

Lecture 05

Google Earth Engine (GEE)

2024-04-01

Sébastien Valade



UNIVERSIDAD NACIONAL
AUTÓNOMA DE
MÉXICO

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](#)

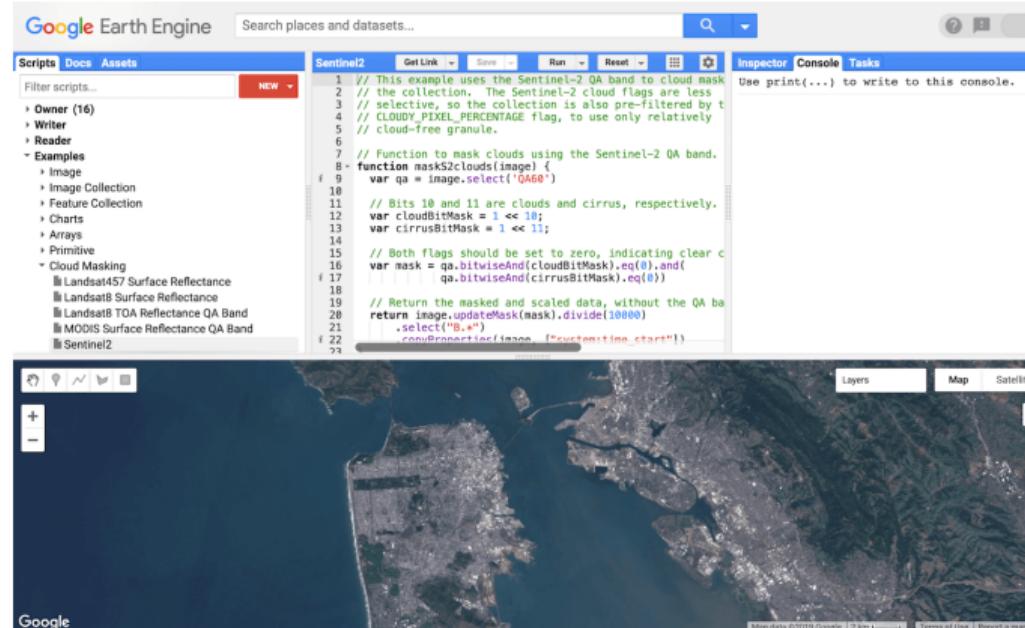
1.1. GEE overview

- **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](#)

1.2. 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 buttons for 'Get Link', 'Save', 'Run', 'Reset', and a dropdown menu. To the right of these are 'Inspector', 'Console', and 'Tasks' tabs, with a note to use `print(...)` to write to the console.

The main area contains a code editor with the following JavaScript code:

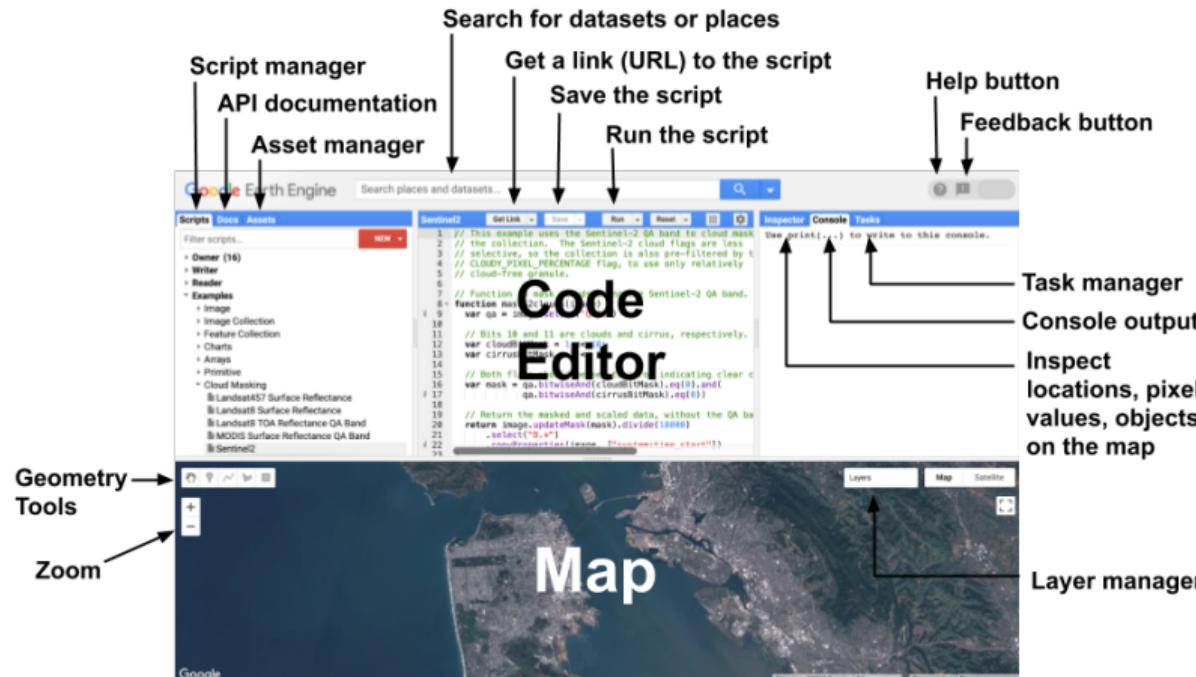
```
1 // This example uses the Sentinel-2 QA band to cloud mask
2 // the collection. The Sentinel-2 cloud flags are less
3 // selective, so the collection is also pre-filtered by t
4 // CLOUDY_PIXEL_PERCENTAGE flag, to use only relatively
5 // cloud-free granule.
6
7 // Function to mask clouds using the Sentinel-2 QA band.
8 function maskS2clouds(image) {
9   var qa = image.select('QA6B');
10
11   // Bits 10 and 11 are clouds and cirrus, respectively.
12   var cloudBitMask = 1 << 10;
13   var cirrusBitMask = 1 << 11;
14
15   // Both flags should be set to zero, indicating clear c
16   var mask = qa.bitwiseAnd(cloudBitMask).eq(0).and(
17     qa.bitwiseAnd(cirrusBitMask).eq(0));
18
19   // Return the masked and scaled data, without the QA ba
20   return image.updateMask(mask).divide(10000)
21     .select(['B.*'])
22     .copyProperties(image, ['customtime_start']);
23 }
```

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 scale bar and a copyright notice: 'Map data ©2019 Google | 2 km'.

