

*Why you need a **reproducible**
computational environment
and how **Binder** can help*

Martina G. Vilas (she/her)

@martinagvilas



important!

- questions?
 - go to sli.do and enter code `#RSCBinder` to ask them!
- notes/resources/links?
 - <https://hackmd.io/@sgibson91/RSCBinder>

who am I?

- PhD student in Neuroscience at Max-Planck-Institute AE
- Core contributor / Maintainer of *The Turing Way*



<https://martinagvilas.github.io/>
@martinagvilas, @turingway #TuringWay

what is a *reproducible computational environment*?

reproducible research

same analysis steps
on the same
dataset produces
same answer

		Data	
		Same	Different
Analysis	Same	(Reproducible)	Replicable
	Different	Robust	Generalisable

*“An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the **complete software development environment** and the complete set of instructions which generated the figures.”*

— Buckheit and Donoho (paraphrasing John Claerbout)

WaveLab and Reproducible Research, 1995

(slide courtesy of Chris Holdgraf and the Jupyter Team)

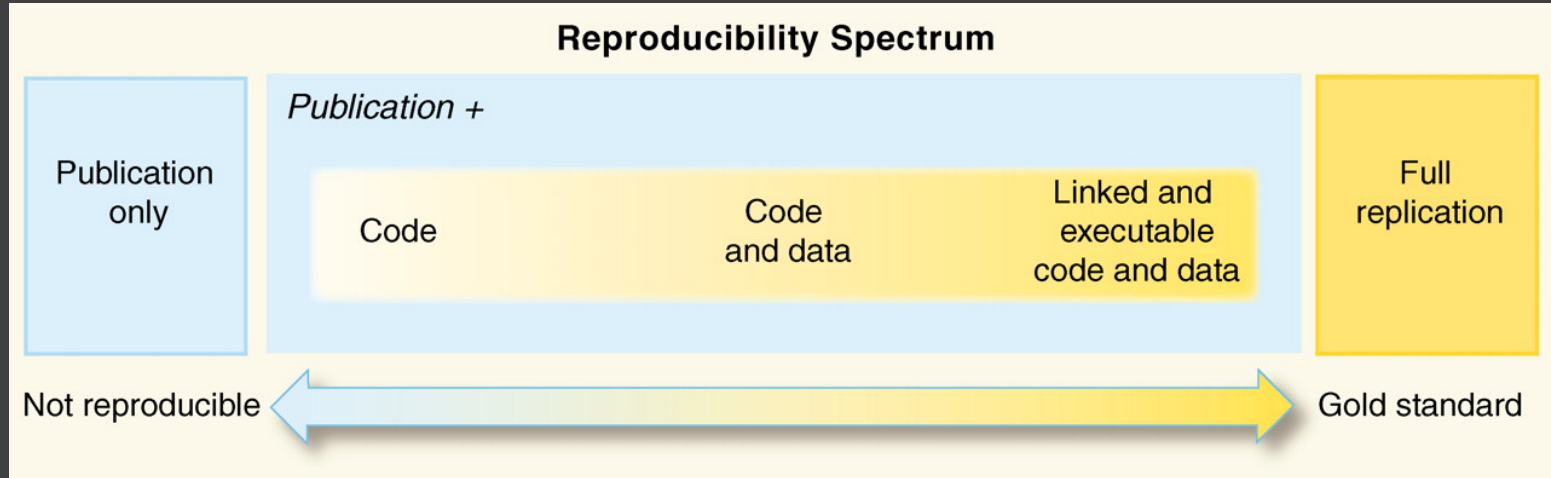
@martinagvilas, @turingway #TuringWay

take home message

sharing your code and
data isn't enough



you need the computational environment too



you need the computational environment too

Publication
only

Not reproducible



Full
replication

Gold standard

Peng, 2011, doi: [10.1126/science.1213847](https://doi.org/10.1126/science.1213847)

@martinagvilas, @turingway #TuringWay

what is a computational environment?

- hardware (e.g. CPU)
- software
 - operating system
 - programming languages
 - packages

what is a computational environment?

- hardware (e.g. CPU)
 - software
 - operating system
 - programming languages
 - packages
- their versions
and their
configuration

what is a computational environment?

- hardware (e.g. CPU)

- software

- operating system

- programming languages

- packages

their versions
and their
configuration

and their
interaction

what is *Binder*?

what is Binder?



