Placement Empowerment Program

Cloud Computing and DevOps Centre

Day 15 – Simple System Summary Report

Create a script to display basic system details like OS, uptime, disk space, memory usage, and current users.

Name:Rogini.P Department: ECE

Introduction

System administrators and developers often need a quick overview of their system's health and configuration. Instead of running multiple

commands individually every time, a simple shell script can automate the process and generate a neat report.

This Proof of Concept (PoC) focuses on building a **Simple System Summary Report** using a bash script. It gathers essential system information such as **operating system details**, **uptime, disk usage, memory stats, and active users**, and presents it in a human-readable format.

This script is especially useful for beginners learning Linux and shell scripting, as it introduces key system commands and their usage in automation.

Overview

The **Simple System Summary Report** is a lightweight bash script designed to provide an at-a-glance view of a Linux system's current status. It consolidates key information from various system utilities into one clean, readable report.

This script captures the following:
Operating System Info: Displays the OS name and version from system files.
Uptime: Shows how long the system has been running without a reboot.
Disk Usage: Summarizes total disk space used and available.
Memory Usage: Reports available and used RAM and swap memory.
Logged-in Users: Lists all current active users.

This PoC helps automate routine health checks, aiding both system monitoring and educational understanding of Linux resource management tools.

Key steps in this PoC:

✓ **Open Terminal**

Launch the terminal on your Linux system to create and execute the script.

✅ **Create a Bash Script File**
Use a text editor like nano to create a script file named **system_summary.sh.**

✅ **Write the Script** Add commands to display:

- OS information using **cat /etc/os-release**
- System uptime using **uptime -p**
- Disk usage using **df -h --total**
- Memory status using **free -h**
- Current users using **who**

✅ **Make the Script Executable**
Use **chmod +x system_summary.sh** to grant execute permission.

✅ **Run the Script**
Execute the script with **./system_summary.sh** to display the system report.

✅ **Save Output to Log File**
Redirect output to a .log file for record-keeping using:
./system_summary.sh > system_report.log

Objectives :

The main objectives of this PoC are:

✅ **Automate System Health Checks**
Create a reusable script to automatically display key system information.

✅ **Learn Core Linux Commands**
Use essential commands like **uptime, df, free**, and **who** to gather system stats.

✅ **Improve Shell Scripting Skills**
Practice writing and executing bash scripts with formatted outputs.

✅ **Enhance System Monitoring**
Provide a quick and clear overview of system status for users or administrators.

✅ **Generate a Readable Report**
Format the output neatly to be easily interpreted or saved as a log file.

Importance:

✅**Quick Diagnostics**
Provides a fast way to check system health without running multiple commands manually.

✅ **System Maintenance Support**
Helps identify performance issues early by regularly monitoring disk, memory, and uptime.

✅ **Foundation for Advanced Monitoring**
Serves as a stepping stone to more advanced tools like **top, htop, Nagios,** or custom monitoring dashboards.

✅**Boosts Scripting Confidence**
Builds confidence in writing shell scripts and automating tasks.

✅**Useful for Reports and Audits**
The generated report can be saved and shared for auditing or troubleshooting purposes.

Step-by-Step Overview

Step 1:Open Terminal

Launch a terminal window on your Linux system.

