

Linux Assignment – Networking, Processes & System Info

Student: *Abdulakeem Ishola*

Email: *isholaabdulakeem@gmail.com*


Date: *28th of August 2025*

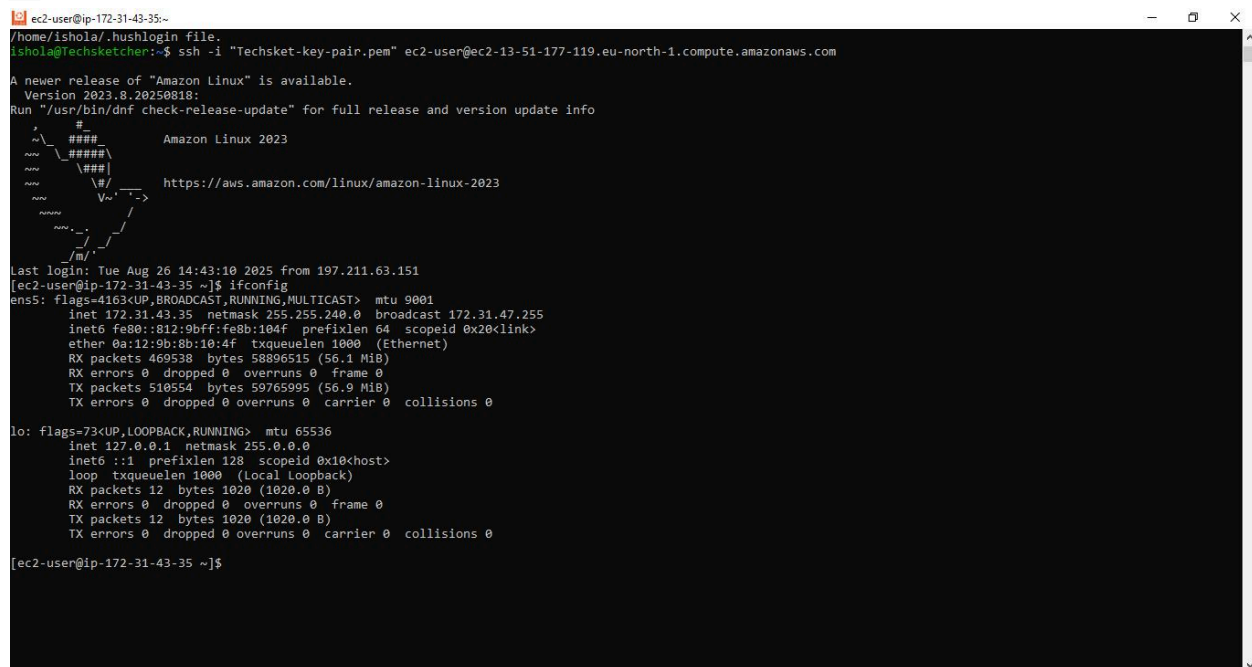
Task 1 – Networking Commands

1. Check server network interface configuration

Command: `ifconfig`

Explanation: Shows details about the server's network interfaces (IP, MAC, subnet). Useful for troubleshooting connectivity.

 Screenshot here



```
ec2-user@ip-172-31-43-35:~$ ssh -i "Techsket-key-pair.pem" ec2-user@ec2-13-51-177-119.eu-north-1.compute.amazonaws.com
A newer release of "Amazon Linux" is available.
Version 2023.8.20250818:
Run "/usr/bin/dnf check-release-update" for full release and version update info

#
##### Amazon Linux 2023
#####
##### https://aws.amazon.com/linux/amazon-linux-2023
#####

Last login: Tue Aug 26 14:43:10 2025 from 197.211.63.151
[ec2-user@ip-172-31-43-35 ~]$ ifconfig
ens5: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.43.35 netmask 255.255.240.0 broadcast 172.31.47.255
    inet6 fe80::812:9bff:fe8b:104f prefixlen 64 scopeid 0x20<link>
    ether 0a:12:9b:8b:10:4f txqueuelen 1000 (Ethernet)
    RX packets 469538 bytes 58896515 (56.1 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 510554 bytes 59765995 (56.9 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 12 bytes 1020 (1020.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 12 bytes 1020 (1020.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0


[ec2-user@ip-172-31-43-35 ~]$
```

