ORIGAMI
Lab

Containers

QLS 612 - May 2023

Sebastian Urchs - @s_urchs

Fondation Brain Canada Foundation
NIH
NIBIB
HEALTHY BRAINS HEALTHY LIVES
neuro Montreal Neurological Institute-Hospital
CONP PCNO

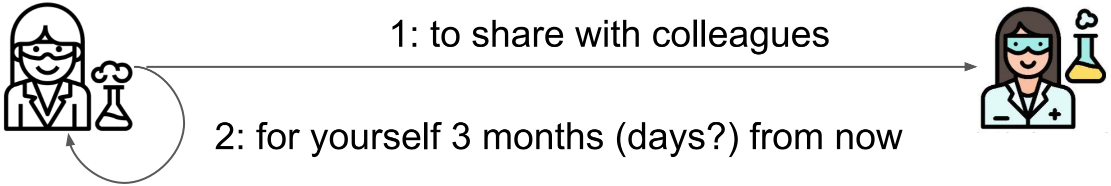
Containers?

What we will talk about

1. **Why** are containers useful for researchers
2. **Virtual Machines** (briefly)
3. Using and building containers with **Docker**
4. Using containers on supercomputers with **Singularity**

Why isolate environments

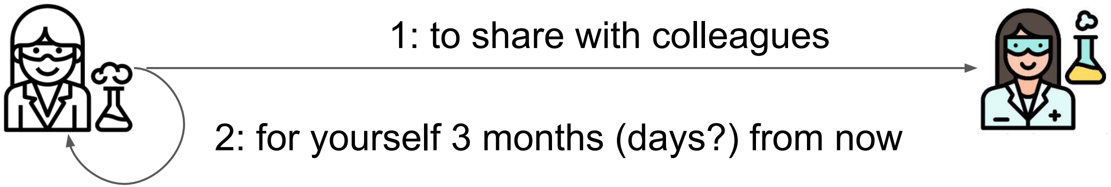
Document your software environment



1: to share with colleagues

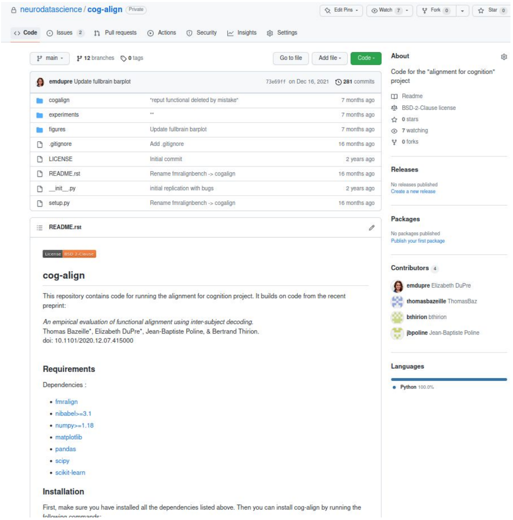
2: for yourself 3 months (days?) from now

Document your software environment



1: to share with colleagues

2: for yourself 3 months (days?) from now



emdupre Update fullbrain barplot 73e69ff on Dec 16, 2021 281 commits

cogalign	"reput functional deleted by mistake"	7 months ago
experiments	""	7 months ago
figures	Update fullbrain barplot	7 months ago
.gitignore	Add .gitignore	16 months ago
LICENSE	Initial commit	2 years ago
README.rst	Rename fmralignbench -> cogalign	16 months ago
__init__.py	initial replication with bugs	2 years ago
setup.py	Rename fmralignbench -> cogalign	16 months ago

README.rst

License BSD 2-Clause

cog-align

This repository contains code for running the alignment for cognition project. It builds on code from the recent preprint:

An empirical evaluation of functional alignment using inter-subject decoding.
Thomas Bazeille*, Elizabeth DuPre*, Jean-Baptiste Poline, & Bertrand Thirion.
doi: 10.1101/2020.12.07.415000

Requirements

Dependencies :

- fmralign
- nibabel>=3.1
- numpy>=1.18
- matplotlib
- pandas
- scipy
- scikit-learn

Installation

First, make sure you have installed all the dependencies listed above. Then you can install cog-align by running the following commands:

About

Code for the "alignment for cognition" project

Readme
BSD-2-Clause license
0 stars
7 watching
0 forks

Releases

No releases published
Create a new release

Packages

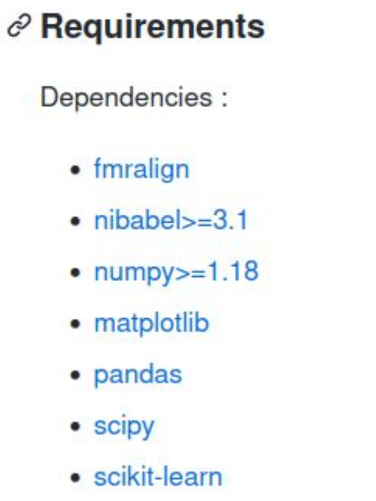
No packages published
Publish your first package

Contributors 4

emdupre Elizabeth DuPre
thomasbazeille ThomasBaz
bthirion bthirion
jbpoline Jean-Baptiste Poline

Languages

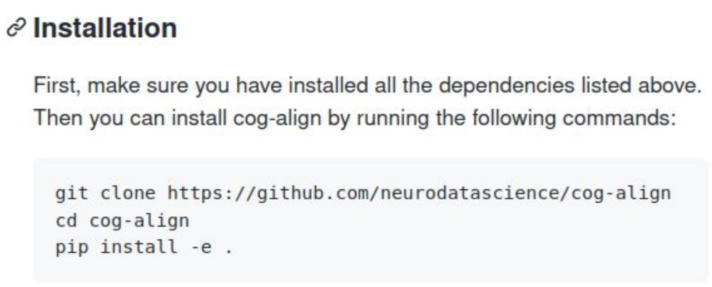
Python 100.0%



Requirements

Dependencies :

- fmralign
- nibabel>=3.1
- numpy>=1.18
- matplotlib
- pandas
- scipy
- scikit-learn



Installation

First, make sure you have installed all the dependencies listed above. Then you can install cog-align by running the following commands:

```
git clone https://github.com/neurodatascience/cog-align
cd cog-align
pip install -e .
```

https://github.com/neurodatascience/cog-align

Don’t use the same environment for all projects

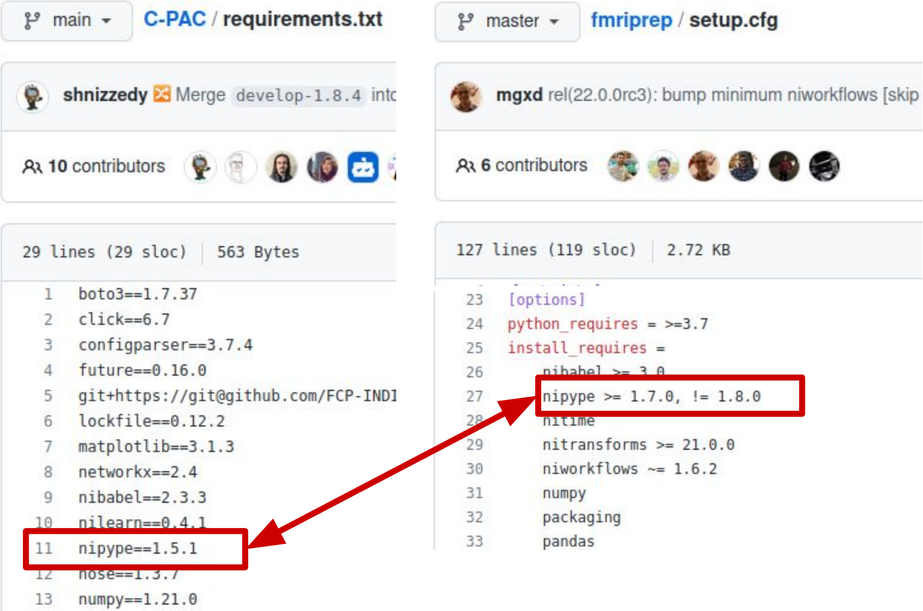
LATEST: 10.17
UPDATE
CHANGES IN VERSION 10.17:
THE CPU NO LONGER OVERHEATS WHEN YOU HOLD DOWN SPACEBAR.
COMMENTS:
LONGTIME USER4 WRITES:
THIS UPDATE BROKE MY WORKFLOW! MY CONTROL KEY IS HARD TO REACH, SO I HOLD SPACEBAR INSTEAD, AND I CONFIGURED EMACS TO INTERPRET A RAPID TEMPERATURE RISE AS "CONTROL".
ADMIN WRITES:
THAT'S HORRIFYING.
LONGTIME USER4 WRITES:
LOOK, MY SETUP WORKS FOR ME. JUST ADD AN OPTION TO REENABLE SPACEBAR HEATING.
EVERY CHANGE BREAKS SOMEONE'S WORKFLOW.

https://xkcd.com/1172/

Changing dependencies may do unexpected things

A. Your updated the dependencies of an existing project

Don’t use the same environment for all projects



C-PAC / requirements.txt

shnizzedy Merge develop-1.8.4 intc

10 contributors

29 lines (29 sloc) | 563 Bytes

```
boto3==1.7.37
click==6.7
configparser==3.7.4
future==0.16.0
git+https://git@github.com/FCP-INDI
lockfile==0.12.2
matplotlib==3.1.3
networkx==2.4
nibabel==2.3.3
nilearn==0.4.1
nipype==1.5.1
nose==1.3.7
numpy==1.21.0
```

fmriprep / setup.cfg

mgxd rel(22.0.0rc3): bump minimum niworkflows [skip

6 contributors

127 lines (119 sloc) | 2.72 KB

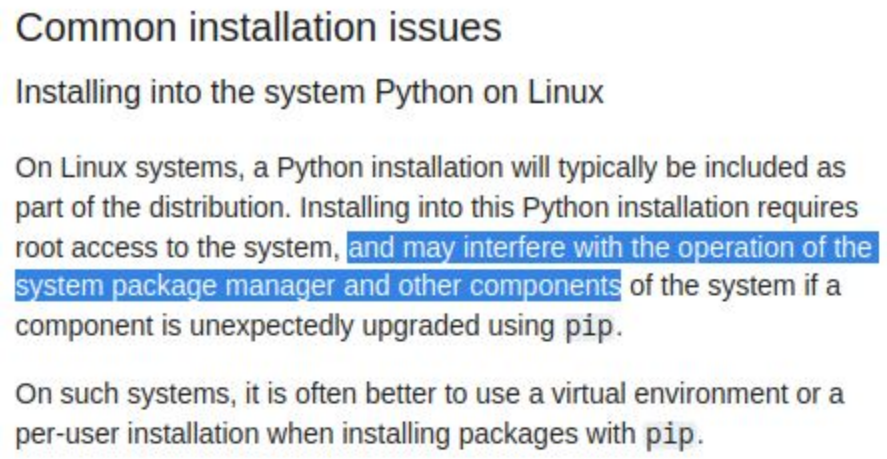
```
[options]
python_requires = >=3.7
install_requires =
    nibabel >= 3.0
    nipype >= 1.7.0, != 1.8.0
    nitime
    nitransforms >= 21.0.0
    niworkflows ~= 1.6.2
    numpy
    packaging
    pandas
```

Changing dependencies may do unexpected things

A. Your updated the dependencies of an existing project

B. **Two projects use the same environment but need different versions of some dependencies**

Do not install things into your system Python!

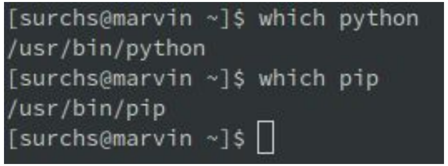


Common installation issues

Installing into the system Python on Linux

On Linux systems, a Python installation will typically be included as part of the distribution. Installing into this Python installation requires root access to the system, and may interfere with the operation of the system package manager and other components of the system if a component is unexpectedly upgraded using `pip`.

