

## 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

```
ibrahim@server:~$ pwd
/home/ibrahim
ibrahim@server:~$ mkdir secure-dir
ibrahim@server:~$ ls -ld secure-dir/
secure-dir/
ibrahim@server:~$
```

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

```
ibrahim@server:~/secure-dir$ echo "RHCSA secure file" > secert.txt
ibrahim@server:~/secure-dir$ ls
secert.txt
ibrahim@server:~/secure-dir$ cat secert.txt
RHCSA secure file
ibrahim@server:~/secure-dir$
```

- 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.

```
ibrahim@server:~/secure-dir$ ls -l
total 4
-rw-r--r--. 1 ibrahim ibrahim 18 Oct 13 17:16 secert.txt
ibrahim@server:~/secure-dir$ sudo chmod 640 secert.txt
[sudo] password for ibrahim:
ibrahim@server:~/secure-dir$ ls -l
total 4
-rw-r-----. 1 ibrahim ibrahim 18 Oct 13 17:16 secert.txt
ibrahim@server:~/secure-dir$
```

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

### 2- Changing Ownership:

- a- Create a new user named secureuser

```
ibrahim@server:~/secure-dir$ sudo useradd -s /bin/bash -p 12345 secureuser
[sudo] password for ibrahim:
ibrahim@server:~/secure-dir$ tail -n 3 /etc/passwd
baduser:x:1003:1003::/home/baduser:/bin/bash
rhcsauser:x:1004:1004::/home/rhcsauser:/bin/bash
secureuser:x:1005:1005::/home/secureuser:/bin/bash
ibrahim@server:~/secure-dir$
```

- 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).

```
ibrahim@server:~/secure-dir
[ibrahim@server secure-dir]$ sudo chown secureuser:secureuser secret.txt
[ibrahim@server secure-dir]$ ls -l
total 4
-rw-r-----. 1 secureuser secureuser 18 Oct 13 17:16 secret.txt
[ibrahim@server secure-dir]$
```

>> 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.

```
ibrahim@server:~/secure-dir
[ibrahim@server secure-dir]$ sudo chmod 700 secret.txt
[sudo] password for ibrahim:
[ibrahim@server secure-dir]$ ls -l
total 4
-rwx-----. 1 secureuser secureuser 18 Oct 13 17:16 secret.txt
[ibrahim@server secure-dir]$
```

>> 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".

```
[ibrahim@server secure-dir]$ echo "#! /bin/bash" > run-as-owner.sh
[ibrahim@server secure-dir]$ echo "echo \"Running as the file owner\"" >> run-as-owner.sh
[ibrahim@server secure-dir]$ cat run-as-owner.sh
#!/bin/bash
echo Running as the file owner
[ibrahim@server secure-dir]$ ls -l
total 8
-rw-r--r--. 1 ibrahim ibrahim 44 Oct 13 17:54 run-as-owner.sh
-rwx-----. 1 secureuser secureuser 18 Oct 13 17:16 secret.txt
[ibrahim@server secure-dir]$ sudo chown secureuser:secureuser run-as-owner.sh
[sudo] password for ibrahim:
[ibrahim@server secure-dir]$ ls -l
total 8
-rw-r--r--. 1 secureuser secureuser 44 Oct 13 17:54 run-as-owner.sh
-rwx-----. 1 secureuser secureuser 18 Oct 13 17:16 secret.txt
[ibrahim@server secure-dir]$ sudo chmod u+s run-as-owner.sh
[ibrahim@server secure-dir]$ ls -lf run-as-owner.sh
run-as-owner.sh
[ibrahim@server secure-dir]$ ls -l run-as-owner.sh
-rwsr--r--. 1 secureuser secureuser 44 Oct 13 17:54 run-as-owner.sh
```

- 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

```

[ibrahim@server ~]$ ps -aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root             1  0.0  0.1 173112 16524 ?        Ss   17:11   0:02 /usr/lib/systemd/systemd rhgb --switched-root --sys
root             2  0.0  0.0      0     0 ?        S    17:11   0:00 [kthreadd]
root             3  0.0  0.0      0     0 ?        I<   17:11   0:00 [rcu_gp]
root             4  0.0  0.0      0     0 ?        I<   17:11   0:00 [rcu_par_gp]
root             5  0.0  0.0      0     0 ?        I<   17:11   0:00 [slub_flushwq]
root             6  0.0  0.0      0     0 ?        I<   17:11   0:00 [netns]
root            10  0.0  0.0      0     0 ?        I<   17:11   0:00 [mm_percpu_wq]
root            12  0.0  0.0      0     0 ?        I    17:11   0:00 [rcu_tasks_kthre]
root            13  0.0  0.0      0     0 ?        I    17:11   0:00 [rcu_tasks_rude_]
root            14  0.0  0.0      0     0 ?        I    17:11   0:00 [rcu_tasks_trace]
root            15  0.0  0.0      0     0 ?        S    17:11   0:00 [ksoftirqd/0]

