



St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119



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: CHANDRU S

Department: INFORMATION TECHNOLOGY



Introduction

Log files are essential components of IT systems, capturing activities and events from applications, servers, and network devices. Monitoring these logs is crucial for identifying errors, warnings, and potential security threats that require prompt attention. Automating this process enhances efficiency and minimizes the risk of overlooking critical information.

This proof of concept (PoC) showcases a PowerShell script designed for real-time log monitoring. The script scans log files for specific keywords, such as "error," and promptly alerts the user upon detection, ensuring timely response to important events.

Overview

This project focuses on developing a PowerShell script that actively monitors a log file for specific keywords in real time. The script:

1. Continuously reads the log file as new entries are added.
2. Checks each new entry against predefined keywords (e.g., "error").
3. Generates an alert when a match is detected.

Designed to be lightweight and efficient, this solution provides system administrators and IT professionals with a practical approach to real-time log monitoring on Windows systems.

Objectives

This project aims to:

1. Automate log file monitoring to detect critical events efficiently.
2. Develop and execute PowerShell scripts on a Windows system.
3. Implement real-time keyword detection, such as identifying "error" entries in logs.
4. Improve troubleshooting by delivering instant alerts for important events.

Importance

1. Proactive Issue Detection

Real-time log monitoring enables immediate identification of errors and issues, minimizing downtime and enhancing system reliability.

2. Hands-On Automation Experience

This project provides beginners with practical exposure to PowerShell scripting, a powerful tool for IT automation.

3. Cost-Effective Monitoring

By leveraging PowerShell, organizations can implement effective log monitoring without investing in expensive third-party solutions.

4. Improved Efficiency

Automating log analysis eliminates the need for manual scanning, allowing IT professionals to focus on resolving critical issues.

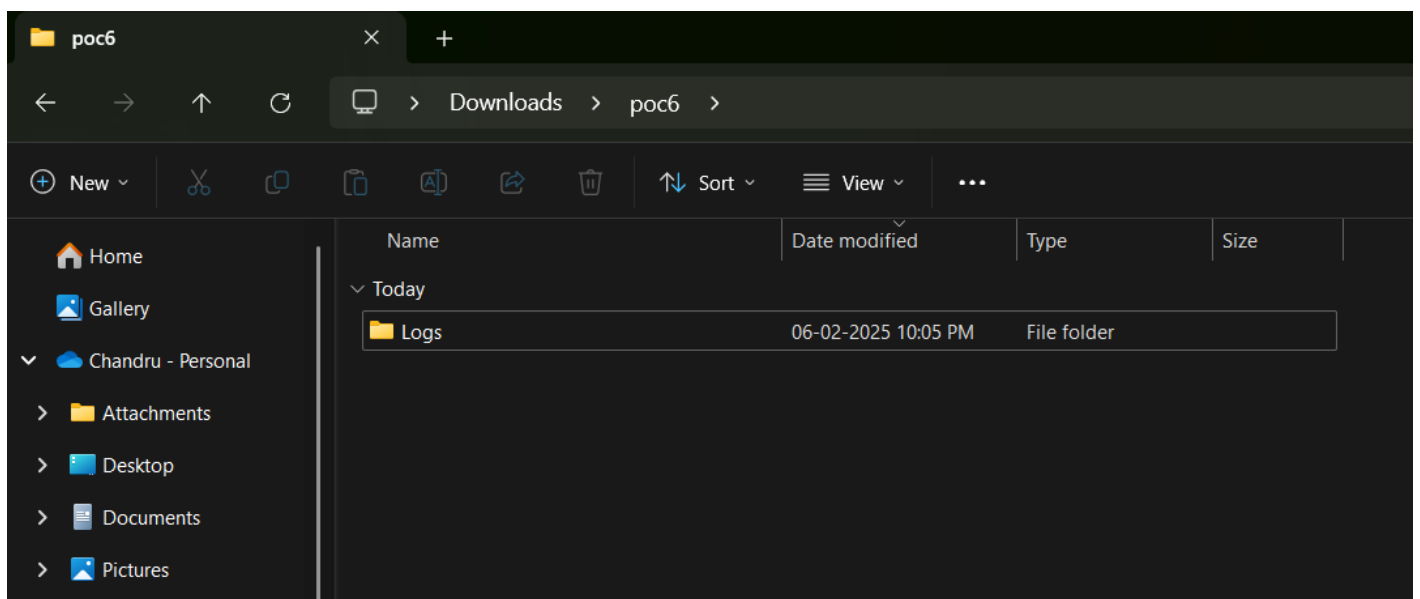
5. Scalability and Adaptability

The script can be extended to monitor multiple log files or customized for more complex use cases, serving as a foundation for advanced automation.

