

Прикладне програмування в науках про Землю

Практична робота №2

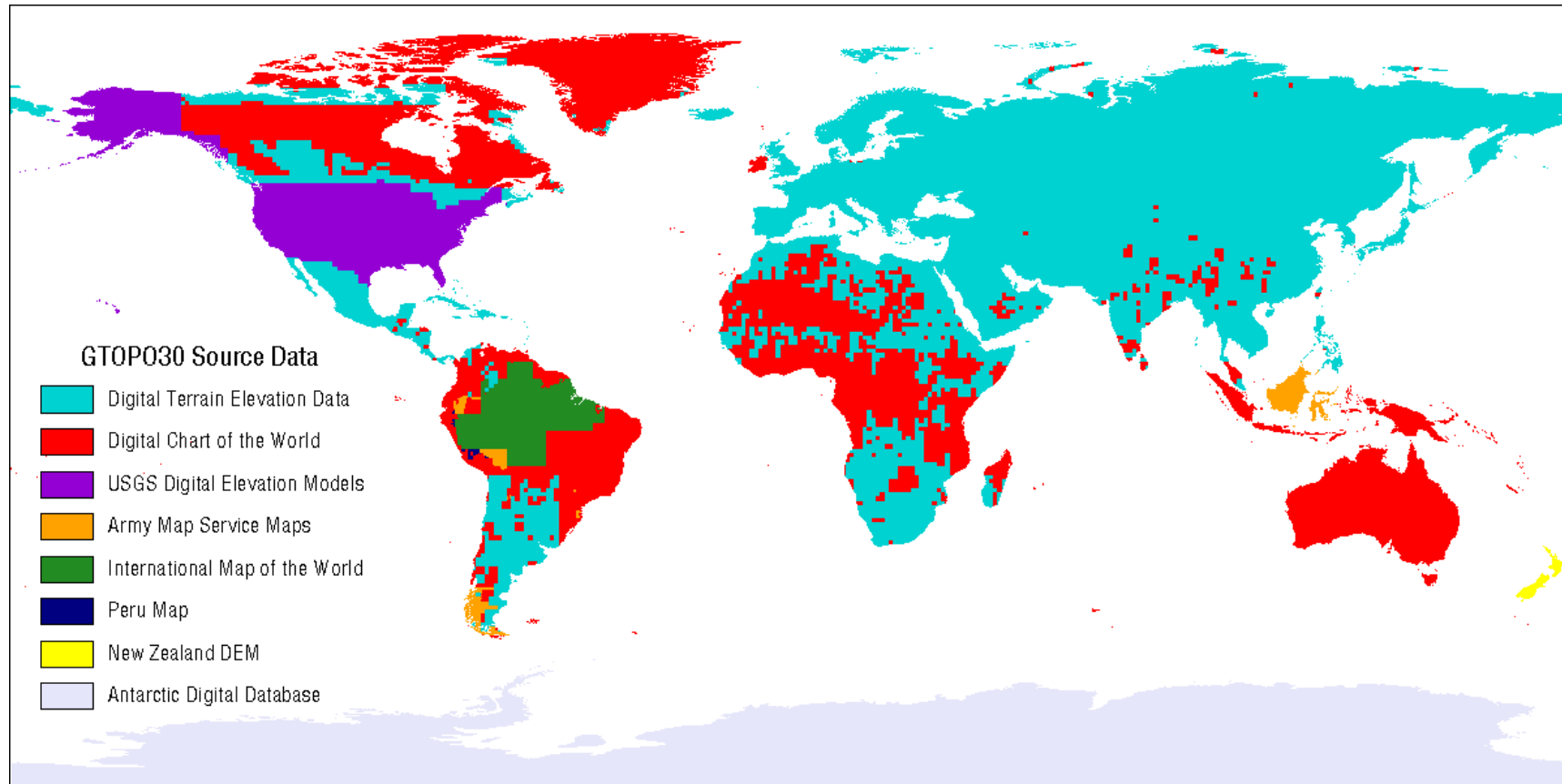
доц. Онищук В.І.
доц. Демидов В.К.
аспір. Охрімчук Р.Ю.

Хід роботи

1. завантаження георастрів із DEM даними;
2. створення метаданих для GTOPO_30AS продукту та його ініціалізація у середовищі ODC;
3. створення датасетів із метаданими та їх ініціалізація у ODC середовищі;
4. знайомство із завантаженням даних у середовищі ODC;
5. зробити звіт про виконану роботу в електронному вигляді.

