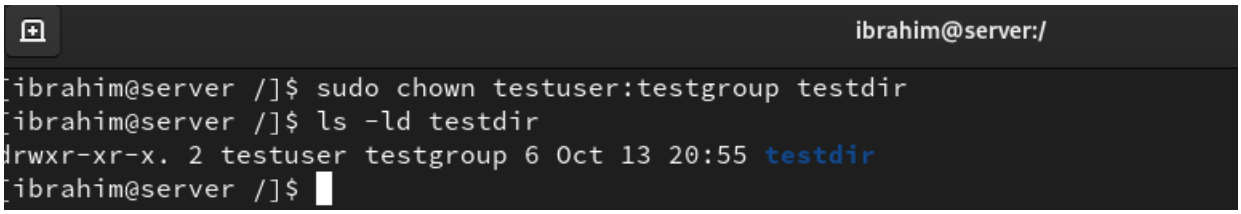
Task 4: Comprehensive Lab - File Permissions, Processes, and Services

1- Comprehensive Scenario:

a- Create a new user named testuser and add them to a new group called testgroup.

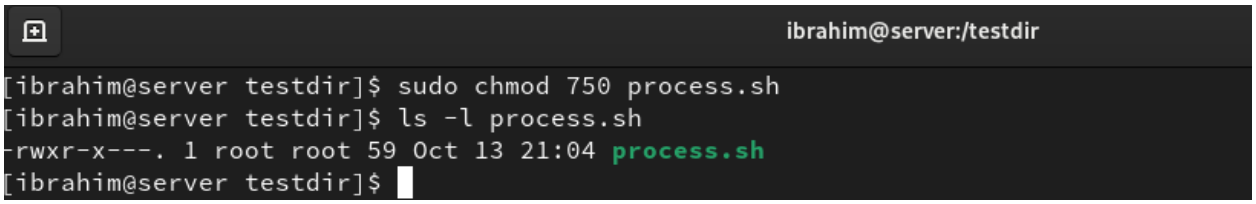
```
ibrahim@server:~
[ibrahim@server ~]$ sudo useradd testuser
[ibrahim@server ~]$ sudo groupadd testgroup
[ibrahim@server ~]$ sudo usermod -aG testgroup testuser
[ibrahim@server ~]$ sudo tail -n 3 /etc/group
apache:x:48:
testuser:x:1006:
testgroup:x:30003:testuser
[ibrahim@server ~]$ sudo tail -n 3 /etc/passwd
secureuser:x:1005:1005::/home/secureuser:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
testuser:x:1006:1006::/home/testuser:/bin/bash
[ibrahim@server ~]$ id testuser
uid=1006(testuser) gid=1006(testuser) groups=1006(testuser),30003(testgroup)
```

b- Create a directory /testdir with testuser as the owner and testgroup as the group

