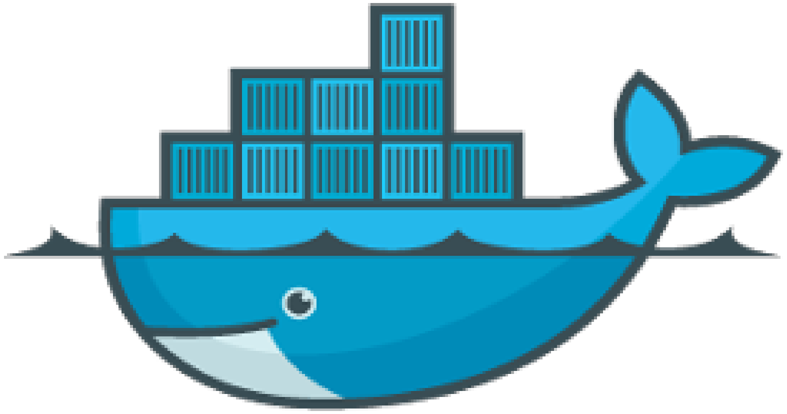


docker

Do any of these sound familiar

- It worked fine on my machine
- I set that up months ago but can't remember the details
- I installed so much stuff trying to get it to work I can't really say which are actually required
- It only works on a linux machine
- I just want to test it without installing
- I want to test/use different versions
- Set-up is too complicated to explain in a paper

Possible solutions

- Hand holding support
- Very detailed documentation
- Virtual machine
- Docker

Virtual Machines

- "is an emulation of a particular computer system"[1]
- Completely separate
- Full set of resources (or as much as possible)
- Current set-up can be saved, copied and shared
- No central repositories of images
- Black-box on how it was set-up
- Full GUI support
- Ideal for working on a host operating system/

[1]https://en.wikipedia.org/w/index.php?titlatorle=Virtual_machine&oldid=669500805

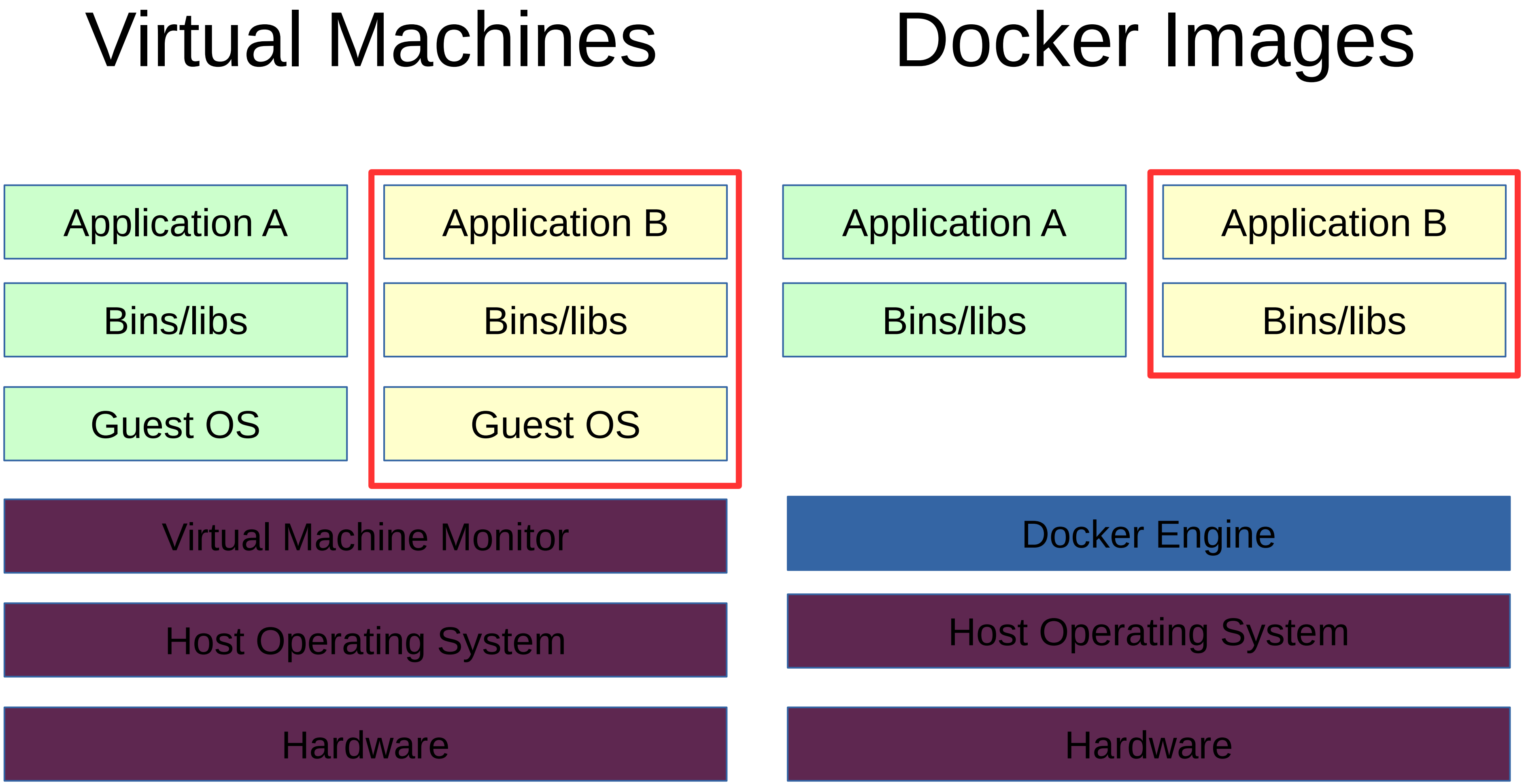
How Docker describes itself

- Build, Ship, Run
- An open platform for distributed applications for developers and sysadmins
- Ship Applications Faster and Easier
- Application Portability and Infrastructure Flexibility
- Dynamically Update, Change and Scale Apps

