

## Task 1: System Monitoring Setup

**Objective:** Configure a monitoring system to ensure the development environment's health, performance, and capacity planning.

---

### Steps:

#### 1. Clone the repository

Use the following command to clone the Git repository:

- `bash: git clone https://github.com/KevinKoreth/monitoring_scripts.git`

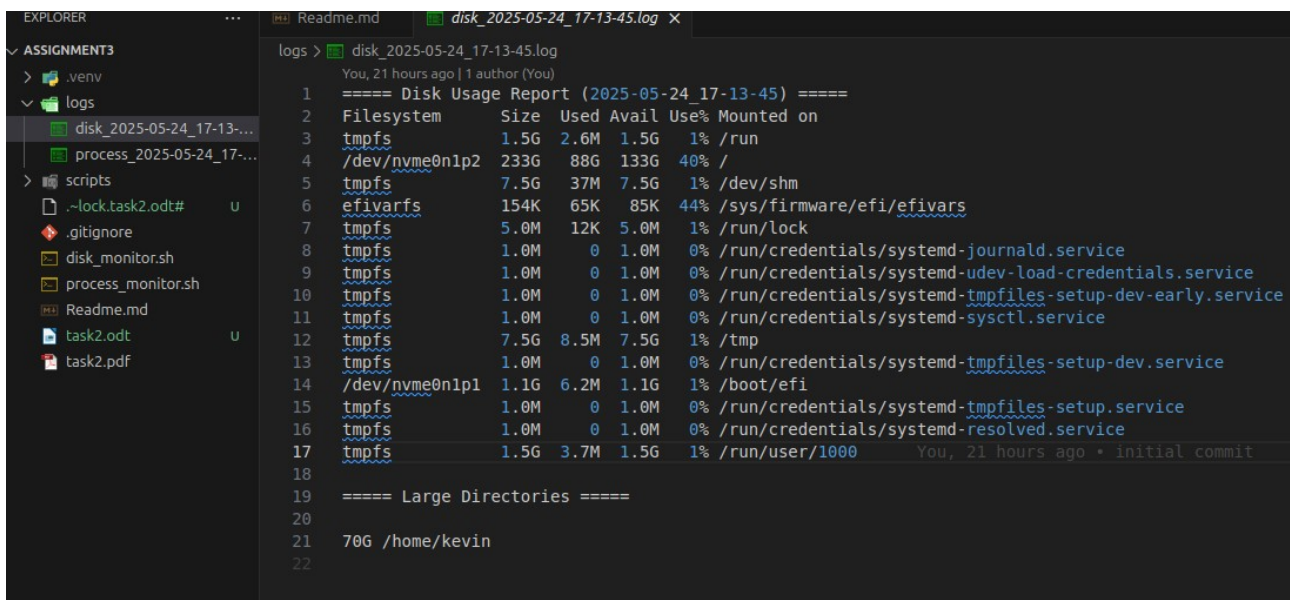
#### • Make the scripts executable

Run the command:

- `cd` into the scripts directory: `cd ./scripts`
- `bash: chmod +x disk_monitor.sh process_monitor.sh`

#### 1. Check overall disk usage

- Execute the `disk_monitor.sh` script to monitor disk usage and identify large directories in `/home`.
- Logs are saved to `logs/disk_<timestamp>.log` (directory auto-created on first run).



```
logs > disk_2025-05-24_17-13-45.log
You, 21 hours ago | 1 author (You)

===== Disk Usage Report (2025-05-24_17-13-45) =====
 1 Filesystem      Size  Used Avail Use% Mounted on
 2 tmpfs           1.5G   2.6M   1.5G   1% /run
 3 /dev/nvme0n1p2  233G   88G  133G  40% /
 4 tmpfs           7.5G   37M   7.5G   1% /dev/shm
 5 efivarfs        154K   65K   85K  44% /sys/firmware/efi/efivars
 6 tmpfs           5.0M   12K   5.0M   1% /run/lock
 7 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-journald.service
 8 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-udev-load-credentials.service
 9 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-tmpfiles-setup-dev-early.service
10 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-sysctl.service
11 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-sysctl.service
12 tmpfs           7.5G   8.5M   7.5G   1% /tmp
13 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-tmpfiles-setup-dev.service
14 /dev/nvme0n1p1  1.1G   6.2M   1.1G   1% /boot/efi
15 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-tmpfiles-setup.service
16 tmpfs           1.0M    0   1.0M   0% /run/credentials/systemd-resolved.service
17 tmpfs           1.5G   3.7M   1.5G   1% /run/user/1000
18
19 ===== Large Directories =====
20
21 70G /home/kevin
22
```

#### 2. Run the process monitor script

- Execute `process_monitor.sh` to list the **top 10 CPU-intensive** and **memory-consuming** processes.

- Logs are saved to `logs/process_<timestamp>.log`.

The screenshot shows a VS Code interface with a file explorer on the left and a terminal on the right. The file explorer shows a directory structure with files like `.env`, `logs`, `disk_2025-05-24_17-13-...`, `process_2025-05-24_17-...`, `scripts`, `task2.odt`, and `task2.pdf`. The terminal shows the output of a command in the `logs` directory, displaying a process report and memory usage for the file `process_2025-05-24_17-14-30.log`.

```

logs > process_2025-05-24_17-14-30.log
You, 21 hours ago | 1 author (You)
===== Process Report (2025-05-24 17-14-30) =====
 1  USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
 2                                     R+  17:14    0:00 ps aux --sort=-%cpu
 3  kevin     76746 300  0.0   11428   4344 pts/0
 4  kevin     8801 48.0  2.5  3333816 396436 ?
 5  kevin     7756 37.8  4.5  12858116 708488 ?
 6  kevin     6965 18.8  2.2  5710616 355660 ?
 7  kevin     14036 5.1  5.9  3794524 932508 ?
 8  kevin     9607 4.3  1.4  2634724 226460 ?
 9  kevin     14787 4.1  2.0  1221569312 327040 ?
10  kevin     41441 3.5  1.4  1216431760 235464 ?
11  kevin     14763 3.2  0.7  34343136 116512 ?
12
13 ===== Memory Usage ===== You, 21 hours ago * initial commit ...
14
15  USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
16  kevin     14036 5.1  5.9  3794524 932508 ?
17  kevin     7756 37.8  4.5  12858116 708488 ?
18  kevin     44345 2.2  3.9  1218090376 619504 ?
19  kevin     42625 1.9  3.8  1218090316 610200 ?
20  kevin     44503 2.7  3.1  9726052 498440 ?
21  kevin     9153 0.7  2.7  2967444 440148 ?
22  kevin     8801 48.0  2.5  3333816 396436 ?
23  kevin     6965 18.8  2.2  5710616 355660 ?
24  kevin     14787 4.1  2.0  1221569312 327040 ?
  
```

## File Locations

- **Scripts:** Located in the root directory of the cloned repository.
- **Logs:** Stored in the `logs/` directory (automatically created when scripts run for the first time).