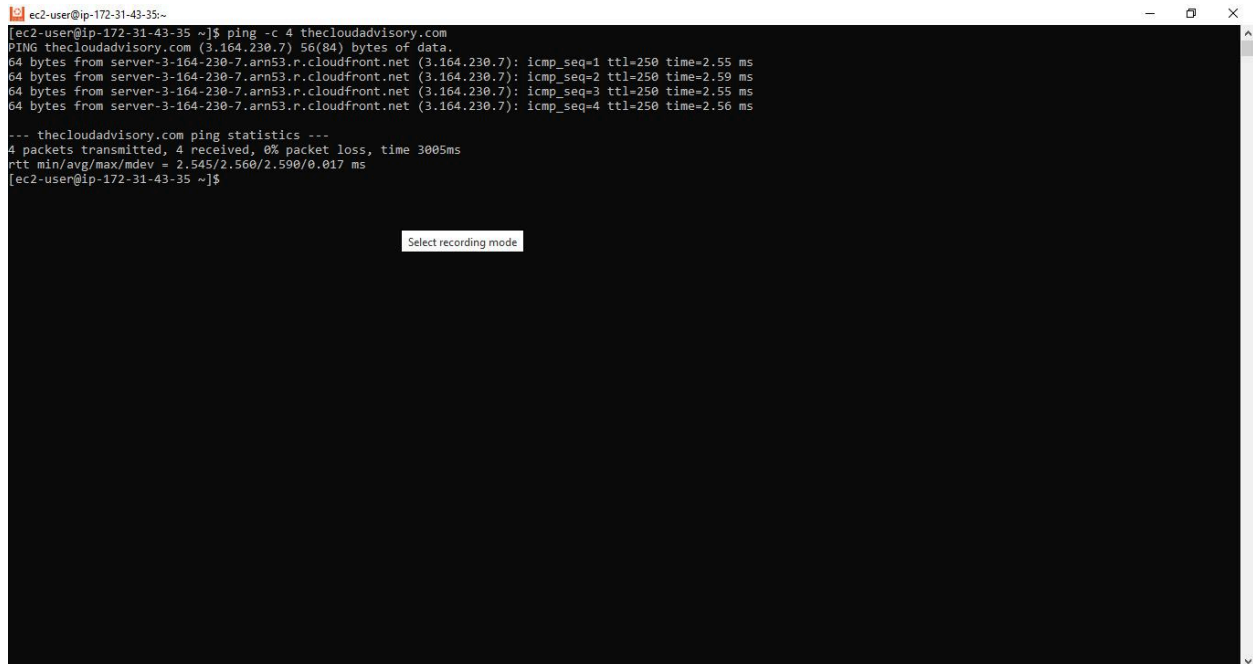
2. Test connectivity to thecloudadvisory.com

Command:

```
ping -c 4 thecloudadvisory.com
```

Explanation: Sends 4 packets to the domain to check reachability and response time. Useful for testing internet and DNS connectivity.

 *Screenshot here*



```
ec2-user@ip-172-31-43-35:~$ ping -c 4 thecloudadvisory.com
PING thecloudadvisory.com (3.164.230.7) 56(84) bytes of data:
64 bytes from server-3-164-230-7.arn53.r.cloudfront.net (3.164.230.7): icmp_seq=1 ttl=250 time=2.55 ms
64 bytes from server-3-164-230-7.arn53.r.cloudfront.net (3.164.230.7): icmp_seq=2 ttl=250 time=2.59 ms
64 bytes from server-3-164-230-7.arn53.r.cloudfront.net (3.164.230.7): icmp_seq=3 ttl=250 time=2.55 ms
64 bytes from server-3-164-230-7.arn53.r.cloudfront.net (3.164.230.7): icmp_seq=4 ttl=250 time=2.56 ms

--- thecloudadvisory.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 2.545/2.560/2.590/0.017 ms
ec2-user@ip-172-31-43-35:~$
```

3. Check open ports and listening services

Command: `sudo netstat -tulnp` (or) `ss -tuln`

Explanation: Displays which ports are open and which services are listening. Helps verify running applications and security.

