

Reproducibility and Extensibility in Scientific Research

Jessica Forde
Project Jupyter
@projectjupyter
@mybinderteam

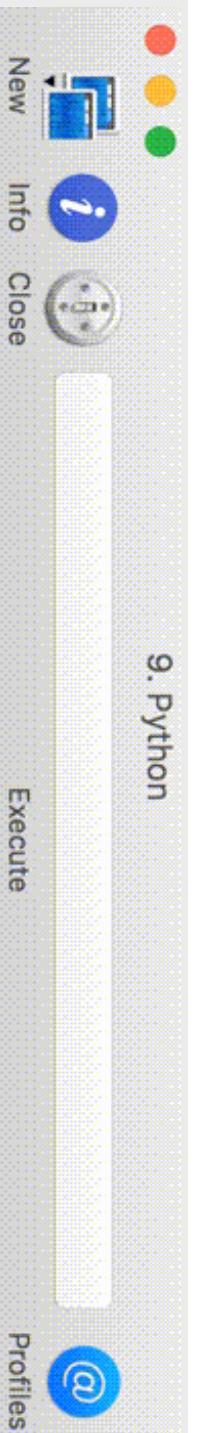


- 
- # Overview
- Project Jupyter
 - IPython
 - Jupyter Notebook
 - Architecture of JupyterHub
 - The problem of reproducibility in science
 - Repo2docker
 - Binder
 - Extending research via interactive computing

Who are we?

•().

9. Python



```
-bash-3.2$ ipython
WARNING: Attempting to work in a virtualenv. If you encounter problems, please
install IPython inside the virtualenv.
Python 2.7.10 (default, Sep 23 2015, 04:34:21)
Type "copyright", "credits" or "license" for more information.
```

```
IPython 4.0.2 -- An enhanced Interactive Python.
```

```
?          -> Introduction and overview of IPython's features.
```

```
%quickref -> Quick reference.
```

```
help       -> Python's own help system.
```

```
object?    -> Details about 'object', use 'object??' for extra details.
```

```
In [1]: from postal.expand import expand_address
```

```
In [2]: 
```

<https://machinelearnings.co/statistical-nlp-on-openstreetmap-b9d573e6cc86>

<https://dataorigami.net/blogs/napkin-folding/18487731-ipython-startup-scripts>

The screenshot shows a Jupyter Notebook interface. The top navigation bar includes tabs for '04-pretty_printing.py', '06-misc.py', 'def fun(x):', '02-autoreload.ipynb', '03-databases.py', 'README.md', '05-data_analysis.py', and '01-pl'. The left sidebar lists files: 'on-startup-files', 'wild_symlink', 'python_analysis', 'tup', '0-imports.py', '1-plotting.py', '2-autoreload.ipynb', '3-databases.py', '4-pretty_printing.py', '5-data_analysis.py', '6-msc.py', 'EADME', ':NSE', 'te.gif', 'ting.gif', 'DME.md', 'uirements.txt', and 's.gif'. The main notebook area displays the following code:

```
In [8]: %matplotlib
Using matplotlib backend: TkAgg
In [9]:
```

~2.1 Million Notebooks On GitHub

Repositories Developers

Trending: this month ▾

All languages
Unknown languages

C++
JavaScript

[fivethirtyeight / data](#)

Data and code behind the articles and graphics at FiveThirtyEight

● Jupyter Notebook

★ 9,553 ⚡ 3,929

Built by 

★ Star

Jupyter Notebook
Python
Matlab

★ 829 stars this month

[fastai / fastai](#)

The fast.ai deep learning library, lessons, and tutorials

● Jupyter Notebook

★ 4,236 ⚡ 1,307

Built by 

★ Star

Python
Matlab

★ 474 stars this month

⋮ Other: Languages ▾

[tensorflow / probability](#)

