Placement Empowerment Program

Cloud Computing and DevOps Centre

Day 10 – Temp File Cleanup and Disk Usage Tracker

Build a script that deletes temporary files older than a set number of days and logs disk usage before and after cleanup.

Name: Rogini.P

Department: ECE

Introduction

In a Linux environment, temporary files (/tmp, cache, or leftover log files) can pile up over time and consume valuable disk space. If not cleaned regularly, they may slow down the system or even lead to storage-related failures.

This Proof of Concept (PoC) demonstrates a bash script that:

Automatically deletes temporary files older than a specific number of days
Tracks and logs disk usage before and after cleanup
Helps maintain disk hygiene and improves system performance

It’s a simple yet powerful solution for system maintenance, especially useful for developers, system administrators, and DevOps teams who manage disk space in shared or production environments.

Overview

This PoC involves creating a bash script that automates the cleanup of temporary files in a specified directory **(e.g., /tmp)** based on their age and logs system disk usage before and after the cleanup process.

Key Features:

Deletes files **older than a defined number of days** (e.g., 7 days)

Logs **disk usage before and after** the cleanup using `df -h`

Saves output to a **log file** for reference or auditing

Can be optionally scheduled using **cron** for automation

Tools & Commands Used:

`bash` – scripting language

`find` – to locate and delete old files

`df -h` – to report disk usage

`chmod` – to make script executable

`cron` (optional) – for periodic execution

This script helps ensure that the system runs efficiently by preventing unnecessary file buildup and keeping disk usage under control.

Objectives :

1. **Automate Cleanup of Temporary Files**
   Automatically identify and delete files older than a defined number of days to free up system space.

2. **Track Disk Usage Before and After Cleanup**
   Monitor how much disk space was being used and how much was reclaimed post-cleanup using df -h.

3. **Generate Cleanup Logs**
   Maintain a detailed log file capturing timestamps, actions taken, and disk usage reports for audit and troubleshooting.

4. **Promote System Efficiency**
   Prevent slowdowns and performance issues by removing unnecessary files and managing disk space proactively.

5. **Enable Periodic Execution**
   Make the script reusable and schedulable using cron to ensure consistent maintenance without manual effort.

Importance :

1. **Prevents Disk Space Exhaustion**
   Regularly removing old temporary files ensures critical disk space isn't consumed by unnecessary data.

2. **Improves System Performance**
   A clean and optimized disk helps Linux systems run faster and more reliably, especially for multi-user or server environments.

3. Reduces Manual Work
Automating the cleanup process saves time and reduces human error in managing system storage.

4. Supports Maintenance Best Practices
Logging every cleanup cycle builds a trackable history for audits, debugging, or capacity planning.

5. Essential for DevOps & SysAdmins
Disk management and automation are core responsibilities in Linux system administration and DevOps pipelines.

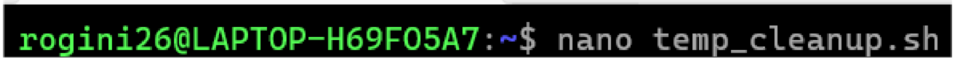
Step-by-Step Overview

Step 1:Open Terminal

Launch a terminal window on your Linux system.

