

# Lecture 01

## GEE Introduction:

*setup, datasets, image visualization*

2025-10-03

Sébastien Valade



## 1. Introduction

1. GEE overview
2. JavaScript API: Earth Engine Code Editor
3. Python API: Google Colaboratory

2. Setup GEE in GoogleColab

3. GEE quick start

## 1.1. GEE overview

- **Google Earth Engine (GEE)** is a cloud-based computing platform for processing satellite imagery and other geospatial datasets.
- Provides access to:
  - large database of satellite imagery (including NASA, USGS, ESA, and other satellite missions)
  - large computational power needed to analyze those images
- Provides API (Application Programming Interfaces) for making requests to the servers in:
  - JavaScript              ⇒ [Earth Engine Code Editor](#)
  - Python                 ⇒ [Google Colaboratory](#)

## 1.1. GEE overview

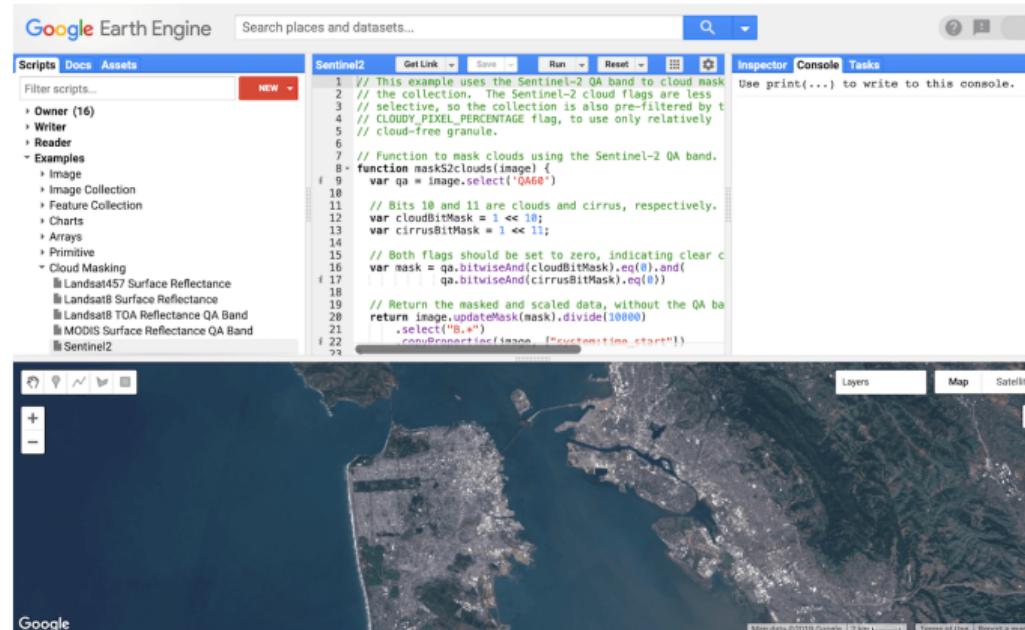
- **Google Earth Engine** (GEE) is a cloud-based computing platform for processing satellite imagery and other geospatial datasets.
- Provides access to:
  - large database of satellite imagery (including NASA, USGS, ESA, and other satellite missions)
  - large computational power needed to analyze those images
- Provides API (Application Programming Interfaces) for making requests to the servers in:
  - JavaScript              ⇒ [Earth Engine Code Editor](#)
  - Python                 ⇒ [Google Colaboratory](#)

- **Google Earth Engine** (GEE) is a cloud-based computing platform for processing satellite imagery and other geospatial datasets.
- Provides access to:
  - large database of satellite imagery (including NASA, USGS, ESA, and other satellite missions)
  - large computational power needed to analyze those images
- Provides API (Application Programming Interfaces) for making requests to the servers in:
  - JavaScript              ⇒ [Earth Engine Code Editor](#)
  - Python                 ⇒ [Google Colaboratory](#)

## 1.2. JavaScript API: Earth Engine Code Editor

## 1. Earth Engine Code Editor (JavaScript API)

⇒ free web-based IDE (*Integrated Development Environment*) using the JavaScript API



The screenshot shows the Google Earth Engine Code Editor interface. At the top, there's a navigation bar with 'Google Earth Engine' and a search bar labeled 'Search places and datasets...'. Below the search bar are tabs for 'Scripts', 'Docs', and 'Assets', with 'Scripts' currently selected. A 'NEW' button is also visible. On the left, a sidebar titled 'Sentinel2' contains a tree view of scripts: 'Owner (16)', 'Writer', 'Reader', 'Examples' (which includes 'Image', 'Image Collection', 'Feature Collection', 'Charts', 'Arrays', 'Primitive', and 'Cloud Masking' with sub-options like 'Landsat457 Surface Reflectance'), and 'Sentinel2'. The main area is a code editor with the following JavaScript code:

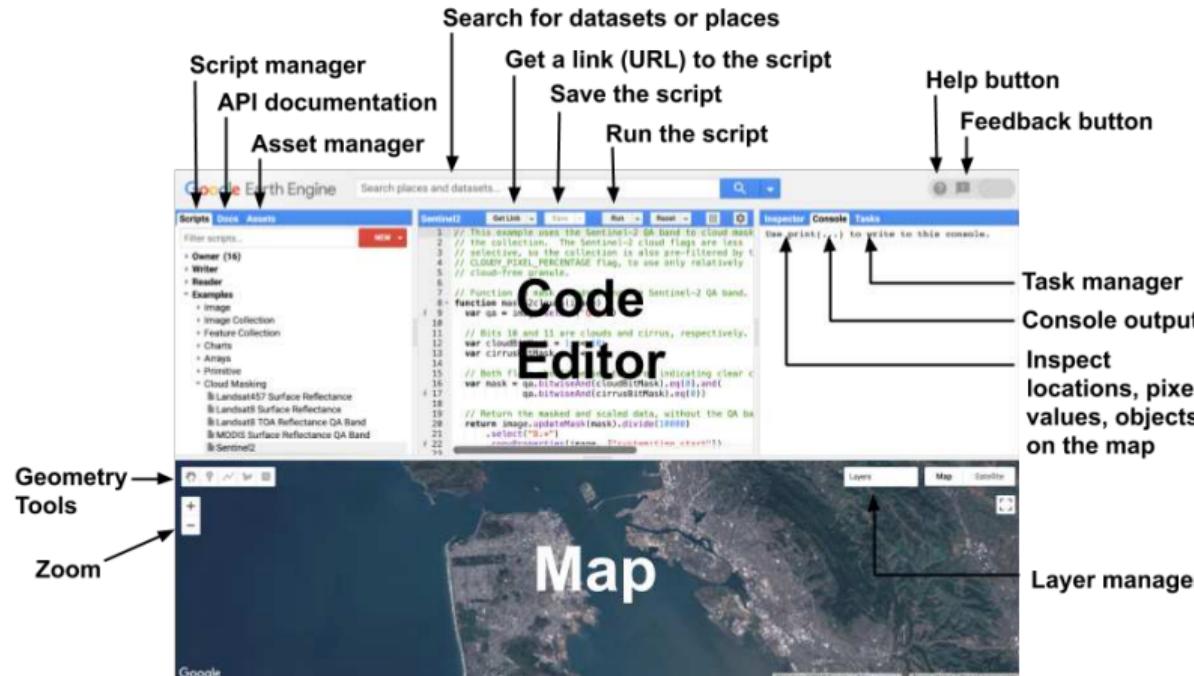
```
// This example uses the Sentinel-2 QA band to cloud mask  
// the collection. The Sentinel-2 cloud flags are less  
// selective, so the collection is also pre-filtered by t  
// CLOUDY_PIXEL_PERCENTAGE flag, to use only relatively  
// cloud-free granule.  
  
// Function to mask clouds using the Sentinel-2 QA band.  
function maskS2clouds(image) {  
  var qa = image.select('QA60');  
  
  // Bits 10 and 11 are clouds and cirrus, respectively.  
  var cloudBitMask = 1 << 10;  
  var cirrusBitMask = 1 << 11;  
  
  // Both flags should be set to zero, indicating clear c  
  var mask = qa.bitwiseAnd(cloudBitMask).eq(0).and(  
    qa.bitwiseAnd(cirrusBitMask).eq(0))  
  
  // Return the masked and scaled data, without the QA ba  
  return image.updateMask(mask).divide(10000)  
    .select(['B.*'])  
    .copyProperties(image, ['customtime_start']);  
}
```

Below the code editor is a map viewer showing a satellite image of a coastal area. The map includes zoom controls (+, -, ×), a location pin, and a compass rose. At the bottom, there are buttons for 'Layers', 'Map', and 'Satellite', along with a 'Google' logo and links for 'Map data ©2019 Google | 2 km | Terms of Use | Report a map error'.

