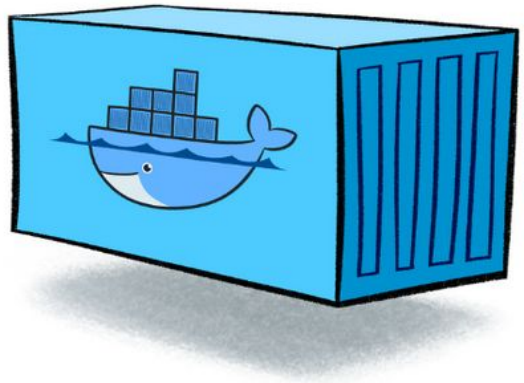
ORIGAMI
Lab

Containerization

QLSC 612 - May 2025
Alyssa Dai

Teaching material adapted from Sebastian Urchs



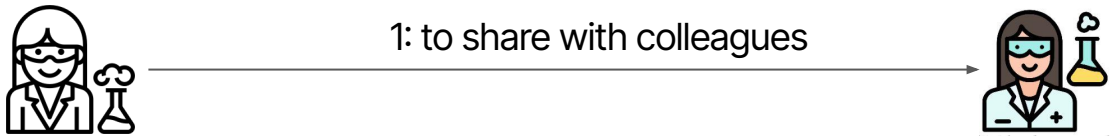
Containers?

What we will cover

1. **Why** are containers useful for researchers?
2. Using and building containers with **Docker**
3. Using containers on supercomputers with **Singularity**

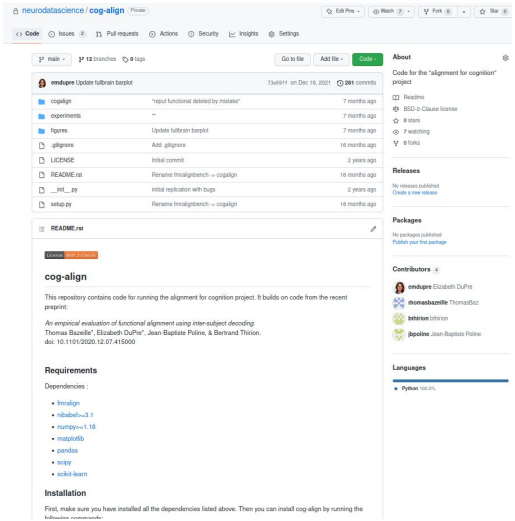
Why isolate environments?

Document your software environment



Document your software environment

1: to share with colleagues



Requirements

Dependencies :

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

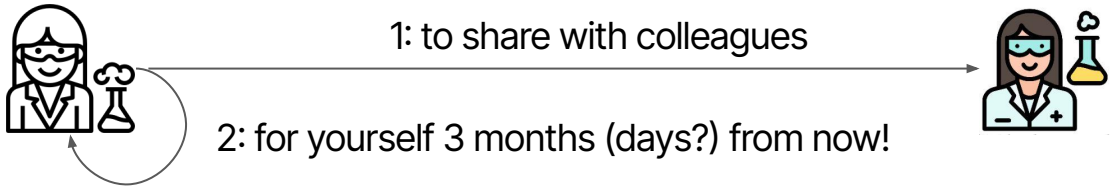
Installation

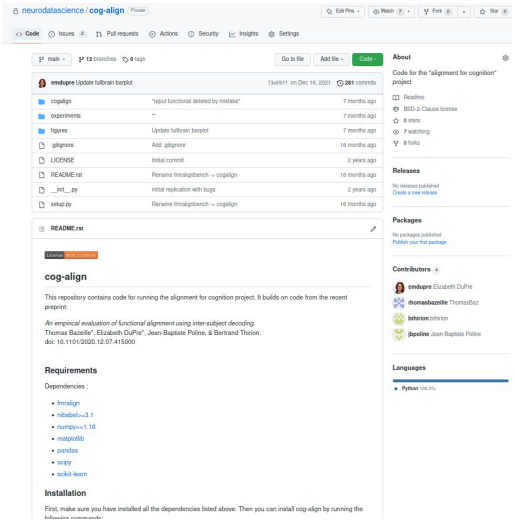
First, make sure you have installed all the dependencies listed above. Then you can install cog-align by running the following commands:

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

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

Document your software environment





Requirements

Dependencies :

