Monitoring and alerts shell script

Problem Statement:
Write a shell script to: Monitor server resource usage (CPU, memory, disk).Send an email alert if usage exceeds a threshold.

Step 1: Creating a script that shows cpu, memory, disk usage in percent

For cpu we make use of top(for resource usage visualization) command, for memory and disk we use dd command(it is for convert and copy file)

```
#!/bin/bash# Thresholds

CPU_THRESHOLD=80    # CPU usage percentage
MEM_THRESHOLD=80    # Memory usage percentage
DISK_THRESHOLD=90   # Disk usage percentage
EMAIL="csborle@gmail.com"
LOG_FILE="/var/log/auth.log"  # Log file location# Function to write logs
log_message() {
  local message="$1"
  echo "$(date '+%Y-%m-%d %H:%M:%S') - $message" >> "$LOG_FILE"
}# Function to check CPU usage
check_cpu_usage() {
  CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}')
  CPU_USAGE=${CPU_USAGE%.*}  # Convert to integer
  if (( CPU_USAGE > CPU_THRESHOLD )); then
    log_message "High CPU usage: $CPU_USAGE%"
    echo "High CPU usage: $CPU_USAGE%" | mail -s "Alert: CPU Usage Exceeded"
"$EMAIL"
  fi
}# Function to check memory usage
check_memory_usage() {
  MEM_TOTAL=$(free | grep Mem: | awk '{print $2}')
  MEM_USED=$(free | grep Mem: | awk '{print $3}')
  MEM_USAGE=$(( MEM_USED * 100 / MEM_TOTAL ))  # Percentage
  if (( MEM_USAGE > MEM_THRESHOLD )); then
    log_message "High Memory usage: $MEM_USAGE%"
    echo "High Memory usage: $MEM_USAGE%" | mail -s "Alert: Memory Usage
Exceeded" "$EMAIL"
  fi
}# Function to check disk usage
```

```
check_disk_usage() {
  DISK_USAGE=$(df -h / | grep / | awk '{print $5}' | sed 's/%//')  # Root
partition
  if (( DISK_USAGE > DISK_THRESHOLD )); then
    log_message "High Disk usage: $DISK_USAGE%"
    echo "High Disk usage: $DISK_USAGE%" | mail -s "Alert: Disk Usage
Exceeded" "$EMAIL"
  fi
}# Main monitoring function
monitor_resources() {
  log_message "Starting resource monitoring..."
  check_cpu_usage
  check_memory_usage
  check_disk_usage
  log_message "Resource monitoring completed."
}# Create log file if it doesn't exist
if [ ! -f "$LOG_FILE" ]; then
  touch "$LOG_FILE"
  chmod 644 "$LOG_FILE"
fi# Execute the monitoring function
monitor_resources
```

Step 2: Configuring the mail server

For sending emails we require a mail server so for that we will install mail utility package that is mail utils

```
sudo apt update
sudo apt install mailutils
```

after installing mailutils it will ask for postfix configuration select internet site(Mail is sent and received directly using SMTP) and click ok in next step click ok again

Postfix is MTA(Mail Transfer Agent) which basically determines the routes and sends emails

Now go to manage your google account option present in google profile enable two-factor authentication and then search for app password and create a password and note it down.

In the next step create mail.rc and enter the following configurations

```
set smtp-auth-user=csborle@gmail.com
set smtp-auth-password=aaaa bbbb cccc dddd
set from="csborle@gmail.com"
set realname="Chandrakant"
```

Now go to postfix folder

```
cd /etc/postfix/
```

```
sudo vim main.cf
```

and add the following lines

```
relayhost = [smtp.gmail.com]:587
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous smtp_tls_security_level = encrypt
smtp_tls_CAfile = /etc/ssl/certs/ca-certificates.crt
```

Configure SASL Authentication
Create a file for your Gmail credentials:

`sudo nano /etc/postfix/sasl_passwd`

Add the following:

```
[smtp.gmail.com]:587 your-email@gmail.com:your-app-password
```

Secure the credentials file:
`sudo chmod 600 /etc/postfix/sasl_passwd`

Generate a Postfix database map:
`sudo postmap /etc/postfix/sasl_passwd`

Restart Postfix service:
`sudo systemctl restart postfix`

Test if sending email through this service works
`echo "This is a test email" | mail -s "Test Email" csborle@gmail.com`

Now schedule it using cronjob to run the script for every 5 minutes

```
crontab -e
```

add the following line.(here we gave the path to our script file)

```
*/5 * * * * /home/ubuntu/alert.sh
```

Step 3: Simulating the stress

now to simulate the stress for cpu we install a linux package called stress-ng

```
stress-ng --cpu 1 --cpu-load 80 --timeout 600
```

this will simulate the cpu load for 10 minutes

for disk load

```
dd if=/dev/zero of=/tmp/testfile bs=1M count=500
```

To remove - `rm /tmp/testfile`

for memory load

```
dd if=/dev/zero of=/dev/null bs=1M count=5000 &
```

To remove - `kill $(pgrep dd)`

Now whenever there is stress on cpu or memory or disk a notification will be sent through email
For example, similar to below ones

Alert: CPU Usage Exceeded

Inbox ×

root <csborle@gmail.com>
to me

High CPU usage: 90%

Reply Forward

Alert: Disk Usage Exceeded

Inbox ×

root <csborle@gmail.com>
to me

High Disk usage: 62%

Reply Forward

Alert: Memory Usage Exceeded

Inbox ×

root <csborle@gmail.com>
to me

High Memory usage: 57%

Reply Forward

```
- created by: Chandrakant
```