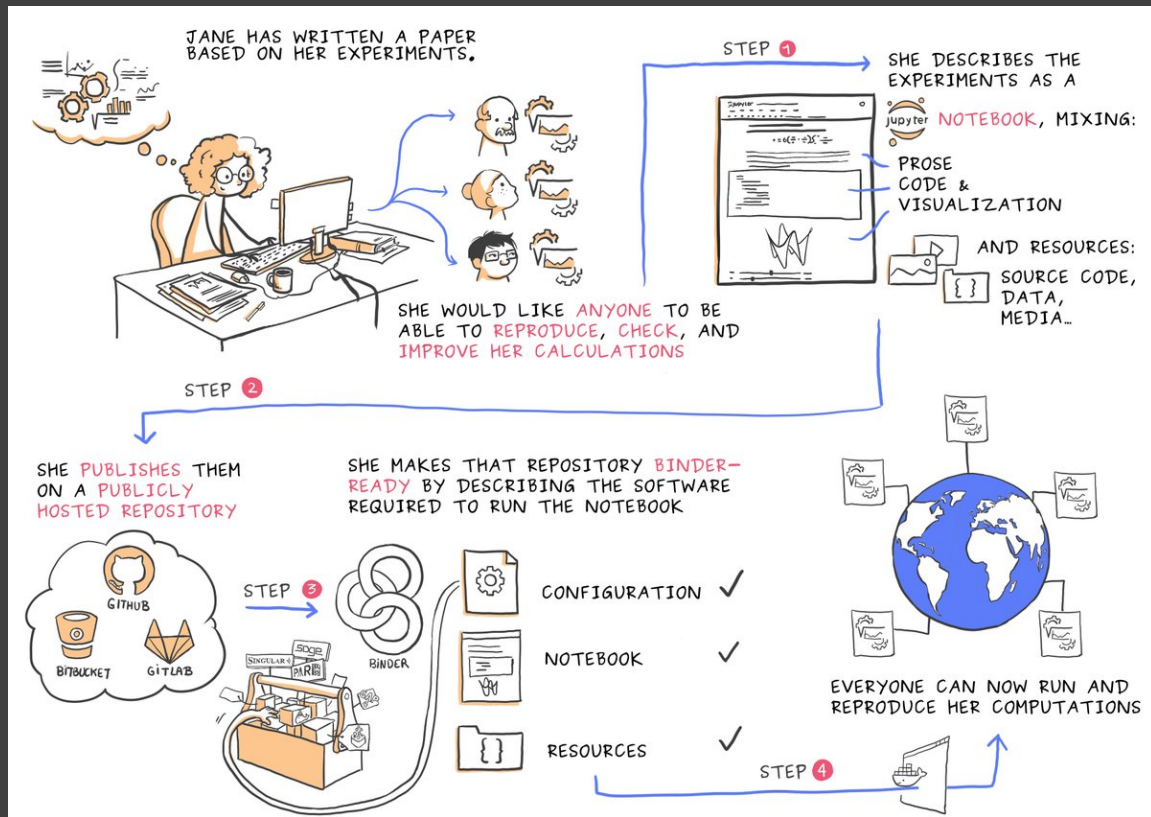
choldgraf Distinguished Contributor

3  Nov '18

The Binder Project helps you create one-click, sharable, live code environments from public code repositories that runs entirely in the cloud.

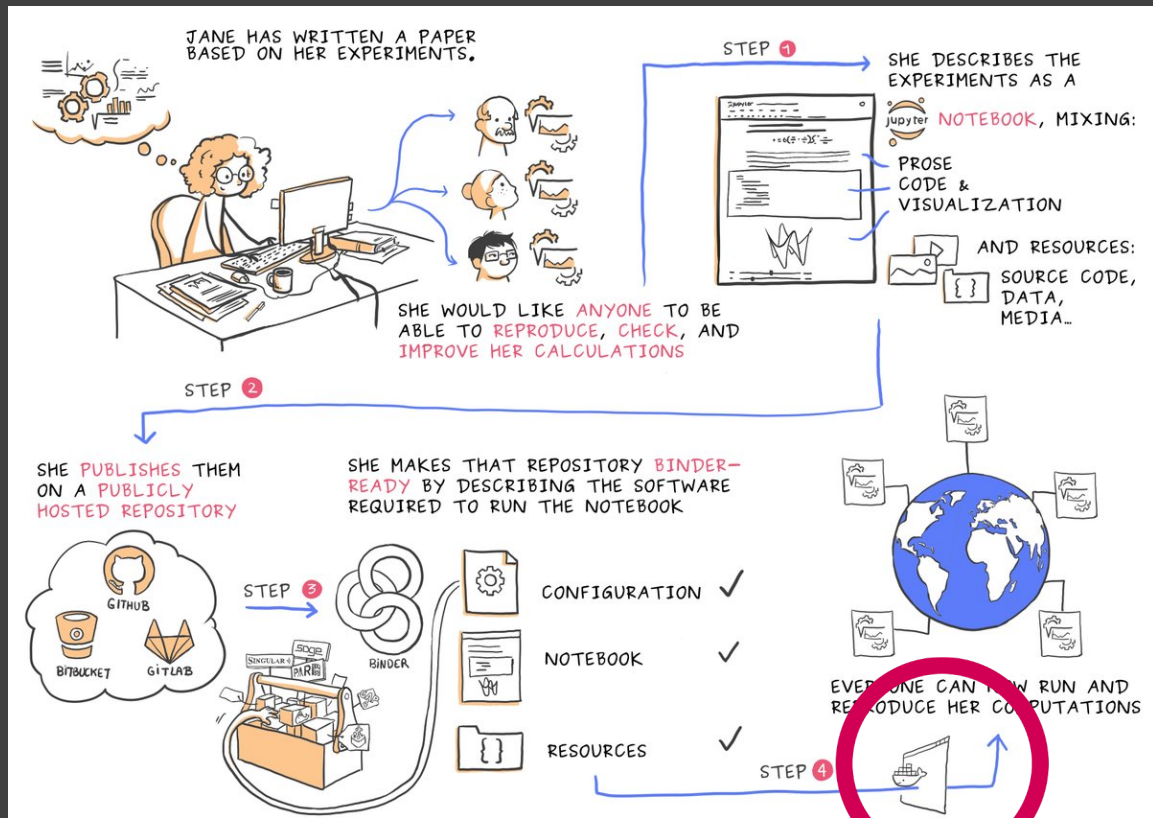
<https://discourse.jupyter.org/t/about-the-binder-category/200>

@martinagvilas, @turingway #TuringWay



Courtesy of Juliette Taka: <https://twitter.com/mybinderteam/status/1082556317842264064>

@martinagvilas, @turingway #TuringWay

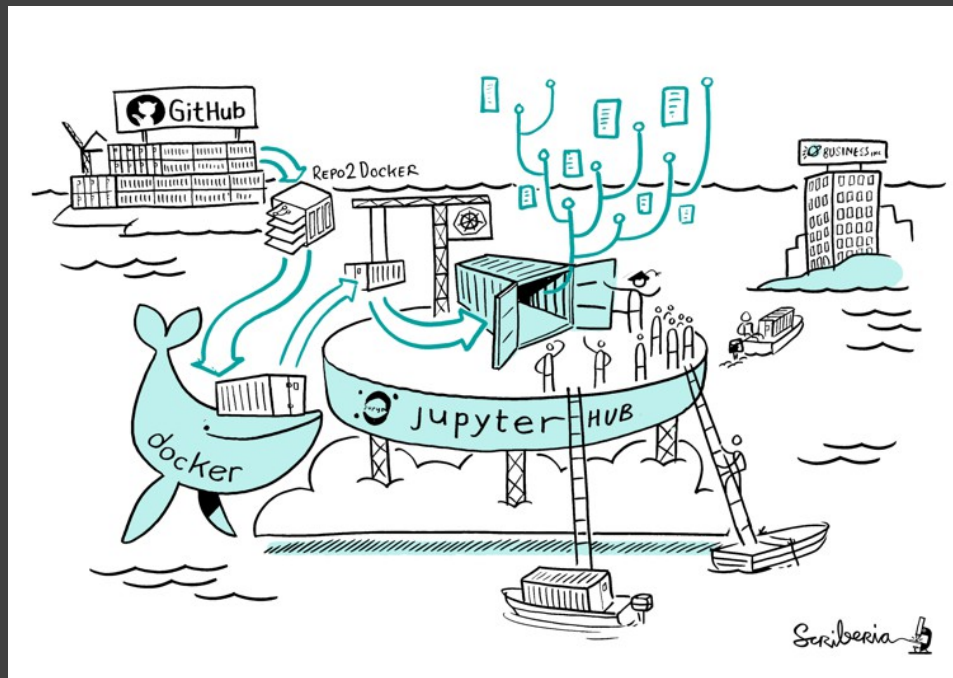


Courtesy of Juliette Taka: <https://twitter.com/mybinderteam/status/1082556317842264064>

