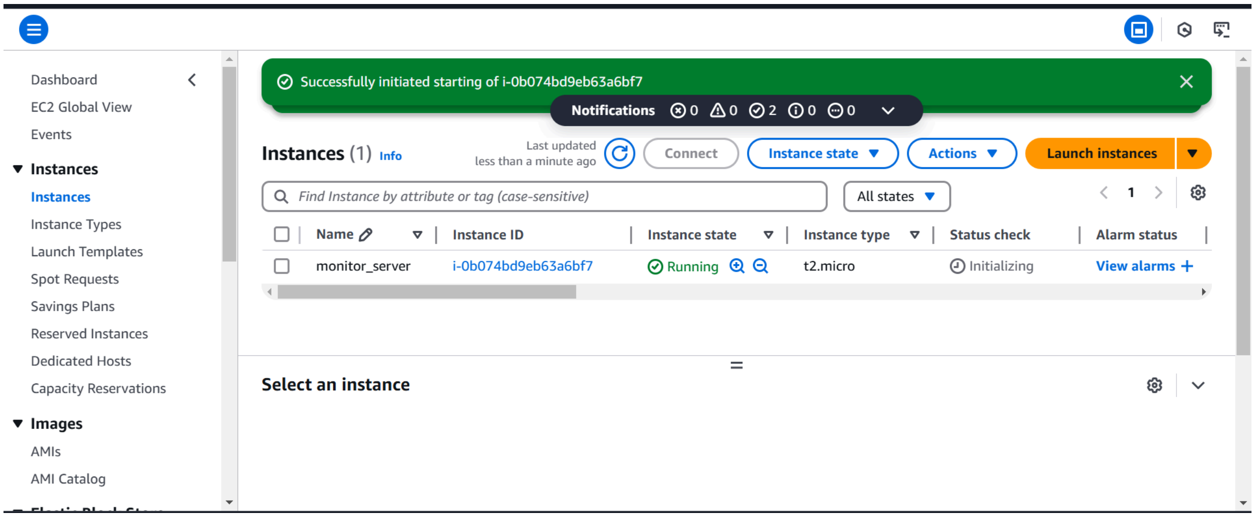
Documentation: Shell Script to Monitor CPU, Memory, Disk Space Utilization

