OpenMC installation guide

Fission Reactor Physics I

Prof. Enrico Padovani
Politecnico di Milano

November 17, 2025

Where is it possible to install?

- **On Windows**: with Docker installation ✓
  - $\sim$ 20 minutes
  - $\sim$ 6 GB
- **On Ubuntu/Linux**: with Conda installation ✓
  - $\sim$ 10 minutes
  - $\sim$ 4 GB
- **On MacOS**: not a unique solution ✗ depending on the Mac architecture, the proposed installation procedures on the OpenMC website may fail or not. Generally, Apple Silicon laptop cannot install OpenMC.
- If none of the reported solutions work, we will share a **Google Colab** ✓ notebook with OpenMC installed $\rightarrow$ useful for some simple examples

Windows

Windows installation: pre-requisites

Enable the Windows Subsystem Linux (WSL).

- Open **Windows Power Shell** in administrator mode and insert

```
wsl --install
```

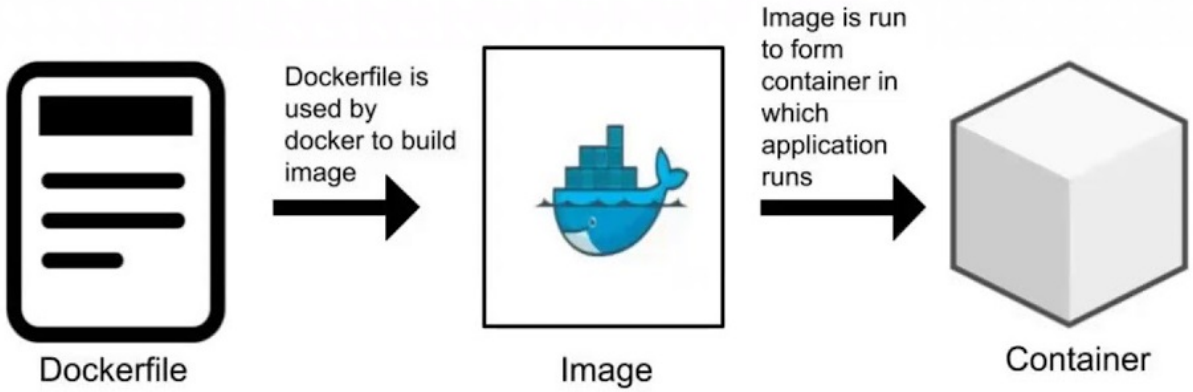
- Restart the computer.

By so doing, most of the common Unix[1] in the input shell (i.e., `ls`, `cd`, `ssh` etc.) are interpreted in the Windows environment.

[1] Unix = Linux or Mac

Docker

Install the ▸ Docker Desktop platform.
The Docker structure is divided in **Images** and **Containers**: Images represents a read-only environment which has all the basic features of the chosen platform; users can clone an Image structure into a Container, allowing them to work in the environment.



