Code for Gmail Authentication via python running on your system:

# batch\_auth.py

from google\_auth\_oauthlib.flow import InstalledAppFlow

import pickle, os, time

SCOPES = ['https://www.googleapis.com/auth/gmail.readonly']

def authenticate(account\_name):

    flow = InstalledAppFlow.from\_client\_secrets\_file(

        'client\_secret.json', SCOPES

    )

    creds = flow.run\_local\_server(port=0)

    os.makedirs("tokens", exist\_ok=True)

    with open(f"tokens/{account\_name}\_token.pickle", "wb") as token\_file:

        pickle.dump(creds, token\_file)

    print(f"✅ Token saved for {account\_name}")

if \_\_name\_\_ == "\_\_main\_\_":

    # List of friendly account names — not the Gmail address itself

    account\_names = [

        "email\_004",

        "email\_005",

        "email\_006",

        # Add more here...

    ]

    for account in account\_names:

        print(f"\n🔐 Authenticating: {account}")

        authenticate(account)

        time.sleep(2)  # Short pause between accounts (optional)

# batch\_auth.py

from google\_auth\_oauthlib.flow import InstalledAppFlow

import pickle, os, time

SCOPES = ['https://www.googleapis.com/auth/gmail.readonly']

def authenticate(account\_name):

    flow = InstalledAppFlow.from\_client\_secrets\_file(

        'client\_secret.json', SCOPES

    )

    creds = flow.run\_local\_server(port=0)

    os.makedirs("tokens", exist\_ok=True)

    with open(f"tokens/{account\_name}\_token.pickle", "wb") as token\_file:

        pickle.dump(creds, token\_file)

    print(f"✅ Token saved for {account\_name}")

if \_\_name\_\_ == "\_\_main\_\_":

    # List of friendly account names — not the Gmail address itself

    account\_names = [

        "email\_004",

        "email\_005",

        "email\_006",

        # Add more here...

    ]

    for account in account\_names:

        print(f"\n🔐 Authenticating: {account}")

        authenticate(account)

        time.sleep(2)  # Short pause between accounts (optional)

Code to Monitor Gmail continuously:

import os

import pickle

import requests

from datetime import datetime, timedelta, timezone

from googleapiclient.discovery import build

from google.auth.transport.requests import Request

BOT\_TOKEN = "7920891849:AAH8ha8USIhdAd0iI-zAZO09Cf0klDIRMhY"

CHAT\_ID = "8055810478"

def send\_telegram(text):

url = f"https://api.telegram.org/bot{BOT\_TOKEN}/sendMessage"

data = {

"chat\_id": CHAT\_ID,

"text": text,

"parse\_mode": "Markdown"

}

try:

requests.post(url, data=data)

except Exception as e:

print(f"Telegram Error: {e}")

def get\_service(token\_file):

with open(token\_file, 'rb') as f:

creds = pickle.load(f)

if creds.expired and creds.refresh\_token:

creds.refresh(Request())

return build('gmail', 'v1', credentials=creds)

def is\_recent(internal\_date\_ms, minutes=5):

try:

email\_time = datetime.fromtimestamp(int(internal\_date\_ms) / 1000, tz=timezone.utc)

now = datetime.now(timezone.utc)

return now - email\_time <= timedelta(minutes=minutes)

except Exception as e:

print(f"Time parse error: {e}")

return False

def check\_emails():

for token\_file in os.listdir('tokens'):

token\_path = os.path.join('tokens', token\_file)

try:

service = get\_service(token\_path)

user\_id = 'me'

# ?? Get Gmail address for this token

profile = service.users().getProfile(userId='me').execute()

account\_email = profile.get('emailAddress', 'Unknown')

results = service.users().messages().list(

userId=user\_id, labelIds=['INBOX'], q='is:unread'

).execute()

messages = results.get('messages', [])

for msg in messages:

msg\_data = service.users().messages().get(userId=user\_id, id=msg['id']).execute()

internal\_date = msg\_data.get('internalDate')

if not is\_recent(internal\_date):

continue

headers = msg\_data['payload']['headers']

subject = next((h['value'] for h in headers if h['name'].lower() == 'subject'), 'No Subject')

sender = next((h['value'] for h in headers if h['name'].lower() == 'from'), 'Unknown')

receiver = next((h['value'] for h in headers if h['name'].lower() == 'to'), 'Unknown')

snippet = msg\_data.get('snippet', '')

alert = (

f"?? \*New Email\*\n"

f"?? Account: `{account\_email}`\n"

f"From: {sender}\n"

f"To: {receiver}\n"

f"Subject: {subject}\n"

f"Snippet: {snippet[:100]}"

)

send\_telegram(alert)

except Exception as e:

print(f"Error processing {token\_file}: {e}")

if \_\_name\_\_ == "\_\_main\_\_":

check\_emails()

Codes to run inside Linux:

Install Python packages (globally for your user)

sudo apt update

sudo apt install python3-pip -y

pip3 install --user google-api-python-client google-auth google-auth-oauthlib google-auth-httplib2 requests

Telegram Chat ID generation Link

https://api.telegram.org/bot<Enter BOT ID HERE>/getUpdates

Code to make system continuously run and monitor

**✅ Step-by-Step Guide**

**📁 Step 1: Move your monitor script to a fixed location**

If it's in ~/gmail-monitor/monitor.py, you're fine.