Probabilistic reasoning and statistical analysis in TensorFlow

● Jupyter Notebook

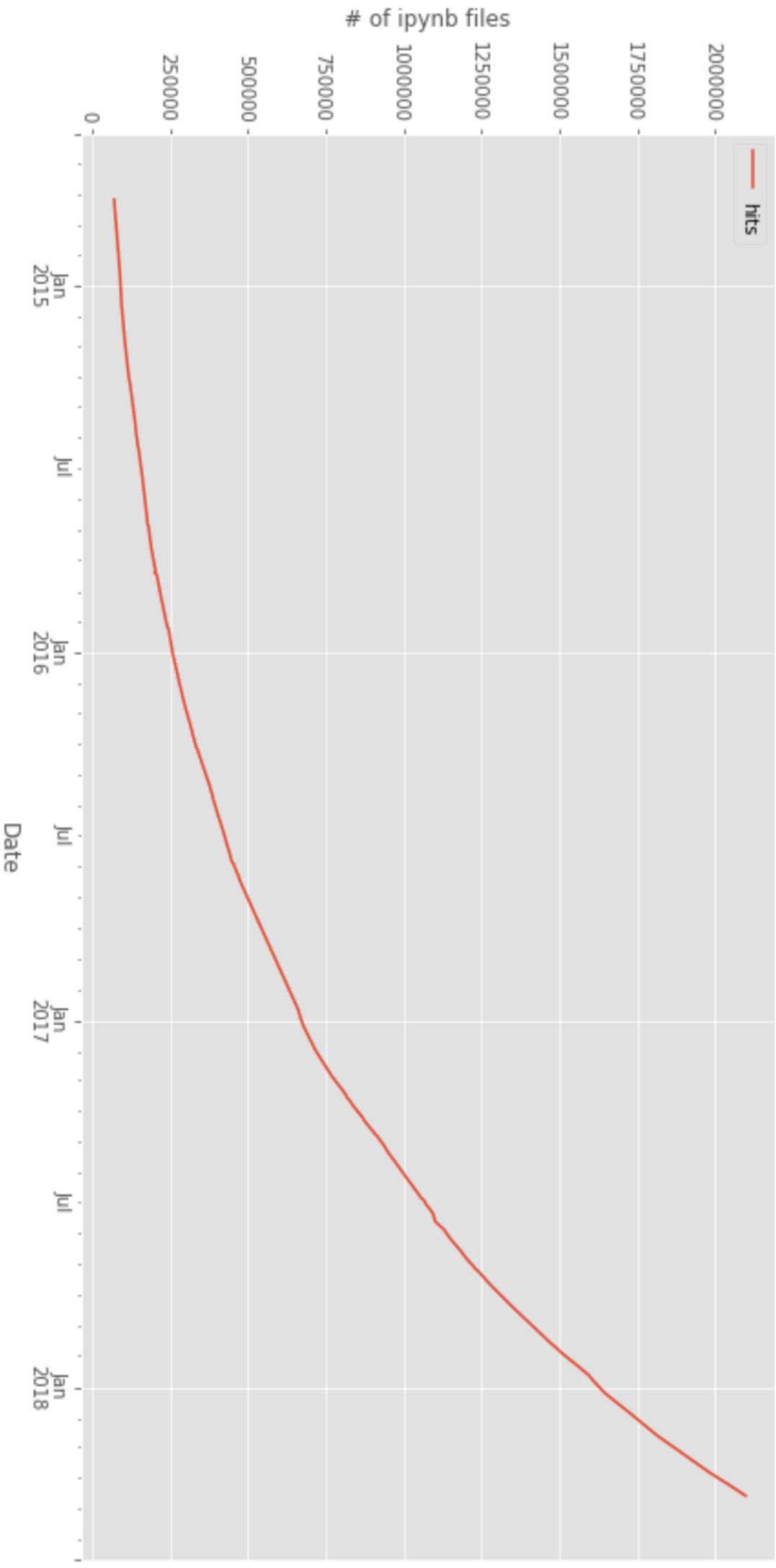
★ 500 ⚡ 62

Built by 

★ Star

<https://github.com/trending/jupyter-notebook?since=monthly>

GitHub search hits for 1291 days sans outliers



<https://mybinder.org/v2/gh/parente/nbestimate/master?filepath=estimate.src.ipynb>

Project Jupyter exists to develop open-source software, open-standards, and services for interactive computing across dozens of programming languages.



Project Jupyter Mission

•

Project Jupyter
Project Jupyter exists to develop open-source software, open standards, and services for interactive
and reproducible computing.
Jul 7, 2015 · 3 min read

GORDON AND BETTY
MOORE
FOUNDA
TION



THE LEONA M. AND HARRY B.
HELMESLEY
C H A R I T A B L E T R U S T

New funding for Jupyter

We are pleased to announce that the Jupyter/!Python project has received \$6M in funding from three organisations:

- The Leona M. and Harry B. Helmsley Charitable Trust
- The Gordon and Betty Moore Foundation
- The Alfred P. Sloan Foundation

The grant, which is being made both to the University of California, Berkeley and California Polytechnic State University, San Luis Obispo, will support the project for three years and includes new collaborations with the University of Southampton, and the Simula Research Lab in Norway.

Mission and Background

Project Jupyter's mission is to create open source tools for interactive scientific computing and data science in research, education and industry, with an emphasis on usability, collaboration and reproducibility.

This project is structured in a “3+1” format, with three main focus areas of research and development and one extra topic of ongoing work. The three focus areas are *Interactive Computing*, *Computational Narratives* and *Collaboration*. The problem of *Sustainability* will require ongoing attention but is conceptually distinct from the first three, as it doesn't focus on specific research questions or deliverables.



Figure 1. Jupyter notebooks exploring the Lorenz system.

Jupyter: Tools for the Life Cycle of a Computational Idea

By Matt Rockar-Kelley, Carol

Willing, and Jason Gross

Computation is increasingly becoming an integral part of science and education across disciplines. The life cycle of a computational idea typically involves interactive exploration and experimentation, as well as publication and communication of results. Reproducible computation demands open research tools, good software practices, and transparent documentation of research processes and results. Project Jupyter is an open community that builds open-source software tools and tools for the life cycle of computational ideas. Two core pieces of the project are an open protocol for interactive computation and an open document format with which to record and share computational ideas. The Jupyter Notebook application builds on these to provide a powerful, interactive computational environment.

LATeX, Markdown, and reStructuredText via Jupyter's conversion tool `nbconvert`.⁴ Using `Notebook` Documents, The Jupyter Notebook server is a web-based application for interacting with mathemat-

ical documents. The notebook document format is free,

transparent, and understandable, in keep-

ing with its aim to facilitate open and accessible science. It is stored as a single JSON-formatted text file, making it easy to manipulate and understand using standard programming tools, without the need for and interfere with mark-

down cells to explain an idea using formatted text and mathematics (see Figure 1).

(see Figure 1).

Many authors communicate using Jupyter Notebooks. GitHub hosts 1.4 million notebooks, and some people have written entire books as collections of notebooks, such as Jake Vanderplas's *Python Data Science Handbook*.⁵ Because notebook documents preserve their content structure and metadata, they are easily convertible to other formats, including plain scripts in the document's language of choice. This also makes them easy to integrate into publishing pipelines via formats such as

LATeX, Markdown, and reStructuredText via Jupyter's conversion tool `nbconvert`.⁴ Using `Notebook` Documents, The Jupyter Notebook server is a web-based application for interacting with mathemat-

⁴The Jupyter Notebook. What is a Notebook? The Jupyter Notebook is a web-based application for interacting with mathematical documents.

⁵What is a Notebook? The Jupyter Notebook is a web-based application for interacting with mathematical documents.

⁶Figure 1. Jupyter notebooks exploring the Lorenz system.

We're not-for-profit

README.md

Project Jupyter Governance

The purpose of this repository is to formalize the governance process that the IPython/Jupyter project has used informally since its inception in 2001. This document clarifies how decisions are made and how the various elements of our community interact, including the relationship between open source collaborative development and work that may be funded by for-profit or non-profit entities.

Table of Contents

- Main Governance Document
- Current Steering Council and Institutional Partners
- New Subproject Incubation Process
- Process for Authoring Jupyter Related Academic Papers

License of Governance Documents

To the extent possible under law, Project Jupyter has waived all copyright and related or neighboring rights to the Project Jupyter Governance documents, in accordance with the Creative Commons [CC0 license](#). This work is published from the United States. See the LICENSE.md file in this repository for details.

<https://github.com/jupyter/governance>

Jupyter Notebook

•().

JUPYTER NOTEBOOK



View and Enter
Stuff Part



Notebook Server

ZeroMQ

Julia

R

Python

Kernels

Language Expert Thing.....

More than Python

•().