Step 2:Create a Shell Script File

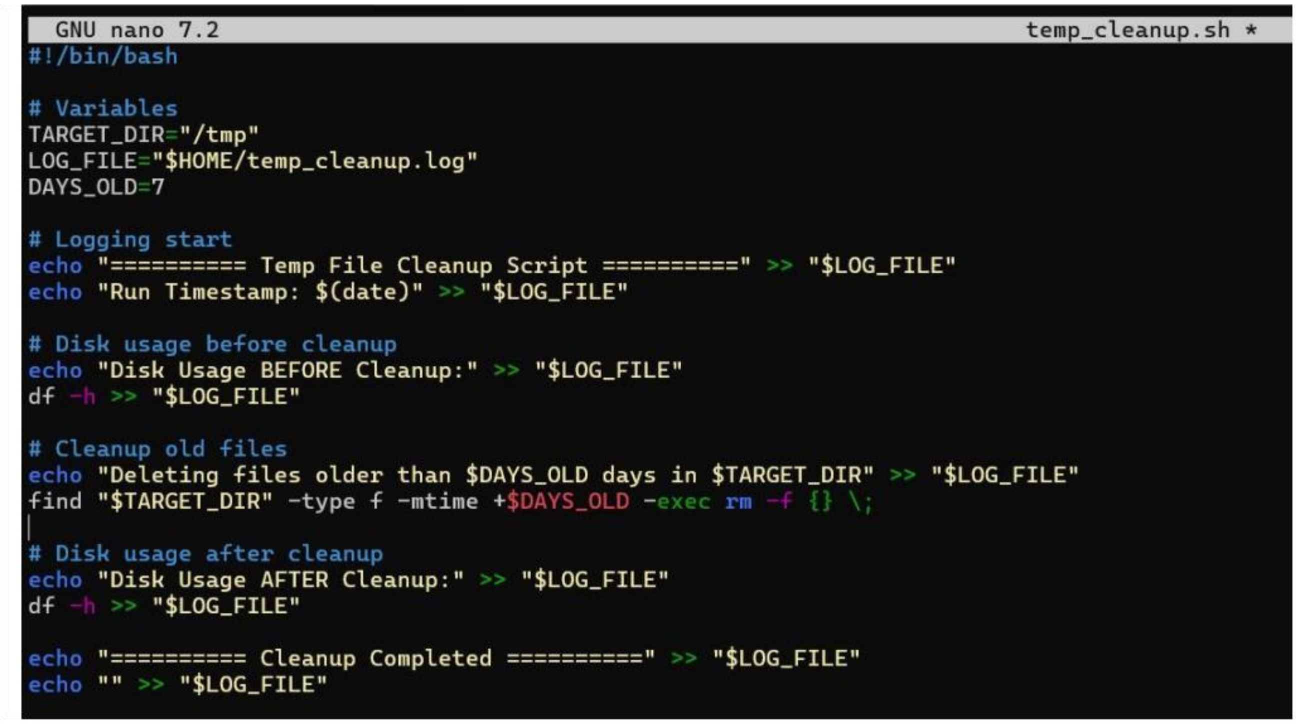
Create a new shell script



```
rogini26@LAPTOP-H69FO5A7:~$ nano temp_cleanup.sh
```

Step 3:Write the Monitoring Script

In the nano editor,Paste the following code:



```bash
#!/bin/bash

# Variables
TARGET_DIR="/tmp"
LOG_FILE="$HOME/temp_cleanup.log"
DAYS_OLD=7

# Logging start
echo "========== Temp File Cleanup Script ==========" >> "$LOG_FILE"
echo "Run Timestamp: $(date)" >> "$LOG_FILE"

# Disk usage before cleanup
echo "Disk Usage BEFORE Cleanup:" >> "$LOG_FILE"
df -h >> "$LOG_FILE"

# Cleanup old files
echo "Deleting files older than $DAYS_OLD days in $TARGET_DIR" >> "$LOG_FILE"
find "$TARGET_DIR" -type f -mtime +$DAYS_OLD -exec rm -f {} \;

# Disk usage after cleanup
echo "Disk Usage AFTER Cleanup:" >> "$LOG_FILE"
df -h >> "$LOG_FILE"

echo "========== Cleanup Completed ==========" >> "$LOG_FILE"
echo "" >> "$LOG_FILE"
```

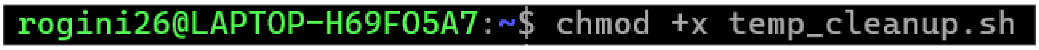
Step 4:Save and Exit

Press Ctrl + O → Enter (to save)

Press Ctrl + X (to exit)

Step 5: Make the Script Executable

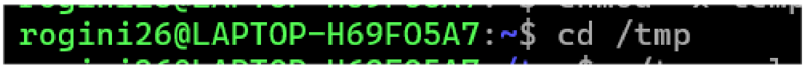
Back in the terminal:



```
rogini26@LAPTOP-H69FO5A7:~$ chmod +x temp_cleanup.sh
```

