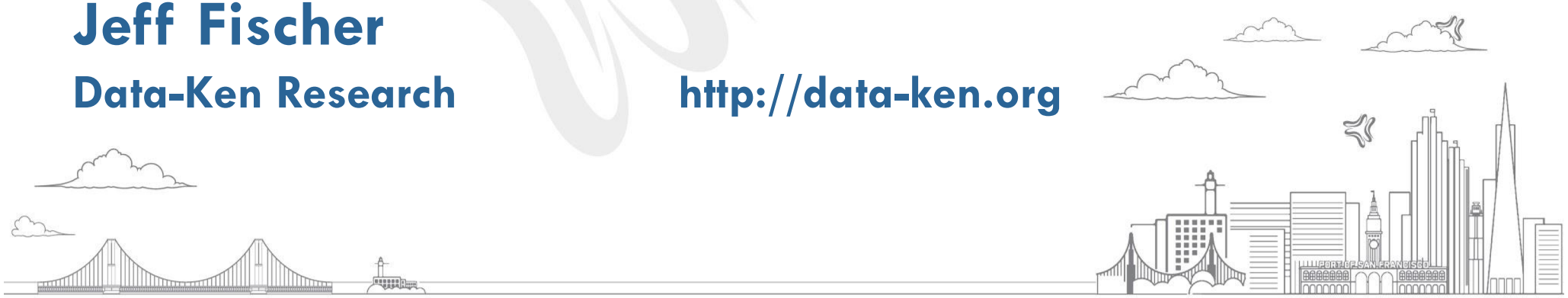
PyBay2018

DOCKER FOR DATA SCIENTISTS:

SIMPLIFY YOUR WORKFLOW AND AVOID PITFALLS

Jeff Fischer

Data-Ken Research **http://data-ken.org**



3rd Annual Regional
Python Conference

August 16 - 19, 2018 | San Francisco, CA

Docker: What and Why?

What is Docker?

- Uses containers to deploy isolated applications on Linux
- Tools for creating, sharing, and building upon layered application stacks
- Forms the basis for more advanced services such as Kubernetes

As a Data Scientist, why should you care?

- Focus on your work, not on maintaining complex software dependencies
- ***Reproduce*** your experiments
- Share and collaborate with your peers
- Build on others