Docker

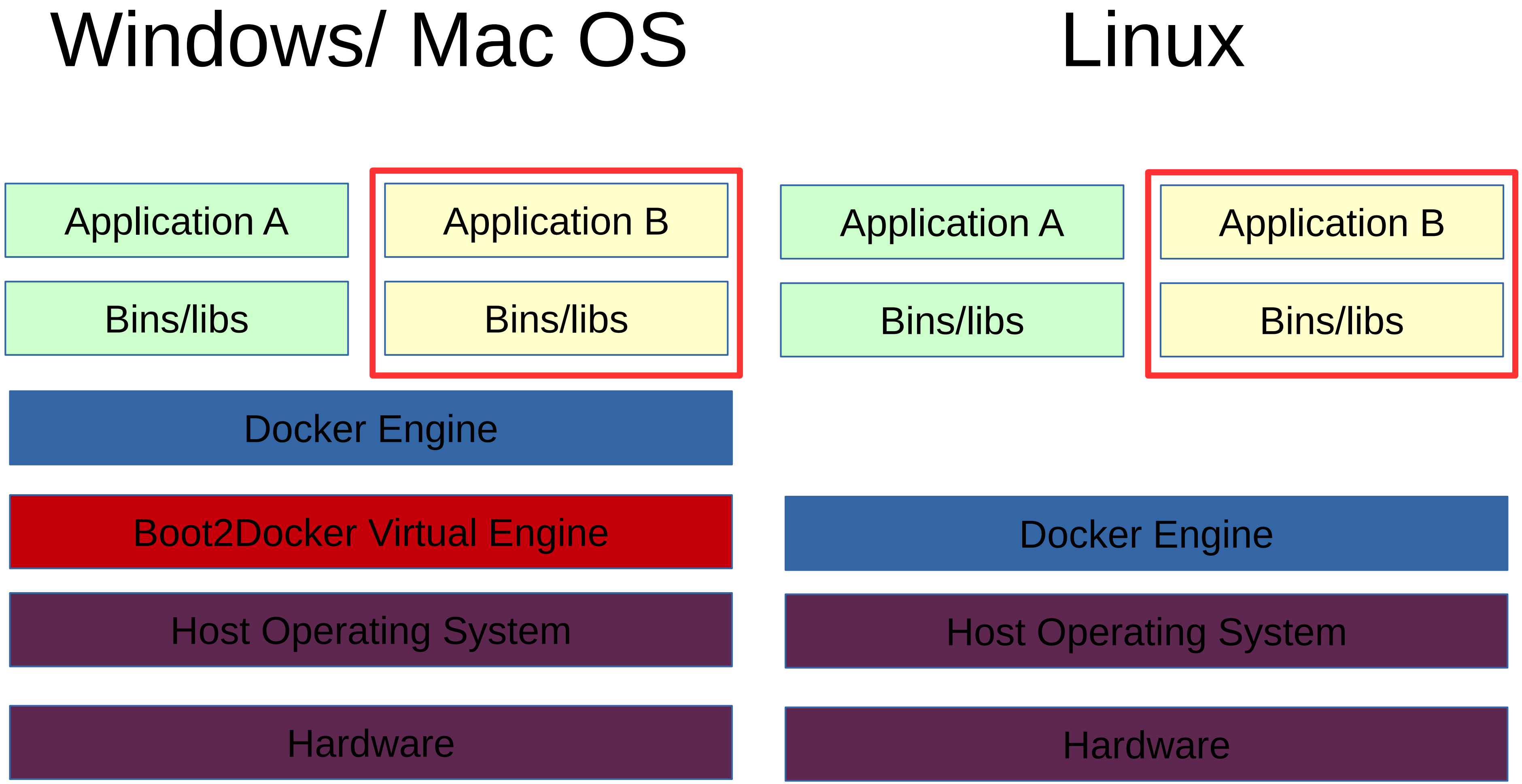
- Uses LinuX Containers (LXC)
- Sharing resources
- Central repositories of images
- Current set-up can be saved, copied and shared
- Dockerfile showing exact set-up
- Typically no GUI support
- Ideal for running a single application or service
- Many Docker images can be run side by side

Docker compared to VMs



http://www.jayway.com/wp-content/uploads/2015/03/vm-vs-docker.png

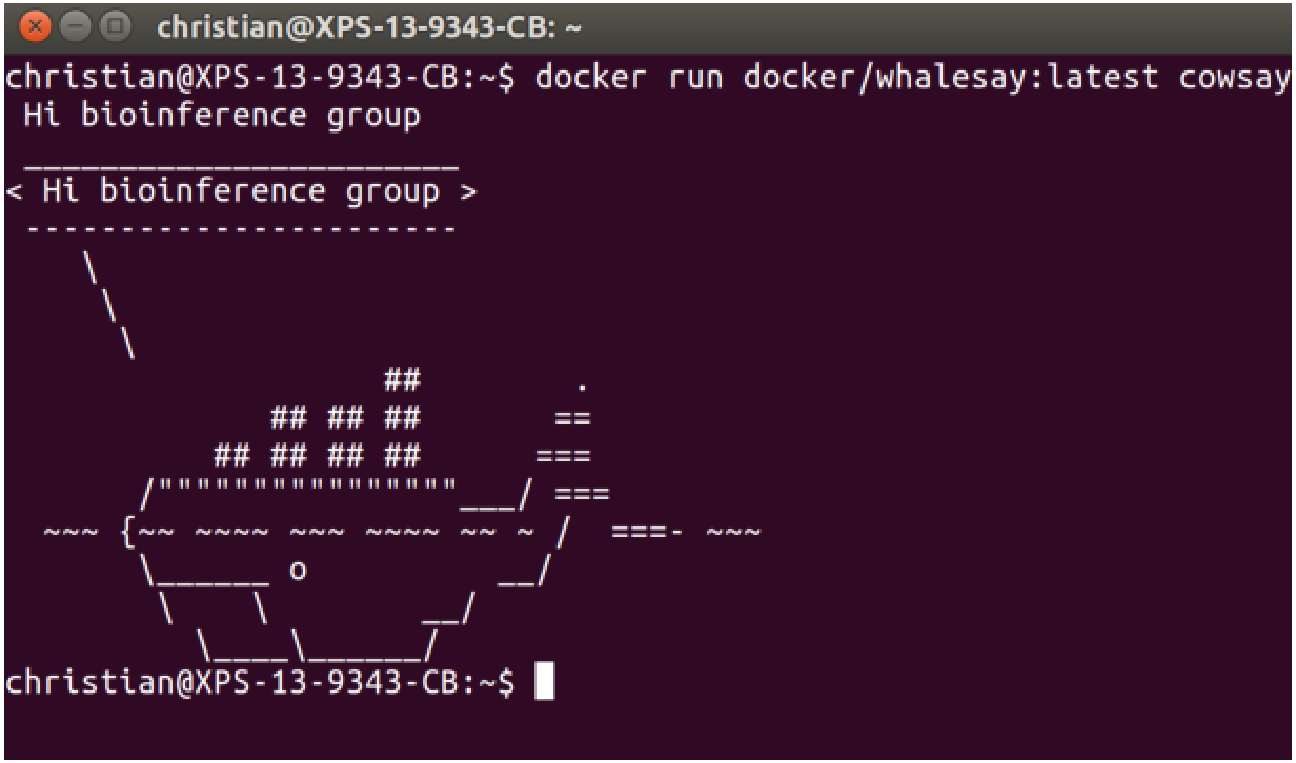
Docker Linux vs Windows



http://www.jayway.com/wp-content/uploads/2015/03/vm-vs-docker.png

A few Run Examples

- docker run docker/whalesay:latest cowsay Hi bioinference group



```
christian@XPS-13-9343-CB:~$ docker run docker/whalesay:latest cowsay
 Hi bioinference group
 _______________________
< Hi bioinference group >
 -----------------------
    \
     \
      \
                    ##        .
              ## ## ##       ==
           ## ## ## ##      ===
       /""""""""""""""""___/ ===
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
christian@XPS-13-9343-CB:~$ 
```

See: http://docs.docker.com/linux/started/

Parts of docker command

- docker : Starts the docker application
- run : docker command to run an image
- docker/whalesay : image to run
  - docker : owner of the repository
  - whalesay : image to run
  - :latest :tag of image to run (:latest is the default
- cowsay : Application inside the image to run
- Hi Bioinference group: parameters for application

