Space Business Innovation Challenge

Guide for Automated Download of Sentinel Imagery

Prepared by:

PHILIPPINE SPACE AGENCY

September 20, 2025

The information contained in this file is intended solely for the use of registered participants of PhilSA Space Business Innovation Challenge, to whom it is addressed, and others authorized to receive it. It may contain confidential or legally privileged information. If you are not the intended recipient, you are hereby notified that any disclosure, copying, distribution, or taking any action in reliance on the contents of this information is strictly prohibited and may be unlawful. If you have received this file in error, please notify the Philippine Space Agency immediately and then delete it from your system.

GUIDE FOR AUTOMATED DOWNLOAD OF SENTINEL IMAGERY

This document is a guide for the downloading of SENTINEL satellite imagery from the Copernicus Datahub using python scripts. This guide will cover the following:

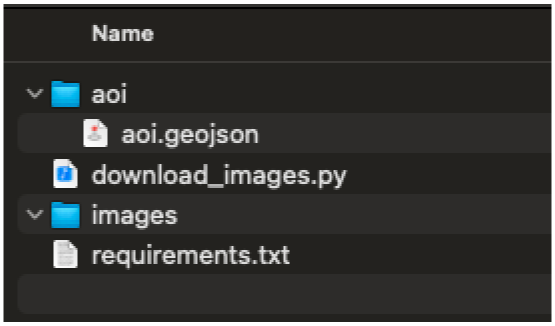
- Installation of a Package and Environment manager for managing python environment
- Creation of a polygon geojson your area of interest using geojson.io
- Obtaining COPERNICUS STAC API access and COPERNICUS S3 access
- Creating of python environment
- Running of the python script provided

This guide assumes that you already have a Copernicus Dataspace Portal. This guide also assumes that you are familiar with Python programming language and have some experience running python programs. If you do not have a Copernicus account yet, please refer to our other guide for the viewing of Copernicus data through its Dataspace portal.

1. Materials to be provided with the guide.

1.1 Breakdown of the files to be provided

Together with the guide, you will be provided with a script.zip that will contain the following files and folders as shown in Figure 1:



```
Name
aoi
    aoi.geojson
download_images.py
images
requirements.txt
```

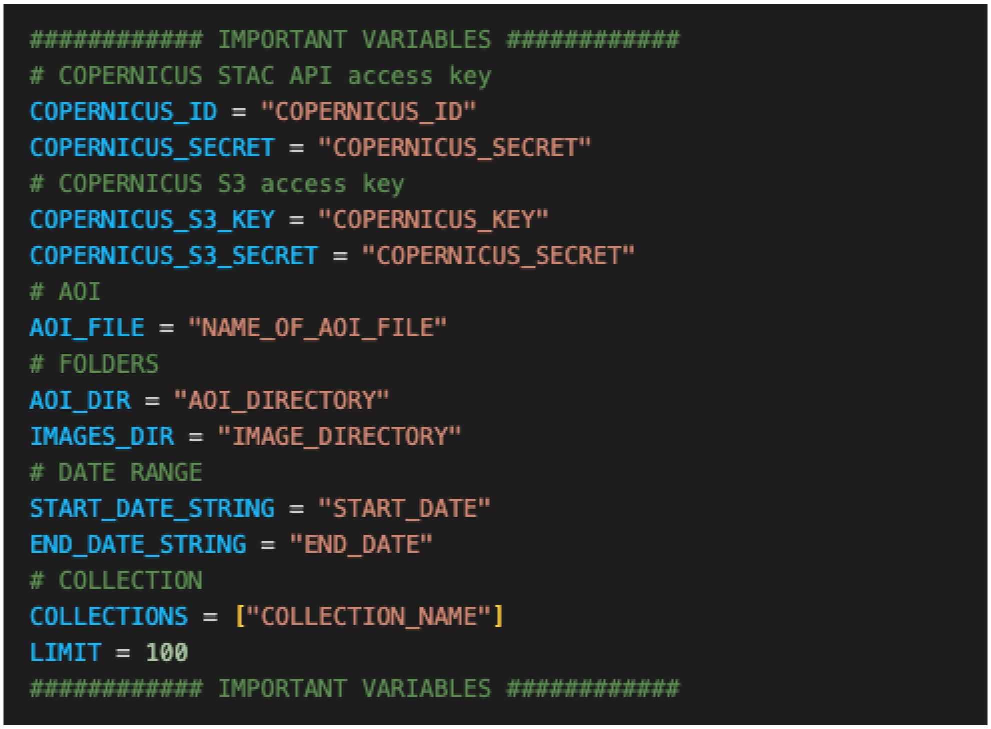
Figure 1. script file directory

- download_images.py - script for downloading sentinel satellite images
- requirements.txt - a list of python packages that are needed to run the script
- aoi – folder where the area of interest created from geojson.io is stored
- images – folder where the downloaded imagery is stored

These files will be provided to you in a zip file. Extract the files and save them in a folder. You will access these files later using your command prompt.

1.2 Important Information needed by the script

To run the script, you need to acquire certain information from its Copernicus account and to provide the parameters of your image search query. Shown in Figure 2 are the parameters that you need to provide to run the script.



```
############ IMPORTANT VARIABLES ############
# COPERNICUS STAC API access key
COPERNICUS_ID = "COPERNICUS_ID"
COPERNICUS_SECRET = "COPERNICUS_SECRET"
# COPERNICUS S3 access key
COPERNICUS_S3_KEY = "COPERNICUS_KEY"
COPERNICUS_S3_SECRET = "COPERNICUS_SECRET"
# AOI
AOI_FILE = "NAME_OF_AOI_FILE"
# FOLDERS
AOI_DIR = "AOI_DIRECTORY"
IMAGES_DIR = "IMAGE_DIRECTORY"
# DATE RANGE
START_DATE_STRING = "START_DATE"
END_DATE_STRING = "END_DATE"
# COLLECTION
COLLECTIONS = ["COLLECTION_NAME"]
LIMIT = 100
############ IMPORTANT VARIABLES ############
```

Figure 2. download images script Important variable

- COPERNICUS STAC API access keys – This access keys are used to query the availability of satellite imagery from the COPERNICUS platform. You can get this information from your COPERNICUS account. The instructions on how to get this information are available in the later part of the document.
    - COPERNICUS_ID - Client ID for calling the Copernicus STAC API
    - COPERNICUS_SECRET - Client Secret for calling the Copernicus STAC API
- COPERNICUS S3 access keys - This access keys are used to download satellite imagery from the COPERNICUS S3 bucket. You can get this information from your COPERNICUS account. Instructions on how to get this information are available.
    - COPERNICUS_S3_KEY- Access key for accessing the Copernicus S3 bucket
    - COPERNICUS_S3_SECRET - Secret key for accessing the Copernicus S3 bucket
- AOI_FILE – the AOI geojson filename that will be used to search satellite imagery. The instructions on how to create this AOI geojson are available in the later part of the document.

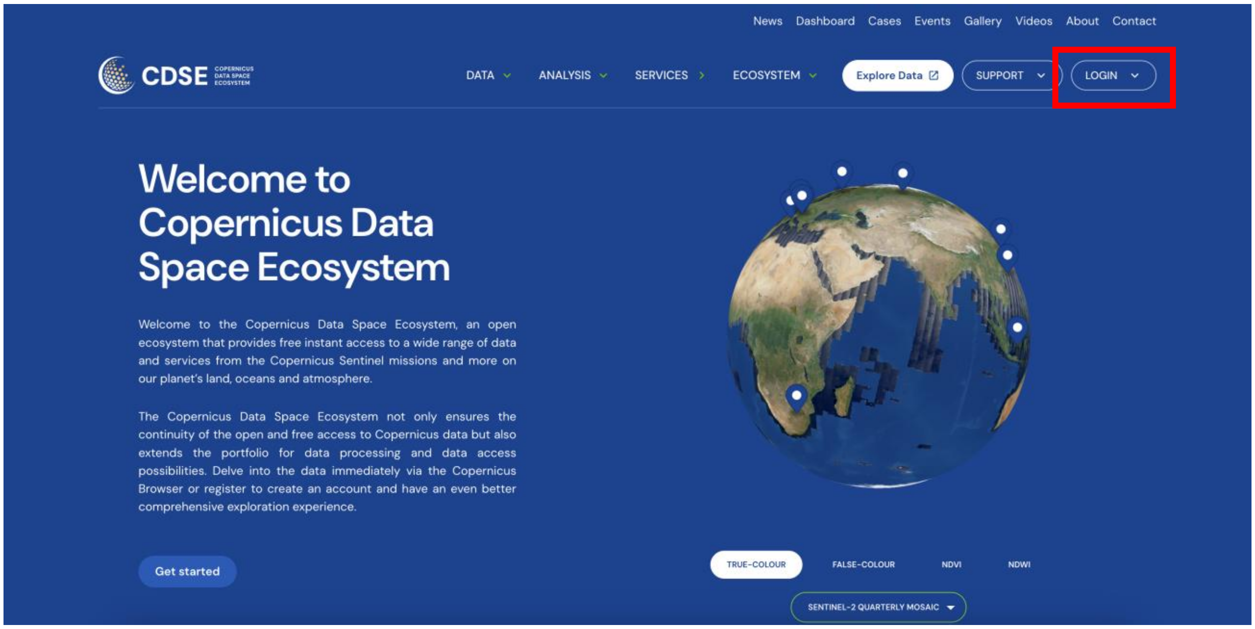
- AOI_DIR – the name of the folder where we will store the AOI geojson will be stored. This directory will be read by the script. For our case, we will be using the AOI_DIR of "aoi".
- IMAGES_DIR – the name of the folder where the COPERNICUS satellite imagery will be downloaded. For our case, we will be using the folder name "images"
- START_DATE_STRING – the start date of the satellite imagery acquisition that we will download. The start date needs to be in YYYY-MM-DD format.
- END_DATE_STRING – the end date of the satellite imagery acquisition that we will download. The end date needs to be in YYYY-MM-DD format.
- COLLECTIONS – the list of collections from the COPERNICUS data that we will be searching for imagery.
- LIMIT – the limit of imagery that will be returned by the query. If the number of satellite imagery that is covered by the filter, the API will only return the number indicated in the limit.

2. Getting the COPERNICUS access key

This section will give you instructions on how to acquire COPERNICUS STAC API access keys and COPERNICUS S3 access keys from your COPERNICUS account. This section assumes that you already have a registered COPERNICUS account. If you do not have a registered COPERNICUS account, please refer to our guide on COPERNICUS account creation and come back here when you have a COPERNICUS account.

2.1 Getting the COPERNICUS STAC API access key

Open your preferred browser and go to the COPERNICUS Dataspace website https://dataspace.copernicus.eu/



Welcome to Copernicus Data Space Ecosystem

Welcome to the Copernicus Data Space Ecosystem, an open ecosystem that provides free instant access to a wide range of data and services from the Copernicus Sentinel missions and more on our planet's land, oceans and atmosphere.

The Copernicus Data Space Ecosystem not only ensures the continuity of the open and free access to Copernicus data but also extends the portfolio for data processing and data access possibilities. Delve into the data immediately via the Copernicus Browser or register to create an account and have an even better comprehensive exploration experience.

Figure 3 Copernicus Data Space home page

From the Home Page as shown in Figure 3 of the Copernicus Data Space Ecosystem, hover on the **Login** button

A selection will apear as shown in Figure 3.1

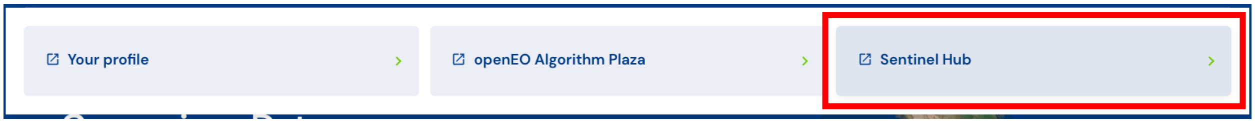
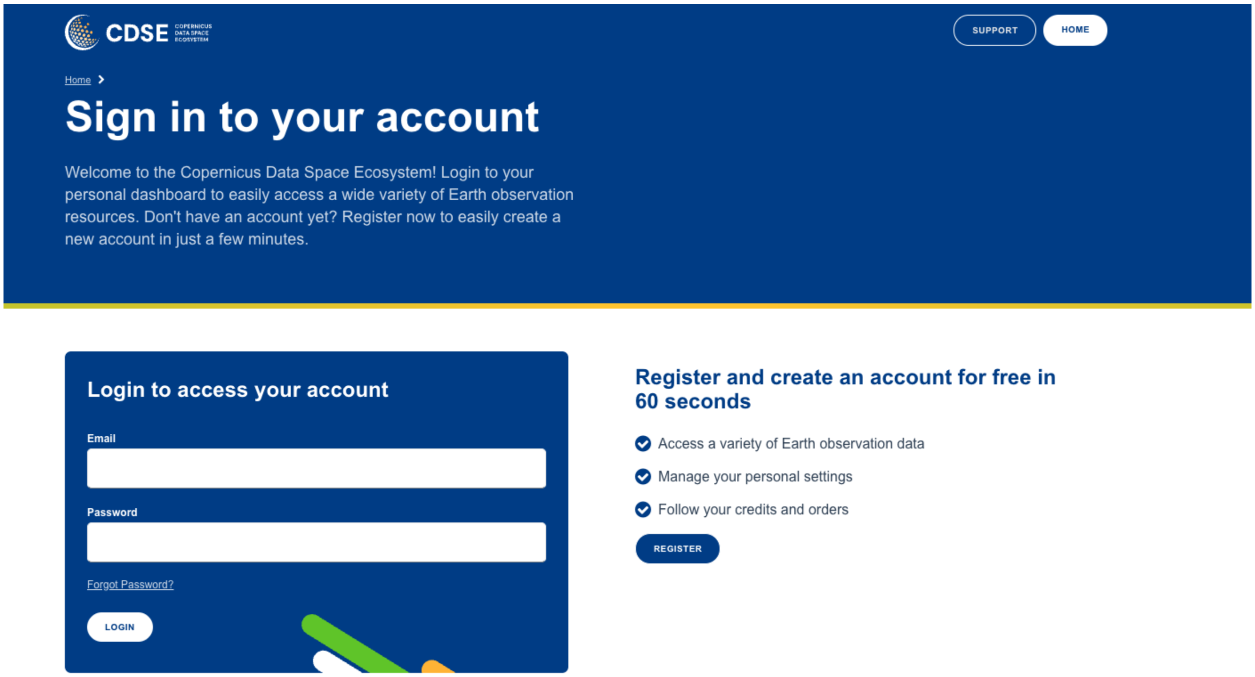


Figure 3.1 Login section

From the selection click on the **Sentinel Hub** button. This will take you to the login page as shown in Figure 4.