1. Завантаження георастрів із DEM даними

ГТОР030 (Global 30 Arc-Second Elevation Data Set) - це цифрова модель рельєфу (DEM), яка має глобальне покриття земної поверхні з роздільною здатністю 30 кутових секунд, що приблизно дорівнює 1 кілометру. Цей набір даних був розроблений і випущений Геологічною службою США (USGS) у 1996 році. Даний продукт був створений на основі радарної альтиметрії (SAR).

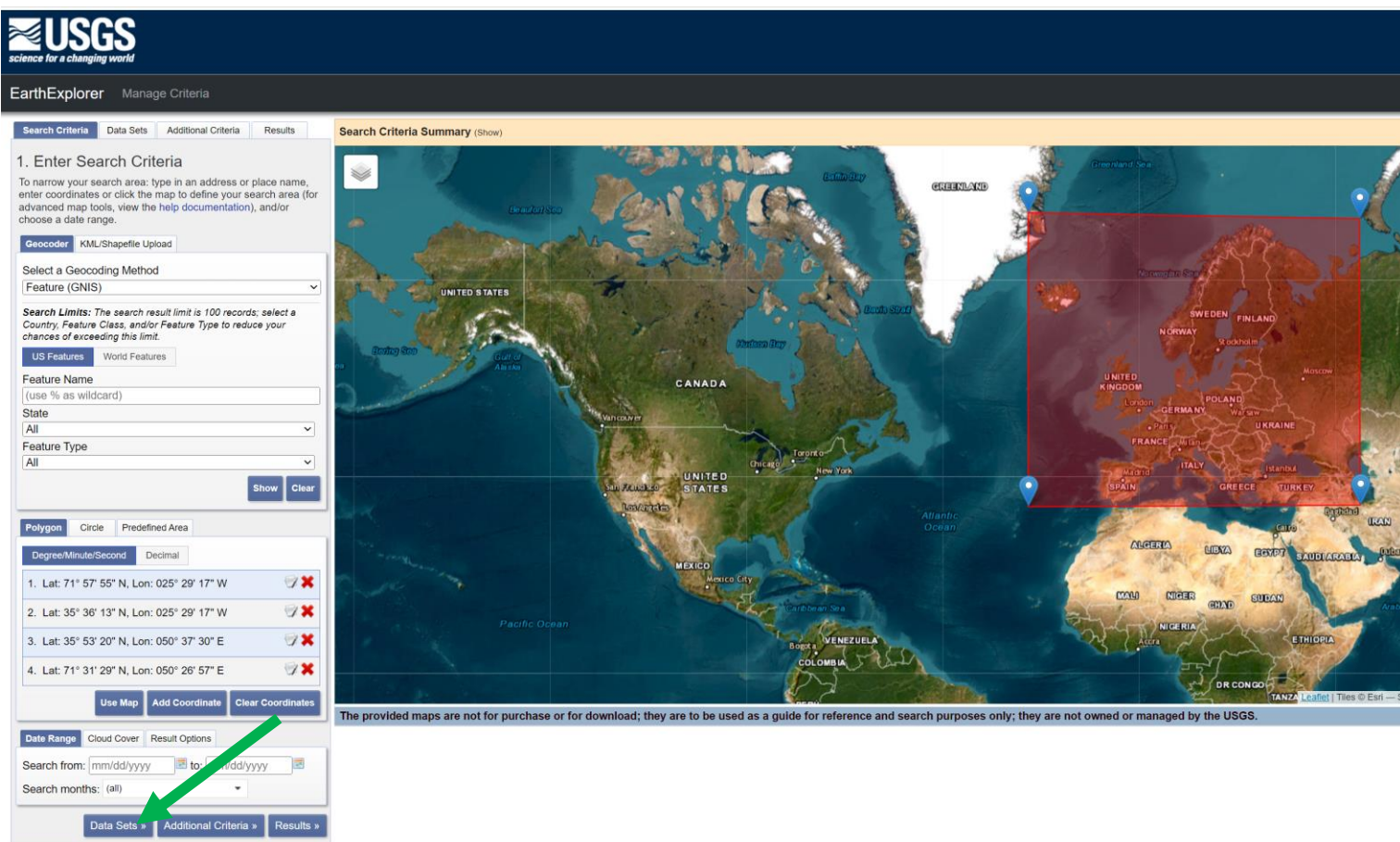


(Джерело: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-global-30-arc-second-elevation-gtopo30?qt-science_center_objects=0#qt-science_center_objects)