On such systems, it is often better to use a virtual environment or a per-user installation when installing packages with `pip`.



```
[surchs@marvin ~]$ which python
/usr/bin/python
[surchs@marvin ~]$ which pip
/usr/bin/pip
[surchs@marvin ~]$ 
```

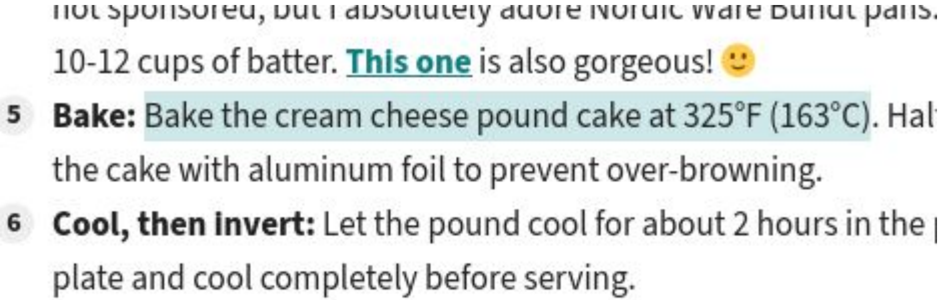
https://docs.python.org/3/installing/index.html

Consider: a cake

Home · Cakes · Perfect Cream Cheese Pound Cake

Perfect Cream Cheese Pound Cake

Published by Sally on February 18, 2019 - 700 comments



not sponsored, but I absolutely adore Nordic Ware Bundt pans.
10-12 cups of batter. This one is also gorgeous! 🙂

5 **Bake:** Bake the cream cheese pound cake at 325°F (163°C). Hal
the cake with aluminum foil to prevent over-browning.

6 **Cool, then invert:** Let the pound cool for about 2 hours in the
plate and cool completely before serving.

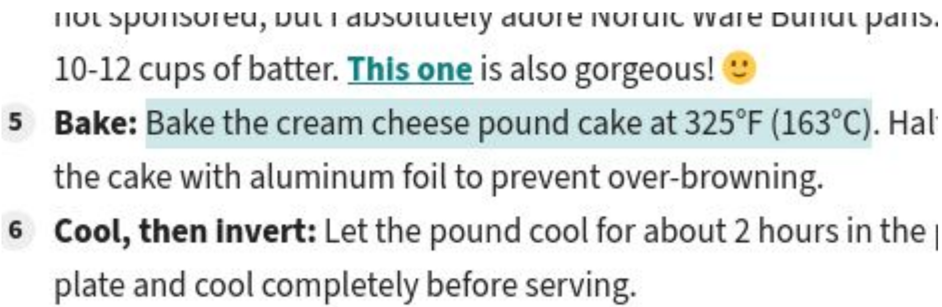
https://sallysbakingaddiction.com/cream-cheese-pound-cake/

Consider: a cake

Home · Cakes · Perfect Cream Cheese Pound Cake

Perfect Cream Cheese Pound Cake

Published by Sally on February 18, 2019 - 700 comments



not sponsored, but I absolutely adore Nordic Ware Bundt pans.
10-12 cups of batter. This one is also gorgeous! 🙂

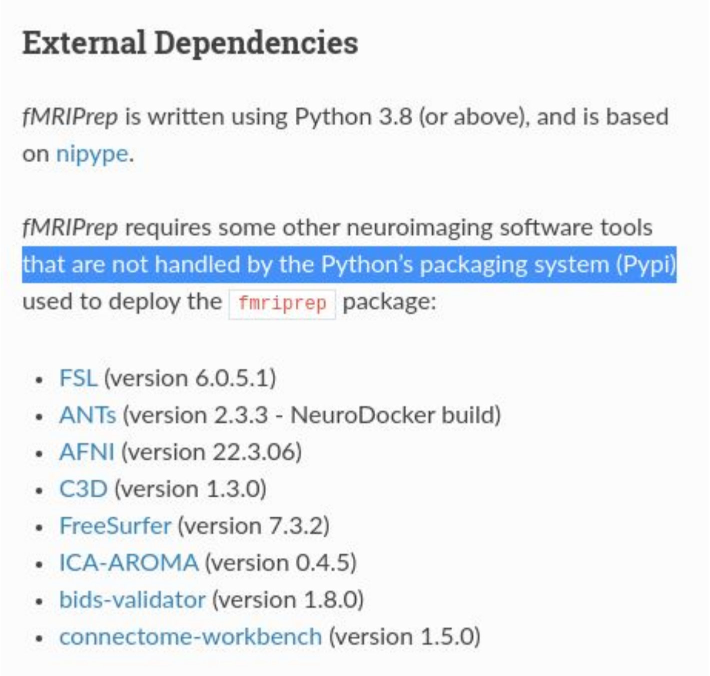
5 **Bake:** Bake the cream cheese pound cake at 325°F (163°C). Hal
the cake with aluminum foil to prevent over-browning.

6 **Cool, then invert:** Let the pound cool for about 2 hours in the
plate and cool completely before serving.

https://sallysbakingaddiction.com/cream-cheese-pound-cake/

Isolate environments to handle different requirements

Why not just python virtual environments?



External Dependencies

fMRIPrep is written using Python 3.8 (or above), and is based on nipype.

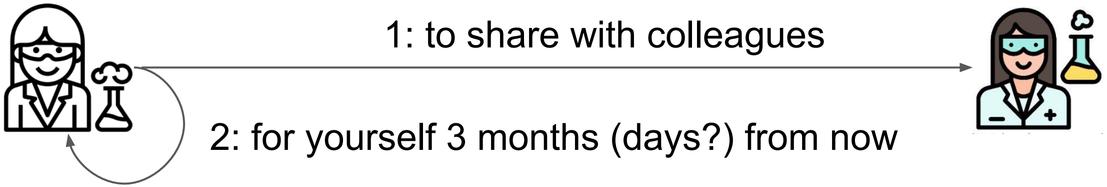
fMRIPrep requires some other neuroimaging software tools that are not handled by the Python's packaging system (Pypi) used to deploy the `fmriprep` package:

- FSL (version 6.0.5.1)
- ANTs (version 2.3.3 - NeuroDocker build)
- AFNI (version 22.3.06)
- C3D (version 1.3.0)
- FreeSurfer (version 7.3.2)
- ICA-AROMA (version 0.4.5)
- bids-validator (version 1.8.0)
- connectome-workbench (version 1.5.0)

https://fmriprep.org/en/stable/installation.html

- Not all binary dependencies are on Anaconda
- Not everything is written in Python or R
- Your Operating System (**OS**) also has packages and a package manager
- The same version problems apply to these

Need to capture the (entire) compute environment



1: to share with colleagues

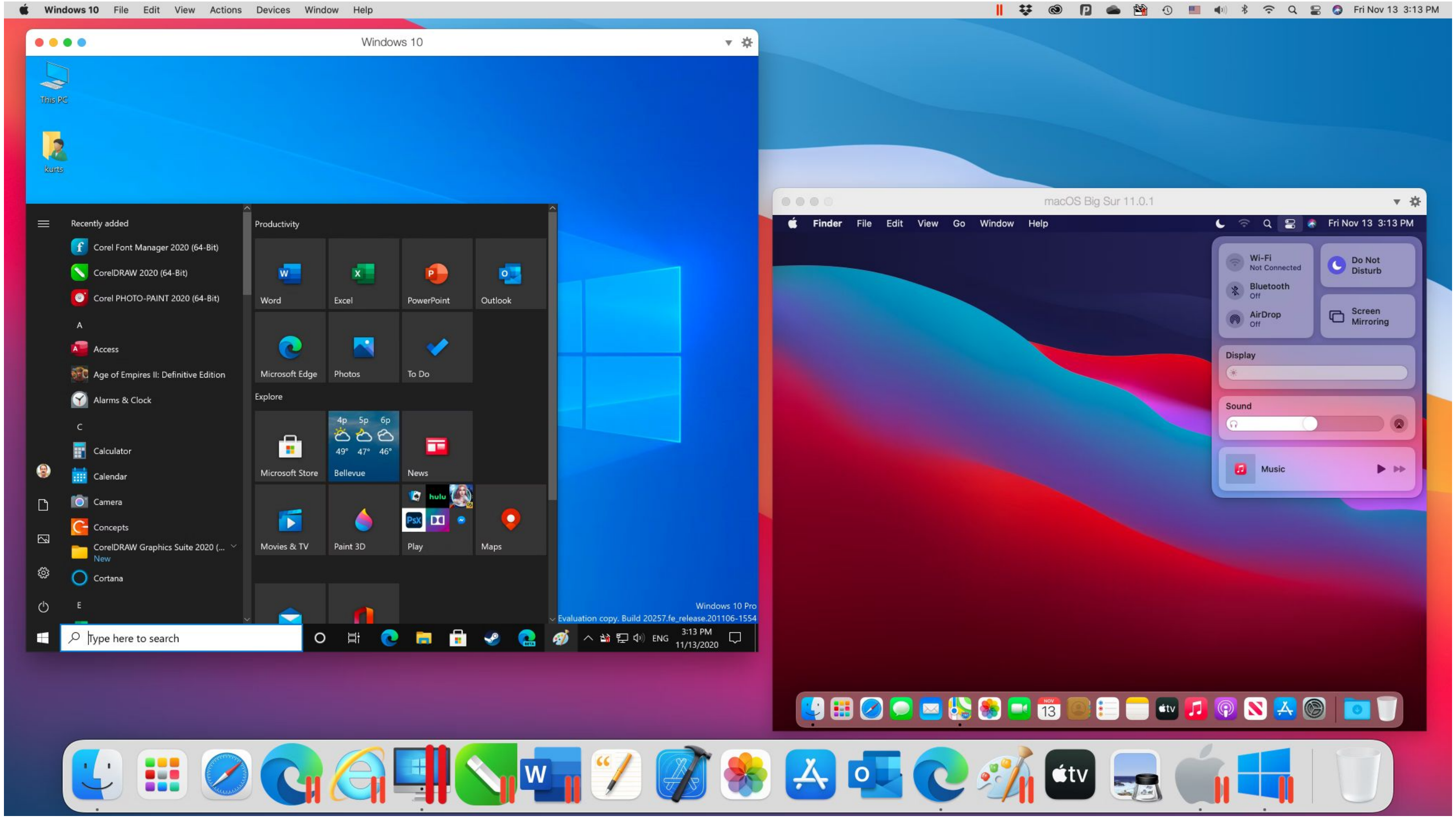
2: for yourself 3 months (days?) from now

Virtual Machines

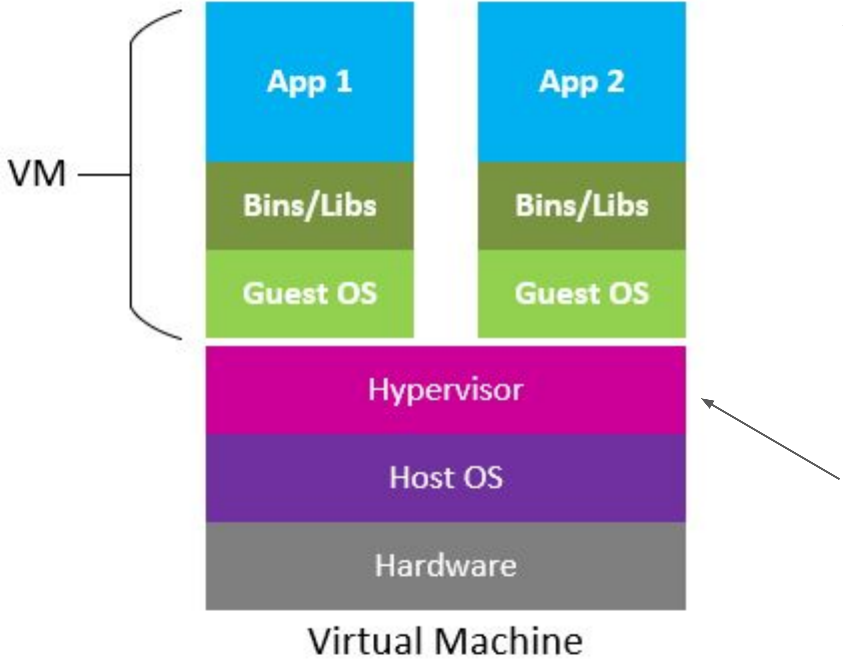
How can we isolate different OS environments?