Sign in to your account

Welcome to the Copernicus Data Space Ecosystem! Login to your personal dashboard to easily access a wide variety of Earth observation resources. Don't have an account yet? Register now to easily create a new account in just a few minutes.

Login to access your account

Email

Password

Forgot Password?

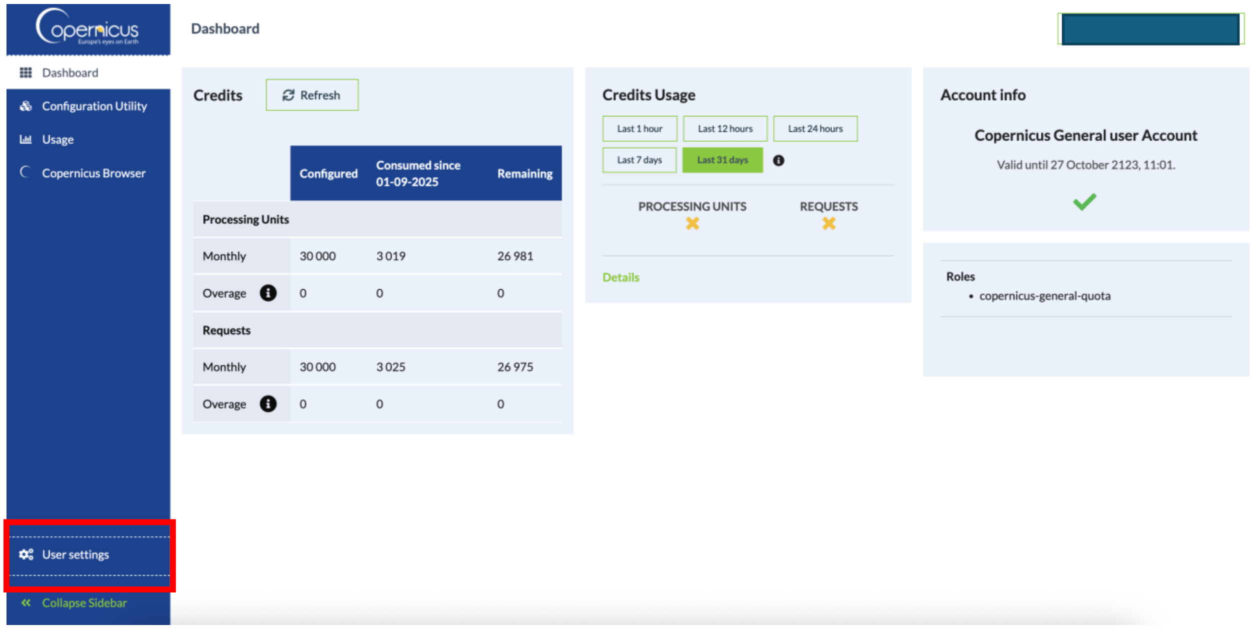
Register and create an account for free in 60 seconds

- Access a variety of Earth observation data
- Manage your personal settings
- Follow your credits and orders

Figure 4 Copernicus Data Space login page

Provide your Copernicus credentials and click login.

This will take you to the Copernicus shapps dashboard.



Dashboard

Credits

	Configured	Consumed since 01-09-2025	Remaining
Processing Units			
Monthly	30 000	3 019	26 981
Overage	0	0	0
Requests			
Monthly	30 000	3 025	26 975
Overage	0	0	0

Credits Usage

PROCESSING UNITS ✗ REQUESTS ✗

Account info

Copernicus General user Account

Valid until 27 October 2123, 11:01.

✓

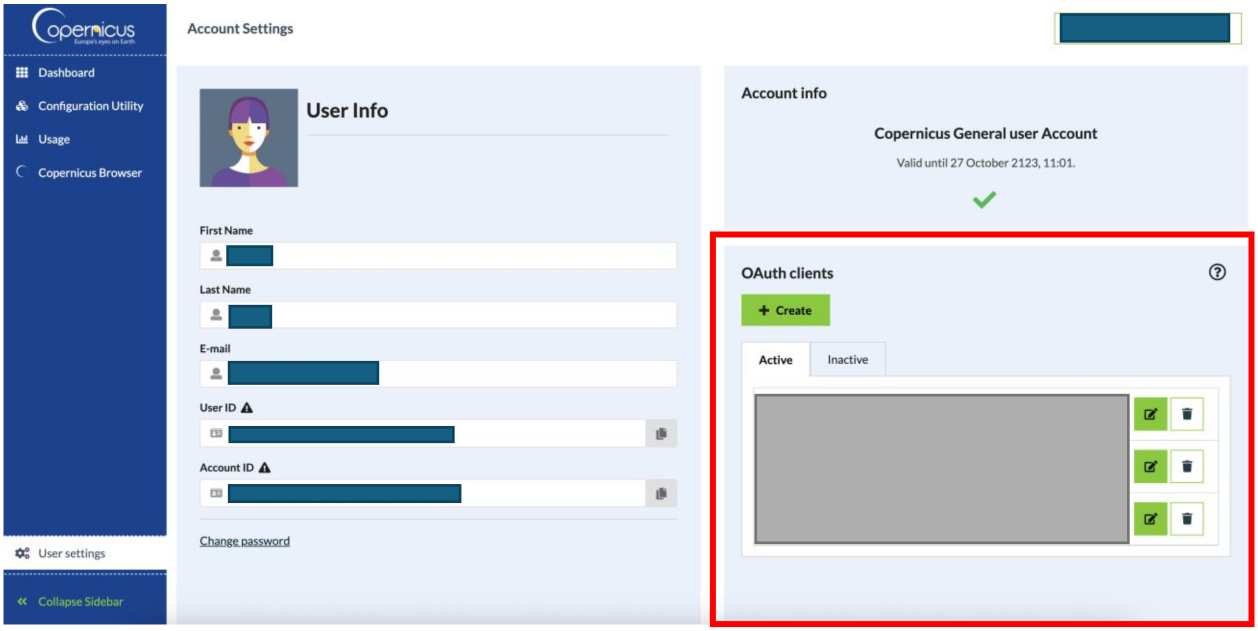
Roles

- copernicus-general-quota

Figure 5 Copernicus Data Space shapps dashboard

From the Copernicus shapps dashboard as shown in Figure 5 click the **User settings** button on the bottom left of the dashboard.

You will be redirected to the Accounts Settings page



Account Settings [redacted]

User Info

First Name
[redacted]

Last Name
[redacted]

E-mail
[redacted]

User ID
[redacted]

Account ID
[redacted]

Change password

Account info

Copernicus General user Account

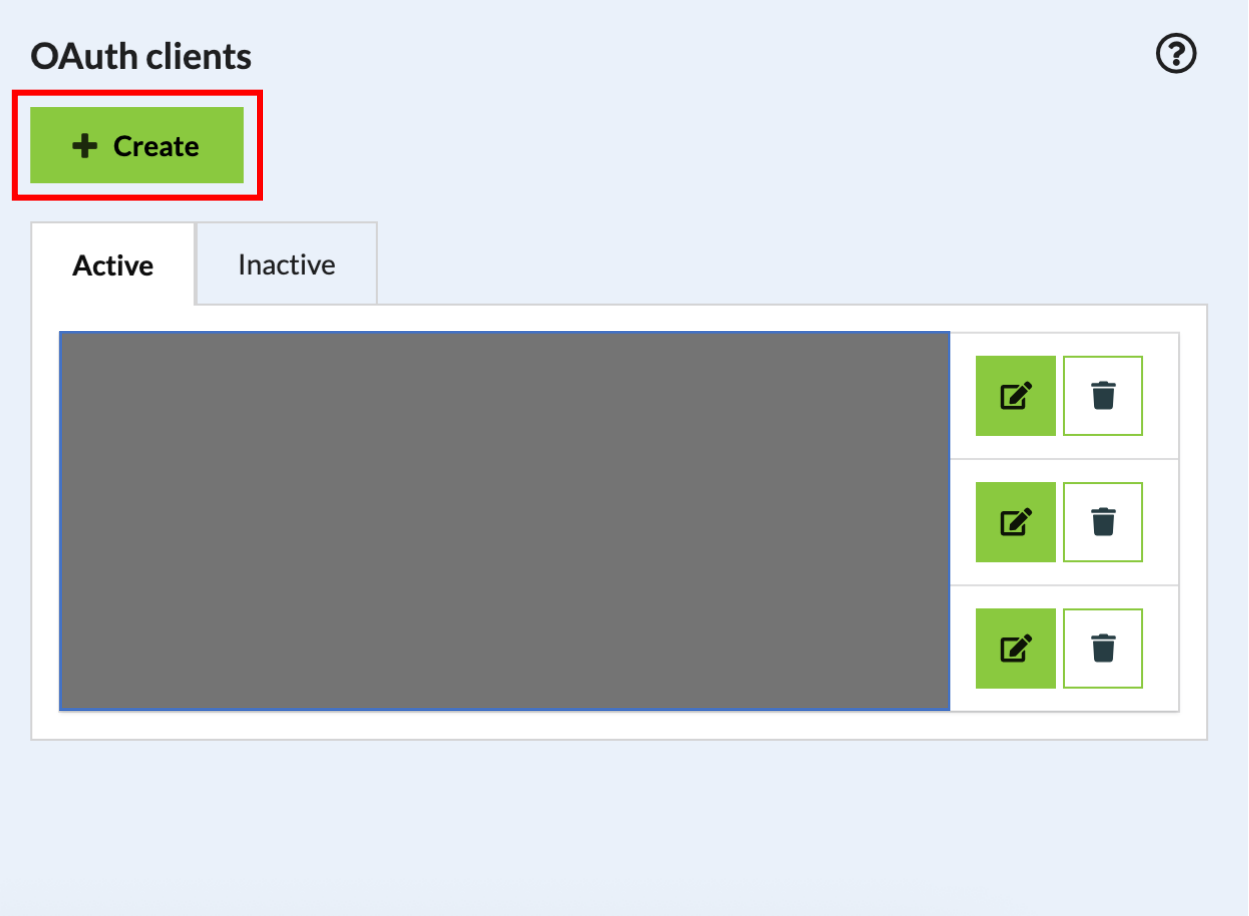
Valid until 27 October 2123, 11:01.

OAuth clients

[redacted]

Figure 6 Account Settings

In the OAuth client section as shown in Figure 6 in the bottom right corner is the list of OAuth clients for accessing the Copernicus STAC API.

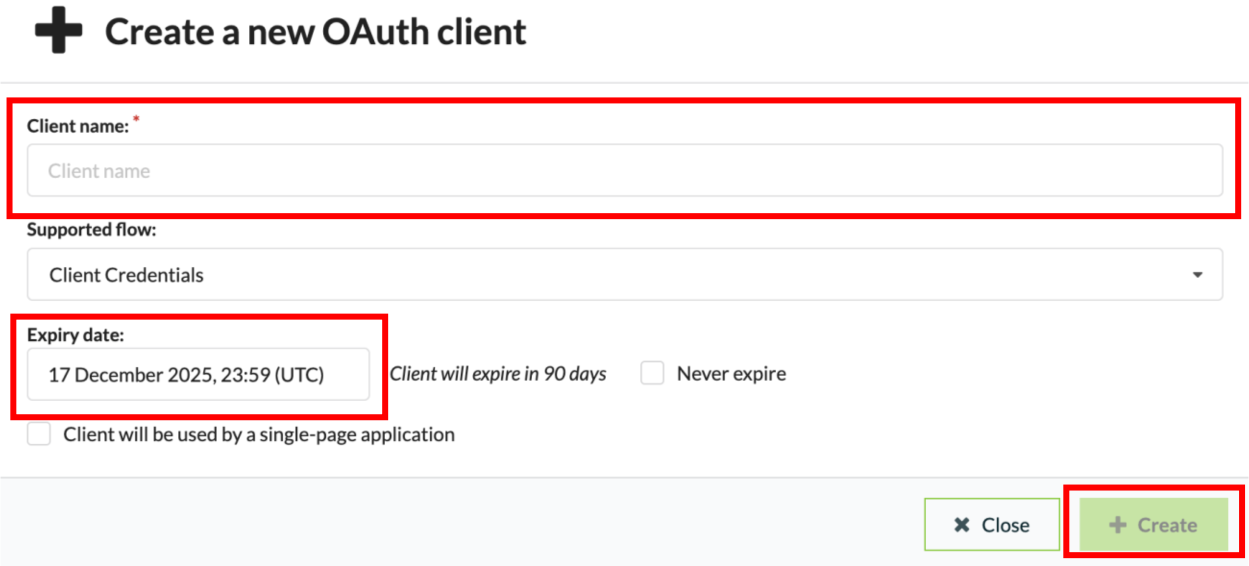


[redacted]

Figure 7 OAuth client section

From the **OAuth clients** section as shown in Figure 7, click on the click on the **Create** button

A Create a new OAuth client modal will appear.