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

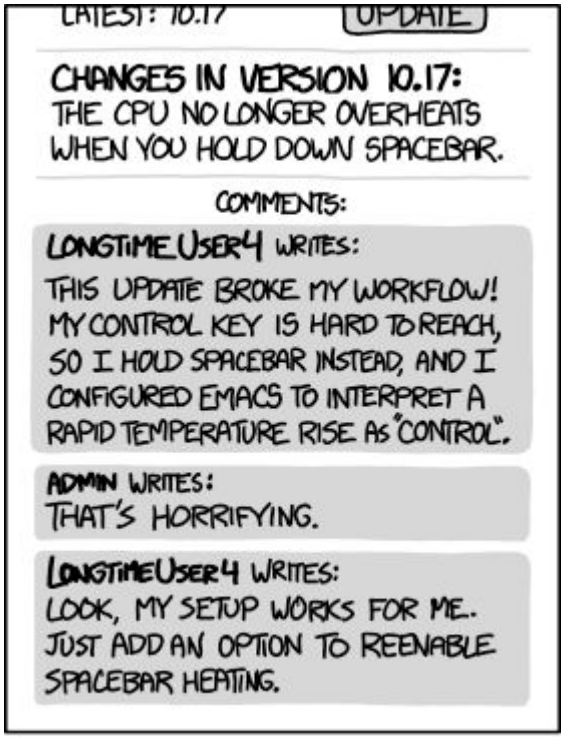
Installation

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

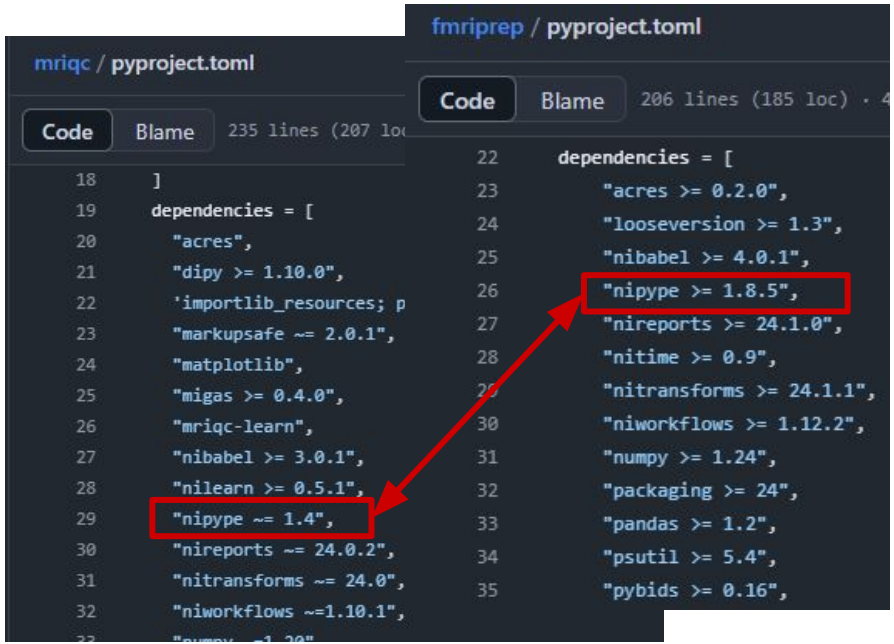


https://xkcd.com/1172/

Changing dependencies may do unexpected things

A. Your updated the dependencies of an existing project

Don’t use the same environment for all projects



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**

Reminder: 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! 🙂

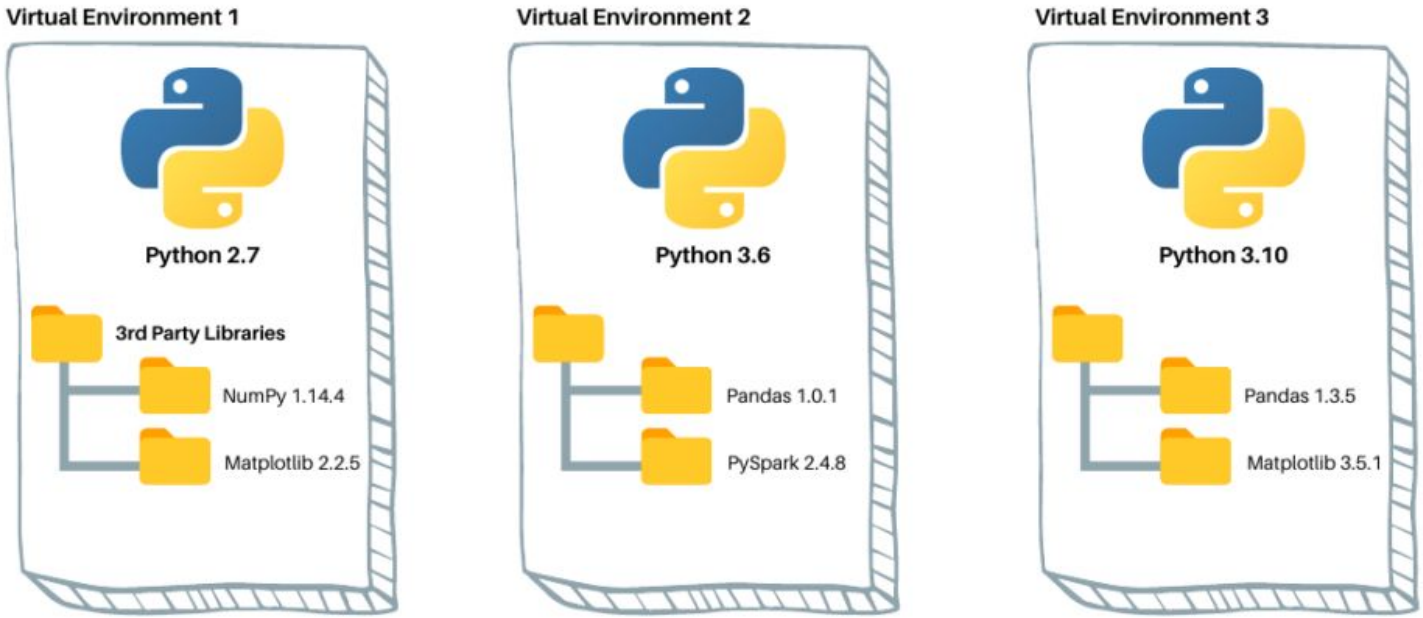
5 **Bake:** Bake the cream cheese pound cake at 325°F (163°C). Hal
the cake with aluminum foil to prevent over-browning.

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

Isolate Python environments



Why not just Python virtual environments?

External Dependencies

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

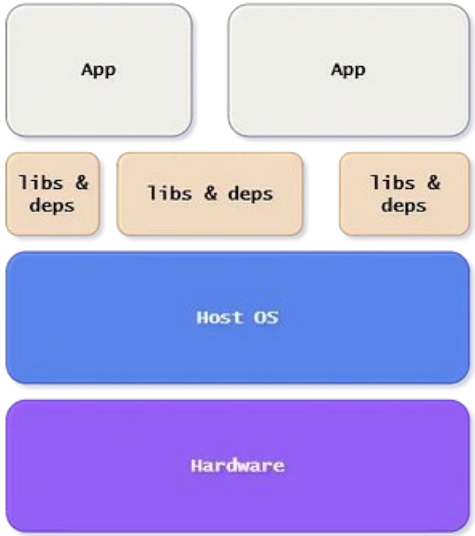
fMRIPrep requires some other neuroimaging software tools that are not handled by the Python's packaging system (Pypi) used to deploy the `fmriprep` package:

- 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

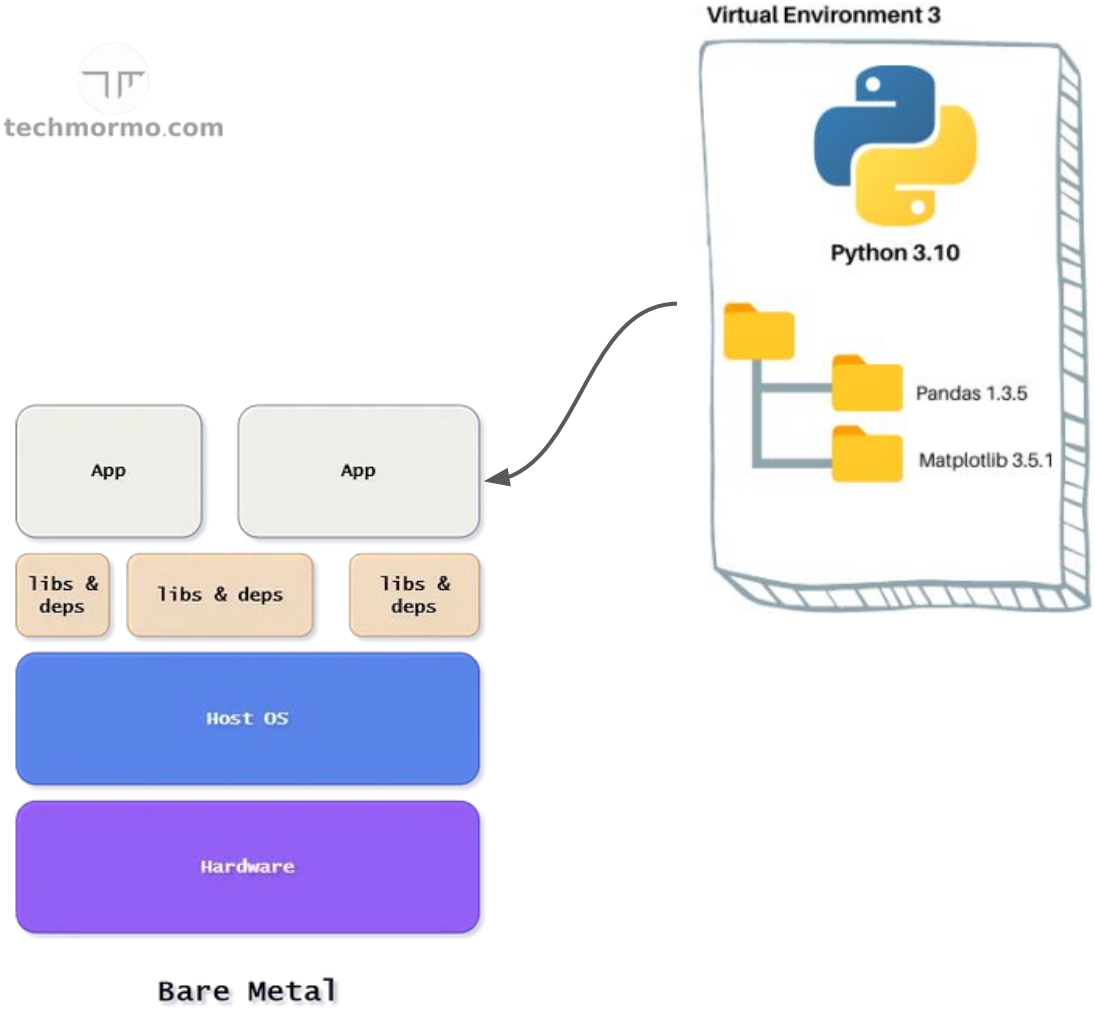
- Not all dependencies are supported in Anaconda
- Not everything is written in Python or R
- Your Operating System (**OS**) also has packages and a package manager
- The same version problems apply to these

