Google Earth Engine Service Account Setup

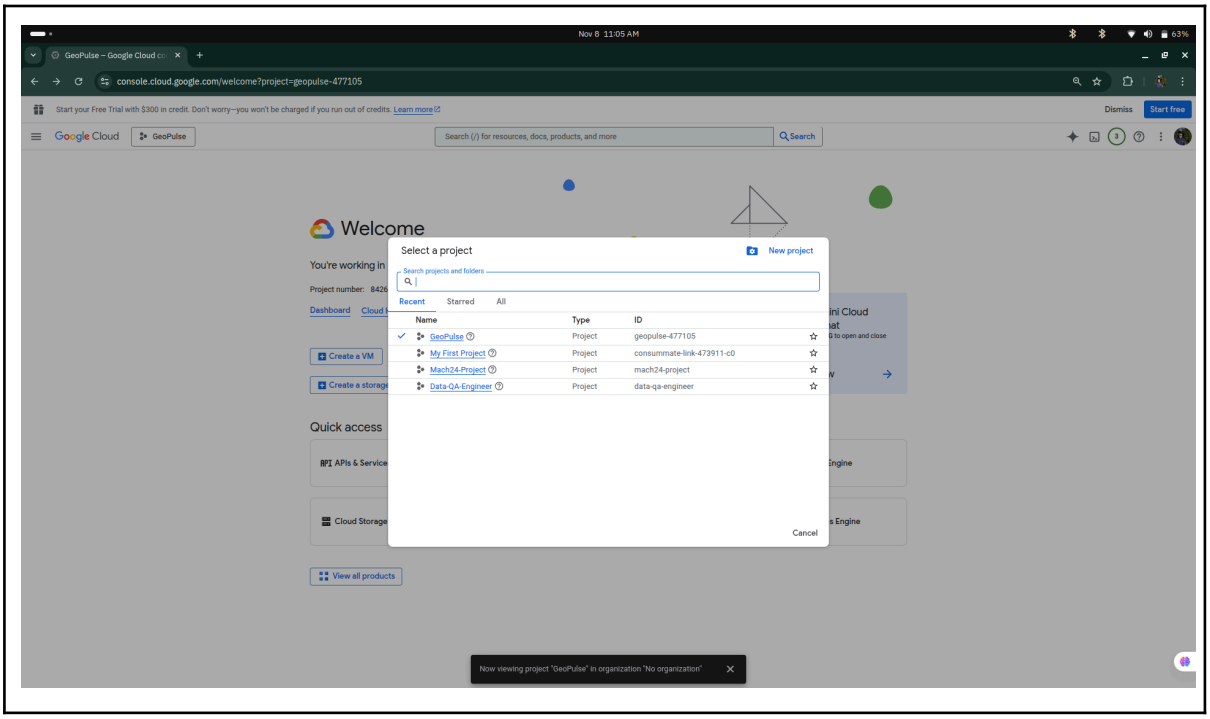
This guide explains how to create a Google Cloud project, set up a service account, and generate the credentials key (key.json) to use with Python scripts for Google Earth Engine (GEE) access.

✅ 1. Create a Google Cloud Project

- Go to **Google Cloud Console.**

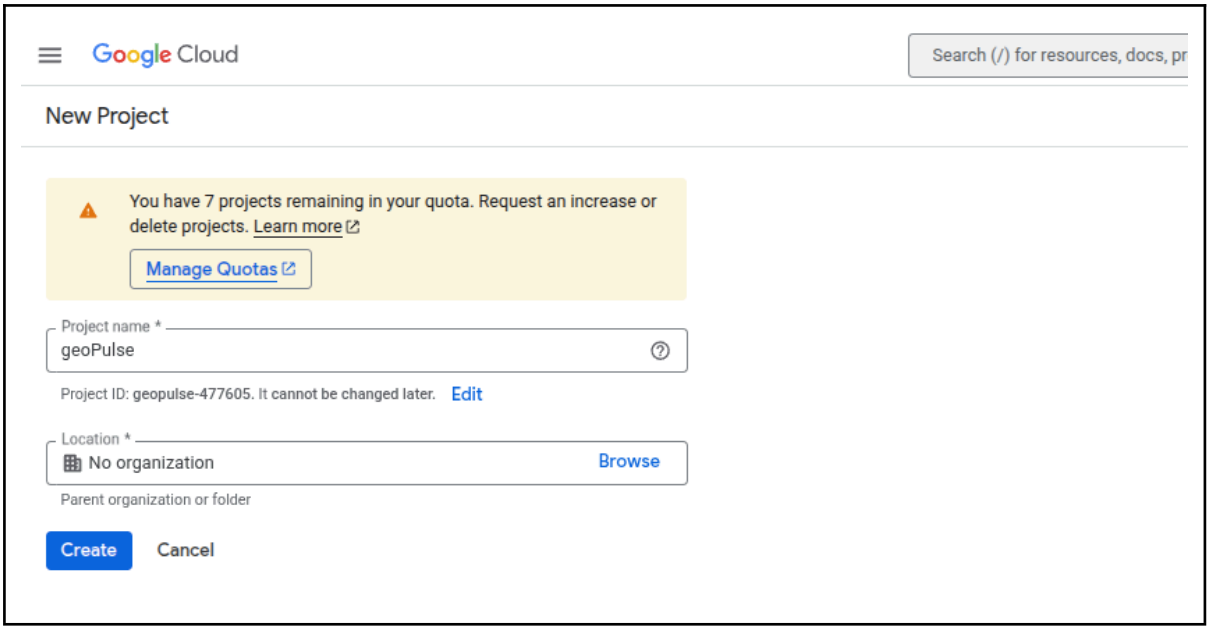
[Google Cloud Console]

- Click **Ctrl + o** and you will see the pop-up below.



