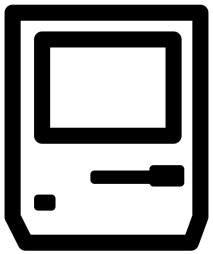
MAX PLANCK INSTITUTE FOR **BIOLOGY OF AGEING**

Reproducible multilang workflows with Jupyter on Docker

Computers are not all the same

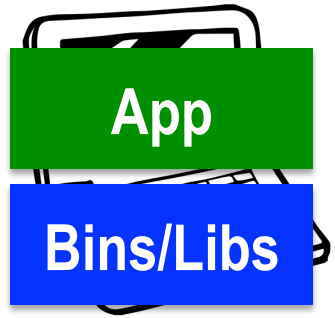
Created by Mike O'Brien
from Noun Project

Created by Mister Pixel
from Noun Project

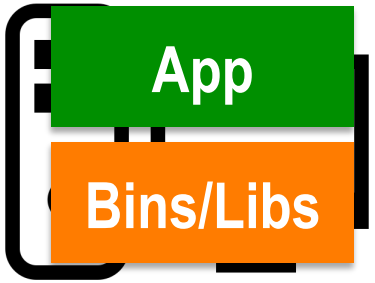


Created by factor[e] design initiative
from Noun Project

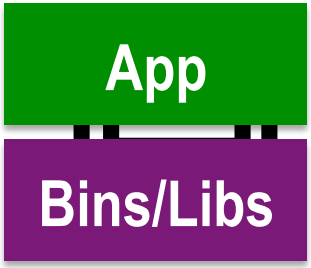
Computers are not all the same



Created by Mike O'Brien
from Noun Project



Created by Mister Pixel
from Noun Project

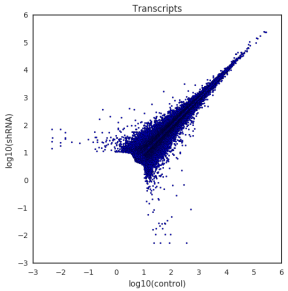


Created by factor[e] design initiative
from Noun Project

The same apparent code can give different results

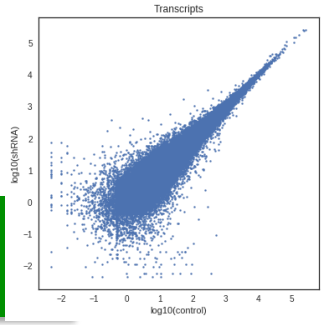
App

Bins/Libs



App

Bins/Libs

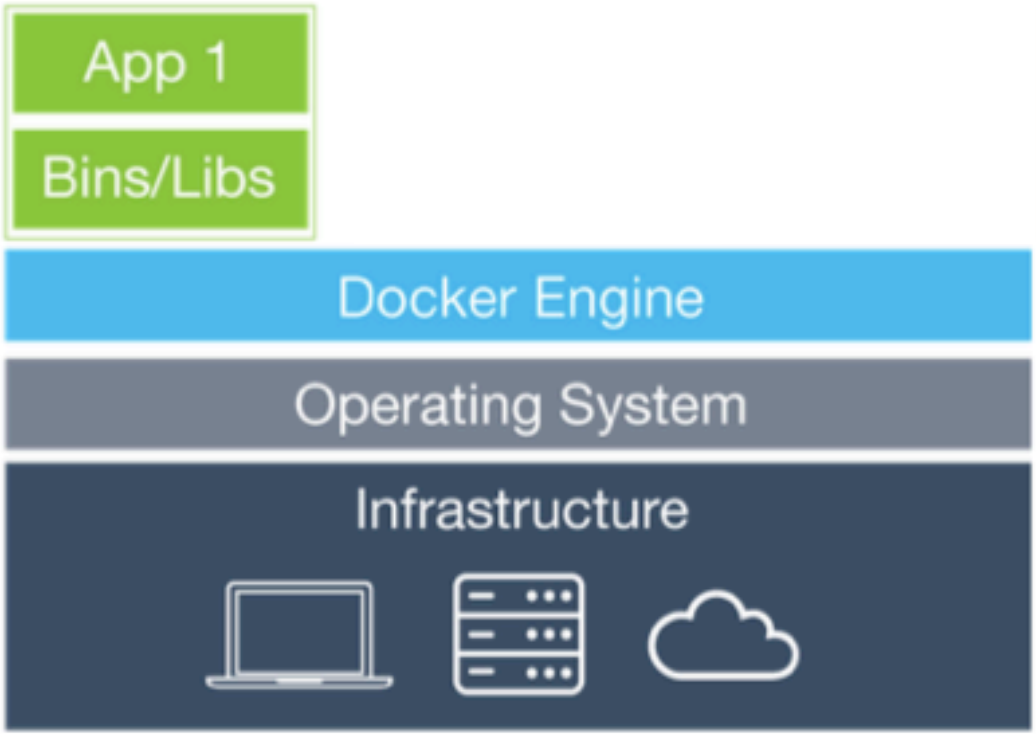


App

Bins/Libs

Different version of dependency in bin/
== different input in one step
== code breaks

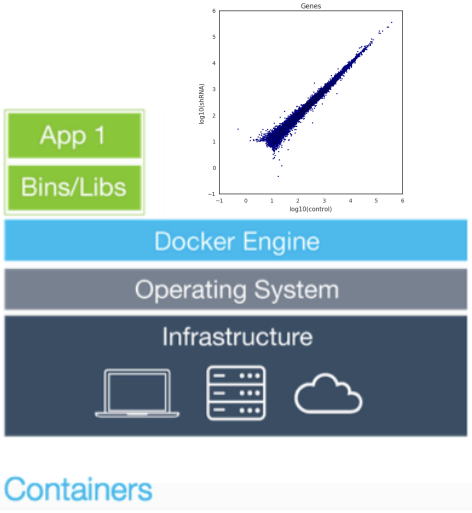
Containers

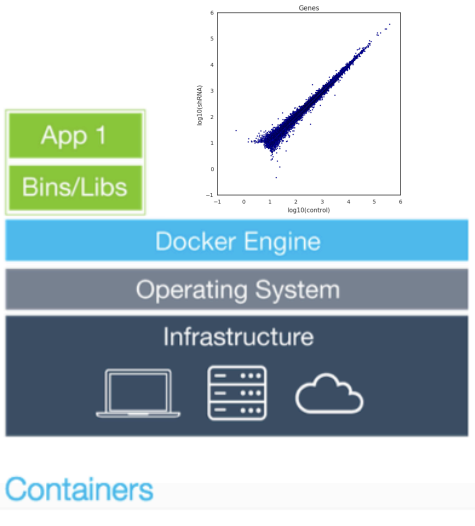


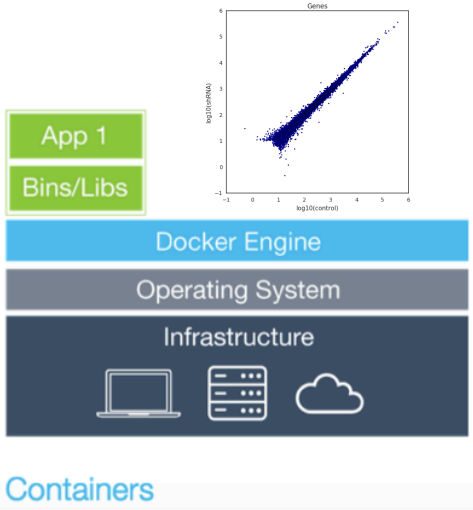
Containers

https://blog.logentries.com/2015/07/an-all-inclusive-log-monitoring-container-for-docker/

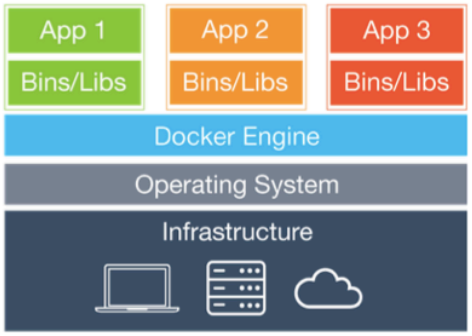
Containers are independent of their host