 *Screenshot here*

```
ec2-user@ip-172-31-43-35:~$ sudo netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1908/sshd: /usr/sbi
tcp        0      0 0.0.0.0:80             0.0.0.0:*               LISTEN      1938/nginx: master
tcp6       0      0 :::22                  :::*                    LISTEN      1908/sshd: /usr/sbi
udp        0      0 127.0.0.1:323          0.0.0.0:*               1943/chronyd
udp        0      0 172.31.43.35:68        0.0.0.0:*               1742/systemd-networ
udp6       0      0 :::1:323               :::*                    1943/chronyd
udp6       0      0 fe80::812:9bff:fe8b:546 :::*                    1742/systemd-networ
ec2-user@ip-172-31-43-35 ~]$
```

Select recording mode

4. Perform DNS lookup for pravinmishra.in

Command:

```
dig pravinmishra.in
```

(or)

```
host pravinmishra.in
```

Explanation: Queries DNS records for the domain. Useful for checking domain resolution.

Screenshot here

```
ec2-user@ip-172-31-43-35:~$ dig pravinmishra.in
;<<>> DiG 9.18.33 <<>> pravinmishra.in
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15460
;; flags: qr rd ra; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;, udp: 4096
;; QUESTION SECTION:
;pravinmishra.in.                IN      A
;; ANSWER SECTION:
pravinmishra.in.        60      IN      A      35.154.180.163
pravinmishra.in.        60      IN      A      43.204.196.43
pravinmishra.in.        60      IN      A      52.66.149.47
;; Query time: 30 msec
;; SERVER: 172.31.0.2#53(172.31.0.2) (UDP)
;; WHEN: Thu Aug 28 12:14:08 UTC 2025
;; MSG SIZE rcvd: 92
[ec2-user@ip-172-31-43-35 ~]$
```

5. Download a file using wget

Command:

```
wget -O /tmp/Untitled-design-40.png
https://pravinmishra.in/wp-content/uploads/2023/08/Untitled-design-40.
png
```

Explanation: Downloads a file from the web and saves it in `/tmp`. Useful for fetching files/scripts directly onto a server.

 Screenshot here

```
ec2-user@ip-172-31-43-35:~$ wget -O /tmp/Untitled-design-40.png https://pravinmishra.in/wp-content/uploads/2023/08/Untitled-design-40.png
--2025-08-28 12:16:30-- https://pravinmishra.in/wp-content/uploads/2023/08/Untitled-design-40.png
Resolving pravinmishra.in (pravinmishra.in)... 43.204.196.43, 52.66.149.47, 35.154.180.163
Connecting to pravinmishra.in (pravinmishra.in)[43.204.196.43]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1702553 (1.6M) [image/png]
Saving to: '/tmp/Untitled-design-40.png'

/tmp/Untitled-design-40.png      100%[=====>] 1.62M  1.57MB/s  in 1.0s
2025-08-28 12:16:32 (1.57 MB/s) - '/tmp/Untitled-design-40.png' saved [1702553/1702553]

[ec2-user@ip-172-31-43-35 ~]$
```

Task 2 – Process Monitoring & Control

1. List all running processes

Command:

```
ps -e
```

Explanation: Lists all active processes on the system. Helps monitor running programs.