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 a web browser window for 'Welcome To Colaboratory' at <https://colab.research.google.com>. The browser interface includes a title bar, address bar, and various toolbar icons. On the left, there's a sidebar with a 'Table of contents' section containing links to 'Getting started', 'Data science', 'Machine learning', 'More Resources', and 'Featured examples'. Below this is a 'Section' button. The main content area displays a 'What is Colab?' page. It explains that Colab allows writing and executing Python in a browser with zero configuration required, access to GPUs for free, and easy sharing. It also mentions that Colab is suitable for students, data scientists, and AI researchers. A 'Getting started' section follows, explaining that the document is an interactive Colab notebook. It shows a code cell with the following Python script:

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

The page also includes instructions for executing code and information about variable scope.

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 'Welcome To Colaboratory' page at https://colab.research.google.com. On the left, there's a sidebar with a 'Table of contents' section containing links to 'Getting started', 'Data science', 'Machine learning', 'More Resources', and 'Featured examples'. Below this is a '+ Section' button. The main content area has a heading 'What is Colab?'. It explains that Colab, or 'Colaboratory', allows you to write and execute Python in your browser, with 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. A 'Getting started' section is expanded, explaining that the document is an interactive environment called a 'Colab notebook' where you can write and execute code. An example code cell is shown:

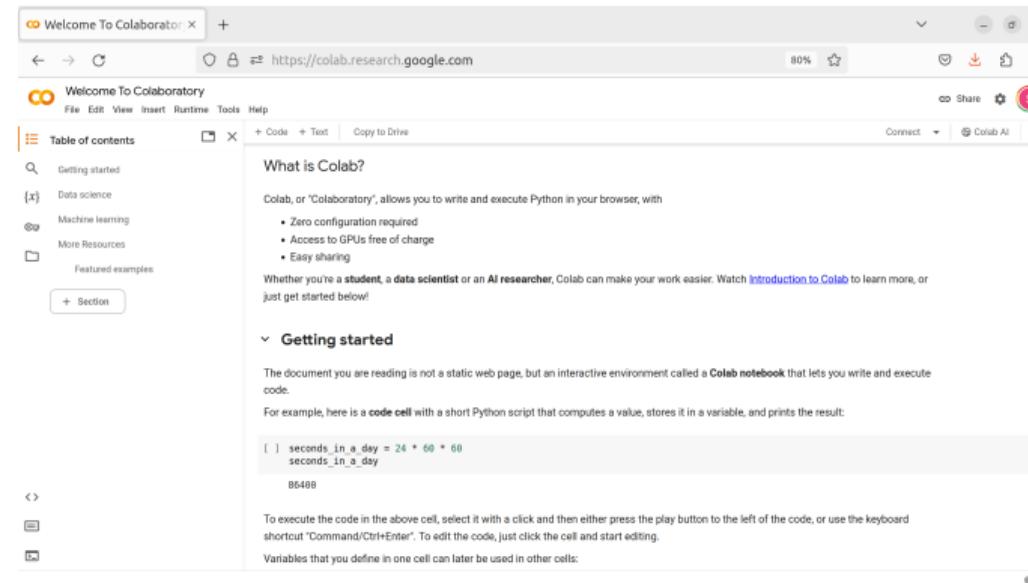
```
[ ] seconds_in_a_day = 24 * 60 * 60
seconds_in_a_day
86400
```

Instructions say to execute the code by selecting it and pressing the play button or using the keyboard shortcut 'Command/Ctrl+Enter'. It also notes that variables defined in one cell can be used in other cells.

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 Colab

3. GEE quick start

Nota Bene

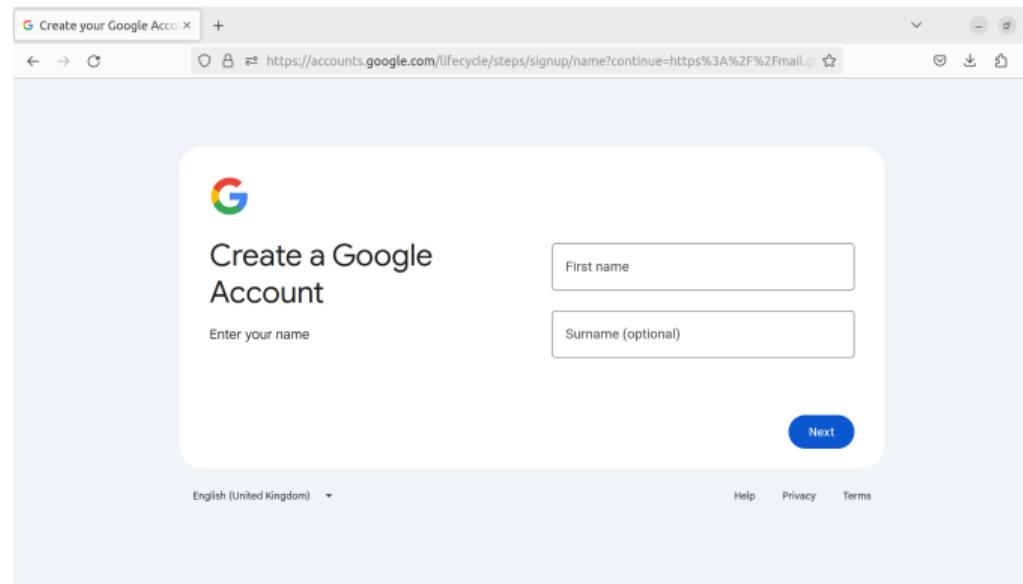
The steps required by Google to access and use GEE APIs are regularly evolving.

⇒ the steps described are those required as of March-2024

⇒ 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 & enable GEE API

2. Create a Google Cloud project & enable GEE API

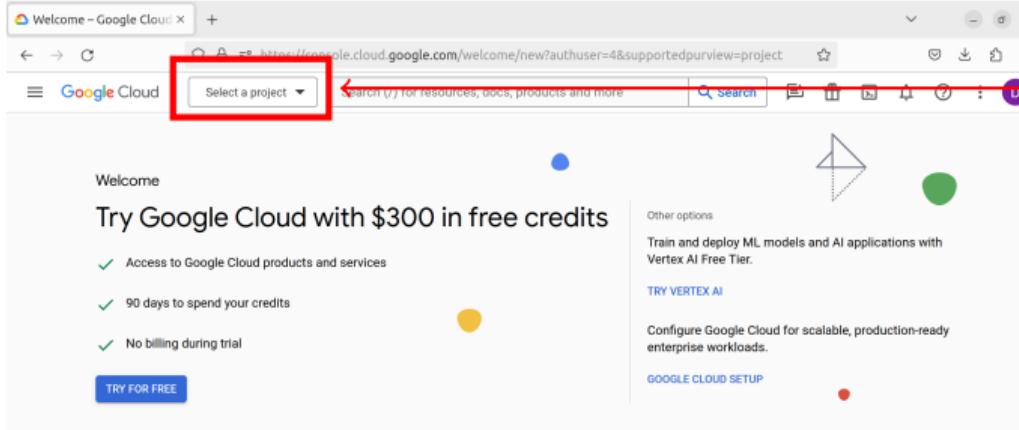
2.1 Access your account's Google Cloud Console

