



Expert Cloud Consulting

Enhance Optimise & Scale

ASCP GPUonCLOUD Pvt Ltd

“Expert Cloud Consulting” -

Documentation for Resource Usage Monitoring and Email Notification Using Shell Script on Linux VM [Title,18, Arial]

07.May.2025 [Subtitle,14, Arial]

version 1.0

—

Contributed by Yogiraj Deshpande [Normal text,14, Arial]

Approved by Akshay Shinde(In Review)

Expert Cloud Consulting

Office #811, Gera Imperium Rise,

Hinjewadi Phase-II Rd, Pune, India – 411057

Week 4: Scripting Basics:

Topics :

- Introduction to Shell scripting and Python scripting.
- Automating basic tasks: file handling, user management.

Assignments:

- Write a shell script to:
 - Monitor server resource usage (CPU, memory, disk).
 - Send an email alert if usage exceeds a threshold.
- Create a Python script to:
 - Parse logs from a web server.
 - Identify IPs causing the most failed login attempts.
 - Block those IPs by dynamically updating firewall rules.

Resources:

- [Shell scripting tutorial](#)
- [Python for Beginners](#)



“Expert Cloud Consulting”

Documentation for Resource Usage Monitoring and Email Notification Using Shell Script on Linux VM

[Title,18, Arial]

1.0: Document Overview:

This document describes the implementation of a Linux shell script that monitors server resource usage and sends email alerts when predefined thresholds are exceeded. The alert system uses Postfix configured with Gmail SMTP to notify system administrators via email.

2.0: Objective:

- Monitor server resources: CPU, memory, and disk usage.
- Set up automated email alerts if any resource usage exceeds a critical threshold.
- Schedule the script to run automatically at regular intervals using `cron`.

3.0: Prerequisites:

1. Ubuntu server with internet access.
2. A valid Gmail account with App Password (for use with Postfix).
3. `mailutils` and `postfix` installed on the system.



4.0: Step-by-Step Implementation:

4.1: Install Mail Utilities:

Steps:

1. Launch an EC2 Instance with Ubuntu.
2. SSH into the instance:

```
ssh -i your-key.pem ubuntu@your-ec2-public-ip
```

3. Install Mail Utilities:

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

Reason: These packages are needed to send email alerts from the script using Gmail SMTP.

4.2: Configure Postfix for Gmail SMTP:

- Edit the Postfix main configuration file:

```
sudo vim /etc/postfix/main.cf
```

- Add the following at the end:

```
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  
header_size_limit = 4096000
```

Reason: This configures Postfix to relay email using Gmail securely.

4.3: Provide Gmail SMTP Credentials:

```
sudo vim /etc/postfix/sasl_passwd
```

- Add:

```
[smtp.gmail.com]:587    your_gmail@gmail.com:your_app_password
```

- Then hash the password file and secure permissions:

```
sudo postmap /etc/postfix/sasl_passwd
sudo chmod 600 /etc/postfix/sasl_passwd /etc/postfix/sasl_passwd.db
```

Reason: postmap converts the password file to a format Postfix uses securely.

4.4: Create the Monitoring Shell Script:

```
vim monitor.sh
```

- Paste the script:

```
#!/bin/bash

# Thresholds
CPU_THRESHOLD=70
MEM_THRESHOLD=70
DISK_THRESHOLD=80

# Email settings
EMAIL_TO="csborle@gmail.com yogirajdeshpande02@gmail.com"
EMAIL_SUBJECT="Server Resource Alert"

# Greeting message
greeting="Hey,\n\nHere is your current server resource usage
alert:\n\n"
alerts=""

# Check CPU usage
CPU_USAGE=$(top -bn1 | grep "Cpu(s)" | awk '{print 100 - $8}' | cut
-d. -f1)

# Check Memory usage
MEM_USAGE=$(free | grep Mem | awk '{print $3/$2 * 100.0}' | cut -d.
-f1)

# Check Disk usage
DISK_USAGE=$(df / | grep / | awk '{ print $5 }' | sed s/%//g)

# Create the alert message
if [ "$CPU_USAGE" -gt "$CPU_THRESHOLD" ]; then
    alerts+="CPU usage is $CPU_USAGE% (threshold: $CPU_THRESHOLD%)\n"
fi

if [ "$MEM_USAGE" -gt "$MEM_THRESHOLD" ]; then
    alerts+="Memory usage is $MEM_USAGE% (threshold:
$MEM_THRESHOLD%)\n"
fi
```

```

if [ "$DISK_USAGE" -gt "$DISK_THRESHOLD" ]; then
    alerts+="Disk usage is $DISK_USAGE% (threshold:
$DISK_THRESHOLD%)\n"
fi

# Send the email if there is an alert
if [ -n "$alerts" ]; then
    final_message="${greeting}${alerts}"
    echo -e "$final_message" | mail -s "$EMAIL_SUBJECT" $EMAIL_TO
fi

