The Alan Turing Institute

The Turing Way Workshop: Reproducible, Open and FAIR Research

Session 2

Emma Karoune & Malvika Sharan

Pronouns: she/her/hers

Assignment: Create a project repository

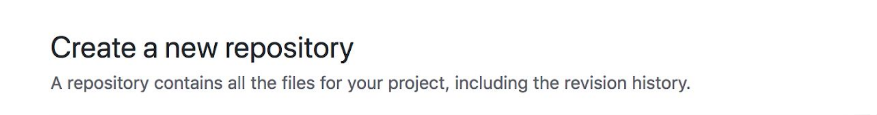
Add top-level files: README and LICENSE

- README to communicate about your work
- Also try to add a License - to allow others to use, modify, build upon your work
  - We will cover licenses in the next session in more detail.

Use The Turing Way chapter for README to guide your assignment

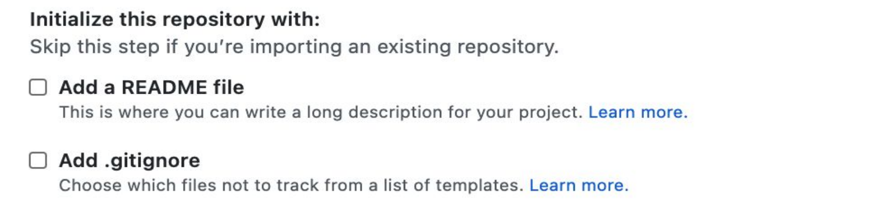
https://the-turing-way.netlify.app/project-design/project-repo/project-repo-readme.html

Create a GitHub Repository with README file



Create a new repository

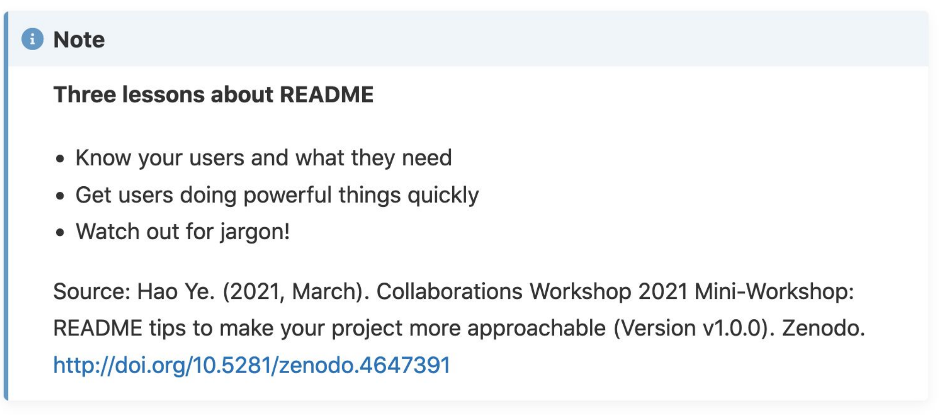
A repository contains all the files for your project, including the revision history.



Initialize this repository with:
Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. Learn more.

☐ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.



Note

Three lessons about README

- Know your users and what they need
- Get users doing powerful things quickly
- Watch out for jargon!

Source: Hao Ye. (2021, March). Collaborations Workshop 2021 Mini-Workshop: README tips to make your project more approachable (Version v1.0.0). Zenodo. http://doi.org/10.5281/zenodo.4647391

End of Part 1: Further Reading and Examples

- *The Turing Way* README
- Open Life Science README
- Purple Booth’s README Template
- Thoughtbot’s Blog on How to Write a Good README
- Matias Singer’s curated List of Awesome READMES

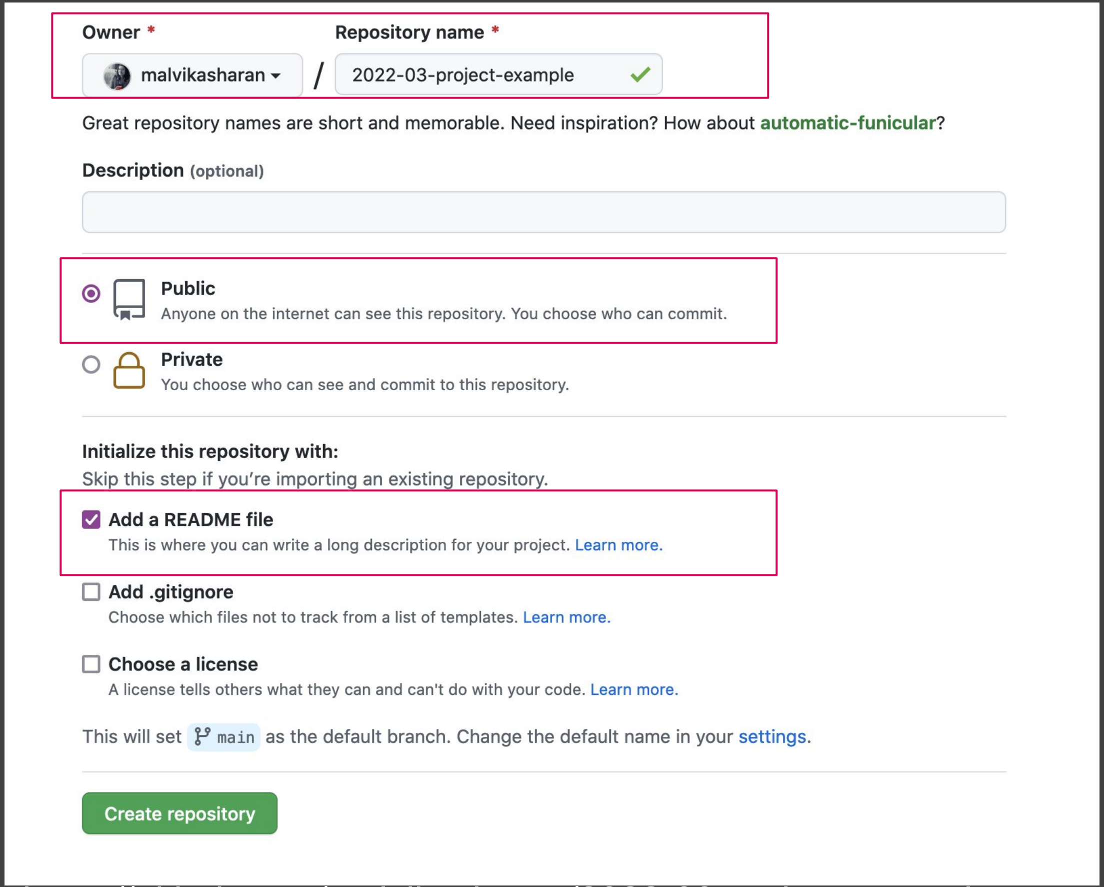
Assignment: Create a project repository

Add top-level files: README and LICENSE

- README to communicate about your work
- Also try to add a License - to allow others to use, modify, build upon your work
    - We will cover licenses in the next session in more detail.

Use The Turing Way chapter for README to guide your assignment

https://the-turing-way.netlify.app/project-design/project-repo/project-repo-readme.html



Owner * malvikasharan / Repository name * 2022-03-project-example ✓

Great repository names are short and memorable. Need inspiration? How about **automatic-funicular**?

Description (optional)

● **Public**
Anyone on the internet can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

Initialize this repository with:
Skip this step if you're importing an existing repository.

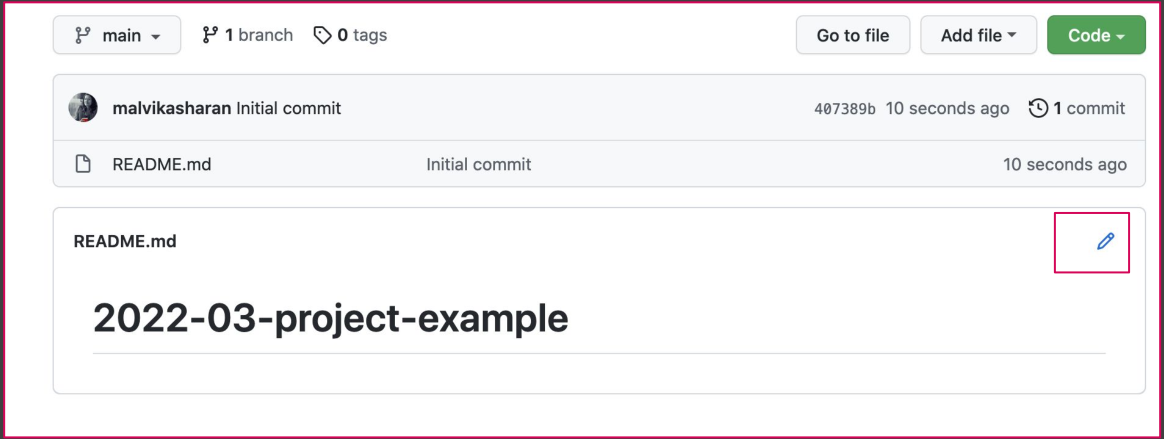
☑ **Add a README file**
This is where you can write a long description for your project. Learn more.

□ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

□ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

This will set `main` as the default branch. Change the default name in your settings.

https://github.com/malvikasharan/2022-03-project-example



malvikasharan Initial commit 407389b 10 seconds ago 1 commit

README.md Initial commit 10 seconds ago

README.md

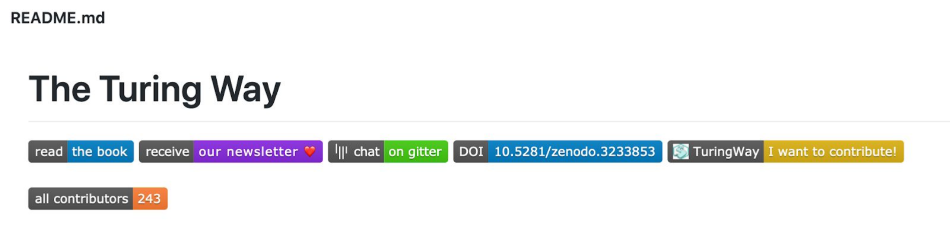
2022-03-project-example

1. Copy Template:

   https://raw.githubusercontent.com/ha0ye/CW21-README-tips/main/template_README.md

2. Add your project information

Bonus: Badges and Additional Information

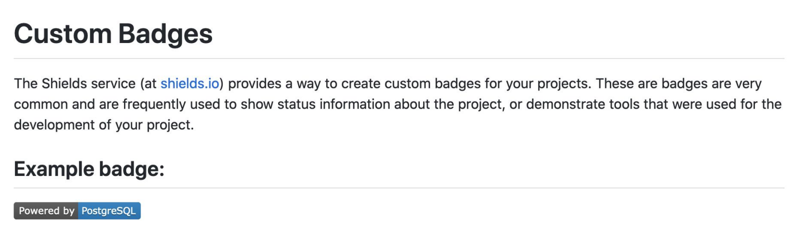


README.md

The Turing Way

read the book | receive our newsletter ❤️ | chat on gitter | DOI 10.5281/zenodo.3233853 | TuringWay I want to contribute!

all contributors 243

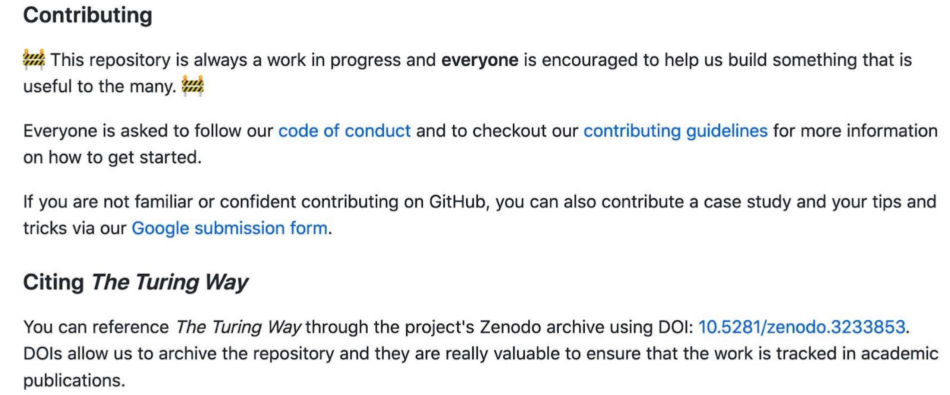


Custom Badges

The Shields service (at shields.io) provides a way to create custom badges for your projects. These are badges are very common and are frequently used to show status information about the project, or demonstrate tools that were used for the development of your project.

Example badge:

Powered by PostgreSQL



Contributing

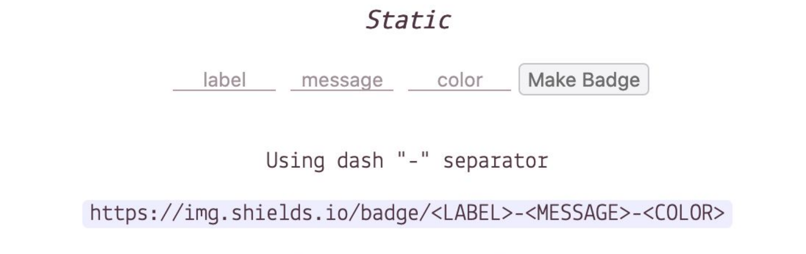
🚧 This repository is always a work in progress and **everyone** is encouraged to help us build something that is useful to the many. 🚧

Everyone is asked to follow our code of conduct and to checkout our contributing guidelines for more information on how to get started.

If you are not familiar or confident contributing on GitHub, you can also contribute a case study and your tips and tricks via our Google submission form.

Citing *The Turing Way*

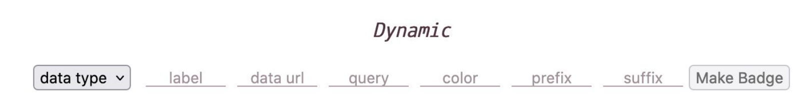
You can reference *The Turing Way* through the project's Zenodo archive using DOI: 10.5281/zenodo.3233853. DOIs allow us to archive the repository and they are really valuable to ensure that the work is tracked in academic publications.



Static

Using dash "-" separator

```
https://img.shields.io/badge/<LABEL>-<MESSAGE>-<COLOR>
```



Dynamic

https://shields.io/, @turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

The Alan Turing Institute

The Turing Way Workshop: Reproducible, Open and FAIR Research

Session 2

Emma Karoune & Malvika Sharan

Pronouns: she/her/hers

Project communication, collaboration and sharing

- License to allow reuse of your research outputs in a way YOU want
- Make your work citable: Digital Object Identifiers, Zenodo
- Setting online repository for collaboration
  - CoC, Contributing guidelines, continuous integration
- Reproducible environment and sharing to aid reproducibility: Binder
- Revisiting how these contribute to implementing FAIR practices
- What more you can do.

Open Licenses

Disclaimer: We are not lawyers

Licensing Chapter in The Turing Way

Motivation:

Allow others to use, remix and share your work.

Method:

Add an open license for use, remixing and sharing.

Addressing Common Concerns

- Sharing something on GitHub does **NOT automatically make it reusable**.
- Sharing with a license does **NOT give away your rights** to publish, sell, etc.
- Work shared with an open license **NOT** to be used **without attribution**.
- *Not attributing to authors violates academic ethics.*

Truly Open Licenses: common elements

use

Anyone can use the work for any purpose

modify

Anyone can modify the work

share

Anyone can redistribute both the original and modified work

> “Open source software is software that can be freely used, modified, and shared (in both modified and unmodified form) by anyone.“

- GitHub Glossary, Open Source

Attribution

Most open licenses require crediting the authors of the work.