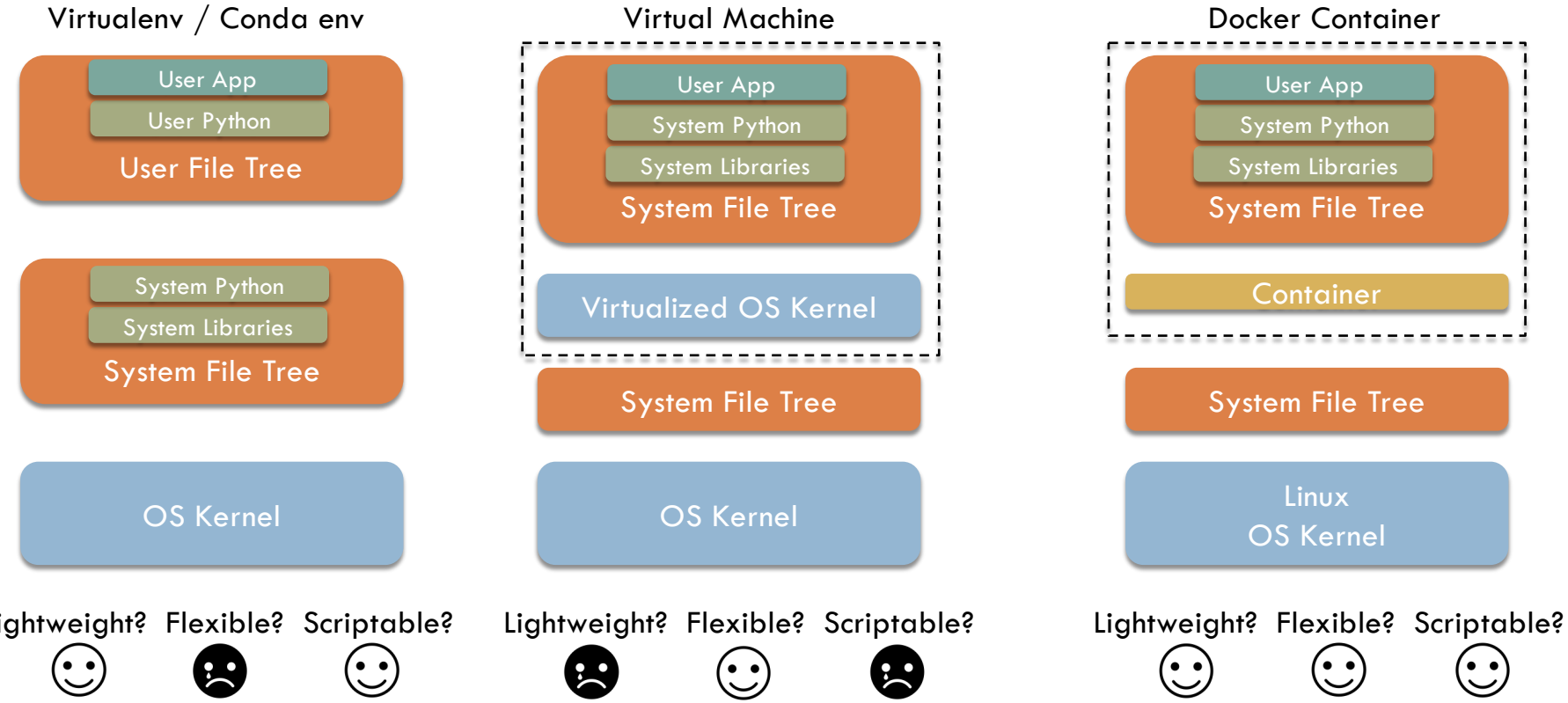
Obligatory picture of a container ship

Image from National Ocean Service, CC By 2.0

Talk Organization and Resources

- This is based on my consulting work with data science teams
- Talk organized around specific tasks ("workflows")
- More resources:
  - Detailed tutorial on my blog: https://data-ken.org/docker-for-data-scientists-part1.html
  - Code up on GitHub: https://github.com/jfischer/docker-for-data-scientist-examples

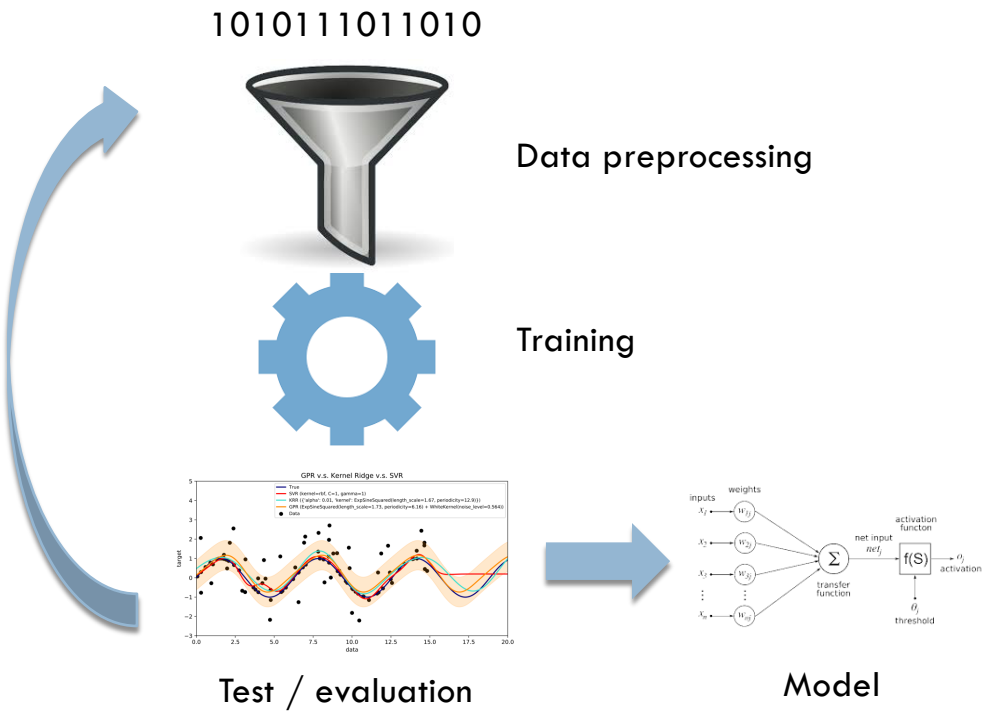
Comparing Deployment and Isolation Approaches