Create a new OAuth client

Client name: *

Client name

Supported flow:

Client Credentials

Expiry date:

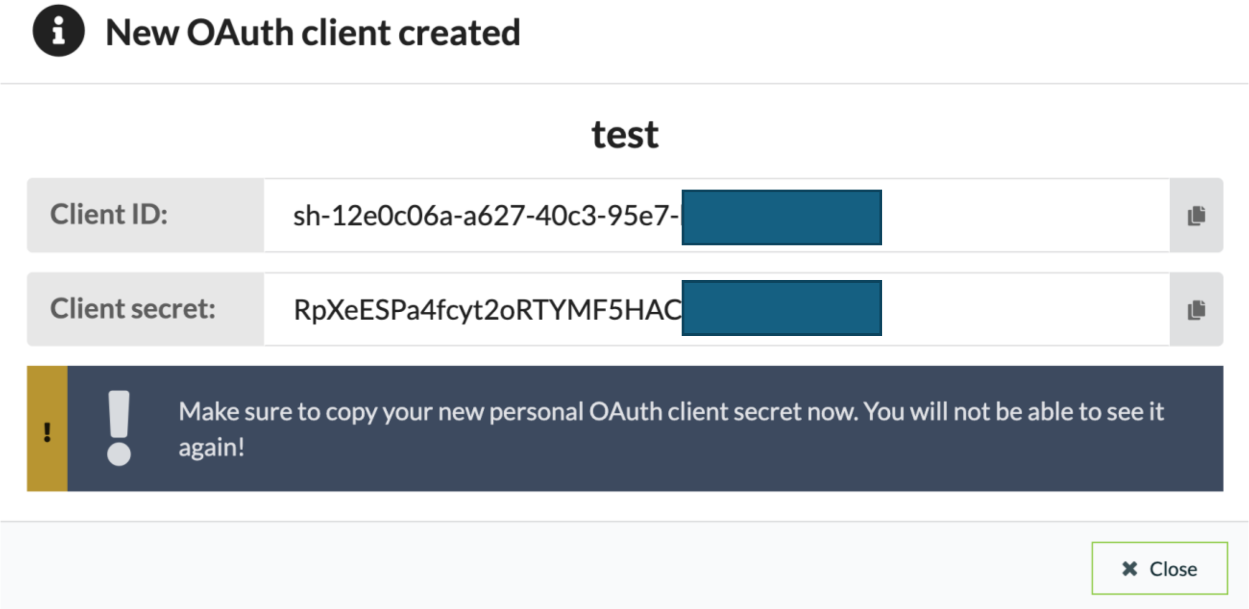
17 December 2025, 23:59 (UTC)

Client will expire in 90 days □ Never expire

□ Client will be used by a single-page application

Figure 8 Create new OAuth client modal

Set the **Client name** and **Expiry date** of the OAuth client as shown in Figure 8. The client name is the label for the access key that will be generated. The expiry date is the date when the access key will expire. Once you fill out the parameters click the **Create** button.



New OAuth client created

test

Client ID: sh-12e0c06a-a627-40c3-95e7-[redacted]

Client secret: RpXeESPa4fcyt2oRTYMF5HAC[redacted]

Make sure to copy your new personal OAuth client secret now. You will not be able to see it again!

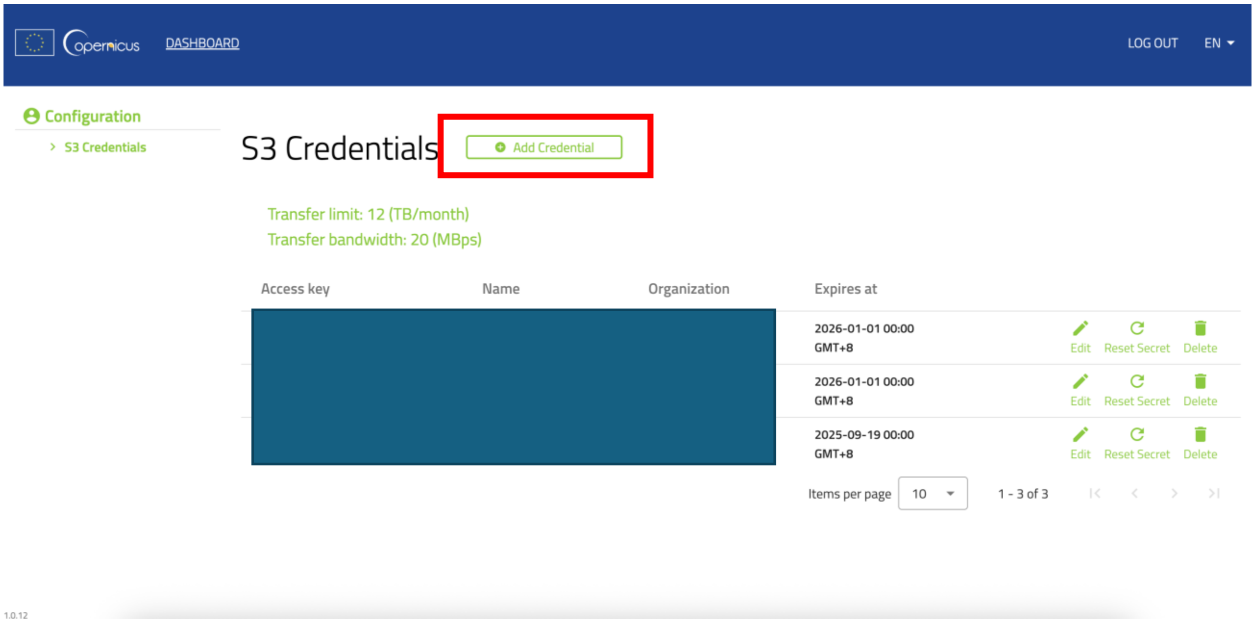
Figure 9 OAuth Client access keys

A modal for the OAuth client created will appear as shown in Figure 9. Copy the **Client ID** and **Client secret.** The Client ID is your **COPERNICUS_ID** and the Client secret is your **COPERNICUS_SECRET.** This will be used in the script. Copy these values because they will not be viewable after closing the modal.

2.2 Getting the COPERNICUS S3 API access key

Go to the Copernicus Data Space S3 keys manager dashboard. https://eodata-s3keysmanager.dataspace.copernicus.eu/panel/s3-credentials

This is the dashboard where you can manage your Copernicus access keys for connecting to the Copernicus S3 data storage.



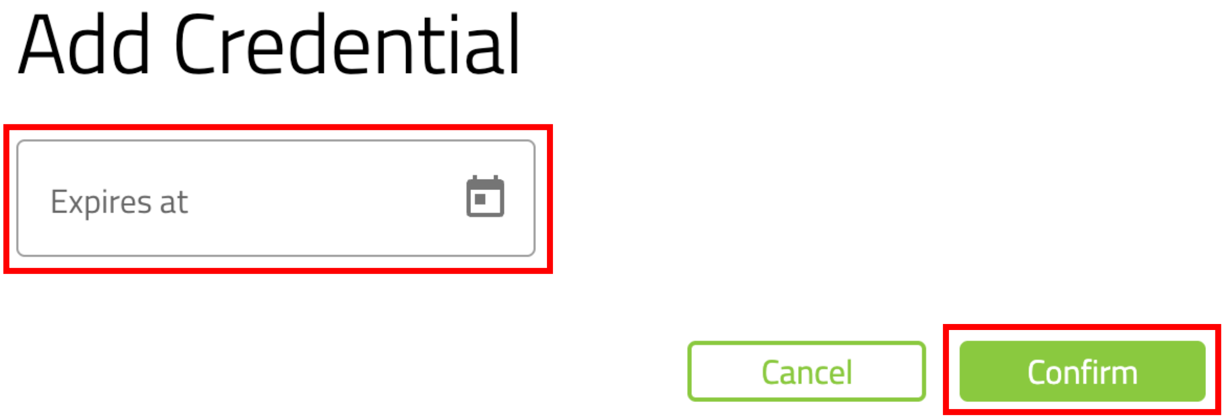
S3 Credentials

Transfer limit: 12 (TB/month)
Transfer bandwidth: 20 (MBps)

Access key	Name	Organization	Expires at
[redacted]	[redacted]	[redacted]	2026-01-01 00:00 GMT+8
[redacted]	[redacted]	[redacted]	2026-01-01 00:00 GMT+8
[redacted]	[redacted]	[redacted]	2025-09-19 00:00 GMT+8

Figure 10 Copernicus Data Space S3 keys manager dashboard

On the dashboard as shown in Figure 10, click the **Add Credential** button.

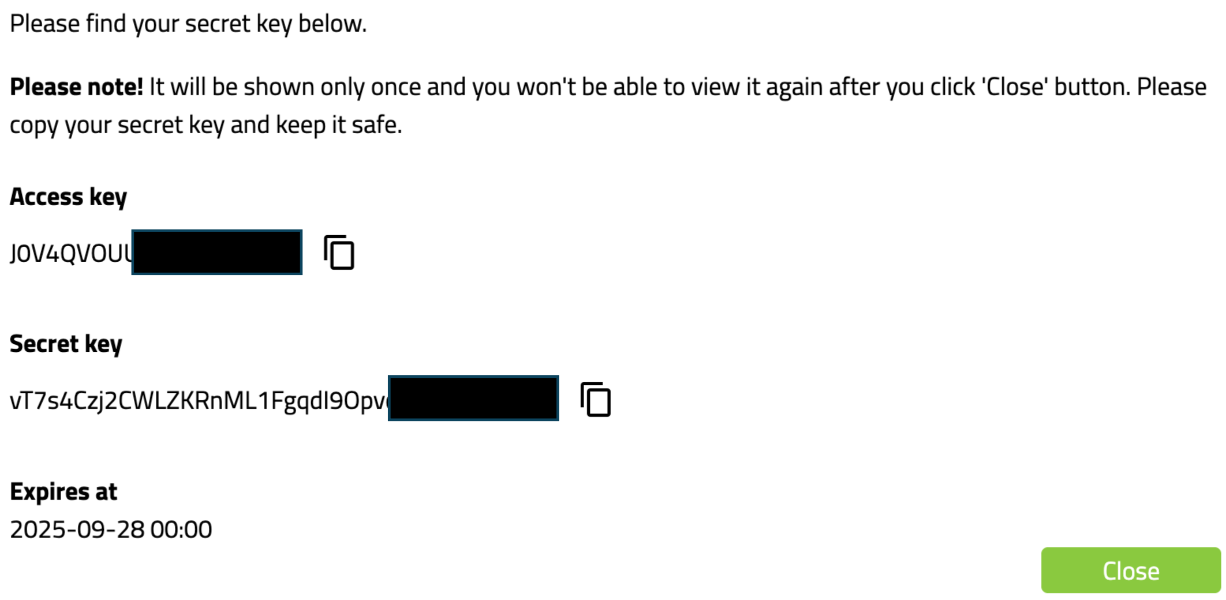


Add Credential

Expires at

Figure 11 Add new credentials modal

An **Add Credential** modal will appear as shown in Figure 11. Set the **Expires at** date in the modal. This will be the expiry date of the S3 access keys. Once set, click the **Confirm** button.



Please find your secret key below.

Please note! It will be shown only once and you won't be able to view it again after you click 'Close' button. Please copy your secret key and keep it safe.

Access key

JOV4QVOU[redacted]

Secret key

vT7s4Czj2CWLZKRnML1FgqdI9Opv[redacted]

Expires at
2025-09-28 00:00

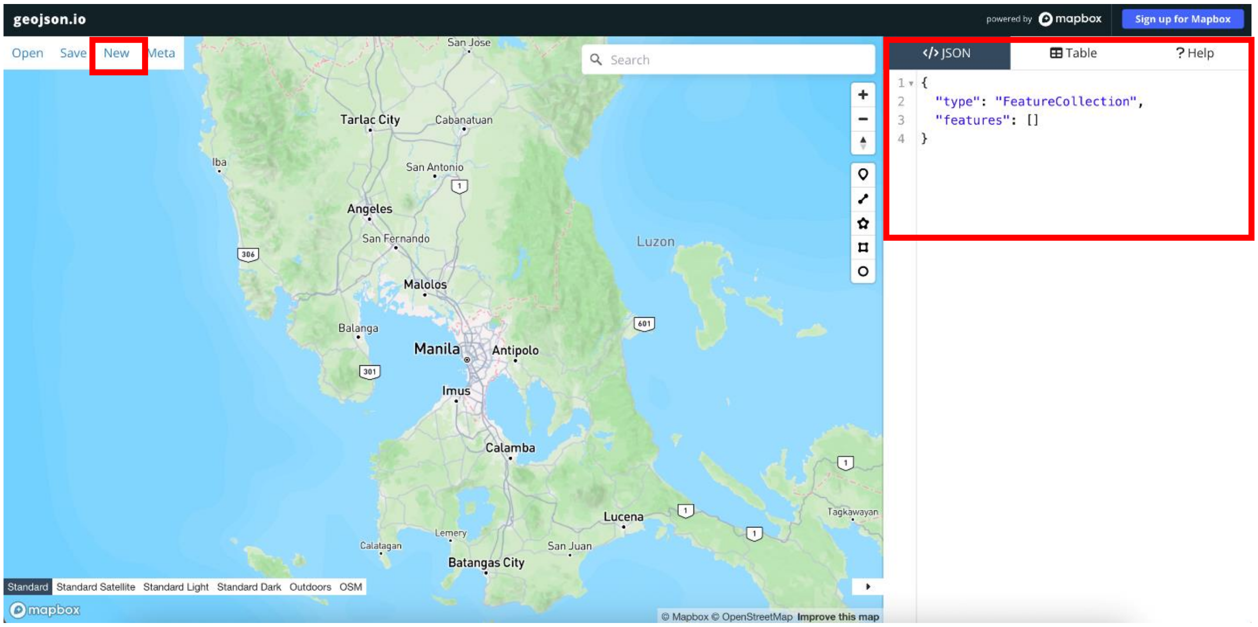
Figure 12 S3 storage access keys

A modal containing your access keys for the S3 data storage will appear as shown in Figure 12. Copy the **Access key** and **Secret key.** The Access key is your **COPERNICUS_S3_KEY** and the Client secret is your **COPERNICUS_S3_SECRET**. This will be used in the script. Copy these values because they will not be viewable after closing the modal.