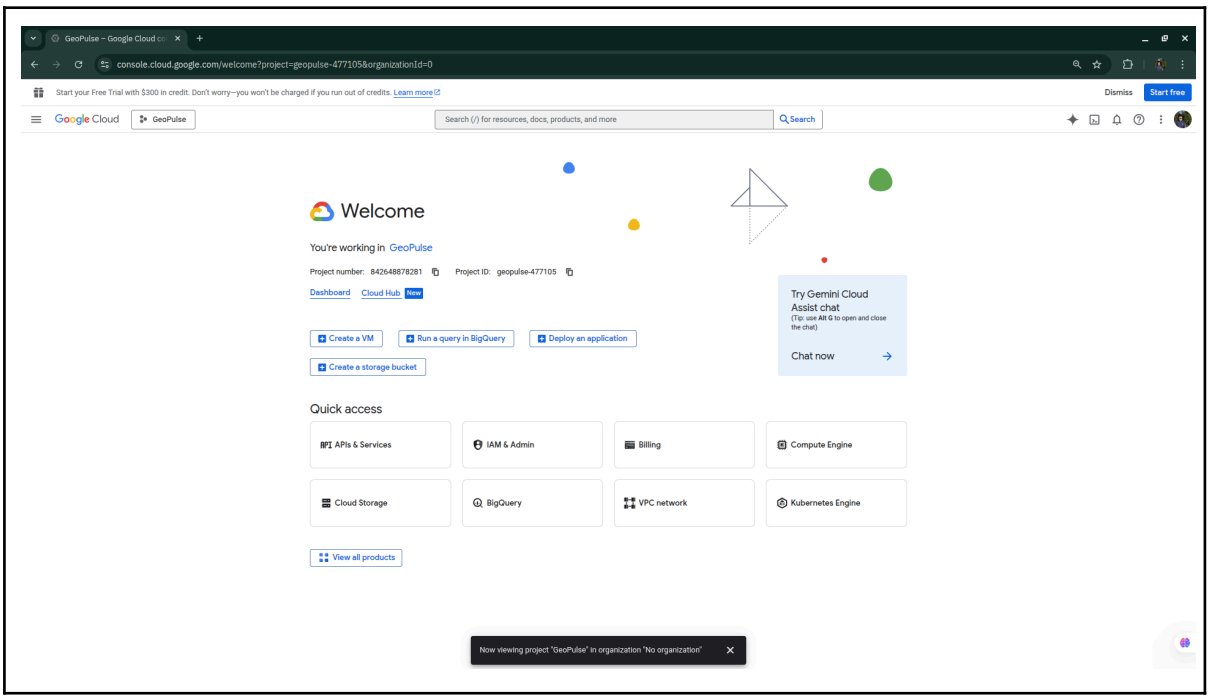
- Click on the **New project.**

- Enter a Project Name (e.g., geoPulse) and note the Project ID (e.g., geoPulse-477605).



- Click **Create**.

- Your new Google Cloud Project will be created.



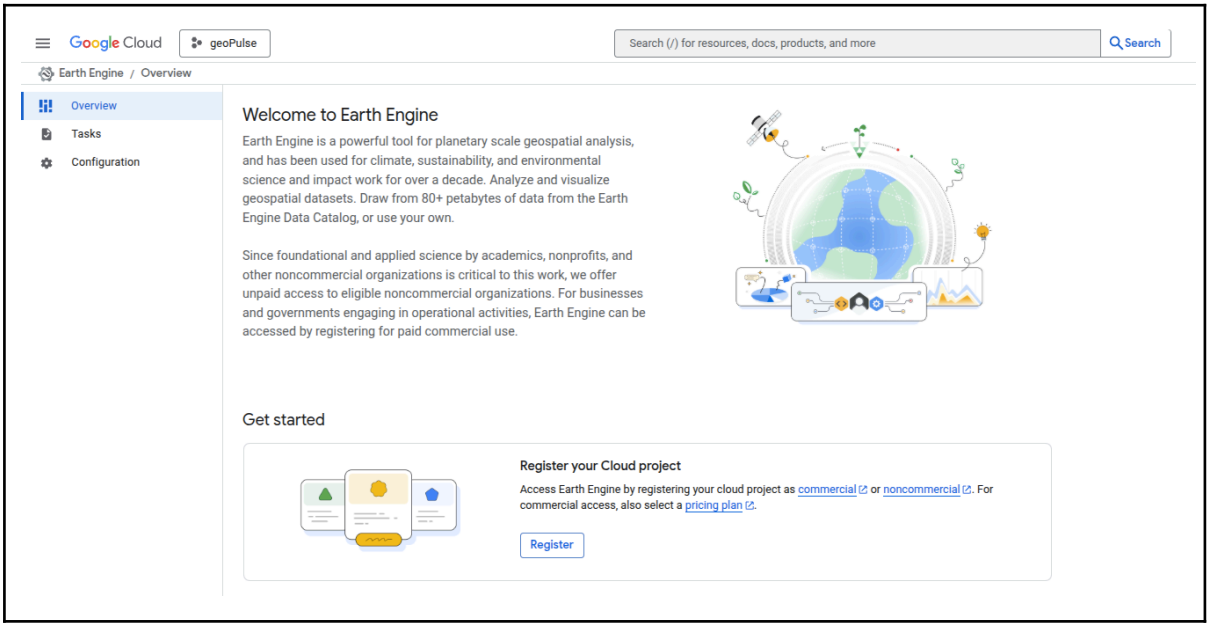
Tip: The project ID is used in our Python code as PROJECT_ID.