Make sure the folder contains:

monitor.py

tokens/

monitor.log

**📝 Step 2: Create a systemd service**

Run:

sudo nano /etc/systemd/system/gmail-monitor.service

Paste this in:

[Unit]

Description=Gmail Monitor Script

After=network.target

[Service]

Type=simple

User=ubuntu

WorkingDirectory=/home/ubuntu/gmail-monitor

ExecStart=/usr/bin/python3 monitor.py

Restart=on-failure

RestartSec=5s

[Install]

WantedBy=multi-user.target

Make sure /usr/bin/python3 matches your actual Python path. Run which python3 to confirm.

OR

[Unit]

Description=Gmail Monitor Script

After=network.target

[Service]

Type=simple

User=ubuntu

WorkingDirectory=/home/ubuntu/gmail-monitor

ExecStart=/usr/bin/python3 /home/ubuntu/gmail-monitor/monitor.py

Restart=always

RestartSec=5s

StandardOutput=append:/home/ubuntu/gmail-monitor/monitor.log

StandardError=append:/home/ubuntu/gmail-monitor/error.log

Environment=PYTHONUNBUFFERED=1

[Install]

WantedBy=multi-user.target

**✅ Step 3: Enable and Start the Service**

sudo systemctl daemon-reexec

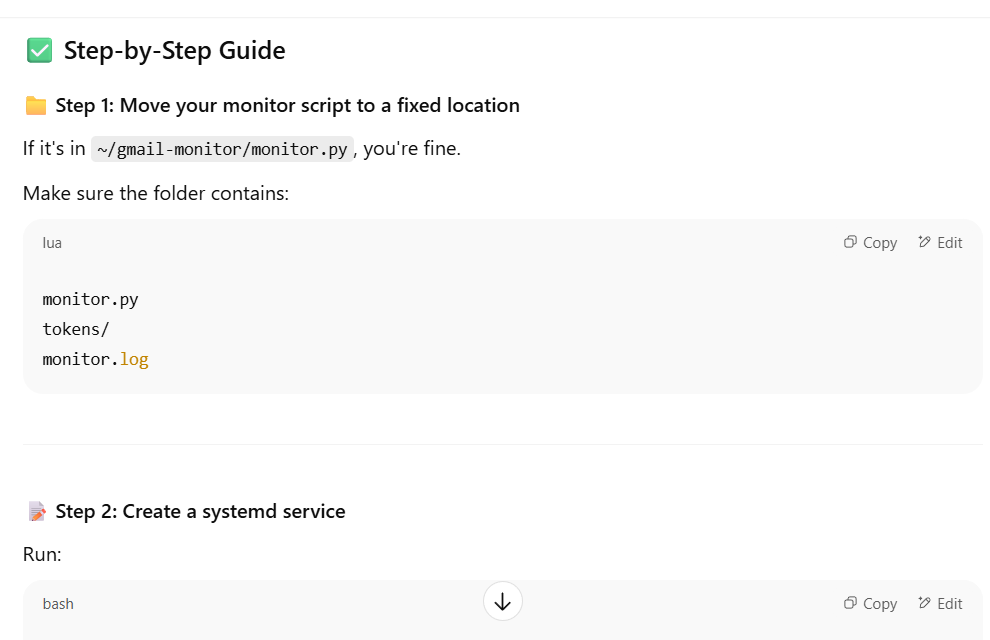
sudo systemctl daemon-reload

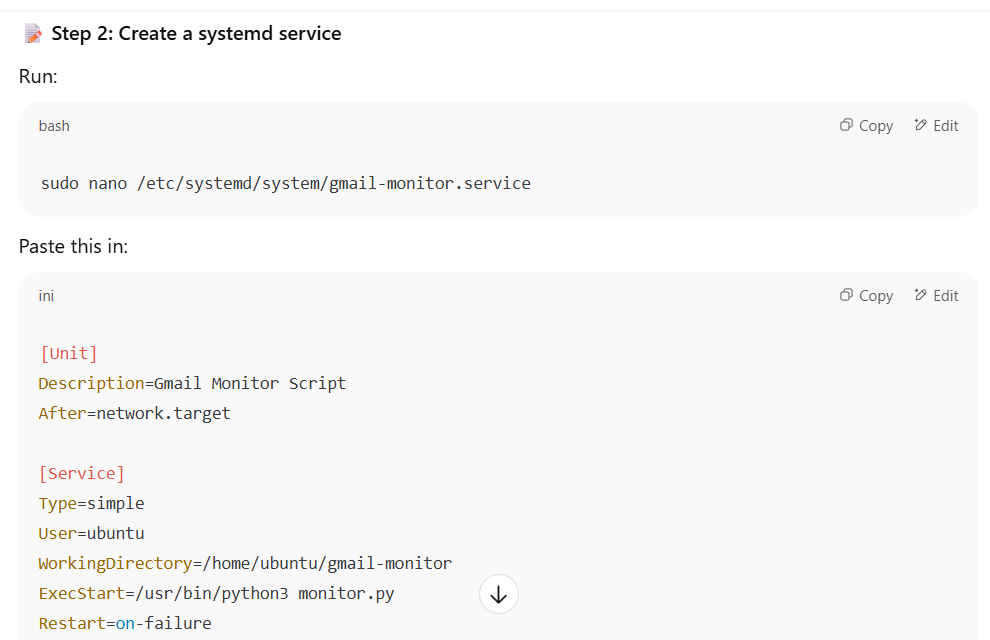
sudo systemctl enable gmail-monitor.service

