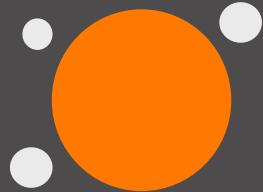


Updates from planet Jupyter



PyDataCluj
2020

Georgiana Dolocan
 jupyter

Hello! ❤

- Georgiana Dolocan
- ~ 2 years
- Outreachy intern
- JupyterHub team member
- Simula
- JupyterHub and Binder Contributor in Residence
- 2i2c The International Interactive Computing Collaboration



Thanks!


Jupyter team

	Afshin Taylor Darian afshin		Dave Willmer dwillmer		Kevin Bates kevin-bates		Peter Parente parente
	Anton Akhmerov akhmerov		Fernando Perez fperez		Logan Page lgpage		Michael Poplavski poplav
	Tim Head betatim		Ian Rose ian-rrose		Luciano Resende lresende		Kyle Kelley rgbkrk
	Steven Sylvester blink1073		Paul Ivanov ivanov		Corey Stubbs Lull3rSkat3r		Reese Netro rnetro
	Cameron Oelsen cameronoelsen		jakirkham		martinRenou martinRenou		Roshan Prabhakar roshancvp
	Safia Abdalla captainsafia		JamieW JamiesHQ		Min RK minrk		Sylvain Corlay SylvainCorlay
	Matthias Bussonnier Carreau		Jason Grout jasongrout		M Pacer mpacer		Thomas Kluyver takluyver
	Charnpreet Singh charnpreetsingh		Joseph Weston jbweston		Nitin Dahyabhai nitind		Carol Willing willingc
	Damian Avila damianavila		Jessica B. Hamrick jhamrick				
	David Brochart davidbrochart		Johan Mabille JohanMabille				
	Doug Blank dsblank		Justin Tyberg jtyberg				



Thanks! ❤

JupyterHub and
Binder team



	Tim Head betatim
	Michael Milligan mbmilligan
	Matthias Bussonnier Carreau
	Chris Holdgraf choldgraf
	Erik Sundell condeRatio
	Georgiana Elena GeorgianaElena
	Christopher Hench henchc
	Jim Crist-Harif jcrist
	Lindsey Heagy lheagy
	Simon Li manics

Thanks!

JupyterLab team ❤
(Lots of people) 

<https://github.com/orgs/jupyterlab/people>



Thanks! ❤

Executable Books team



AakashGC
AakashGfude



Chris Holdgraf
choldgraf



Chris Sewell
chrisjsewell



Taneli Hukkanen
hukkinj1



John Stachurski
jstac



mmcky



najuzilu



Jupyter Notebooks

The screenshot shows a Jupyter Notebook interface. At the top, there's a header bar with the Jupyter logo, the title "ProxyPerformance", the last checkpoint date "07/13/2020", and a "Not Trusted" status. Below the header is a toolbar with various icons for file operations like Open, Save, and Run, along with a Python 3 logo and a Logout button.

The main area contains two code cells:

In [7]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

In [8]:

```
toml_df_concurrent = pd.read_csv('/results/toml_methods_concurrent.csv')
etcd_df_concurrent = pd.read_csv('/results/etcd_methods_concurrent.csv')
chp_df_concurrent = pd.read_csv('/results/chp_methods_concurrent.csv')
consul_df_concurrent = pd.read_csv('/results/consul_methods_concurrent.csv')
patched_consul_df_concurrent = pd.read_csv('/results/patched_consul_methods_concurrent.csv')
patched_consul_df_concurrent = patched_consul_df_concurrent.replace('ConsulProxy', 'PatchedConsulProxy')

frames = [toml_df_concurrent, etcd_df_concurrent, consul_df_concurrent,
          patched_consul_df_concurrent, chp_df_concurrent]
proxies_df_concurrent = pd.concat(frames)
```

In [9]:

```
def plot_proxy_method_perf(
    plot_proxy_method_perf, df, proxy, method, run_type, plt_title=None, time_type=None, color=None
):
    q = f'proxy == "{proxy}" and method == "{method}"'

    if not time_type:
        proxy_method_df = df.query(q)[['route_idx', 'proxy', 'cpu_time', 'real_time']]
    else:
        proxy_method_df = df.query(q)[['route_idx', 'proxy', time_type]]

    grouped = proxy_method_df.groupby(['route_idx'])
    errors = grouped.std()
    means = grouped.mean()

    if method == "delete" and run_type == "sequential":
        means.index = np.flip(means.index)

    if plt_title is None:
        plt.title = proxy
    means.plot()
```

Lots of different kernels

jupyter / jupyter

Code Issues 173 Pull requests 4 Actions Projects Wiki Security Insights

Jupyter kernels

Mark Janssen edited this page 15 days ago · 129 revisions

Edit View Page

Jupyter kernels

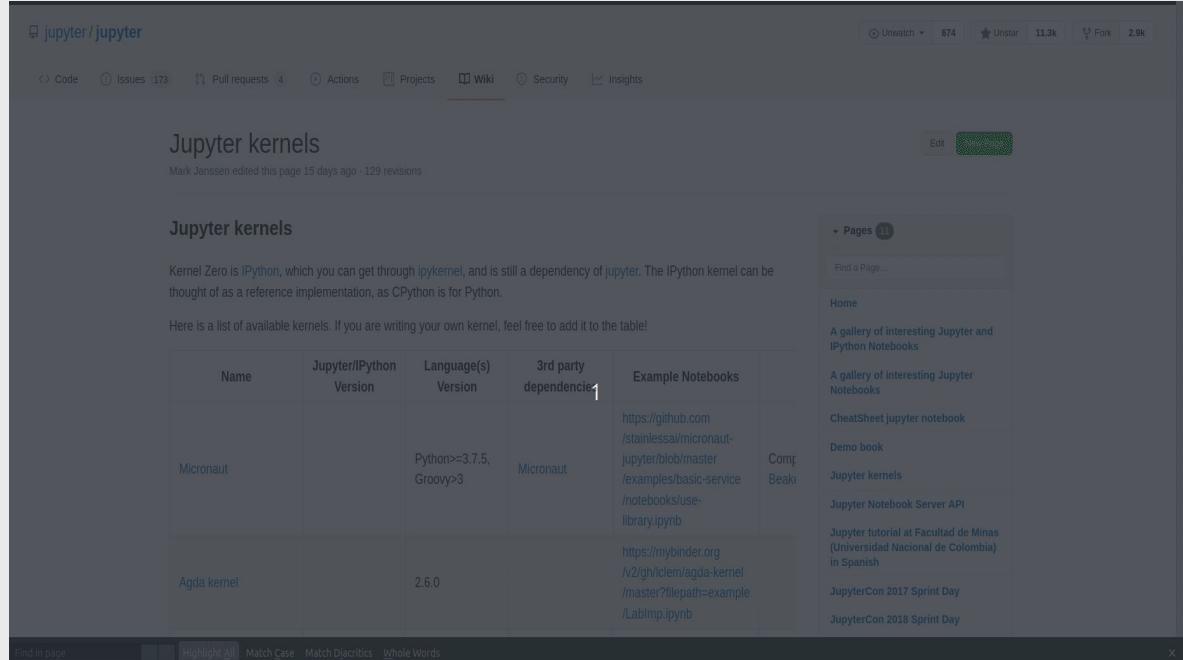
Kernel Zero is IPython, which you can get through [ipykernel](#), and is still a dependency of jupyter. The IPython kernel can be thought of as a reference implementation, as CPython is for Python.

