

EGI Notebooks and Replay services

Marco Rorro

AI Solutions Architect / EGI Foundation

TLP: WHITE Public

March 24th 2023

TGB Meeting

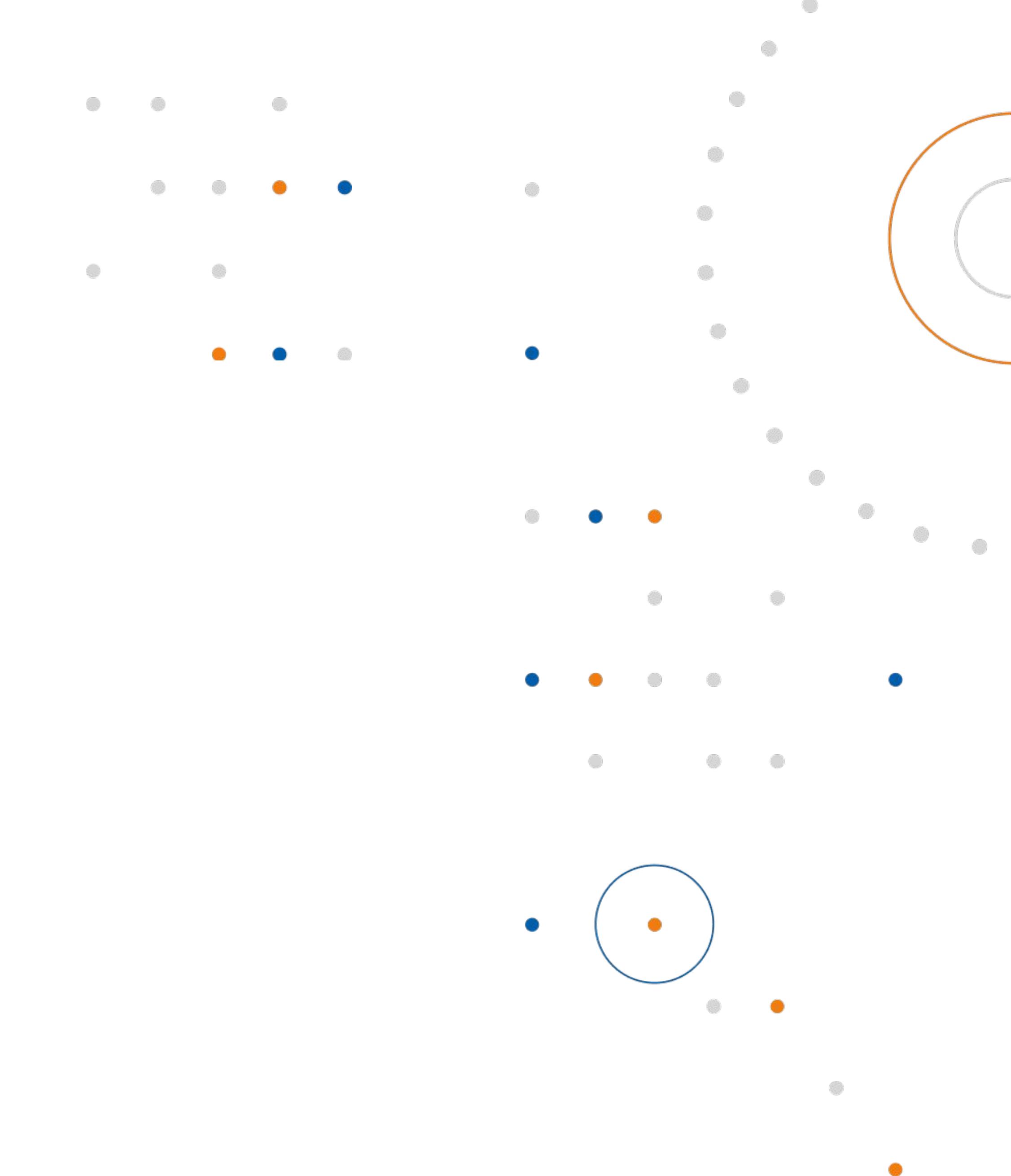
- **EGI Jupyter Notebooks**
 - Default Environments
 - User Custom Environments

- **EGI Replay**
 - A Binder service built on BinderHub



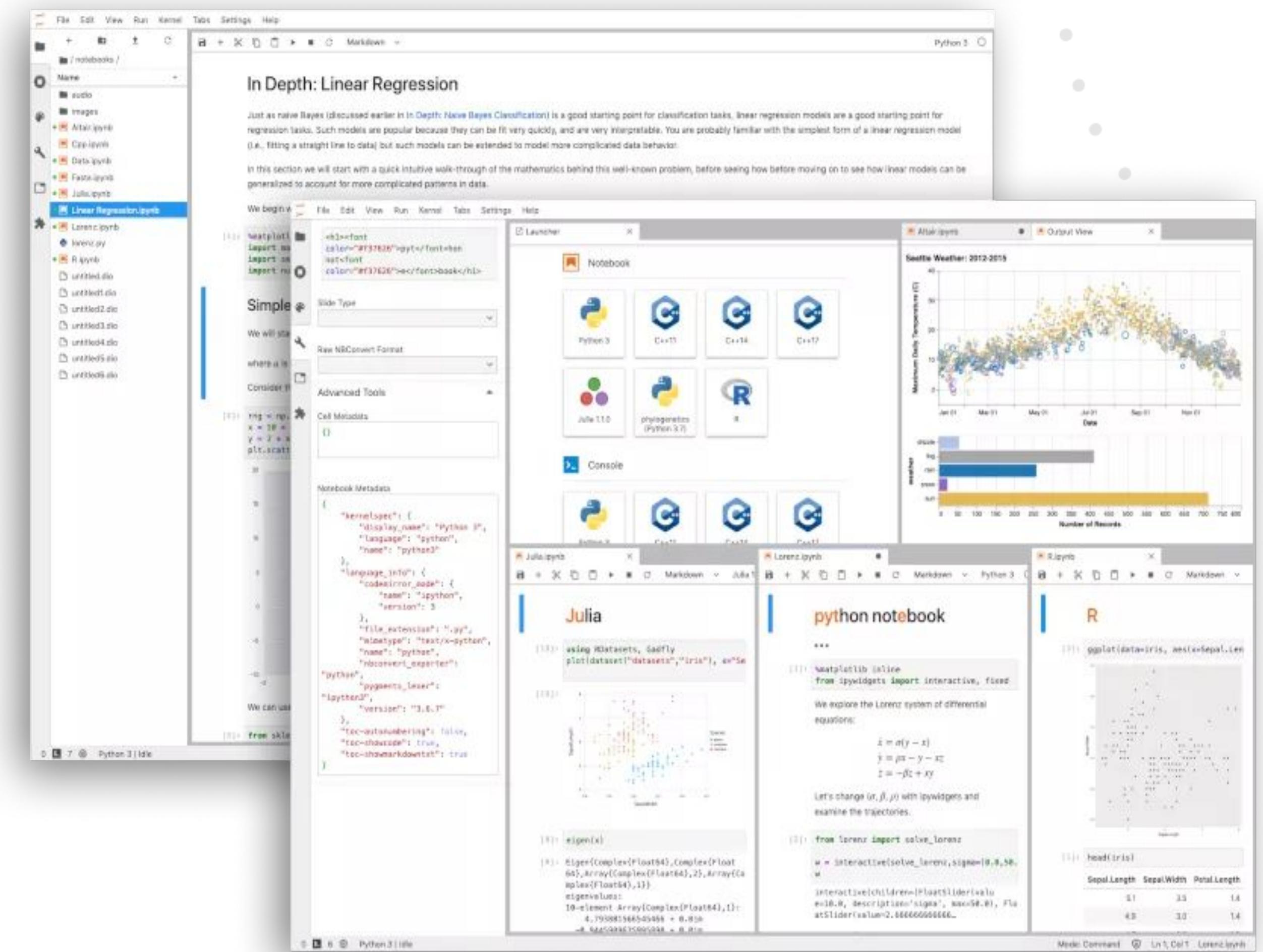
Interactive Computing

Jupyter Notebook



The Jupyter Notebook is an open-source **web application** that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.





