

Linux Assignment Report

Student Name: **Saima Usman**

Date: **01-Jan-2026**

Instructor: **Rajan Chettri - HeroVired**

Course Name: **PPMCAD-15**

What is included in Final Report

Introduction

- Objective of the assignment
- Role as Fresher DevOps Engineer

Task 1 – System Monitoring

- Tools used (`htop`, `df`, `du`)
- Logging setup
- Screenshots

Task 2 – User Management

- User creation
- Directory isolation
- Password policy enforcement

Task 3 – Backup Automation

- Backup scripts

- Cron scheduling
- Verification logs

Challenges & Learnings

- Permission issues
- Cron testing
- Importance of backups
- Backup scripts
- Cron scheduling
- Verification logs

Objective

The goal of this assignment was to set up a secure, monitored, and well-maintained development environment for two developers, **Sarah** and **Mike**. Key tasks included:

1. System Monitoring Setup
2. User Management and Access Control
3. Automated Backup Configuration for Apache and Nginx

(All steps were implemented on an Ubuntu server)

Role as a Fresher DevOps Engineer

As a Fresher DevOps Engineer, my role involved assisting in setting up a secure and reliable development environment by implementing system monitoring, user management, and automated backup solutions. I was responsible for configuring access controls, enforcing security policies, and ensuring system health visibility. I also automated backup processes using shell scripts and cron jobs to ensure data integrity and recovery. Through this assignment, I gained hands-on experience in Linux administration, automation, and DevOps best practices.

Task 1: System Monitoring Setup

Objective: Configure monitoring tools to track **system health**, **CPU**, **memory**, and **disk usage**.

Steps Taken:

Step 1: Server Upgrade

Upgrading the Server is a good practice to start. Before doing anything, always upgrade/update the server first.

```
sudo apt upgrade -y
```

```
Get:48 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 libfwupd2 am
d64 1.9.31-0ubuntu1~24.04.1 [136 kB]
Get:49 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 libmbim-prox
y amd64 1.31.2-0ubuntu3.1 [6,172 B]
Get:50 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 libmbim-glib
4 amd64 1.31.2-0ubuntu3.1 [233 kB]
Get:51 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 fwupd amd64
1.9.31-0ubuntu1~24.04.1 [4,592 kB]
Get:52 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 libpackageki
t-glib2-18 amd64 1.2.8-2ubuntu1.4 [120 kB]
Get:53 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 gir1.2-packa
gekitglib-1.0 amd64 1.2.8-2ubuntu1.4 [25.6 kB]
Get:54 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 landscape-co
mmon amd64 24.02-0ubuntu5.7 [93.8 kB]
Get:55 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 libmbim-util
s amd64 1.31.2-0ubuntu3.1 [71.6 kB]
Get:56 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 linux-module
s-6.8.0-90-generic amd64 6.8.0-90.91 [39.4 MB]
Get:57 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 linux-image-
6.8.0-90-generic amd64 6.8.0-90.91 [14.8 MB]
Get:58 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 wireless-reg
db all 2025.07.10-0ubuntu1~24.04.1 [7,502 B]
Get:59 http://ae.archive.ubuntu.com/ubuntu noble-updates/main amd64 linux-module
s-extra-6.8.0-90-generic amd64 6.8.0-90.91 [113 MB]
45% [59 linux-modules-extra-6.8.0-90-generic 18.2 MB/113 MB 16%]
```