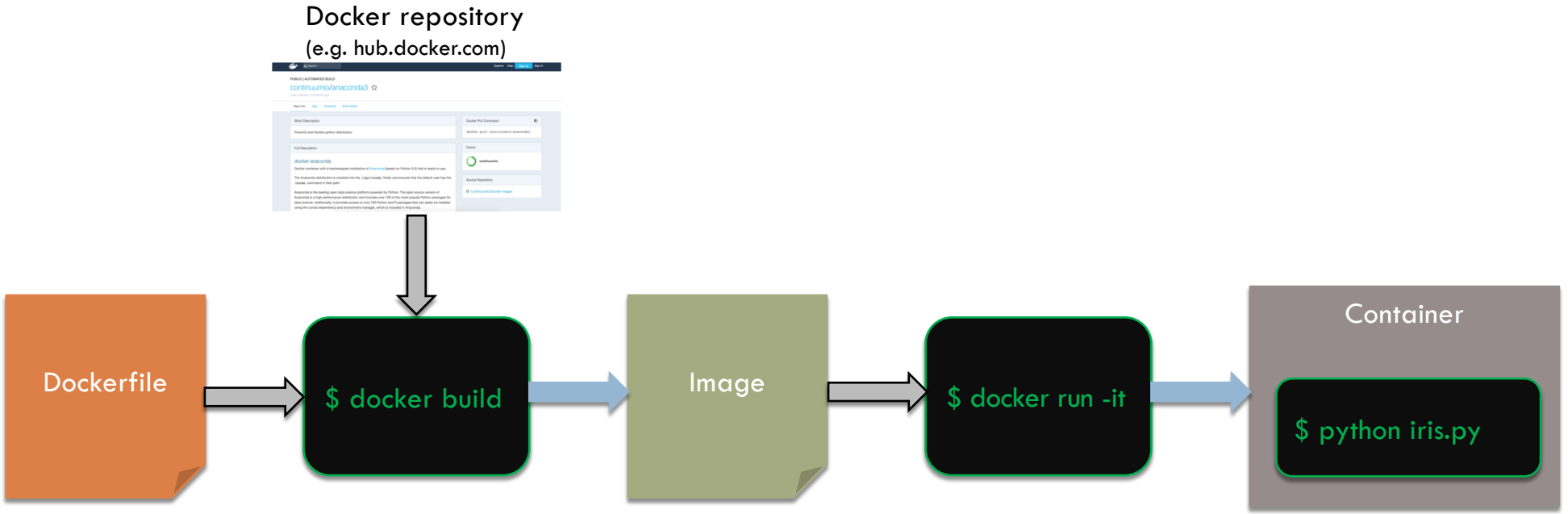
Workflows

In a container…

1. Run a Python script
2. Run a development session
3. Run a Jupyter notebook
4. Run a database



Run a Python Script: Overview



Run a Python Script: iris.py


```
!/usr/bin/env python3
# Scikit-learn Iris example

from sklearn import datasets, svm
from sklearn.model_selection import train_test_split

# load the data
iris = datasets.load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Train a Support Vector Classifier
clf = svm.SVC()
print(clf)
clf.fit(X_train, y_train)

# Classify the test data
accuracy = clf.score(X_test, y_test)
print("Accuracy is %0.3f" % accuracy)
```

Run a Python Script: Details

Dockerfile

```
FROM continuumio/anaconda3:latest
RUN mkdir /scripts
COPY iris.py /scripts

CMD /bin/bash
```

Shell commands

```
docker pull continuumio/anaconda3

docker build -t pybay-workflow-1 .

docker run -it --rm \
  pybay-workflow-1:latest \
  /opt/conda/bin/python /scripts/iris.py
```