Need to capture the (entire) compute environment

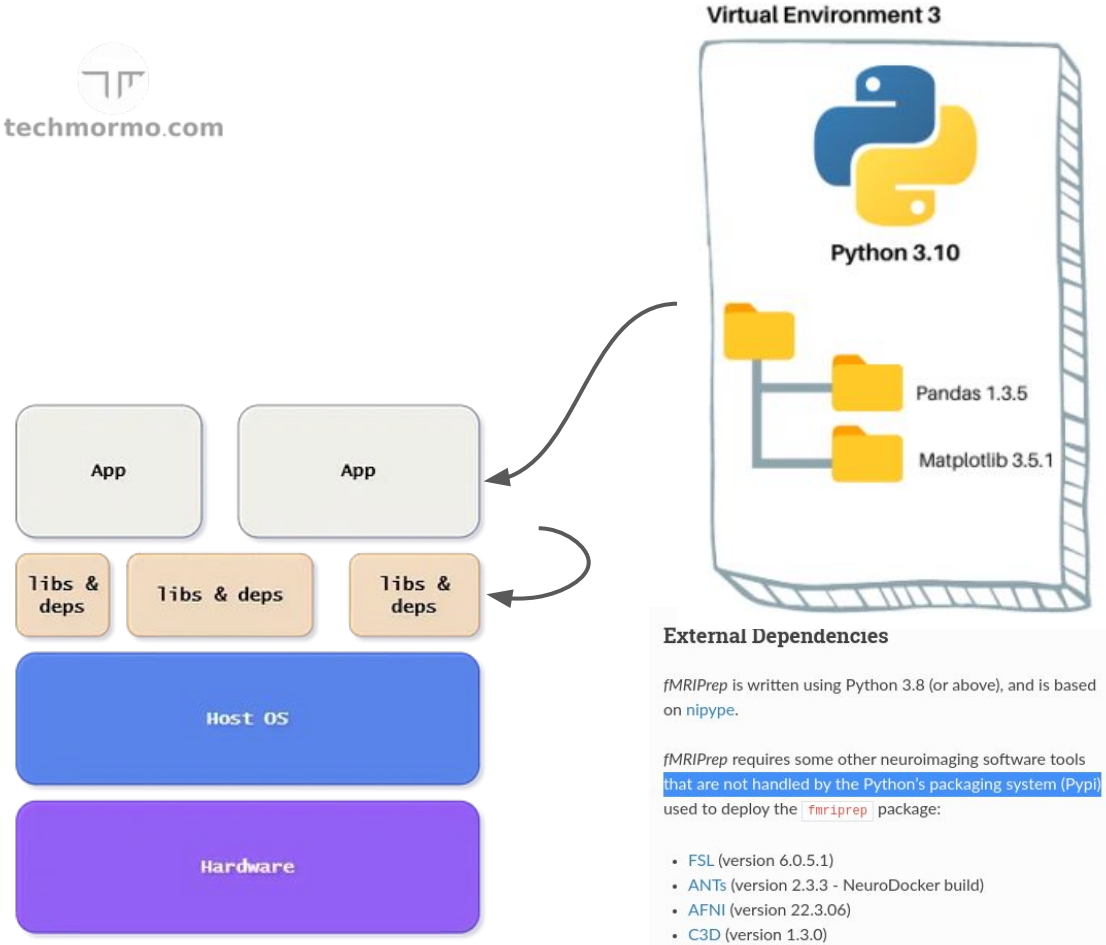


Bare Metal

techmormo.com



techmormo.com



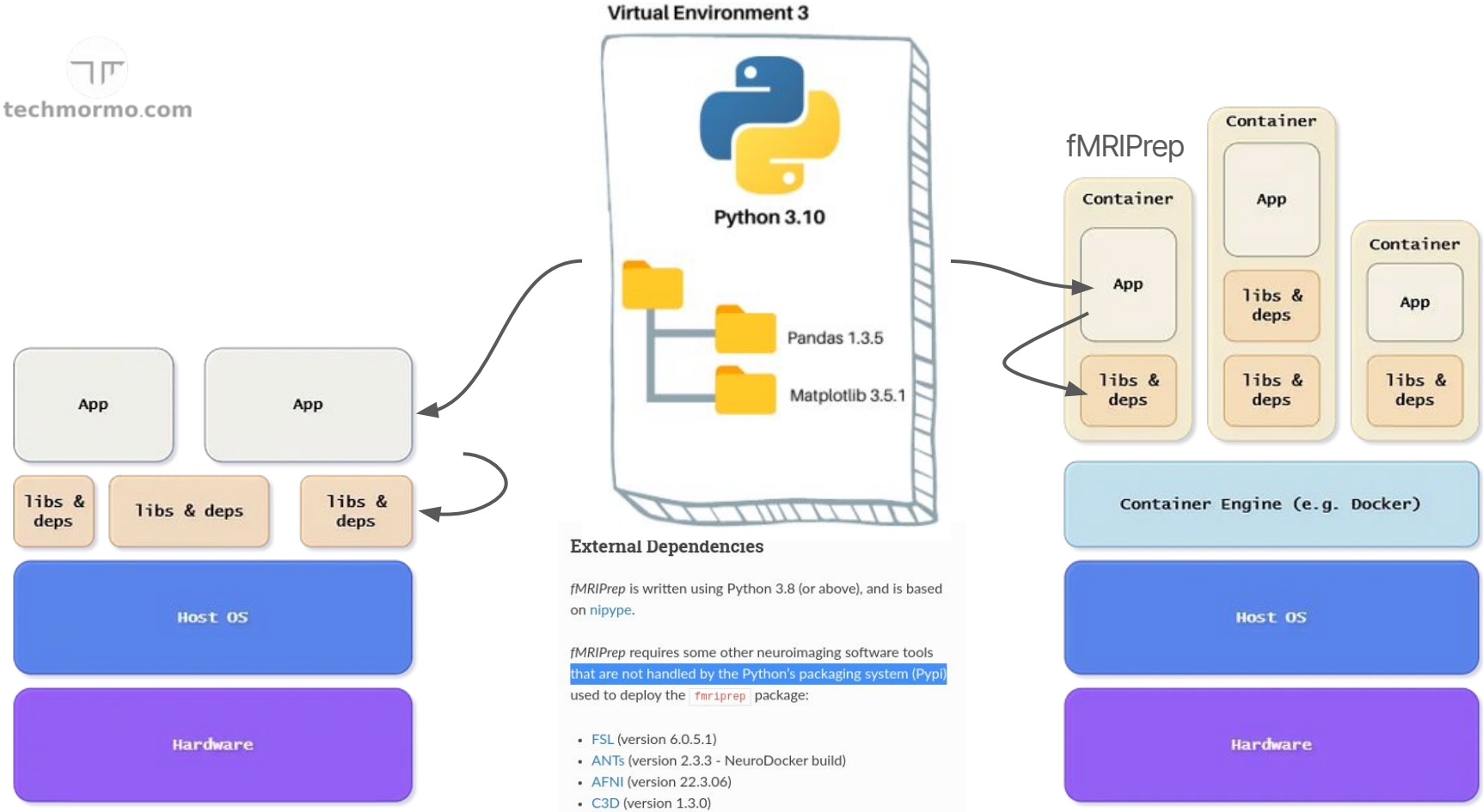
External Dependencies

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)

techmormo.com



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)

Containers

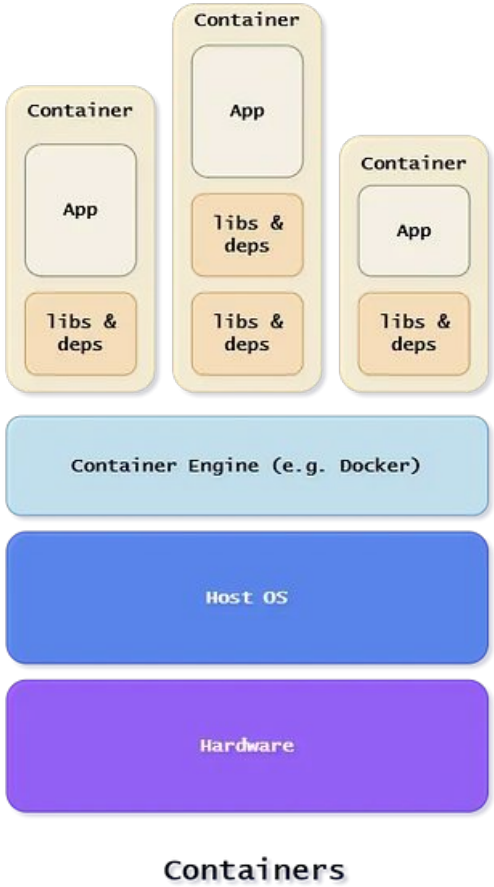
Containers

What are they

- isolated environments sharing the same host 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