@martinagvilas, @turingway #TuringWay

BinderHub

- cloud-based technology
- can launch a repository of code in a browser
- allows the user to execute and interact with the code

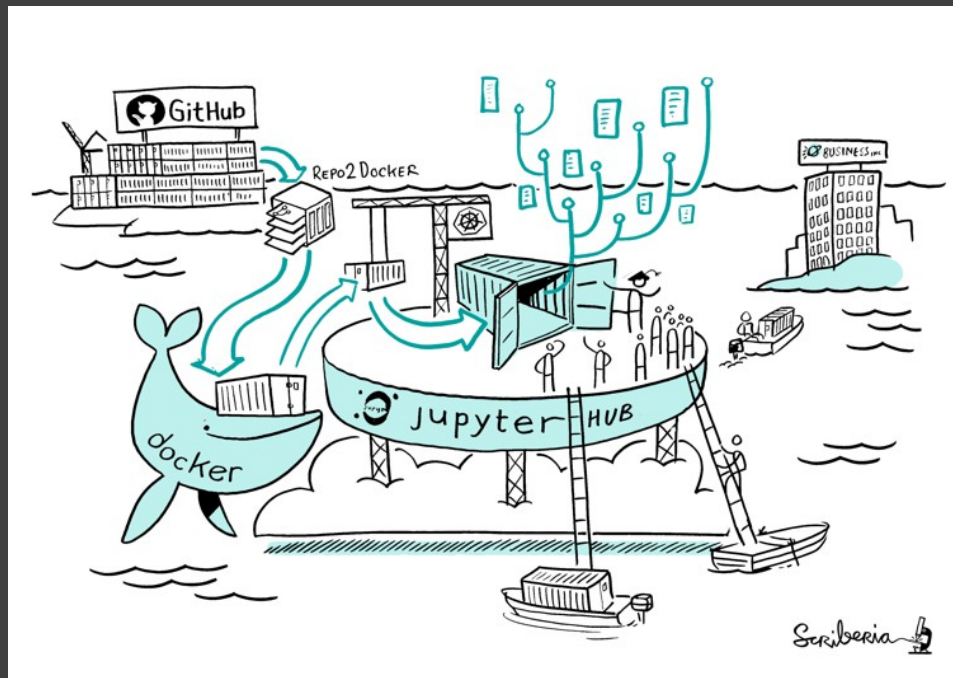


<https://the-turing-way.netlify.app/reproducible-research/binderhub/binderhub-compute.html>

@martinagvilas, @turingway #TuringWay

BinderHub

- repo2docker
- Kubernetes
- Helm
- JupyterHub
- a cloud service platform




<https://the-turing-way.netlify.app/reproducible-research/binderhub/binderhub-compute.html>

@martinagvilas, @turingway #TuringWay

mybinder.org

- online service for sharing computational environments from online repositories
- a federation of BinderHub deployments

Thanks to Google Cloud, OVH, 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

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

HEAD File

Copy the URL below and share your Binder with others:

Copy the text below, then paste into your README to show a binder badge:  [launch binder](#)

<https://mybinder.org/>

@martinagvilas, @turingway #TuringWay

gke.mybinder.org



Run by [The Binder Team](#)

Funded by [Google Cloud Platform](#)

ovh.mybinder.org



Run by [The OVH Team](#)

Funded by [OVH](#)

gegis.mybinder.org



Run by [The GESIS Notebooks Team](#)

Funded by [GESIS](#)

turing.mybinder.org



Run by [The Turing Way team led by Sarah Gibson](#)


Funded by [The Alan Turing Institute](#)











<https://mybinder.readthedocs.io/en/latest/about/about.html>

@martinagvilas, @turingway #TuringWay

Example 1


This example demonstrates a reproducible 4-step workflow for predicting a protein fold classification using a Machine Learning approach.







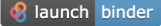
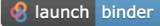
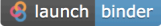
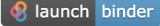
Rule 9: Design Your Notebooks to Be Read, Run, and Explored. The nbviewer links below provide a non-interactive preview of notebooks and  buttons launch Jupyter Notebook or Jupyter Lab in your web browser using the Binder (mybinder.org) server (may be slow!). (See the Binder website how to setup links to a Git repository.) The HTML links provide a permanent static record of the notebooks. All notebooks can also be launched directly from the links in the 0-Workflow.ipynb top-level notebook.

Nbviewer	Jupyter Notebook	Jupyter Lab	HTML
0-Workflow.ipynb			HTML
1-CreateDataset.ipynb			HTML
2-CalculateFeatures.ipynb			HTML
3-FitModel.ipynb			HTML
4-Predict.ipynb			HTML

Example 1

This example demonstrates a reproducible 4-step workflow for predicting a protein fold classification using a Machine Learning approach.

Rule 9: Design Your Notebooks to Be Read, Run, and Explored. The nbviewer links below provide a non-interactive preview of notebooks and  buttons launch Jupyter Notebook or Jupyter Lab in your web browser using the Binder (mybinder.org) server (may be slow!). (See the Binder website how to setup links to a Git repository.) The HTML links provide a permanent static record of the notebooks. All notebooks can also be launched directly from the links in the 0-Workflow.ipynb top-level notebook.

