

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: Mohana Priya M

Department: CSE

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.

Step-by-Step Overview

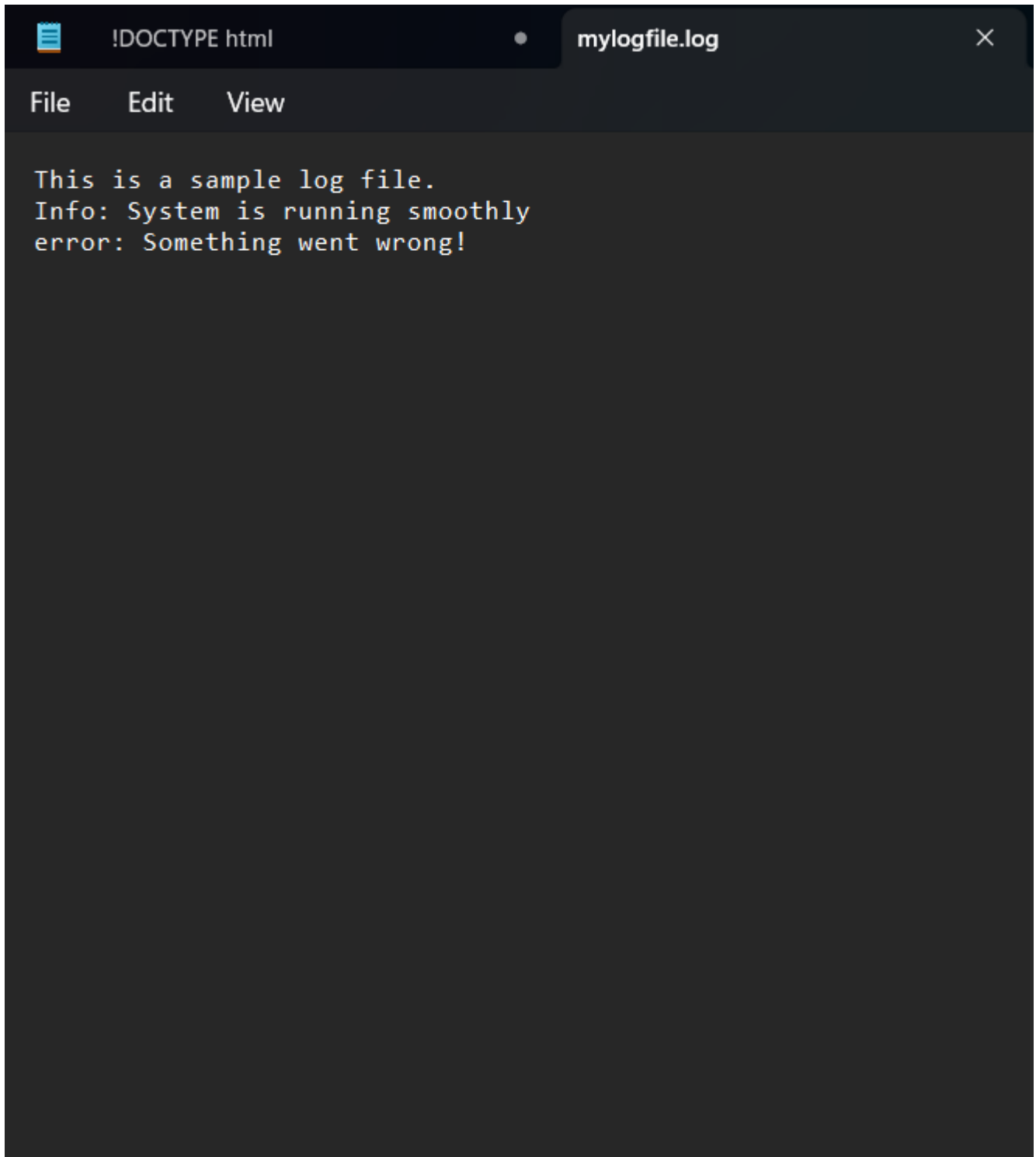
Step 1:

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

A screenshot of a Notepad application window. The title bar shows a single tab named 'mylogfile.log' with a close button on the right. The menu bar includes 'File', 'Edit', and 'View'. The text area contains three lines of sample log data: 'This is a sample log file.', 'Info: System is running smoothly', and 'error: Something went wrong!'. The text is displayed in a monospaced font with syntax highlighting: 'This' is blue, 'Info:' is green, and 'error:' is red. The rest of the text is black. The background of the Notepad window is dark gray.