Screenshot here

```
ec2-user@ip-172-31-43-35:~$ ps -e
PID TTY          TIME CMD
  1 ?            00:00:30 systemd
  2 ?            00:00:00 kthreadd
  3 ?            00:00:00 pool_workqueue_release
  4 ?            00:00:00 kworker/R-kvfree_rcu_reclaim
  5 ?            00:00:00 kworker/R-rcu_gp
  6 ?            00:00:00 kworker/R-sync_wq
  7 ?            00:00:00 kworker/R-slab_flushwq
  8 ?            00:00:00 kworker/R-netns
 10 ?            00:00:00 kworker/0:0H-events_highpri
 13 ?            00:00:00 kworker/R-mm_percpu_wq
 14 ?            00:00:00 rcu_tasks_kthread
 15 ?            00:00:00 rcu_tasks_rude_kthread
 16 ?            00:00:00 rcu_tasks_trace_kthread
 17 ?            00:00:04 ksoftirqd/0
 18 ?            00:00:10 rcu_preempt
 19 ?            00:00:00 rcu_exp_par_gp_kthread_worker/0
 20 ?            00:00:00 rcu_exp_gp_kthread_worker
 21 ?            00:00:02 migration/0
 22 ?            00:00:00 cpuhp/0
 23 ?            00:00:00 cpuhp/1
 24 ?            00:00:02 migration/1
 25 ?            00:00:04 ksoftirqd/1
 27 ?            00:00:00 kworker/1:0H-events_highpri
 30 ?            00:00:00 kdevtmpfs
 31 ?            00:00:00 kworker/R-inet_frag_wq
 32 ?            00:00:00 kauditd
 33 ?            00:00:00 khungtaskd
 34 ?            00:00:00 oom_reaper
 36 ?            00:00:00 kworker/R-writeback
 37 ?            00:00:21 kcompactd0
 38 ?            00:00:00 khugepaged
 39 ?            00:00:00 kworker/R-cryptd
 40 ?            00:00:00 kworker/R-kintegrityd
 41 ?            00:00:00 kworker/R-kblockd
 42 ?            00:00:00 kworker/R-blkcg_punt_bio
 43 ?            00:00:00 irq/9-acpi
 45 ?            00:00:00 kworker/R-tpm_dev_wq
 46 ?            00:00:00 kworker/R-md
 47 ?            00:00:00 kworker/R-md_bitmap
 48 ?            00:00:00 kworker/R-edac-poller
 49 ?            00:00:00 watchdogd
 50 ?            00:00:01 kworker/0:1H-kblockd
```

2. Search for the nginx process

Command:

```
ps aux | grep nginx
```

Explanation: Filters running processes to display only nginx. Useful to confirm if a service is active.

 Screenshot here

```
ec2-user@ip-172-31-43-35:~$ ps aux | grep nginx
root      1938  0.0  0.2 16688 2664 ?        Ss   Aug19   0:00 nginx: master process /usr/sbin/nginx
nginx     1939  0.0  0.4 17192 4176 ?        S    Aug19   0:00 nginx: worker process
nginx     1940  0.0  0.4 17192 4164 ?        S    Aug19   0:00 nginx: worker process
ec2-user  373925  0.0  0.2 222316 2356 pts/0    S+   12:19   0:00 grep --color=auto nginx
ec2-user@ip-172-31-43-35:~$
```

3. View process hierarchy

Command:

`pstree`

Explanation: Displays processes in a tree structure. Useful for understanding parent-child relationships.

Screenshot here

```
ec2-user@ip-172-31-43-35:~$ pstree
systemd--2*[agetty]
--amazon-ssm-agent--8*[amazon-ssm-agent]
--atd
--auditd--{auditd}
--chronyd
--dbus-broker-lau--dbus-broker
--gssproxy--5*[gssproxy]
--irqbalance--{irqbalance}
--lsmd
--nginx--2*[nginx]
--sshd--sshd--sshd--bash--pstree
--systemd--(sd-pam)
--systemd-homed
--systemd-inhibit--acpid
--systemd-journal
--systemd-logind
--systemd-network
--systemd-resolve
--systemd-udev
--systemd-userdbd--3*[systemd-userwor]
[ec2-user@ip-172-31-43-35 ~]$
```

4. Terminate a process

Command:


```
kill <PID>
```

(or)

```
kill -9 <PID>
```

Explanation: Ends a process by its PID. Useful for stopping stuck or unwanted applications.