EGI Notebooks

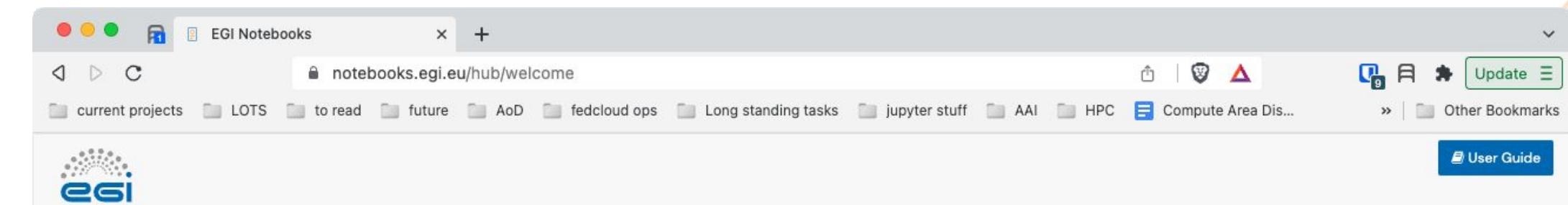


- **Jupyter hosted in the EGI Cloud**

- Offers Jupyter notebooks 'as a Service'
- One-click solution: login and start using

- **Main Features:**

- Easy access: Login with the EGI AAI Check-In service
- Persistent storage for notebooks
- Use EGI computing and storage resources from your notebooks



Notebooks

Notebooks is an environment based on [Jupyter](#) and the [EGI cloud service](#) that offers a browser-based, scalable tool for interactive data analysis. The Notebooks environment provides users with notebooks where they can combine text, mathematics, computations and rich media output.

Access requires a valid [EGI account](#) and enrolling to the [vo.notebooks.egi.eu VO](#).

 Continue with EGI Check-in

User communities/advanced users can have their customised EGI Notebooks service instance. EGI offers consultancy and support, as well as can operate the setup. Order a [community notebooks instance](#) via the EOSC Marketplace.



cesnet

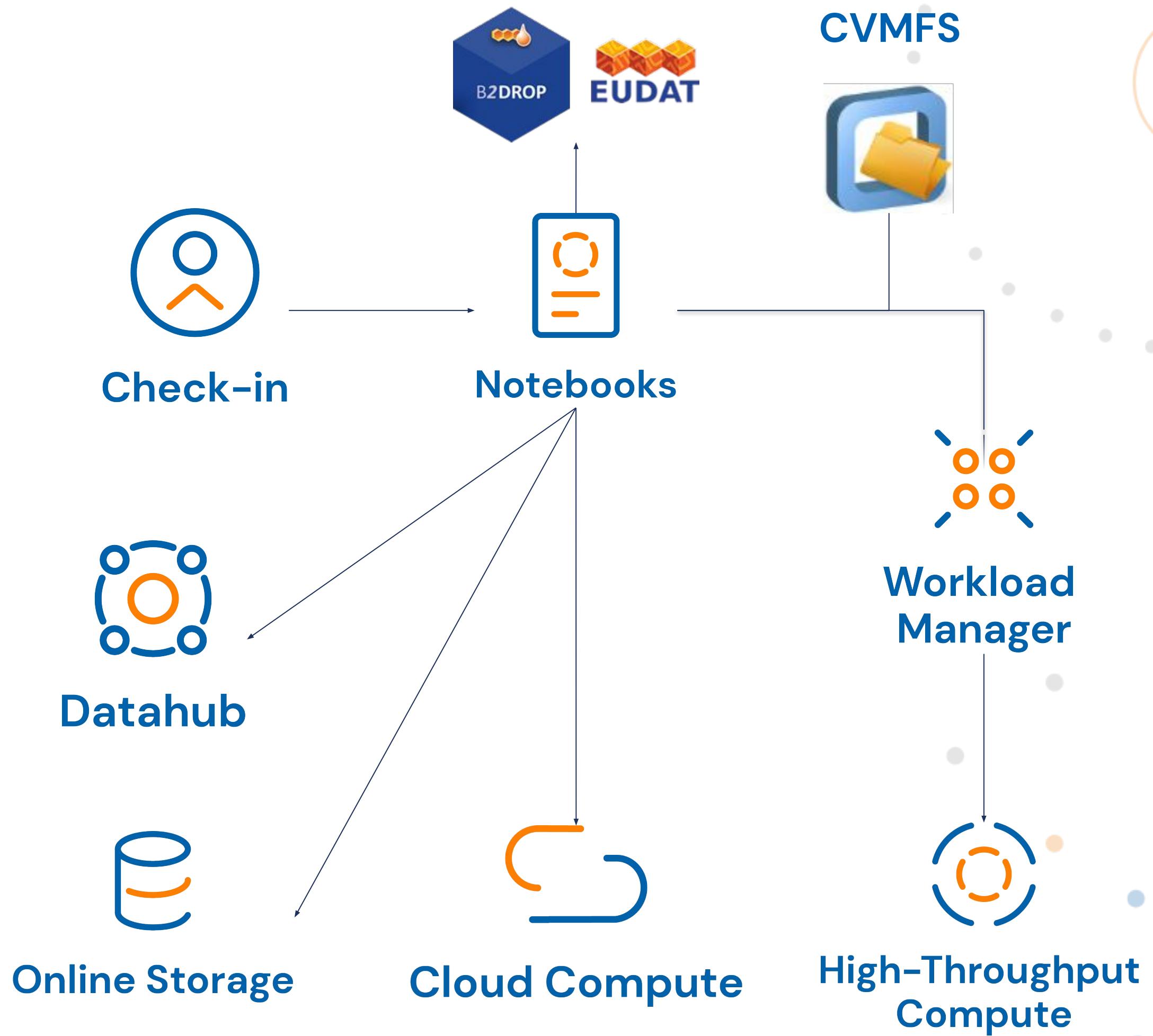
cesnet

Notebooks is a service provided by CESNET, co-funded by EGI-ACE.