3. Creation of area of interest bounding box geojson

This section will show you how to create a geojson file of your area of interest using a free website, geojson.io

Go to https://geojson.io/ in the map section navigate to your general area of interest

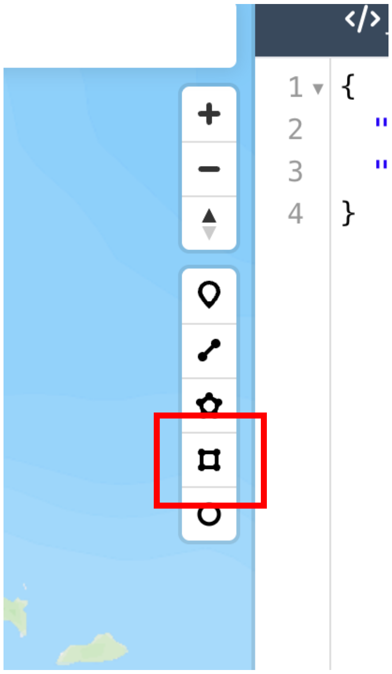


```
{
  "type": "FeatureCollection",
  "features": []
}
```

Figure13 Geojson.io home page

From the Geojson.io home page, as shown in Figure 13 ,ake sure that in the **</>JSON** tab on the right, that the features array is blank. If it is not blank, click on **New** in the top left of the map to clear the features.

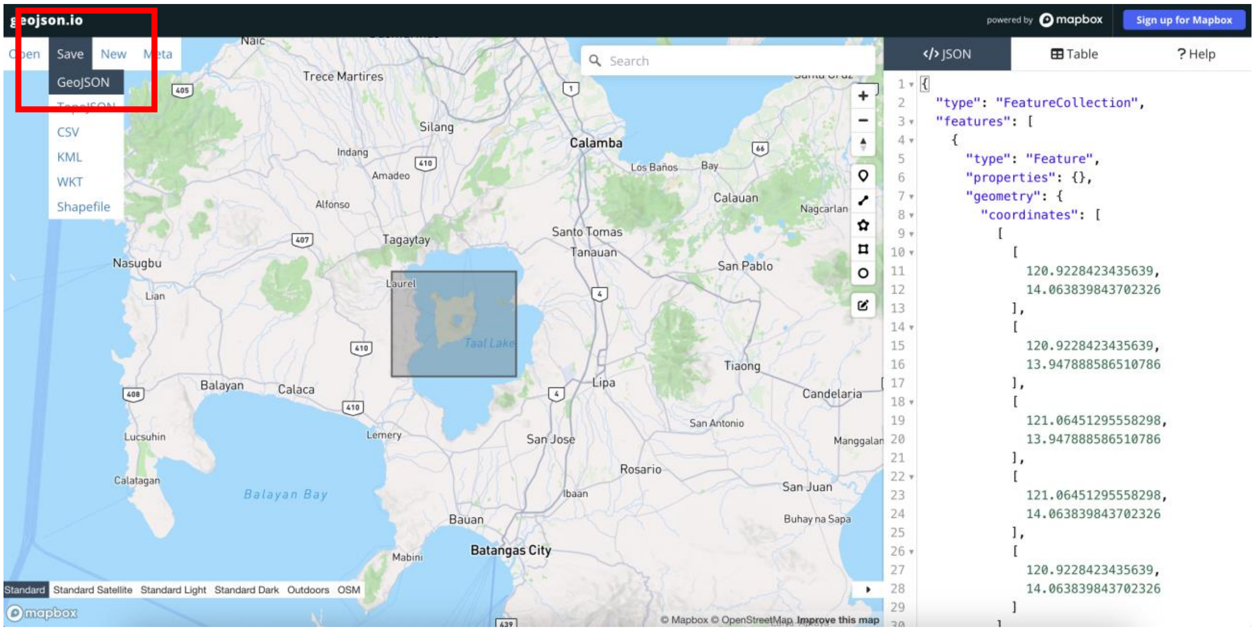
The right side of the map has a column of tools for interacting with the map as shown in Figure 14.



```
{
    "
    "
}
```

Figure 14 Map tools

Click on the **Draw Rectangular Polygon** button. After clicking the button, on the map click the first corner of your area of interest, then drag your mouse to the other corner of your area of interest. As you drag your mouse, a rectangular polygon should appear in the map showing you the area where you will create your area of interest. When you have decided on the corners of your area, click the map again.



```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [
          [
            [
              120.9228423435639,
              14.063839843702326
            ],
            [
              120.9228423435639,
              13.947888586510786
            ],
            [
              121.06451295558298,
              13.947888586510786
            ],
            [
              121.06451295558298,
              14.063839843702326
            ],
            [
              120.9228423435639,
              14.063839843702326
            ]
          ]
```

Figure 15 Save geojson file

The map should have a polygon of your area of interest as shown in Figure 15. The panel on the right of the map shows the geojson text representation of your area of interest. We want to save that to a .geojson file. Click on the **Save** button on the upper left of the map. After that click the **GeoJSON** button. This will download a **map.geojson** file. This file will be used by the script. Rename the geojson file to the **AOI_FILE** variable in the script.

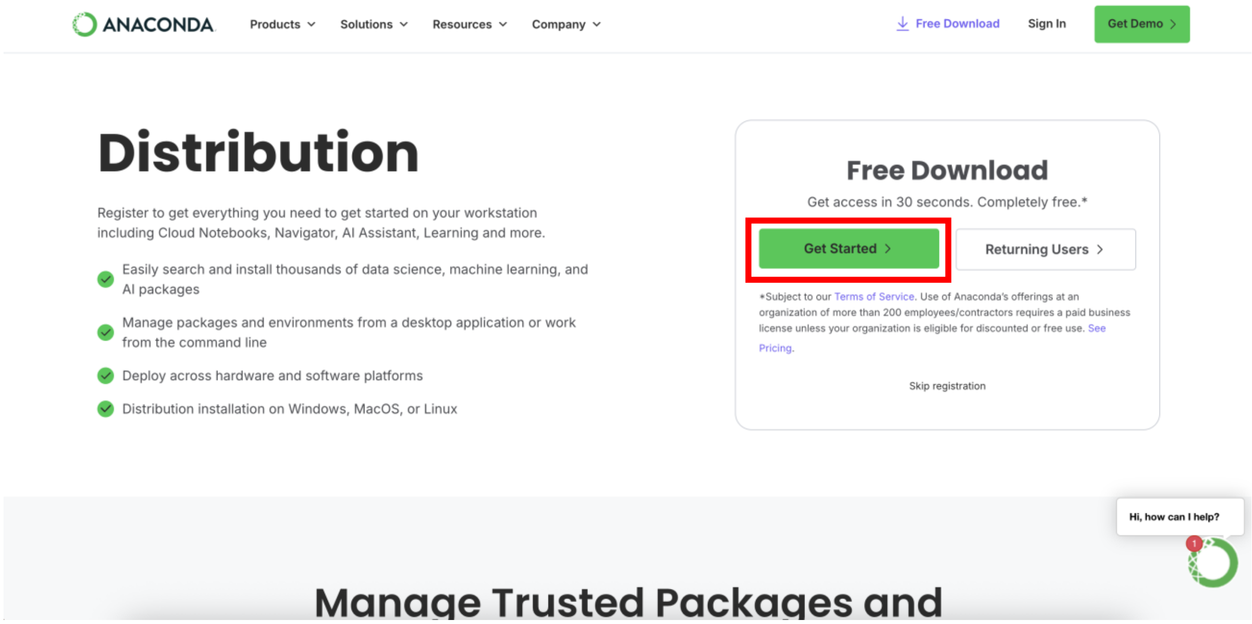
Copy this file to the **aoi** folder of the script.

4. Setting up your Python environment

This section will instruct you on how to setup your Python environment. For this, you will be using **Miniconda**, a package and environment manager. You can skip this part if you already have a python manager installed. Please note that we will be using Python 3.11 or higher for the setup.

4.1 Downloading of Miniconda installer

Go to the Anaconda download page https://www.anaconda.com/download



Distribution

Register to get everything you need to get started on your workstation including Cloud Notebooks, Navigator, AI Assistant, Learning and more.

- Easily search and install thousands of data science, machine learning, and AI packages
- Manage packages and environments from a desktop application or work from the command line
- Deploy across hardware and software platforms
- Distribution installation on Windows, MacOS, or Linux

Free Download

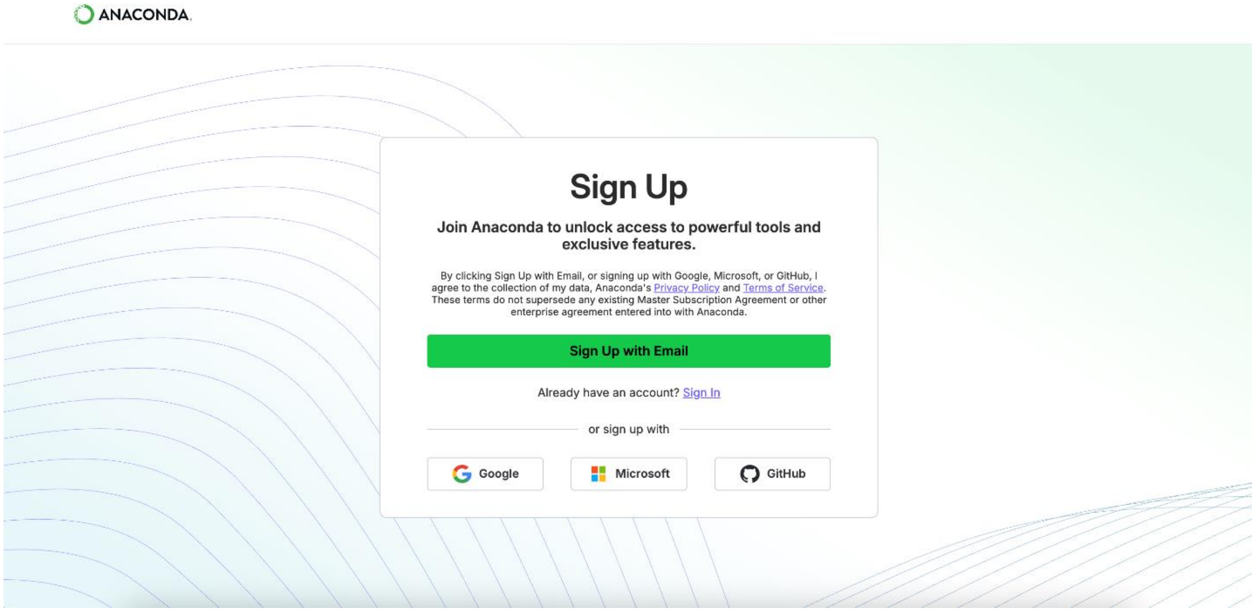
Get access in 30 seconds. Completely free.*

*Subject to our Terms of Service. Use of Anaconda's offerings at an organization of more than 200 employees/contractors requires a paid business license unless your organization is eligible for discounted or free use. See Pricing.

Manage Trusted Packages and

Figure 16 Anaconda distribution page

In the download page as shown in Figure 16 click **Get Started**.



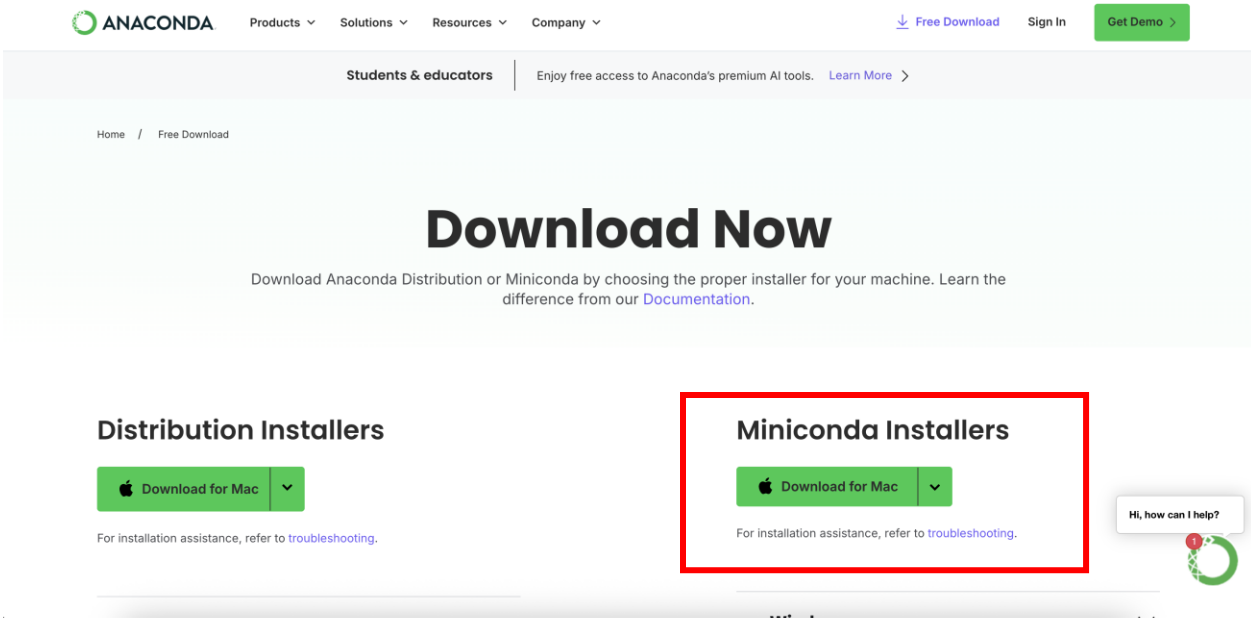
Sign Up

Join Anaconda to unlock access to powerful tools and exclusive features.

