Project Documentation: Automated Backup and Rotation System with Google Drive Integration

---

1. Project Title

Automated Backup and Rotation System with Google Drive Integration

---

2. Objective

To automate daily, weekly, and monthly backups of a project directory, upload backups to Google Drive, delete old backups, monitor storage quota, send optional email notifications, and allow manual restoration—all using shell scripting, `rclone`, and `cron` on an Ubuntu EC2 instance.

---

3. Key Benefits

- Reliable automated offsite backups
- Zero manual intervention after setup
- Low-cost, simple alternative to commercial tools
- Restore and recovery possible anytime
- Storage usage monitoring
- Customizable retention policy

---

4. Tools and Technologies Used

- **Ubuntu EC2 Instance (Cloud Host)**
- **Shell Scripting** (Bash)
- **Rclone** (Sync with Google Drive)
- **Cron** (Task Scheduling)
- **Msmtp** (Optional Email Notification)
- **Zip/Unzip** (Compression/Extraction)
- **tree**, **du**, **df** (for structure/logging)

---

5. Folder Structure

```
backup_project/
├── daily_backup.sh
├── weekly_backup.sh
├── monthly_backup.sh
├── restore_backup.sh
├── check_quota.sh
├── my_project_folder/
│   ├── file1.txt
│   └── file2.txt
├── backup_log.txt
├── quota_log.txt
```

```
└── project-backups/
    ├── daily/
    ├── weekly/
    └── monthly/
```

---

6. Steps to Set Up the Project

Step 1: Launch Ubuntu EC2 Instance

- Login to AWS Console
- Launch EC2 instance (Ubuntu 22.04 preferred)
- Allow SSH (port 22)

Step 2: Install Required Packages

```
sudo apt update
sudo apt install rclone zip unzip msmtp mailutils tree -y
```

Step 3: Configure Rclone for Google Drive

```
rclone config
# Choose: n (new remote)
# Name: gdrive
# Type: drive
# Follow the prompts and authorize using your browser
```

Step 4: Prepare Project Directory

```
mkdir ~/backup_project
cd ~/backup_project
mkdir my_project_folder project-backups
mkdir project-backups/daily project-backups/weekly project-backups/monthly
```

Step 5: Create Your Scripts

Create the following shell scripts:

daily_backup.sh

- Compresses the folder
- Uploads to Google Drive
- Logs actions
- Deletes old backups beyond 7 days

weekly_backup.sh

- Same logic as daily but separate path
- Retain last 4 weekly backups

monthly_backup.sh

- Retain last 3 monthly backups

check_quota.sh

- Uses `rclone about gdrive:` to check space

restore_backup.sh

- Asks for backup name
- Downloads and extracts it

Step 6: Make Scripts Executable

```
chmod +x *.sh
```

---

7. Automate Using Cron

```
crontab -e

# Daily backup at 1 AM
0 1 * * * /home/ubuntu/backup_project/daily_backup.sh

# Weekly backup at 2 AM every Sunday
0 2 * * 0 /home/ubuntu/backup_project/weekly_backup.sh

# Monthly backup at 3 AM on 1st
0 3 1 * * /home/ubuntu/backup_project/monthly_backup.sh
```

---

8. Optional: Setup Email Notification

Step 1: Install msmtp

```
sudo apt install msmtp
```

Step 2: Configure

```
nano ~/.msmtprc
```

Add:

```
defaults
auth on
tls on
tls_trust_file /etc/ssl/certs/ca-certificates.crt
logfile ~/.msmtp.log
account gmail
host smtp.gmail.com
port 587
from your_email@gmail.com
user your_email@gmail.com
password your_app_password
account default : gmail
```

---

9. Testing Scripts

Run each script manually and verify:

```
./daily_backup.sh
./weekly_backup.sh
./monthly_backup.sh
./check_quota.sh
./restore_backup.sh
```

Use `rclone ls gdrive:project-backups/` to check files in Drive.

10. Final Output and Logs

- `backup_log.txt` - Contains success/failure info
- `quota_log.txt` - Shows drive space report
- Google Drive has three folders: `daily/`, `weekly/`, `monthly/`

---

11. Conclusion

This project delivers a production-ready, fully automated solution for backing up any critical data to the cloud. It ensures:

- Data Safety
- Disaster Recovery
- Cost Savings
- Minimal Manual Intervention

---

12. GitHub Repository

 GitHub - Automated Backup and Rotation System

For any queries or improvements, feel free to connect!