# Project Overview:

Develop a comprehensive shell script for sysops to automate system monitoring and generate detailed reports. The script will leverage advanced Linux shell scripting techniques to monitor system metrics, capture logs, and provide actionable insights for system administrators.

**# Script Initialization:**

Initialize script with necessary variables and configurations.

Validate required commands and utilities availability.

**# System Metrics Collection:**

Monitor CPU usage, memory utilization, disk space, and network statistics.

(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/" | awk '{print 100 - $1}')

(df -h | awk '$NF=="/"{printf "%s", $5}')

Capture process information including top processes consuming resources.

(free | grep Mem | awk '{print $3/$2 \* 100.0}')

## **# Log Analysis:**

Parse system logs (e.g., syslog) for critical events and errors.

(tail -n 200 /var/log/syslog | grep -iE 'error|critical')

Generate summaries of recent log entries based on severity.

(tail -n 20 /var/log/syslog)

## **# Health Checks:**

Check the status of essential services (e.g., Nginx, MySQL).

systemct status nginx

Verify connectivity to external services or databases.

## **# Alerting Mechanism:**

Implement thresholds for critical metrics (CPU, memory) triggering alerts.

Send email notifications to sysadmins with critical alerts.

## **# Report Generation:**

Compile all collected data into a detailed report.

Include graphs or visual representations where applicable.

## **# Automation and Scheduling:**

Configure the script to run periodically via cron for automated monitoring. Edit crontab file

crontab –e

Add following content in it

\* \*/1 \* \* \* /home/einfochips/Desktop/SP/Training/Day10/sysops.sh

check list of crontab list

crontab –l

Ensure the script can handle both interactive and non-interactive execution modes.

## **# User Interaction:**

Provide options for interactive mode to allow sysadmins to manually trigger checks or view specific metrics.

Ensure the script is user-friendly with clear prompts and outputs.

## **# Documentation:**

Create a README file detailing script usage, prerequisites, and customization options.

Include examples of typical outputs and how to interpret them.

# **# Solution:**

Create Script File

nano system\_ops.sh

Add Following Code in it

#!/bin/bash

REPORTDIR="/home/einfochips/Desktop/SP/Training/Day10/"

ALERT\_THRESHOLD\_CPU=75

ALERT\_THRESHOLD\_MEM=40

SERVICE\_STATUS=("nginx" "mysql")

EXTERNAL\_SERVICES=("google.com" "mysql://db.test.com")

CPU\_USAGE=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/")

MEMORY\_USAGE=$(free | awk '/Mem/{printf("%.2f"), $3/$2\*100}')

System\_Metrics() {

echo -e "\n CPU Usage: "

echo "CPU\_USAGE=$(top -bn1 | grep "Cpu(s)" | sed "s/.\*, \*\([0-9.]\*\)%\* id.\*/\1/")"

echo "MEMORY\_USAGE=$(free | awk '/Mem/{printf("%.2f"), $3/$2\*100}')"

echo "DISK\_SPACE=$(df -h / | awk '/\//{print $(NF-1)}')"

echo -e "\n Network Statistics: "

echo "NETWORK\_STATS=$(netstat -i)"

echo -e "\n Top Processes: "

echo "TOP\_PROCESSES=$(top -bn 1 | head -n 10)"

}

Log\_analysis() {

echo -e "\Recent Critical Events: "

echo "CRITICAL\_EVENTS=$(tail -n 200 /var/log/syslog | grep -iE 'error|critical')"

echo -e "\n Recent Logs: "

echo "RECENT\_LOGS=$(tail -n 20 /var/log/syslog)"

}

Health\_check() {

echo -e "\n Service Status: "

for service in "${SERVICE\_STATUS[@]}"; do

systemctl is-active --quiet "$service"

if [ $? -eq 0 ]; then

echo " $service is running."

else

echo " Alert: $service is not running."

fi

done

echo -e "\n Connectivity Check: "

for service in "${EXTERNAL\_SERVICES[@]}"; do

ping -c 1 "$service" >/dev/null 2>&1

if [ $? -eq 0 ]; then

echo " Connectivity to $service is okay."

else

echo " Alert: Unable to connect to $service."

fi

done

}

if (( $(echo "$CPU\_USAGE >= $ALERT\_THRESHOLD\_CPU" | bc -l) )); then

echo "High CPU Usage Alert: $CPU\_USAGE%"

fi

if (( $(echo "$MEMORY\_USAGE >= $ALERT\_THRESHOLD\_MEM" | bc -l) )); then

echo "High Memory Usage Alert: $MEMORY\_USAGE%"

fi

mkdir -p "$REPORTDIR"

REPORTFILE="$REPORTDIR/sysreport\_$(date +'%Y-%m-%d\_%H-%M-%S').txt"

echo "System Report $(date)" >> "$REPORTFILE"

System\_Metrics >> "$REPORTFILE"

Log\_analysis>> "$REPORTFILE"

Health\_check >> "$REPORTFILE"

echo "Select an option:

1. Check system metrics

2. View logs

3. Check service status

4. Exit"

read choice

case $choice in

1) System\_Metrics

;;

2) Log\_analysis

;;

3) Health\_check

;;

4) exit ;;

\*) echo "Invalid option";;

esac

Give neccessory permissions to File for execution

chmod 755 system\_ops.sh

Run the script using following command

./system\_ops.sh

After run the script it will create report file which will store all info like CPU usage, memory usage and all.