Step-by-step Overview

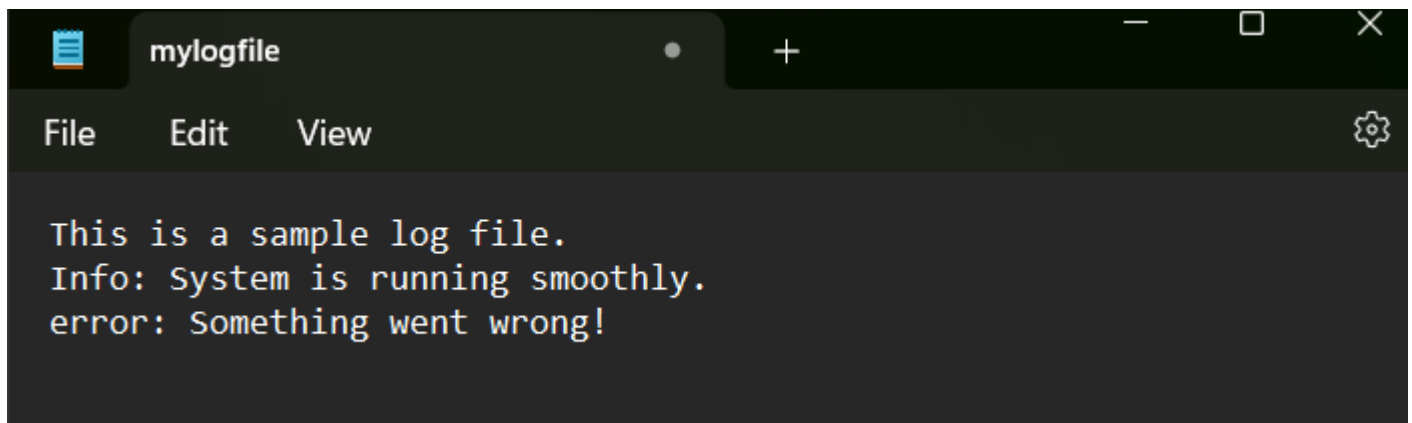
Step 1:

Create a folder named "logs" to store your log files and script.



Step 2:

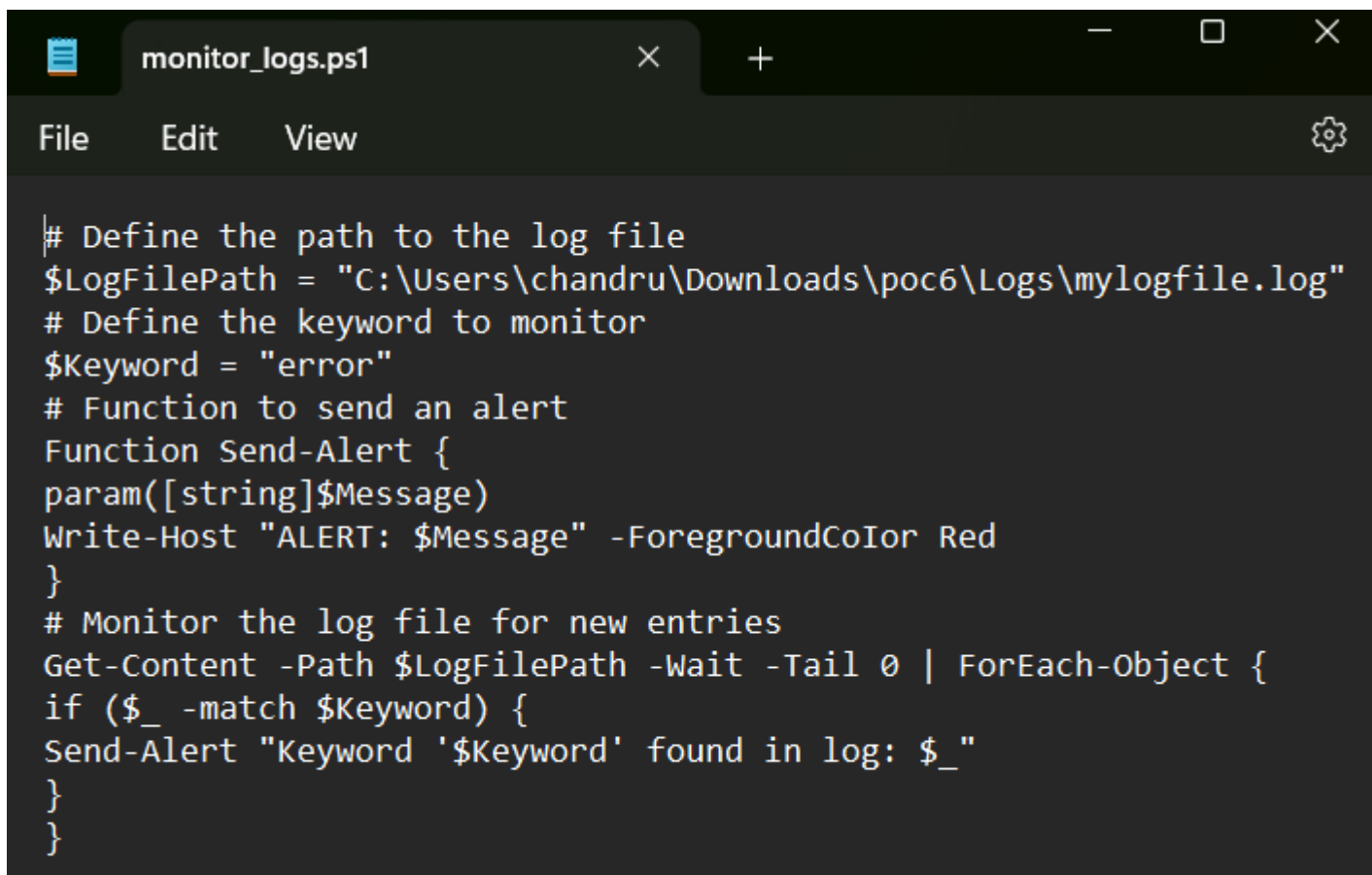
Open Notepad, add the following sample text, and save the file as mylogfile.log inside the logs folder.

A screenshot of a Notepad window titled 'mylogfile'. The window has a dark theme and a menu bar with 'File', 'Edit', and 'View'. The text inside the window is:

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

Step 3:

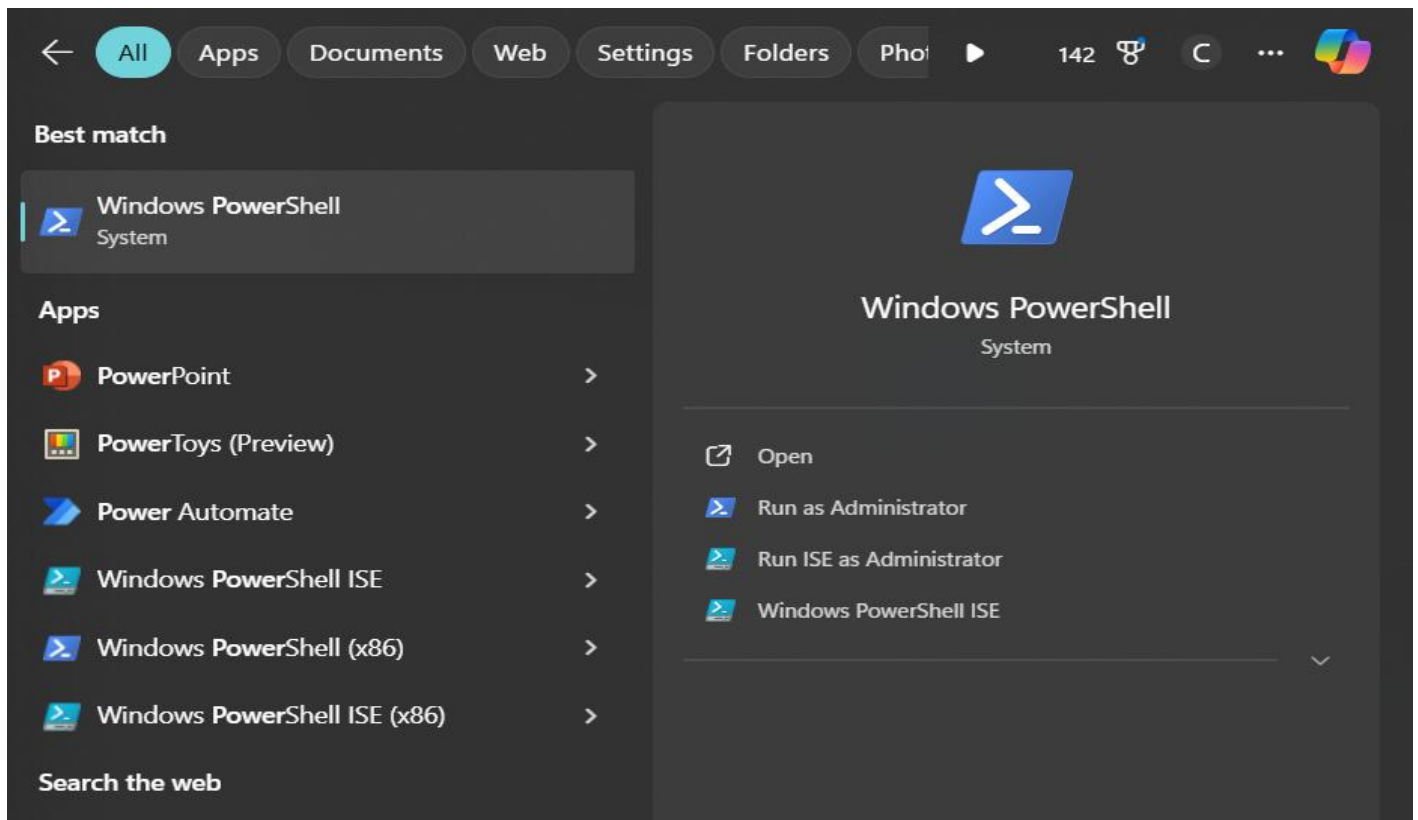
Open Notepad, type the PowerShell script, and set the \$LogFilePath to point to mylogfile.log inside the logs folder. Save the script as monitor_logs.ps1 in the same folder.

A screenshot of a Notepad window titled 'monitor_logs.ps1'. The window has a dark theme and a menu bar with 'File', 'Edit', and 'View'. The PowerShell script inside the window is:

```
# Define the path to the log file  
$LogFilePath = "C:\Users\chandru\Downloads\poc6\Logs\mylogfile.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  
}  
# 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:

Press the Windows Key, search for Windows PowerShell, and select Run as Administrator.



Step 5:

Enable script execution by running the following command in PowerShell:

Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned

When prompted, type **Y** and press Enter.

Step 6:

Navigate to the logs folder in PowerShell using the cd command.

```
PS C:\WINDOWS\system32> cd "C:\Users\chandru\Downloads\poc6\Logs"
```

Step 7:

Run the script:

.\monitor_logs.ps1

```
Select 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> cd "C:\Users\chandru\Downloads\poc6\Logs"
PS C:\Users\chandru\Downloads\poc6\Logs> .\monitor_logs.ps1
```

Step 8:

Open mylogfile.log in Notepad, add a new line containing the word "error", and save the file.

```
mylogfile

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 similar to:

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

```
Select 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> cd "C:\Users\chandru\Downloads\poc6\Logs"
PS C:\Users\chandru\Downloads\poc6\Logs> .\monitor_logs.ps1
ALERT: Keyword 'error' found in log: error: A new issue occurred! -ForegroundColor Red
```

Explanation

1. The script continuously monitors the log file in real time.
2. When a new line containing the keyword "error" is added, it immediately triggers an alert in PowerShell.

Outcome

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

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