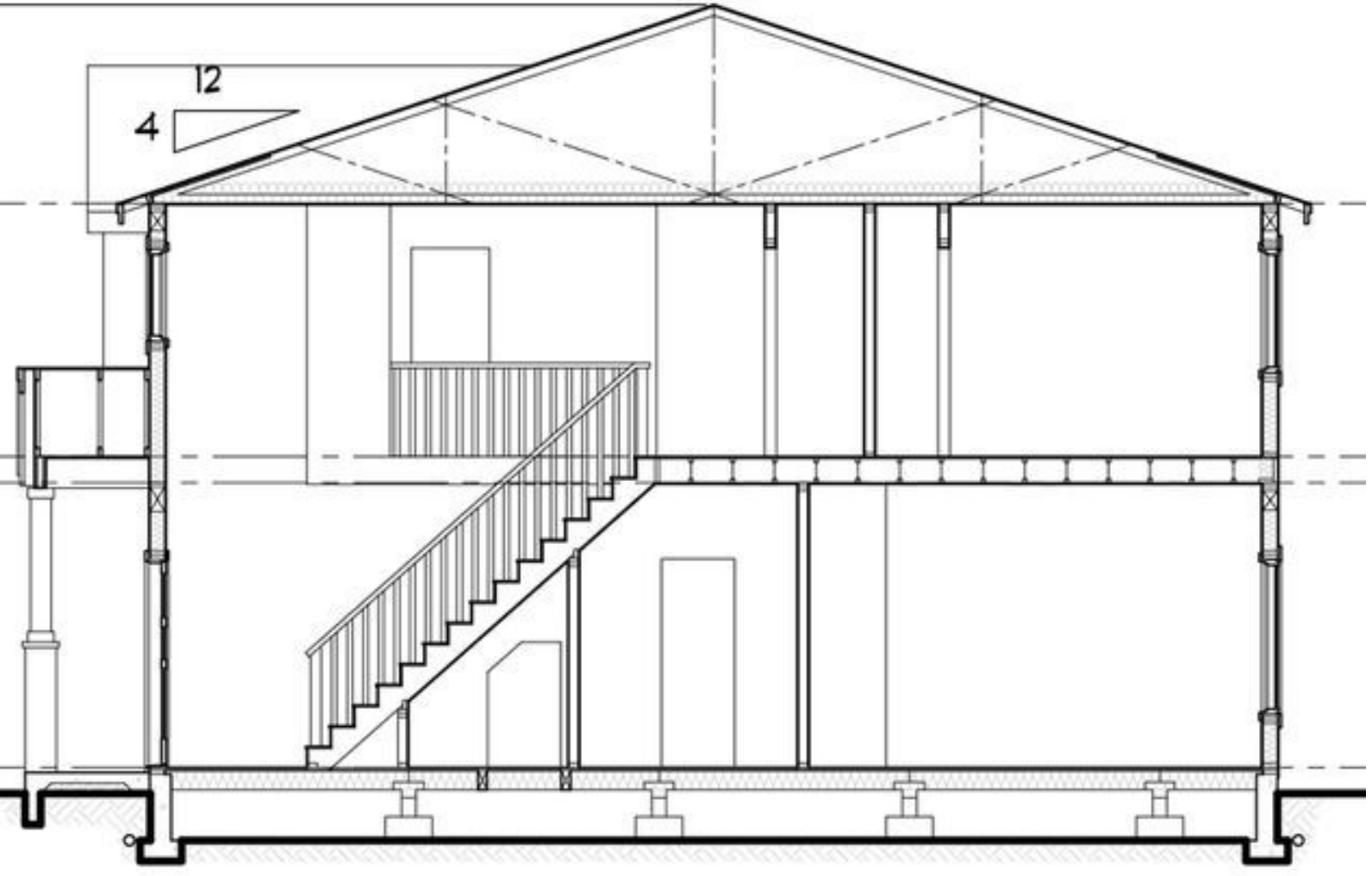
How do I make one

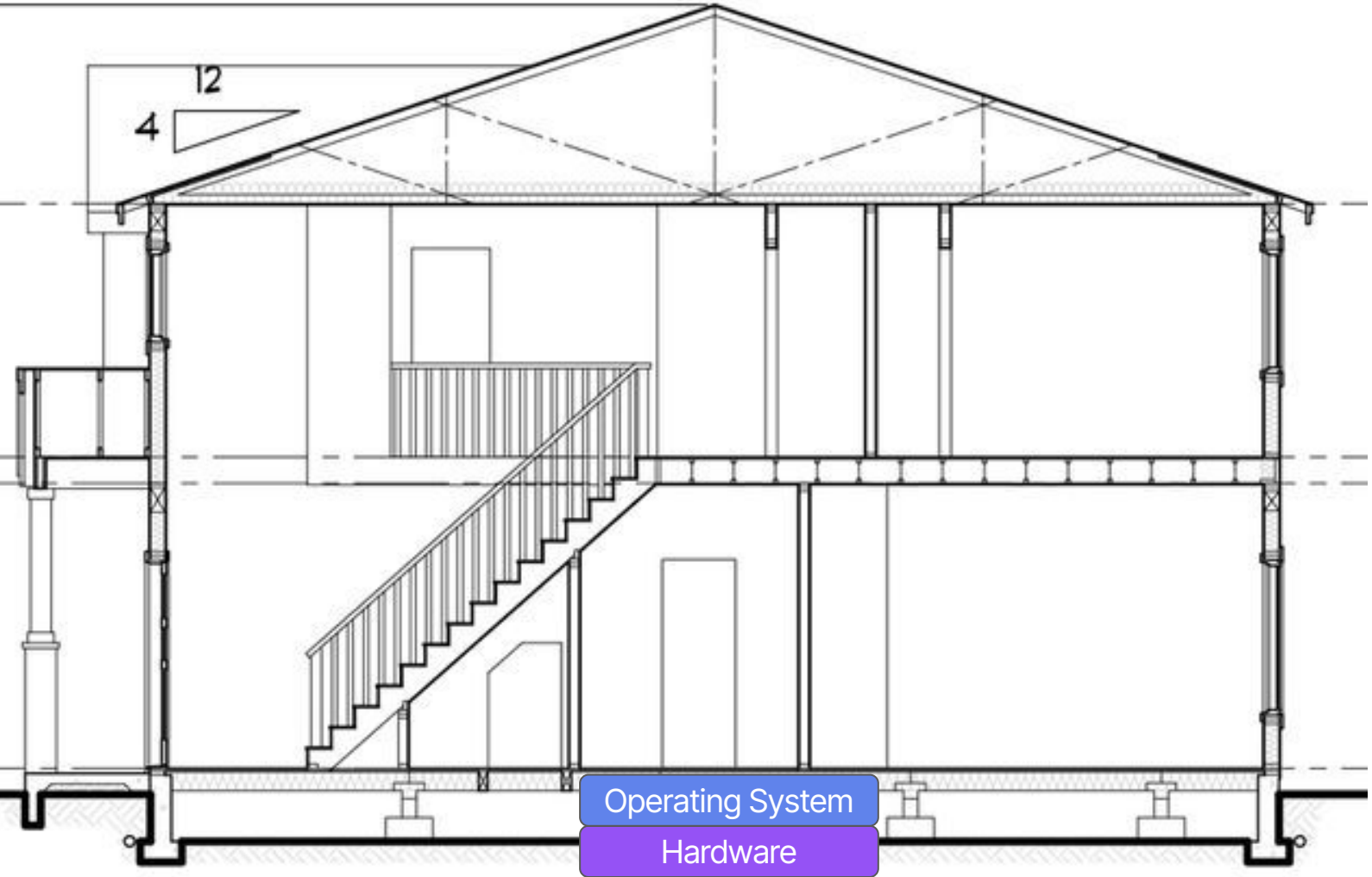
- use a container engine
- **Docker** is the most widely used
- Singularity is used on supercomputers

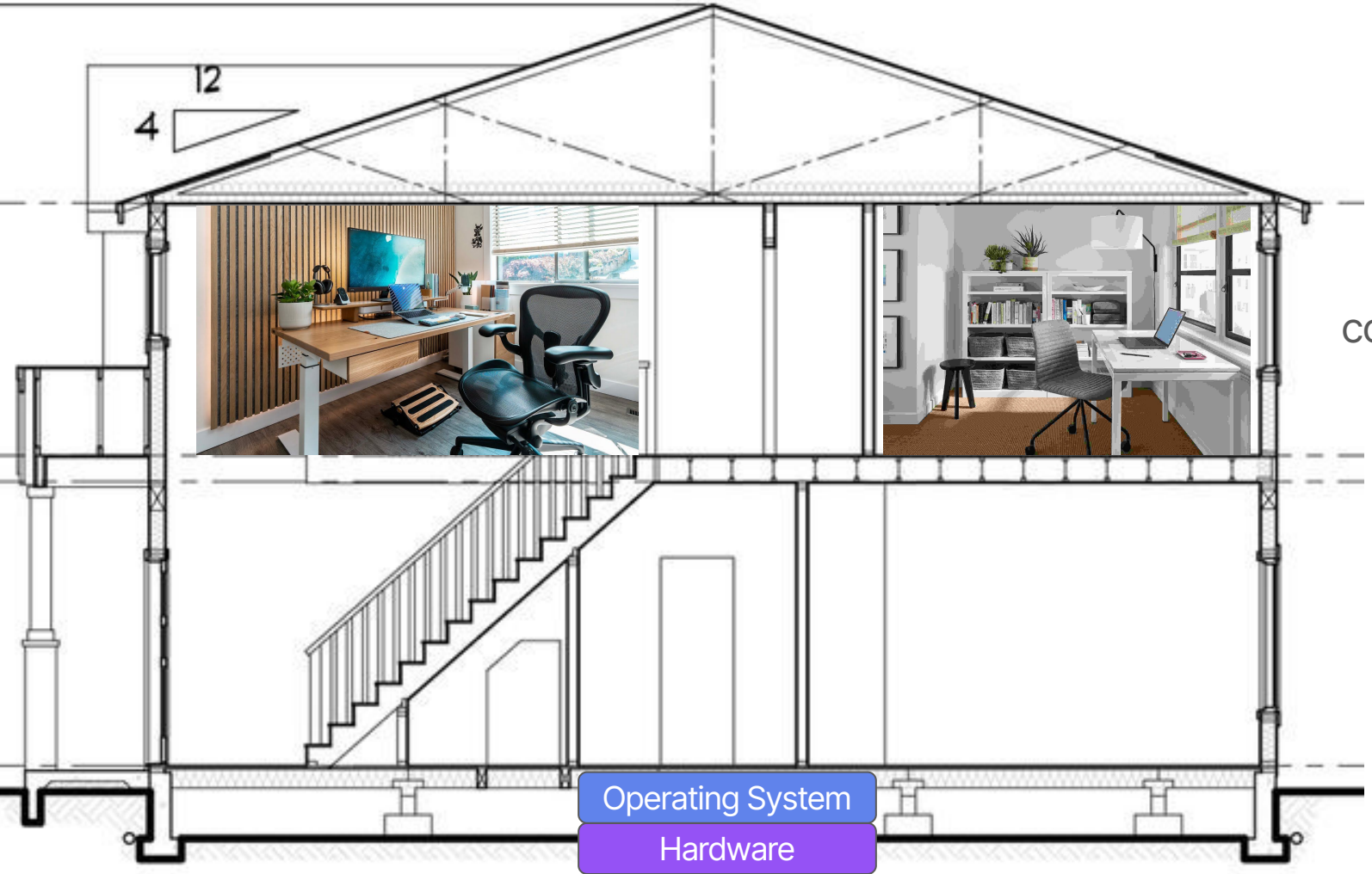


Containers

https://medium.com/techmormo/what-is-virtualization-bare-metal-vs-virtual-machines-vs-containers-f3d8be22ad34