By clicking Sign Up with Email, or signing up with Google, Microsoft, or GitHub, I agree to the collection of my data, Anaconda's Privacy Policy and Terms of Service. These terms do not supersede any existing Master Subscription Agreement or other enterprise agreement entered into with Anaconda.

Figure 17 Anaconda sign up page

You will be prompted with a signup page as shown in Figure 17. Continue with the signup page, use any of the available options to sign up. Once you have signed up, you will be directed to a download page as shown in Figure 18.



Enjoy free access to Anaconda's premium AI tools. Learn More

Download Now

Download Anaconda Distribution or Miniconda by choosing the proper installer for your machine. Learn the difference from our Documentation.

For installation assistance, refer to troubleshooting.

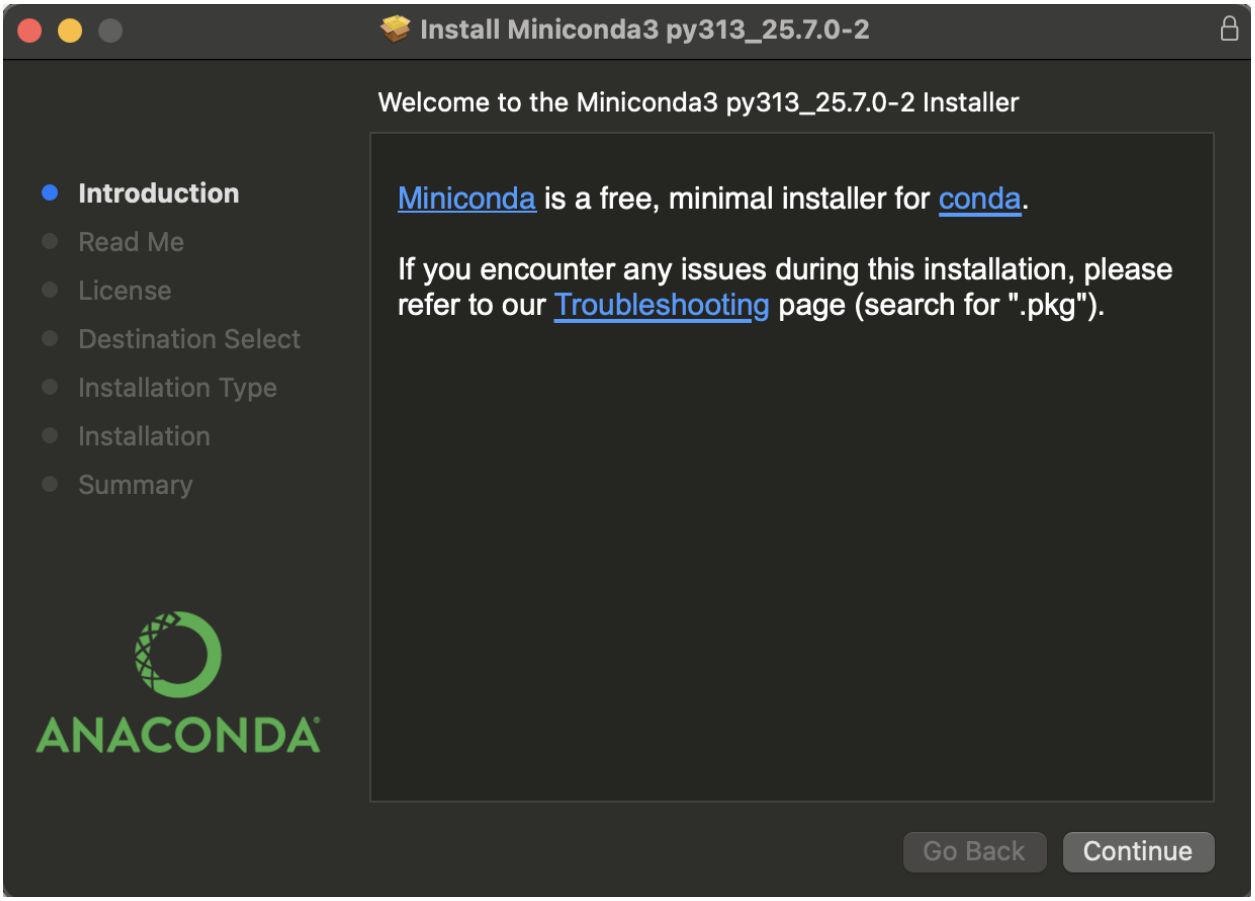
For installation assistance, refer to troubleshooting.

Figure18 Anaconda download page

Click the download button for the **Miniconda Installers.** Select the installer for your particular device. The installer will be downloaded to your device.

4.2 Install the Miniconda installer

Double click the miniconda installer. An install wizard will appear as shown in Figure 19



Install Miniconda3 py313_25.7.0-2

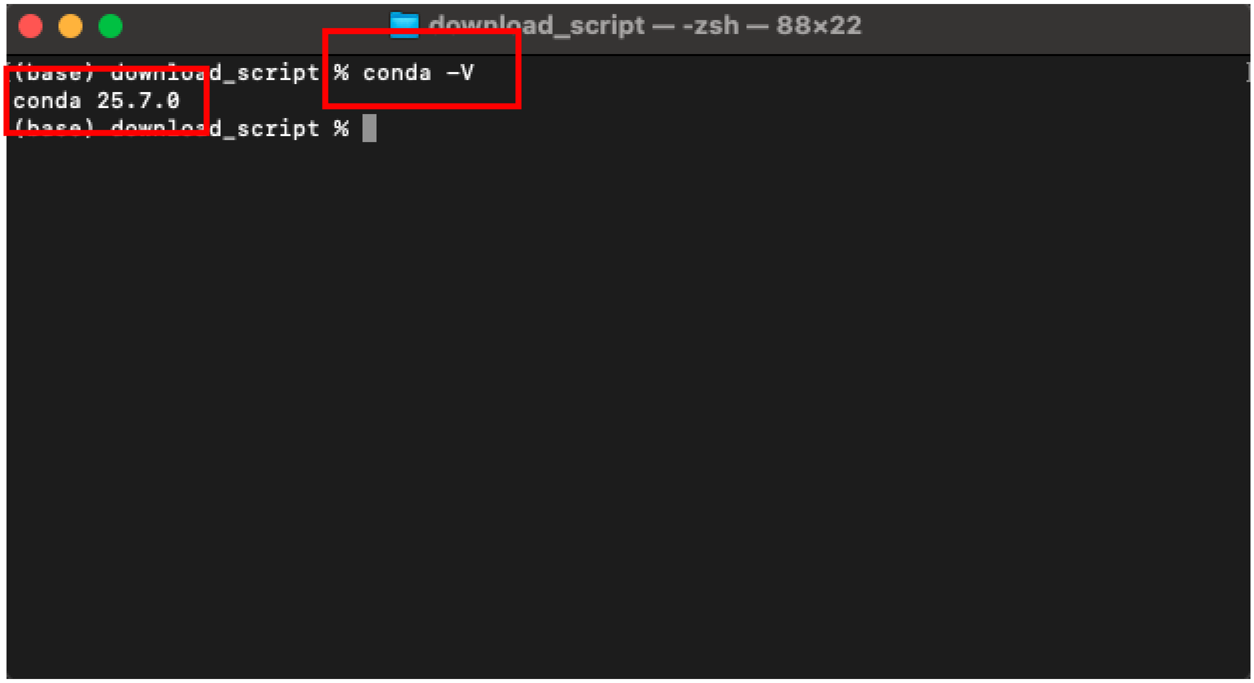
Welcome to the Miniconda3 py313_25.7.0-2 Installer

Miniconda is a free, minimal installer for conda.

If you encounter any issues during this installation, please refer to our Troubleshooting page (search for ".pkg").

Figure 19 Miniconda installation

Continue with the installation of the Miniconda. Once done, open your Command Prompt / Terminal application.



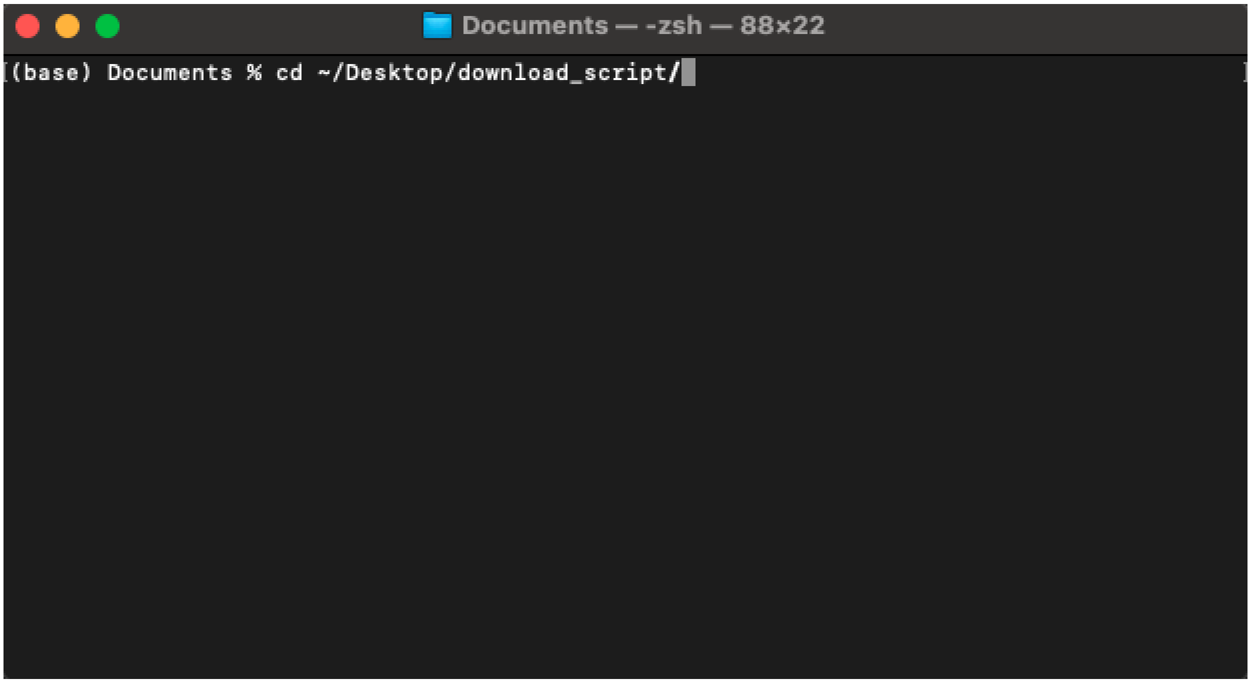
```
(base) download_script % conda -V
conda 25.7.0
(base) download_script % 
```

Figure 20 Checking of miniconda version in terminal

Run the **conda –V** command as shown in Figure 20. It should respond with a version of conda. Conda is the package manager that you will use for your python environment.

4.3 Creation of Python environment

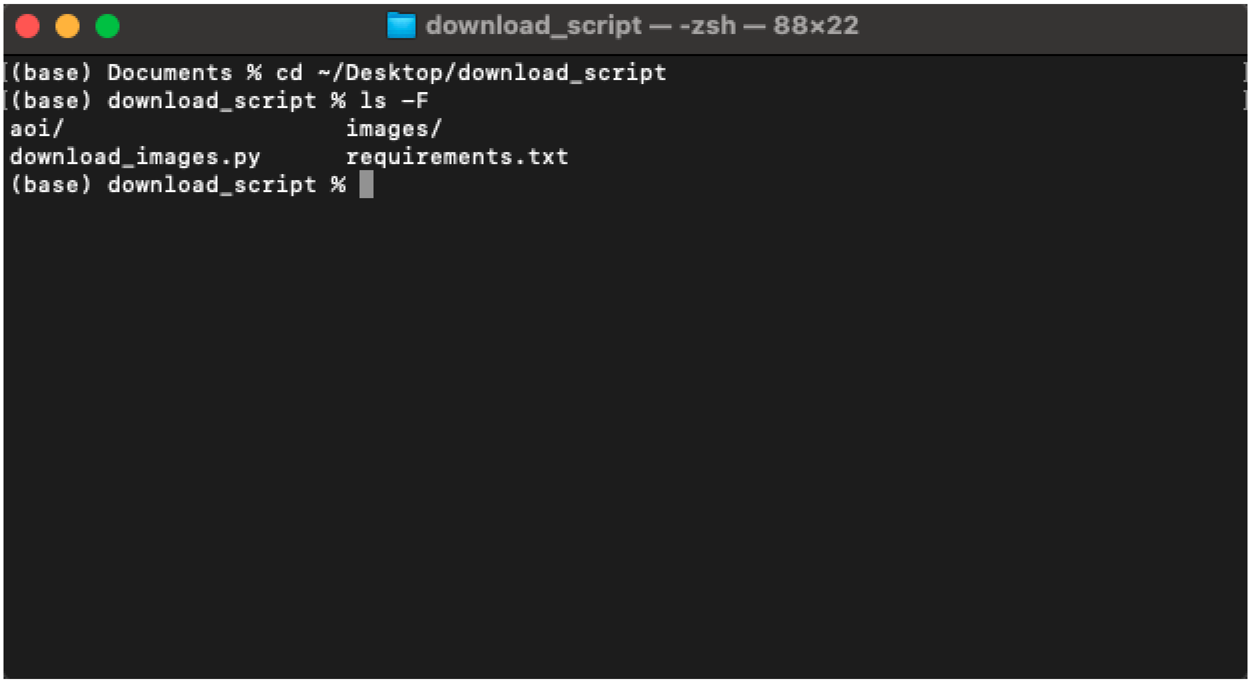
Open your command prompt / terminal.



```
(base) Documents % cd ~/Desktop/download_script/
```