Step #1:Launch the new ec2 instance

Go to AWS account and launch new ubuntu instance



Successfully initiated starting of i-0b074bd9eb63a6bf7

Instances (1) Info

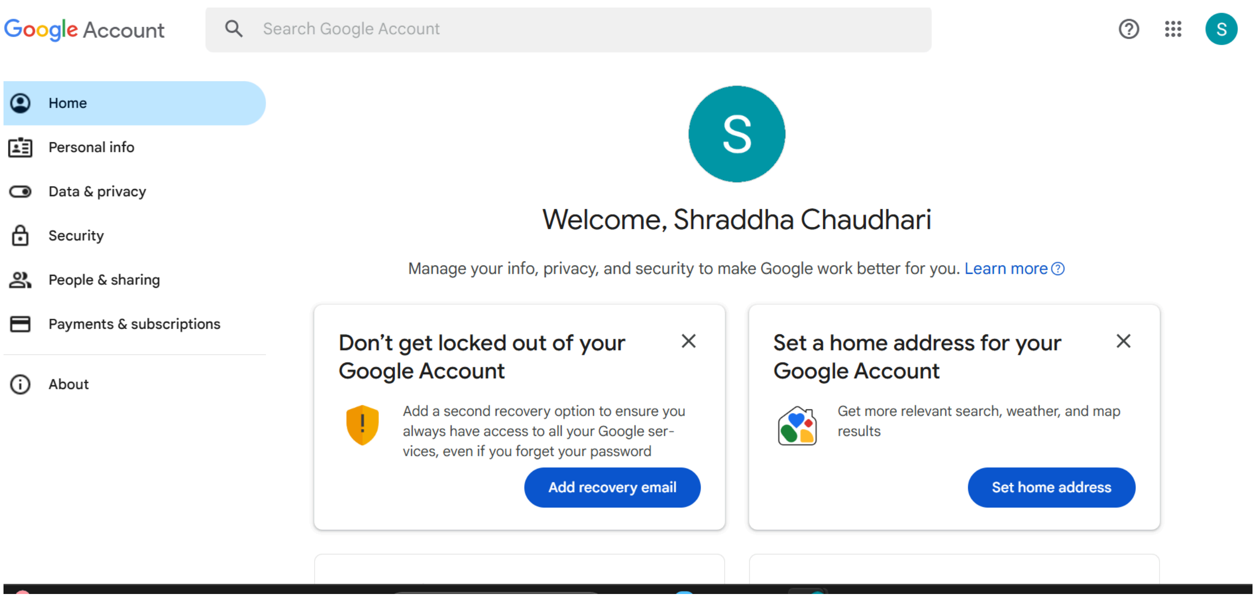
Last updated
less than a minute ago

Find Instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
monitor_server	i-0b074bd9eb63a6bf7	Running	t2.micro	Initializing	View alarms +

Step #2:Generate the App Password

First go to your Google account.



Welcome, Shraddha Chaudhari

Manage your info, privacy, and security to make Google work better for you. Learn more

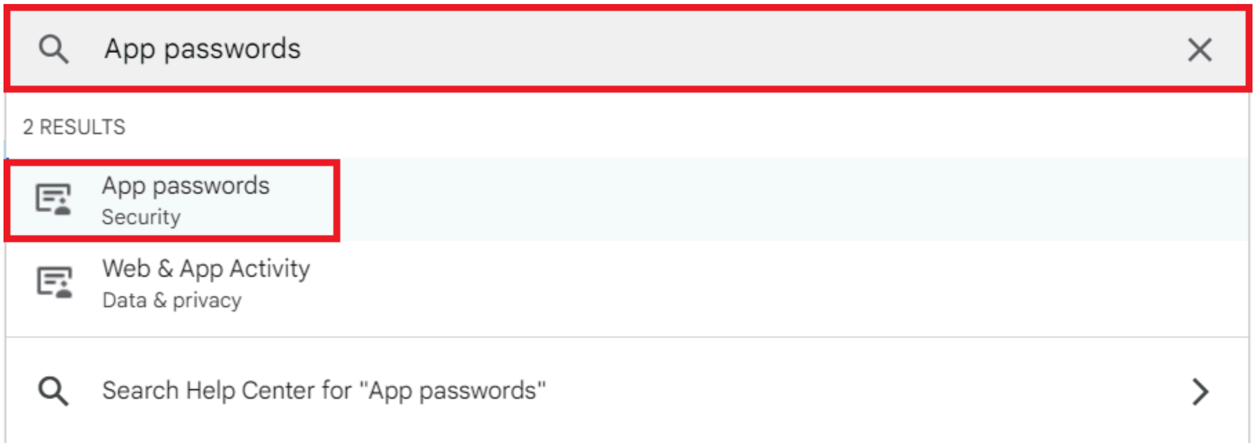
Don't get locked out of your Google Account

Add a second recovery option to ensure you always have access to all your Google services, even if you forget your password