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

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

```
File Edit View

# Define the path to the log file
$LogFilePath = "C:\Users\mjnev\OneDrive\Desktop\LOGS"

# Define the keyword to monitor
$Keyword = "error"

# Function to send an alert
Function Send-Alert {
    param([string]$Message)
    Write-Host "ALERT: $Message" -ForegroundColor Red
}

# Monitor the log file for new entries
Get-Content -Path $LogFilePath -Wait -Tail 0 | ForEach-Object {
    if ($_ -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.



All

Apps

Documents

Web

Settings

Folders

Pl



200



M



Best match



Windows PowerShell
System

Apps



Windows Tools



Windows Defender Firewall with Advanced Security



Windows Media Player Legacy



Windows PowerShell ISE



Windows PowerShell (x86)



Windows Memory Diagnostic



jdk-21_windows-x64_bin (1).exe



jdk-21_windows-x64_bin.exe



Settings (8+)

Search the web



windows - See more search results



Windows PowerShell
System



Open



Run as Administrator



Run ISE as Administrator



Windows PowerShell ISE



Step 5:

Run the following command to allow script execution:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

When prompted, type Y and press Enter.

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

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

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
PS C:\Windows\system32> _
```

Step 6:

Navigate to the logs folder

```
PS C:\Windows\system32> cd C:\Users\mjnev\OneDrive\Desktop\LOGS
```

Step 7:

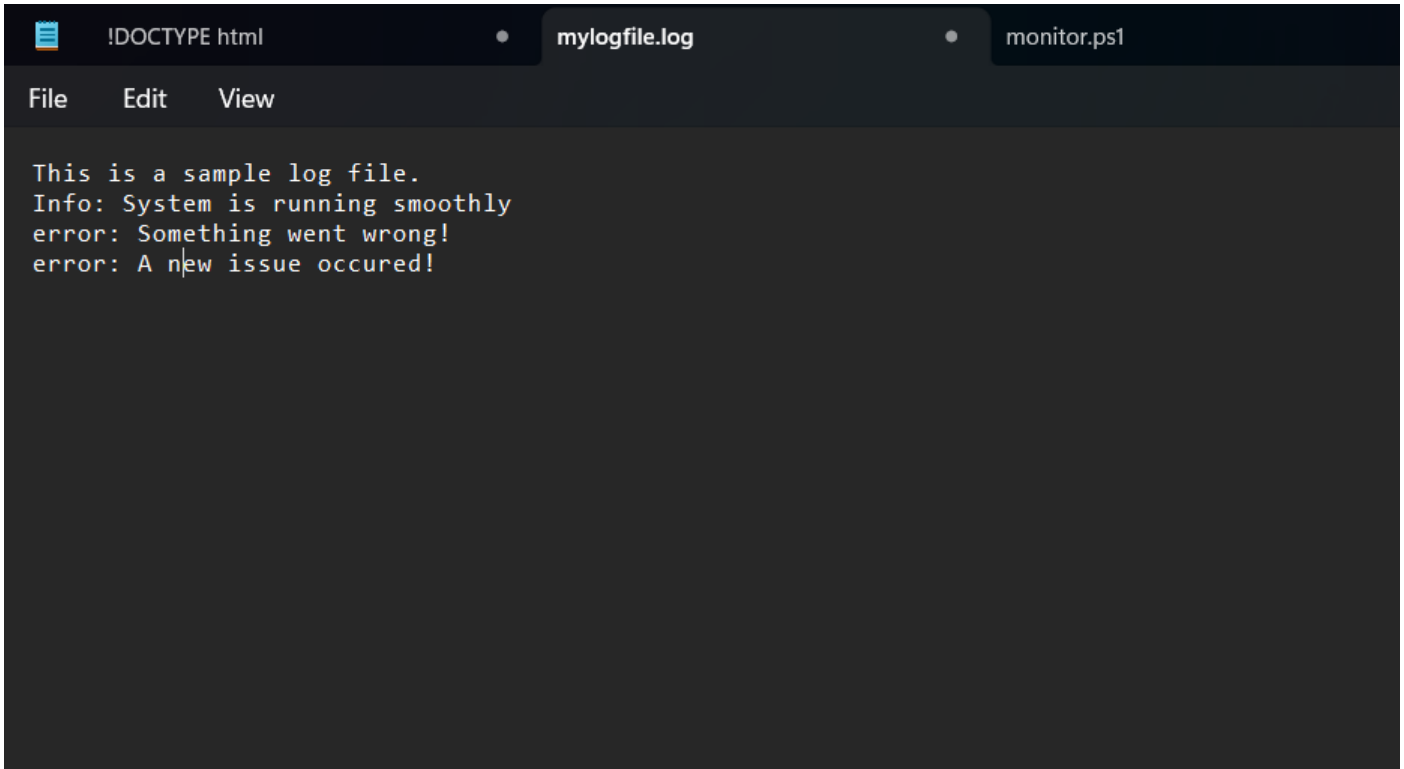
Run the script:

.\monitor.ps1

```
PS C:\Users\mjnev\OneDrive\Desktop\LOGS> ./monitor.ps1
```

Step 8:

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

A screenshot of a Notepad window with a dark theme. The title bar shows three tabs: "!DOCTYPE html", "mylogfile.log", and "monitor.ps1". The "mylogfile.log" tab is active. The menu bar includes "File", "Edit", and "View". The text content of the file is as follows:

```
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!
```


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.