

Methane Emission Sensing Tools - How To Guide

Nicholas Kinsella
Ulster University

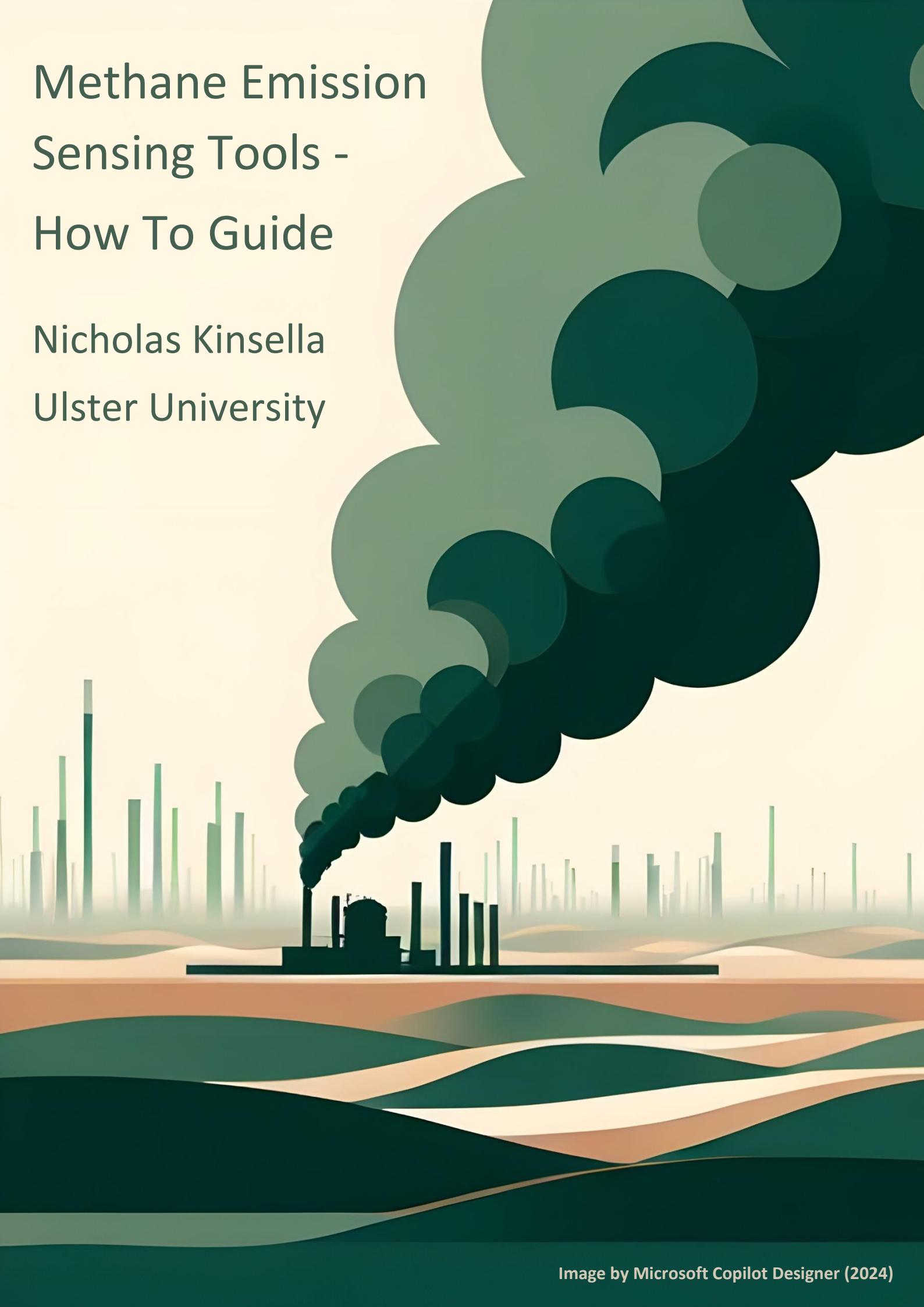


Table of Contents

1. Introduction	3
2. Sentinel-2 Multi-Band-Multi-Pass CH ₄ Mapper (S2MBMP)	5
2.1 Methodology	5
2.2 Data and Dependencies.....	6
2.3 Setup.....	7
2.3.1 Anaconda Navigator	7
2.3.2 Creating a Conda Environment.....	7
2.3.3 Setting up Jupyter Lab	8
2.3.4 OpenEO setup using Anaconda Navigator	10
2.3.5 OpenEO setup using PyPi.....	11
2.3.6 Registering with Copernicus Data Space Ecosystem.....	12
2.3.7 Running the tools in Jupyter-lab.....	13
2.3.8 Authentication with OpenEO	15
2.4 S2MBMP Code	16
2.5 Expected Results and Demo	23
2.6 Troubleshooting.....	25
2.6.1 Remote disconnected error (All tools)	25
2.6.2 Concurrent job error (S5-AGM).....	25
3. Integrated Methane Inversion (IMI)	27
3.1 Methodology	27
3.2 Data and Dependencies.....	28
3.3 Setup.....	29
3.3.1 Creating an AWS account	29
3.3.2 Add Amazon Simple Storage Service (S3) user permissions.....	29
3.3.3 Launching an IMI instance	32
3.3.4 Connecting to instance.....	38
3.3.5 Configuring and running the IMI Preview	41
3.3.5 Viewing the IMI Preview data.....	44
3.3.6 Configuring and running the IMI	47
3.3.7 Accessing the IMI data.....	50
3.3.8 Interpreting the IMI data.....	52
3.3.9 Shutting down the instance.....	54
3.5 Troubleshooting.....	56
4. References	57

1. Introduction

Methane (CH_4), a greenhouse gas with a warming potential 28 times greater than carbon dioxide over a 100-year period, has seen its atmospheric concentration increase over 250% since the industrial revolution. Despite CH_4 's shorter atmospheric lifespan, it still has contributed to at least a quarter of anthropogenic warming since the Industrial Revolution (Pandey et al., 2023; Vigano et al., 2008).

Mismanaged oil and gas fields can produce significant CH_4 emissions (Jackson et al., 2020; Pandey et al., 2023) the 2021 Global Methane Pledge (GMP), was created to reduce anthropogenic emission levels by 30% of 2020 levels by 2030, with 157 countries participating (GMP, 2024; International Energy Agency, 2022; Malley et al., 2023).

In 2022 Algeria made a pledge to reduce its GHG emissions by between 7% and 22% (United Nations Development Programme, 2022) and the European Union has plans to launch a pilot scheme to encourage the Algerian state petrochemical company Sonatrach, to capture CH_4 under the 'You collect, we buy' scheme by the end of 2024 (European Commission, 2023). Despite this, as of 2024, Algeria has yet to join the GMP and currently does not have government led commitments to reduce CH_4 emissions specifically (International Energy Agency, 2024).

Managed by state-owned company, Sonatrach, there are well over a hundred identified oil and gas fields in Algeria, two of which are considered giant (Abada et al., 2018). Situated in northern Algeria, 550 km south of Algiers (fig.1), the Hassi R'Mel gas field covers an area of around 3,500 km^2 and is a key resource for Europe, producing around 45% of Algeria's national gas production (Abada and Bouharkat, 2018; Naus et al., 2023; Rosenthal., 2023; Talamali., 2016). A further 350 km further to the southeast lies the Hassi Messaoud oil field, which covers 1,600 km^2 and produces around 36% of Algeria's national oil production (fig.1) (Kamr Eddine Aissou., 2024; Naus et al., 2023). Despite being an oil field, Hassi Messaoud is by far the worse emitter of CH_4 emissions, with Sentinel 5P detecting 154 super emissions since 2019, compared to just 8 in Hassi R'Mel over the same period, (Kayros, 2024; Naus et al., 2023). This is backed up by the study of Naus et al. (2023) which detected 1 super-emitter event in Hassi R'Mel and 9 in Hassi Messaoud for the year 2020. Another study by Varon et al. (2021) detected 101 plumes between the 9th of October 2019 and 9th of August 2020 at Hassi Messaoud. This implies that current management practices of CH_4 emissions at Hassi Messaoud could be improved.

Satellite missions have in recent years improved their CH_4 measurement capabilities, offering advantages over ground-based detectors (Parker et al., 2011). This guide outlines the use of two tools that can be used to monitor methane emissions.

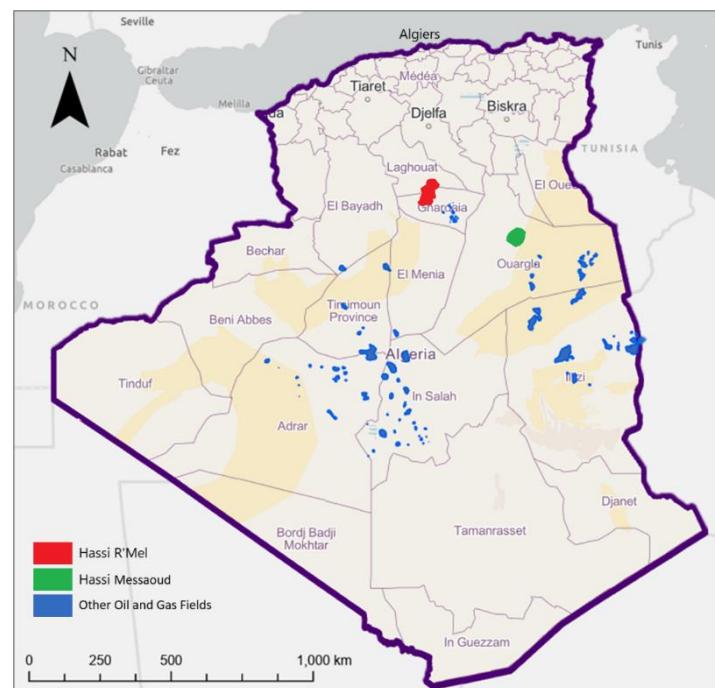


Figure 1: Hassi Messaoud oil field in Green (Lat 31.69, Long 6.03) and Hassi R'Mel gas field in Algeria in red (Lat 32.94, Long 3.27). (Field locations produced by manual survey of satellite images by author.)

- **Sentinel-2 Multi-Band-Multi-Pass CH₄ Mapper (S2MBMP):** This creates a high-resolution map that can show super emitting CH₄ point sources for selected Algerian petrochemical fields for a specific date. This tool can be downloaded or cloned from the following Github repository. ([GITHUB GOES HERE](#))
- **The Integrated Methane Inversion (IMI):** An Amazon Web Services based tool for measuring methane emissions using the Sentinel-5P methane product and the GEOS-Chem model of atmospheric chemistry (Varon et al., 2022)

This guide outlines their installation and use.

2. Sentinel-2 Multi-Band-Multi-Pass CH₄ Mapper (S2MBMP)

2.1 Methodology

To help users locate emission sources in a field, Sentinel 2's MSI instrument with a 20m² spatial resolution and return frequency of 3-5 days was employed.

Varon et al. (2021) demonstrated that methane columns from point sources can be measured by exploiting the SWIR-1 and SWIR-2 bands of Sentinel-2. A Python code has been developed accessing Sentinel 2's L1C data using the Copernicus OpenEO API. Scenes without obscuring cloud or smoke will be chosen as the non-emission scene and then these will be tested against other scenes with a maximum of 5% cloud cover. No emission and active emission scenes were each processed using the multi-band-single-pass equation as outlined in Varon et al. (2021):

$$\text{MBSP} = \mathbf{B11} - c\mathbf{B12}$$

Where: **B12** is the Sentinel-2 SWIR-2 band, **B11** is the Sentinel-2 SWIR-1 band and **c** is calculated by least squares fitting B12 against B11.

Once the MBSP raster had been calculated, the following equation was then used to calculate the multi-band-multi-pass raster:

$$\text{MBMP} = \text{ActiveMBSP} - \text{NoMBSP}$$

Where **ActiveMBSP** is the multiband single pass for the active emission scene and **NoMBSP** is the multiband single pass for the no emission scene.

Figure 2 provides an overview of the main steps in processing.

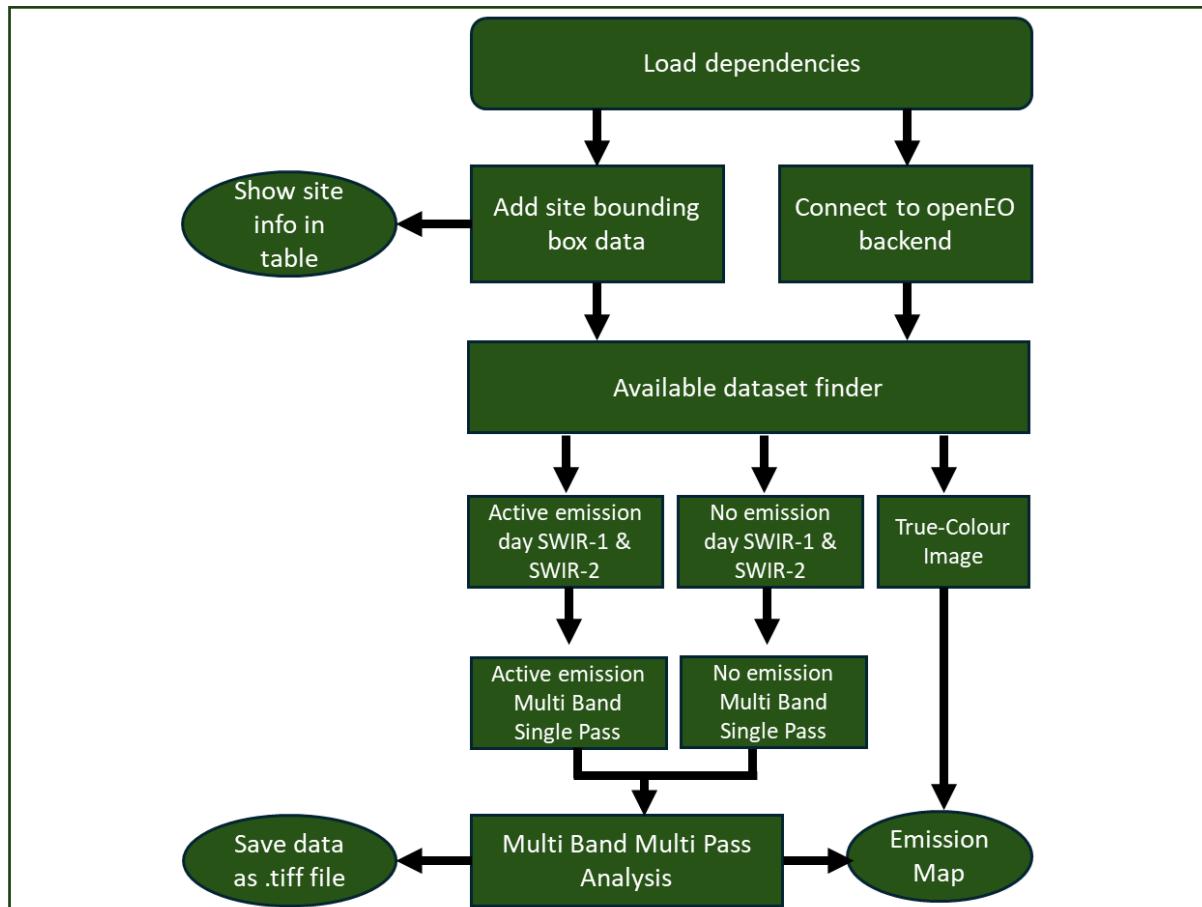


Figure 2: Flowchart of processes used for Sentinel 2 CH₄ Map

2.2 Data and Dependencies

The datasets used by these tools are outlined in table 1.

Table 1: Datasets used by tools

Description	Data	Source	For use in tool(s)
Oil and Gas Field Bounding	Long/Lat .csv file.	Manual surveying of arial images of Algeria by author	S2-MBMP
Sentinel 2 (MSI) bands 2, 3, 4, 11 and 12	Raster 10m2 for 2, 3 and 4 and 20m2 for 11 and 12	Copernicus Dataspace – OpenEO.	S2-MBMP

The dependencies contained in the environment.yml file are outlined in table 2.

Table 2: dependencies required to use the tools.

Name	Version	Description
Python	3.12.2	Programming language to run the code
Jupyterlab	4.1.4	Computing environment for Python
Folium	0.16.0	Generates interactive maps with Leaflet.js.
Pandas	2.2.1	Data manipulation and analysis.
Geopandas	0.14.3	Geospatial data manipulation and analysis.
Matplotlib	3.8.3	Creates map visualizations
Rasterio	1.3.9	Reads and edits raster datasets.
Numpy	1.26.4	Performs numerical computations on arrays
Requests	2.31.0	Used fetch data from web services or APIs.
Rasterstats	0.19.0	Linear regression for brightness correction
OpenEO	0.28.0	API for Earth observation data processing
Shapely	2.0.3	Used to manage points, lines and polygon data

2.3 Setup

Anaconda Navigator, containing the Python programming language and ‘Conda’, a package manager for creating shareable environments that tools like this can run on will be installed. This includes Jupyter Lab which will be used to run this tool’s code.

2.3.1 Anaconda Navigator

To download Anaconda, navigate to <https://docs.anaconda.com/anaconda/install/> and follow the instructions of your operating system.

2.3.2 Creating a Conda Environment

In the Anaconda Navigator side bar, click the ‘Environments’ tab. You will see the installed packages (fig.3).

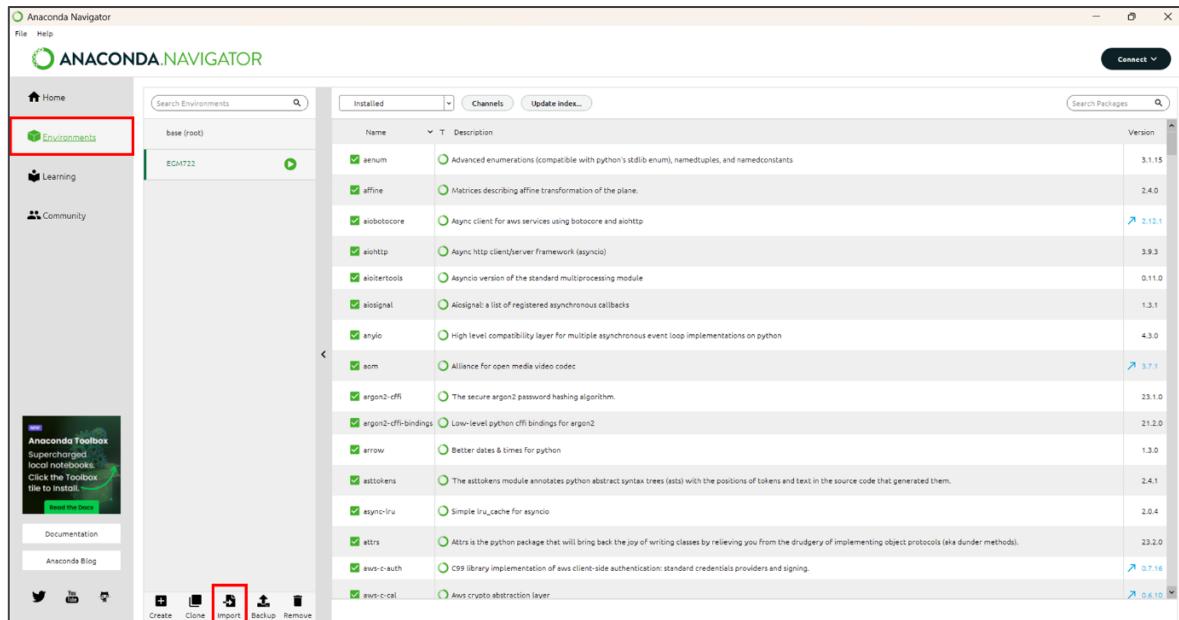


Figure 3: Environments tab of Anaconda Navigator with environments tab and import button highlighted in red.

Next click on the imports tab (fig.3) and select the file ‘environment.yml’ contained in the .zip file of the tool’s download, choosing an appropriate name for the environment (fig.4).

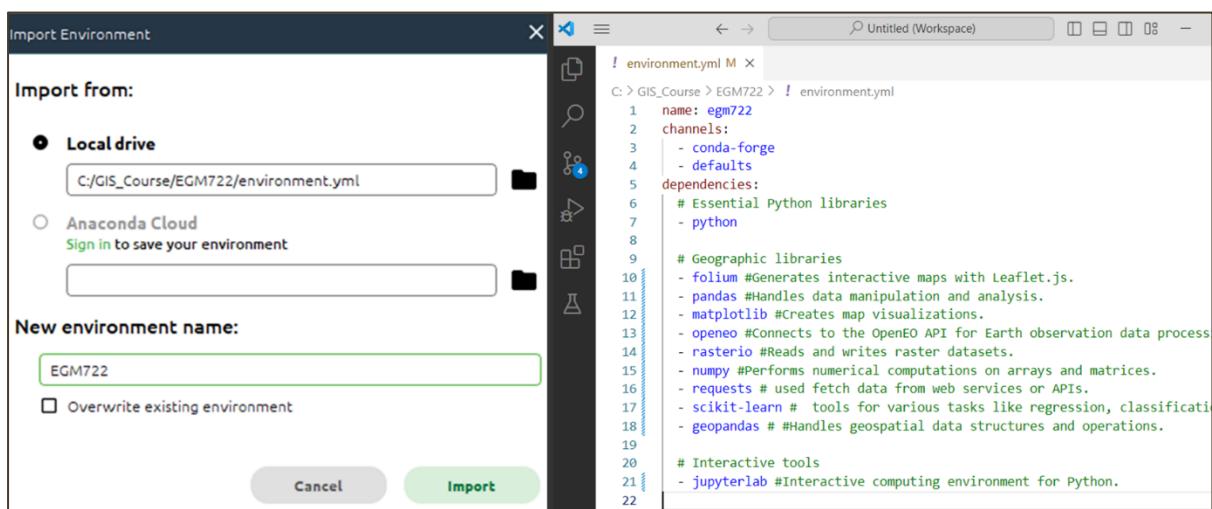


Figure 4: The import config box (left) and the contents of ‘environment.yml’ (right).

Click Import. The process of installing all the packages may take several minutes. Once finished you will be returned to the environments tab (fig.3)

Next click on the ‘Home’ tab in Anaconda Navigator’s sidebar (fig.5).

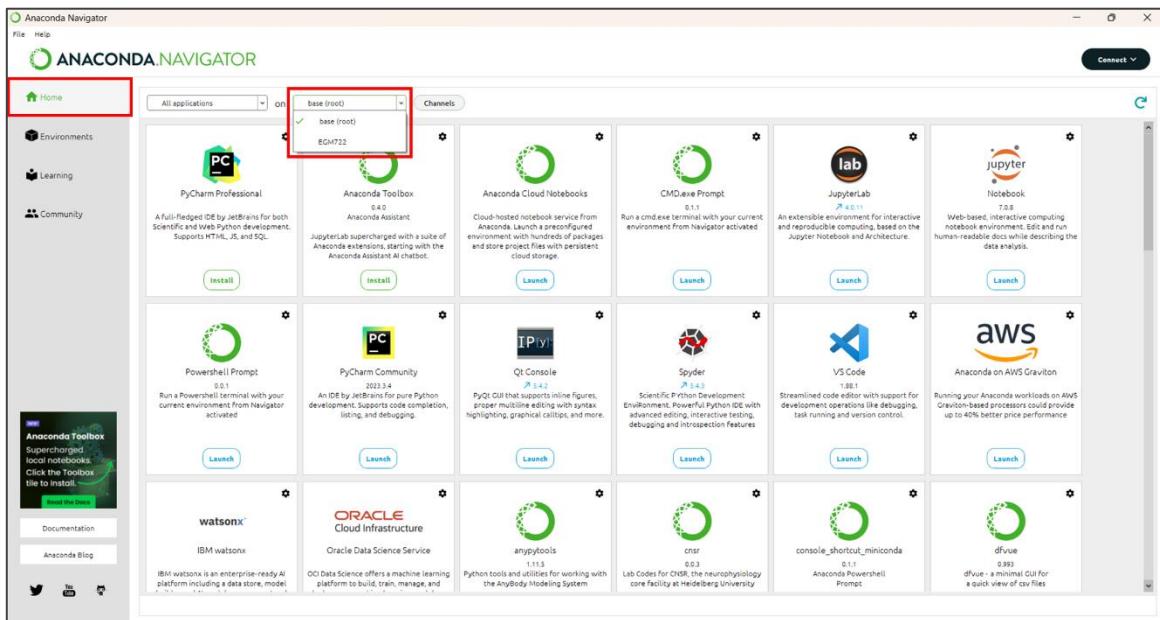


Figure 5: Anaconda Navigator with home tab and environment switching dropdown in red.

The dropdown highlighted in figure 3 should display two options, ‘base (root)’ and the name of your new environment (in figure 3 this is ‘EGM722'). **Ensure your environment name is always selected here or the dependencies installed earlier will not be available to it.**

2.3.3 Setting up Jupyter Lab

A configuration file (‘.config’) needs to be created to change the settings used by Jupyter Lab by default. Launch the CMD.exe prompt (fig.6)

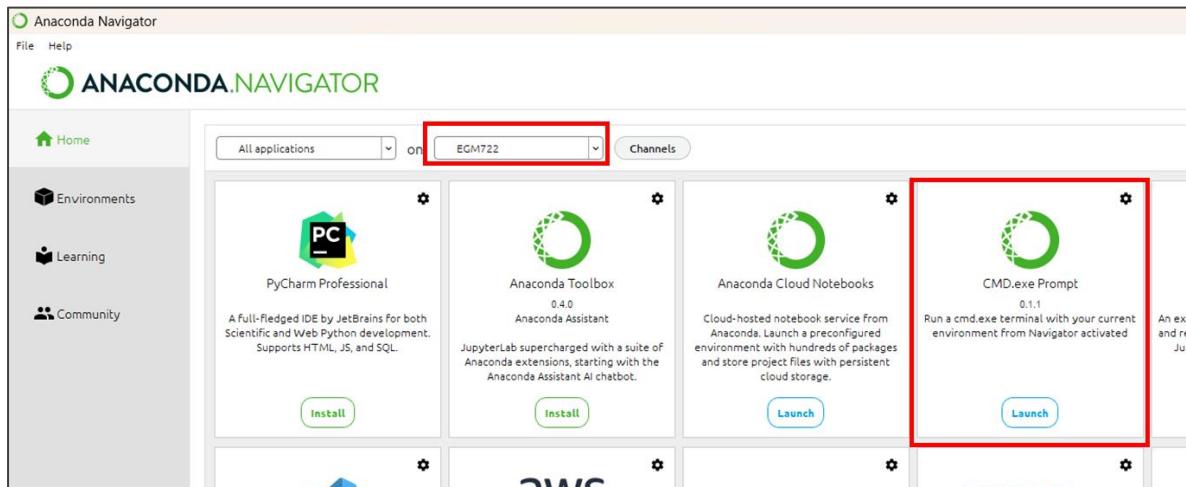


Figure 6: Highlighted locations of selected environment and CMD.exe Prompt

In the command prompt, enter the command:

```
jupyter lab --generate-config
```

This will create a new folder in your user directory called ‘jupyter’ containing a python script jupyter_lab_config.py. On Windows this is usually ‘C:\Users\<your_username>’.

Jupyter lab will by default open in your user directory. Due to security restrictions, it is not possible to navigate to the parent directory of the launch location. So if Jupyter launches in 'C:\Users\RockyBalboa', it is not possible to move to 'C:\Users' or, 'C:\EGM722'. If the directory you are keeping your data in is outside your user directory, you will need to change the default opening folder to your data directory.

