Got it! Let’s break this down step by step for GCP.

I’ll guide you through both manual credential setup and then the Python automation script that uses them to init, setup, and reset resources. I’ll be thorough so you can implement this without confusion.

1️⃣ Manual Preparation: GCP Credentials

Before your Python script can interact with GCP, you need service account credentials. Here’s how to do it:

Step 1: Create a GCP Project

1. Go to **Google Cloud Console**.
2. Click the **project dropdown** → **New Project**.
3. Name it (e.g., `automation-test`) and note the **Project ID** (we’ll use it in the script).

Step 2: Enable Required APIs

The script will need APIs for storage, cloud functions, container registry, logging, etc.
For example:

- Cloud Storage API
- Cloud Functions API
- Container Registry API
- Cloud Build API
- Logging API
- Artifact Registry Writer
- Storage Admin
- Cloud Run Admin
- Cloud Run Admin
- Cloud Build Editor
- Cloud Run Admin API

How to enable:

1. Go to **APIs & Services** → **Library**.

2. Search and **Enable** the APIs listed above.

Step 3: Create a Service Account

This account will be used by your Python script:

1. Go to **IAM & Admin → Service Accounts**.
2. Click **Create Service Account**.
3. Give it a name like `automation-sa`.
4. Assign roles needed for your resources. For simplicity, you can assign:
   - Storage Admin
   - Cloud Functions Admin
   - Viewer (or customize per your permissions JSON)
5. Click **Done**.

Step 4: Generate a Key File

1. In the Service Account list, click your newly created account → **Keys** → **Add Key** → **Create New Key**.
2. Choose **JSON** format.
3. Download the file (e.g., `gcp-service-account.json`).
4. Keep this file secure – it’s essentially your credentials.