Here is a list of available kernels. If you are writing your own kernel, feel free to add it to the table!

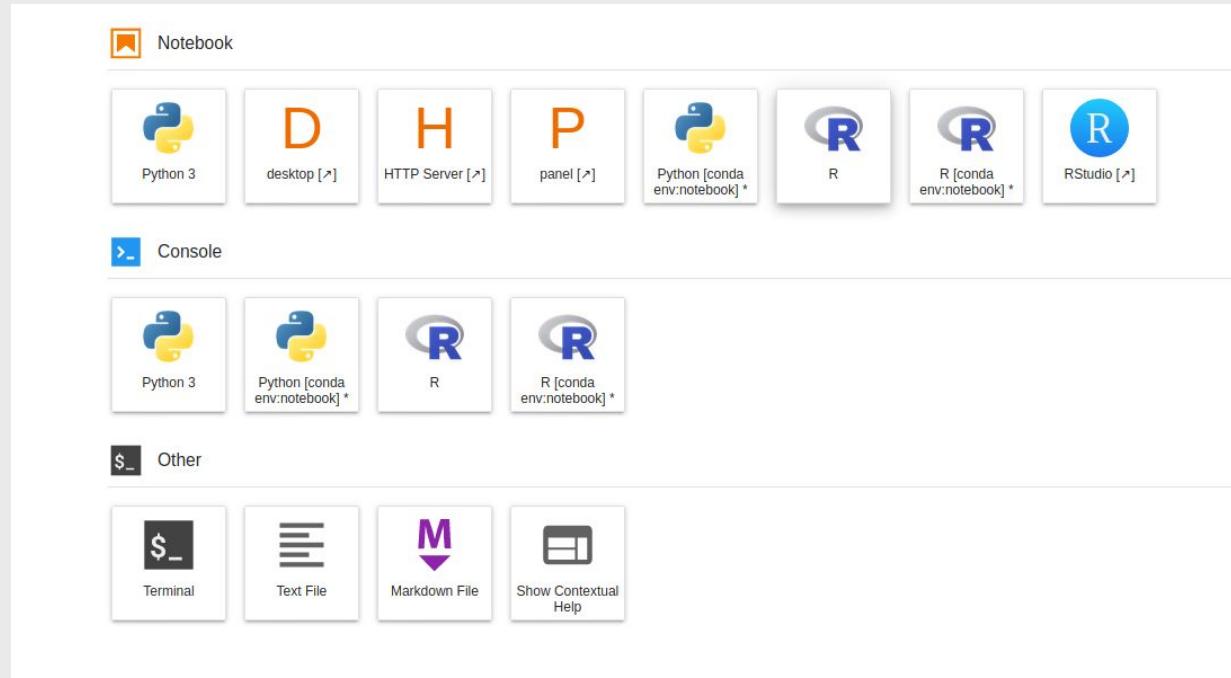
Name	Jupyter/IPython Version	Language(s) Version	3rd party dependencies	Example Notebooks	Beak
Micronaut		Python>=3.7.5, Groovy>3	Micronaut	https://github.com/stainlessai/micronaut-jupyter/blob/master/examples/basic-service/notebooks/use-library.ipynb	Comp Beak
Agda kernel		2.6.0		https://mybinder.org/v2/gh/clem/agda-kernel/master?filepath=example/LabImp.ipynb	

Pages 1 Find a Page... Home A gallery of interesting Jupyter and IPython Notebooks A gallery of interesting Jupyter Notebooks CheatSheet jupyter notebook Demo book Jupyter kernels Jupyter Notebook Server API Jupyter tutorial at Facultad de Minas (Universidad Nacional de Colombia) in Spanish JupyterCon 2017 Sprint Day JupyterCon 2018 Sprint Day

Find in page Highlight Match Case Match Diacritics Whole Words



Lots of different kernels



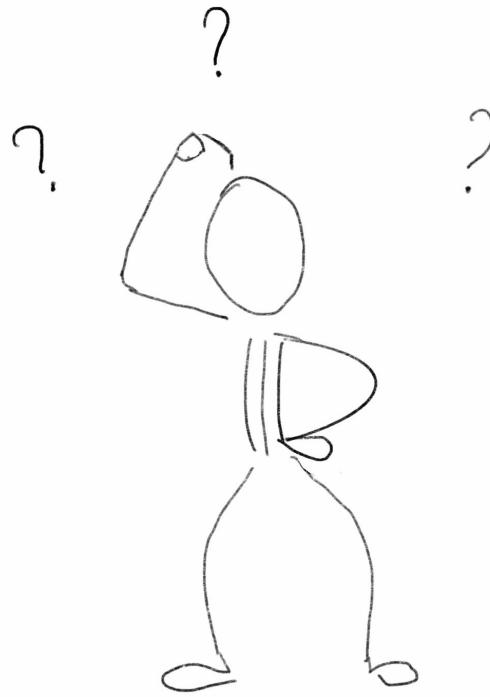
There's more

- Jupyter Notebook
- JupyterLab
- Binder - mybinder.org



- JupyterHub
- Zero to JupyterHub
- TLJH
- Jupyter Book

What are all
these?





Putting the **Lab** in Jupyter



Jupyter**Lab**

JupyterLab

the next-generation user
interface for Project Jupyter

<https://github.com/jupyterlab/jupyterlab>

Debugger

JupyterLab debugger

- UI extension
- *Xeus-python*
- Xeus library: c++ implementation of the Jupyter kernel protocol
- *xeus-cling*

JupyterLab debugger

-demo from the docs-

Debugger

<https://raw.githubusercontent.com/jupyterlab/debugger/master/screencast.gif>

¡Bienvenido!
Bienvenue!
Welcome!
Bun venit!

- [Discourse forum thread](#)
- [Crowdin Platform](#)
- JupyterLab 3.x

JupyterLab

Home Activity Discussions

Translations:



Catalan

16% • 0%



Chinese Simplified

100% • 0%



Chinese Traditional

2% • 0%



Czech

3% • 0%



Danish

16% • 0%



Dutch

2% • 0%



Estonian

16% • 0%



Finnish

14% • 0%



French

38% • 0%



German

29% • 0%



Greek

1% • 0%



Hungarian

0% • 0%



Italian

4% • 0%



Japanese

1% • 0%



Korean

1% • 0%



Norwegian

4% • 0%



Polish

72% • 0%



Portuguese, Brazilian

61% • 23%



Romanian

25% • 0%



Russian

78% • 0%



Spanish

49% • 0%



Turkish

100% • 0%



Ukrainian

100% • 0%



Vietnamese

4% • 0%



jupyter

Description

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Currently ready for users.

JupyterLab is the next-generation user interface for Project Jupyter offering all the familiar building blocks of the classic Jupyter Notebook (notebook, terminal, text editor, file browser, rich outputs, etc.) in a flexible and powerful user interface. JupyterLab will eventually replace the classic Jupyter Notebook.

JupyterLab can be extended using npm packages

Show More

Share Project



Details

Sign in

Warning: JupyterHub seems to be served over an unsecured HTTP connection. We strongly recommend enabling HTTPS for JupyterHub.