✅ 2. Register Your Project for Earth Engine

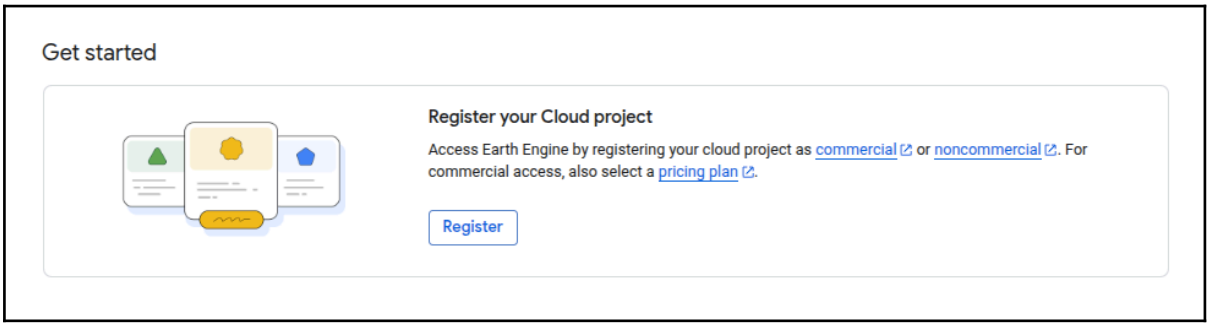
- Go to the **Earth Engine Overview** page.

https://console.cloud.google.com/earth-engine/welcome

- Select your previously created project (**Ctrl + o**).

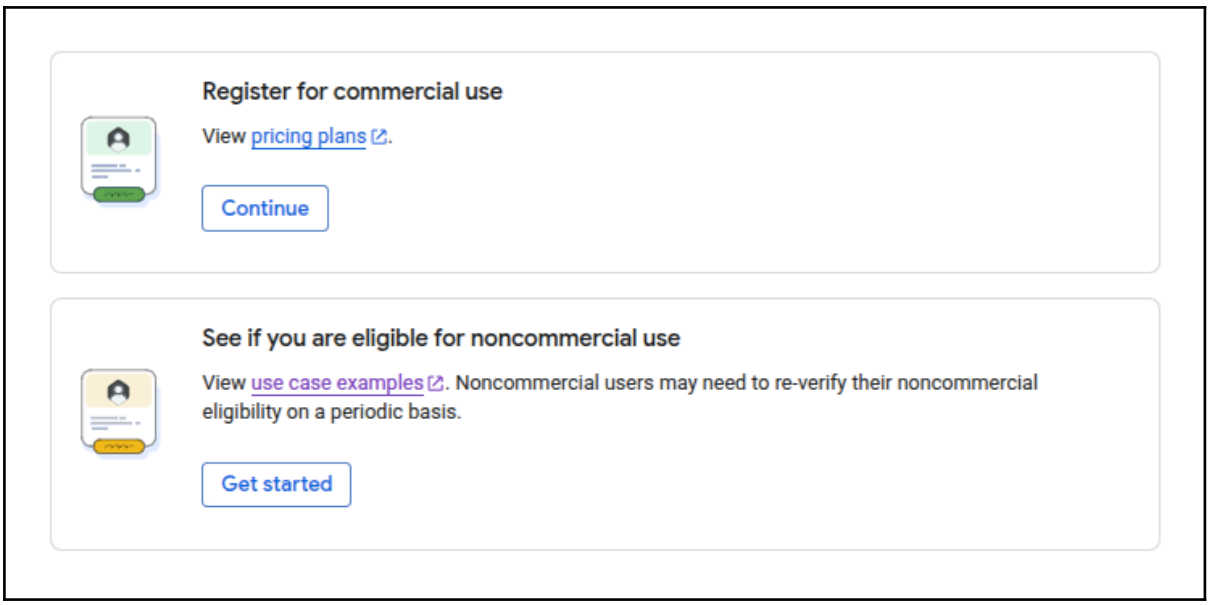


- Register your cloud project by clicking **Register**.



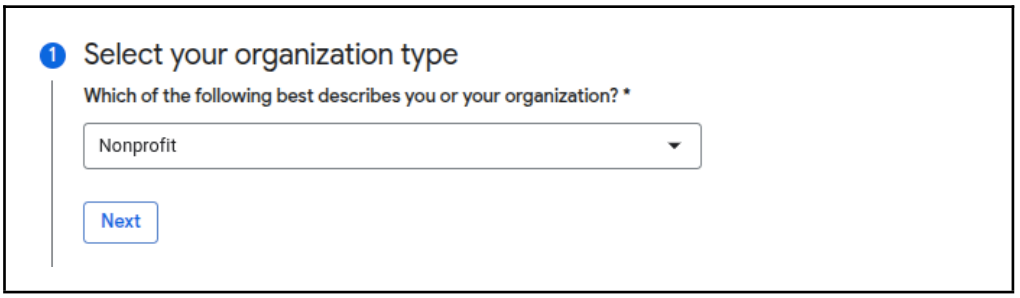
- You will be redirected to the **Earth Engine Configuration** page, where you need to decide to register your cloud project as:
    - Commercial (**pricing plans**) or,
    - Non-commercial (**Unpaid offer based on eligibility**).