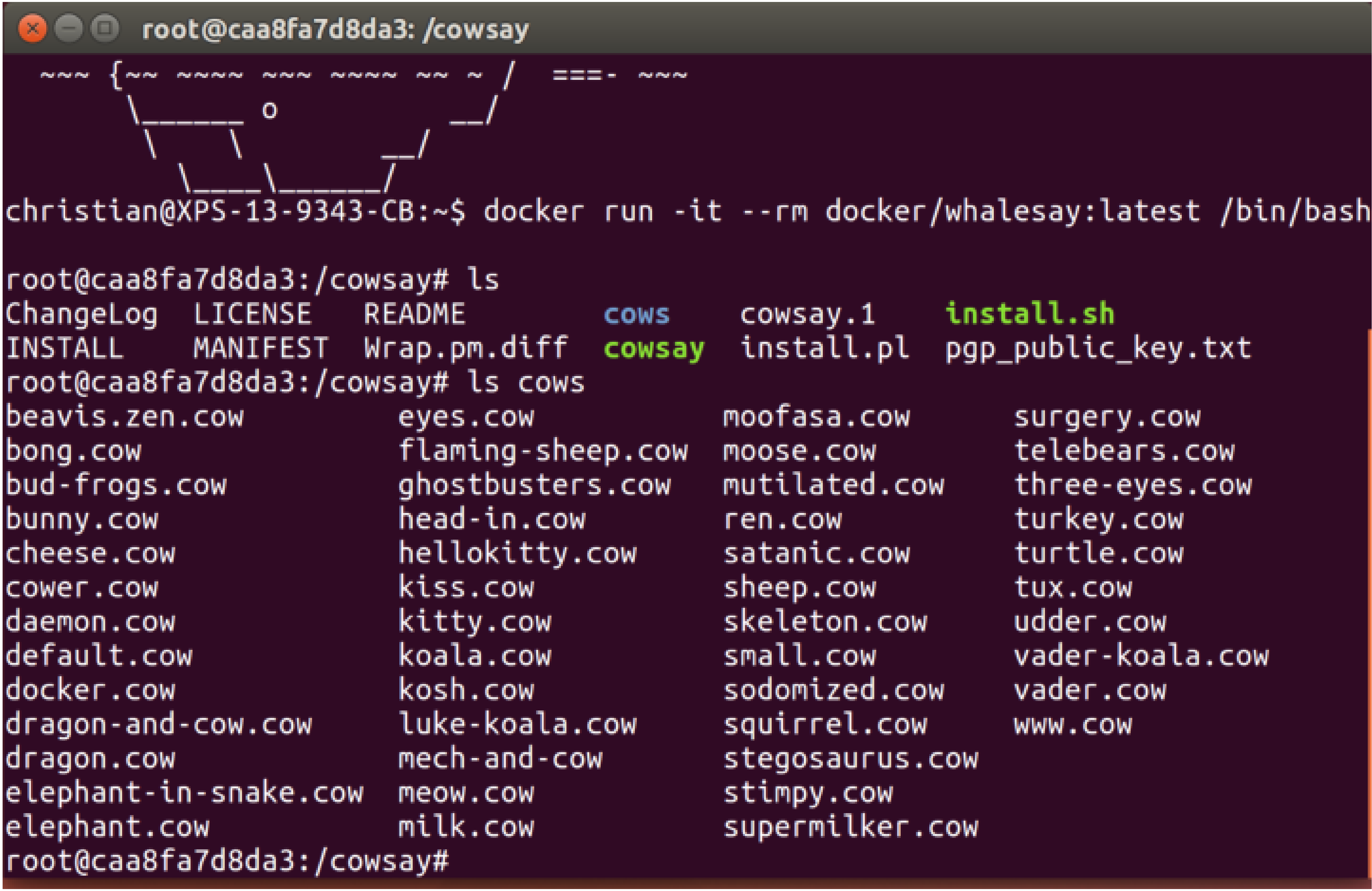
Separate Run environment

- docker run -i -t --rm docker/whalesay

  -i = Keep STDIN open even if not attached

  -t = Allocate a pseudo-TTY

  --rm = Automatically remove the container when it exits
- Open a bin/bash terminal
- ls
  - Cows directory , cowsay progam

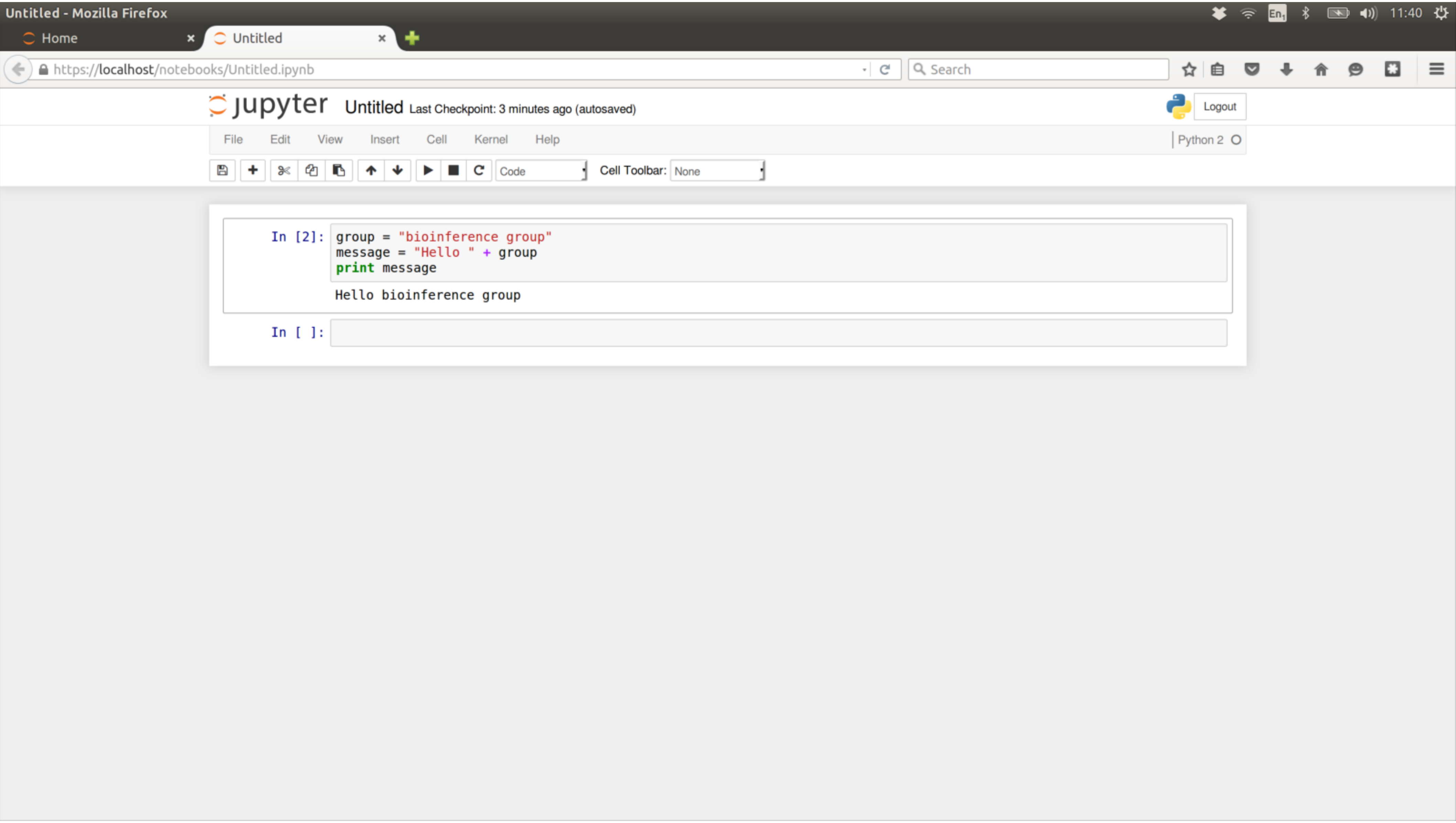


```
  ~~~ {~~ ~~~~ ~~~ ~~~~ ~~ ~ /  ===- ~~~
       \______ o          __/
        \    \        __/
          \____\______/
christian@XPS-13-9343-CB:~$ docker run -it --rm docker/whalesay:latest /bin/bash

