Project 3

Automated Backup and Rotation System with Google Drive Integration using Python and rclone

Batch

MCA 7 Feb 2025 &
DevOps 2 Jun 2025

Name

Suraj Molke

Name of the Mentor

Ravindra Bagle Sir
Swati Zampal Ma'am

Fortune Cloud Technologies

2nd Floor, Shirodkar House, Opposite To Amit Cafe, Congress House Road, Near Pune Municipal Corporation, Shivajinagar, Pune - 411005.

smolke9@gmail.com Suraj Molke Smolke9

Project Title: Automated Backup and Rotation System with Google Drive Integration using Python and rclone

Objective:

The goal of this project is to automate the process of backing up important project folders.

It creates .zip files from a source folder, stores them locally, uploads them to Google Drive, and manages (deletes) old backups.

It also sends a notification to confirm the backup and writes a log for tracking.

This system runs daily using a cron job, and helps ensure that data is never

Lost.

Steps:

1. Create Project Folder Structure:

```
ubuntu@ip-172-31-41-228:~$ mkdir -p ~/backup_Gdrive/{logs,backups}
ubuntu@ip-172-31-41-228:~$ ls
backup_Gdrive
ubuntu@ip-172-31-41-228:~$ cd backup_Gdrive/
ubuntu@ip-172-31-41-228:~/backup_Gdrive$ ls
backups  logs
ubuntu@ip-172-31-41-228:~/backup_Gdrive$
```

Purpose:

logs/: Stores a log file showing all backup operations

backups/: Temporarily stores zipped files before uploading to Google Drive

2. Install Required Tools:

```
ubuntu@ip-172-31-41-228:~/backup_Gdrive$ sudo apt update && sudo apt install -y pyth
on3 python3-pip zip unzip curl
```

1)python3, pip3: For running Python scripts

2)zip/unzip: For creating .zip files

3)curl: For uploading to webhook

4) python-dotenv is required for your script to read .env variables:

i) To enable python3 -m venv command

```
ubuntu@ip-172-31-41-228:~/backup_Gdrive$ sudo apt install python3-venv python3-full
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

ii) python3 -m venv venv-To create isolated python environment

iii) source venv/bin/activate - To enter the environment

```
ubuntu@ip-172-31-41-228:~/backup_Gdrive$ python3 -m venv venv
ubuntu@ip-172-31-41-228:~/backup_Gdrive$ source venv/bin/activate
(venv) ubuntu@ip-172-31-41-228:~/backup_Gdrive$ ls
backups  logs  venv
(venv) ubuntu@ip-172-31-41-228:~/backup_Gdrive$
```

iii) pip install python-dotenv - To read .env file in your script

v) venv folder – Keeps project clean and dependency-safe