The screenshot shows the Google Cloud Welcome page. At the top right, there is a purple circular profile icon with a white letter 'D' inside. A red box highlights this icon, and a red arrow points to it from the text '1. select your account here'. The page features a 'Welcome' section with a banner offering '\$300 in free credits'. Below this, there are three main sections: 'Other options' (with a 'TRY VERTEX AI' button), 'Configure Google Cloud for scalable, production-ready enterprise workloads.' (with a 'GOOGLE CLOUD SETUP' button), and 'Popular getting started resources'. At the bottom, there are filter buttons for 'General' (which is selected) and other categories like 'Web, mobile, game, storage', 'Containers, VMs, hybrid/multi, move workload', 'Data, AI/ML, SAP', and 'Maps, APIs'. There is also a 'Pre-built solution templates' section.

1. select your account here

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

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

The screenshot shows the Google Cloud Welcome page. At the top left, there's a 'Select a project' dropdown menu with a red box around it. A red arrow points from the text '2. select a project' to this dropdown. The page features a 'Welcome' section with a 'Try Google Cloud with \$300 in free credits' offer. It lists three benefits: 'Access to Google Cloud products and services', '90 days to spend your credits', and 'No billing during trial'. Below these is a 'TRY FOR FREE' button. To the right, there's an 'Other options' section for 'Vertex AI Free Tier' with a 'TRY VERTEX AI' button, and another section for 'GOOGLE CLOUD SETUP'. At the bottom, there's a 'Popular getting started resources' section with a 'Filter by' dropdown and several categories like 'Web, mobile, game, storage', 'Containers, VMs, hybrid/multi, move workload', 'Data, AI/ML, SAP', 'Maps, APIs', and 'General'. There's also a 'Pre-built solution templates' section.

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

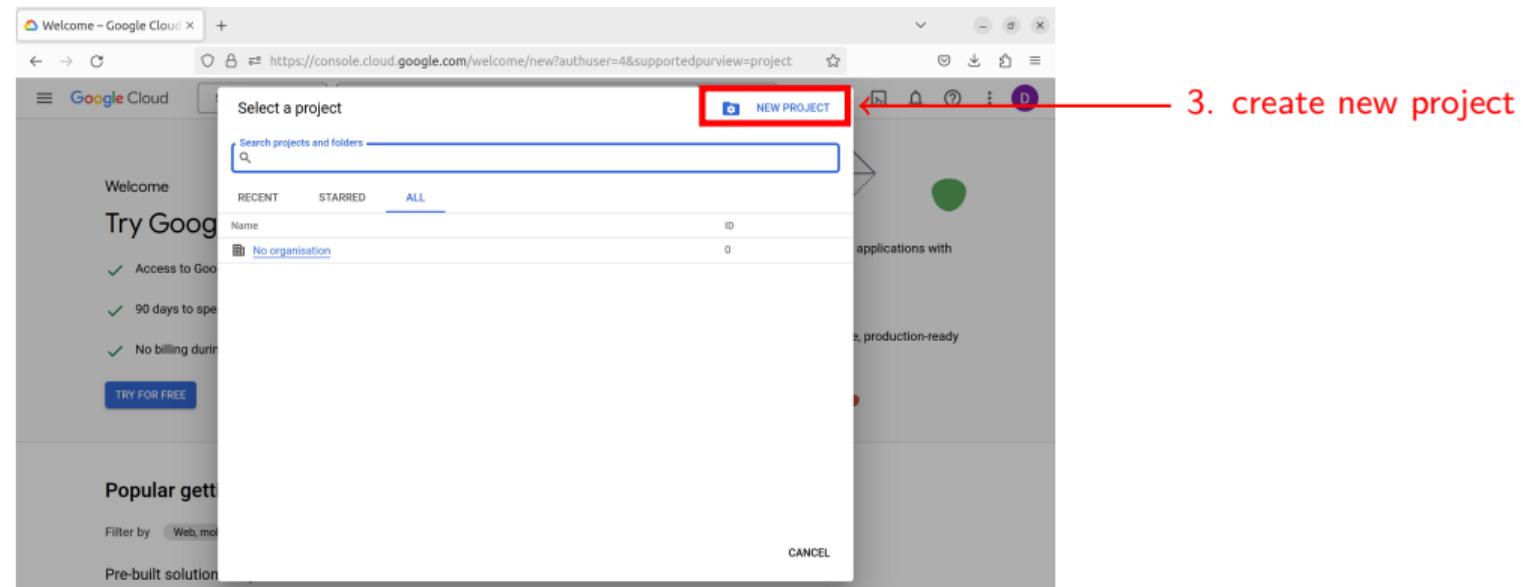
Filter by

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

Pre-built solution templates

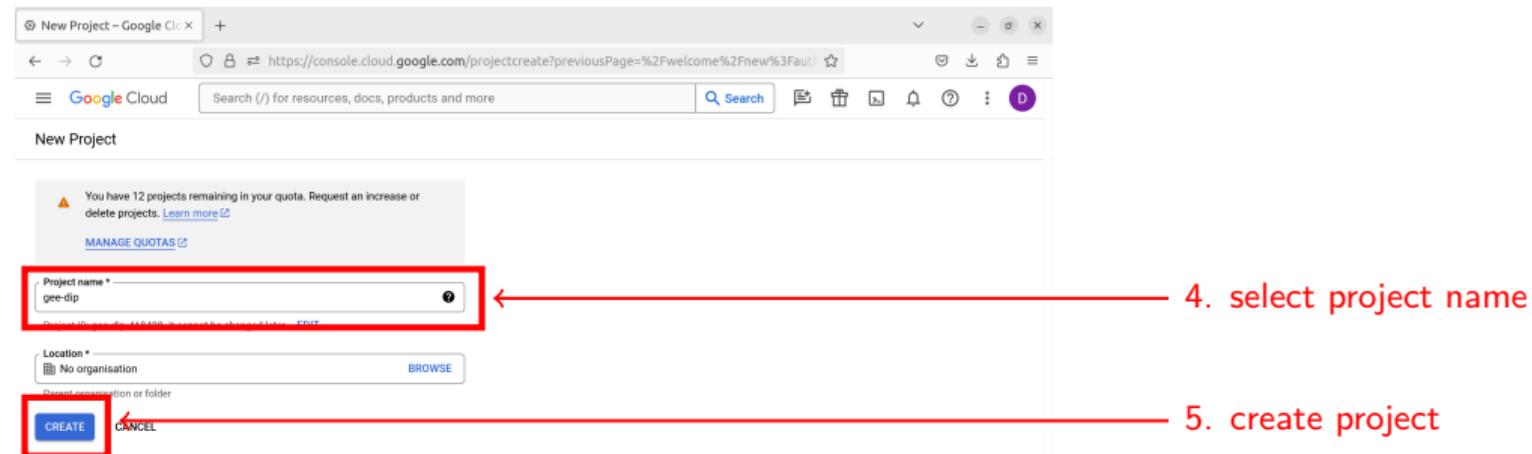
2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

