

Google Earth Engine Service Account Setup

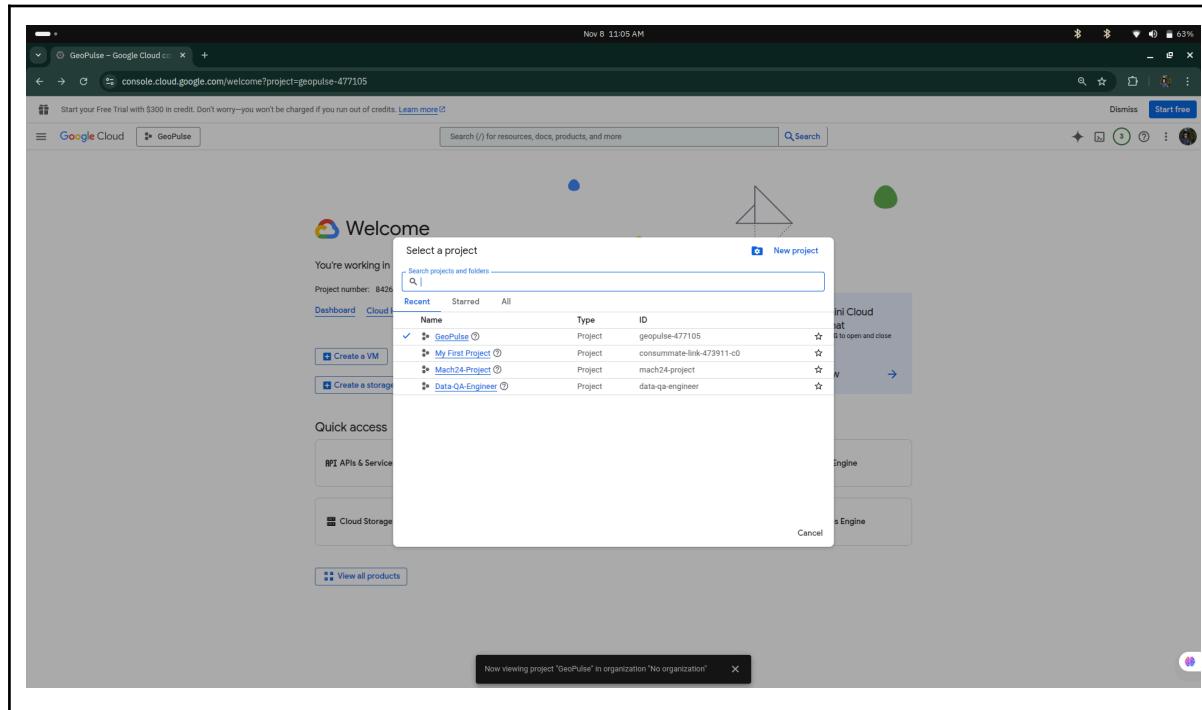
This guide explains how to create a Google Cloud project, set up a service account, and generate the credentials key (key.json) to use with Python scripts for Google Earth Engine (GEE) access.

✓ 1. Create a Google Cloud Project

- Go to **Google Cloud Console**.