1. Завантаження георастрів із DEM даними

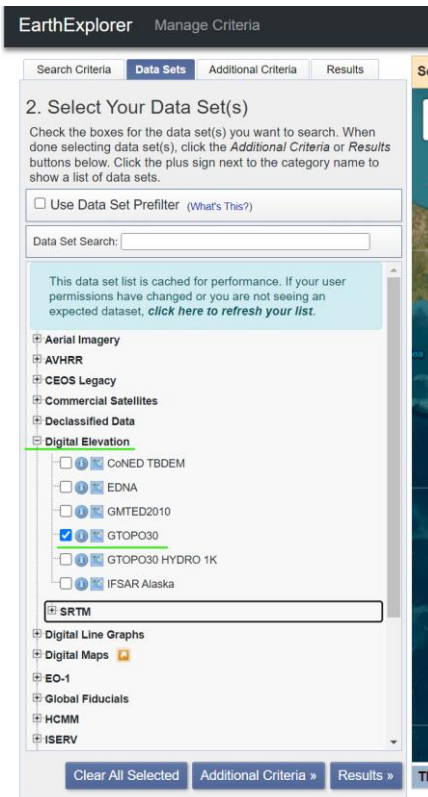
1.1 Завантажити **ГТОРО30** георастри можна за посиланням: <https://earthexplorer.usgs.gov/> - це веб-сайт, розроблений Геологічною службою США (USGS), який дозволяє користувачам шукати, переглядати та завантажувати різноманітні геопросторові дані. Для завантаження потрібно створити обліковий запис;


1.2 Наступний кроком потрібно обрати область, для якої необхідно підгрузити тайли, та натиснути на кнопку – **Data Sets**;

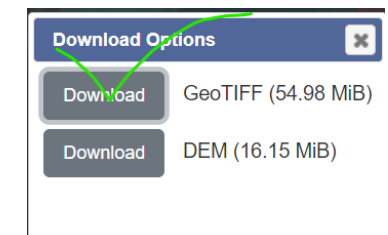
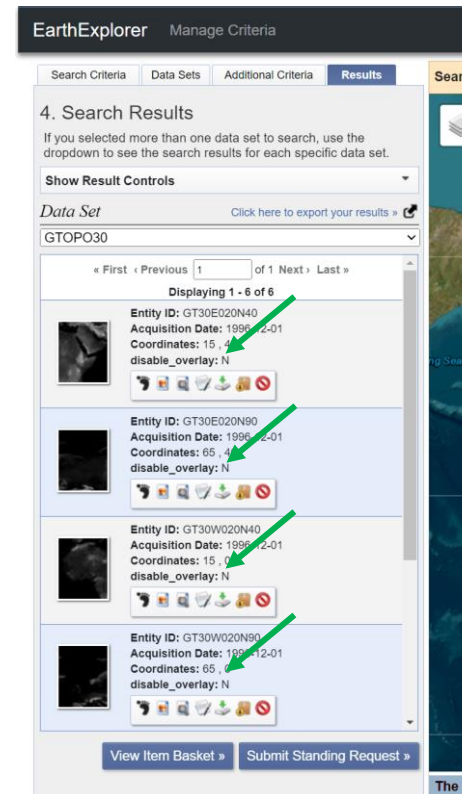


1. Завантаження георастрів із DEM даними

1.3 Оберіть у меню зі списком датасетів - **GTOPO30** продукт та натиснути на кнопку **Results**;