Username:

Password:

Sign In

JupyterHub login page

Putting the **Hub** in Jupyter



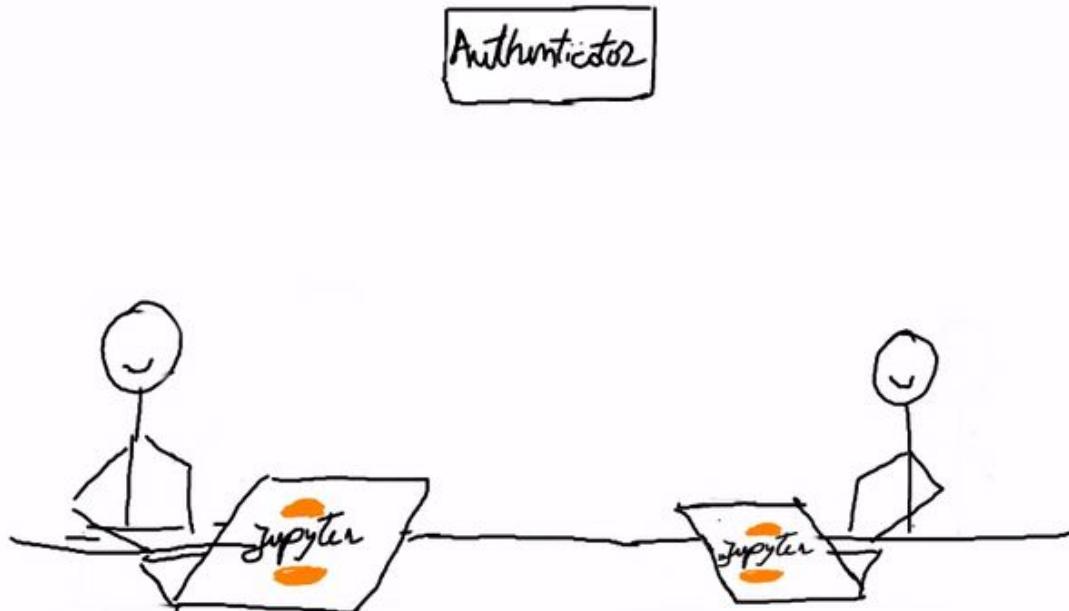
Jupyter**Hub**

Jupyter vs. JupyterHub



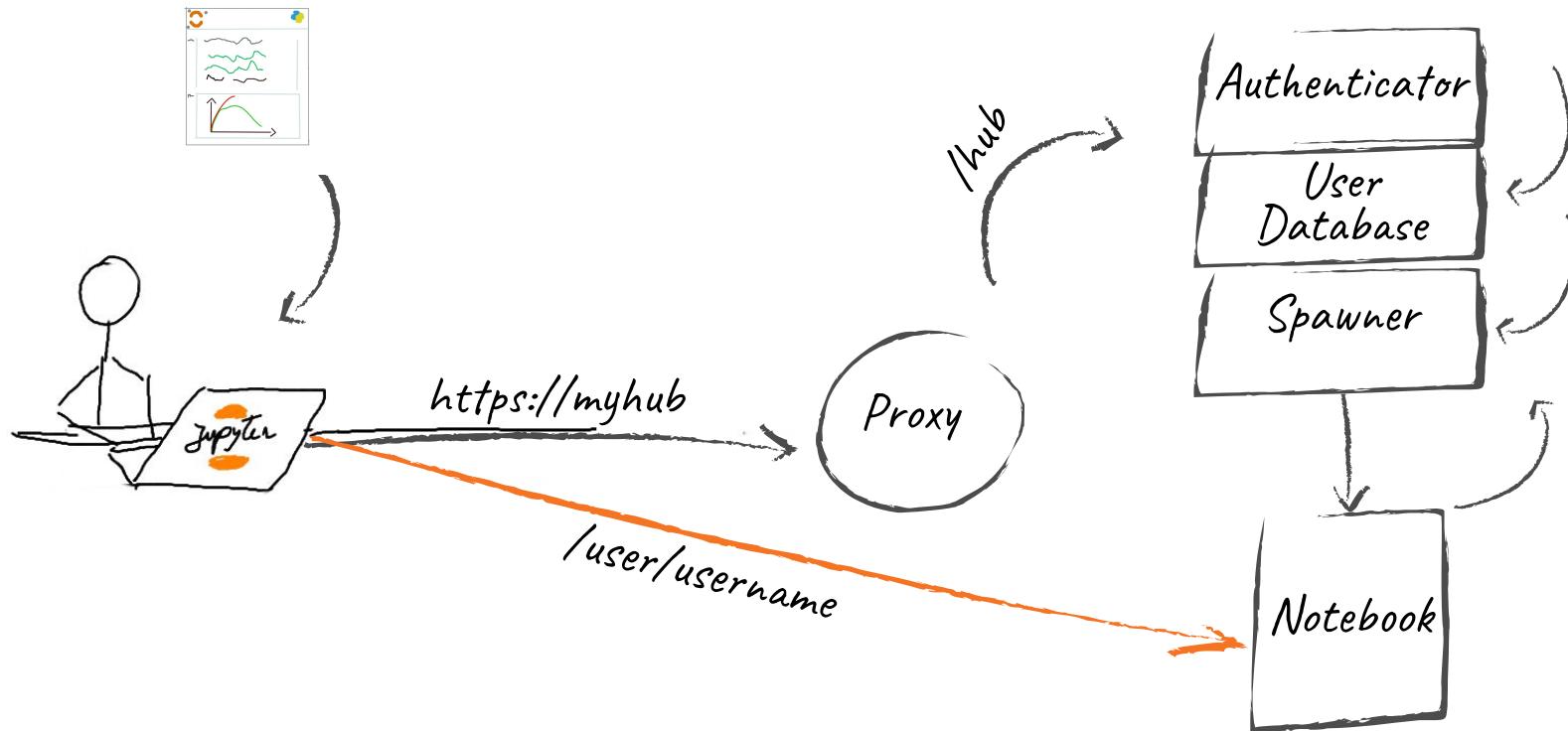
JupyterHub

Multiple users can have access to a Jupyter Notebook with a click of their favorite browser.



JupyterHub

High-level workflow



Things you can change to your hub and still be a JupyterHub



Impressive modularity

You can change pretty much everything...