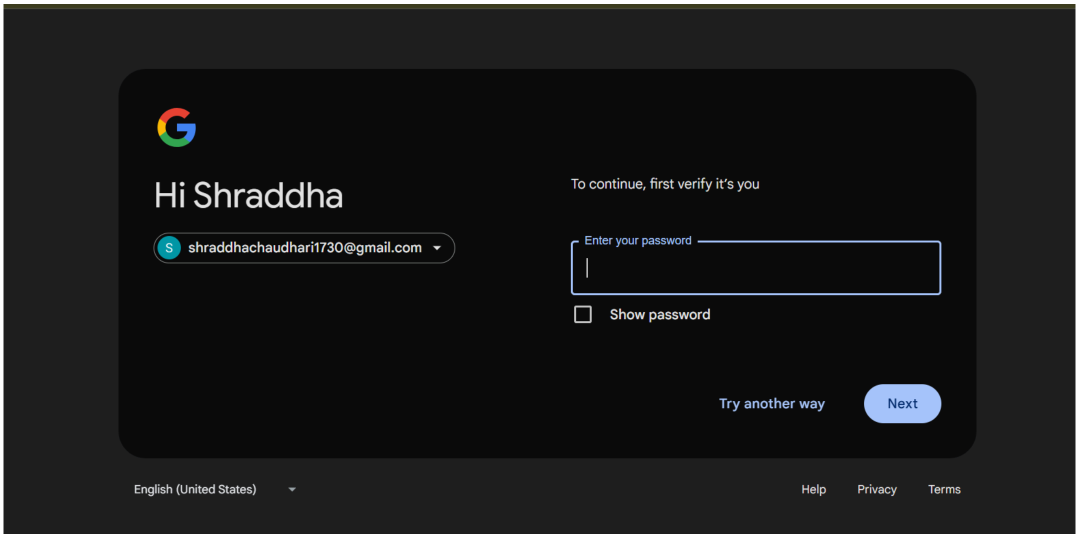
Set a home address for your Google Account

Get more relevant search, weather, and map results

Search for App Password in the search bar.



Enter your google account password to verify it's you.



Hi Shraddha

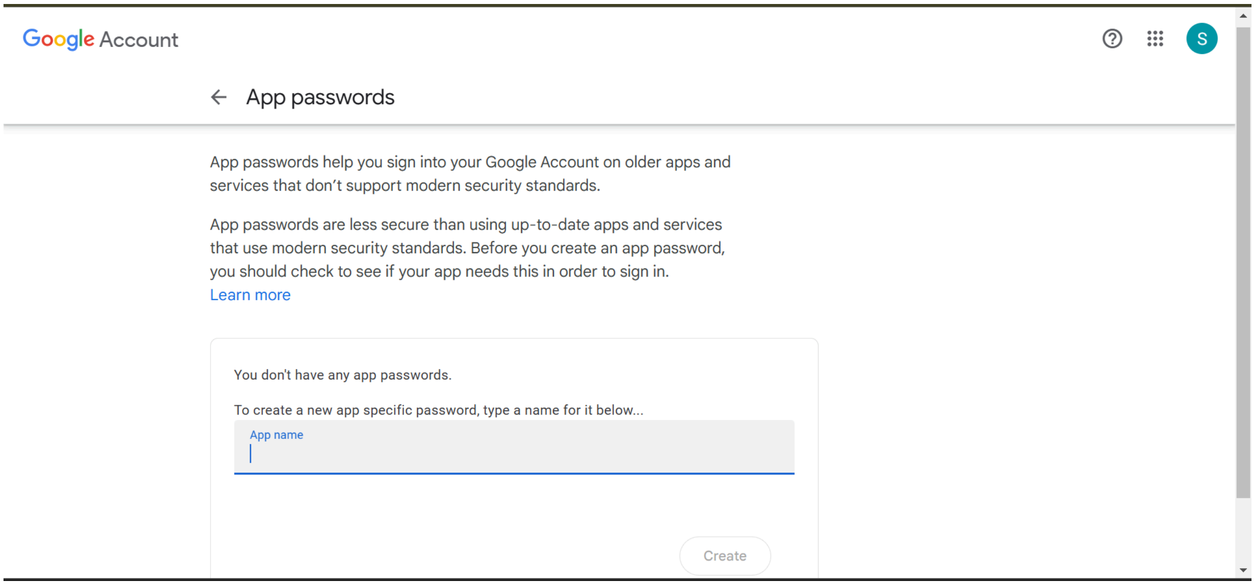
shraddhachaudhari1730@gmail.com

To continue, first verify it's you

Enter your password

Now Enter the app name for which you wanted to get the app password like Diskspace alert

And click on Create to create it



App passwords help you sign into your Google Account on older apps and services that don't support modern security standards.

