

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 alert you

Name:Dhamini B

Department: IT

Introduction

Log files play a critical role in IT systems, as they record activities and events generated by applications, servers, and network devices. Monitoring these logs helps identify issues such as errors, warnings, and suspicious activities that may require immediate attention. Automating the monitoring process ensures efficiency and reduces the risk of missing critical information.

This PoC demonstrates the creation of a **PowerShell script** to monitor logs in real-time. The script will detect specific keywords (like "error") in a log file and alert the user when such events occur.

Overview

This project involves writing and running a PowerShell script that continuously scans a log file for specific keywords. The script:

1. Reads the log file in real time.
2. Matches new entries in the log against predefined keywords
3. Triggers an alert when a match is found.

This solution is lightweight and practical for system administrators and IT professionals to monitor logs on Windows systems.

Objective

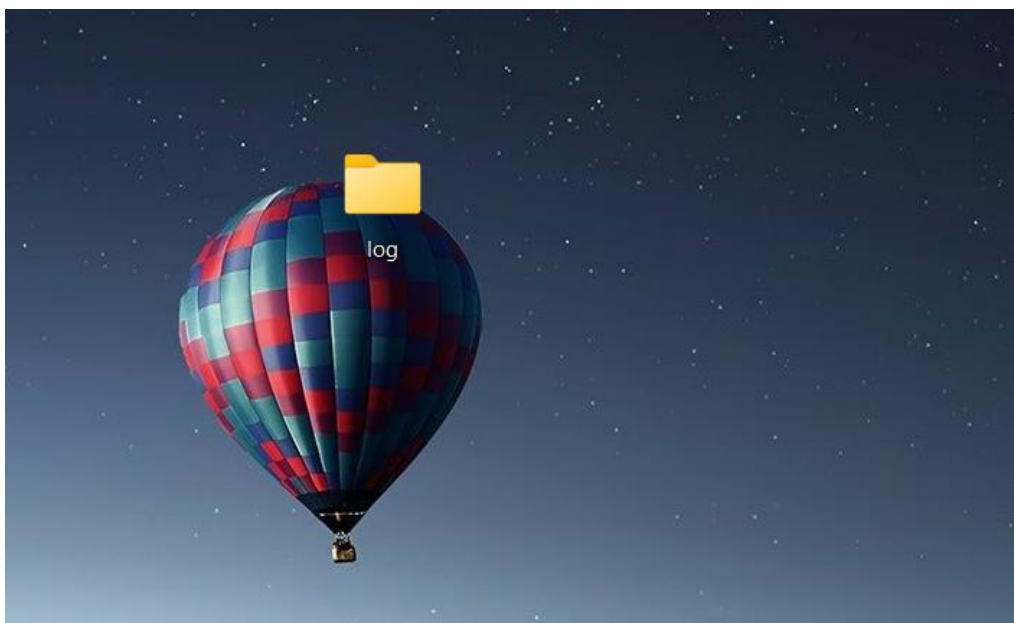
The objective of this project is to:

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

Step 1:

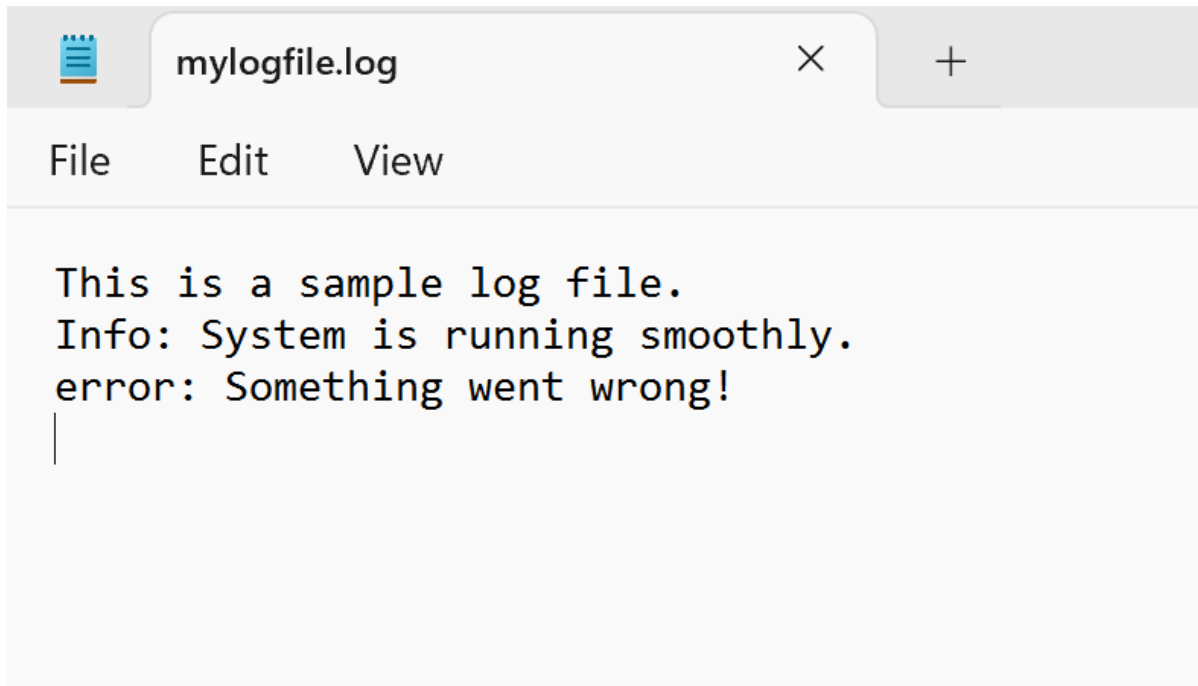
Create a Folder called logs for Your Logs and Script