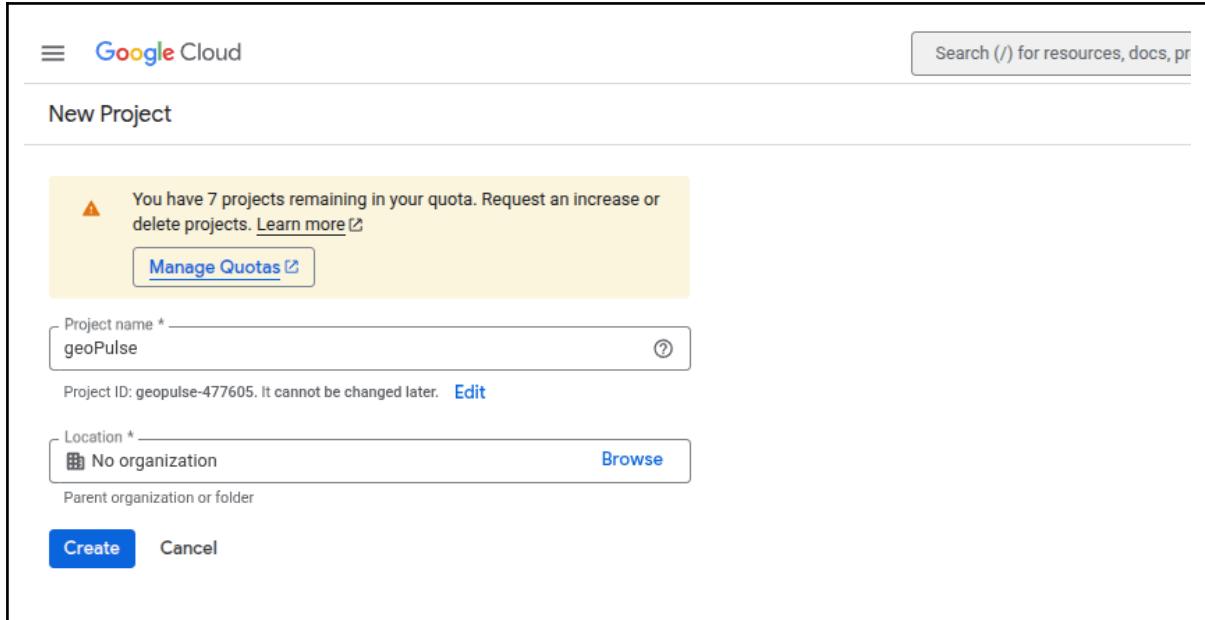
[Google Cloud Console](#)