root@caa8fa7d8da3:/cowsay# ls
ChangeLog  LICENSE   README        cows    cowsay.1    install.sh
INSTALL    MANIFEST  Wrap.pm.diff  cowsay  install.pl  pgp_public_key.txt
root@caa8fa7d8da3:/cowsay# ls cows
beavis.zen.cow         eyes.cow           moofasa.cow      surgery.cow
bong.cow               flaming-sheep.cow  moose.cow        telebears.cow
bud-frogs.cow          ghostbusters.cow   mutilated.cow    three-eyes.cow
bunny.cow              head-in.cow        ren.cow          turkey.cow
cheese.cow             hellokitty.cow     satanic.cow      turtle.cow
cower.cow              kiss.cow           sheep.cow        tux.cow
daemon.cow             kitty.cow          skeleton.cow     udder.cow
default.cow            koala.cow          small.cow        vader-koala.cow
docker.cow             kosh.cow           sodomized.cow    vader.cow
dragon-and-cow.cow     luke-koala.cow     squirrel.cow     www.cow
dragon.cow             mech-and-cow       stegosaurus.cow
elephant-in-snake.cow  meow.cow           stimpy.cow
elephant.cow           milk.cow           supermilker.cow
root@caa8fa7d8da3:/cowsay#
```

Ipython example

- docker run -d -p 443:8888 -e "PASSWORD=test" --name iserver ipython/scipyserver
- docker ps
- https://0.0.0.0/tree (use password entered in run command)
- If using boot2docker
  - boot2docker ip   (to get ip address vm uses)
  - https://*.*.*.*/tree
- https://www.ibm.com/developerworks/community/blogs/jfp/entry/using_ipython_notebooks_in_d

Docker ipython

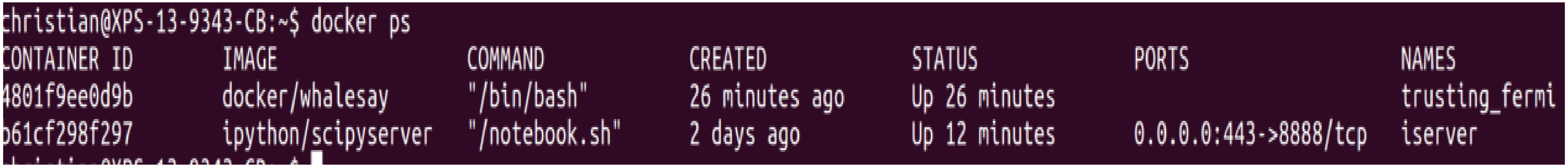

Untitled - Mozilla Firefox
Home
Untitled
11:40
https://localhost/notebooks/Untitled.ipynb
Search
jupyter
Untitled Last Checkpoint: 3 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Help
Python 2
Code
Cell Toolbar: None

```
In [2]: group = "bioinference group"
        message = "Hello " + group
        print message

        Hello bioinference group

In [ ]:
```

Ipython continued

- -d
  - Run container in background and print container ID
- -p 443:8888
  - Publish a container's port(s) to the host
- -e "PASSWORD=test"
  - Set environment variables
- --name iserver
  - Assign a name to the container
- ipython/scipyserver
  - Name of the image

Container



```
christian@XPS-13-9343-CB:~$ docker ps
CONTAINER ID        IMAGE                  COMMAND            CREATED            STATUS             PORTS                    NAMES
4801f9ee0d9b        docker/whalesay        "/bin/bash"        26 minutes ago     Up 26 minutes                               trusting_fermi
b61cf298f297        ipython/scipyserver    "/notebook.sh"     2 days ago         Up 12 minutes      0.0.0.0:443->8888/tcp    iserver
```

- docker ps
  - CONTAINER ID b61cf298f297
  - IMAGE ipython/scipyserver
  - COMMAND "/notebook.sh"
  - CREATED 28 minutes ago
  - STATUS Up 28 minutes
  - PORTS 0.0.0.0:443->8888/tcp
  - NAMES iserver

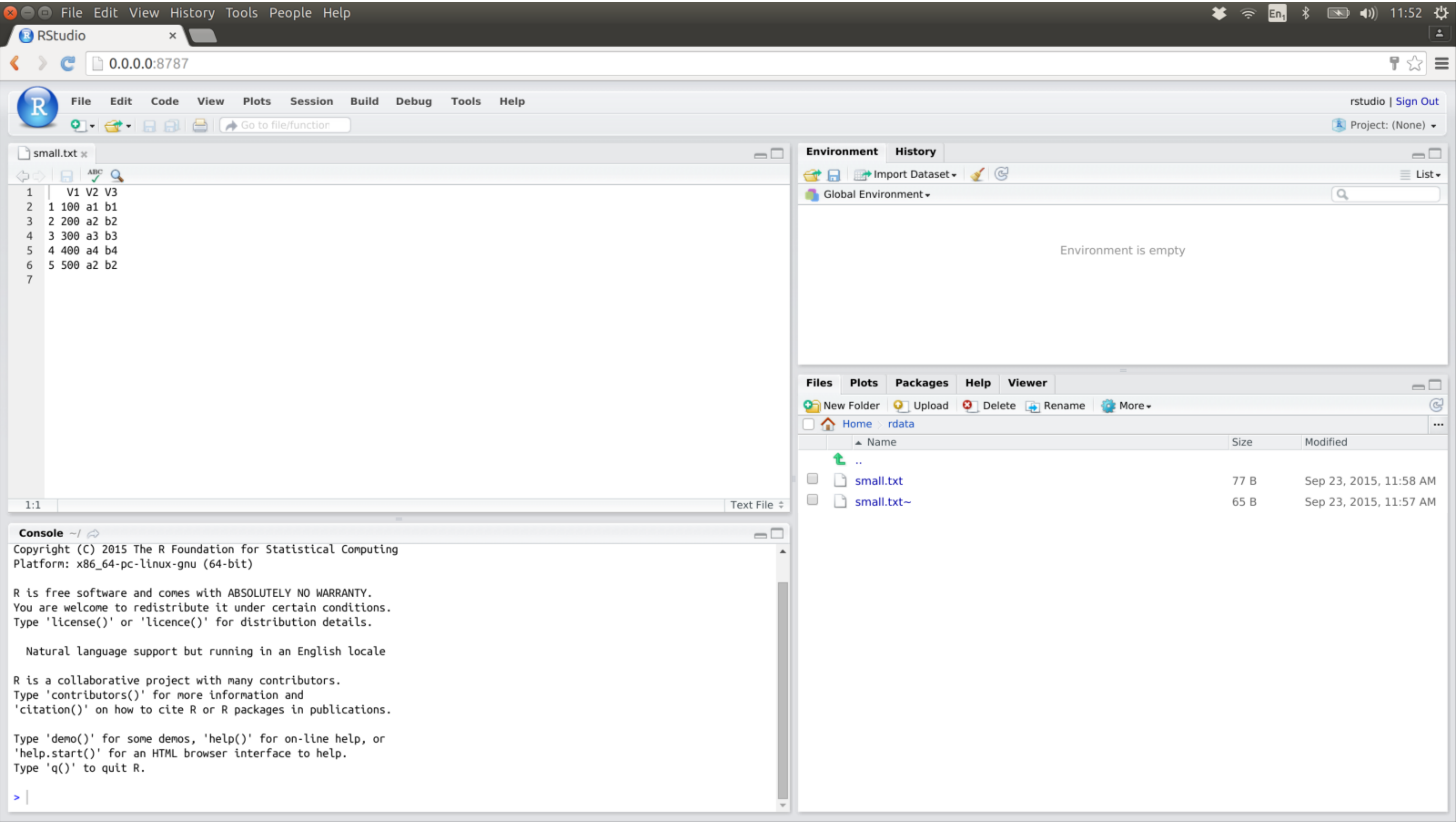
Container start and start

- At https://0.0.0.0/tree
  - New Python 2
  - print “hello world”
  - Run Button
- Close and reopen Jupiter
- docker stop iserver
- See https://0.0.0.0/tree fails
- docker start iserver
- See https://0.0.0.0/tree saves still there

RStudio