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

2.2. Create a Google Cloud project & enable GEE API

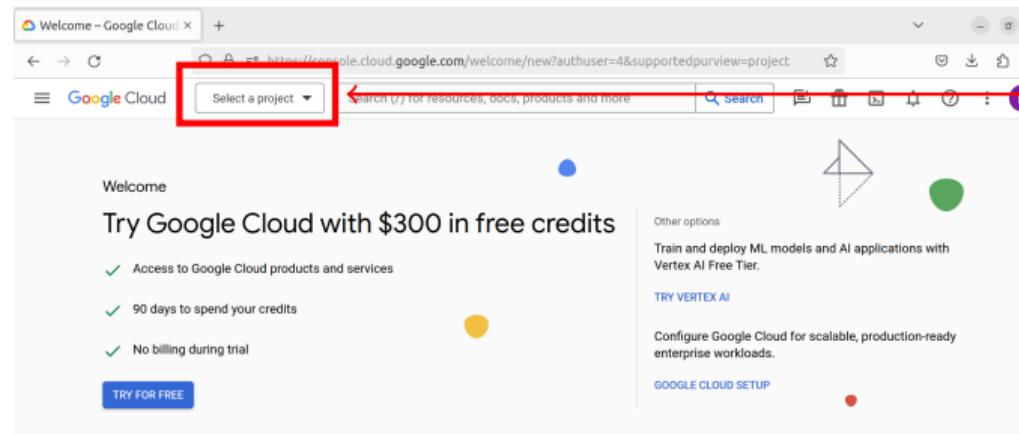
2. Create a Google Cloud project & enable GEE API