Buy a lot of computers?

Virtual Machines: Let’s simulate the computers



Virtual Machine (VM)



What is it?

- entire computer hardware simulated (**hardware virtualization**)
- you can choose what hardware is exposed (CPU, RAM, hard-drive, …)
- the VM is fully isolated from the Host OS: you get to install any OS inside the VM

How do I make one?

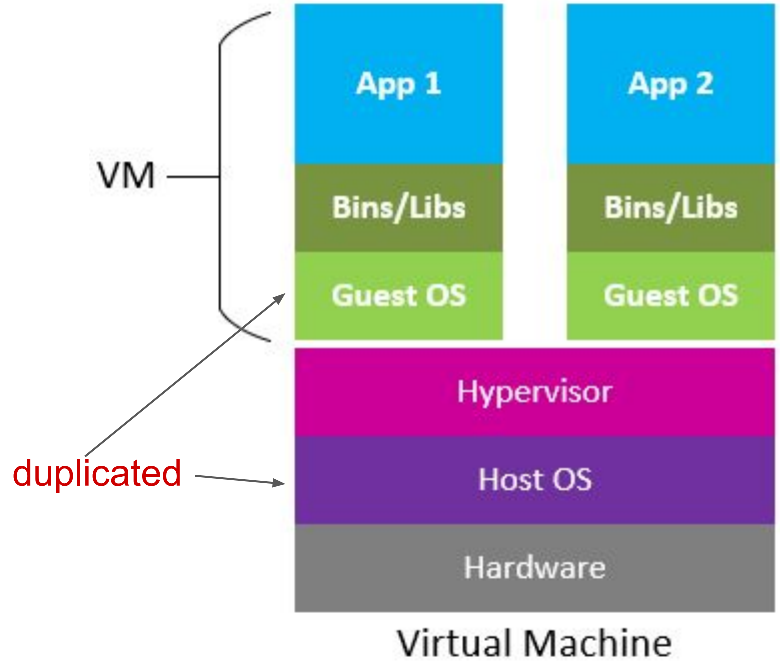
Get Hypervisor / Virtualization software

- VirtualBox (partly OSS)
- VMWare (commercial)
- Hyper-V (Windows)
- …

https://cloudblogs.microsoft.com/opensource/2019/07/15/how-to-get-started-containers-docker-kubernetes/

https://www.virtualbox.org/

Virtual Machine (VM)



Good:

- can run anything a computer runs (Windows, Linux, Hackintosh)
- can make a snapshot to share with others (Neurodebian used to have a VM)
- good way to test things across many systems
- full system isolation

But:

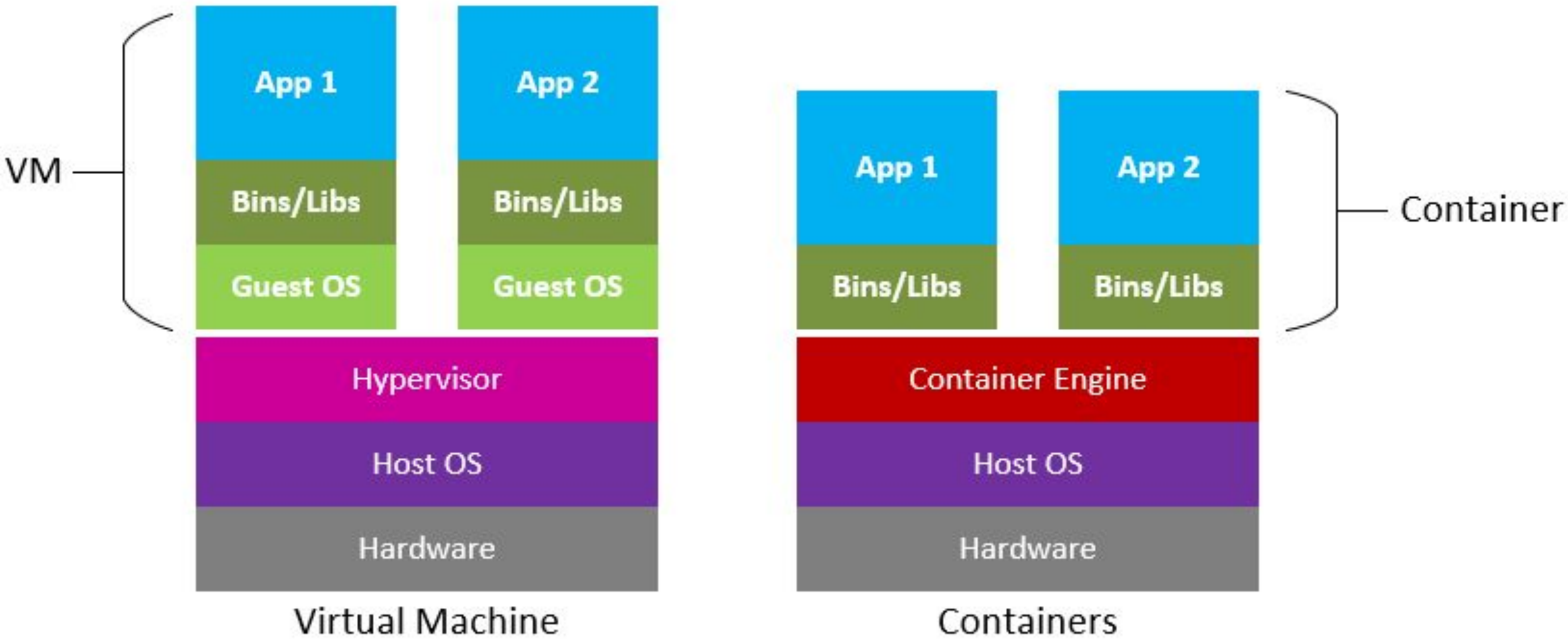
- doesn’t share resources with host -> BIG
- slow to start up, stop, resume
- cumbersome to configure for each project
- duplicates things (every VM needs OS / Kernel)
- no easy way to “get” and use VMs from other people
- you can’t use it on a supercomputer

https://cloudblogs.microsoft.com/opensource/2019/07/15/how-to-get-started-containers-docker-kubernetes/

https://www.virtualbox.org/

Containers

Containers: let’s all share the same kernel, but in boxes



https://cloudblogs.microsoft.com/opensource/2019/07/15/how-to-get-started-containers-docker-kubernetes/

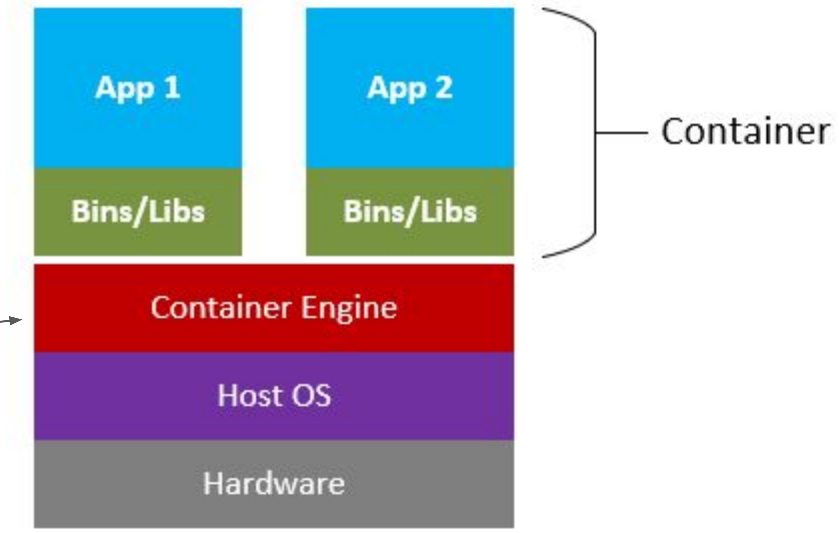
Containers

What are they

- isolated environments sharing the same kernel / OS -> (**OS virtualization**)
- from the inside, a container looks like a separate computer, can't see outside
- within each container, you can have your desired binary and library dependencies

How do I make one

- use a container implementation
- **docker** is the most widely used
- Singularity is used on supercomputers



Containers

https://cloudblogs.microsoft.com/opensource/2019/07/15/how-to-get-started-containers-docker-kubernetes/

Virtual Machine

Container

What is Docker

Docker Engine

- a command line program
- gets and builds Docker images and runs Docker containers

https://docs.docker.com/engine/

Docker Hub

- a website / web service
- a central repository to store and share Docker container images (commercial)
- other container image registries exist

https://hub.docker.com/

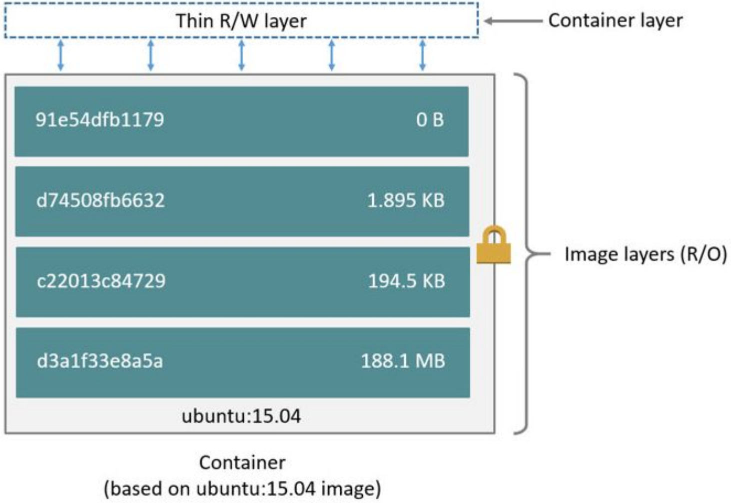
Docker **image** and **container**: what’s the difference

Docker image

- a **read-only** snapshot of an environment
- organized in **layers**
- changing an image adds more layers
- can be stored on Dockerhub or as a file
- images can share identical layers
- can make your own with a **Dockerfile**

Docker container

- a **live instance** of a Docker image
- has a thin writable layer that dies with it
- **one image** can spawn **many containers**

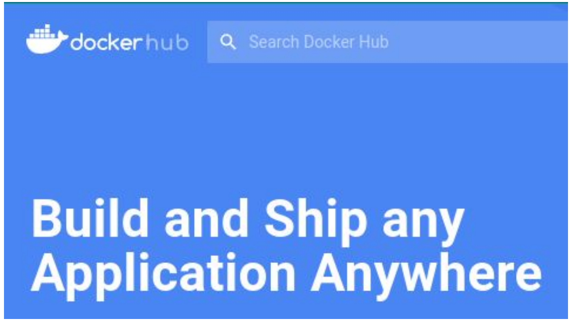


https://docs.docker.com/storage/storagedriver/

How can I get my own Docker container going

Use an **existing image**

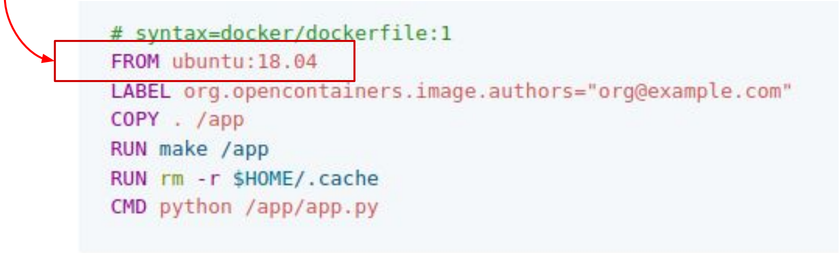
- Dockerhub: a repository of docker images https://hub.docker.com/
- You can pull an image with `docker pull`