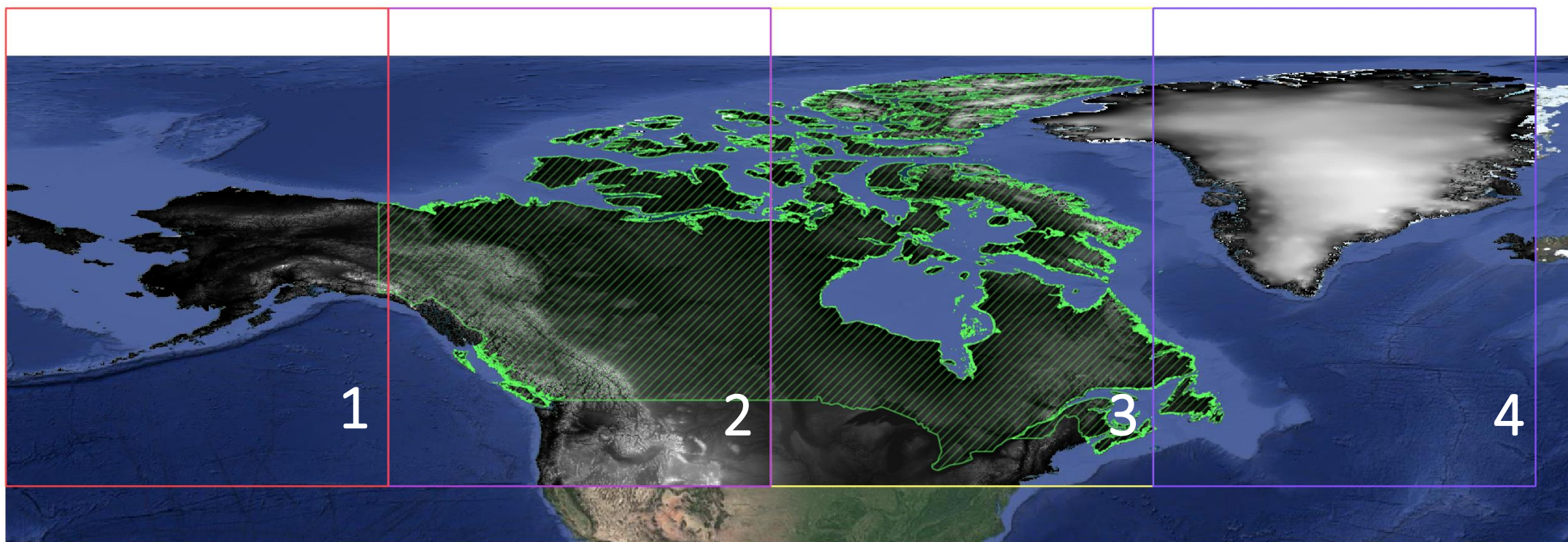
1.4 Завантажте георастри для всіх тайлів, які входять у вибрану вами область. Для завантаження натисніть на  піктограму, а потім у вікні, що з'явиться, оберіть опцію – **GeoTIFF**. Необхідно завантажити всі 33 тайли, що покривають весь світ;



(Даний спосіб завантаження георастрів є найпростішим, проте не найефективнішим. Для виконання цього завдання можна обрати інші методи, які можуть бути більш оптимальними)

1. Завантаження георастрів із DEM даними

Для кращого розуміння матеріалу, додаю візуалізацію тайлів у середовищі QGIS. Наприклад, територія Канади покривається 4 тайлами, які наведено на рисунку. Дані тайли знаходяться у папці **.data/gtopo30_example_tiff.zip**. Для них також, буде створений приклад .yaml файлу із їх метаданими у форматі – eo3 (детальніше про це описано у 3 розділі - Створення датасетів із метаданими)



2. Створення метаданих для GTOPO_30AS продукту та його ініціалізація у середовищі ODC

Створення метаданих буде відбуватися у середовищі conda. Для початку роботи, потрібно активувати раніше створене середовище – **odc_env** і запустити **Jupyter Notebook**;