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

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

2.3 Enable GEE API in the newly created project



6. select the project

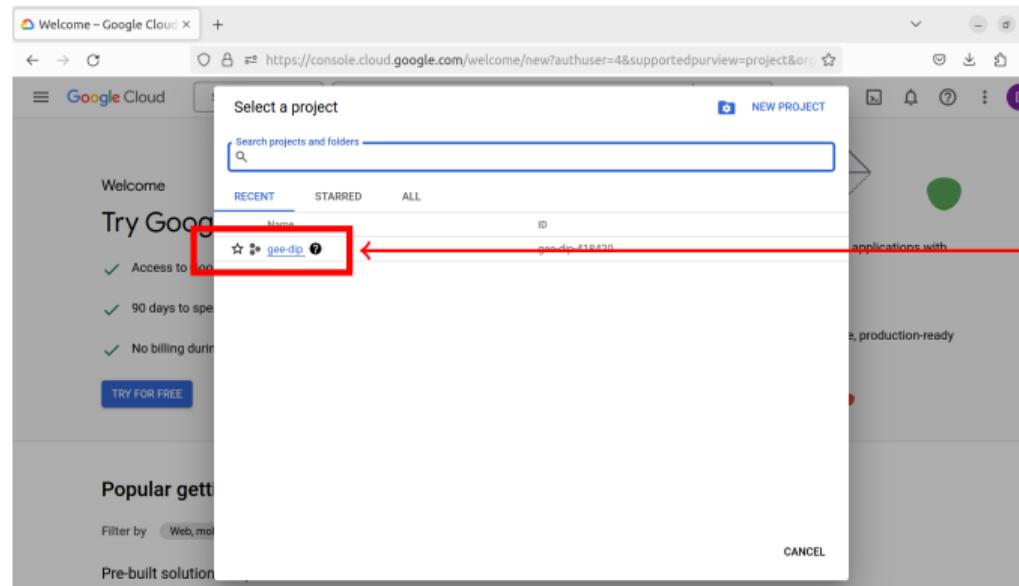
Popular getting started resources

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 & enable GEE API

2. Create a Google Cloud project & enable GEE API

2.3 Enable GEE API in the newly created project

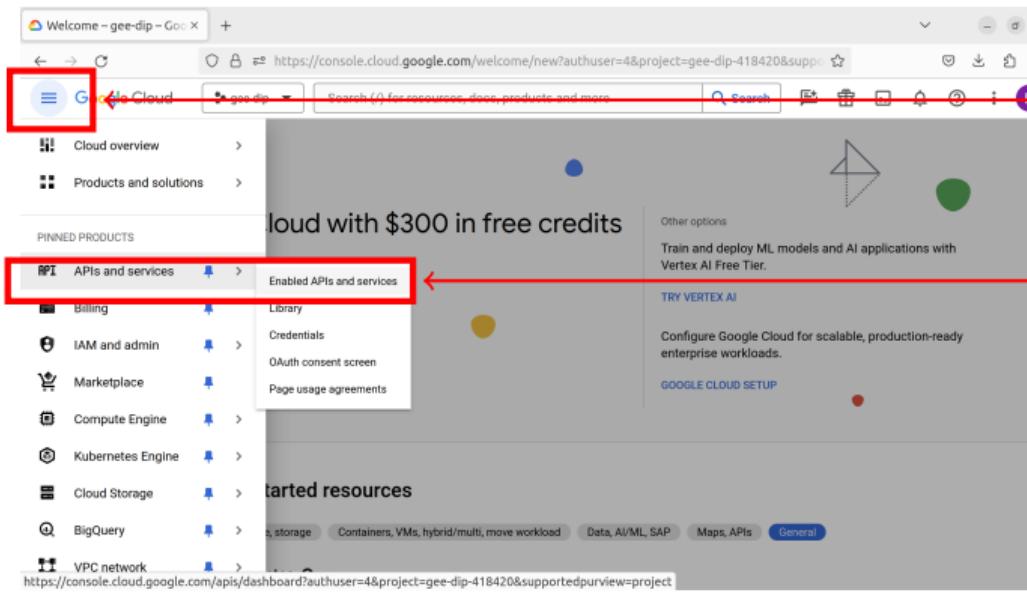


7. select the project

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

2.3 Enable GEE API in the newly created project



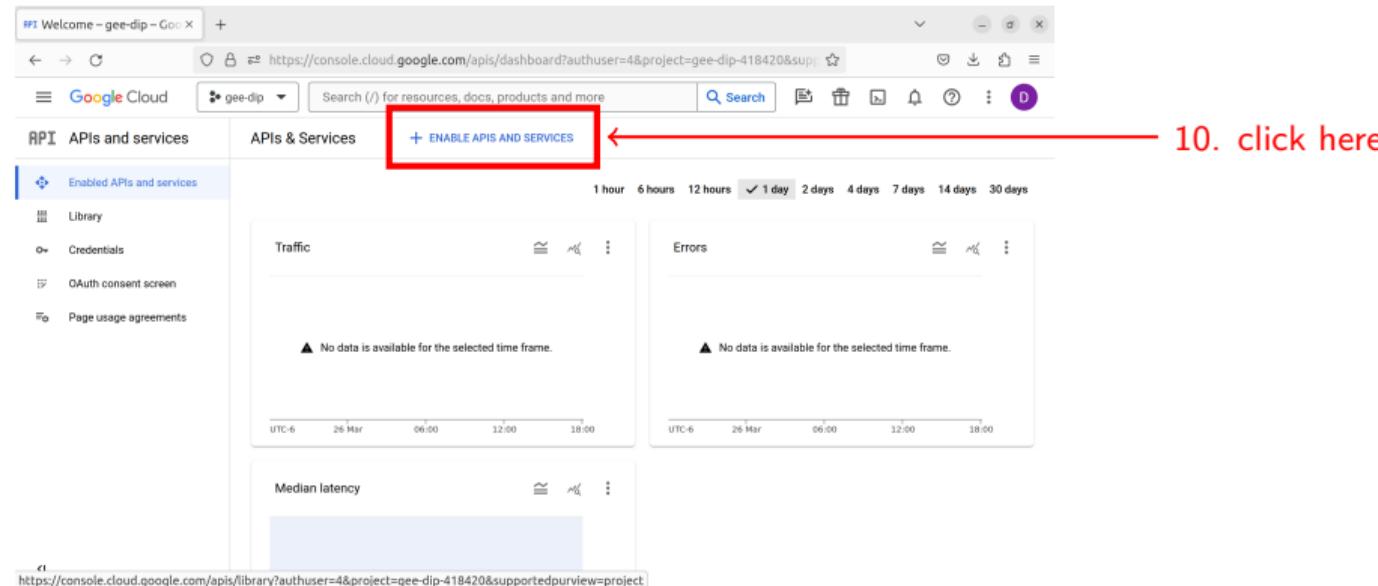
8. select navigation menu

9. "Enable API and services"

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

2.3 Enable GEE API in the newly created project

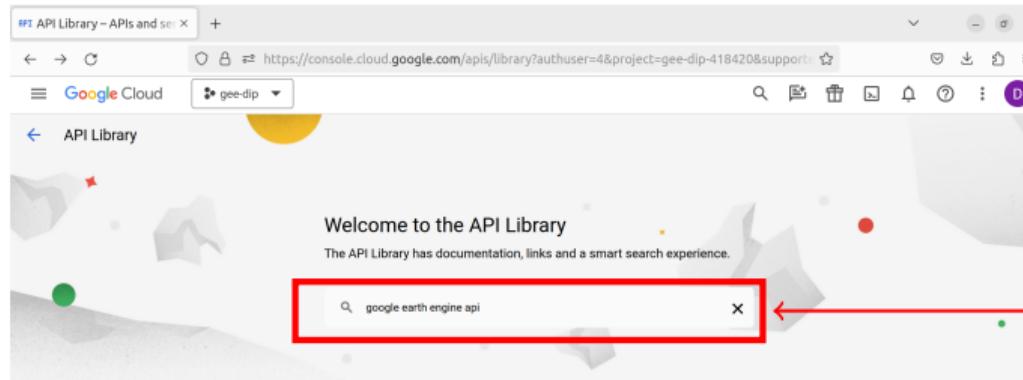


10. click here

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

2.3 Enable GEE API in the newly created project



Welcome to the API Library
The API Library has documentation, links and a smart search experience.

11. search/select “Google Earth Engine API”

Filter Type to filter

Maps

VIEW ALL (23)

Visibility

Public (452)

Private (2)

Category

Maps

Maps SDK for Android
Google
Maps for your native Android app.

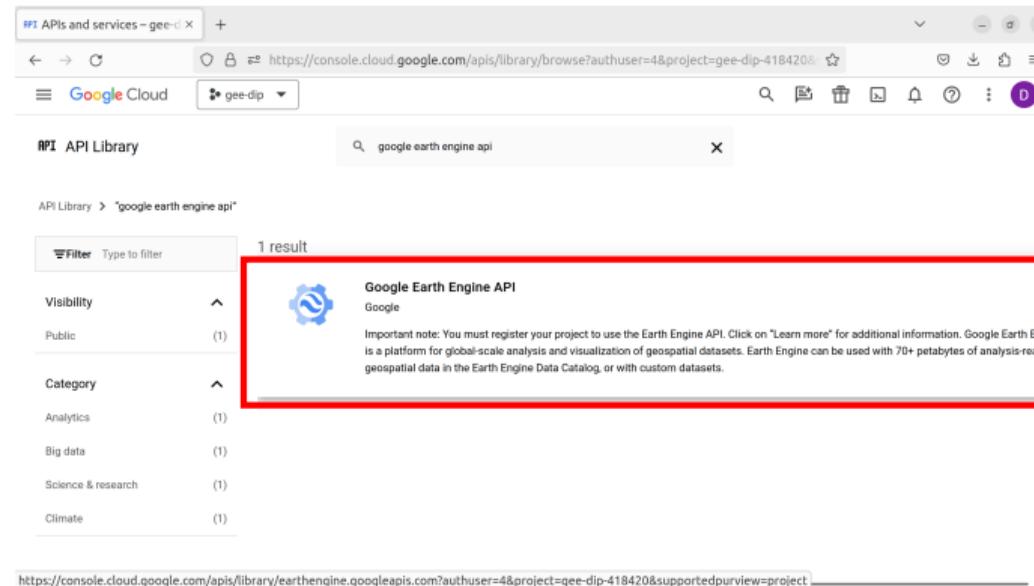
Maps SDK for iOS
Google
Maps for your native iOS app.