Nbviewer	Jupyter Notebook	Jupyter Lab	HTML
0-Workflow.ipynb			HTML
1-CreateDataset.ipynb			HTML
2-CalculateFeatures.ipynb			HTML
3-FitModel.ipynb			HTML
4-Predict.ipynb			HTML

Thanks to [Google Cloud](#), [OVH](#), [GESIS Notebooks](#) and the [Turing Institute](#) for supporting us! 🍷



Starting repository: `jupyter-guide/ten-rules-jupyter/master`

Take a look at our [gallery of example repositories](#).

Build logs

[show](#)

<https://github.com/jupyter-guide/ten-rules-jupyter#example-1>

@martinagvilas, @turingway #TuringWay



Create Dataset

This notebook extracts from the Protein Data Bank information about the secondary structure of proteins. The ultimate goal is to assign a fold classification from a protein sequence.

Rule 2: Document the Process, Not Just the Results. Here we describe the steps how to produce the dataset.

Rule 7: Build a Pipeline. Besides documenting all steps, the entire process of dataset creation from the original data files in the /data directory is automated. There are no manual steps.


Rule 8: Share and Explain Your Data. To enable reproducibility we provide a /data directory with data files and a file that describes the datasets with download locations and dates.











```
In [1]: # column names
value_col = "foldClass" # fold class to be predicted
```

```
In [2]: import pandas as pd
import numpy as np
import pdbutils
```


Example 1

This example demonstrates a reproducible 4-step workflow for predicting a protein fold classification using a Machine Learning approach.

Rule 9: Design Your Notebooks to Be Read, Run, and Explored. The nbviewer links below provide a non-interactive preview of notebooks and  buttons launch Jupyter Notebook or Jupyter Lab in your web browser using the Binder (mybinder.org) server (may be slow!). (See the Binder website how to setup links to a Git repository.) The HTML links provide a permanent static record of the notebooks. All notebooks can also be launched directly from the links in the 0-Workflow.ipynb top-level notebook.

Nbviewer	Jupyter Notebook	Jupyter Lab	HTML
0-Workflow.ipynb			HTML
1-CreateDataset.ipynb			HTML
2-CalculateFeatures.ipynb			HTML
3-FitModel.ipynb			HTML
4-Predict.ipynb			HTML

<https://github.com/jupyter-guide/ten-rules-jupyter#example-1>

@martinagvilas, @turingway #TuringWay

The screenshot shows a JupyterLab environment. On the left is a file browser for the directory `/example1/`. It contains a table of files and folders, all last modified 5 months ago:

Name	Last Modified
data	5 months ago
intermediate_data	5 months ago
0-Workflow.html	5 months ago
0-Workflow.ipynb	5 months ago
1-CreateDataset....	5 months ago
1-CreateDataset....	5 months ago
2-CalculateFeatu...	5 months ago
2-CalculateFeatu...	5 months ago
3-FitModel.html	5 months ago
3-FitModel.ipynb	5 months ago
4-Predict.html	5 months ago
4-Predict.ipynb	5 months ago
mlutils.py	5 months ago
pdbutils.py	5 months ago
protectors.py	5 months ago

The main notebook window is titled `1-CreateDataset.ipynb` and shows the following content:

Create Dataset

This notebook extracts from the Protein Data Bank information about the secondary structure of proteins. The ultimate goal is to assign a fold classification from a protein sequence.

Rule 2: Document the Process, Not Just the Results. Here we describe the steps how to produce the dataset.

Rule 7: Build a Pipeline. Besides documenting all steps, the entire process of dataset creation from the original data files in the `/data` directory is automated. There are no manual steps.

Rule 8: Share and Explain Your Data. To enable reproducibility we provide a `/data` directory with data files and a file that describes the datasets with download locations and dates.

```
[1]: # column names
value_col = "foldClass" # fold class to be predicted
```

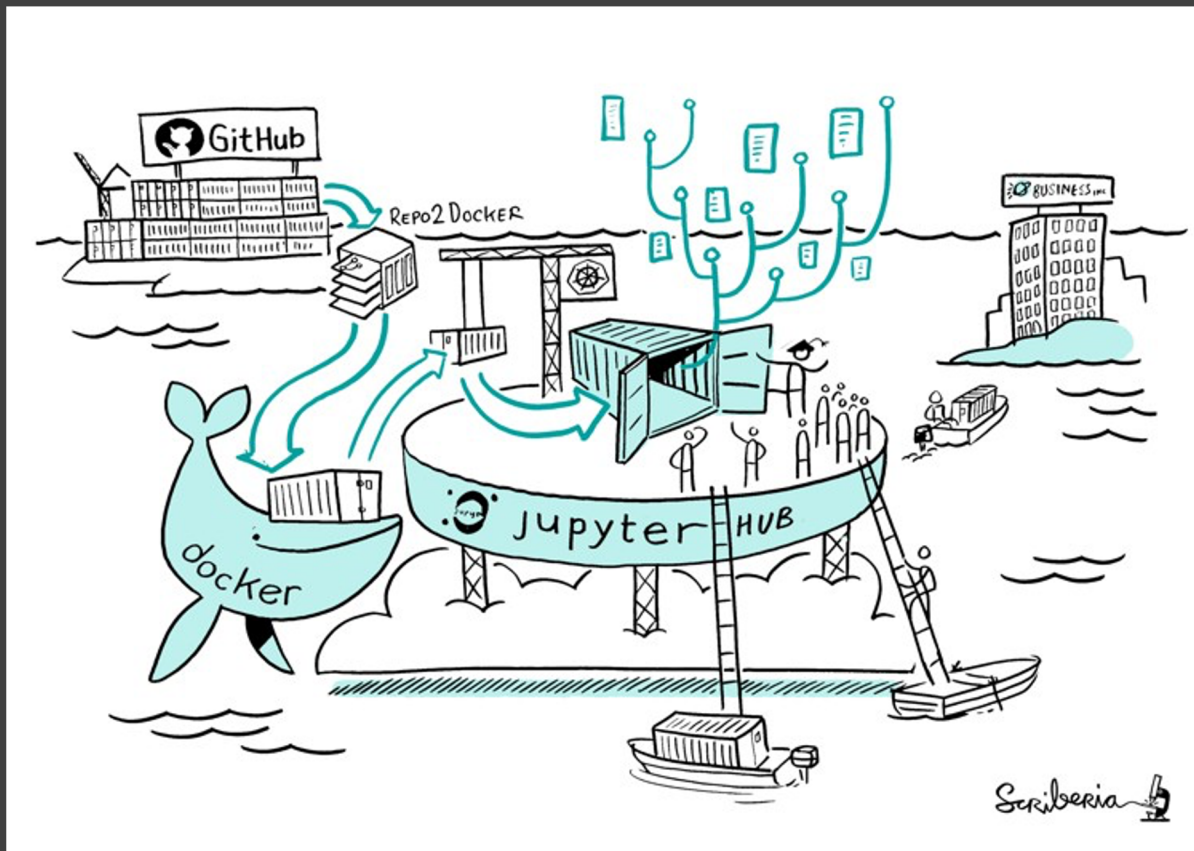
```
[2]: import pandas as pd
import numpy as np
import pdbutils
```

Kirstie Whitaker

“Binder is great because it also encourages reproducible practices in the communication.”