[Privacy policy](#) | [Terms of use](#)

EGI Notebooks integrations

- Runs on EGI Cloud providers (**Cloud Compute / Online Storage**)
- Uses **Check-in** for authentication + access to other services
- Makes user-level software available from **CVMFS**
 - DIRAC client for submission of jobs to Workload Manager / High Throughput Compute
 - Fedcloudclient for interaction with Cloud Compute
- Transparent access to Datahub spaces
- Access to 3rd party services: **B2DROP**



Notebooks for researchers: getting access

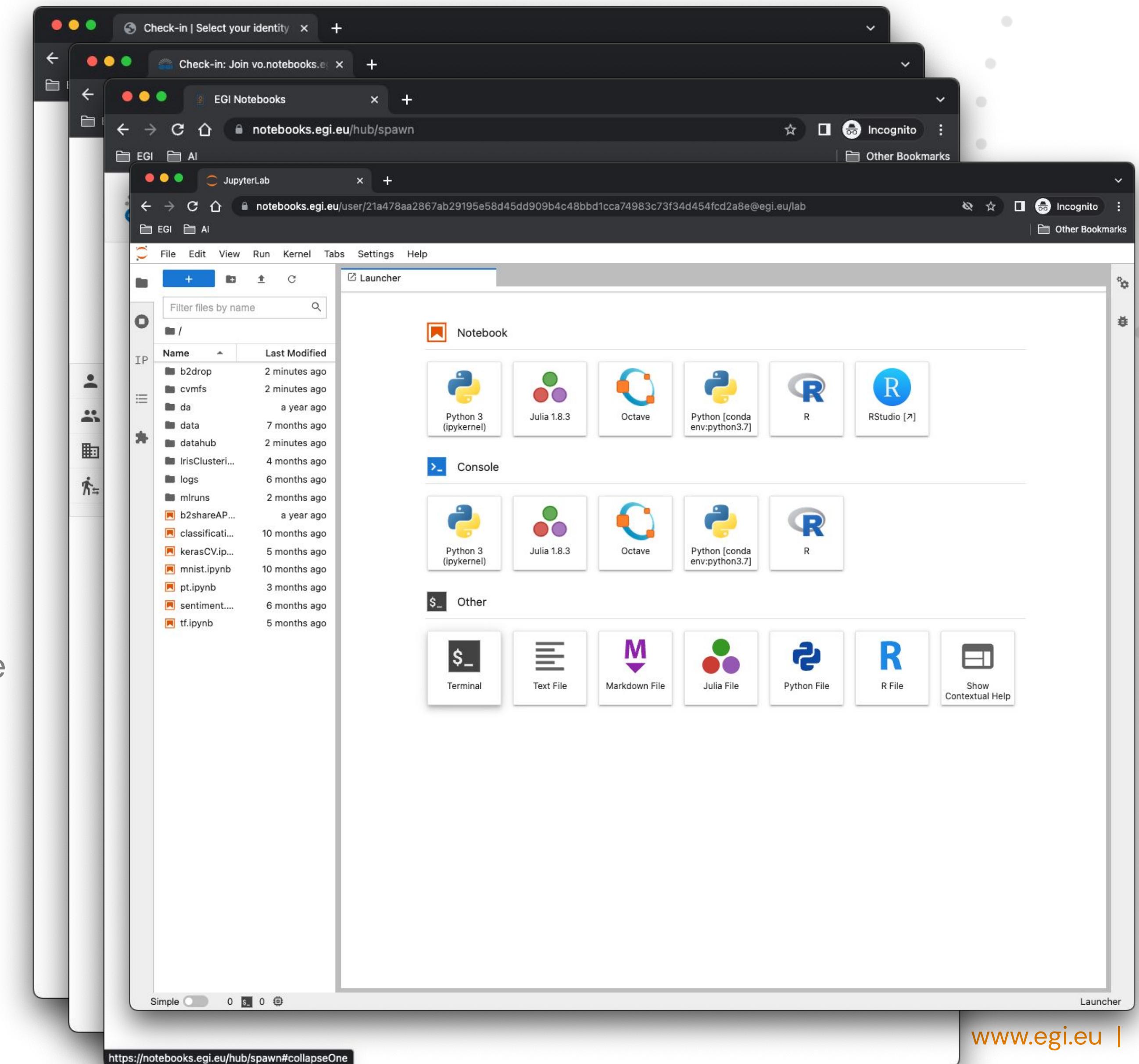
1. Get an EGI account

2. Enroll to one of the supported VOs (Virtual Organisations):

- vo.notebooks.egi.eu
- vo.access.egi.eu
- vo.reliance-project.eu
- eiscat.se
- vo.environmental.egi.eu
- ...

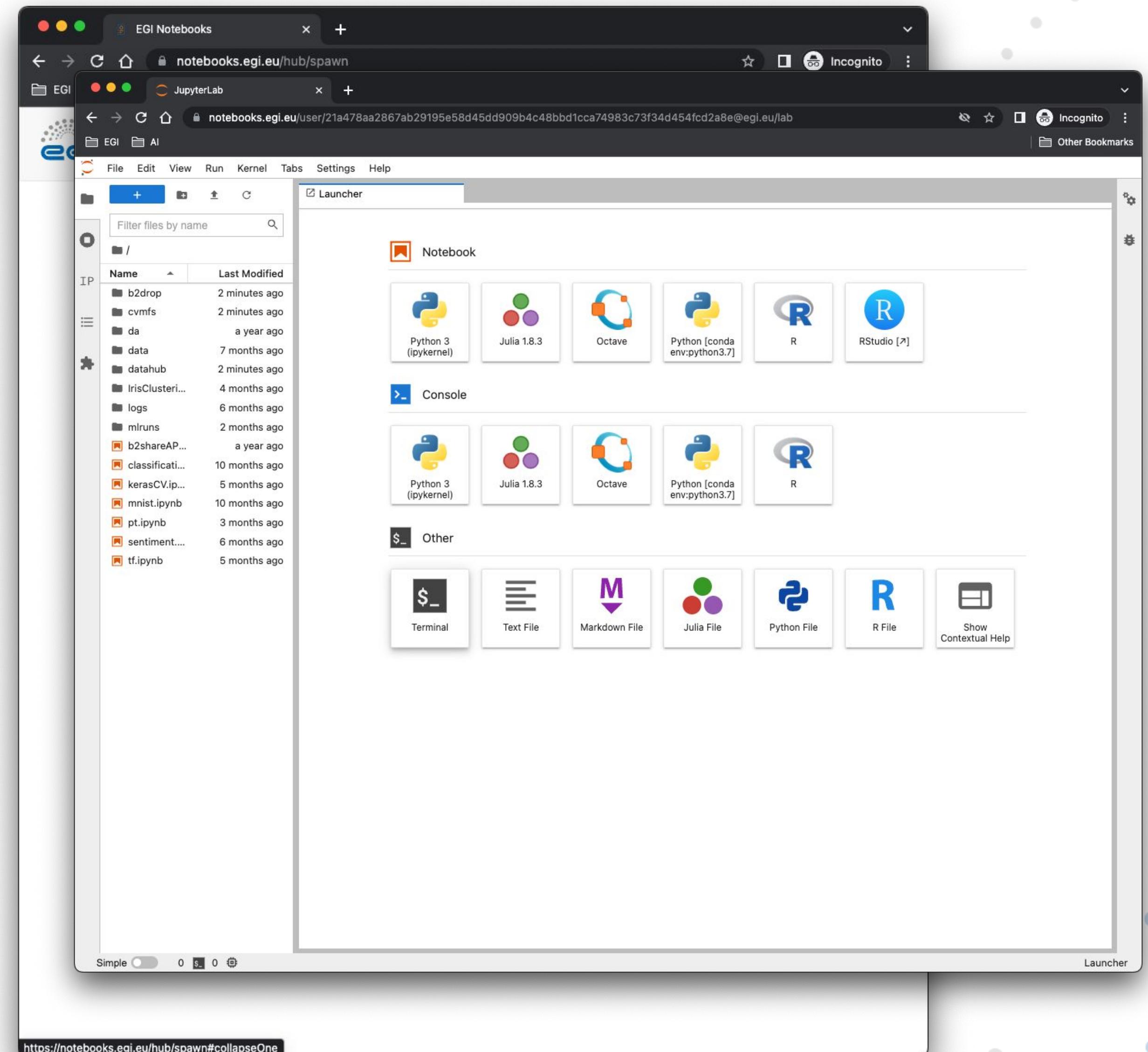
3. Access <https://notebooks.egi.eu/>

- Default: Python, R, Julia, Octave and a wide range of data science libraries
- MATLAB (Basic/Full): Run MATLAB on EGI resources (requires a license!)
- Community environments (RELIANCE, EISCAT): tuned to meet the needs of specific user communities