2.1 Потрібно відкрити **lab_2_1.ipynb**, який знаходиться у папці – **./lab_2** та виконати всі наведенні там завдання;

```
[1]: import yaml
    from pathlib import Path

[2]: product_name = 'GTOPO_30AS'
    description = "GTOPO30 is a global digital elevation model (DEM) with a horizontal grid spacing of 30 arc seconds (approximately 1 kilometer)"
    path_save_product = r'D:\odc\GTOPO_30AS\metadata\product_definition' # Вказати власний шлях

[3]: var_name = 'DEM'
    measurements = [{"name": var_name,
                      "aliases": [var_name.lower()],
                      "units": "masl (meters above mean sea level)",
                      "dtype": 'int16',
                      "nodata": -9999}]

[4]: def create_eo3_product(product_name, description, measurements, path_save_product):

    eo3_prod_dict = {
        "name": product_name,
        "description": description,
        "metadata_type": "eo3",

        "metadata": {
            "product": {
                "name": product_name
            },
            "measurements": measurements
        }

    print(eo3_prod_dict)

    path_product_definition = str(Path(path_save_product)/f'{product_name}_product_definition.yaml')

    with open(path_product_definition, "w") as dest:
        yaml.dump(eo3_prod_dict, dest, explicit_start=True, sort_keys=False) #TODO Потрібно описати, параметри функції explicit_start та sort_keys, яку в

    return path_product_definition

[5]: create_eo3_product(product_name, description, measurements, path_save_product)

{'name': 'GTOPO_30AS', 'description': 'GTOPO30 is a global digital elevation model (DEM) with a horizontal grid spacing of 30 arc seconds (approximately 1 kilometer)', 'metadata_type': 'eo3', 'metadata': {'product': {'name': 'GTOPO_30AS'}}, 'measurements': [{'name': 'DEM', 'aliases': ['dem'], 'units': 'masl (meters above mean sea level)', 'dtype': 'int16', 'nodata': -9999}]}

[5]: 'D:\odc\GTOPO_30AS\metadata\product_definition\GTOPO_30AS_product_definition.yaml'

[6]: #TODO Потрібно написати функцію, яка зможе читати словник із створеного файлу, також у коментарі навести приклад її роботи

def read_yaml(path_yaml):
    ...
    ARGS:
    - path_yaml: (str)
    RETURNS:
    ...
    - content_dict: (dict)
    ...

    #TODO - тут потрібно написати код

    return content_dict
```

2. Створення метаданих для GTOPO_30AS продукту та його ініціалізація у середовищі ODC

2.2 Використовуючи командний рядок **Anaconda Prompt** потрібно перейти в директорію, де був створений .yaml файл із продуктом (за допомогою функції - **create_eo3_product**);

```
(odc_env) D:\>cd D:/odc/GTOPO_30AS/metadata/product_definition
```

2.3 Для перевірки, чи середовище ODC є доступним потрібно виконати команду **datacube**. Якщо ви отримали повідомлення, яке показано на картинці нижче, це означає, що бібліотека буда інстальована вірно;

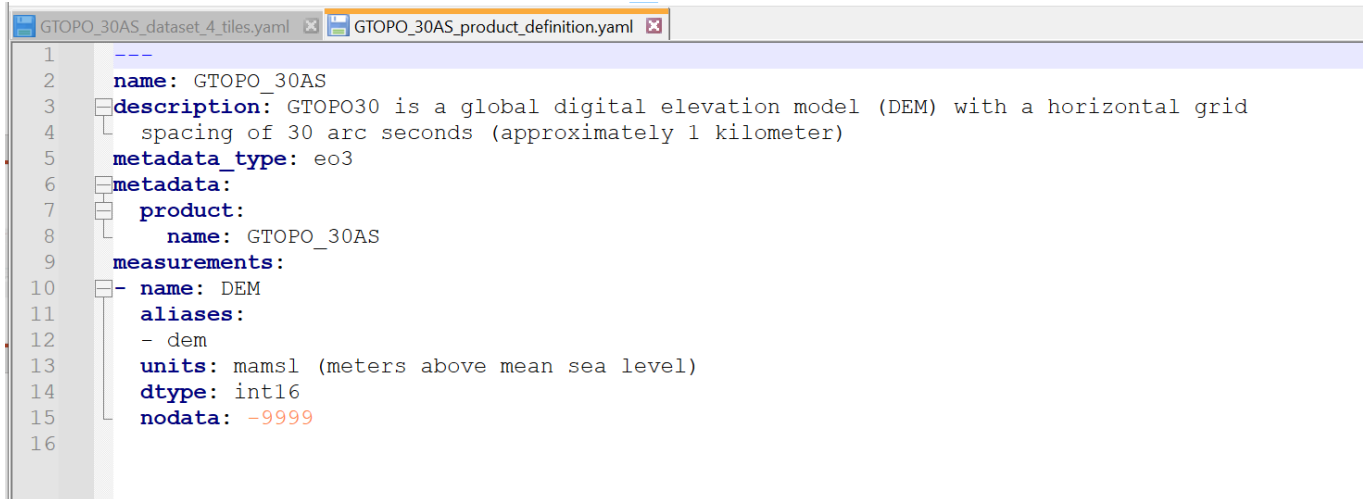
```
(odc_env) D:\odc\GTOPO_30AS\metadata\product_definition>datacube
Usage: datacube [OPTIONS] COMMAND [ARGS]...

Data Cube command-line interface

Options:
  --version
  -v, --verbose           Use multiple times for more verbosity
  --log-file TEXT         Specify log file
  -E, --env TEXT
  -C, --config, --config_file TEXT
  --log-queries           Print database queries.
  -h, --help              Show this message and exit.

Commands:
  dataset  Dataset management commands
  ingest   WARNING: Ingestion has been deprecated in v1.8.14 and will be...
  metadata Metadata type commands
  product  Product commands
  system   System commands
  user     User management commands
```