Figure 21 Change of directory to script directory

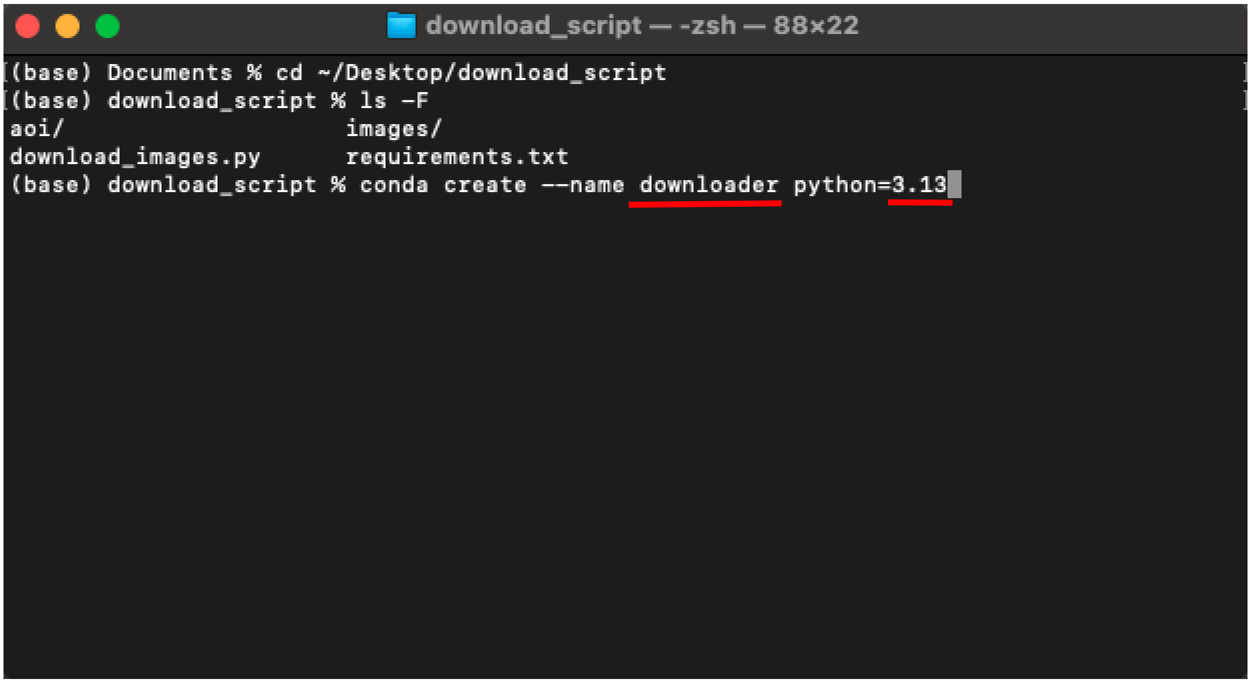
Change directory to the directory of the folder in the zip file provided to you as shown in Figure 21. Run **cd {directory path of the download script folder}.** In the case above, the download_script folder is in the Desktop folder, so the command is cd to Desktop/download_script/



```
(base) Documents % cd ~/Desktop/download_script
(base) download_script % ls -F
aoi/                    images/
download_images.py      requirements.txt
(base) download_script % 
```

Figure 22 Check files inside script directory

Run **ls –F** for mac users and **dir** for windows users as shown in Figure 22. This will list down the files and directories inside our current directory. In our case, the folder contains download_images.py file, requirements.txt file, aoi folder and images folder.



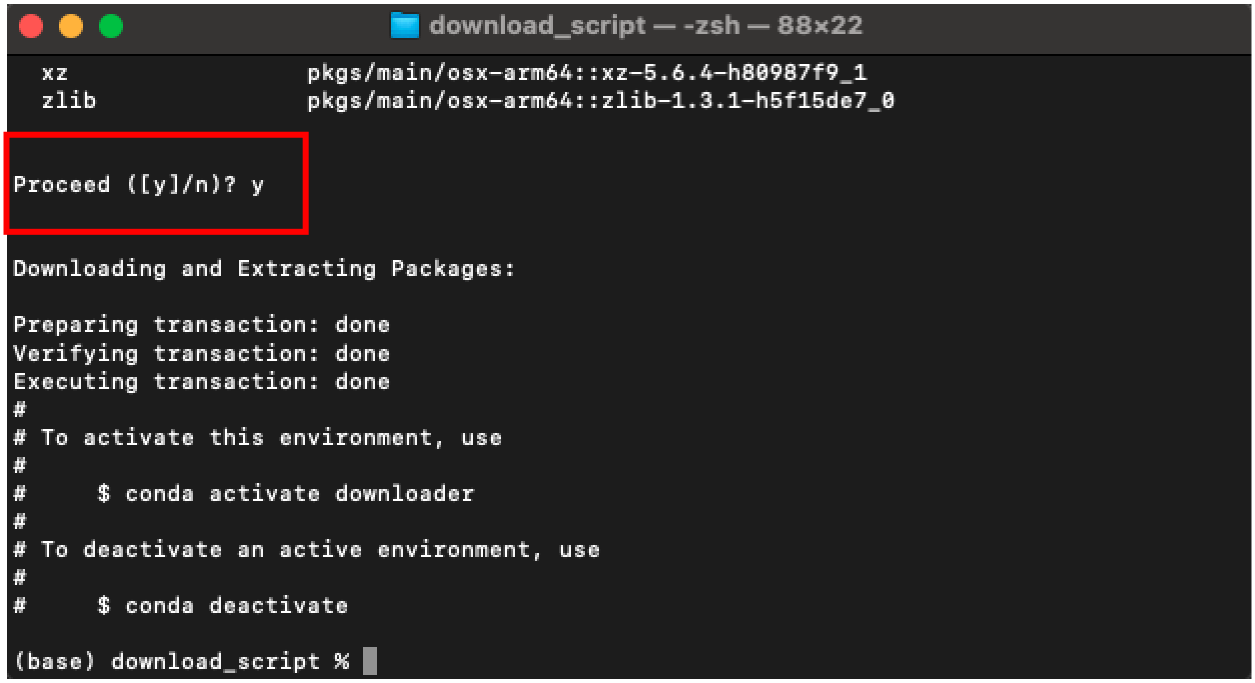
```
(base) Documents % cd ~/Desktop/download_script
(base) download_script % ls -F
aoi/                     images/
download_images.py       requirements.txt
(base) download_script % conda create --name downloader python=3.13
```

Figure 23 Create new conda environment command

To create a python environment, call the following command (replace the **environment**_name and **python_version** to your desired variables):

conda create --name {environment_name} python={python_version}

Refer to Figure 23. In the case above, the environment name was set to **downloader,** and the python version was set to **3.13**



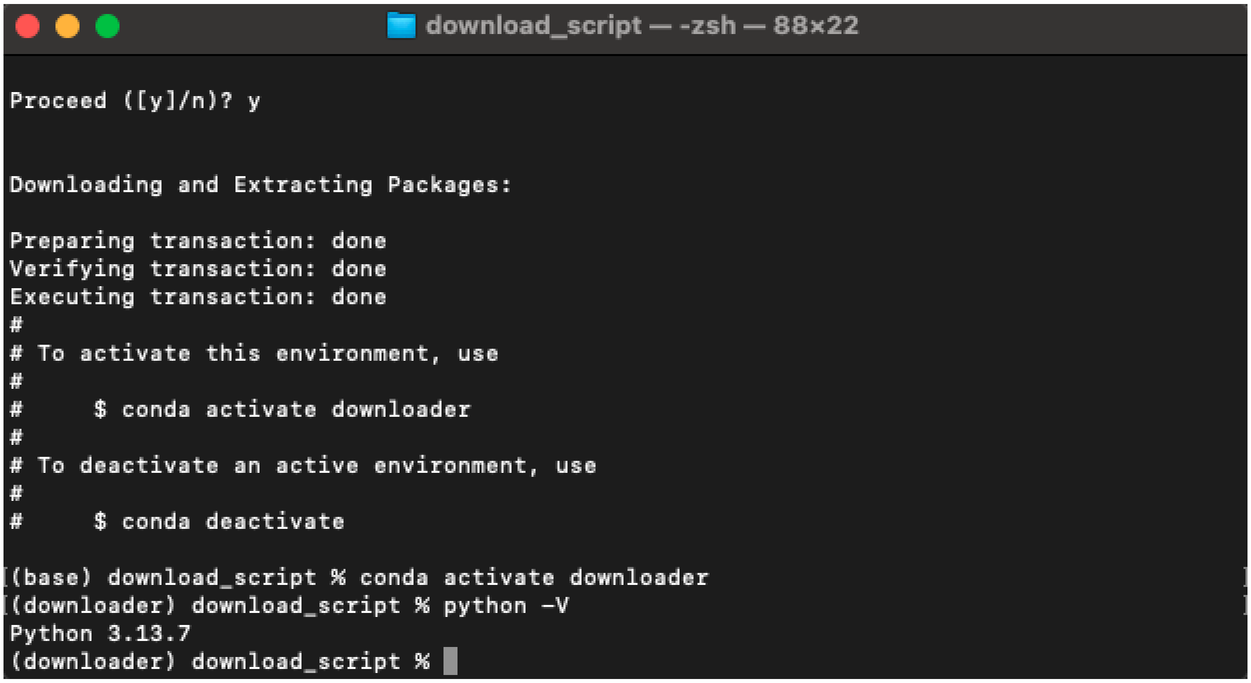
```
  xz                  pkgs/main/osx-arm64::xz-5.6.4-h80987f9_1
  zlib                pkgs/main/osx-arm64::zlib-1.3.1-h5f15de7_0


Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate downloader
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) download_script %
```

Figure 24 conda environment installation

When the command is executed, the creation of the environment will begin. In the middle of the installation of the environment, it will ask you if you would like to proceed. Type y to proceed as shown in Figure 24. After the creation of the environment, the prompt will show you how you can activate the environment. In the case above, the command is:

conda activate downloader

Run the command to activate the environment as shown in Figure 24.



```
Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate downloader
#
# To deactivate an active environment, use
#
#     $ conda deactivate

[(base) download_script % conda activate downloader
[(downloader) download_script % python -V
Python 3.13.7
(downloader) download_script % 
```

Figure 25 Checking python version on conda environment

After running the command to activate the environment, run the command:

python –V

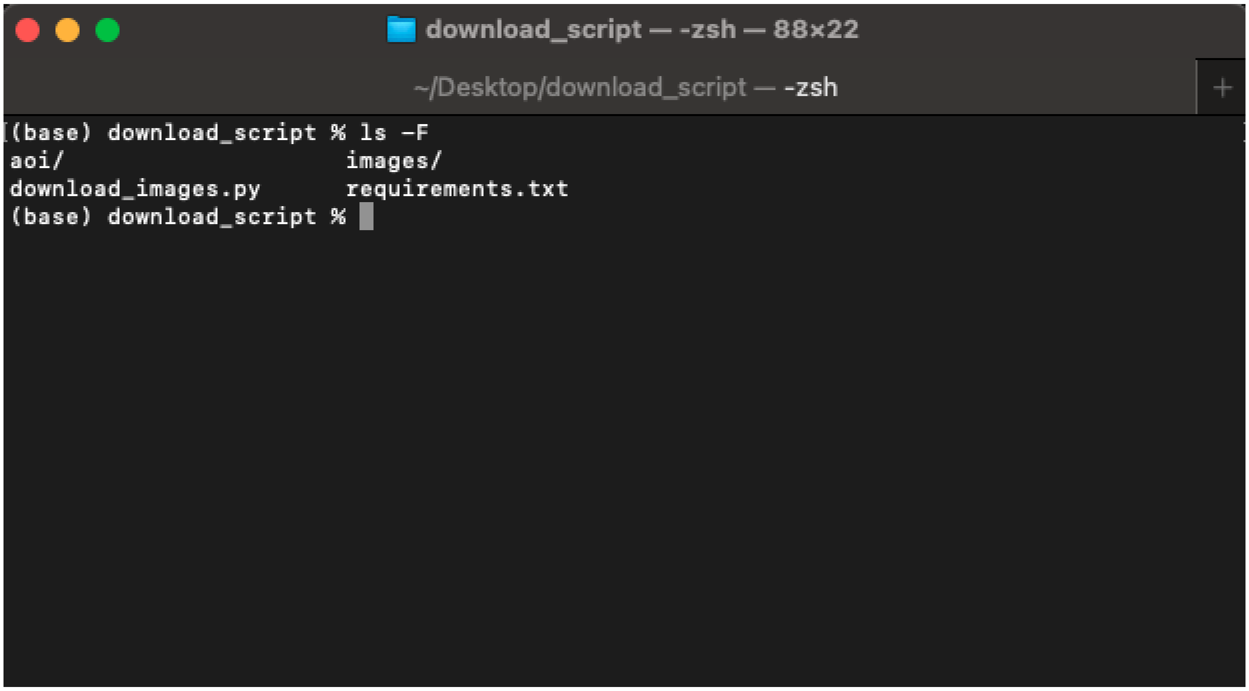
Refer to Figure 25. It should show you the version of python currently activated. In the case above, the python version is 3.13.7 that is the same version that we used in the environment creation. That indicates that the creation of the environment is successful.

5. Running the script for downloading images from the Copernicus Data Space programmatically

This section will show you how to run the script for downloading Copernicus data.