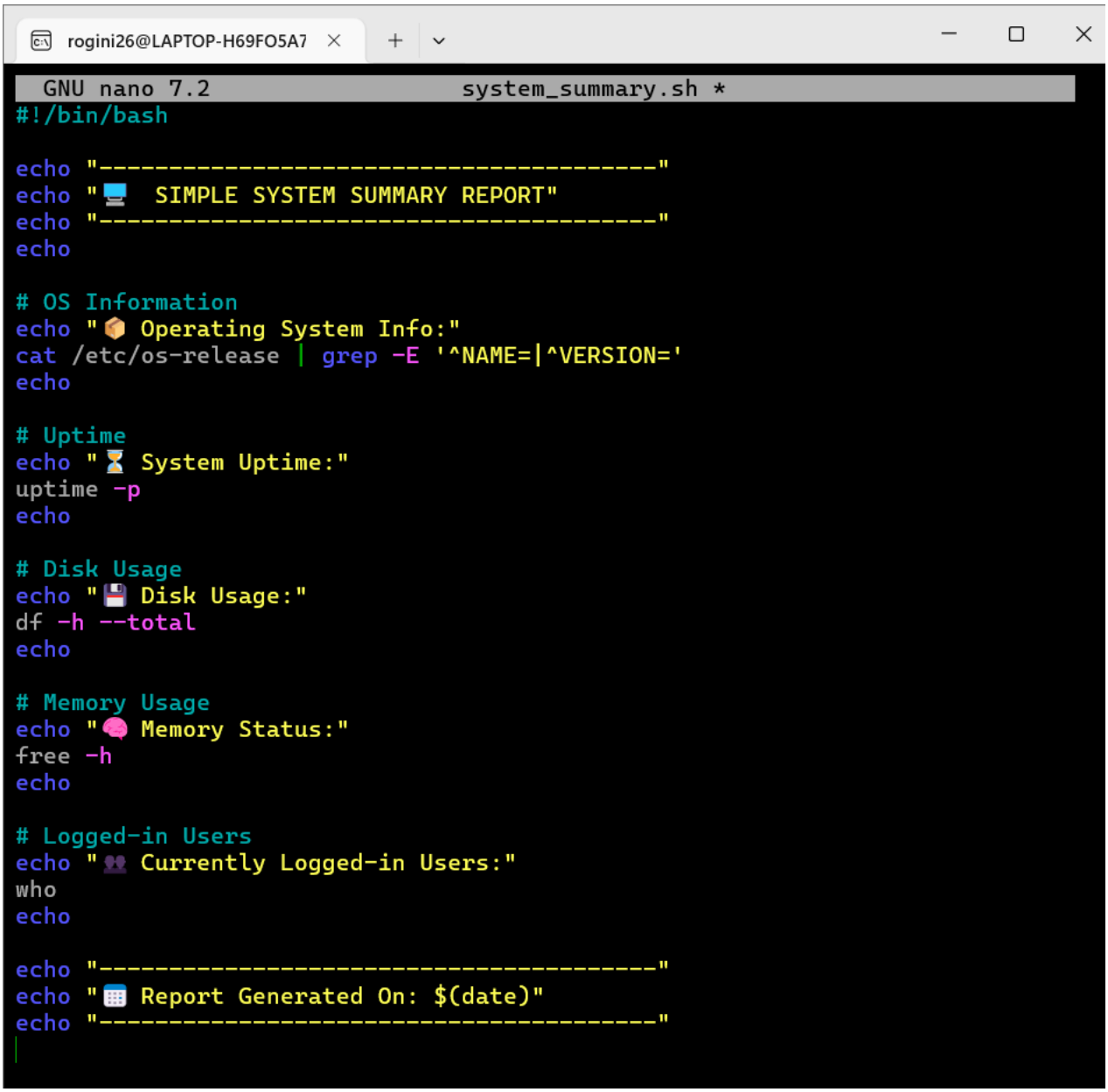
Step 2:Createa ShellScriptFile

Use the nano editor to create a new file

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

Step 3:Write the Monitoring Script

In the nano editor,Paste the following code:



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:

This gives the script permission to run as a program.

Step 6: Run the Script

Run the script to see the system summary:

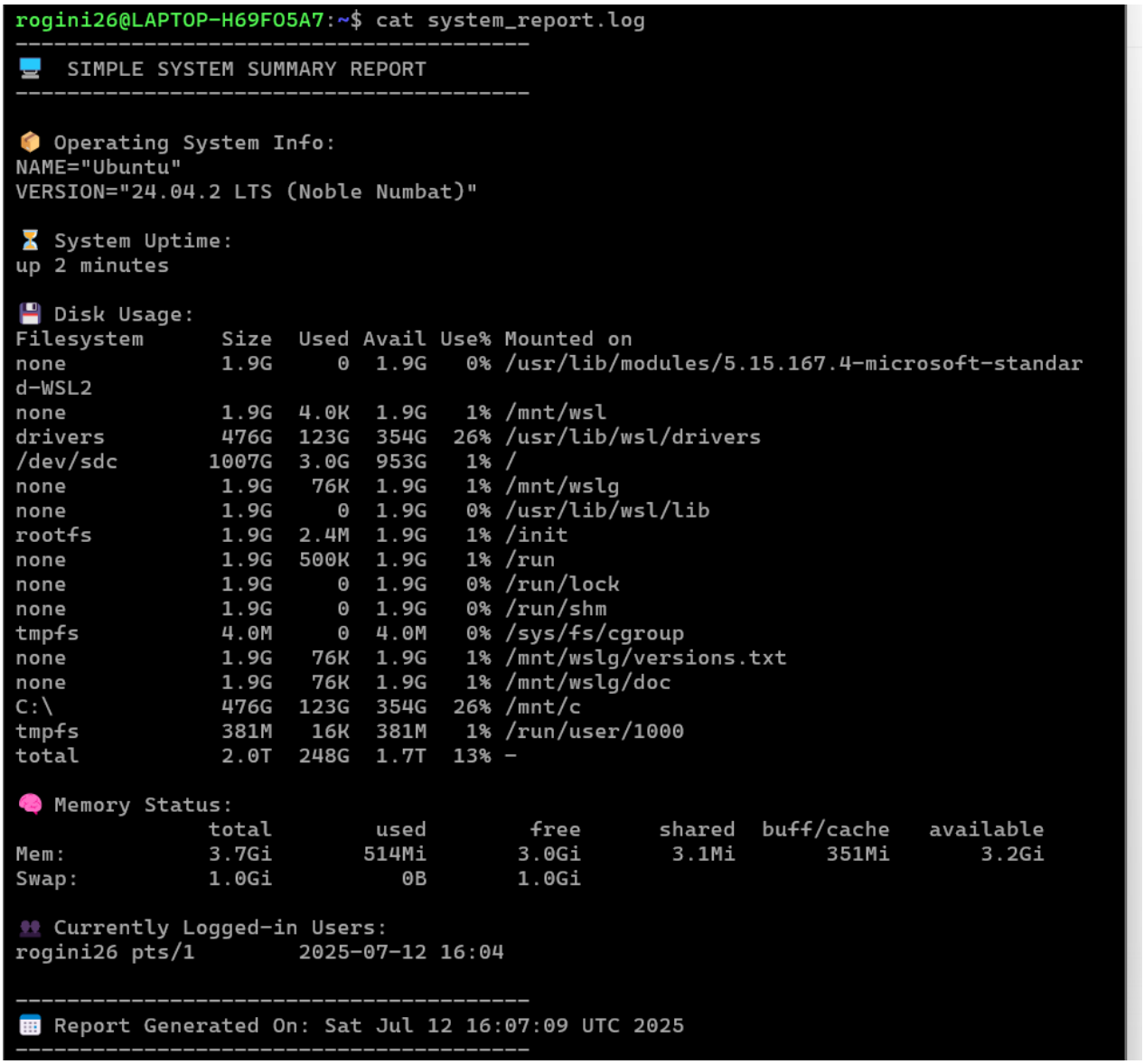
```
rogini26@LAPTOP-H69FO5A7:~$ ./system_summary.sh
----------------------------------------
🖥️  SIMPLE SYSTEM SUMMARY REPORT
----------------------------------------

📦 Operating System Info:
NAME="Ubuntu"
VERSION="24.04.2 LTS (Noble Numbat)"

⏳ System Uptime:
up 2 minutes

💾 Disk Usage:
Filesystem      Size  Used Avail Use% Mounted on
none            1.9G     0  1.9G   0% /usr/lib/modules/5.15.167.4-microsoft-standar
d-WSL2
none            1.9G  4.0K  1.9G   1% /mnt/wsl
drivers         476G  123G  354G  26% /usr/lib/wsl/drivers
/dev/sdc       1007G  3.0G  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  500K  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  123G  354G  26% /mnt/c
tmpfs           381M   16K  381M   1% /run/user/1000
total           2.0T  248G  1.7T  13% -

🧠 Memory Status:
               total        used        free      shared  buff/cache   available
Mem:           3.7Gi       513Mi       3.0Gi       3.1Mi       351Mi       3.2Gi
Swap:          1.0Gi          0B       1.0Gi

👥 Currently Logged-in Users:
rogini26 pts/1        2025-07-12 16:04

----------------------------------------
📅 Report Generated On: Sat Jul 12 16:06:59 UTC 2025
----------------------------------------
```

Step 7 : Save Output to Log File

If you want to store the output:

```
rogini26@LAPTOP-H69FO5A7:~$ ./system_summary.sh > system_report.log
```

You can then view it later using:



Outcomes:

✅ **Understood Key System Commands**
Learned how to use uptime, df, free, who, and cat /etc/os-release.

✅ **Created a Reusable Bash Script**
Built a shell script to automate system status checks.

✅ **Improved Shell Scripting Skills**
Practiced script writing, file permissions, and output formatting.

✅ **Generated a Readable System Report**
Produced clear and organized output summarizing system information.

✅ **Captured Output to a Log File**
Learned how to redirect command output to a file for future reference.

✅ **Strengthened Linux CLI Confidence**
Boosted hands-on experience with Linux command-line operations.

✅ **Prepared for Basic Sysadmin Tasks**
Gained practical knowledge useful for system monitoring and troubleshooting.