12
4

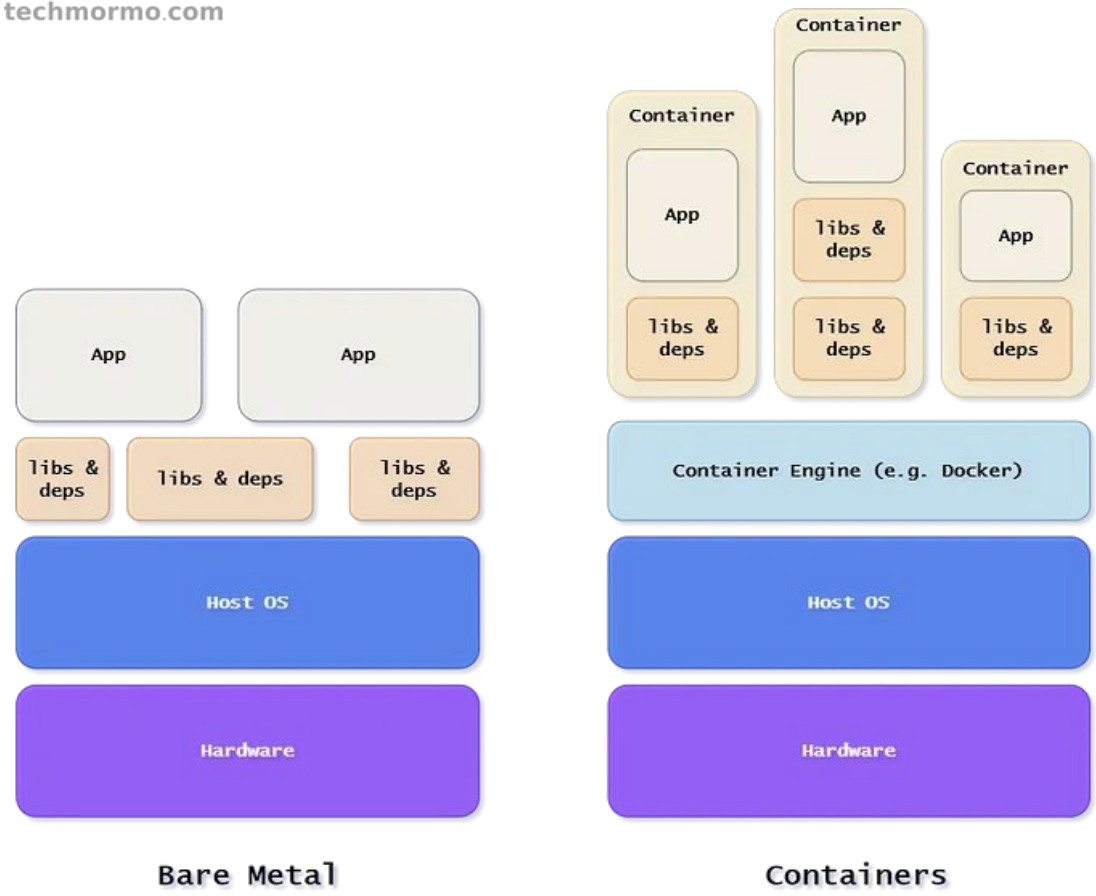

12
4
Operating System
Hardware



containers


12
4
containers
virtual
environments
Operating System
Hardware

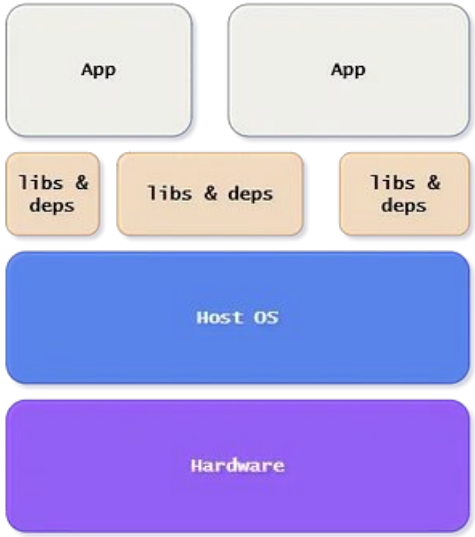
techmormo.com



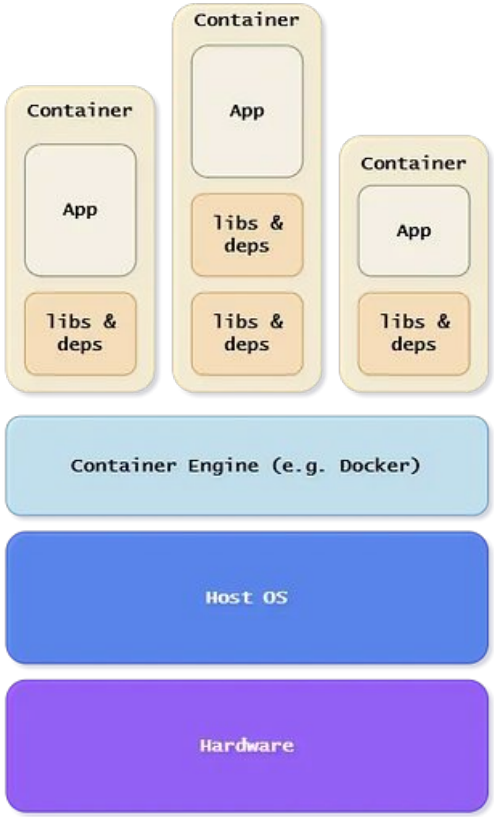
Bare Metal

Containers

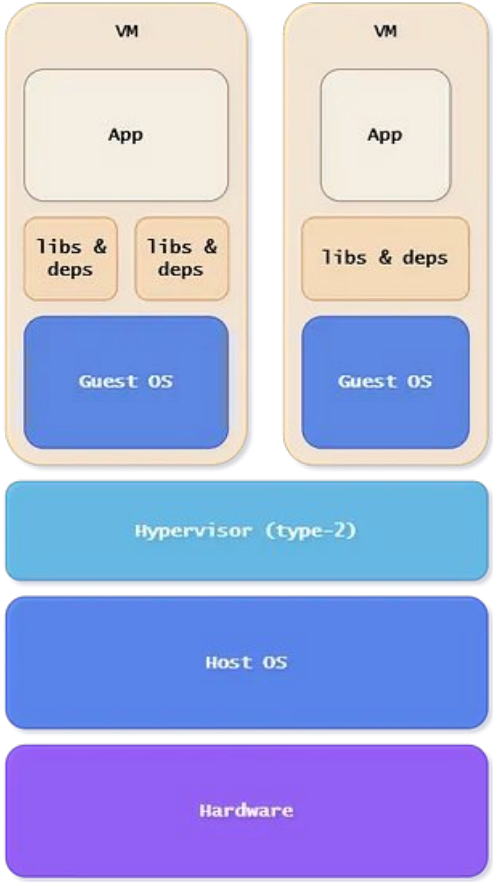
techmormo.com



Bare Metal

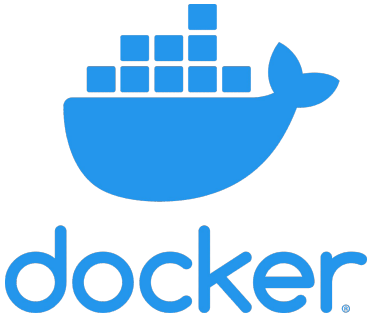


Containers



Virtual Machines

What is Docker?



Docker Engine

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

Docker Hub

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

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

https://hub.docker.com/

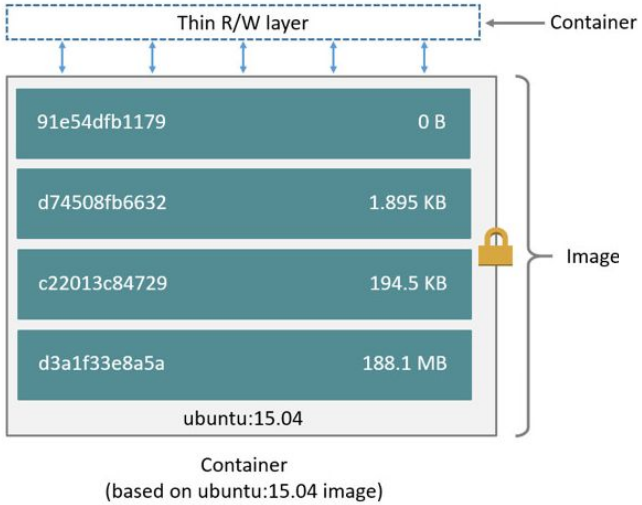
Docker **image** and **container**: what's the difference?