## 1.2. JavaScript API: Earth Engine Code Editor

## 1. Earth Engine Code Editor (JavaScript API)

⇒ free web-based IDE (*Integrated Development Environment*) using the JavaScript API



## 1.3. Python API: Google Colaboratory

### 2. Google Colaboratory (Python API)

⇒ free cloud-based Jupyter notebook environment for writing and executing Python code

⇒ avoids the need to set up a local development environment, i.e. software (libraries) & hardware (GPU)

⇒ provides access to GEE Python API, free GPU and TPU resources, enabling users to perform computationally intensive tasks

The screenshot shows the Google Colaboratory interface in a web browser. The title bar says "Welcome To Colaboratory". The address bar shows the URL <https://colab.research.google.com>. The main content area displays the "Welcome To Colaboratory" page, which includes a "Table of contents" sidebar with sections like "Getting started", "Data science", "Machine learning", and "More Resources". The main content area has a heading "What is Colab?", followed by a list of benefits: "Zero configuration required", "Access to GPUs free of charge", and "Easy sharing". It also mentions that Colab can make work easier for students, data scientists, and AI researchers. Below this is a section titled "Getting started" with a note about interactive environments and code cells. A code cell example is shown with the following Python script:

```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
86400
```

Instructions for executing the code are provided, along with a note that variables defined in one cell can be used in others.

## 1.3. Python API: Google Colaboratory

### 2. Google Colaboratory (Python API)

- ⇒ free cloud-based Jupyter notebook environment for writing and executing Python code
- ⇒ avoids the need to set up a local development environment, i.e. software (libraries) & hardware (GPU)
- ⇒ provides access to GEE Python API, free GPU and TPU resources, enabling users to perform computationally intensive tasks

The screenshot shows the Google Colaboratory interface. At the top, there's a header bar with a logo, a title 'Welcome To Colaboratory', and a URL 'https://colab.research.google.com'. Below the header is a navigation menu with options like File, Edit, View, Insert, Runtime, Tools, Help, and a 'Share' button. On the left, there's a sidebar titled 'Table of contents' with sections for Getting started, Data science, Machine learning, More Resources, and Featured examples. A 'Section' button is at the bottom of this sidebar. The main content area has a heading 'What is Colab?' followed by a list of benefits: Zero configuration required, Access to GPUs free of charge, and Easy sharing. It also mentions that Colab can make work easier for students, data scientists, and AI researchers. Below this is a section titled 'Getting started' with a note about it being an interactive environment. It shows a code cell with the following Python script:

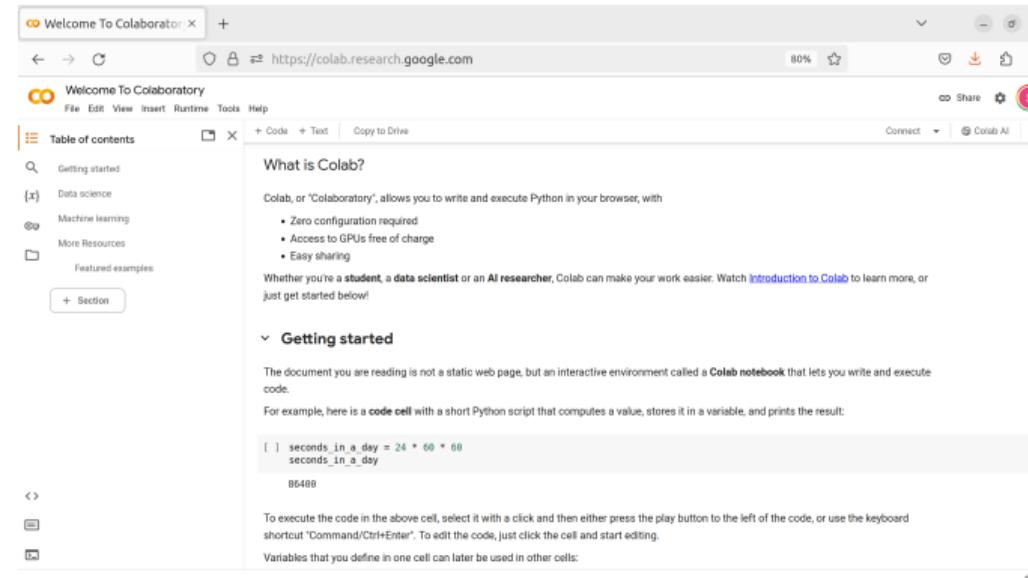
```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
86400
```

A note says to execute the code by selecting it and pressing the play button or keyboard shortcut 'Command/Ctrl+Enter'. It also mentions that variables defined in one cell can be used in others.

## 1.3. Python API: Google Colaboratory

### 2. Google Colaboratory (Python API)

- ⇒ free cloud-based Jupyter notebook environment for writing and executing Python code
- ⇒ avoids the need to set up a local development environment, i.e. software (libraries) & hardware (GPU)
- ⇒ provides access to GEE Python API, free GPU and TPU resources, enabling users to perform computationally intensive tasks



## 1. Introduction

## 2. Setup GEE in GoogleColab

1. Create a Google account
2. Create a Google Cloud project & enable GEE API
3. Register Google Cloud project for use with GEE
4. Access GEE in JavaScript IDE
5. Access GEE in Colab

## 3. GEE quick start

### Nota Bene

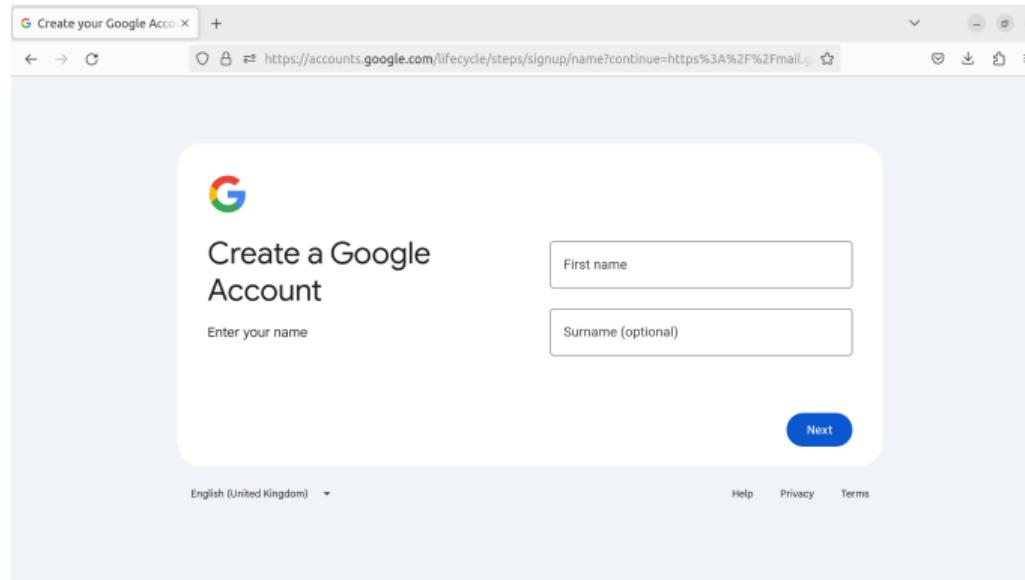
The steps required by Google to access and use GEE APIs are regularly evolving.

⇒ the steps described are those required as of September-2025

⇒ visit the [Earth Engine access guidelines](#) for the most up-to-date information

## 2.1. Create a Google account

## 1. Create a Google account (if you have one, skip this step)



## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

## 2.1 Access your account's Google Cloud Console

The screenshot shows the Google Cloud Welcome page. At the top right, there is a purple circular icon with a white letter 'D'. A red arrow points from the text '1. select your account here' to this icon. The page features a 'Welcome' section with a 'Try Google Cloud with \$300 in free credits' offer. It lists three benefits with checkmarks: 'Access to Google Cloud products and services', '90 days to spend your credits', and 'No billing during trial'. Below this is a 'TRY FOR FREE' button. To the right, there are sections for 'Other options' (Vertex AI Free Tier) and 'GOOGLE CLOUD SETUP'. At the bottom, there's a 'Popular getting started resources' section with a 'General' tab selected, and a 'Pre-built solution templates' section.