Step 2: Install **htop** for interactive monitoring:

```
sudo apt update
```

```
Setting up libsystemd-shared:amd64 (255.4-1ubuntu8.12) ...
Setting up dhcpcd-base (1:10.0.6-1ubuntu3.2) ...
Setting up gir1.2-glib-2.0:amd64 (2.80.0-6ubuntu3.5) ...
Setting up usbmuxd (1.1.1-5~exp3ubuntu2.1) ...
usbmuxd.service is a disabled or a static unit not running, not starting it.
Setting up sosreport (4.9.2-0ubuntu0~24.04.1) ...
Installing new version of config file /etc/sos/sos.conf ...
Setting up python3-urllib3 (2.0.7-1ubuntu0.3) ...
Setting up libfdisk1:amd64 (2.39.3-9ubuntu6.4) ...
Setting up python-apt-common (2.7.7ubuntu5.1) ...
Setting up mount (2.39.3-9ubuntu6.4) ...
Setting up linux-headers-6.8.0-90 (6.8.0-90.91) ...
Setting up uuid-runtime (2.39.3-9ubuntu6.4) ...
uuid.service is a disabled or a static unit not running, not starting it.
Setting up libdrm-common (2.4.122-1~ubuntu0.24.04.2) ...
Setting up linux-tools-common (6.8.0-90.91) ...
Setting up libbim-glib4:amd64 (1.31.2-0ubuntu3.1) ...
Setting up python3-apt (2.7.7ubuntu5.1) ...
Setting up linux-headers-6.8.0-90-generic (6.8.0-90.91) ...
Setting up libfwupd2:amd64 (1.9.31-0ubuntu1~24.04.1) ...
Setting up libglib2.0-bin (2.80.0-6ubuntu3.5) ...
Setting up libpackagekit-glib2-18:amd64 (1.2.8-2ubuntu1.4) ...
Setting up systemd (255.4-1ubuntu8.12) ...
```

```
Progress: [ 77%] [#####.....]
```

```
s [378 kB]
Get:8 http://ae.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Compone
nts [940 B]
Get:9 http://ae.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components
[7,308 B]
Get:10 http://ae.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Comp
onents [216 B]
Get:11 http://ae.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Compon
ents [10.5 kB]
Get:12 http://ae.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Comp
onents [212 B]
Get:13 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [2
1.5 kB]
Get:14 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Compone
nts [212 B]
Get:15 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Component
s [71.5 kB]
Get:16 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Compone
nts [208 B]
Fetched 1,044 kB in 14s (74.5 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
All packages are up to date.
sam@sambuntusrv:~$
```

`sudo apt install htop -y`

`htop`

```
sam@sambuntusrv:~$ sudo apt install htop -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
htop is already the newest version (3.3.0-4build1).
htop set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 70 not upgraded.
sam@sambuntusrv:~$
```

What to observe:

- CPU usage per core
- Memory & swap usage
- Running processes
- Sorting by CPU or memory (F6)

`Exit with q.`

```

CPU[          0.0%] Tasks: 20, 22 thr, 74 kthr; 1 running
Mem[|||||] 179M/1.92G Load average: 0.00 0.07 0.13
Swp[          0K/1.87G] Uptime: 00:13:16

Main I/O
PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%  TIME+  Command
1 root        20    0 22050 13184  9472 S   0.0  0.7  0:02.69 /sbin/init
295 root       19   -1 66836 17124 15972 S   0.0  0.8  0:00.33 /usr/lib/syst
345 root       RT    0 282M 27392  8704 S   0.0  1.4  0:00.05 /sbin/multipa
356 root       20    0 29058  7680  4992 S   0.0  0.4  0:00.23 /usr/lib/syst
362 root       20    0 282M 27392  8704 S   0.0  1.4  0:00.00 /sbin/multipa
364 root       RT    0 282M 27392  8704 S   0.0  1.4  0:00.00 /sbin/multipa
365 root       RT    0 282M 27392  8704 S   0.0  1.4  0:00.00 /sbin/multipa
366 root       RT    0 282M 27392  8704 S   0.0  1.4  0:00.00 /sbin/multipa
367 root       RT    0 282M 27392  8704 S   0.0  1.4  0:00.10 /sbin/multipa
368 root       RT    0 282M 27392  8704 S   0.0  1.4  0:00.00 /sbin/multipa
531 systemd-ne 20    0 19004  9472  8320 S   0.0  0.5  0:00.13 /usr/lib/syst
539 systemd-re 20    0 21588 12928 10624 S   0.0  0.6  0:00.24 /usr/lib/syst
545 systemd-ti 20    0 91024  7808  6912 S   0.0  0.4  0:00.09 /usr/lib/syst
579 systemd-ti 20    0 91024  7808  6912 S   0.0  0.4  0:00.00 /usr/lib/syst
652 messagebus 20    0  9788  5376  4608 S   0.0  0.3  0:00.16 @dbus-daemon
659 polkitd    20    0 300M  7936  7040 S   0.0  0.4  0:00.08 /usr/lib/polkitd
672 root       20    0 17995  8832  7808 S   0.0  0.4  0:00.13 /usr/lib/syst

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