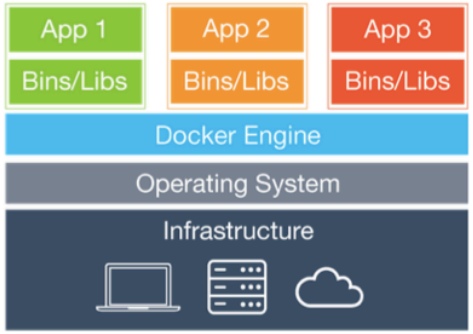




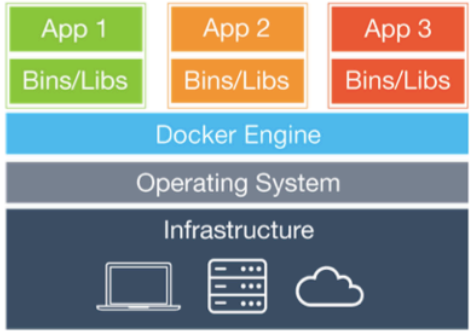
Multiple environments per machine



Containers

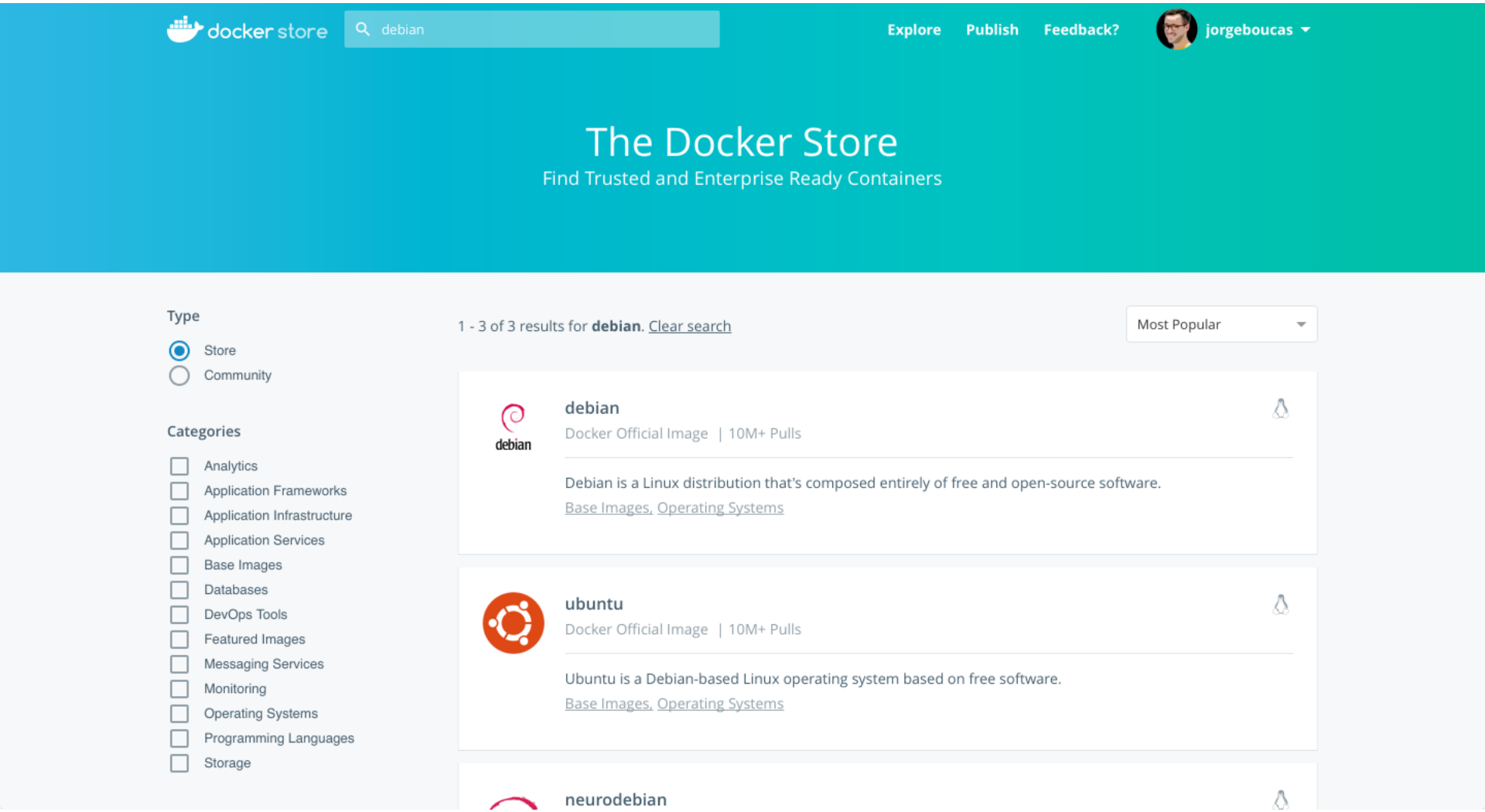


Containers

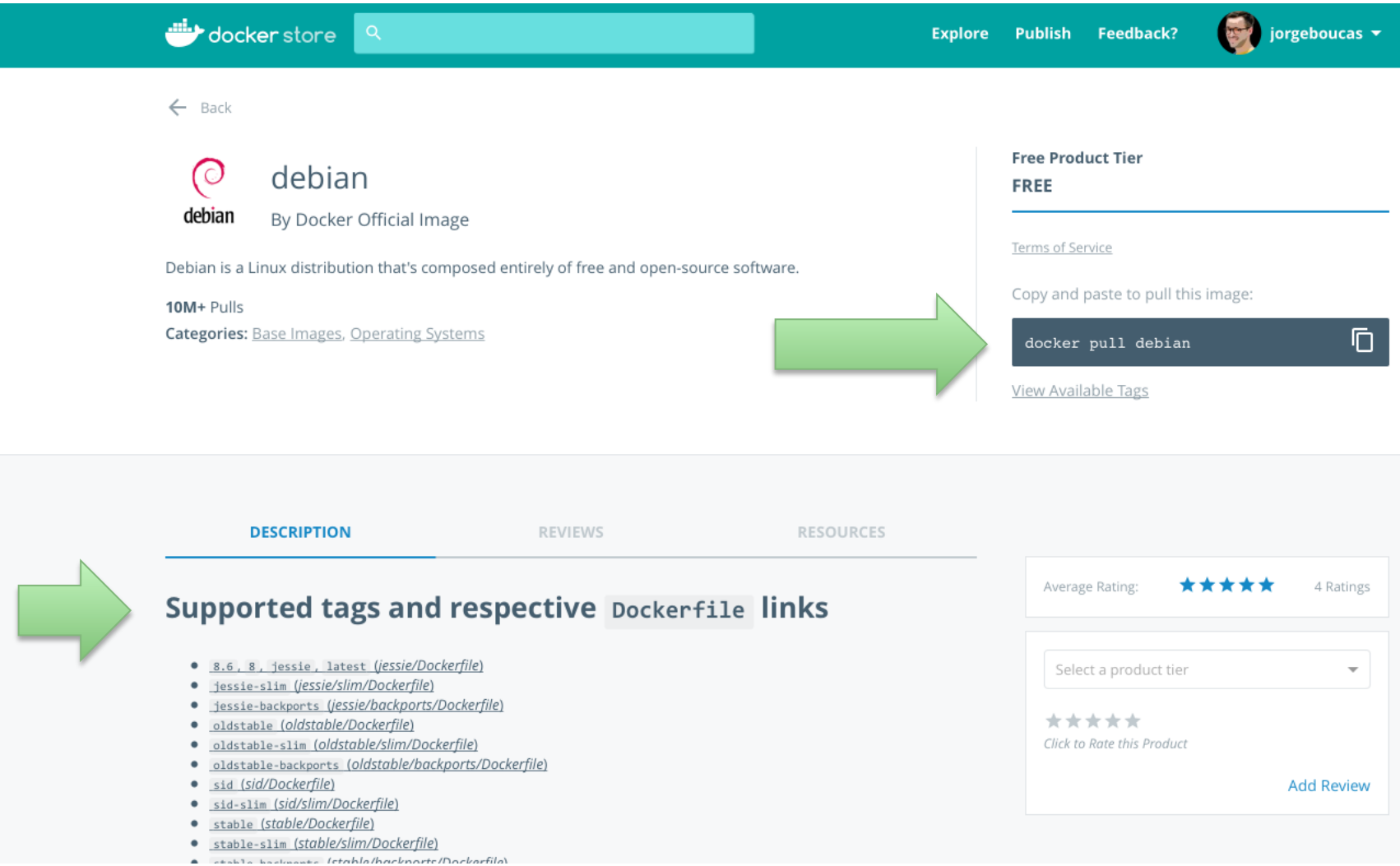


Containers

https://store.docker.com



docker pull debian



running a container

```
## pull an image from the docker store

jboucas-mbpr:~$ docker pull debian

Using default tag: latest
latest: Pulling from library/debian
5040bd298390: Pull complete
Digest: sha256:abbe80c8c87b7e1f652fe5e99ff1799cdf9e0878c7009035afe1bccac129cad8
Status: Downloaded newer image for debian:latest

## start a container from the downloaded image
## and initiate an interactice bash session

jboucas-mbpr:~$ docker run --name my_first_container -t -i debian /bin/bash

root@43530985a83d:/# pwd
/
root@43530985a83d:/# ls
bin  boot  dev  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin
srv  sys  tmp  usr  var
```

Working on a container

```
## change the content of the container

root@43530985a83d:/# touch jorge.txt
root@43530985a83d:/# ls
bin  boot  dev  etc  home  jorge.txt  lib  lib64  media  mnt  opt  proc  root
run  sbin  srv  sys  tmp  usr  var
root@43530985a83d:/# exit
exit

## list containers

jboucas-mbpr:~$ docker ps -a

CONTAINER ID        IMAGE                  COMMAND                  CREATED
STATUS                      PORTS                    NAMES
43530985a83d        debian                 "/bin/bash"              58 minutes
ago      Exited (0) 3 seconds ago                        my_first_container
c3ad2c600b24        encode_rnaseq_eclip    "tini -- start-not..."   10 days ago
Up 3 days                   0.0.0.0:8888->8888/tcp   encode_rnaseq_eclip
```

Restarting a container

```
## start a container

jboucas-mbpr:~$ docker start my_first_container
my_first_container

## start an interactive bash session on a running container

jboucas-mbpr:~$ docker exec -i -t my_first_container /bin/bash

root@43530985a83d:/# ls
bin  boot  dev  etc  home  jorge.txt  lib  lib64  media  mnt  opt  proc  root
run  sbin  srv  sys  tmp  usr  var
root@43530985a83d:/# exit
exit

## stop a container

jboucas-mbpr:~$ docker stop my_first_container
my_first_container
```