This location is also where you should store the following files and folder:

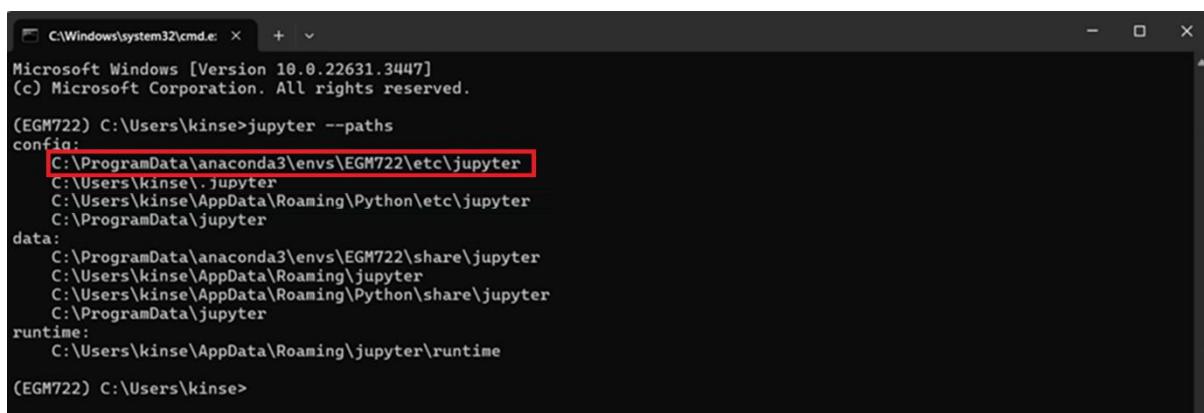
- Sentinel_5P_Atmospheric_Gas_Time_Series.ipynb
- Sentinel_5P_Atmospheric_Gas_Map.ipynb
- Sentinel_2_CH4_Multi-Band-Single-Pass.ipynb
- The folder "Data"

If your data directory is in your user directory, you should be able to click and navigate there using the interface of Jupyter Lab. If that is not the case, you will need to do the following:

Open an Anaconda Navigator CMD.exe prompt and type the following command:

```
jupyter --paths
```

This will show something like figure 7 although your file paths will be unique to you.



```
C:\Windows\system32\cmd.exe > + ▾
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

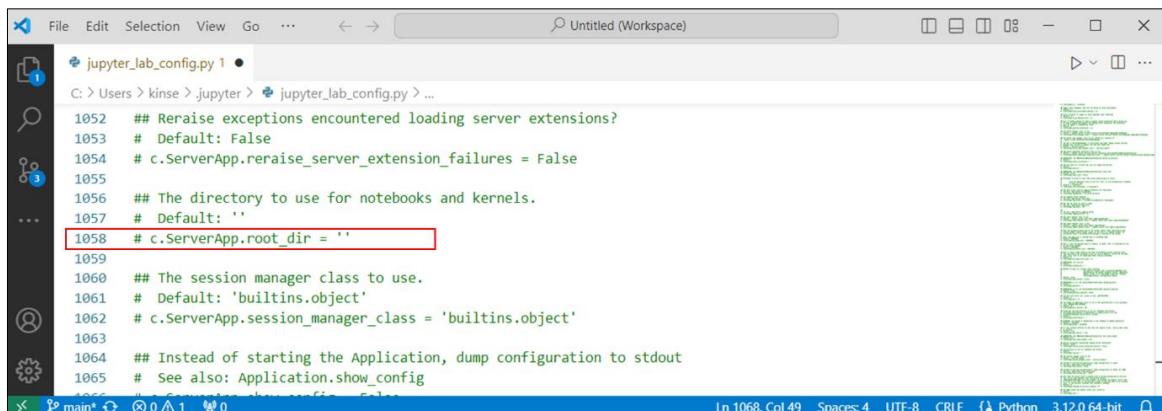
(EGM722) C:\Users\kinse>jupyter --paths
config:
C:\ProgramData\anaconda3\envs\EGM722\etc\jupyter
C:\Users\kinse\.jupyter
C:\Users\kinse\AppData\Roaming\Python\etc\jupyter
C:\ProgramData\jupyter
data:
C:\ProgramData\anaconda3\envs\EGM722\share\jupyter
C:\Users\kinse\AppData\Roaming\jupyter
C:\Users\kinse\AppData\Roaming\Python\share\jupyter
C:\ProgramData\jupyter
runtime:
C:\Users\kinse\AppData\Roaming\jupyter\runtime

(EGM722) C:\Users\kinse>
```

Figure 7: results of 'jupyter –paths' command showing path used by environment highlighted in red.

The 'jupyter_lab_config.py' file mentioned earlier needs to be copy pasted into that folder.

Once 'jupyter_lab_config.py' file has been moved, open it in Notepad++, Visual Studio Code or if you don't have those, Notepad. Using the shortcut 'CTRL + F' type in the following line: 'c.ServerApp.root_dir' (without quotes) and you should find the section highlighted in figure 8.



```
jupyter_lab_config.py 1 ●
C: > Users > kinse > jupyter > jupyter_lab_config.py > ...
1052 ## Reraise exceptions encountered loading server extensions?
1053 # Default: False
1054 # c.ServerApp.reraise_server_extension_failures = False
1055
1056 ## The directory to use for notebooks and kernels.
1057 # Default: ''
1058 # c.ServerApp.root_dir = '' 1059
1060 ## The session manager class to use.
1061 # Default: 'builtins.object'
1062 # c.ServerApp.session_manager_class = 'builtins.object'
1063
1064 ## Instead of starting the Application, dump configuration to stdout
1065 # See also: Application.show_config
1066 # c.ServerApp.dump_configuration = False
```

Figure 8: location of 'c.ServerApp.root_dir' in jupyter_lab_config.py

Remove the '#' and space from the start and add the path used by your environment between the quote marks, adding a 'r' beforehand (fig.9).

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, ...
- Toolbar:** Back, Forward, Search (Untitled (Workspace)), Minimize, Maximize, Close.
- Left Sidebar:** Includes icons for file operations (New, Open, Save, etc.), search, and help.
- Code Area:** The main workspace displays the following Python code from `jupyter_lab_config.py`:

```
1052     ## Reraise exceptions encountered loading server extensions?
1053     # Default: False
1054     # c.ServerApp.reraise_server_extension_failures = False
1055
1056     ## The directory to use for notebooks and kernels.
1057     # Default: ''
1058     c.ServerApp.root_dir = r'C:\GIS_Course\EGM722'
1059
1060     ## The session manager class to use.
1061     # Default: 'builtins.object'
1062     # c.ServerApp.session_manager_class = 'builtins.object'
1063
1064     ## Instead of starting the Application, dump configuration to stdout
1065     # See also: Application.show_config
1066     # c.ServerApp.dump_config = False
```
- Bottom Status Bar:** Shows the current file as `main*`, line count as `Ln 1068, Col 41`, and encoding as `UTF-8 CRLF Python 3.12.0 64-bit`.

Figure 9: path to data directory added to jupyter_lab_config.py

Save and close this file and return to the Anaconda Navigator ‘Home’ tab. Launch Jupyter Lab and if you have followed the steps correctly, you should see that your data directory is automatically displayed (fig.10).

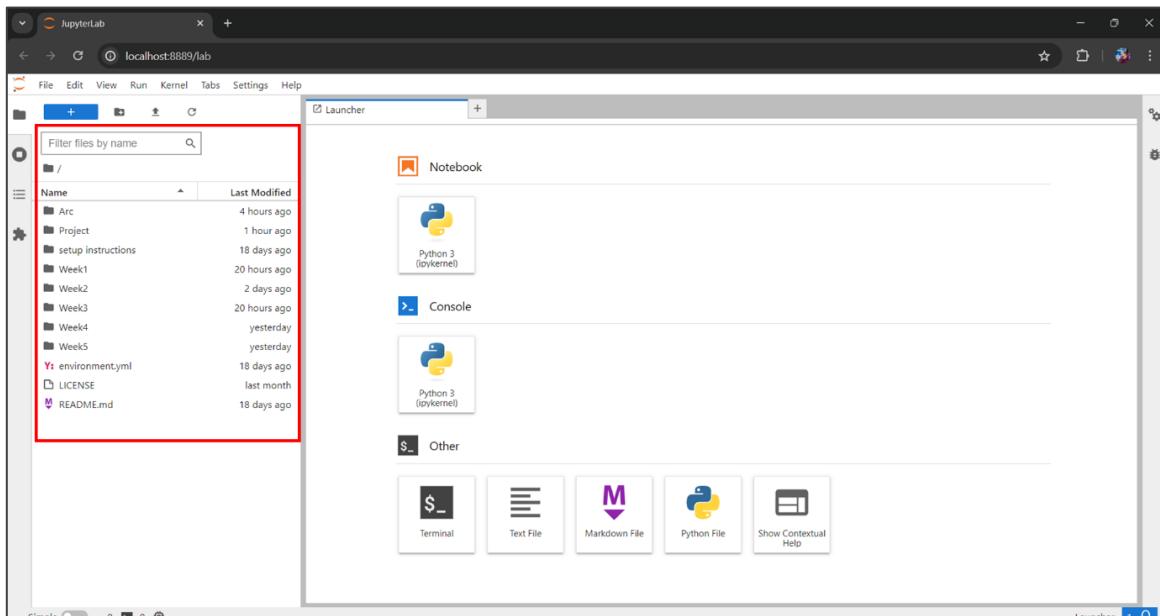


Figure 10: Jupyter Lab showing by default the data directory

2.3.4 OpenEO setup using Anaconda Navigator

OpenEO is an open-source API that allows access to the earth observation satellite missions run by the Copernicus program. These include the satellites used by this tool.

First search in the Anaconda Navigator environments tab for ‘openeo’. Make sure that ‘Not installed’ is selected (fig.11). If the package appears here, click its tick box and select apply. If you can’t see it here, please go to section 2.5.

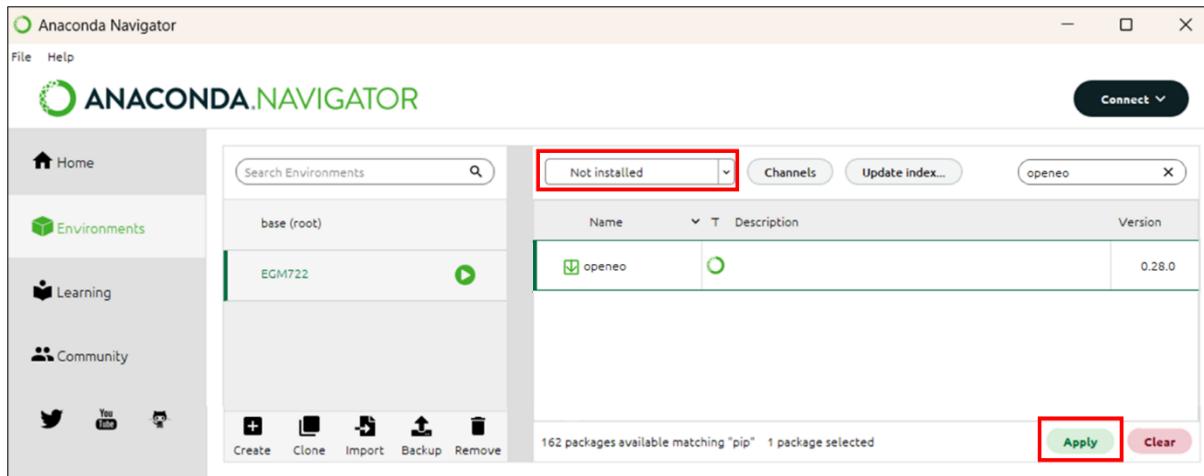


Figure 11: installing OpenEO from Anaconda Navigator.

Next you will be presented with the following screen (fig.12). Once this has finished processing the request. Simply click ‘apply’ to begin the installation.

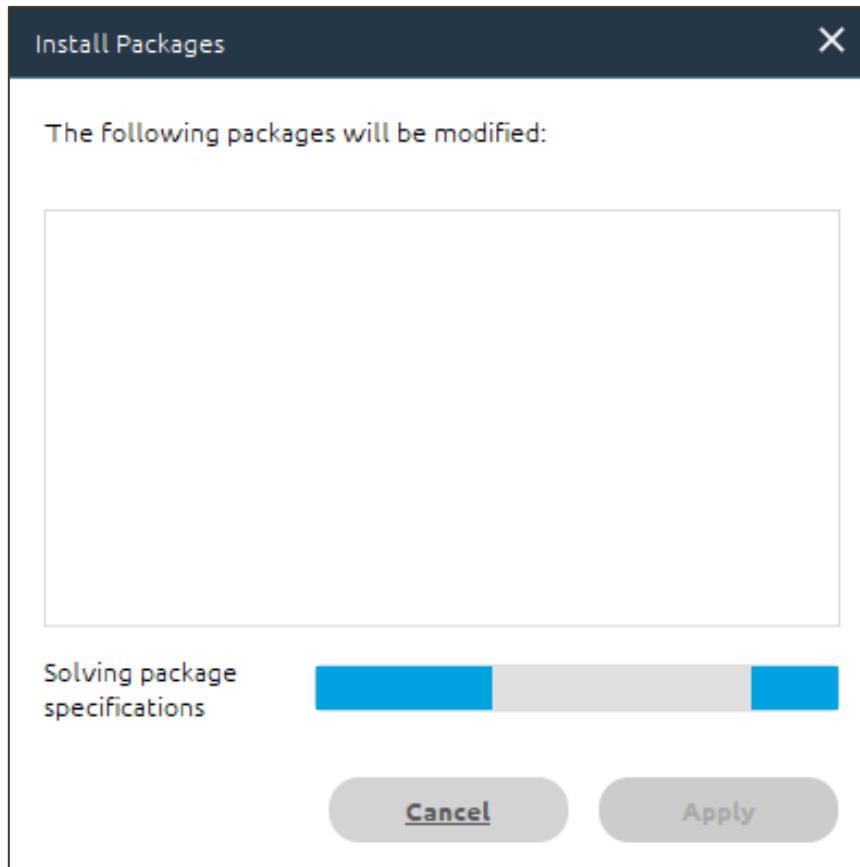


Figure 3: Anaconda Navigator package installer loading screen.

2.3.5 OpenEO setup using PyPi

If Anaconda Navigator cannot find OpenEO you can use PyPi, the official third-party software library for Python. Search for ‘pip’, selecting the appropriate tick-box and then clicking apply (fig.11), then clicking apply once the install packages prompt has finished loading (fig.13).

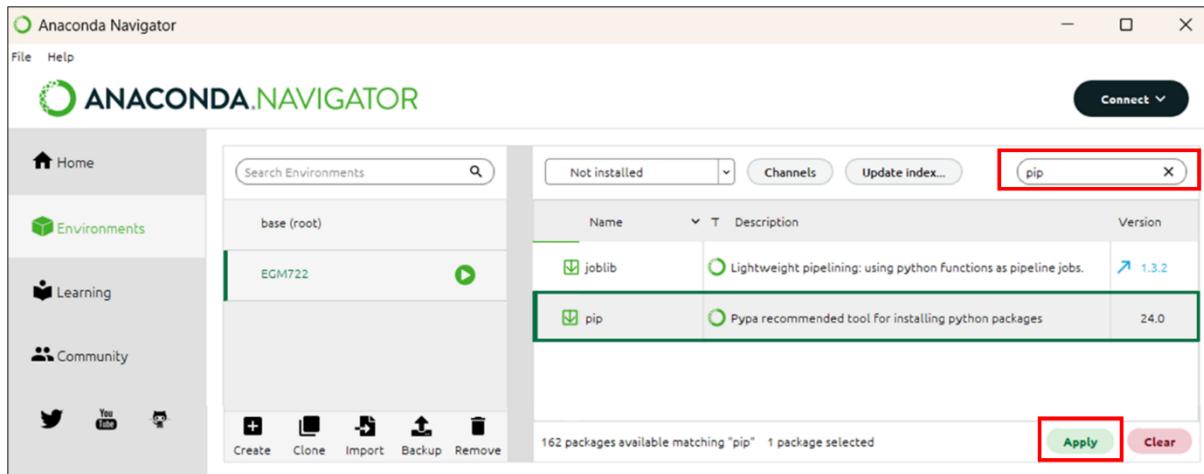


Figure 4: Installing pip via Anaconda Navigator

Open an Anaconda Navigator CMD.exe prompt and type the following command:

```
pip install openeo
```

Once the process has completed, you can close the CMD.exe prompt window.

2.3.6 Registering with Copernicus Data Space Ecosystem.

Accessing OpenEO requires an authentication. To do this, you need to complete a Copernicus Dataspace Registration. Go to <https://dataspace.copernicus.eu/> and click the green login button (fig.14)

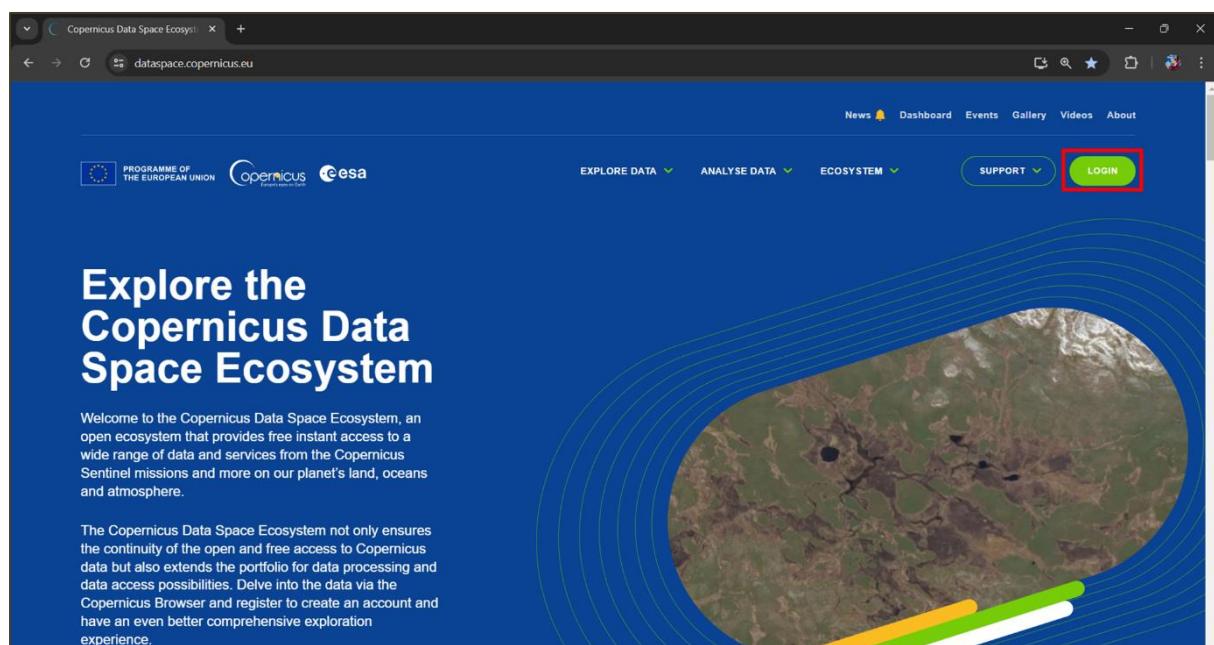


Figure 5: Copernicus Dataspace landing page with login button highlighted in red.

Next click the green 'register' button (fig.15):

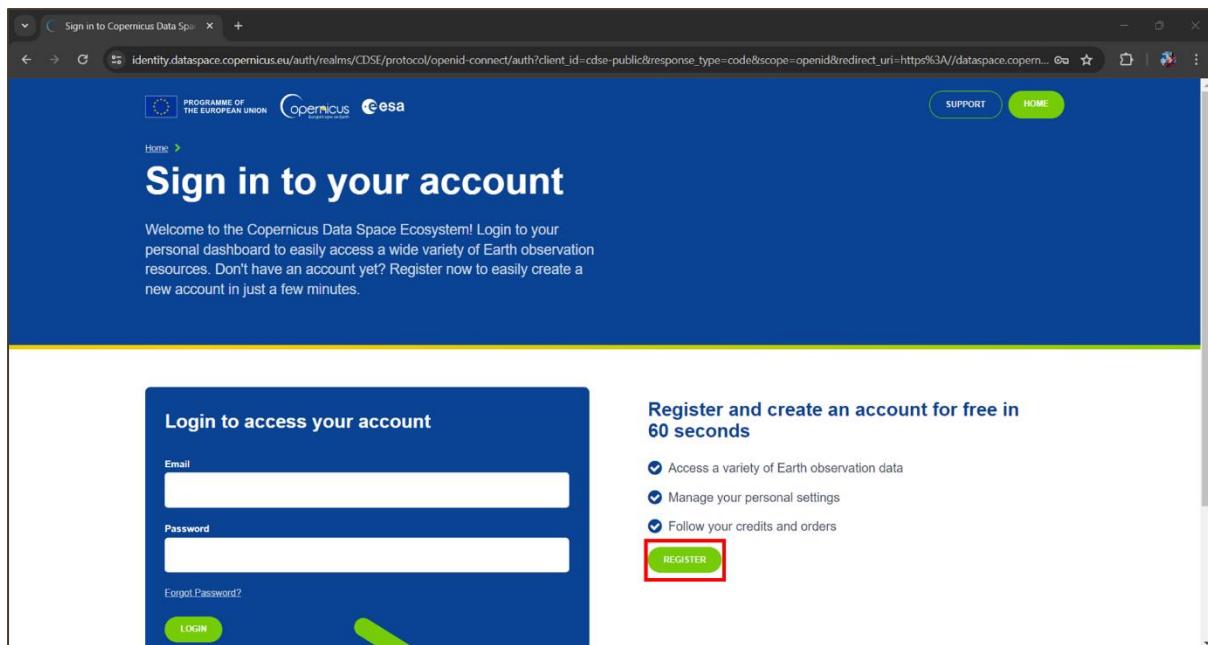


Figure 6: Copernicus Dataspace sign in page.

On the following page, fill out the application form and then at the bottom click the green ‘register’ button (fig.16).

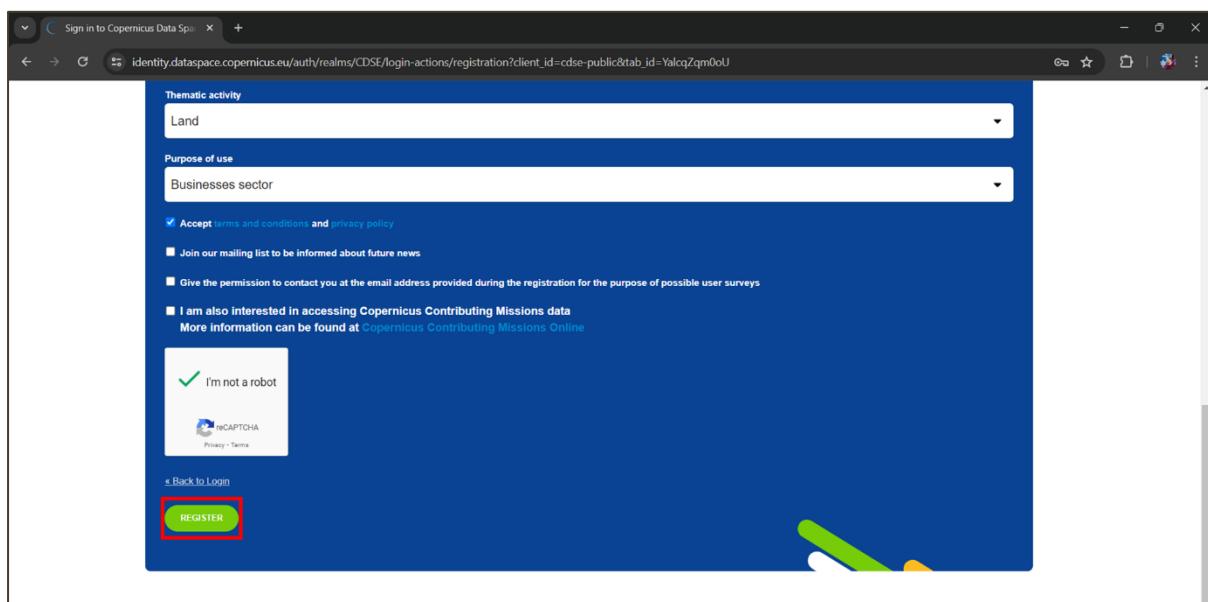


Figure 7: End of Copernicus registration page with register button highlighted in red.

Once registered, you will receive an email asking to verify your address. You can then log-in with your email and chosen password.

For any registration problems, email: help-login@dataspace.copernicus.eu

2.3.7 Running the tools in Jupyter-lab

Now that (almost) everything has been setup you can launch Jupyter Lab in the Anaconda Navigator (fig.17). Remember as always that your project environment (here ‘EGM722’) should be selected and not ‘base (root)’.

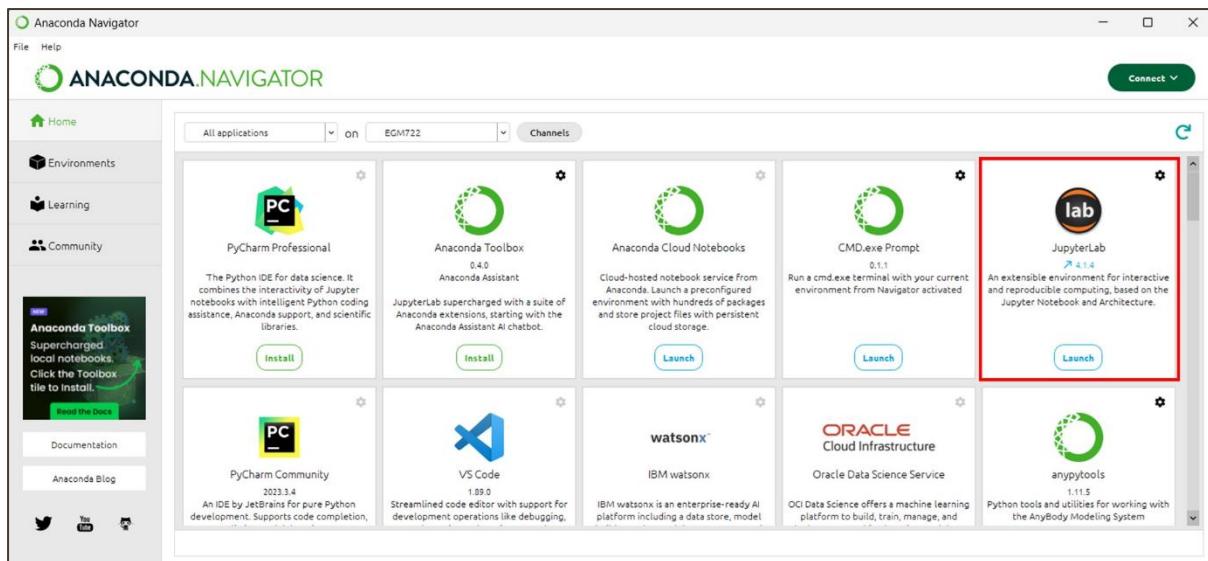


Figure 8: Location of Jupyter Lab in Anaconda Navigator highlighted in red.

Once Jupyter lab opens, you should see the three tools on the left (fig.18).

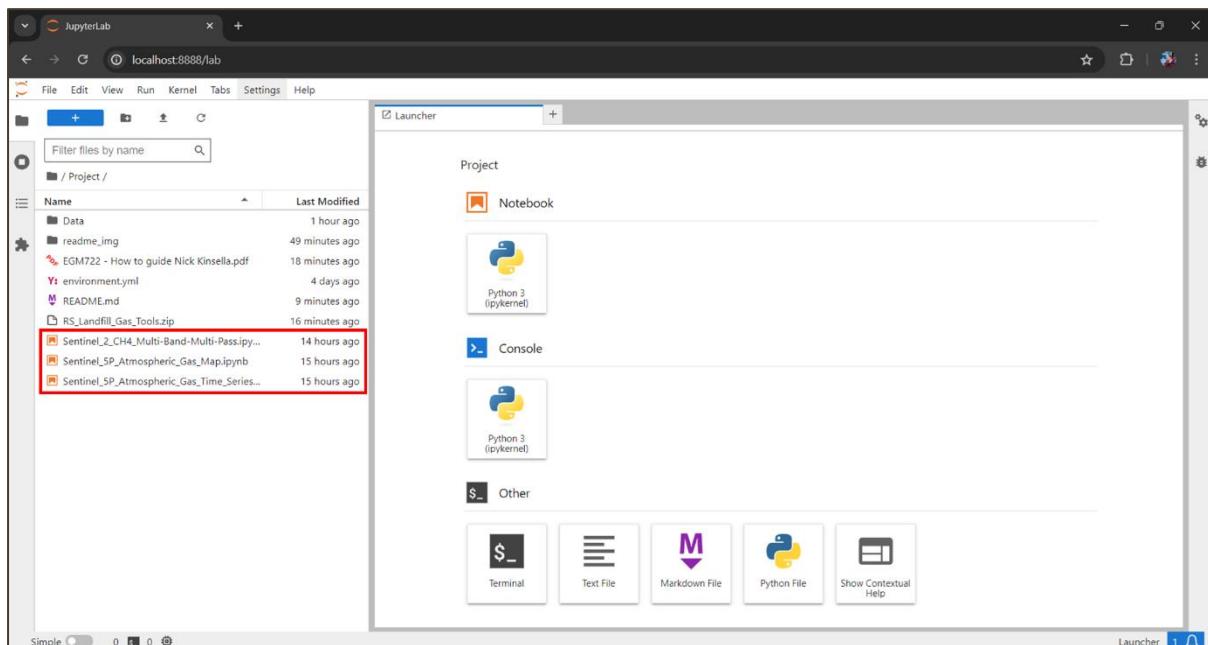


Figure 9: Location of tool scripts in Jupyter lab highlighted in red.

Click on one of the tools to open it. You can then follow the instructions, running the code by clicking the play button (fig.19).

