

# 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: Keerthana Sri G

Department: ECE

# Introduction

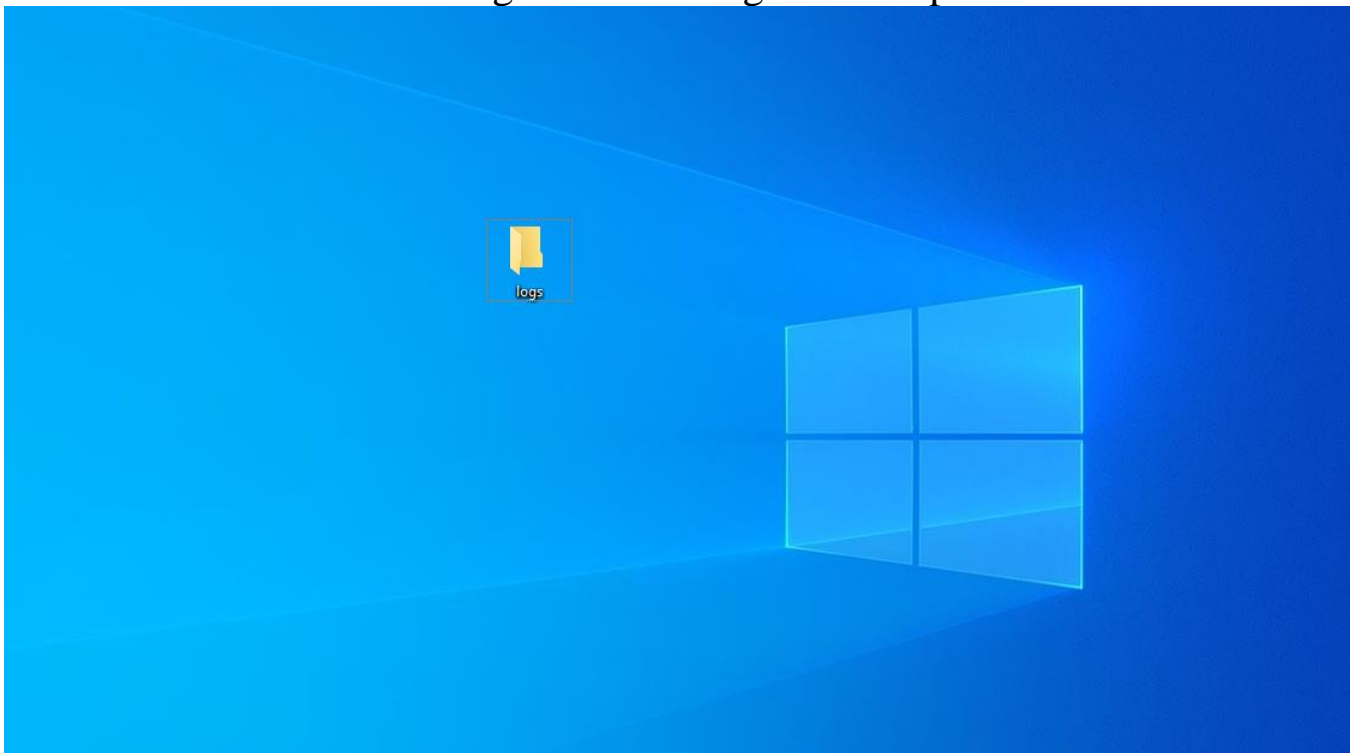
Log files play a critical role in IT systems, as they record activities and events generated by applications, servers, and network devices. Monitoring these logs helps identify issues such as errors, warnings, and suspicious activities that may require immediate attention. Automating the monitoring process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

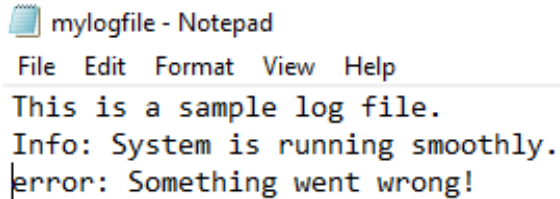
### Step 1:

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 window titled "mylogfile - Notepad". The window has a menu bar with "File", "Edit", "Format", "View", and "Help". The text inside the window is:  
This is a sample log file.  
Info: System is running smoothly.  
Error: Something went wrong!  
The text is displayed in a monospaced font, with "Info:" and "Error:" in blue and the rest in black.

