

# Conda Hands-on

## 7 June 2024

Conda is a powerful command line tool for package and environment management that runs on Windows, macOS, and Linux.

Conda can install entire software stacks such as Python, Python packages, R, R libraries, Java programs, C and C++ programs and libraries, Perl programs. It can be adapted for many use cases, but it is mainly oriented towards the scientific community.

Conda can install complex software stacks on a system without needing root privileges.

Two keywords you need to be familiar with before starting: **environment** and **channel**

**Environment:** A collection of packages (libraries, programs, etc...). Conda organizes the packages that you install into “environments”. A package can be part of different environments. You **create** an environment by specifying the software you would like to have. When you **activate** the environment, you can use the packages you have installed freely. When you **deactivate** it, your system will revert to its original state.

Environments give you the opportunity to test multiple versions of a software avoiding conflicts. It is also a good idea to have a separate environment for each of your projects.

**Channels** are the remote locations where packages are stored and that you can use to download them on your computer.

By default, packages are automatically downloaded and updated from the default channel, curated, built, maintained, and served by Anaconda (a software company)

Another channel that we will use is **conda-forge**, that is instead a channel managed by a community. This channel contains the **Copernicus Marine Service Toolbox** that we are going to install.

Let's start! We will now install conda and anaconda-navigator, a tool to administrate the conda environments.

Open the following web page:

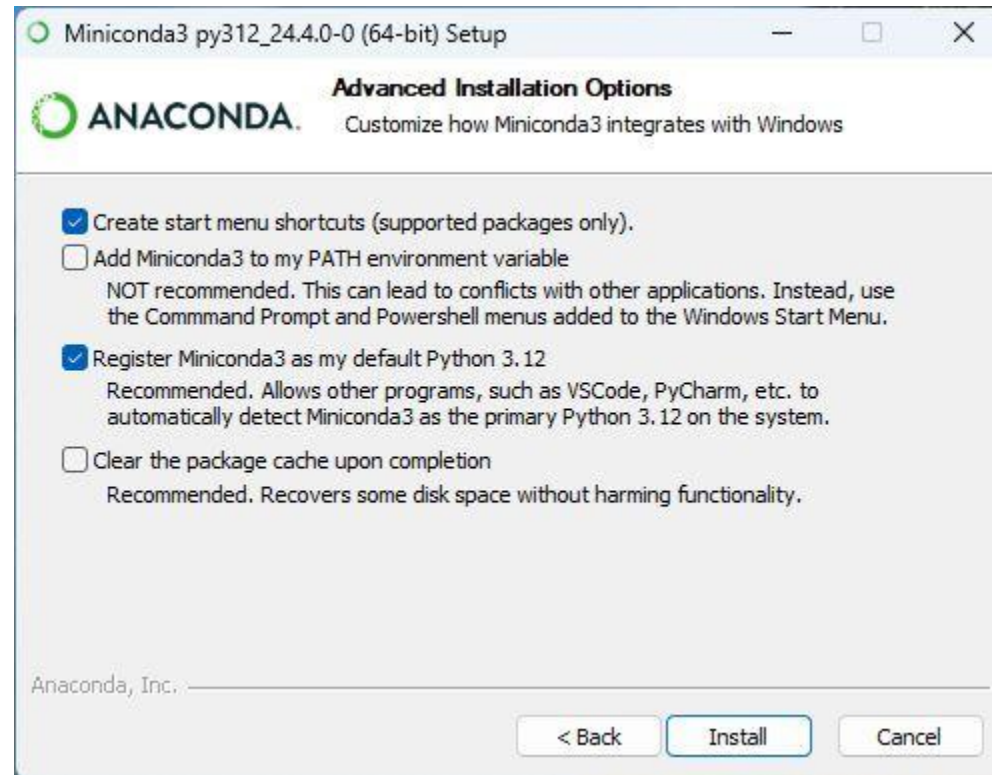
`https://docs.anaconda.com/free/miniconda/`

and download an appropriate installer for your system.