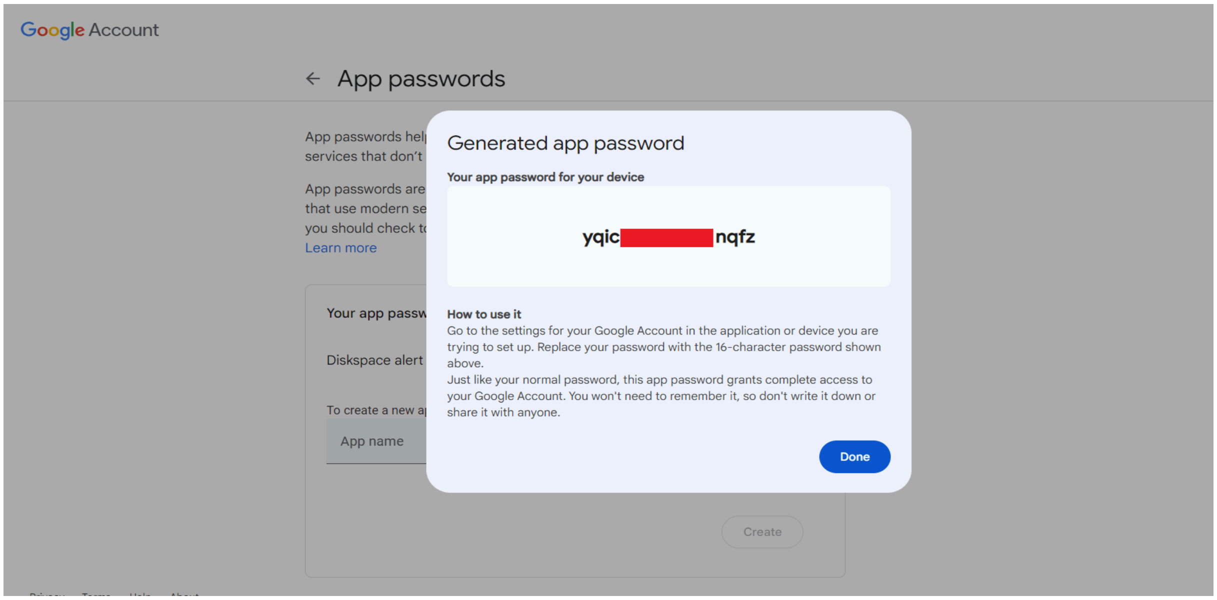
App passwords are less secure than using up-to-date apps and services that use modern security standards. Before you create an app password, you should check to see if your app needs this in order to sign in.
Learn more

You don't have any app passwords.

To create a new app specific password, type a name for it below...

App name

the password will be generated. Note it down cause will be using it.



Generated app password

Your app password for your device

yqic[redacted]nqfz

How to use it

Go to the settings for your Google Account in the application or device you are trying to set up. Replace your password with the 16-character password shown above.

Just like your normal password, this app password grants complete access to your Google Account. You won't need to remember it, so don't write it down or share it with anyone.

Step 3:Combine Simulations-to add artificial load on server for testing

Install the `stress` Tool

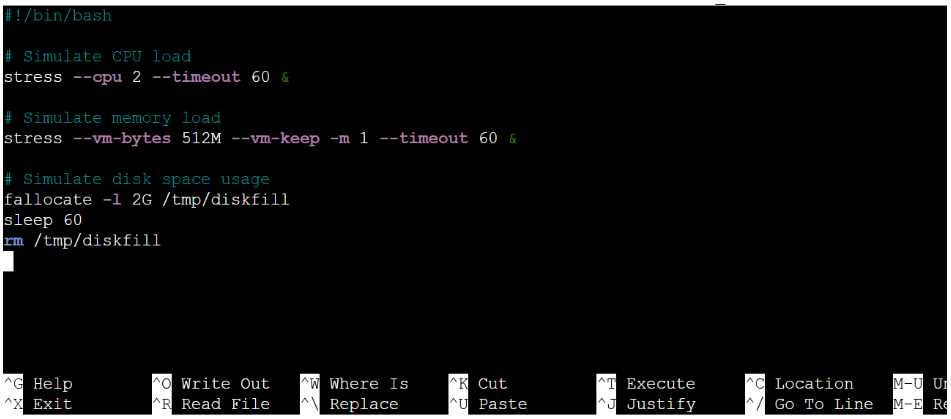
Run the following command to install `stress` on Ubuntu:

```
-> sudo apt update
-> sudo apt install stress -y
```

