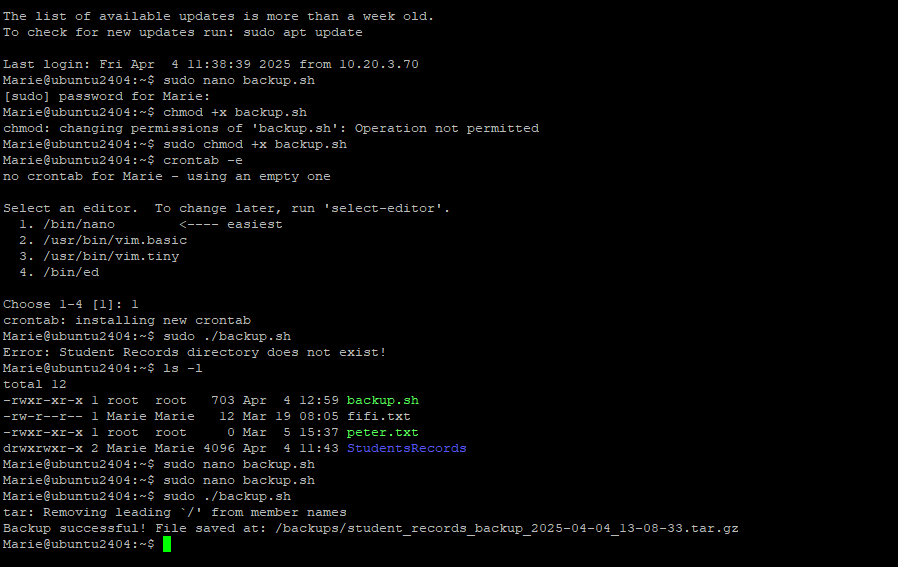
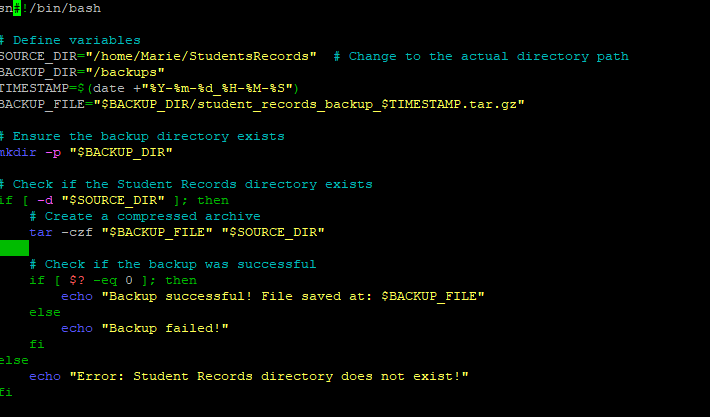
**NANKINGA MARIA – 2023-B071-10729**

**SYSTEM ADMINISTRATION ASSIGNMENT**

Youtube link: <https://youtu.be/OGJUAb6Cnbc?si=ZHJX73PU4Gx9rYc2>

This Script ;

First, Defines the source directory containing student records.

Creates a timestamped filename for version control.

Afterwards Sets the destination for backups.

Ensures the backup folder exists (creates it if not).

We check if the student records directory exists.

Compresses it into a .tar.gz file.

We also check if the tar command was successful.

Lastly, Error handling for a missing source directory.

Testing

Make sure /home/Marie/StudentsRecords exists with some test files.

Give execute permission

Run it manually

Results:

If Successful  
Backup successful! File saved at: /backups/student\_records\_backup\_YYYY-MM-DD\_HH-MM-SS.tar.gz

MONITORING SCRIPT

monitor.sh



Set The threshold to 80%’.

We uses free and awk to calculate percentage of used RAM.

We check If usage is ≥ 80%, it sends a desktop notification and logs a terminal warning.

We loop evey 10 seconds.