Maps JavaScript API
Google
Maps for your website

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & 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 page displays the "API Library" for the project "gee-dip". A search bar at the top right contains the query "google earth engine api". The results section shows "1 result" for "Google Earth Engine API" under the "Google" category. A red box highlights this result. A red arrow points from the text "12. click here" to the highlighted result. The left sidebar shows filters for "Visibility" (Public) and "Category" (Analytics, Big data, Science & research, Climate).

12. click here

2.2. Create a Google Cloud project & enable GEE API

2. Create a Google Cloud project & enable GEE API

2.3 Configure the newly created project

The screenshot shows a web 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". Below the title, there is a logo for "Google Earth Engine API" and a subtext "Geospatial insights for a more sustainable world.". A red box highlights the "ENABLE" button, which is located in a blue rectangular button. 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

Overview

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

13. click "ENABLE"

2.2. Create a Google Cloud project & enable GEE API

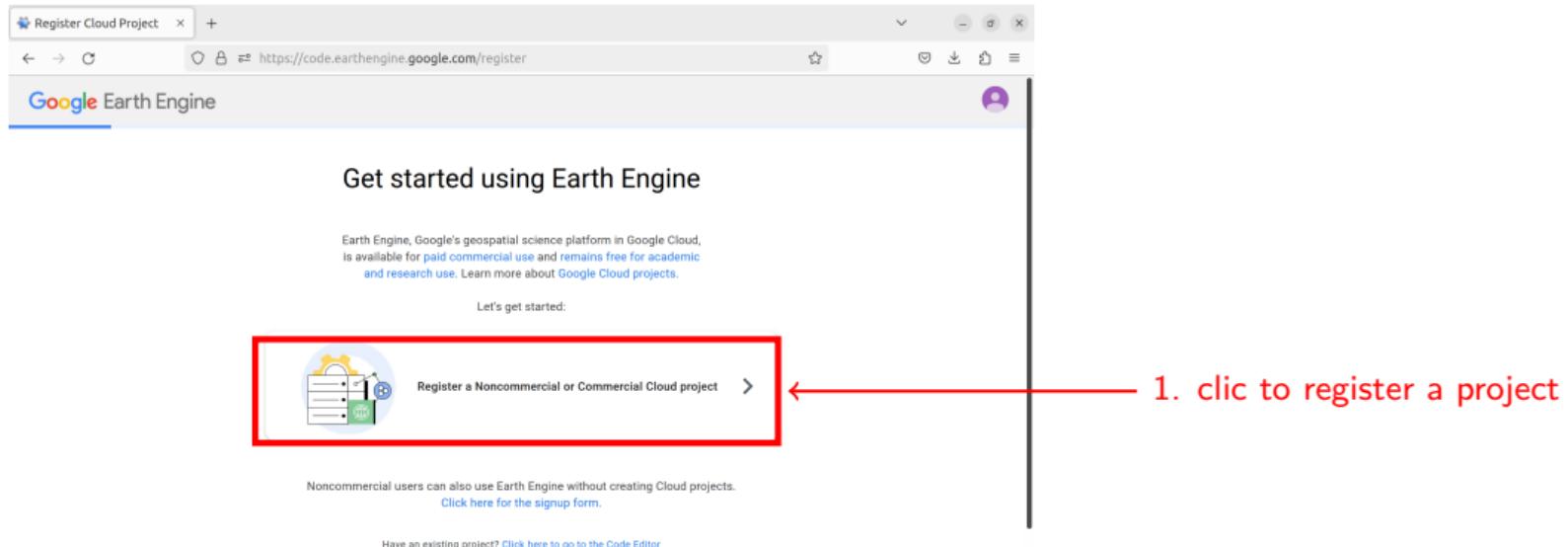
2. Create a Google Cloud project & enable GEE API

2.3 Enable GEE API in the newly created project

The screenshot shows the 'API/Service details' page for the Google Earth Engine API in the Google Cloud Platform. 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' with options like Library, Credentials, OAuth consent screen, and Page usage agreements. The main content area shows the 'Google Earth Engine API' with a status of 'Enabled'. A red arrow points to the 'Status' section, which is highlighted with a red box. The status is listed as 'Enabled'. Below the status, there is a 'CREATE CREDENTIALS' button and a 'VIEW DOCUMENTATION' link. At the bottom, there are tabs for METRICS, QUOTAS AND SYSTEM LIMITS, CREDENTIALS, and COST. The METRICS tab is selected, showing a dropdown for 'Select graphs' set to '4 Graphs', and a time range from '1 hour' to '30 days' with '30 days' checked. There are also filters for 'Versions' (v1, v1 alpha and v1 beta), 'Credentials' (Unspecified, Anonymous...), and 'Methods' (125 options selected).

2.3. Register Google Cloud project for use with GEE

3. Register Google Cloud 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

How do you want to use Earth Engine?

Paid usage
Commercial businesses, government operations. See examples

Unpaid usage
Non-profits, education, government research, training, media. See examples

Project type*
Academia & Research

Please note: If you will be accessing Earth Engine as a customer of a Google Cloud Platform reseller, please contact your reseller for terms and pricing governing your use of Earth Engine.

BACK NEXT

2. select:
- Unpaid usage
 - Academia & Research

2.3. Register Google Cloud project for use with GEE

3. Register Google Cloud project for use with GEE

3.1 Register project

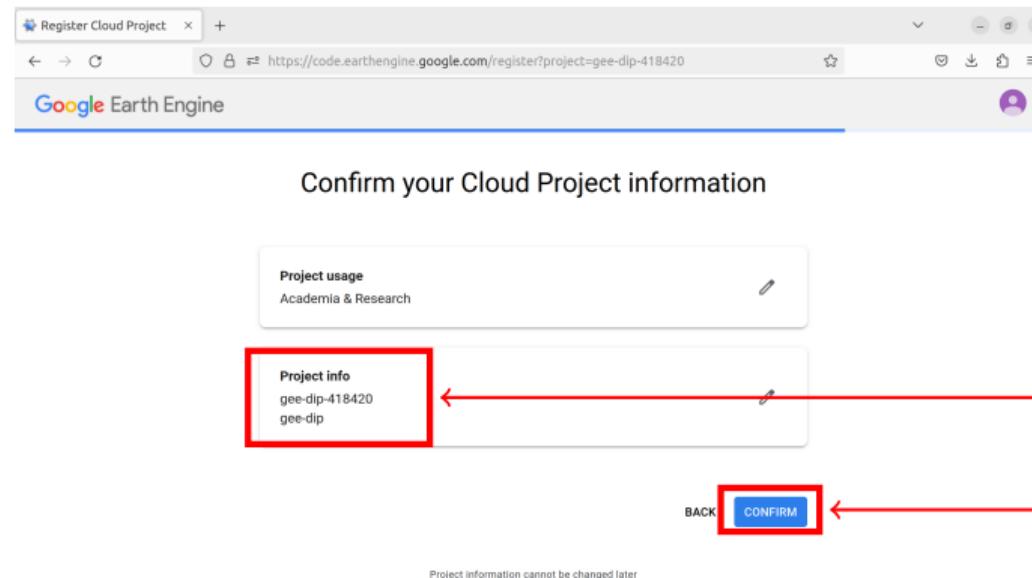
The screenshot shows a web browser window titled "Register Cloud Project". The URL in the address bar is <https://code.earthengine.google.com/register>. The page header says "Google Earth Engine". The main content area has a heading "Create or choose a Cloud Project to register". Below it, a note says "Create a new project in Google Cloud, or choose one you are authorized to access to enable the API:". There are two radio button options: "Create a new Google Cloud Project" (unchecked) and "Choose an existing Google Cloud Project" (checked). A red box highlights the "Choose an existing Google Cloud Project" section. Inside this box, there is a "Project" input field with "Type to filter" placeholder text, a "Refresh" button, and a list of "All Cloud Projects". One project, "gee-dip", is listed with its ID "gee-dip-418420". A red arrow points from the right side of the image towards the "gee-dip" entry in the list.