You can simulate all three (CPU, memory, and disk) simultaneously:

Create a Testing Script:

```
-> nano simulate_load.sh
```



```
#!/bin/bash

# Simulate CPU load
stress --cpu 2 --timeout 60 &

# Simulate memory load
stress --vm-bytes 512M --vm-keep -m 1 --timeout 60 &

# Simulate disk space usage
fallocate -l 2G /tmp/diskfill
sleep 60
rm /tmp/diskfill
```

Make it executable:

```
-> chmod +x simulate_load.sh
```

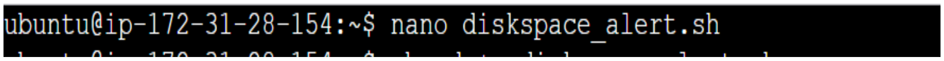
Run it:

```
-> ./simulate_load.sh
```

Step 4:Create a file in Ubuntu

Open the terminal and use the nano command to create a new file.

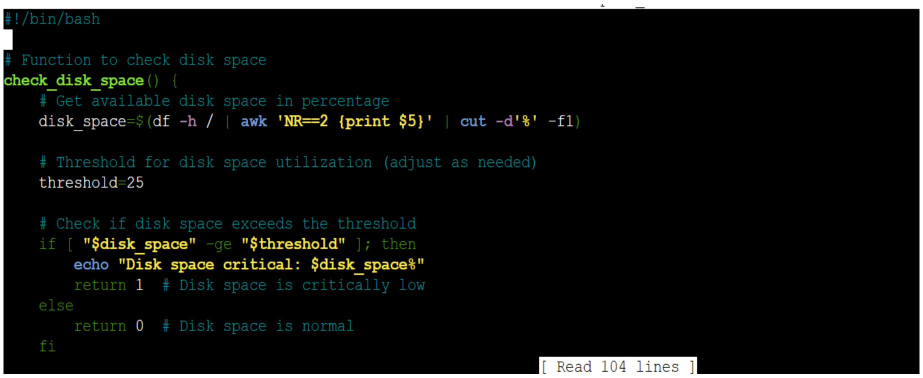
```
 -> nano diskspace_alert.sh
```



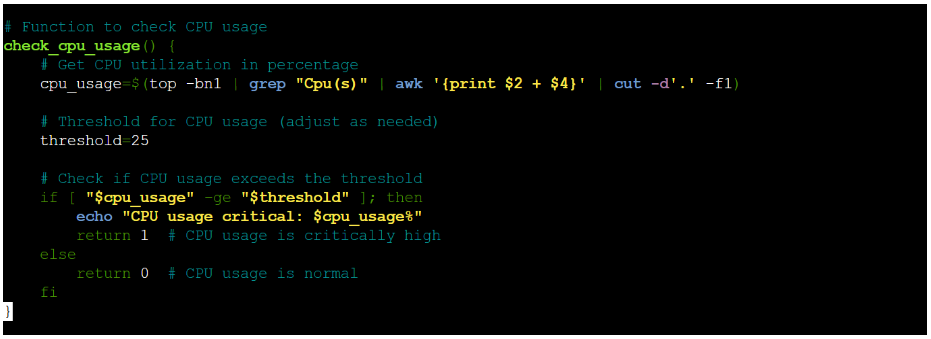
```
ubuntu@ip-172-31-28-154:~$ nano diskspace_alert.sh
```

Step 5:Write a Script to Monitor CPU, Memory, and Disk Space Utilization and Send Email Alerts

Below is a basic script to Monitor Disk Space Utilization and Send Email Alerts.