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

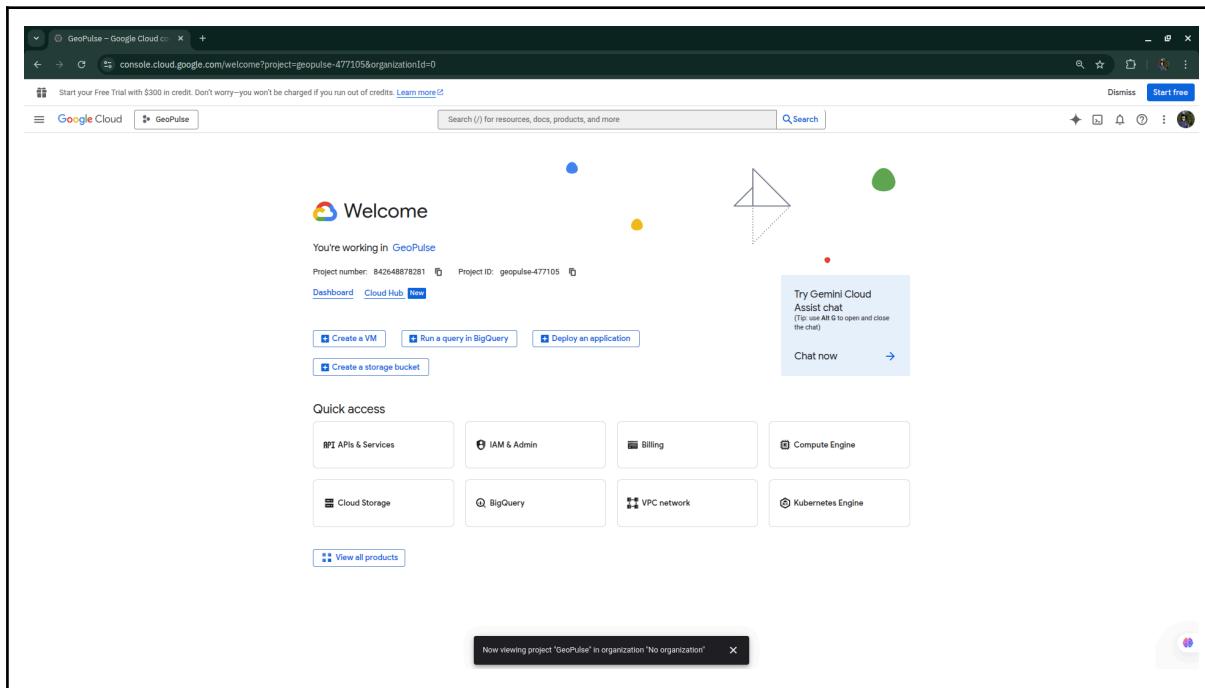


- Click on the **New project**.

- 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.



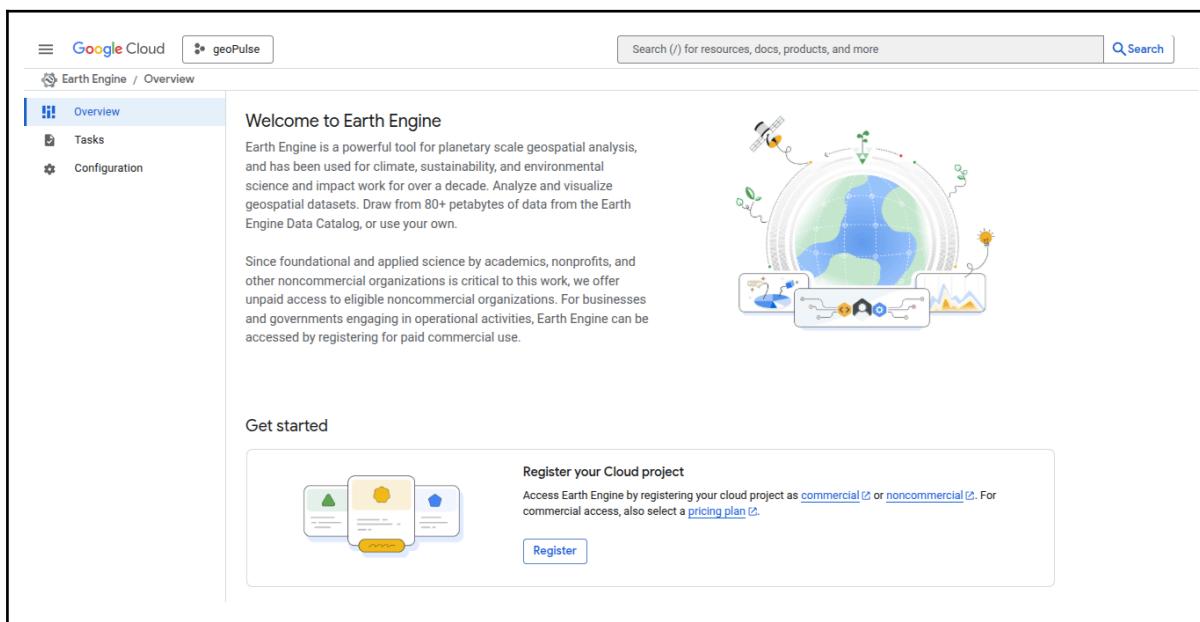
Tip: The project ID is used in our Python code as PROJECT_ID.

