

Network & System Tools + Compression & Archiving + Text Process Tools Assignment

Assignment Objective

You will simulate a real-world Linux system administrator or DevOps engineer role by:

- Setting up user and group permissions
- Running real-time network & system diagnostics
- Performing data compression and decompression
- Using powerful text processing tools like grep and awk in combination.

You must capture and document command outputs as if working in a production or staging server environment.

Industry Scenario

You are a junior system administrator in a company called CloudOps Ltd.. Your manager has asked you to:

1. Set up proper user and group permissions for the network team.
2. Run network and system diagnostics to check connectivity and performance.
3. Archive and compress log files for backup.
4. Use grep and awk to extract meaningful data from logs.

Task 1.1 — Create users & groups

Command:

```
sudo groupadd network_team
```

Explanation: Creates a new group called network_team.

Command:

```
sudo useradd -m -G network_team alice
```

Explanation: Creates user alice and adds her to network_team.

Command:

```
sudo useradd -m -G network_team bob
```

Explanation: Creates user bob and adds him to network_team.

SCREENSHOT:

```
Last login: Sat Jun 14 07:25:18 2025 from 202.47.53.138
[ec2-user@ip-172-31-11-3 ~]$ sudo groupadd network_team
[ec2-user@ip-172-31-11-3 ~]$ sudo useradd -m -G network_team alice
[ec2-user@ip-172-31-11-3 ~]$ sudo useradd -m -G network_team bob
[ec2-user@ip-172-31-11-3 ~]$ ls
```

Task 1.2 — Set directory permissions

Command:

`sudo mkdir /opt/network_data`

Explanation: Creates a shared directory for network data.

Command:

`sudo chown root:network_team /opt/network_data`

Explanation: Changes group ownership to network_team.

Command:

`sudo chmod 770 /opt/network_data`

Explanation: Gives read/write/execute permissions to the group.

SCREENSHOT:

```
[ec2-user@ip-172-31-11-3 ~]$ sudo mkdir /opt/network_data
[ec2-user@ip-172-31-11-3 ~]$ sudo chown root:network_team /opt/network_data
[ec2-user@ip-172-31-11-3 ~]$ sudo chmod 770 /opt/network_data
[ec2-user@ip-172-31-11-3 ~]$ ls -ld /opt/network_data
lrwxrwx---. 2 root network_team 6 Jun 18 08:56 /opt/network_data
[ec2-user@ip-172-31-11-3 ~]$
```

Task 2.1 — Check connectivity to google.com

Command:

`ping -c 4 google.com`

Explanation: Checks network connectivity to google.com.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ ping -c 4 google.com
PING google.com (142.250.183.78) 56(84) bytes of data.
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=1 ttl=114 time=1.82 ms
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=2 ttl=114 time=2.24 ms
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=3 ttl=114 time=1.86 ms
64 bytes from bom12s12-in-f14.1e100.net (142.250.183.78): icmp_seq=4 ttl=114 time=2.28 ms

--- google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3005ms
rtt min/avg/max/mdev = 1.815/2.048/2.277/0.210 ms
```

Command:

`traceroute google.com`

Explanation: Shows the route packets take to google.com.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ traceroute google.com
traceroute to google.com (142.250.183.78), 30 hops max, 60 byte packets
 1 240.2.196.15 (240.2.196.15) 2.642 ms 1.702 ms 240.2.196.12 (240.2.196.12) 2.602 ms
 2 * * *
 3 99.82.178.53 (99.82.178.53) 1.603 ms * 2.485 ms
 4 * * *
 5 142.250.238.198 (142.250.238.198) 3.074 ms 142.250.227.74 (142.250.227.74) 3.986 ms 142.250.238.196 (142.250.238.196) 2.375 ms
 6 108.170.238.199 (108.170.238.199) 2.358 ms 192.178.110.248 (192.178.110.248) 2.460 ms 142.250.226.66 (142.250.226.66) 3.382 ms
 7 bom12s12-in-f14.1e100.net (142.250.183.78) 1.663 ms 2.213 ms 2.178 ms
[ec2-user@ip-172-31-11-3 ~]$
```

Command:

`mtr --report google.com`

Explanation: Combines ping and traceroute to report network path and performance.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ mtr --report google.com
Start: 2025-06-18T09:04:48+0000
HOST: ip-172-31-11-3.ap-south-1.c Loss% Snt Last Avg Best Wrst StDev
 1. |-- 240.2.196.12 0.0% 10 1.9 2.3 1.9 2.7 0.2
 2. |-- ??? 100.0 10 0.0 0.0 0.0 0.0 0.0
 3. |-- 99.82.178.53 0.0% 10 2.4 2.0 1.5 2.4 0.2
 4. |-- 142.251.225.9 0.0% 10 4.2 3.5 2.9 4.2 0.4
 5. |-- 142.251.77.101 0.0% 10 1.5 1.5 1.2 1.9 0.2
 6. |-- pnbomb-bo-in-f14.1e100.ne 0.0% 10 1.9 2.1 1.8 2.9 0.3
[ec2-user@ip-172-31-11-3 ~]$
```

Task 2.2 — Check open ports & listening services

Command:

`sudo netstat -tuln`

Explanation: Displays open ports and services using netstat.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ sudo netstat -tuln
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
tcp6       0      0 :::22                  :::*                     LISTEN
udp        0      0 172.31.11.3:68         0.0.0.0:*               *
udp        0      0 127.0.0.1:323          0.0.0.0:*               *
udp6       0      0 fe80::827:beff:fe82:546 :::*                     *
udp6       0      0 :::1:323               :::*                     *
```

Command:

`sudo ss -tulwn`

Explanation: Lists open ports and services using ss (faster alternative to netstat).