Build and Ship any Application Anywhere

Build your **own image**

- a `Dockerfile` lets you describe the exact image you want to create
- Start from one image and add to it



```
# syntax=docker/dockerfile:1
FROM ubuntu:18.04
LABEL org.opencontainers.image.authors="org@example.com"
COPY . /app
RUN make /app
RUN rm -r $HOME/.cache
CMD python /app/app.py
```

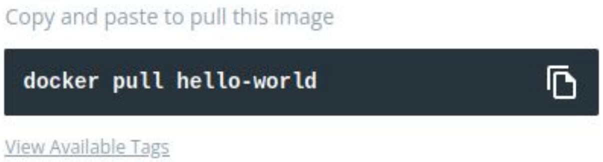
Let’s look at both

Exercise 1: Run a container from Dockerhub

- Find an image we like: https://hub.docker.com/_/hello-world
- Take a look at it
- Pull the image
- Run the image

Exercise 1: Run a container from Dockerhub

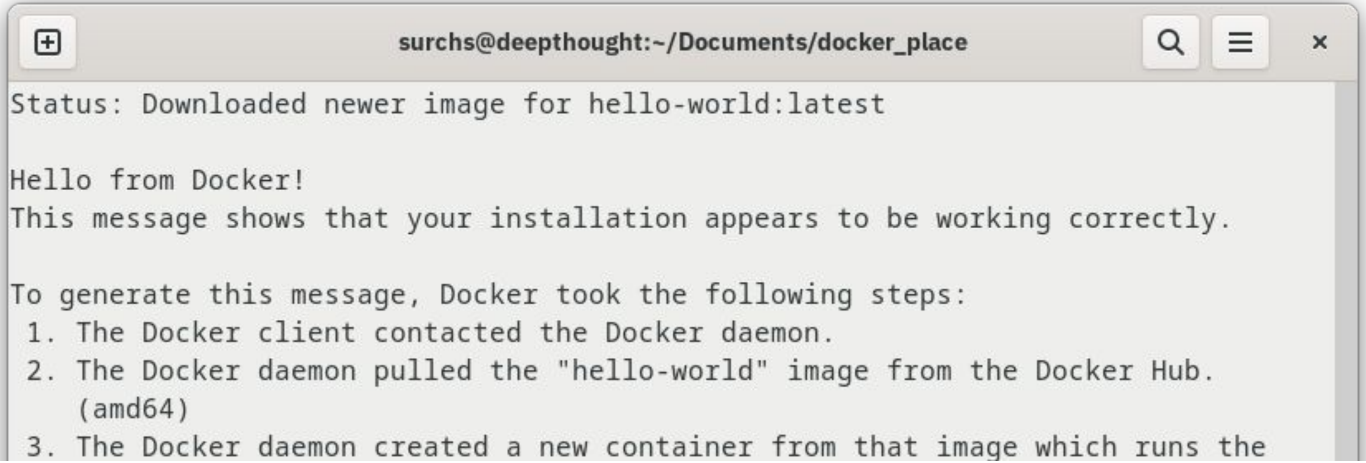
- Find an image we like: https://hub.docker.com/_/hello-world
- Take a look at it
- Pull the image
- Run the image



Copy and paste to pull this image

```
docker pull hello-world
```

View Available Tags



```
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
```

Exercise 1: Run a container from Dockerhub

Summary

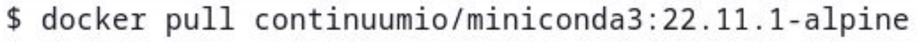
- Dockerhub has images
- Image tags are important
- We can
  - retrieve images with `docker pull`
  - create and immediately run a container from an image with `docker run`

Let’s find a more useful image

Exercise 2: work with a container that has conda

I don’t have conda installed on my system. But there is a Docker image. Let’s try!

- Find a Docker image: https://hub.docker.com/r/continuumio/miniconda3/
- Pick a tag, then pull the image



```
$ docker pull continuumio/miniconda3:22.11.1-alpine
```

- Run it

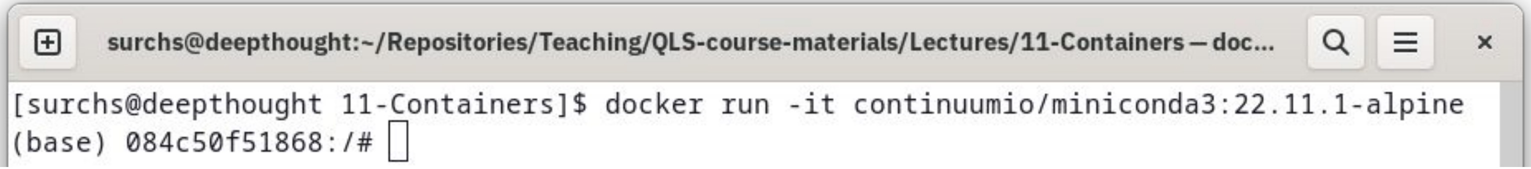
Exercise 2: work with a container that has conda

I don’t have conda installed on my system. But there is a Docker image. Let’s try!

- Find a Docker image: https://hub.docker.com/r/continuumio/miniconda3/
- Pick a tag, then pull the image `$ docker pull continuumio/miniconda3:22.11.1-alpine`
- Run it

Oh, nothing happened?

- Let’s run it interactively to take a look inside



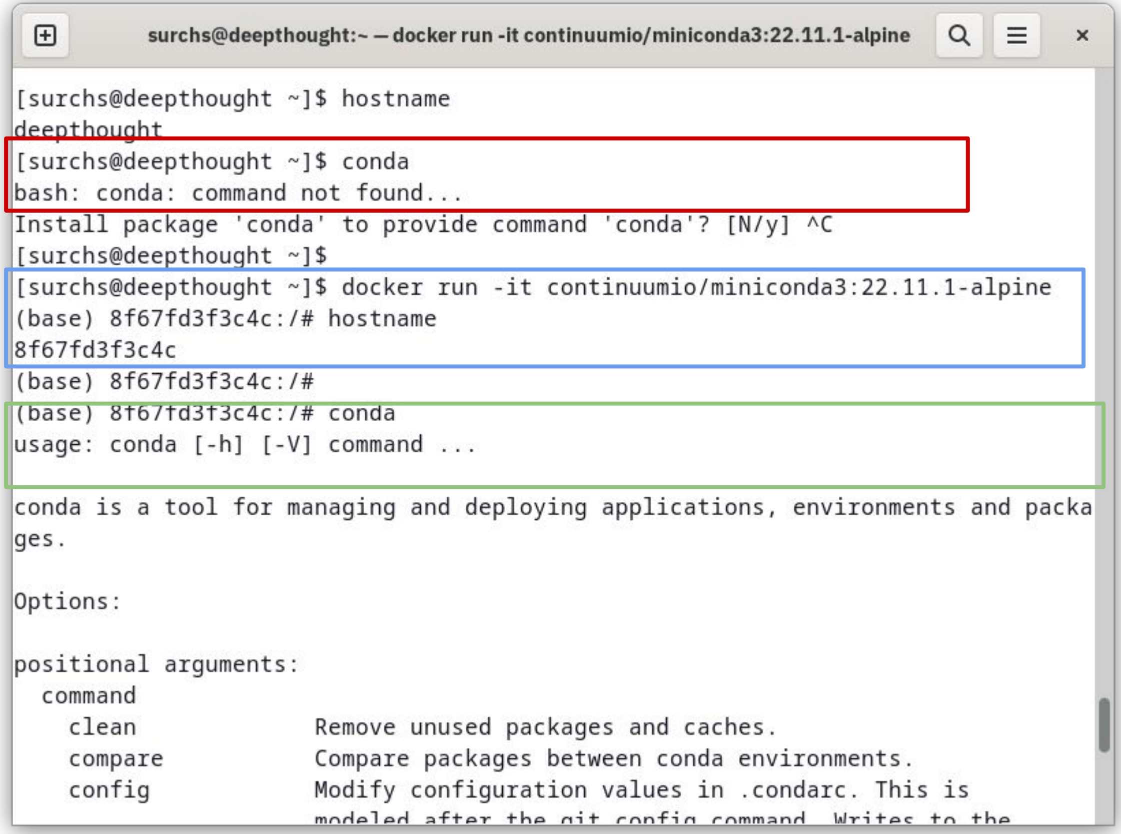
surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers — doc...

```
[surchs@deepthought 11-Containers]$ docker run -it continuumio/miniconda3:22.11.1-alpine
(base) 084c50f51868:/# 
```

https://github.com/ContinuumIO/docker-images/blob/master/miniconda3/alpine/Dockerfile

Looking around inside a container

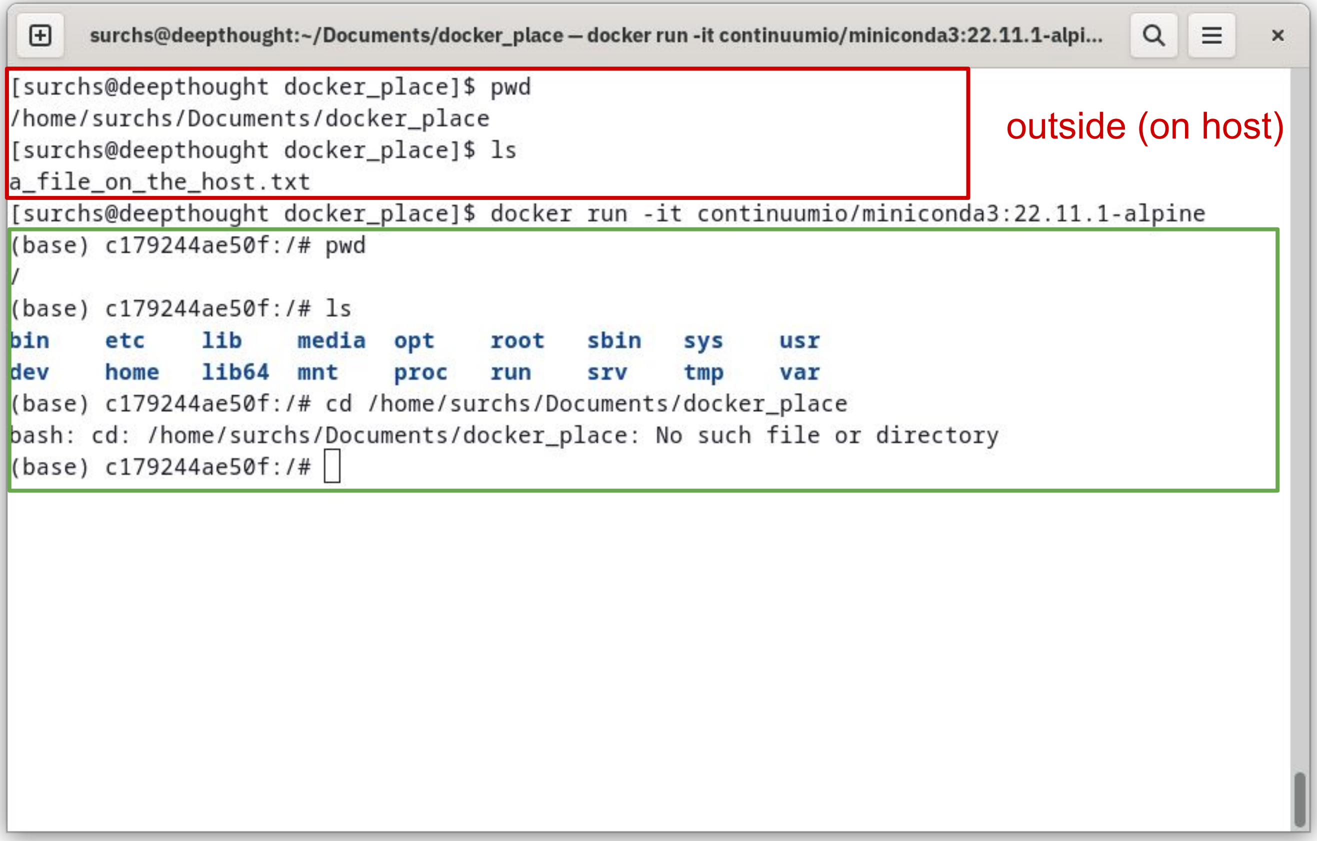
- no conda on my machine
- starting container changes the look of my terminal and the name of my computer
- inside of the container I have access to conda



surchs@deepthought:~ — docker run -it continuumio/miniconda3:22.11.1-alpine

```
[surchs@deepthought ~]$ hostname
deepthought
[surchs@deepthought ~]$ conda
bash: conda: command not found...
Install package 'conda' to provide command 'conda'? [N/y] ^C
[surchs@deepthought ~]$
[surchs@deepthought ~]$ docker run -it continuumio/miniconda3:22.11.1-alpine
(base) 8f67fd3f3c4c:/# hostname
8f67fd3f3c4c
(base) 8f67fd3f3c4c:/#
(base) 8f67fd3f3c4c:/# conda
usage: conda [-h] [-V] command ...

conda is a tool for managing and deploying applications, environments and packa
ges.

Options:

positional arguments:
  command
    clean              Remove unused packages and caches.
    compare            Compare packages between conda environments.
    config             Modify configuration values in .condarc. This is
                       modeled after the git config command. Writes to the
```