```

```

CPU[          0.0%] Tasks: 20, 22 thr, 75 kthr; 1 running
Mem[|||||] 179M/1.92G Load average: 0.00 0.04 0.10
Swp[          0K/1.87G] Uptime: 00:15:40

Main I/O
TIME+ Command
:02.69 /sbin/init
:00.33 /usr/lib/systemd/systemd-journald
:00.05 /sbin/multipathd -d -s
:00.23 /usr/lib/systemd/systemd-udev
:00.00 /sbin/multipathd -d -s
:00.00 /sbin/multipathd -d -s
:00.00 /sbin/multipathd -d -s
:00.00 /sbin/multipathd -d -s
:00.11 /sbin/multipathd -d -s
:00.00 /sbin/multipathd -d -s
:00.13 /usr/lib/systemd/systemd-networkd
:00.24 /usr/lib/systemd/systemd-resolved
:00.10 /usr/lib/systemd/systemd-timesyncd
:00.00 /usr/lib/systemd/systemd-timesyncd
:00.16 @dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-a
:00.08 /usr/lib/polkit-1/polkitd --no-debug
:00.13 /usr/lib/systemd/systemd-logind

F1Help F2Setup F3Search F4Filter F5Tree F6SortBy F7Nice -F8Nice +F9Kill F10Quit

```

htop shows CPU/memory usage per process.

Step 3. Disk Usage Monitoring

1. Checked disk usage *with df* (filesystem level)

df -h

```
sam@sambuntusrv:~$ df -h
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     197M        1.1M   196M    1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 9.8G       5.0G       4.4G   54% /
tmpfs                     985M          0    985M    0% /dev/shm
tmpfs                     5.0M          0    5.0M    0% /run/lock
/dev/sda2                 1.8G       197M       1.5G   12% /boot
tmpfs                     197M        12K    197M    1% /run/user/1000
sam@sambuntusrv:~$ _
```

2. Checked directory size with *du*

*sudo du -sh /var/**

```
sam@sambuntusrv:~$ sudo du -sh /var/*
728K    /var/backups
149M    /var/cache
4.0K    /var/crash
266M    /var/lib
4.0K    /var/local
0       /var/lock
203M    /var/log
4.0K    /var/mail
4.0K    /var/opt
0       /var/run
4.0K    /var/snap
20K     /var/spool
60K     /var/tmp
12K     /var/www
sam@sambuntusrv:~$ _
```

df → overall disk usage (file-level usage)

du → directory-level usage

Step 4. Identified resource-intensive processes:

```
ps aux --sort=-%cpu | head
```

```
sam@sambuntusrv:~$ ps aux --sort=-%cpu | head
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         56  1.7  0.0      0     0 ?        I    08:50    0:22 [kworker/0:3-
events]
root       143  0.2  0.0      0     0 ?        I    08:51    0:03 [kworker/0:5-
cgroup_destroy]
root         1  0.2  0.6  22060  13184 ?        Ss   08:50    0:02 /sbin/init
root        27  0.0  0.0      0     0 ?        I    08:50    0:00 [kworker/u2:2
-events_unbound]
root        16  0.0  0.0      0     0 ?        S    08:50    0:00 [ksoftirqd/0]
root       295  0.0  0.8  66836  17124 ?        S<S  08:51    0:00 /usr/lib/syst
emd/systemd-journald
root        43  0.0  0.0      0     0 ?        I<   08:50    0:00 [kworker/0:1H
-kblockd]
root        25  0.0  0.0      0     0 ?        I    08:50    0:00 [kworker/u2:1
-events_power_efficient]
root       705  0.0  1.1 109640  22912 ?        Ss   08:51    0:00 /usr/bin/pyth
on3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
sam@sambuntusrv:~$
```

```
ps aux --sort=-%mem | head
```

```
sam@sambuntusrv:~$ ps aux --sort=-%mem | head
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root    14298  0.0  1.3 223580 27264 ?        SLs  11:05    0:00 /sbin/multipa
thd -d -s
root       715  0.0  1.1 109640 22912 ?        Ss   10:50    0:00 /usr/bin/pyth
on3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root     3980  0.0  1.0 372672 20992 ?        Ss   11:03    0:00 /usr/libexec/
packagekitd
root     3753  0.0  0.7  50332  14848 ?        S<S  11:03    0:00 /usr/lib/syst
emd/systemd-journald
root         1  0.5  0.6  22656  13952 ?        Ss   10:50    0:06 /usr/lib/syst
emd/systemd --system --deserialize=63
root    14301  0.0  0.6 468976 13568 ?        Ss   11:05    0:00 /usr/libexec/
udisks2/udisksd
systemd+  4323  0.0  0.6  21588  13056 ?        Ss   11:04    0:00 /usr/lib/syst
emd/systemd-resolved
root    14328  0.0  0.6 392108 13056 ?        Ss   11:05    0:00 /usr/sbin/Mod
emManager
sam       916  0.0  0.5  20036  11008 ?        Ss   10:52    0:00 /usr/lib/syst
emd/systemd --user --deserialize=12
sam@sambuntusrv:~$ _
```