-
  ```
  docker run -d -p 8787:8787 -v /home/christian/docker/rdata/:/home/rstudio/rdata --name=rstudio -e USER=rstudio -e PASSWORD=rstudio rocker/rstudio
  ```
  - v maps a directory into the docker container
- http://0.0.0.0:8787/
- Outside changes to ../rdata are visible in rstudio
- see https://github.com/rocker-org/rocker/wiki/Using-the-RStudio-image

Docker RStudio



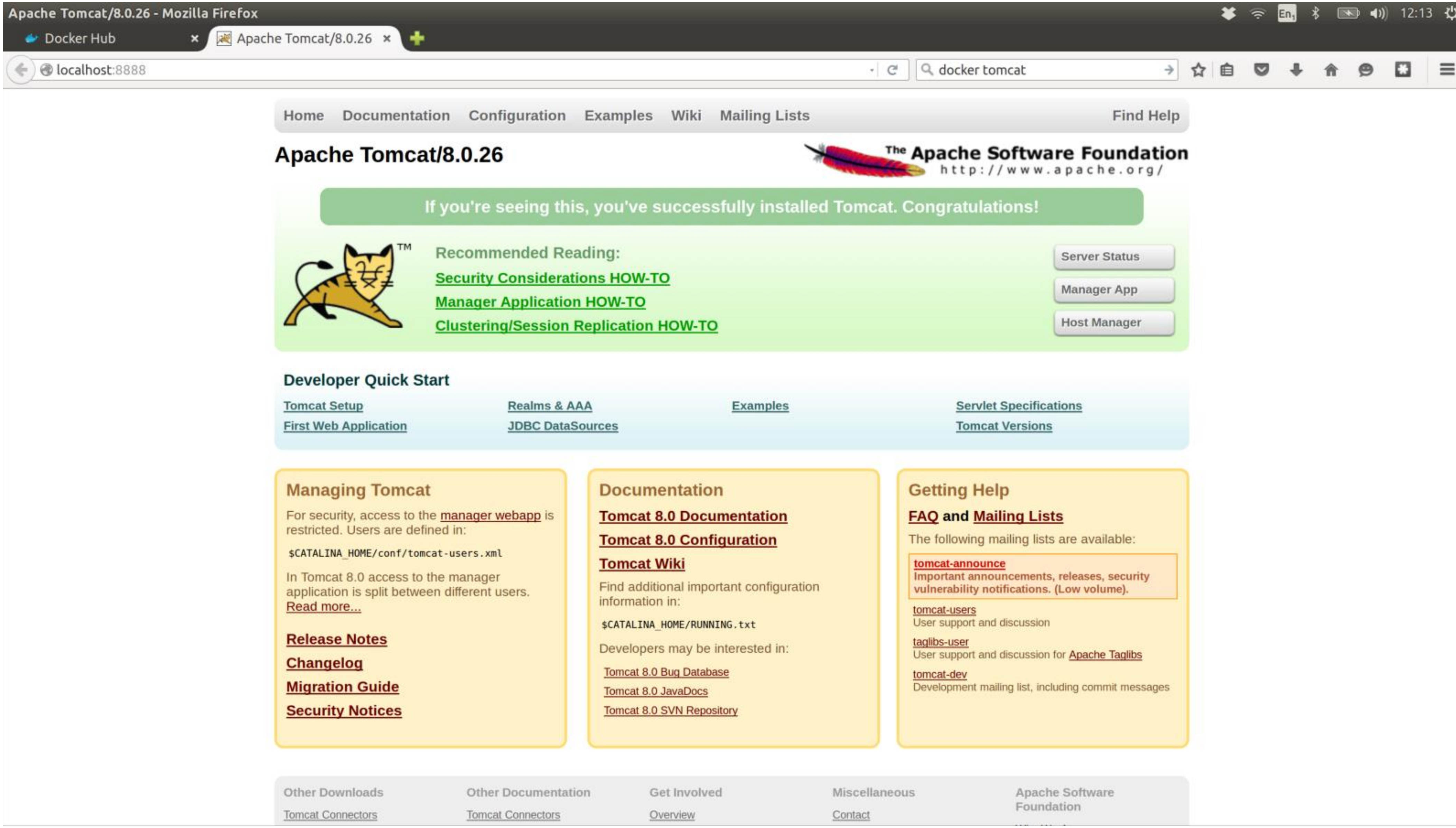
small.txt:

```
    V1 V2 V3
1 100 a1 b1
2 200 a2 b2
3 300 a3 b3
4 400 a4 b4
5 500 a2 b2

```

Name	Size	Modified
..		
small.txt	77 B	Sep 23, 2015, 11:58 AM
small.txt~	65 B	Sep 23, 2015, 11:57 AM