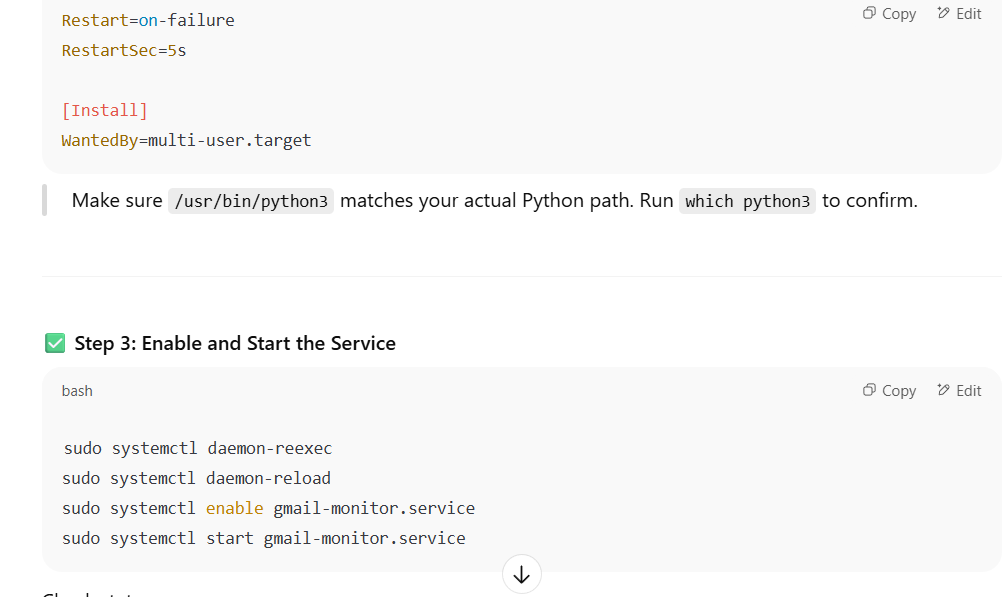
sudo systemctl start gmail-monitor.service

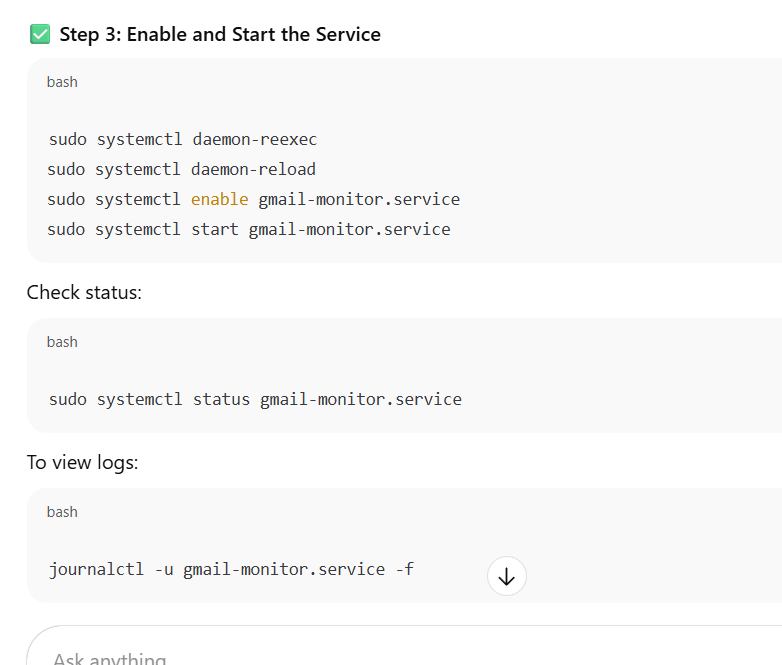
Check status:

sudo systemctl status gmail-monitor.service









Step 2 Commands:

sudo nano /etc/systemd/system/gmail-monitor.service

Paste this in:

[Unit]

Description=Gmail Monitor Script

After=network.target

[Service]

Type=simple

User=ubuntu

WorkingDirectory=/home/ubuntu/gmail-monitor

ExecStart=/usr/bin/python3 monitor.py

Restart=on-failure

RestartSec=5s

[Install]

WantedBy=multi-user.target

Step3:

sudo systemctl daemon-reexec

sudo systemctl daemon-reload

sudo systemctl enable gmail-monitor.service

sudo systemctl start gmail-monitor.service

Status Check: sudo systemctl status gmail-monitor.service