- In this guide, we will be registering our cloud project as **non-commercial**.

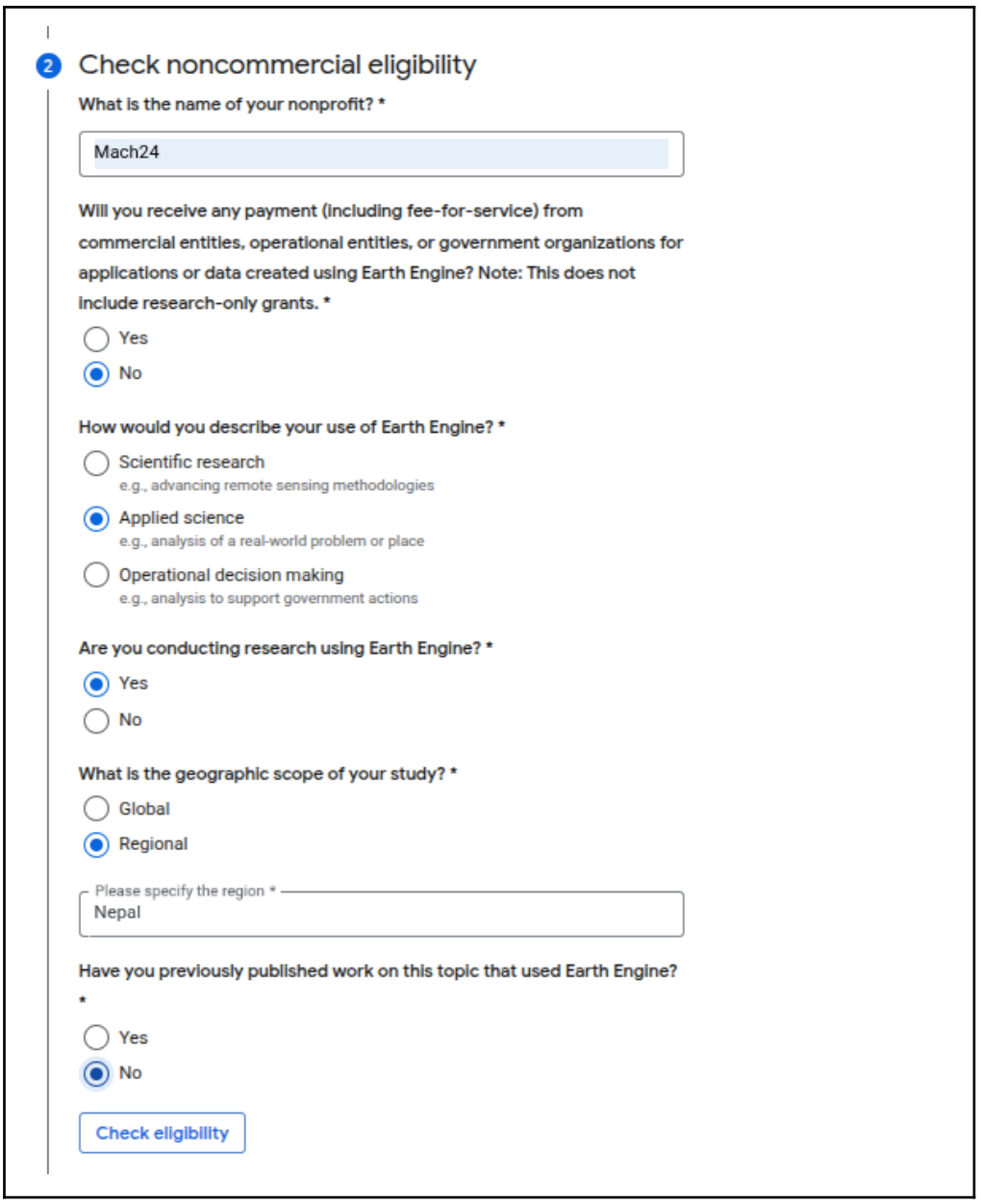


- Click on **Get started.**

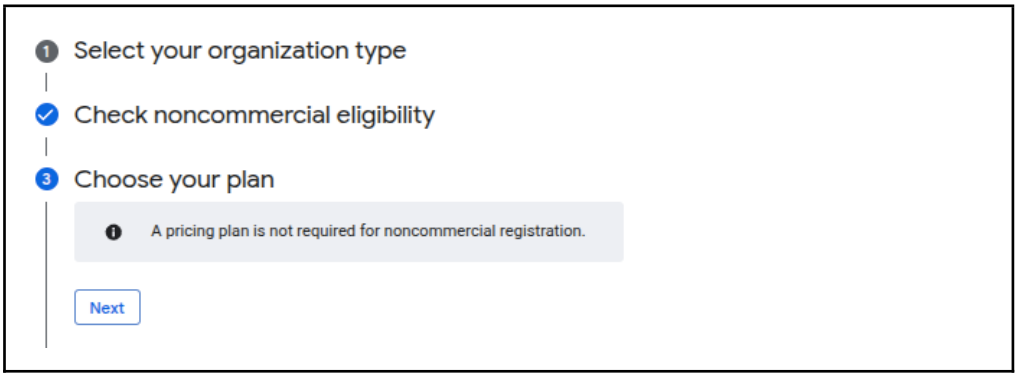
- **Fill in:**
    - Select your organization type:



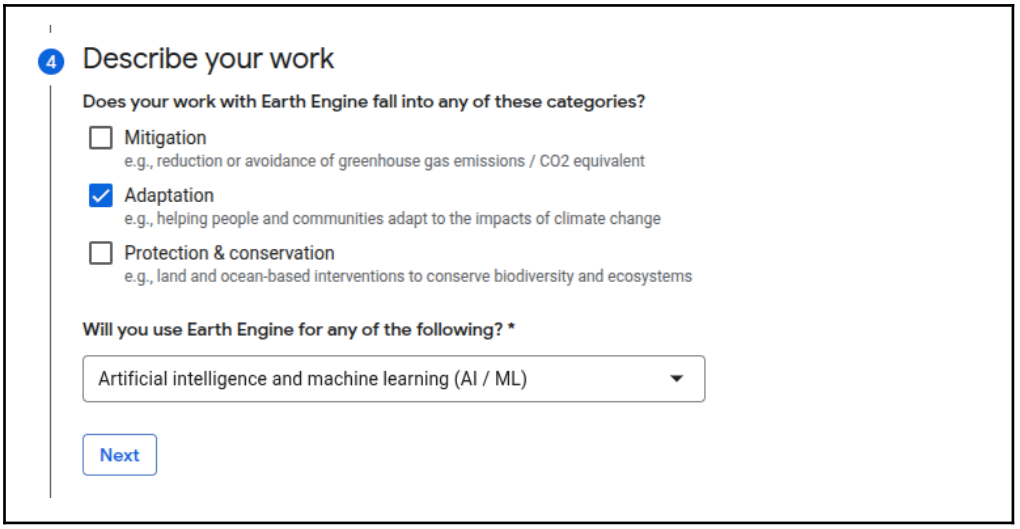
- Check noncommercial eligibility:



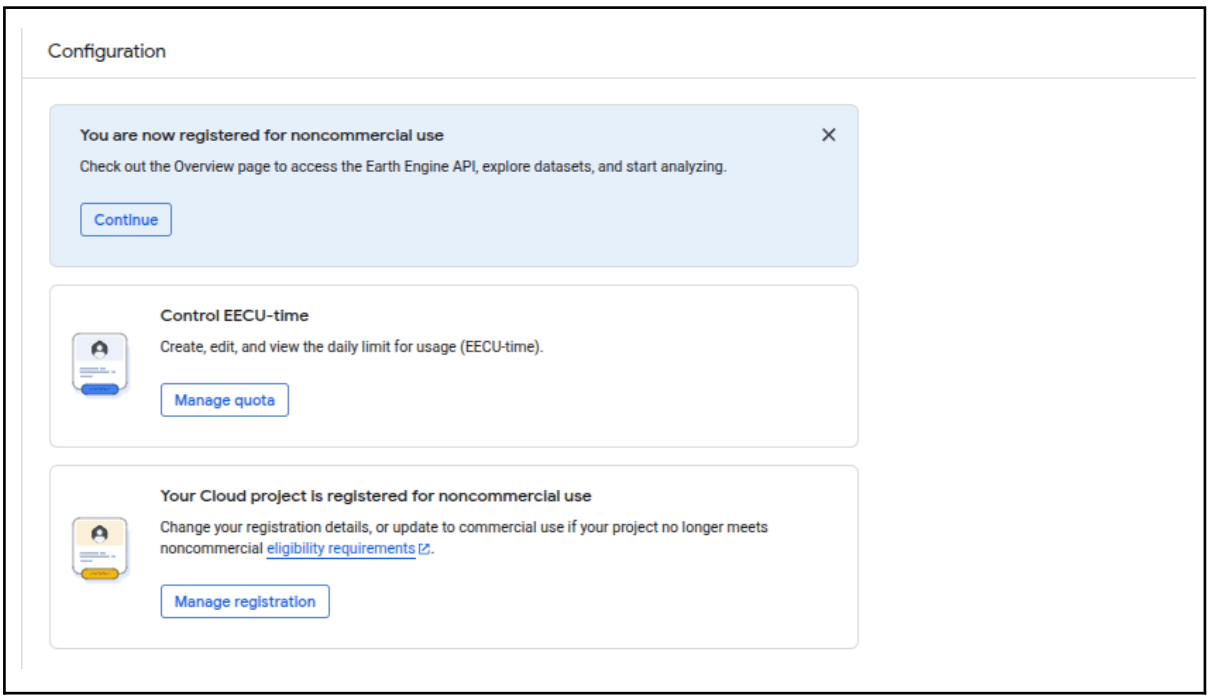
- Choose your plan.