```
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> 
```

docker run -it --rm -p 8888:8080 tomcat:8.0



Apache Tomcat/8.0.26

If you're seeing this, you've successfully installed Tomcat. Congratulations!

Recommended Reading:

Security Considerations HOW-TO
Manager Application HOW-TO
Clustering/Session Replication HOW-TO

Developer Quick Start

Tomcat Setup	Realms & AAA	Examples	Servlet Specifications
First Web Application	JDBC DataSources		Tomcat Versions

Managing Tomcat

For security, access to the manager webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 8.0 access to the manager application is split between different users. Read more...

Release Notes
Changelog
Migration Guide
Security Notices

Documentation

Tomcat 8.0 Documentation
Tomcat 8.0 Configuration
Tomcat Wiki

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

Tomcat 8.0 Bug Database
Tomcat 8.0 JavaDocs
Tomcat 8.0 SVN Repository

Getting Help

FAQ and Mailing Lists

The following mailing lists are available:

tomcat-announce
Important announcements, releases, security vulnerability notifications. (Low volume).

tomcat-users
User support and discussion

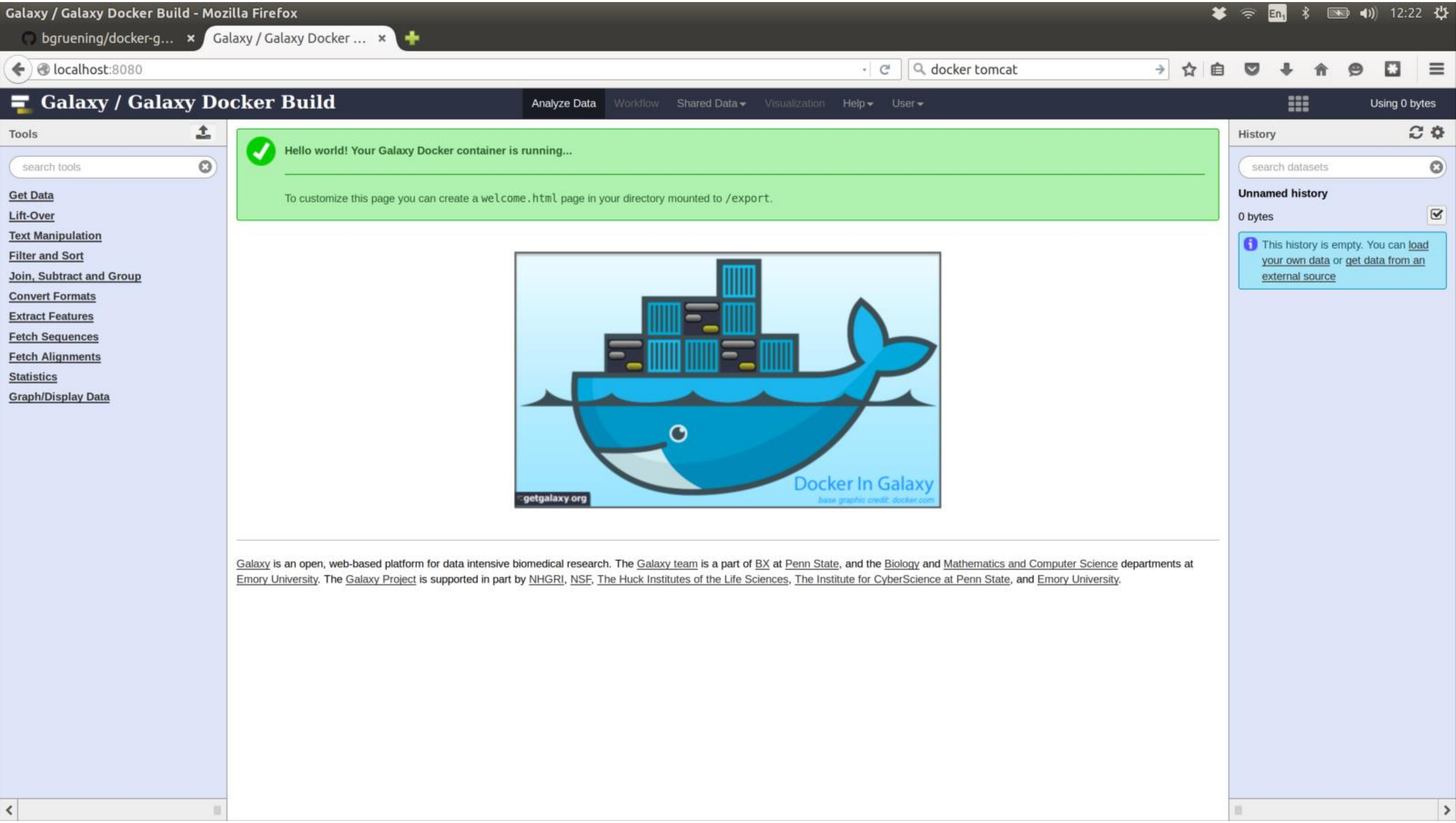
taglibs-user
User support and discussion for Apache Taglibs

tomcat-dev
Development mailing list, including commit messages

Other Downloads	Other Documentation	Get Involved	Miscellaneous	Apache Software Foundation
Tomcat Connectors	Tomcat Connectors	Overview	Contact	

- https://hub.docker.com/_/tomcat/

docker run -d -p 8080:80 -p 8021:21 bgruening/galaxy-stable



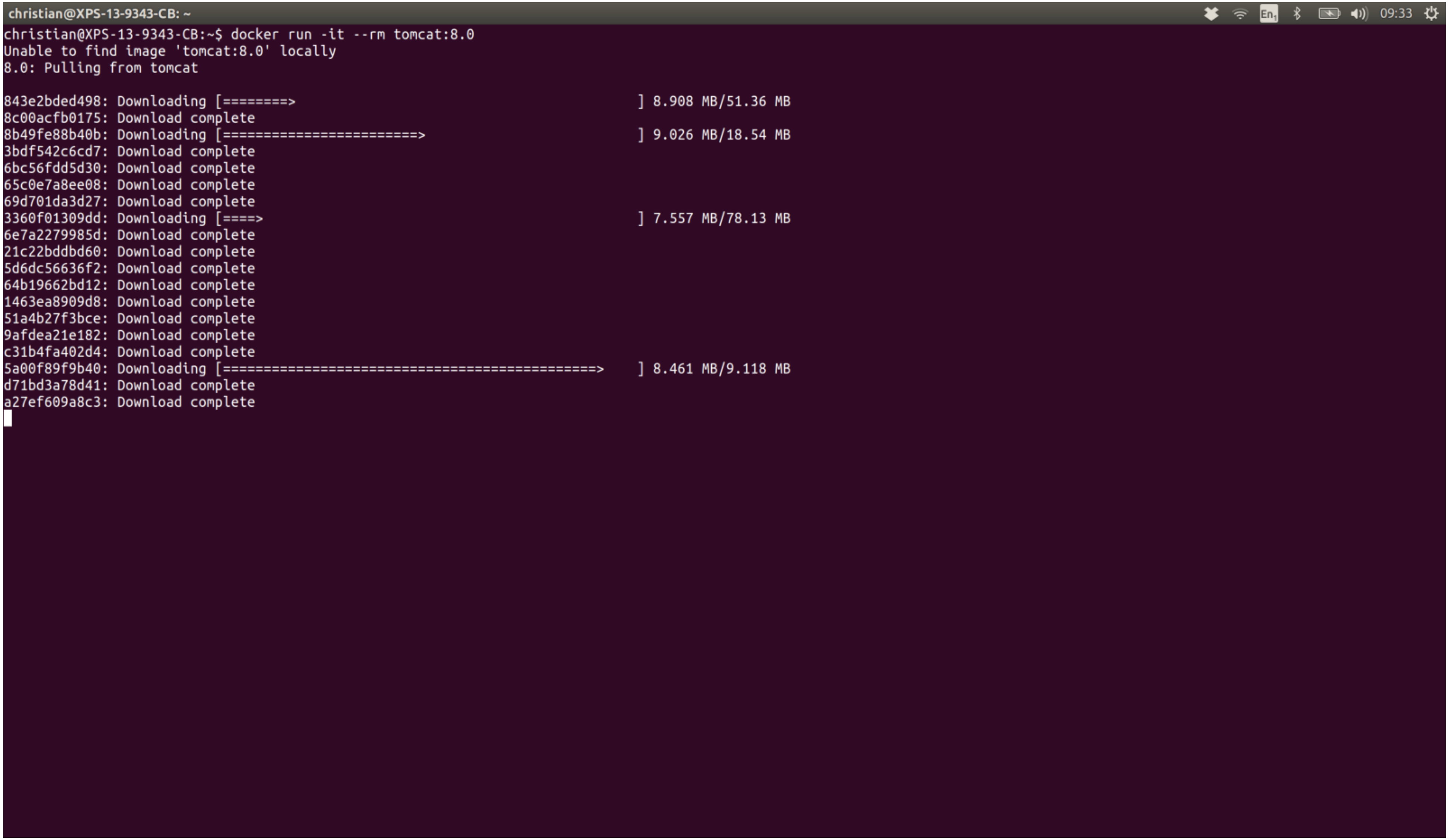
Hello world! Your Galaxy Docker container is running...

To customize this page you can create a `welcome.html` page in your directory mounted to `/export`.

This history is empty. You can load your own data or get data from an external source

Galaxy is an open, web-based platform for data intensive biomedical research. The Galaxy team is a part of BX at Penn State, and the Biology and Mathematics and Computer Science departments at Emory University. The Galaxy Project is supported in part by NHGRI, NSF, The Huck Institutes of the Life Sciences, The Institute for CyberScience at Penn State, and Emory University.

Docker downloads the first time

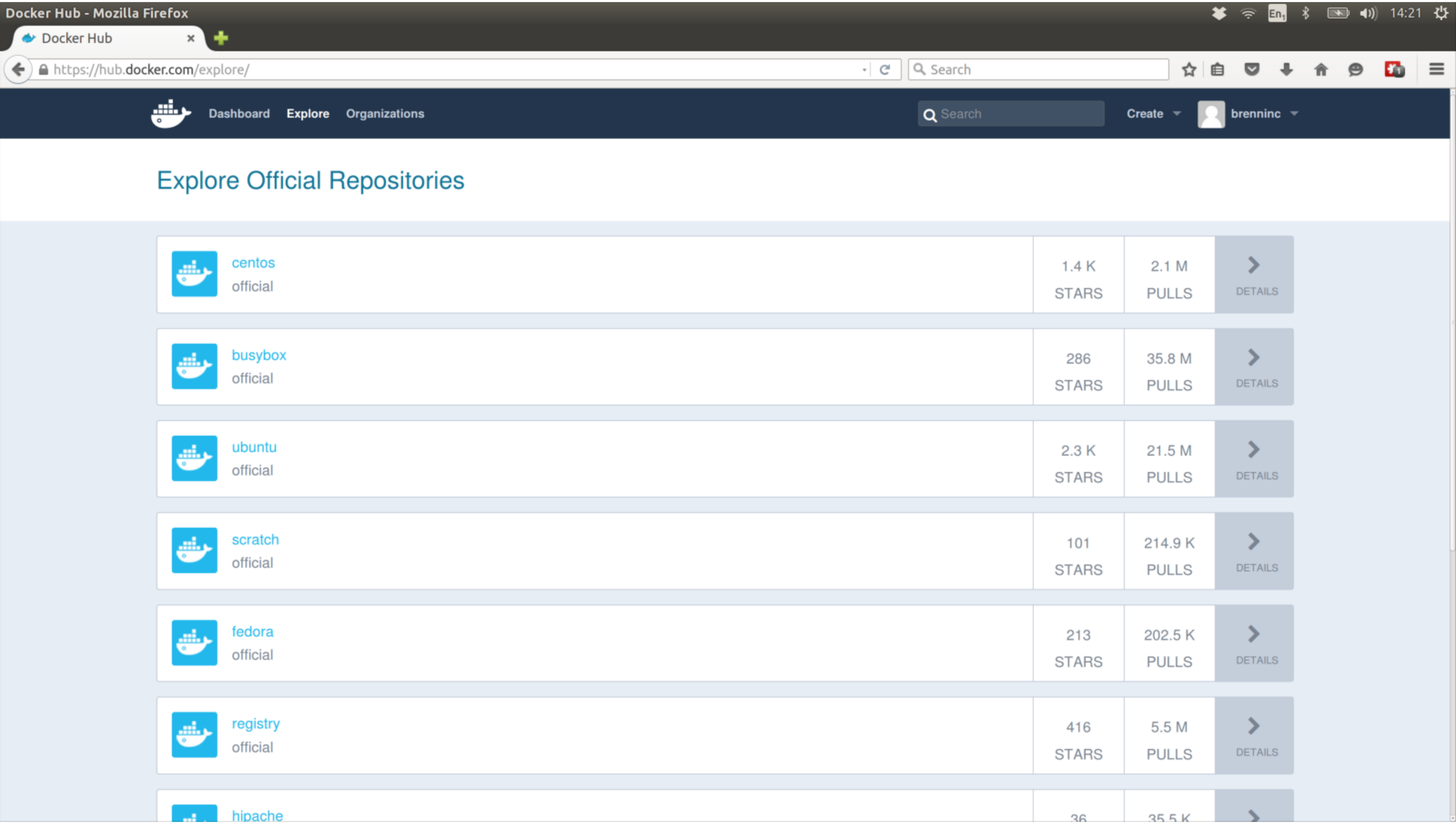