Demo: Run a Python Script

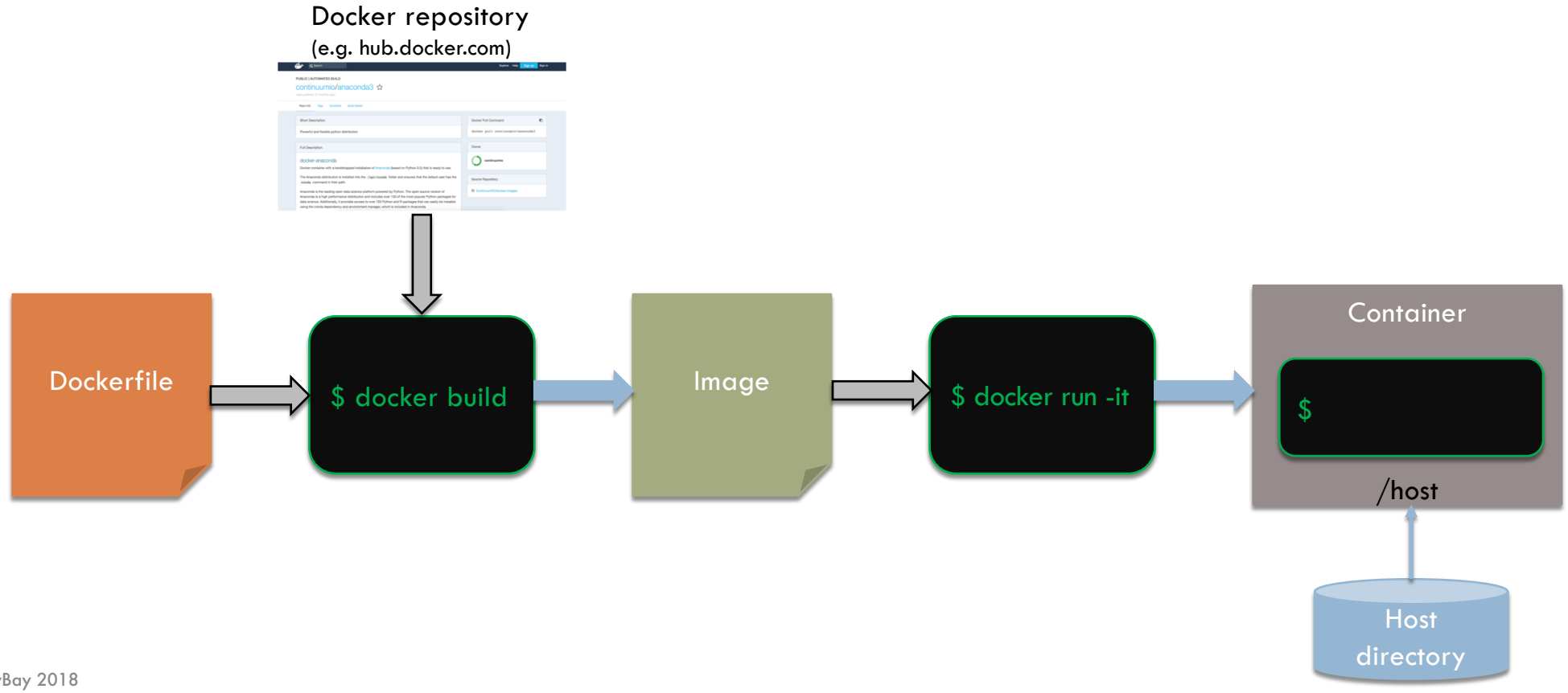
workflow-1

Pitfall: Data Changes in Container

- **Problem:** I started my container, but the changes I made inside it are gone!
- **Causes:**
  - `--rm` creates an ephemeral container
  - Easy to mix up `run` and `start` commands
  - Easy to lose track of which container you made changes
- **Fix:**
  - Do not make any code/data changes in the container!
  - Mount the host filesystem and make the changes there (see next workflow)

Image from Wikipedia, low res fair use

Run a Development Session: Overview



Run a Development Session: Details

Dockerfile

```
FROM continuumio/anaconda3:latest
RUN apt-get -y -q install vim-tiny
VOLUME /host
WORKDIR /host
CMD /bin/bash
```

Shell commands

```
docker pull continuumio/anaconda3

docker build -t pybay-workflow-2a .

docker run -it --rm \
 --volume `pwd`:/host \
 pybay-workflow-2a:latest /bin/bash
```

Mount the current directory as /host

Demo: Run a Development Session

workflow-2a

Pitfall: Access Permissions

- **Problem:** my container on Linux cannot write to files in my home directory!
- **Cause:** Enterprise Edition enables user remapping and other security features
- **Fix:** map to host user in `docker run` command – see blog for details