1. select your account here

Welcome

Try Google Cloud with \$300 in free credits

- ✓ Access to Google Cloud products and services
- ✓ 90 days to spend your credits
- ✓ No billing during trial

TRY FOR FREE

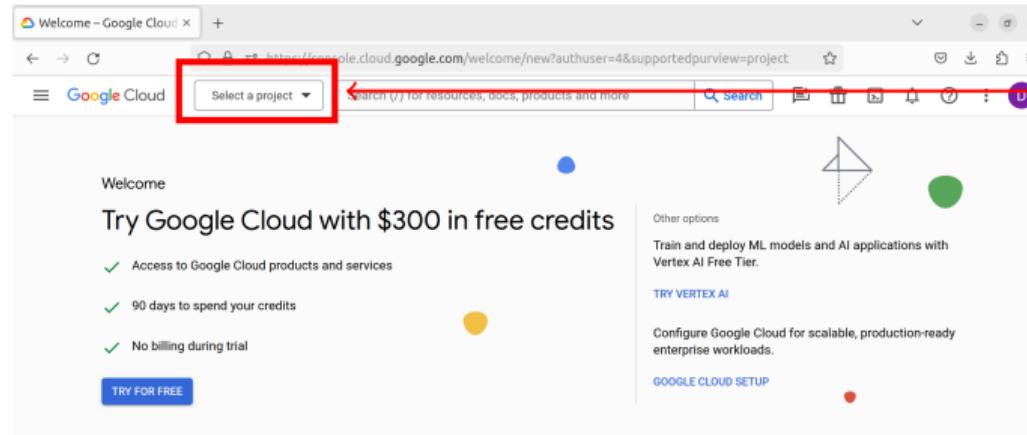
Popular getting started resources

Filter by: Web, mobile, game, storage | Containers, VMs, hybrid/multi, move workload | Data, AI/ML, SAP | Maps, APIs | General

Pre-built solution templates

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

2.2 Create a new project in your [Google Cloud Console](#)

2. select a project

Welcome

Try Google Cloud with \$300 in free credits

- ✓ Access to Google Cloud products and services
- ✓ 90 days to spend your credits
- ✓ No billing during trial

TRY FOR FREE

Other options

Train and deploy ML models and AI applications with Vertex AI Free Tier.

TRY VERTEX AI

Configure Google Cloud for scalable, production-ready enterprise workloads.

GOOGLE CLOUD SETUP

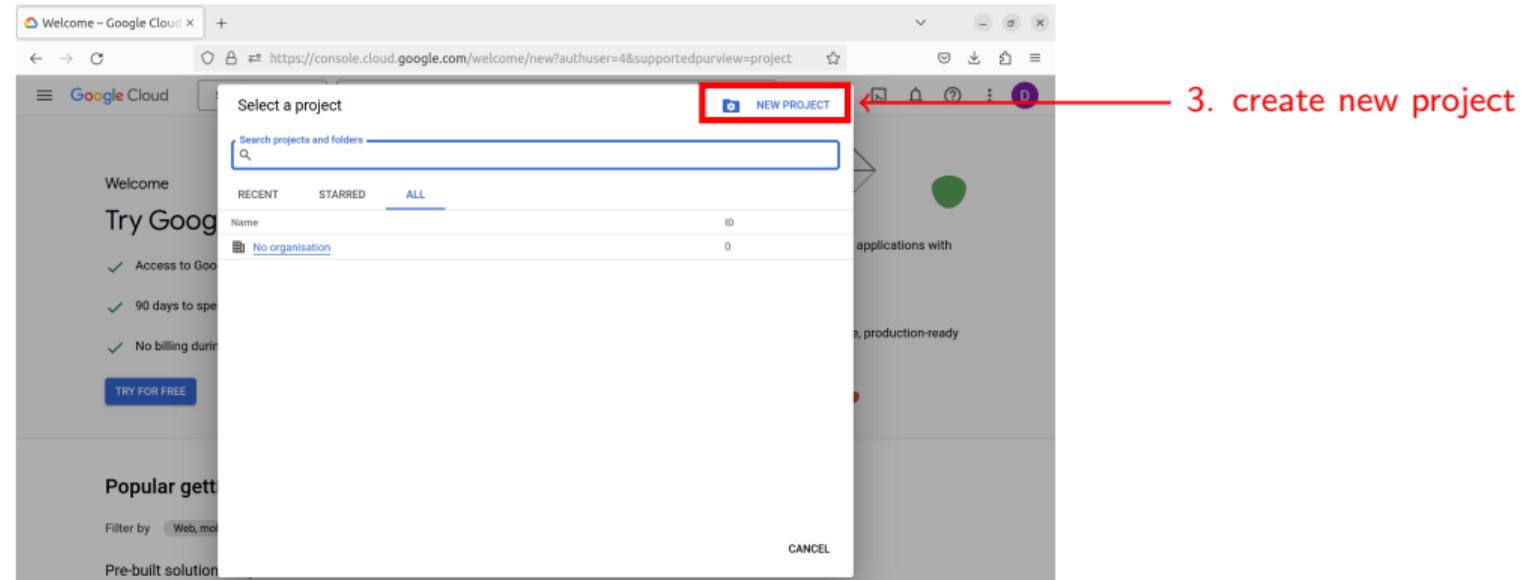
Popular getting started resources

Filter by Web, mobile, game, storage Containers, VMs, hybrid/multi, move workload Data, AI/ML, SAP Maps, APIs General

Pre-built solution templates

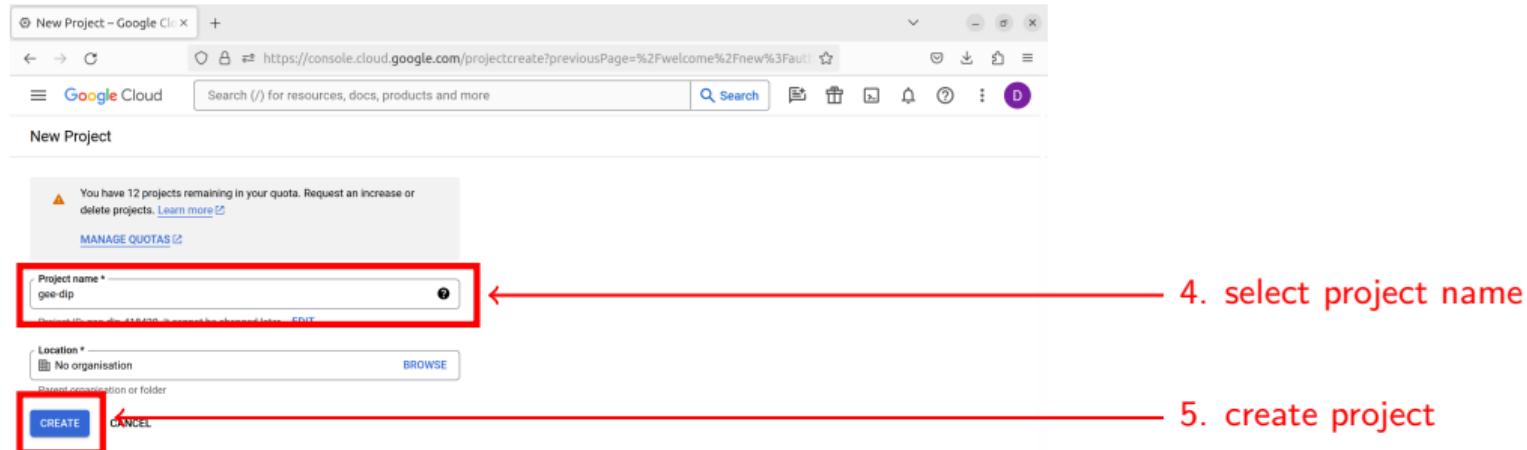
## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

2.2 Create a new project in your [Google Cloud Console](#)

## 2.2. Create a Google Cloud project &amp; enable GEE API

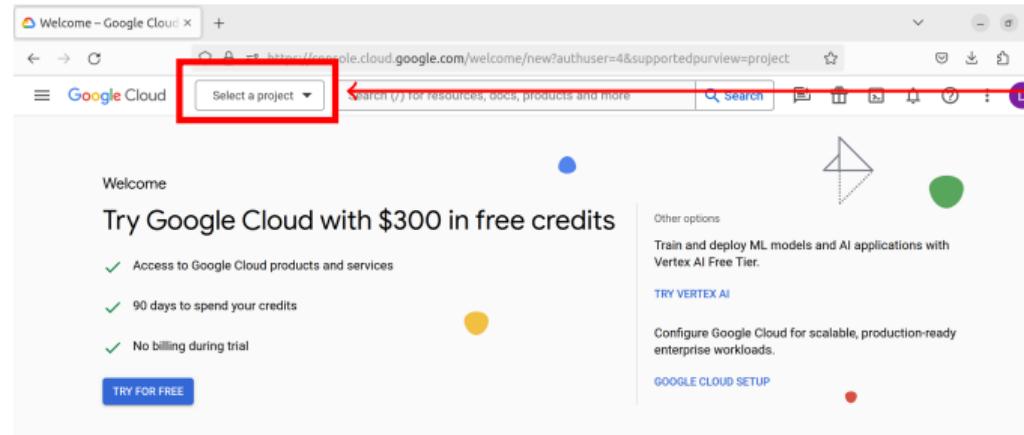
## 2. Create a Google Cloud project &amp; enable GEE API

2.2 Create a new project in your [Google Cloud Console](#)

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

## 2.3 Enable GEE API in the newly created project



6. select the project

Welcome

Try Google Cloud with \$300 in free credits

- ✓ Access to Google Cloud products and services
- ✓ 90 days to spend your credits
- ✓ No billing during trial

TRY FOR FREE

Other options

Train and deploy ML models and AI applications with Vertex AI Free Tier.

TRY VERTEX AI

Configure Google Cloud for scalable, production-ready enterprise workloads.