```
christian@XPS-13-9343-CB:~$ docker run -it --rm tomcat:8.0
Unable to find image 'tomcat:8.0' locally
8.0: Pulling from tomcat

843e2bded498: Downloading [========>                                          ] 8.908 MB/51.36 MB
8c00acfb0175: Download complete
8b49fe88b40b: Downloading [========================>                          ] 9.026 MB/18.54 MB
3bdf542c6cd7: Download complete
6bc56fdd5d30: Download complete
65c0e7a8ee08: Download complete
69d701da3d27: Download complete
3360f01309dd: Downloading [====>                                              ] 7.557 MB/78.13 MB
6e7a2279985d: Download complete
21c22bddbd60: Download complete
5d6dc56636f2: Download complete
64b19662bd12: Download complete
1463ea8909d8: Download complete
51a4b27f3bce: Download complete
9afdea21e182: Download complete
c31b4fa402d4: Download complete
5a00f89f9b40: Downloading [==============================================>    ] 8.461 MB/9.118 MB
d71bd3a78d41: Download complete
a27ef609a8c3: Download complete
```

Ship

- https://hub.docker.com
  - Images that can be downloaded
- docker pull xyz (gets an image and its parents)
- docker run xyz (pulls if required)
- Many images linked to a github account
  - Dockerfile
  - Extra files
  - Info files
- Automatically built so you know exactly what you get

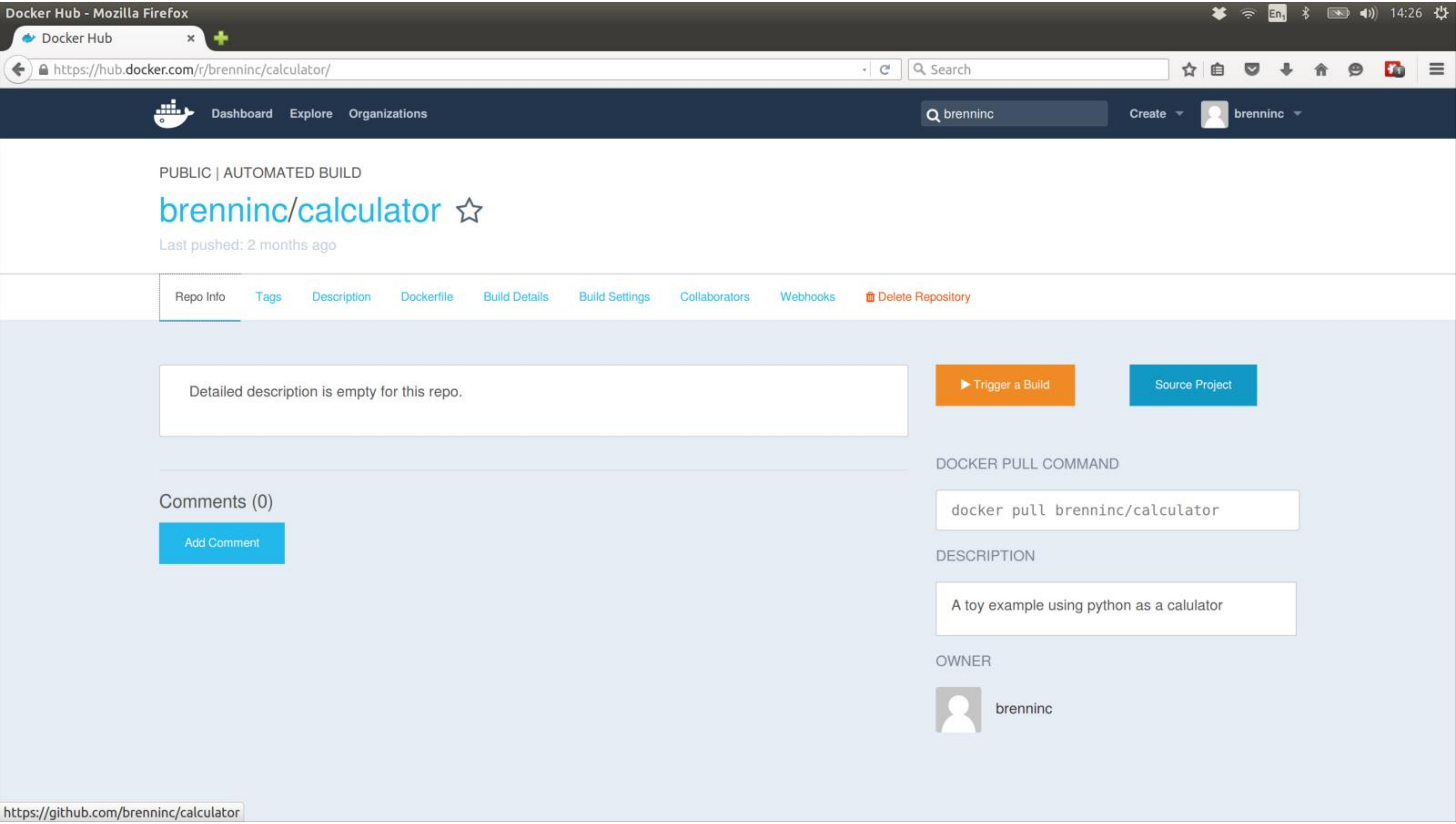
https://hub.docker.com/explore/



Explore Official Repositories

Repository	Stars	Pulls	
centos official	1.4 K STARS	2.1 M PULLS	DETAILS
busybox official	286 STARS	35.8 M PULLS	DETAILS
ubuntu official	2.3 K STARS	21.5 M PULLS	DETAILS
scratch official	101 STARS	214.9 K PULLS	DETAILS
fedora official	213 STARS	202.5 K PULLS	DETAILS
registry official	416 STARS	5.5 M PULLS	DETAILS
hipache	36	35.5 K	

https://hub.docker.com/r/brenninc/calculator/



Docker Hub - Mozilla Firefox
Docker Hub
https://hub.docker.com/r/brenninc/calculator/
Search
14:26
Dashboard
Explore
Organizations
brenninc
Create
brenninc

PUBLIC | AUTOMATED BUILD

brenninc/calculator ☆

Last pushed: 2 months ago

Detailed description is empty for this repo.

Comments (0)

DOCKER PULL COMMAND