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ sudo ss -tulwn
Netid  State  Recv-Q  Send-Q           Local Address:Port      Peer Address:Port
icmp6  UNCONN 0        0                *%enX0:58               *:
udp    UNCONN 0        0                172.31.11.3%enX0:68      0.0.0.0:*
udp    UNCONN 0        0                127.0.0.1:323           0.0.0.0:*
udp    UNCONN 0        0                [fe80::827:beff:fe82:7ec5]%enX0:546 [::]:*
udp    UNCONN 0        0                [::1]:323               [::]:*
tcp    LISTEN 0        128             0.0.0.0:22              0.0.0.0:*
tcp    LISTEN 0        128             [::]:22                 [::]:*
```

Task 2.3 — Test remote port connectivity

Command:

`telnet google.com 443`

Explanation: Tests if port 443 is open using telnet.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ telnet google.com 443
Trying 142.250.67.238...
Connected to google.com.
Escape character is '^]'.
Connection closed by foreign host.
[ec2-user@ip-172-31-11-3 ~]$
```

*Command:**`nc -zv google.com 443`*

Explanation: Checks connectivity to port 443 using netcat.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ nc -zv google.com 443
Ncat: Version 7.93 ( https://nmap.org/ncat )
Ncat: Connected to 142.250.67.238:443.
Ncat: 0 bytes sent, 0 bytes received in 0.02 seconds.
[ec2-user@ip-172-31-11-3 ~]$
```

Task 2.4 — Check network interfaces*Command:**`ifconfig`*

Explanation: Displays active network interfaces.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ ifconfig
enx0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9001
    inet 172.31.11.3 netmask 255.255.240.0 broadcast 172.31.15.255
    inet6 fe80::827:beff:fe82:7ec5 prefixlen 64 scopeid 0x20<link>
    ether 0a:27:be:82:7e:c5 txqueuelen 1000 (Ethernet)
    RX packets 5224 bytes 3983217 (3.7 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 3155 bytes 333875 (326.0 KiB)
    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 24 bytes 2040 (1.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 24 bytes 2040 (1.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```


Task 2.5 — DNS lookup

Command:

nslookup google.com

Explanation: Performs DNS query using nslookup.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ nslookup google.com
Server:          172.31.0.2
Address:         172.31.0.2#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.207.174
Name:   google.com
Address: 2404:6800:4009:807::200e

[ec2-user@ip-172-31-11-3 ~]$
```

Command:

dig google.com

Explanation: Performs DNS query using dig for more detail.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ dig google.com

;<>> DiG 9.18.33 <>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47352
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:;; udp: 4096
;; QUESTION SECTION:
;google.com.                IN      A
;; ANSWER SECTION:
google.com.                207     IN      A      142.250.207.174

;; Query time: 0 msec
;; SERVER: 172.31.0.2#53(172.31.0.2) (UDP)
;; WHEN: Wed Jun 18 09:20:54 UTC 2025
;; MSG SIZE rcvd: 55
```

Task 2.6 — Download test file

Command:

```
wget https://example.com/testfile.txt
```

Explanation: Downloads a file using wget.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ wget https://example.com/testfile.txt
--2025-06-18 09:27:47-- https://example.com/testfile.txt
Resolving example.com (example.com)... 23.192.228.84, 23.215.0.136, 23.215.0.138, ...
Connecting to example.com (example.com)|23.192.228.84|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2025-06-18 09:27:48 ERROR 404: Not Found.
```

Command:

```
curl -O https://example.com/testfile.txt
```

Explanation: Downloads a file using curl.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ curl -O https://example.com/testfile.txt
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Dload  Upload  Total   Spent    Left     Speed
100 1256  100 1256    0    0  1089      0  0:00:01  0:00:01 --:--:-- 1090
[ec2-user@ip-172-31-11-3 ~]$ |
```

Task 2.7 — Monitor bandwidth in real time

Command:

```
sudo iftop -i eth0
```

Explanation: Monitors real-time bandwidth usage using iftop.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ sudo iftop -i eth0
interface: eth0
Error getting hardware address for interface: eth0
ioctl(SIOCGIFHWADDR): No such device
Unable to get IP address for interface: eth0
ioctl(SIOCGIFADDR): No such device
pcap_open_live(eth0): eth0: No such device exists (SIOCGIFHWADDR: No such device)
[ec2-user@ip-172-31-11-3 ~]$ |
```

Task 3 — Compression & Decompression

Command:

tar cvf network_data.tar /opt/network_data

Explanation: Creates a tar archive of the directory.