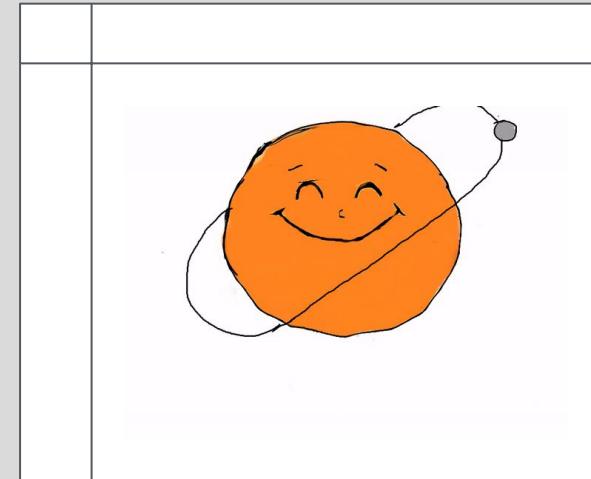
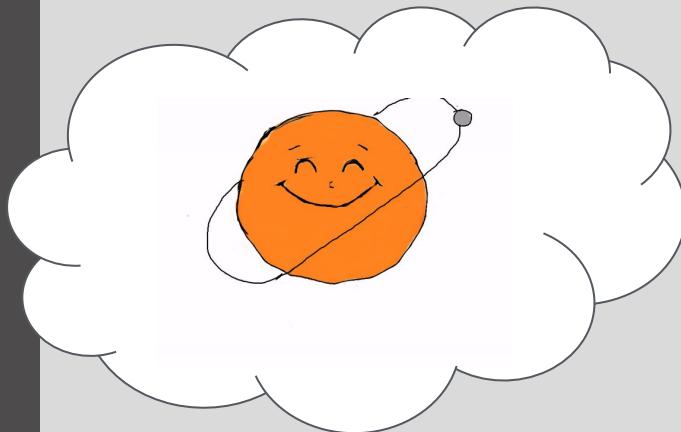
- Extensible
- Serves a variety of use-cases
- Create your own component
- Community-developed
- jupyterhub_config.py



Deploying JupyterHub - Distributions



JupyterHub the hard way



The Littlest JupyterHub

TLJH

- **single**-server
- **multi**-user
- small number of users
- small costs
- reduced complexity
- shared conda environment
- easy-deployable
- no sysadmin skills
- base use-case, no ssh

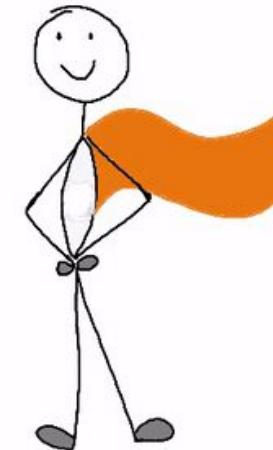


Zero to JupyterHub with Kubernetes

Z2JH

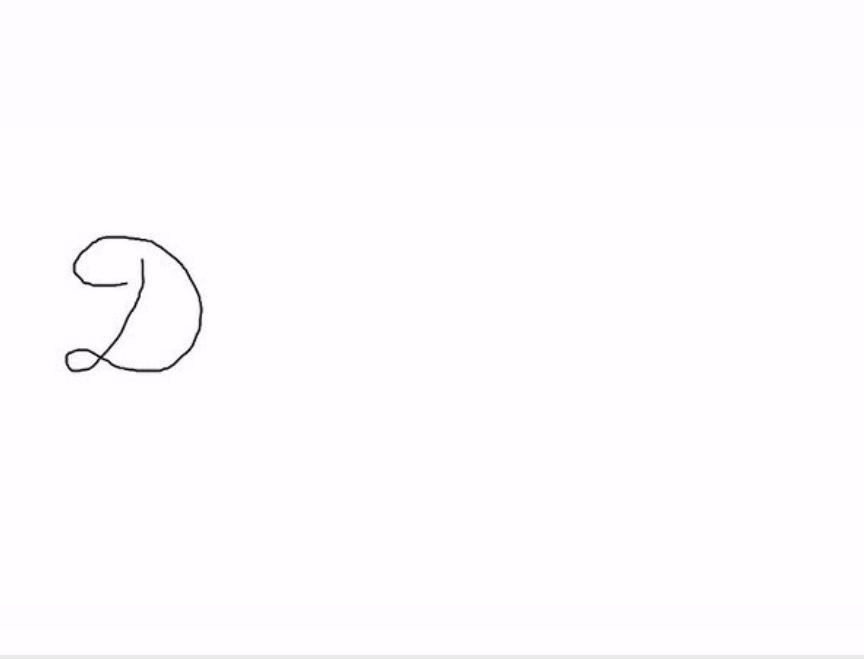
- **multi**-server
- **multi**-user
- lots of users
- cost depends on size
- complex
- Kubernetes “laws”
 - scalable
- admin k8s skills required

Z2JH





First day of class



2

- <https://tjh.jupyter.org>
- <https://cloud.digitalocean.com/projects>



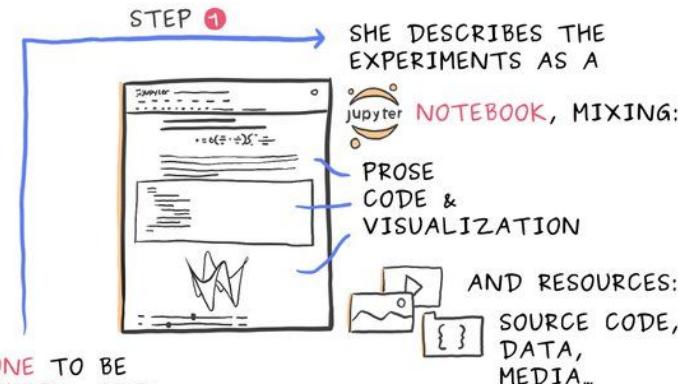
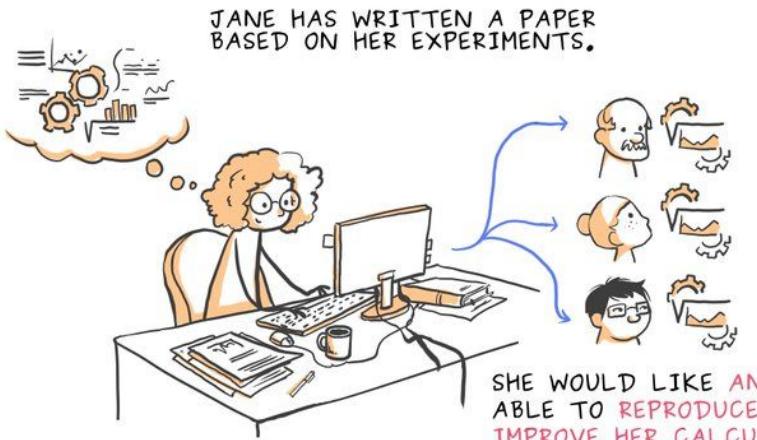
Putting the **reproducibility** in
Jupyter



Binder

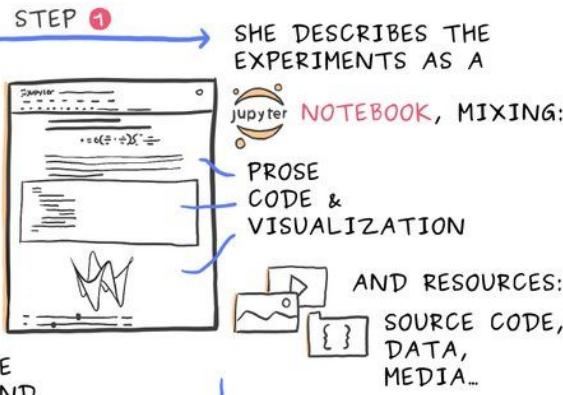
What do you need to do open-source science?