```
docker pull brenninc/calculator
```

DESCRIPTION

A toy example using python as a calulator

OWNER

brenninc

https://github.com/brenninc/calculator

Build (the BAD WAY)

- docker run -i -t --name=bad ubuntu:14.04
  - curl --version

    curl: command not found
  - sudo apt-get install curl
  - curl --version
    - curl 7.35.0 .....
  - exit
- docker run -i -t --rm ubuntu:14.04
  - curl --version

    curl: command not found
- docker start -i bad

Build (The bad way)

- These images can be uploaded to docker hub
- No Dockerfile will be available
- No Automatic build

- Would you trust someone else’s black box?

Build using Docker files

- Saved in a text file called Dockerfile
- Exact record of how the system was built
- Dockerfile can built upon other docker images
- Built up in layers
  - Max 128 layers
- Each command in a Dockerfile is a layer
- Docker file allow for “automatic builds” on Docker hub
- Docker files typically shared via github

Calculator Example

- docker run –rm brenninc/calculator 4+5*2
  - 4+5*2 = 14
- docker run –rm brenninc/calculator
  - 1 + 2 * 3 = 7

Dockerfile instructions

- FROM
- MAINTAINER
- LABEL
- RUN
- ENTRYPOINT
- CMD
- EXPOSE
- ENV
- COPY

From

- Base or parent image
- Can be an operating system
  - FROM ubuntu:14.04
  - FROM centos
  - FROM febora
- Only Linux family operating system
- Can be a base image
  - ipython/scipyserver
  - ipython/scipystack
  - ipython/ipython:3.x

MAINTAINER

- A way of signalling who is responsible for the image
- MAINTAINER Christian Brenninkmeijer <Christian.Brenninkmeijer@manchester.ac.uk>
- MAINTAINER IPython Project <ipython-dev@scipy.org>
- Does count towards the 128 layer limit

LABEL

- key-value paired metadata

- LABEL com.example.label-with-value="foo"
- LABEL version="1.0"
- LABEL description="This text illustrates \
- that label-values can span multiple lines."

- Exposed vai
- docker inspect image_name

RUN

- Executes command on base image and saves a new image
- apt-get Install stuff
- Download stuff
- Unzip stuff
- Create directories
- Run setup and config scripts
- Delete temporary files

Run examples

-
  ```
  RUN apt-get update && apt-get install -y python
  ```

-
  ```
  RUN curl -L http://downloads.sourceforge.net/project/libpng/libpng16/older-releases/1.6.7/libpng-1.6.7.tar.gz > libpng-1.6.7.tar.gz && \
    tar -xzf libpng-1.6.7.tar.gz &&
    rm libpng-1.6.7.tar.gz && \
    mkdir libpng && \
    cd libpng-1.6.7 && \
  ```

Run notes

- Multiple command can be combined
  - These then count as one layer (out of 128 max)
- Temporary files must be removed in same layer as used or they stay in the image
  - Next image builds on previous
- cd (change directory) only effects that layer
  - Each new layer starts in home
- export only effects that layer
  - See ENV command

ENTRYPOINT and CMD

- Command to run then the image is run
- There can only be one of each
  - Earlier ones are ignored
- Both are optional and independent
- Various different formats possible

- Example:

  ```
  ENTRYPOINT ["python","calculator.py"]
  CMD ["1","+","2","*","3"]
  ```

  - Runs "python calculator.py 1+2*3

ENTRYPOINT

- Command part expected to be used every time
- Makes the image an executable file

- If docker run is provided arguments the ENTRYPOINT commands are still included

- Can be ignored with the docker run flag --entrypoint

CMD

- Default arguments for Docker run
- Ignored if any arguments are provided when docker images is run

```
docker run –rm brenninc/calculator 4+5*2
    4+5*2 = 14
docker run –rm brenninc/calculator
    1 + 2 * 3 = 7
```

EXPOSE

- “informs Docker that the container will listen on the specified network ports at runtime”
- Connects ports of any application/ service to be run to the outside of the docker
- Note requires the -p flag at runtime to expose it from docker to the host

ENV

- Sets Key value environment variable
- Persist on all future layers and runtime
- Can be overwritten

- ENV myName John Doe
- ENV myDog Rex The Dog
- ENV myCat fluffy

- ENV myName="John Doe" myDog=Rex\ The\ Dog \

COPY

- COPY source destination
- Copies local files or directories into the docker image
- Source must be in the same context as the Dockerfile
  - Files in the same context as the Docker file are only available in the image if copied in
- Multiple sources can be specified but then destination must be a folder

- COPY calculator.py calculator.py

ADD

- Similar to COPY but with extra functionality
  - Docker recommends using COPY when possible
- If source is a local tar archive in a recognized compression format (identity, gzip, bzip2 or xz) then it is unpacked as a directory.
- ADD can add data from URLS
  - Never unpacked

VOLUME

- Creates a mount point
  - Creates a directory in /var/lib/docker/volumes/
  - With a random name
- Used by containers that save data
  - Example Ipython
- Similar to the -v flag in docker run image
- Directory created when a container is created are not removed even if the container is

USER

- Allow you to run image as other than root user
- User must be created

WORKDIR

- Sets the working directory
- Should be an absolute directory
  - Absolute within docker image not the host
- Unlike cd persists between layers

ONBUILD

- Used in images that will be parents to other images
- Adds instructions to run then child image builds
- example

```
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app
ONBUILD COPY Gemfile /usr/src/app/
ONBUILD COPY Gemfile.lock /usr/src/app/
ONBUILD RUN bundle install
```

Calculator Dockerfile

- FROM ubuntu:14.04
- MAINTAINER Christian Brenninkmeijer <Christian.Brenninkmeijer@manchester.ac.uk>
- LABEL "description"="An example docker app using python as a calculator"
- #Install python via apt-get
- RUN apt-get update && apt-get install -y python
- #copy in the code
- COPY calculator.py calculator.py
- ENTRYPOINT ["python","calculator.py"]

Calculator.py

- import parser
- import sys

- command = " ".join(sys.argv[1:])
- st = parser.expr(command)
- code = st.compile('file.py')
- print command,"=",eval(code)

Build Calculator

- docker build -t brenninc/calculator .

-t provides a tag (name) for your image

- Docker will reuse existing images layers wherever this is possible
- Automatically detecting the first layer that changed
  - Including if a file copied in has changed
- All subsequent layers are built

SHIP Calculator

- Source file uploaded to:https://github.com/brenninc/calculator
  - Dockerfile
  - Calculator.py
- Linked to:https://hub.docker.com/r/brenninc/calculator/
- Automatically built image (by docker hub)
- docker pull brenninc/calculator