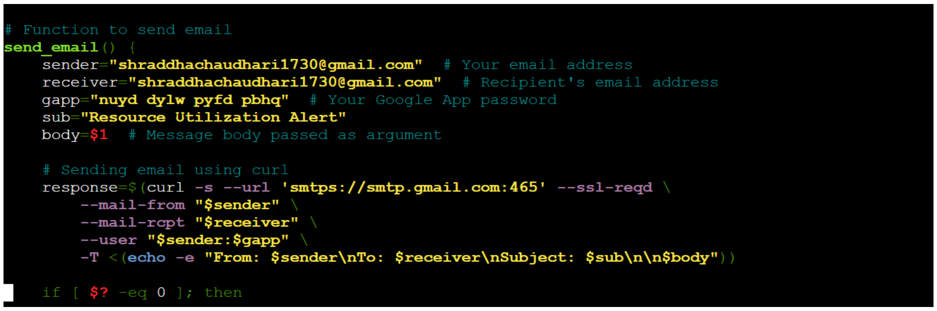


```
#!/bin/bash

# Function to check disk space
check_disk_space() {
    # Get available disk space in percentage
    disk_space=$(df -h / | awk 'NR==2 {print $5}' | cut -d'%' -f1)

    # Threshold for disk space utilization (adjust as needed)
    threshold=25

    # Check if disk space exceeds the threshold
    if [ "$disk_space" -ge "$threshold" ]; then
        echo "Disk space critical: $disk_space%"
        return 1  # Disk space is critically low
    else
        return 0  # Disk space is normal
    fi
                                        [ Read 104 lines ]
```



```
# Function to check CPU usage
check_cpu_usage() {
    # Get CPU utilization in percentage
    cpu_usage=$(top -bn1 | grep "Cpu(s)" | awk '{print $2 + $4}' | cut -d'.' -f1)

    # Threshold for CPU usage (adjust as needed)
    threshold=25

    # Check if CPU usage exceeds the threshold
    if [ "$cpu_usage" -ge "$threshold" ]; then
        echo "CPU usage critical: $cpu_usage%"
        return 1  # CPU usage is critically high
    else
        return 0  # CPU usage is normal
    fi
}
```

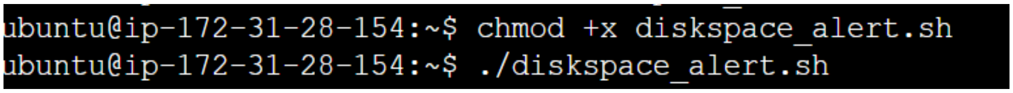
Step 6:Make file executable



```
# Function to send email
send_email() {
    sender="shraddhachaudhari1730@gmail.com"  # Your email address
    receiver="shraddhachaudhari1730@gmail.com"  # Recipient's email address
    gapp="nuyd dylw pyfd pbhq"  # Your Google App password
    sub="Resource Utilization Alert"
    body=$1  # Message body passed as argument

    # Sending email using curl
    response=$(curl -s --url 'smtps://smtp.gmail.com:465' --ssl-reqd \
        --mail-from "$sender" \
        --mail-rcpt "$receiver" \
        --user "$sender:$gapp" \
        -T <(echo -e "From: $sender\nTo: $receiver\nSubject: $sub\n\n$body"))

    if [ $? -eq 0 ]; then
```

Change the file permissions to make it executable using the chmod command.

```
-> chmod +x diskspace_alert.sh
```

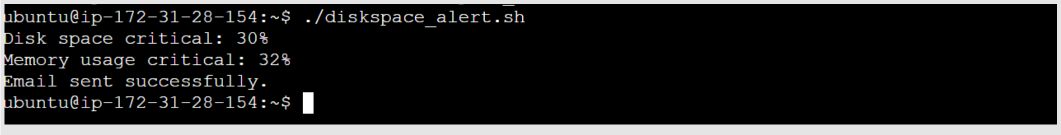


```
ubuntu@ip-172-31-28-154:~$ chmod +x diskspace_alert.sh
ubuntu@ip-172-31-28-154:~$ ./diskspace_alert.sh
```