**Windows users:** Start the executable program

**Mac / Linux users:** Execute the script inside a console

**Windows users:** Choose the default options

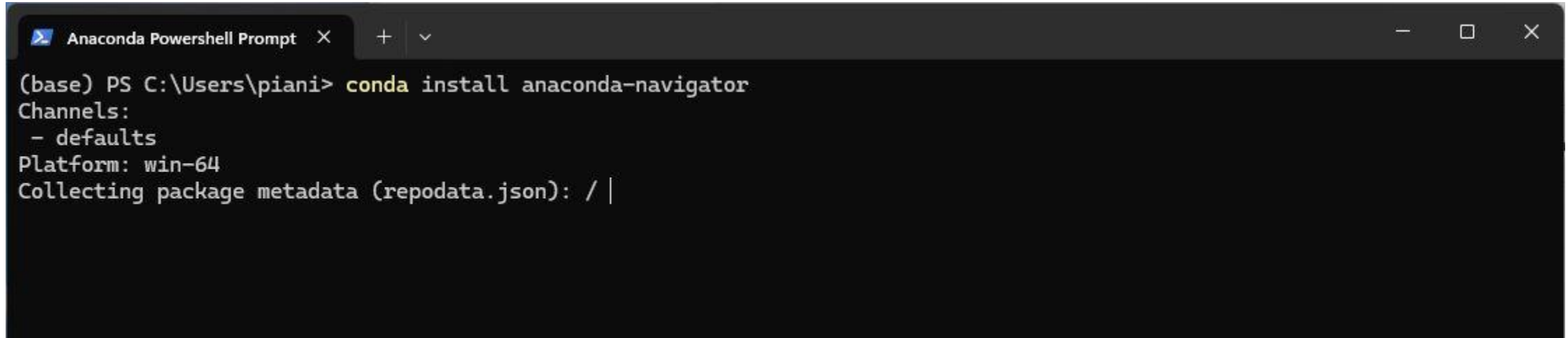


## Windows users:

Once the installation is complete, start the anaconda powershell prompt and run:

```
conda install anaconda-navigator
```

to install the navigator. This is the application that we will use to administrate conda

A screenshot of the Anaconda Powershell Prompt terminal window. The window title bar shows "Anaconda Powershell Prompt" with standard Windows window controls. The terminal content shows the command "conda install anaconda-navigator" being executed. Below the command, it displays "Channels:" followed by "- defaults", "Platform: win-64", and "Collecting package metadata (repodata.json): / |".

```
Anaconda Powershell Prompt X + v
(base) PS C:\Users\piani> conda install anaconda-navigator
Channels:
- defaults
Platform: win-64
Collecting package metadata (repodata.json): / |
```

## Mac / Linux users:

Execute the script:

- Accept the license
- Choose a path for the installation (usually, the default is fine)
- Do NOT execute conda init (choose no)

## Mac / Linux users:

In the last part of the text written by the installation script, there is the command to activate conda; it is something like

```
eval "$ (MINICONDAPATH/bin/conda shell.YOUR_SHELL_NAME hook) "
```

Execute this command to activate conda. Substitute MINICONDAPATH with the path where you have installed conda and YOUR\_SHELL\_NAME with the name of your shell (usually, bash or zsh). If you don't know what shell you are using, just type:

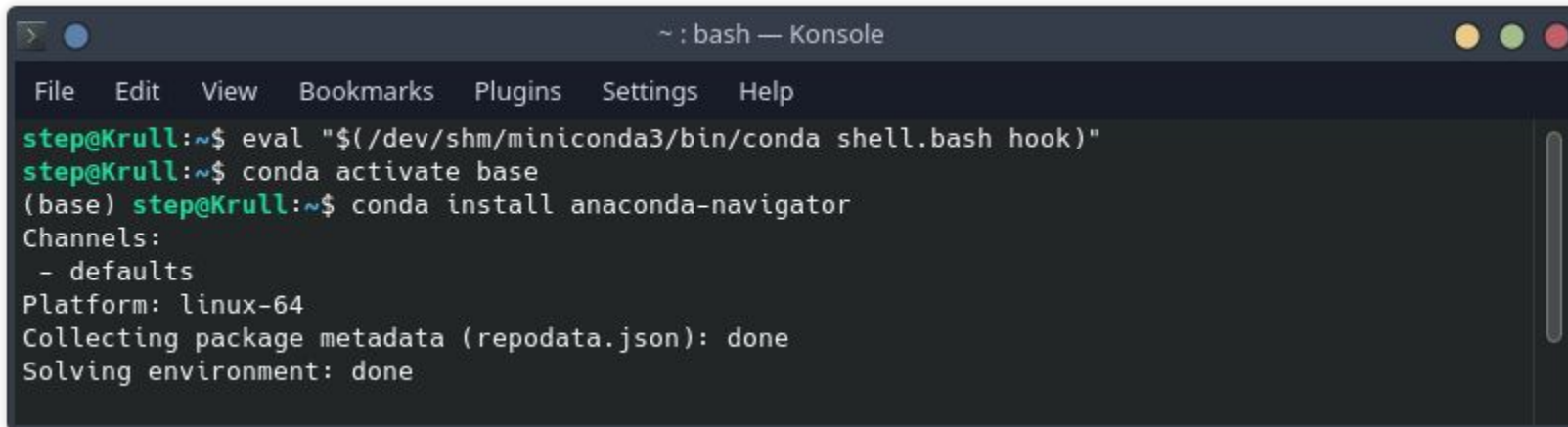
```
echo $SHELL
```

## Mac / Linux users:

Do not forget the command that you used to activate conda, because you will need that command every time you want to use conda.

Now run the following commands to install the anaconda-navigator:

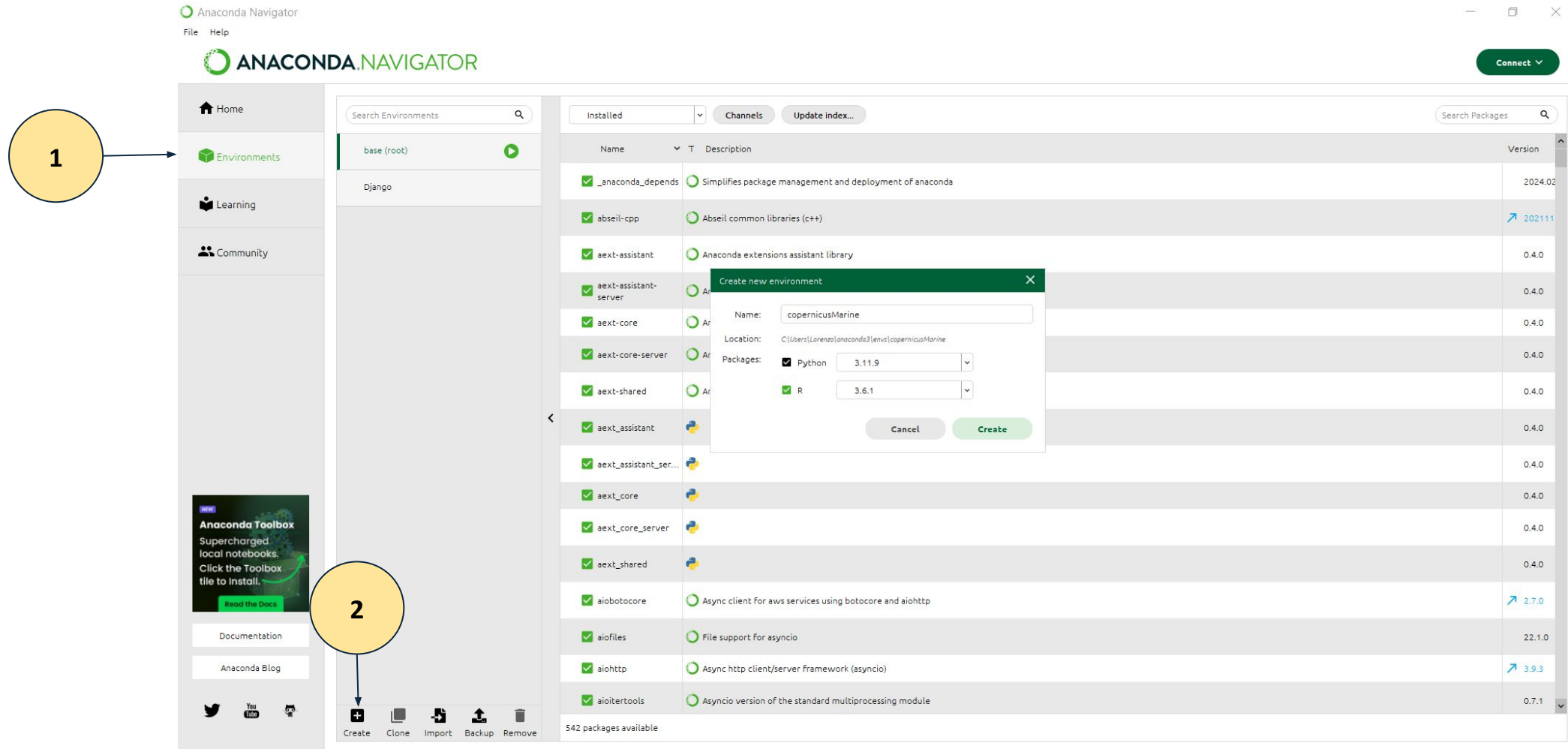
- `conda activate base`
- `conda install anaconda-navigator`
- `anaconda-navigator`



```
~ : bash — Konsole
File Edit View Bookmarks Plugins Settings Help
step@Krull:~$ eval "$(/dev/shm/miniconda3/bin/conda shell.bash hook)"
step@Krull:~$ conda activate base
(base) step@Krull:~$ conda install anaconda-navigator
Channels:
- defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
```



## Create a new environment called “copernicusmarine” (for example) with Python and R



1

2

Anaconda Navigator

File Help

ANACONDA.NAVIGATOR

Connect

Home

Environments

Learning

Community

Search Environments

base (root)

Django

Installed

Channels

Update index...

Search Packages

Create new environment

Name: copernicusMarine

Location: C:\Users\Lorenzo\anaconda3\envs\copernicusMarine

Packages: ☒ Python 3.11.9 ☒ R 3.6.1

Cancel Create

542 packages available

Name	Description	Version
✓ _anaconda_depends	Simplifies package management and deployment of anaconda	2024.02
✓ abseil-cpp	Abseil common libraries (c++)	202111
✓ aext-assistant	Anaconda extensions assistant library	0.4.0
✓ aext-assistant-server		0.4.0
✓ aext-core		0.4.0
✓ aext-core-server		0.4.0
✓ aext-shared		0.4.0
✓ aext_assistant		0.4.0
✓ aext_assistant_ser...		0.4.0
✓ aext_core		0.4.0
✓ aext_core_server		0.4.0
✓ aext_shared		0.4.0
✓ aiobotocore	Async client for aws services using botocore and aiohttp	2.7.0
✓ aiofiles	File support for asyncio	22.1.0
✓ aiohttp	Async http client/server framework (asyncio)	3.9.3
✓ aiortools	Asyncio version of the standard multiprocessing module	0.7.1

Add a new channel named “conda-forge” (this time the name is mandatory)

File Help

ANACONDA.NAVIGATOR

Connect

Home

Environments

Learning

Community

Search Environments

base (root)

Django

copernicusMarine

Channels

Update index...

Search Channels

Manage channels you want Navigator to include.

Add...