Non-copyleft: Permissive, non-reciprocal - CC BY, MIT, BSD

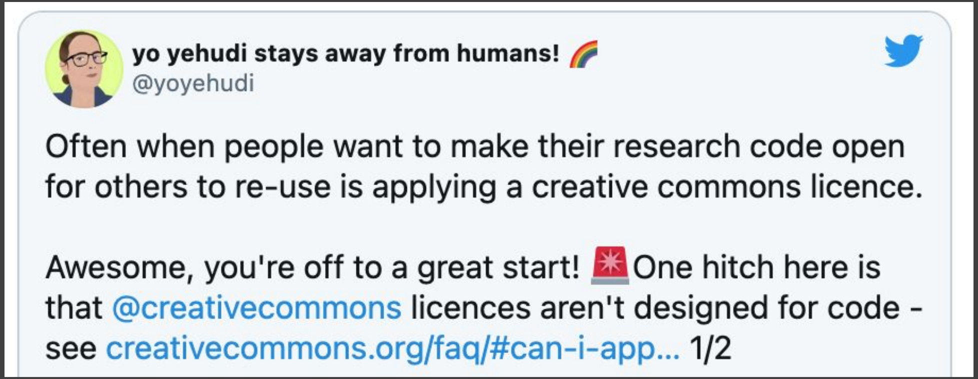
- **do not require** derivative works to shared with the same license

Copyleft: Viral, reciprocal - CC BY-SA, GPLv3, MPL-2.0

- **require** derivative works to shared with the same license

Exception: CC0 (public domain, no copyright holder)

How to Apply License



yo yehudi stays away from humans! 🌈
@yoyehudi

Often when people want to make their research code open for others to re-use is applying a creative commons licence.

Awesome, you're off to a great start! 🚨One hitch here is that @creativecommons licences aren't designed for code - see creativecommons.org/faq/#can-i-app... 1/2

Place the **full text of the license in a file named LICENSE** in the root directory.

Apply licenses to all components, content: CC-BY, software: MIT, data: CC0

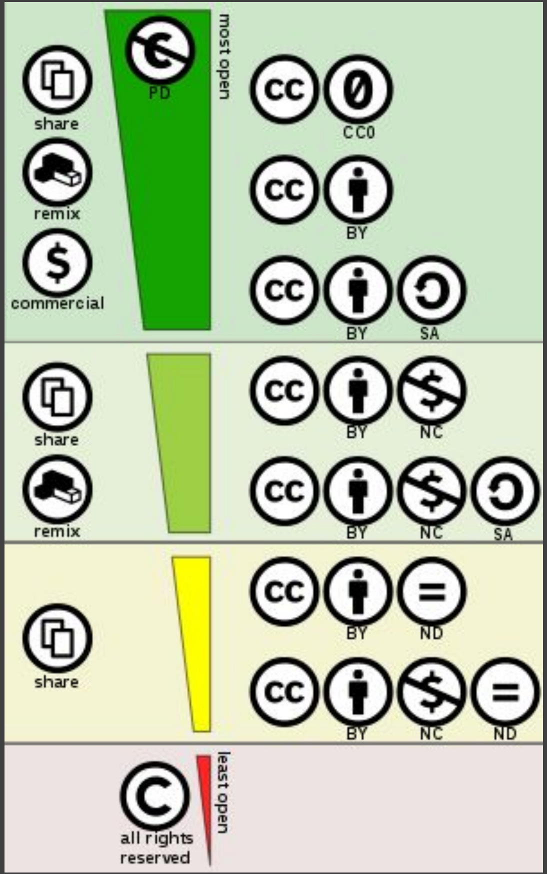
Describe in README which license applies to which parts of your work.

Assignment: Add a License

Use a different license for content, code and data. For example

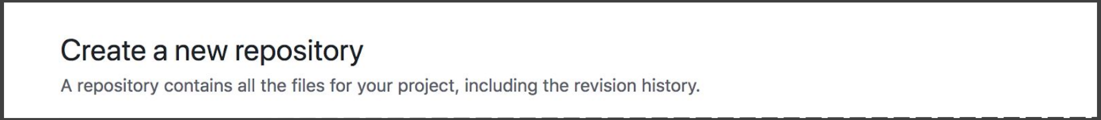
- Writing/docs/images/metadata: CC-BY
- Code: MIT License
- Data will also have different license

  based on what you can/can't share!



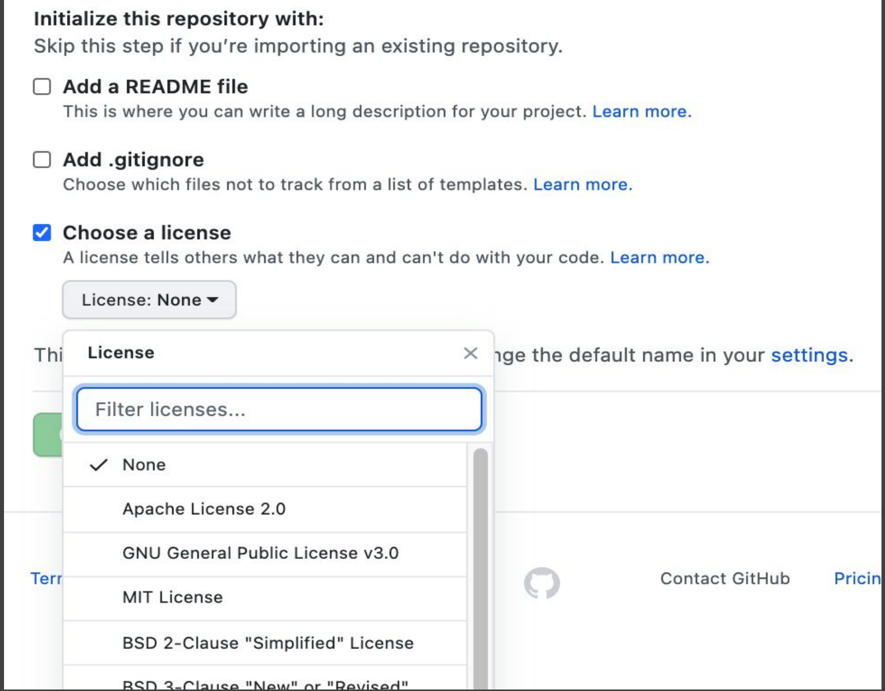
@openlifesci, CC-BY 4.0, The Turing Way, Image: Shaddim; Original CC license icons licensed under CC BY 4.0

GitHub can add a license for you



Create a new repository

A repository contains all the files for your project, including the revision history.



Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ **Add a README file**
This is where you can write a long description for your project. Learn more.

☐ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

☑ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

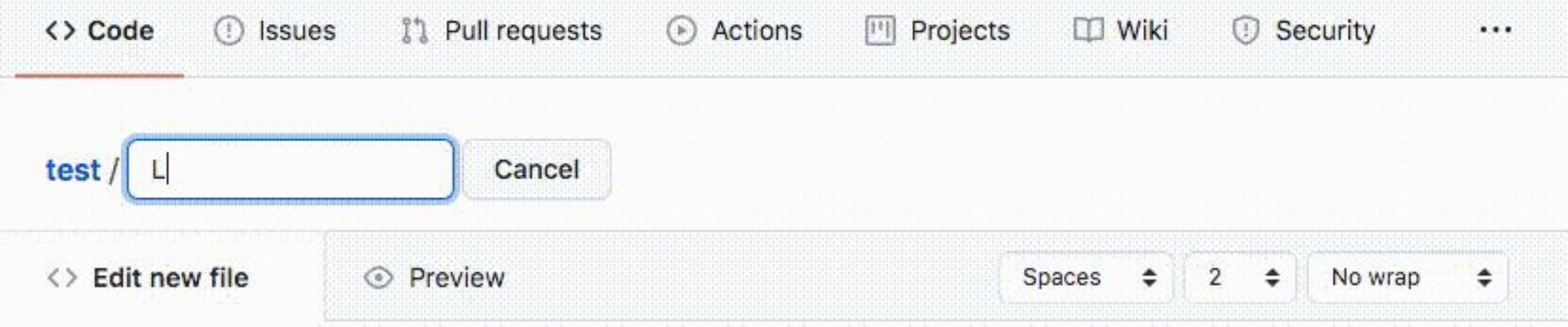
Thi nge the default name in your settings.

License
Filter licenses...
✓ None
Apache License 2.0
GNU General Public License v3.0
MIT License
BSD 2-Clause "Simplified" License
BSD 3-Clause "New" or "Revised"
Terr
Contact GitHub
Pricin

@openlifesci, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.4609987

GitHub can add a license for you, Part 2

when adding a new file named LICENSE...



Further Reading

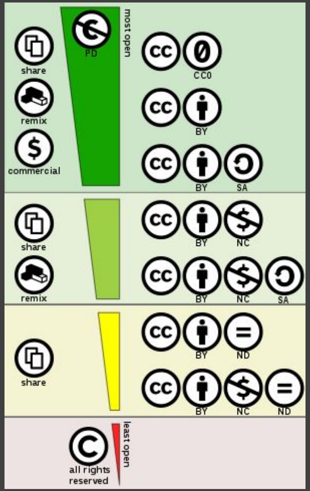
Licensing Chapter in *The Turing Way*

- The Open Source Definition (10 Criteria) | opensource.org
- Legal Matters | producingoss.com
- Open Source Guide | opensource.guide
- Software: Choose an Open Source License | choosealicense.com
- Content: Choose a License | creativecommons.org

Breakout

Make sure that you have done the following:

- Set a project repository
- Add a minimal README
- Add a License

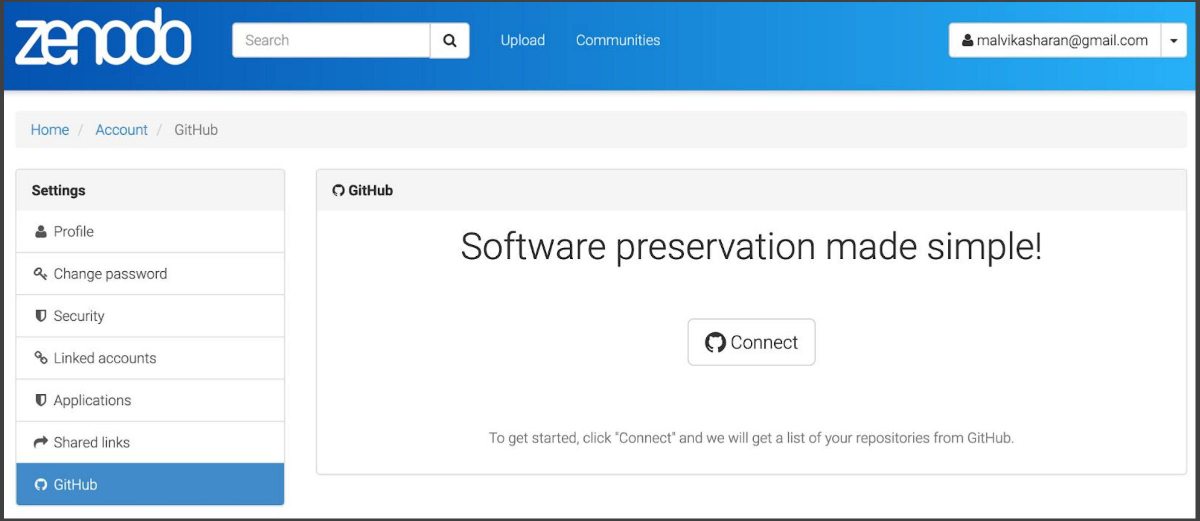


You can use part of your project to use as example for this workshop!

You can fork: https://github.com/malvikasharan/2022-03-project-example

@openlifesci, CC-BY 4.0, The Turing Way, Image: Shaddim; Original CC license icons licensed under CC BY 4.0

Making your work citable



zenodo

Search

Upload

Communities

malvikasharan@gmail.com

Home / Account / GitHub

Settings

- Profile
- Change password
- Security
- Linked accounts
- Applications
- Shared links
- GitHub

GitHub

Software preservation made simple!

Connect

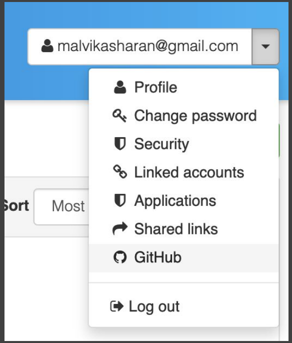
To get started, click "Connect" and we will get a list of your repositories from GitHub.

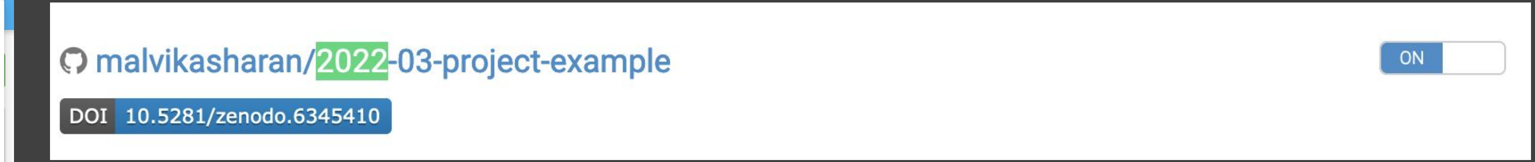
A digital object identifier (DOI) is a persistent identifier used to identify research objects uniquely.

- Zenodo, Figshare (research objects)
- Data Dryad (data)
- Preprint (manuscript)
- Docker (env)
- ORCID (researchers)

https://guides.github.com/activities/citable-code/, @turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

Zenodo

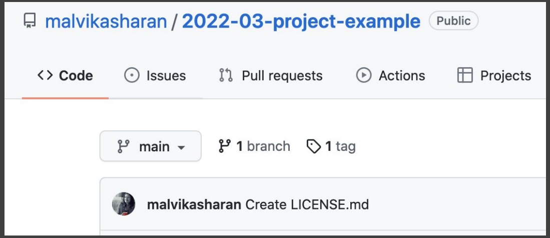



malvikasharan/2022-03-project-example

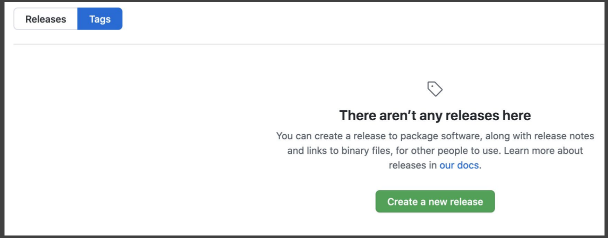
ON

DOI 10.5281/zenodo.6345410

GitHub

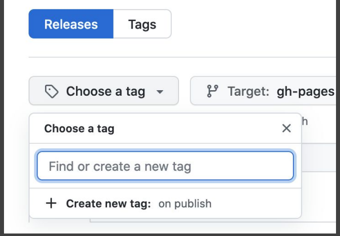

malvikasharan / 2022-03-project-example Public

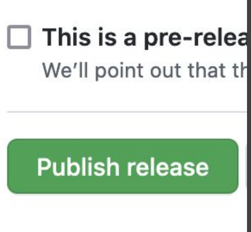
malvikasharan Create LICENSE.md



There aren't any releases here

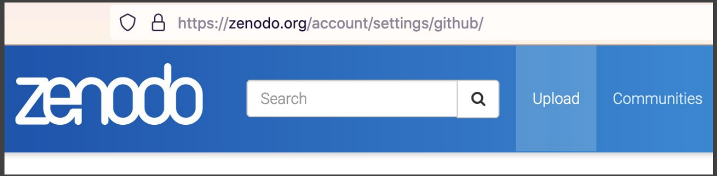
You can create a release to package software, along with release notes and links to binary files, for other people to use. Learn more about releases in our docs.