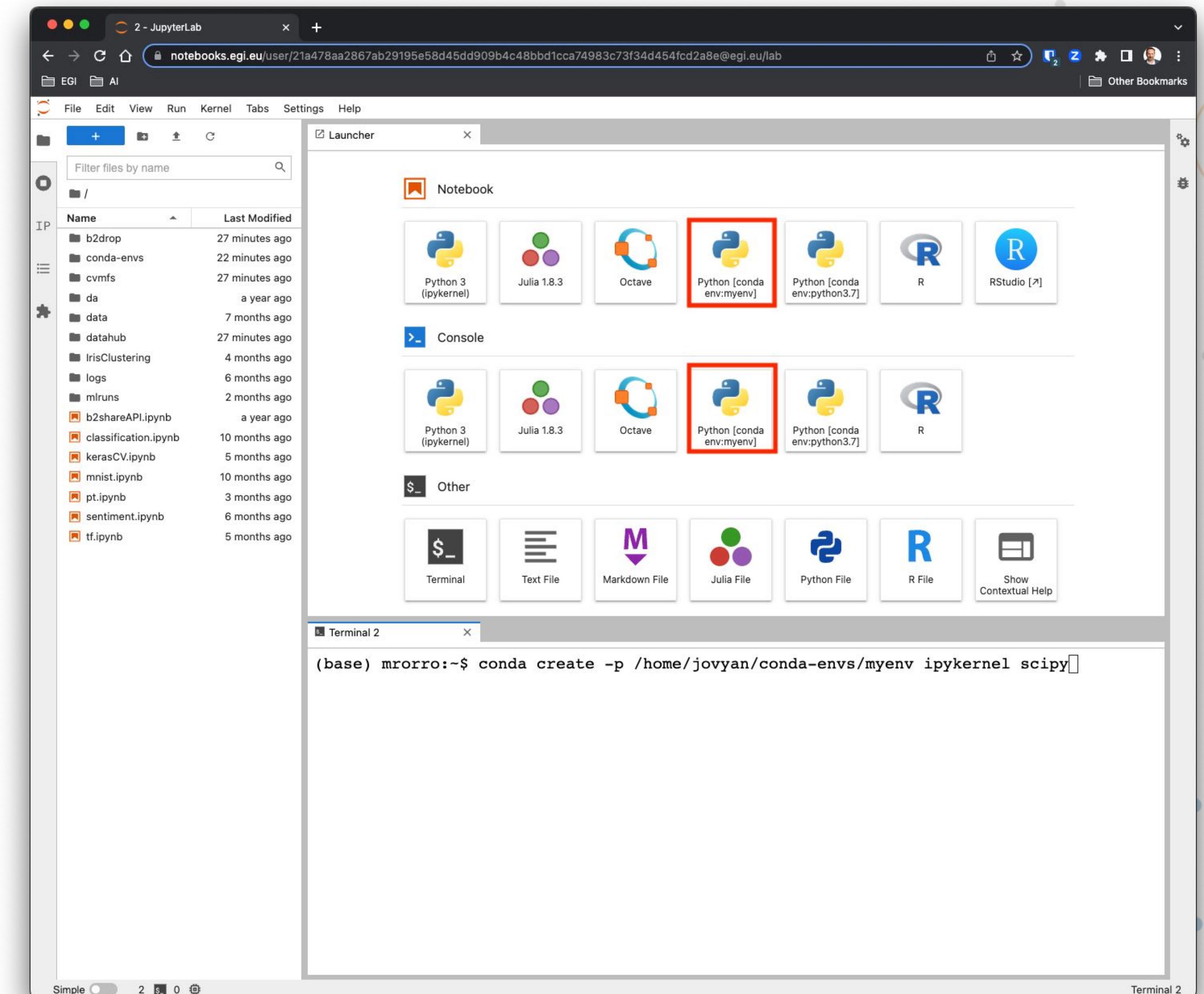
- Relies on custom images

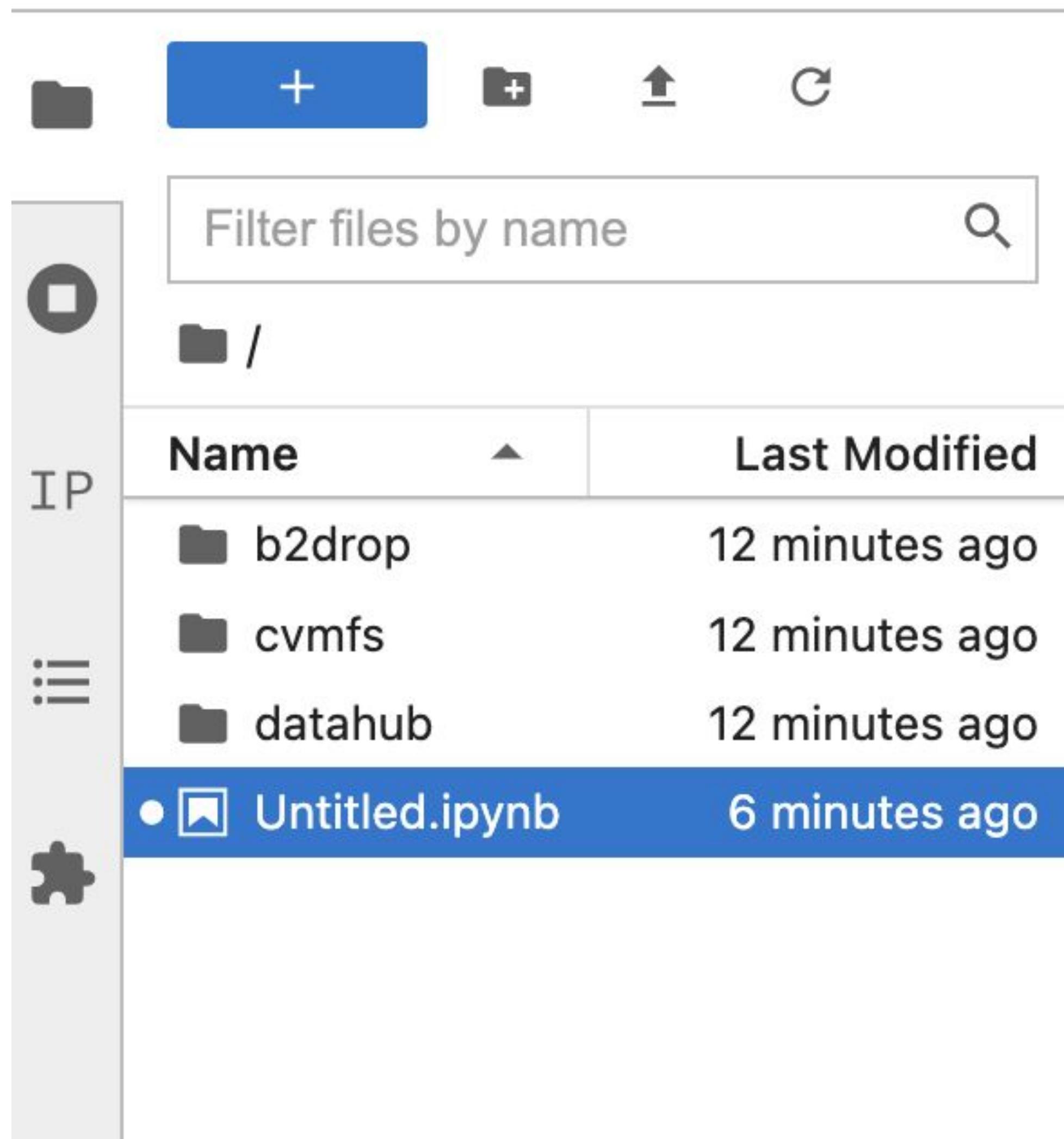
- Dockerfiles are available at [EGI Notebooks images git repository](#)
- GitHub actions automatically build for every commit pushed to the repository to [eginotebooks @ dockerhub](#)
- Builds from [Jupyter datascience-notebook](#) and adds a wide range of libraries
- Uses multiple profiles (`profileList` JupyterHub Helm chart option) to let users select their environment
- defining a custom pre-spawn hook that populates the profile list based on user identity
- Populates the profiles list based on the user VO memberships defining a pre-spwan hook