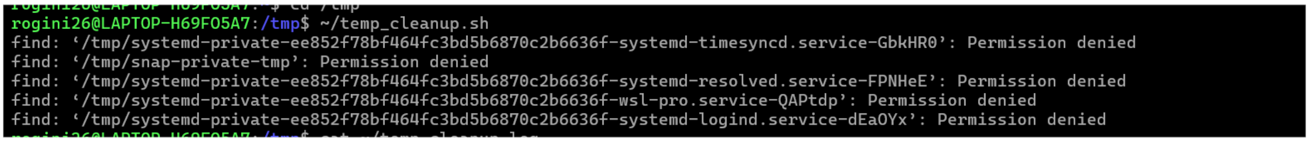
This gives the script permission to run as a program

Step 6: Create Dummy Files for Testing (Optional)



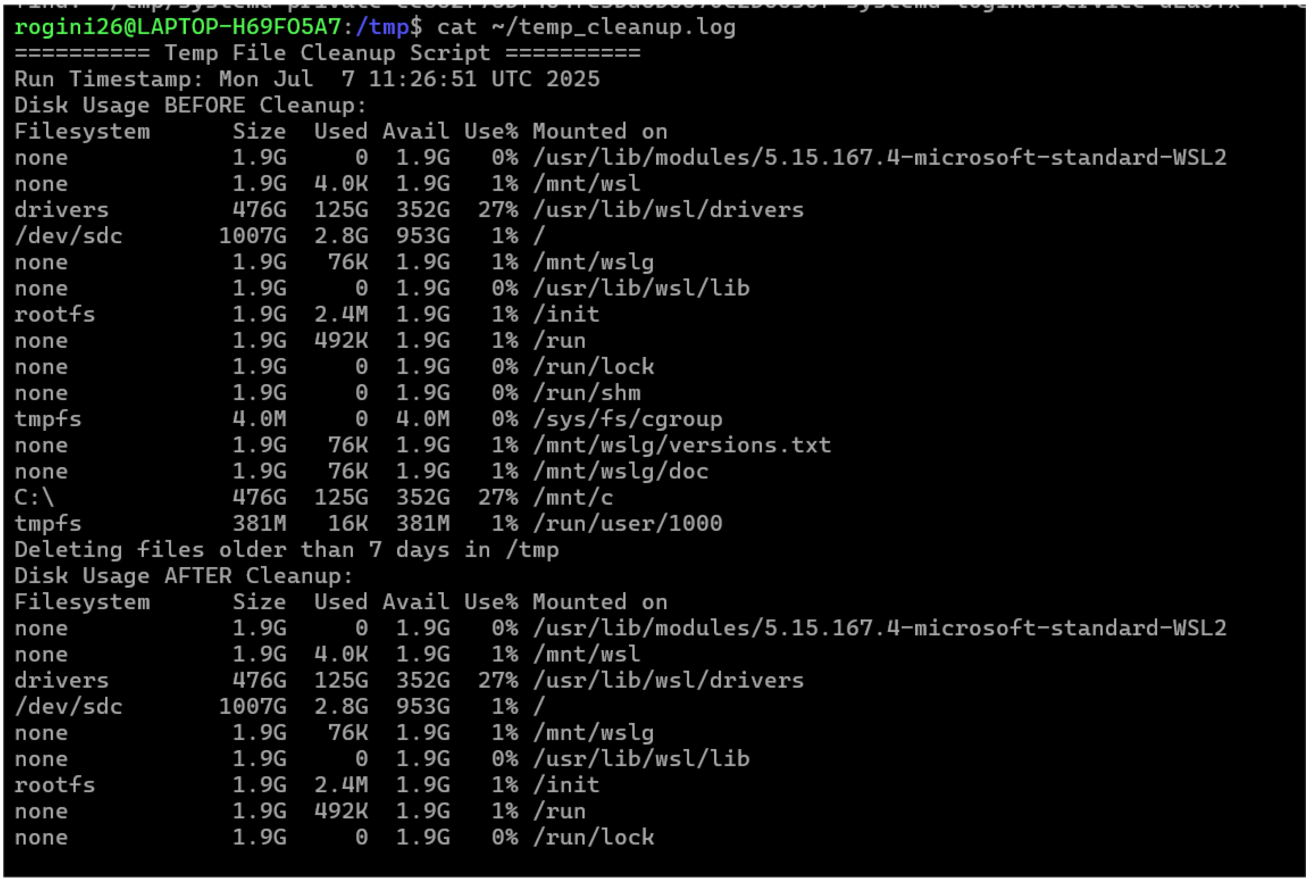
```
rogini26@LAPTOP-H69FO5A7:~$ cd /tmp
```