By default the container doesn’t see files on the host …



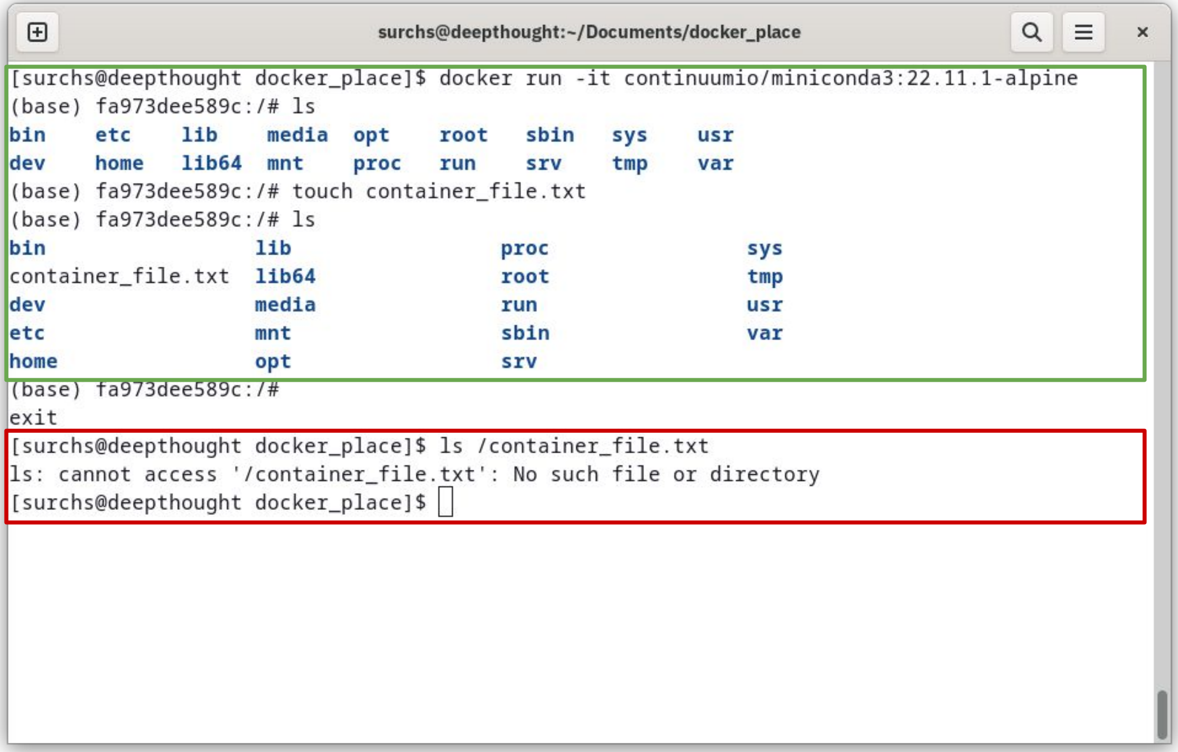
surchs@deepthought:~/Documents/docker_place — docker run -it continuumio/miniconda3:22.11.1-alpi...

```
[surchs@deepthought docker_place]$ pwd
/home/surchs/Documents/docker_place
[surchs@deepthought docker_place]$ ls
a_file_on_the_host.txt
[surchs@deepthought docker_place]$ docker run -it continuumio/miniconda3:22.11.1-alpine
(base) c179244ae50f:/# pwd
/
(base) c179244ae50f:/# ls
bin     etc     lib     media   opt     root    sbin    sys     usr
dev     home    lib64   mnt     proc    run     srv     tmp     var
(base) c179244ae50f:/# cd /home/surchs/Documents/docker_place
bash: cd: /home/surchs/Documents/docker_place: No such file or directory
(base) c179244ae50f:/# 
```

outside (on host)

inside
(in container)

… and the host can’t see files on the container …

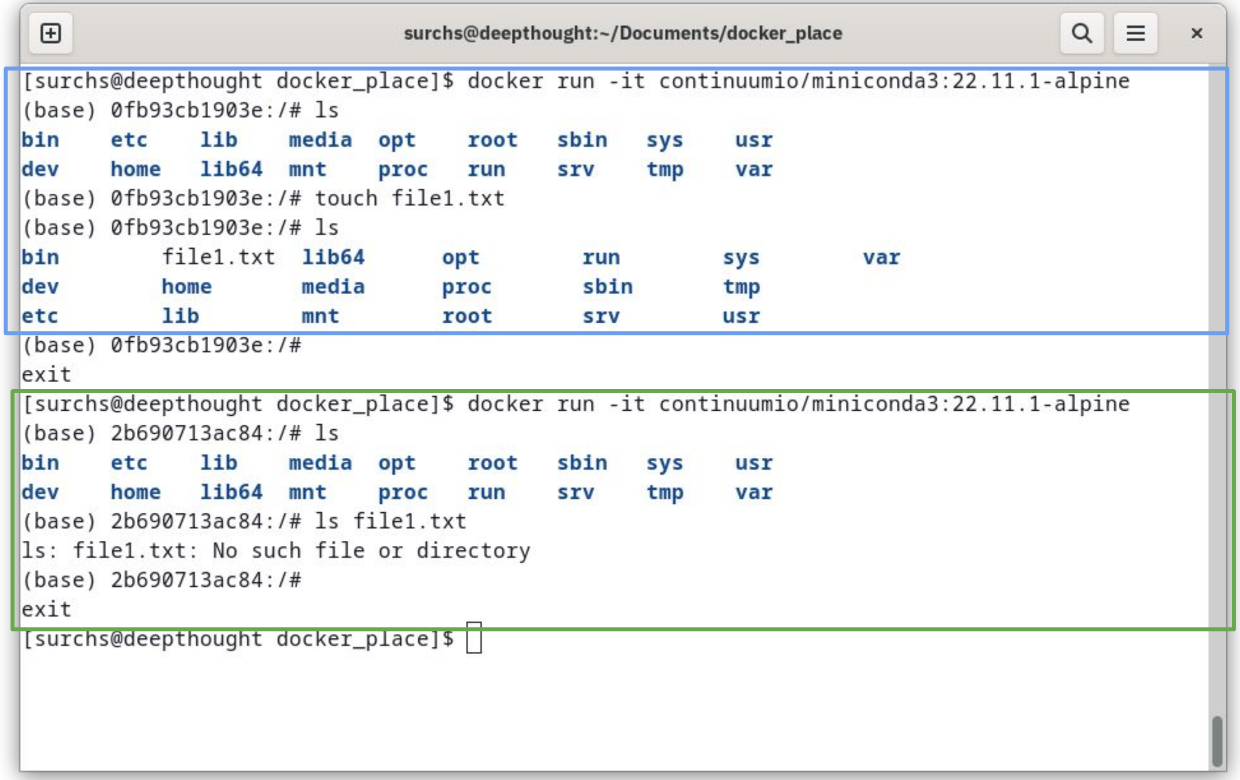


```
[surchs@deepthought docker_place]$ docker run -it continuumio/miniconda3:22.11.1-alpine
(base) fa973dee589c:/# ls
bin     etc     lib     media   opt     root    sbin    sys     usr
dev     home    lib64   mnt     proc    run     srv     tmp     var
(base) fa973dee589c:/# touch container_file.txt
(base) fa973dee589c:/# ls
bin                 lib                 proc                sys
container_file.txt  lib64               root                tmp
dev                 media               run                 usr
etc                 mnt                 sbin                var
home                opt                 srv
(base) fa973dee589c:/#
exit
[surchs@deepthought docker_place]$ ls /container_file.txt
ls: cannot access '/container_file.txt': No such file or directory
[surchs@deepthought docker_place]$
```

inside
(in container)

outside (on host)

… and files written to a container are tied to it!



```
[surchs@deepthought docker_place]$ docker run -it continuumio/miniconda3:22.11.1-alpine
(base) 0fb93cb1903e:/# ls
bin     etc     lib     media   opt     root    sbin    sys     usr
dev     home    lib64   mnt     proc    run     srv     tmp     var
(base) 0fb93cb1903e:/# touch file1.txt
(base) 0fb93cb1903e:/# ls
bin         file1.txt  lib64      opt        run        sys        var
dev         home       media      proc       sbin       tmp
etc         lib        mnt        root       srv        usr
(base) 0fb93cb1903e:/#
exit
[surchs@deepthought docker_place]$ docker run -it continuumio/miniconda3:22.11.1-alpine
(base) 2b690713ac84:/# ls
bin     etc     lib     media   opt     root    sbin    sys     usr
dev     home    lib64   mnt     proc    run     srv     tmp     var
(base) 2b690713ac84:/# ls file1.txt
ls: file1.txt: No such file or directory
(base) 2b690713ac84:/#
exit
[surchs@deepthought docker_place]$ 
```

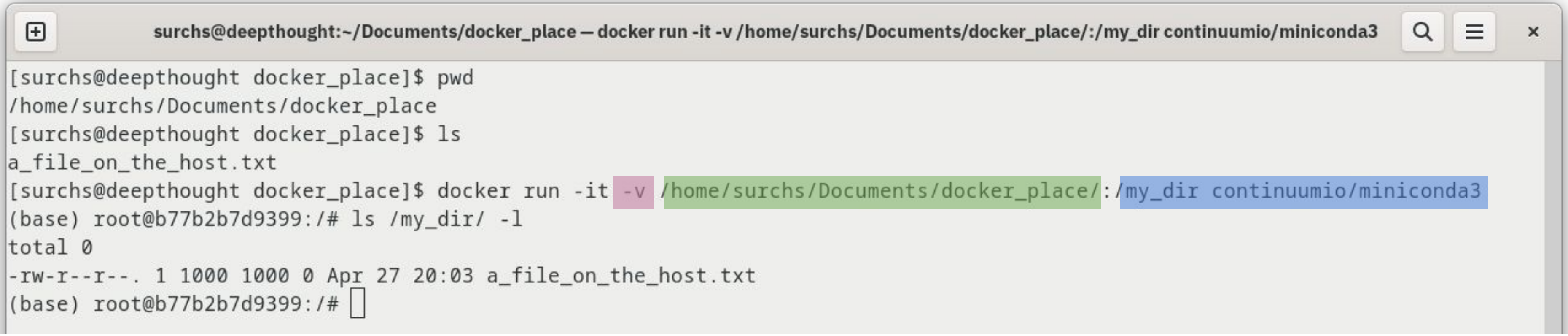
first container instance

second container instance

-> Don’t write anything (worth keeping) to a container

How do we share files between host and container?