Note: `kill <PID>` terminates a process by its ID. It's useful for stopping unresponsive or unwanted processes. In this case, I killed a harmless `sleep` process for demonstration.

 Screenshot here

```
ec2-user@ip-172-31-43-35:~$ ps -e | grep sleep
373986 ?        00:00:00 kworker/u8:0
373987 ?        00:00:00 systemd-userwor
373988 ?        00:00:00 systemd-userwor
373989 ?        00:00:00 systemd-userwor
374047 pts/0    00:00:00 ps
[ec2-user@ip-172-31-43-35 ~]$ sleep 300 &
[1] 374105
[ec2-user@ip-172-31-43-35 ~]$ ps -e | grep sleep
374105 pts/0    00:00:00 sleep
[ec2-user@ip-172-31-43-35 ~]$ ps -e | grep sleep
[1]+  Done                  sleep 300
[ec2-user@ip-172-31-43-35 ~]$ kill -9 374200
-bash: kill: (374200) - No such process
[ec2-user@ip-172-31-43-35 ~]$ ps -e | grep sleep
[ec2-user@ip-172-31-43-35 ~]$
```

Task 3 – System Information

1. Display full system information

Command:

```
uname -a
```

Explanation: Shows system kernel, version, and architecture. Useful for identifying system details.

 *Screenshot here*

```
ec2-user@ip-172-31-43-35:~$ uname -a
Linux ip-172-31-43-35.eu-north-1.compute.internal 6.12.40-63.114.amzn2023.x86_64 #1 SMP PREEMPT_DYNAMIC Thu Aug 7 19:30:51 UTC 2025 x86_64 x86_64 GNU/Linux
ec2-user@ip-172-31-43-35:~$
```

2. Check system uptime

Command: `uptime`

Explanation: Displays how long the system has been running. Useful for monitoring stability.

 *Screenshot here*

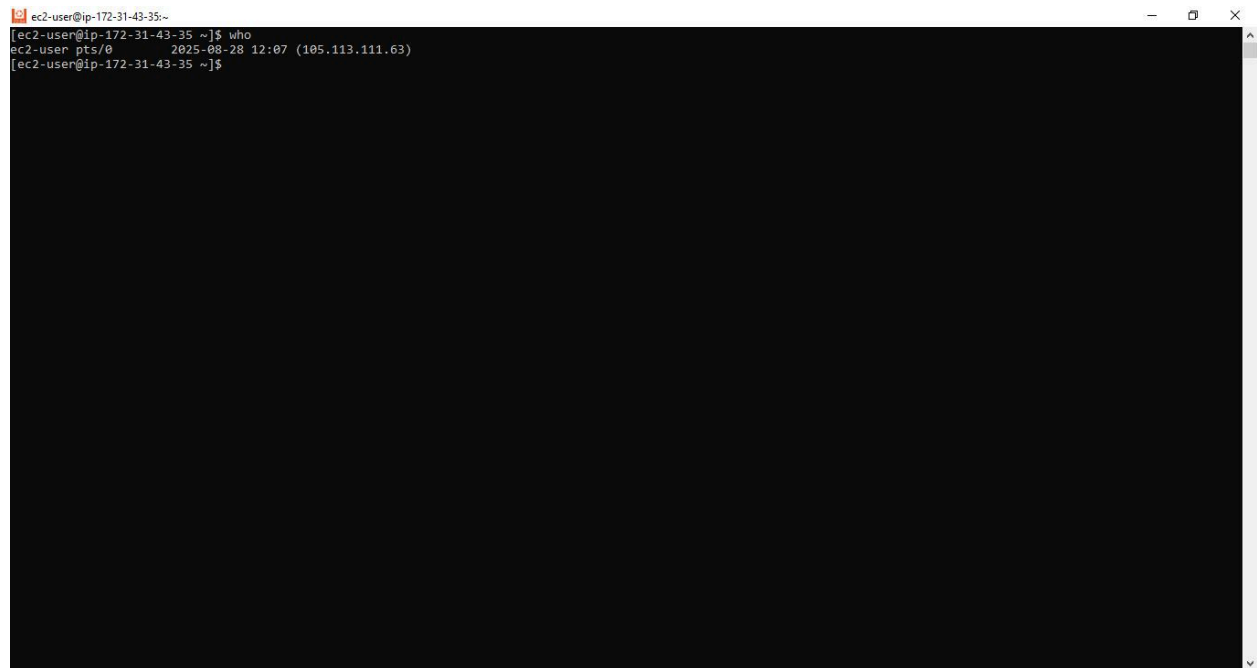
```
ec2-user@ip-172-31-43-35:~$ uptime
12:38:07 up 8 days, 13:06, 1 user, load average: 0.00, 0.00, 0.00
ec2-user@ip-172-31-43-35:~$
```