Step 2:

Open Notepad and Add the following sample text to it and Save the file as **mylogfile.log** inside the logs folder

xxxx



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_logs.ps1 inside the same logs folder

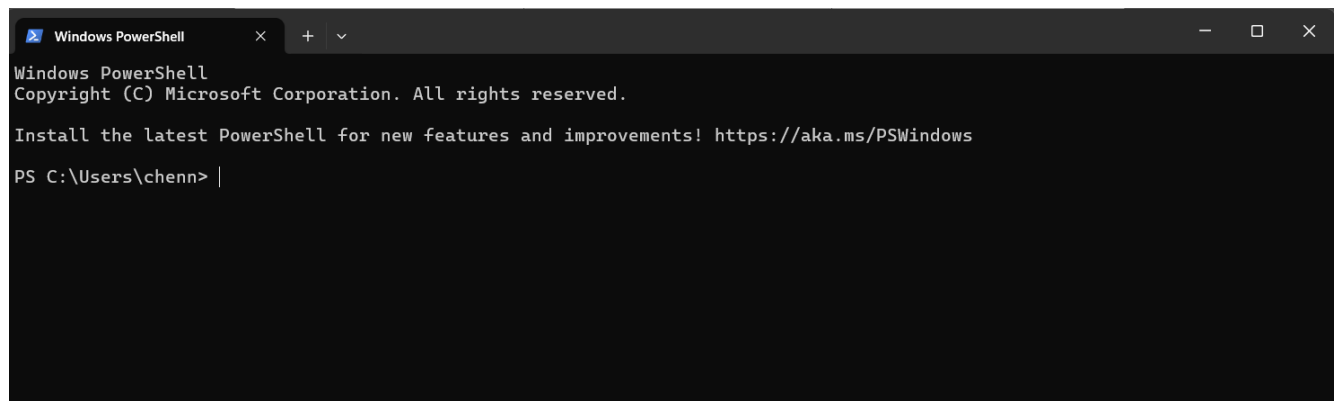
A screenshot of a Notepad window with a dark theme. The title bar shows 'Hi)' and two tabs: 'index.bat' and 'mylogfile.log'. The menu bar includes 'File', 'Edit', and 'View'. The script content is as follows:

```
# Define the path to the log file
$LogFilePath ="C:\Users\chenn\OneDrive\Desktop\log"
# Define the keyword to monitor
$keyword = "error"

# Function to send an alert
Function Send-Alert {
    param([string]$Message)
    Write-Host "ALERT: $Message" -ForegroundColor Red
    Add-Content -Path "C:\Users\chenn\OneDrive\Desktop\log.txt" -Value "$(Get-Date) - $Message"
}
# Monitor the log file for new entries
Get-Content -Path $LogFilePath -Wait -Tail 10 | ForEach-Object {
    if ($_ -match $keyword) {
        Send-Alert "Keyword '$keyword' found in log: $_"
    }
}
```

Step 4:

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

A screenshot of a Windows PowerShell window titled 'Windows PowerShell'. The title bar includes standard window controls. The window content shows the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\chenn> |
```

Step 5:

Run the following command to allow script execution:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

When prompted, type Y and press Enter.

```
PS C:\Windows\system32> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https://go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Y
```

Step 6:

Navigate to the logs folder

```
PS C:\Users\chenn> "C:\Users\chenn\OneDrive\Desktop\log"
```

Step 7:

Run the script:

.\monitor_logs.ps1

```
PS C:\Users\chenn> .\monitor_logs.ps1
```

Step 8:

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

```
File Edit View

This is a sample log file.
Info: System is running smoothly.
error: Something went wrong!
error: A new issue occurred!
```

Step 9:

Check PowerShell — you should see an alert like:

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

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

keyword "error" is added to the file, it immediately triggers an alert in PowerShell.

Outcome:

By completing this Proof of Concept (PoC), we will:

1. Successfully create and execute a PowerShell script to monitor log files in real time.
2. Detect and alert on predefined keywords (e.g., "error") to highlight critical events.
3. Gain hands-on experience with PowerShell scripting and automation on a Windows system.
4. Understand the importance of log monitoring in proactive system maintenance and troubleshooting.
5. Learn to customize and scale the script for more advanced monitoring scenarios in future projects.