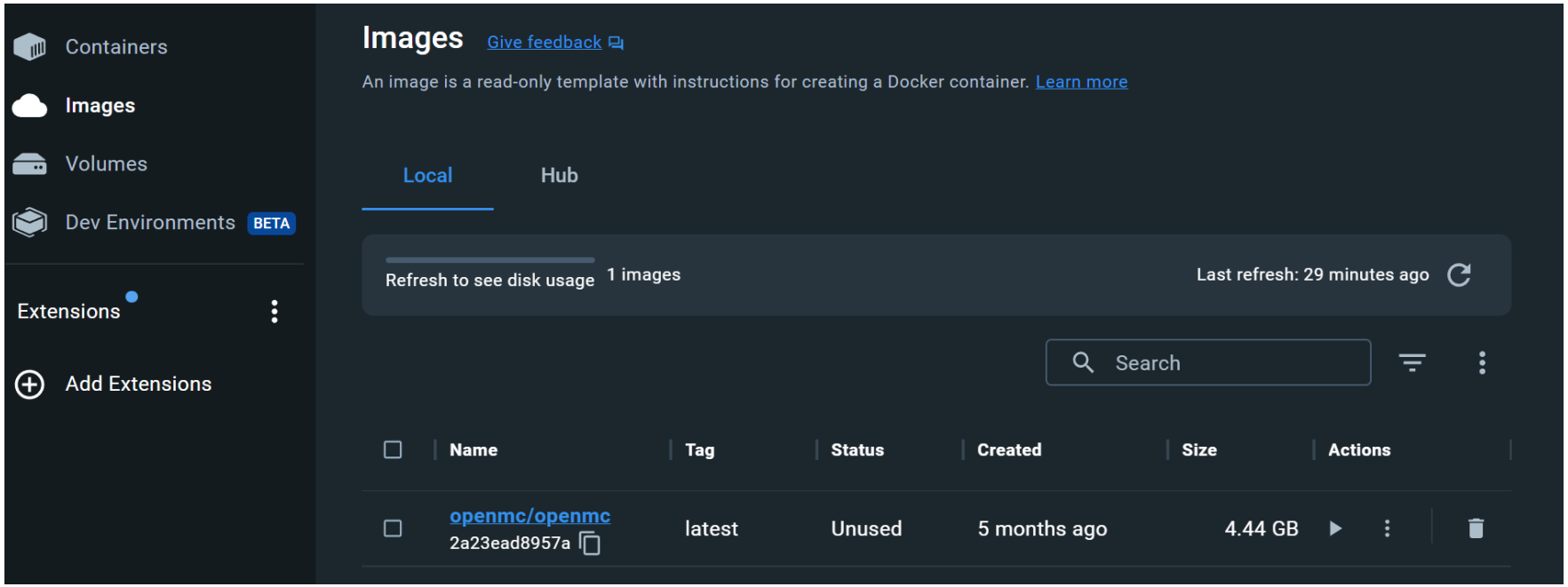
Our aim is to first build the Ubuntu Image with already installed the OpenMC packages. Then, create a custom Container.

Docker Image

Paste in the Power Shell

```
docker pull openmc/openmc:latest
```

to start the Image download.



Once completed, it may require to re-start your laptop.

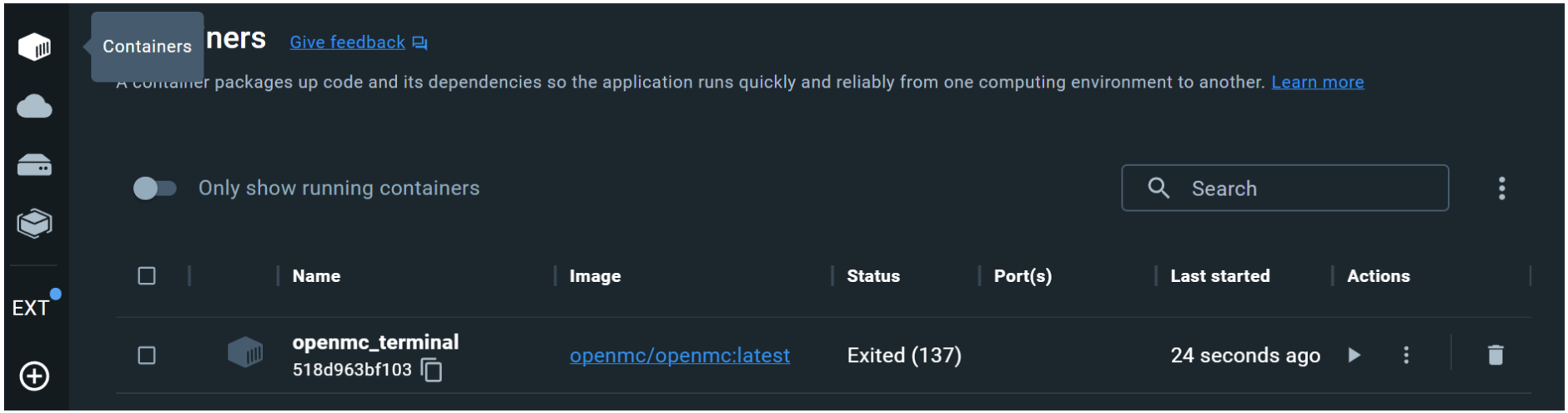
Docker Container (I)

The Image is a *clean copy* of the virtual machine with OpenMC installed. The environment where you can work is called **Container**. To create a Container, go to the Power Shell and paste:

```
docker run -it -p 8888:8888 --name openmc_terminal openmc/openmc:latest
```

A Container called **openmc_terminal** is now created, and you can access it from the Docker Desktop application.