```

```

[ibrahim@server ~] -- top
top - 19:11:30 up 2:00, 2 users, load average: 0.07, 0.05, 0.01
Tasks: 212 total, 1 running, 211 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.8 us, 1.3 sy, 0.0 ni, 97.5 id, 0.0 wa, 0.0 hi, 0.4 si, 0.0 st
MiB Mem : 8654.2 total, 7088.3 free, 1168.1 used, 658.7 buff/cache
MiB Swap: 953.0 total, 953.0 free, 0.0 used, 7486.1 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND
 1636 ibrahim  20   0 4936280 367836 125000 S   17.3   4.2   4:16.20 gnome-shell
 3764 root      20   0 238116   8960   8064 S    1.0   0.1   0:00.03 nm-dispatcher
1702 ibrahim  20   0 526796  11552   6784 S    0.7   0.1   0:30.58 ibus-daemon
2225 ibrahim  20   0 765004  56160  40976 S    0.7   0.6   0:27.24 gnome-terminal-
   1 root      20   0 173112  16524  10792 S    0.3   0.2   0:02.31 systemd
  17 root      20   0      0      0      0 I    0.3   0.0   0:02.00 rcu_preempt
  644 avahi     20   0  15936   6400   5760 S    0.3   0.1   0:00.17 avahi-daemon
  649 polkitd  20   0 2713744 25692  19776 S    0.3   0.3   0:02.14 polkitd
3352 root      20   0      0      0      0 I    0.3   0.0   0:00.63 kworker/1:2-mm_percpu_wq

```

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

```

[ibrahim@server ~]$ pidof sshd
820
[ibrahim@server ~]$

```

- 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.

```
ibrahim@server:~  
[ibrahim@server ~]$ sleep 500 &  
[1] 3835  
[ibrahim@server ~]$ jobs  
[1]+  Running                  sleep 500 &  
[ibrahim@server ~]$
```

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

```
ibrahim@server:~  
[ibrahim@server ~]$ ps  
  PID TTY          TIME CMD  
 2243 pts/0    00:00:02 bash  
 3835 pts/0    00:00:00 sleep  
 3858 pts/0    00:00:00 ps  
[ibrahim@server ~]$ kill 3835  
[1]+  Terminated              sleep 500  
[ibrahim@server ~]$
```

- c- Verify that the process has been terminated.

```
ibrahim@server:~  
[ibrahim@server ~]$ pgrep sleep  
[ibrahim@server ~]$
```

>> 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.

```
ibrahim@server:~ — sleep 300  
[ibrahim@server ~]$ sleep 300
```

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

```
ibrahim@server:~  
[ibrahim@server ~]$ sleep 300  
^Z  
[1]+  Stopped                  sleep 300  
[ibrahim@server ~]$ jobs  
[1]+  Stopped                  sleep 300  
[ibrahim@server ~]$
```

- c- Resume the process in the background using bg.

```
ibrahim@server:~  
[ibrahim@server ~]$ jobs  
[1]+  Stopped                  sleep 300  
[ibrahim@server ~]$ bg %1  
[1]+  sleep 300 &  
[ibrahim@server ~]$ jobs  
[1]+  Running                  sleep 300 &  
[ibrahim@server ~]$
```

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

```
ibrahim@server:~ — sleep 300  
[ibrahim@server ~]$ jobs  
[1]+  Running                  sleep 300 &  
[ibrahim@server ~]$ fg %1  
sleep 300
```

#### 4- Monitoring System Activity:

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

```
ibrahim@server:~  
[ibrahim@server ~]$ uptime  
19:37:55 up 2:26, 2 users, load average: 0.00, 0.00, 0.00  
[ibrahim@server ~]$ vmstat  
procs -----memory----- ---swap-- ----io---- -system-- -----cpu-----  
r b swpd free buff cache si so bi bo in cs us sy id wa st  
0 0 0 7225152 1780 672916 0 0 19 1 41 51 0 0 99 0 0  
[ibrahim@server ~]$
```