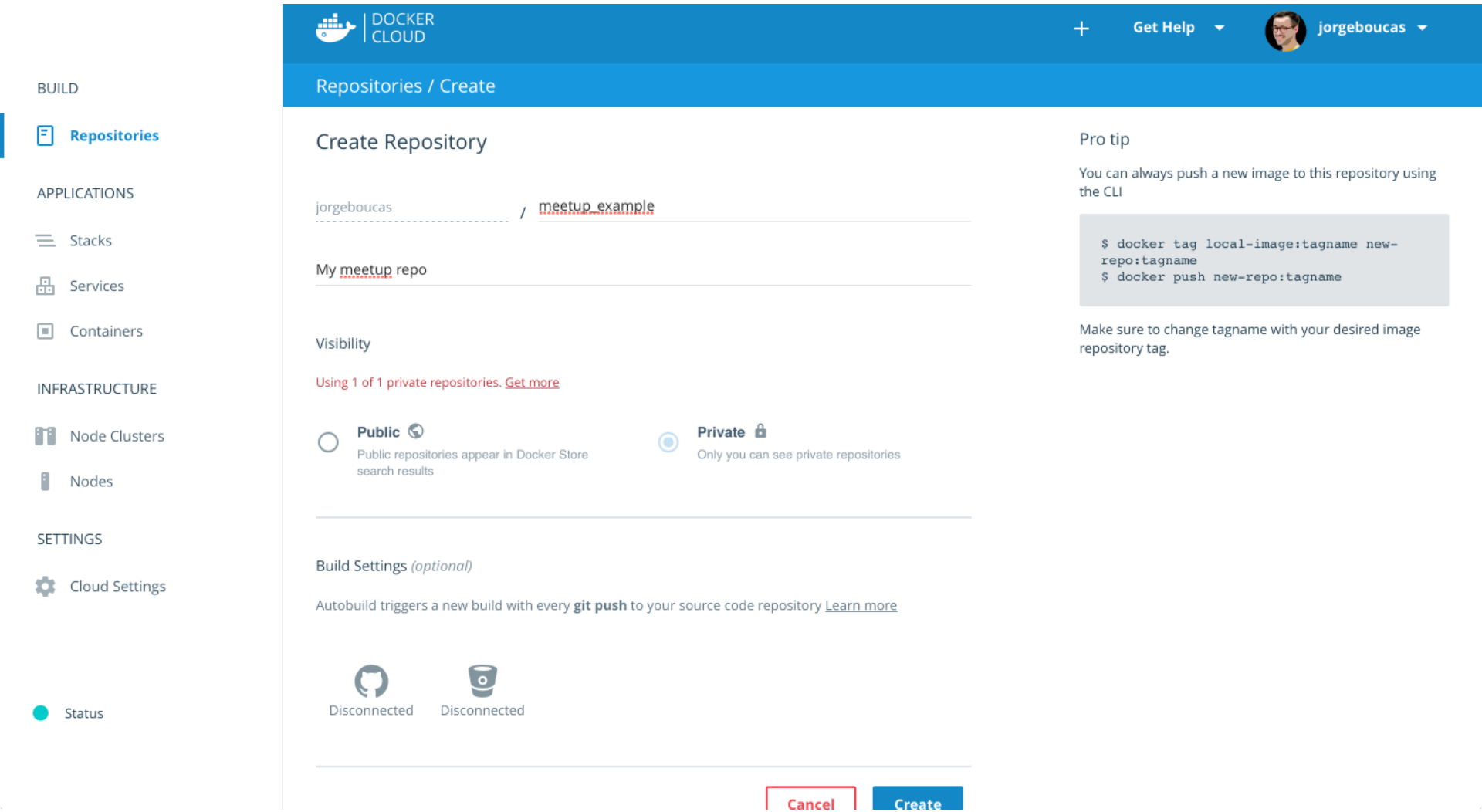
Listing containers and images

```
## list containers

jboucas-mbpr:~$ docker ps -a
CONTAINER ID        IMAGE                  COMMAND                  CREATED
STATUS                      PORTS                    NAMES
43530985a83d        debian                 "/bin/bash"              About an
hour ago   Exited (0) 36 seconds ago
my_first_container
c3ad2c600b24        encode_rnaseq_eclip    "tini -- start-not..."   10 days ago
Up 3 days                   0.0.0.0:8888->8888/tcp   encode_rnaseq_eclip

## list images

jboucas-mbpr:~$ docker images
REPOSITORY                                 TAG                 IMAGE ID
CREATED             SIZE
debian                                     latest              e5599115b6a6
6 days ago          123 MB
encode_rnaseq_eclip                        latest              40aed2032ff1
10 days ago         4.02 GB
mpgagebioinformatics/encode_rnaseq_eclip   latest              8a22f9e82bf6
8 weeks ago         4.02 GB
```

from container to image to cloud

```
## commit a containter into an image

jboucas-mbpr:~$ docker commit my_first_container my_first_image
sha256:686efbb118cdef3a5cda0413a684ab77c5d67645f37c85e7450212fc5d406575

## tag, push and pull your image

jboucas-mbpr:~$ docker tag 686efbb118cd jorgeboucas/meetup_example:v1
jboucas-mbpr:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't
have a Docker ID, head over to https://hub.docker.com to create one.
Username (jorgeboucas): jorgeboucas
Password:
Login Succeeded
```

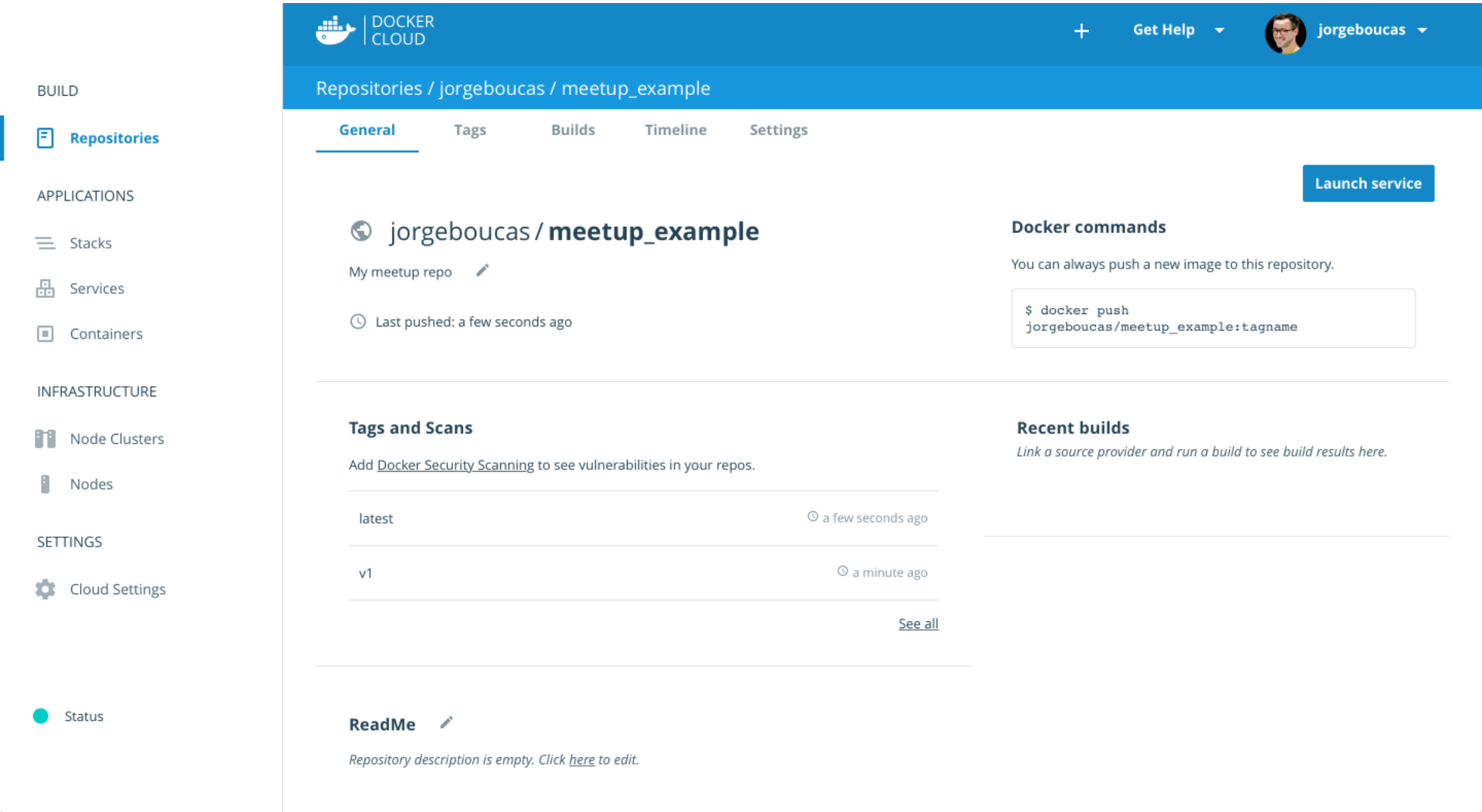
cloud.docker.com



pushing an image

```
jboucas-mbpr:~$ docker push jorgeboucas/meetup_example:v1
The push refers to a repository [docker.io/jorgeboucas/meetup_example]
570a668954b5: Pushed
a2ae92ffcd29: Mounted from library/debian
v1: digest:
sha256:0d4c34cdd21c80c32791c09b5902dfd7c28815a2a990e2c75edc0fb66c3d57fb size:
736

jboucas-mbpr:~$ docker tag 686efbb118cd jorgeboucas/meetup_example:latest

jboucas-mbpr:~$ docker push jorgeboucas/meetup_example:latest
The push refers to a repository [docker.io/jorgeboucas/meetup_example]
570a668954b5: Layer already exists
a2ae92ffcd29: Layer already exists
latest: digest:
sha256:0d4c34cdd21c80c32791c09b5902dfd7c28815a2a990e2c75edc0fb66c3d57fb size:
736
```