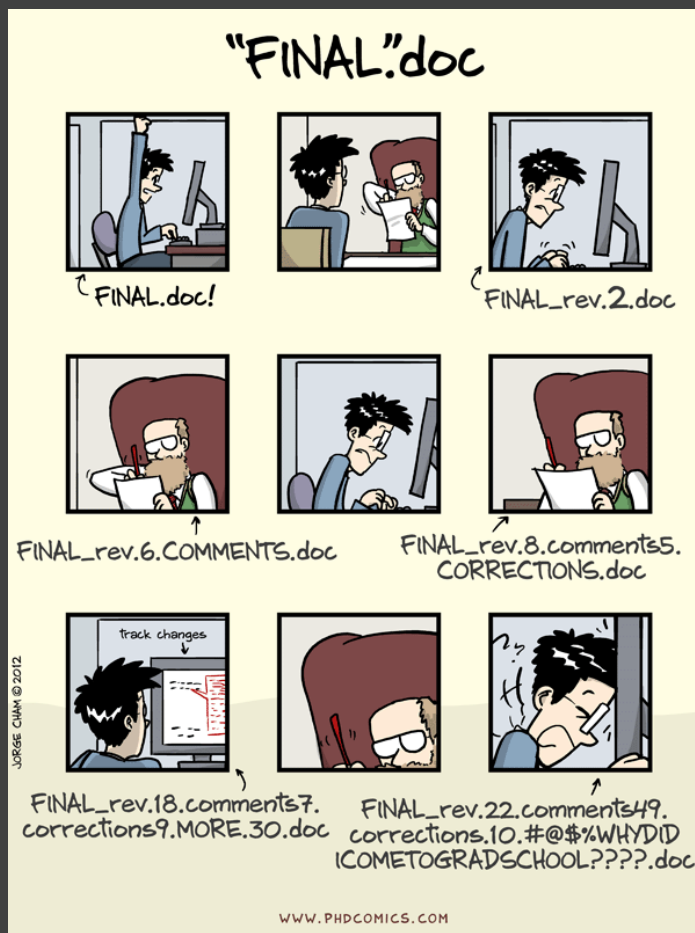


how does *Binder* work?



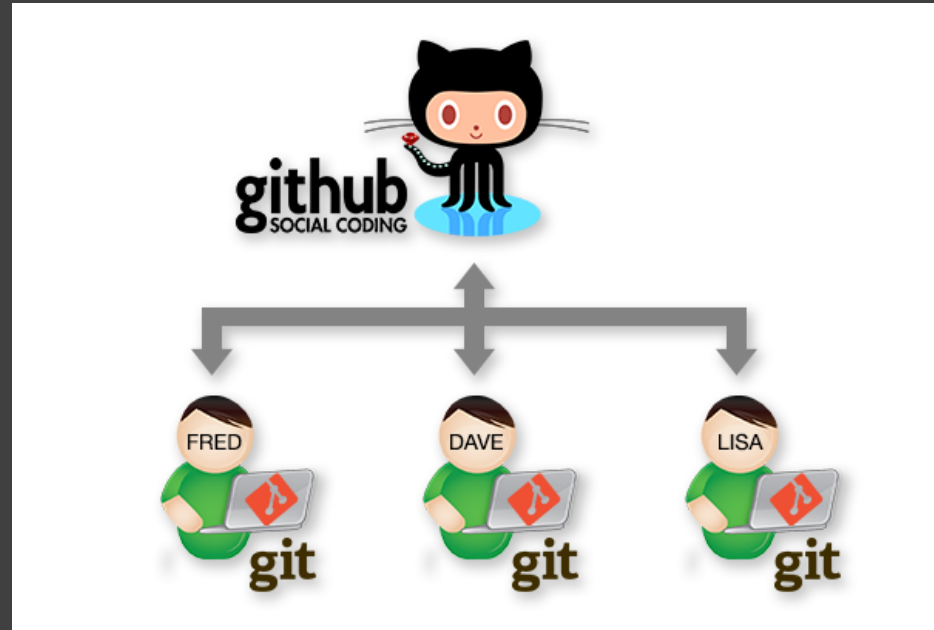


- version control system
- records changes to a file or set of files over time
- provides access to any specific version



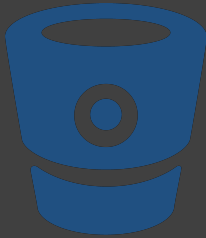


- online hosting platform for git repositories
- enables collaborative work



preparing a repository for Binder

1. The repository is in a public location online



preparing a repository for Binder

1. The repository is in a public location online
2. The repository does not require any personal or sensitive information (such as passwords)

preparing a repository for Binder

1. The repository is in a public location online
2. The repository does not require any personal or sensitive information (such as passwords)
3. The repository contains content designed for people to read

preparing a repository for Binder

1. The repository is in a public location online
2. The repository does not require any personal or sensitive information (such as passwords)
3. The repository contains content designed for people to read
4. The repository has configuration files that specify its computational environment

preparing a repository for Binder

1. The repository is in a public location online
2. The repository does not require any personal or sensitive information (such as passwords)
3. The repository contains content designed for people to read.
4. The repository has **configuration files** that specify its computational environment

configuration file

- defines your computational environment

```
requirements.txt ×
1 # Requirements for the demo notebooks
2 # Useful for MyBinder configuration
3 pandas==1.0.5
4 numpy==1.19.0
5 matplotlib==3.2.2
6 datascience
7 folium
8 jupyter-book==0.8.2
9 sphinxcontrib-bibtex==1.0.0
10
```

pip freeze

- Python specific
- captures the versions of all packages that you're currently using
- can print to screen or save in a file named `requirements.txt`