3. Display currently logged-in users

Command: `who`

Explanation: Shows logged-in users and their sessions. Useful for monitoring user access.

 *Screenshot here*



```
ec2-user@ip-172-31-43-35:~$ who
ec2-user pts/0    2025-08-28 12:07 (105.113.111.63)
ec2-user@ip-172-31-43-35:~$
```

4. Check memory usage

Command: `free -h`

Explanation: Displays available and used memory in human-readable format. Helps monitor RAM usage.

 *Screenshot here*

```
ec2-user@ip-172-31-43-35:~$ free -h
[ec2-user@ip-172-31-43-35 ~]$ free -h
              total        used        free      shared  buff/cache   available
Mem:           911Mi       173Mi       380Mi         2.0Mi        357Mi        607Mi
Swap:              0B              0B              0B
```

5. Check disk usage

Command: `df -h`

Explanation: Displays storage usage of all mounted filesystems. Useful for monitoring disk capacity.


 Screenshot here

```
ec2-user@ip-172-31-43-35:~$ free -h
[ec2-user@ip-172-31-43-35 ~]$ free -h
             total        used        free      shared  buff/cache   available
Mem:           911Mi       173Mi       380Mi        2.0Mi       357Mi       607Mi
Swap:              0B              0B              0B
[ec2-user@ip-172-31-43-35 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs         4.0M   0   4.0M   0% /dev
tmpfs            456M   0   456M   0% /dev/shm
tmpfs           183M  432K   182M   1% /run
efivarfs        128K   3.2K   120K   3% /sys/firmware/efi/efivars
/dev/nvme0n1p1   8.0G   3.3G   4.8G  41% /
tmpfs            456M   1.7M   455M   1% /tmp
/dev/nvme0n1p128 10M   1.3M   8.7M  13% /boot/efi
tmpfs            92M    0    92M   0% /run/user/1000
[ec2-user@ip-172-31-43-35 ~]$
```

6. Check directory size usage in /var

Command: `sudo du -sh /var/*`

Explanation: Shows disk usage of subdirectories in `/var`. Useful for finding large files/directories.

 Screenshot here

```
ec2-user@ip-172-31-43-35:~$ sudo du -sh /var/*
0      /var/account
0      /var/adm
104M    /var/cache
0      /var/db
0      /var/empty
0      /var/ftp
0      /var/games
0      /var/kerberos
38M     /var/lib
0      /var/local
0      /var/lock
211M    /var/log
0      /var/mail
0      /var/nis
0      /var/opt
0      /var/preserve
0      /var/run
0      /var/spool
16K     /var/tmp
350M    /var/www
0      /var/yp
[ec2-user@ip-172-31-43-35 ~]$
```