Step 7:Run the script

Run the script manually by executing the following command.

```
-> ./diskspace_alert.sh
```

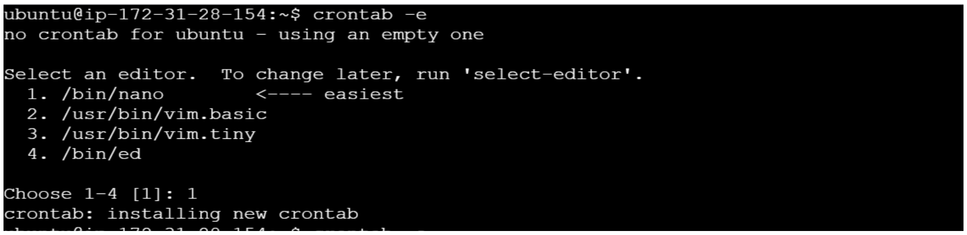


```
ubuntu@ip-172-31-28-154:~$ ./diskspace_alert.sh
Disk space critical: 30%
Memory usage critical: 32%
Email sent successfully.
ubuntu@ip-172-31-28-154:~$
```

Step 7: Automate the Script with Cron

Schedule the script to run periodically using `cron`:

```
-> crontab -e
```



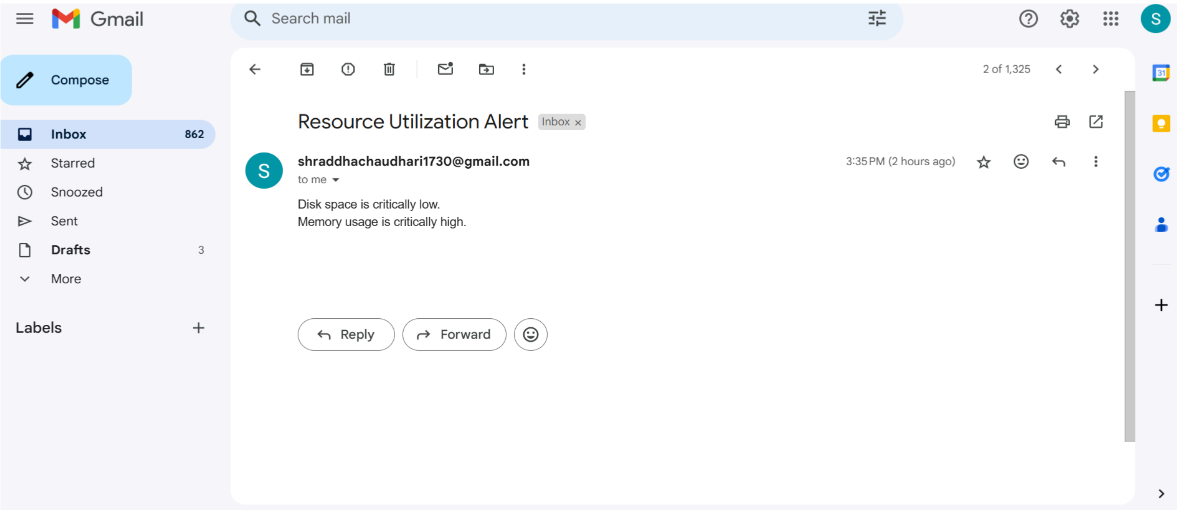
```
ubuntu@ip-172-31-28-154:~$ crontab -e
no crontab for ubuntu - using an empty one

Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed

Choose 1-4 [1]: 1
crontab: installing new crontab
```

Add the following line to run the script every 5 minutes:

```
-> */5 * * * * /path/to/monitor.sh
```

You will get the message Email sent successfully.

Now check the mail box to see if you receive the mail or not.



Resource Utilization Alert

shraddhachaudhari1730@gmail.com

to me

3:35 PM (2 hours ago)

Disk space is critically low.
Memory usage is critically high.