Command:

gzip network_data.tar

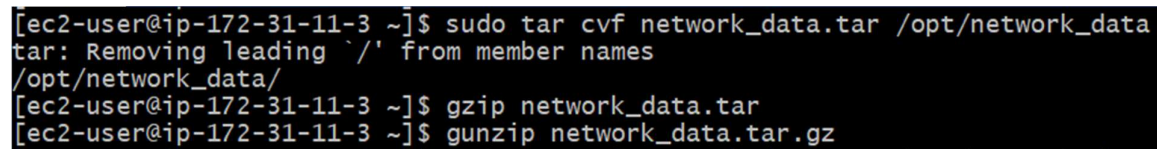
Explanation: Compresses the tar archive using gzip.

Command:

gunzip network_data.tar.gz

Explanation: Decompresses the gzip file.

Screenshot/Output:



```
[ec2-user@ip-172-31-11-3 ~]$ sudo tar cvf network_data.tar /opt/network_data
tar: Removing leading '/' from member names
/opt/network_data/
[ec2-user@ip-172-31-11-3 ~]$ gzip network_data.tar
[ec2-user@ip-172-31-11-3 ~]$ gunzip network_data.tar.gz
```

Command:

bzip2 network_data.tar

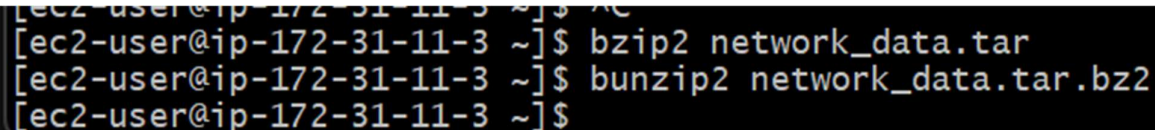
Explanation: Compresses the archive using bzip2.

Command:

bunzip2 network_data.tar.bz2

Explanation: Decompresses the bzip2 archive.

Screenshot/Output:



```
[ec2-user@ip-172-31-11-3 ~]$ bzip2 network_data.tar
[ec2-user@ip-172-31-11-3 ~]$ bunzip2 network_data.tar.bz2
[ec2-user@ip-172-31-11-3 ~]$
```


Command:**`ls -lh`**

Explanation: Lists files with human-readable sizes.

Screenshot/Output:

```
[ec2-user@ip-172-31-11-3 ~]$ ls -lh
total 36K
-rw-r--r--. 1 root      root           45 May 10 06:52 archive.tar.gz
drwxr-xr-x. 6 ec2-user ec2-user       167 Jun 14 07:25 django-todo
-rwxr-xr-x. 1 ec2-user ec2-user       137 May 17 07:22 login.sh
-rwxr-xr-x. 1 ec2-user ec2-user       135 May 17 06:52 myscript.sh
-rw-r--r--. 1 ec2-user ec2-user      10K Jun 18 09:39 network_data.tar
drwxrwx---. 2 john     developers    24 May 16 12:16 projectB
drwxr-xr-x. 3 ec2-user ec2-user        22 May 16 12:11 projects
-rwxr-xr-x. 1 ec2-user ec2-user       118 May 17 07:16 sum.sh
-rw-r--r--. 1 ec2-user ec2-user       258 May 10 07:22 syslog.txt
-rw-r--r--. 1 ec2-user ec2-user      1.3K Jun 18 09:28 testfile.txt
[ec2-user@ip-172-31-11-3 ~]$
```

Task 4 — Text Processing with grep & awk

Command:**`grep "error" /var/log/syslog`**

Explanation: Searches for 'error' messages in syslog.

Command:**`grep -c "error" /var/log/syslog`**

Explanation: Counts number of 'error' entries in syslog.

Command:**`grep "error" /var/log/syslog | awk '{print $1, $2, $3, $5}'`**

Explanation: Extracts timestamp and source field from error lines.

Command:**`grep "error" /var/log/syslog | awk '{print $5}' | sort | uniq -c | sort -nr`**

Explanation: Summarizes error sources by frequency.

SCREENSHOT:

```
[ec2-user@ip-172-31-11-3 ~]$ ls /var/log
README  bttmp-20250614  dnf.librepo.log  hawkey.log-20250614  sa
amazon  chrony          dnf.log          journal              sssd
audit   cloud-init-output.log  dnf.rpm.log     lastlog              tallylog
bttmp   cloud-init.log   hawkey.log       private              wtmp
[ec2-user@ip-172-31-11-3 ~]$ ^C
[ec2-user@ip-172-31-11-3 ~]$ grep "error" /var/log/cloud-init.log
[ec2-user@ip-172-31-11-3 ~]$ grep -c "error" /var/log/cloud-init.log
0
[ec2-user@ip-172-31-11-3 ~]$ grep "error" /var/log/cloud-init.log | awk '{print $1, $2, $3, $4}'
[ec2-user@ip-172-31-11-3 ~]$ grep "error" /var/log/cloud-init.log | awk '{print $4}' | sort
| uniq -c | sort -nr
[ec2-user@ip-172-31-11-3 ~]$
```