- Describe your work:



- Click on **Next,** and after reading the Review summary, click on **Register.**



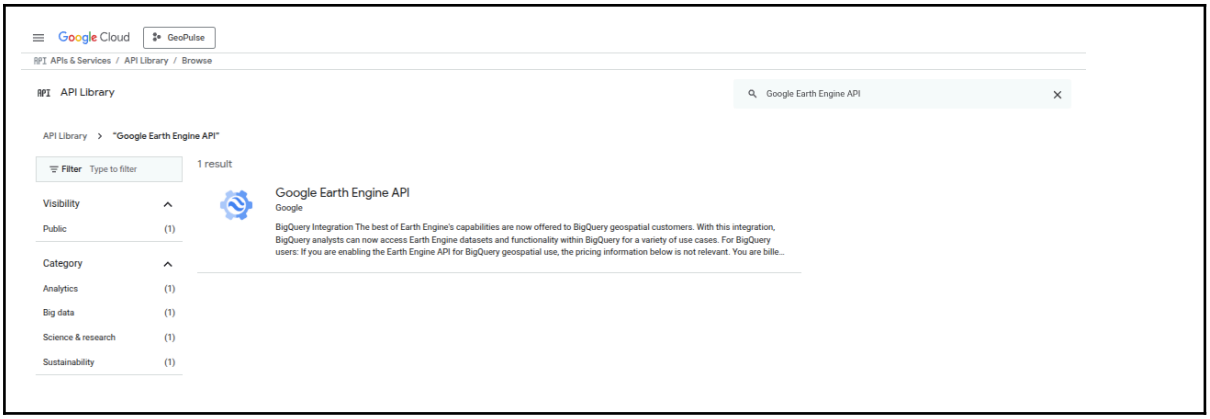
- Your cloud project is now registered for **non-commercial** use.

✅ 3. Enable Earth Engine API

- In **Google Cloud Console** under **Quick Access**, go to **APIs & Services** and then the **Library tab**.

[Google Cloud Console]

- Search for **Google Earth Engine API**.

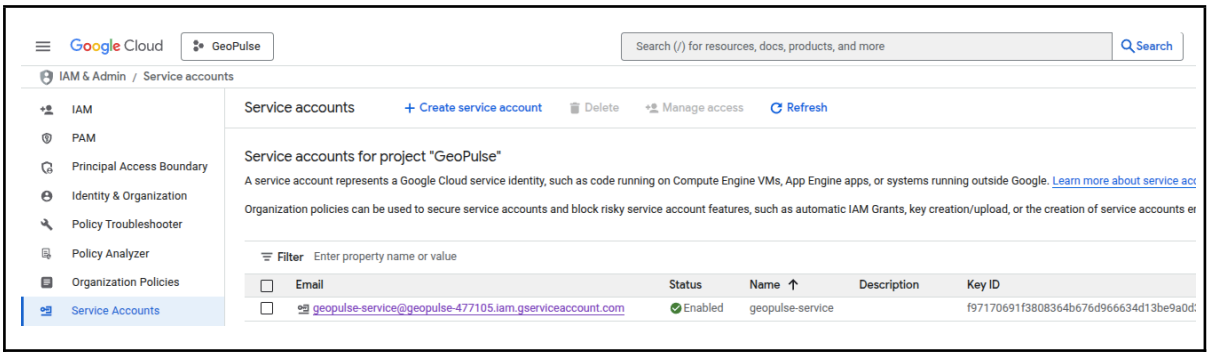


- Click **Enable**.

✅ 4. Create a Service Account

- In **Google Cloud Console**, go to **IAM & Admin → Service Accounts**.

Google Cloud Console



- Click **Create Service Account**.

- Fill in:
  - Service account name: **geopulse-service**
  - Service account ID: **geopulse-service-439** (Auto-generated)**.**
  - This will form the email: **geopulse-service-439@<PROJECT_ID>.iam.gserviceaccount.com**

- Click **Create and Continue.**

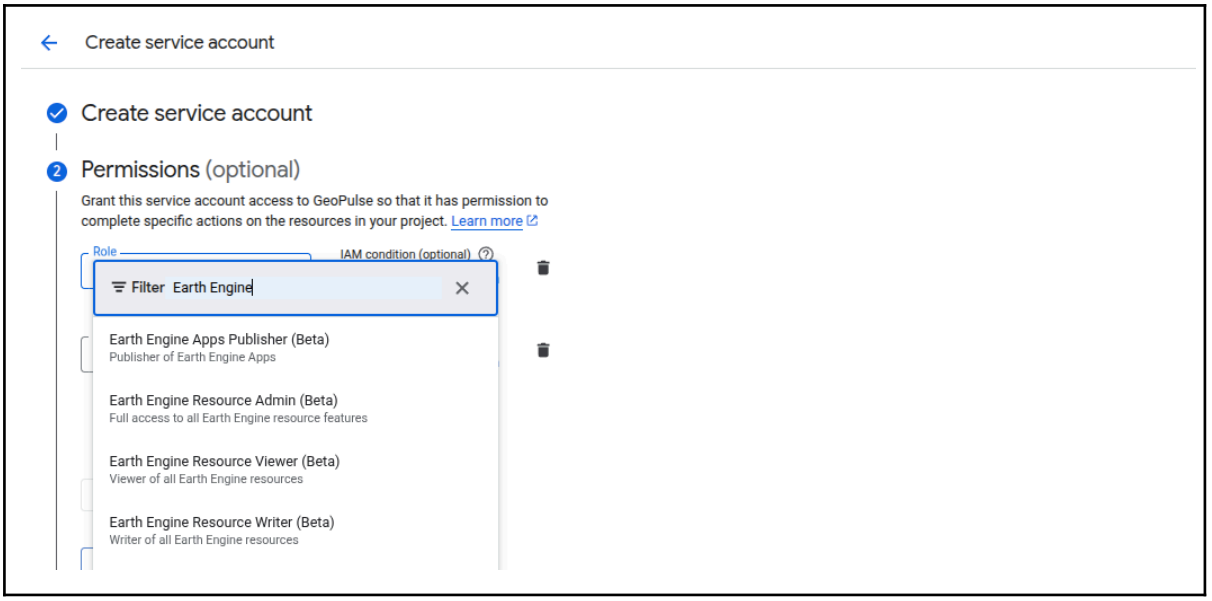
✅ 5. Assign Roles to the Service Account

- After you fill in the **Service account name** and **ID** and click **Create and Continue**, you'll see **"Grant this service account access to the project"** (Permissions) and optionally **Principals with access**.