- Bind-mount a path on the host to a path in the container


surchs@deepthought:~/Documents/docker_place — docker run -it -v /home/surchs/Documents/docker_place/:/my_dir continuumio/miniconda3

```
[surchs@deepthought docker_place]$ pwd
/home/surchs/Documents/docker_place
[surchs@deepthought docker_place]$ ls
a_file_on_the_host.txt
[surchs@deepthought docker_place]$ docker run -it -v /home/surchs/Documents/docker_place/:/my_dir continuumio/miniconda3
(base) root@b77b2b7d9399:/# ls /my_dir/ -l
total 0
-rw-r--r--. 1 1000 1000 0 Apr 27 20:03 a_file_on_the_host.txt
(base) root@b77b2b7d9399:/# 
```

- The bind-mount will be created in the container, even if it exists already!
- Make sure you provide a path (starts with / or ./) - or use `--mount`

https://docs.docker.com/storage/bind-mounts/

Exercise 2: work with a container that has conda

Summary

- We can connect to an interactive shell in a container with `docker run -it`
- By default the container cannot see or write the filesystem of the host
- We can "mount" a path on the host into the container with

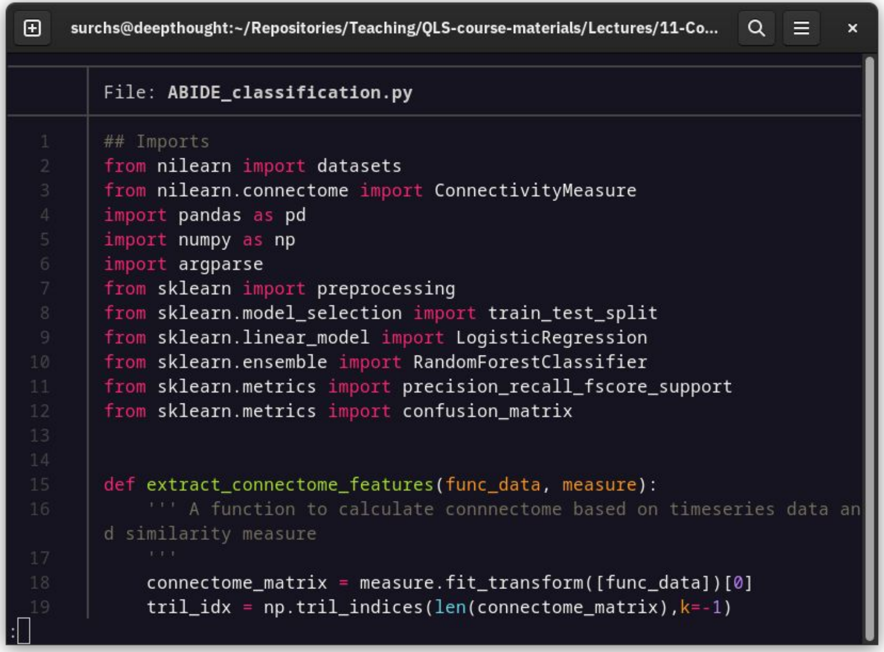
  ```
  docker run -v /host/path:/container/path OR
  docker run --mount type=bind,source=/host/path,target=/container/path
  ```

- It's a bad idea to write into the container directly

So how do I make persistent changes to the image?

Exercise 3: Containerize a classifier

- I want to run the classification script from `08-Machine_Learning_1`
- but I did not setup all of the requirements to run it



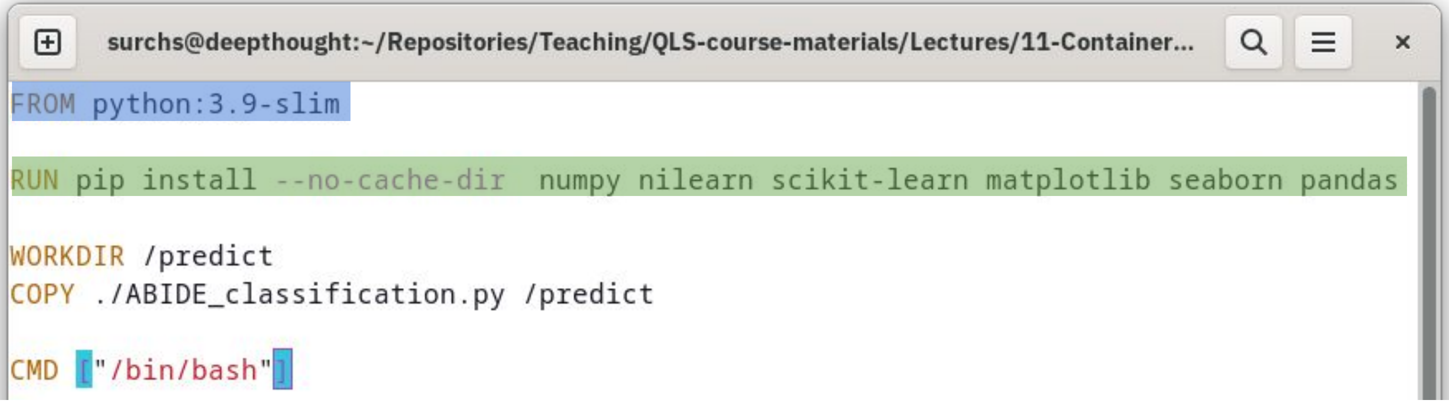
surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Co...

```
File: ABIDE_classification.py

## Imports
from nilearn import datasets
from nilearn.connectome import ConnectivityMeasure
import pandas as pd
import numpy as np
import argparse
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import confusion_matrix


def extract_connectome_features(func_data, measure):
    ''' A function to calculate connnectome based on timeseries data and similarity measure
    '''
    connectome_matrix = measure.fit_transform([func_data])[0]
    tril_idx = np.tril_indices(len(connectome_matrix),k=-1)
:
```

docker

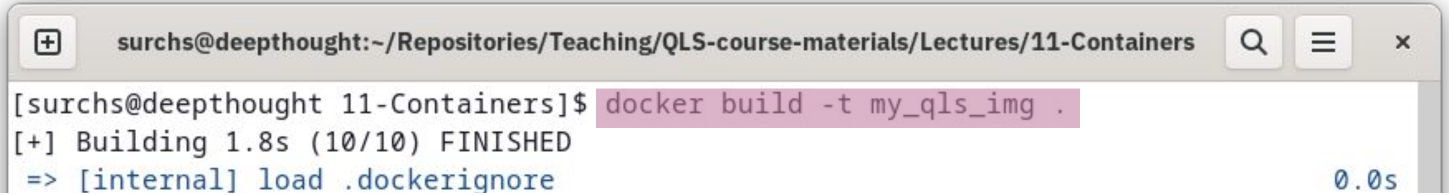
Exercise 3: Containerize a classifier

1. Start **from** an existing image
2. Define **changes** on top of that with a **Dockerfile**
3. Build a new image using the **docker build** command



surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Container...

```
FROM python:3.9-slim

RUN pip install --no-cache-dir  numpy nilearn scikit-learn matplotlib seaborn pandas

WORKDIR /predict
COPY ./ABIDE_classification.py /predict

CMD ["/bin/bash"]
```

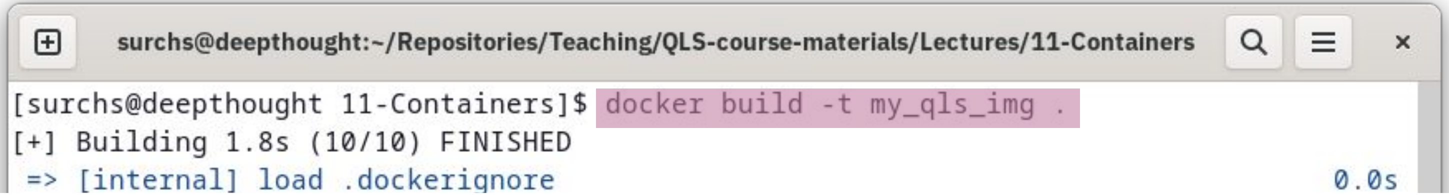


surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers

```
[surchs@deepthought 11-Containers]$ docker build -t my_qls_img .
[+] Building 1.8s (10/10) FINISHED
 => [internal] load .dockerignore                                            0.0s
```

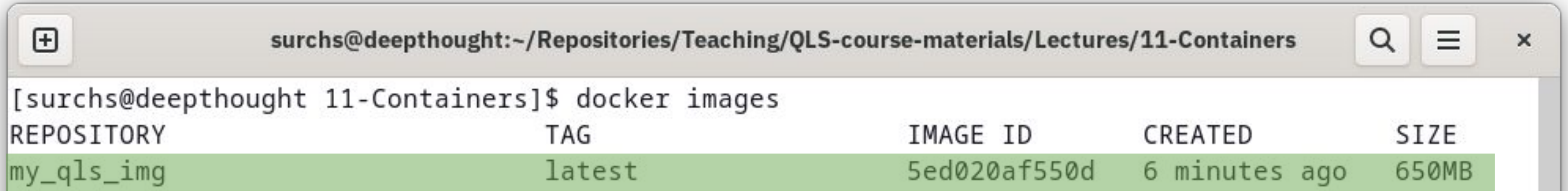
Exercise 3: Containerize a classifier

4. We have **built our own docker image and stored it locally** on the computer



surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers

```
[surchs@deepthought 11-Containers]$ docker build -t my_qls_img .
[+] Building 1.8s (10/10) FINISHED
 => [internal] load .dockerignore                                    0.0s
```

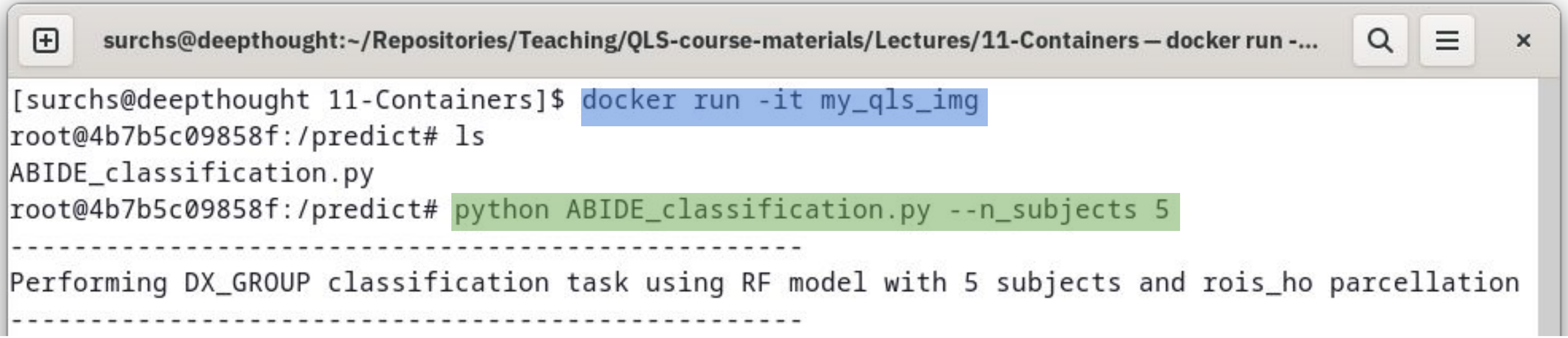


surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers

```
[surchs@deepthought 11-Containers]$ docker images
REPOSITORY                    TAG                 IMAGE ID       CREATED          SIZE
my_qls_img                    latest              5ed020af550d   6 minutes ago    650MB
```