✓ 2. Register Your Project for Earth Engine

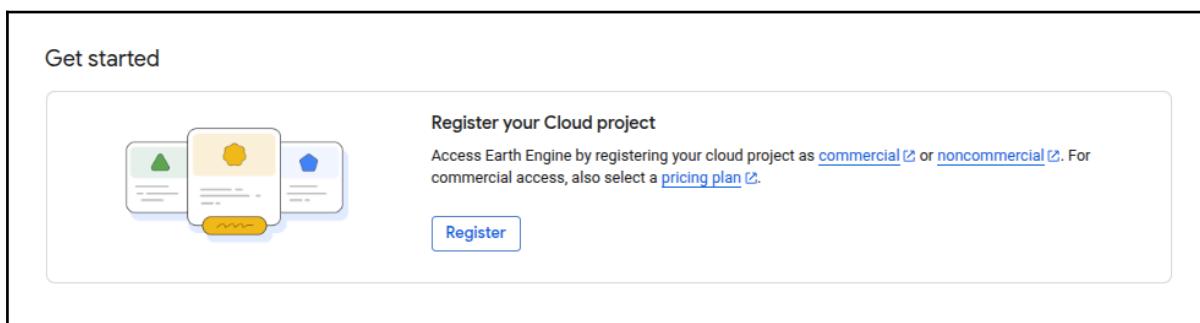
- Go to the **Earth Engine Overview** page.

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

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

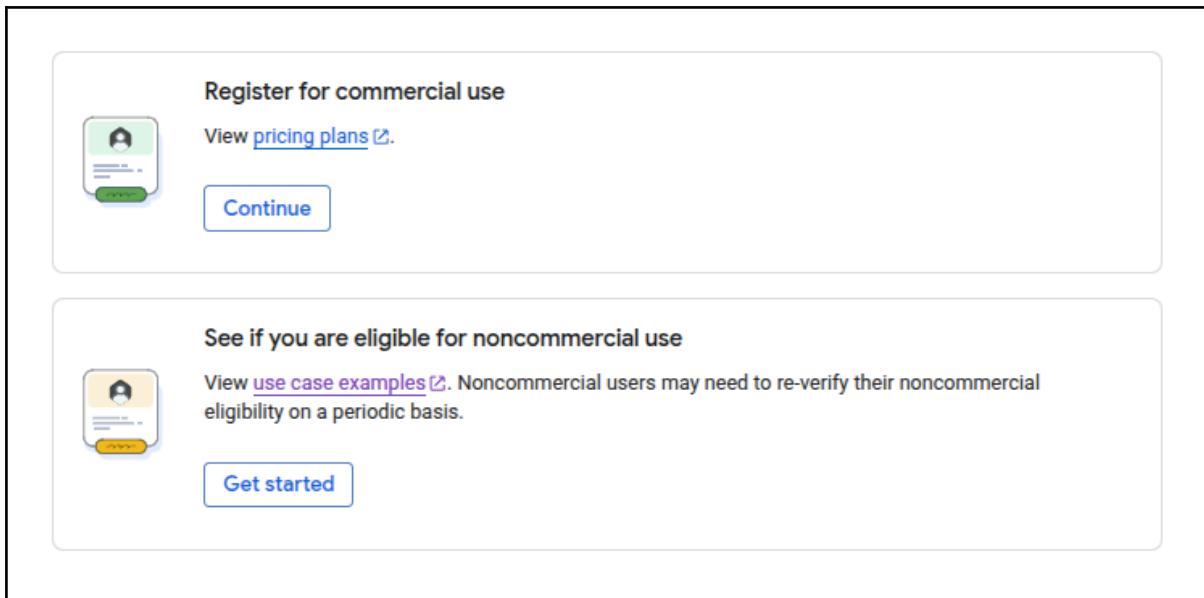


- Register your cloud project by clicking **Register**.



- 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:

This screenshot shows the first step of a form titled '1 Select your organization type'. The question asks, 'Which of the following best describes you or your organization? *'. A dropdown menu is open, showing 'Nonprofit' as the selected option. Below the dropdown is a blue 'Next' button.

1 Select your organization type
Which of the following best describes you or your organization? *

Nonprofit

Next

- Check noncommercial eligibility:

② Check noncommercial eligibility

What is the name of your nonprofit? *

Will you receive any payment (including fee-for-service) from commercial entities, operational entities, or government organizations for applications or data created using Earth Engine? Note: This does not include research-only grants. *

Yes
 No

How would you describe your use of Earth Engine? *

Scientific research
e.g., advancing remote sensing methodologies
 Applied science
e.g., analysis of a real-world problem or place
 Operational decision making
e.g., analysis to support government actions

Are you conducting research using Earth Engine? *

Yes
 No

What is the geographic scope of your study? *

Global
 Regional

Please specify the region * —

Have you previously published work on this topic that used Earth Engine? *

Yes
 No

[Check eligibility](#)

- Choose your plan.

