Copernicus and openEO

OER4SDI

Open Educational Resources for Spatial Data Infrastructures

In this technical tutorial on the Copernicus Data Infrastructure, you will learn about the Copernicus Data Space Ecosystem and how to process EO data using Python and OpenEO.

Tobias Krumrein, Albert Remke
Institute for Geoinformatics, University of Münster

September 2024

ein Kooperationsprojekt, empfohlen durch:

DIGITALE HOCHSCHULE NRW

gefördert durch:

Ministry of Culture and Science of the State of North Rhine-Westphalia

1. Overview

In this technical tutorial on the Copernicus data infrastructure, you will learn about the **Copernicus Data Space Ecosystem** and **OpenEO**. After working through the tutorial, you will be able to

- understand the purpose and scope of the Copernicus EO infrastructure
- understand the role of the Copernicus Data Space Ecosystem and OpenEO for accessing and processing Copernicus data
- use Python to create information products from Earth observation data using the Copernicus Data Space Ecosystem and OpenEO.

This tutorial is structured as follows:

This Tutorial takes about 60 minutes for reading and viewing the provided materials, downloading the software and for conducting the hands-on exercises and tasks.

This tutorial is designed for students and professionals who want to improve their understanding of Spatial Information Infrastructures. We assume that you have some basic knowledge about remote sensing, Python and web technologies. However, even without this prior knowledge you will be able to follow and to achieve the main learning objectives. Your computer should have 8 GB of usable RAM and 2 GB of usable hard disk space to download and use the software for this tutorial.

This Tutorial has been developed at the Institute for Geoinformatics, University of Münster. Authors are Tobias Krumrein and Albert Remke. The latest version of this tutorial is always available on GitHub. You are invited to use GitHub issues to provide feedback and suggest improvements.

generally excluded. Any code provided with the tutorial can be used under the terms of the MIT license. Please see the full license terms at:
https://github.com/oer4sdi/OER-CopernicusAndOpenEO/blob/main/LICENSE.md

The tutorial can be referenced as follows: “OER-DataAccessVia-OGC-API-Features”, OER4SDI project / University Münster, CC BY-SA 4.0.

The OER4SDI project has been recommended by the Digital University NRW and is funded by the Ministry of Culture and Science NRW.

Let's Get Started!

Begin with familiarizing yourself with the background on Copernicus, the Copernicus Data Space Ecosystem and openEO as presented in the next section. The technical exercises then provide you with the opportunity to gain hands-on experience with these concepts and technologies.

2. Background

This section will provide you with a brief introduction to Copernicus, the Copernicus Data Space Ecosystem and openEO.

2.1 The Copernicus Spatial Information Infrastructure

The European Copernicus spatial information infrastructure is one of the largest Earth observation programs in the world. It was launched in 1998 by the European Commission and the European Space Agency (ESA). It provides us with Earth observation data, data services and specific information products at global scale. Since 2013, most Copernicus services are openly accessible and free of charge for everyone.

Please watch this short introductory video, which presents an overview on the Copernicus program's objectives, components and actors. After viewing the video, we'll discuss some aspects in more detail.

As you have seen, the purpose of the Copernicus program is:

a) **..to provide high-quality earth observation data obtained from satellite and in-situ observations and measurements**

   Copernicus builds and operates its own fleet of Sentinel satellites with a broad range of specialized sensors. This requires many components such as launching facilities, control centers, ground stations, data centers and cloud based XaaS service offerings. Furthermore, Copernicus has contracted data from other satellite programs such as EnviSat, terraSAR or TandDEM-X.

Copernicus uses the term “in situ data” for all the data that is needed to create Copernicus information products and does not come from space. Examples are ground-based and airborne observation and measurement data on weather phenomena, air quality, biomass, solar radiation, etc.. This data does not come from Copernicus but from other institutions. Copernicus mainly coordinates and supports the availability of in situ data for those who need them for building Copernicus information products.

Please check the following website for an overview on the satellite missions and in situ data sources that are in the scope of Copernicus:
https://www.copernicus.eu/en/about-copernicus/infrastructure-overview

b) … to provide a set of high quality information products in six thematic areas: land monitoring, marine environment, atmosphere, climate change, emergency management, and security

The information products, such as land use maps, data on ground movements, sea ice cover, forest fires, etc., are produced from Copernicus satellite and in-situ data and made available via topic-specific portals. Examples are the Copernicus Land Monitoring Service (CLMS) and its products at https://land.copernicus.eu/, or the Copernicus Marine Environment Monitoring Service (CMEMS) and its portal https://marine.copernicus.eu.