This shows top CPU & memory consumers.

Step 5: Log System Metrics (Basic Reporting)

a. Created a log directory:

```
sudo mkdir -p /var/log/system-monitor
```

b. Created a script:

```
sudo nano /usr/local/bin/system_monitor.sh
```

```
GNU nano 7.2 /usr/local/bin/system_monitor.sh
#!/bin/bash
echo "===== $(date) =====" >> /var/log/system-monitor/metrics.log
df -h >> /var/log/system-monitor/metrics.log
free -m >> /var/log/system-monitor/metrics.log
ps aux --sort=-%cpu | head >> /var/log/system-monitor/metrics.log
echo "" >> /var/log/system-monitor/metrics.log

[ Read 6 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

c. Made it executable:

```
sudo chmod +x /usr/local/bin/system_monitor.sh
```

d. Run it:

```
sudo /usr/local/bin/system_monitor.sh
```

e. Viewed log:

```
cat /var/log/system-monitor/metrics.log
```

```

sam@sambuntusrv:~$ sudo /usr/local/bin/system_monitor.sh
sam@sambuntusrv:~$ cat /usr/local/bin/system_monitor.sh
#!/bin/bash
echo "===== $(date) =====" >> /var/log/system-monitor/metrics.log
df -h >> /var/log/system-monitor/metrics.log
free -m >> /var/log/system-monitor/metrics.log
ps aux --sort=-%cpu | head >> /var/log/system-monitor/metrics.log
echo "" >> /var/log/system-monitor/metrics.log
sam@sambuntusrv:~$

```

Logged outputs for review:

```
sudo top -b -n1 | sudo tee /system_metrics.log
```

```

3747 systemd+ 20 0 19008 9728 8576 S 0.0 0.5 0:00.06 systemd+
3753 root 19 -1 50332 15232 14208 S 0.0 0.8 0:00.15 systemd+
3898 systemd+ 20 0 91024 7808 6912 S 0.0 0.4 0:00.09 systemd+
3959 root 20 0 28908 7748 5060 S 0.0 0.4 0:00.08 systemd+
3960 root -2 0 0 0 0 S 0.0 0.0 0:00.00 psimon
4323 systemd+ 20 0 21588 13056 10752 S 0.0 0.6 0:00.15 systemd+
4800 syslog 20 0 222508 5504 4608 S 0.0 0.3 0:00.04 rsyslogd
13433 root 0 -20 0 0 0 I 0.0 0.0 0:00.00 kworker+
14298 root rt 0 223580 27264 8704 S 0.0 1.4 0:00.24 multipa+
14301 root 20 0 468976 13568 11392 S 0.0 0.7 0:00.06 udisksd
14305 polkitd 20 0 308164 8064 7168 S 0.0 0.4 0:00.10 polkitd
14328 root 20 0 392108 13056 11008 S 0.0 0.6 0:00.07 ModemMa+
15263 root 20 0 6808 5004 3712 S 0.0 0.2 0:00.13 apache2
15266 www-data 20 0 754140 5532 3840 S 0.0 0.3 0:00.00 apache2
15267 www-data 20 0 754140 5532 3840 S 0.0 0.3 0:00.00 apache2
15374 www-data 20 0 3620 1544 1408 S 0.0 0.1 0:00.11 htcache+
15464 root -2 0 0 0 0 S 0.0 0.0 0:00.00 psimon
15559 root 20 0 0 0 0 I 0.0 0.0 0:01.07 kworker+
15580 root 20 0 0 0 0 I 0.0 0.0 0:00.00 kworker+
15581 root 20 0 16732 7040 5888 S 0.0 0.3 0:00.00 sudo
15582 root 20 0 16728 7040 5888 S 0.0 0.3 0:00.01 sudo
15583 root 20 0 16732 2480 1280 S 0.0 0.1 0:00.00 sudo
15585 root 20 0 16728 2476 1280 S 0.0 0.1 0:00.00 sudo
15586 root 20 0 5692 1792 1792 S 0.0 0.1 0:00.00 tee
sam@sambuntusrv:~$

```