Docker image

- a **read-only** snapshot of an environment
- changing an image adds more layers
- can be stored on Docker Hub 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**

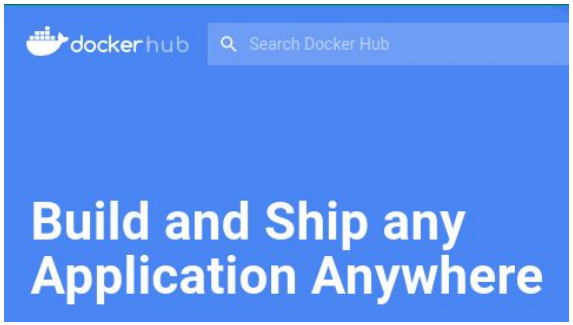


https://docs.docker.com/storage/storagedriver/

How can I get my own Docker container going?

Use an **existing image**

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

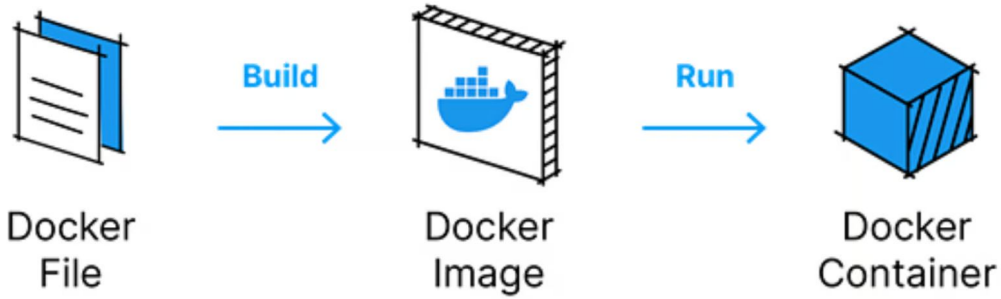


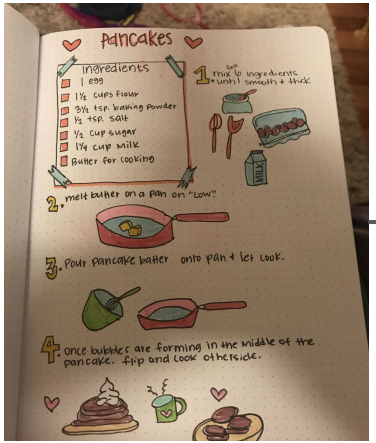
Build your **own image**

- 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 python:3.10-slim-buster
WORKDIR /app
COPY . /app/src
RUN pip install -r /app/src/requirements.txt
RUN pip install --no-cache-dir --no-deps /app/src[all]