PyROOT_Example Last Checkpoint: 16 minutes ago (autosaved)

File Edit View Insert Cell Kernel Help



Markdown

Not Trusted Python 2 O

Interleave Python with C++: the %%cpp magic [¶](#)

Thanks to ROOT, it is possible to write cells in C++ within a Python notebook. This can be done using the %%cpp magic. Magics are a feature of Jupyter notebooks and when importing the ROOT module, the %%cpp magic was registered.

```
In [10]: %%cpp
cout << "This is a C++ cell" << endl;
```

This is a C++ cell

Not bad. On the other hand, ROOT offers much more than this. Thanks to its [interpreter](#) and [type system](#), entities such as functions, classes and variables, created in a C++ cell, can be accessed from within Python (and viceversa, partially).

```
In [11]: %%cpp
class A{
public:
A(){cout << "Constructor of A!" << endl;}
};
```

```
In [12]: a = ROOT::A()
```

Constructor of A!

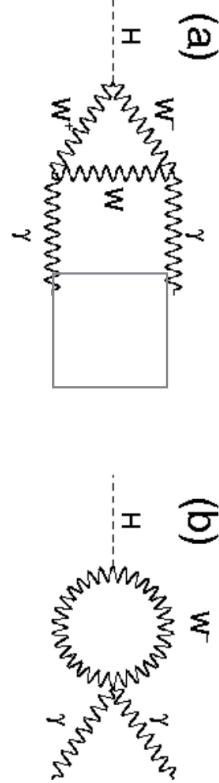
The Python and C++ worlds are so entangled that we can find back in C++ the entities created in Python. To illustrate this, from within a C++ cell, we are going to fit a function in the gauss histogram displayed above and then re-draw the canvas.

```
In [13]: %%cpp
gauss->Fit("gaus", "S");
myCanvasName->Draw();
```

```
FCN=47.4997 FROM MIGRAD   STATUS=CONVERGED  53 CALLS      54 TOTAL
EDM=8.44224e-09  STRATEGY= 1  ERROR MATRIX ACCURATE
```

Higgs decay to two photons

The Standard Model predicted the decay of the Higgs bosons into photons. The process is depicted by the diagrams below:

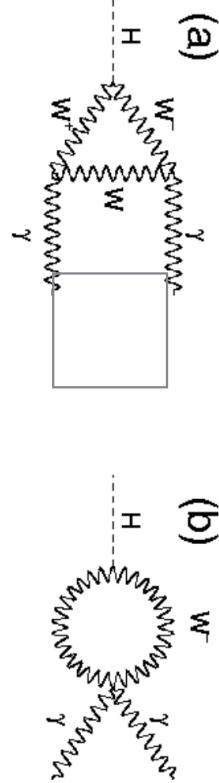


At the [Large Hadron Collider](#), this process has been measured. This figure shows how an Higgs boson decay looks in the CMS detector:



Higgs decay to two photons

The Standard Model predicted the decay of the Higgs bosons into photons. The process is depicted by the diagrams below:



At the [Large Hadron Collider](#), this process has been measured. This figure shows how an Higgs boson decay looks in the CMS detector:



```
In [ ]: import (
    "fmt"
    "time"
)

In [ ]: // sum calculates the sum of two integers
func sum(x, y int) int {
    return x + y
}

In [ ]: a, b := 3, 4

In [ ]: fmt.Sprintln("sum(%d, %d) = %d", a, b, sum(a, b))

In [ ]: start := time.Now()
defer func() {
    end := time.Now()
    fmt.Println("Interrupted an infinite loop after",
               end.Sub(start))
}()

for {}
```

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 Note!
Coarray-Fortran	Jupyter 4.0	Fortran 2008/2015	GFortran >= 7.1, OpenCoarrays , MPICH >= 3.2	Demo Bind demo
sparkmagic	Jupyter >=4.0	Pyspark (Python 2 & 3), Spark (Scala), SparkR (R)	Livy	Notebook Docker Image

<https://github.com/jupyter/jupyter/wiki/Jupyter-kernels>

Creating new Jupyter kernels

[Making kernels for Jupyter](#) in the documentation.

[Simple example kernel](#)

[IHaskeil creator blog post](#)

[Testing kernels against message specification \(work in progress\)](#)

[Tool to test a kernel against specification \(work in progress\)](#)

More than Notebooks

•().

