**Project: System Resource Monitoring and Reporting with Bash**

**Step 1: Define Monitoring Metrics for System Resource Monitoring Script**

Before writing the script, we need to **identify** which system resources to monitor and **set threshold values** that determine when an alert should be triggered.

**Key Metrics to Monitor**

We will track the following **three critical system resources**:

**CPU Usage**

* **Why?** High CPU usage can indicate system overload, inefficient processes, or possible failures.
* **How to Measure?** We will check the percentage of CPU usage.
* **Threshold:** If CPU usage **exceeds 80%**, send an alert.

**Memory (RAM) Usage**

* **Why?** Running out of memory can cause slowdowns or system crashes.
* **How to Measure?** We will check the percentage of used memory compared to total available memory.
* **Threshold:** If memory usage **exceeds 80%**, send an alert.

**Disk Space Usage**

* **Why?** If a disk is nearly full, system performance may degrade, and logs or processes may fail.
* **How to Measure?** We will check the percentage of disk space used on the root partition (/).
* **Threshold:** If disk space **exceeds 90%**, send an alert.

**Threshold Summary**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Measurement** | **Threshold** |
| **CPU Usage** | % used | **80%** |
| **Memory Usage** | % used | **80%** |
| **Disk Space Usage** | % used (on /) | **90%** |

**Optional: Additional Monitoring Metrics (Advanced)**

If needed, we can expand monitoring by tracking:

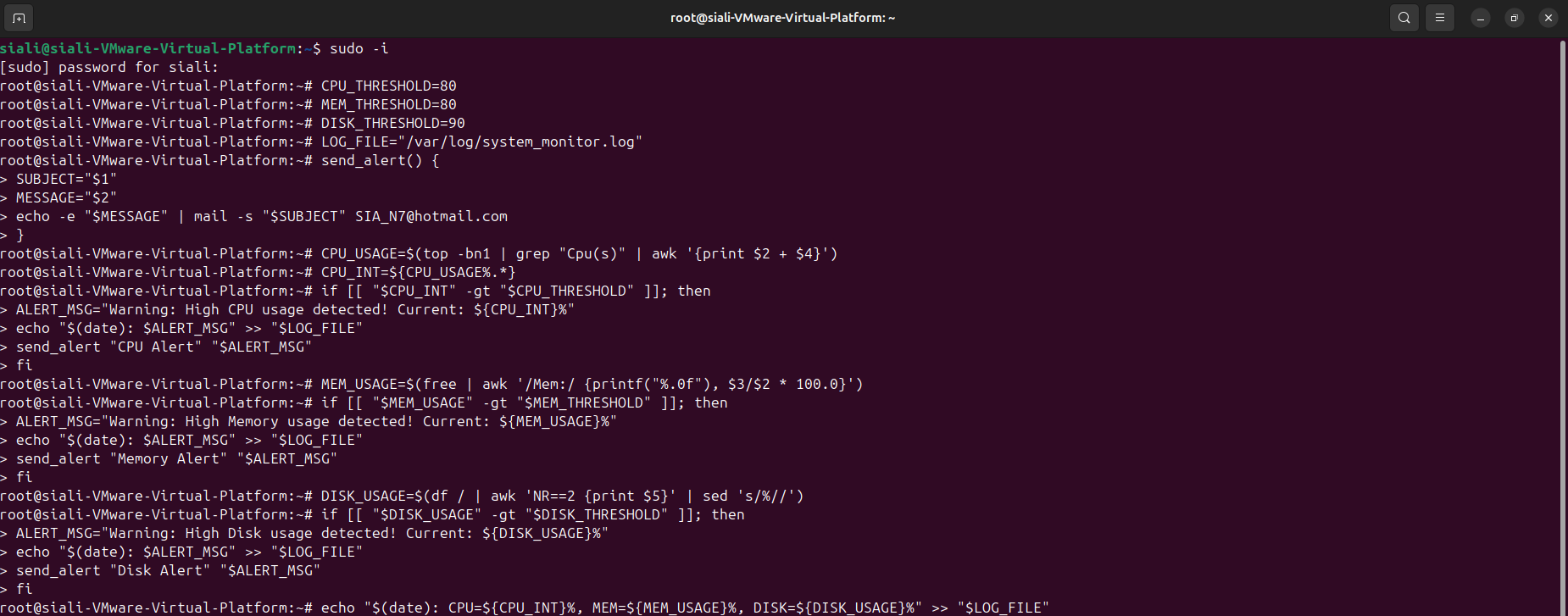
* **Network Usage** (Check upload/download speeds)
* **Process Monitoring** (Detect high-resource-consuming processes)
* **System Load Average** (Check overall system performance)

**Step 2: Create a Bash Script (monitor.sh)**

This script will:

1. Collect system resource data.
2. Compare metrics against predefined thresholds.
3. Send alerts via email if thresholds are exceeded.
4. Log data for reporting.