```
df -h | sudo tee -a /system_metrics.log
```

```
sam@sambuntusrv:~$ df -h | sudo tee -a /system_metrics.log
Filesystem                Size      Used Avail Use% Mounted on
tmpfs                     197M        1.1M  196M   1% /run
/dev/mapper/ubuntu--vg-ubuntu--lv 9.8G      5.0G      4.4G  54% /
tmpfs                     985M         0   985M   0% /dev/shm
tmpfs                     5.0M         0   5.0M   0% /run/lock
/dev/sda2                 1.8G      197M      1.5G  12% /boot
tmpfs                     197M        12K   197M   1% /run/user/1000
sam@sambuntusrv:~$ _
```

Challenges and Fixes

`du` returned **Permission denied** on some directories → solved by using `sudo`.

👉 While checking directory sizes using the `du` command, permission denied errors were encountered for system directories under `/var`. This occurred because these directories are restricted to root access for security reasons. Running the command with `sudo` allowed accurate disk usage monitoring.

Logged metrics provide a snapshot for **capacity planning** and **troubleshooting**.

Screenshots provided for Review:

- ✓ `htop` interface
- ✓ Output of `df -h`
- ✓ Output of `du -sh /var/*`

--- ***** TASK-01 DONE ***** ---

Task 2: User Management and Access Control

Objective: Create secure user accounts for **Sarah** and **Mike**, isolate directories, and enforce password policies.

Steps Taken:

1. Created users with secure passwords:

```
sudo adduser sarah
```

```
sudo adduser mike
```

- ✓ Passwords were set interactively; strong passwords were enforced.

```
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for sarah
Enter the new value, or press ENTER for the default
    Full Name []: Sarah
    Room Number []: 1
    Work Phone []: 123456
    Home Phone []: 123456
    Other []: 123456
Is the information correct? [Y/n] y
info: Adding new user `sarah' to supplemental / extra groups `users' ...
info: Adding user `sarah' to group `users' ...
sam@sambuntusrv:~$
```

```
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `mike' (1002) ...
info: Adding new user `mike' (1002) with group `mike (1002)' ...
info: Creating home directory `/home/mike' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
Sorry, passwords do not match.
passwd: Authentication token manipulation error
passwd: password unchanged
Try again? [y/N] y
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for mike
Enter the new value, or press ENTER for the default
    Full Name []: Mike
    Room Number []: 2
    Work Phone []: 123456
    Home Phone []: 123456
    Other []: 123456
Is the information correct? [Y/n] y
info: Adding new user `mike' to supplemental / extra groups `users' ...
info: Adding user `mike' to group `users' ...
sam@sambuntusrv:~$
```

2. Created isolated workspace directories:

```
sudo mkdir -p /home/sarah/workspace
```

```
sudo mkdir -p /home/mike/workspace
```

```
sudo chown -R sarah:sarah /home/sarah/workspace
```

```
sudo chown -R mike:mike /home/mike/workspace
```

```
sudo chmod 700 /home/sarah/workspace
```

```
sudo chmod 700 /home/mike/workspace
```

```
sam@sambuntusrv:~$ sudo mkdir -p /home/sarah/workspace
sam@sambuntusrv:~$ sudo mkdir -p /home/mike/workspace
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo chown -R sarah:sarah /home/sarah/workspace
sam@sambuntusrv:~$ sudo chown -R mike:mike /home/sarah/workspace
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo chmod 700 /home/sarah/workspace
sudo: chmod: command not found
sam@sambuntusrv:~$ sudo chmod 700 /home/sarah/workspace
sam@sambuntusrv:~$ sudo chmod 700 /home/mike/workspace
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ ls -ld /home/sarah/workspace
ls: cannot access '/home/sarah/workspace': Permission denied
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo ls -ld /home/sarah/workspace
drwx----- 2 mike mike 4096 Dec 31 10:40 /home/sarah/workspace
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo ls -ld /home/mike/workspace
drwx----- 2 root root 4096 Dec 31 10:40 /home/mike/workspace
sam@sambuntusrv:~$
sam@sambuntusrv:~$
```

👉 The **-p** option in the **mkdir** command ensures that parent directories are created automatically if they do not exist and prevents errors if the directory already exists, making the command safe for automation.

👉 The **-R** (recursive) option in the **chown** command changes ownership of the specified directory and all its subdirectories and files, ensuring consistent access permissions throughout the directory structure.