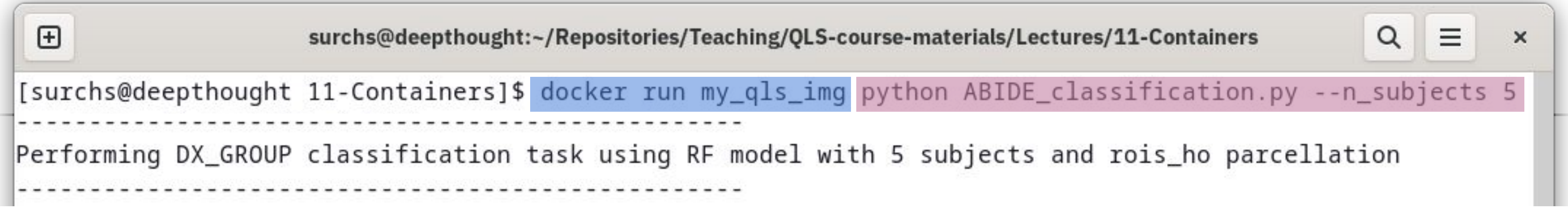
Run the classifier script

1. **Run the container and attach** an interactive shell
2. Inside the container, **run the classification script** like we would on the host


surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers — docker run -...

```
[surchs@deepthought 11-Containers]$ docker run -it my_qls_img
root@4b7b5c09858f:/predict# ls
ABIDE_classification.py
root@4b7b5c09858f:/predict# python ABIDE_classification.py --n_subjects 5
------------------------------------------------------
Performing DX_GROUP classification task using RF model with 5 subjects and rois_ho parcellation
------------------------------------------------------
```

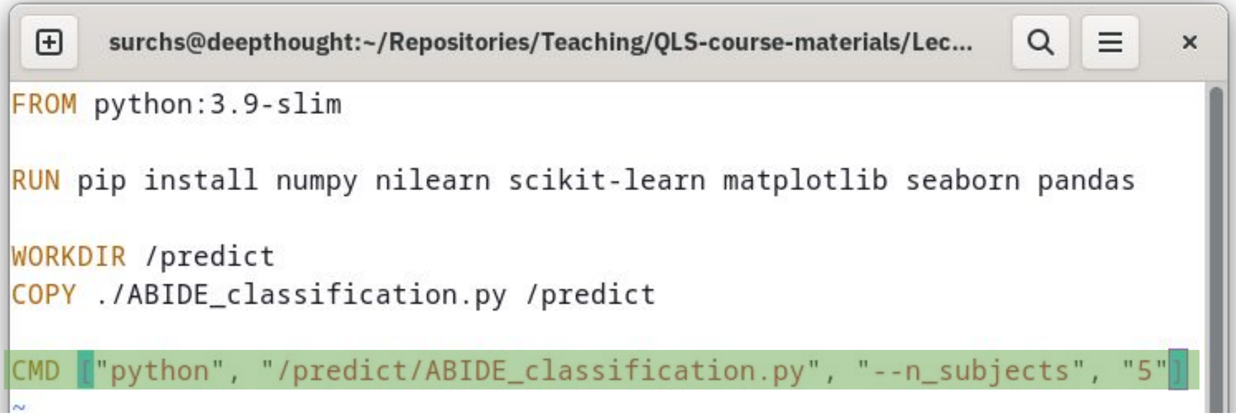
3. We can also **run the container** and then **run the script as an argument**


surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers

```
[surchs@deepthought 11-Containers]$ docker run my_qls_img python ABIDE_classification.py --n_subjects 5
------------------------------------------------------
Performing DX_GROUP classification task using RF model with 5 subjects and rois_ho parcellation
------------------------------------------------------
```

I keep writing the same command, what if I didn't have to?

1. The `CMD` **keyword in the Dockerfile** defines a default command
2. Unless the user provides their own command, the default CMD is run



surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lec...

```
FROM python:3.9-slim

RUN pip install numpy nilearn scikit-learn matplotlib seaborn pandas

WORKDIR /predict
COPY ./ABIDE_classification.py /predict

CMD ["python", "/predict/ABIDE_classification.py", "--n_subjects", "5"]
~
```

3. To reflect the changes in the Dockerfile, I have to **rebuild** my image

Exercise 3: Containerize a classifier

First summary:

- A `Dockerfile` lets us specify exactly what environment we want in a container
- `CMD` defines a default command that runs if the user doesn’t provide one
- We can build our own docker image on top of an existing one
- `docker build` creates the specified docker image on our host computer

Exercise 3: Containerize a classifier

First summary:

- A `Dockerfile` lets us specify exactly what environment we want in a container
- `CMD` defines a default c ser doesn’t provide one
- We can build our ow existing one
- `docker build` creat ge on our host computer

Don’t write anything to a container

Every time I run a new container, I have to download my data again …

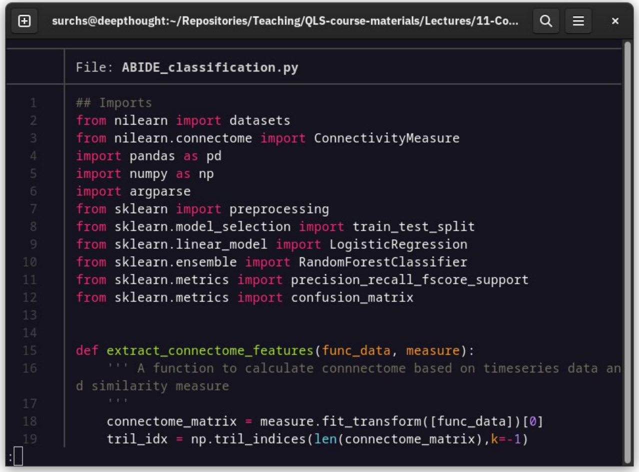
Loading or storing data in a container is not great

docker

ABIDE

Autism Brain Imaging Data Exchange

PREPROCESSED



surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Co...

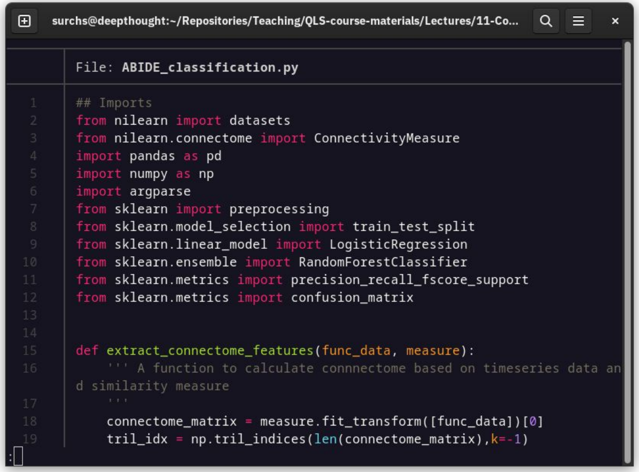
```
File: ABIDE_classification.py

## Imports
from nilearn import datasets
from nilearn.connectome import ConnectivityMeasure
import pandas as pd
import numpy as np
import argparse
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import confusion_matrix


def extract_connectome_features(func_data, measure):
    ''' A function to calculate connnectome based on timeseries data an
d similarity measure
    '''
    connectome_matrix = measure.fit_transform([func_data])[0]
    tril_idx = np.tril_indices(len(connectome_matrix),k=-1)
:
```

not persistent, not shareable with other container, slow, …

Containers are for processes, not data

docker

ABIDE
Autism Brain Imaging
Data Exchange
PREPROCESSED



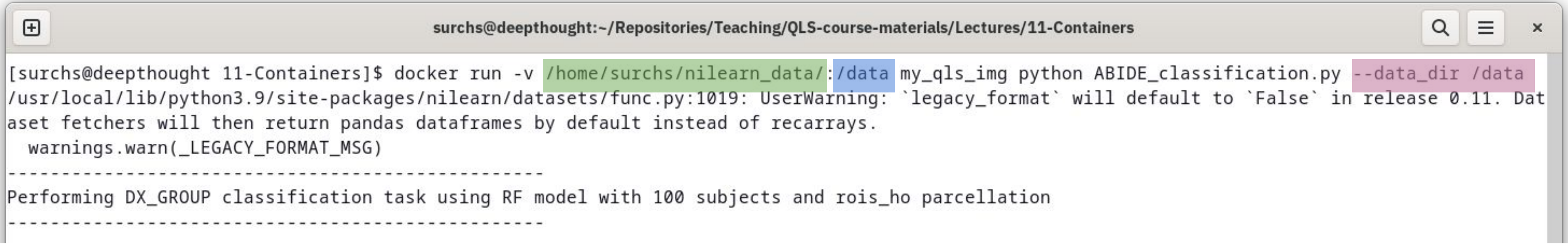
surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Co...

```
File: ABIDE_classification.py

## Imports
from nilearn import datasets
from nilearn.connectome import ConnectivityMeasure
import pandas as pd
import numpy as np
import argparse
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import precision_recall_fscore_support
from sklearn.metrics import confusion_matrix


def extract_connectome_features(func_data, measure):
    ''' A function to calculate connnectome based on timeseries data and similarity measure
    '''
    connectome_matrix = measure.fit_transform([func_data])[0]
    tril_idx = np.tril_indices(len(connectome_matrix),k=-1)
:
```

-> let’s bind-mount the data into the container from our host machine

Provide the downloaded data via bind-mount

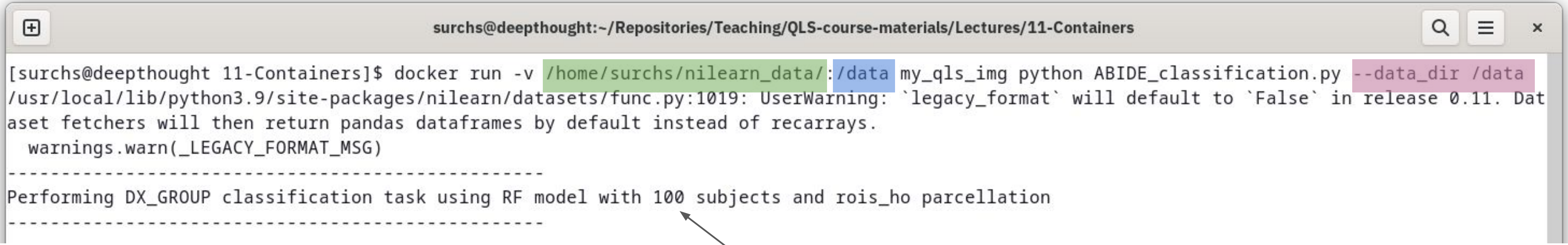
1. Bind the **path to the data on your computer** to a **path in the container**
2. Provide the path in the container **to the classifier script**
3. The script inside the container only sees the mounted path


surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers

```
[surchs@deepthought 11-Containers]$ docker run -v /home/surchs/nilearn_data/:/data my_qls_img python ABIDE_classification.py --data_dir /data
/usr/local/lib/python3.9/site-packages/nilearn/datasets/func.py:1019: UserWarning: `legacy_format` will default to `False` in release 0.11. Dat
aset fetchers will then return pandas dataframes by default instead of recarrays.
  warnings.warn(_LEGACY_FORMAT_MSG)
--------------------------------------------------
Performing DX_GROUP classification task using RF model with 100 subjects and rois_ho parcellation
--------------------------------------------------
```

Provide the downloaded data via bind-mount

1. Bind the **path to the data on your computer** to a **path in the container**
2. Provide the path in the container **to the classifier script** as a parameter
3. The script inside the container only sees the mounted path


surchs@deepthought:~/Repositories/Teaching/QLS-course-materials/Lectures/11-Containers

```
[surchs@deepthought 11-Containers]$ docker run -v /home/surchs/nilearn_data/:/data my_qls_img python ABIDE_classification.py --data_dir /data
/usr/local/lib/python3.9/site-packages/nilearn/datasets/func.py:1019: UserWarning: `legacy_format` will default to `False` in release 0.11. Dat
aset fetchers will then return pandas dataframes by default instead of recarrays.
  warnings.warn(_LEGACY_FORMAT_MSG)
--------------------------------------------------
Performing DX_GROUP classification task using RF model with 100 subjects and rois_ho parcellation
--------------------------------------------------
```

Bonus: why is the classifier running with 100 subjects now?

Exercise 3: Containerize a classifier

- We have a `Dockerfile` that describes our desired environment
- We have built a docker image that contains all dependencies and the code
- We have a default command so a user can just run a container
- **-> We have containerized the classifier**

Exercise 3: Containerize a classifier

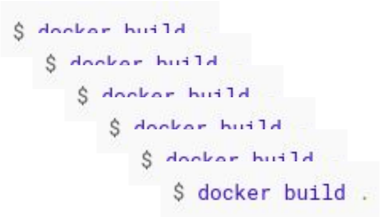
- We have a `Dockerfile` that describes our desired environment
- We have built a docker image that contains all dependencies and the code
- We have a default command so a user can just run a container
- **-> We have containerized the classifier**

What if I'd like to make a change to the code?

