Server Resource Monitoring

Overview

This guide explains how to set up automated server resource monitoring using a shell script. The script monitors CPU, memory, and disk usage, sending email alerts when usage exceeds defined thresholds.

Prerequisites

- Postfix mail server

- Gmail account with App Password configured

- Basic understanding of shell scripting

- Root or sudo access to the server

Resource Monitoring Script:

Script defines alerts when CPU usage exceeds threshold, Memory Usage: Monitors RAM utilization, Disk Usage: Tracks storage space on specified mount points, Logging: Maintains detailed log of all checks and alerts

```
root@ip-172-31-21-126:~# cat monitor_script.sh
#!/bin/bash

# Thresholds for resource usage (adjust as needed)
CPU_THRESHOLD=40    # Percentage
MEMORY_THRESHOLD=20 # Percentage
DISK_THRESHOLD=10  # Percentage
EMAIL="tejalbkale@gmail.com"

#alert messages
alert_message=""

send_alert() {
        local subject=$1
        local message=$2
        echo "$message" | mailx -s "$subject" "$EMAIL"
}

# 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
        alert_message+="CPU usage has reached: ${CPU_USAGE}%."
    fi
}

# Function to check memory usage
check_memory_usage() {
    MEMORY_USAGE=$(free | awk '/Mem:/ {printf "%.0f", $3/$2 * 100.0}')
    if (( "$MEMORY_USAGE" > "$MEMORY_THRESHOLD" )); then
        alert_message+="Memory usage has reached: ${MEMORY_USAGE}%."
    fi
```

```
# Function to check memory usage
check_memory_usage() {
    MEMORY_USAGE=$(free | awk '/Mem:/ {printf "%.0f", $3/$2 * 100.0}')
    if (( "$MEMORY_USAGE" > "$MEMORY_THRESHOLD" )); then
        alert_message+="Memory usage has reached: ${MEMORY_USAGE}%."
    fi
}

# Function to check disk usage
check_disk_usage() {
    while read -r line; do
        USAGE=$(echo "$line" | awk '{print $1}')
        PARTITION=$(echo "$line" | awk '{print $2}')
        if (( "$USAGE" > "$DISK_THRESHOLD" )); then
            alert_message+="Disk usage is high on $PARTITION: ${USAGE}%."
        fi
    done < <(df -h --output=pcent,target | tail -n +2 | awk '{print $1 " " $2}' | sed 's/%//')
}



# Send email if there are alerts
if [ -n "$alert_message" ]; then
   send_alert "Usage alert" "$alert_message"
fi

root@ip-172-31-21-126:~#
```