```
sam@sambuntusrv:~$ su - mike
Password:
mike@sambuntusrv:~$ ls
workspace
mike@sambuntusrv:~$ su - sam
Password:
sam@sambuntusrv:~$ su - sarah
Password:
sarah@sambuntusrv:~$ ls
workspace
sarah@sambuntusrv:~$ ls -ld
drwxr-x--- 3 sarah sarah 4096 Dec 31 10:40 .
sarah@sambuntusrv:~$ su - mike
Password:
mike@sambuntusrv:~$ ls -ld
drwxr-x--- 3 mike mike 4096 Dec 31 10:40 .
mike@sambuntusrv:~$
```

3. Enforced password policy (expiry 30 days, added complexity):

```
sudo chage -M 30 sarah
```

```
sudo chage -M 30 mike
```

4. Enforce Password Complexity

Edited PAM configuration:

```
sudo nano /etc/pam.d/common-password
```

```
password requisite pam_pwquality.so retry=3 minlen=8 ucredit=-1  
lcredit=-1 dcredit=-1
```

```
GNU nano 7.2 /etc/pam.d/common-password *  
  
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.  
# To take advantage of this, it is recommended that you configure any  
# local modules either before or after the default block, and use  
# pam-auth-update to manage selection of other modules. See  
# pam-auth-update(8) for details.  
  
# here are the per-package modules (the "Primary" block)  
password [success=1 default=ignore] pam_unix.so obscure yescrypt  
# here's the fallback if no module succeeds  
password requisite pam_deny.so  
# prime the stack with a positive return value if there isn't one already;  
# this avoids us returning an error just because nothing sets a success code  
# since the modules above will each just jump around  
password required pam_permit.so  
# and here are more per-package modules (the "Additional" block)  
# Enforce Password Complexity  
password requisite pam_pwquality.so retry=3 minlen=8 ucredit=-1 lcredit=-1 dcr>  
  
# end of pam-auth-update config  
  
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location  
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

Meaning:

- Minimum 8 characters
- Uppercase, lowercase, digit required

4. Verified password settings:

```
sudo chage -l sarah
```

```
sam@sambuntusrv:~$ sudo chage -M 30 sarah
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo chage -M 30 mike
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo chage -l sarah
Last password change                : Dec 31, 2025
Password expires                    : Jan 30, 2026
Password inactive                   : never
Account expires                    : never
Minimum number of days between password change : 0
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
sam@sambuntusrv:~$
```

👉 The `chage -l sarah` command is used to list the password aging and expiration details of the user, allowing administrators to verify that password expiration and security policies are correctly enforced.

`sudo chage -l mike`

```
sam@sambuntusrv:~$ sudo chage -l mike
Last password change                : Dec 31, 2025
Password expires                    : Jan 30, 2026
Password inactive                   : never
Account expires                    : never
Minimum number of days between password change : 0
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
sam@sambuntusrv:~$ _
```

Challenges and Fixes

- Some permission errors occurred → resolved with sudo and correct ownership.
- Isolated directories ensure data confidentiality.

Screenshots provided for Review:

✓ `ls -ld /home/sarah/workspace`

✓ `ls -ld /home/mike/workspace`

```
sam@sambuntusrv:~$  
sam@sambuntusrv:~$ sudo ls -ld /home/sarah/workspace  
drwx----- 2 mike mike 4096 Dec 31 10:40 /home/sarah/workspace  
sam@sambuntusrv:~$  
sam@sambuntusrv:~$ sudo ls -ld /home/mike/workspace  
drwx----- 2 root root 4096 Dec 31 10:40 /home/mike/workspace  
sam@sambuntusrv:~$  
sam@sambuntusrv:~$
```