Cancel

Update

Name	Type	Version
<input checked="" type="checkbox"/> bash_kernel		0.7.2
<input type="checkbox"/> Fortran_kernel		0.1.7
<input type="checkbox"/> jupyter_kernel_g...	Jupyter	3.0.1
<input type="checkbox"/> jupyter_kernel_test	Jupyter	0.7.0
<input type="checkbox"/> matlab_kernel		0.17.1
<input type="checkbox"/> nb_conda_kernels		2.5.1
<input type="checkbox"/> octave_kernel		0.36.0
<input type="checkbox"/> sas_kernel		2.4.13
<input type="checkbox"/> stata_kernel		1.12.2
<input type="checkbox"/> wwt_kernel_data...		0.3.0

10 packages available matching "kerne"

1

2

3

Anaconda Toolbox

Supercharged local notebooks. Click the Toolbox tile to install.

Read the Docs

Documentation

Anaconda Blog

Create Clone Import Backup Remove

22°C Salsgite 16:08

Search for the “copernicusmarine” package on the right and install it inside your current environment

File Help

ANACONDA.NAVIGATOR

Connect

Home

Environments

Learning

Community

Search Environments

base (root)

Django

copernicusMarine

1

2

3

copernicusmarine

1.2.3

1 package available matching "coper"

Anaconda Toolbox  
Supercharged local notebooks.  
Click the Toolbox tile to install.  
Read the Docs