cloud.docker.com



removing images

```
## remove images

jboucas-mbpr:~$ docker images
REPOSITORY                                  TAG                 IMAGE ID
CREATED             SIZE
debian                                      latest              e5599115b6a6
6 days ago          123 MB
jorgeboucas/meetup_example                  latest              686efbb118cd
7 hours ago         123 MB
jorgeboucas/meetup_example                  v1                  686efbb118cd
7 hours ago         123 MB

jboucas-mbpr:~$ docker rmi jorgeboucas/meetup_example
Untagged: jorgeboucas/meetup_example:latest

jboucas-mbpr:~$ docker rmi jorgeboucas/meetup_example:v1
Untagged: jorgeboucas/meetup_example:v1
Untagged: jorgeboucas/
meetup_example@sha256:0d4c34cdd21c80c32791c09b5902dfd7c28815a2a990e2c75edc0fb66
c3d57fb
```

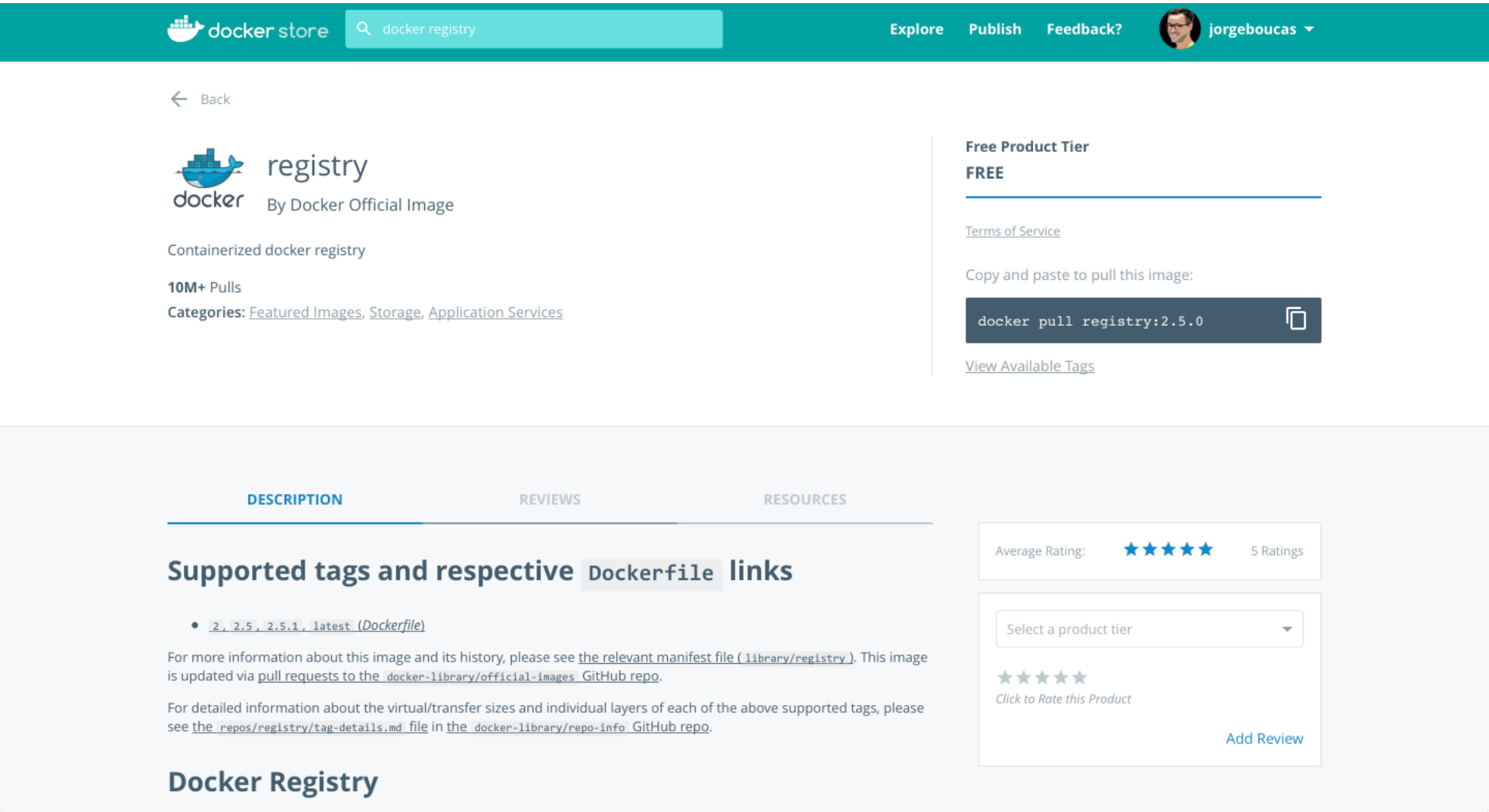
pull your image back

```
jboucas-mbpr:~$ docker images
REPOSITORY                                  TAG                 IMAGE ID
CREATED             SIZE
debian                                      latest              e5599115b6a6
6 days ago          123 MB

## pull your own image and check the content

jboucas-mbpr:~$ docker pull jorgeboucas/meetup_example
Using default tag: latest
latest: Pulling from jorgeboucas/meetup_example
Digest: sha256:0d4c34cdd21c80c32791c09b5902dfd7c28815a2a990e2c75edc0fb66c3d57fb
Status: Downloaded newer image for jorgeboucas/meetup_example:latest

jboucas-mbpr:~$ docker images
REPOSITORY                                  TAG                 IMAGE ID
CREATED             SIZE
jorgeboucas/meetup_example                  latest              686efbb118cd
7 hours ago         123 MB
debian                                      latest              e5599115b6a6
6 days ago          123 MB
```

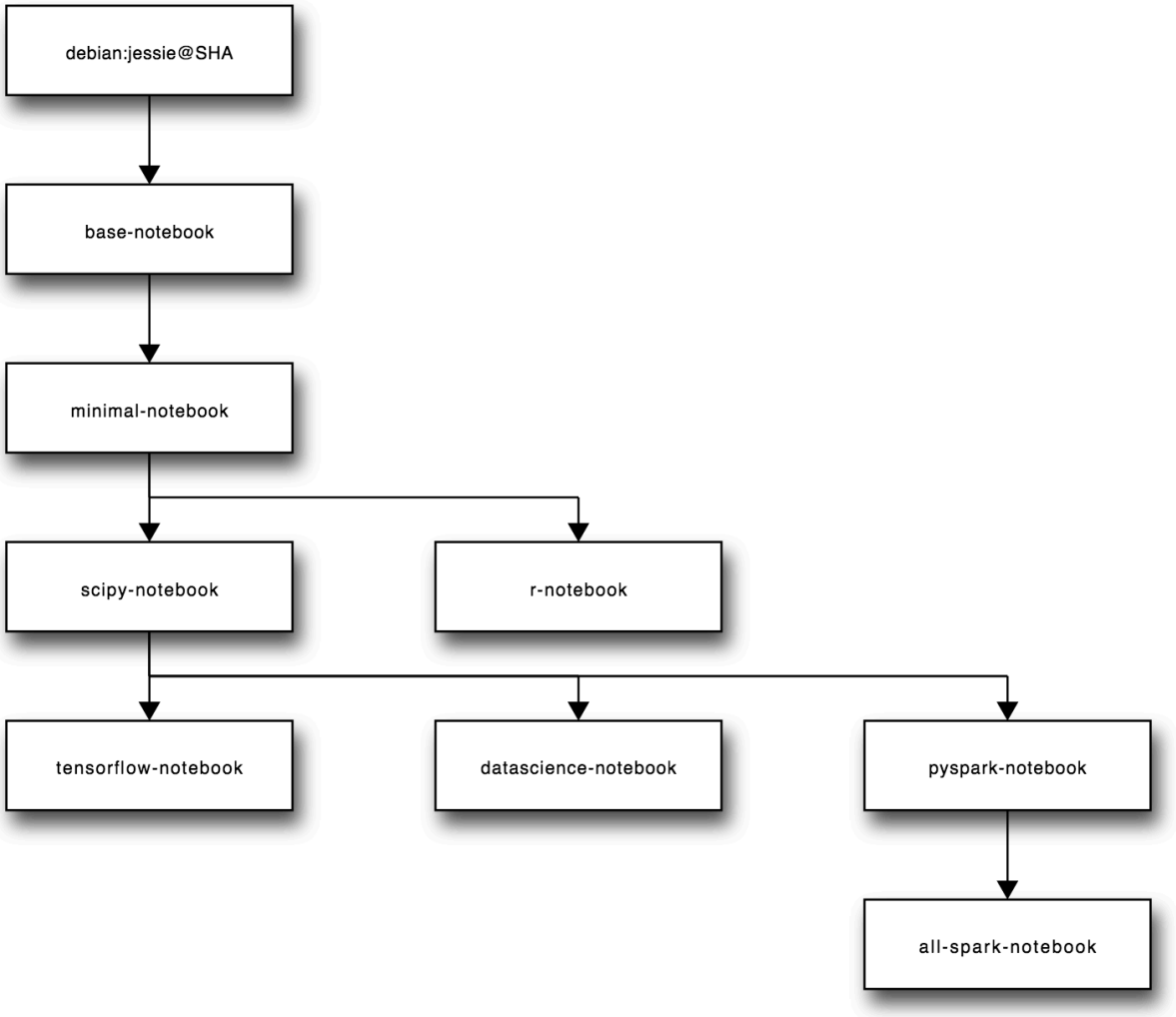
run your image again

```
jboucas-mbpr:~$ docker run --name my_own_second_container -t -i jorgeboucas/
meetup_example /bin/bash

root@cdaa6dcb587b:/# ls
bin  boot  dev  etc  home  jorge.txt  lib  lib64  media  mnt  opt  proc  root
run  sbin  srv  sys  tmp  usr  var
root@cdaa6dcb587b:/# exit
exit
```