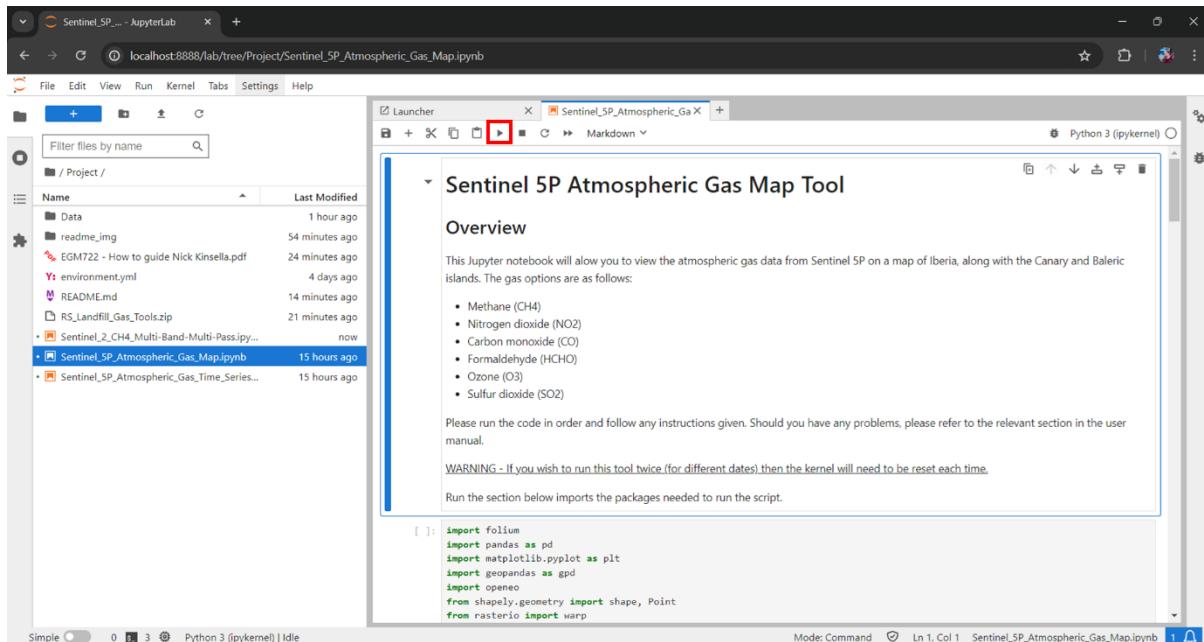


Figure 10: Location of the ‘play’ button which runs each of the code segments of the workbooks.

2.3.8 Authentication with OpenEO

The very first time one of the tools are run, the following section of code...

```
connection = openeo.connect(url="openeo.dataspace.copernicus.eu")
connection.authenticate_oidc()
```

... will provide you with a URL that will look something like this:

Visit https://auth.example.com/device?user_code=EAXD-RQXV to authenticate.

Copy this into your web browser and login using the Copernicus Data Space. Once this is complete, run tool’s Python script again and it will receive an authentication token, printing the message:

Authorized successfully.

In future you may be prompted with a new URL to create a new authentication token, whereby you should repeat the steps of this section.

2.4 S2MBMP Code

Code 1 shows the loaded dependencies.

```
import folium
import pandas
import matplotlib.pyplot as plt
import openeo
import rasterio
from rasterio.transform import from_origin
import numpy as np
import requests
from folium import Map, LayerControl, LatLngPopup, GeoJson
from folium.raster_layers import ImageOverlay
from IPython.display import display
from skimage import exposure
import geopandas as gpd
```

Code 1: Loading of dependencies for the code to run.

Code 2 connects to the OpenEO backend.

```
connection = openeo.connect(url="openeo.dataspace.copernicus.eu")
connection.authenticate_oidc()
```

Code 2: Connecting to OpenEO

Code 3 reads a pre-processed file that contains the field location bounding boxes, and then displays it for the end user to see.

```
studysite_csv =
pandas.read_csv(r'C:\GIS_Course\Methane_Point_Detection\all_files\Data\Algerian_Oil_and_Gas_Fields.csv')
pandas.set_option('display.max_rows', None)
print(studysite_csv)
```

Code 3: Displaying the contents of the site file for easy reference.

Code 4 provides the available datasets for a date range and site selected by the user. It also has a cloud filter set at 15% as any cloud in the scene interferes with the tool. This section of code was provided by Sonneveld, E. (2024)

```
def get_spatial_extent(site_id):
    site = studysite_csv[studysite_csv['id'] == site_id].iloc[0]
    return {
        "west": site['west'],
        "south": site['south'],
        "east": site['east'],
        "north": site['north']
    }

def fetch_available_dates(site_id, temporal_extent):
    spatial_extent = get_spatial_extent(site_id)
    catalog_url =
f"https://catalogue.dataspace.copernicus.eu/resto/api/collections/Sentinel12/search.json?box={spatial_extent['west']}%2C{spatial_extent['south']}%2C{spat
```

```

ial_extent['east']}%2C{spatial_extent['north']}&sortParam=startDate&sortOrde
r=ascending&page=1&maxRecords=1000&status=ONLINE&dataset=ESA-
DATASET&productType=L2A&startDate={temporal_extent[0]}T00%3A00%3A00Z&complet
ionDate={temporal_extent[1]}T00%3A00%3A00Z&cloudCover=%5B0%2C{cloud_cover}%5
D"
    response = requests.get(catalog_url)
    response.raise_for_status()
    catalog = response.json()
    dates = [date.split('T')[0] for date in map(lambda x:
x['properties']['startDate'], catalog['features'])]
    return dates

# Please enter your parameters here.
site_id = 86 # Specify the site ID.
temporal_extent = ["2020-01-01", "2020-04-30"] # Specify the the date range
you want to check for available data.
cloud_cover = 5

available_dates = fetch_available_dates(site_id, temporal_extent)
print("Available dates:", available_dates)

```

Code 4: Code for determining available dates for analysis.

Next the user will select the day to be investigated for emissions (code 5).

```

def active_emission(site_id, temporal_extent):
    site = studysite_csv[studysite_csv['id'] == site_id].iloc[0]

    active_emission = connection.load_collection(
        "SENTINEL2_L2A",
        temporal_extent=temporal_extent,
        spatial_extent={
            "west": site['west'],
            "south": site['south'],
            "east": site['east'],
            "north": site['north']
        },
        bands=["B11", "B12"],
    )
    active_emission.download("Sentinel-2_active_emissionMBMP.Tiff")

# Enter parameters for the active emission day
site_id = 86 # Specify the site ID
temporal_extent = ["2020-01-07", "2020-01-07"]

active_emission(site_id, temporal_extent)

```

Code 5: Code for downloading active emission dataset.

Then the user will choose a “no emission day” to compare the “active emission day” to. (code 6).

```

def no_emission(site_id, temporal_extent):
    site = studysite_csv[studysite_csv['id'] == site_id].iloc[0]

```

```

no_emission = connection.load_collection(
    "SENTINEL2_L2A",
    temporal_extent=temporal_extent,
    spatial_extent={
        "west": site['west'],
        "south": site['south'],
        "east": site['east'],
        "north": site['north']
    },
    bands=["B11", "B12"],
)
no_emission.download("Sentinel-2_no_emissionMBMP.Tiff")

# Enter parameters for the active emission day
site_id = 86 # Specify the site ID
temporal_extent = ["2020-01-04", "2020-01-04"]

no_emission(site_id, temporal_extent)

```

Code 6: Code for downloading active emission dataset.

Code 7 downloads a true colour satellite image to aid in the visualisation of the data.

```

def truecolour_image(site_id, temporal_extent):
    site = studysite_csv[studysite_csv['id'] == site_id].iloc[0]

    truecolour_image = connection.load_collection(
        "SENTINEL2_L2A",
        temporal_extent=temporal_extent,
        spatial_extent={
            "west": site['west'],
            "south": site['south'],
            "east": site['east'],
            "north": site['north']
        },
        bands=["B02", "B03", "B04"],
    )
    truecolour_image.download("Sentinel-2_truecolourMBMP.Tiff")

# Enter parameters for the no emission day
site_id = 86 # Specify the site ID
temporal_extent = ["2020-01-07", "2020-01-07"]

truecolour_image(site_id, temporal_extent)

```

Code 7: Code for downloading true colour satellite image for data visualisation

Code 8 runs the analysis as outlined at the beginning of this section (4.3).

```

# Define file paths
Active_Multiband = "Sentinel-2_active_emissionMBMP.Tiff"
No_Multiband = "Sentinel-2_no_emissionMBMP.Tiff"

```

```

import numpy as np
import rasterio

# Define a function for least squares fitting
def least_squares_fit(x, y):
    # Remove NaNs (if any) for valid calculations
    mask = ~np.isnan(x) & ~np.isnan(y)
    x_valid = x[mask]
    y_valid = y[mask]

    # Calculate least squares fit parameters
    A = np.vstack([x_valid, np.ones_like(x_valid)]).T
    m, c = np.linalg.lstsq(A, y_valid, rcond=None)[0]
    return m, c

# Open datasets and perform least squares fitting
with rasterio.open(Active_Multiband) as Active_img,
rasterio.open(No_Multiband) as No_img:
    Active_B11 = Active_img.read(1)
    Active_B12 = Active_img.read(2)
    No_B11 = No_img.read(1)
    No_B12 = No_img.read(2)

    # Perform least squares fitting for Active_B11 vs Active_B12
    m_active, c_active = least_squares_fit(Active_B11.flatten(),
Active_B12.flatten())
    Corrected_Active_B12 = m_active * Active_B12 + c_active

    # Perform least squares fitting for No_B11 vs No_B12
    m_no, c_no = least_squares_fit(No_B11.flatten(), No_B12.flatten())
    Corrected_No_B12 = m_no * No_B12 + c_no

    # Calculate the fractional change
    SWIR_diff = (Active_B11 - Corrected_Active_B12) - (No_B11 -
Corrected_No_B12)

# Define the output file path
output_file = "SWIR_diff_output.tif"

# Define the transform and metadata for the output file
transform = Active_img.transform
meta = Active_img.meta.copy()
meta.update({
    "driver": "GTiff",
    "height": SWIR_diff.shape[0],
    "width": SWIR_diff.shape[1],
})

```

```

    "count": 1,
    "dtype": SWIR_diff.dtype,
    "crs": Active_img.crs,
    "transform": transform
  })

# Save the SWIR_diff array to the output file
with rasterio.open(output_file, "w", **meta) as dest:
  dest.write(SWIR_diff, 1)

print(f"SWIR_diff has been saved to {output_file}")

```

Code 8: Code running analysis

Code 9 displays the map. It firstly takes the true colour satellite image bands, stacks them and applies a brightness factor to make the image clearer. Then it adds the SWIR_diff data calculated in the previous code and sets it to only display data between 1.5 and 3 standard deviations above the mean. It adds scaled grid lines to help understand distances on the image. Finally it downloads the map as a .jpg file.

```

# Load the true color image
truecolour_sat = 'Sentinel-2_truecolourMBMP.Tiff'
img = rasterio.open(truecolour_sat)
blue = img.read(1)
green = img.read(2)
red = img.read(3)

# Adjust brightness
brightness_factor = 0.03 # Increase brightness factor
blue = np.clip(blue * brightness_factor, 0, 255)
green = np.clip(green * brightness_factor, 0, 255)
red = np.clip(red * brightness_factor, 0, 255)

# Stack bands to create RGB image
rgb = np.dstack((red, green, blue))
rgb = rgb / rgb.max()

# Apply logarithmic transformation to enhance contrast
rgb = np.log1p(rgb)

# Normalize the transformed image
rgb = rgb / rgb.max()

# Create folium map
m = Map(location=[31.7294, 6.0200], zoom_start=10, control_scale=True)

# Add true color image overlay
truecolour_overlay = ImageOverlay(
  image=rgb,
  bounds=[[31.485122001357873, 5.732553997080988], [31.973701421209405,
  6.308794405143235]],
  opacity=1,

```

```
    interactive=True,
    cross_origin=False,
    zindex=1,
)
truecolour_overlay.add_to(m)

# Calculate mean and standard deviation of SWIR_diff
mean = np.nanmean(SWIR_diff)
std = np.nanstd(SWIR_diff)

# Standard deviation stretch
std_factor = 2 # Number of standard deviations
lower_bound = mean - std_factor * std
upper_bound = mean + std_factor * std

# Normalize SWIR_diff using standard deviation stretch
normalized_SWIR_diff = (SWIR_diff - lower_bound) / (upper_bound -
lower_bound)
normalized_SWIR_diff[normalized_SWIR_diff < 0] = 0
normalized_SWIR_diff[normalized_SWIR_diff > 1] = 1

# Apply colormap
cmap = plt.get_cmap('plasma')
SWIR_colored = cmap(normalized_SWIR_diff)

# Add SWIR_diff overlay
SWIR_overlay = ImageOverlay(
    image=SWIR_colored,
    bounds=[[31.485122001357873, 5.732553997080988], [31.973701421209405,
6.308794405143235]],
    opacity=1,
    interactive=True,
    cross_origin=False,
    zindex=2,
)
SWIR_overlay.add_to(m)

# Load GeoJSON file (already in EPSG:3857)
vector_point_path =
"C:/GIS_Course/Methane_Point_Detection/all_files/Data/known_point_sources.ge
ojson"
gdf = gpd.read_file(vector_point_path)

# Add vector point file overlay
GeoJson(gdf.to_json()).add_to(m)

# Add layer control
LayerControl().add_to(m)
```

```
# Add a click event to show latitude and longitude when clicking on the map  
m.add_child(LatLngPopup())  
  
# Display the map  
display(m)
```

Code 9: Code displaying the map

2.5 Expected Results and Demo

This section will illustrate the functionality of the tools by showcasing documented methane plumes and their corresponding depiction in the results. To ensure that this tool works as intended, a known emission in Algeria used by Varon et al. (2021) was imaged (fig.20).

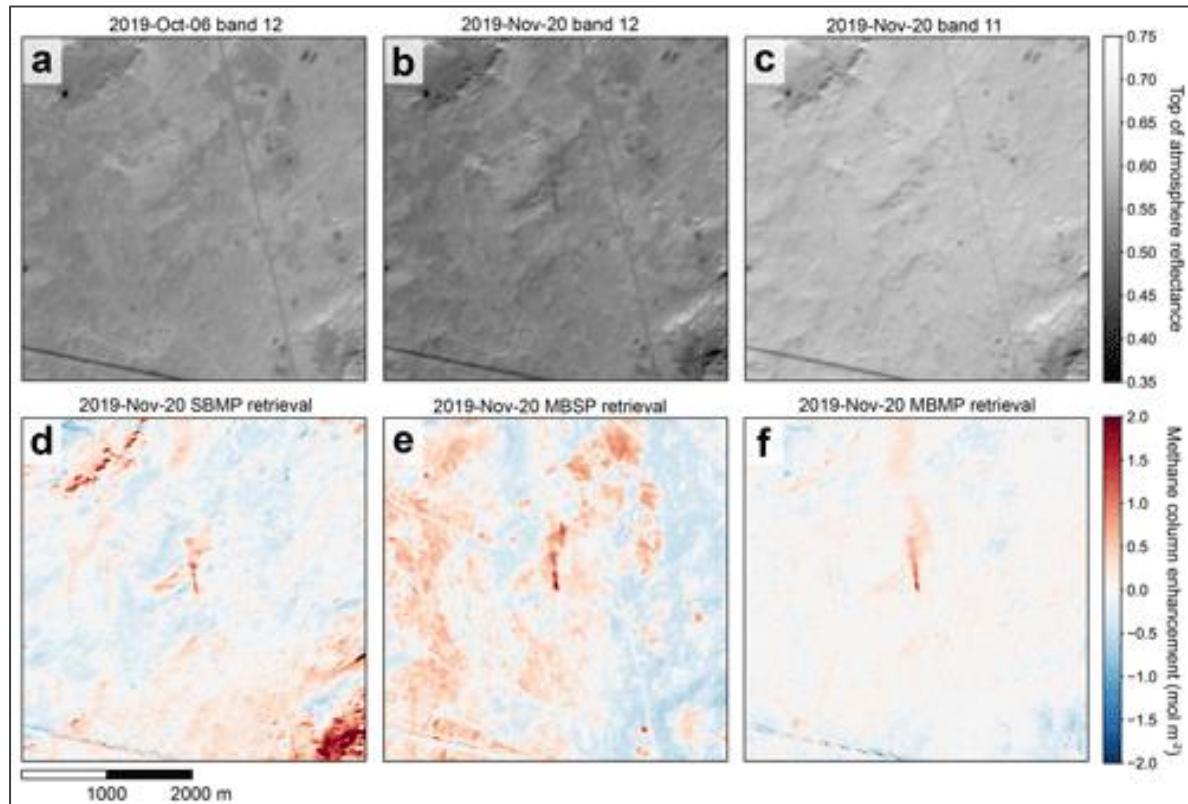


Figure 11: Sentinel 2 SWIR-2 (b), SWIR-1 (c) and Methane column retrieval for Multi Band Multi Pass method (e) from Varon et al. (2021)

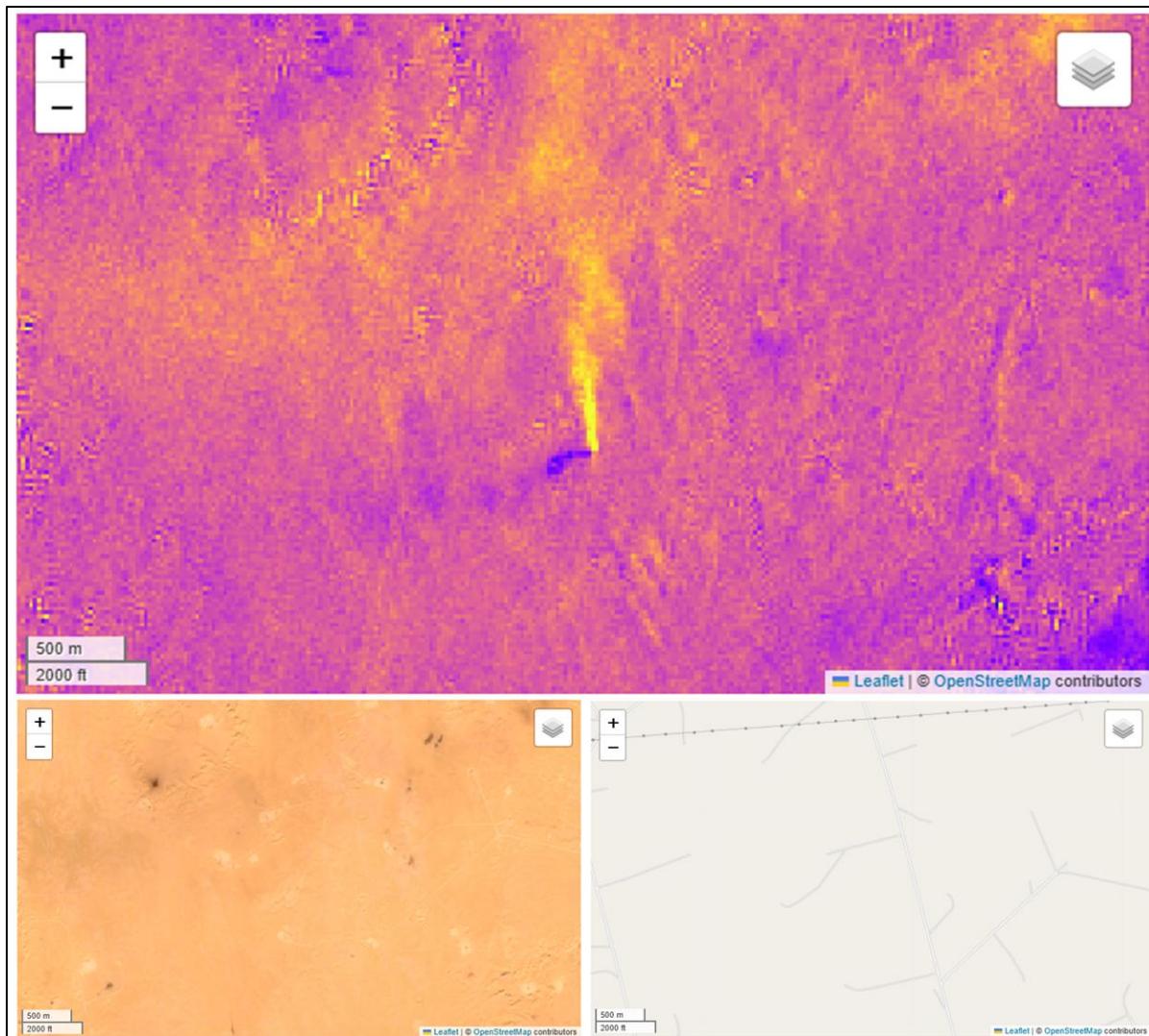


Figure 12: Methane column retrieval using S2-MBSP tool.

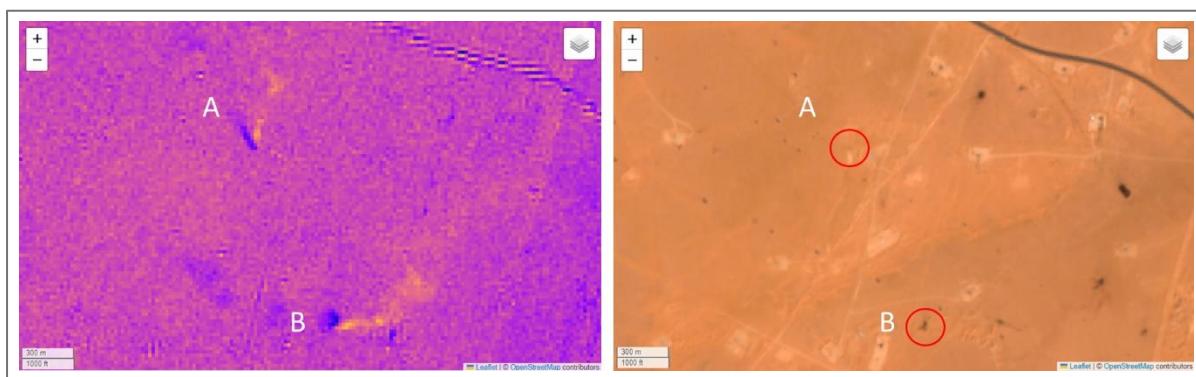


Figure 13: Images from the 19th of August 2021. Plume A location: Lat: 31.7771, Long: 5.9982. Plume B location: Lat: 31.7679, Long: 6.0029.

2.6 Troubleshooting

Input errors have been covered in the tool notebooks, however there are errors unrelated to a user input which can cause problems. These are detailed below.

2.6.1 Remote disconnected error (All tools)

Error: Remote disconnected

Or

OpenEoApiError: [500] Internal: Server error: KazooTimeoutError('Connection time-out') (ref: r-2405 108830e742b59ef8ac2f28647fb5)

This can occur when there are issues with the Copernicus network. In the event that you see an error like this you can check page <https://dataspace.copernicus.eu/news> for any downtime messages and you can also contact the Copernicus dataspace team via the form at <https://helpcenter.dataspace.copernicus.eu/hc/en-gb/requests/new>

2.6.2 Concurrent job error (S5-AGM)

OpenEoApiError: [400] ConcurrentJobLimit: Job was not started because concurrent job limit (2) is reached. (ref: r-240413b5d1b240118da9f9ed90807c58)

This can happen when S5-AGT tool is run, cancelled and then run again. If this happens the process is still running in the background and needs to be cancelled.

To do this go to the following URL: <https://openeo.dataspace.copernicus.eu/>

You will be presented with the following screen (fig.23). Please click the highlighted login button.

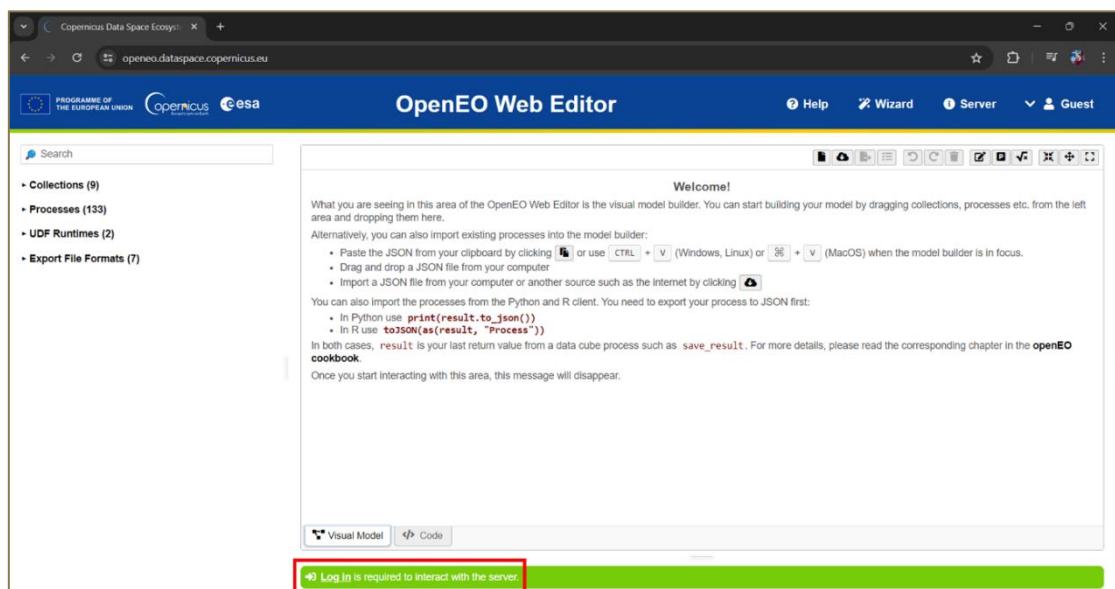


Figure 14: OpenEO Web Editor with login button highlighted in red

You will then see the following screen (fig.24).

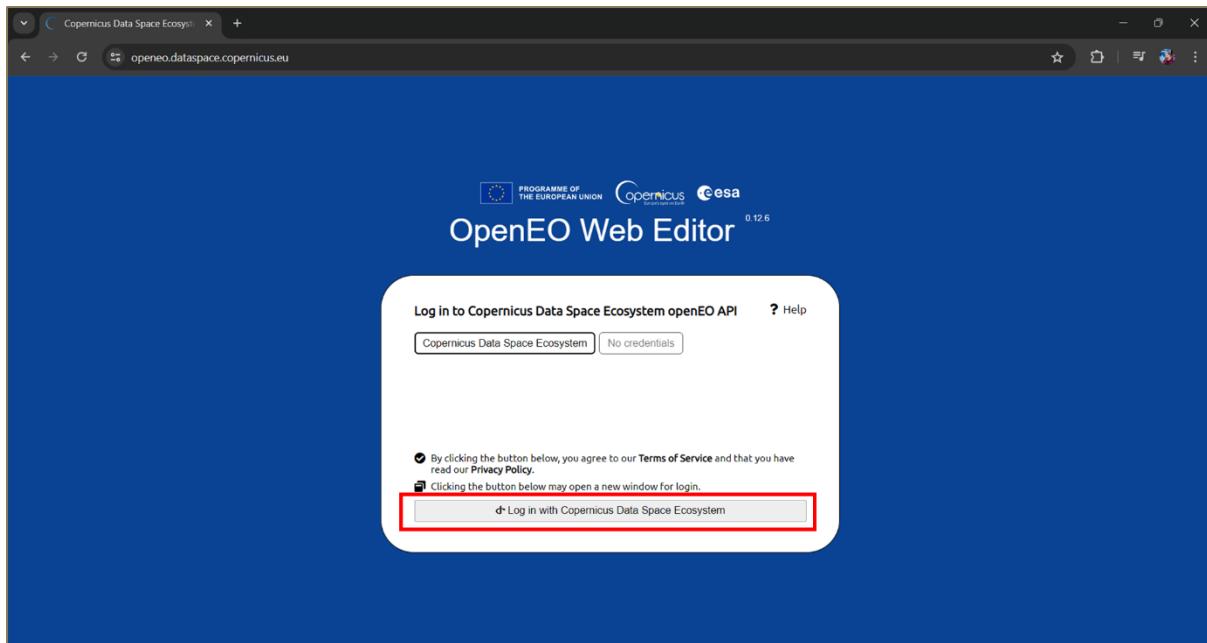


Figure 24: OpenEO Web editor login prompt with login button highlighted in red.

Simply click the highlighted button (fig.25) and follow the process. You should then be returned to the OpenEO web editor but now you will see a list of processes including the active ones.