GOOGLE CLOUD SETUP

Popular getting started resources

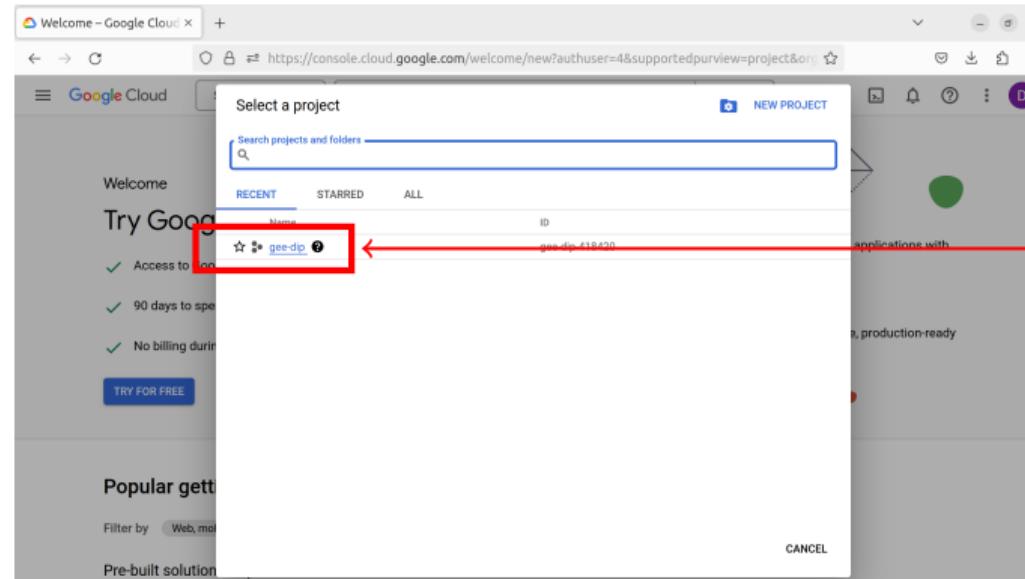
Filter by Web, mobile, game, storage Containers, VMs, hybrid/multi, move workload Data, AI/ML, SAP Maps, APIs General

Pre-built solution templates

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

## 2.3 Enable GEE API in the newly created project

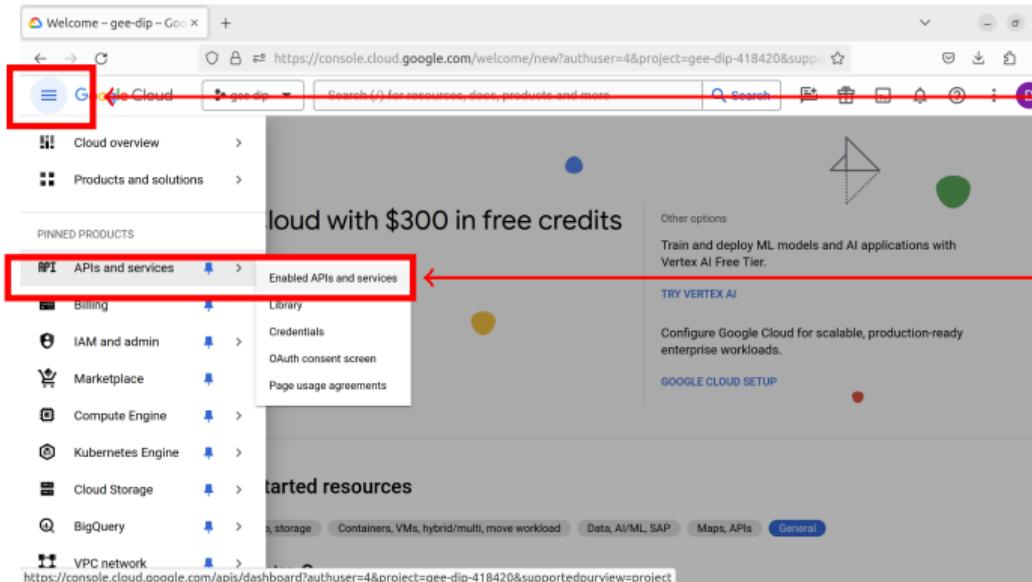


7. select the project

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project & enable GEE API

### 2.3 Enable GEE API in the newly created project



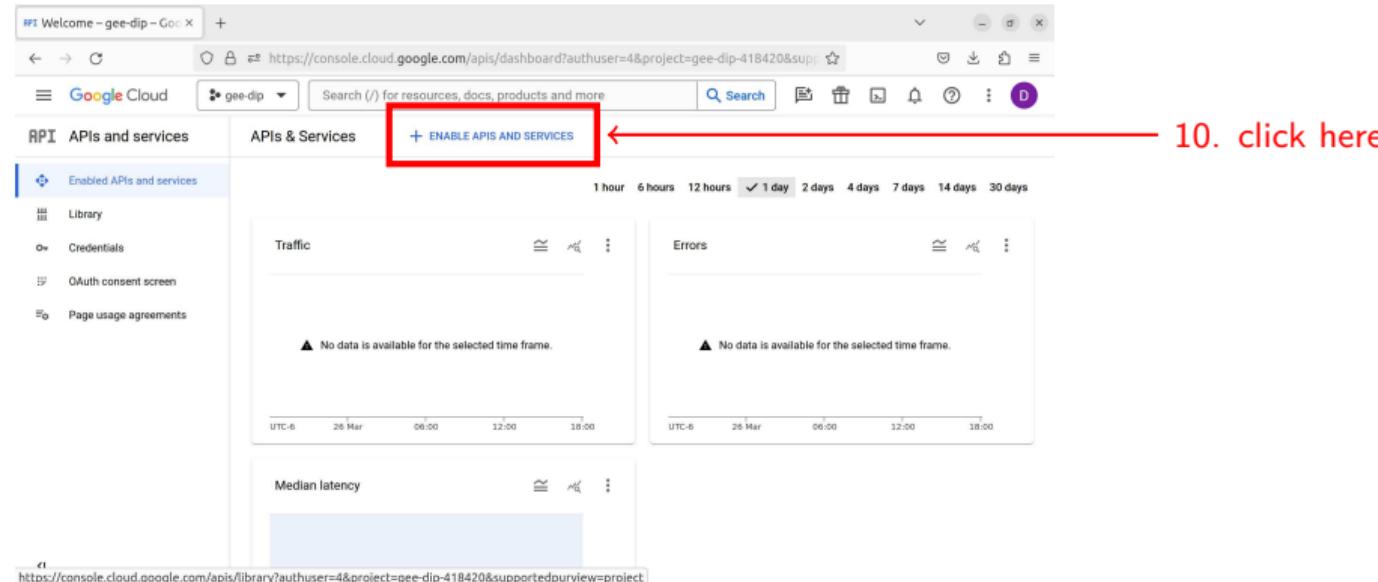
8. select navigation menu

9. "Enable API and services"

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

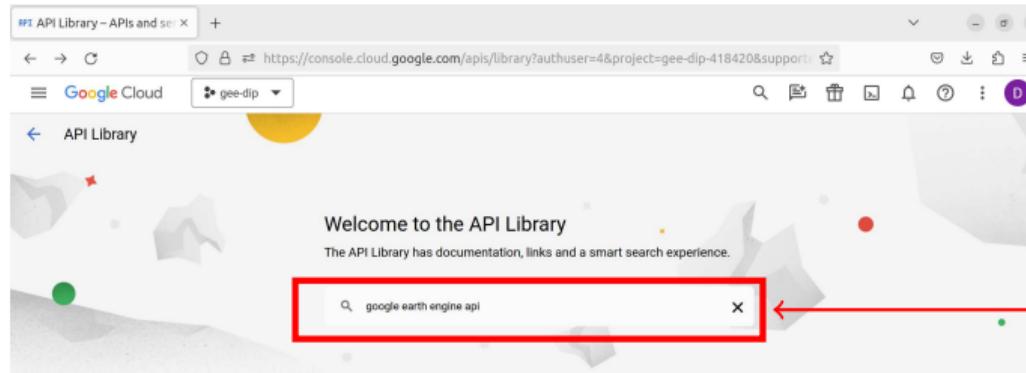
## 2.3 Enable GEE API in the newly created project



## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project & enable GEE API

### 2.3 Enable GEE API in the newly created project



11. search/select “Google Earth Engine API”

The screenshot shows the Google Cloud API Library interface. A red box highlights the search bar at the top, which contains the text "google earth engine api". A red arrow points from the right side of the image towards the search bar. Below the search bar, there is a "VIEW ALL (23)" link. On the left, there is a sidebar with filters for "Visibility" (Public: 452, Private: 2) and "Category". The main area displays a grid of API entries under the "Maps" category. Each entry includes an icon, the API name, and the provider.