✓ Output of `sudo chage -l sarah` and `sudo chage -l mike`

```
sam@sambuntusrv:~$ sudo chage -M 30 sarah
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo chage -M 30 mike
sam@sambuntusrv:~$
sam@sambuntusrv:~$
sam@sambuntusrv:~$ sudo chage -l sarah
Last password change                : Dec 31, 2025
Password expires                    : Jan 30, 2026
Password inactive                    : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
sam@sambuntusrv:~$
```

```
sam@sambuntusrv:~$ sudo chage -l mike
Last password change                : Dec 31, 2025
Password expires                    : Jan 30, 2026
Password inactive                    : never
Account expires                     : never
Minimum number of days between password change : 0
Maximum number of days between password change : 30
Number of days of warning before password expires : 7
sam@sambuntusrv:~$ _
```

--- ***** TASK-02 DONE ***** ---

Task 3: Automated Backup Configuration for Web Servers

Objective: Configure automated backups for Sarah's Apache server and Mike's Nginx server.

Step 1: Created Backup Directory

```
sudo mkdir /backups
```

```
sudo chmod 700 /backups
```

Challenges on Ubuntu

- Directories `/etc/httpd` (Apache) and `/etc/nginx` (Nginx) do not exist by default.
- Solution: Create placeholder directories and test files for backup scripts.

a. Prepared directories and test files

Backup directory

```
sudo mkdir -p /backups
```

Apache placeholders

```
sudo mkdir -p /etc/httpd /var/www/html
```

```
echo "Apache test config" | sudo tee /etc/httpd/test.conf
```

```
echo "Apache test page" | sudo tee /var/www/html/index.html
```

Nginx placeholders

```
sudo mkdir -p /etc/nginx /usr/share/nginx/html
```

```
echo "Nginx test config" | sudo tee /etc/nginx/nginx.conf
```

```
echo "Nginx test page" | sudo tee /usr/share/nginx/html/index.html
```

Log files

```
sudo touch /var/log/apache_backup.log  
/var/log/apache_backup_verify.log
```

```
sudo touch /var/log/nginx_backup.log /var/log/nginx_backup_verify.log
```

```
sam@sambuntusrv:~$ sudo mkdir /backups  
[sudo] password for sam:  
sam@sambuntusrv:~$ sudo chmod 700 /backups  
sam@sambuntusrv:~$ ls -ld  
drwxr-x--- 8 sam sam 4096 Dec 31 09:04 .  
sam@sambuntusrv:~$ ls -ld /backups  
drwx----- 2 root root 4096 Jan  1 14:55 /backups  
sam@sambuntusrv:~$
```

```
sam@sambuntusrv:~$ sudo ls -lh /backups  
total 8.0K  
-rw-r--r-- 1 root root 45 Jan  1 15:35 apache_backup_2026-01-01.tar.gz  
-rw-r--r-- 1 root root 45 Jan  1 15:44 nginx_backup_2026-01-01.tar.gz  
sam@sambuntusrv:~$ _
```

Step 2: Created backup scripts

a. Created a backup script for Apache web server for user Sarah

```
sudo nano /usr/local/bin/apache_backup.sh
```

Apache Backup Script: `/usr/local/bin/apache_backup.sh`

```
GNU nano 7.2 /usr/local/bin/apache_backup.sh
#!/bin/bash

# Variables
DATE=$(date +%F)
BACKUP_FILE="/backups/apache_backup_${DATE}.tar.gz"

# Create Backup
sudo tar -czf $BACKUP_FILE /etc/httpd /var/www/html

# Log backup success
echo "$(date) - Apache Backup Created: $BACKUP_FILE" >> /var/log/apache_backup.log

# Verify Backup
tar -tzf $BACKUP_FILE >> /var/log/backups/apache_backup_verify.log 2>&1

sam@sambuntusrv:~$
```


b. Created a backup script for the Nginx web server for user Mike

```
sudo nano /usr/local/bin/nginx_backup.sh
```

Nginx Backup Script: /usr/local/bin/nginx_backup.sh

```
GNU nano 7.2 /usr/local/bin/nginx_backup.sh *
#!/bin/bash
DATE=$(date +%F)
BACKUP_FILE="/backups/nginx_backup_${DATE}.tar.gz"
tar -czf $BACKUP_FILE /etc/nginx /usr/share/nginx/html 2>> /var/log/nginx_backup.log
tar -tzf $BACKUP_FILE >> /var/log/nginx_backup_verify.log
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line

Step 3: Made scripts executable:

```
sudo chmod +x /usr/local/bin/apache_backup.sh
```

```
sudo chmod +x /usr/local/bin/nginx_backup.sh
```

Step 4: Tested backup scripts manually