Docker Container (II)

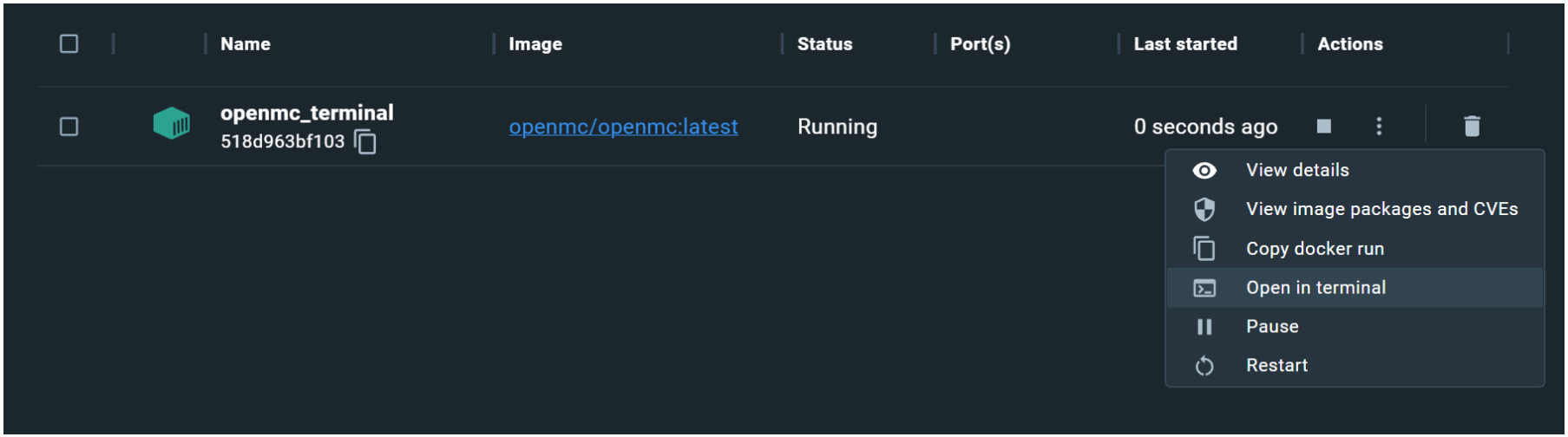


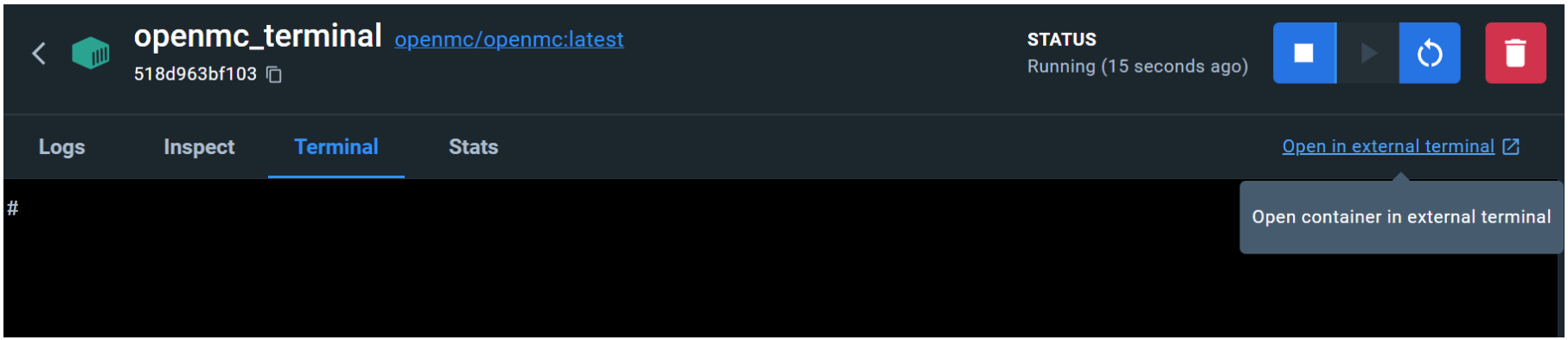
Be careful! From now one, you will work in the **Container environment** and not in the **Image environment**.

