St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

Placement Empowerment Program

Cloud Computing and DevOps Centre

Day 11 – Disk Usage Monitor & Old File Cleaner
Check if disk usage crosses a certain threshold and delete old files in a specific directory if needed.

Name: Rogini.P

Department: ECE

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

In Linux systems, disk space can fill up quickly due to temporary files, logs, backups, or unused data. If not monitored and managed properly, a full disk can lead to application crashes, system errors, or service downtime.

This proof of concept (PoC) introduces a simple yet powerful **bash script** that automates the monitoring of disk usage. When the usage exceeds a defined threshold, the script automatically cleans up old files from a specified directory — helping maintain healthy disk space without manual intervention.

This kind of automation is essential for system administrators and developers managing production servers or even personal systems with limited storage.

Overview

This PoC focuses on building a **bash script** to monitor disk usage and clean up old files when necessary. It works by checking the current disk usage percentage of the root (/) partition and comparing it with a defined threshold. If the usage exceeds this threshold, the script automatically deletes files older than a specified number of days from a target directory.

The process helps prevent potential system issues caused by full disks and ensures that storage is consistently optimized. The script also maintains a log file to record disk usage status and cleanup activities.

Objectives :

✅**Automate Disk Usage Monitoring**
To regularly check the system’s disk usage and detect when it exceeds a safe threshold.

✅**Perform Conditional File Cleanup**

To automatically delete old and unnecessary files from a specific directory only when disk usage is high.

✅**Optimize System Storage**

To ensure the system remains efficient and responsive by freeing up disk space before it becomes critical.

✅**Prevent System Failures**

To avoid crashes, logging issues, or service interruptions caused by a full disk.

✅**Practice Shell Scripting Skills**

To improve hands-on experience with bash scripting, file management, and Linux command-line tools like **df, find, awk, and sed.**

✅ **Enable Log Tracking**

To maintain a history of disk checks and cleanup actions for review, debugging, or audit purposes.

Importance :

✅ **Prevents System Failures**

Full disk space can cause critical applications and services to crash. This script helps avoid such issues by cleaning up space proactively.

✅ **Ensures Smooth System Performance**

Low available disk space can slow down processes or prevent new files from being written. Regular monitoring keeps the system healthy and responsive.

✅ **Reduces Manual Effort**

Automating disk checks and file cleanup saves time and reduces the need for manual maintenance by system admins or developers.

✅ **Useful in Production Servers & Dev Machines**

Especially in cloud-based systems or shared environments where log files and temporary data grow rapidly, this script is highly valuable.

✅ **Improves Resource Efficiency**
It helps you make the most of available storage by removing unnecessary old files that are no longer useful.

✅ **Builds Real-World Scripting Skills**
Working with this script enhances your command-line proficiency and teaches essential system administration tasks using bash.

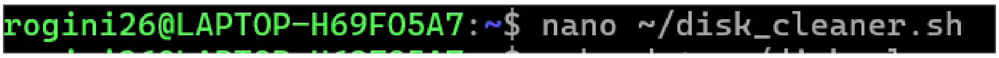
Step-by-Step Overview

Step 1:Open Terminal

Launch a terminal window on your Linux system.

Step 2: Create the Script File

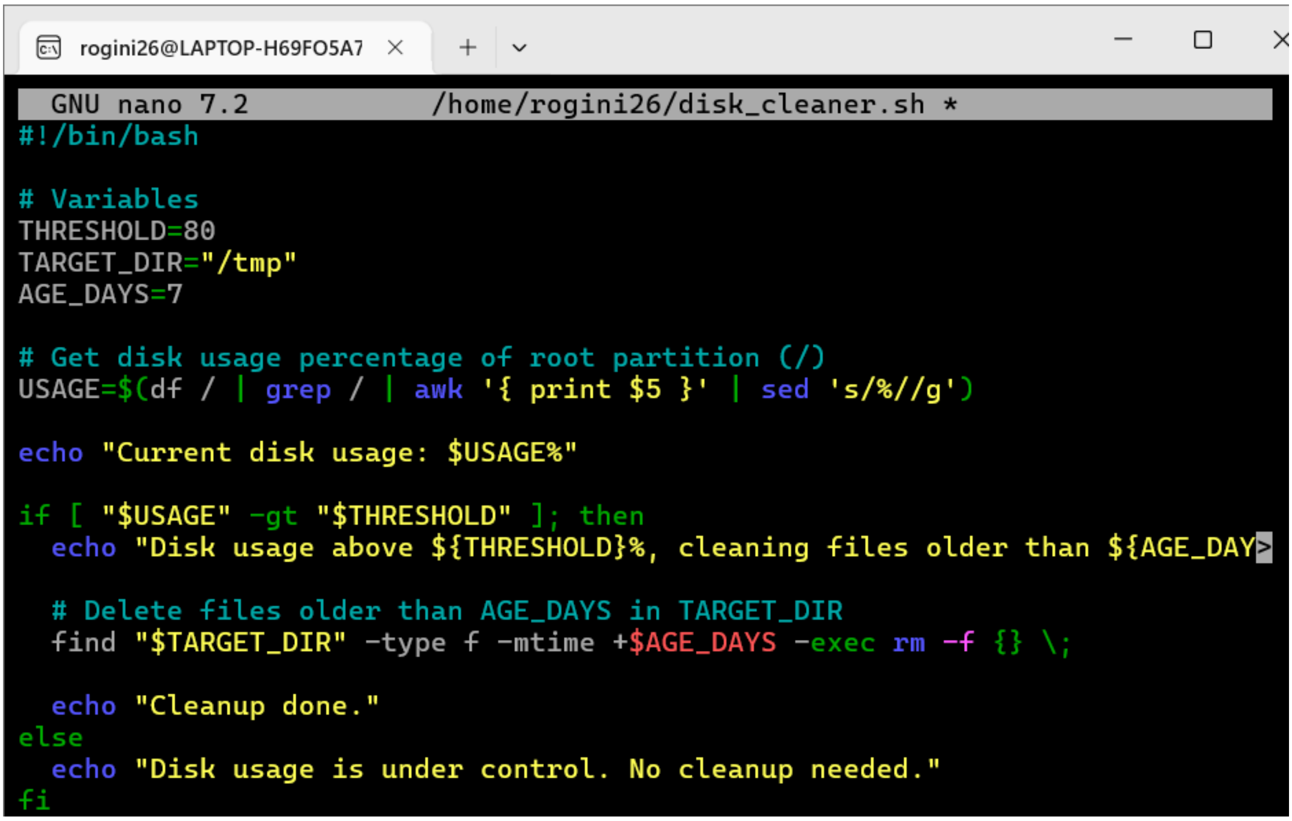
Use nano (or any editor) to create the script:



```
rogini26@LAPTOP-H69F05A7:~$ nano ~/disk_cleaner.sh
```