5.1 Installation of the python library dependencies

Open the command prompt or terminal and go to the download_script directory



download_script — -zsh — 88×22
~/Desktop/download_script — -zsh

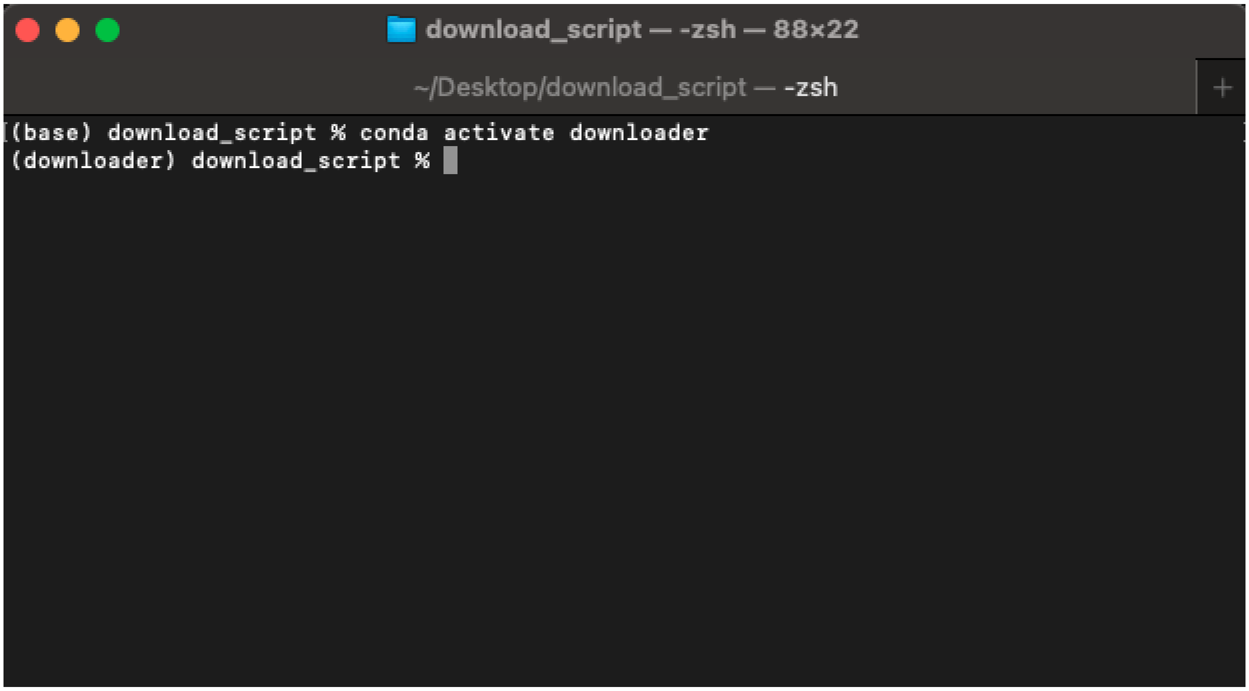
```
(base) download_script % ls -F
aoi/                    images/
download_images.py      requirements.txt
(base) download_script % 
```

Figure 26 checking of script directory files

The directory should have the requirements.txt file. Refer to Figure 26. Activate the conda environment using the following command:

conda activate {conda_env}

Refer to Figure 27. In the case below, the conda_env is downloader

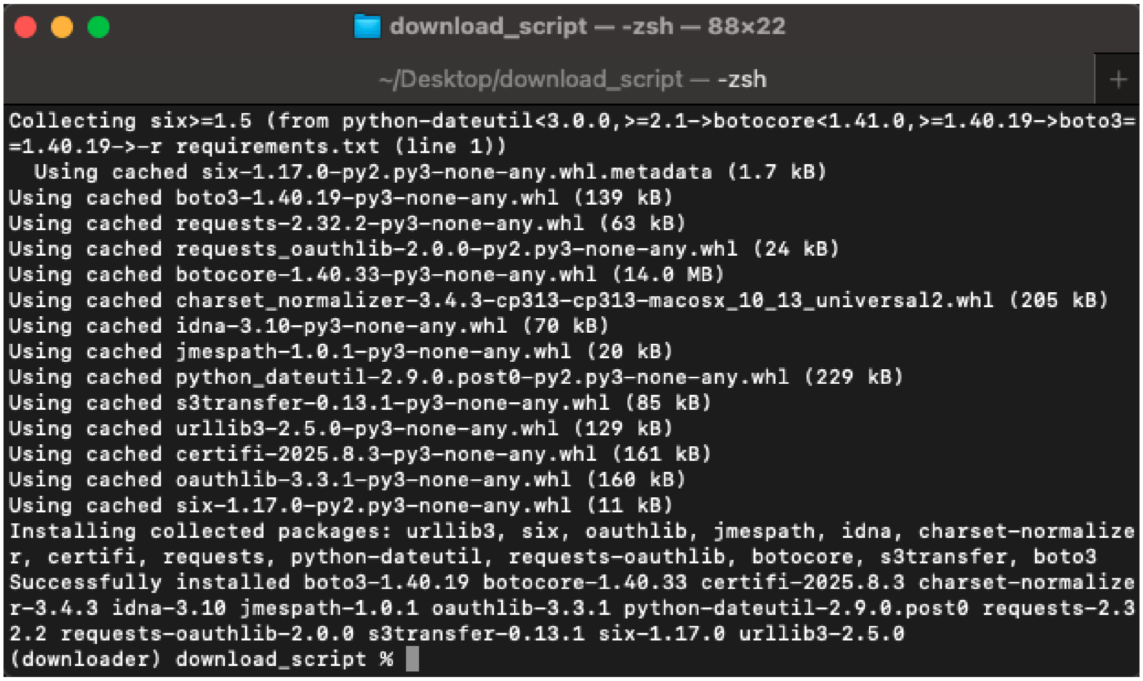

download_script — -zsh — 88×22
~/Desktop/download_script — -zsh

```
(base) download_script % conda activate downloader
(downloader) download_script % 
```

Figure 27 Running command to activate conda environment `downloader`

To install the library dependencies in the file, run the following command:

pip install -r requirements.txt


download_script — -zsh — 88×22
~/Desktop/download_script — -zsh

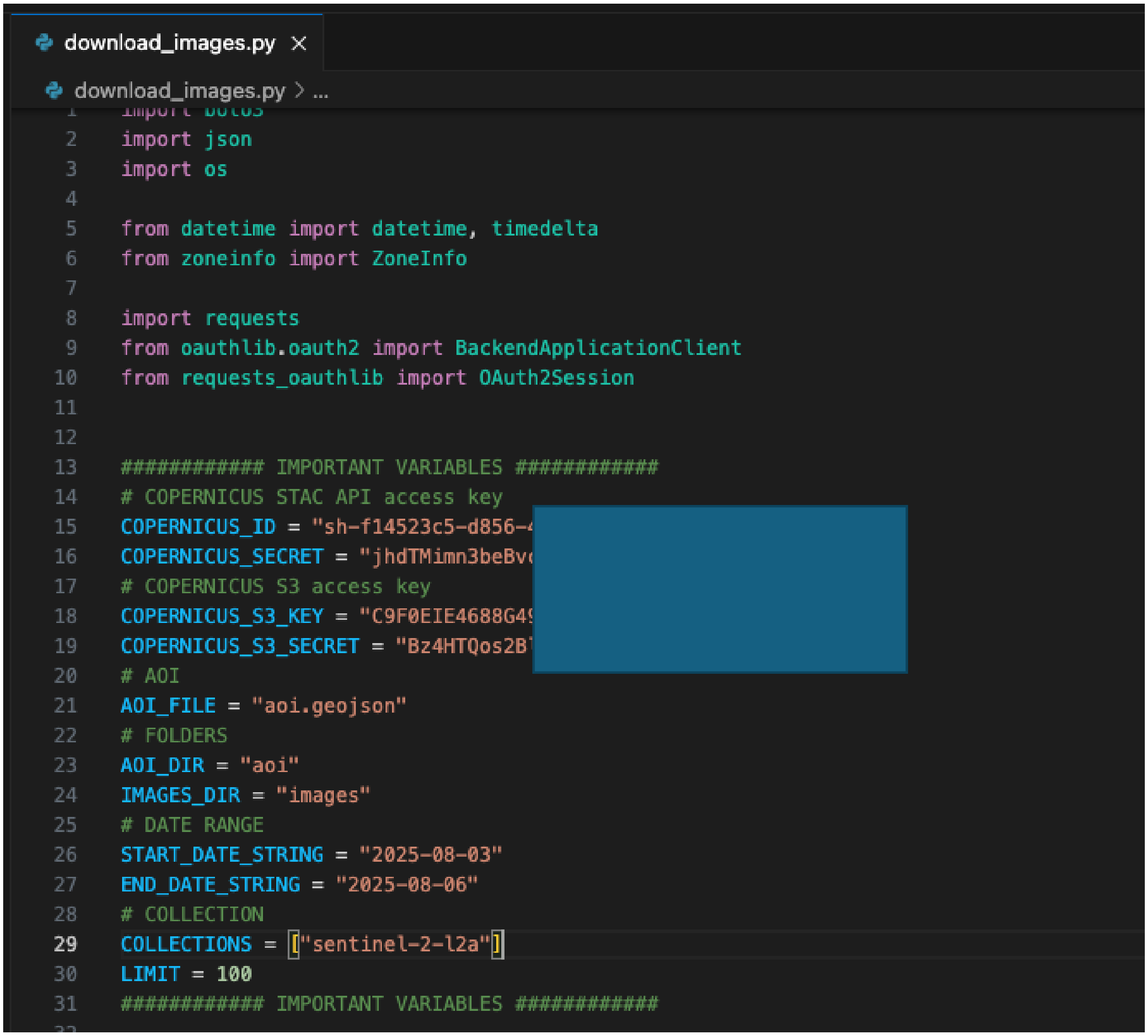
```
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore<1.41.0,>=1.40.19->boto3=
=1.40.19->-r requirements.txt (line 1))
  Using cached six-1.17.0-py2.py3-none-any.whl.metadata (1.7 kB)
Using cached boto3-1.40.19-py3-none-any.whl (139 kB)
Using cached requests-2.32.2-py3-none-any.whl (63 kB)
Using cached requests_oauthlib-2.0.0-py2.py3-none-any.whl (24 kB)
Using cached botocore-1.40.33-py3-none-any.whl (14.0 MB)
Using cached charset_normalizer-3.4.3-cp313-cp313-macosx_10_13_universal2.whl (205 kB)
Using cached idna-3.10-py3-none-any.whl (70 kB)
Using cached jmespath-1.0.1-py3-none-any.whl (20 kB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Using cached s3transfer-0.13.1-py3-none-any.whl (85 kB)
Using cached urllib3-2.5.0-py3-none-any.whl (129 kB)
Using cached certifi-2025.8.3-py3-none-any.whl (161 kB)
Using cached oauthlib-3.3.1-py3-none-any.whl (160 kB)
Using cached six-1.17.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: urllib3, six, oauthlib, jmespath, idna, charset-normalize
r, certifi, requests, python-dateutil, requests-oauthlib, botocore, s3transfer, boto3
Successfully installed boto3-1.40.19 botocore-1.40.33 certifi-2025.8.3 charset-normalize
r-3.4.3 idna-3.10 jmespath-1.0.1 oauthlib-3.3.1 python-dateutil-2.9.0.post0 requests-2.3
2.2 requests-oauthlib-2.0.0 s3transfer-0.13.1 six-1.17.0 urllib3-2.5.0
(downloader) download_script % 
```

Figure 28 Running command to install python dependencies in requirements.txt file

Refer to Figure 28. This will install the python dependencies

5.2 Run the download_images script

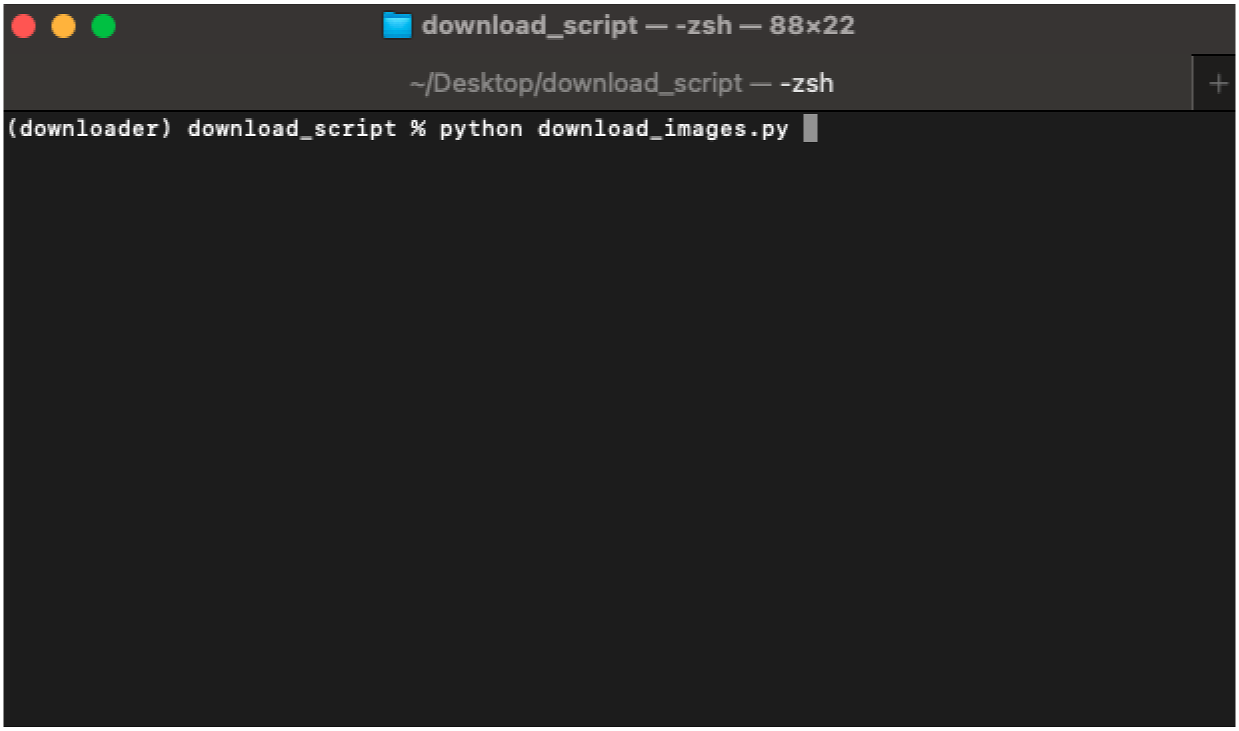
Open the download_script.py file in your preferred text editor. In the **IMPORTANT VARIABLES** section, provide the needed access keys and parameters needed. Important variables are shown in Figure 29



```
import boto3
import json
import os

from datetime import datetime, timedelta
from zoneinfo import ZoneInfo

import requests
from oauthlib.oauth2 import BackendApplicationClient
from requests_oauthlib import OAuth2Session


############ IMPORTANT VARIABLES ############
# COPERNICUS STAC API access key
COPERNICUS_ID = "sh-f14523c5-d856-[redacted]
COPERNICUS_SECRET = "jhdTMimn3beBv
# COPERNICUS S3 access key
COPERNICUS_S3_KEY = "C9F0EIE4688G4
COPERNICUS_S3_SECRET = "Bz4HTQos2B
# AOI
AOI_FILE = "aoi.geojson"
# FOLDERS
AOI_DIR = "aoi"
IMAGES_DIR = "images"
# DATE RANGE
START_DATE_STRING = "2025-08-03"
END_DATE_STRING = "2025-08-06"
# COLLECTION
COLLECTIONS = ["sentinel-2-l2a"]
LIMIT = 100
############ IMPORTANT VARIABLES ############
```