① Select your organization type

Check noncommercial eligibility

③ Choose your plan

ⓘ A pricing plan is not required for noncommercial registration.

[Next](#)

- Describe your work:

④ Describe your work

Does your work with Earth Engine fall into any of these categories?

Mitigation
e.g., reduction or avoidance of greenhouse gas emissions / CO₂ equivalent

Adaptation
e.g., helping people and communities adapt to the impacts of climate change

Protection & conservation
e.g., land and ocean-based interventions to conserve biodiversity and ecosystems

Will you use Earth Engine for any of the following? *

Artificial intelligence and machine learning (AI / ML) ▾

[Next](#)

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

Configuration

You are now registered for noncommercial use X

Check out the Overview page to access the Earth Engine API, explore datasets, and start analyzing.

[Continue](#)

Control EECU-time

Create, edit, and view the daily limit for usage (EECU-time).
[Manage quota](#)

Your Cloud project is registered for noncommercial use

Change your registration details, or update to commercial use if your project no longer meets noncommercial [eligibility requirements](#).
[Manage registration](#)

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

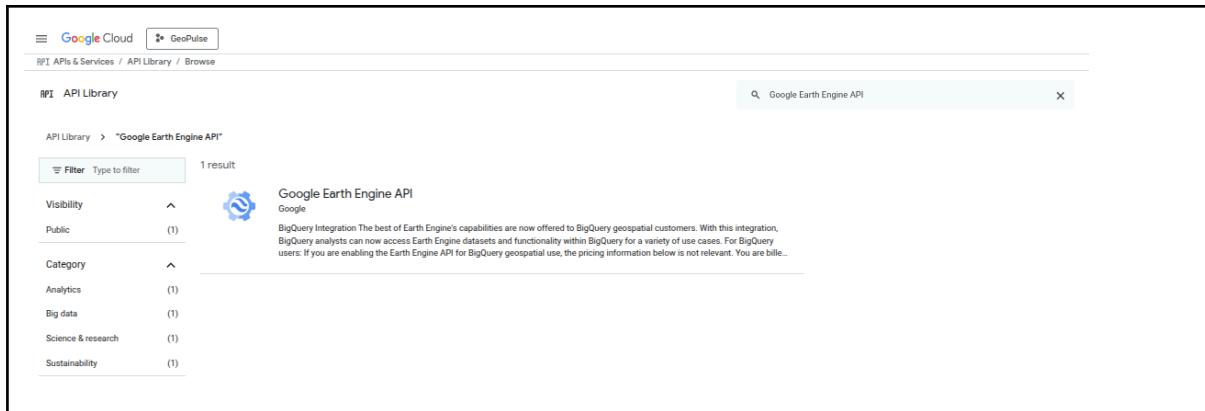
✓ 3. Enable Earth Engine API

- 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](#)

The screenshot shows the Google Cloud Console interface. At the top, there's a navigation bar with 'Google Cloud' and 'GeoPulse' selected. Below it, a search bar says 'Search (/) for resources, docs, products, and more'. The main area is titled 'Service accounts' under 'IAM & Admin / Service accounts'. It shows a list of service accounts for the project 'GeoPulse'. One account is listed: 'geopulse-service@geopulse-477105.iam.gserviceaccount.com'. The account is enabled and has the name 'geopulse-service'. A key ID is also visible.

- 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**.

[Create service account](#)

2 Permissions (optional)

Grant this service account access to GeoPulse so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

Role	IAM condition (optional)
Filter Earth Engine	X
Earth Engine Apps Publisher (Beta) Publisher of Earth Engine Apps	
Earth Engine Resource Admin (Beta) Full access to all Earth Engine resource features	
Earth Engine Resource Viewer (Beta) Viewer of all Earth Engine resources	
Earth Engine Resource Writer (Beta) Writer of all Earth Engine resources	

- 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**.