Step 3: Add the Script Content

Paste the following code:



```
  GNU nano 7.2              /home/rogini26/disk_cleaner.sh *
#!/bin/bash

# Variables
THRESHOLD=80
TARGET_DIR="/tmp"
AGE_DAYS=7

# Get disk usage percentage of root partition (/)
USAGE=$(df / | grep / | awk '{ print $5 }' | sed 's/%//g')

echo "Current disk usage: $USAGE%"

if [ "$USAGE" -gt "$THRESHOLD" ]; then
  echo "Disk usage above ${THRESHOLD}%, cleaning files older than ${AGE_DAY>

  # Delete files older than AGE_DAYS in TARGET_DIR
  find "$TARGET_DIR" -type f -mtime +$AGE_DAYS -exec rm -f {} \;

  echo "Cleanup done."
else
  echo "Disk usage is under control. No cleanup needed."
fi
```

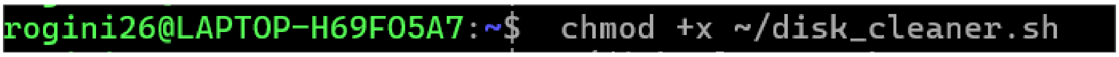
Step 4:Save and Exit

Press Ctrl + O → Enter (to save)

Press Ctrl + X (to exit)

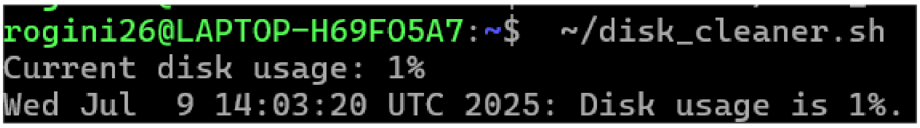
Step 5: Make the Script Executable

Back in the terminal:



```
rogini26@LAPTOP-H69F05A7:~$  chmod +x ~/disk_cleaner.sh
```

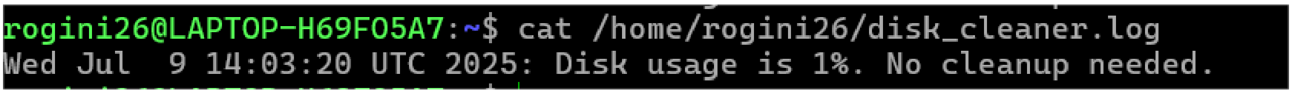
Step 6: Run the Script



```
rogini26@LAPTOP-H69F05A7:~$  ~/disk_cleaner.sh
Current disk usage: 1%
Wed Jul  9 14:03:20 UTC 2025: Disk usage is 1%.
```

Step 7: View the Log File

Check cleanup logs and status:



```
rogini26@LAPTOP-H69F05A7:~$ cat /home/rogini26/disk_cleaner.log
Wed Jul  9 14:03:20 UTC 2025: Disk usage is 1%. No cleanup needed.
```

Outcomes:

✅ **Manual Disk Usage Monitoring Implemented**
You created a bash script that checks disk space usage manually when executed.

✅ **Old File Cleanup Triggered by Script**
When disk usage exceeds the defined threshold, the script deletes files older than a set number of days from a specific folder.

✅ **Log File Maintained Successfully**
The script logs every run with a timestamp, showing current disk usage and whether cleanup was performed.

✅ **No Scheduled Automation (No Cron)**
The script is triggered only when manually run. This is suitable for occasional maintenance or testing purposes.

✅ **Skill Gained: Basic Shell Automation**
You practiced core Linux concepts: file system monitoring, log management, bash scripting, and file operations using **find.**