**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

# Write a shell script to moniter logs

**Name: Monika K** **Department:ADS**

**Introduction and Overview**

In this POC, we will learn how to monitor logs in real time using a shell script. This process involves creating a script that continuously monitors logs, identifies errors or anomalies, and alerts users accordingly. By following these steps, you'll gain hands-on experience in log management, a crucial skill for system administration and DevOps.

**Objective**

**The goal of this project is to:**

1. Set up a shell script to monitor log files.
2. Detect and report specific log patterns (e.g., errors, warnings).
3. Automate log monitoring for proactive issue resolution.

**Importance of Log Monitoring**

Log monitoring is essential for maintaining system stability, security, and performance. It offers several advantages, such as:

- Proactive Issue Detection: Identify errors and issues before they escalate.

- Security Monitoring: Detect unauthorized access or security threats.
- Performance Insights: Analyze logs for system optimization.

**Step-by-Step Overview**

**Step 1:** Identify Log Files to Monitor Determine which log files are crucial for monitoring, such as /var/log/syslog, /var/log/auth.log, or application-specific logs.

**Step 2:** Create a Log Monitoring Script Create a shell script to monitor logs in real time.

**log_monitor.sh:**

```
#!/bin/bash


# Define log file to monitor
LOG_FILE="/var/log/syslog"


# Keywords to watch for
KEYWORDS="error|failed|critical"


# Monitor logs in real time using tail
```

```
tail -F "$LOG_FILE" | while read LINE
 do
    echo "$LINE" | grep -E "$KEYWORDS" && echo "ALERT: Found keyword in logs!"
done
```

**Step 3:** Make the Script Executable Run the following command to make the script executable:

```
chmod +x log_monitor.sh
```

**Step 4:** Execute the Script To start monitoring logs in real time, run:

```
./log_monitor.sh
```

**Step 5:** Automate Script Execution To ensure the script runs continuously, consider adding it to a cron job or system service.

To run every minute via cron:

```
* * * * * /path/to/log_monitor.sh
```

This ensures continuous log monitoring and real-time alerts for critical events.