**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**Write the shell Script to Monitor logs:**

Create a script that monitors server logs

For errors and alerts you.

Name: Bhavadharani S Department : CSE



**Introduction and Overview**

In today's digital landscape, maintaining server reliability is critical for business continuity and system performance. Log files serve as vital records of system activity, capturing errors, security threats, and performance issues. However, manually monitoring logs is time-consuming and prone to oversight. Automating this process with a shell script ensures continuous, real-time detection of anomalies, enabling quick responses to prevent downtime, security breaches, or system failures. By scanning critical logs for predefined error patterns, the script acts as an early warning system, enhancing overall server health, security, and efficiency.

**Objective**

The objective of this task is to implement an automated log monitoring system using a shell script to detect and alert administrators about critical issues. Why is this important?

* Proactive Issue Resolution: Early detection prevents minor issues from escalating into major failures.
* Reduced Downtime: Immediate alerts ensure quick response, minimizing service interruptions.
* Enhanced Security: Detects unauthorized access attempts, system breaches, or unusual activities.
* Operational Efficiency: Eliminates the need for manual log checks, allowing administrators to focus on higher-priority tasks.

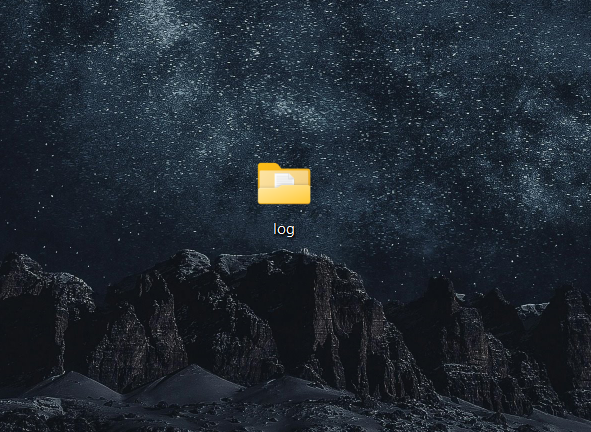
**Importance of Automated Server Log Monitoring**

Automating server log monitoring is **crucial** for maintaining system reliability, security, and performance. It enables **early detection of issues**, preventing downtime and system failures. Automated alerts ensure **quick response**, reducing troubleshooting time and improving efficiency. It also enhances **security** by detecting unauthorized access and anomalies. By eliminating manual log reviews, it **saves time and operational costs**, making IT management more efficient.

**Step-by-Step Overview**

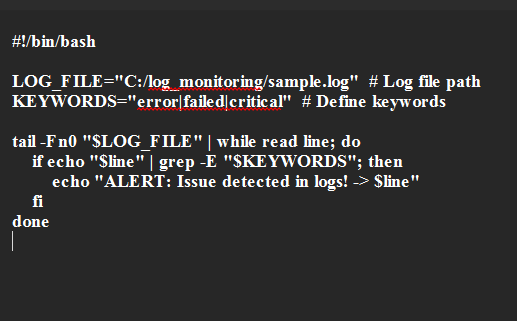
**Step 1: Create a Folder**

Create a folder on your desktop or in a desired location to store your log monitoring script.



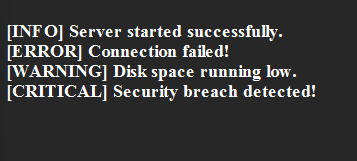
**Step 2: Create the Script**

Write a shell script (monitor\_logs.sh) to monitor server logs for specific error patterns and send alerts when detected.

+

**Step 3: Create a Sample Log File**

Create a sample log file (sample.log) with some example log entries (e.g., errors, warnings) in the folder created in Step 1.

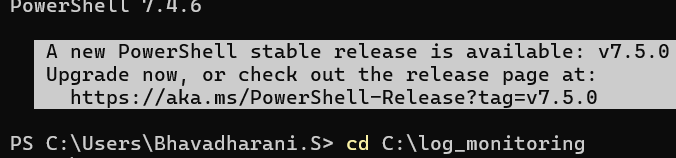


**Step 4: Install Git Bash (Optional)**

If not installed, download and install **Git Bash** from [Git's website](https://git-scm.com/downloads).

**Step 5: Open PowerShell or Git Bash**

Open **PowerShell** (or **Git Bash**) and navigate to the folder where the script is saved (cd C:\log\_monitoring).1



**Step 6: Set Execution Policy (PowerShell Only)**

If using PowerShell, set the execution policy to RemoteSigned using the command:

***Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned***



**Step 7: Run the Script**

Run the shell script (monitor\_logs.sh) using either **Git Bash** or **WSL** (if configured), with the command:

./***monitor\_logs.sh***



**Step 8: Test and Verify**

Add error or warning log entries in the sample log file and check if the script detects and alerts you about them.

These steps will guide you through monitoring server logs for errors using a shell script.