Testing

Run the script in one terminal

./monitor.sh

To simulate high memory usage

stress --vm 1 --vm-bytes 710M --timeout 60s

Results

When RAM exceeds 80%:

Terminal shows: WARNING: High RAM usage detected! Current usage: XX%

SCHEDULED LOGOUT



This command “wall "System maintenance: You will be logged out in 1 minute. Save your work!" will send a broadcast warning to all users.

Logs the action for auditing.

It gives users 60 seconds to save work.

Lists all logged-in users (except root) and forcefully logs them out.

Testing

Open multiple SSH sessions or user terminals (except root).

Run the script with sudo

Sudo ./logout\_users.sh

Check /var/log/logout.log for timestamped entry

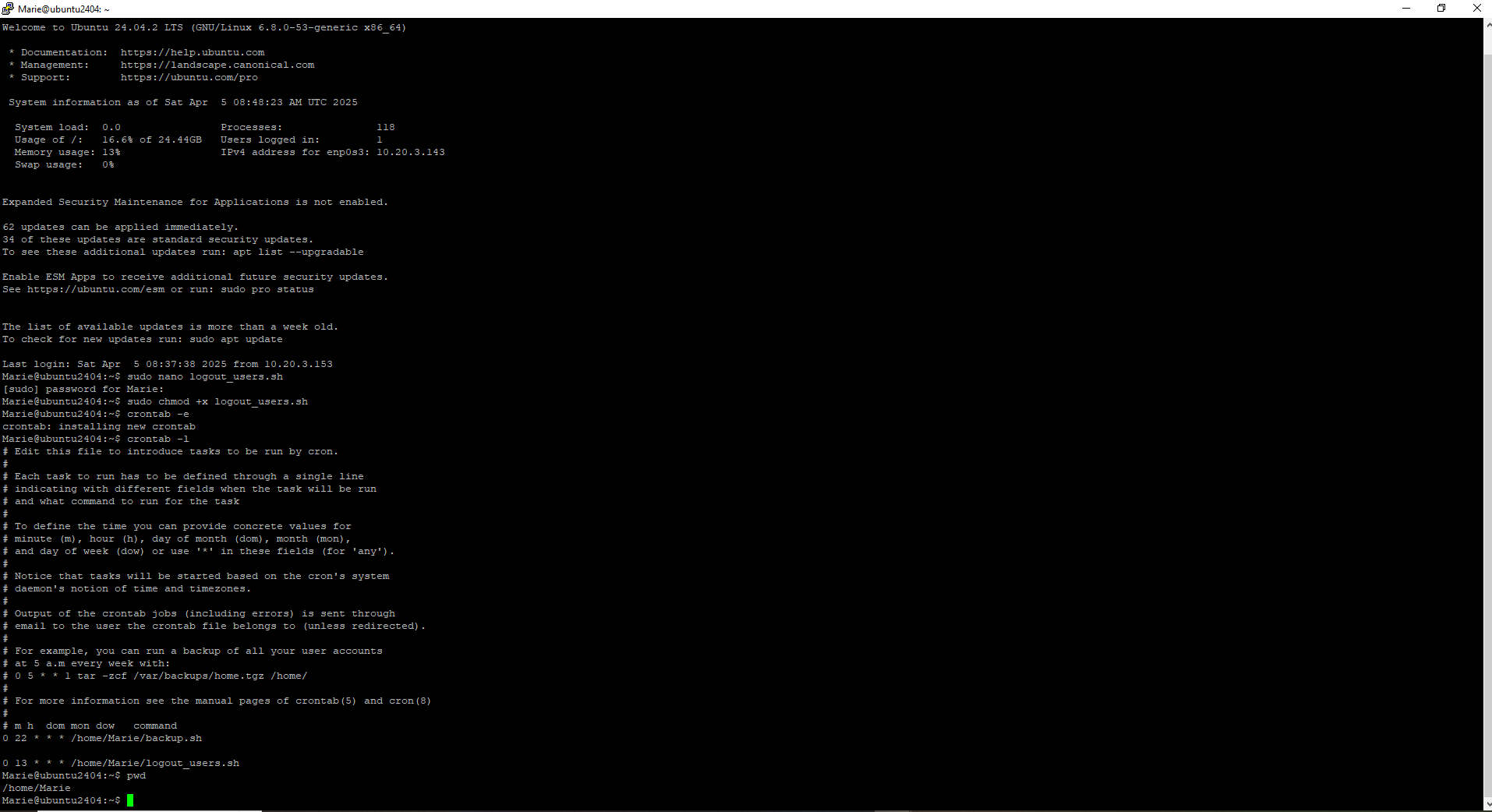
Results

Users see Broadcast message: System maintenance: You will be logged out in 1 minute. Save your work!

After 1 minute, users are logged out.  
Expected output from the logfile

Users logged out at Sat Apr 5 14:00:00 EAT 2025

CronTab for automation



SECURITY CONSIDERATION

**1. Managing Unsaved Work During Scheduled Logout**

One of the potential risks of automating user logouts is the **loss of unsaved data**. If users are logged out without proper warning or time to save their work, it can lead to data loss and disruption, especially for staff or students working on the system at the time. To mitigate this:

* The system broadcasts a **warning message** one minute before logout using wall, allowing users to save work.
* For enhanced security and user awareness, future updates could include **automated session snapshots**, **desktop notifications**, or **graceful application shutdowns**.
* Integrating **audit trails** or logs ensures administrators can trace activity and address incidents of data loss or irregular behavior.

**2. Secure Handling of Backup Files**

The student records directory contains **sensitive academic data**, and improper handling of these backups could expose the university to data breaches. To secure the backups:

* Backup archives are stored in a **restricted directory (/backups)** with proper **permissions** to ensure only administrators can access them.
* It's recommended to **encrypt** backup files using tools like gpg or openssl before storing them to prevent unauthorized access even if files are copied or stolen.