Step 7: Run the Script



```
rogini26@LAPTOP-H69FO5A7:/tmp$ ~/temp_cleanup.sh
find: ‘/tmp/systemd-private-ee852f78bf464fc3bd5b6870c2b6636f-systemd-timesyncd.service-GbkHR0’: Permission denied
find: ‘/tmp/snap-private-tmp’: Permission denied
find: ‘/tmp/systemd-private-ee852f78bf464fc3bd5b6870c2b6636f-systemd-resolved.service-FPNHeE’: Permission denied
find: ‘/tmp/systemd-private-ee852f78bf464fc3bd5b6870c2b6636f-wsl-pro.service-QAPtdp’: Permission denied
find: ‘/tmp/systemd-private-ee852f78bf464fc3bd5b6870c2b6636f-systemd-logind.service-dEaOYx’: Permission denied
```

Step 8: View the Cleanup Log



```
rogini26@LAPTOP-H69FO5A7:/tmp$ cat ~/temp_cleanup.log
========== Temp File Cleanup Script ==========
Run Timestamp: Mon Jul  7 11:26:51 UTC 2025
Disk Usage BEFORE Cleanup:
Filesystem      Size  Used Avail Use% Mounted on
none            1.9G     0  1.9G   0% /usr/lib/modules/5.15.167.4-microsoft-standard-WSL2
none            1.9G  4.0K  1.9G   1% /mnt/wsl
drivers         476G  125G  352G  27% /usr/lib/wsl/drivers
/dev/sdc       1007G  2.8G  953G   1% /
none            1.9G   76K  1.9G   1% /mnt/wslg
none            1.9G     0  1.9G   0% /usr/lib/wsl/lib
rootfs          1.9G  2.4M  1.9G   1% /init
none            1.9G  492K  1.9G   1% /run
none            1.9G     0  1.9G   0% /run/lock
none            1.9G     0  1.9G   0% /run/shm
tmpfs           4.0M     0  4.0M   0% /sys/fs/cgroup
none            1.9G   76K  1.9G   1% /mnt/wslg/versions.txt
none            1.9G   76K  1.9G   1% /mnt/wslg/doc
C:\             476G  125G  352G  27% /mnt/c
tmpfs           381M   16K  381M   1% /run/user/1000
Deleting files older than 7 days in /tmp
Disk Usage AFTER Cleanup:
Filesystem      Size  Used Avail Use% Mounted on
none            1.9G     0  1.9G   0% /usr/lib/modules/5.15.167.4-microsoft-standard-WSL2
none            1.9G  4.0K  1.9G   1% /mnt/wsl
drivers         476G  125G  352G  27% /usr/lib/wsl/drivers
/dev/sdc       1007G  2.8G  953G   1% /
none            1.9G   76K  1.9G   1% /mnt/wslg
none            1.9G     0  1.9G   0% /usr/lib/wsl/lib
rootfs          1.9G  2.4M  1.9G   1% /init
none            1.9G  492K  1.9G   1% /run
none            1.9G     0  1.9G   0% /run/lock
```

Outcomes:

1. **Successfully Deleted Old Temporary Files**
   All files older than 7 days in the target directory (e.g., /tmp) are automatically removed.

2. **Disk Space Reclaimed**
The script helps free up storage space by clearing out unnecessary files.

3. **Disk Usage Logged Before and After Cleanup**
Disk usage statistics are captured and stored in a log file (temp_cleanup.log), showing how much space was recovered.

4. **Automated Cleanup Process Established**
The script can be reused or scheduled via cron for regular, hands-free execution.

5. **Improved System Health and Maintainability**
Regular cleanup improves performance, prevents storage-related issues, and supports good system hygiene.

6. **Learned and Applied Shell Scripting Skills**
   Reinforced knowledge of bash, find, df -h, logging, permission handling, and automation with cron.