- Allow users to create their own conda environments

- Rely on the nb_conda_kernels package
- Install user environments to a folder within \$HOME
- Make sure to install a kernel (ipykernel) for it to show automatically





- **Persistent home**

- Can be used to store data (10GB limit)
- Files will be kept even if the notebook server dies

- **datahub**

- Access to your accessible spaces in datahub
- Share data and assign PIDs to shared spaces

- **b2drop**

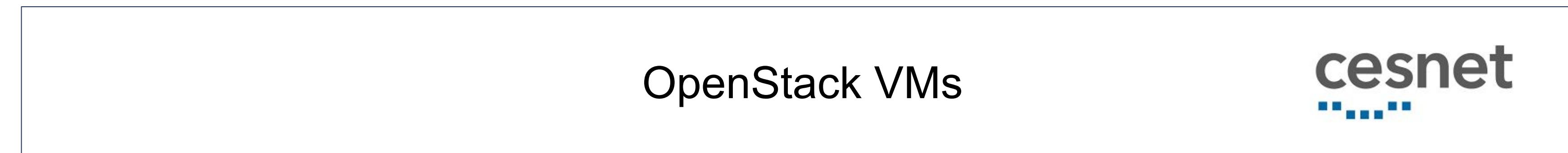
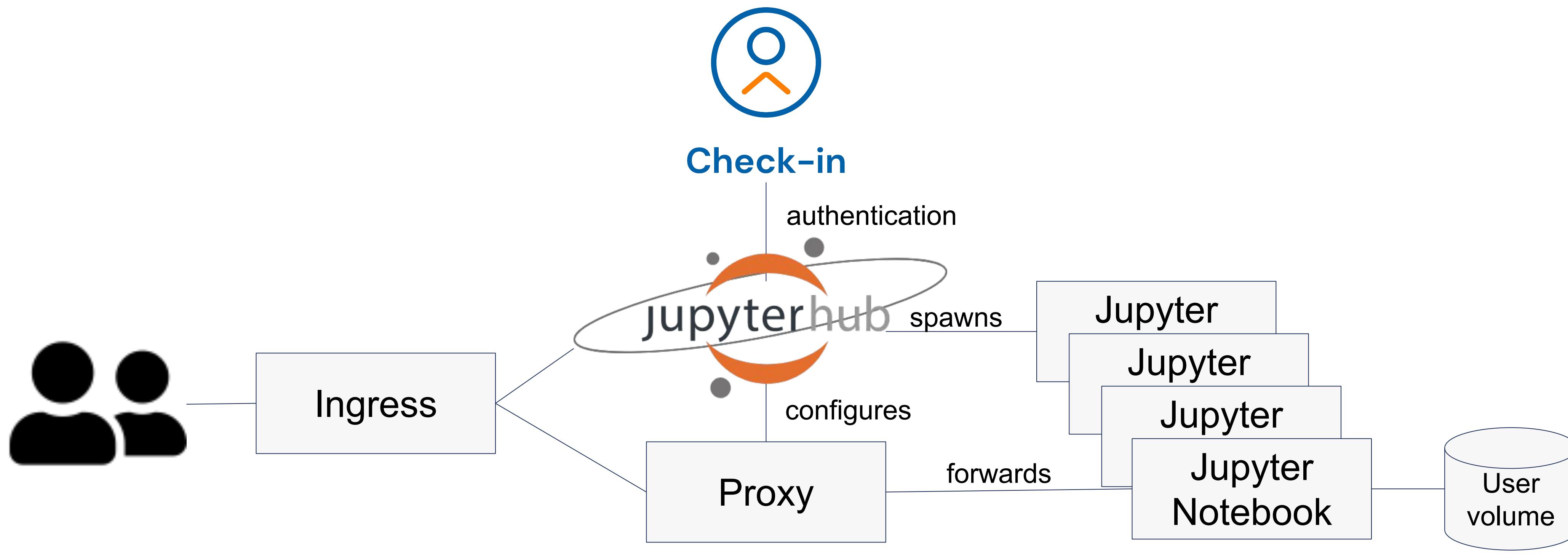
- Access data stored in EUDAT's B2DROP service

- **cvmfs**

- Selected CVMFS repositories available
- Easy to add community specific ones

- **nbgitpuller**

- Get code from any git repository from a single URL
- <https://hub.jupyter.org/nbgitpuller/link.html>