- The computational **tools** to solve a problem
- An **interface** to facilitate coding / creating
- A way to **communicate** your work
- A way to **share** your work
- A way to **pack** it all for replication
- A way to do all of this relatively **easily and accessibly**





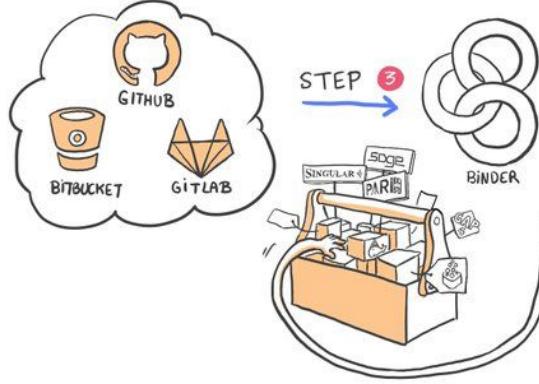
SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS



STEP 2

SHE PUBLISHES THEM ON A PUBLICLY HOSTED REPOSITORY

SHE MAKES THAT REPOSITORY BINDER-READY BY DESCRIBING THE SOFTWARE REQUIRED TO RUN THE NOTEBOOK



STEP 3



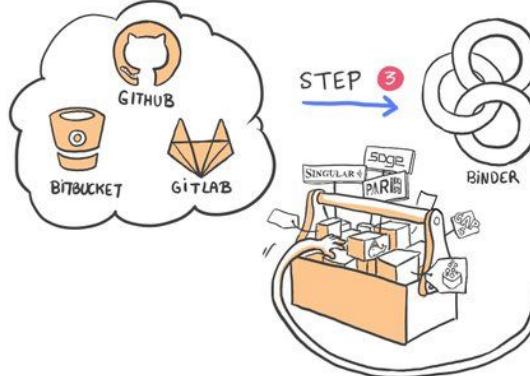
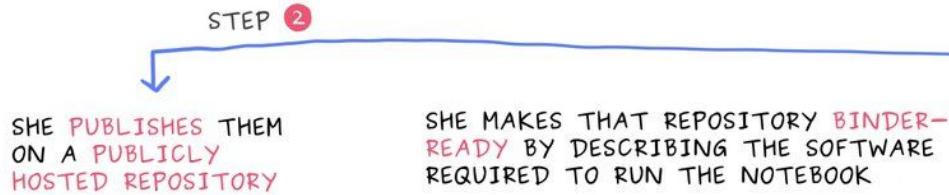
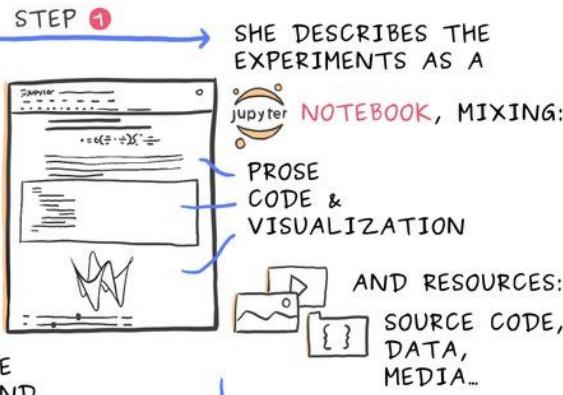
CONFIGURATION ✓

NOTEBOOK ✓

RESOURCES ✓



SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS



CONFIGURATION ✓

NOTEBOOK ✓

RESOURCES ✓



EVERYONE CAN NOW RUN AND REPRODUCE HER COMPUTATIONS



Building a Dockerfile might a barrier to entry

- JupyterHub needs a **pre-built** Docker image.
- Creating the image can be hard to learn, debug, etc
- Most languages already have a way to specify dependencies
- Could we just use those files to generate a **Dockerfile**?

repo2docker

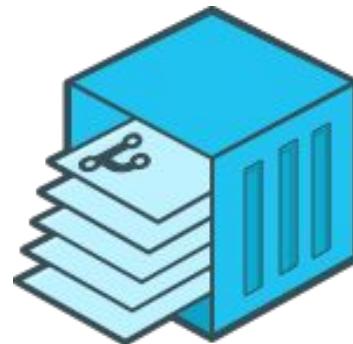
+

binder



repo2docker

Convert a repository into a reproducible Docker image



repo2docker.readthedocs.io

repo2docker

*deterministically build a
docker image
from a repository.*

- Parse a repo for “configuration files”
- Use them to build the environment needed to run the code
- Many kinds of workflows:
 - requirements.txt
 - environment.yml
 - apt.txt
 - postBuild
 - REQUIRE
 - install.R
 - runtime.txt
 - Dockerfile

if you already **specify your dependencies**
your repository should just work with
Binder.



Project Binder

*sharable computational
environments*

- JupyterHub + repo2docker + some glue
- Quickly generate **interactive, reproducible, sharable computational environments**

BinderHub

*One-click sharable,
interactive, reproducible
environments from your
public git repository*

- mybinder.org
- binderhub.readthedocs.io

Thanks to Google Cloud, ONR, GESIS Notebooks and the Turing Institute for supporting us! 



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.

New to Binder? Get started with a [Zero-to-Binder tutorial](#) in Julia, Python or R.

Build and launch a repository

GitHub repository name or URL
 GitHub GitHub repository name or URL

Git ref (branch, tag, or commit)
 HEAD Path to a notebook file (optional)
 Path to a notebook file (optional)

Copy the URL below and share your Binder with others:
 Fill in the fields to see a URL for sharing your Binder.

Copy the text below, then paste 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.

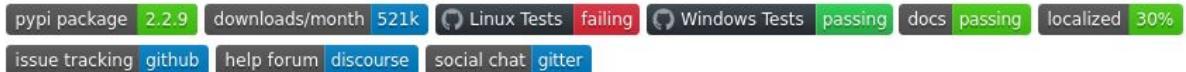
questions?