2. Створення метаданих для GTOPO_30AS продукту та його ініціалізація у середовищі ODC

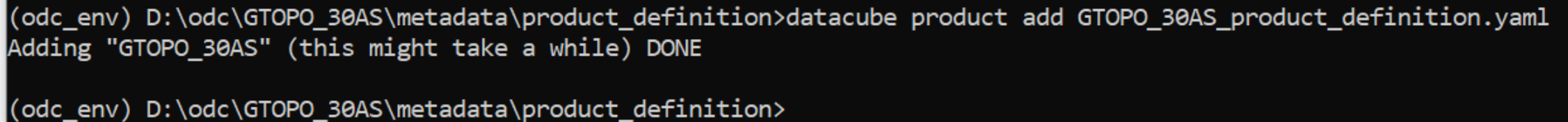


```
1 ---
2 name: GTOPO_30AS
3 description: GTOPO30 is a global digital elevation model (DEM) with a horizontal grid
4   spacing of 30 arc seconds (approximately 1 kilometer)
5 metadata_type: eo3
6 metadata:
7   product:
8     name: GTOPO_30AS
9   measurements:
10    - name: DEM
11      aliases:
12        - dem
13      units: mams1 (meters above mean sea level)
14      dtype: int16
15      nodata: -9999
16
```

(Приклад .yaml файла із метаданими для створення GTOPO_30AS продукту)

2.4 Для ініціалізації продукту у середовищі ODC потрібно виконати команду:

datacube product add GTOPO_30AS_product_definition.yaml (шлях до .yaml файлу із метаданими продукту);



```
(odc_env) D:\odc\GTOPO_30AS\metadata\product_definition>datacube product add GTOPO_30AS_product_definition.yaml
Adding "GTOPO_30AS" (this might take a while) DONE

(odc_env) D:\odc\GTOPO_30AS\metadata\product_definition>
```

(При успішному виконанні команди, буде отримане відповідне сповіщення)

2.5 Для закріплення матеріалу опишіть різницю між бібліотеками **pathlib** та **os** (вони використовуються для взаємодії з операційною системою на рівні файлової системи і є базовими) та додайте це до вихідного звіту;

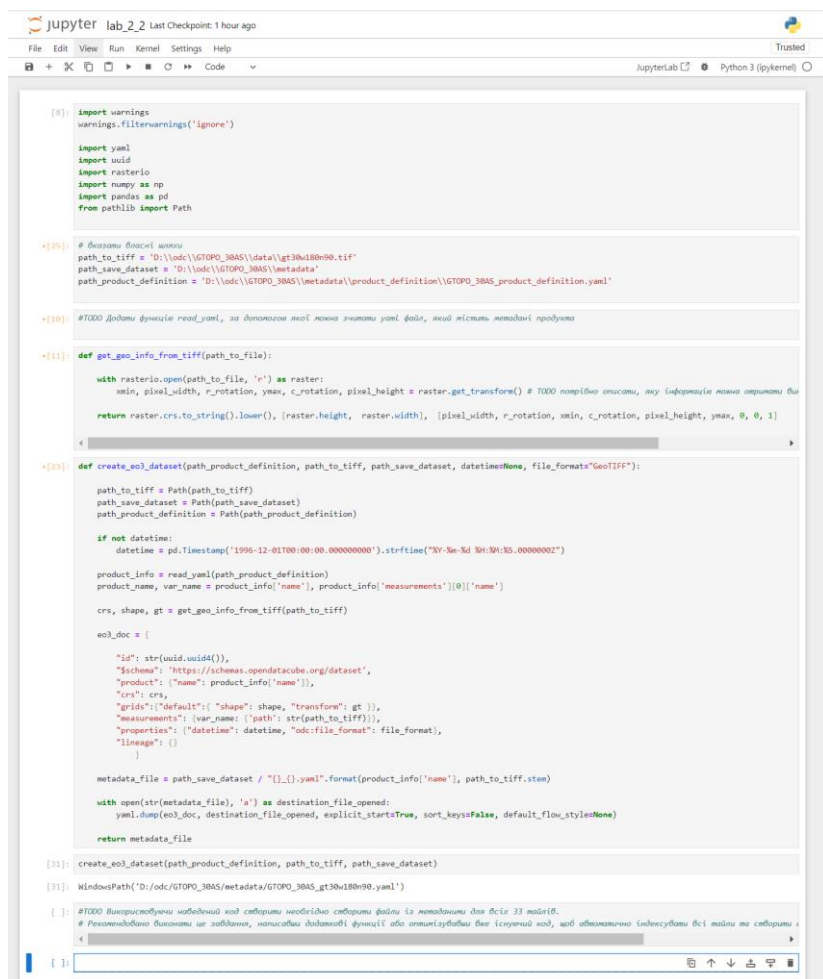
3. Створення датасетів із метаданими та їх ініціалізація у ODC середовищі