Most of the information products are publicly accessible and free of charge. For sensitive data, access is restricted to authorized users.

Please check the following WebSite for an overview on the existing Copernicus information services: https://www.copernicus.eu/en/copernicus-services

c) … to provide tools and services that enable users to work with Copernicus data

The Sentinel satellite missions alone produce more than 20 terabytes of data every day. The data archives already comprise many petabytes of data. Deriving high-quality information products from this data requires specialized software, significant storage volume and computing capacity, which can only be provided by high-performance data centers.

This is why the satellite data itself is provided via scalable cloud platforms that allow users to access, analyze and visualize the raw data according to their individual needs. These users include, for example, researchers or public sector institutions, but also private companies that want to develop and market their own products and services.

These cloud platforms offer XaaS, i.e. cloud-based services that provide various IT resources and functions as a service that are required to use Earth observation data. This includes:

- **IaaS - Infrastructure as a Service**
  Virtualized computing resources over the internet, such as servers, storage, and networks
- **DaaS - Data as a Service**
  Access to large volumes of Copernicus satellite and selected in-situ data.
- **PaaS - Platform as a Service**
  Services for developers to analyze data and to build, run, and manage applications without needing to care for managing the underlying infrastructure
- **SaaS - Software as a Service**
  These are end-user applications, supporting the needs of their target groups.

Further service types may support marketplace functionality and support for business transactions (contracting, billing, payment). Essentially, XaaS enables on-demand access to IT resources and usage-based billing models, which significantly improves scalability and cost efficiency for resource-intensive applications.

Copernicus has used public-private partnerships to support the development of these cloud platforms that provide special services for working with Copernicus data. Examples are WEkEO[1], CREODIAS[2], CODE-DE[3], and also the Copernicus Data Space Ecosystem (CDSE)[4], which we will discuss in more detail later.

It is important to note that in addition to the platform offerings that are co-financed by EU funds and/or national budgets, there are also a number of commercial cloud platforms that offer XaaS services for processing and using Earth observation data, such as Google Earth Engine[5], the Earth in AWS Program[6], Microsoft's Planetary Computer[7].

And who is responsible for what in the context of Copernicus?

The development and operation of the Copernicus information infrastructure is driven by the European Commission. However, Copernicus is designed as a joint effort of all public and private sectors in order to achieve maximum efficiency and maximum benefits from this infrastructure. The following list gives an idea on the governance structure of the Copernicus infrastructure.

a) **Copernicus Policies and funding**
   - The Copernicus program is managed by the **European Commission**, specifically by its Directorate General Defense, Industry and Space (DG DEFIS).

[1] Landing page of the WEkEO platform: https://www.wekeo.eu/
[2] Landing page of the CREODIAS platform: https://creodias.eu/
[3] Landing page of the CODE-DE platform: https://code-de.org/
[4] Landing page of the CDSE platform: https://dataspace.copernicus.eu/
[5] Landing page of the Google Earth Engine platform: https://earthengine.google.com/
[6] Landing page of the Earth on AWS program:https://aws.amazon.com/de/earth/
[7] Landing page of Microsoft's PLanetary Computer: https://planetarycomputer.microsoft.com/

- It is supported by the **Copernicus Committee** where all member states and other relevant entities are represented and which has an advisory role regarding strategic decisions and funding allocation.
- The Committee is complemented by the **Copernicus User Forum** with experts from the member states and from European institutions. Its role is to gather and represent the needs of Copernicus end-users.
- The **European Member States** participate in policy development, funding discussions, and strategic decision-making, ensuring national alignment with Copernicus policies
- The **European Parliament** approves the Copernicus legal framework and funding as well as its alignment with EU policies and priorities.