join the discussion, read the docs, see the code

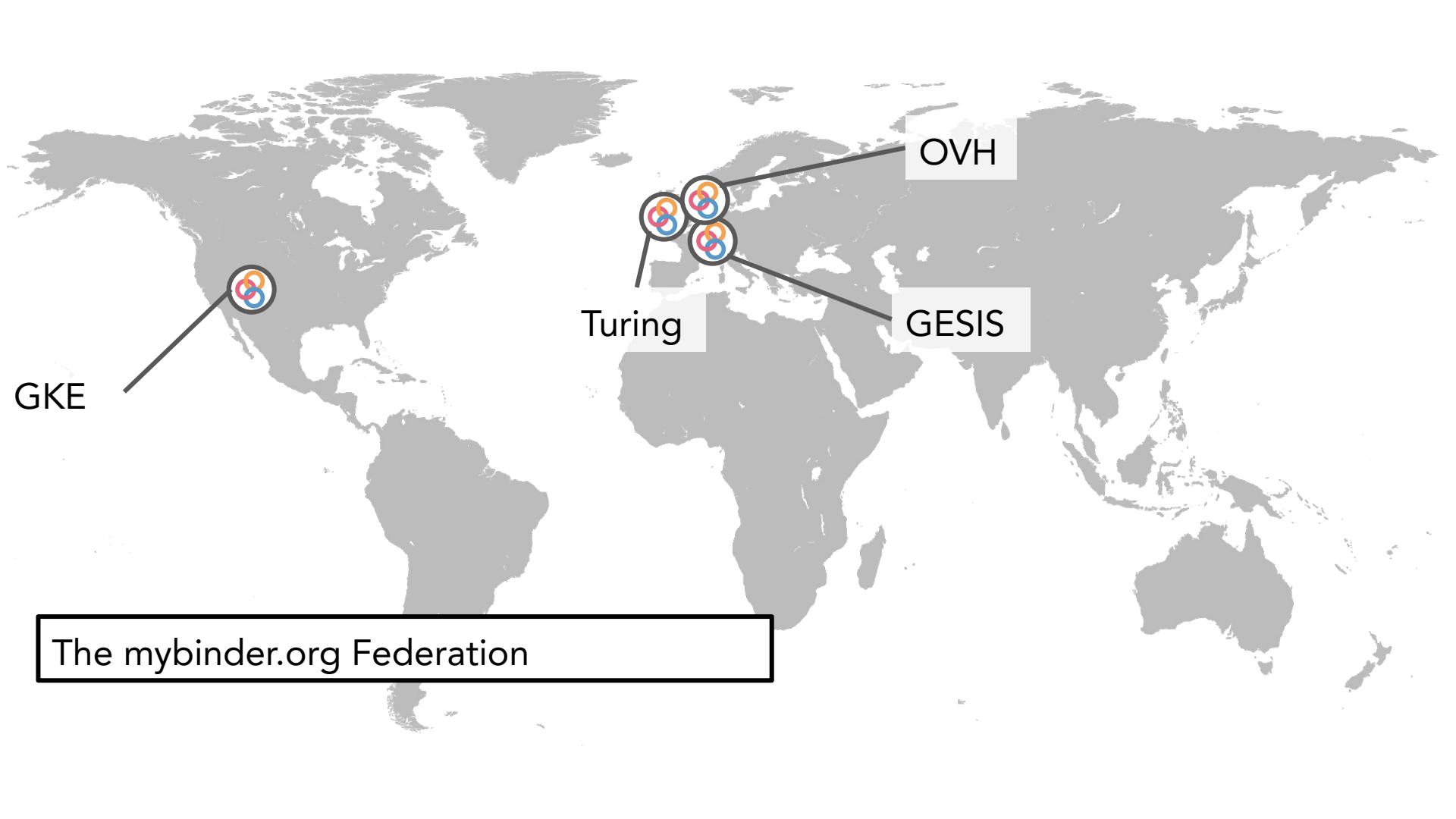
mybinder.org

- Is a BinderHub hosted for public use
- Suitable for workshops, research reproducibility and more
- Relies on a JupyterHub on Kubernetes but adds custom dependency installation!

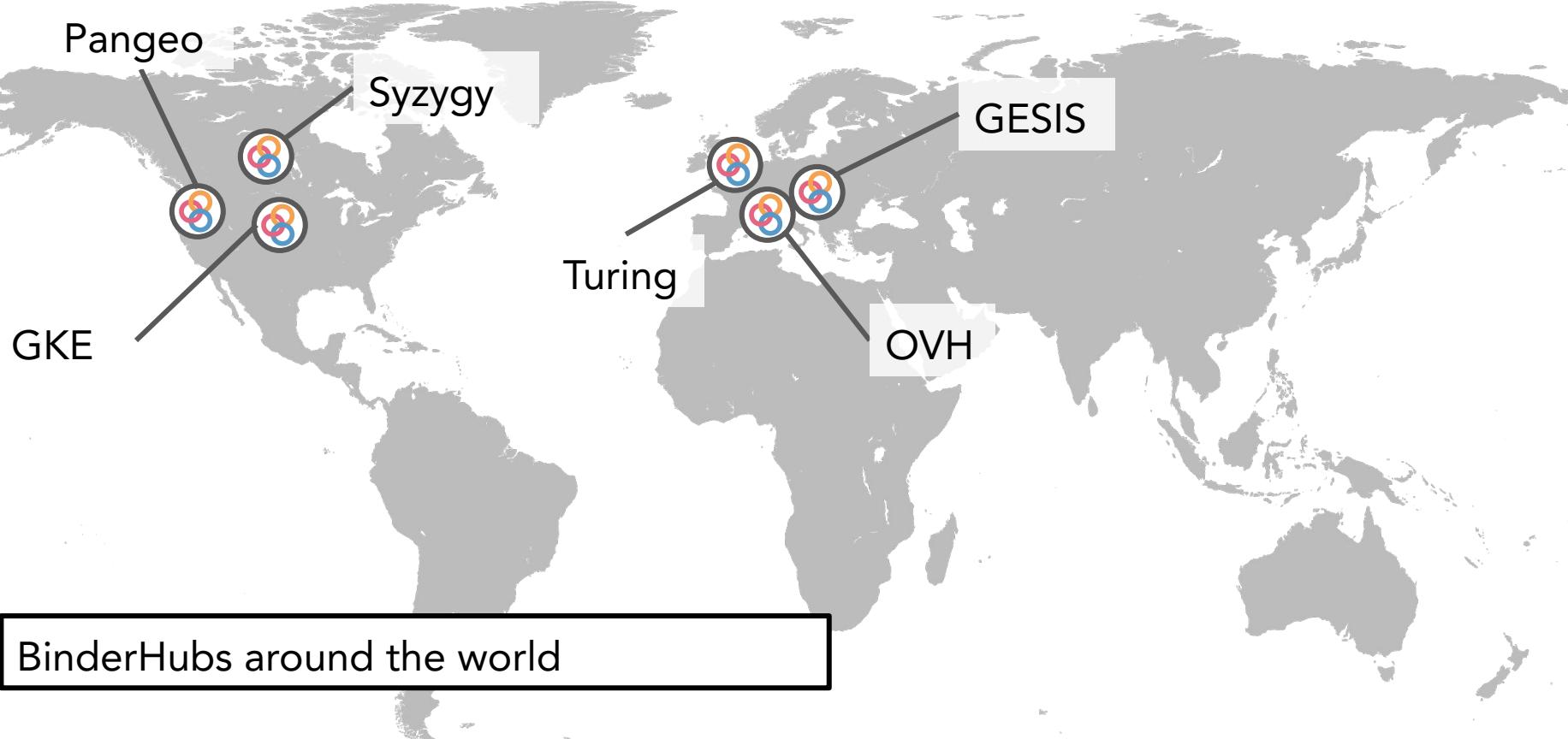
↗ JupyterLab



 launch binder



The mybinder.org Federation





Putting the **book** in Jupyter



Jupyter **Book**

What is Jupyter Book?

Build an online book with
Jupyter Notebooks and Markdown

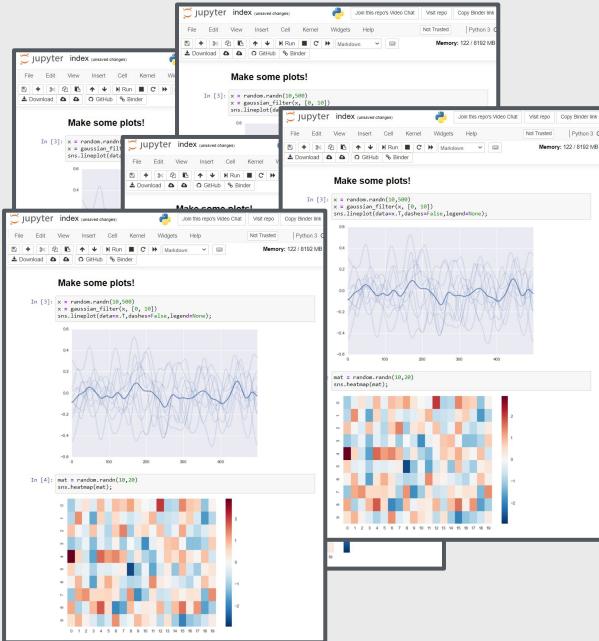


jupyter.org/jupyter-book



Content

Notebooks / Markdown files



Book

HTML / PDF / and more

The figure shows a screenshot of a Jupyter book interface. On the left, there is a sidebar with navigation links such as "Get started", "Write book content", and "Make your book interactive". The main content area features a heading "Plotly" and a sub-section "Plotly". It includes a search bar, a "Search this book..." input field, and a "Plotly" logo. Below the heading, there is a text block about Plotly and a code block demonstrating its use with the Iris dataset. To the right of the text is a scatter plot of Sepal Length vs Sepal Width, colored by species (setosa, versicolor, virginica).