Whatever you do in the container, it stays in the container.

Docker Container - (III)

Now, *run* the container and *open in terminal*.





Docker Container (IV)

The first time you enter the terminal, the jupyter package has to be installed:

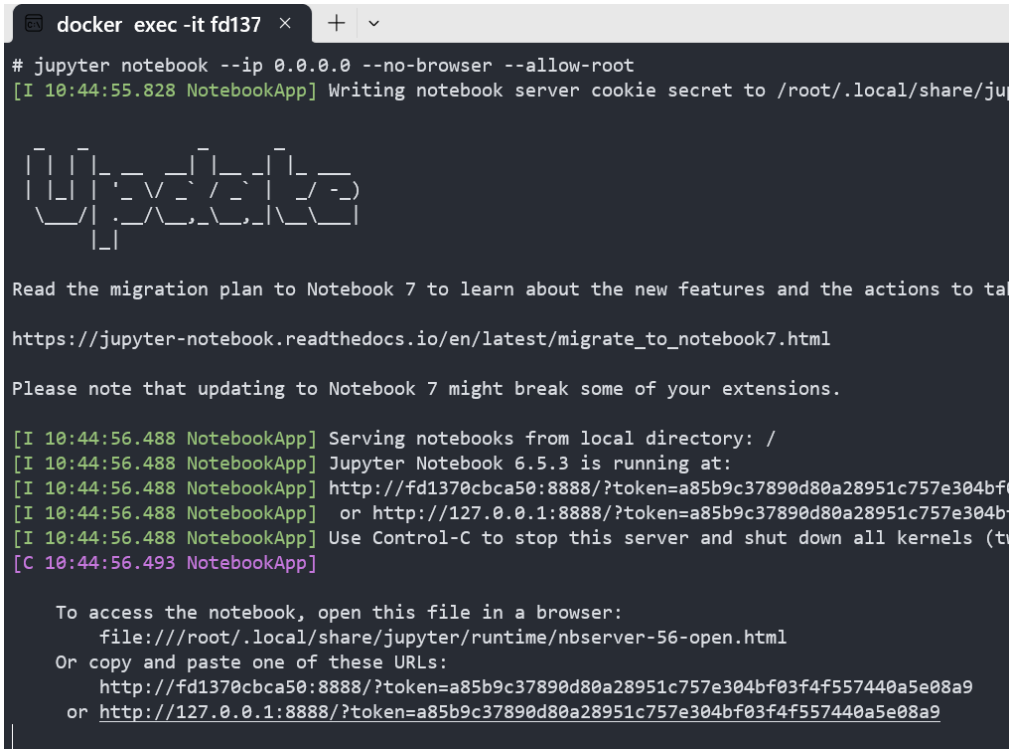
```
pip install jupyter
```

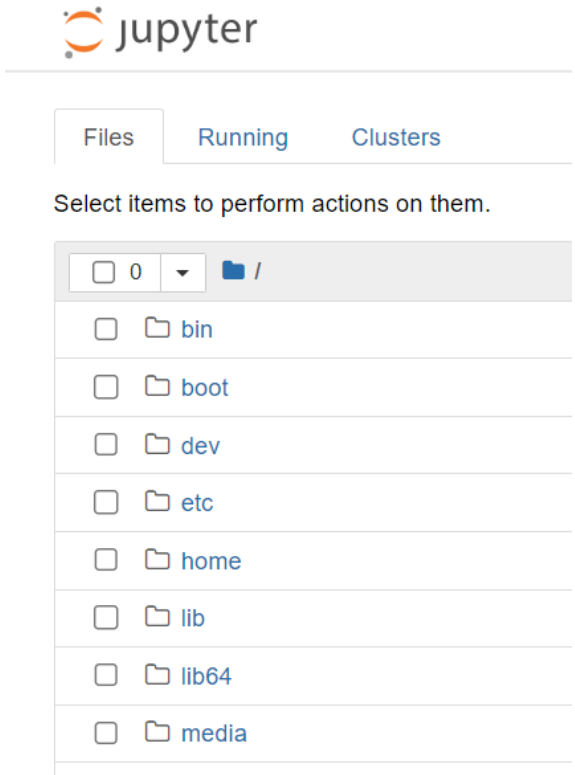
From now on, to open a script, paste this in the container terminal:

```
jupyter notebook --ip 0.0.0.0 --no-browser --allow-root
```

Docker Container - Jupyter

A couple of links will show. Following one of them, we will be linked to the jupyter view of the Container.