The screenshot shows the OpenEO Web Editor interface. At the top, there are navigation links for 'Copernicus Data Space Ecosystem', 'Help', 'Wizard', 'Server', and a user profile for 'Nicholas Kinsella'. The main area has a sidebar with 'Collections (9)', 'Processes (133)', 'UDF Runtimes (2)', and 'Export File Formats (7)'. The main workspace is titled 'Welcome!' and contains instructions for importing processes. Below this is a table titled 'Batch Job' with the following data:

Batch Job	Status	Submitted	Last update	Actions
CH4 timeseries	running	15/04/2024, 8:54:30 UTC	15/04/2024, 9:00:21 UTC	
CH4 timeseries	finished	15/04/2024, 8:49:39 UTC	15/04/2024, 8:52:36 UTC	
CH4 timeseries	finished	13/04/2024, 11:04:00 UTC	13/04/2024, 11:06:12 UTC	
CH4 timeseries	finished	13/04/2024, 8:56:24 UTC	13/04/2024, 9:03:07 UTC	
CH4 timeseries	finished	13/04/2024, 8:28:37 UTC	13/04/2024, 8:32:10 UTC	
SWIR timeseries	finished	13/04/2024, 8:16:12 UTC	13/04/2024, 8:20:10 UTC	
SWIR timeseries	finished	12/04/2024, 22:15:30 UTC	12/04/2024, 22:18:08 UTC	

Figure 25: OpenEO Web Editor showing batch job screen, with running job and delete button highlighted.

To stop the process, simply click the highlighted bin button (delete). This should allow the tool to work normally again.

3. Integrated Methane Inversion (IMI)

3.1 Methodology

The Sentinel-5P TROPOMI Methane product provides atmospheric CH₄ concentrations in parts per billion (ppb). Its data has been shown to be in close agreement with ground-based measurements from the Total Carbon Column Observing Network (Liu et al., 2021; Lorente et al., 2021). Additional steps are required to estimate ground level CH₄ emissions from these observed whole atmosphere concentrations, because CH₄ can travel significant distances from its source and may be present in high altitudes over otherwise low-emission areas, and vice versa (Varon et al., 2022), resolving this requires an atmospheric transport model to be applied the data

The IMI (Integrated Methane Inversion) is a Python-based tool developed by Varon et al. (2022) on Amazon Web Services to estimate emissions in a selected area (fig. 26). It starts by using official emission inventories to establish a baseline, known as the prior estimate. Then, the GEOS-Chem atmospheric transport model predicts how these emissions should appear in satellite observations, factoring in environmental conditions like wind speed and direction. These predictions are compared with actual observations from Sentinel-5P. By analysing the differences, the tool calculates a posterior estimate, which quantifies the total emissions and reveals whether the prior estimate was too high or too low.

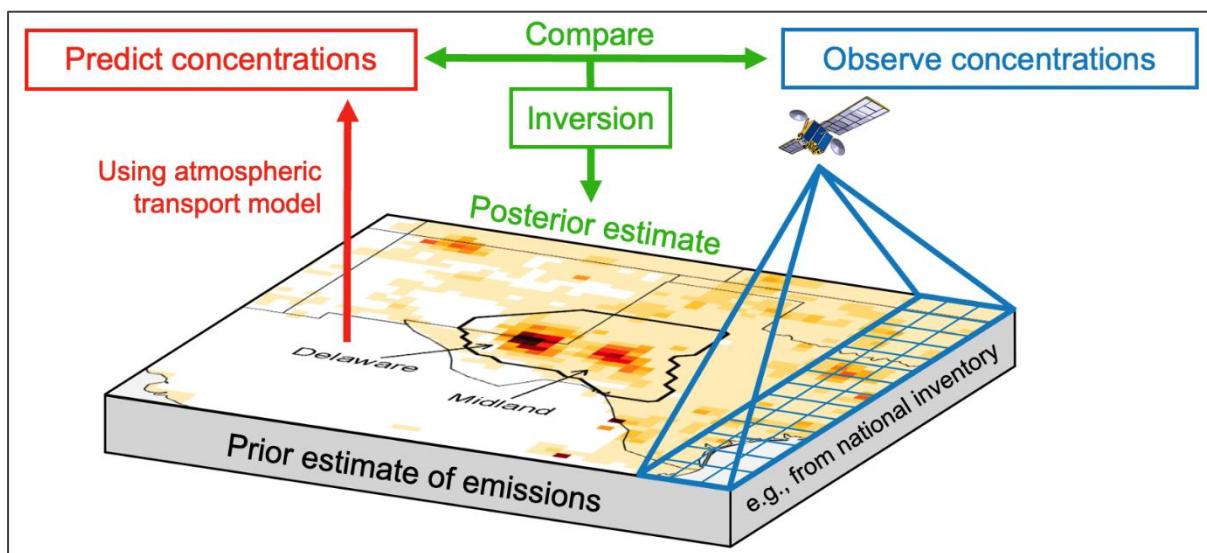


Figure 26: IMI processing step (Integrated Methane Inversion, 2024).

3.2 Data and Dependencies

The datasets used by these tools are outlined in table 1.

Table 3: Datasets used by tools

Description	Data	Source	For use in tool(s)

Table 4: dependencies required to use the tools.

3.3 Setup

The following steps will show you how to setup a AWS account and run the IMI tool

3.3.1 Creating an AWS account

Creating an AWS account requires providing some personal and bank card information, as it is not a free service. The cost of running the IMI will be provided as an estimate prior to you running the full analysis. Open the following link in a browser <http://aws.amazon.com/> and click on create an AWS account. Thereafter, follow the prompts.

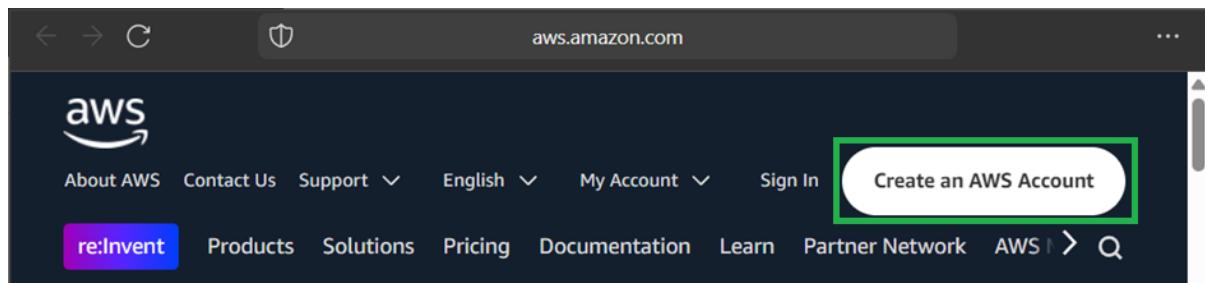


Figure 27: AWS sign in and create account landing page.

3.3.2 Add Amazon Simple Storage Service (S3) user permissions

This is an optional step if you wish to store your results long term. Firstly navigate to the AWS management console by clicking on the “Sign In to the Console” button or by using the “My Account” dropdown (fig. 28).

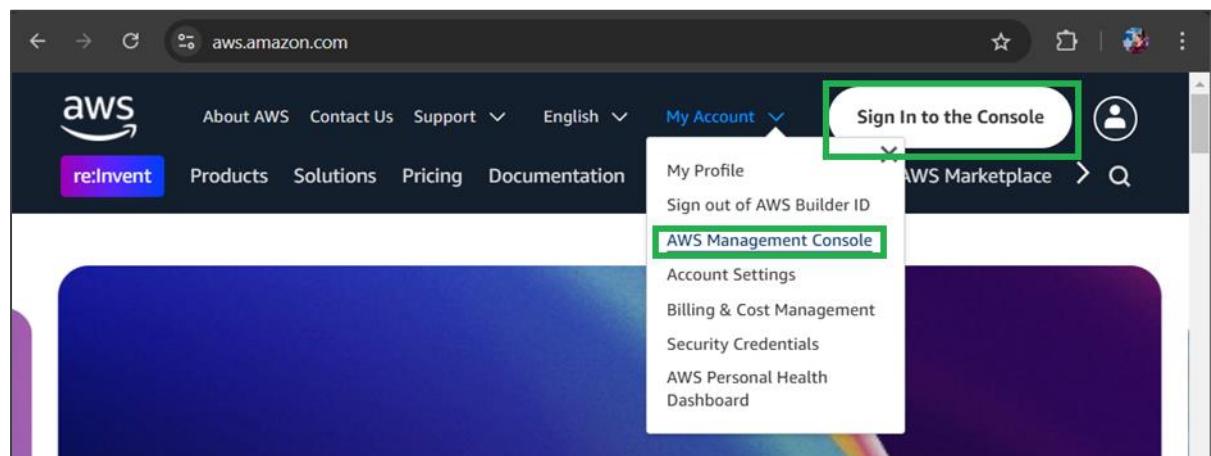


Figure 28: Sign in to my Console access on AWS landing page.

Once on the management console page, click the magnifying glass (fig. 29)

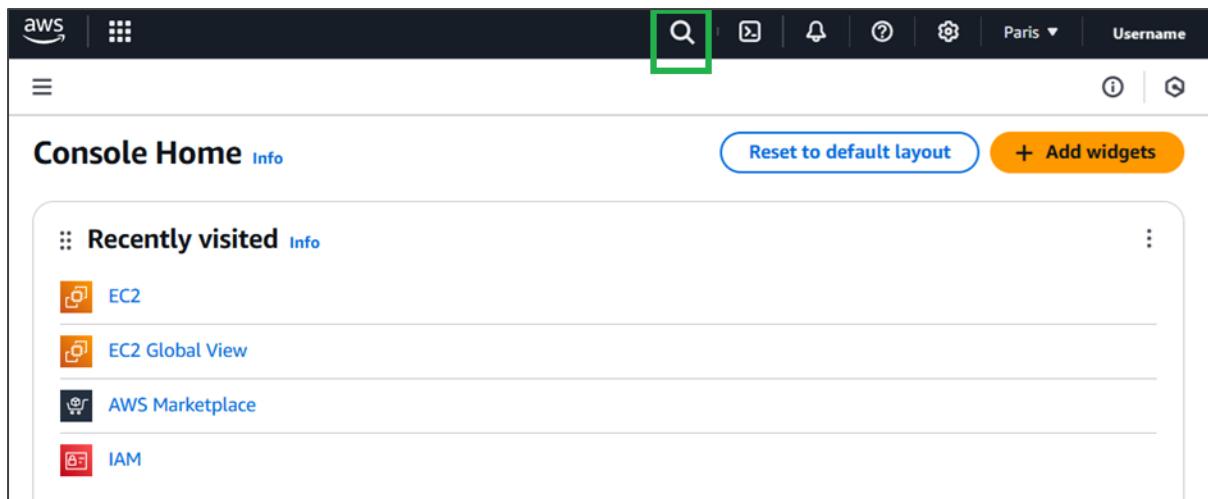


Figure 29: Location of search function in AWS console.

Then search for “IAM” (fig. 30).

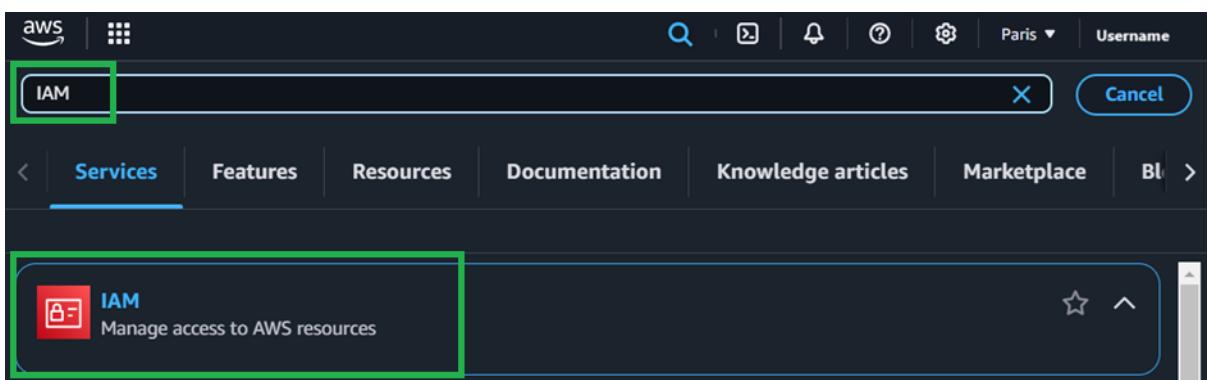


Figure 30: search function finding the AWS resource management access.

Under the “Access management” sidebar, click on Roles and then “Create role” (fig. 31)

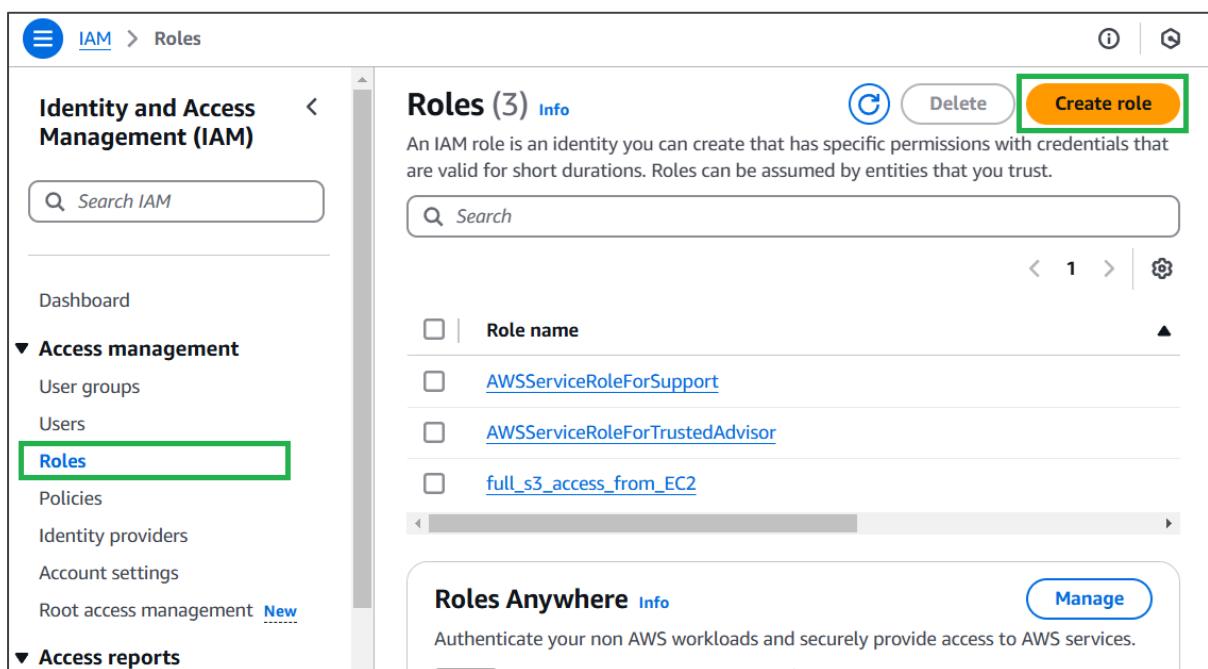


Figure 31: create role button location

In the “Use Case”, select EC2 and then click “next”.

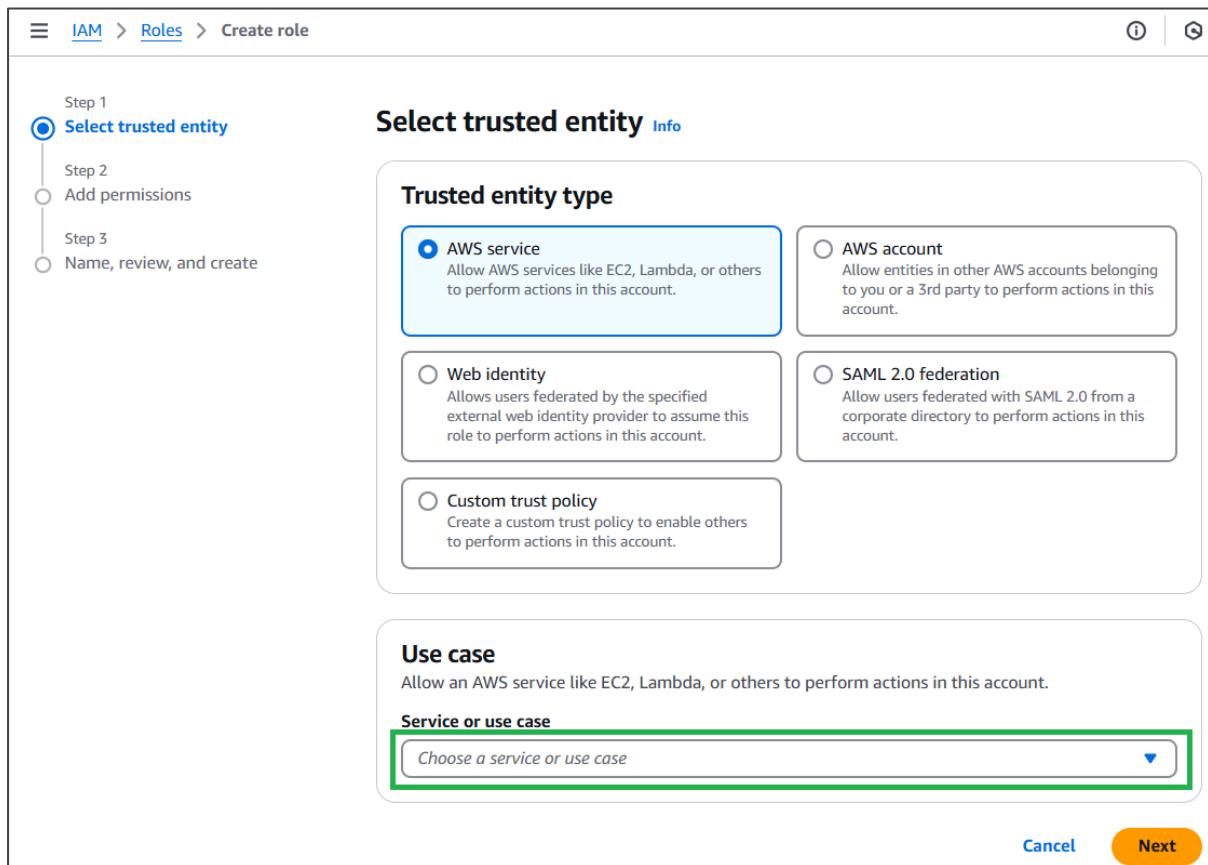


Figure 32: Use case selection location.

In the “Add permissions” screen type S3 in the search box and select AmazonS3FullAccess (fig.33). Click next.

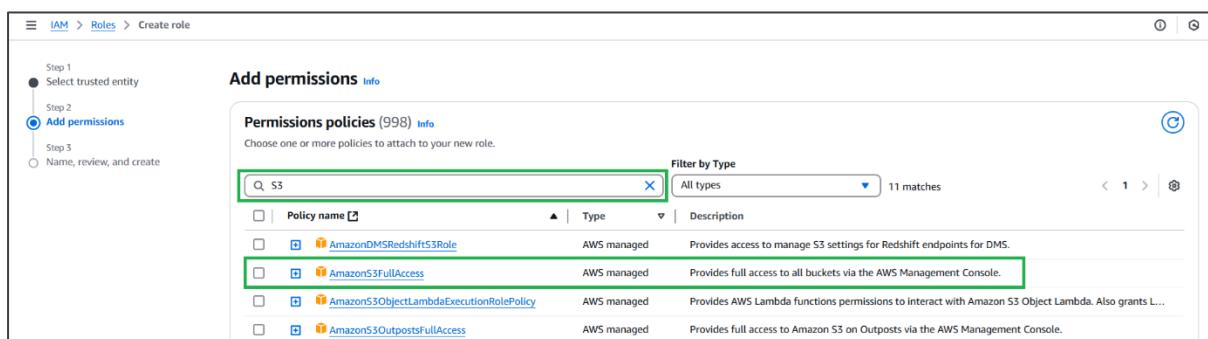


Figure 33: Use case selection location.

Lastly, choose a descriptive name for the role like “full_S3_access_from_EC2”. Click “create role” at the bottom of the page (fig. 34).

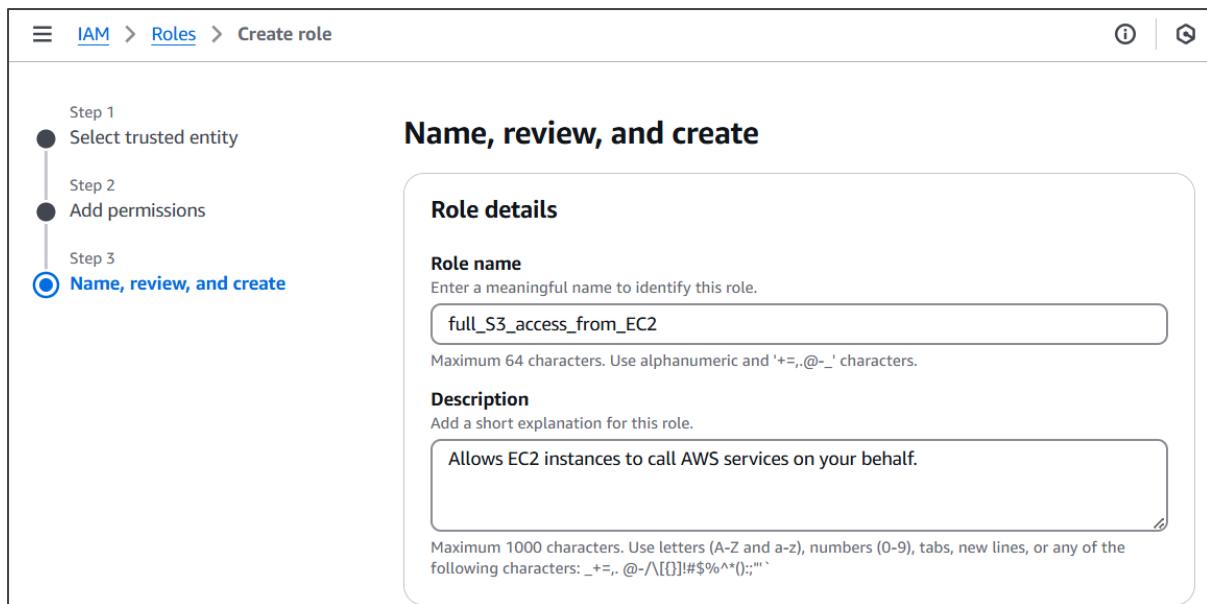


Figure 34: Create role naming screen.

Now a new IAM role is created. This only needs to be done once.

3.3.3 Launching an IMI instance

Navigate to <https://aws.amazon.com/marketplace/pp/prodview-hkuxx4h2vpjba> or search for “Integrated Methane Inversion”. You should find the page shown in figure 35. Click “view purchase options”

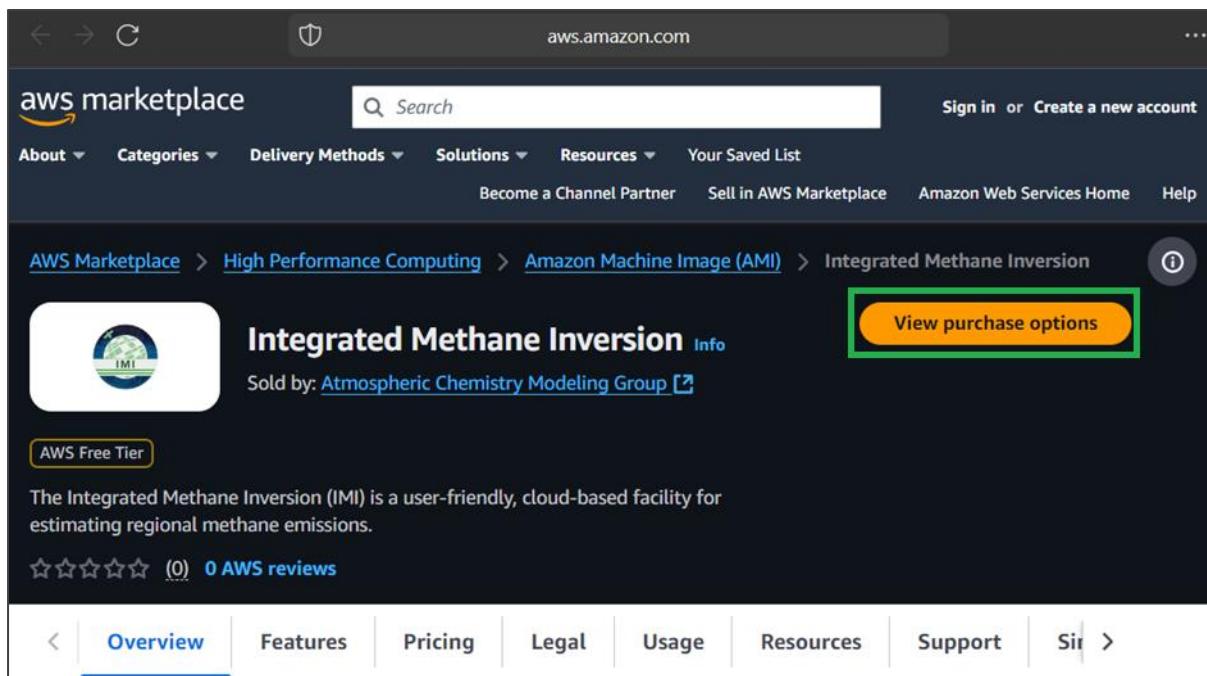


Figure 35: IMI landing page and purchase options button.

Next click “continue to configuration”.

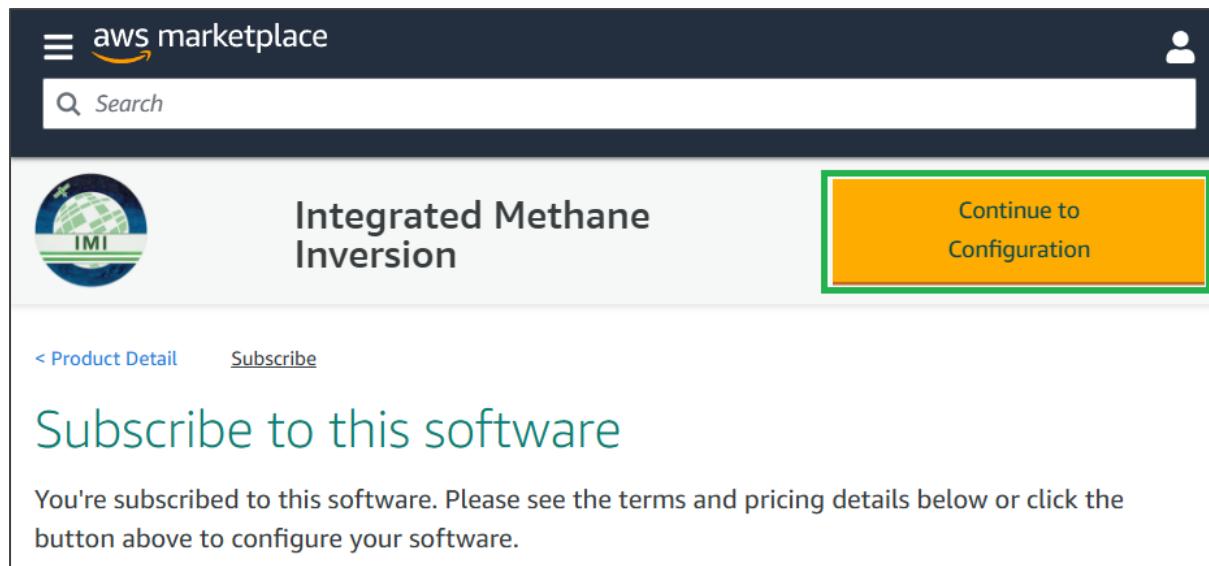


Figure 36: Subscription page with continue to configuration button.

Choose your preferred region and IMI version, then click “Continue to Launch.” (fig. 37) Opting for a region closer to your physical location can enhance network connectivity but may lead to higher costs compared to selecting the region where GEOS-Chem data is hosted (us-east-1, N. Virginia).