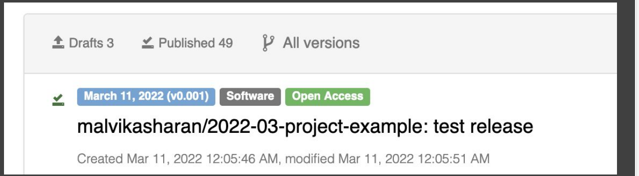



□ This is a pre-relea

We'll point out that th

Zenodo

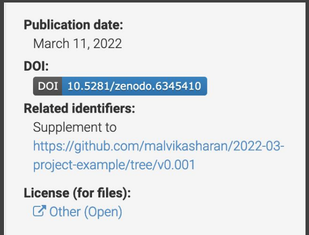

https://zenodo.org/account/settings/github/



March 11, 2022 (v0.001) Software Open Access

malvikasharan/2022-03-project-example: test release

Created Mar 11, 2022 12:05:46 AM, modified Mar 11, 2022 12:05:51 AM


Publication date:
March 11, 2022

DOI:
DOI 10.5281/zenodo.6345410

Related identifiers:
Supplement to
https://github.com/malvikasharan/2022-03-project-example/tree/v0.001

License (for files):
Other (Open)

Citable DOI

https://guides.github.com/activities/citable-code/, @turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

Upload to free openly accessible online repositories

Open Science Framework

figshare
credit for all your research

zenodo

The Dataverse® Project

GitHub

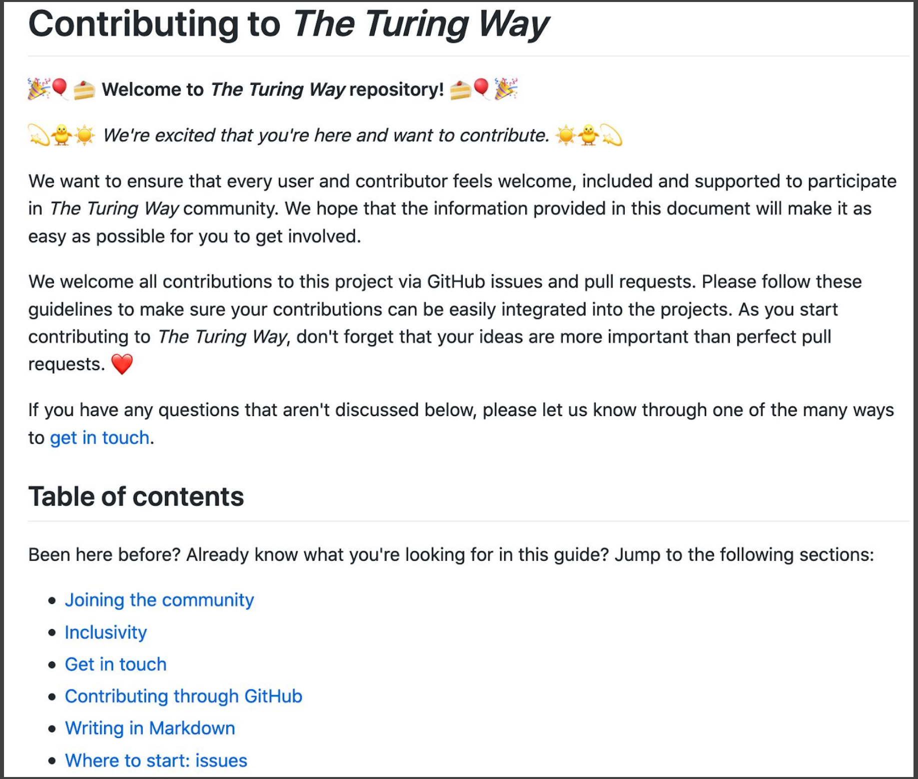
- Not a TDR
- cannot ensure long-term access

✓

- Trustworthy digital repository (TDR)
- Allows to generate shareable DOI - Digital Object Identifier

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

Contribution Guideline



Contributing to *The Turing Way*

🎉🎈🍰 **Welcome to *The Turing Way* repository!** 🍰🎈🎉

💫🐥☀️ *We're excited that you're here and want to contribute.* ☀️🐥💫

We want to ensure that every user and contributor feels welcome, included and supported to participate in *The Turing Way* community. We hope that the information provided in this document will make it as easy as possible for you to get involved.

We welcome all contributions to this project via GitHub issues and pull requests. Please follow these guidelines to make sure your contributions can be easily integrated into the projects. As you start contributing to *The Turing Way*, don't forget that your ideas are more important than perfect pull requests. ❤️

If you have any questions that aren't discussed below, please let us know through one of the many ways to get in touch.

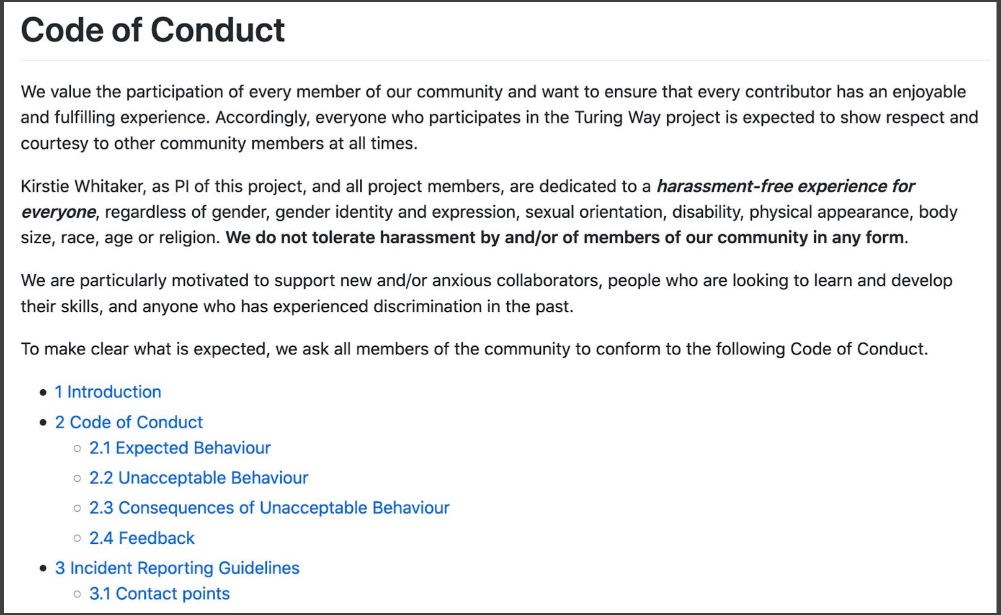
Table of contents

Been here before? Already know what you're looking for in this guide? Jump to the following sections:

CONTRIBUTING.md

- how to file a bug report
- how to suggest a feature
- how to contribute changes
- roadmap & project vision
- how contributors should ask for help and guidance

Code of Conduct



Code of Conduct

We value the participation of every member of our community and want to ensure that every contributor has an enjoyable and fulfilling experience. Accordingly, everyone who participates in the Turing Way project is expected to show respect and courtesy to other community members at all times.

Kirstie Whitaker, as PI of this project, and all project members, are dedicated to a ***harassment-free experience for everyone***, regardless of gender, gender identity and expression, sexual orientation, disability, physical appearance, body size, race, age or religion. **We do not tolerate harassment by and/or of members of our community in any form**.

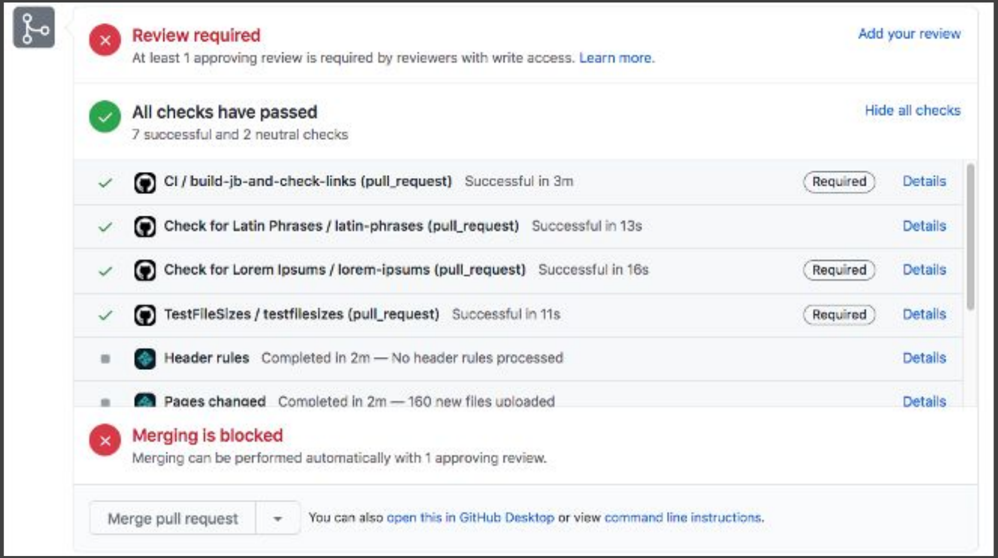
We are particularly motivated to support new and/or anxious collaborators, people who are looking to learn and develop their skills, and anyone who has experienced discrimination in the past.

To make clear what is expected, we ask all members of the community to conform to the following Code of Conduct.

CODE_OF_CONDUCT.md

- What is expected of the contributors
- What culture do we want to promote
- What if something should be reported

Continuous Integration



Review required Add your review

At least 1 approving review is required by reviewers with write access. Learn more.

All checks have passed Hide all checks

7 successful and 2 neutral checks

- ✓ CI / build-jb-and-check-links (pull_request) Successful in 3m Required Details
- ✓ Check for Latin Phrases / latin-phrases (pull_request) Successful in 13s Details
- ✓ Check for Lorem Ipsums / lorem-ipsums (pull_request) Successful in 16s Required Details
- ✓ TestFileSizes / testfilesizes (pull_request) Successful in 11s Required Details
- ■ Header rules Completed in 2m — No header rules processed Details
- ■ Pages changed Completed in 2m — 160 new files uploaded Details

Merging is blocked

Merging can be performed automatically with 1 approving review.

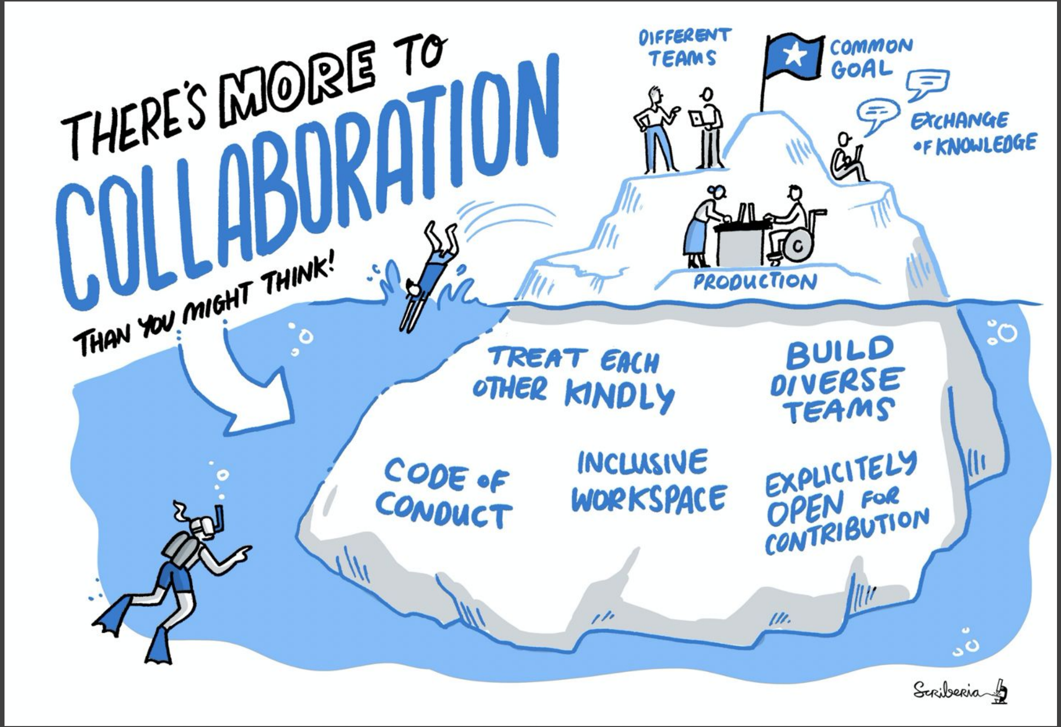
Merge pull request ▾ You can also open this in GitHub Desktop or view command line instructions.

Practice of integrating changes to a project made by individuals into a main, shared version

-- frequently

All these aspects enhance collaboration in your project!

How to make your code easy to test?

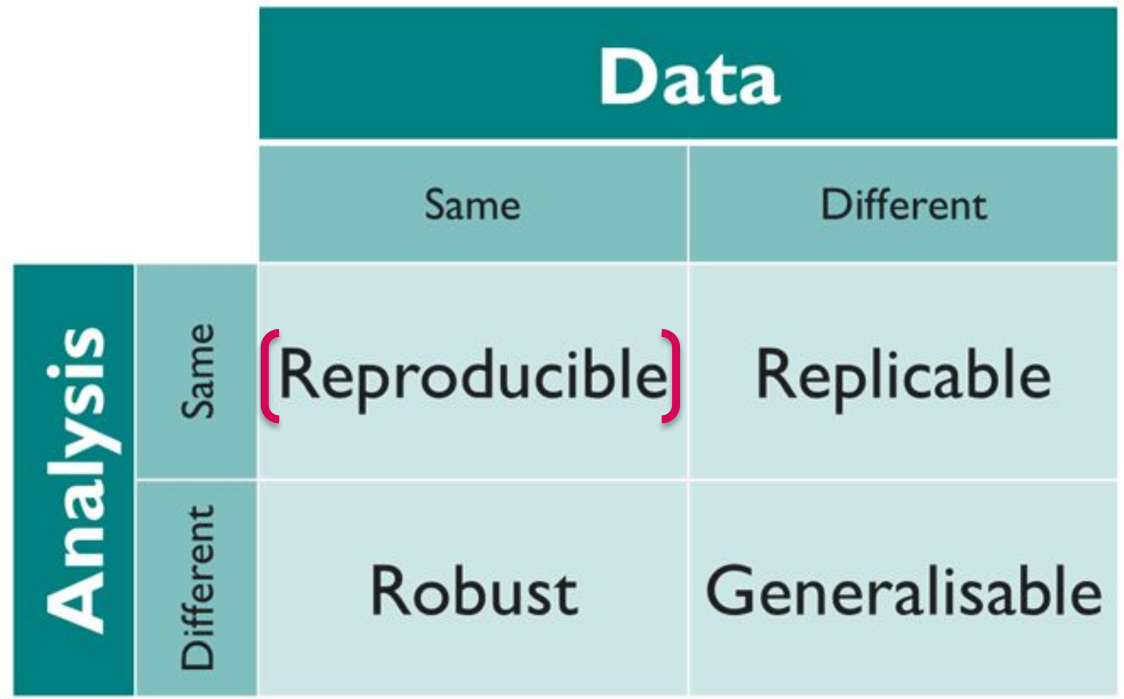


@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

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

https://the-turing-way.netlify.app/reproducible-research/overview/overview-definitions.html

“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

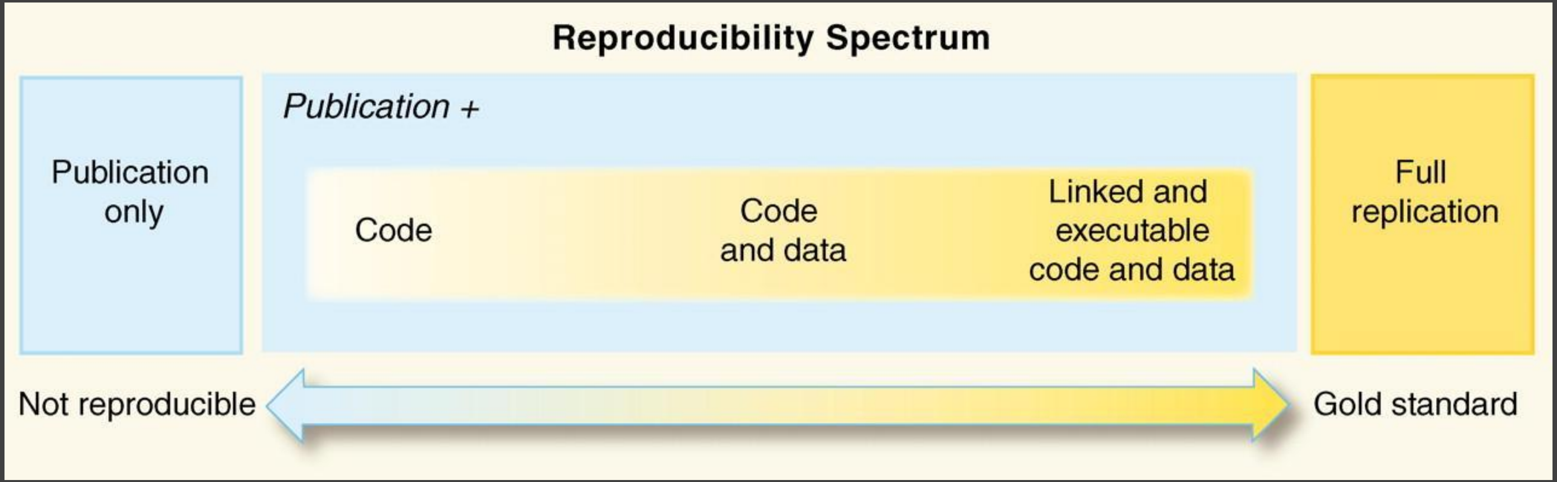
(slide courtesy of Chris Holdgraf and the Jupyter Team)

Take home message

sharing your code and data isn't enough

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

You need the computational environment too



Peng, 2011, doi: 10.1126/science.1213847

You need the computational environment too

Peng, 2011, doi: 10.1126/science.1213847

What is a computational environment?