Examples

1. Generate output suitable for a requirements file.

Unix/macOS **Windows**

```
$ python -m pip freeze
docutils==0.11
Jinja2==2.7.2
MarkupSafe==0.19
Pygments==1.6
Sphinx==1.2.2
```

2. Generate a requirements file and then install from it in another environment.

Unix/macOS **Windows**

```
env1/bin/python -m pip freeze > requirements.txt
env2/bin/python -m pip install -r requirements.txt
```

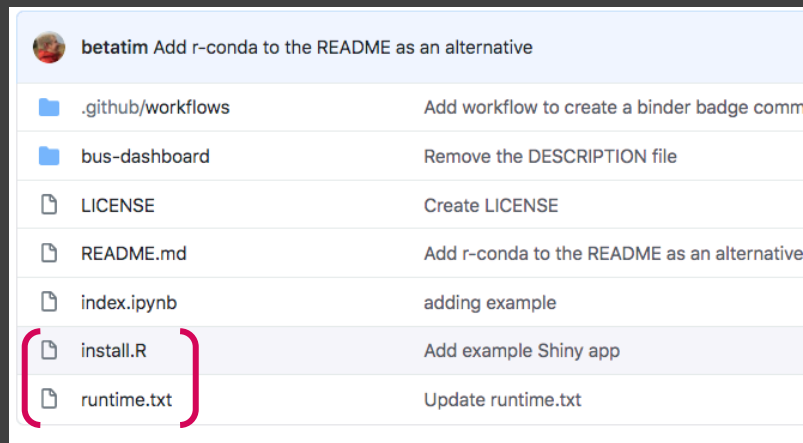









- environment, package and version management system
- for multiple languages
- Information about installed software saved in file called `environment.yml`

```
! environment.yml x
1  name: example-environment
2  channels:
3    - conda-forge
4  dependencies:
5    - python
6    - numpy
7    - pip
8    - pip:
9      - nbgitpuller
10     - sphinx-gallery
11     - pandas
12     - matplotlib
13
```



R environment



- support for R and RStudio with libraries pinned to a specific snapshot on MRAN, defined in `runtime.txt`
- `install.R` specifies one library to install per line






betatim Add r-conda to the README as an alternative	
 .github/workflows	Add workflow to create a binder badge comm
 bus-dashboard	Remove the DESCRIPTION file
 LICENSE	Create LICENSE
 README.md	Add r-conda to the README as an alternative
 index.ipynb	adding example
 install.R	Add example Shiny app
 runtime.txt	Update runtime.txt

r / runtime.txt

 **betatim** Update runtime.txt 



👤 2 contributors  





Raw Blame   




1 lines (1 sloc) | 17 Bytes

```
1 r-3.6-2019-04-12
```

r / install.R

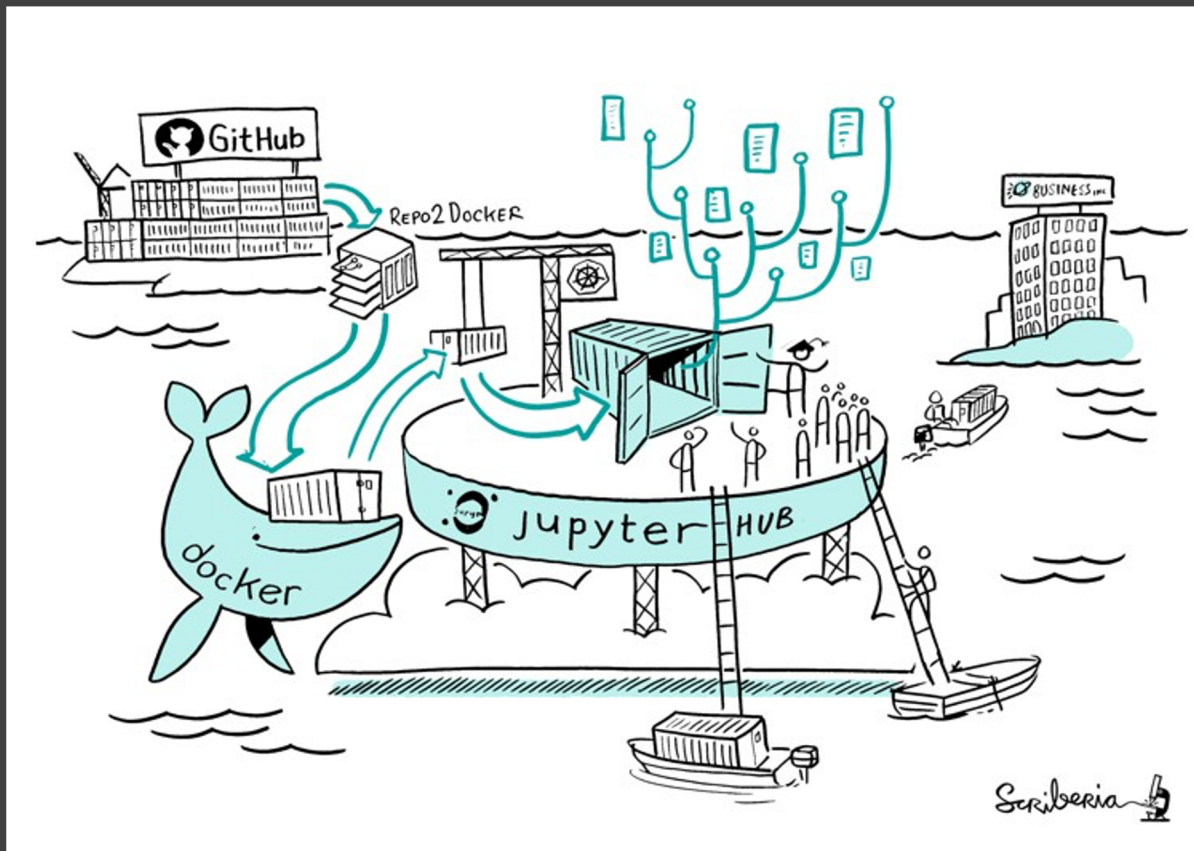
 **betatim** Add example Shiny app 

👤 4 contributors    

Raw Blame   

5 lines (5 sloc) | 148 Bytes

```
1 install.packages("tidyverse")
2 install.packages("rmarkdown")
3 install.packages("httr")
4 install.packages("shinydashboard")
5 install.packages('leaflet')
```





repo2docker

- automatically builds a Docker image from a code repository given a configuration file