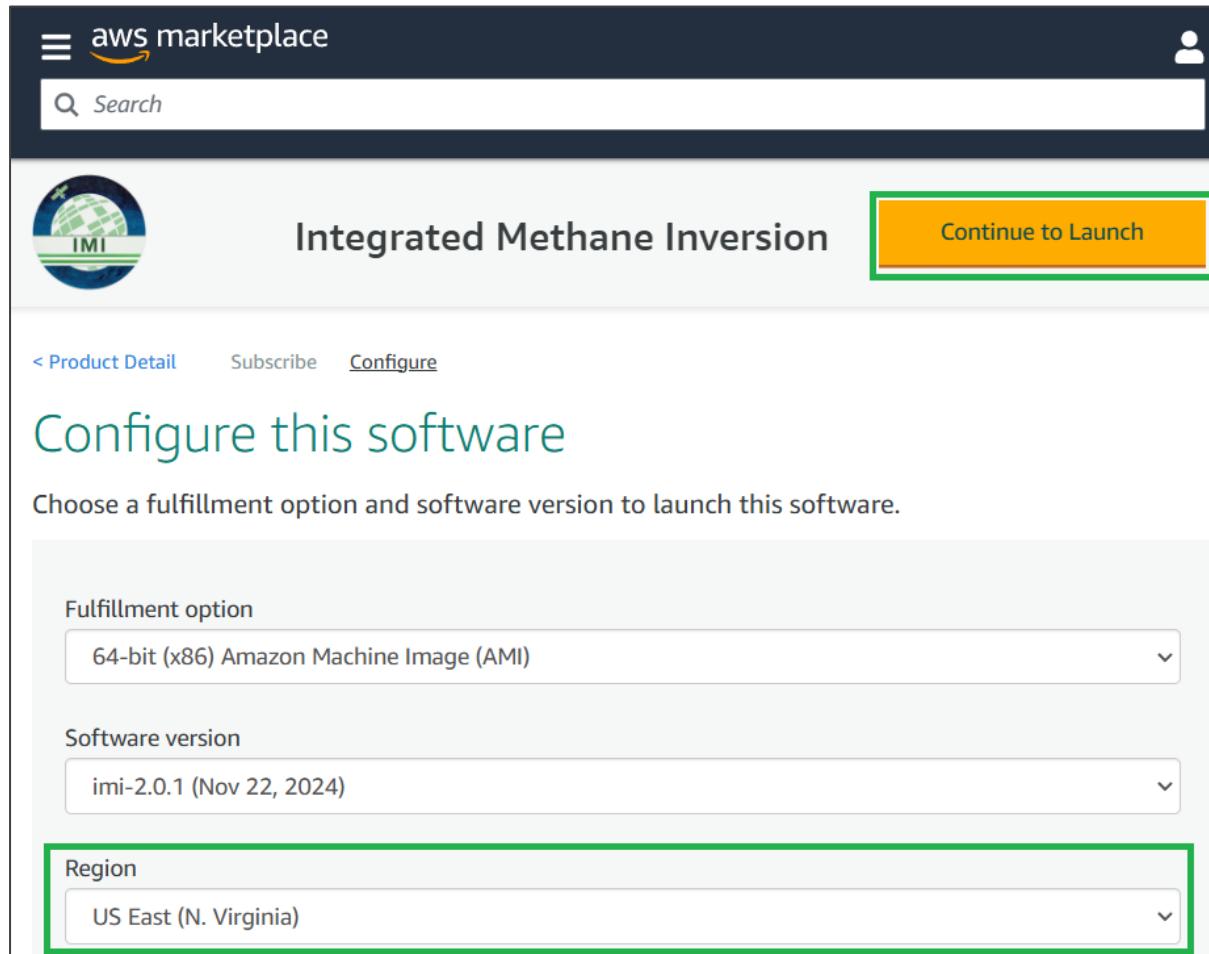


Figure 37: region configuration page and “continue to launch button”.

In the “Launch this software” page, in the choose action prompt, select “Launch through EC2” and then click “Launch” (fig. 38).

The screenshot shows the AWS Marketplace interface for the 'Integrated Methane Inversion' software. At the top, there's a navigation bar with links for 'About', 'Categories', 'Delivery Methods', 'Solutions', 'Resources', 'Your Saved List', and options to 'Become a Channel Partner', 'Sell in AWS Marketplace', 'Amazon Web Services Home', and 'Help'. A search bar is also present. On the left, there's a logo for 'IMI' featuring a globe icon. The main title 'Integrated Methane Inversion' is displayed prominently. Below the title, there are four buttons: '< Product Detail', 'Subscribe', 'Configure', and 'Launch'. The 'Launch' button is underlined, indicating it's the active or intended action. A large teal section titled 'Launch this software' contains the instruction: 'Review the launch configuration details and follow the instructions to launch this software.' Below this, a 'Configuration details' section lists the following information:

Fulfillment option	64-bit (x86) Amazon Machine Image (AMI)
	Integrated Methane Inversion
	<i>running on c5.9xlarge</i>
Software version	imi-2.0.1
Region	US East (N. Virginia)

A blue-bordered box labeled 'Usage instructions' is positioned below the configuration details. In the bottom right corner of the main content area, there's a yellow 'Launch' button. To the left of the 'Launch' button, a 'Choose Action' section contains a dropdown menu with the option 'Launch through EC2' selected. A tooltip next to the dropdown explains: 'Choose this action to launch your configuration through the Amazon EC2 console.'

Figure 37: Launch software page.

Now it's time to determine the name of your instance and the hardware to run the IMI. The primary considerations are the number of CPUs and the amount of RAM. In the "Instance Type" section, you can choose from a wide range of options. The IMI will perform faster with a higher number of CPUs. The c5.9xlarge instance type is the recommended setting. It offers 36 CPU cores and 72GB of RAM. However, you can choose a different instance type with more or fewer cores and memory based on your specific needs (fig. 38).

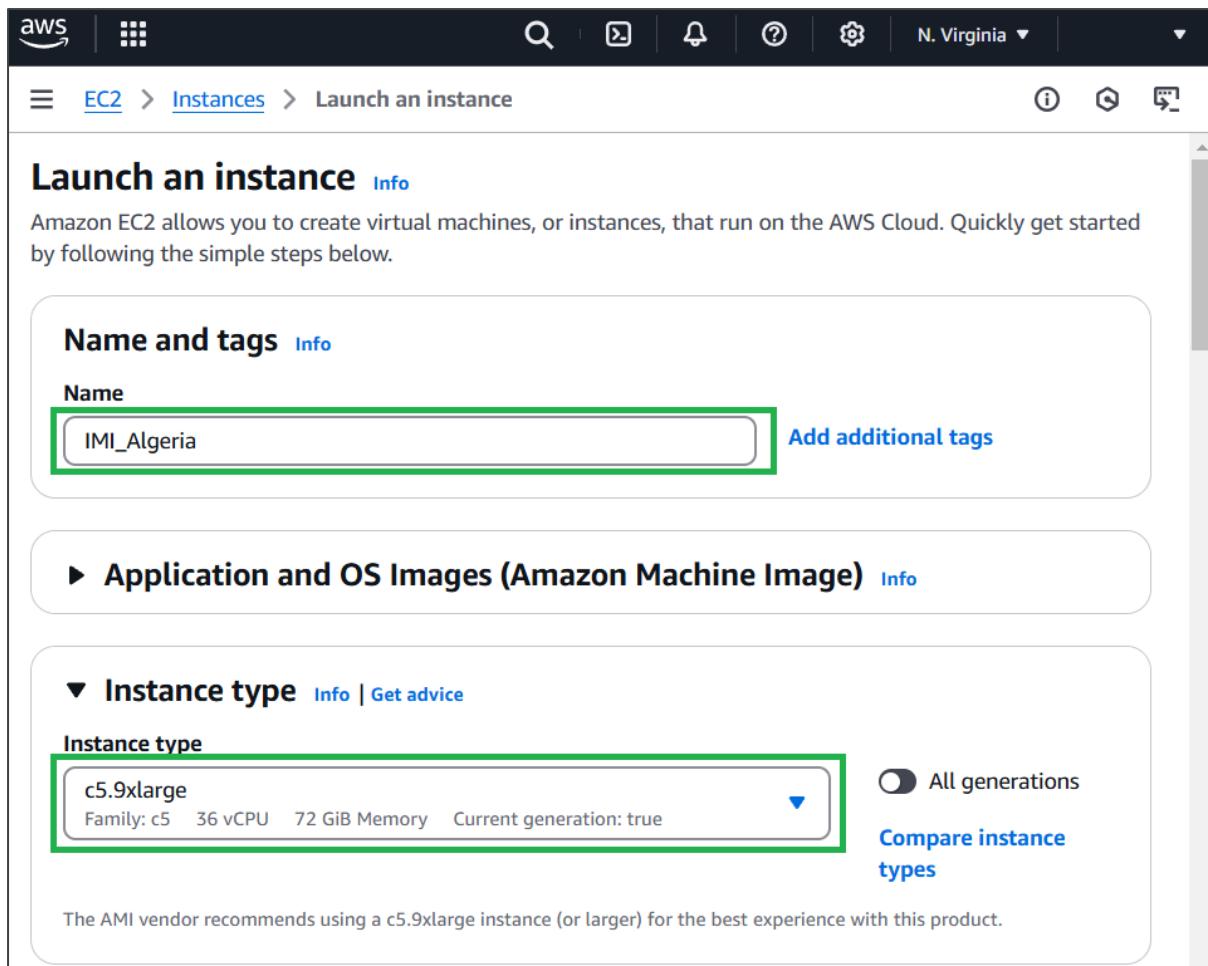


Figure 38: Hardware selection page.

In the next step, you'll create a new SSH key pair or select an existing one. This key pair acts as the password for accessing your server via SSH. To create a new key pair, click "Create new key pair". In the dialog box, assign a name to your key pair (e.g., imi_testing) and click "Create key pair". Once created, the key pair will be available for future use, and you can simply select it from the dropdown menu.

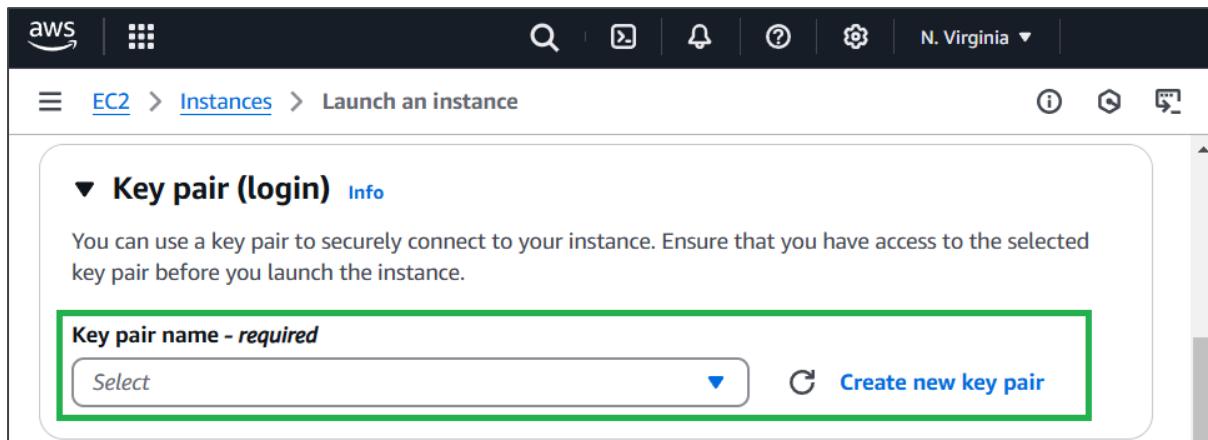


Figure 39: keypair selection.

If you are creating a key pair, use the settings shown in figure 40, choosing an appropriate name. Click "create key pair" once you have finished, saving it in a memorable directory that will be used later.

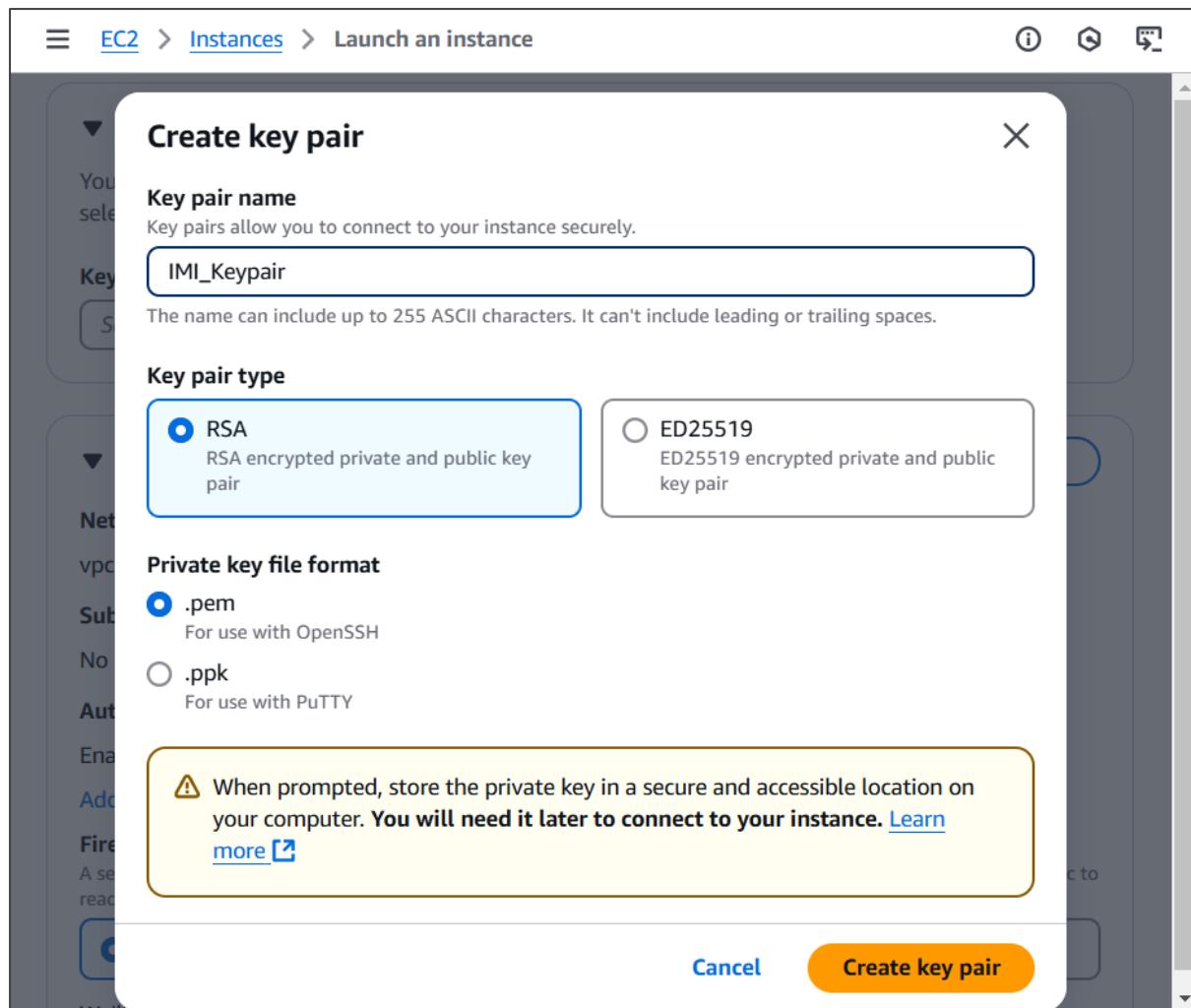


Figure 40: keypair setup.

Proceed to the "Configure Storage" section (fig.41), where you will select the size of the storage volume. Storage costs are around USD \$100 per month per TB of space. The amount of storage you choose will depend on how long you intend to analyse an area. According to the estimates provided by the IMI team, 100 gigabytes is sufficient for a 1-week inversion of the Permian Basin, which spans an area of 223,000 km².

To determine the storage required for Hassi Messaoud and its surrounding area, which covers a much smaller area of 3,562 km², we can calculate the storage needed for a 1-week inversion:

For every 1 km², approximately 0.000448 GB of storage is required for a 1-week inversion.

Using this, the calculation for the storage needed for 3,562 km² is:

$$0.000448 \text{ GB} \times 3,562 \text{ km}^2 = 1.59 \text{ GB for a 1-week inversion.}$$

For a full year of data (52 weeks), the total storage required for Hassi Messaoud would be:

$$1.59 \text{ GB} \times 52 = 82.98 \text{ GB.}$$

It is sensible to add slightly more, so in this example it will be set at 100GB (fig. 41).

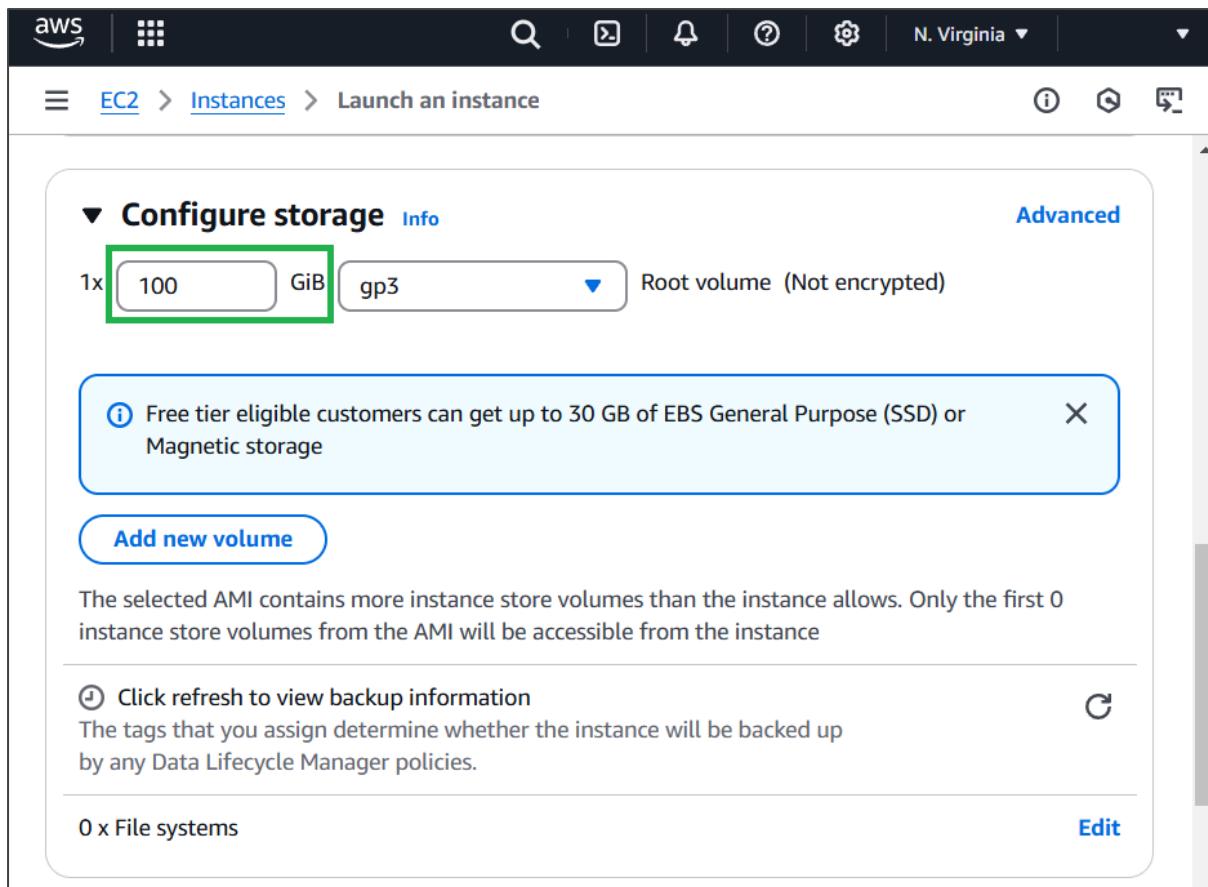


Figure 41: storage configuration section.

Next, open the “Advanced Details” section and choose the IAM role you created in step 3 from the “IAM Instance Profile” dropdown. This will grant your EC2 instance access to S3 for uploading output data. You can leave all other configuration settings in the “Advanced Details” section as the default values.

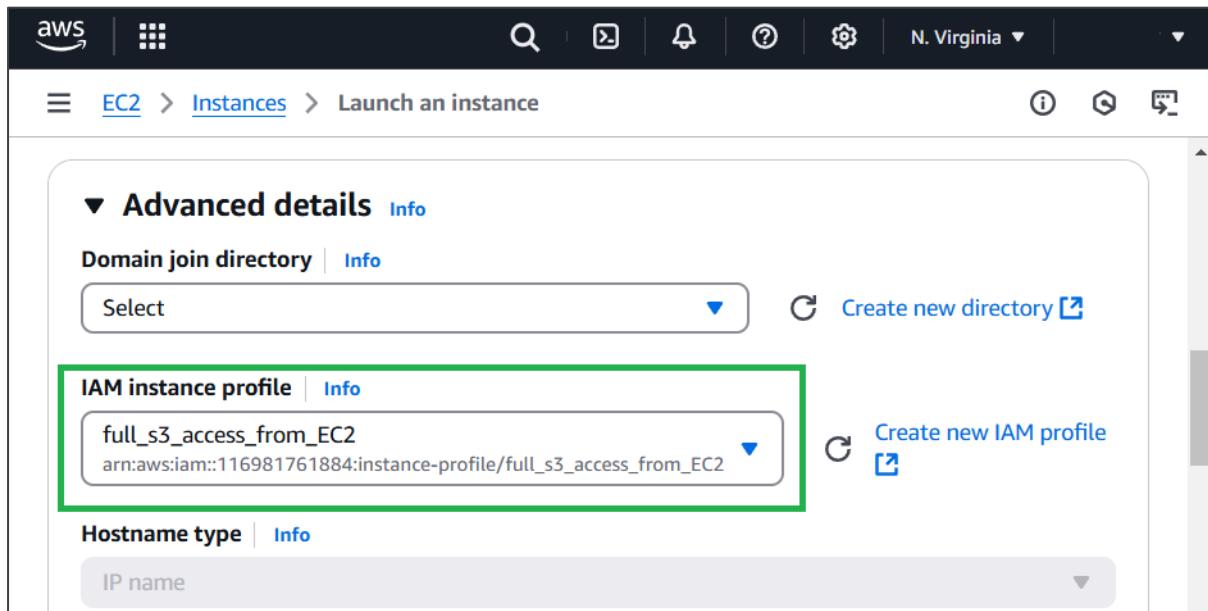


Figure 42: advanced details section.

Finally click the launch instance button in the section showed in figure 43.

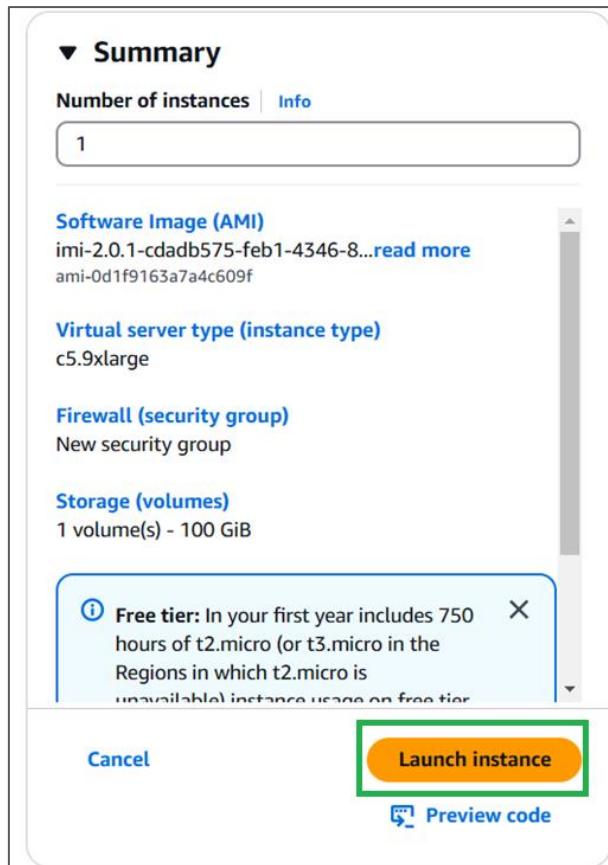


Figure 43: Summary section with “launch instance” button.

3.3.4 Connecting to instance

To connect to the instance and run the IMI, a program called Git will need to be downloaded and used. You can download Git from the following URL: <https://git-scm.com/downloads/win>, choosing the appropriate version of the software for your system.

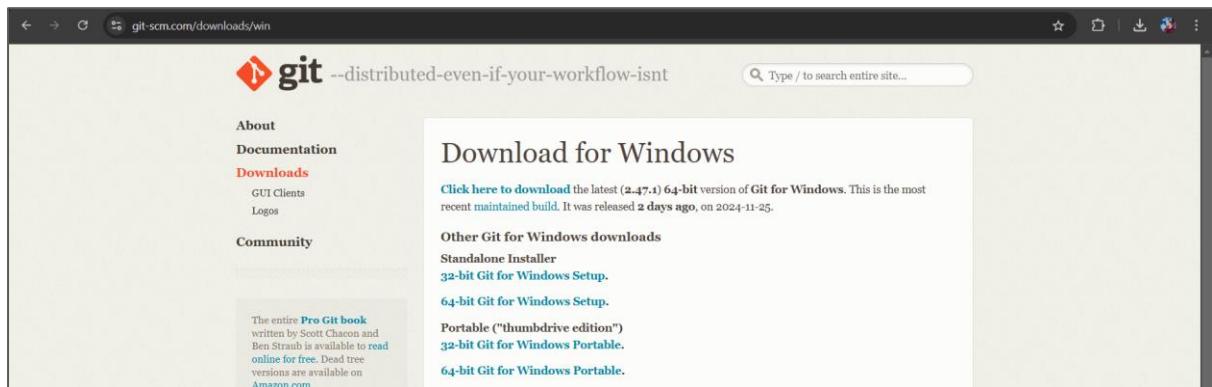


Figure 44: download page for Git

Once downloaded, run and follow the instructions on the installation wizard.

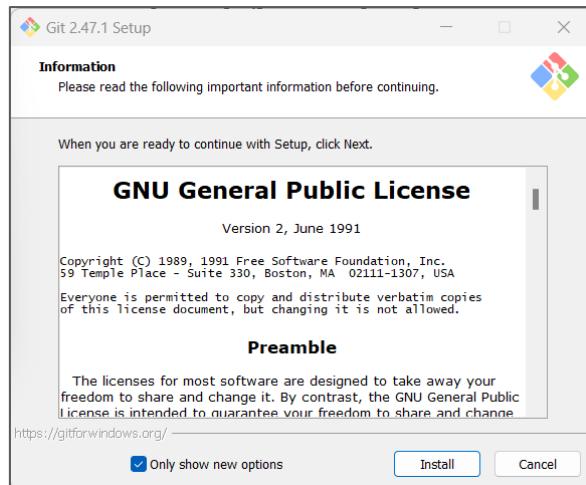


Figure 45: Git installation wizard

Once completed, tick the box that says launch Gitbash. We will come back to it shortly.

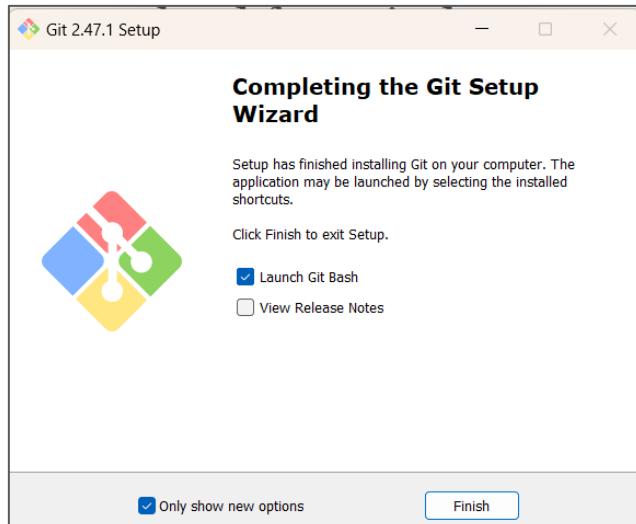


Figure 46: Git installation wizard

Returning to the AWS management console, the first time you launch an instance it should be running already (fig. 47).

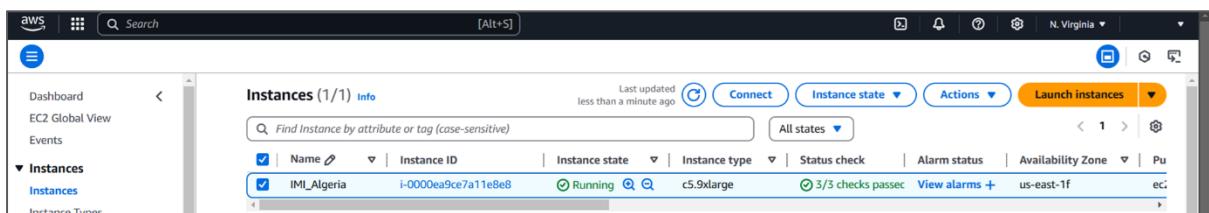


Figure 47:

If it isn't running then select the instance using the tick box and then under "instance state" select "start instance" (fig. 48).