```
(venv) ubuntu@ip-172-31-41-228:~/backup_Gdrive$ pip install python-dotenv
Collecting python-dotenv
  Downloading python_dotenv-1.2.1-py3-none-any.whl.metadata (25 kB)
Downloading python_dotenv-1.2.1-py3-none-any.whl (21 kB)
Installing collected packages: python-dotenv
Successfully installed python-dotenv-1.2.1
(venv) ubuntu@ip-172-31-41-228:~/backup_Gdrive$ ls
backups  logs  venv
```

3. Install and Configure rclone:

Installed rclone (used to upload to Google Drive):

During rclone config:

• We created a remote called gdrive-backup

• Selected drive as the storage type (Google Drive)

• Logged in to Google to give rclone permission

• We skipped setting client ID/secret (left it empty)

• Chose full access scope

• Did not use service account or shared drive

Why:

rclone acts as the middleman to upload zipped backups to Google Drive

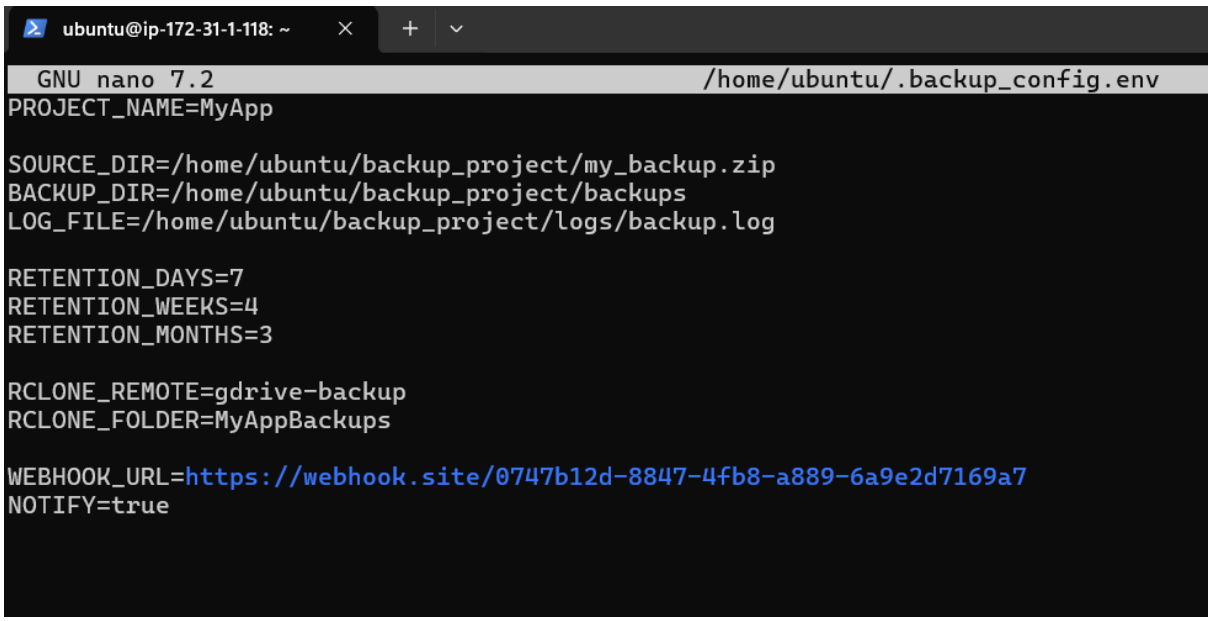
I)Using - “curl https://rclone.org/install.sh | sudo bash

ii)rclone config.

4.Create Configuration File:

We created the config file:

nano ~/.backup_config.env



*we get the webhook url from webhook website:

Why:

Stores values like source folder, backup location, retention policy, etc.

Helps make the Python script reusable and easy to change later

5.Write the python Backup Script:

We created the script:

nano ~/backup_project/backup_script.py

This script does the following:

• Loads variables from the .env file

• Creates a .zip file of the source directory

• Stores it in the backup folder with a timestamp

• Uploads the zip to Google Drive using rclone

• Sends a webhook notification (optional)

• Logs each operation

• Deletes old backups based on retention policy

We made it executable:

chmod +x ~/backup_project/backup_script.py

6.Test the Script:

We tested the script manually

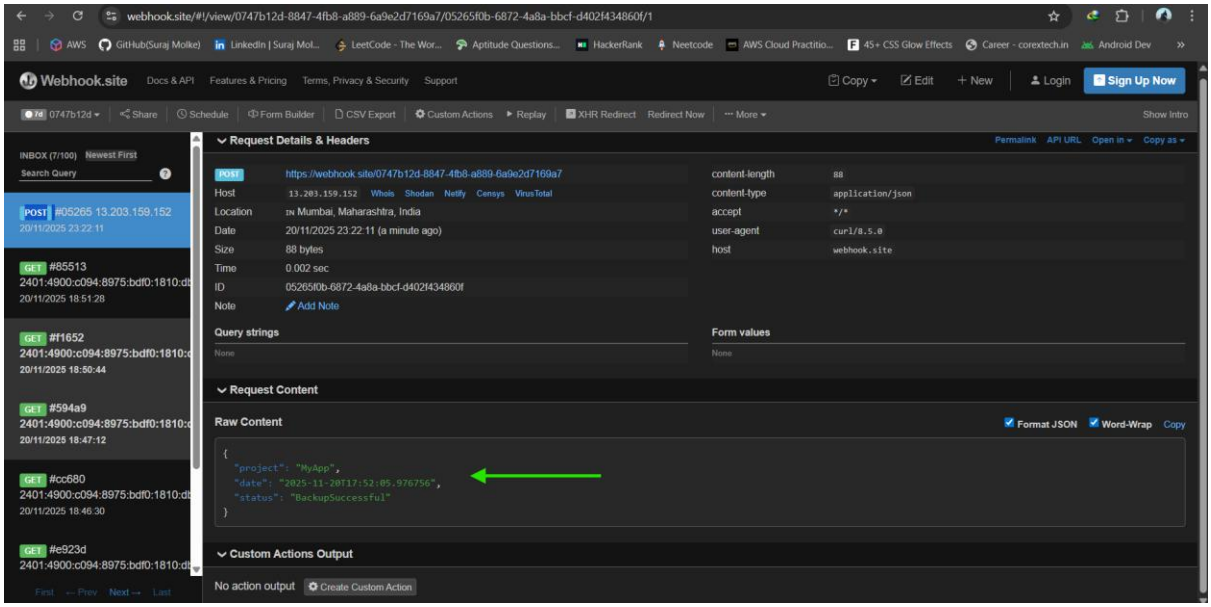
```
ModuleNotFoundError: No module named 'dotenv'
ubuntu@ip-172-31-1-118:~/backup_project$ cd ~/backup_project
source venv/bin/activate
(venv) ubuntu@ip-172-31-1-118:~/backup_project$ python3 backup_script.py
[INFO] Creating backup: /home/ubuntu/backup_project/backups/2025/11/20/MyApp_20251120_175205.zip
[INFO] Uploading to Google Drive using rclone...
This URL has no default content configured. <a href="https://webhook.site/#!/edit/0747b12d-8847-4fb8-a88
te</a>.[INFO] Webhook notification sent.
```

New zip file created inside backups/YYYY/MM/DD/

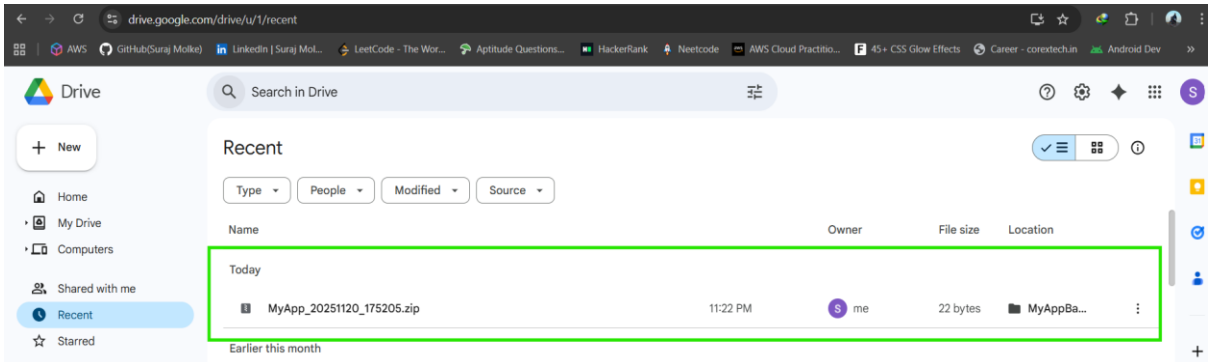
```
ubuntu@ip-172-31-1-118:~/backup_project/backups$ tree
.
└── 2025
    └── 11
        └── 20
            └── MyApp_20251120_175205.zip