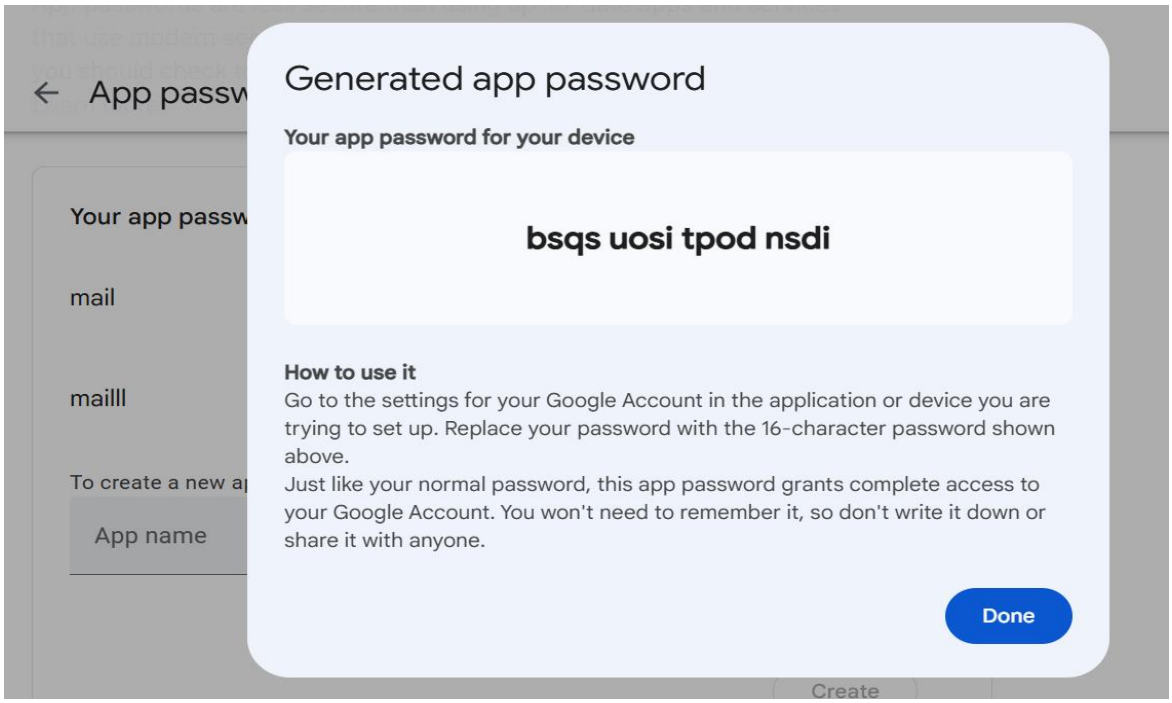
Set Up Mail Server

1. Install Required Packages

- sudo apt-get update
- sudo apt-get install postfix mailutils

2. Configure Gmail App Password

- Go to Google Account settings
- Enable 2-Factor Authentication
- Generate App Password under Security settings
- Save the generated password



3. Configure Postfix for Gmail

- sudo nano /etc/postfix/main.cf

Add/modify these lines:

```
relayhost = [smtp.gmail.com]:587
smtp_use_tls = yes
smtp_sasl_auth_enable = yes
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtp_sasl_security_options = noanonymous
```

4. Create SASL Password File

- sudo nano /etc/postfix/sasl_passwd
- [smtp.gmail.com]:587 your-email@gmail.com:your-app-password

5. Secure and Update Postfix Configuration

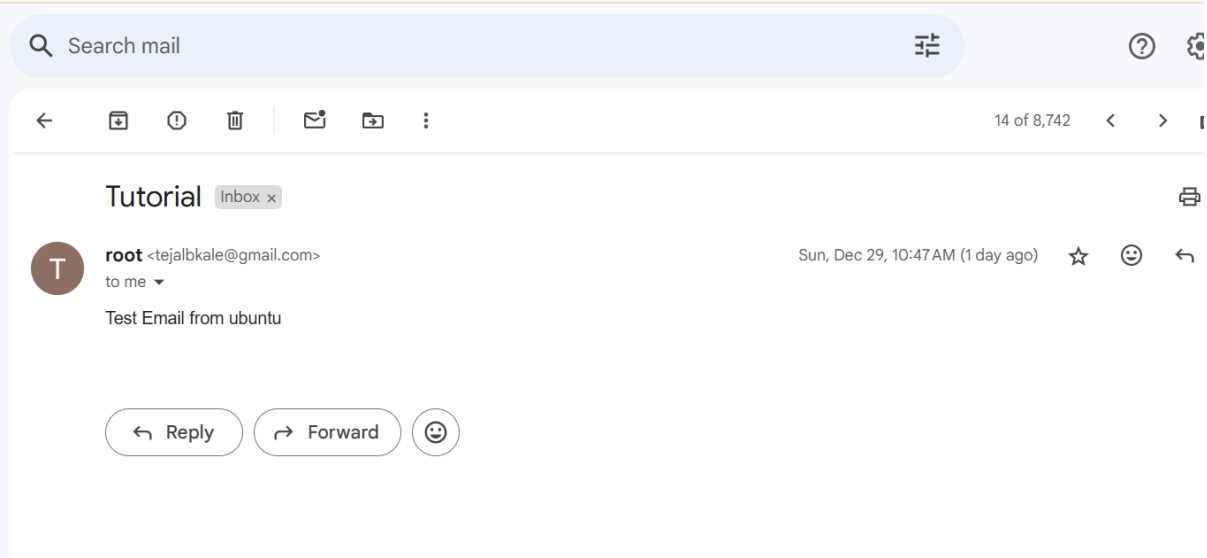
- sudo chmod 600 /etc/postfix/sasl_passwd
- sudo systemctl restart postfix

6. Set Up Cron Job

- sudo crontab -e
- */5 * * * * /usr/local/bin/server-monitor.sh
- This runs the script every 5 minutes.

7. Test Email Configuration

echo "Test Email" | mail -s "Test Subject" your-email@gmail.com



8. Testing

You can simulate high resource usage using the `stress` command:

Install stress tool

- sudo apt-get install stress

9. Test CPU stress

- stress --cpu 8 --timeout 60s