Перед початком роботи рекомендується встановити додаткові бібліотеки у вже існуюче середовище **odc_env**, запустивши наступні команди у командному рядку Anaconda Prompt:

```
conda install pandas
```

```
conda install conda-forge::matplotlib
```

3.1 Потрібно відкрити файл **lab_2_2.ipynb**, який знаходиться у папці у **./lab_2** та виконати всі TODO, які там наведені;



```
[X] import warnings
warnings.filterwarnings('ignore')

import yaml
import uuid
import rasterio
import numpy as np
import pandas as pd
from pathlib import Path

# TODO: додати функцію read_yaml, за допомогою якої можна зачитати yaml файл, який містить метадані продукту

# TODO: додати функцію create_eo3_dataset, за допомогою якої можна створити датасет з метаданими

def get_geo_info_from_tiff(path_to_file):
    with rasterio.open(path_to_file, 'r') as raster:
        min_, pixel_width, r_rotation, ymax, c_rotation, pixel_height = raster.get_transform() # TODO: потрібно отримати, яку інформацію можна отримати від
    return raster.crs.to_string().lower(), [raster.height, raster.width], [pixel_width, r_rotation, min_, c_rotation, pixel_height, ymax, 0, 0, 1]

def create_eo3_dataset(path_product_definition, path_to_tiff, path_save_dataset, datetime=None, file_format="GeoTIFF"):
    path_to_tiff = Path(path_to_tiff)
    path_save_dataset = Path(path_save_dataset)
    path_product_definition = Path(path_product_definition)

    if not datetime:
        datetime = pd.Timestamp('1996-12-01T00:00:00.000000000').strftime("%Y-%m-%d %H:%M:%S.000000000")

    product_info = read_yaml(path_product_definition)
    product_name, var_name = product_info['name'], product_info['measurements'][0]['name']

    crs, shape, gt = get_geo_info_from_tiff(path_to_tiff)

    eo3_doc = {
        "id": str(uuid.uuid4()),
        "$schema": "https://schemas.opendatacube.org/dataset",
        "product": {"name": product_info['name']},
        "crs": crs,
        "grid": {"default": {"shape": shape, "transform": gt}},
        "measurements": [{"var_name": var_name, "path": str(path_to_tiff)}],
        "properties": {"datetime": datetime, "odc:file_format": file_format},
        "lineage": {}
    }

    metadata_file = path_save_dataset / "[{}].yaml".format(product_info['name'], path_to_tiff.stem)

    with open(str(metadata_file), 'a') as destination_file_opened:
        yaml.dump(eo3_doc, destination_file_opened, explicit_start=True, sort_keys=False, default_flow_style=None)

    return metadata_file

[31]: create_eo3_dataset(path_product_definition, path_to_tiff, path_save_dataset)

[32]: WindowsPath('D:/odc/GTOPO_384S/metadata/GTOPO_384S_gt384180x90.yaml')
```

[] # TODO: Використовуючи необхідний код створити необхідний набір файлів із метаданими для всіх 33 майбл.
Рекомендується виконати це за допомогою наступної функції або адаптувати вже існуючий код, щоб автоматично ініціалізувати всі файли та створити і

3. Створення датасетів із метаданими та їх ініціалізація у ODC середовищі

3.2 Після створення файлу (або файлів) із метаданими для всіх 33 тайлів, їх необхідно додати до бази даних, що можна зробити, використовуючи команду:

datacube dataset add *GTOPO_30AS_dataset_4_tiles.yaml* (шлях до *.yaml* файлу із метаданими датасетів)