Jupyter hub

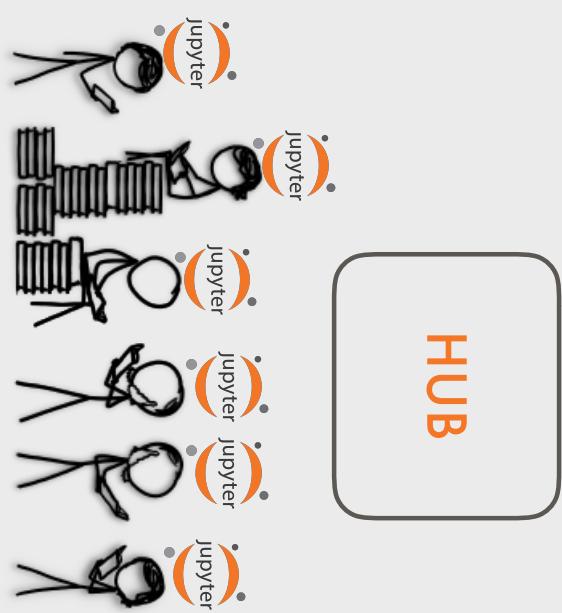
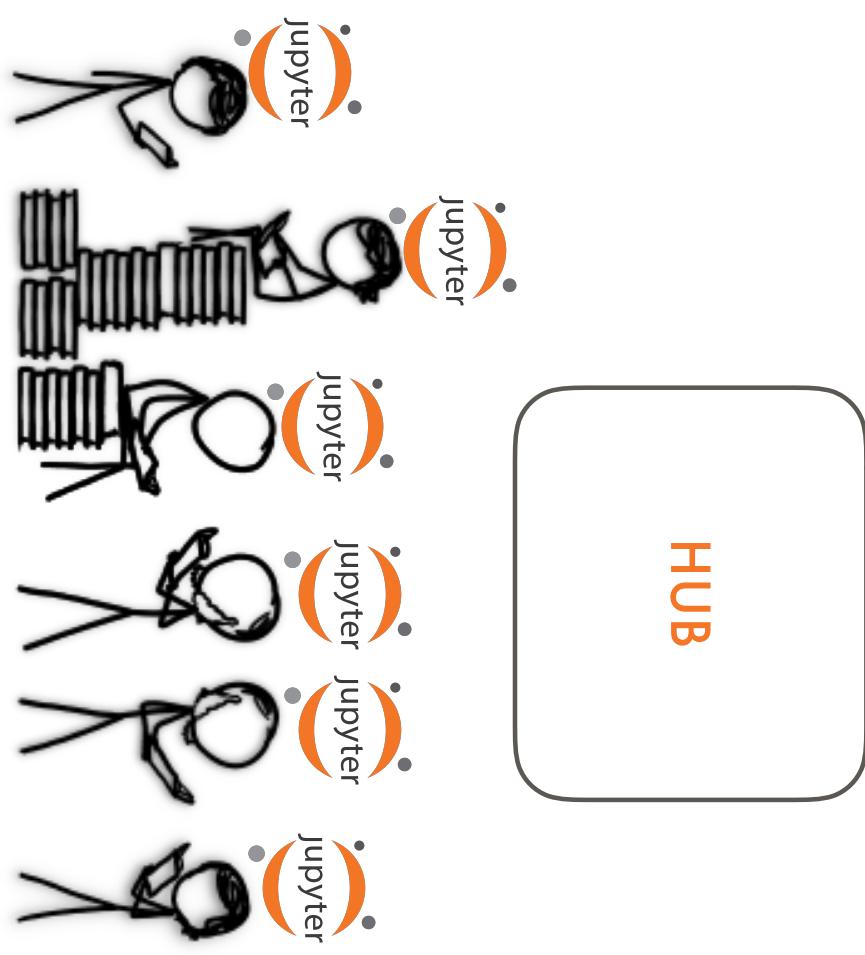
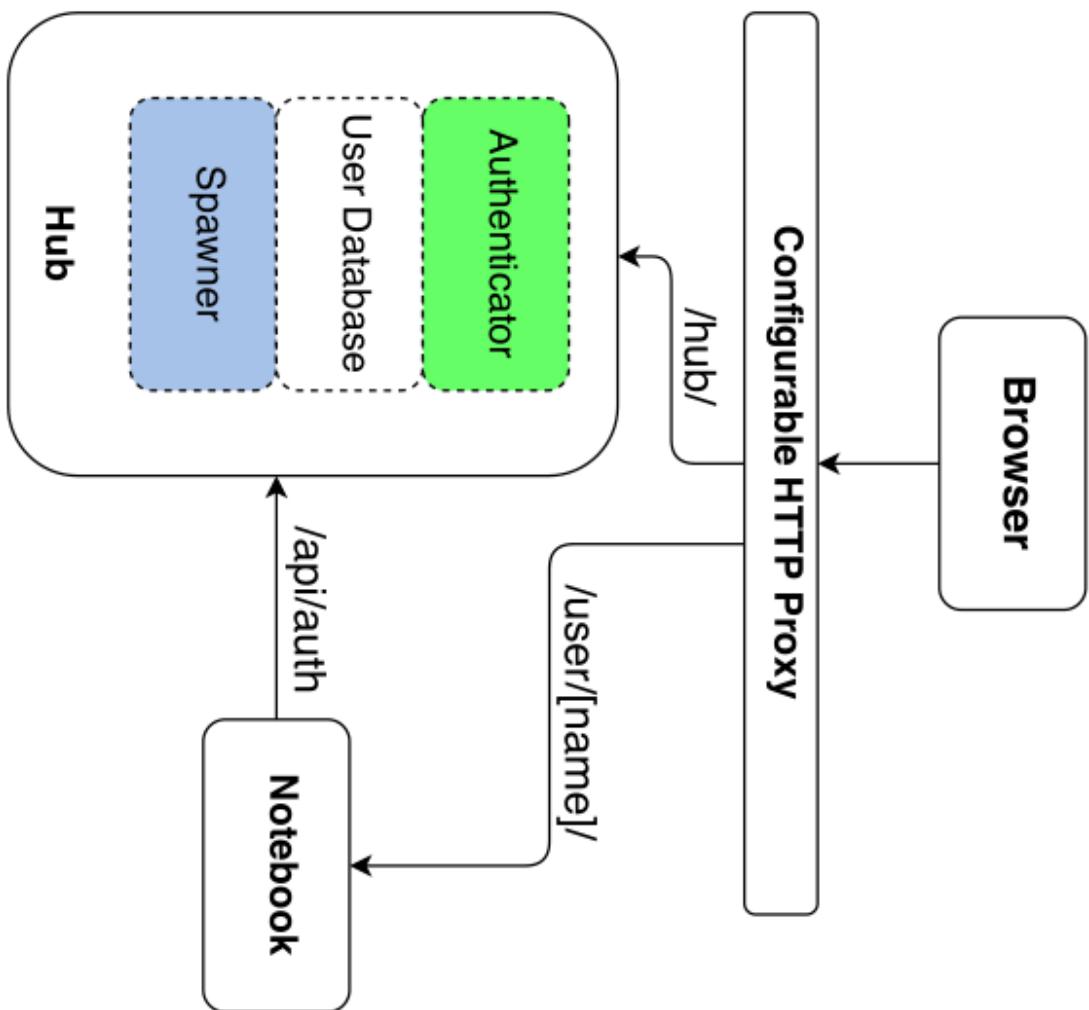


photo: xkcd, Jupyter, C. Willing

- JupyterHub provides a single user Jupyter Notebook server for each person in a group
- Similar to SWAN