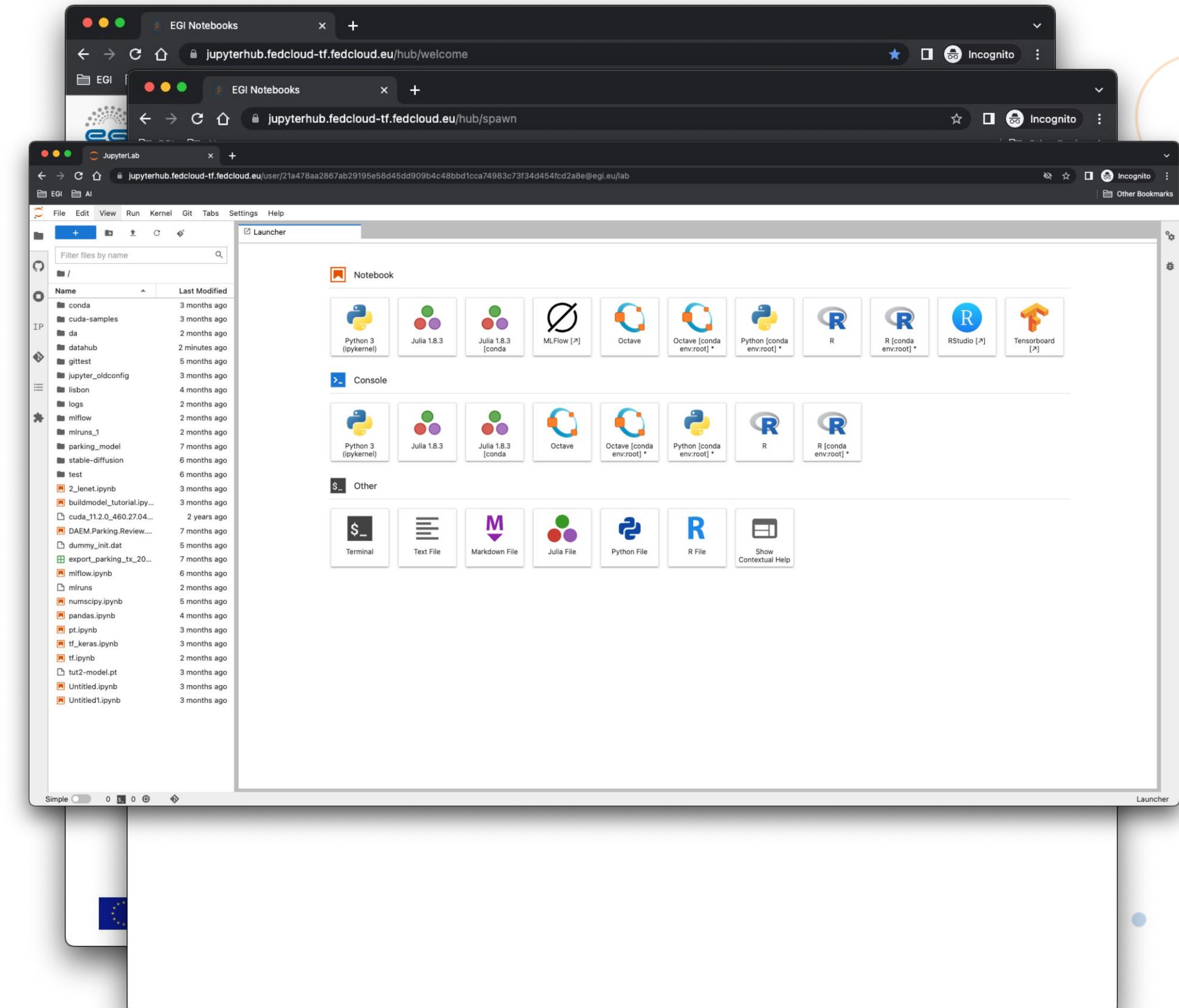


- **User communities can have their customised EGI Notebooks service instance**

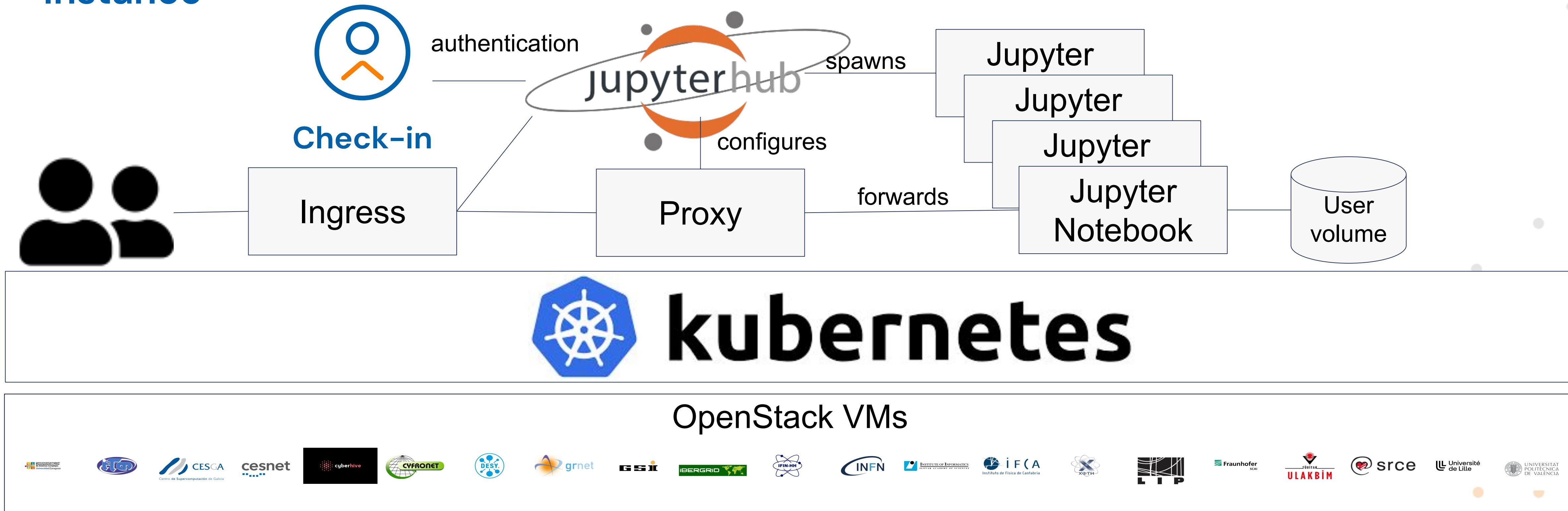
- custom resource limits
- use the community's own Virtual Organisation
- have fine grained control on who can access the instance
- add custom libraries into Jupyter

- **EGI currently operates instances for:**

- D4Science. These instances are accessed through specific Gateways: SoBigData, Blue-Cloud, D4Science Services and EOSC-Pillar.
- AI4PublicPolicy
- Training for training events



1. Communities instance can be deployed at one of the EGI cloud providers
2. Terraform creates the VMs
3. Ansible configures the VMs, install kubernetes and deploys the notebooks instance



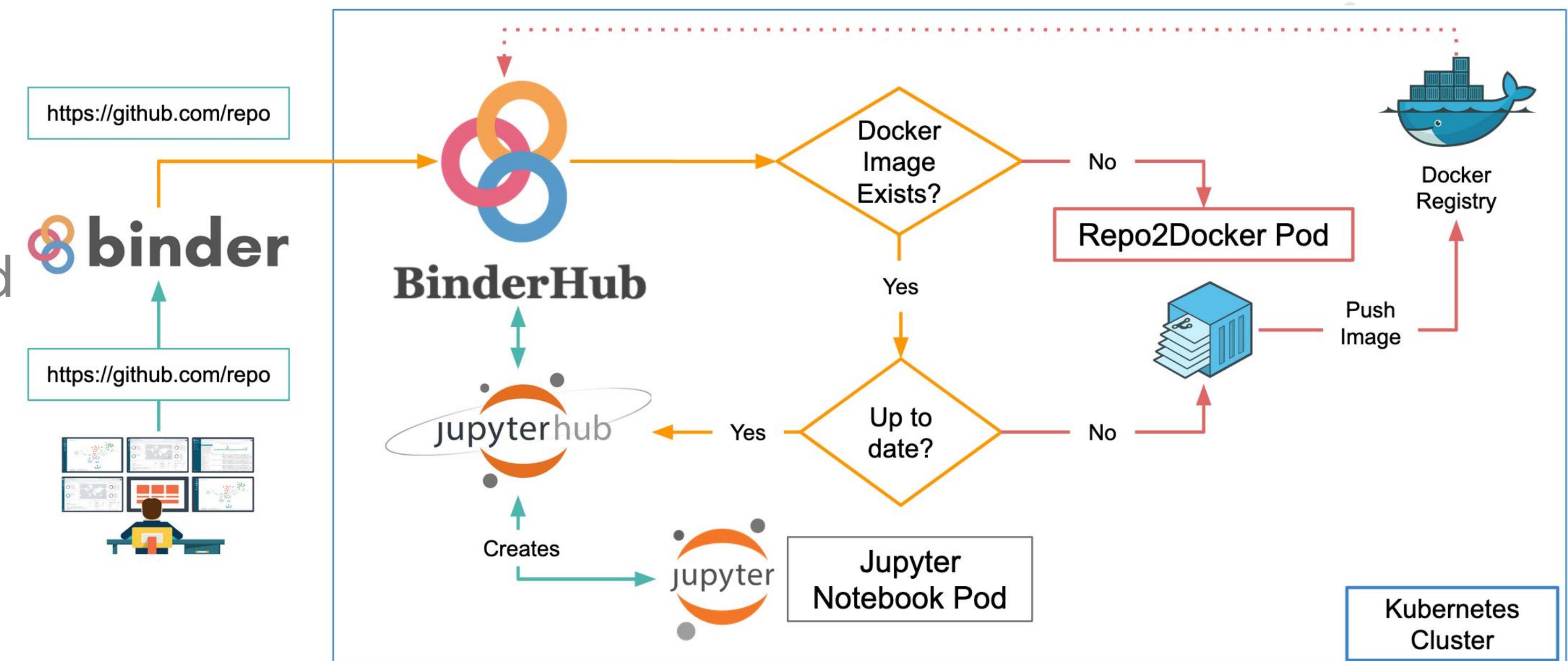


Reproducible Open Science

Replay

The Binder Project allow to create shareable, interactive, reproducible environments.