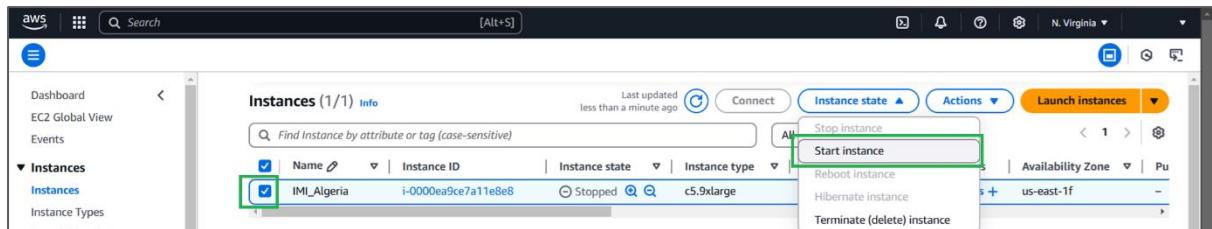


Figure 48:

Once running, with the instance selected, select “actions” and then “connect” (fig. 49)

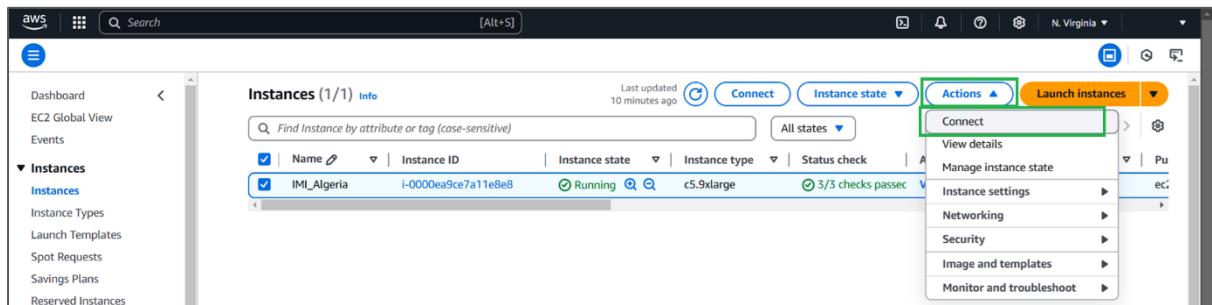


Figure 49:

In the connect to instance page, you have a list of instructions to follow using the Gitbash terminal you installed.

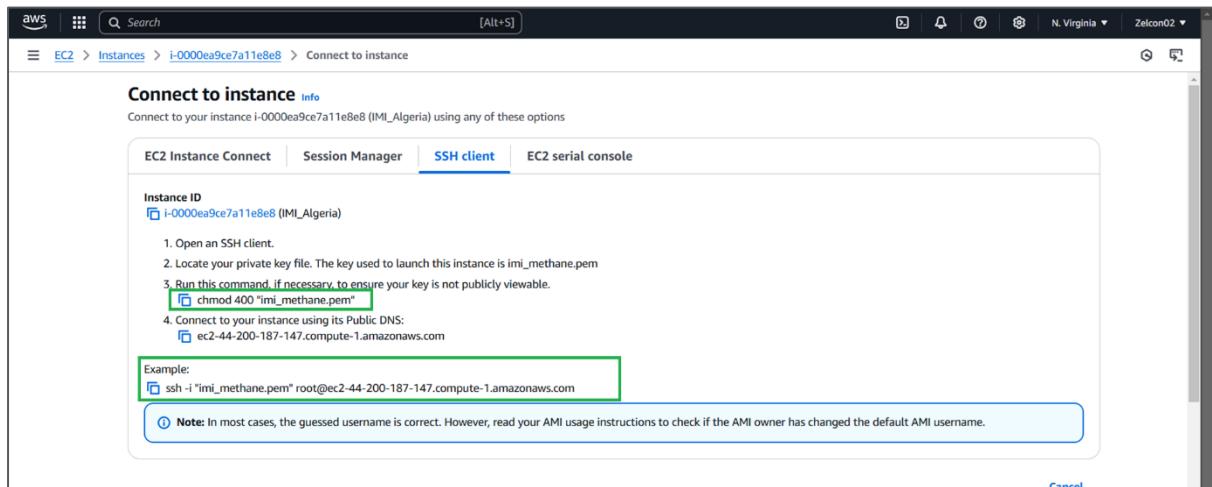


Figure 50:

Firstly change directory to where you keep your keypair by typing cd <path to your keypair folder> (fig. 51)

```
MINGW64:/c/GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ cd C:\GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
```

Figure 51:

Next you are going to enter in the command that ensures your keypair is not publicly viewable (fig. 52).

```

MINGW64:/c/GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ cd C:\GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ chmod 400 "imi_methane.pem"
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ |

```

Figure 52:

Finally you are going to take the long “example” code (at the bottom of fig. 50) and change the word “root” to “ubuntu”.

```

ssh -i "imi_methane.pem" root@ec2-44-200-187-147.compute-1.amazonaws.com
ssh -i "imi_methane.pem" ubuntu@ec2-44-200-187-147.compute-1.amazonaws.com

```

Then paste it into the gitbash terminal using the right click function of the mouse. Run the code and answer “yes” to the prompt, “are you sure you want to continue connecting?” (fig. 53). Once this has run, you should receive a “welcome to ubuntu” message.

```

MINGW64:/c/GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ cd C:\GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ chmod 400 "imi_methane.pem"
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ ssh -i "imi_methane.pem" ubuntu@ec2-44-200-187-147.compute-1.amazonaws.com
The authenticity of host 'ec2-44-200-187-147.compute-1.amazonaws.com (44.200.187.147)' can't be established.
ED25519 key fingerprint is SHA256:zGQ8D8dJ74RQJNEni/cABdUfcqPALojgrWGK7UvGPGI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes

```

Figure 53:

3.3.5 Configuring and running the IMI Preview

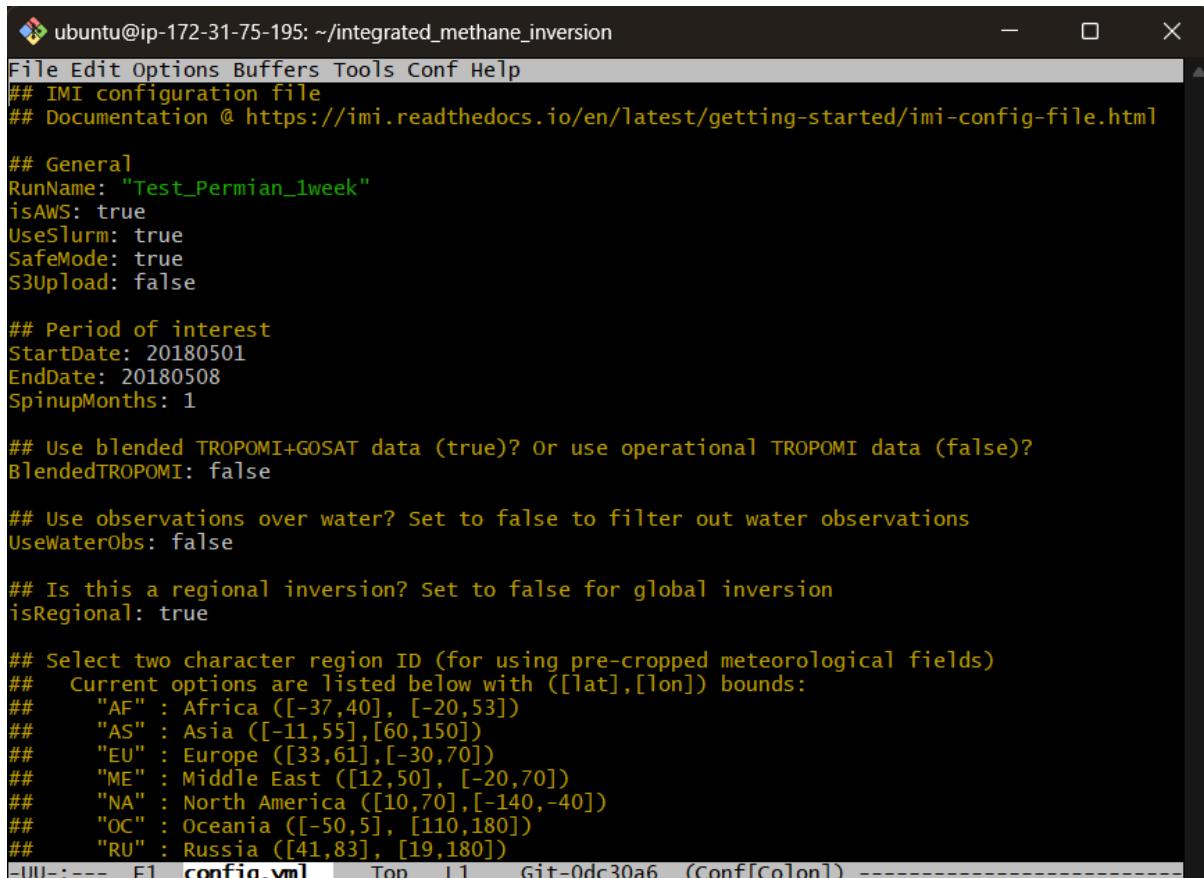
In the gitbash terminal, navigate to the IMI setup directory by typing:

```
cd ~/integrated_methane_inversion
```

Next open the inversion configuration file by typing the following:

```
emacs config.yml
```

The following screen should appear after a pause of a few moments (fig. 54)



```

ubuntu@ip-172-31-75-195: ~/integrated_methane_inversion
File Edit Options Buffers Tools Conf Help
## IMI configuration file
## Documentation @ https://imi.readthedocs.io/en/latest/getting-started/imi-config-file.html

## General
RunName: "Test_Permit_1week"
isAWS: true
UseSlurm: true
SafeMode: true
S3Upload: false

## Period of interest
StartDate: 20180501
EndDate: 20180508
SpinupMonths: 1

## Use blended TROPOMI+GOSAT data (true)? Or use operational TROPOMI data (false)?
BlendedTROPOMI: false

## Use observations over water? Set to false to filter out water observations
UseWaterObs: false

## Is this a regional inversion? Set to false for global inversion
isRegional: true

## Select two character region ID (for using pre-cropped meteorological fields)
## Current options are listed below with ([lat],[lon]) bounds:
##   "AF" : Africa ([-37,40], [-20,53])
##   "AS" : Asia ([-11,55],[60,150])
##   "EU" : Europe ([33,61],[-30,70])
##   "ME" : Middle East ([12,50], [-20,70])
##   "NA" : North America ([10,70],[-140,-40])
##   "OC" : Oceania ([-50,5], [110,180])
##   "RU" : Russia ([41,83], [19,180])
-UU--- F1 config.yml Top L1 Git-0dc30a6 (Conf[Colon]) -----

```

Figure 54:

By default, this will run the IMI preview which will provide among other things, the expected cost of running the full inversion.

Make a note of the Runname. This must be unique for every inversion you run. Slightly further down, you can modify the date fields (format: YYYYMMDD) to choose the start and end dates for the analysis (fig. 55).



```

ubuntu@ip-172-31-75-195: ~/integrated_methane_inversion
File Edit Options Buffers Tools Conf Help
## IMI configuration file
## Documentation @ https://imi.readthedocs.io/en/latest/getting-started/imi-config-file.html

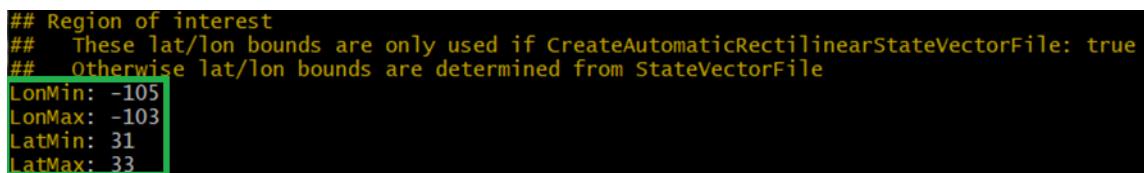
## General
RunName: "Test_Permit_1week"
isAWS: true
UseSlurm: true
SafeMode: true
S3Upload: false

## Period of interest
StartDate: 20180501
EndDate: 20180508
SpinupMonths: 1

```

Figure 55:

The bounding box settings are found further down the page in the “region of interest” section.



```

## Region of interest
## These lat/lon bounds are only used if CreateAutomaticRectilinearStateVectorFile: true
## Otherwise lat/lon bounds are determined from StateVectorFile
LonMin: -105
LonMax: -103
LatMin: 31
LatMax: 33

```

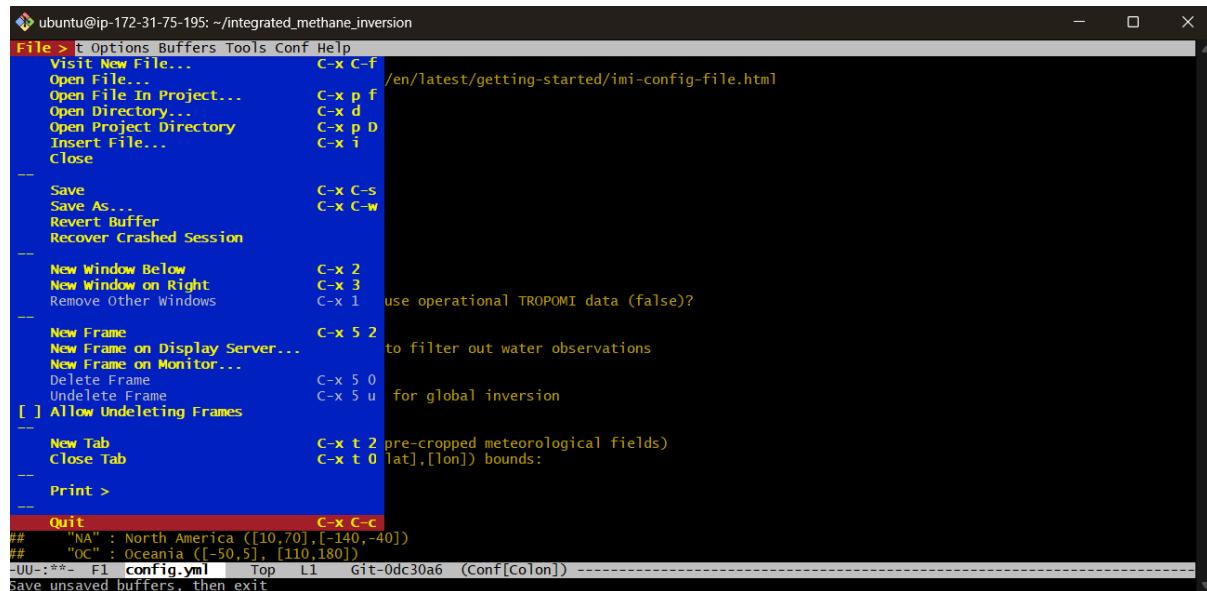
Figure 56:

Still further down is the IMI preview section which is by default set to “true”. You will change this to “false” when deciding to go ahead with the inversion.

```
## IMI preview
## NOTE: RunSetup must be true to run preview
DoPreview: true
DOFSThreshold: 0
```

Figure 57:

Once you are satisfied with the settings, press F10 on your keyboard and using the arrow keys, navigate to “Exit”, then press enter (fig. 58).

**Figure 58:**

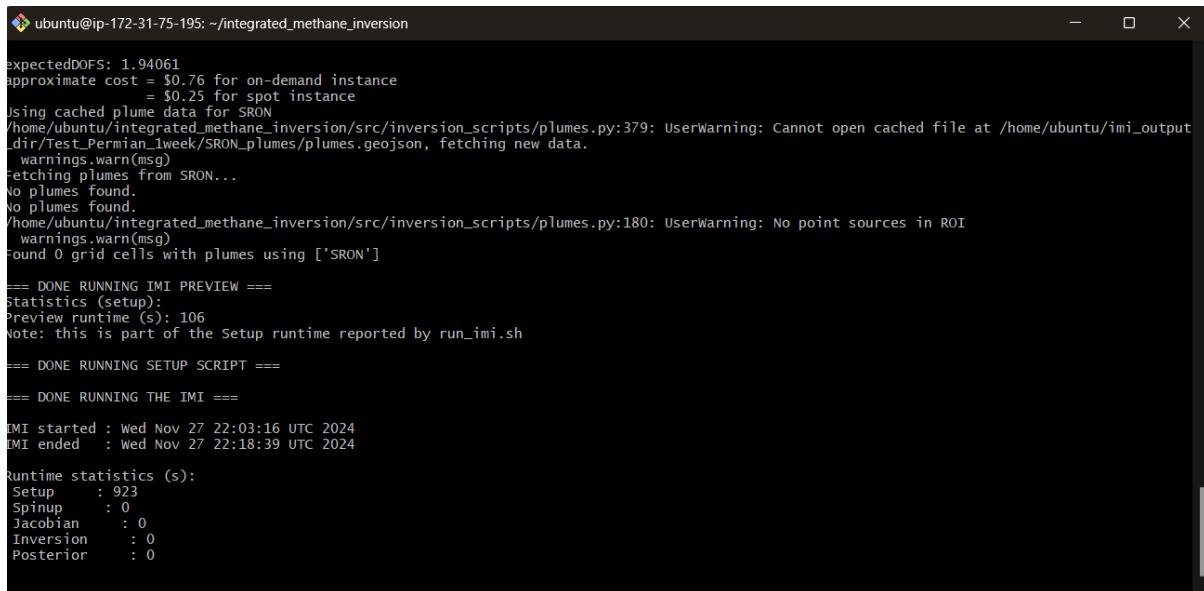
After exiting, run the following command to initiate the inversion preview:

sbatch run_imi.sh

and the following command to follow its progress:

tail --follow imi_output.log

Once the script has finished running it will look like figure 59 showing that the IMI preview for 1 week on the Permian Basin took around 15 minutes.



```
ubuntu@ip-172-31-75-195: ~/integrated_methane_inversion
expectedDOFS: 1.94061
approximate cost = $0.76 for on-demand instance
= $0.25 for spot instance
Using cached plume data for SRON
/home/ubuntu/integrated_methane_inversion/src/inversion_scripts/plumes.py:379: UserWarning: Cannot open cached file at /home/ubuntu/im_i_output_dir/Test_Permin_1week/SRON_plumes/plumes.geojson, fetching new data.
warnings.warn(msg)
Fetching plumes from SRON...
No plumes found.
No plumes found.
/home/ubuntu/integrated_methane_inversion/src/inversion_scripts/plumes.py:180: UserWarning: No point sources in ROI
warnings.warn(msg)
Found 0 grid cells with plumes using ['SRON']

== DONE RUNNING IMI PREVIEW ==
Statistics (setup):
Preview runtime (s): 106
Note: this is part of the Setup runtime reported by run_imi.sh

== DONE RUNNING SETUP SCRIPT ==

== DONE RUNNING THE IMI ==

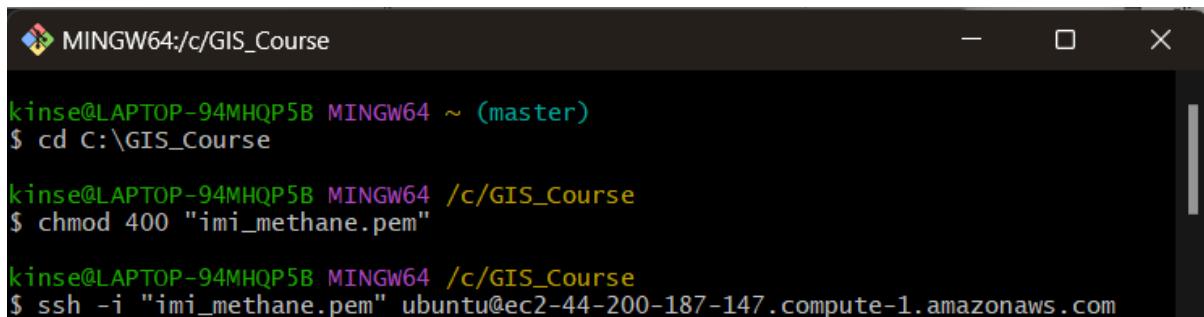
IMI started : Wed Nov 27 22:03:16 UTC 2024
IMI ended   : Wed Nov 27 22:18:39 UTC 2024

Runtime statistics (s):
Setup      : 923
Spinup     : 0
Jacobian   : 0
Inversion   : 0
Posterior  : 0
```

Figure 59: completed inversion process.

3.3.5 Viewing the IMI Preview data

Open a new gitbash terminal and log in as before (fig. 60)



```
MINGW64:/c/GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ cd C:\GIS_Course

kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ chmod 400 "imi_methane.pem"

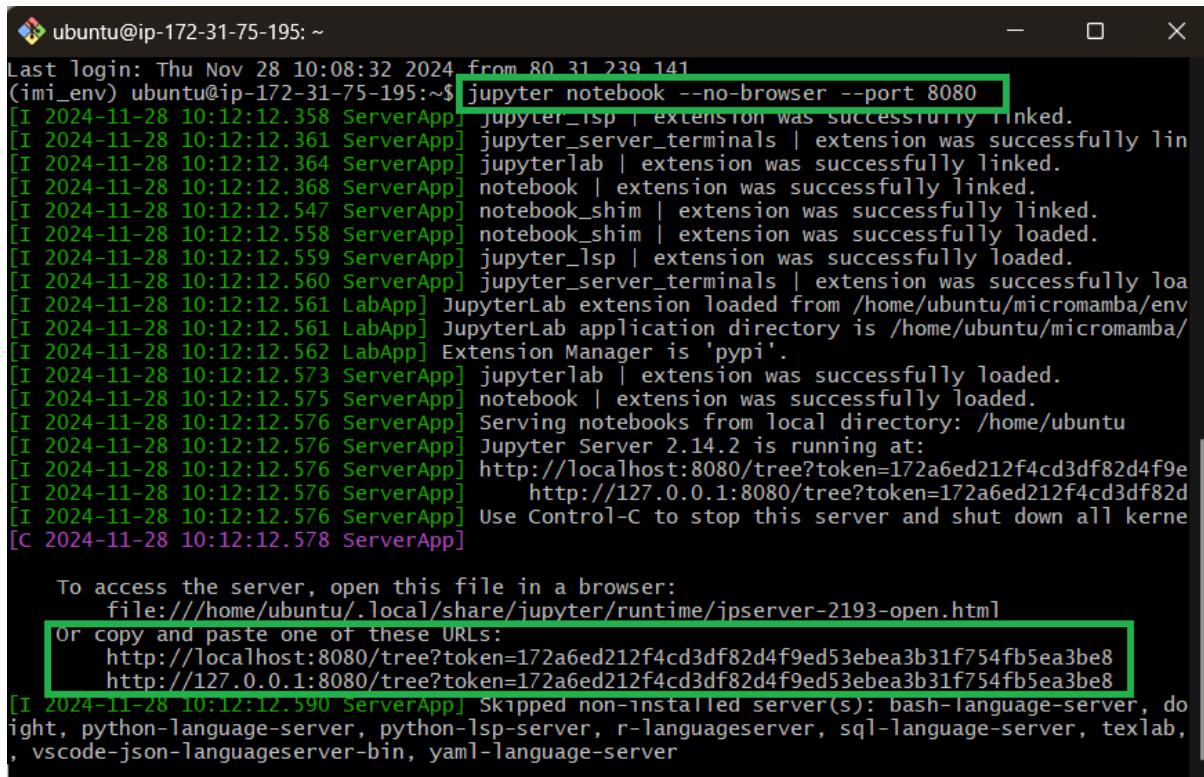
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ ssh -i "imi_methane.pem" ubuntu@ec2-44-200-187-147.compute-1.amazonaws.com
```

Figure 60:

Once you have logged into the instance, enter the code:

```
jupyter notebook --no-browser --port 8080
```

This starts a Jupyter server on port 8080 and will print out two URLs with an authentication token. This as shown in figure 61.



```

ubuntu@ip-172-31-75-195: ~
Last login: Thu Nov 28 10:08:32 2024 from 80.31.239.141
(iml_env) ubuntu@ip-172-31-75-195:~$ jupyter notebook --no-browser --port 8080
[I 2024-11-28 10:12:12.358 ServerApp] jupyter_lsp | extension was successfully linked.
[I 2024-11-28 10:12:12.361 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-11-28 10:12:12.364 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-11-28 10:12:12.368 ServerApp] notebook | extension was successfully linked.
[I 2024-11-28 10:12:12.547 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-11-28 10:12:12.558 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-11-28 10:12:12.559 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-11-28 10:12:12.560 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-11-28 10:12:12.561 LabApp] JupyterLab extension loaded from /home/ubuntu/micromamba/env
[I 2024-11-28 10:12:12.561 LabApp] JupyterLab application directory is /home/ubuntu/micromamba/
[I 2024-11-28 10:12:12.562 LabApp] Extension Manager is 'pypi'.
[I 2024-11-28 10:12:12.573 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-11-28 10:12:12.575 ServerApp] notebook | extension was successfully loaded.
[I 2024-11-28 10:12:12.576 ServerApp] Serving notebooks from local directory: /home/ubuntu
[I 2024-11-28 10:12:12.576 ServerApp] Jupyter Server 2.14.2 is running at:
[I 2024-11-28 10:12:12.576 ServerApp] http://localhost:8080/tree?token=172a6ed212f4cd3df82d4f9e
[I 2024-11-28 10:12:12.576 ServerApp] http://127.0.0.1:8080/tree?token=172a6ed212f4cd3df82d4f9e
[I 2024-11-28 10:12:12.576 ServerApp] Use Control-C to stop this server and shut down all kernel
[C 2024-11-28 10:12:12.578 ServerApp]

To access the server, open this file in a browser:
file:///home/ubuntu/.local/share/jupyter/runtime/jpserver-2193-open.html
or copy and paste one of these URLs:
http://localhost:8080/tree?token=172a6ed212f4cd3df82d4f9e53ebea3b31f754fb5ea3be8
http://127.0.0.1:8080/tree?token=172a6ed212f4cd3df82d4f9e53ebea3b31f754fb5ea3be8
[I 2024-11-28 10:12:12.590 ServerApp] Skipped non-installed server(s): bash-language-server, do
ight, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab,
, vscode-json-languageserver-bin, yaml-language-server

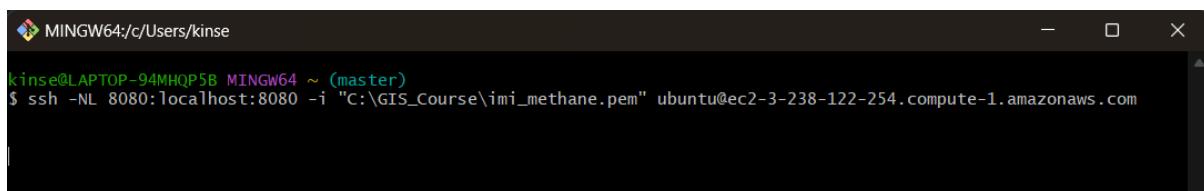
```

Figure 61:

Before you use these links you need to open an ssh tunnel from the ec2 instance to your local computer via port 8080. Open a new Gitbash terminal. Instead of logging into the instance as usual, you are going to use the following command to create the tunnel, changing the private key path to where your private key is stored on your local machine, and the host name to the same as the one shown in figure 50 on your connect to instance page.