Figure 29 Adding the Important Variables in the download_images script

For this particular query, the request is for SENTINEL 2 – L2A images over the aoi of Taal Volcano from August 03, 2025, to August 06, 2025.


download_script — -zsh — 88×22
~/Desktop/download_script — -zsh

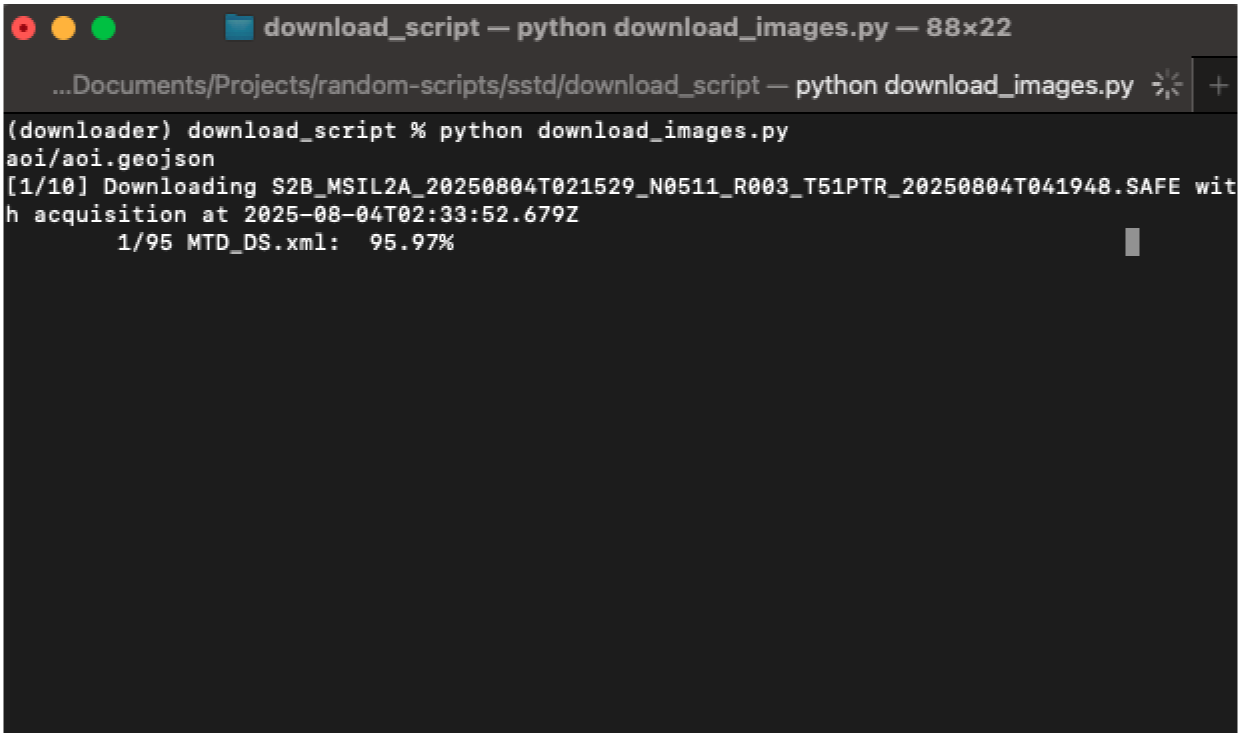
```
(downloader) download_script % python download_images.py
```

Figure 30 Running of download_images script

Run the script by running the command:

python download_images.py

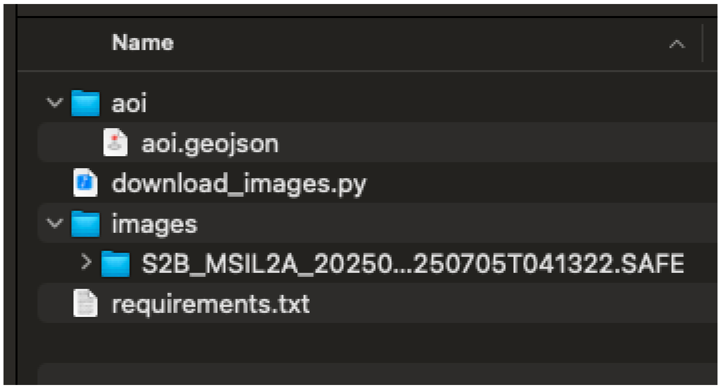
Refer to Figure 30.


download_script — python download_images.py — 88×22
...Documents/Projects/random-scripts/sstd/download_script — python download_images.py

```
(downloader) download_script % python download_images.py
aoi/aoi.geojson
[1/10] Downloading S2B_MSIL2A_20250804T021529_N0511_R003_T51PTR_20250804T041948.SAFE wit
h acquisition at 2025-08-04T02:33:52.679Z
        1/95 MTD_DS.xml:   95.97%
```

Figure 31 Progress of downloading of images

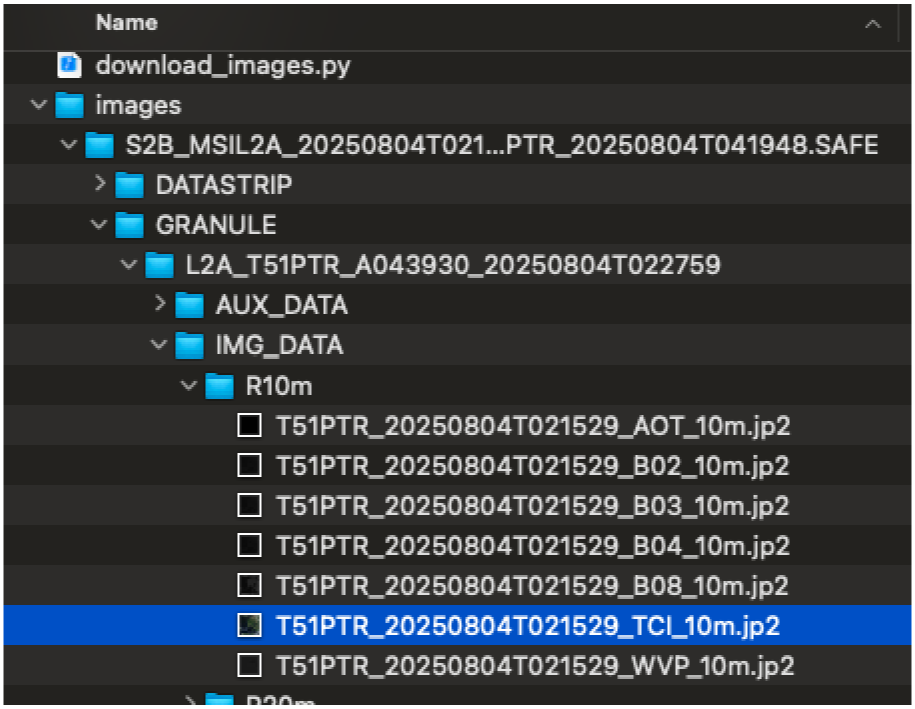
The script will detect the number of scenes that fit the parameters and download the files of each scene into the images folder. It will run the download for all the scenes detected. Each of the files of the scene of the satellite image will be downloaded by the script as shown in Figure 31.



```
Name
aoi
    aoi.geojson
download_images.py
images
    S2B_MSIL2A_20250...250705T041322.SAFE
requirements.txt
```

Figure 32 SENTINEL 2 satellite image downloaded into the image directory

The images downloaded by the scene will be downloaded into the **images** folder as shown in Figure 32.

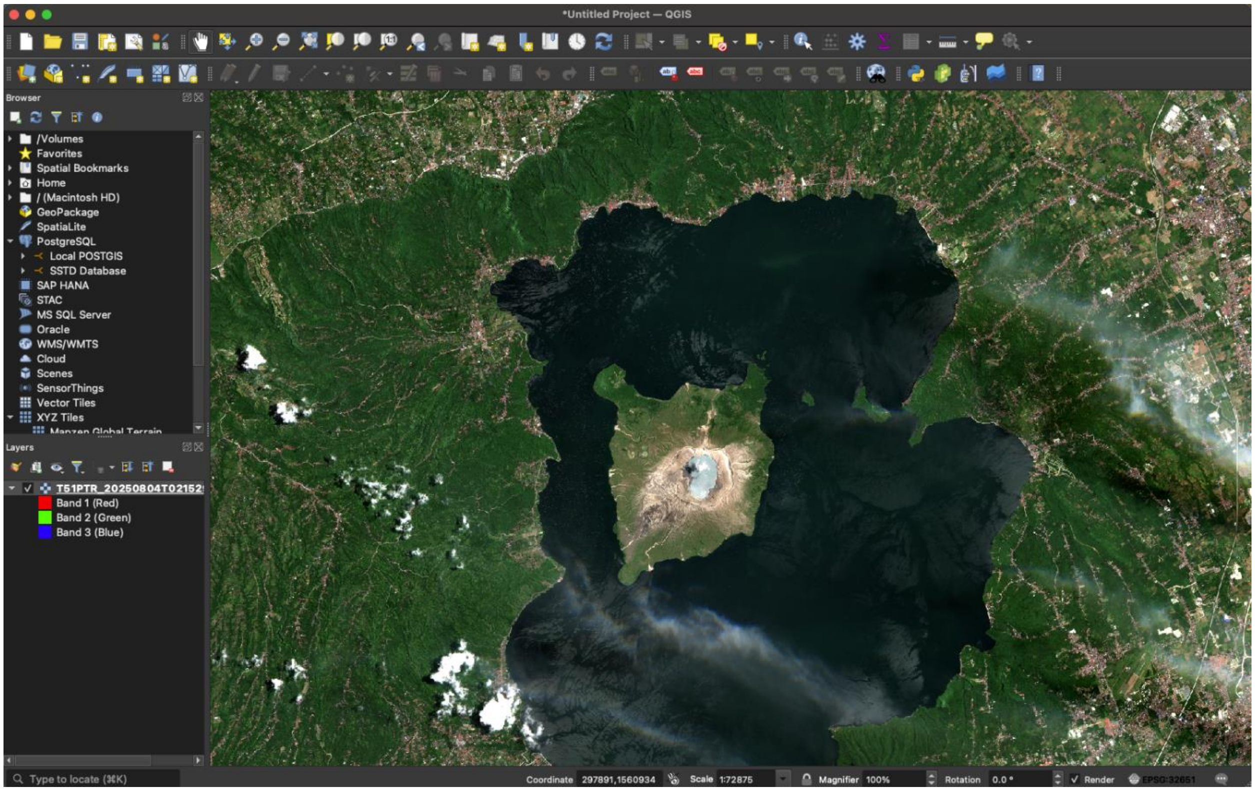


```
Name
download_images.py
images
  S2B_MSIL2A_20250804T021...PTR_20250804T041948.SAFE
    DATASTRIP
    GRANULE
      L2A_T51PTR_A043930_20250804T022759
        AUX_DATA
        IMG_DATA
          R10m
            T51PTR_20250804T021529_AOT_10m.jp2
            T51PTR_20250804T021529_B02_10m.jp2
            T51PTR_20250804T021529_B03_10m.jp2
            T51PTR_20250804T021529_B04_10m.jp2
            T51PTR_20250804T021529_B08_10m.jp2
            T51PTR_20250804T021529_TCI_10m.jp2
            T51PTR_20250804T021529_WVP_10m.jp2
          R20m
```

Figure 33 Directory path for the True color image of the SENTINEL 2 satellite image

The True Color Image of the SENTINEL 2 satellite image scene downloaded is stored in the R10m folder as shown in Figure 33.

You can use GIS software like QGIS to view the image. In the above example we will view the True Color image of the scene. For instructions on how to use QGIS please refer to the beginner guide.



☑ T51PTR_20250804T02152

🟥 Band 1 (Red)
🟩 Band 2 (Green)
🟦 Band 3 (Blue)

Coordinate 297891,1560934 Scale 1:72875 Magnifier 100% Rotation 0.0 ° ☑ Render EPSG:32651

Figure 34 SENTINEL 2 True color image loaded in QGIS

The image in Figure 34 shows the True Color image of the SENTINEL 2 satellite with a resolution of 10 meters captured last August 04, 2025.