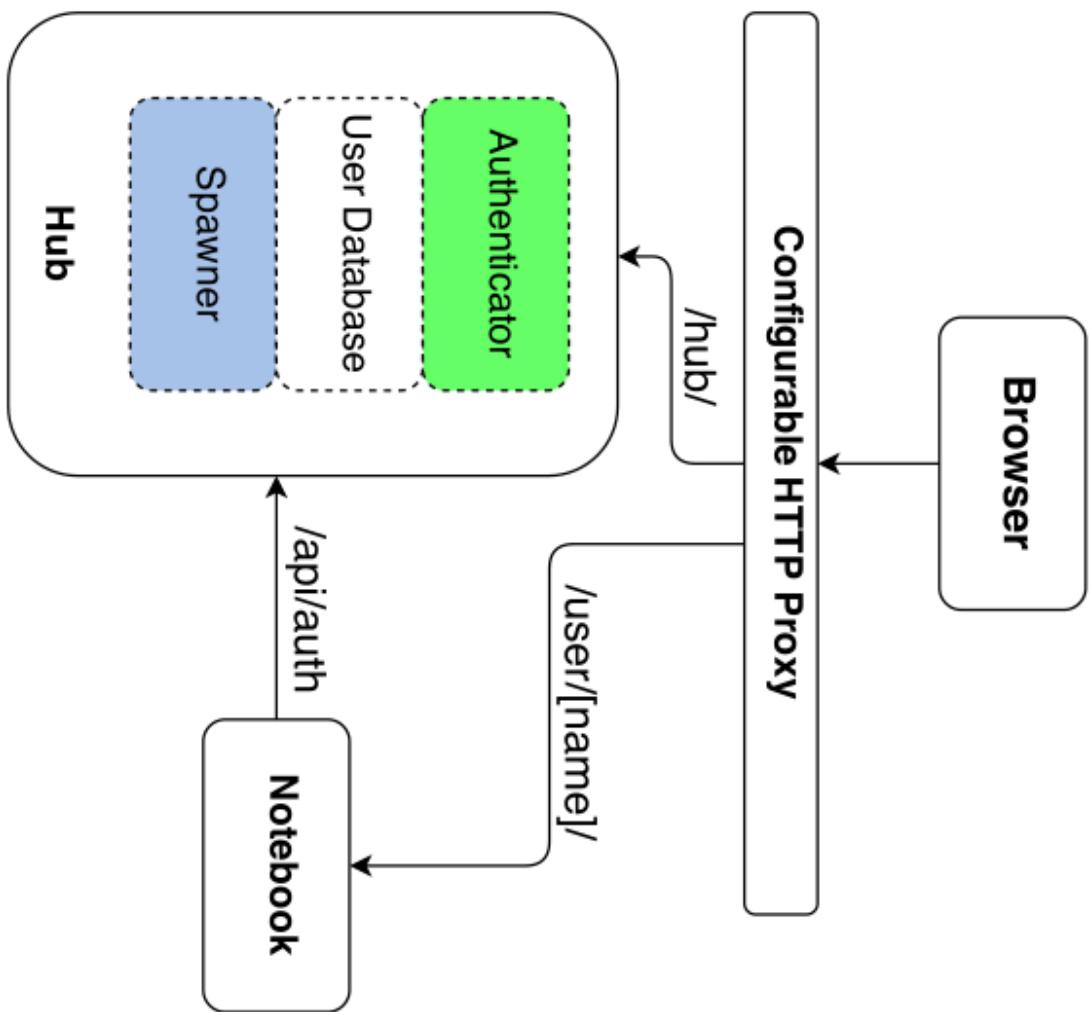


- **Hub**: manages user accounts, authentication, and coordinates Single User Notebook Servers using a Spawner
- **Proxy**: routes HTTP requests to the Hub and Single User Notebook Servers.
- **Spawner**: starts a **single-user notebook servers** when a user logs in



•()•

- Hub launches a proxy
- Proxy forwards all requests to Hub by default
- Hub handles login, and spawns single-user servers on demand
- Hub configures proxy to forward url prefixes to the single-user notebook servers



Kubernetes as Spawner

[Next »](#)

Zero to JupyterHub



JupyterHub is a tool that allows you to quickly utilize cloud computing infrastructure to manage a hub that enables users to interact remotely with a computing environment that you specify. JupyterHub offers a useful way to standardize the computing environment of a group of people (e.g., for a class of students or an analytics team), as well as allowing people to access the hub remotely.

Zero to JupyterHub

This growing collection of information will help you set up your own JupyterHub instance. It is in an early stage, so the information and tools may change quickly. If you see anything that is incorrect or have any questions, feel free to reach out at the [issues page](#).

Kubernetes

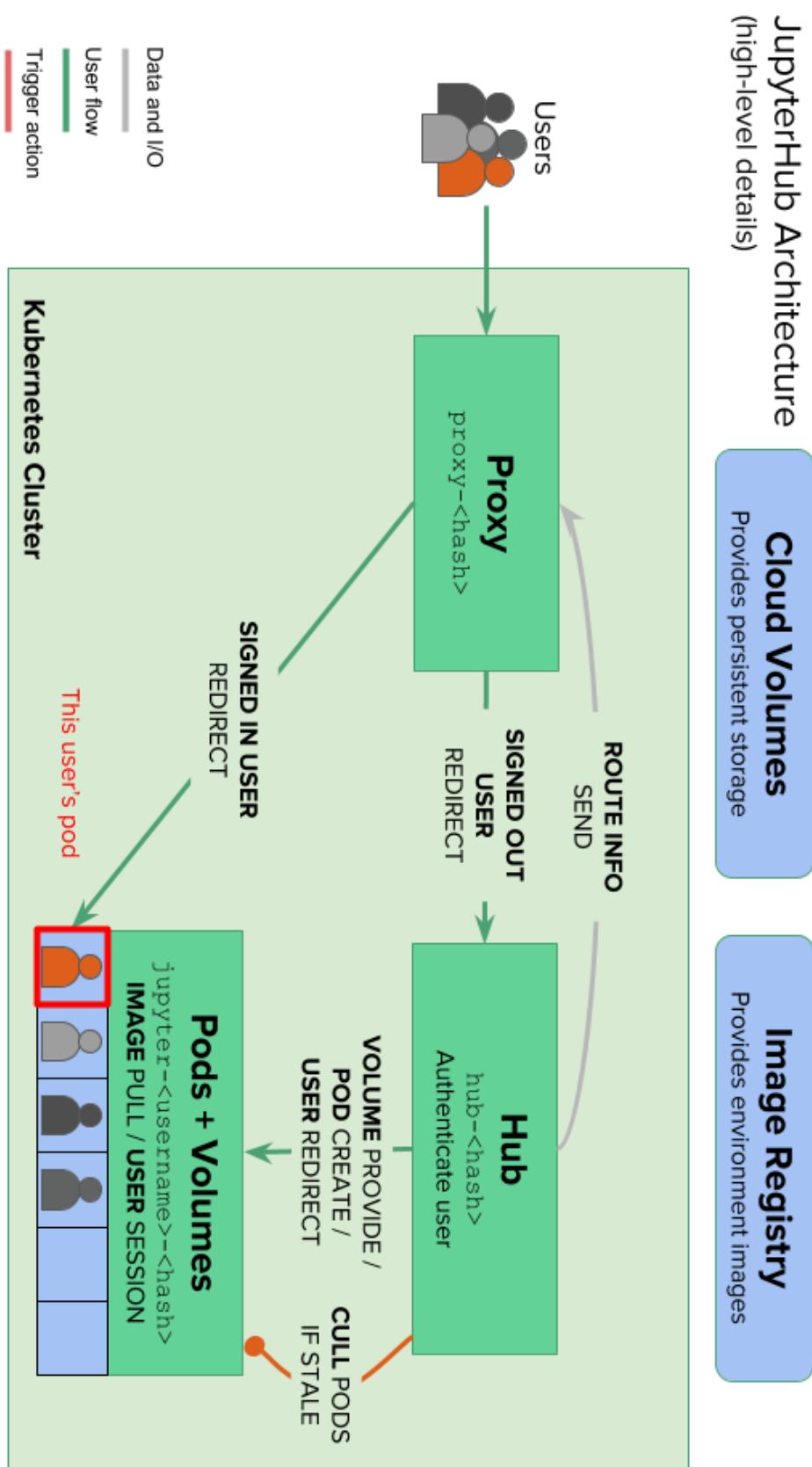
A tutorial to help install and manage JupyterHub with Kubernetes