The service builds on BinderHub, an Open Source tool that allows to build docker images from a public repository like Git and then provides interactive session through JupyterHub of those images.





BinderHub hosted by EGI

- Offered 'as Service'
- Same access conditions as EGI Notebooks

Main Features:

- Use any binder-compatible repository
- Reproduce your notebooks with access to EGI resources (e.g. datahub)
- No hard limits on sessions duration, customisable resource limits for users/communities

The screenshot shows a web browser window with two tabs. The top tab is titled 'EGI Replay' and the bottom tab is titled 'Binder'. The URL in the address bar is 'replay.notebooks.egi.eu'. The main content area displays the 'binder' logo and the text: 'Turn a Git repo into a collection of interactive notebooks. Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.' Below this, there is a section titled 'Build and launch a repository' with fields for 'GitHub repository name or URL' (set to 'GitHub'), 'Git ref (branch, tag, or commit)' (set to 'HEAD'), and 'Path to a notebook file (optional)'. A 'launch' button is present. Further down, there is a 'Copy the URL below and share your Binder with others:' field and a 'Fill in the fields to see a URL for sharing your Binder.' placeholder. At the bottom, there is a link to 'Expand to see the text below, paste it into your README to show a binder badge.'

How it works

- ① Enter your repository information
Provide in the above form a URL or a GitHub repository that contains Jupyter notebooks, as well as a branch, tag, or commit hash. Launch will build your Binder repository. If you specify a path to a notebook file, the notebook will be opened in your browser after building.
- ② We build a Docker image of your repository
Binder will search for a dependency file, such as requirements.txt or environment.yml, in the repository's root directory (more details on more complex dependencies in documentation). The dependency files will be used to build a Docker image. If an image has already been built for the given repository, it will not be rebuilt. If a new commit has been made, the image will automatically be rebuilt.
- ③ Interact with your notebooks in a live environment!
A JupyterHub server will host your repository's contents. We offer you a reusable link and badge to your live repository that you can easily share with others.



A Binder repository

- A code repository that contains:**
- **Code to reproduce (i.e. set of notebooks)**
 - **+ description of the software runtime (e.g. a conda environment)**
 - **+ any auxiliary files needed to run the code**

The screenshot shows a browser window with two tabs. The left tab is a GitHub repository page for 'enolfc/isgc-2023-enes' containing files like 'QuickStart_GeoPy.ipynb', 'LICENSE', and 'README.md'. The right tab is a JupyterLab session titled 'QuickStart_GeoPy.ipynb' showing Python code for geolocation and mapping using GeoPy, Folium, and hvplot, with a resulting map of Taipei.

Data Visualization with GeoPy, Folium and hvplot

GeoPy: a Python client to locate the coordinates of addresses, cities, countries, and landmarks across the globe using third-party geocoders and other data sources. <https://geopy.readthedocs.io/en/stable/>

Folium: to visualize data that's been manipulated in Python on an interactive leaflet map. <http://python-visualization.github.io/folium/>

hvPlot: a high-level plotting API built on HoloViews that provides a general and consistent API to create interactively explorable Bokeh plots with panning, zooming, hovering, and clickable/selectable legends. <https://hvplot.holoviz.org/>

```
[1]: import numpy as np
import pandas as pd
import xarray as xr
import intake
from ipywidgets import widgets
from geopy.geocoders import Nominatim
import folium
import hvplot.pandas
import warnings
warnings.filterwarnings("ignore")
from os.path import expanduser
home = expanduser("~/")
```

Choose a location

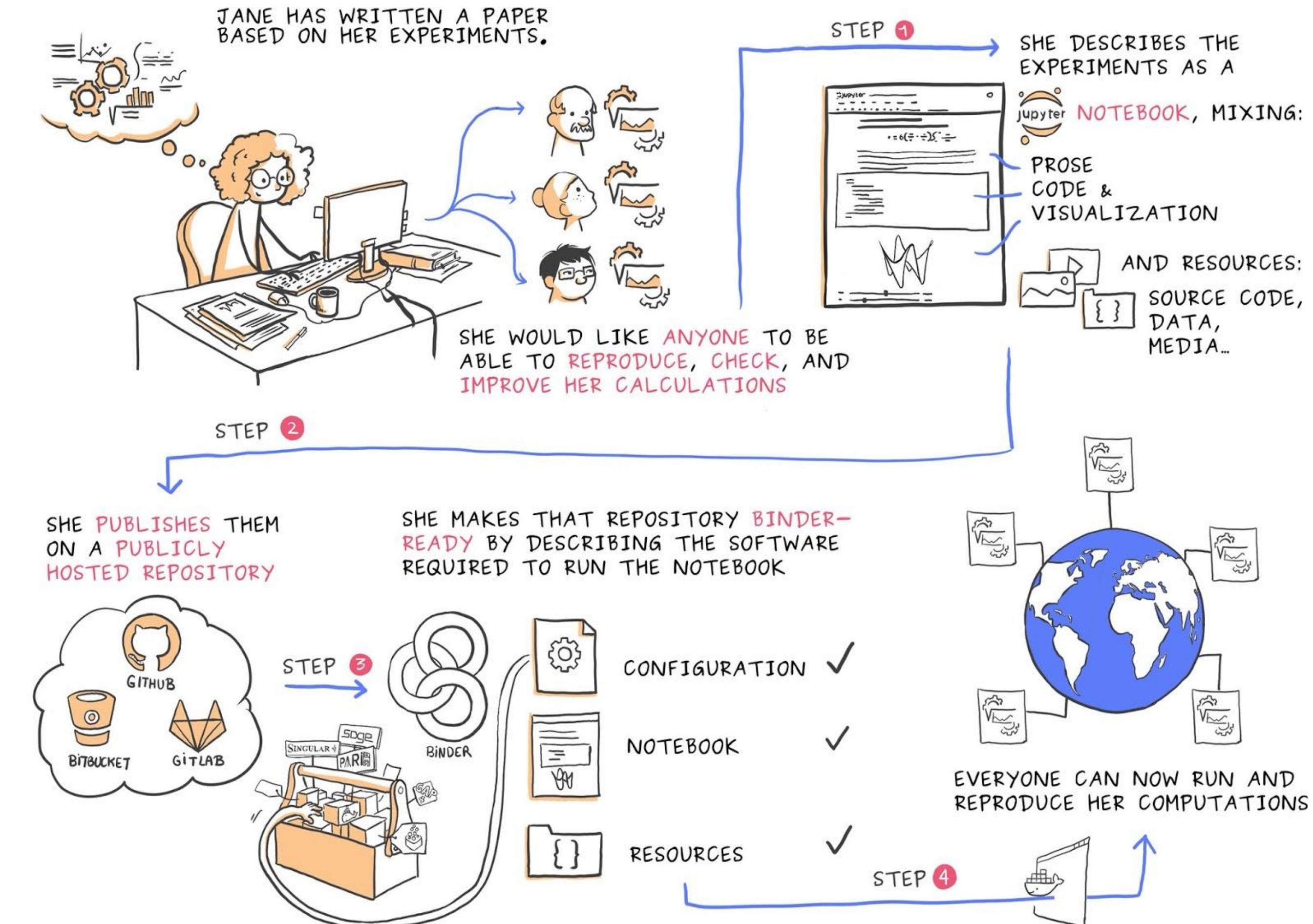
```
[2]: place_box = widgets.Text(description="Enter place:", value="Taipei")
display(place_box)
Enter place: Taipei
```

```
[3]: # The Nominatim module is used to retrieve the geographical coordinates of the selected place
geolocator = Nominatim(user_agent="any_agent")
location = geolocator.geocode(place_box.value)

print(location.address)
print((location.latitude, location.longitude))
臺北市, 臺灣
(25.0375198, 121.5636796)
```

```
[4]: # The folium package is used to plot our selected geolocation on a map
m = folium.Map(location=[location.latitude, location.longitude])
tooltip = location.latitude, location.longitude
folium.Marker([location.latitude, location.longitude], tooltip=tooltip).add_to(m)
display(m)
```