Private Docker Registry



github.com/jupyter/docker-stacks



github.com/jupyter/docker-stacks/blob/master/base-notebook/Dockerfile

```
# Copyright (c) Jupyter Development Team.
# Distributed under the terms of the Modified BSD License.

# Debian Jessie image released 2016 May 03.
FROM debian@sha256:f7062cf040f67f0c26ff46b3b44fe036c29468a7e69d8170f37c57f2eec1261b

MAINTAINER Jupyter Project <jupyter@googlegroups.com>

USER root

# Install all OS dependencies for notebook server that starts but lacks all
# features (e.g., download as all possible file formats)
ENV DEBIAN_FRONTEND noninteractive
RUN REPO=http://cdn-fastly.deb.debian.org \
 && echo "deb $REPO/debian jessie main\ndeb $REPO/debian-security jessie/updates main" > /etc/apt/sources.list \
 && apt-get update && apt-get -yq dist-upgrade \
 && apt-get install -yq --no-install-recommends \
    wget \
    bzip2 \
    ca-certificates \
    sudo \
    locales \
 && apt-get clean \
 && rm -rf /var/lib/apt/lists/*

RUN echo "en_US.UTF-8 UTF-8" > /etc/locale.gen && \
    locale-gen

# Install Tini
RUN wget --quiet https://github.com/krallin/tini/releases/download/v0.10.0/tini && \
    echo "1361527f39190a7338a0b434bd8c88ff7233ce7b9a4876f3315c22fce7eca1b0 *tini" | sha256sum -c - && \
    mv tini /usr/local/bin/tini && \
    chmod +x /usr/local/bin/tini
```

github.com/jupyter/docker-stacks/blob/master/base-notebook/Dockerfile

```
34
35  # Configure environment
36  ENV CONDA_DIR /opt/conda
37  ENV PATH $CONDA_DIR/bin:$PATH
38  ENV SHELL /bin/bash
39  ENV NB_USER jovyan
40  ENV NB_UID 1000
41  ENV HOME /home/$NB_USER
42  ENV LC_ALL en_US.UTF-8
43  ENV LANG en_US.UTF-8
44  ENV LANGUAGE en_US.UTF-8
45
46  # Create jovyan user with UID=1000 and in the 'users' group
47  RUN useradd -m -s /bin/bash -N -u $NB_UID $NB_USER && \
48      mkdir -p $CONDA_DIR && \
49      chown $NB_USER $CONDA_DIR
50
51  USER $NB_USER
52
53  # Setup jovyan home directory
54  RUN mkdir /home/$NB_USER/work && \
55      mkdir /home/$NB_USER/.jupyter && \
56      echo "cacert=/etc/ssl/certs/ca-certificates.crt" > /home/$NB_USER/.curlrc
57
58  # Install conda as jovyan
59  RUN cd /tmp && \
60      mkdir -p $CONDA_DIR && \
61      wget --quiet https://repo.continuum.io/miniconda/Miniconda3-4.2.12-Linux-x86_64.sh && \
62      echo "c59b3dd3cad550ac7596e0d599b91e75d88826db132e4146030ef471bb434e9a *Miniconda3-4.2.12-Linux-x86_64.sh" | sha256sum -c - && \
63      /bin/bash Miniconda3-4.2.12-Linux-x86_64.sh -f -b -p $CONDA_DIR && \
64      rm Miniconda3-4.2.12-Linux-x86_64.sh && \
65      $CONDA_DIR/bin/conda config --system --add channels conda-forge && \
66      $CONDA_DIR/bin/conda config --system --set auto_update_conda false && \
67      conda clean -tipsy
```

github.com/jupyter/docker-stacks/blob/master/base-notebook/Dockerfile

```
# Install Jupyter Notebook and Hub
RUN conda install --quiet --yes \
    'notebook=4.3*' \
    jupyterhub=0.7 \
    && conda clean -tipsy

USER root

EXPOSE 8888
WORKDIR /home/$NB_USER/work

# Configure container startup
ENTRYPOINT ["tini", "--"]
CMD ["start-notebook.sh"]

# Add local files as late as possible to avoid cache busting
COPY start.sh /usr/local/bin/
COPY start-notebook.sh /usr/local/bin/
COPY start-singleuser.sh /usr/local/bin/
COPY jupyter_notebook_config.py /home/$NB_USER/.jupyter/
RUN chown -R $NB_USER:users /home/$NB_USER/.jupyter

# Switch back to jovyan to avoid accidental container runs as root
USER $NB_USER
```

https://github.com/krallin/tini/issues/8
https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/
https://docs.docker.com/engine/reference/builder/

github.com/mpg-age-bioinformatics/ ENCODE_RNAseq_eCLIP

ENCODE_RNAseq_eCLIP

This is a *Jupyter* notebook using *Python*, *R* and *bash* to integrate RNAseq and eCLIP data from ENCODE. It uses an RNAseq data set of KHSRP-shRNA on K562 cells and respective shRNA-mock control. It uses as well an eCLIP data set on KHSRP.

This pipeline comes with a *Docker* image containing all required software.

Installation

You will need to have Docker and git installed.

Start by cloning this repository into your home directory:

```
cd ~/
git clone https://github.com/mpg-age-bioinformatics/ENCODE_RNAseq_eCLIP
```

Then build the Docker image:

```
cd ~/ENCODE_RNAseq_eCLIP
sudo docker build -t encode_rnaseq_eclip .
```

github.com/mpg-age-bioinformatics/ENCODE_RNAseq_eCLIP

Usage

The analysis pipeline was tested on a Docker image running with 8gb of memmory and 4 cpus.

Start your container with:

```
sudo docker run -d -p 8888:8888 \
-e GRANT_SUDO=yes -e NB_UID=1000 --user root \
-e PASSWORD="YOURPASS" -e USE_HTTPS=yes \
-v ~/ENCODE_RNAseq_eCLIP/notebooks:/home/jovyan/work/notebooks \
-v ~/ENCODE_RNAseq_eCLIP/results:/home/jovyan/work/results \
--name encode_rnaseq_eclip \
-i -t encode_rnaseq_eclip
```

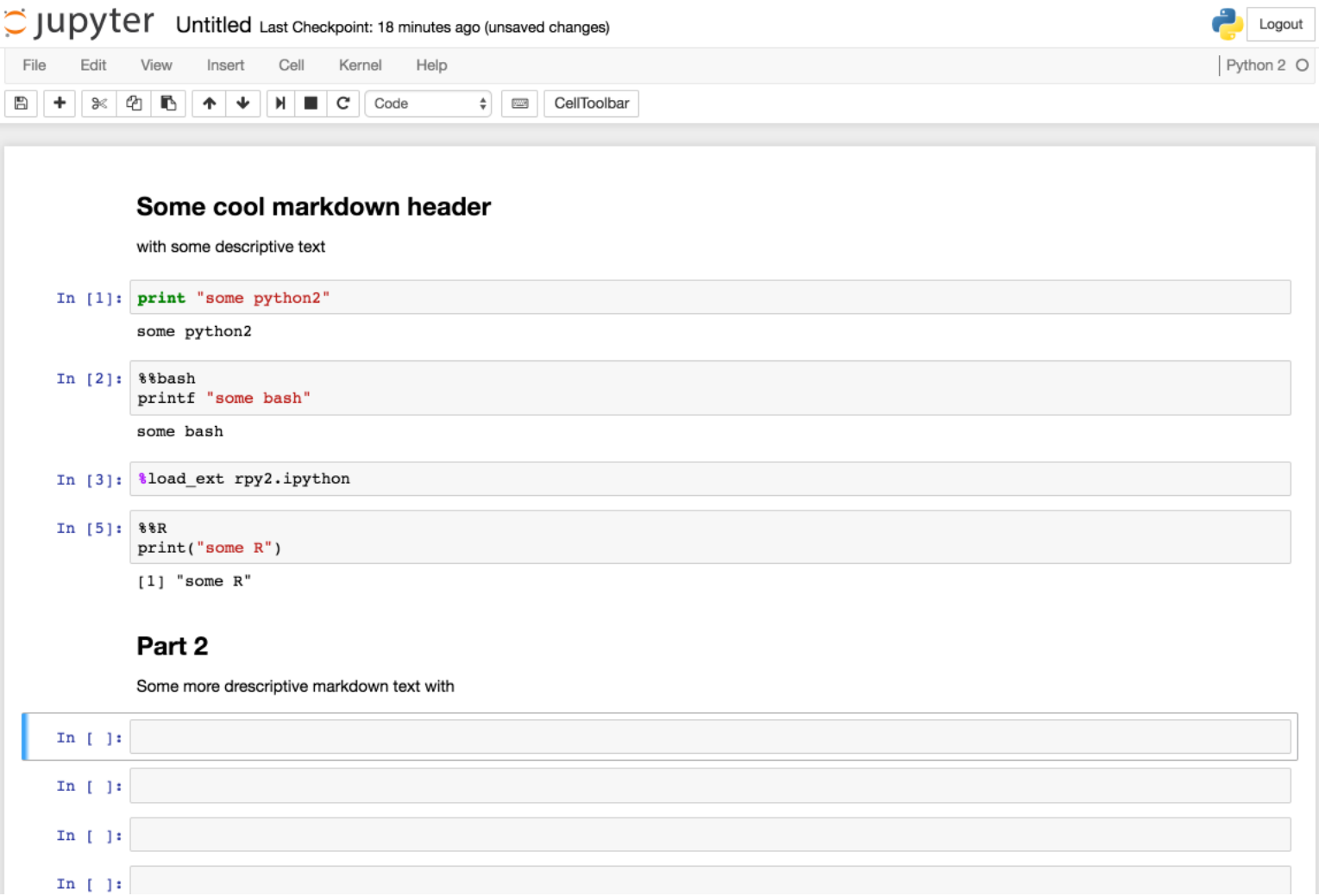
Instead of `"YOURPASS"` type in a password of your choice.