```
sudo /usr/local/bin/apache_backup.sh
```

```
sudo /usr/local/bin/nginx_backup.sh
```

```
sam@sambuntusrv:~$ sudo chmod +x /usr/local/bin/nginx_backup.sh
[sudo] password for sam:
sam@sambuntusrv:~$ sudo /usr/local/bin/nginx_backup.sh
tar: Removing leading '/' from member names
tar: Removing leading '/' from hard link targets
Thu Jan  1 05:16:09 PM UTC 2026 - Nginx Backup Created: /backups/nginx_backup_20
26-01-01.tar.gz
etc/nginx/
etc/nginx/nginx.conf
usr/share/nginx/html/
usr/share/nginx/html/index.html
sam@sambuntusrv:~$ _
```

Step 5: Checked backup files:

```
ls -lh /backups
```

```
sam@sambuntusrv:~$ sudo ls -lh /backups
total 8.0K
-rw-r--r-- 1 root root 45 Jan  1 15:35 apache_backup_2026-01-01.tar.gz
-rw-r--r-- 1 root root 45 Jan  1 15:44 nginx_backup_2026-01-01.tar.gz
sam@sambuntusrv:~$ _
```

Step 6: Checked verification logs:

```
sudo cat /var/log/apache_backup_verify.log
```

```
sudo cat /var/log/nginx_backup_verify.log
```

```
sam@sambuntusrv:~$ sudo cat /var/log/nginx_backup_verify.log
etc/nginx/
etc/nginx/nginx.conf
usr/share/nginx/html/
usr/share/nginx/html/index.html
etc/nginx/
etc/nginx/nginx.conf
usr/share/nginx/html/
usr/share/nginx/html/index.html
etc/nginx/
etc/nginx/nginx.conf
usr/share/nginx/html/
usr/share/nginx/html/index.html
etc/nginx/
etc/nginx/nginx.conf
usr/share/nginx/html/
usr/share/nginx/html/index.html
sam@sambuntusrv:~$ _
```

✓ All files are backed up correctly, and logs populated.

Step 7: Scheduled cron jobs

```
sudo crontab -e
```

```
sam@sambuntusrv:~$ sudo crontab -e
no crontab for root - using an empty one

Select an editor. To change later, run 'select-editor'.
 1. /bin/nano      <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]:
```

```
GNU nano 7.2 /tmp/crontab.iFiYxY/crontab *
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
# Schedule Jobs (Tuesday 12:00 am)
0 0 * * 2 /usr/local/bin/apache_backup.sh
0 0 * * 2 /usr/local/bin/nginx_backup.sh

[ Wrote 30 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line
```

```
# Apache backup every Tuesday at 12:00 AM
```

```
0 0 * * 2 /usr/local/bin/apache_backup.sh
```

```
# Nginx backup every Tuesday at 12:00 AM
```

```
0 0 * * 2 /usr/local/bin/nginx_backup.sh
```

Verified cron:

`sudo crontab -l`

```
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
#
# Schedule Jobs (Tuesday 12:00 am)

0 0 * * 2 /usr/local/bin/apache_backup.sh
0 0 * * 2 /usr/local/bin/nginx_backup.sh

sam@sambuntusrv:~$
```

✓ Cron ensures **automatic weekly backups**.

Notes / Challenges

tar: removing leading '/' from member names → normal warning; safe to ignore.

Initial verification logs were empty → fixed by creating placeholder files.

Directories /etc/httpd and /etc/nginx are placeholders; real servers do not need placeholders.

Verified Backup Integrity

```
tar -tzf /backups/apache_backup_YYYY-MM-DD.tar.gz
```

```
cat /var/log/apache_backup_verify.log
```

```
sam@sambuntusrv:~$ sudo tar -tzf /backups/apache_backup_2026-01-01.tar.gz
var/www/html/
var/www/html/index.html
sam@sambuntusrv:~$ _
```

Conclusion of the whole Assignment

- **System monitoring** implemented via `htop`, `df`, `du`, and logged metrics.
- **User accounts** for Sarah and Mike were created with isolated directories and password policies enforced.
- **Backup automation** is fully functional for Apache and Nginx, with verification logs and cron scheduling.
- Assignment objective achieved by: **secure, monitored, automated, and verifiable development environment.**

---- ***** END OF REPORT ***** ----