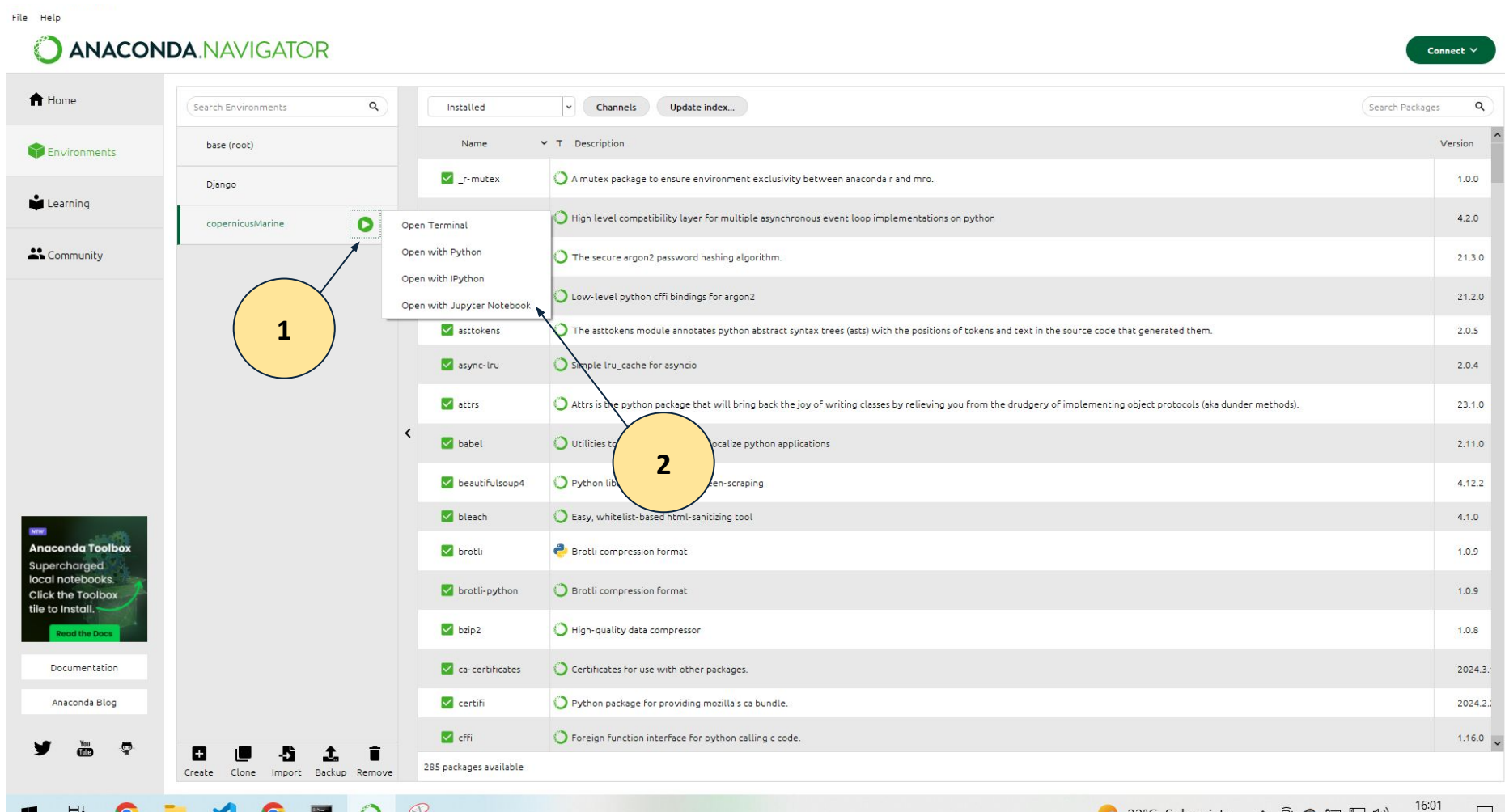
Documentation

Anaconda Blog

Create Clone Import Backup Remove

22°C Salento 16:09

## Start a **jupyter notebook** inside your new environment



The screenshot displays the Anaconda Navigator application interface. On the left sidebar, the 'Environments' section is active, showing a list of environments: 'base (root)', 'Django', and 'copernicusMarine'. A yellow circle with the number '1' highlights the 'copernicusMarine' environment, which has a green play button icon next to it. A context menu is open over this icon, with the option 'Open with Jupyter Notebook' selected. A second yellow circle with the number '2' highlights this menu option. The main panel on the right shows a list of installed packages for the 'copernicusMarine' environment, including \_r-mutex, aiohttp, asttokens, async-lru, attrs, babel, beautifulsoup4, bleach, brotli, brotli-python, bzip2, ca-certificates, certifi, and cffi. The bottom status bar indicates '285 packages available'.

File Help

ANACONDA.NAVIGATOR

Connect

Home

Environments

Learning

Community

Search Environments

base (root)

Django

copernicusMarine

Open Terminal

Open with Python

Open with IPython

Open with Jupyter Notebook

1

2

Search Packages

Installed Channels Update index...

Name	Description	Version
✓ _r-mutex	A mutex package to ensure environment exclusivity between anaconda r and mro.	1.0.0
✓ aiohttp	High level compatibility layer for multiple asynchronous event loop implementations on python	4.2.0
✓ argon2	The secure argon2 password hashing algorithm.	21.3.0
✓ argon2-cffi	Low-level python cffi bindings for argon2	21.2.0
✓ asttokens	The asttokens module annotates python abstract syntax trees (asts) with the positions of tokens and text in the source code that generated them.	2.0.5
✓ async-lru	Simple lru_cache for asyncio	2.0.4
✓ attrs	Attrs is the python package that will bring back the joy of writing classes by relieving you from the drudgery of implementing object protocols (aka dunder methods).	23.1.0
✓ babel	Utilities to localize python applications	2.11.0
✓ beautifulsoup4	Python library for web scraping	4.12.2
✓ bleach	Easy, whitelist-based html-sanitizing tool	4.1.0
✓ brotli	Brotli compression format	1.0.9
✓ brotli-python	Brotli compression format	1.0.9
✓ bzip2	High-quality data compressor	1.0.8
✓ ca-certificates	Certificates for use with other packages.	2024.3.1
✓ certifi	Python package for providing mozilla's ca bundle.	2024.2.1
✓ cffi	Foreign function interface for python calling c code.	1.16.0

285 packages available

Create Clone Import Backup Remove

23°C Salsgite 16:01