Quick search

Creating your JupyterHub

This tutorial starts from “step zero” and walks through how to install and configure a complete JupyterHub deployment in the cloud. Using Kubernetes and the JupyterHub Helm chart provides sensible defaults for an initial deployment.

Kubernetes as Spawner



Scaling JupyterHub



KubeCon + CloudNativeCon North America 2017 has ended

Thursday, December 7 • 3:50pm - 4:25pm

Large Scale Teaching Infrastructure with Kubernetes - Yuvi Panda, Berkeley University

Sign up or log in to save this to your schedule and see who's attending!

<http://sched.co/CU7L>

Tweet

Share

Data Science & Programming literacy is an important aspect of literacy in the 21st century, but teaching these skills at scale is quite difficult. At UC Berkeley, we are trying - our 'Foundations of Data Science' course has no pre-requisites, and routinely attracts more than a 1000 students from across majors.

<https://kccncna17.sched.com/event/CU7U/large-scale-teaching-infrastructure-with-kubernetes-yuvi-panda-berkeley-university>

Scaling JupyterHub

PAWS

Phabricator project: [#paws](#)

PAWS: A Web Shell (PAWS) is a Jupyter notebooks [deployment](#) that has been customized to make interacting with Wikimedia wikis easier. It allows users to create and share documents that contain live code, visualizations such as graphs, rich text, etc. The user created notebooks are a powerful tool that enables data analysis and scientific research, and also transforms the way in which programmers write code - by enabling an exploratory environment with a quick feedback loop, and a low barrier for entry through it's easy to use graphical interface.

Sign in with your wiki account and tada!

Contents [hide]

- 1 Usage
- 2 Documentation
- 3 Other notes
- 4 See also



Scaling JupyterHub

Pangeo: JupyterHub, Dask, and XArray on the Cloud

This work is supported by Anaconda Inc, the NSF EarthCube program, and UC Berkeley BIDS

A few weeks ago a few of us stood up pangeo.pydata.org, an experimental deployment of JupyterHub, Dask, and XArray on Google Container Engine (GKE) to support atmospheric and oceanographic data analysis on large datasets. This follows on [recent work](#) to deploy Dask and XArray for the same workloads on super computers. This system is a proof of concept that has taught us a great deal about how to move forward. This blogpost briefly describes the problem, the system, then describes the collaboration, and finally discusses a number of challenges that we'll be working on in coming months.

The Problem

Atmospheric and oceanographic sciences collect (with satellites) and generate (with simulations) large datasets that they would like to analyze with distributed systems. Libraries like Dask and XArray already solve this problem computationally if scientists have their own clusters, but we seek to expand access by deploying on cloud-based systems. We build a system to which people can log in, get Jupyter Notebooks, and launch Dask clusters without much hassle. We hope that this increases access, and connects more scientists with more cloud-based datasets.

Kubeflow

The Kubeflow project is dedicated to making **deployments** of machine learning (ML) workflows on **Kubernetes** simple, portable and scalable. Our goal is **not** to recreate other services, but to provide a straightforward way to deploy best-of-breed open-source systems for ML to diverse infrastructures. Anywhere you are running Kubernetes, you should be able to run Kubeflow.

This repository contains the manifests for creating:

- A **JupyterHub** to create and manage interactive **Jupyter notebooks**. **Project Jupyter** is a non-profit, open-source project to support interactive data science and scientific computing across all programming languages.
- A **TensorFlow Training Controller** that can be configured to use either CPUs or GPUs and dynamically adjusted to the size of a cluster with a single setting
- A **TensorFlow Serving** container to export trained TensorFlow models to Kubernetes

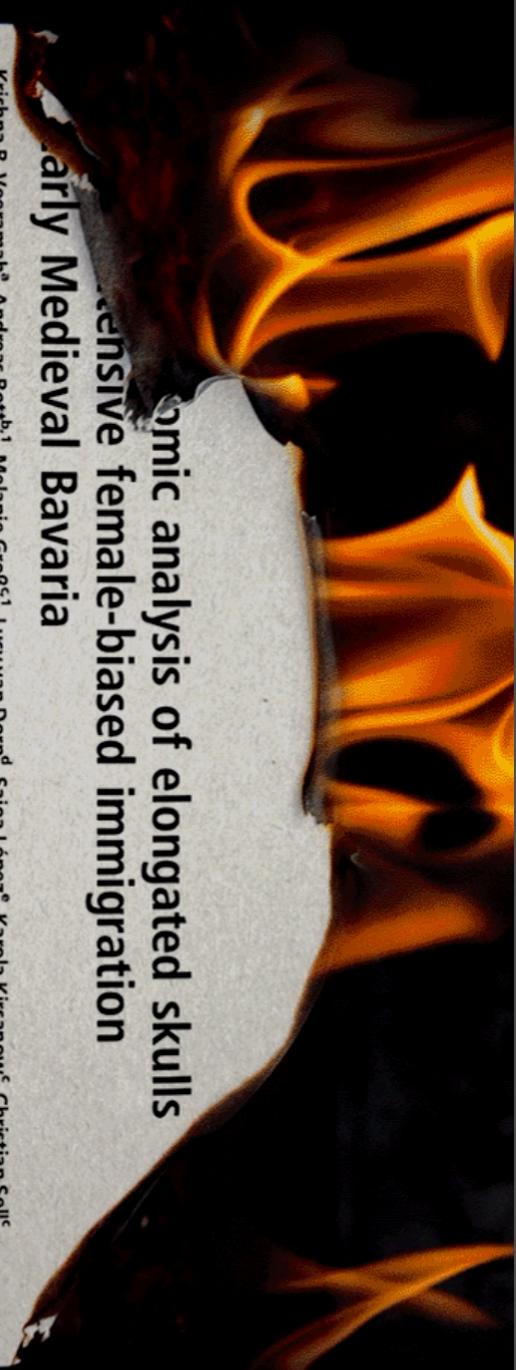
This document details the steps needed to run the Kubeflow project in any environment in which Kubernetes runs.