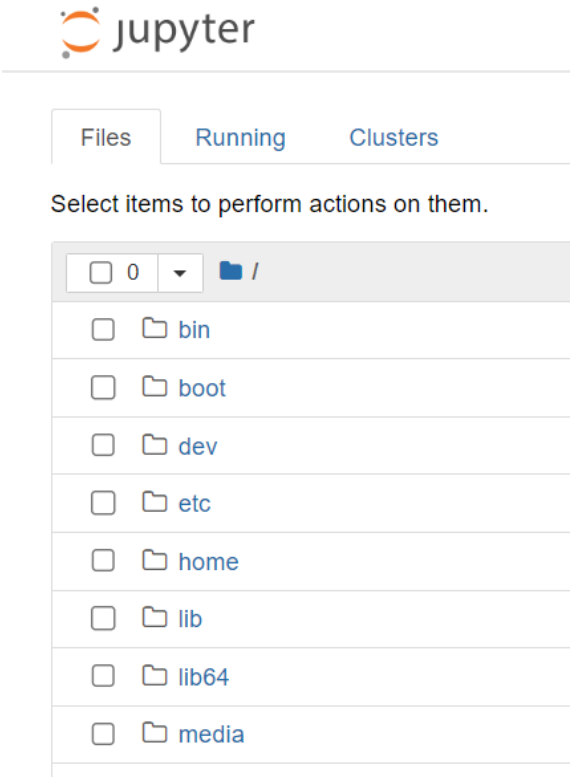




Docker Container - Jupyter

The installation is complete!

- From the Jupyter window, it is possible to create new files and run scripts;
- To check that you are working in an environment where OpenMC is correctly installed create a new Python Notebook (i.e., *file.ipynb*) and in the first cell write `import openmc`. If the cell runs, you can start working with the environment!



Final remarks

- All the saved files remain in the created container;
- In order to re-enter in the same container (or, better, to open the same jupyter window):
  1. Open the already created container as in slide n9;
  2. Paste in the terminal the line:

```
jupyter notebook --ip 0.0.0.0 --no-browser --allow-root
```

By so doing, it is possible to continue working on the same files!

Mac/Linux

Unix installation

For Linux and Mac (not Apple Silicon), OpenMC can be installed directly on the laptop and it is possible to work with VisualStudio

1. (Pre-requisite) Install the `Conda` package: it should be installed with one of the following installers: Miniconda or Anaconda;
2. Follow the instructions from the official OpenMC website, installation using Conda.

With Google Colab

How does it work?

1. Load the notebook *colab_openmc.ipynb* in Colab
2. The first cells will install a temporary OpenMC and load the nuclear libraries (some cells may be runned twice)
3. The notebook is ready to work! Keep in mind that whenever you restart the kernel/session, all the steps must be repeated (also, all the saved files are deleted).

Setting XS

Nuclear libraries

- **Windows**: the Docker environment already has some cross section within the container (nndc data library). However, up to version 0.13.2, these cross sections are evaluated at one temperature (~300 K). So, if calculations want to exploit higher temperatures, different XS has to be properly set in the container. Being the container a Linux environment, follow that instructions from the container terminal.
- **Linux/Mac**:
  1. Download Cross Section data from here and store them in the laptop;
  2. Within each code, before exporting the `materials` object to xml, add the following line[2]:

```
materials = openmc.Materials([mat1, mat2, ...])
materials.cross_sections = '/path/to/nucleardata/cross_sections.xml'
materials.export_to_xml()
```

[2] /path/to/nucleardata/ is the path pointing to the file *cross_sections.xml*