Plotly is another interactive plotting library that provides a high-level API for visualization. See the [Plotly JupyterLab documentation](#) to get started with Plotly in the notebook.

Below is some example output.

```
import plotly.io as pio
import plotly.express as px
import plotly.offline as py

df = px.data.iris()
fig = px.scatter(df, x="sepal_width", y="sepal_length", color="species", size="sepal_length")
fig
```

species

- setosa
- versicolor
- virginica

sepal-length

sepal_width



Search this book...

GET STARTED

- Overview
- Build your book
- Publish your book online
- Configure book settings
- Table of contents structure
- Types of content source files

WRITE BOOK CONTENT

- MyST Markdown overview
- Special content blocks
- References and citations
- Math and equations
- Images and figures
- Control the page layout
- Execute and cache your pages
- Formatting code outputs

MAKE YOUR BOOK INTERACTIVE

- Launch buttons for interactivity
- Hide or remove content
- Interactive data visualizations
- Commenting and annotating

ADVANCED AND MISCELLANEOUS

- PDFs for your book
- Custom Sphinx configuration

Books with Jupyter

Jupyter Book is an open source project for building beautiful, publication-quality books and documents from computational material.

Here are some of the features of Jupyter Book:

- ✓ **Write publication-quality content in Markdown**
You can write in either Jupyter Markdown, or an extended flavor of Markdown with [publishing features](#). This includes support for rich syntax such as [citations](#) and [cross-references](#), [math and equations](#), and [figures](#).
- ✓ **Write content in Jupyter Notebook**
This allows you to include your code and outputs in your book. You can also write notebooks [entirely in Markdown](#) that get executed when you build your book.
- ✓ **Execute and cache your book's content**
For [.ipynb](#) and Markdown notebooks, execute code and insert the latest outputs into your book. In addition, [cache](#) and [re-use](#) outputs to be used later.
- ✓ **Insert notebook outputs into your content**
Generate outputs as you build your documentation, and insert them in-line with your content across pages.
- ✓ **Add interactivity to your book**
You can [toggle cell visibility](#), [include interactive outputs](#) from Jupyter, and [connect with online services](#) like Binder.
- ✓ **Generate a variety of outputs**
This includes single- and multi-page websites, as well as [PDF outputs](#).
- ✓ **Build books with a simple command-line interface**
You can quickly generate your books with one command, like so: `jupyter-book build mybook/`

This website is built with Jupyter Book! You can browse its contents to the left to see what is possible.

💡 **Get involved with Jupyter Book!**

Jupyter Book is an open community that welcomes your feedback, input, and contributions!

Open an issue
to provide feedback and new ideas, and to help others.

Vote for new features
by adding a  to issues you'd like to see completed.

Contribute to Jupyter Book

Connect with cloud infrastructure

The screenshot shows the Jupyter Book website interface. On the left is a sidebar with a dark background containing navigation links for various sections like 'GET STARTED', 'WRITE BOOK CONTENT', 'MAKE YOUR BOOK INTERACTIVE', and 'ADVANCED AND MISCELLANEOUS'. The main content area has a light background and displays the title 'Books with Jupyter' with a back arrow icon. Below the title is a brief introduction: 'Jupyter Book is an open source project for building beautiful, publication-quality books and documents from computational material.' A section titled 'Here are some of the features of Jupyter Book:' lists several items with checkmarks: 'Write publication-quality content in Markdown', 'Write content in Jupyter Notebook', 'Execute and cache your book's content', 'Insert notebook outputs into your content', 'Add interactivity to your book', 'Generate a variety of outputs', and 'Build books with a simple command-line interface'. At the bottom of the main content area, a note says 'This website is built with Jupyter Book! You can browse its contents to the left to see what is possible.' At the very bottom, there is a call-to-action button labeled 'Get involved with Jupyter Book'.

jupyter {book}

Search this book...

GET STARTED

- Overview
- Build your book
- Publish your book online
- Configure book settings
- Table of contents structure
- Types of content source files

WRITE BOOK CONTENT

- MyST Markdown overview
- Special content blocks
- References and citations
- Math and equations
- Images and figures
- Control the page layout
- Execute and cache your pages
- Formatting code outputs

MAKE YOUR BOOK INTERACTIVE

- Launch buttons for interactivity
- Hide or remove content
- Interactive data visualizations
- Commenting and annotating

ADVANCED AND MISCELLANEOUS

- PDFs for your book
- Custom Sphinx configuration

Books with Jupyter

Jupyter Book is an open source project for building beautiful, publication-quality books and documents from computational material.

Here are some of the features of Jupyter Book:

- ✓ Write publication-quality content in Markdown
- ✓ Write content in Jupyter Notebook
- ✓ Execute and cache your book's content
- ✓ Insert notebook outputs into your content
- ✓ Add interactivity to your book
- ✓ Generate a variety of outputs
- ✓ Build books with a simple command-line interface

This website is built with Jupyter Book! You can browse its contents to the left to see what is possible.

Get involved with Jupyter Book

Jupyter Book is an open community that welcomes your feedback, input, and contributions!

Open an issue

to provide feedback and new ideas, and to help others.

Vote for new features

by adding a ⚡ to issues you'd like to see completed.

Contribute to Jupyter Book

Contents

- Install Jupyter Book
- Get started
- A small example project
- Under the hood - the components of Jupyter Book
- Contribute to Jupyter Book
- Acknowledgements



GitHub Pages: free website hosting

GitHub Pages typical web-address



GitHub Actions: automate build / publish

GitHub Actions

The screenshot shows the GitHub Actions interface for the repository `neurohackademy/nh2020-jupyterhub`. The `Actions` tab is selected. A search bar at the top right contains the query `workflow:deploy-book`. Below the search bar, there are filters for Event, Status, Branch, and Actor. Two workflow runs are listed:

- docs: make note about the symbolic link ~/data -...**
deploy-book #41: Commit 2443b3d pushed by considerRatio
master 8 days ago 1m 14s
- docs: add notes about curriculum repo sync**
deploy-book #40: Commit c6fa35a pushed by considerRatio
master 8 days ago 57s