Reproducing Execution Environments



Juliette Taka, & Nicolas M. Thiéry. (2018). Publishing reproducible logbooks explainer comic strip. Zenodo. <https://doi.org/10.5281/zenodo.4421040>



Making repositories citable

The screenshot shows the Zenodo account settings page for GitHub integration. On the left, there's a sidebar with 'Settings' and 'GitHub' selected. The main area is titled 'GitHub Repositories' and contains two sections: '1 Flip the switch' (with a toggle switch set to 'ON') and '2 Create'. Below these are 'Enabled Repositories' listed with DOIs:

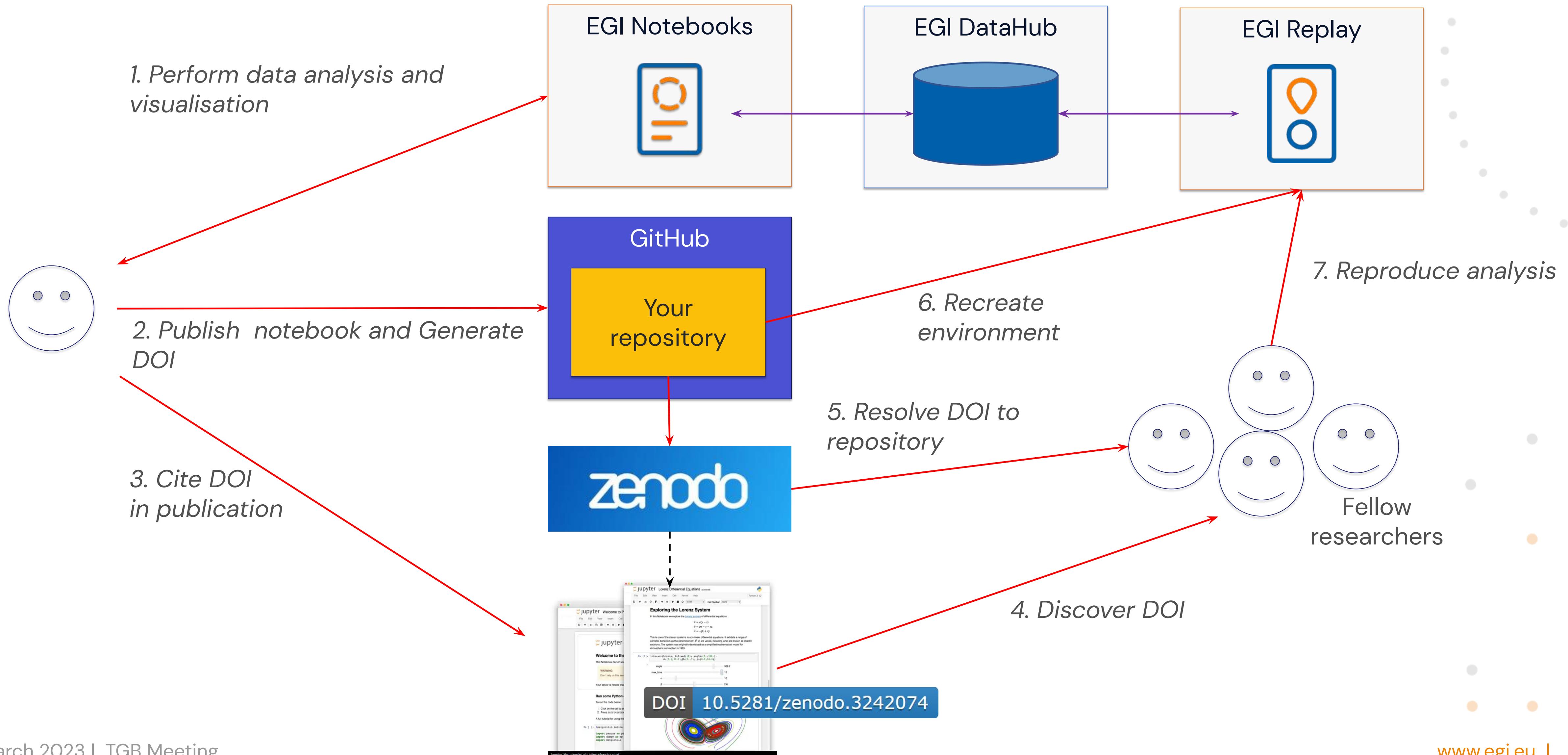
- EGI-Federation/data-transfer-pilot DOI 10.5281/zenodo.4218562
- EGI-Federation/nagios-plugins-srm DOI 10.5281/zenodo.3963998
- EGI-Federation/training-notebooks-climat DOI 10.5281/zenodo.3469564
- EGI-Federation/training-notebooks-seadat DOI 10.5281/zenodo.3561323
- c-scale-community/use-case-aquamonito DOI 10.5281/zenodo.7673807
- egi-qc/umd-verification DOI 10.5281/zenodo.3747669
- enolfernandez/seadatnet DOI 10.5281/zenodo.3561323

The screenshot shows the Zenodo record page for a training notebook. At the top, it says 'December 4, 2019' and has 'Software Open Access' buttons. The main content area features the ESI binder logo and instructions: 'Turn a Git repo into a collection of interactive notebooks' and 'Have a repository full of Jupyter notebooks? With Binder, open those notebooks in an executable environment, making your code immediately reproducible by anyone, anywhere.' It includes a 'New to Binder?' link and a 'Zero-to-Binder tutorial' section.

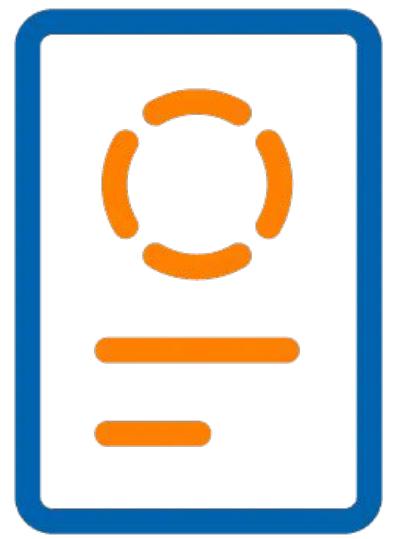
On the right, there's a detailed view of the record:

- s-seadatnet:** A software entry with 148 views and 11 downloads.
- Available in:** GitHub and OpenAIRE.
- Publication date:** December 4, 2019.
- DOI:** DOI 10.5281/zenodo.3561323
- Subject(s):** EOSC Jupyter Notebook
- Related identifiers:** Supplement to https://github.com/EGI-Foundation/training-notebooks-seadatnet/tree/0.4
- License (for files):** Other (Open)
- Versions:** Version 0.4 (Dec 4, 2019) and Version 0.3 (Oct 8, 2019).

EGI Replay and EGI Notebooks



Start using the services!



<https://notebooks.egi.eu/>



<https://replay.notebooks.egi.eu/>



Join us





Thank you

Marco Rorro

www.egi.eu



This work is partially funded by the EU research
and innovation programme