Category	API Name	Provider
Maps	Maps SDK for Android	Google
Maps	Maps SDK for iOS	Google
Maps	Maps JavaScript API	Google

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

## 2.3 Configure the newly created project

The screenshot shows a browser window titled "APIs and services - gee-dip" with the URL <https://console.cloud.google.com/apis/library/browse?authuser=4&project=gee-dip-418420&supportedpurview=project>. The search bar contains "google earth engine api". The main content area displays the "API Library > 'google earth engine api'" results. A single result, "Google Earth Engine API" by Google, is shown. This result is highlighted with a red box and a red arrow pointing to it from the right, labeled "12. click here". To the left of the result, there are filters for "Visibility" (Public) and "Category" (Analytics, Big data, Science & research, Climate). The URL at the bottom of the page is <https://console.cloud.google.com/apis/library/earthengine.googleapis.com?authuser=4&project=gee-dip-418420&supportedpurview=project>.

## 2.2. Create a Google Cloud project &amp; enable GEE API

## 2. Create a Google Cloud project &amp; enable GEE API

## 2.3 Configure the newly created project

The screenshot shows a browser window with the URL <https://console.cloud.google.com/apis/library/earthengine.googleapis.com?authuser=4&project=gee-dip>. The page title is "Google Earth Engine API". The main content area displays the "Google Earth Engine API" logo and the tagline "Geospatial insights for a more sustainable world." Below the logo is a large blue button labeled "ENABLE". A red arrow points from the text "13. click \"ENABLE\"" to the "ENABLE" button. At the bottom of the page, there are tabs for "OVERVIEW" (which is selected), "PRICING", "SUPPORT", and "RELATED PRODUCTS".

Google Earth Engine API

Geospatial insights for a more sustainable world.

ENABLE

OVERVIEW PRICING SUPPORT RELATED PRODUCTS

Important note: You must register your project to use the Earth Engine API. Click on "Learn more" for additional information.

Google Earth Engine is a platform for global-scale analysis and visualization of geospatial datasets. Earth Engine can be used with 70+ petabytes of analysis-ready geospatial data in the Earth Engine Data Catalog, or with custom datasets.

Additional details

Type: SaaS & APIs  
Last product update: 01/08/2022  
Category: Big data, Analytics, Science & research, Climate  
Service name: earthengine.googleapis.com

## 2.2. Create a Google Cloud project &amp; enable GEE API

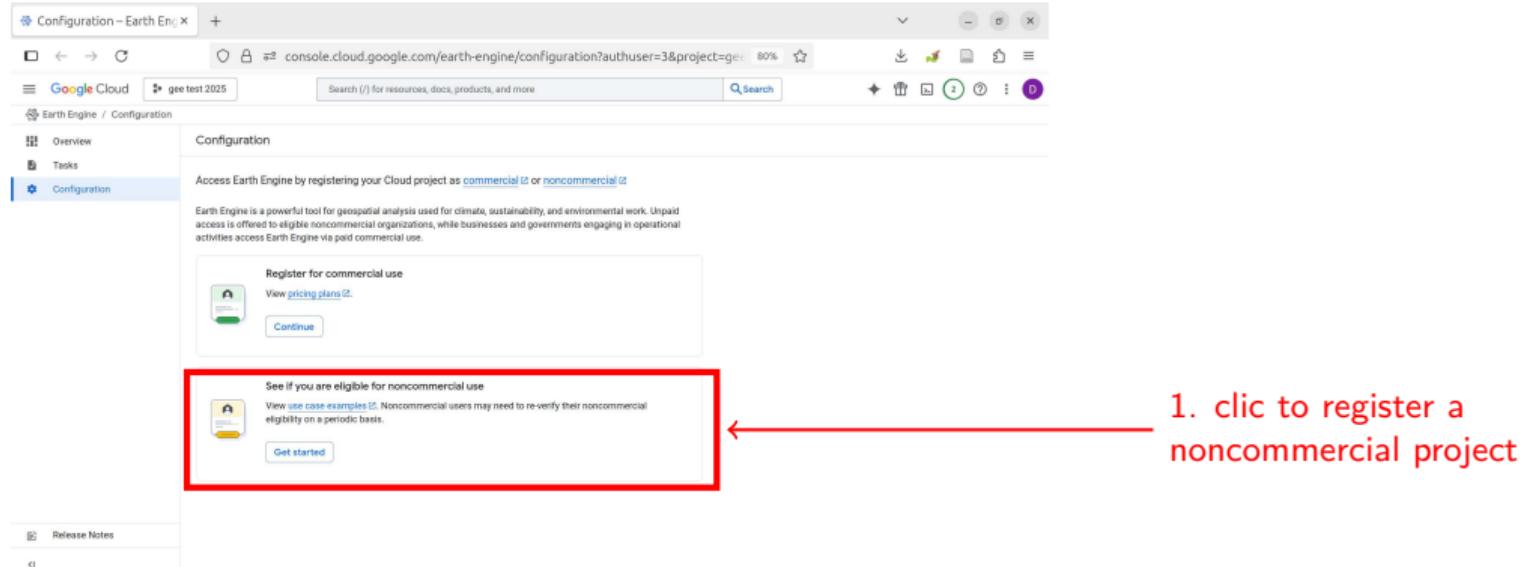
## 2. Create a Google Cloud project & enable GEE API

### 2.3 Enable GEE API in the newly created project

The screenshot shows the 'API/Service details' page for the GEE API in the Google Cloud console. The URL is <https://console.cloud.google.com/apis/api/earthengine.googleapis.com/metrics?project=gee-dip>. The left sidebar shows 'Enabled APIs and services'. The main area displays the 'Google Earth Engine API' with its service name as 'earthengine.googleapis.com' and type as 'Public API'. A red box highlights the 'Status' field which is set to 'Enabled'. To the right of the status, a red arrow points to the text 'GEE API is now enabled'. Below the status, there are tabs for METRICS, QUOTAS AND SYSTEM LIMITS, CREDENTIALS, and COST. There are also filters for 'Select graphs' (4 Graphs), time intervals (1 hour, 6 hours, 12 hours, 1 day, 2 days, 4 days, 7 days, 14 days, 30 days), and dropdowns for 'Filters', 'Versions' (v1, v1alpha and v1beta), 'Credentials' (Unspecified, Anonymous ...), and 'Methods' (125 options selected).

## 2.3. Register Google Cloud project for use with GEE

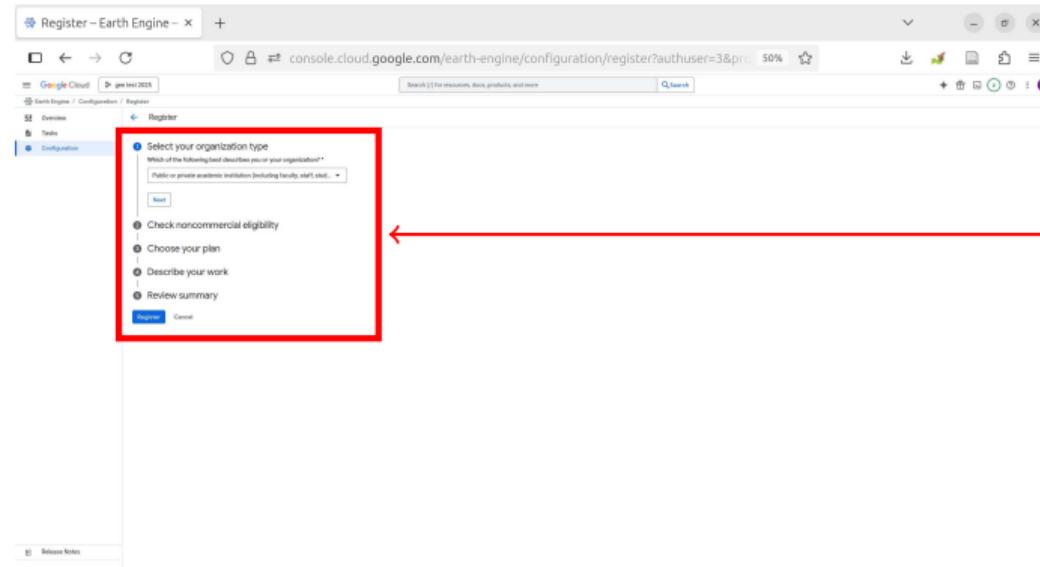
## 3. Register Google Cloud project project for use with GEE