Calling repo2docker

repo2docker is called with this command:

```
jupyter-repo2docker <source-repository>
```

where `<source-repository>` is:

- a URL of a Git repository (<https://github.com/binder-examples/requirements>),
- a Zenodo DOI ([10.5281/zenodo.1211089](https://doi.org/10.5281/zenodo.1211089)),
- a SWHID ([swh:1:rev:999dd06c7f679a2714dfe5199bdca09522a29649](https://www.softwareherald.org/swh/packages/999dd06c7f679a2714dfe5199bdca09522a29649)), or
- a path to a local directory (`a/local/directory`)

of the source repository you want to build.

For example, the following command will build an image of Peter Norvig's `Pytudes` repository:

```
jupyter-repo2docker https://github.com/norvig/pytudes
```

Building the image may take a few minutes.

containers



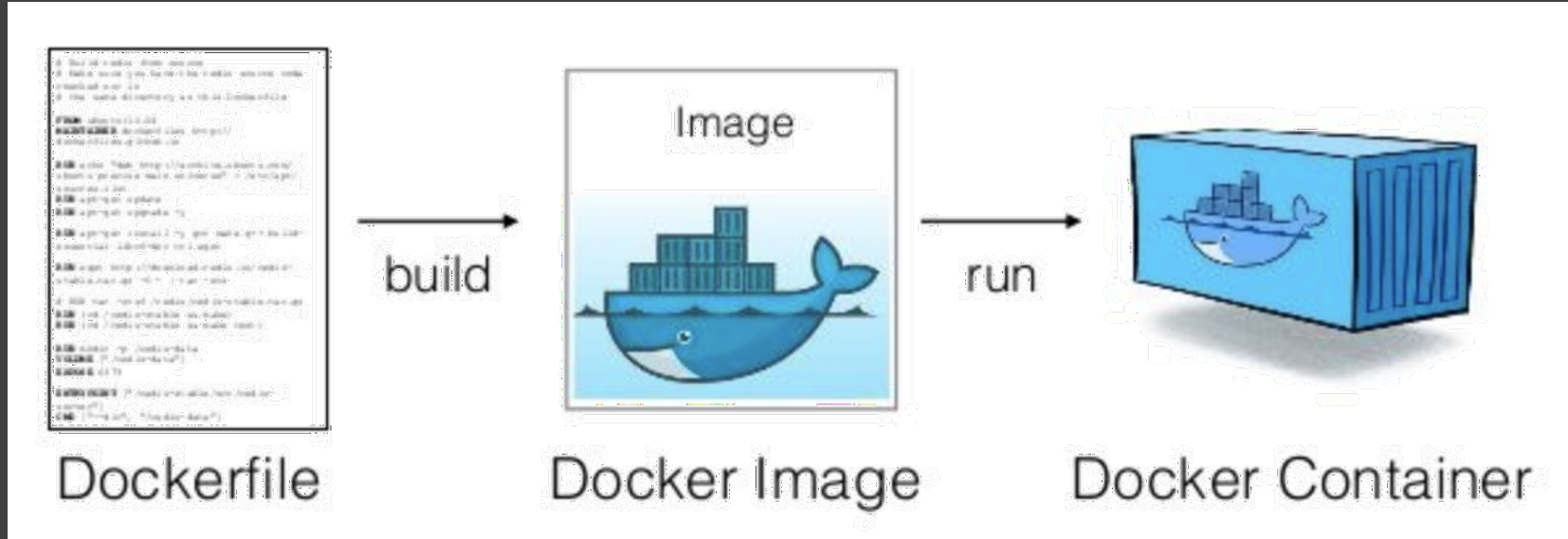
- package a project with all of the parts it needs - such as libraries, dependencies, and system settings
- anyone can then open up a container and work within it
- the computational environment of the container is identical across instances

containers



- behaves like a virtual machine
- more lightweight -> only contains the individual components needed to operate the project

containers



containers



master minimal-dockerfile / Dockerfile Go to file ...

betatim Switch USER at the end to not run as ro... Latest commit dbcf36b on 28 Feb 2019 History