3. select project

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**

4. Confirm

2.3. Register Google Cloud project for use with GEE

3. Register Google Cloud project project for use with GEE

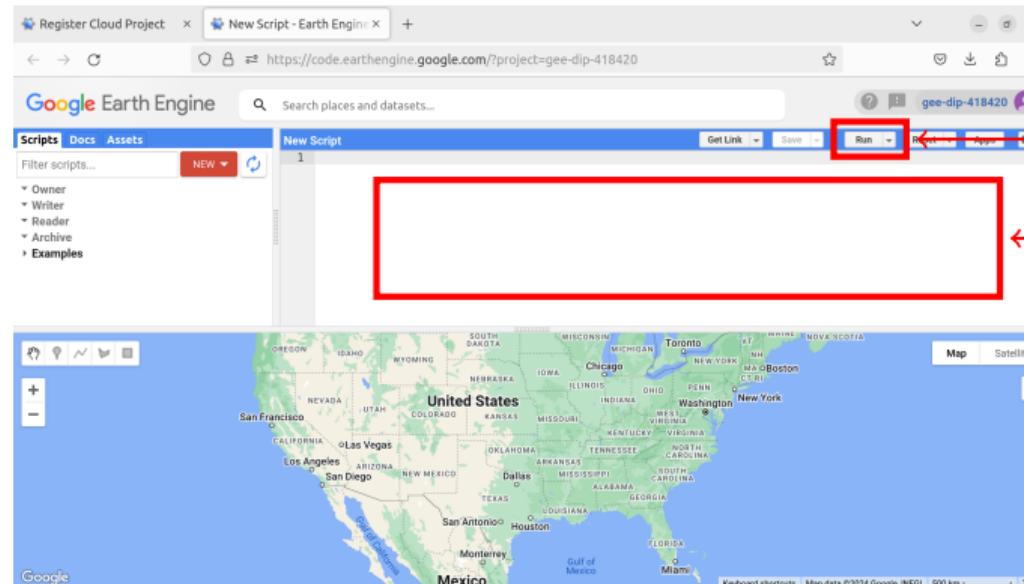
3.2 Register project



5. register successful
⇒ redirecting to Code Explorer
⇒ JavaScript IDE

2.3. Register Google Cloud project for use with GEE

3. Register Google Cloud project for use with GEE

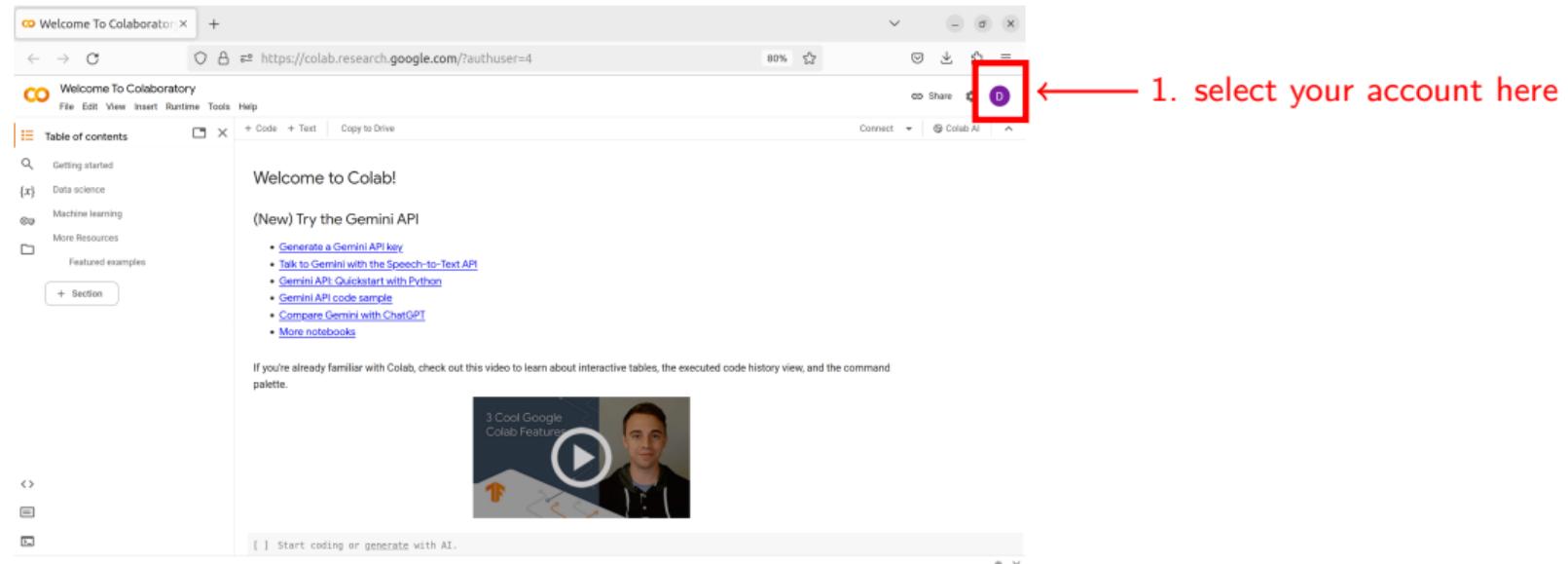
3.4 Try accessing GEE in **Code Editor** (JavaScript IDE)