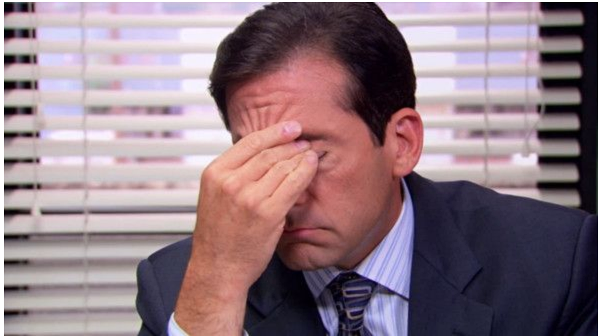
Exercise 3: Containerize a classifier

- We have a `Dockerfile` that describes our desired environment
- We have built a docker image that contains all dependencies and the code
- We have a default command so a user can just run a container
- **-> We have containerized the classifier**

What if I’d like to make a change to the code?



```
$ docker build .
$ docker build .
$ docker build .
$ docker build .
$ docker build .
$ docker build .
```

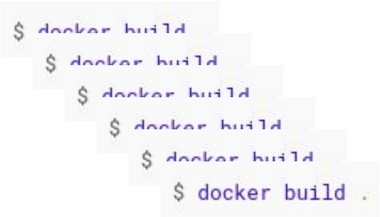


Exercise 3: Containerize a classifier

- We have a `Dockerfile` that describes our desired environme
- We have built a docker image that contains al
- We have a default command so a user can i
- **-> We have containerized the classifie**

The bind-mount will be created in the container, even if it exists already!

What if I'd like to make a change to t



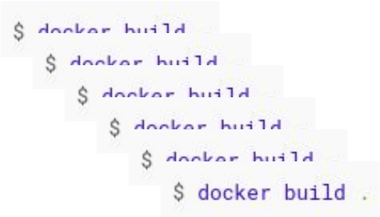
```
$ docker build .
$ docker build .
$ docker build .
$ docker build .
$ docker build .
$ docker build .
```

Exercise 3: Containerize a classifier

- We have a `Dockerfile` that describes our desired environme
- We have built a docker image that contains al
- We have a default command so a user can i
- **-> We have containerized the classifie**

The bind-mount will be created in the container, <u>even if it exists already</u>!

What if I’d like to make a change to t



```
$ docker build .
$ docker build .
$ docker build .
$ docker build .
$ docker build .
$ docker build .
```

Exercise 3: Keep developing the containerized classifier

- I can bind-mount my local source code over the copied code in the container
- I can keep editing my files locally and test them in the container
- When I build a new image, the most recent files will be copied into the image

🎉

Exercise 3: Containerize a classifier

Summary:

- Dockerfile + docker build create a **controlled image** for our code
- bind-mounts allow us to **expose data** to the container and store results
- We can combine **local editing** of source files and
  the **controlled environment** of the container by bind-mounting our source code
- -> We can use the **same environment** for development, analysis, and sharing

Exercise 3: Containerize a classifier

Summary:

- Dockerfile + docker build create a **controlled image** for our code
- bind-mounts allow us to **expose data** to the container and store results
- We can combine **local editing** of source files and
  the **controlled environment** of the container by bind-mounting our source code
- -> We can use the **same environment** for development, analysis, and sharing

Can I use my new container on a supercomputer?

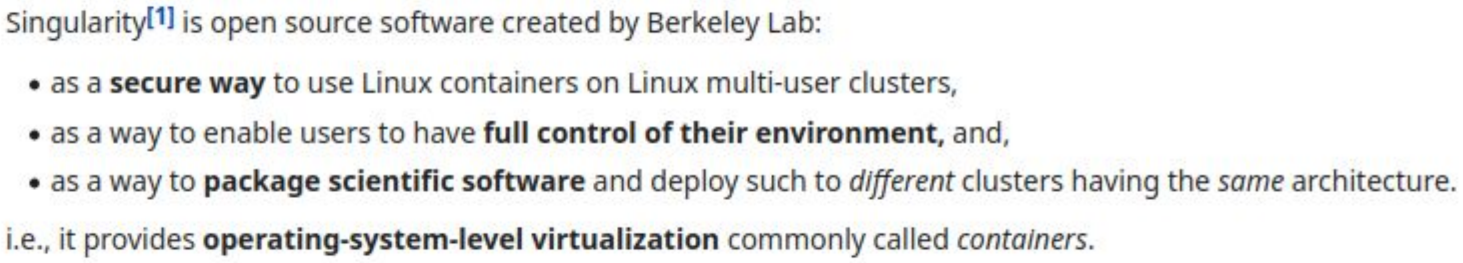
Singularity

Singularity is the container solution for HPCs

On shared systems (like a supercomputer), you shouldn’t / can’t use Docker

- Docker isn’t as isolated as a VM
- By default you run docker with root privileges and are root inside a container
- A malicious actor can escalate privileges and “break out” of a container

Apptainer (formerly Singularity) **is a container solution in these cases**



Singularity[1] is open source software created by Berkeley Lab:

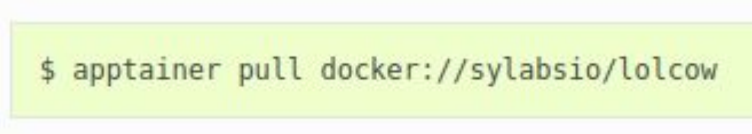
- as a **secure way** to use Linux containers on Linux multi-user clusters,
- as a way to enable users to have **full control of their environment,** and,
- as a way to **package scientific software** and deploy such to *different* clusters having the *same* architecture.

i.e., it provides **operating-system-level virtualization** commonly called *containers*.

https://apptainer.org/docs/user/main/ and https://apptainer.org/docs/user/main/security.html https://docs.alliancecan.ca/wiki/Singularity

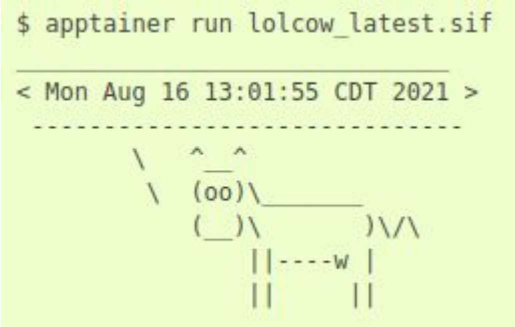
Build images with Docker, run them with Singularity

1. Pull an image from Dockerhub and create a local **S**ingularity**I**mage**F**ile



```
$ apptainer pull docker://sylabsio/lolcow
```

2. Run the Singularity File using **apptainer run**



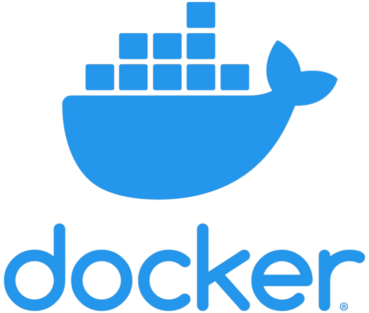
```
$ apptainer run lolcow_latest.sif
 ______________________________
< Mon Aug 16 13:01:55 CDT 2021 >
 ------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

https://apptainer.org/docs/user/main/quick_start.html#

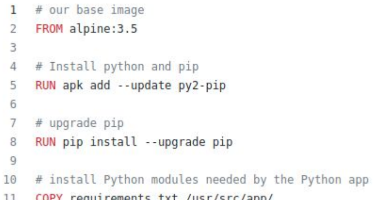
https://docs.alliancecan.ca/wiki/Singularity

Container Summary

- Docker **images** are **read-only snapshots**, you can find them on Dockerhub
- Images have tags (or version), that you should specify when pulling
- A Docker **container** is an isolated, **live instance** of an image
- Docker **containers** have their own file system, but this is **not persistent**
- With **volumes** we can **expose directories** on the host to the container
- We can **build** our own images on top of existing ones using a **Dockerfile**
- Use **Singularity/Apptainer** to run Docker images on a compute cluster



Docker engine

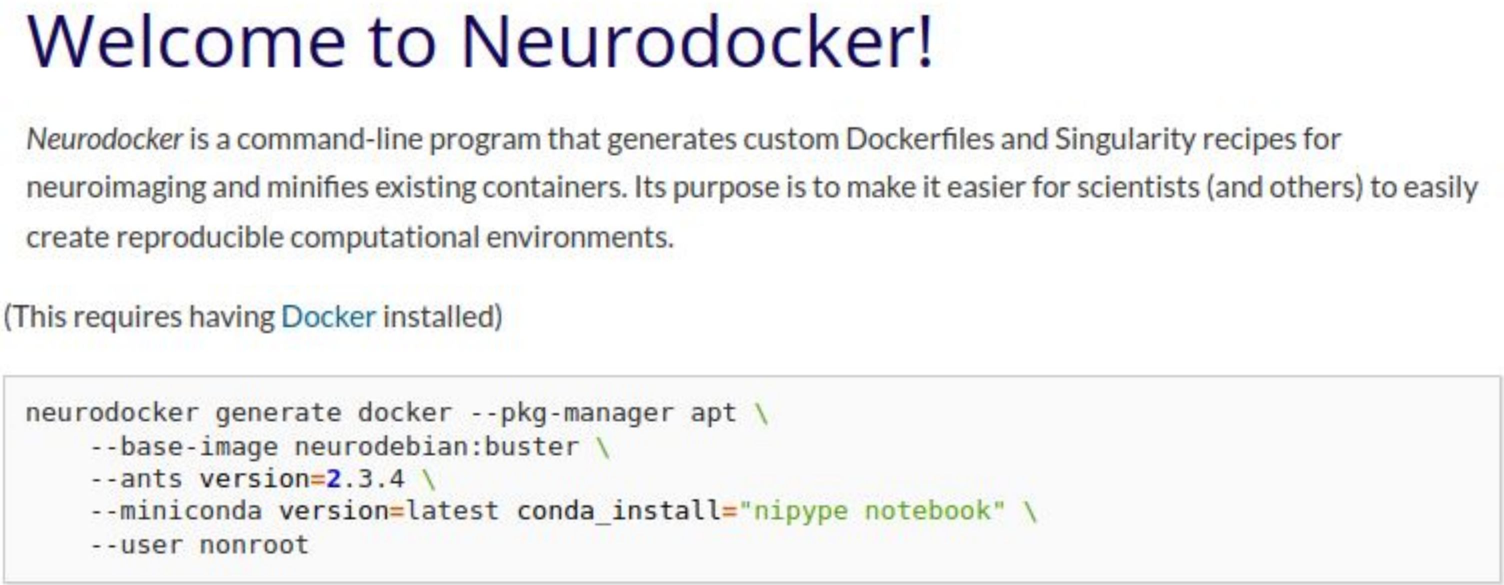


```
# our base image
FROM alpine:3.5

# Install python and pip
RUN apk add --update py2-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
```

Dockerfile

Docker image registry

There are tools to help make Dockerfiles



Welcome to Neurodocker!

Neurodocker is a command-line program that generates custom Dockerfiles and Singularity recipes for neuroimaging and minifies existing containers. Its purpose is to make it easier for scientists (and others) to easily create reproducible computational environments.

(This requires having Docker installed)

```
neurodocker generate docker --pkg-manager apt \
    --base-image neurodebian:buster \
    --ants version=2.3.4 \
    --miniconda version=latest conda_install="nipype notebook" \
    --user nonroot
```

https://www.repronim.org/neurodocker/user_guide/index.html

Additional Resources

- Docker tutorial https://github.com/docker/labs
- Neurohackweek container course https://neurohackweek.github.io/docker-for-scientists/
- The Turing Way on reproducible research environments https://the-turing-way.netlify.app/reproducible-research/renv.html