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

https://the-turing-way.netlify.app/reproducible-research/renv.html

What is a computational environment?

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

their versions and their configuration

https://the-turing-way.netlify.app/reproducible-research/renv.html

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

What is a computational environment?

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

their versions and their configuration

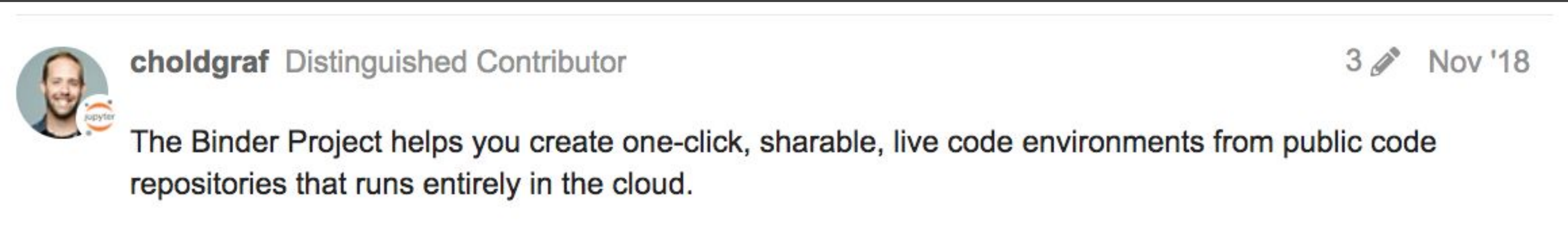
and their interaction

https://the-turing-way.netlify.app/reproducible-research/renv.html

What is ***Binder***?

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

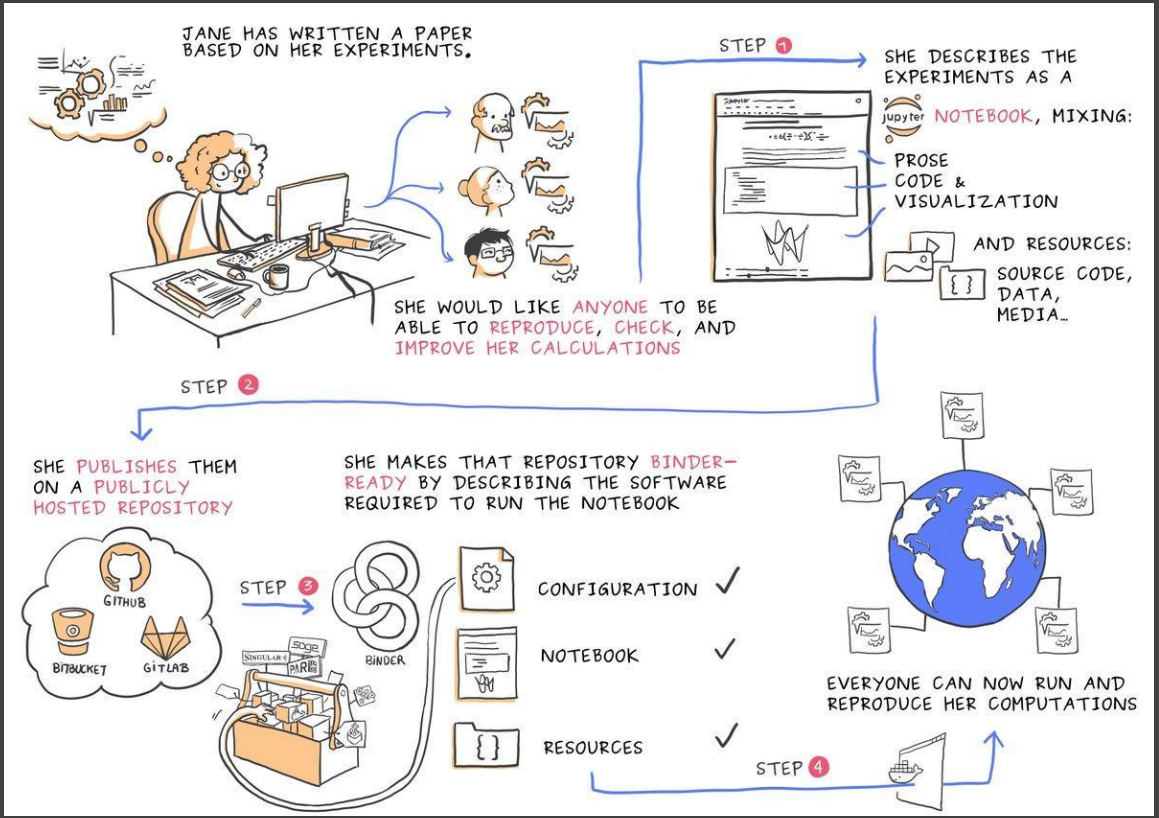
What is Binder?



choldgraf Distinguished Contributor 3 Nov '18

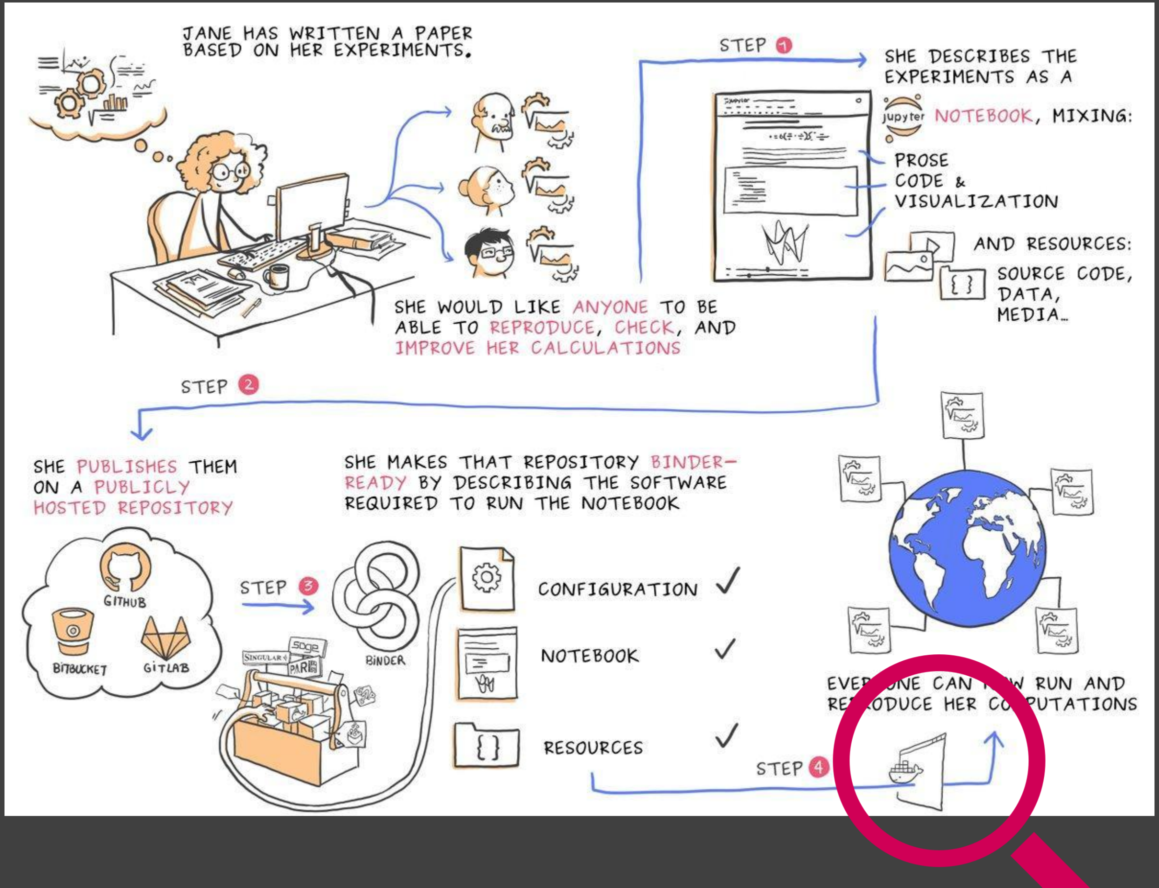
The Binder Project helps you create one-click, sharable, live code environments from public code repositories that runs entirely in the cloud.

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


JANE HAS WRITTEN A PAPER BASED ON HER EXPERIMENTS.
SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS
STEP 1
SHE DESCRIBES THE EXPERIMENTS AS A
jupyter NOTEBOOK, MIXING:
PROSE
CODE &
VISUALIZATION
AND RESOURCES:
SOURCE CODE,
DATA,
MEDIA...
STEP 2
SHE PUBLISHES THEM ON A PUBLICLY HOSTED REPOSITORY
GITHUB
BITBUCKET
GITLAB
STEP 3
SHE MAKES THAT REPOSITORY BINDER-READY BY DESCRIBING THE SOFTWARE REQUIRED TO RUN THE NOTEBOOK
BINDER
SINGULAR
sage
PARI
CONFIGURATION ✓
NOTEBOOK ✓
RESOURCES ✓
STEP 4
EVERYONE CAN NOW RUN AND REPRODUCE HER COMPUTATIONS

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

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

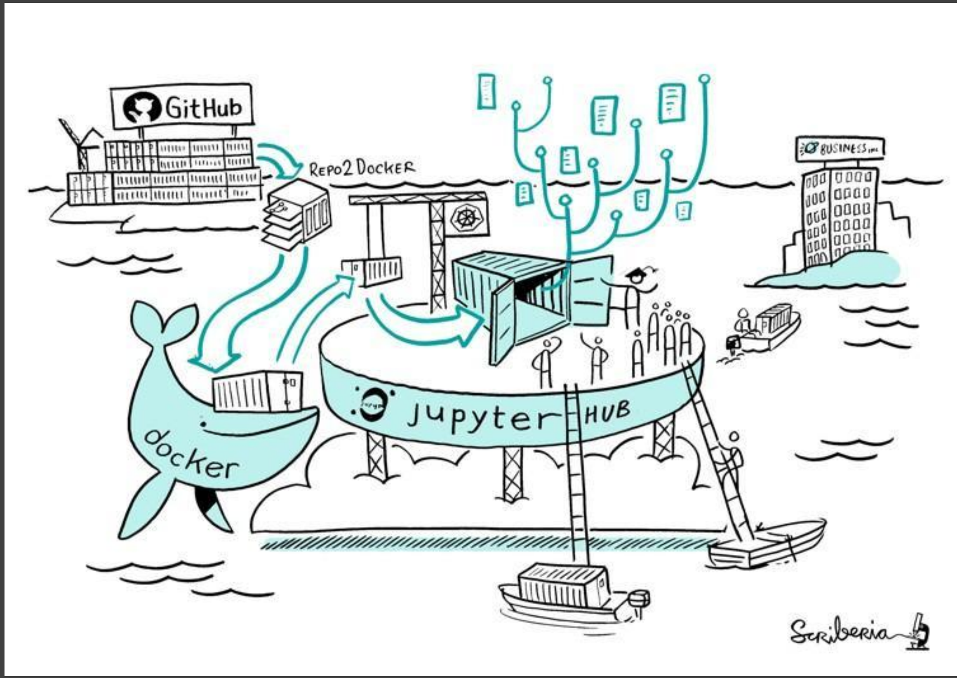

JANE HAS WRITTEN A PAPER BASED ON HER EXPERIMENTS.
SHE WOULD LIKE ANYONE TO BE ABLE TO REPRODUCE, CHECK, AND IMPROVE HER CALCULATIONS
STEP 1
SHE DESCRIBES THE EXPERIMENTS AS A
jupyter NOTEBOOK, MIXING:
PROSE
CODE &
VISUALIZATION
AND RESOURCES:
SOURCE CODE,
DATA,
MEDIA…
STEP 2
SHE PUBLISHES THEM ON A PUBLICLY HOSTED REPOSITORY
GITHUB
BITBUCKET
GITLAB
STEP 3
SHE MAKES THAT REPOSITORY BINDER-READY BY DESCRIBING THE SOFTWARE REQUIRED TO RUN THE NOTEBOOK
BINDER
SINGULAR
sage
PARI
CONFIGURATION ✓
NOTEBOOK ✓
RESOURCES ✓
STEP 4
EVERYONE CAN NOW RUN AND REPRODUCE HER COMPUTATIONS

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

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

BinderHub

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

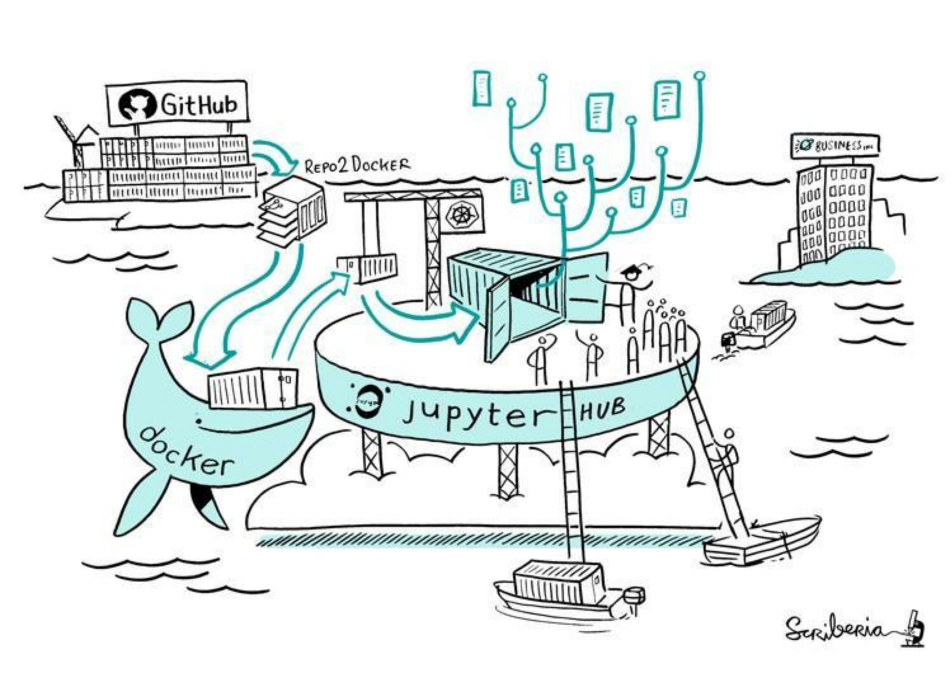


https://the-turing-way.netlify.app/reproducible-research/binderhub/binderhub-compute.html