2 Create service account

3 Permissions (optional)

3 Principals with access (optional)

Grant access to users or groups that need to perform actions as this service account. [Learn more](#)

Service account users role	?
Grant users the permissions to deploy jobs and VMs with this service account	
Service account admins role	?
Grant users the permission to administer this service account	

Done **Cancel**

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

The screenshot shows the Google Cloud IAM & Admin / Service accounts interface. On the left, there's a sidebar with various options like IAM, PAM, Principal Access Boundary, etc. The 'Service Accounts' option is selected. The main area displays a table of service accounts for the project "GeoPulse". There are two entries:

Email	Status	Name	Description	Key ID	Key creation date
geopulse-service@geopulse-477105.iam.gserviceaccount.com	Enabled	geopulse-service		f97170691f3808364b676d966634d13be9a0d389	Nov 3, 2025
geopulse-service-439@geopulse-477105.iam.gserviceaccount.com	Enabled	geopulse-service	No keys		

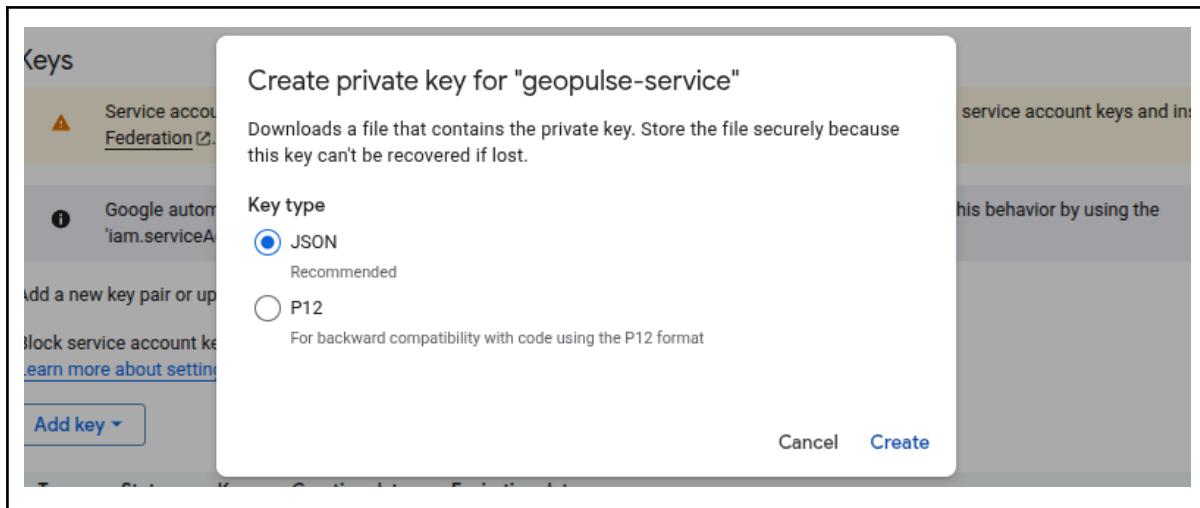
✓ 6. Generate Service Account Key (.json)

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

The screenshot shows the Google Cloud IAM & Admin / Service accounts / geopulse-service / Keys page. The sidebar shows the 'Service Accounts' option is selected. The main area has tabs for Details, Permissions, Keys, Metrics, Logs, and Principals with access. The 'Keys' tab is active. It contains a warning about service account keys being a security risk if compromised. Below that, it says "Google automatically disables service account keys detected in public repositories. You can customize this behavior by using the 'iam.serviceAccountKeyExposureResponse' organization policy." At the bottom, there's a "Create new key" button and a table for managing keys.

Create new key	Key	Creation date	Expiration date
Upload existing 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**.