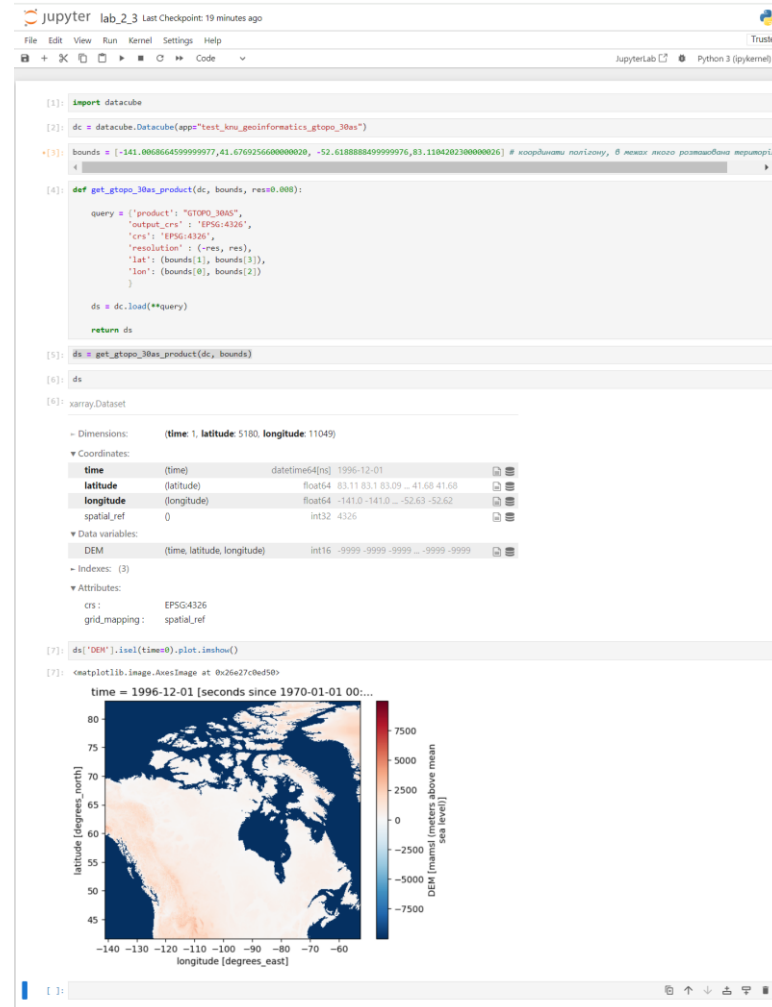
```
(odc_env) D:\odc\GTOPO_30AS\metadata\product_definition>cd ..  
  
(odc_env) D:\odc\GTOPO_30AS\metadata>datacube dataset add GTOPO_30AS_dataset_4_tiles.yaml  
Indexing datasets [#####] 100%  
  
(odc_env) D:\odc\GTOPO_30AS\metadata>_
```

(При успішному виконанні буде отримане дане сповіщення)

*(Для прикладу, у папці **./configs** знаходиться файл - **GTOPO_30AS_dataset_4_tiles.yaml**, який містить метадані для 4 тайлів, що покривають територію Канади. Георастри із відповідними тайлами знаходяться за шляхом - **./data/gtopo30_example_tiff.zip**)*

4. Знайомство із завантаженням даних у середовищі ODC

4.1 Потрібно відкрити файл **lab_2_3.ipynb**, який знаходиться у папці у **./lab_2**, та зробити візуалізацію DEM даних для продукту - GTOPO_30AS, як це показано у прикладі, тільки для іншого полігону. Для цього необхідно вказати інші координати (bounds). Для підтвердження виконання цього завдання, необхідно зробити скріншот із отриманими результатами, який буде демонструвати, що не сталося жодних збоїв у роботі ODC при візуалізації даних для іншої частини світу.



5. Зробити звіт про виконану роботу в електронному вигляді

5.1 Для успішного виконання практичної роботи команда `ds['DEM'].isel(time=0).plot.imshow()` має виконуватися без помилок. Це буде означати, що продукт та датасети були успішно додані до середовища ODC. Як завершальний етап виконання цієї роботи, потрібно підготувати звіт про виконану роботу і надіслати його в Google Classroom. Також потрібно додати .yaml файл із датасетами, який буде містити метадані для всіх 33 тайлів, проіндексованих локально на вашому ПК.