ENTRYPOINT [ "bagel" ]
CMD ["--help"]
```


Build
Run
Docker File
Docker Image
Docker Container



recipe

combine

dough/mix

make from

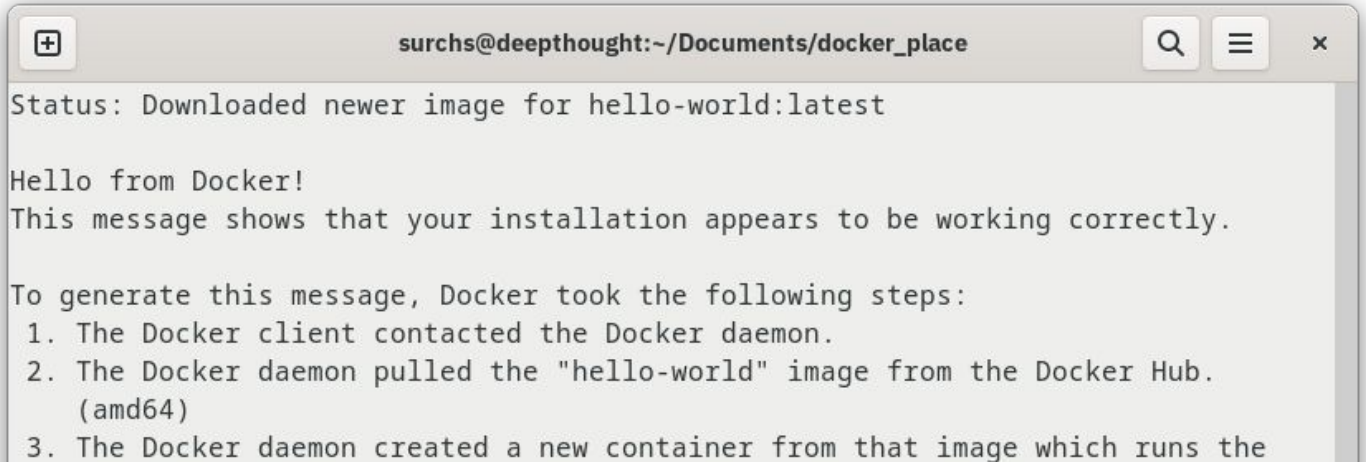
pancake (with toppings)

Exercise 1: Run a container from Docker Hub

- Find an image we like: https://hub.docker.com/_/hello-world
  - Already did this? Try https://hub.docker.com/r/grycap/cowsay/
- Pull the image (a specific tag)
- Run the image

Exercise 1: Run a container from Docker Hub

- Find an image we like: https://hub.docker.com/_/hello-world
  - Already did this? Try https://hub.docker.com/r/grycap/cowsay/
- Pull the image (a specific tag)
- Run the image



Exercise 1: Run a container from Docker Hub

💡 `docker --help` lists main docker commands!

Summary

- Docker Hub has pre-made images
- Image tags are important
- We can
  - retrieve images with `docker pull`
  - view images we have pulled with `docker images`
  - (pull 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

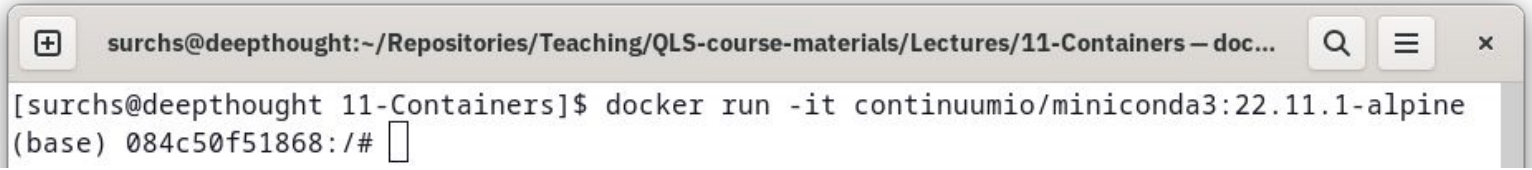
Exercise 2: Work with a container that has conda

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?

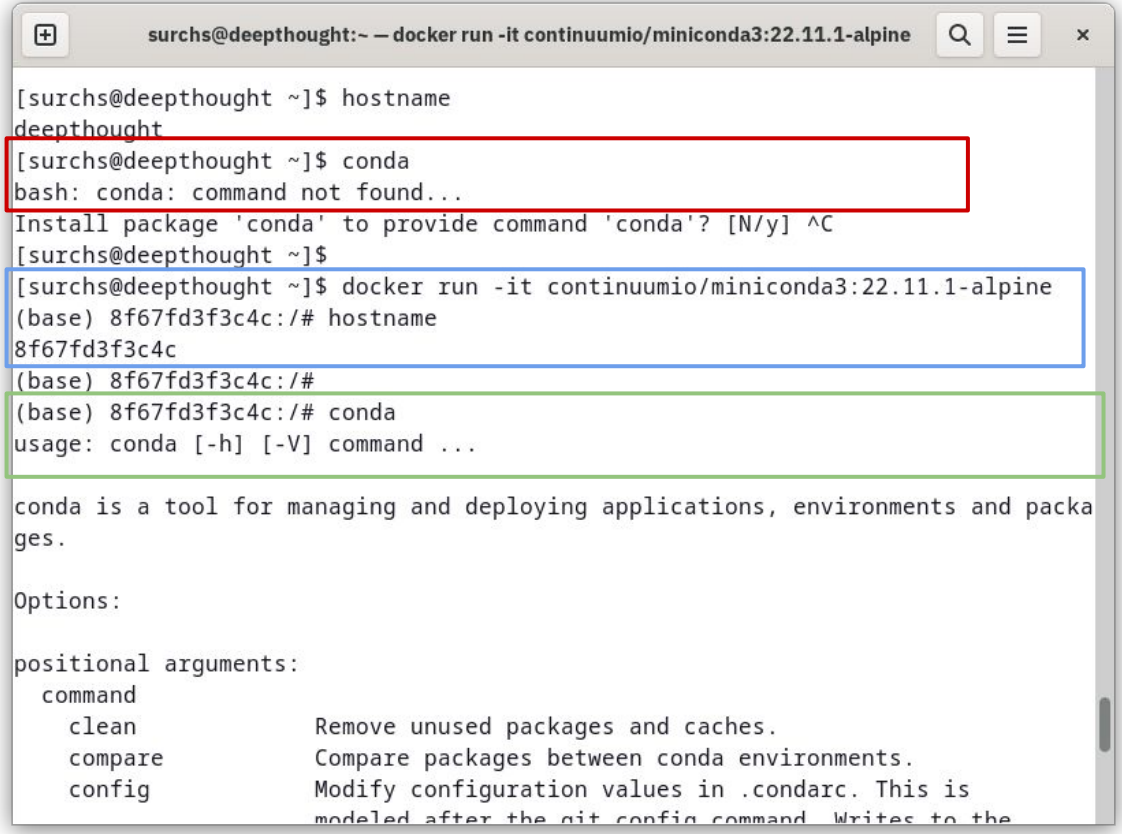
- Let's run it interactively to take a look inside



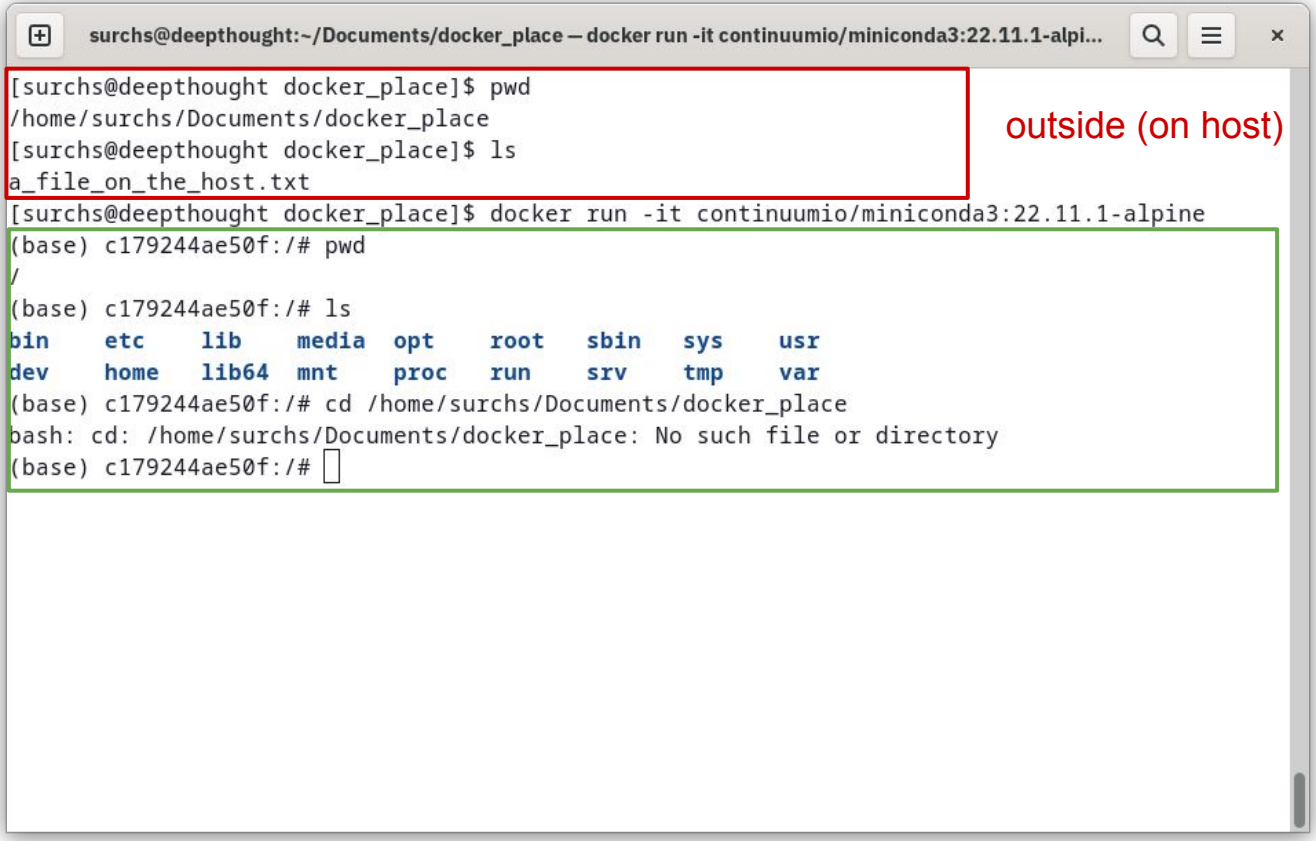
https://github.com/ContinuumIO/docker-images/blob/master/miniconda3/alpine/Dockerfile

Looking around inside a container

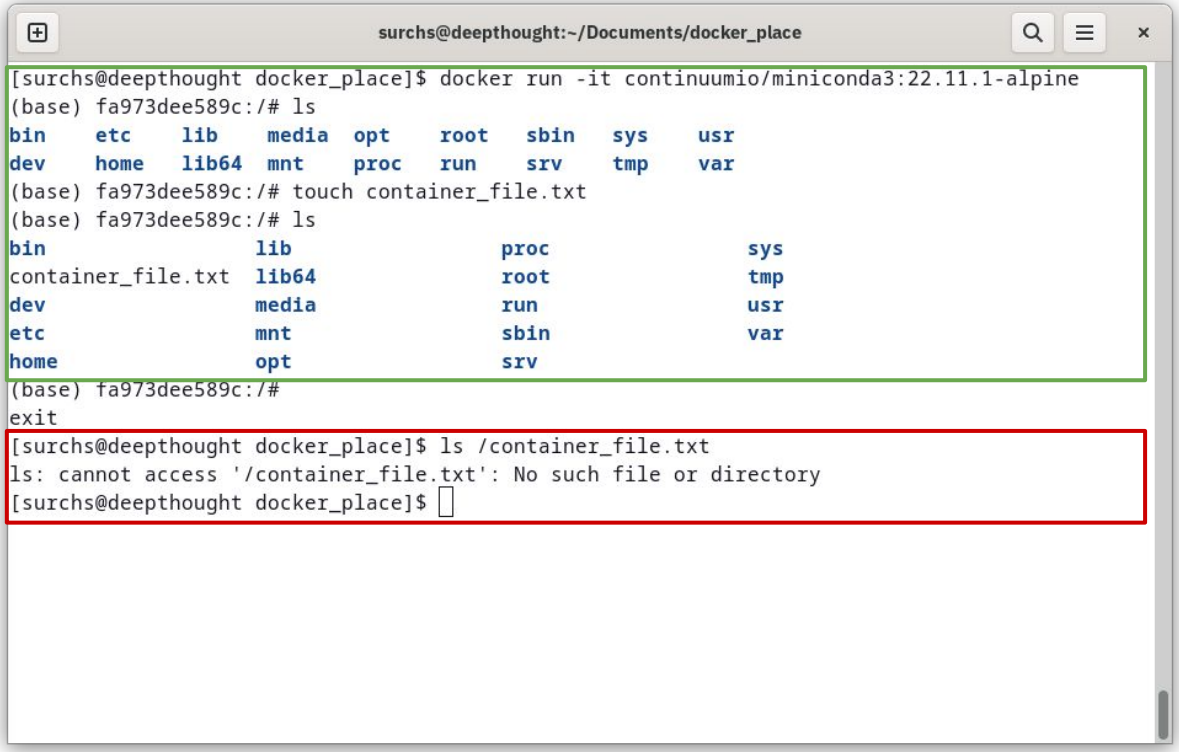
- no conda on my machine
- starting container changes the look of my terminal and the name of my computer
- inside of the container I have access to conda



By default the container doesn't see files on the host ...



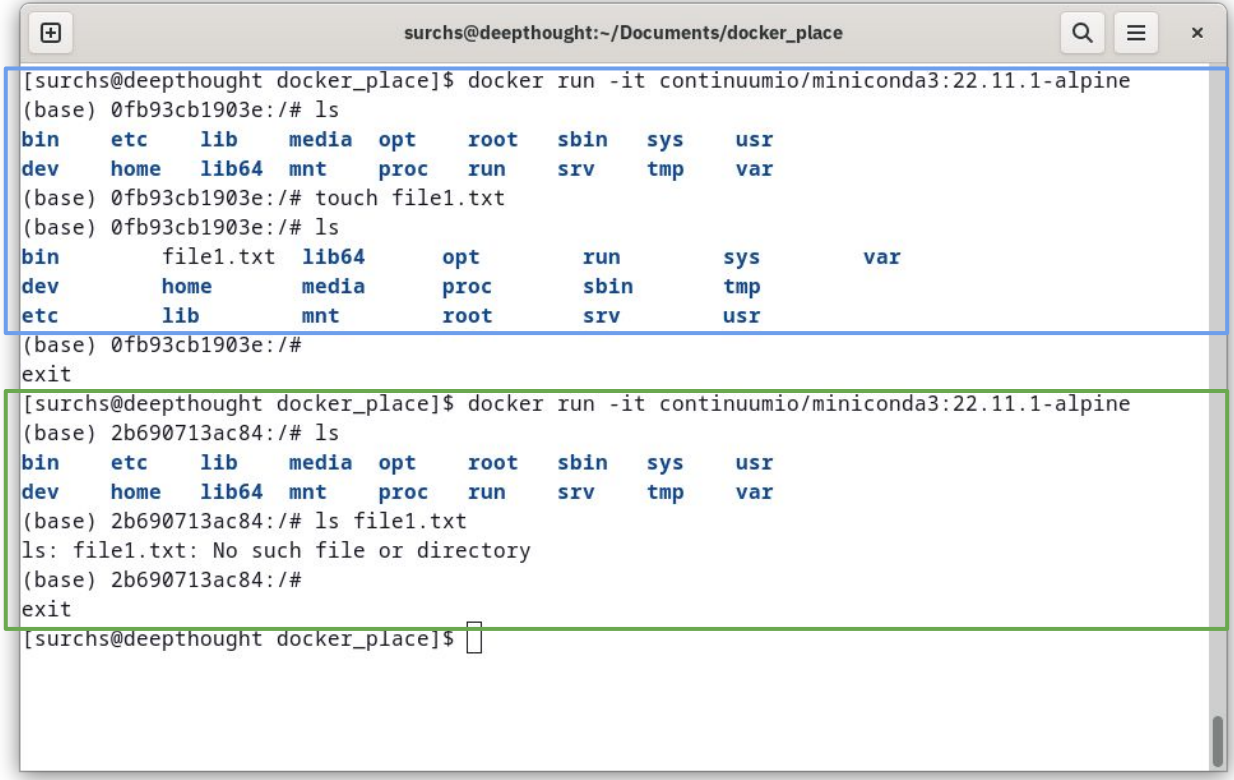
… and the host can’t see files on the container …



inside
(in container)

outside (on host)

... and files written to a specific container are tied to it!



first container instance

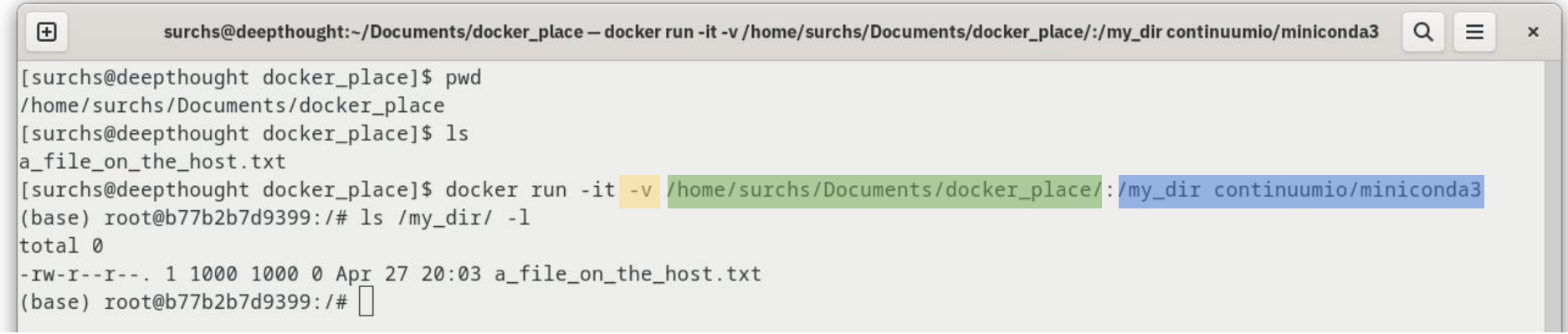
second container instance

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

💡 `docker ps -a` lists all containers

How do we share files between host and container?

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

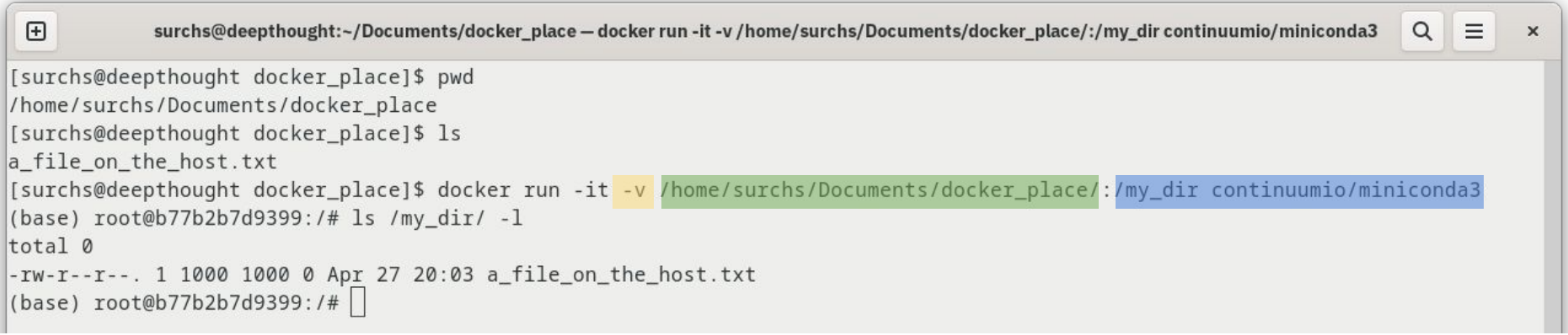


- The target path will be created in the container, regardless of if it already exists
- Host path must start with `/` or `./` ! Or use `--mount` instead
- Container path must be an absolute path (start with `/`)

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

How do we share files between host and container?

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



- The target path will be created in the container, regardless of if it already exists
- Host path must start with `/` or `./` ! Or use `--mount` instead
- Container path must be an absolute path (start with `/`)

🤔 which user owns files created by the container?

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 to 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`

  - This allows changes to persist beyond the container
- It’s a bad idea to write directly into the container’s filesystem

Exercise 3: Run a Dockerized tool on local data

We've explored how a Docker container behaves on the inside. Let's follow the more "typical" steps to run a tool available as a Docker image.

inside

outside

Exercise 3: Run a Dockerized tool on local data

Let's try running the BIDS validator on a test dataset `ds006`, to see if the dataset complies with the Brain Imaging Data Structure.

1. **Pull** the bids/validator Docker image tagged `latest`, and **list** the images on your system to confirm that bids/validator is there