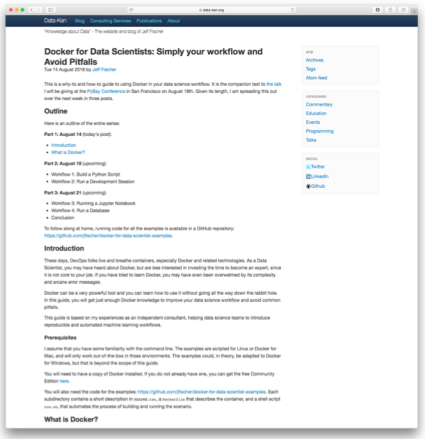
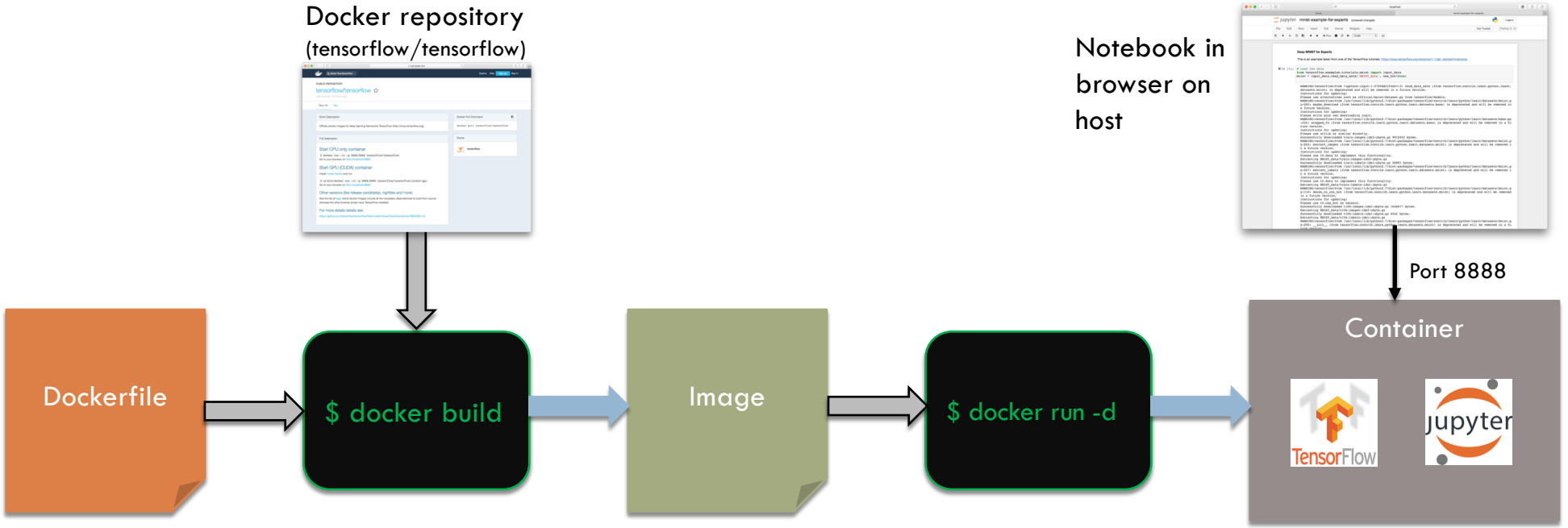


Image from Wikipedia, low res fair use

Run a Jupyter Notebook: Overview



Run a Jupyter Notebook: Details

Dockerfile

```
FROM tensorflow/tensorflow:latest
COPY mnist-example-for-experts.ipynb /notebooks
ENV PASSWORD test

WORKDIR "/notebooks"

CMD ["/run_jupyter.sh", "--allow-root"]
```

Shell commands

```
docker pull tensorflow/tensorflow

docker build -t pybay-workflow-3a .

docker run -d -p 8888:8888 \
 --name workflow-3a-container \
 pybay-workflow-3a:latest
```

Demo: Run a Jupyter Notebook

workflow-3a

Pitfall: GPU Access

- **Problem:** I want to run a GPU-enabled application, but my container does not see my GPU!
- **Cause:** you need a special plugin-from Nvidia
- **Fix:** install `nvidia-docker`, etc. – see blog for details