b) Development and operation of Copernicus

- The **European Commission** (specifically DG DEFIS) coordinates all Copernicus program activities, allocates budgets, and manages contracts with partners and service providers.
- The new **European Union Agency for the Space Programme (EUSPA)** coordinates the security accreditation of the Copernicus infrastructure and supports the integration and market uptake of the various components of the EU Space program.
- The **European Space Agency (ESA)** coordinates the space segment, i.e. manages satellite design, launches and operates satellite missions. It also coordinates the European Data Space Ecosystem which is a platform for accessing and processing Copernicus EO data.
- **European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT)** coordinates the operation of Sentinel-3, -4, -5, and -6, primarily supporting marine and atmospheric monitoring.
- The **European Environment Agency (EEA)** coordinates the in situ data component and the provision of local and pan European land monitoring services.
- The **Joint Research Centre (JRC)** coordinates the provision of global land monitoring services and emergency management services
- The **European Organisation for the Exploitation of Meteorological Satellites (EUMETSAT)** coordinates the provision of atmosphere monitoring and climate change information services.
- **Mercator Ocean** coordinates monitoring services for the marine environment
- **FRONTEX** and the **European Maritime Safety Agency (EMSA)** are coordinating information services supporting border and maritime surveillance

Many more institutions are involved in the development and operation of all infrastructure components, including national agencies such as the German Aerospace Center (DLR), as well as many research institutions and private companies.

2.2 The Sentinel-2 Satellite Mission

As mentioned before, Copernicus operates its own series of satellite missions. In our exercise we will work with data from the Sentinel-2 mission.

The Sentinel-2 mission consists of two identical satellites, Sentinel-2A and Sentinel-2B, launched in 2015 and 2017 respectively. These satellites scan every place on Earth every 5 days, providing high-resolution optical imagery.

Sentinel-2 uses a Multispectral Instrument (MSI), capturing high-resolution images with spatial resolutions of 10 meters for visible and near-infrared bands, 20 meters for red-edge and short-wave infrared bands, and 60 meters for atmospheric correction bands. This allows for detailed observations of the Earth's surface. Unlike radar systems, Sentinel-2 relies on optical and infrared sensors, which means it depends on daylight and good weather conditions.

Sentinel-2's capabilities in capturing detailed optical and infrared images make it an extremely valuable tool for continuous observation and analysis of the Earth's surface, supporting a wide range of environmental, agricultural, and scientific applications, for example:

- **Land Use and Land Cover Mapping**
  High-resolution images help to detect and map changes in land use and land cover, which is crucial for many planning and monitoring tasks (e.g. nature and landscape conservation, regional planning, subsidy control in agriculture).

- **Monitoring Vegetation**
  The MSI's spectral bands are optimized for monitoring plant health, chlorophyll content, and vegetation growth, making it essential for agricultural management.

- **Disaster Management**:
  Frequent revisit times enable timely monitoring of areas affected by natural disasters such as floods, wildfires, and landslides, aiding in emergency response and recovery efforts.

- **Water Quality Monitoring**: Sentinel-2 provides data on turbidity, chlorophyll concentration, and the extent of water bodies, supporting the monitoring of inland and coastal waters.

Please visit https://dataspace.copernicus.eu/explore-data/data-collections/sentinel-data/sentinel-2 for more information on the Sentinel 2 satellite mission.

2.3 The Copernicus Data Space Ecosystem (CDSE)

Let's take a closer look at the Copernicus Data Space Ecosystem (CDSE), a fairly new development that gives us an idea on how the European Commission intends to further develop Copernicus.

For a first impression you can look-up this video,which provides you with a high-level introduction to the CDSE platform: https://youtu.be/LImwmjTidFw.

Here are some key features of the CDSE platform:

- **Earth Observation Data**

  The main purpose of CDSE is to facilitate the accessibility and usability of **Copernicus Earth observation data** from the Copernicus Program. It provides access to more than 80 petabytes of data from the Copernicus Sentinels and contributing missions to selected sources of in-situ data. For more details on the data offerings please visit: https://dataspace.copernicus.eu/explore-data/data-collections.

- **Copernicus Browser**

  Within this platform the Copernicus Browser is a rich web application that allows you to
  - search for satellite data by type, time, location, and quality
  - to visualize satellite data in an interactive map using customizable scripts for pixel based map calculations and to create animated time series visualizations
  - to download links for raw and processed data, enabling integration with GIS tools and further analysis