```

- Make it executable:

```

chmod +x monitor.sh

```

4.5: Test the Script:

```

bash -x monitor.sh

```

Reason: Ensures the script and email alerts are functioning correctly before automation.

4.6: Automate the Script with Cron:

- Edit the crontab:

```

crontab -e

```

- Add the following line::

```

*/5 * * * * /home/ubuntu/monitor.sh

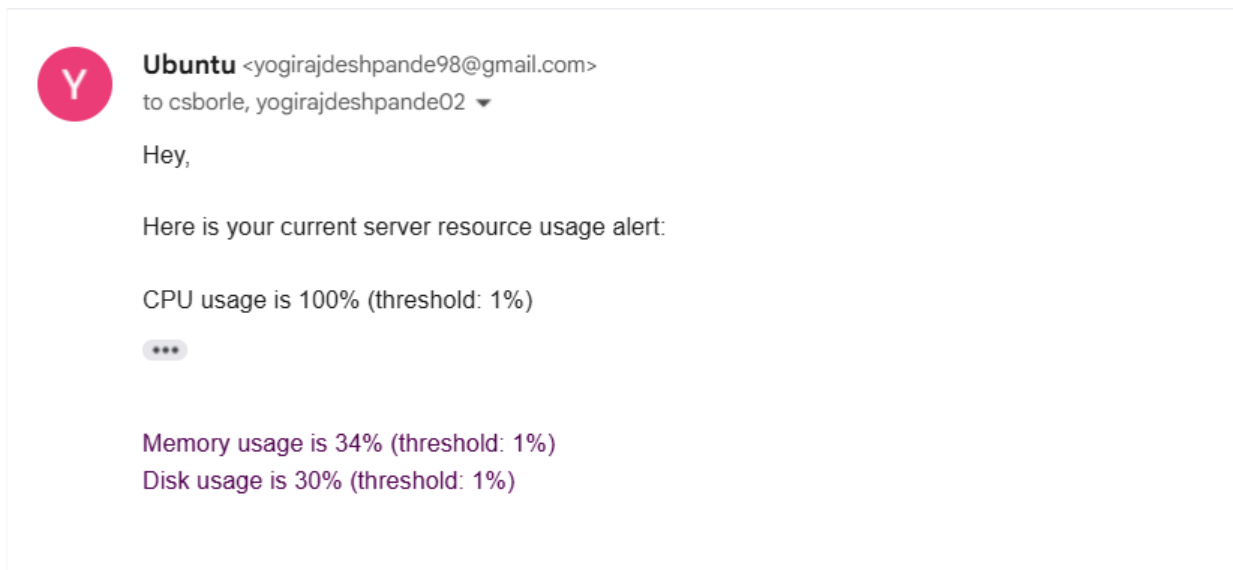
```

Reason: This schedules the script to run every 5 minutes and automatically monitors system health.

5.0: Result and Outcome:

- When resource usage exceeds the defined thresholds, an email is sent immediately.
- The subject line of the email is: **Server Resource Alert**.
- The email body details CPU, memory, and disk usage with threshold comparisons.

6.0: Sample Output (Email):



7.0: Security Considerations:

- Always use an App Password instead of your main Gmail password.
- Restrict file permissions to protect sensitive credentials.
- Review crontab and logs periodically to ensure stability.

7.0: Conclusion:

This implementation provides a lightweight yet effective server monitoring system tailored for Linux-based virtual machines. By combining shell scripting with Postfix and Gmail SMTP, administrators can proactively monitor CPU, memory, and disk usage and receive timely alerts via email when usage exceeds defined thresholds.