Here is my script (monitor.sh) screenshot:



**Step 3: Automate Script Execution with Cron**

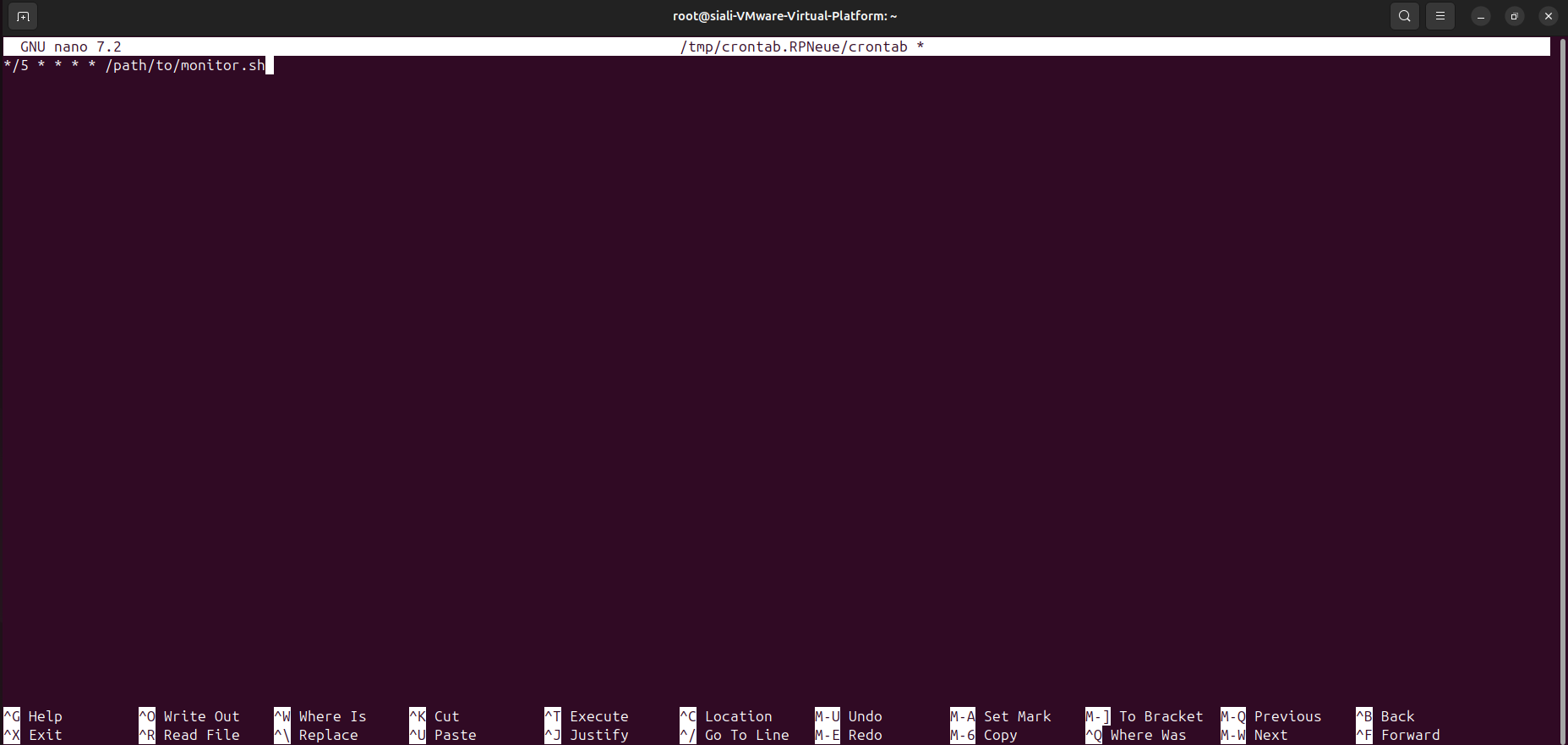
To run the script every 5 minutes, do the following:

1. Open the cron job editor and choose nano:



1. Add the following line:

\*/5 \* \* \* \* /path/to/monitor.sh



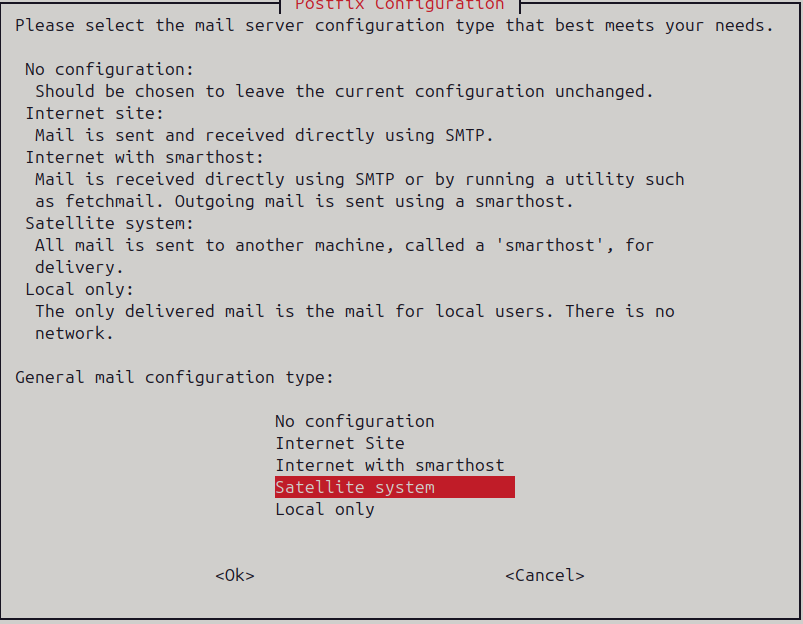
This ensures the script runs automatically.

**Step 4: Set Up Email Alerts**

For email notifications, install mailutils (for Debian-based systems like Ubuntu):



Choose “Internet Site” when going through the ‘mailutils’ installation. Also, Type in your email address as your System Mail Name.



**Step 5: Verify and Test**

Run the script manually to test:



Check the logs:



root@siali-VMware-Virtual-Platform:~# cat /var/log/system\_monitor.log

Mon Feb 10 09:49:58 PM GMT 2025: CPU=5%, MEM=37%, DISK=5%

This setup provides **automated system monitoring and alerting**, improving reliability and proactive issue detection.