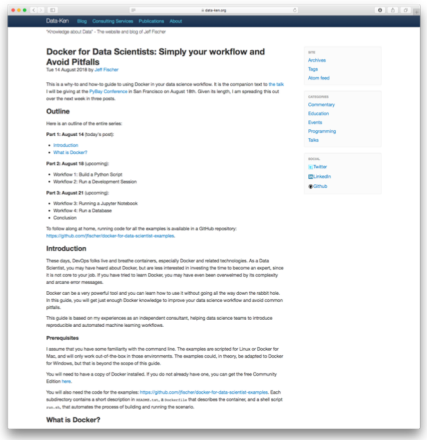
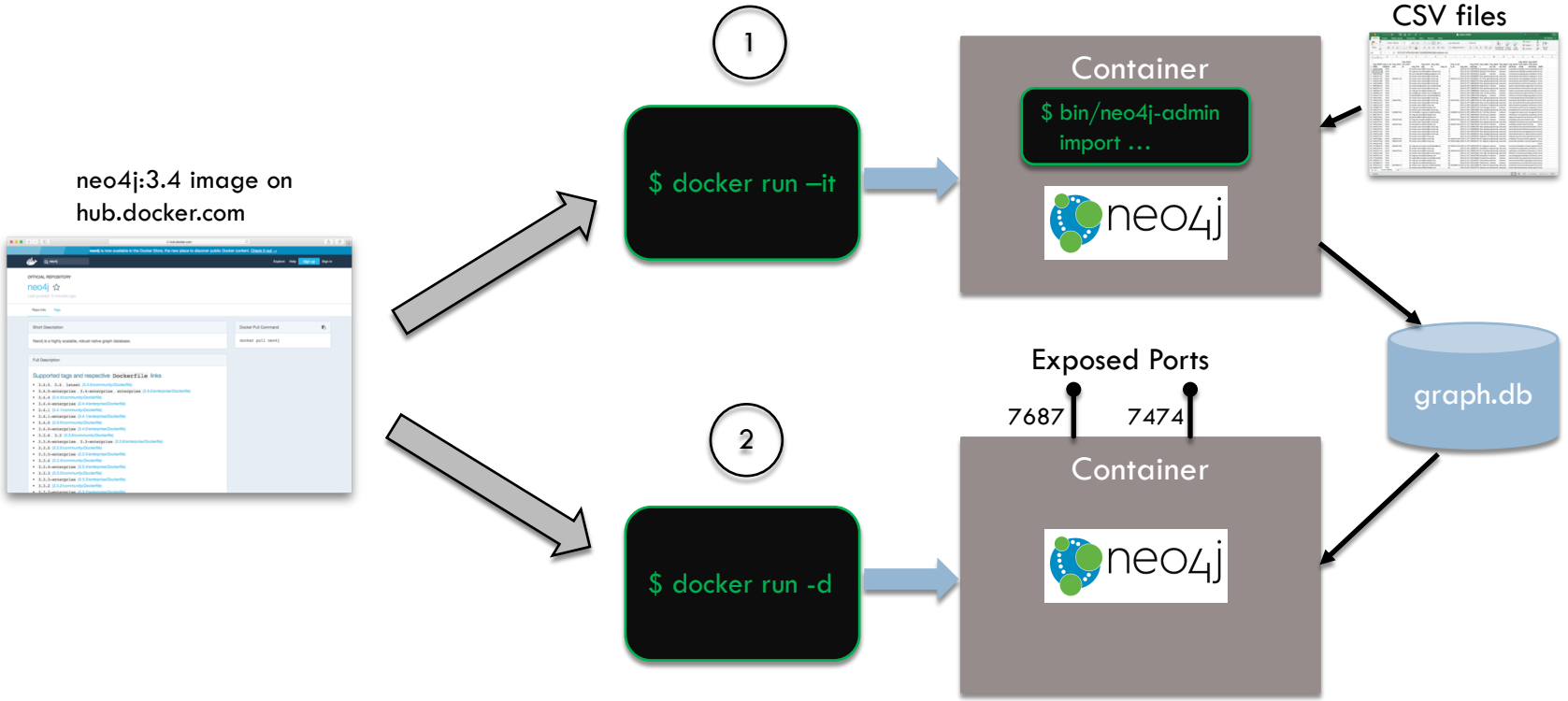


Image from Wikipedia, low res fair use

Run a Database: Overview



See workflow-4 in code repository for details

Demo: Run a Database

workflow-4

Summary

- Use Docker when:
  1. You have a complex stack to maintain
  2. You need to collaborate
  3. You need to run on multiple machines

- Design patterns:
  1. Containers for computation and immutable state
  2. Treat containers as a cheap, throwaway resource
  3. Integrate into your personal workflow and automation

- Contact me if you have questions:
  - Email: jeff@data-ken.org
  - Blog: http://data-ken.org
  - LinkedIn: https://www.linkedin.com/in/fischerjeff/
  - Twitter: @fischer_jeff

Thank you!

Get your work done and enjoy nature.