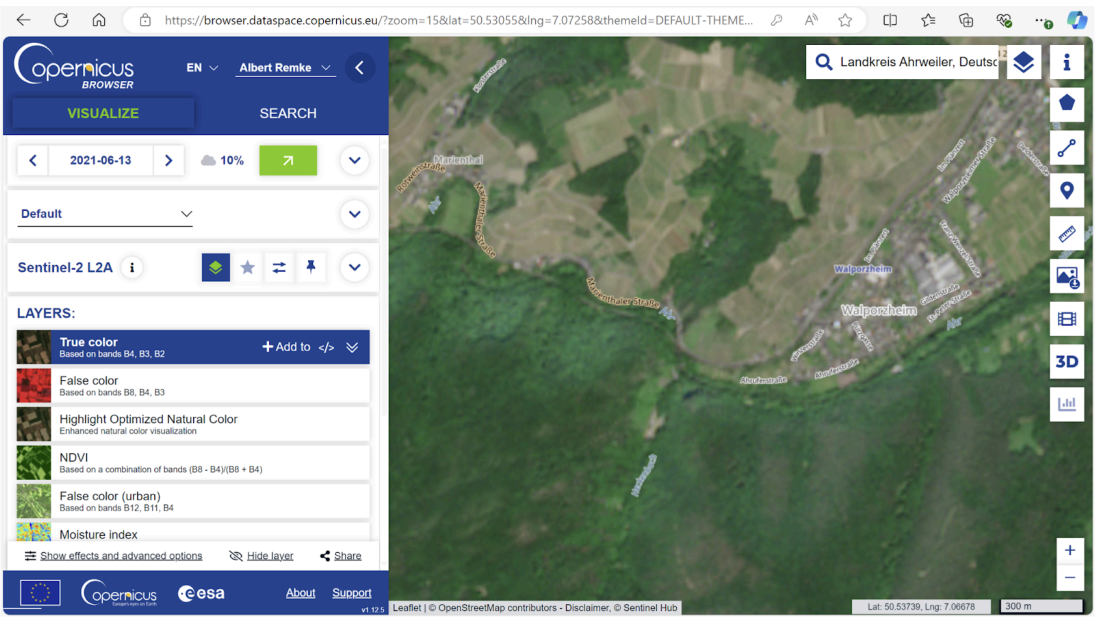

Fig. 1: The CDSE's Copernicus Browser

- **Application programming interfaces (APIs)**

  The CDSE provides a set of APIs that can be used to search for, access, process, analyze, visualize and download Copernicus data. This supports the automation of data processing and the integration of the Copernicus infrastructure with your IT environment. And, most importantly, the processing is done using the capacity of cloud services! This means that you are not limited by the capacity of your local IT environment.

- **JupyterLab**

  The **JupyterLab** of the CDSE supports working with Jupyter notebooks and programming languages such as Python or R. You can customize the notebook's environment to use the software tools that serve your needs and that you are familiar with. You can use CDSE's APIs to integrate the data and processing capabilities of the platform with your individual code. All this is running in the cloud. You are supported to share your code with others, making your work reproducible and useful for others.

- **QGIS PlugIn**

  The APIs are also used by standard software such as QGIS to connect to the data space ecosystem. The **QGIS SentinelHub plug-in** allows you to connect to CDSE's cloud services, search for and access Copernicus data and map services, and combine them with your local data.

For more information on the CDSE platform, visit the website https://dataspace.copernicus.eu. There you will find many introductory video tutorials on the platform's features, as well as comprehensive documentation.

To use the CDSE platform, you need to create an account that comes with free monthly quotas for using the platform services. If you want more, you may need to use commercial "ecosystem services" provided by service providers such as CREODIAS or Mundi Webservices that have integrated their services into the Copernicus Data Space ecosystem.

The Copernicus Data Space ecosystem is coordinated by ESA and implemented and operated by a consortium of European cloud and Earth observation service providers.

2.4 Processing EO-Data with Python and OpenEO

As mentioned earlier, the CDSE provides a set of APIs that can be used to retrieve, process, analyze, visualize, and download Earth Observation data. The advantage of these APIs is that client-side applications do not have to deal with the complexity of the server-side implementations. They can simply focus on using the exposed functionality.

openEO is a web-based API for EO cloud infrastructures that aims to facilitate the discovery, processing, analysis, visualization, and download of Earth Observation data across different cloud platforms. Each backend that supports openEO offers a server-side implementation of the openEO interface. The goal is that the same code can be executed on Google Earth Engine, Sentinel Hub, or CDSE.

openEO is tailored to the needs of data scientists and their typical workflows and toolsets when working with EO data. openEO provides client libraries for Python, R and Javascript that allow to integrate the functionality of EO cloud services into their programs.

openEO is a high-level abstraction layer that hides all the technical details of resource management in the backend. openEO enables users to perform complex geospatial analysis of satellite data without having to manage data storage or the computing infrastructure themselves.