Then, on your web browser connect to https://localhost:8888.

You will be requested to accepted the self-signed certificate and to give in the password you used in `"YOURPASS"`.

Jupyter, https://localhost:8888

Jupyter, https://localhost:8888



Jupyter, https://localhost:8888

jupyter Untitled Last Checkpoint: 3 minutes ago (autosaved) Logout

File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

In [21]:
```
import pandas as pd
# Make a pandas DataFrame
df = pd.DataFrame({'Alphabet': ['a', 'b', 'c', 'd','e', 'f', 'g', 'h','i'],
                   'A': [4, 3, 5, 2, 1, 7, 7, 5, 9],
                   'B': [0, 4, 3, 6, 7, 10,11, 9, 13],
                   'C': [1, 2, 3, 1, 2, 3, 1, 2, 3]})
df.head()
```

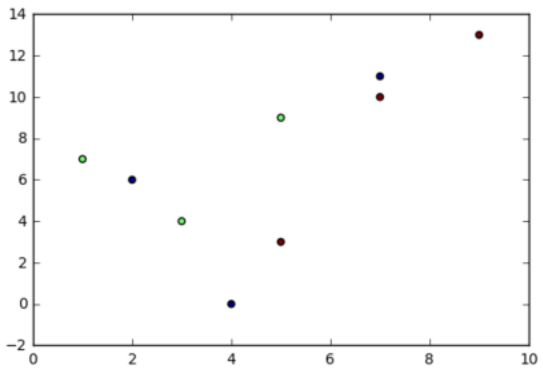
Out[21]:

	A	Alphabet	B	C
0	4	a	0	1
1	3	b	4	2
2	5	c	3	3
3	2	d	6	1
4	1	e	7	2

In [18]:
```
%matplotlib inline
print "Inline graphics"
plt.scatter(df["A"],df["B"],c=df["C"])
```

```
Inline graphics
```

Out[18]:
```
<matplotlib.collections.PathCollection at 0x7f3df7b4b8d0>
```



Jupyter, https://localhost:8888

jupyter Untitled Last Checkpoint: 5 minutes ago (autosaved) Logout

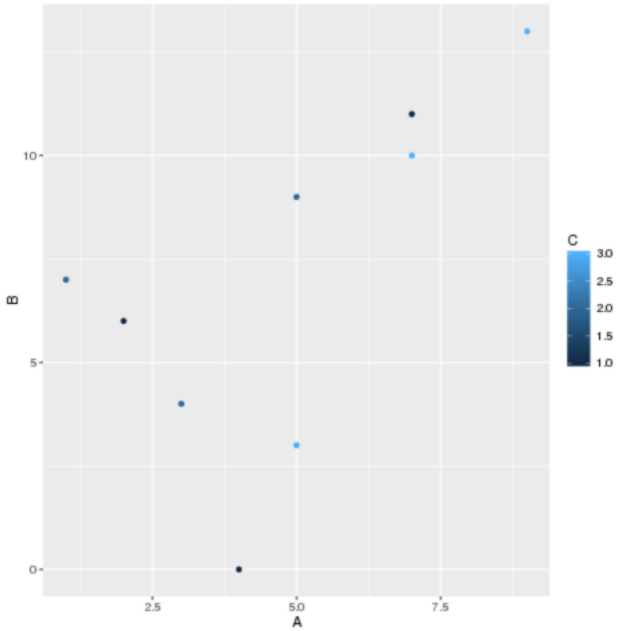
File Edit View Insert Cell Kernel Help Python 2

Code CellToolbar

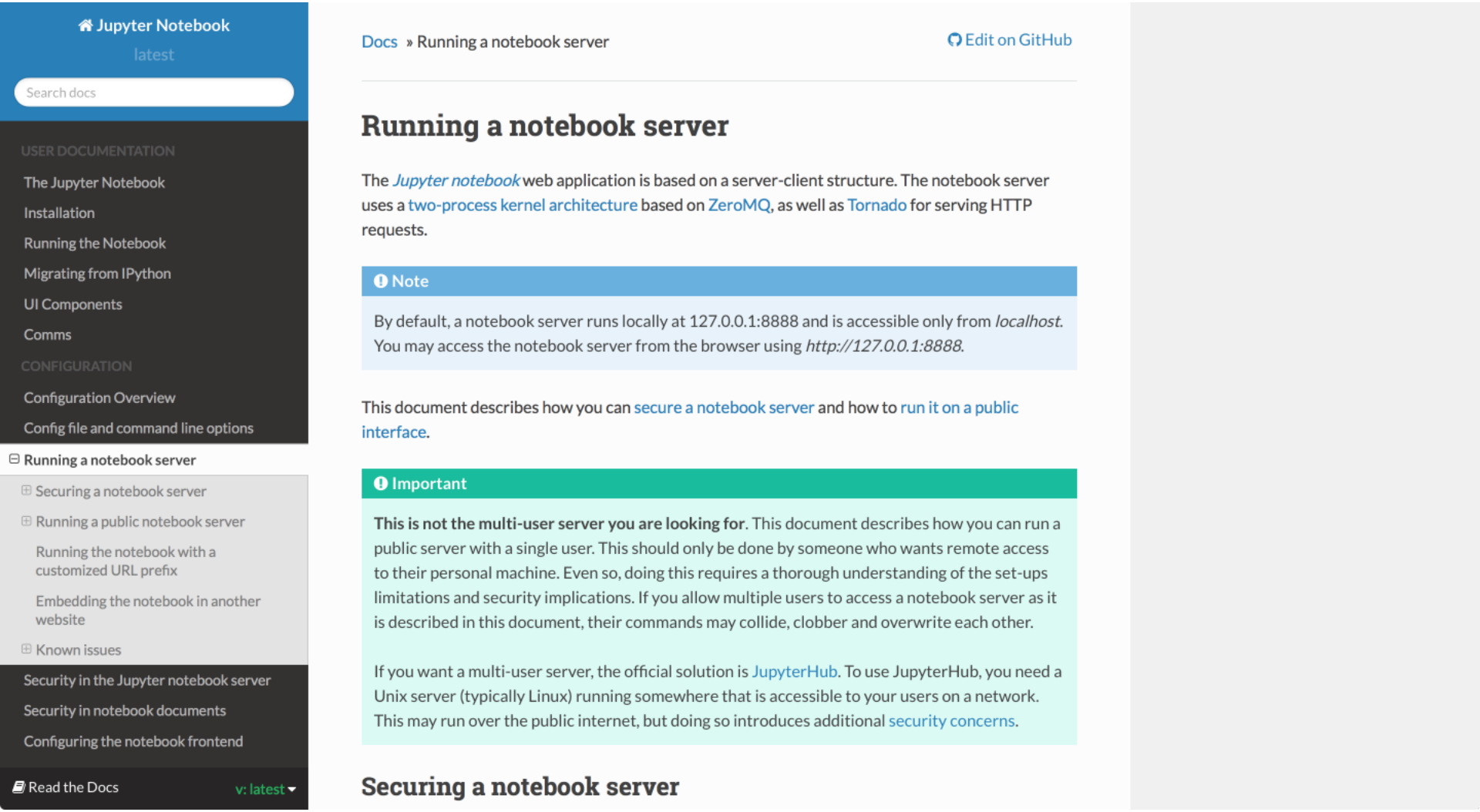
Autotranslation

In [19]:
```
%R -i df
```

In [20]:
```
%%R
require(ggplot2)
ggplot(data=df) + geom_point(aes(x=A, y=B, color=C))
```



Jupyter on a remote server



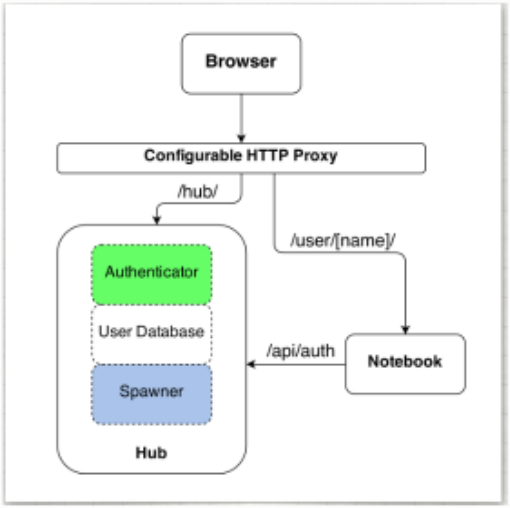
multi-user Jupyter server

JupyterHub

With JupyterHub you can create a **multi-user Hub** which spawns, manages, and proxies multiple instances of the single-user Jupyter notebook server. Due to its flexibility and customization options, JupyterHub can be used to serve notebooks to a class of students, a corporate data science group, or a scientific research group.