3.1 Access register page at <https://code.earthengine.google.com/register>

## 2.3. Register Google Cloud project for use with GEE

## 3. Register Google Cloud project project for use with GEE

## 3.1 Register project



2. Fill in questionnaire  
steps 1-5

## 2.3. Register Google Cloud project for use with GEE

## 3. Register Google Cloud project for use with GEE

## 3.1 Register project

Step 1	Step 2	Step 3	Step 4	Step 5
<p><b>1 Select your organization type</b></p> <p>Which of the following best describes you or your organization?</p> <p><input type="checkbox"/> Public or private academic institution (including faculty, staff, students)</p> <p><b>Next</b></p>	<p><b>2 Select your organization type</b></p> <p><b>3 Check noncommercial eligibility</b></p> <p>What is the name of your academic institution? *</p> <p>Universidad Nacional Autónoma de México</p> <p>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.</p> <p><input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p>How would you describe your use of Earth Engine? *</p> <p><input checked="" type="radio"/> Scientific research e.g., advancing remote sensing methodologies <input type="radio"/> Decision making e.g., analysis that focuses on decision support</p> <p>What is your research question? *</p> <p>What is the geographic scope of your study? *</p> <p><input checked="" type="radio"/> Global <input type="radio"/> Regional</p> <p>Have you previously published work on this topic that used Earth Engine? *</p> <p><input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p><b>Check eligibility</b></p> <p><b>4 Choose your plan</b></p> <p><b>5 Describe your work</b></p> <p><b>6 Review summary</b></p> <p><b>Register</b> <b>Cancel</b></p>	<p><b>1 Select your organization type</b></p> <p><b>2 Check noncommercial eligibility</b></p> <p><b>3 Choose your plan</b></p> <p>A pricing plan is not required for noncommercial registration.</p> <p><b>Next</b></p> <p><b>4 Describe your work</b></p> <p><b>5 Review summary</b></p> <p><b>Register</b> <b>Cancel</b></p>	<p><b>1 Select your organization type</b></p> <p><b>2 Check noncommercial eligibility</b></p> <p><b>3 Choose your plan</b></p> <p><b>4 Describe your work</b></p> <p>Does your work with Earth Engine fall into any of these categories?</p> <p><input type="checkbox"/> Mitigation e.g., reduction or avoidance of greenhouse gas emissions / CO<sub>2</sub> equivalent <input type="checkbox"/> Adaptation e.g., helping people and communities adapt to the impacts of climate change <input type="checkbox"/> Protection &amp; conservation e.g., land and freshwater interventions to conserve biodiversity and ecosystems</p> <p>Will you use Earth Engine for any of the following? *</p> <p>Natural Disasters / Climate Risk</p> <p><b>Next</b></p> <p><b>5 Review summary</b></p> <p><b>Register</b> <b>Cancel</b></p>	<p><b>1 Review summary</b></p> <p>Create and edit the daily limit for usage (EEU-time) through the Earth Engine Configuration page.</p> <p><b>Organization type</b></p> <p>Which of the following best describes you or your organization?</p> <p>Public or private academic institution (including faculty, staff, students)</p> <p><b>Noncommercial eligibility</b></p> <p>What is the name of your academic institution?</p> <p>Universidad Nacional Autónoma de México</p> <p>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.</p> <p>No</p> <p>How would you describe your use of Earth Engine?</p> <p>Scientific research</p> <p>What is your research question?</p> <p>How can satellite imagery can help better monitor volcanoes</p> <p>What is the geographic scope of your study?</p> <p>Global</p> <p>Have you previously published work on this topic that used Earth Engine?</p> <p>No</p> <p><b>Your work</b></p> <p>Will you use Earth Engine for any of the following?</p> <p>Natural Disasters / Climate Risk</p> <p>This information is collected to verify noncommercial eligibility, inform product improvements, and assess the sustainability impact of Earth Engine usage, subject to the <a href="#">Google Cloud Privacy Notice</a>.</p> <p><b>Register</b> <b>Cancel</b></p>

## 2.3. Register Google Cloud project for use with GEE

## 3. Register Google Cloud project project for use with GEE

## 3.2 Register project

The screenshot shows a browser window with the URL `console.cloud.google.com/earth-engine/configuration;success=true?authuser=1`. The page title is "Configuration – Earth Engine". On the left, there's a sidebar with "Google Cloud" and "gee test 2025" selected. The main content area has a heading "Configuration" and a success message: "You are now registered for noncommercial use. Check out the Overview page to access the Earth Engine API, explore datasets, and start analyzing." Below this message is a "Continue" button. A red box highlights this message area. To the right of the red box, a red arrow points to the text "3. register successful". Further down the page, there are sections for "Control EECU-time" (with "Manage quota" button) and "Your Cloud project is registered for noncommercial use" (with "Manage registration" button). At the bottom left, there's a "Release Notes" section.

## 2.3. Register Google Cloud project for use with GEE

## 3. Register Google Cloud project project for use with GEE

## 3.2 Register project

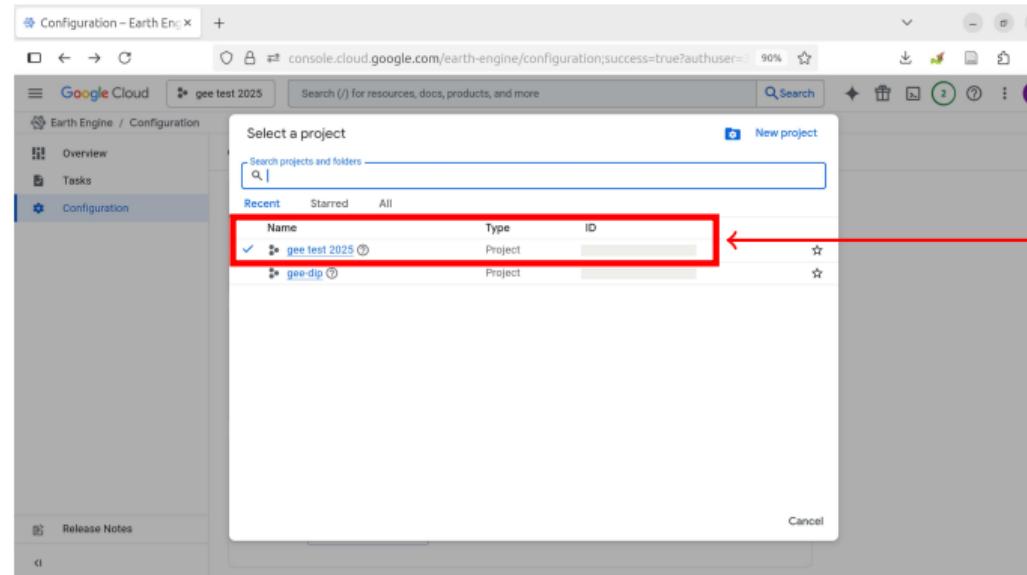
The screenshot shows the 'Configuration' page of the Earth Engine web interface. The URL in the browser bar is `console.cloud.google.com/earth-engine/configuration;success=true?authuser=1`. A red box highlights the project name 'gee test 2025' in the top navigation bar. The main content area displays a message: 'You are now registered for noncommercial use. Check out the Overview page to access the Earth Engine API, explore datasets, and start analyzing.' Below this is a section titled 'Control EECU-time' with a 'Manage quota' button. At the bottom, there's a section about registration status: '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](#).' A 'Manage registration' button is present here as well.

4. check project id

## 2.3. Register Google Cloud project for use with GEE

## 3. Register Google Cloud project project for use with GEE

## 3.1 Register project



Project info:  
- **project ID**  
- **project Name**

## 2.4. Access GEE in JavaScript IDE

4. (Run GEE in JavaScript IDE) → *in case you want to try*

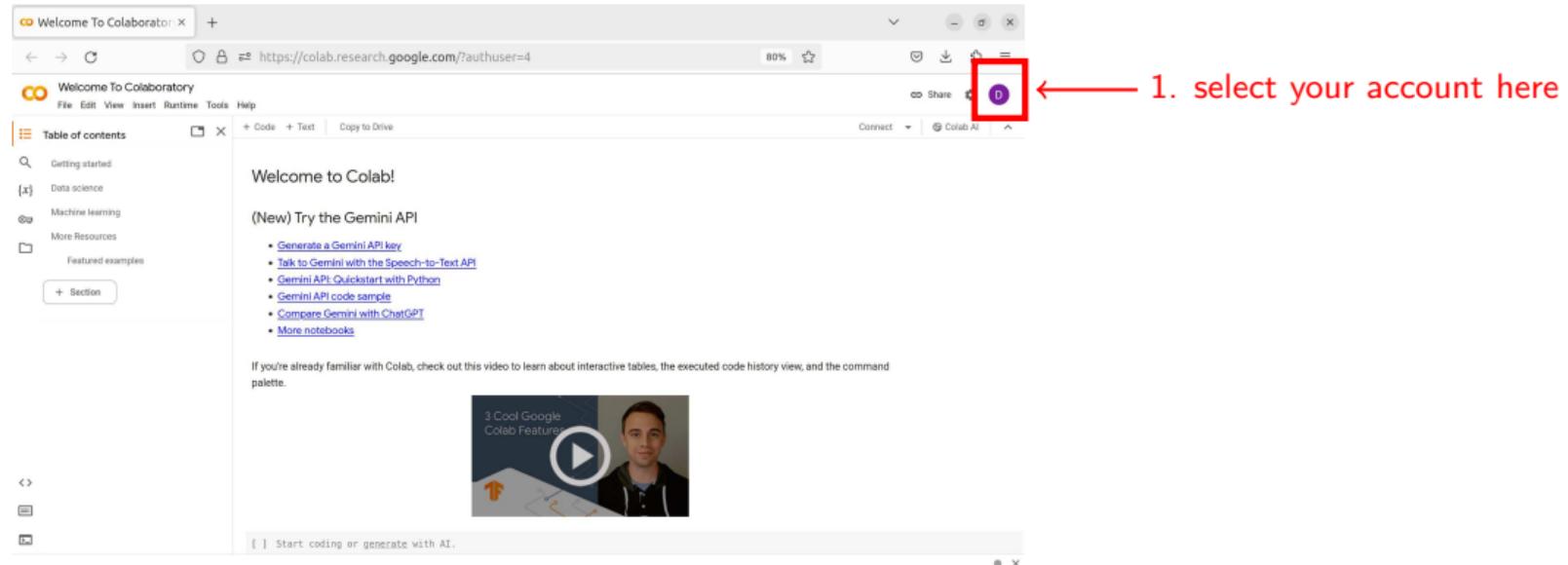
4.1 Access GEE JavaScript IDE at <https://code.earthengine.google.com/>