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

BinderHub

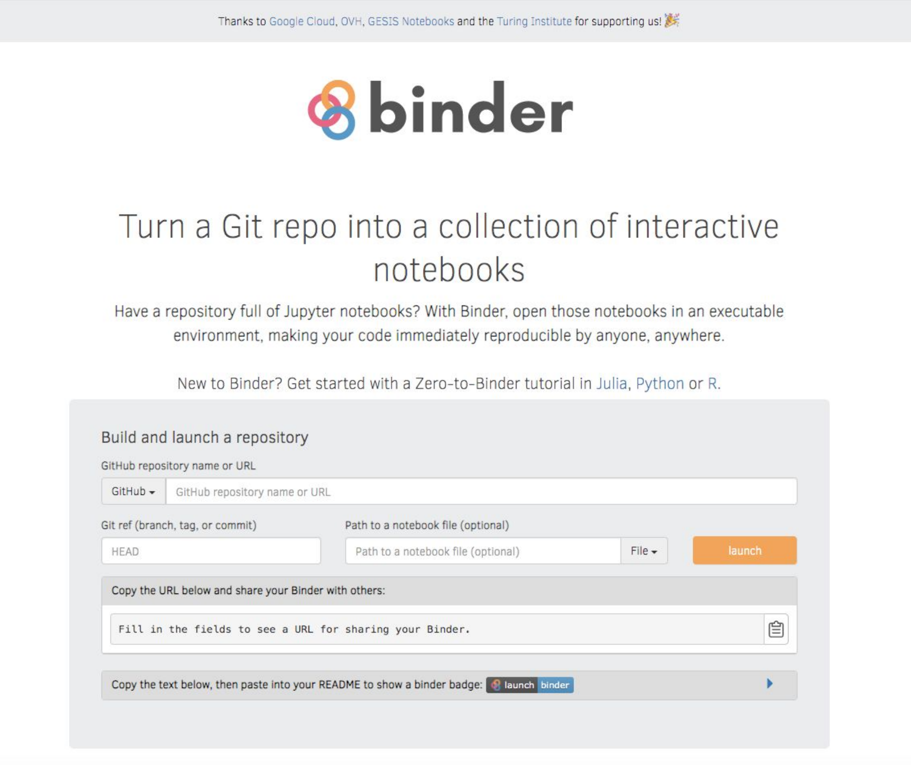
- repo2docker
- Kubernetes
- Helm
- JupyterHub
- a cloud service platform



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

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! 🎉

binder

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

Git ref (branch, tag, or commit)

Path to a notebook file (optional)

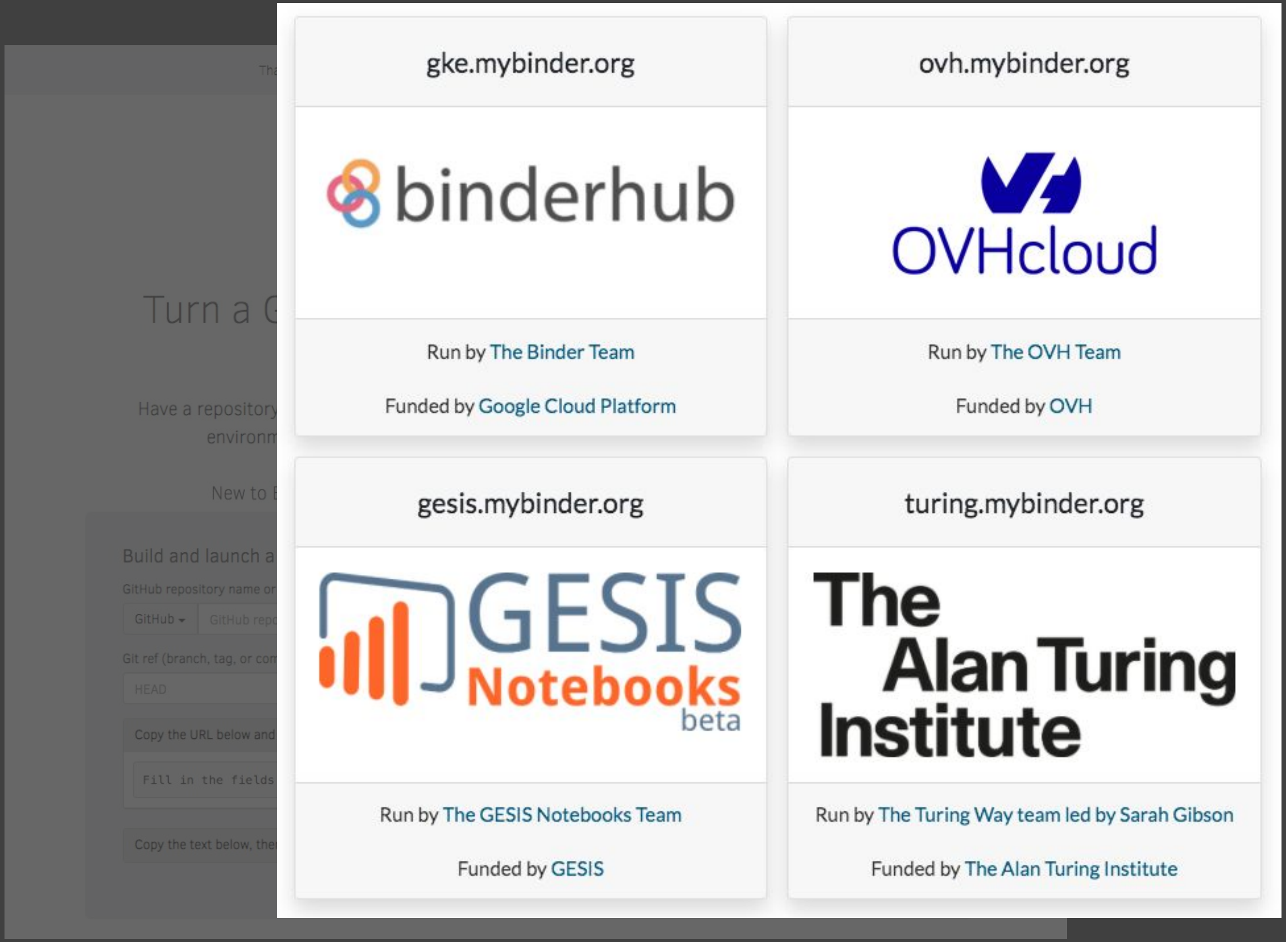
Copy the URL below and share your Binder with others:

```
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: launch binder

https://mybinder.org/

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555



gke.mybinder.org

Run by The Binder Team

Funded by Google Cloud Platform

ovh.mybinder.org

Run by The OVH Team

Funded by OVH

gesis.mybinder.org

Run by The GESIS Notebooks Team

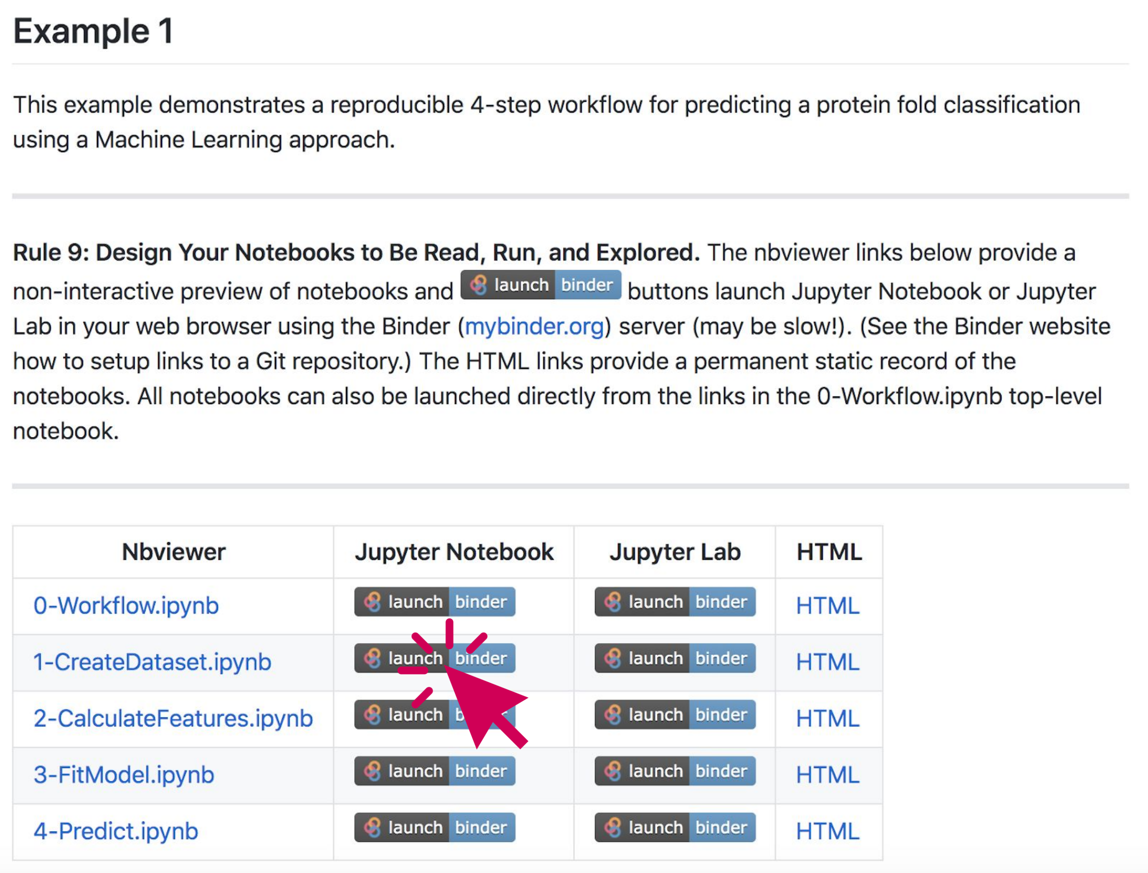
Funded by GESIS

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



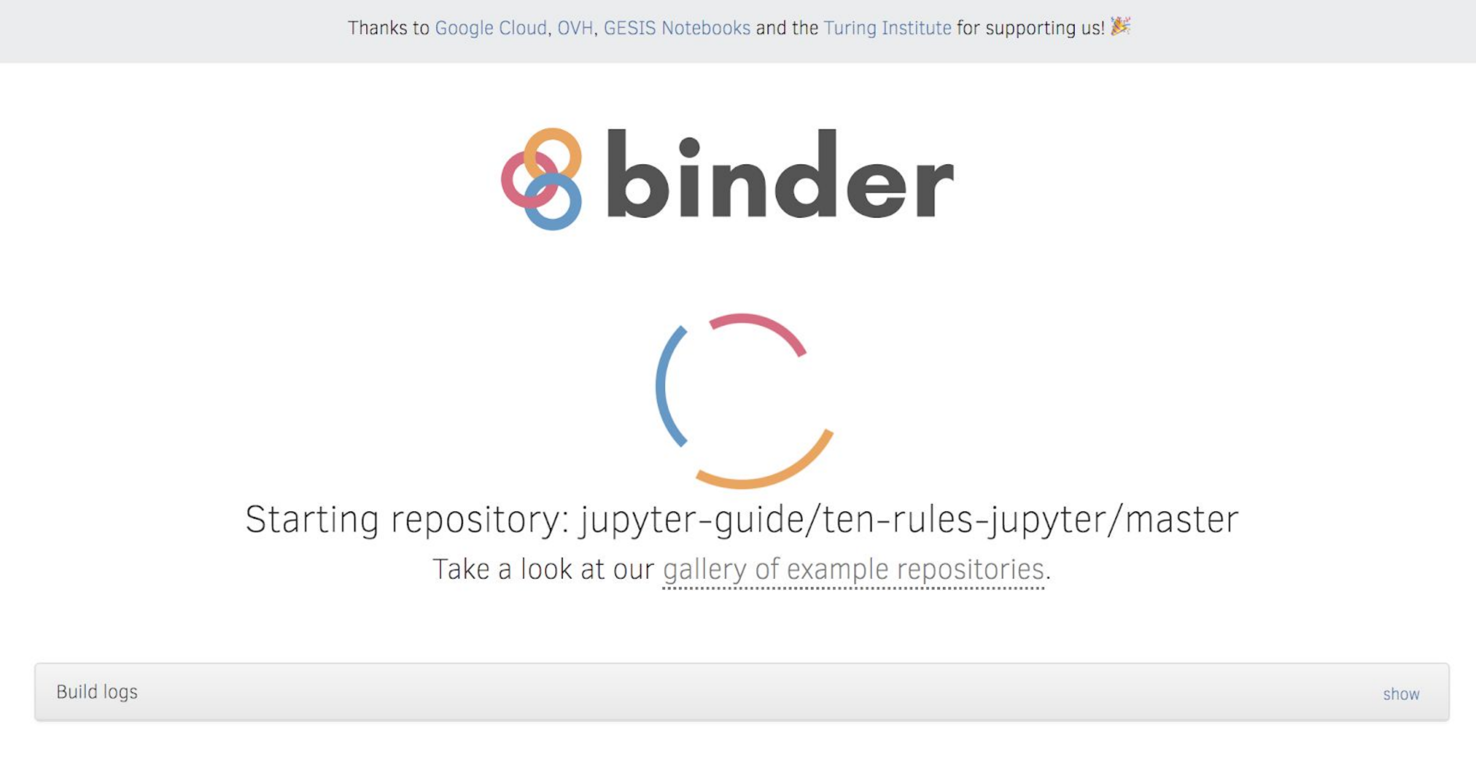
Example 1

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 launch binder 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	launch binder	launch binder	HTML
1-CreateDataset.ipynb	launch binder	launch binder	HTML
2-CalculateFeatures.ipynb	launch binder	launch binder	HTML
3-FitModel.ipynb	launch binder	launch binder	HTML
4-Predict.ipynb	launch binder	launch binder	HTML

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



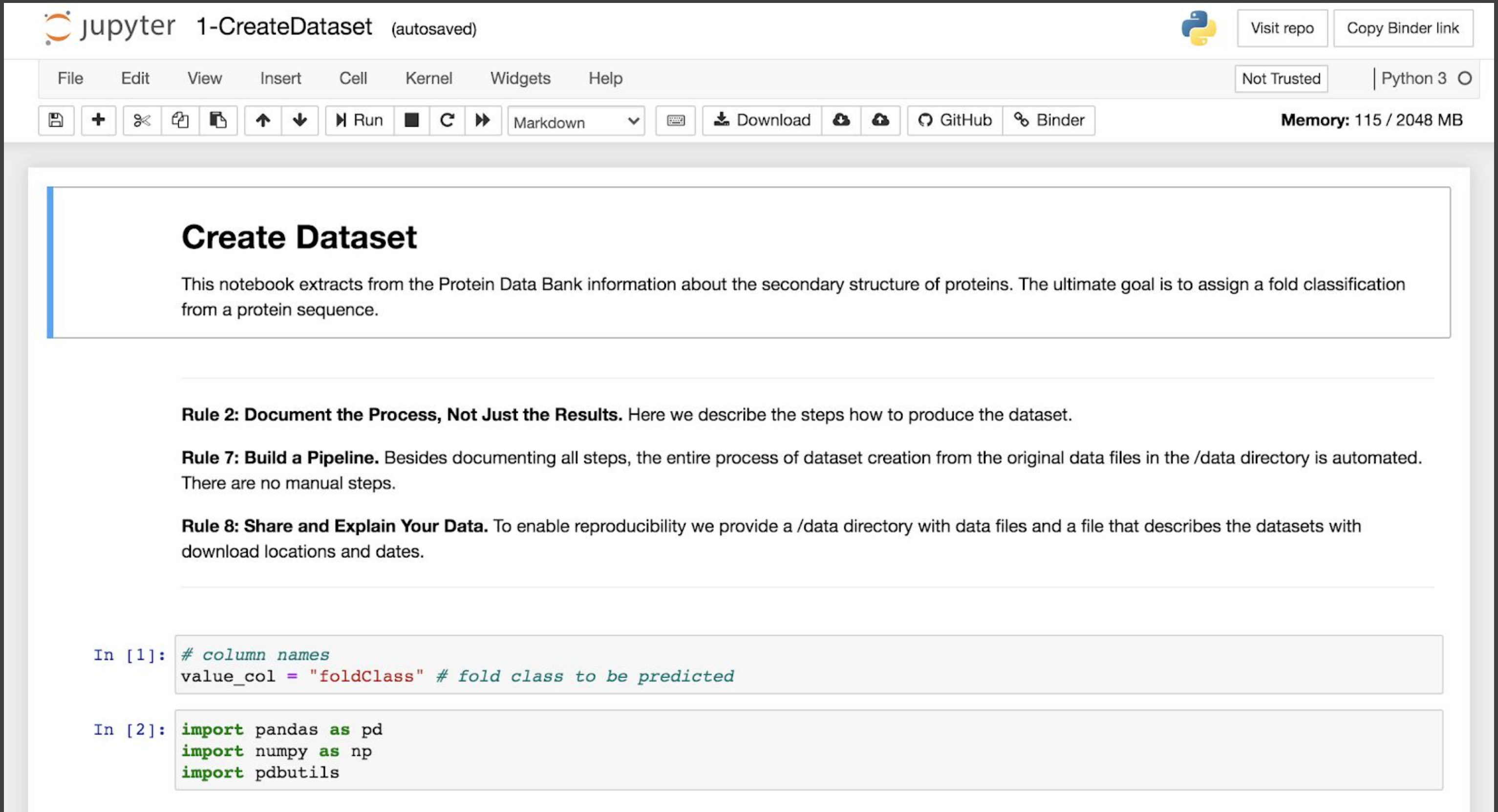
Thanks to Google Cloud, OVH, GESIS Notebooks and the Turing Institute for supporting us! 🎉

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

Take a look at our gallery of example repositories.

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

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555



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]:

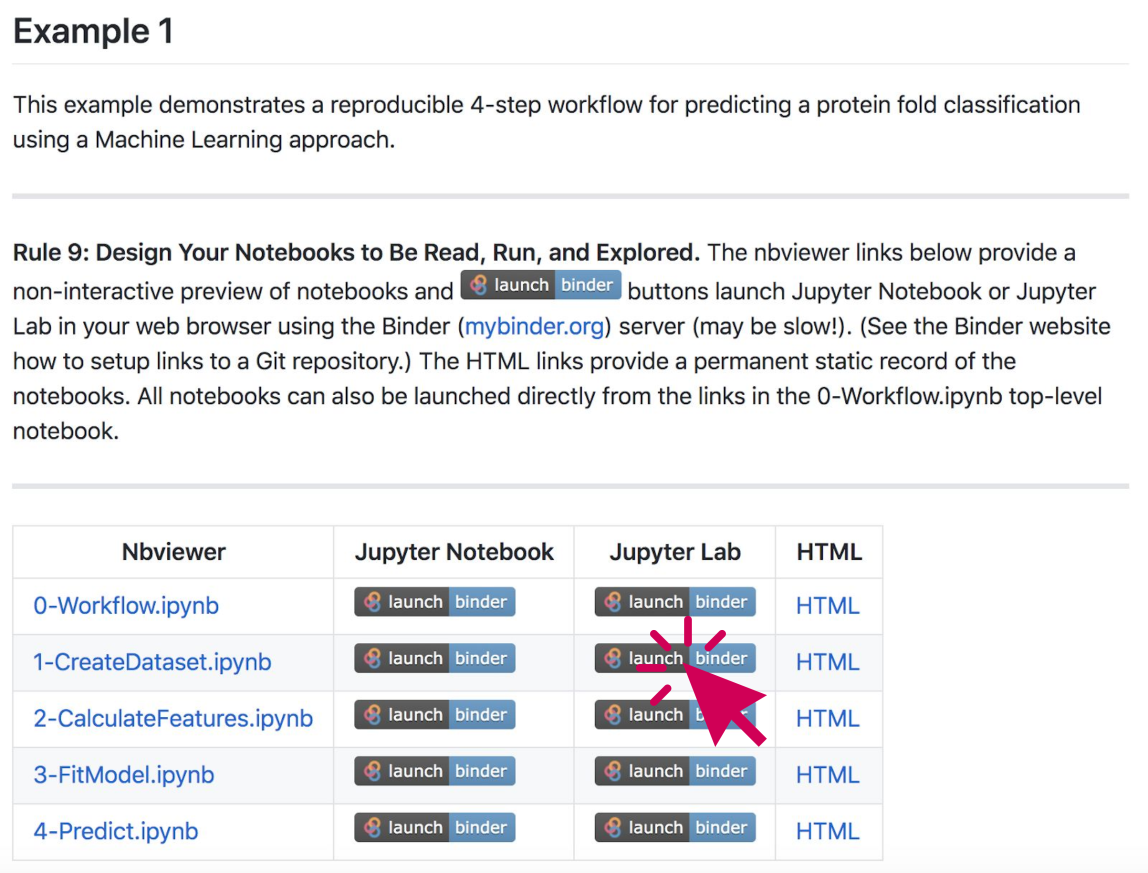
```
# column names
value_col = "foldClass" # fold class to be predicted
```

In [2]:

```
import pandas as pd
import numpy as np
import pdbutils
```

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

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555



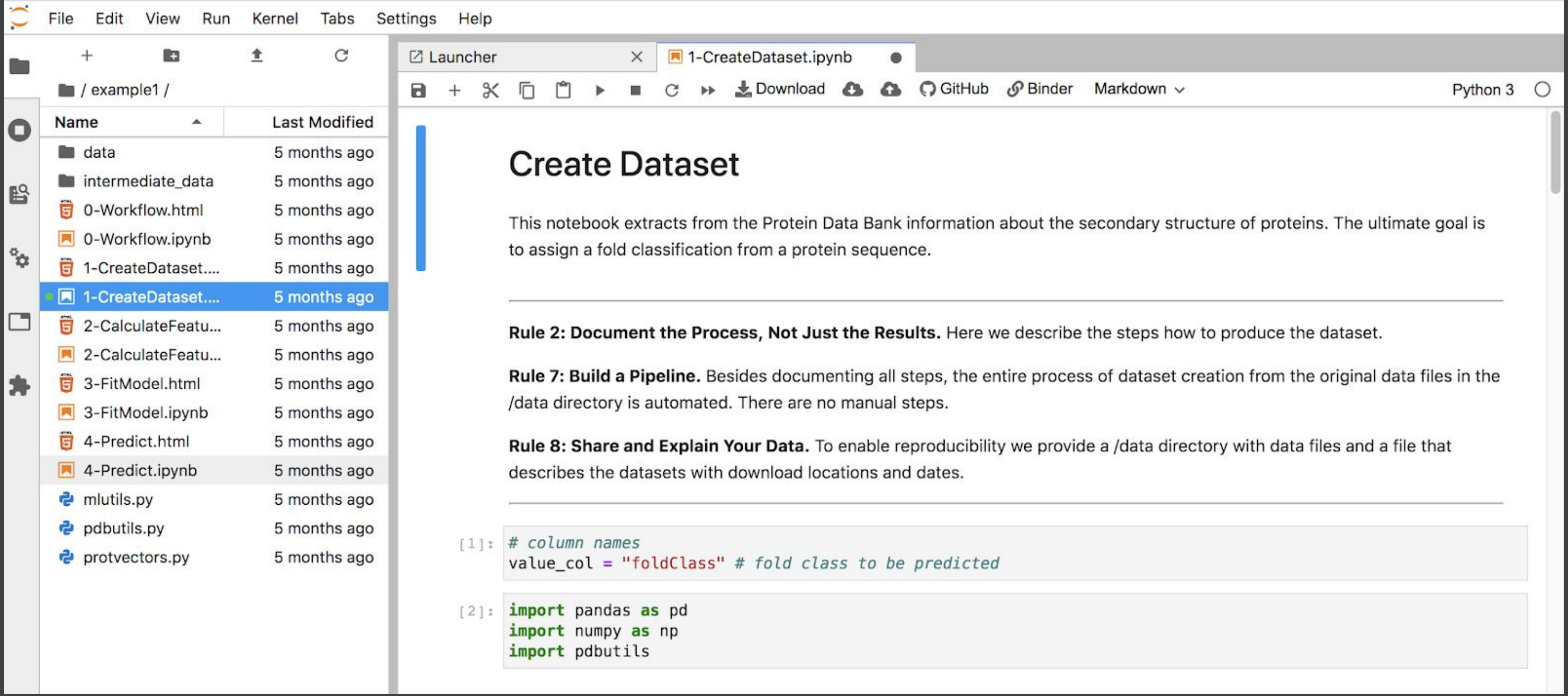
Example 1

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 launch binder 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	launch binder	launch binder	HTML
1-CreateDataset.ipynb	launch binder	launch binder	HTML
2-CalculateFeatures.ipynb	launch binder	launch binder	HTML
3-FitModel.ipynb	launch binder	launch binder	HTML
4-Predict.ipynb	launch binder	launch binder	HTML

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



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
protvectors.py	5 months ago

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
```

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

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

Kirstie Whitaker

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

@kirstie_j

@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

Hands-on session

Zero to Binder, a guided tour of building a Binder resource:

* Link to the full Tutorial: http://bit.ly/zero-to-binder-python

Preparing to Launch your First Repo

Zero to Binder, a guided tour of building a Binder resource:

* Link to the full Tutorial: http://bit.ly/zero-to-binder-python

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)

https://mybinder.readthedocs.io/en/latest/introduction.html

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

https://mybinder.readthedocs.io/en/latest/introduction.html

Launch your First Repo

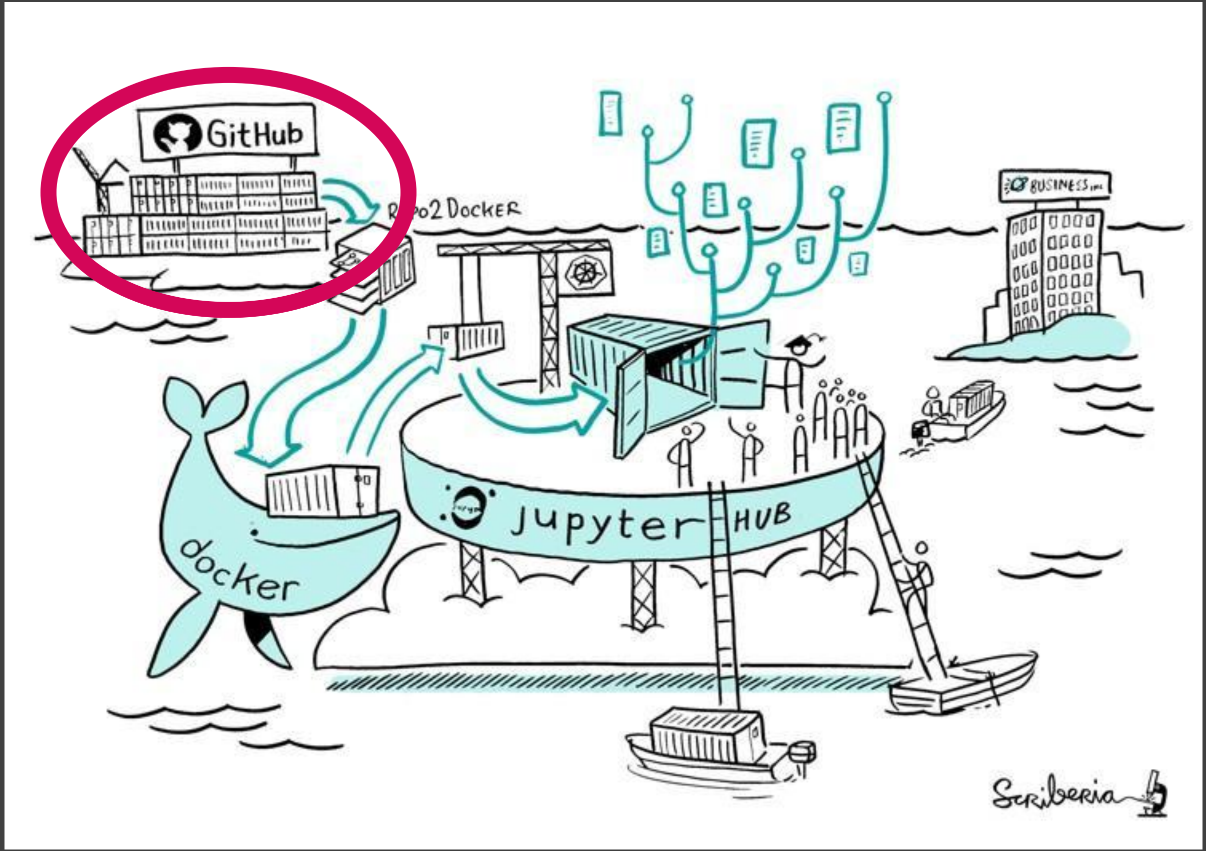
1. Go to https://mybinder.org
2. Type the URL of your repo into the "GitHub repo or URL" box. It should look like this: https://github.com/your-username/my-first-binder
3. As you type, the webpage generates a link in the "Copy the URL below..." box It should look like this: https://mybinder.org/v2/gh/your-username/my-first-binder/HEAD
4. Copy it, open a new browser tab and visit that URL.
    - You will see a "spinner" as Binder launches the repo.

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

https://mybinder.readthedocs.io/en/latest/introduction.html

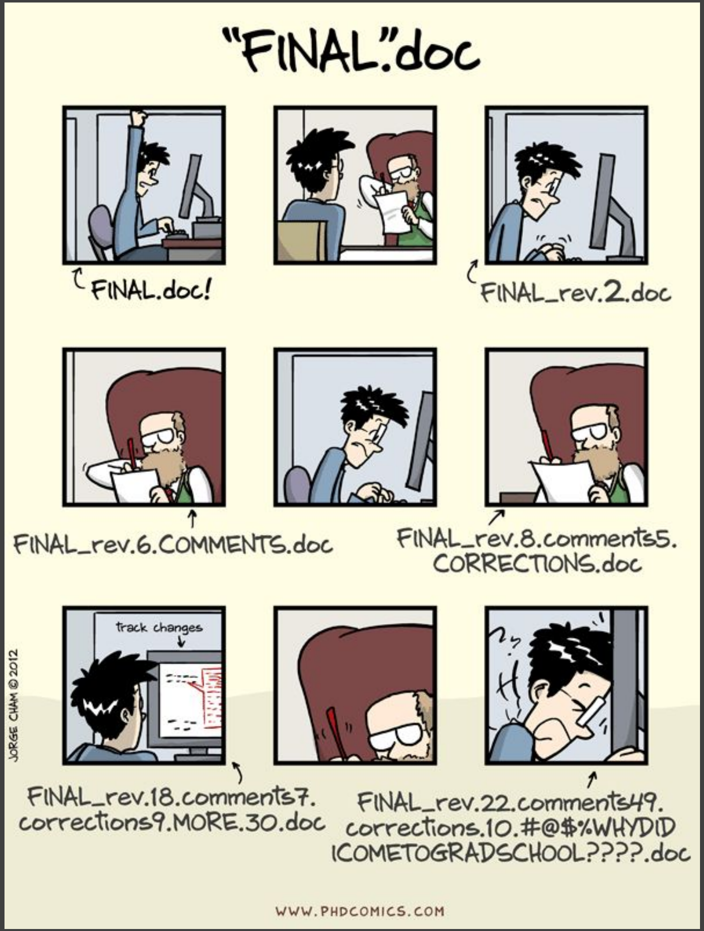
How does ***Binder*** work?