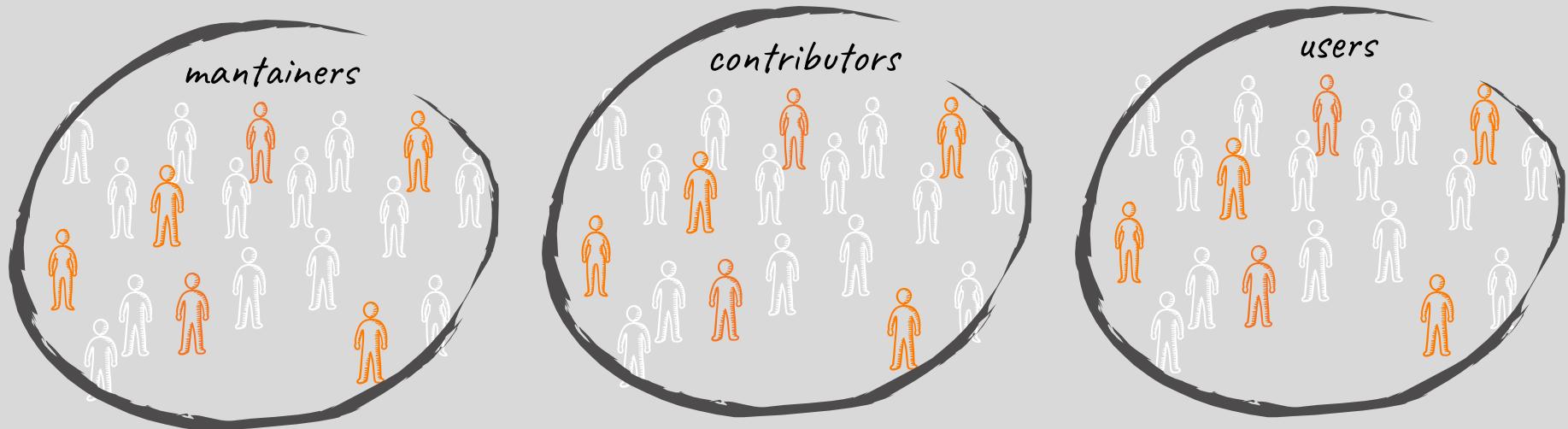


Check on TLJH!

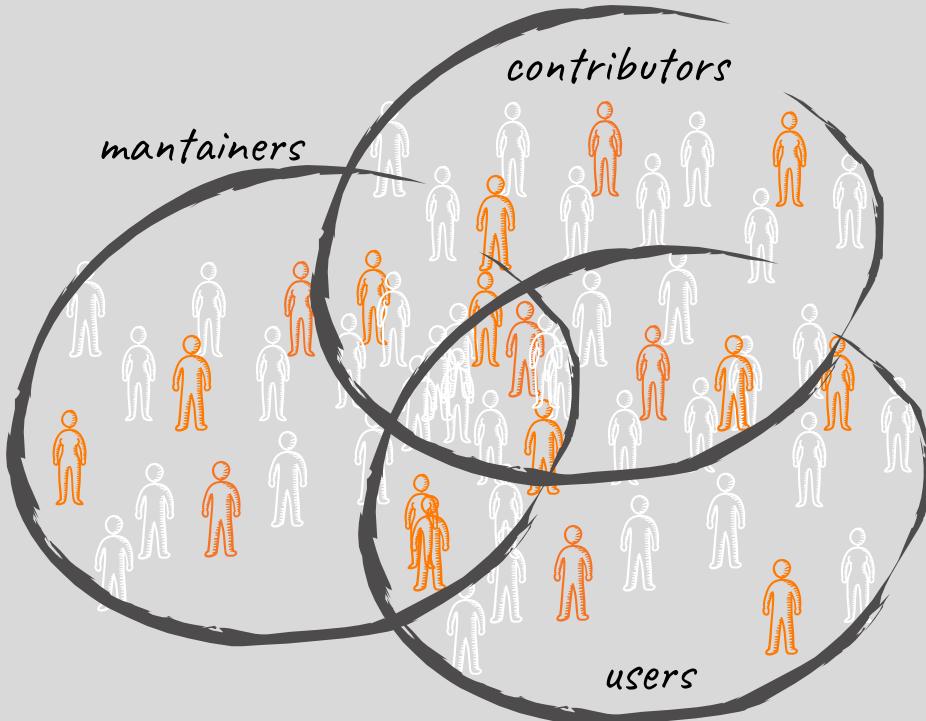
<may the demo gods be kind with georgiana>



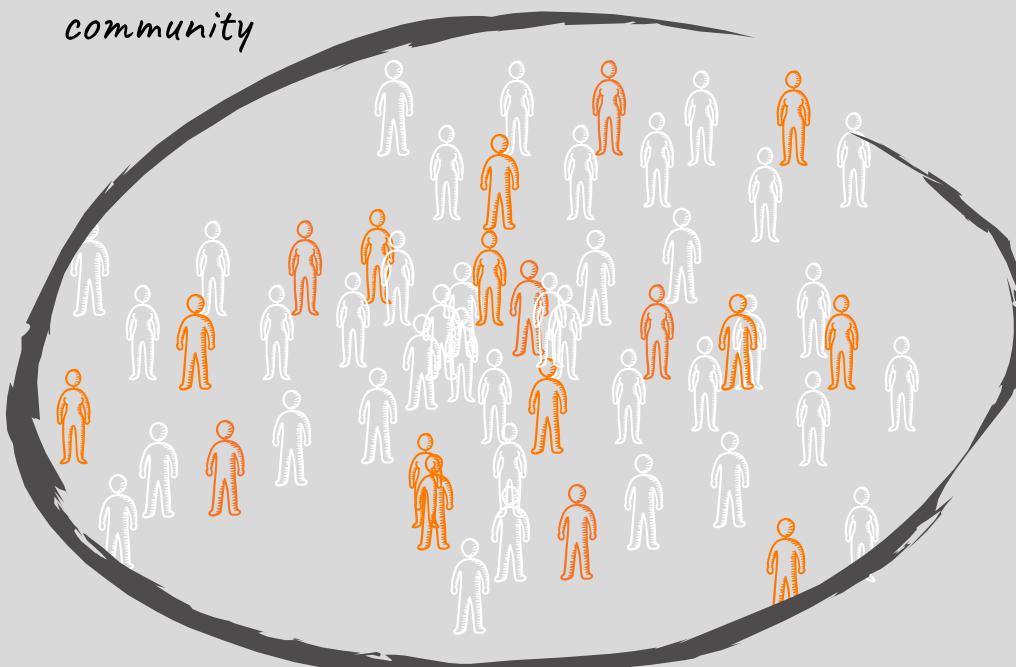
The Community



The Community



The Community





Takeaways

- Jupyter ecosystem is powered by the community
 - Something is always going on
 - same goal
 - Contribute, become a Jovyan



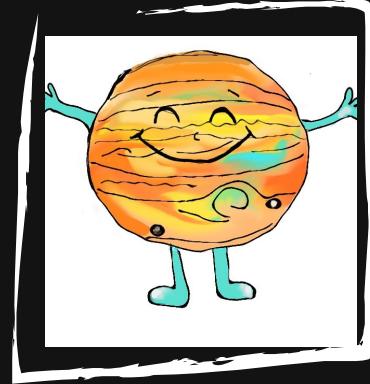
Try Jupyter
jupyter.org

Connect!
discourse.jupyter.org

JupyterCon
youtube.com/JupyterCon

Blog
blog.jupyter.org/





Thanks!

You can find me at

 @GeorgianaElena

 georgiana.dolocan@gmail.com

 @GeorgianaElenaD