## 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

```

monitor_logs - Notepad
File Edit Format View Help
#Define the path to the log file

$LogFilePath = "C:\Users\hp\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: ForEach-Object (

if ($match $Keyword) (

Send-Alert "Keyword '$Keyword' found in log: $"

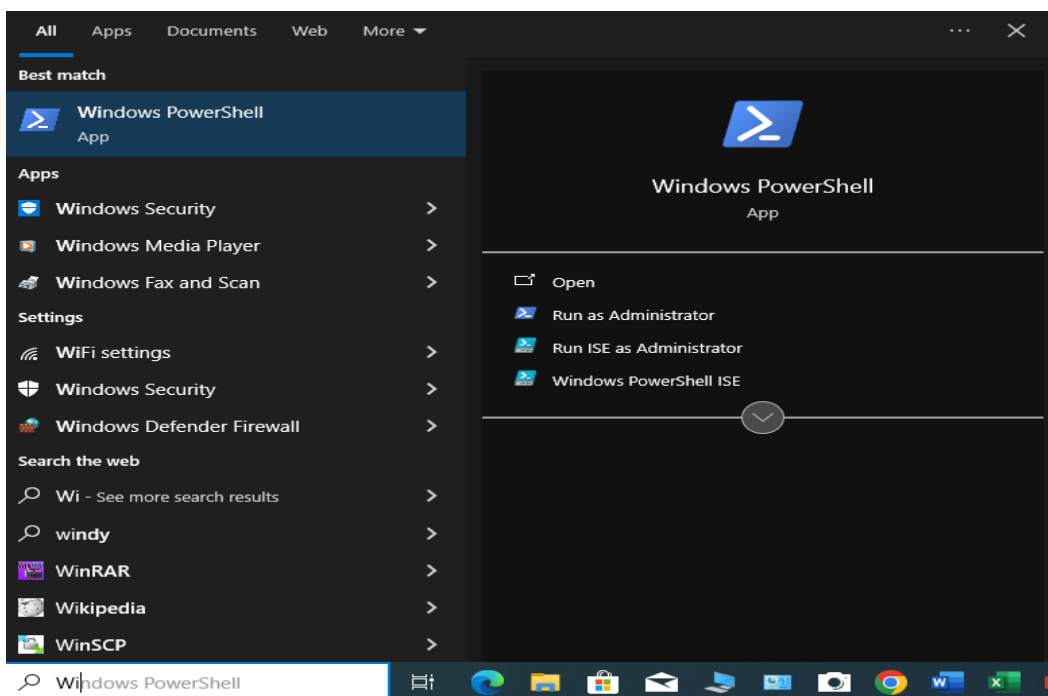
}

}

```

## Step 4:

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

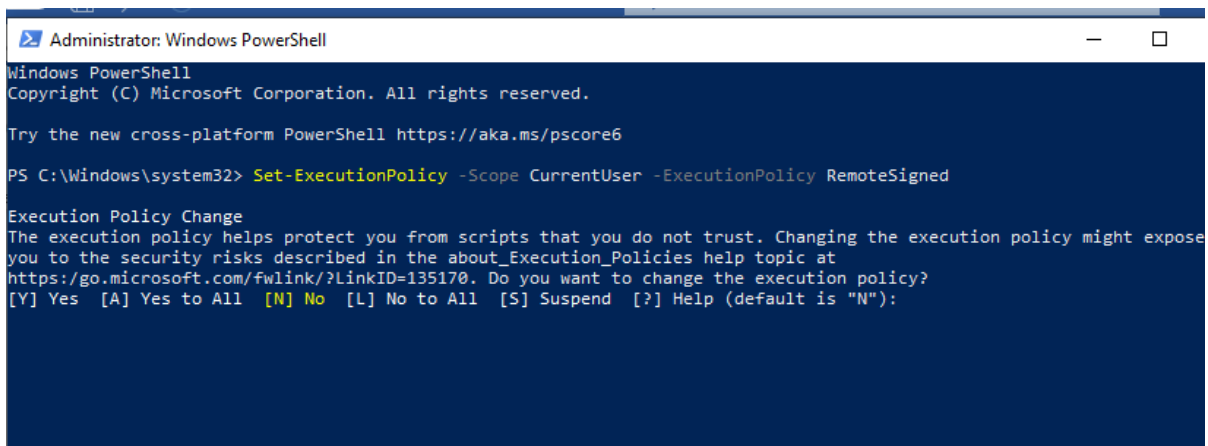


## Step 5:

Run the following command to allow script execution:

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

When prompted, type Y and press Enter.

A screenshot of a Windows PowerShell terminal window titled "Administrator: Windows PowerShell". The window has a dark blue background. The text inside shows the PowerShell prompt "PS C:\Windows\system32>" followed by the command "Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned". Below the command, a message appears: "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?". At the bottom of the message, it says "[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is 'N'):". The letter 'Y' is highlighted in yellow, indicating it has been entered as input.

## Step 6:

Navigate to the logs folder

```
PS C:\Windows\system32> C:\Users\hp\Desktop\logs
```

## Step 7:

Run the script:

**.\monitor\_logs.ps1**

```
PS C:\Windows\system32> .\monitor_logs.ps1
```

## Step 8:

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



```
*mylogfile - Notepad
File Edit Format View Help
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.