Format: ssh -NL 8080:localhost:8080 -i /path/to/private_key ubuntu@<host-name>

Example: ssh -NL 8080:localhost:8080 -i "C:\GIS_Course\imi_methane.pem" ubuntu@ec2-3-238-122-254.compute-1.amazonaws.com



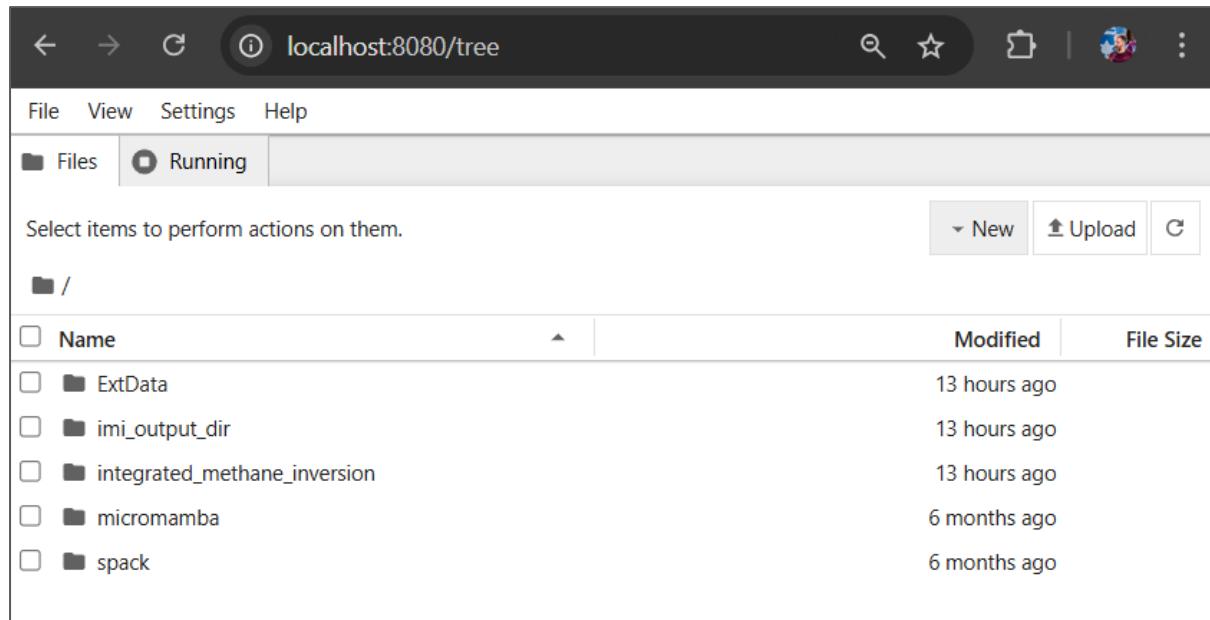
```

MINGW64:/c/Users/kinse
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ ssh -NL 8080:localhost:8080 -i "C:\GIS_Course\imi_methane.pem" ubuntu@ec2-3-238-122-254.compute-1.amazonaws.com

```

Figure 62:

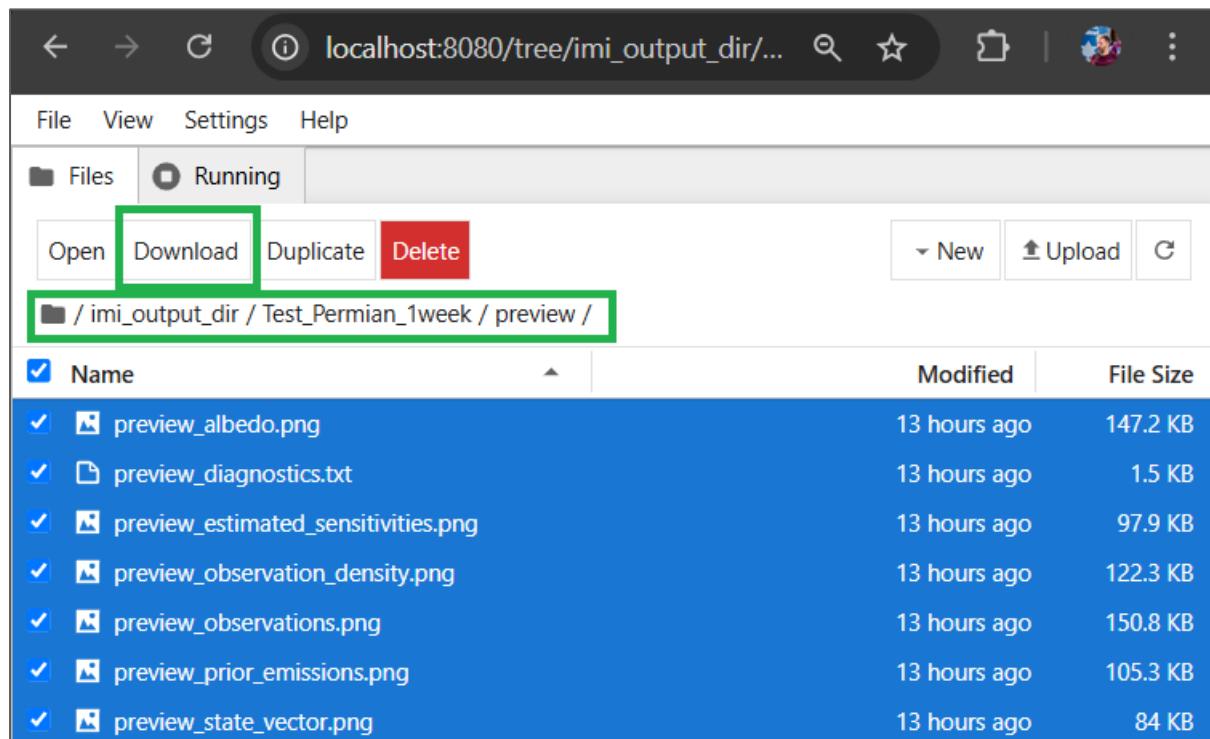
Once you do this, nothing will seem to have happened but in the background the tunnel will have been created. You can now use one of the two URLs provided in figure 61 to access and view the IMI preview data (fig. 63).

**Figure 63:**

Once you have access to the file directory shown above, navigate to:

/imi_output_dir/Test_Permit_1week/preview/

From there you can download and view the IMI preview data

**Figure 64:**

The IMI preview data should look something like the following (fig. 65).

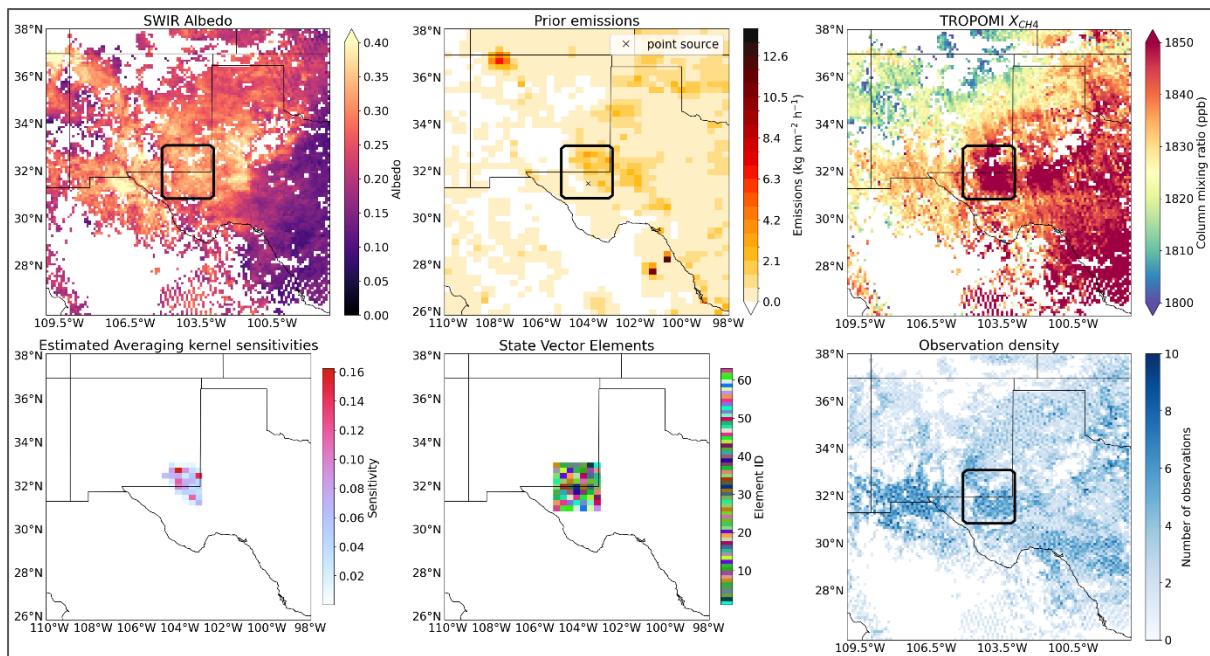


Figure 65:

And the file “preview_diagnostics.txt” gives you an estimate of the cost of running the inversion (fig. 66).

```
preview_diagnostics.txt
File Edit View
##approximate cost = $0.76 for on-demand instance
##                 = $0.25 for spot instance
##Total prior emissions in region of interest = 0.3917637345532142 Tg/y
##Found 38858.0 observations in the region of interest
##k = [1.25903 1.25903 1.25903 1.25903 1.25903 1.25903 1.25903 1.25903
      1.25903 1.25903 1.25903 1.25903 1.25903 1.25903 1.25903 1.25903]
```

Figure 66:

3.3.6 Configuring and running the IMI.

If the IMI preview looks correct and the cost is within your budget, you will now proceed with running the IMI. Open a new Gitbash terminal and connect to your instance as before (fig. 67).

```
MINGW64:c/GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ cd C:\GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ chmod 400 "imi_methane.pem"
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ ssh -i "imi_methane.pem" ubuntu@ec2-44-200-187-147.compute-1.amazonaws.com
The authenticity of host 'ec2-44-200-187-147.compute-1.amazonaws.com (44.200.187
.147)' can't be established.
ED25519 key fingerprint is SHA256:zGQ8D8dJ74RQJNEni/cABdUfCqPALojgrWGK7UvGPGI.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Figure 67:

In the Gitbash terminal, navigate to the IMI setup directory by typing:

```
cd ~/integrated_methane_inversion
```

Next open the inversion configuration file by typing the following:

```
emacs config.yml
```

The following screen should appear after a pause of a few moments (fig. 68)

```
ubuntu@ip-172-31-75-195: ~/integrated_methane_inversion
File Edit Options Buffers Tools Conf Help
## IMI configuration file
## Documentation @ https://imi.readthedocs.io/en/latest/getting-started/imi-config-file.html

## General
RunName: "Test_Permit_1week"
isAWS: true
UseSlurm: true
SafeMode: true
S3Upload: false

## Period of interest
StartDate: 20180501
EndDate: 20180508
SpinupMonths: 1

## Use blended TROPOMI+GOSAT data (true)? Or use operational TROPOMI data (false)?
BlendedTROPOMI: false

## Use observations over water? Set to false to filter out water observations
UseWaterObs: false

## Is this a regional inversion? Set to false for global inversion
isRegional: true

## Select two character region ID (for using pre-cropped meteorological fields)
## Current options are listed below with ([lat],[lon]) bounds:
## "AF" : Africa ([-37,40], [-20,53])
## "AS" : Asia ([-11,55],[60,150])
## "EU" : Europe ([33,61],[-30,70])
## "ME" : Middle East ([12,50], [-20,70])
## "NA" : North America ([10,70],[-140,-40])
## "OC" : Oceania ([-50,5], [110,180])
## "RU" : Russia ([41,83], [19,180])
-UU--- F1 config.yml Top L1 Git-0dc30a6 (Conf[Colon]) -----
```

Figure 68:

Before you changed the following fields to match the area and time you were interested in.

- StartDate
- EndDate
- LongMin
- LongMax
- LatMin
- LatMax

Now you are going to set the following fields to the values below to run the IMI instead of the IMI preview.

```
## Setup modules
## Turn on/off different steps in setting up the inversion
    • RunSetup: true
    • SetupTemplateRundir: false
    • SetupSpinupRun: true
```

- SetupJacobianRuns: true
- SetupInversion: true
- SetupPosteriorRun: true

```

## Run modules

## Turn on/off different steps in performing the inversion



- DoHemcoPriorEmis: false
- DoSpinup: true
- DoJacobian: true
- ReDoJacobian: true
- DoInversion: true
- DoPosterior: true



## IMI preview

## NOTE: RunSetup must be true to run preview



- DoPreview: false

```

Once you have adjusted those settings, press F10 and save the changes (fig. 69), then press F10 again and Quit to return to the console.

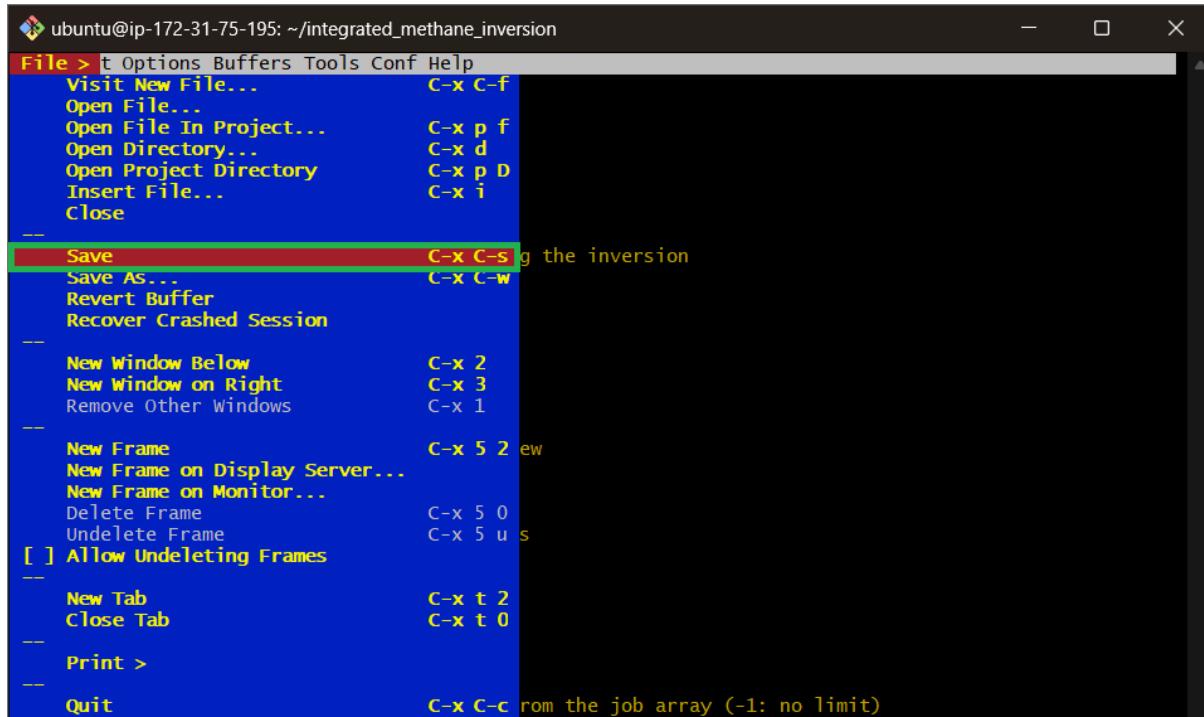


Figure 69:

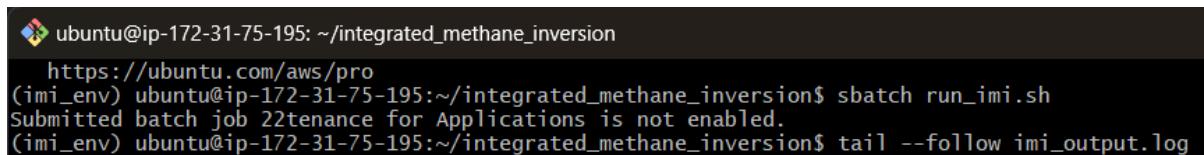
In the Gitbash terminal issue this command to run the IMI:

sbatch run_imi.sh

As before you can follow its progress by using the following command:

tail --follow imi_output.log

As shown in figure 70.



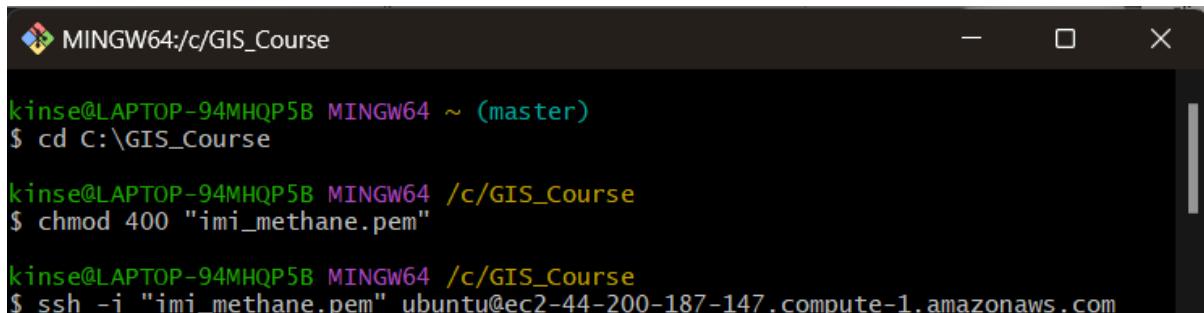
```
ubuntu@ip-172-31-75-195: ~/integrated_methane_inversion
https://ubuntu.com/aws/pro
(imi_env) ubuntu@ip-172-31-75-195:~/integrated_methane_inversion$ sbatch run_imi.sh
Submitted batch job 22tenance for Applications is not enabled.
(imi_env) ubuntu@ip-172-31-75-195:~/integrated_methane_inversion$ tail --follow imi_output.log
```

Figure 70:

At this point you can disconnect from your instance as the IMI will take hours to run (**state quantity for the example**).

3.3.7 Accessing the IMI data

Open a new Gitbash terminal and log in to your instance (fig. 71)



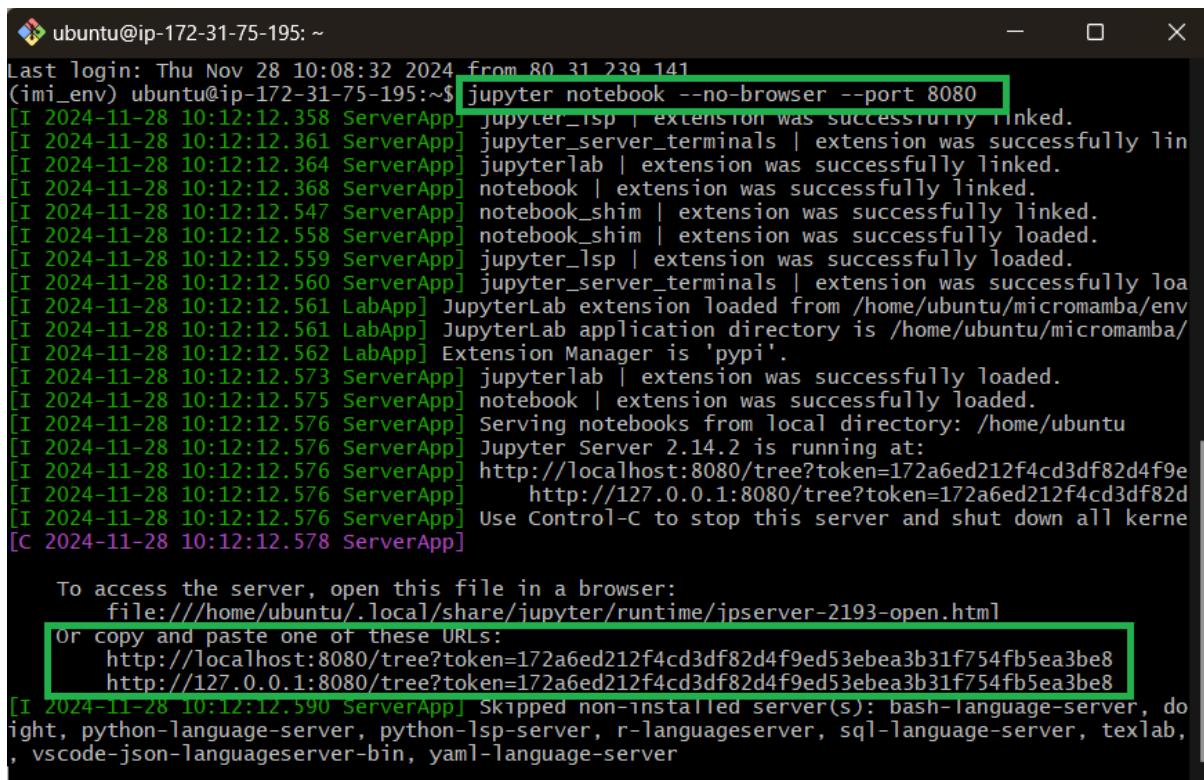
```
MINGW64:/c/GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ cd C:\GIS_Course
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ chmod 400 "imi_methane.pem"
kinse@LAPTOP-94MHQP5B MINGW64 /c/GIS_Course
$ ssh -i "imi_methane.pem" ubuntu@ec2-44-200-187-147.compute-1.amazonaws.com
```

Figure 71:

Once you have logged into the instance, enter the code:

jupyter notebook --no-browser --port 8080

This starts a Jupyter server on port 8080 and will print out two URLs with an authentication token. This as shown in figure 72.



```
Last login: Thu Nov 28 10:08:32 2024 from 80.31.239.141
(imi_env) ubuntu@ip-172-31-75-195:~$ jupyter notebook --no-browser --port 8080
[I 2024-11-28 10:12:12.358 ServerApp] jupyter_rsp | extension was successfully linked.
[I 2024-11-28 10:12:12.361 ServerApp] jupyter_server_terminals | extension was successfully linked.
[I 2024-11-28 10:12:12.364 ServerApp] jupyterlab | extension was successfully linked.
[I 2024-11-28 10:12:12.368 ServerApp] notebook | extension was successfully linked.
[I 2024-11-28 10:12:12.547 ServerApp] notebook_shim | extension was successfully linked.
[I 2024-11-28 10:12:12.558 ServerApp] notebook_shim | extension was successfully loaded.
[I 2024-11-28 10:12:12.559 ServerApp] jupyter_lsp | extension was successfully loaded.
[I 2024-11-28 10:12:12.560 ServerApp] jupyter_server_terminals | extension was successfully loaded.
[I 2024-11-28 10:12:12.561 LabApp] JupyterLab extension loaded from /home/ubuntu/micromamba/env
[I 2024-11-28 10:12:12.561 LabApp] JupyterLab application directory is /home/ubuntu/micromamba/
[I 2024-11-28 10:12:12.562 LabApp] Extension Manager is 'pypi'.
[I 2024-11-28 10:12:12.573 ServerApp] jupyterlab | extension was successfully loaded.
[I 2024-11-28 10:12:12.575 ServerApp] notebook | extension was successfully loaded.
[I 2024-11-28 10:12:12.576 ServerApp] Serving notebooks from local directory: /home/ubuntu
[I 2024-11-28 10:12:12.576 ServerApp] Jupyter Server 2.14.2 is running at:
[I 2024-11-28 10:12:12.576 ServerApp] http://localhost:8080/tree?token=172a6ed212f4cd3df82d4f9e
[I 2024-11-28 10:12:12.576 ServerApp] http://127.0.0.1:8080/tree?token=172a6ed212f4cd3df82d4f9e
[I 2024-11-28 10:12:12.576 ServerApp] Use Control-C to stop this server and shut down all kernels...
[C 2024-11-28 10:12:12.578 ServerApp]

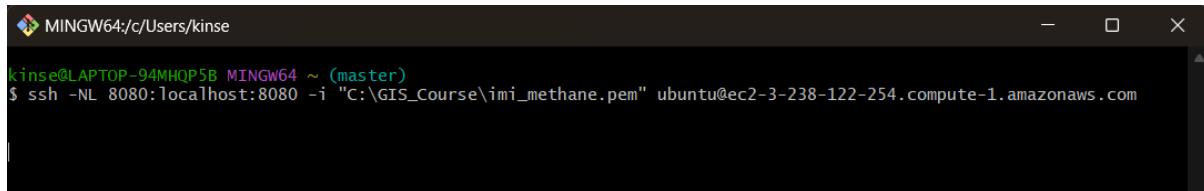
To access the server, open this file in a browser:
file:///home/ubuntu/.local/share/jupyter/runtime/jpserver-2193-open.html
Or copy and paste one of these URLs:
http://localhost:8080/tree?token=172a6ed212f4cd3df82d4f9ed53ebea3b31f754fb5ea3be8
http://127.0.0.1:8080/tree?token=172a6ed212f4cd3df82d4f9ed53ebea3b31f754fb5ea3be8
[I 2024-11-28 10:12:12.590 ServerApp] Skipped non-installed server(s): bash-language-server, do
ight, python-language-server, python-lsp-server, r-languageserver, sql-language-server, texlab,
, vscode-json-languageserver-bin, yaml-language-server
```

Figure 72:

Before you use these links you need to open an ssh tunnel from the ec2 instance to your local computer via port 8080. Open a new Gitbash terminal. Instead of logging into the instance as usual, you are going to use the following command to create the tunnel, changing the private key path to where your private key is stored on your local machine, and the host name to the same as the one shown in figure 50 on your connect to instance page.

Format: ssh -NL 8080:localhost:8080 -i /path/to/private_key ubuntu@<host-name>

Example: ssh -NL 8080:localhost:8080 -i "C:\GIS_Course\imi_methane.pem" ubuntu@ec2-3-238-122-254.compute-1.amazonaws.com



```
MINGW64:/c/Users/kinsel
kinse@LAPTOP-94MHQP5B MINGW64 ~ (master)
$ ssh -NL 8080:localhost:8080 -i "C:\GIS_Course\imi_methane.pem" ubuntu@ec2-3-238-122-254.compute-1.amazonaws.com
```

Figure 73:

Once you do this, nothing will seem to have happened in this Gitbash window but there but in the background the tunnel will have been created. You can now use one of the two URLs provided in figure 61 to access and view the IMI preview data (fig. 74).

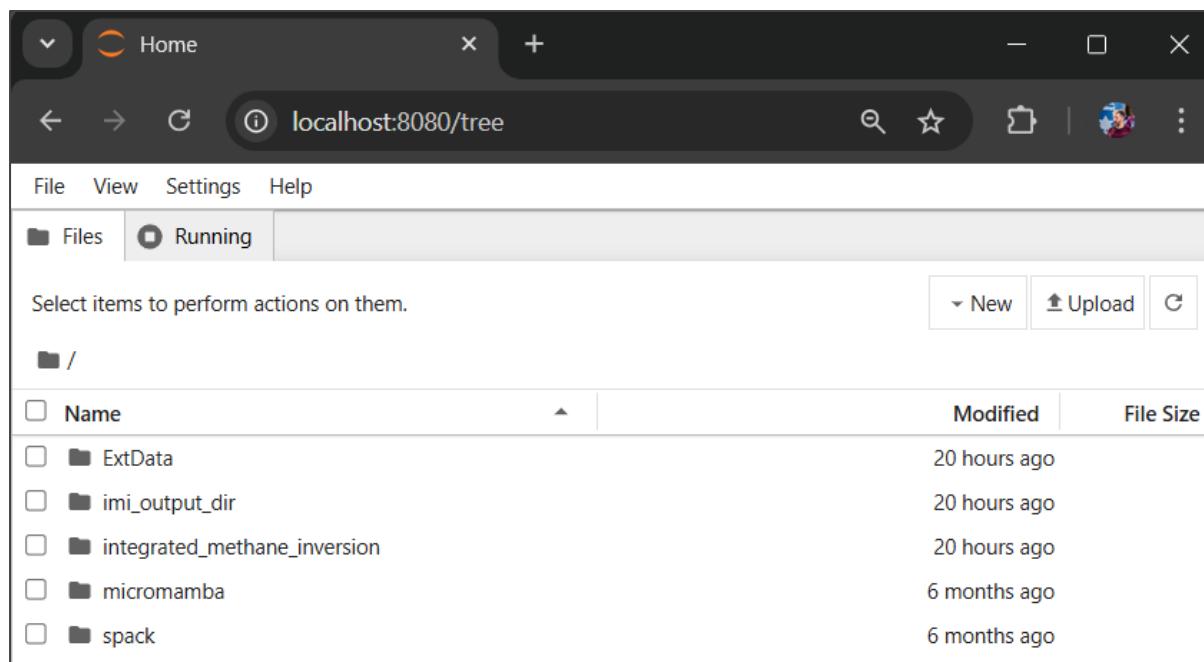
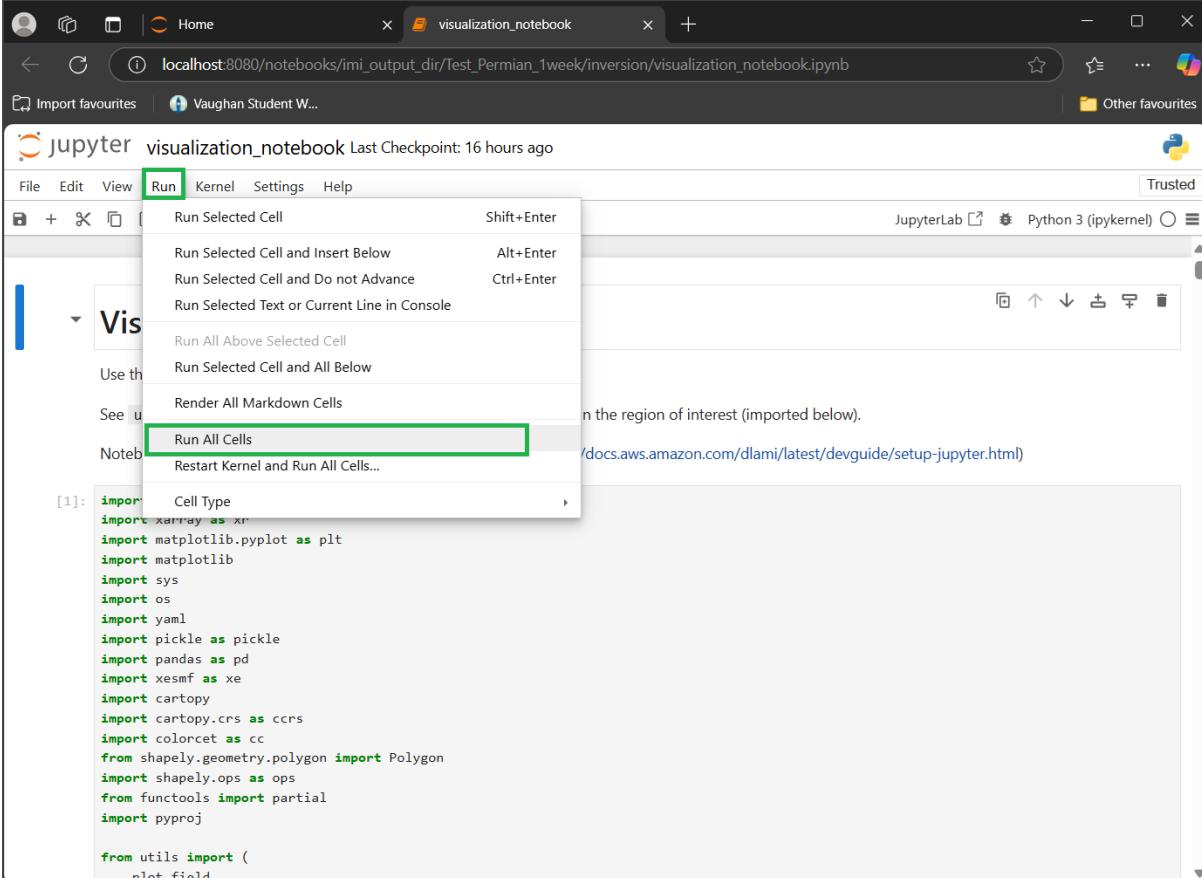


Figure 73:

Navigate to:

/imi_output_dir/Test_Permian_1week/inversion/

and look for "visualization_notebook.ipynb". Open this and a Jupyter notebook will appear. In this notebook, select from the menu "Run" and then "Run all cells" (fig. 74).



The screenshot shows a Jupyter Notebook window titled "visualization_notebook". The "Run" menu is open, and the option "Run All Cells" is highlighted with a green box. The code cell [1] contains several import statements and function definitions. The output area shows a link to the Jupyter documentation.

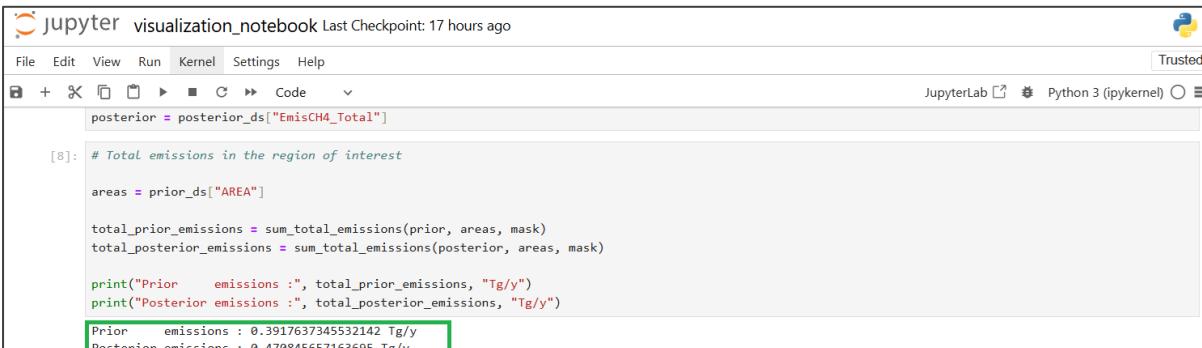
```
[1]: import Cell Type
import xarray as xr
import matplotlib.pyplot as plt
import matplotlib
import sys
import os
import yaml
import pickle as pickle
import pandas as pd
import xesmf as xe
import cartopy
import cartopy.crs as ccrs
import colorcet as cc
from shapely.geometry.polygon import Polygon
import shapely.ops as ops
from functools import partial
import pyproj

from utils import (
    plot_field,
```

Figure 74:

3.3.8 Interpreting the IMI data

Once this is complete, scroll down to view the data. The first point of interest is code box 8 which shows the previous estimate for emissions based on official inventories (Prior) and the measured emissions based on Sentinel-5P and IMI (Posterior). As can be seen in figure 75, the official “prior” estimate is 0.079 Teragrams per year (Tg/y) below the IMI measured figure, which is 79,000 metric tonnes.



The screenshot shows a Jupyter Notebook window titled "visualization_notebook". Code box 8 displays the calculation of total emissions. The output shows the Prior emissions as 0.3917637345532142 Tg/y and the Posterior emissions as 0.470845657163695 Tg/y. The output is highlighted with a green box.