7. clic "Run" to execute code

6. type code in JavaScript
⇒ ex: official tutorials

2.4. Access GEE in Colab

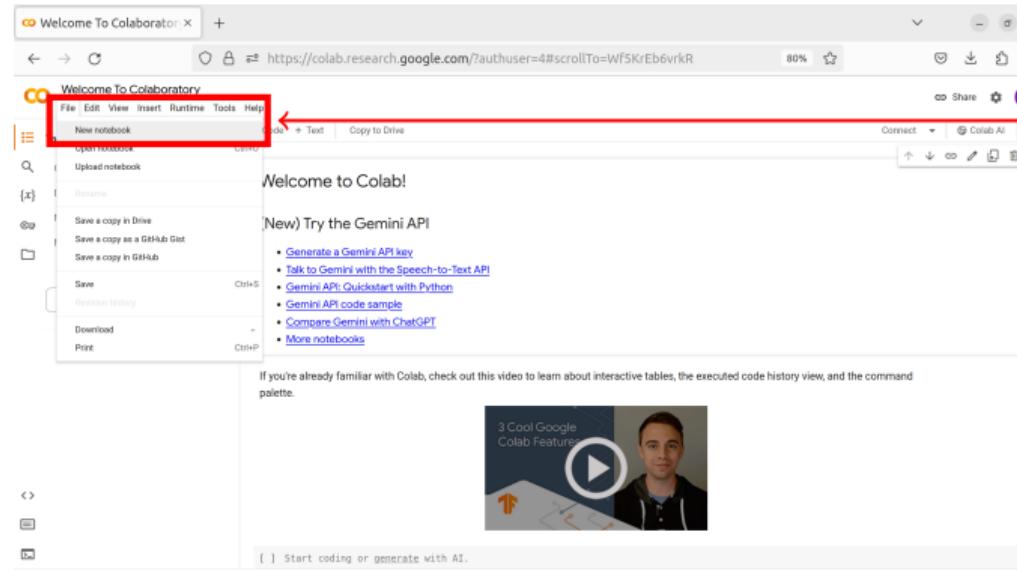
4. Access GEE in Colab

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

2.4. Access GEE in Colab

4. Access GEE in Colab

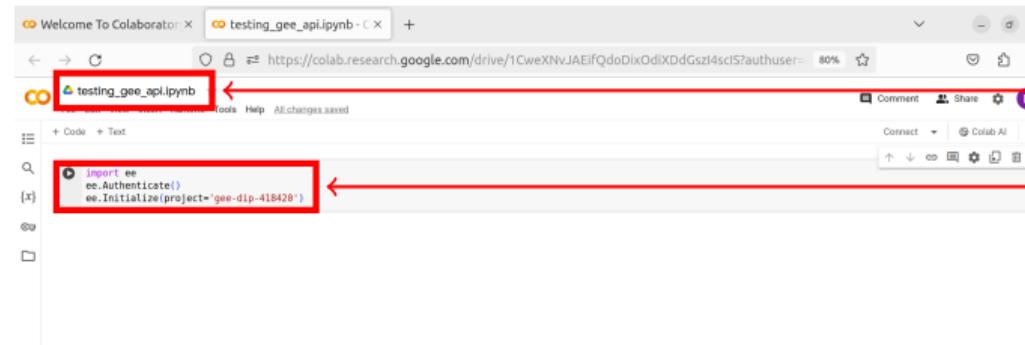
4.2 Create new notebook



2. File - New notebook

2.4. Access GEE in Colab

4. Access GEE in Colab

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

```
import ee
ee.Authenticate()
ee.Initialize(project='gee-dip-418420')
```

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='gee-dip-418420') # <--- use project-id (NOT project-name!)
```

2.4. Access GEE in Colab

4. Access GEE in Colab

4.3 Execute cell & give authorizations in pop-up windows

The image consists of three screenshots illustrating the authorization process for Google Colab Notebooks.

- 5.** A modal dialog titled "Allow this notebook to access your Google credentials?" is displayed. It contains the text: "This will allow code executed in this notebook to access your Google Drive and Google Cloud data. Review the code in this notebook prior to allowing access." Two buttons are present: "No thanks" and "Allow". The "Allow" button is highlighted with a red box.
- 6.** A "Sign in - Google accounts" window titled "Choose an account" is shown. It displays the email address "dip.remotesensing@gmail.com" with a red box around it. Below it is the option "Use another account".
- 7.** A "Sign in - Google accounts" window titled "Sign in to Colaboratory Runtimes" is shown. It displays the email address "dip.remotesensing@gmail.com" with a red box around it. Below it is the option "Use another account". At the bottom, there are "Cancel" and "Continue" buttons, with the "Continue" button highlighted with a red box.
- 8.** A "Sign in - Google accounts" window titled "Collaboratory Runtimes wants access to your Google Account" is shown. It displays the email address "dip.remotesensing@gmail.com" with a red box around it. Below it is the option "Use another account". At the bottom, there are "Cancel" and "Continue" buttons, with the "Continue" button highlighted with a red box.
- 9.** A "Sign in - Google accounts" window titled "Before you can sign in with Google, Collaboratory Runtimes needs to verify your account" is shown. It displays the email address "dip.remotesensing@gmail.com" with a red box around it. Below it is the option "Use another account". At the bottom, there are "Cancel" and "Continue" buttons, with the "Continue" button highlighted with a red box.

2.4. Access GEE in Colab

4. Access GEE in Colab

4.4 Start coding with GEE in Colab !

Welcome To Colaboratory X testing_gee_api.ipynb - C X + https://colab.research.google.com/drive/1CweXNvJAEifQdoDixOdiXDdGsI4scIS?auth=1 80% Comment Share RAM Disk Colab AI

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[x] In [1]:

```
import ee
ee.Authenticate()
ee.Initialize(project='gee-dip-418420')

# 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
```

Out [1]:

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: MSS (Multispectral Scanner)
- Landsat 4	(1982–1993)	Sensor: TM (Thematic Mapper)
- Landsat 5	(1984–2012)	Sensor: TM (Thematic Mapper)
- Landsat 7	(1999–2021)	Sensor: ETM+ (Enhanced Thematic Mapper Plus)
- Landsat 8	(2013–Present)	Sensor: OLI/TIRS (Op. Land Imager / Therm. Infrared Sensor)
- Landsat 9	(2021–Present)	Sensor: OLI/TIRS (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), GRD scenes (Ground Range Detected)
- Sentinel 2	(2015–Present)	Sensor: MSI (Multispectral Instrument)
- Sentinel 3	(2016–Present)	Sensor: OLCI (Ocean and Land Color Instrument)
- Sentinel 5P	(2018–Present)	Sensor: TROPOMI (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='gee-dip-418420') # 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
```