2. clic "Run" to execute code

1. type code in JavaScript  
⇒ ex: official tutorials

## 2.5. Access GEE in Colab

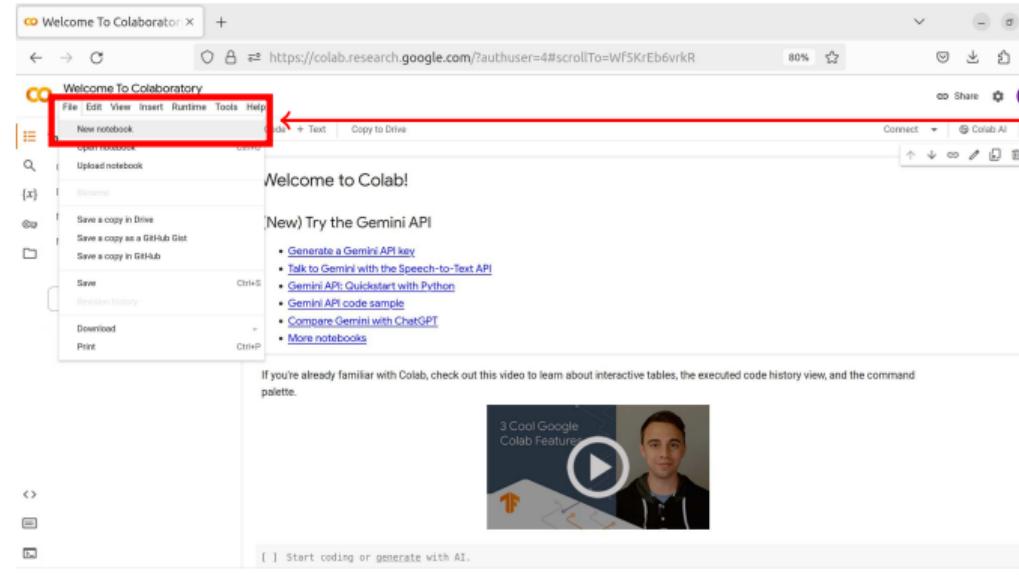
## 4. Run GEE in Colab

4.1 Access Google Colaboratory at <https://colab.research.google.com/>

## 2.5. Access GEE in Colab

## 4. Access GEE in Colab

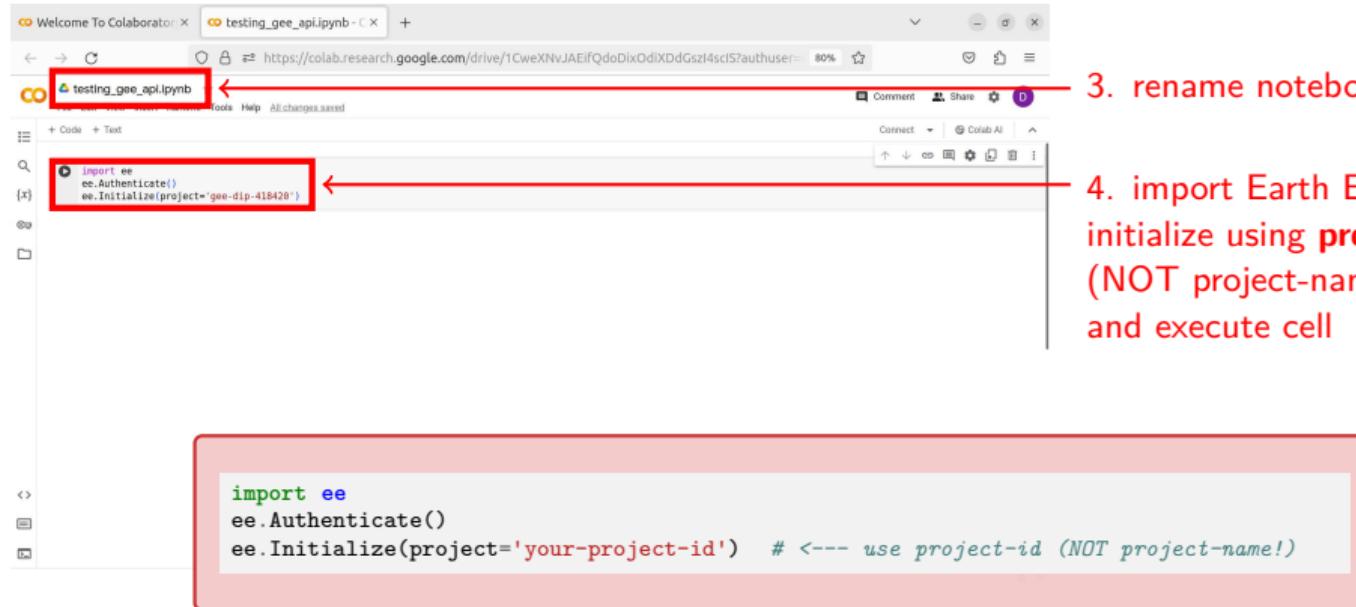
## 4.2 Create new notebook



2. File - New notebook

## 2.5. Access GEE in Colab

## 4. Access GEE in Colab

4.3 Import ee library & initialize with **project-id** (in which GEE API was enabled)

3. rename notebook (optional)

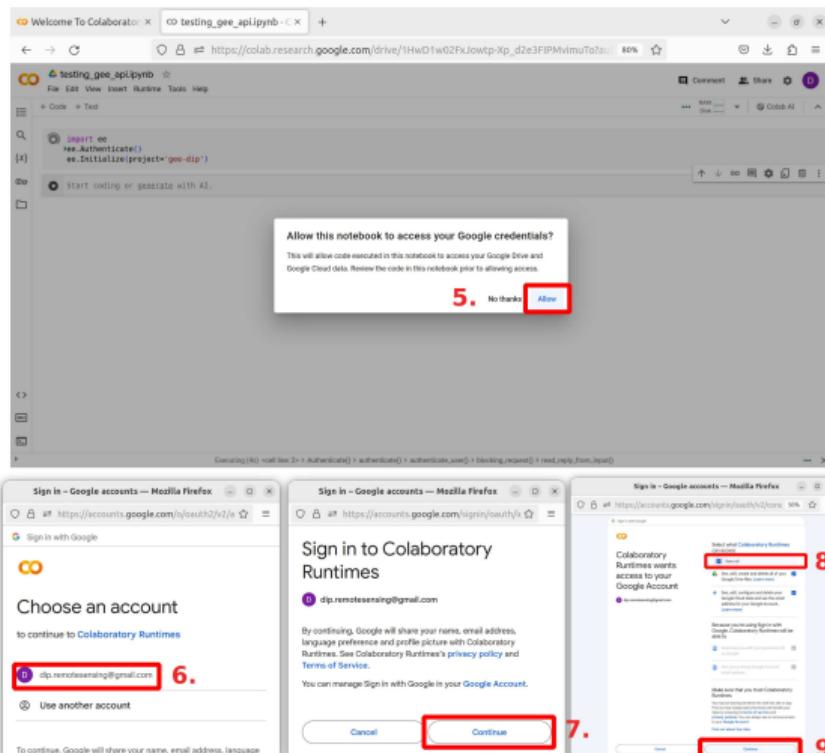
4. import Earth Engine lib (ee),  
initialize using **project-id**,  
(NOT project-name!),  
and execute cell

```
import ee
ee.Authenticate()
ee.Initialize(project='your-project-id') # <--- use project-id (NOT project-name!)
```

## 2.5. Access GEE in Colab

## 4. Access GEE in Colab

## 4.3 Execute cell &amp; give authorizations in pop-up windows



## 2.5. Access GEE in Colab

## 4. Access GEE in Colab

## 4.4 Start coding with GEE in Colab !

```
[1]: import ee  
ee.Authenticate()  
ee.Initialize(project='gee-dip-418428')  
  
[2]: # Initialize map  
import geemap  
Map = geemap.Map()  
  
# Select image and visualization parameters  
image = ee.Image('LANDSAT/LC08/C02/T1_TOA/LC08_044034_20140318');  
vizParams = {'bands': ['B4', 'B3', 'B2'], 'min': 0, 'max': 0.3, 'gamma': 1.3}  
  
# Center map on image and display  
Map.centerObject(image, 9)  
Map.addLayer(image, vizParams, 'Landsat 8 true color')  
Map
```

The screenshot shows a Google Colab interface with a notebook titled "testing\_gee\_api.ipynb". The code cell at index [2] contains Python code for initializing the Earth Engine API and creating a map centered on a Landsat 8 image. A red box highlights the code block, and a red arrow points from it to the text "10. start coding!".