- In the **Permissions** section, you can assign roles to the service account.

- Assign the roles required for Earth Engine Access:
  - **Earth Engine Resource Viewer** (minimum)
  - **Earth Engine Admin** (if you need full access)

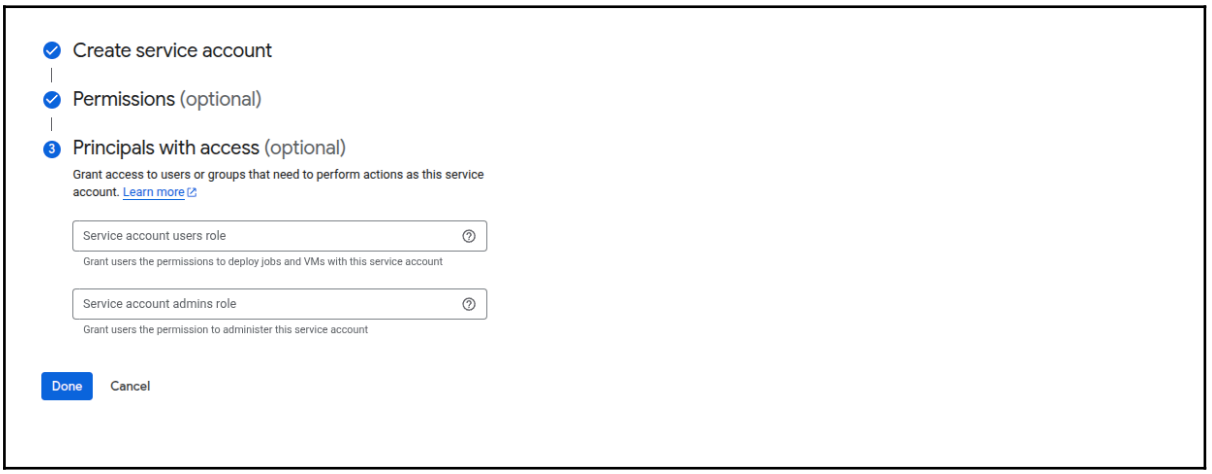
- To assign the above roles, use the **filter** option (type in **Earth Engine**) from the **role** box of the **Permissions.**



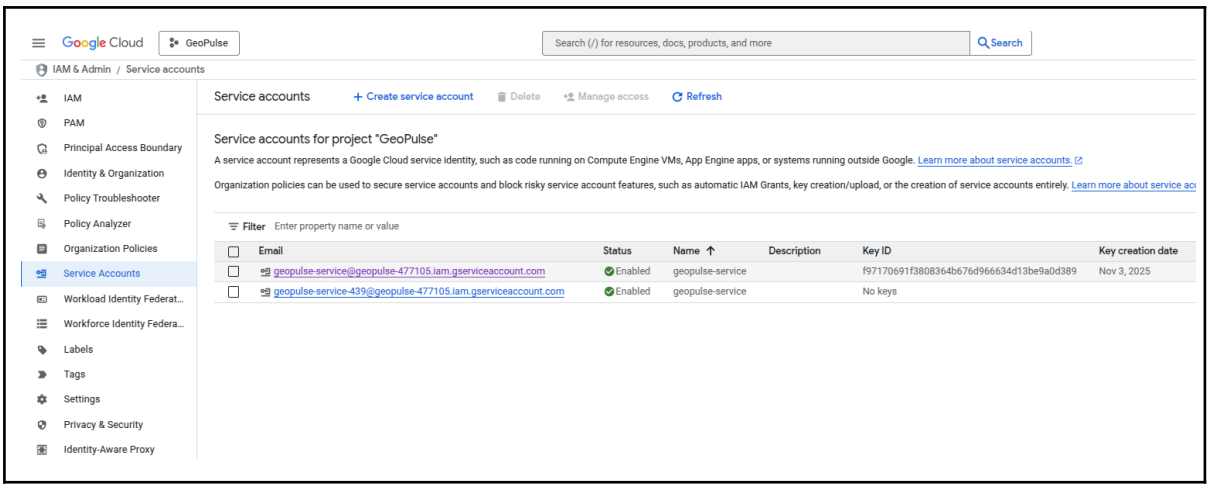
- You'll see roles like:
  - **Earth Engine Resource Viewer.**
  - **Earth Engine Resource Admin.**