4 directories, 1 file
ubuntu@ip-172-31-1-118:~/backup_project/backups$
```

iv) Notification seen on webhook.site



File was uploaded to Google Drive



7. Automating the Script via Cron with Virtual

Environment:

Problem occured when running cron job without shell script:

The cron job was not running the Python backup script properly, even though

it worked when we ran it manually in the terminal.

Steps:

1) Create run_backup.sh Shell Script:

“nano ~/backup_project/run_backup.sh”

2) Make It Executable:

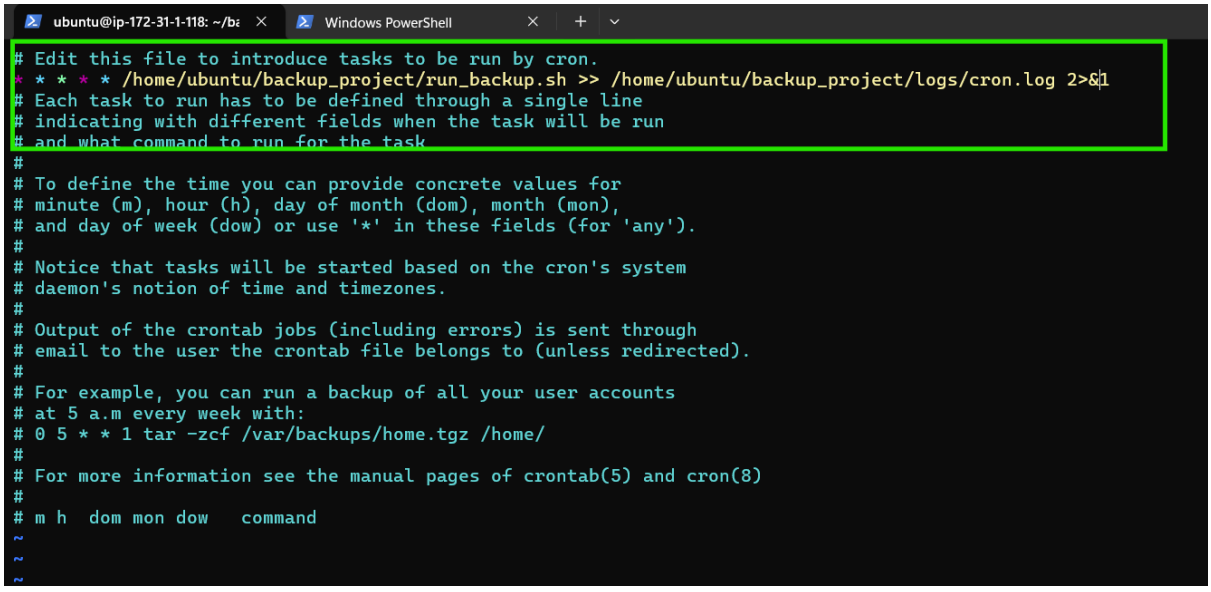
“chmod +x ~/backup_project/run_backup.sh”

3) Schedule It with Cron:

“crontab -e”

4)Add this line to run it every minute or you can modify it

according to you:



7. Automating the Script via Cron with Virtual Environment:

Problem occured when running cron job without shell script:

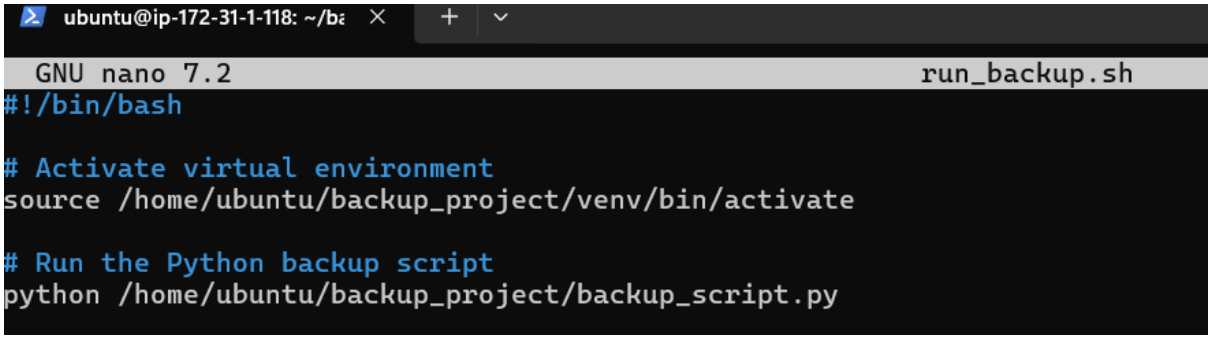
The cron job was not running the Python backup script properly, even though

it worked when we ran it manually in the terminal.

Steps:

1) Create run_backup.sh Shell Script:

"nano ~/backup_project/run_backup.sh"



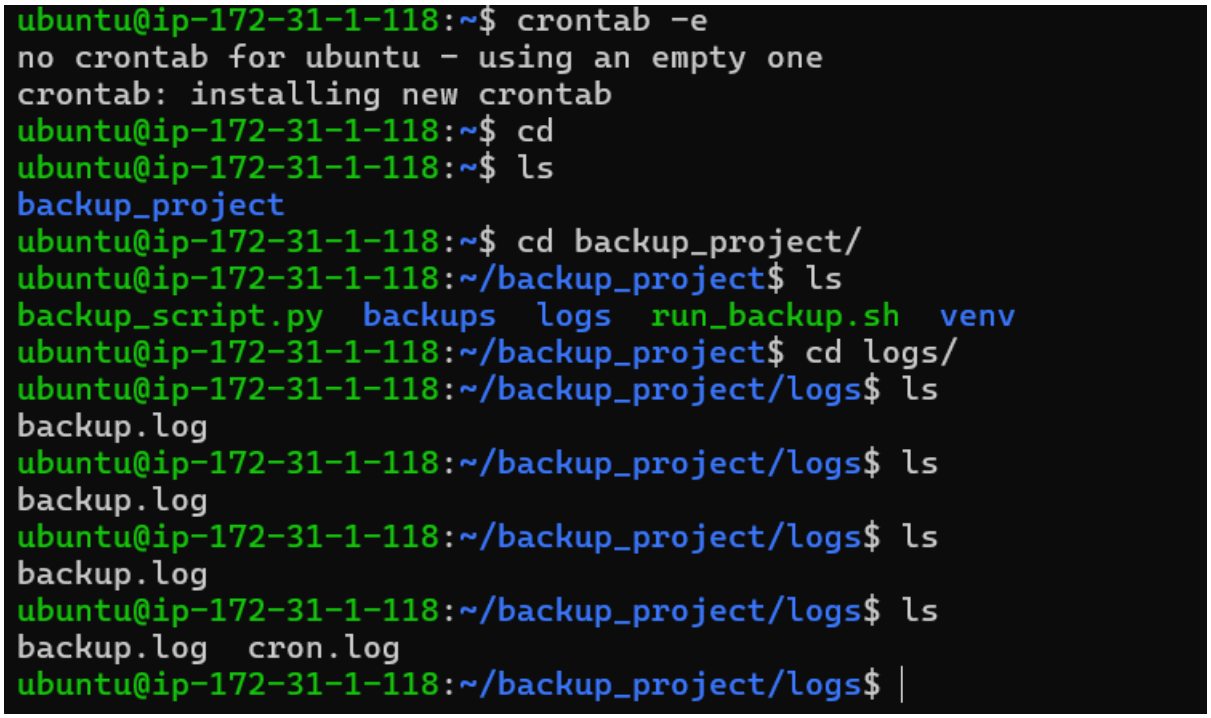
What this does:

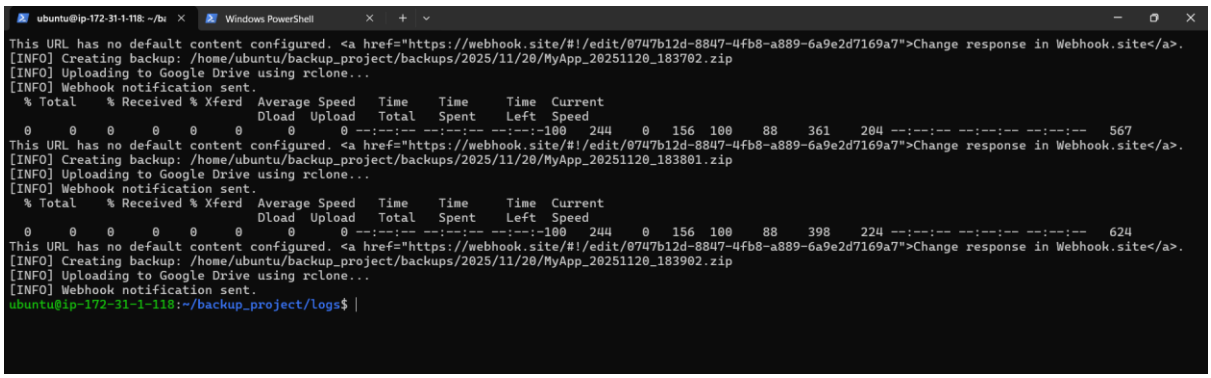
• Runs the backup automatically every 1 minute

• Saves any output (success/failure logs) into

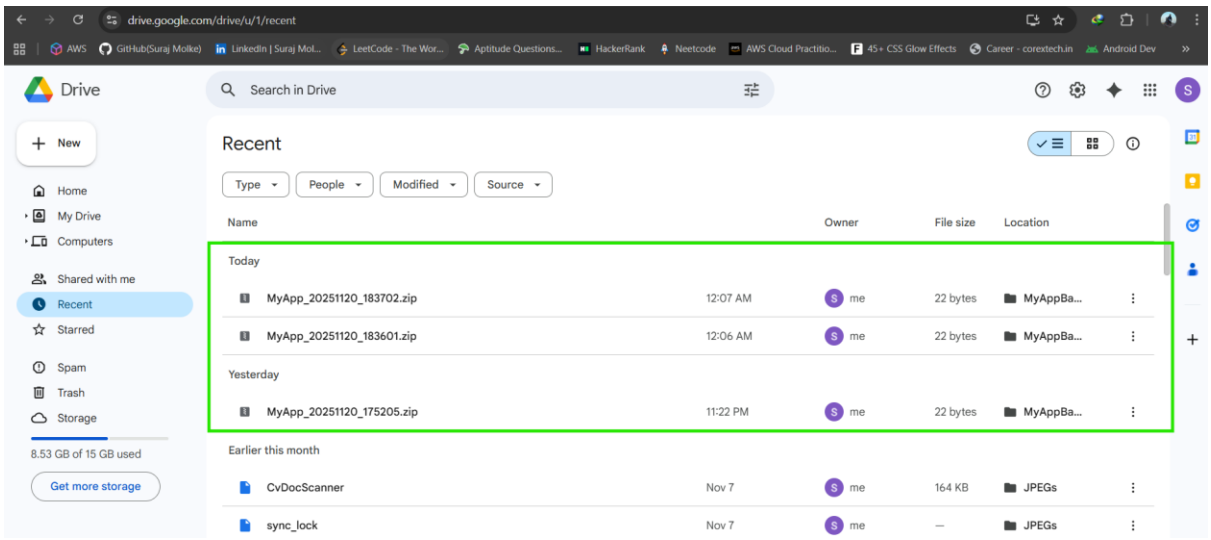
cron_output.log.

Output:





On google Drive uploaded



Conclusion

This project provides a fully automated and reliable backup solution by combining directory compression, cloud synchronization, and intelligent retention policies. Using rclone, the system securely uploads timestamped backups to Google Drive, while automated rotation ensures efficient storage management across daily, weekly, and monthly cycles. The integration of a webhook notification system offers real-time status updates, and detailed logging makes the process transparent and easy to audit. With configuration handled through a simple .env file and execution scheduled via cron, the solution is both flexible and easy to maintain, ensuring consistent and dependable backups with minimal manual intervention.

Key Features:

- Environment-based Configuration (.env file)
- Timestamped Backup Zip Creation
- Google Drive Upload via rclone
- Retention Policy: Automatically deletes older backups (daily, weekly,monthly)
- Webhook Notification after successful backup
- Logging: Logs each action (success/failure/deletions)
- Automated Execution via Cron