```
[8]: posterior = posterior_ds["EmisCH4_Total"]

# Total emissions in the region of interest

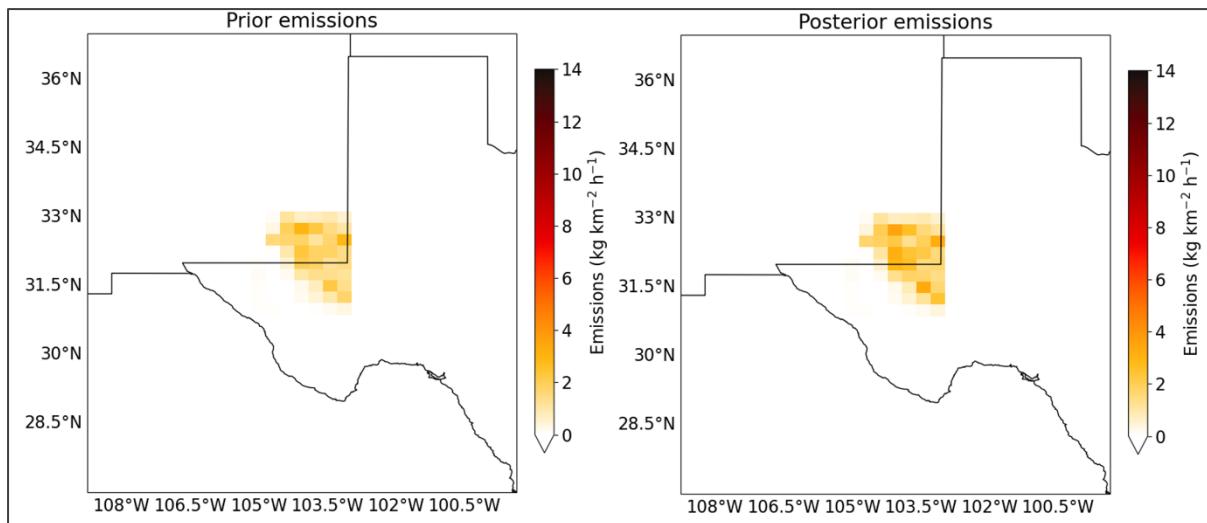
areas = prior_ds["AREA"]

total_prior_emissions = sum_total_emissions(prior, areas, mask)
total_posterior_emissions = sum_total_emissions(posterior, areas, mask)

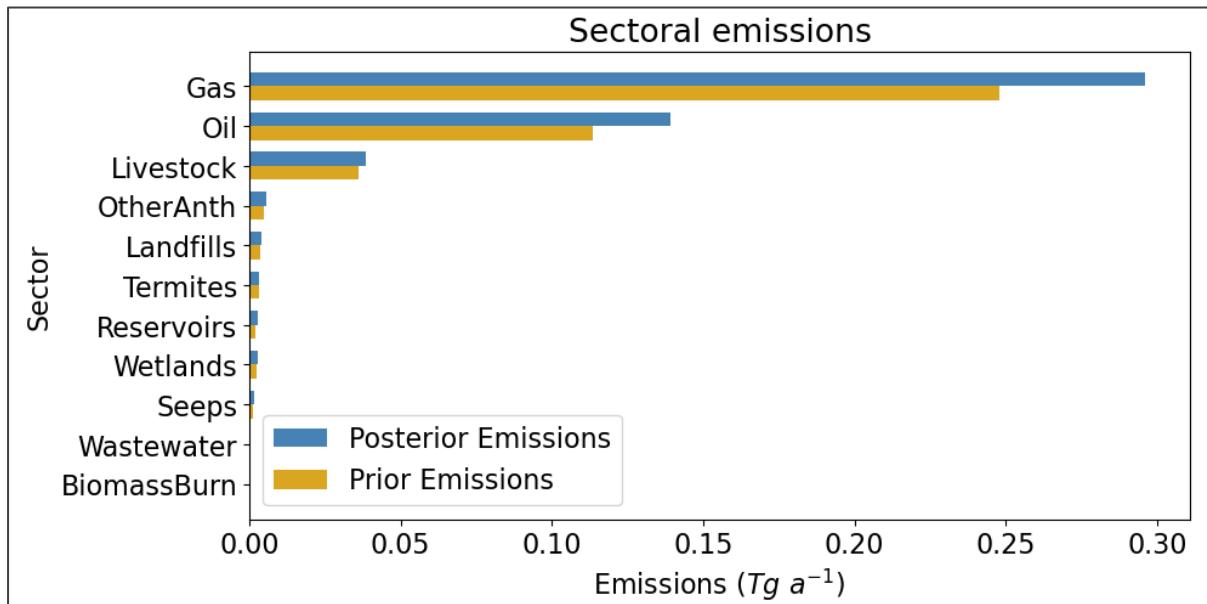
print("Prior emissions : ", total_prior_emissions, "Tg/y")
print("Posterior emissions : ", total_posterior_emissions, "Tg/y")
```

Figure 75:

Code box 9 further visualises this and shows the official “prior” estimates on a map, whereas code box 10 shows the mapped IMI measured “posterior” emissions (fig. 76).

**Figure 76:**

Code box 11 shows the sectoral emissions. This is determined by applying the ratios given by the official “prior” estimates to the IMI results (fig. 77).

**Figure 77:**

If you wish to download any of the figures from the visualisation notebook, you can do so by navigating to:

/imi_output_dir/Test_Permian_1week/inversion/output/

Selecting the tick boxes for the figures and then clicking the download button (fig. 78)

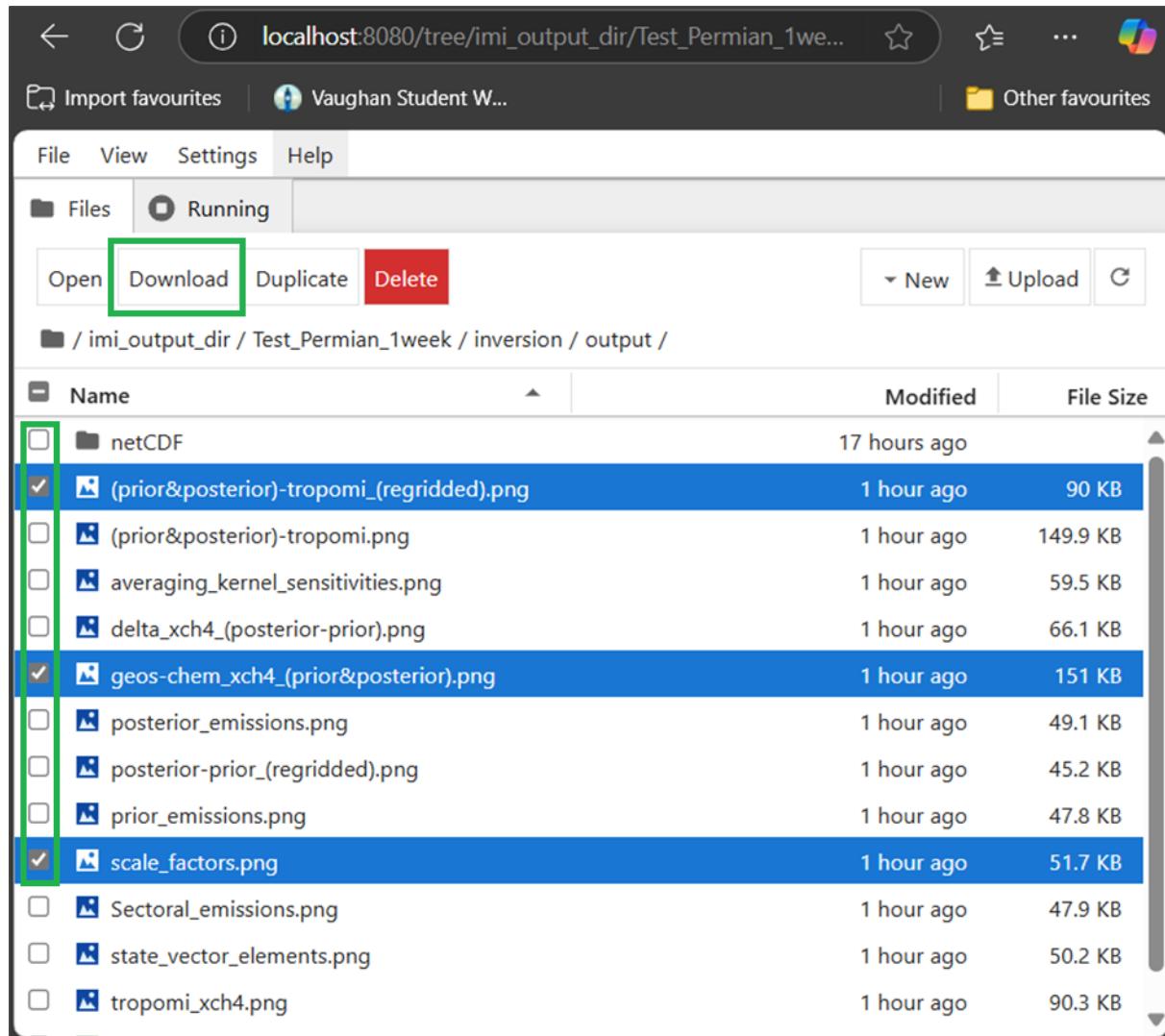


Figure 77:

3.3.9 Shutting down the instance

Once you have run the inversion and collected the data you need, you must shut down the inversion to avoid any unnecessary costs. Open a browser and navigate to <https://aws.amazon.com/>, then login to your console using the user details you set up previously (fig. 78).

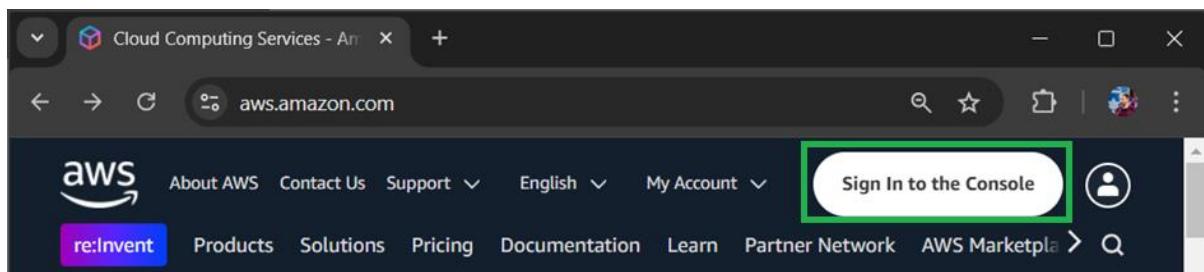


Figure 78:

Once you have logged into your console, click on the EC2 Global view link (fig. 79).

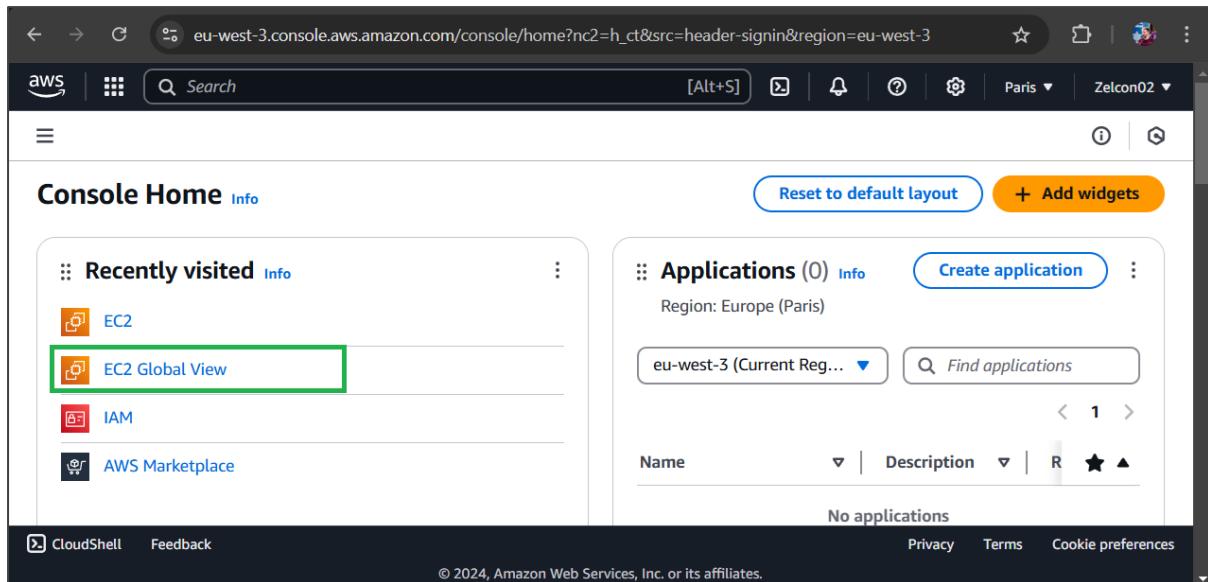


Figure 79:

You should see under “instances” 1 in 1 region (or however many instances you set up). Click on that link (fig. 80) and select “view all”.

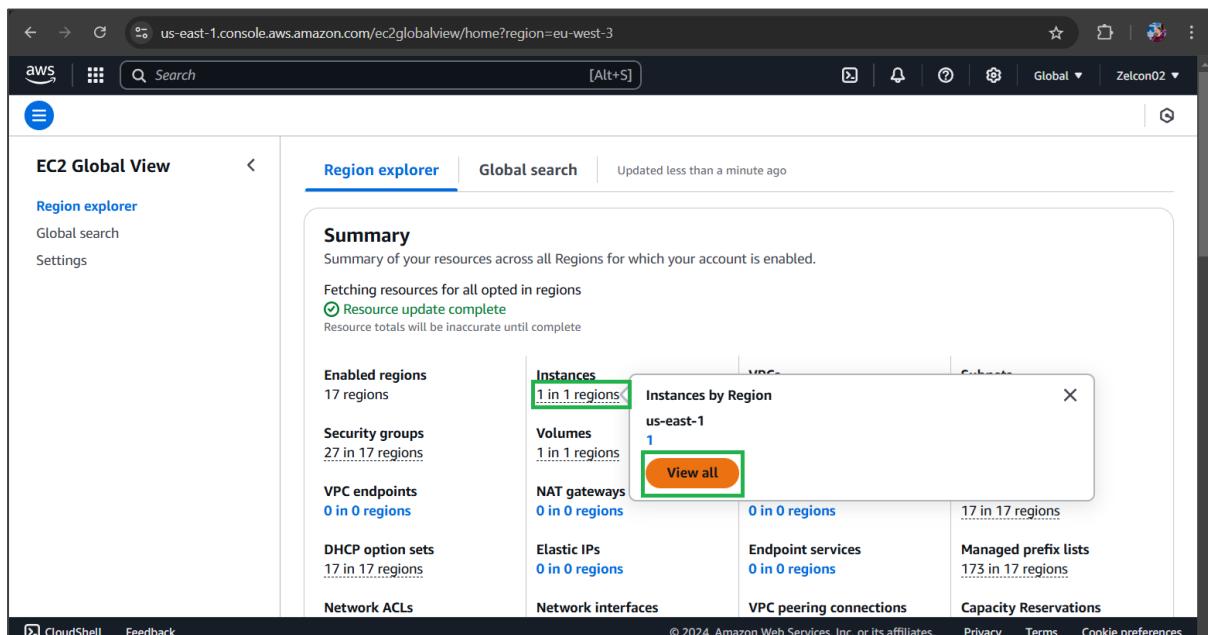


Figure 80:

Next click on the blue “resource ID” link (fig. 81)

The screenshot shows the AWS EC2 Global View interface. On the left, there's a sidebar with 'EC2 Global View' and links for 'Region explorer', 'Global search' (which is selected), and 'Settings'. The main area has tabs for 'Region explorer' and 'Global search'. Below the tabs, it says 'Updated 1 minute ago'. A section titled 'Global search (1)' is shown, with a note: 'Perform a global search to search for specific resources across all Regions for which your account is enabled'. There are buttons for 'Download CSV' and 'Manage'. A search bar contains the placeholder 'Find resources by attribute or tag'. Below the search bar are filters: 'Resource Type = Instance' (selected) and 'Clear filters'. A table lists one instance: Name 'IMI_Algeria', Resource ID 'i-076065ecc9406cb9' (highlighted with a green border), Resource Type 'Instance', and Region 'us-east-1'.

Figure 81:

Finally open the “instance state” dropdown menu and select “stop instance” (fig. 82). Thereafter you should receive a message confirming the operation.

The screenshot shows the AWS EC2 Instances page. The left sidebar has 'Instances' expanded, showing 'Instances', 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', and 'Dedicated Hosts'. The main area shows an 'Instance summary for i-076065ecc9406cb9 (IMI_Algeria)'. It includes a 'Connect' button, an 'Instance state' dropdown (set to 'Running'), and an 'Actions' dropdown. The 'Actions' dropdown is open, showing options: 'Stop instance' (highlighted with a green border), 'Start instance', 'Reboot instance', 'Hibernate instance', and 'Terminate (delete) instance'. To the right, there's information about private and public IP addresses and DNS names.

Figure 82:

3.5 Troubleshooting

If you are having problems running the IMI, please submit a ticket at the following Github page where the developers will attempt to help you out.

https://github.com/geoschem/integrated_methane_inversion/issues

4. References

- Abada, Z. and Bouharkat, M., 2018. Study of management strategy of energy resources in Algeria. *Energy Reports*, 4, pp.1-7. Available at: https://www.researchgate.net/publication/328663133_Study_of_management_strategy_of_energy_resources_in_Algeria [Accessed 18 Sep. 2024].
- European Commission, 2023. EU announces €175m financial support to reduce methane emissions at COP28. *IP/23/6057*. Available at: https://ec.europa.eu/commission/presscorner/detail/en/IP_23_6057
- Ferronato, N., Torretta, V., Ragazzi, M., & Rada, E.C. (2017). Waste mismanagement in developing countries: A case study of environmental contamination. *UPB Sci. Bull.*, 79(2), 185–196.
- Google Earth. (2024). Satellite view of Algerian petrochemical facility. Retrieved 10th of May 2024.
- Integrated Methane Inversion (2024) *IMI processing steps*, Harvard University. Available at: <https://imi.seas.harvard.edu/overview> (Accessed: 27 November 2024).
- International Energy Agency, 2022. Global Methane Tracker 2022. *IEA, Paris*. Available at: <https://www.iea.org/reports/global-methane-tracker-2022>
- International Energy Agency, 2024. Global Methane Tracker 2024: Tracking pledges, targets, and action. Available at: <https://www.iea.org/reports/global-methane-tracker-2024/tracking-pledges-targets-and-action> [Accessed 17 Sep. 2024].
- Jet Propulsion Laboratory. (2024). VISIONS: The EMIT Open Data Portal. Retrieved 10th of May 2024 from: <https://tinyurl.com/2s4kkwe5>
- Jackson, R.B., Saunois, M., Bousquet, P., Canadell, J.G., Poulter, B., Stavert, A.R., Bergamaschi, P., Niwa, Y., Segers, A., and Tsuruta, A., 2020. Increasing anthropogenic methane emissions arise equally from agricultural and fossil fuel sources. *Environmental Research Letters*, 15(7), p.071002. Available at: <https://doi.org/10.1088/1748-9326/ab9ed2> [Accessed 4 Jul. 2024].
- Liu, M., Van Der A, R., Van Weele, M., Eskes, H., Lu, X., Veefkind, P., De Laat, J., Kong, H., Wang, J., Sun, J. and Ding, J., 2021. A new divergence method to quantify methane emissions using observations of Sentinel-5P TROPOMI. *Geophysical Research Letters*, 48(18), p.e2021GL094151.
- Lorente, A., Borsdorff, T., Butz, A., Hasekamp, O., Aan De Brugh, J., Schneider, A., Wu, L., Hase, F., Kivi, R., Wunch, D. and Pollard, D.F., 2021. Methane retrieved from TROPOMI: Improvement of the data product and validation of the first 2 years of measurements. *Atmospheric Measurement Techniques*, 14(1), pp.665–684.
- Naus, S., Maasakkers, J.D., Gautam, R., Omara, M., Stikker, R., Veenstra, A.K., Nathan, B., Irakulis-Loitxate, I., Guanter, L., Pandey, S. and Girard, M., 2023. Assessing the relative importance of satellite-detected methane superemitters in quantifying total emissions for oil and gas production areas in Algeria. *Environmental Science & Technology*, 57(48), pp.19545-19556.
- Malley, C.S., Borgford-Parnell, N., Haeussling, S., Howard, I.C., Lefèvre, E.N. and Kuylenstierna, J.C., 2023. A roadmap to achieve the global methane pledge. *Environmental Research: Climate*, 2(1), p.011003.
- McNabb, R. (2024). Reprojecting Rasters Using Rasterio [Blog post]. Retrieved 24th of April 2024 from <https://iamdonovan.github.io/teaching/egm722/practicals/raster.html>
- Microsoft Copilot Designer. (2024). Cover artwork. Generated on 25th of November 2024
- OpenEO. (2024). NDVI Timeseries. [Online]. Retrieved 22nd of April 2024, from: https://documentation.dataspace.copernicus.eu/notebook-samples/openeo/NDVI_Timeseries.html

Pandey, S., van Nistelrooij, M., Maasakkers, J.D., Sutar, P., Houweling, S., Varon, D.J., Tol, P., Gains, D., Worden, J., & Aben, I. (2023). Daily detection and quantification of methane leaks using Sentinel-3: a tiered satellite observation approach with Sentinel-2 and Sentinel-5p. *Remote Sensing of Environment*, 296, 113716.

Parker, R., Boesch, H., Cogan, A., Fraser, A., Feng, L., Palmer, P.I., Messerschmidt, J., Deutscher, N., Griffith, D.W., Notholt, J., & Wennberg, P.O. (2011). Methane observations from the Greenhouse Gases Observing SATellite: Comparison to ground based TCCON data and model calculations. *Geophysical Research Letters*, 38(15).

Rosenthal, J., 2023. In Reverse: Natural Gas and Politics in the Maghreb and Europe. *FPRI: Foreign Policy Research Institute*. United States of America. Available at: <https://coilink.org/20.500.12592/bf2q0b> [Accessed 29 September 2024].

Sentinel Hub. (2024) "About Sentinel-2 Data." Retrieved 5th of May 2024, from: <https://docs.sentinel-hub.com/api/latest/data/sentinel-2-l2a/>

Sentinel Hub. (2024) "About Sentinel-5P Data." Retrieved 23rd April 2024, from: <https://docs.sentinel-hub.com/api/latest/data/sentinel-5p-l2/>

Sonneveld, E. (2024). Obtaining a list of available Sentinel 2 datasets for a location. Copernicus Dataspace Forum. Retrieved 25th of April 2024 from <https://tinyurl.com/4t2trevy>

Talamali, S., Chaouchi, R. and Benayad, S., 2016. Sedimentological evolution of the Lower Series formation in the southern area of the Hassi R'Mel field, Saharan Platform, Algeria. *Arabian Journal of Geosciences*, 9, p.481. DOI: 10.1007/s12517-016-2506-7.

United Nations Development Programme, 2022. Country Programme Document for Algeria (2023-2027). DP/DCP/DZA/4. Available at: <https://www.undp.org/sites/g/files/zskgke326/files/2023-03/CPD%20Algeria%202023-2027.pdf> [Accessed 17 September 2024].

Varon, D. J., Jervis, D., McKeever, J., Spence, I., Gains, D., & Jacob, D. J. (2021). High-frequency monitoring of anomalous methane point sources with multispectral Sentinel-2 satellite observations. *Atmospheric Measurement Techniques*, 14, 2771–2785.

Varon, D.J., Jacob, D.J., Sulprizio, M., Estrada, L.A., Downs, W.B., Shen, L., Hancock, S.E., Nesser, H., Qu, Z., Penn, E., Chen, Z., Lu, X., Lorente, A., Tewari, A. and Randles, C.A., 2022. Integrated Methane Inversion (IMI 1.0): a user-friendly, cloud-based facility for inferring high-resolution methane emissions from TROPOMI satellite observations. *Geoscientific Model Development*, 15, pp.5787–5805. <https://doi.org/10.5194/gmd-15-5787-2022>.

Varon, D.J., Jacob, D.J., Sulprizio, M., Estrada, L.A., Downs, W.B., Shen, L., Hancock, S.E., Nesser, H., Qu, Z., Penn, E., Chen, Z., Lu, X., Lorente, A., Tewari, A. and Randles, C.A., 2022. Integrated Methane Inversion (IMI 1.0): a user-friendly, cloud-based facility for inferring high-resolution methane emissions from TROPOMI satellite observations. *Geoscientific Model Development*, 15, pp.5787–5805. <https://doi.org/10.5194/gmd-15-5787-2022>.

Vigano, I., Van Weelden, H., Holzinger, R., Keppler, F., McLeod, A., & Röckmann, T. (2008). Effect of UV radiation and temperature on the emission of methane from plant biomass and structural components. *Biogeosciences*, 5(3), 937-947.