- 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.

```
ibrahim@server:~ — systemctl list-units --type=service --state active
[ibrahim@server ~]$ systemctl list-units --type=service --state active
UNIT                                LOAD    ACTIVE SUB    DESCRIPTION
accounts-daemon.service            loaded active running Accounts Service
alsa-state.service                 loaded active running Manage Sound Card State (restore and store)
atd.service                         loaded active running Deferred execution scheduler
auditd.service                     loaded active running Security Auditing Service
avahi-daemon.service                loaded active running Avahi mDNS/DNS-SD Stack
colord.service                      loaded active running Manage, Install and Generate Color Profiles
crond.service                      loaded active running Command Scheduler
cups.service                        loaded active running CUPS Scheduler
dbus-broker.service                loaded active running D-Bus System Message Bus
dracut-shutdown.service             loaded active exited Restore /run/initramfs on shutdown
firewalld.service                  loaded active running firewalld - dynamic firewall daemon
fwupd.service                      loaded active running Firmware update daemon
gdm.service                         loaded active running GNOME Display Manager
```

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

```
ibrahim@server:~
[ibrahim@server ~]$ systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-10-13 17:11:25 EEST; 2h 53min ago
     Docs: man:firewalld(1)
    Main PID: 687 (firewalld)
      Tasks: 2 (limit: 55007)
    Memory: 42.8M
       CPU: 880ms
    CGroup: /system.slice/firewalld.service
            └─687 /usr/bin/python3 -s /usr/sbin/firewalld --nofork --nopid
[ibrahim@server ~]$
```

>>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

```
ibrahim@server:~
[ibrahim@server ~]$ systemctl is-enabled firewalld
enabled
[ibrahim@server ~]$
```

#### 2- Enable and Disable Services:

- a- Disable the httpd service

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

```
ibrahim@server:~
[ibrahim@server ~]$ systemctl status httpd
Unit httpd.service could not be found.
[ibrahim@server ~]$
```

>>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
httpd-devel                    x86_64      2.4.57-11.el9_4.1            rhel-9-for-x86_64-appstream-rpms 18 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/systemd/system/httpd.service
[ibrahim@server ~]$ systemctl is-enabled httpd
enabled
[ibrahim@server ~]$
```

### 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

```
ibrahim@server:/  
[ibrahim@server /]$ sudo chown testuser:testgroup testdir  
[ibrahim@server /]$ ls -ld testdir  
drwxr-xr-x. 2 testuser testgroup 6 Oct 13 20:55 testdir  
[ibrahim@server /]$
```

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.

```
[ibrahim@server /]$ cd testdir/  
[ibrahim@server testdir]$ nano process.sh  
ibrahim@server:/testdir — nano process.sh  
GNU nano 5.6.1 process.sh  
#!/bin/bash  
  
top -b -n 60 -d 0.5 > /testdir/topoutput.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

```
ibrahim@server:/testdir  
[ibrahim@server testdir]$ sudo chmod 750 process.sh  
[ibrahim@server testdir]$ ls -l process.sh  
-rwxr-x---. 1 root root 59 Oct 13 21:04 process.sh  
[ibrahim@server testdir]$
```

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

```
root@server:/testdir  
[ibrahim@server testdir]$ sudo su  
[root@server testdir]# ./process.sh  
[root@server testdir]# less top-output.txt  
[root@server testdir]#
```

```
root@server:/testdir  
top - 21:09:29 up 3:58, 2 users, load average: 0.32, 0.17, 0.10  
Tasks: 227 total, 1 running, 226 sleeping, 0 stopped, 0 zombie  
%Cpu(s): 3.5 us, 3.5 sy, 0.0 ni, 91.2 id, 0.0 wa, 0.0 hi, 1.8 si, 0.0 st  
MiB Mem : 8654.2 total, 6741.2 free, 1308.0 used, 876.2 buff/cache  
MiB Swap: 953.0 total, 953.0 free, 0.0 used. 7346.2 avail Mem  
  
  PID USER      PR  NI   VIRT   RES   SHR S  %CPU  %MEM    TIME+  COMMAND  
 1636 ibrahim   20   0 4947072 379164 125128 S   37.5   4.3   8:57.81 gnome-shell  
 36157 apache    20   0 2226544 21280   5760 S    6.2   0.2   0:04.42 httpd  
 37208 root      20   0 225884   4224   3456 R    6.2   0.0   0:00.01 top  
    1 root      20   0 174668 17804 10792 S    0.0   0.2   0:03.94 systemd  
    2 root      20   0      0      0      0 S    0.0   0.0   0:00.09 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
```

## 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

```
root@server:~
GNU nano 5.6.1 service-managment-summary.txt Modified
1- switch to the root account
2- stop the ssh service using systemctl stop sshd command
3- then waited for 10s by sleep 10 command
4- start the sshd again using the systemctl start sshd
5- capture the logs using journalctl -u sshd command to makesure that the sshd service stoped and started
```