- Add **both roles** to the service account, then **press continue.**

- Click **Done.**

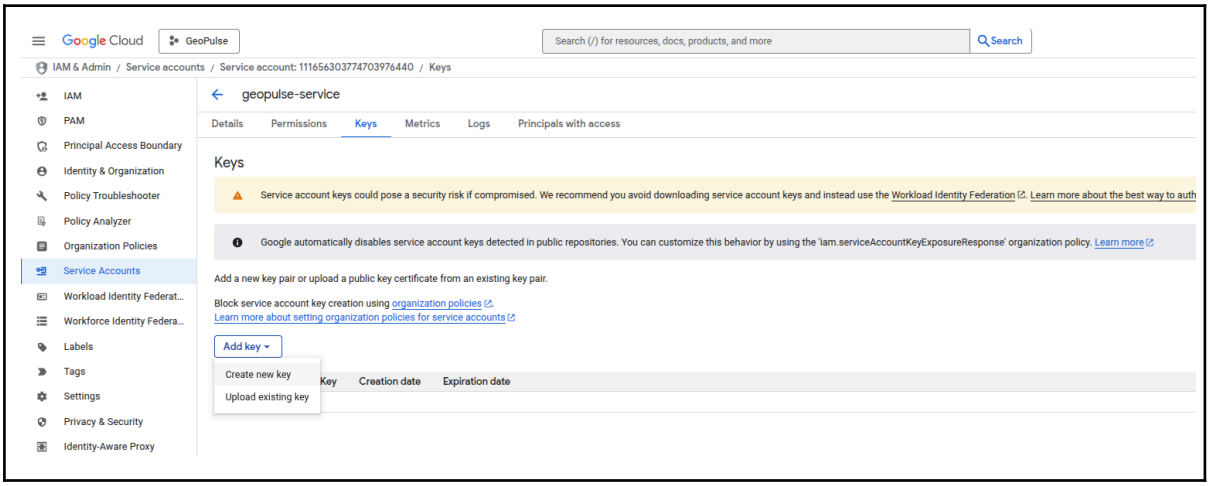


- Check your newly created service account in the **Service Accounts** tab.

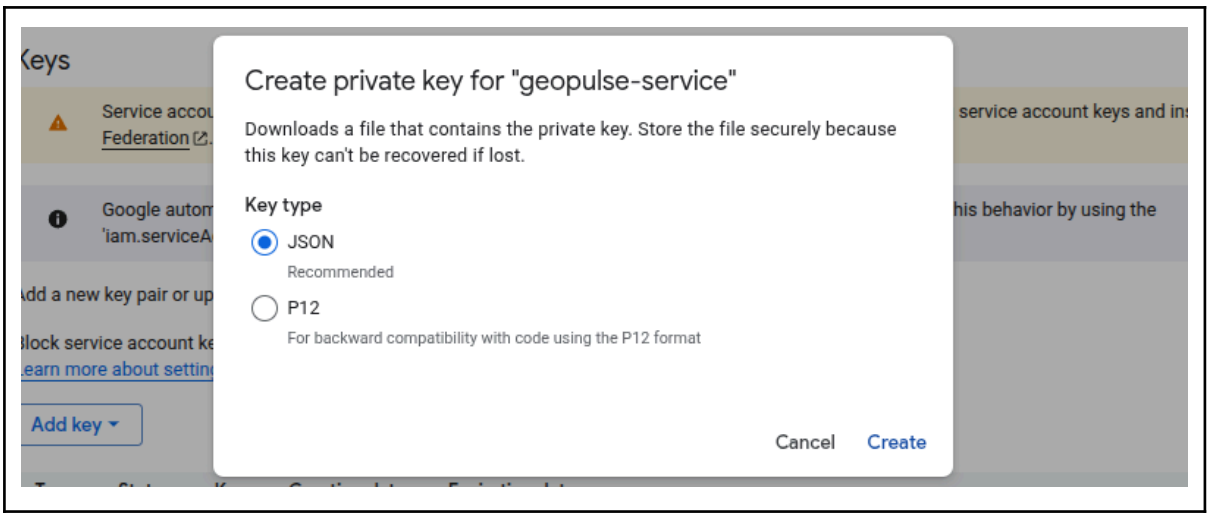


✅ 6. Generate Service Account Key (.json)

- In the **Service Accounts,** select your service account.
- Go to **Keys → Add Key → Create New Key**.



- Choose **JSON** format.



- Click **Create.**
- Your newly created key will be automatically downloaded into your computer in **JSON** format.
- Save the downloaded key as **geopulse-key.json** in your project directory.

Important Tip: Keep this file secret! Do not commit it to version control (e.g., GitHub).

✅ 7. Initialize Earth Engine in Python

- Create a **.env** file in your project root.

```
PROJECT_ID=" Your Previously Created Project Id "
SERVICE_ACCOUNT= "Your service account email (.iam.gserviceaccount.com)"
```

In Python, you can load them with:

```
import os
from dotenv import load_dotenv

load_dotenv()   # Load environment variables from .env
project_id = os.getenv(“PROJECT_ID”)
service_account = os.getenv(“SERVICE_ACCOUNT”)
```

- Initialize Earth Engine in Python using the service account.

```
import ee
import os

credentials = ee.ServiceAccountCredentials(
    os.getenv("SERVICE_ACCOUNT"), "geopulse-key.json"
)
ee.Initialize(credentials, project=os.getenv("PROJECT_ID"))
```

- If the above code does not throw an error, that means:
    - Your **Earth Engine service account** credentials (**geopulse-key.json**) are valid.
    - Your environment variables **SERVICE_ACCOUNT** and **PROJECT_ID** are correctly set.
    - Your **Google Cloud project** is properly linked and has **Earth Engine API enabled**.

In other words, if **ee.Initialize()** runs without raising an exception — your setup is complete and you’re successfully authenticated into **Google Earth Engine**.

🎉 Congratulations!