Reproducibility

• ().

The Scientific Paper Is Obsolete

Genomic analysis of elongated skulls suggests extensive female-biased immigration to Early Medieval Bavaria



Krishna R. Veeramah^a, Andreas Rott^{b,1}, Melanie Groß^{c,1}, Lucy van Dorp^d, Saioa López^e, Karola Kirsanow^c, Christian Sell^c, Jens Blöcher^c, Daniel Wegmann^{f,g}, Vivian Link^{h,g}, Zuzana Hofmanová^g, Joris Peters^{b,h}, Bernd Trautmann^b, Anja Gaihohsⁱ, Jochen Haberstroh^j, Bernd Päffgen^k, Garrett Hellenthal^d, Brigitte Haas-Gebhardⁱ, Michaela Harbeck^{b,2,3}, and Joachim Burger^{c,2,3}

^aDepartment of Ecology and Evolution, Stony Brook University, Stony Brook, NY 11794-5245; ^bState Collection for Anthropology and Palaeoanthropology, Bavarian Natural History Collections, 80333 Munich, Germany; ^cPalaeogenetics Group, Institute of Organismic and Molecular Evolution, Johannes Gutenberg University Mainz, 55099 Mainz, Germany; ^dUCL Genetic Institute, Department of Genetics, Evolution and Environment, University College London, WC1E 6BT London, United Kingdom; ^eCancer Institute, University College London, WC1E 6DD London, United Kingdom; ^fDepartment of Biology, University of Fribourg, 1700 Fribourg, Switzerland; ^gSwiss Institute of Bioinformatics, 1700 Fribourg, Switzerland; ^hArchaeoBioCenter and Institute for Palaeoanthropology, Institute of Archaeological Research, Ludwig-Maximilians-Universität München, 80539 Munich, Germany; ⁱBavarian State Archaeological Museum, 80333 Munich, Germany; ^jInstitute of Archaeology, Ludwig-Maximilians-Universität München, 80539 Munich, Germany; ^kInstitute of Archaeology, University of Bonn, 53101 Bonn, Germany

Edited by Eske Willerslev, University of Copenhagen, Copenhagen, Denmark, and approved January 30, 2018 (received for review November 21, 2017)

Modern European genetic structure demonstrates strong correlations with geography, while genetic analysis of prehistoric humans has indicated at least two major waves of immigration from outside the continent during periods of cultural change. However, population-level genome data that could shed light on the demographic processes occurring during the intervening periods have been absent. Therefore, we generated genomic data

to form in the 5th century AD, and that it emanated from a combination of the romanized local population of the border province of the former Roman Empire and immigrants from north of the Danube (2). While the Baiuvari are less well known than some other contemporary groups, an interesting archaeological feature in Bavaria from this period is the presence of skeletons with artificially deformed or elongated skulls (Fig. 1*A*).

POPULATION
BIOLOGY

PNAS / Richard Goerg / Getty / The Atlantic

Here's what's next.

•().

The more sophisticated science becomes, the harder it is to communicate results. Papers today are longer than ever and full of jargon and symbols. They depend on chains of computer programs that generate data, and clean up data, and plot data, and run statistical models on data. These programs tend to be both so sloppily written and so central to the results that it's contributed to a replication crisis, or put another way, a failure of the paper to perform its most basic task: to report what you've actually discovered, clearly enough that someone else can discover it for themselves.

- James Somers



•().

Reproducibility:
Producing similar results
with the same data

•().

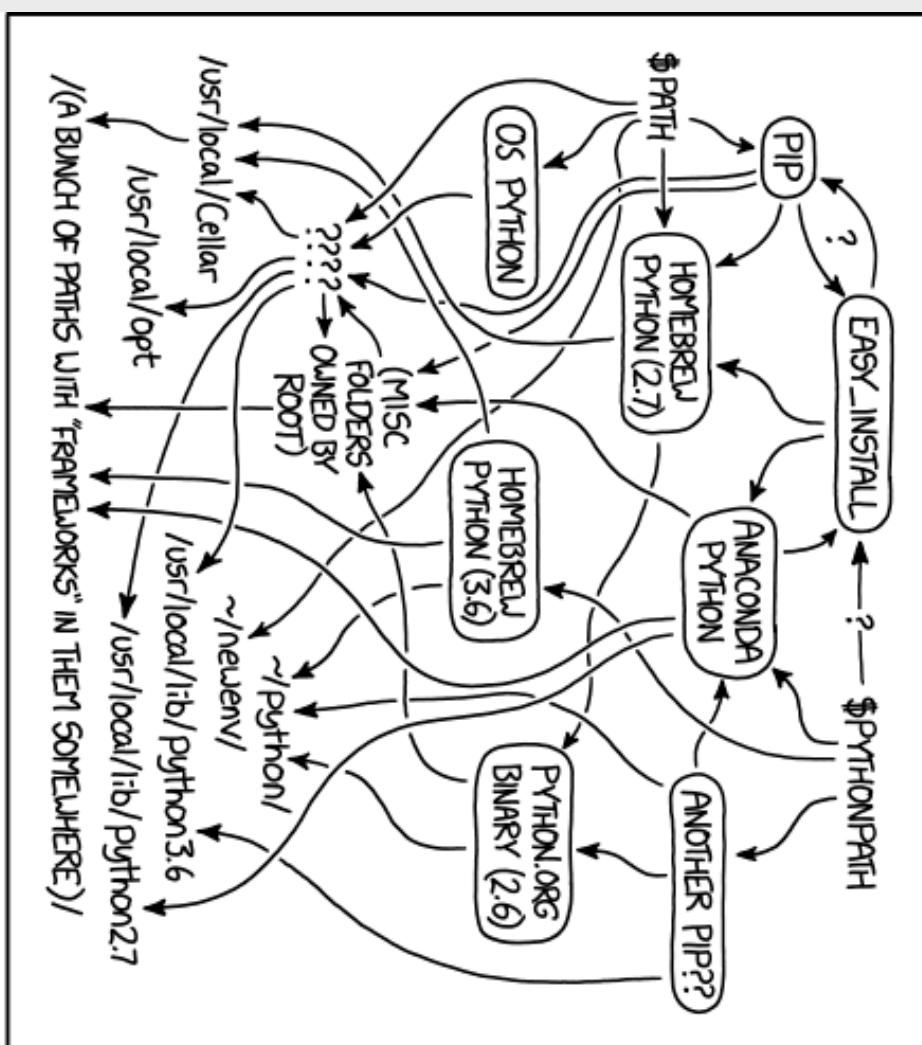
Reproducibility: A software problem

•().