CDSE provides you with many options to use openEO with Python, for example:

- **openEO Editor**

  You can use the **openEO Editor,** a web-application that enables users to create, edit, and run Earth Observation data processing workflows visually, using drag-and-drop operations without needing to write code. Under the hood the editor creates scripts that use the openEO API to connect to CDSE backends. These scripts can be exported, allowing further customization and integration into other platforms or applications.

- **CDSE JupyterLab**

  You can use the CDSE's JupyterLab, which is a development environment that allows users to interact with and analyze Copernicus Earth Observation data using Python or R in Jupyter Notebooks. You can use the openEO Python client library to directly interact with the openEO-supported backends. The processing can be done in the cloud, which is ideal for working with large datasets.

- **Python Scripts in your local IT-environment**

  Once you have installed the openEO client library in your Python environment you can connect to the CDSE using openID Connect to get full access to all the data and computing resources that are provided by the platform.

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

connection.authenticate_oidc() # Triggers a browser login flow
```

To summarize, these are the advantages of working with openEO:

- **Simplified access to EO data**

  Instead of manually downloading satellite data from multiple platforms, OpenEO allows you to access data directly through its API. Whether you're looking for data from Sentinel-2 or other satellites, you can easily retrieve what you need.

- **Processing EO data in the Cloud**

  OpenEO allows you to process large amounts of data in the cloud, which means you don't need a powerful computer to handle complex analyses. You can send requests to OpenEO, and the actual data processing happens on remote servers.

- **Standardized API**

  OpenEO provides a standardized way to work with resources from different EO cloud platforms. You can use the same commands and methods across all platforms. This simplifies your workflow, as you don't need to learn and adapt to the individual APIS of all these platforms.

- **Python Client Library**

  In this tutorial, we will use the openEO Python client library. This client allows us to connect to the OpenEO API, search for data, and perform analyses—all using Python code. This is especially convenient because Python is a widely used programming language, and many tools for data analysis are built around it.

Now that you have an idea of the **Copernicus program**, the **Data Space ecosystem** and how **OpenEO** can simplify satellite data processing, it is time to apply these concepts.

Let's move on to the practical part, where you will gain hands-on experience in using Python, openEO and the CDSE platform for utilizing Earth observation data.

3. Hands-On Exercise

In this part of the tutorial, you will apply the previously presented concepts and technologies in a practical exercise. Referring to a simple use case, you will use Python and openEO to leverage the capabilities of the Copernicus Data Space Ecosystem Platform to access and visualize Sentinel 2 data.

3.1 Exercise Overview

Scenario:

Imagine you work for a governmental disaster management agency. After an extreme heavy rainfall event in 2021 in Western Germany, you want to get a quick first impression of the effects of the flooding caused by the heavy rainfall on agricultural land and urban infrastructure. You want to use Sentinel-2 satellite data to look at the affected areas and detect changes in land cover. Visualizing the Normalized Difference Vegetation Index (NDVI) should help you to identify damage to vegetation.

For this task, you will use the Python notebook that we provided with the tutorial.

Workflow:

For the purpose described above, you want to access Sentinel-2 data captured shortly before and shortly after the rain event. Since Sentinel-2 is an optical sensor, you have to find data with little cloud cover.

Then you want to calculate the NDVI from the Sentinel-2 data and compare it for situations.

The NDVI is a simple and widely used vegetation index that relates to the density and health of plant life in a given area. Vegetation absorbs visible light (especially red) for photosynthesis and strongly reflects near-infrared light. NDVI uses these characteristics to estimate vegetation health by calculating the ratio of the difference between NIR and red reflectance to the sum of NIR and red reflectance:

$$NDVI = (NIR - Red) / (NIR + Red)$$

The NDVI's values range from -1 to +1.

- Values close to +1 indicate healthy, dense vegetation
- Values close to 0 indicate bare soil or sparse vegetation
- Values near -1 often represent water, clouds, or snow

For Sentinel-2, the relevant spectral bands are:

- B08 (NIR): Near-Infrared band (842 nm)
- B04 (Red): Red band (665 nm)

Which means, for computing the NDVI from Sentinel-2 data you use the bands B08 (NIR) and B08 (Red) and apply the NDVI formula for a pixel based calculation:

NDVI = (B08−B04) / (B08+B04)

3.2 Preparation

For the exercise you need to register for Copernicus Data Space Ecosystem and also need to install Docker for using a Python notebook.

3.2.1 Copernicus Data Space Ecosystem Account

First you need to register and create an account for Copernicus Data Space Ecosystem. You can register by following this link: https://dataspace.copernicus.eu/ and clicking on Login in the top right corner.
After you register you need to verify your email address before continuing.

3.2.2 Installing Docker, Jupyter and a Python Notebook

The software environment that we need for this exercise is Docker:

1. **Install Docker Desktop**
   Go to the official web site https://docs.docker.com/get-docker/ and follow the guidance which is provided there to install docker on your computer (8GB usable RAM recommended).

2. **Download the GitHub Repository of this Tutorial**
   - Create a working directory on your local computer.
   - Use either git clone to download the GitHub repository https://github.com/oer4sdi/OER-CopernicusAndOpenEO to your working directory or use your browser, go to https://github.com/oer4sdi/OER-CopernicusAndOpenEO, download the repository as a zip file and unzip it in your working directory

3. **Make sure Docker Desktop is up and running**

4. **Open a terminal and change to your working directory**

5. **Use "docker-compose up --build" to build and run the docker container**

Once this is complete, your terminal output should look similar to this:

```
    To access the notebook, open this file in a browser:
        file:///home/jovyan/.local/share/jupyter/runtime/nbserver-7-open.html
    Or copy and paste one of these URLs:
        http://177a0a74ba1b:8888/?token=3a2e5088e85cde486ab00cfbf5a5ef7346e51092980f43ce
     or http://127.0.0.1:8888/?token=3a2e5088e85cde486ab00cfbf5a5ef7346e51092980f43ce
```

Fig. 2: Example of a correct response in the terminal after the docker container is running

To access the python notebook look out for the link http://127.0.0.1:8888/?token=... in your output (it should look like the one on the bottom in the picture above). Copy the link and the token (the latter will be created new every time you start the container). Paste the link to your preferred browser to access the Jupyter server which displays the notebooks that can be used.

3.3 Walk through the Python Notebook

Now that you have set up your environment, you can use the Python Notebook to implement the workflow as outlined above. .

Open the Python notebook, read and activate one cell after another. This will lead you through the following steps:

1. Configuring the environment: integrating all the libraries that you want to use to access and work with the EO data
2. Connecting to the CDSE platform and authenticate yourself
3. Retrieving Sentinel-2 data for the flood-affected region from two time periods: one before the flood and one after the heavy rainfall event
4. Calculate the NDVI for both time periods and visually compare the results to identify the most affected agricultural and urban areas.

Please be aware that some steps may take some time before you can see the result. So, be patient ;)

4. Summary

In this tutorial, we started with a brief introduction to the **Copernicus Earth Observation Infrastructure**, which provides us with a wealth of space born EO data, derived information products and facilities which support working with these EO data. We saw that Copernicus is a full fledged Spatial Information Infrastructure which has been set up over the last two and a half decades with tremendous investments from the European Commission and the member states.

You have learned that the **Copernicus Data Space Ecosystem** is a cloud-based platform that provides a rich set of XaaS offerings for accessing, processing, analyzing, visualizing and downloading EO data.

You learned to know the **openEO API** that provides you with the possibility to use the capabilities of the CDSE with your local Python code and you gained practical experience in working with these concepts and technologies.

We have focused on some technical aspects of the Copernicus infrastructure, assuming that you either already have previous knowledge of remote sensing or will further develop your knowledge in this area in the future.

If you want to dive deeper into the topics of this tutorial we recommend to use the following resources:

- **General information in the Copernicus EO Infrastructure**

  The WebSite https://www.copernicus.eu/ is an excellent starting point for deepening your knowledge on the Copernicus EO Infrastructure. It provides you with links and further materials on many aspects of Copernicus.

- **Copernicus Data Space Ecosystem**

  We already referred to the DCSE website https://dataspace.copernicus.eu, which provides you with many useful video tutorials and comprehensive documentation.

- **openEO**

  You'll find some good introductory materials at the DCSE website (videos and documentation) as mentioned above.

  For more information on openEO and for contacting the developer community around this openSource project visit https://openeo.org/. Here you find the full documentation, links to the open source software repositories at GitHub and a list of all known openEO providers.