* Backups should be **transferred securely** (e.g., via scp over SSH) if stored off-site or in the cloud.
* Implementing **automated backup verification** can ensure integrity and detect corruption early.

**3. Script Security and Permissions**

Since these scripts run with elevated privileges (especially logout.sh and backup operations), it is essential to:

* Restrict access using appropriate **file permissions** (e.g., chmod 700).
* Avoid storing passwords or sensitive information in plain text within scripts.
* Log script activities in secure log files (like /var/log/logout.log) with permissions that prevent tampering.

FUTURE ENHANCEMENTS

**1. Use of Systemd Timers Instead of Cron Jobs**

While cron is simple and effective, **systemd timers** provide better integration with modern Linux systems, including:

* Improved logging and monitoring.
* Easier management of start/stop behaviors.
* Dependency handling (e.g., run after network or user login is ready).

**2. User-Friendly Notification System**

Enhancing the logout warning system with **graphical alerts** or **interactive countdown timers** can provide a better user experience. Users could be allowed to request a temporary delay or save session snapshots automatically before logout.

**3. Centralized Monitoring Dashboard**

Expanding RAM monitoring to include **CPU, disk usage, and network activity**, and displaying it on a centralized dashboard (e.g., using tools like **Grafana + Prometheus**) would offer real-time visibility into server health and performance.

**4. Automated Backup Rotation and Cleanup**

To avoid disk space exhaustion and clutter:

* Implement a **retention policy** (e.g., keep only the last 7 daily backups).
* Automate cleanup of older backups with a script or cron job.

**5. Notification and Alert Integration**

Integrate email or SMS alerts for critical events, such as:

* Backup failures.
* Excessive RAM or disk usage.
* Unsuccessful logouts or permission errors.

Tools like mail, ssmtp, or APIs like Twilio can be used for such notifications.

**6. Audit and Compliance Reporting**

Introduce logging for all automated tasks to support **audit trails**. Logs should be:

* Time-stamped.
* Stored securely and reviewed regularly.
* Used to generate reports for compliance and review.