@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

git

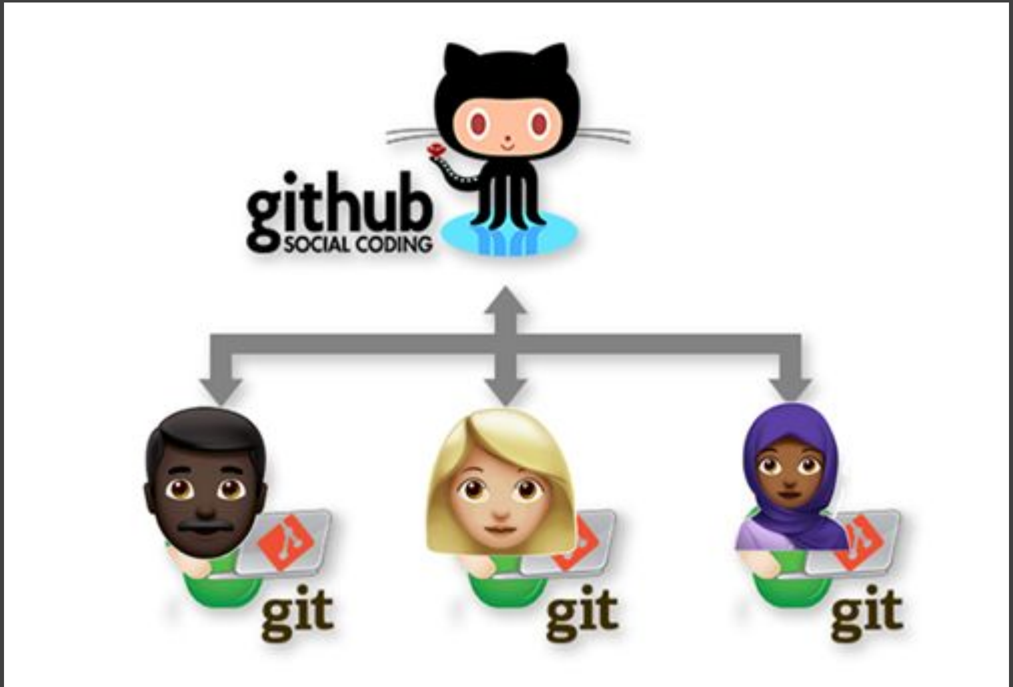
- version control system
- records changes to a file or set of files over time
- provides access to any specific version



http://phdcomics.com/comics/archive.php?comicid=1531

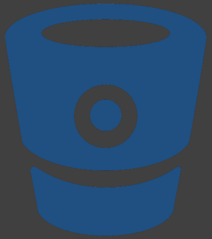
GitHub

- online hosting platform for git repositories
- enables collaborative work



Preparing a repository for Binder

1. The repository is in a public location online



https://mybinder.readthedocs.io/en/latest/introduction.html

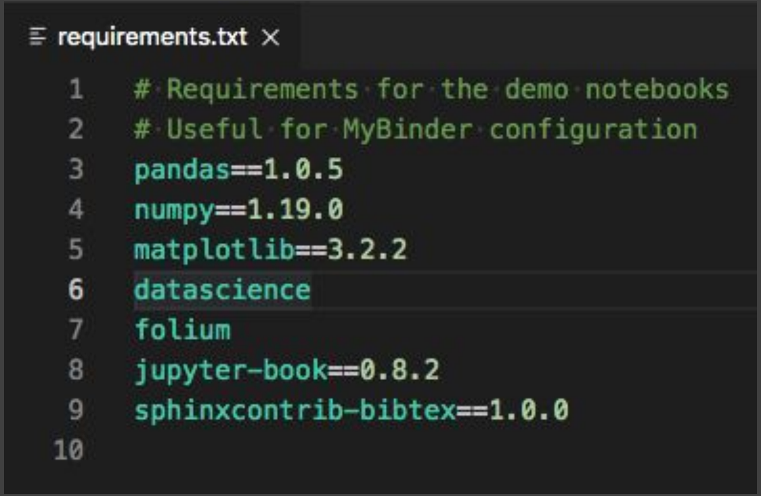
Preparing a repository for Binder

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

https://mybinder.readthedocs.io/en/latest/introduction.html

Configuration file

- defines your computational environment

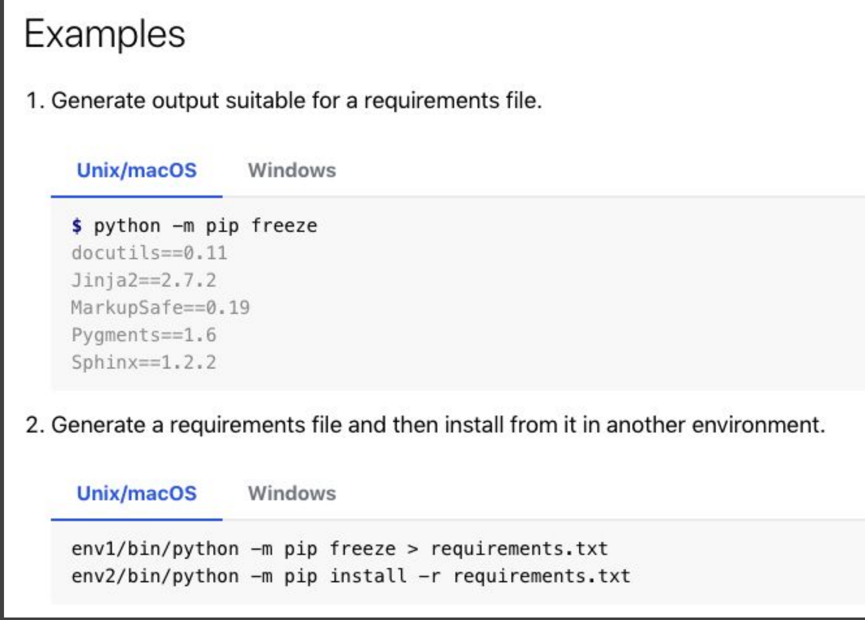


```
# Requirements for the demo notebooks
# Useful for MyBinder configuration
pandas==1.0.5
numpy==1.19.0
matplotlib==3.2.2
datascience
folium
jupyter-book==0.8.2
sphinxcontrib-bibtex==1.0.0
```

https://mybinder.readthedocs.io/en/latest/using/config_files.htmll

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.

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

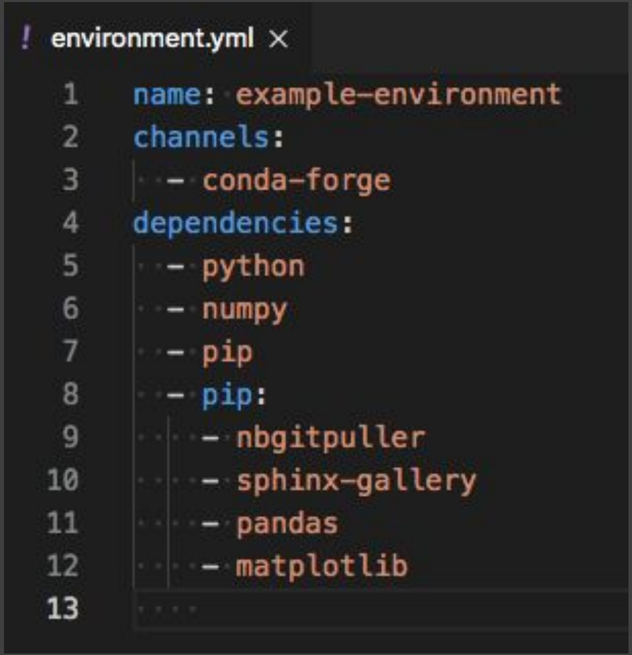
2. Generate a requirements file and then install from it in another environment.

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

https://pip.pypa.io/en/stable/reference/pip_freeze/

CONDA

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

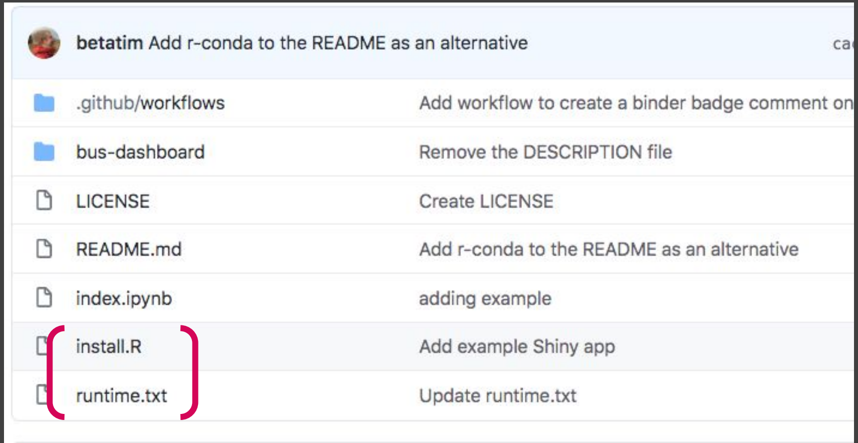


```yaml
name: example-environment
channels:
  - conda-forge
dependencies:
  - python
  - numpy
  - pip
  - pip:
    - nbgitpuller
    - sphinx-gallery
    - pandas
    - matplotlib
    
```

https://github.com/binder-examples/python-conda_pip/blob/master/environment.yml

R environment

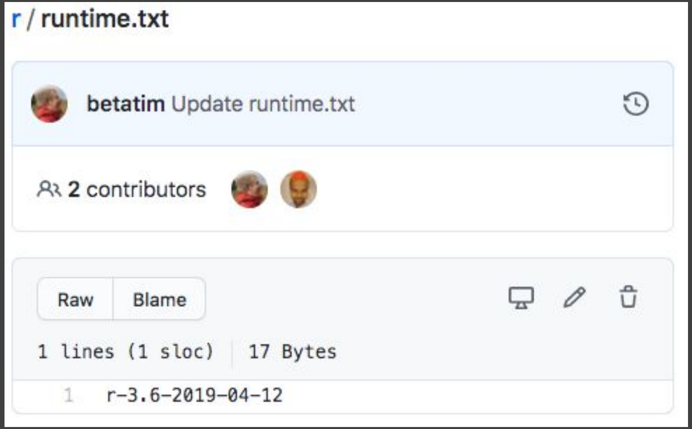
- support for R and RStudio with libraries pinned to a specific snapshot on MRAN, defined in runtime.txt
- install.R specifies one library to install per line



betatim Add r-conda to the README as an alternative ca

.github/workflows	Add workflow to create a binder badge comment on
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

https://mybinder.readthedocs.io/en/latest/examples/sample_repos.html#specifying-an-r-environment-with-a-runtime-txt-file



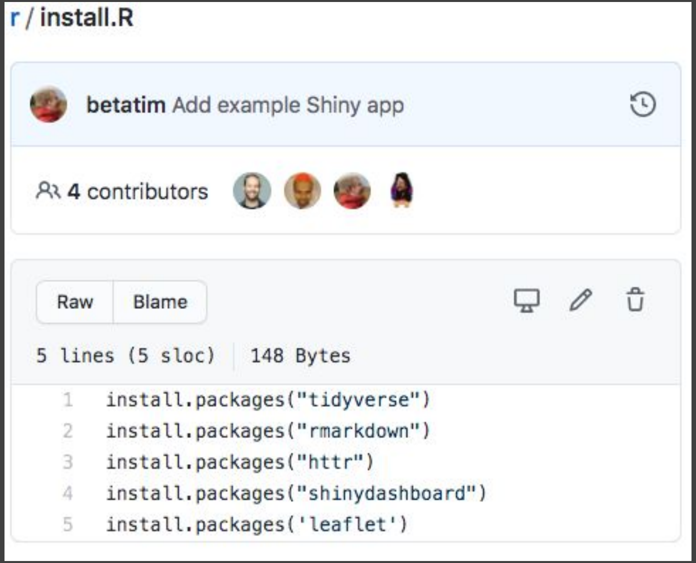
r / runtime.txt

betatim Update runtime.txt

2 contributors

1 lines (1 sloc) | 17 Bytes

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



r / install.R

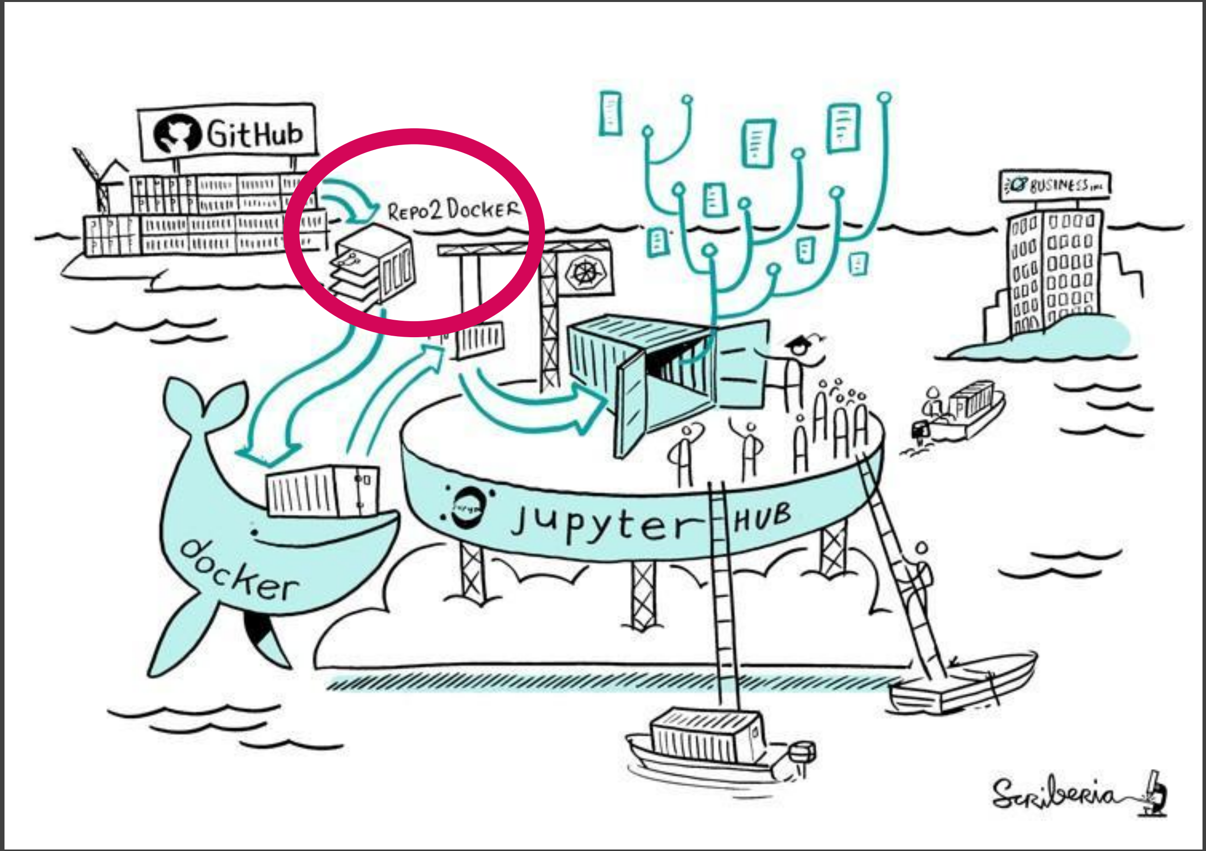
betatim Add example Shiny app

4 contributors

5 lines (5 sloc) | 148 Bytes

```
install.packages("tidyverse")
install.packages("rmarkdown")
install.packages("httr")
install.packages("shinydashboard")
install.packages('leaflet')
```

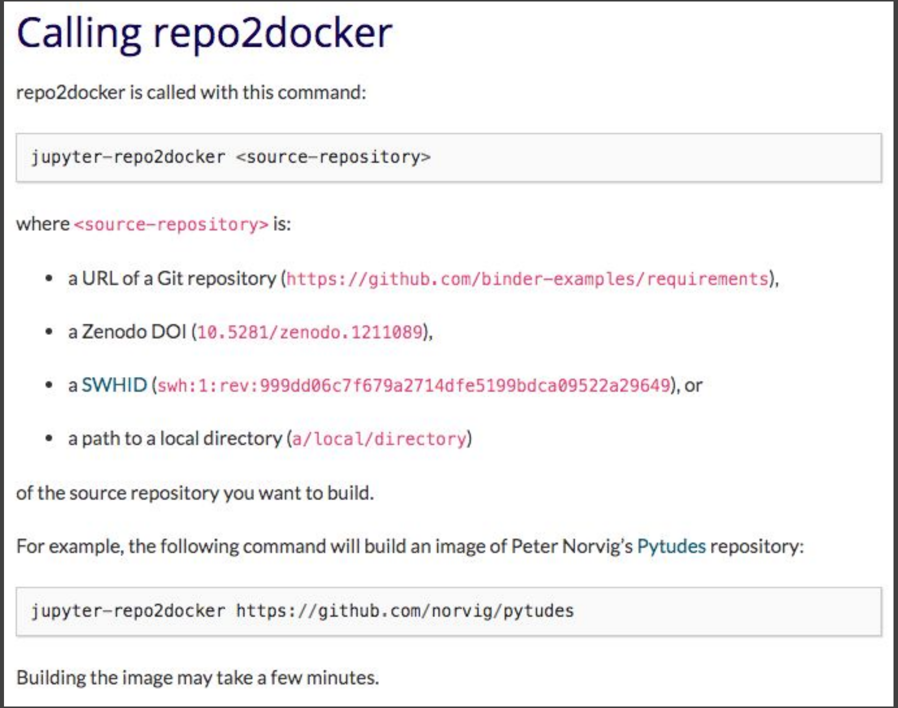
https://github.com/binder-examples/r/blob/master



@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

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`),
- a SWHID (`swh:1:rev: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.

