



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



Placement Empowerment Program Cloud Computing and DevOps Centre

Write a Shell Script to Monitor Logs

Name: Shalini D

Department: IT



Introduction

In Unix-like systems, logs are used to record system activities and application events. Monitoring these logs can help in troubleshooting, performance tuning, and ensuring system security. A Shell script can be used to automate the process of monitoring logs and alerting when specific events occur.

Overview

The shell script continuously monitors a specified log file in real time and detects error messages such as "ERROR," "CRITICAL," or "FAIL." Upon detecting an error, the script logs the occurrence and optionally sends an email notification to the system administrator.

Features

1. Monitors logs in real-time
2. Detects specific error keywords
3. Logs detected errors to a separate alert file

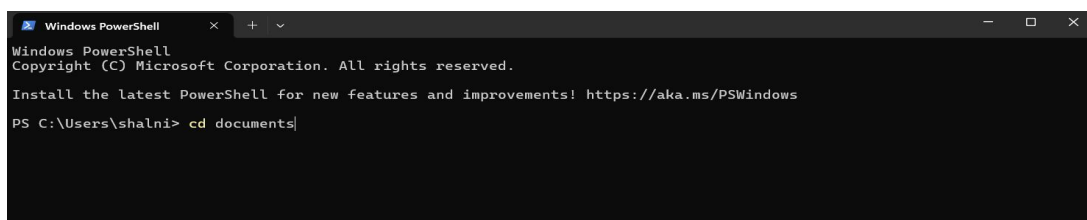
4. Optionally sends email notifications
5. Can run in the background continuously.

Steps to Implement

Step 1:

Create the Shell Script

1. Open a terminal and navigate to the directory where you want to create the script:

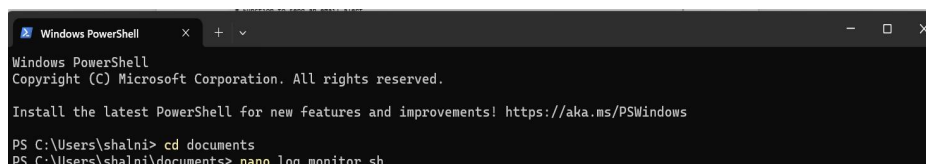


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\shalni> cd documents|
```

2.Create a new shell script file:

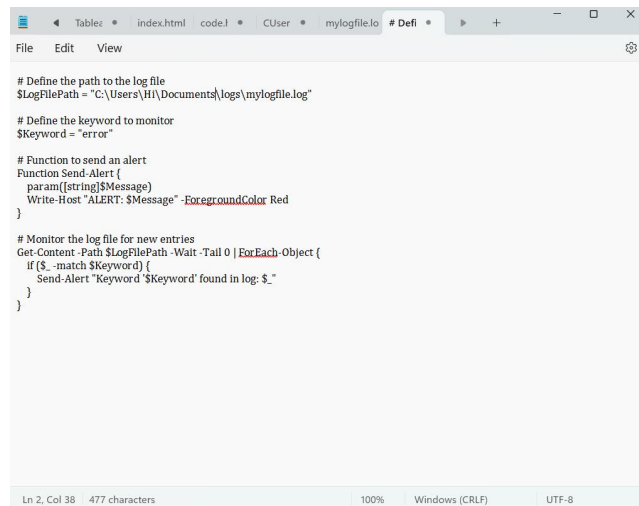


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\shalni> cd documents
PS C:\Users\shalni\documents> nano log_monitor.sh
```

3.Create and save a document file in your documents path.



```
# Define the path to the log file
$LogFilePath = "C:\Users\HI\Documents\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: $_"
    }
}
```

4. Save and exit:

- Press CTRL + X
- Press Y and then Enter

Step 2:

Grant Execution Permission

Make the script executable:



```
PS C:\Users\shalni\documents> chmod +x log_monitor.sh
```

Step 3:

Run the Script

Execute the script to start monitoring logs:



```
PS C:\Users\shalni> ./log_monitor.sh
```

Expected Output

The expected outcome of running the shell script is that it will continuously display any new lines appended to the specified log file in real-time. If the log file does not exist, the script will alert you and exit.

Conclusion

This shell script automates log monitoring, helping system administrators detect critical issues promptly. By customizing it further, you can enhance server security and reliability.