Technical Solutions

- GitHub
- Open Science Framework
- Codalab
- RunMyCode
- Research Journals

This week in xkcd



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Reproducible scientific software pipelines

- Repository would contain:
 - Data
 - Dependencies
 - Hyperparameters
 - Scripts to run the jobs on similar hardware
 - Analysis code

Reproducible scientific software pipelines

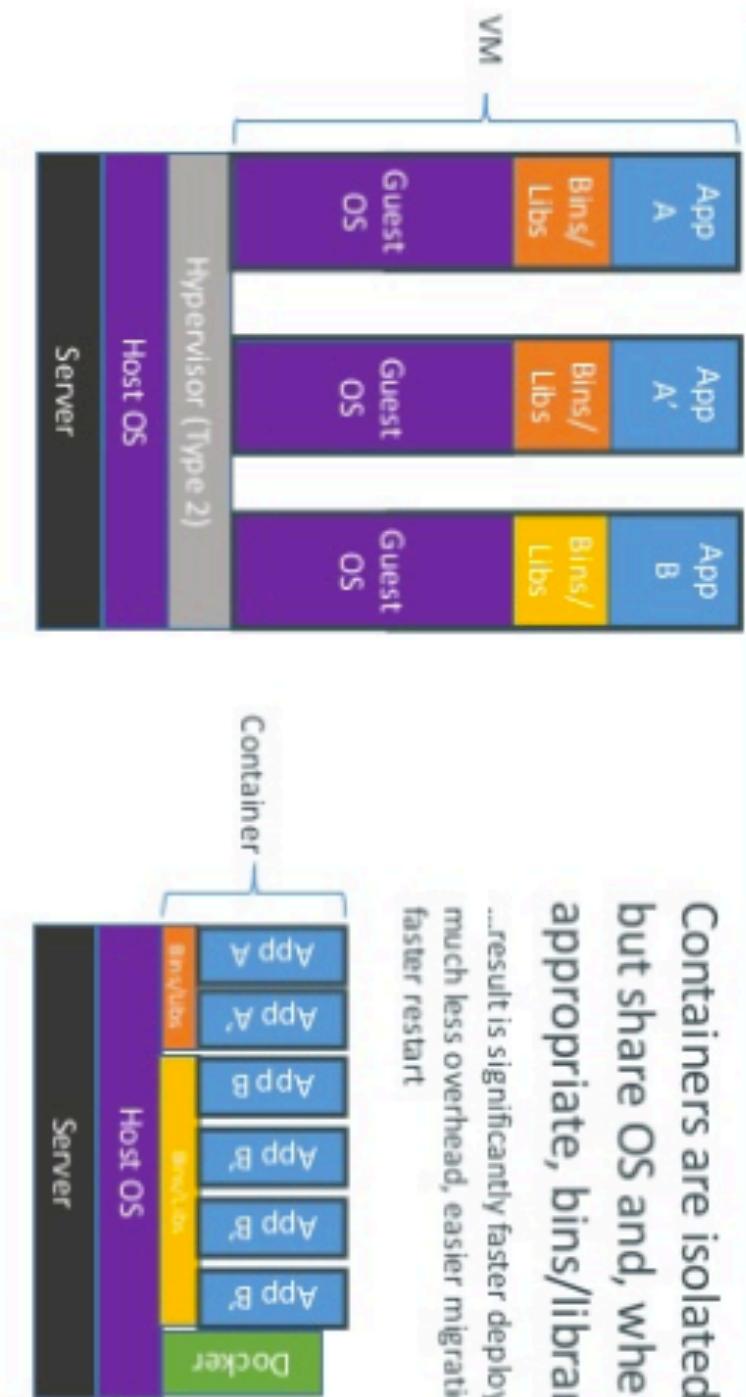
- Repository would contain:

- Data
- Dependencies
- Hyperparameters
- Scripts to run the jobs on similar hardware
- Analysis code

No mention of OS or
lower-level software

Docker for reproducible software

Containers vs. VMs



source: Docker



Dockerfiles from GitHub Repos

jupyter-repo2docker

[build](#) passing [docs](#) passing

`jupyter-repo2docker` takes as input a repository source, such as a GitHub repo. It then builds, runs, and/or pushes Docker images built from that source.

See the [repo2docker documentation](#) for more information.

Pre-requisites

1. Docker to build & run the repositories. The [community edition](#) is recommended.
2. Python 3.4+.

Supported on Linux and macOS. See [documentation note about Windows support](#).

Using repo2docker

Note that Docker needs to be running on your machine for this to work.

Example:

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

After building (it might take a while!), it should output in your terminal something like:

Copy/paste this URL into your browser when you connect **for** the first time,
to login with a token:

```
http://0.0.0.0:36511/?token=f94f8fabb92e22f5bfab116c382b4707fc2cade56adace0
```



repo2docker as reproducibility unit-test

- repo2docker looks for configuration files to determine how to build the docker image
- Typically describe dependencies and other instructions to create the environment
- configs in either root or folder called binder

Supported config files

- **Dockerfile**: full environment setup
- **environment.yml**: conda
- **requirements.txt**: pip
- **REQUIRE**: Julia
- **apt.txt**: Debian packages
- **postBuild**: custom install script
- **runtime.txt**: Python runtime

Example repo

Branch: master ▾ [GAN_tutorial](#) / environment.yml

 mickypaganini Update environment.yml

1 contributor

9 lines (8 sloc) | 139 Bytes

```
1 name: binder
2 dependencies:
3   - pytorch
4   - torchvision
5   - matplotlib==2.0.2
6   - numpy==1.14.1
7   - ipython==6.2.1
8   - scikit-learn==0.19.1
```

Branch: master ▾ [GAN_tutorial](#) / runtime.txt

 mickypaganini Create runtime.txt

1 contributor

2 lines (1 sloc) | 11 Bytes

```
1 python-2.7
```