Three subsystems make up JupyterHub:

- a multi-user **Hub** (tornado process)
- a **configurable http proxy** (node-http-proxy)
- multiple **single-user Jupyter notebook servers** (Python/IPython/tornado)

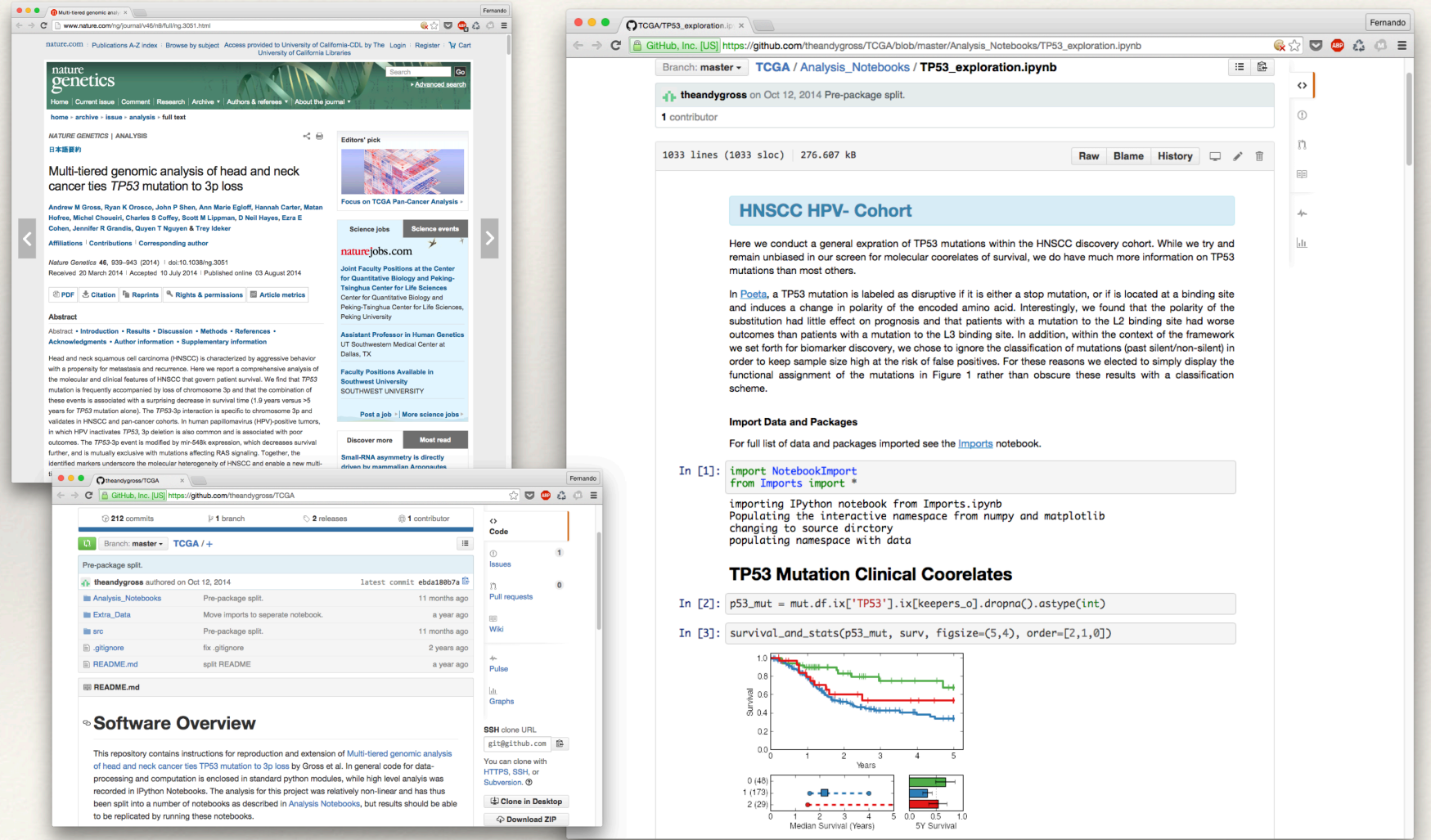


JupyterHub's basic flow of operations includes:

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

For convenient administration of the Hub, its users, and Services (added in version 7.0), JupyterHub also provides a REST API.

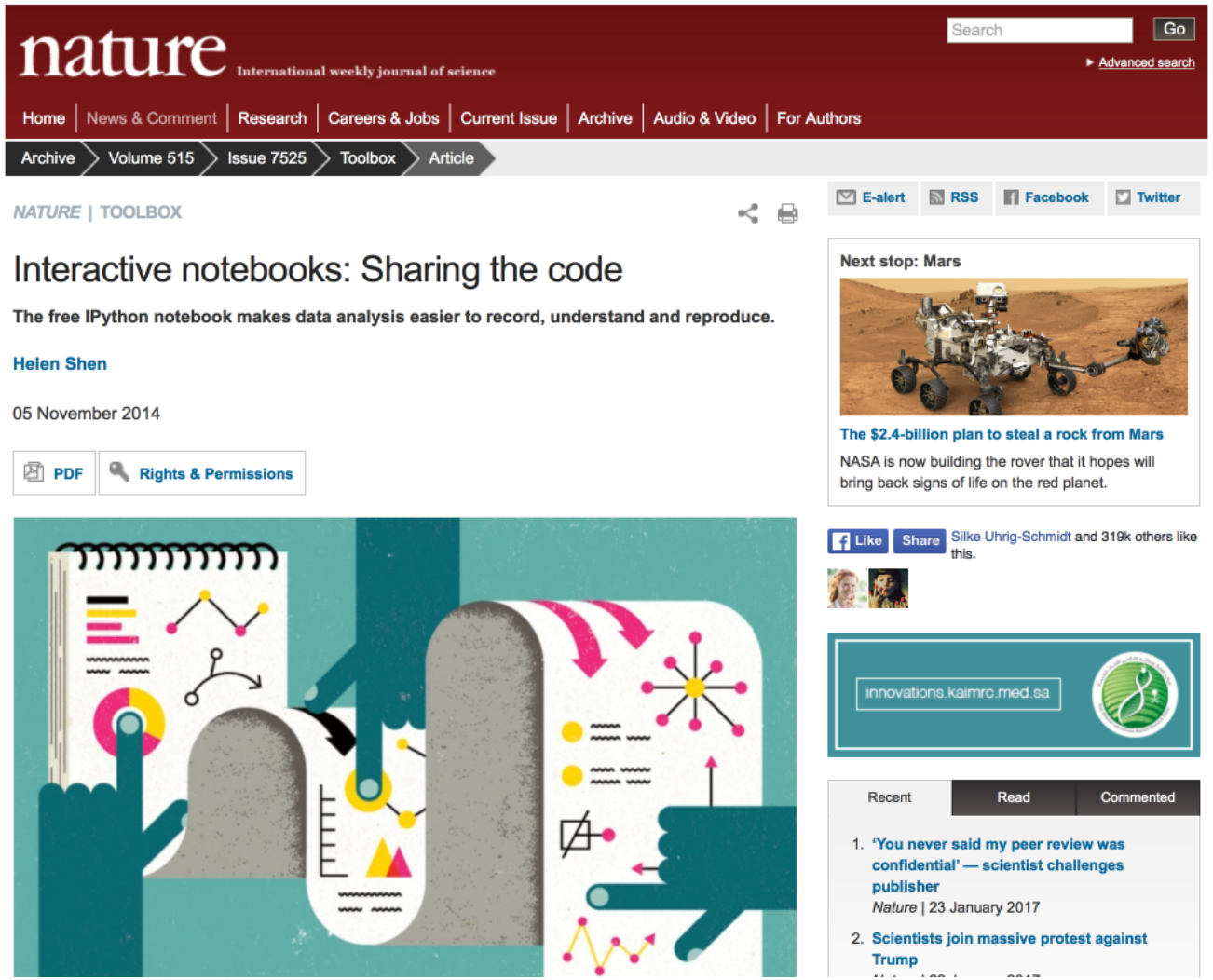
Notebooks on Github: the “actual scolarship”



https://figshare.com/articles/Project_Jupyter_interactive_computing_in_the_context_of_modern_science_gateways/4495976