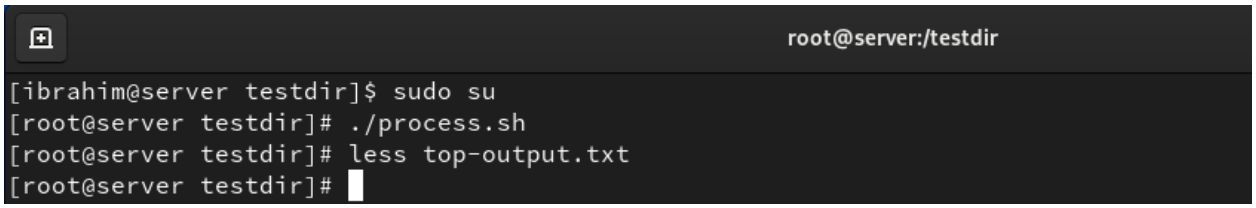


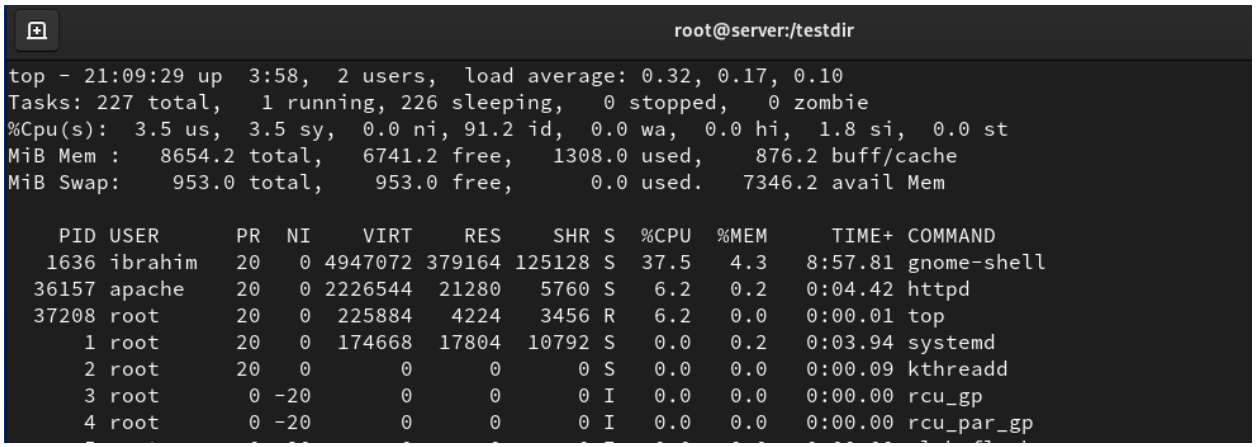
c- Inside /testdir, create a file named process.sh that runs the top command for 30 seconds and saves the output to top-output.txt.

d- Set the following permissions for /testdir/process.sh: The owner has read, write, and execute permissions.The group has read and execute permissions.Others have no permissions



e- As the root user, execute process.sh and verify that it creates the top-output.txt file in /testdir.





2- Service Management:

a- As the root user, stop the sshd service, wait for 10 seconds, and then start it again

```
[root@server ~]# systemctl stop sshd
[root@server ~]# sleep 10
[root@server ~]# systemctl start sshd
[root@server ~]# systemctl status sshd
● sshd.service - OpenSSH server daemon
     Loaded: loaded (/usr/lib/systemd/system/sshd.service; enabled; preset: enabled)
     Active: active (running) since Sun 2024-10-13 21:13:30 EEST; 6s ago
       Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 37241 (sshd)
      Tasks: 1 (limit: 55007)
     Memory: 1.4M
        CPU: 57ms
     CGroup: /system.slice/sshd.service
             └─37241 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

Oct 13 21:13:30 server.com systemd[1]: Starting OpenSSH server daemon...
Oct 13 21:13:30 server.com sshd[37241]: Server listening on 0.0.0.0 port 22.
Oct 13 21:13:30 server.com systemd[1]: Started OpenSSH server daemon.
Oct 13 21:13:30 server.com sshd[37241]: Server listening on :: port 22.
[root@server ~]#
```

b- Use journalctl to confirm that the service was stopped and restarted.

```
[root@server ~]# journalctl -u sshd
Oct 13 17:11:25 server.com systemd[1]: Starting OpenSSH server daemon...
Oct 13 17:11:26 server.com sshd[820]: Server listening on 0.0.0.0 port 22.
Oct 13 17:11:26 server.com sshd[820]: Server listening on :: port 22.
Oct 13 17:11:26 server.com systemd[1]: Started OpenSSH server daemon.
Oct 13 21:13:10 server.com systemd[1]: Stopping OpenSSH server daemon...
Oct 13 21:13:10 server.com sshd[820]: Received signal 15; terminating.
Oct 13 21:13:10 server.com systemd[1]: sshd.service: Deactivated successfully.
Oct 13 21:13:10 server.com systemd[1]: Stopped OpenSSH server daemon.
Oct 13 21:13:30 server.com systemd[1]: Starting OpenSSH server daemon...
Oct 13 21:13:30 server.com sshd[37241]: Server listening on 0.0.0.0 port 22.
Oct 13 21:13:30 server.com systemd[1]: Started OpenSSH server daemon.
Oct 13 21:13:30 server.com sshd[37241]: Server listening on :: port 22.
[root@server ~]#
```

c- Write a brief summary of the steps you followed and include the command output in a file named service-management-summary.txt.

>> nano service-management-summary.txt