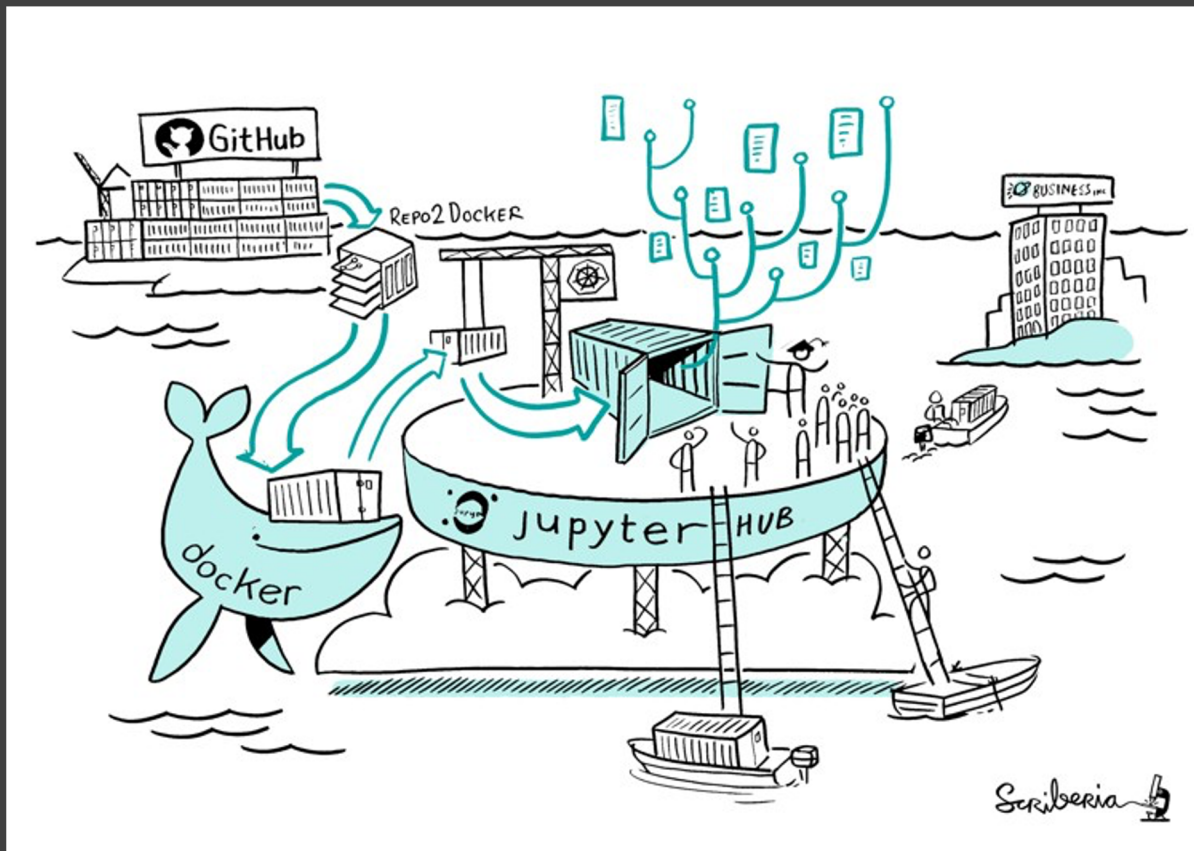
2 contributors

17 lines (15 sloc) 370 Bytes Raw Blame

```
1 FROM python:3.7-slim
2 # install the notebook package
3 RUN pip install --no-cache --upgrade pip && \
4     pip install --no-cache notebook
5
6 # create user with a home directory
7 ARG NB_USER
8 ARG NB_UID
9 ENV USER ${NB_USER}
10 ENV HOME /home/${NB_USER}
11
12 RUN adduser --disabled-password \
13     --gecos "Default user" \
14     --uid ${NB_UID} \
15     ${NB_USER}
16 WORKDIR ${HOME}
17 USER ${USER}
```

<https://github.com/binder-examples/minimal-dockerfile/blob/master/Dockerfile>

@martinagvilas, @turingway #TuringWay



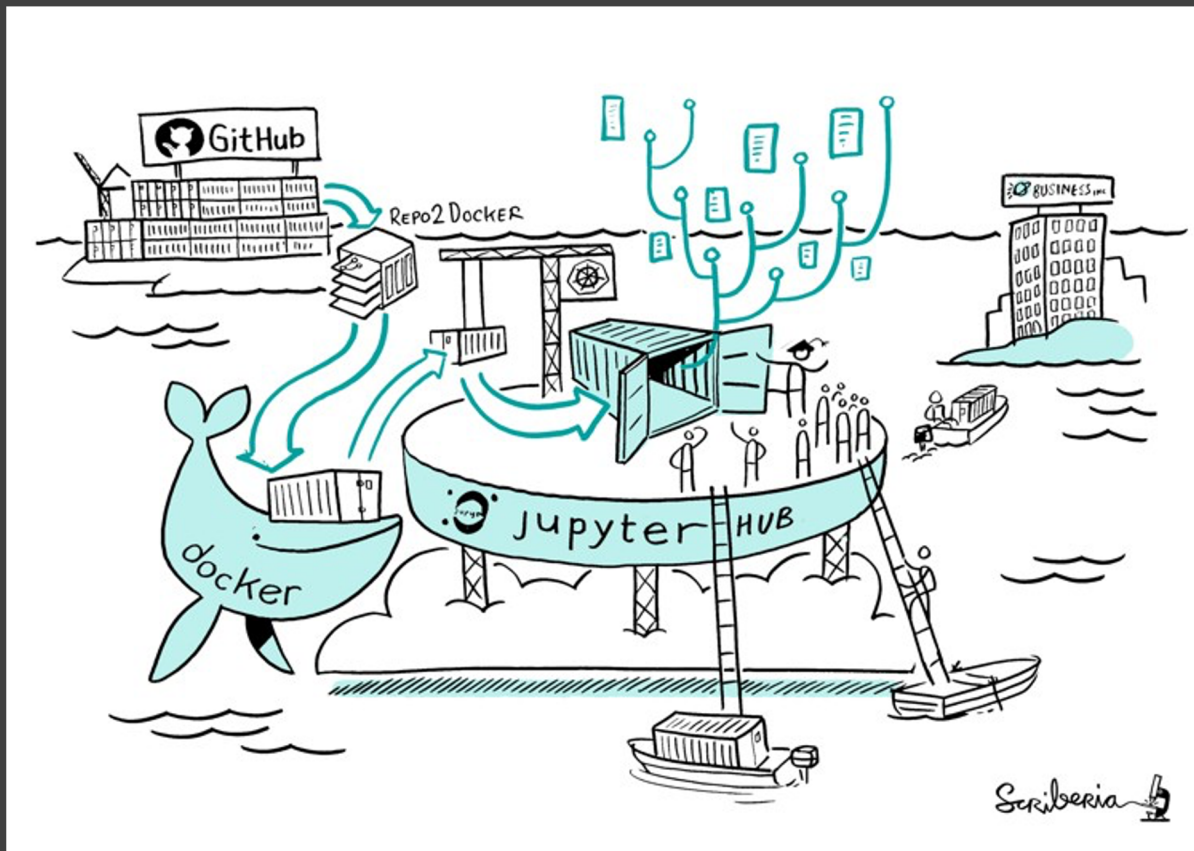
JupyterHub

- allows users to interact with a computing environment through a webpage
- “the cloud is just someone else’s computer” @kirstie_j



JupyterHub

1. JupyterHub creates a Kubernetes pod for the user that serves the built Docker image for the repository.
2. JupyterHub monitors the user's pod for activity, and destroys it after a short period of inactivity.



Sarah Gibson

“It took me a while to feel like I knew enough to contribute to Binder. But the team are always so excited to have my input. Its really motivating to be part of such a welcoming community.”



small *group exercise*

small group exercise

https://github.com/alan-turing-institute/the-turing-way/blob/master/workshops/boost-research-reproducibility-binder/paired_examples.md