https://repo2docker.readthedocs.io/en/latest/?badge=latest

Containers

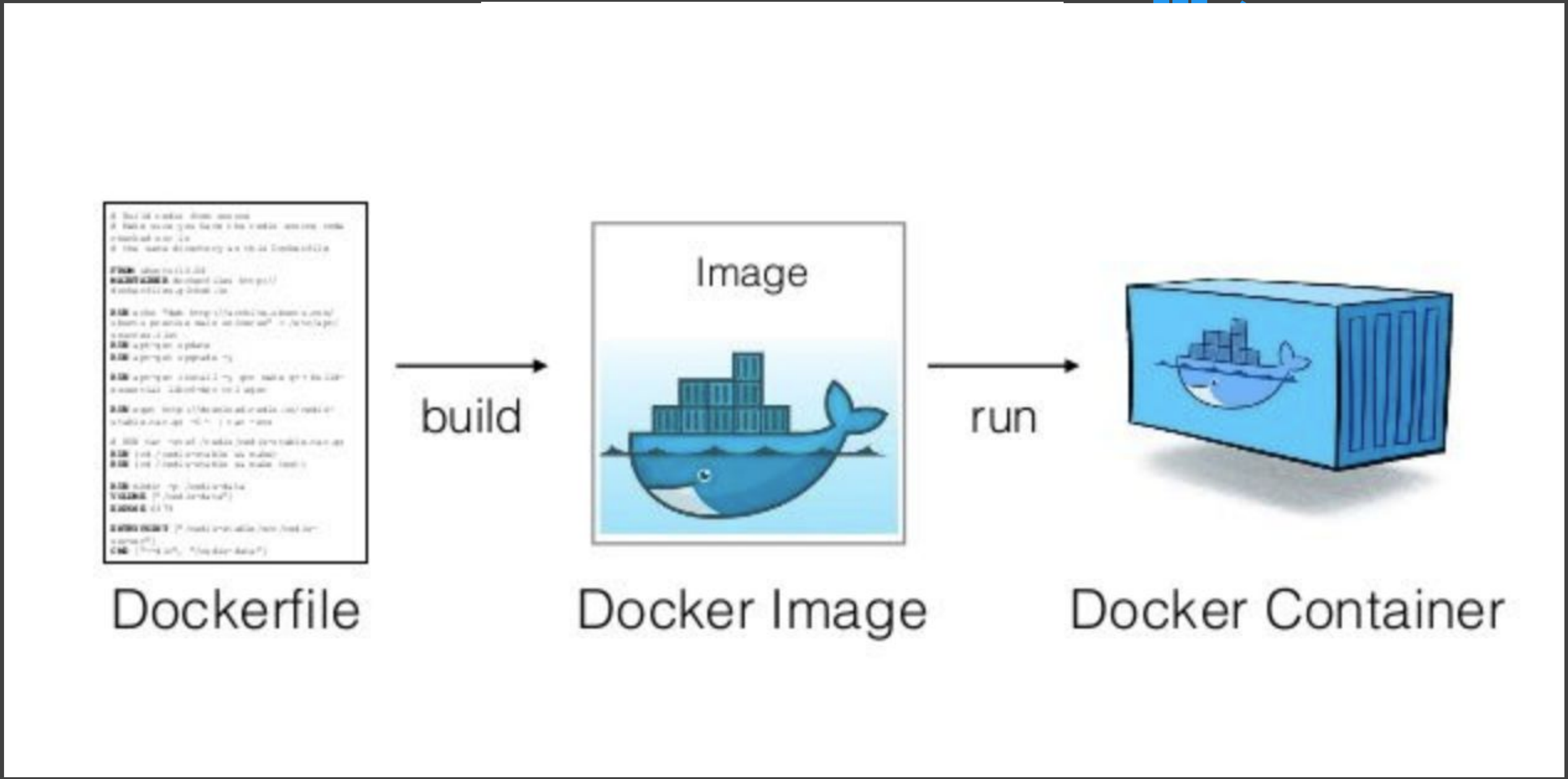
- package a project with all of the parts it needs - such as libraries, dependencies, and system settings
- anyone can then open up a container and work within it
- the computational environment of the container is identical across instances

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

Containers

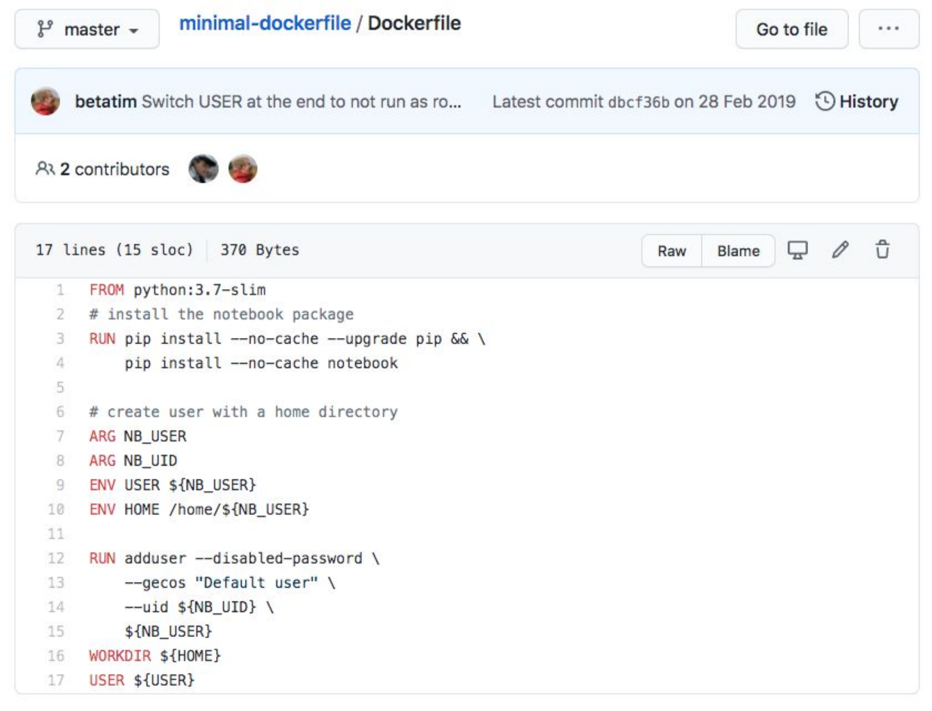
- behaves like a virtual machine
- more lightweight -> only contains the individual components needed to operate the project

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



https://medium.com/platformer-blog/practical-guide-on-writing-a-dockerfile-for-your-application-89376f88b3b5

Containers



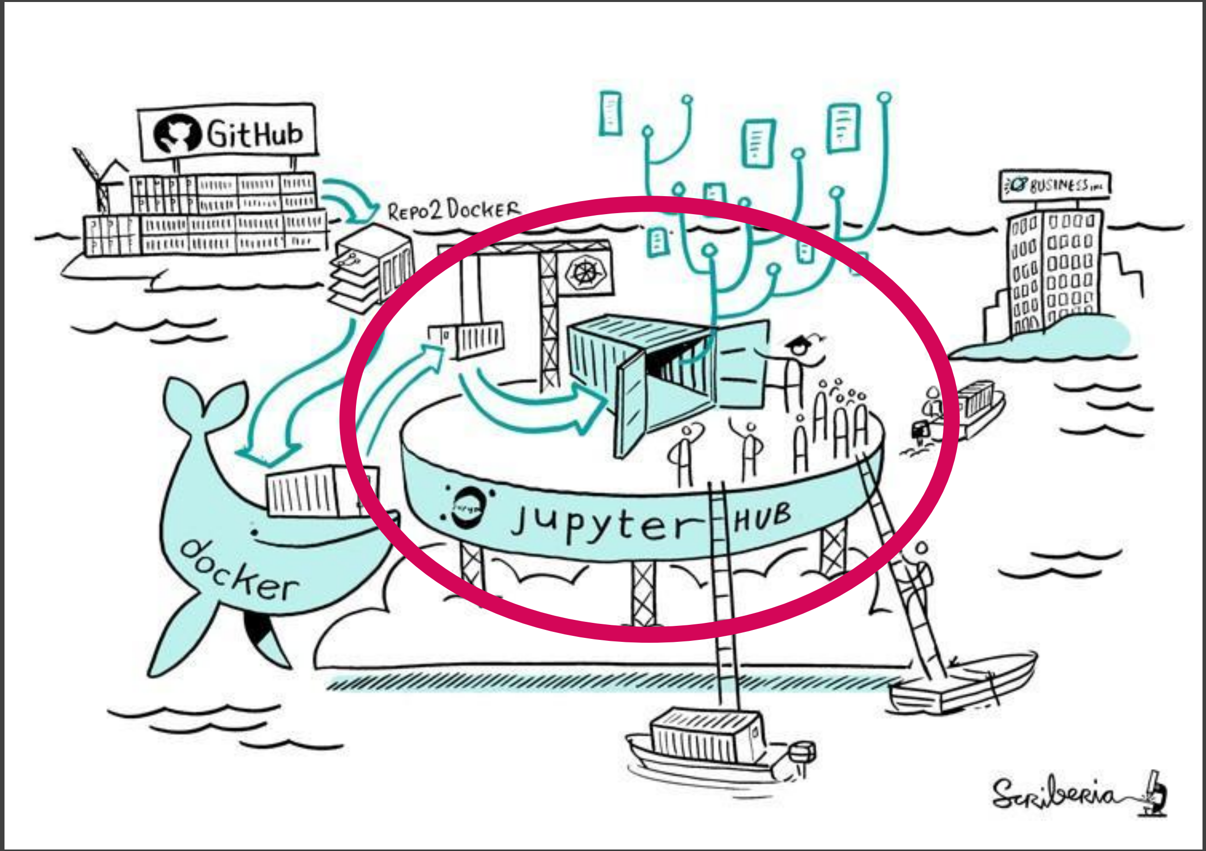
master
minimal-dockerfile / Dockerfile
Go to file
betatim Switch USER at the end to not run as ro...
Latest commit dbcf36b on 28 Feb 2019
History
2 contributors
17 lines (15 sloc) | 370 Bytes
Raw
Blame

```
FROM python:3.7-slim
# install the notebook package
RUN pip install --no-cache --upgrade pip && \
    pip install --no-cache notebook

# create user with a home directory
ARG NB_USER
ARG NB_UID
ENV USER ${NB_USER}
ENV HOME /home/${NB_USER}

RUN adduser --disabled-password \
    --gecos "Default user" \
    --uid ${NB_UID} \
    ${NB_USER}
WORKDIR ${HOME}
USER ${USER}
```

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



@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

JupyterHub

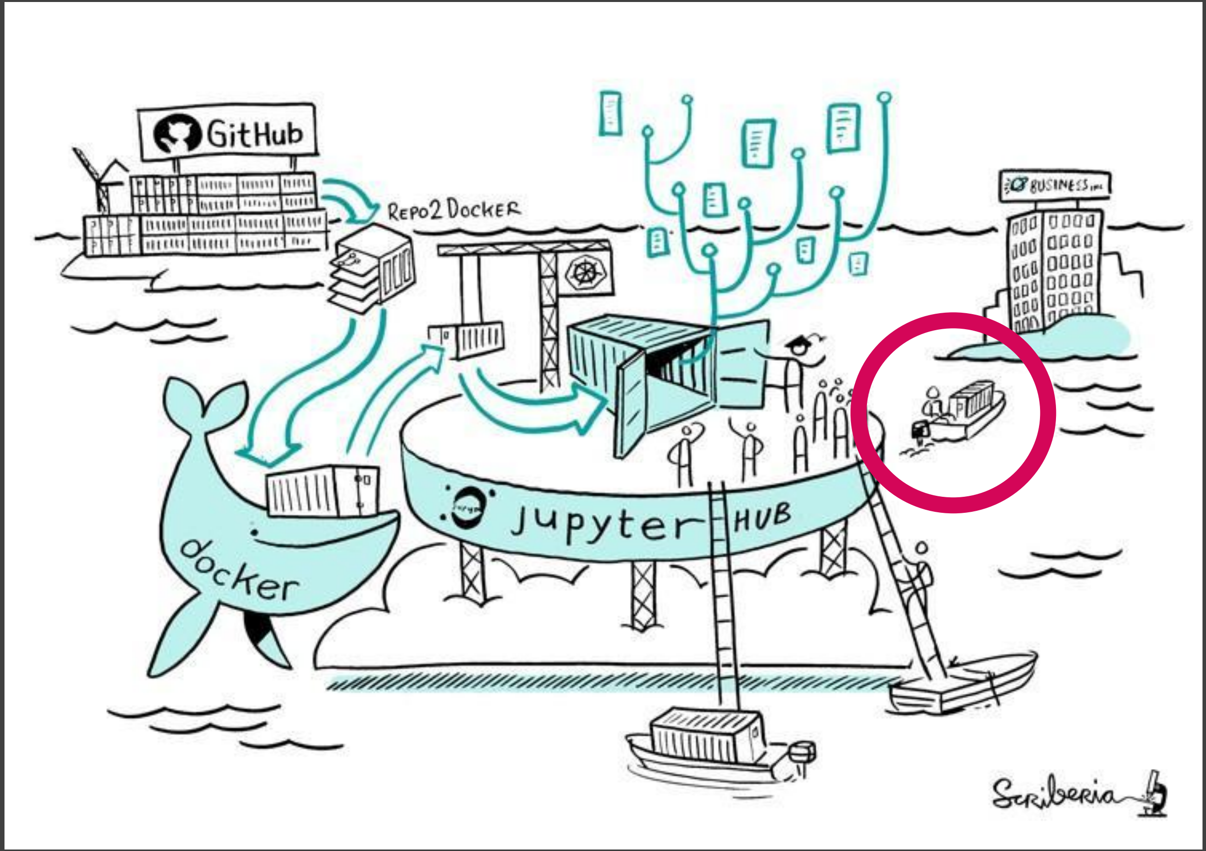
- allows users to interact with a computing environment through a webpage
- “the cloud is just someone else’s computer” @kirstie_j

https://zero-to-jupyterhub.readthedocs.io/en/stable/

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.

https://binderhub.readthedocs.io/en/latest/overview.html



@turingway, CC-BY 4.0, The Turing Way, DOI: 10.5281/zenodo.6346555

Launch your First Repo

1. Go to https://mybinder.org
2. Type the URL of your repo into the "GitHub repo or URL" box. It should look like this: https://github.com/your-username/my-first-binder
3. As you type, the webpage generates a link in the "Copy the URL below..." box It should look like this: https://mybinder.org/v2/gh/your-username/my-first-binder/HEAD
4. Copy it, open a new browser tab and visit that URL.
    - You will see a "spinner" as Binder launches the repo.

What happens in the background?

While you wait for the launch, BinderHub (the backend of Binder) is:

- Fetching your repo from GitHub
- Reading the contents
- Creating a Docker file based on your repo requirement
- Launching that Docker image in the Cloud
- Connecting you to it via your browser

Let’s run a Python file

- Create a file called hello.py via the web interface with print("Hello from Binder!") on the first line and commit to the main branch
- In the top right corner, click "New" ➡️ "Terminal"
- In the new tab with the terminal, type python hello.py and press return

Let’s add dependencies

- In your repo, create a file called requirements.txt
- Add a line that says: numpy==1.14.5
- Check for typos! Then commit to the main branch.
- Visit https://mybinder.org/v2/gh/your-username/repo/HEAD again in a new tab
- *BinderHub will read requirements.txt and install version 1.14.5 of the numpy package.*

Check the Environment

- In the top right corner, click "New" ➡️ "Python 3" to open a new notebook
- Type the following into a new cell

```
import numpy
print(numpy.__version__)
numpy.random.randn()
```

- Run the cell to see the version number and a random number printed
- Continue with this Binder Lesson:

https://github.com/alan-turing-institute/the-turing-way/blob/main/workshops/boost-research-reproducibility-binder/workshop-presentations/zero-to-binder-python.md#1-creating-a-repo-to-binderize

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. It’s really motivating to be part of such a welcoming community.”

https://www.turing.ac.uk/people/researchers/sarah-gibson

Thank you!

The Alan Turing Institute

Acknowledgements:

- These slides reuse previously developed materials by Kirstie Whitaker, Sarah Gibson, Martina Vilas, Malvika Sharan, Emma Karoune, + many in *The Turing Way* community
- README and LICENSE section reuses Open Life Science Materials

Useful Links

- Book: the-turing-way.netlify.com
- Twitter: twitter.com/turingway
- Newsletter: tinyletter.com/TuringWay
- GitHub: github.com/alan-turing-institute/the-turing-way
- Original artwork by Scriberia: https://doi.org/10.5281/zenodo.3332807