 A logo with a smile

Description automatically generated

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

**TASK-6** Write the Shell Script to Monitor Logs : Create a script that monitors server logs for errors and alert you

Name:VASANTH S Department:CSE

A black and white logo with a circle and a circle with a logo on it

Description automatically generated with medium confidence

**Introduction**

Log files are essential for tracking system activities, application events, and potential security threats. They help in diagnosing errors, identifying warnings, and detecting unusual activities. However, manually monitoring logs can be time-consuming and prone to oversight.

This Proof of Concept (PoC) focuses on automating log monitoring using a PowerShell script. The script continuously scans log files in real time, detecting specific keywords (such as "error") and generating alerts when critical events are found. This automation enhances efficiency, ensures prompt issue detection, and minimizes manual effort.

**Overview**

Log monitoring is crucial for maintaining system health, identifying performance issues, and ensuring security compliance. This PoC aims to automate log file monitoring using a PowerShell script that scans logs in real time, searching for predefined keywords like "error" or "warning."

The script continuously checks log files, triggers alerts upon detecting critical entries, and provides immediate feedback to system administrators. This automation reduces manual effort, improves response time, and ensures that no critical log entry goes unnoticed. By implementing this solution, organizations can proactively address potential system failures and security threats.

**Objective**

The objective of this project is to:

1. Automate the process of monitoring log files for critical events.

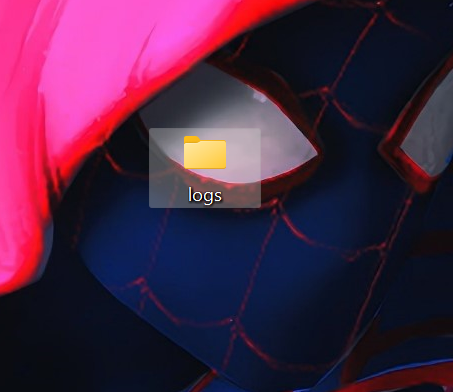
2. Learn how to create and execute PowerShell scripts on a Windows system.

3. Demonstrate real-time detection of keywords like "error" in log files. 4. Enhance troubleshooting efficiency by providing immediate alerts for critical events.

**Step-by-Step Overview**

Step1:

Create a Folder called logs for Your Logs and Script.



Step 2:

Open Notepad and Add the following sample text to it and Save the file as **Log.log**inside the logsfolder.

A screenshot of a computer

Description automatically generated

Step 3:

Open Notepad and Type the following PowerShell script into it and Set the $LogFilePath address to the mylogfile.log which you saved in logs folder. Save the file as monitor.ps1 inside the same logs folder.

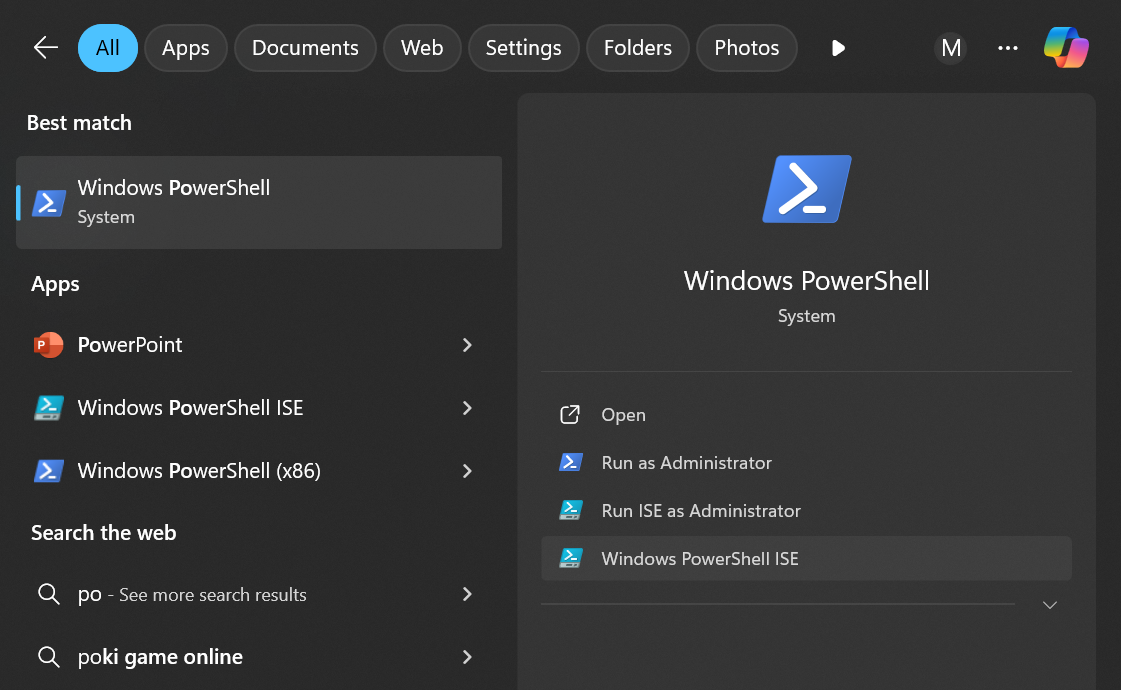
(Note:change your log address to the path given in the second like of the image)

A screenshot of a computer

Description automatically generated

Step 4:

Click the Windows Key and Search for Windows PowerShell and click Run as Administratostrator.

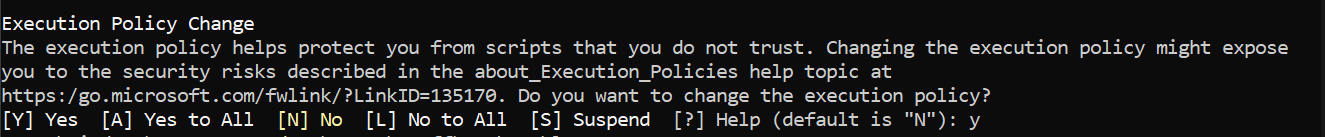


Step 5:

Run the following command to allow script execution:

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

When prompted, type Y and press Enter.



Step 6:

Navigate to the logsfolder. (use cd path of the folder)



Step 7:

Run the script:

**.\monitor.ps1 (give your name of monitoring file created in the format of ps1)**



Step 8:

Open logs.log in Notepad and Add a new line with the word "error" and Save the file

A screenshot of a computer

Description automatically generated

Step 9:

Check PowerShell — you should see an alert like:

**ALERT: Keyword 'error' found in log: error: A new issue occurred!**



This occurs why because while script is running, it conintously monitors the log file we given. If any change done with the keyword “error” it triggers and alerts you!!.