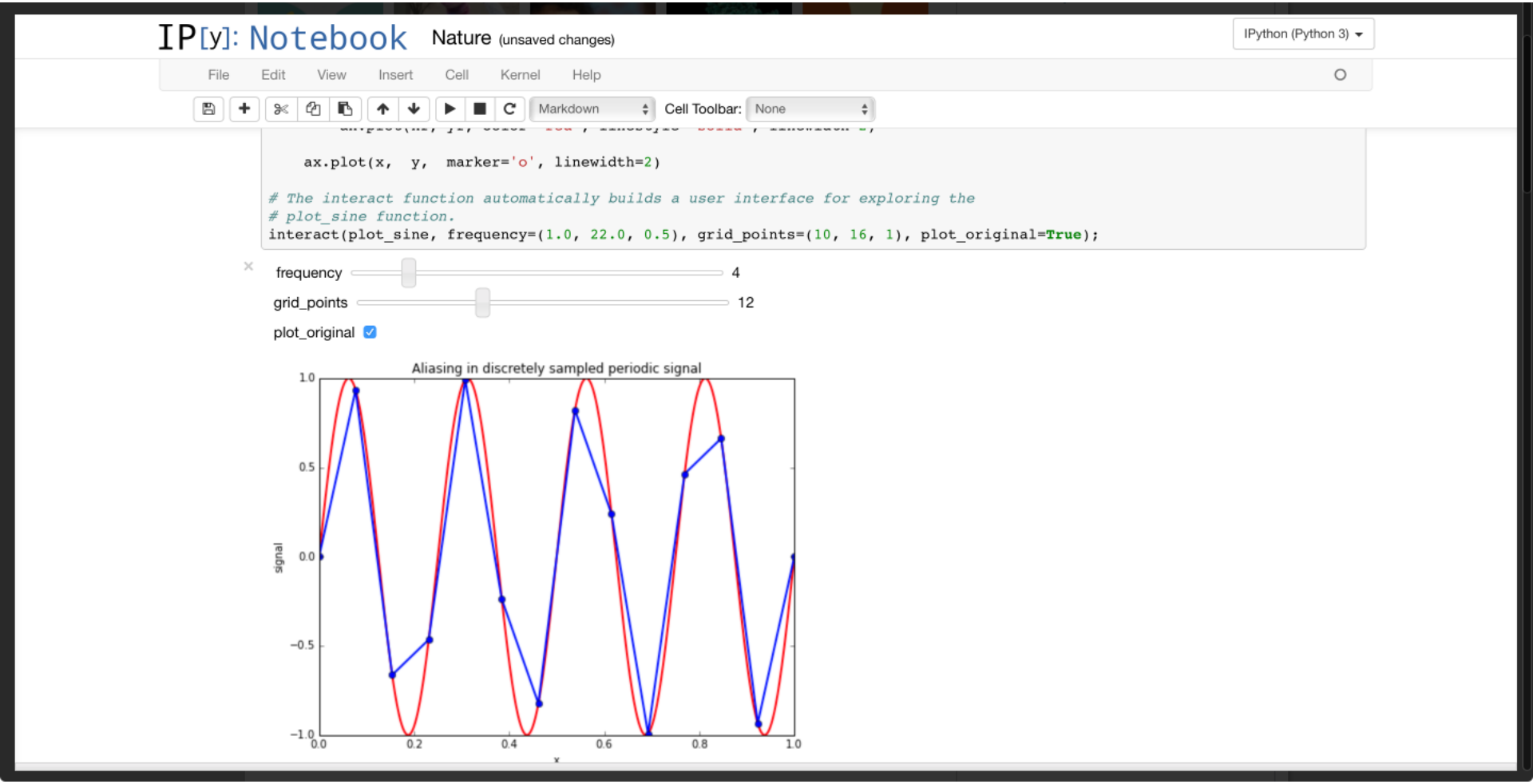
Interactive notebooks: Sharing the code

http://www.nature.com/news/interactive-notebooks-sharing-the-code-1.16261

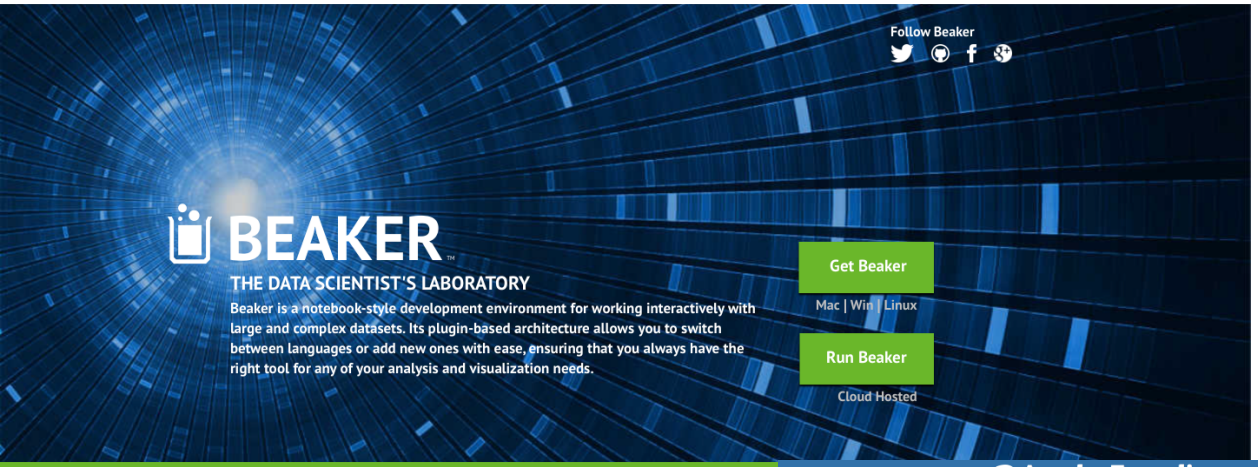


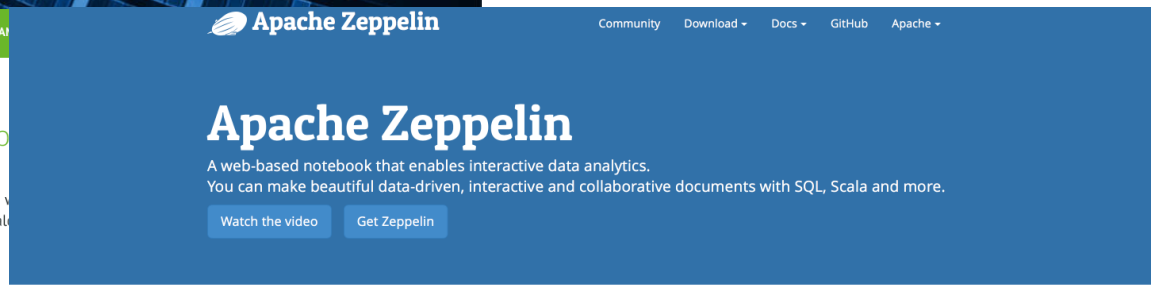
Executable papers

https://figshare.com/articles/
Project_Jupyter_interactive_computing_in_the_context_of_modern_science_gateways/4495976



Alternatives to Jupyter





Beaker Lab is currently in Beta and so we are offering complete use of the system for free.

Alternatives to Docker

- **VAGRANT :** https://www.vagrantup.com
- **Open Container Initiative (OCI)** : https://www.opencontainers.org
- **Kubernetes :** https://kubernetes.io
- **CoreOS and rkt :** https://coreos.com/rkt/
- **Apache Mesos and Mesosphere :** http://mesos.apache.org
- **Canonical and LXD :** https://www.ubuntu.com/cloud/lxd

http://searchcloudapplications.techtarget.com/tip/Five-development-containers-to-consider-that-arent-Docker

Containers for HPC

- **Singularity: http://singularity.lbl.gov (sudo for Docker imports ?)**

Singularity natively supports InfiniBand, Lustre, and works seamlessly with all resource managers (e.g. SLURM, Torque, SGE, etc.) because it works like running any other command on the system.

```
$ singularity exec /tmp/Demo.img xterm

$ singularity exec /tmp/Demo.img python script.py

$ singularity exec /tmp/Demo.img python < /path/to/python/script.py

$ cat /path/to/python/script.py | singularity exec /tmp/Demo.img python

$ mpirun -np X singularity exec /path/to/container.img /usr/bin/
mpi_program_inside_container (mpi program args)
```

Containers for HPC

- **Shifter:** https://github.com/NERSC/shifter

With SLURM plugin, spank
https://github.com/SchedMD/slurm/blob/master/slurm/spank.h

```
```
#!/bin/bash
#SBATCH --image=dmjacobsen/mpitest
#SBATCH --volume=/scratch/dmj/i:/input
#SBATCH --volume=/scratch/dmj/o:/output
#SBATCH -N 250
```
srun shifter /app


shifterimg pull debian:latest # instead of docker pull
shifter --image=debian:latest bash -login # for interactive session
```

Containers for HPC

- **Shifter:** https://github.com/NERSC/shifter

With SLURM plugin, spank
https://github.com/SchedMD/slurm/blob/master/slurm/spank.h

```
```
#!/bin/bash
#SBATCH --image=dmjacobsen/mpitest
#SBATCH --volume=/scratch/dmj/i:/input
#SBATCH --volume=/scratch/dmj/o:/output
#SBATCH -N 250
```
srun shifter /app


shifterimg pull debian:latest # instead of docker pull
shifter --image=debian:latest bash -login # for interactive session
```

END

bioinformatics@age.mpg.de

https://mpg-age-bioinformatics.github.io