10. start coding!  
(ex: Tutorial intro-to-python-api)

1. Introduction

2. Setup GEE in GoogleColab

### 3. GEE quick start

1. GEE data catalog
2. GEE data model
3. Jumpstart into image visualization

## 3.1. GEE data catalog

GEE's public [data archive](#) includes >40 years of **satellite imagery** expanded daily:

## 1. **Landsat** collections

- ⇒ [NASA/USGS Program](#), since 1972
- ⇒ 9 generations of satellites (polar-orbiting):
  - **Landsat-1** (1972) - **Landsat-3** (1978): optical & infrared imaging (VIS/NIR)
  - **Landsat-4** (1982) - **Landsat-9** (2021): optical & infrared imaging (VIS/NIR/SWIR/TIR)
- ⇒ GEE archive includes:

- Landsat 1-5	(1972–1999)	Sensor: <a href="#">MSS</a> (Multispectral Scanner)
- Landsat 4	(1982–1993)	Sensor: <a href="#">TM</a> (Thematic Mapper)
- Landsat 5	(1984–2012)	Sensor: <a href="#">TM</a> (Thematic Mapper)
- Landsat 7	(1999–2021)	Sensor: <a href="#">ETM+</a> (Enhanced Thematic Mapper Plus)
- Landsat 8	(2013–Present)	Sensor: <a href="#">OLI/TIRS</a> (Op. Land Imager / Therm. Infrared Sensor)
- Landsat 9	(2021–Present)	Sensor: <a href="#">OLI/TIRS</a> (Op. Land Imager / Therm. Infrared Sensor)

## 3.1. GEE data catalog

GEE's public [data archive](#) includes >40 years of **satellite imagery** expanded daily:

## 2. **Sentinel** collections

⇒ [ESA/Copernicus Program](#), since 2014

⇒ constellation of satellites consisting comprising various sensors:

- **Sentinel-1**: radar imaging (C-band SAR)
- **Sentinel-2**: optical & infrared imaging (VIS/SWIR)
- **Sentinel-3**: optical & infrared imaging (VIS/SWIR/TIR)
- **Sentinel-5P**: ultra-violet, optical, infrared imaging (UV/VIS/NIR/SWIR)

⇒ GEE archive includes:

- Sentinel 1	(2014–Present)	Sensor: SAR (C-band), <a href="#">GRD</a> scenes (Ground Range Detected)
- Sentinel 2	(2015–Present)	Sensor: <a href="#">MSI</a> (Multispectral Instrument)
- Sentinel 3	(2016–Present)	Sensor: <a href="#">OLCI</a> (Ocean and Land Color Instrument)
- Sentinel 5P	(2018–Present)	Sensor: <a href="#">TROPOMI</a> (TROPOspheric Monitoring Instrument)

## 3.1. GEE data catalog

GEE's public [data archive](#) includes >40 years of **satellite imagery** expanded daily:

### 3. MODIS collections

- ⇒ NASA's "Moderate Resolution Imaging Spectroradiometer"
- ⇒ sensor on board 2 satellites: Terra (since 1999) & Acqua (since 2002)
- ⇒ GEE archive includes: daily surface spectral reflectances from MODIS, as well as several derived products (e.g., vegetation indices, snow cover, etc)

### 4. High-Resolution Imagery

- ⇒ GEE archive currently includes: [Planet SkySat](#) Multispectral imagery, and aerial imagery acquired by the NAIP (*National Agriculture Imagery Program*) during the agricultural growing seasons in the continental U.S.

## 3.1. GEE data catalog

GEE's public [data archive](#) includes >40 years of **satellite imagery** expanded daily:

### 3. MODIS collections

- ⇒ NASA's "Moderate Resolution Imaging Spectroradiometer"
- ⇒ sensor on board 2 satellites: Terra (since 1999) & Acqua (since 2002)
- ⇒ GEE archive includes: daily surface spectral reflectances from MODIS, as well as several derived products (e.g., vegetation indices, snow cover, etc)

### 4. High-Resolution Imagery

- ⇒ GEE archive currently includes: [Planet SkySat](#) Multispectral imagery, and aerial imagery acquired by the NAIP (*National Agriculture Imagery Program*) during the agricultural growing seasons in the continental U.S.

## 3.1. GEE data catalog

In addition to satellite imagery, GEE also includes **other scientific datasets**:

1. Digital Elevation Models (DEMs) collections

- ⇒ DEMs describe Earth's topography
- ⇒ GEE archive includes:

- global DEMs: [SRTM DEM](#) (NASA's Shuttle Radar Topography Mission) data at 30-meter resolution, [Copernicus DEM](#) (ESA) data at 30-meter resolution, ALOS
- regional DEMs at higher resolutions

2. Thematic datasets:

- [Surface Temperature](#): includes land and sea surface temperature products derived from several spacecraft sensors, including MODIS, ASTER, and AVHRR, in addition to raw Landsat thermal data
- [Climate](#): includes climate models generate both long-term climate predictions and historical interpolations of surface variables
- [Atmospheric](#): includes ozone data from NASA's TOMS and OMI instruments and the MODIS Monthly Gridded Atmospheric Product

## 3.1. GEE data catalog

In addition to satellite imagery, GEE also includes **other scientific datasets**:

1. Digital Elevation Models (DEMs) collections

- ⇒ DEMs describe Earth's topography
- ⇒ GEE archive includes:

- global DEMs: [SRTM DEM](#) (NASA's Shuttle Radar Topography Mission) data at 30-meter resolution, [Copernicus DEM](#) (ESA) data at 30-meter resolution, ALOS
- regional DEMs at higher resolutions

2. Thematic datasets:

- [Surface Temperature](#): includes land and sea surface temperature products derived from several spacecraft sensors, including MODIS, ASTER, and AVHRR, in addition to raw Landsat thermal data
- [Climate](#): includes climate models generate both long-term climate predictions and historical interpolations of surface variables
- [Atmospheric](#): includes ozone data from NASA's TOMS and OMI instruments and the MODIS Monthly Gridded Atmospheric Product

## 3.1. GEE data catalog

In addition to satellite imagery, GEE also includes **other scientific datasets**:

## 2. Thematic datasets (continued):

- [Weather](#): includes forecasted and measured conditions over short periods of time, including precipitation, temperature, humidity, and wind, and other variables. Includes in particular NOAA's Global Forecast System (GFS) and the NCEP Climate Forecast System (CFSv2)
- [Land Cover](#): includes the physical landscape in terms of land cover classes such as forest, grassland, and water
- [Cropland](#): includes a number of cropland data products
- [Other Geophysical Data](#): includes data from other satellite image sensors

The GEE data model revolves around the following components:

- **Image objects**

⇒ `ee.Image`

- ⇒ Image objects represent raster data (i.e., satellite imagery, climate data, or any gridded data)
- ⇒ Image objects consist of one or more bands, where each band represents a different type of information (e.g., red, green, blue bands for RGB imagery)

- **Geometry objects**

⇒ `ee.Geometry`

- ⇒ Geometry objects represent vector data (i.e., points, lines, or polygons)
- ⇒ Geometry objects support different geometries: `Point` (a list of coordinates in some projection), `LineString` (a list of points), `LinearRing` (a closed `LineString`), `Polygon` (a list of `LinearRings` where the first is a shell and subsequent rings are holes), as well as `MultiPoint`, `MultiLineString`, and `MultiPolygon`

- **Feature objects**

- ⇒ `ee.Feature`
- ⇒ Feature objects are Geometry objects with attributes
- ⇒ Feature objects store a Geometry object (or null) and a properties property storing a dictionary of other properties

- **Collection objects**

- ⇒ Collections are groups of Image or Feature objects
- ⇒ `ee.ImageCollection`: group of Image objects, which can be organized and filtered based on various criteria such as date, metadata, or spatial location
- ⇒ `ee.FeatureCollection`: group of Feature objects

## 3.3. Jumpstart into image visualization

```
# Initialize
import geemap
import ee
ee.Authenticate()
ee.Initialize(project='your-project-id') # Initialize using project-id with enabled GEE API
Map = geemap.Map() # Initialize map

# Select image and visualization parameters
image = ee.Image('LANDSAT/LC08/C02/T1_TOA/LC08_026047_20200116'); # Landsat 8 Top of Atmosphere (TOA) image over Popocatépetl
vis_param = {'bands': ['B4', 'B3', 'B2'], 'min': 0, 'max': 0.3, 'gamma': 1.3} # Select bands for true color RGB

# Center map on image and display
Map.centerObject(image, 9)
Map.addLayer(image, vizParams, 'Landsat 8 true color')
Map
```