```
docker pull bids/validator:latest
docker images
```

Exercise 3: Run a Dockerized tool on local data

Recall: `docker run [docker-options] <imagename> <command>`

Print the help text for the bids/validator:

```
docker run bids/validator --help
```

Output:

```
Usage: bids-validator <dataset_directory> [options]
...
```

Exercise 3: Run a Dockerized tool on local data

Recall: `docker run [docker-options] <imagename> <command>`

Print the help text for the bids/validator:

```
docker run bids/validator --help
```

Output:

```
Usage: bids-validator <dataset_directory> [options]
...
```

...How do I run other validator commands?

```
docker run [options] bids/validator <dataset_directory> [options]
```

Exercise 3: Run a Dockerized tool on local data

```
docker run [options] bids/validator <dataset_directory> [options]
```

Our test dataset is located on our **host machine** under:
`~/QLS-course-materials/Lectures/2025/data/ds006`

2. Run a bids/validator container again, but provide this dataset directory to validate.

Exercise 3: Run a Dockerized tool on local data

```
docker run [options] bids/validator <dataset_directory> [options]
```

Our test dataset is located on our **host machine** under:
`~/QLS-course-materials/Lectures/2025/data/ds006`

2. Run a bids/validator container again, but provide this dataset directory to validate.

HINTS:

- Mount the dataset directory into the container for it to find the dataset (remember: the path must start with / or ./ !)
  `-v /path/on/host:/path/in/container`
- Where is the bids/validator app actually running? What files can it see?

Exercise 3: Run a Dockerized tool on local data

```
docker run [options] bids/validator <dataset_directory> [options]
```

Our test dataset is located on our **host machine** under:
`~/QLS-course-materials/Lectures/2025/data/ds006`

2. Run a bids/validator container again, but provide this dataset directory to validate.

HINTS:

- Mount the dataset directory into the container for it to find the dataset (remember: the path must start with / or ./ !)
  `-v /path/on/host:/path/in/container`
- Where is the bids/validator app actually running? What files can it see?

```
cd ~/QLS-course-materials/Lectures/2025/data
docker run -v./ds006:/data bids/validator /data
```

Exercise 3: Run a Dockerized tool on local data

The BIDS validator returns errors `[ERR]` because the test dataset contains empty data files.

3. To make the validator happy, run it again on the dataset, but this time add **command options** to:
   - Ignore warnings
   - Ignore NIfTI header content

Exercise 3: Run a Dockerized tool on local data

The BIDS validator returns errors `[ERR]` because the test dataset contains empty data files.

3. To make the validator happy, run it again on the dataset, but this time add **command options** to:
   - Ignore warnings
   - Ignore NIfTI header content

HINTS:
   - Print the help text again to see all command options for bids/validator
     ```
     docker run --rm bids/validator --help
     ```

Exercise 3: Run a Dockerized tool on local data

The BIDS validator returns errors `[ERR]` because the test dataset contains empty data files.

3. To make the validator happy, run it again on the dataset, but this time add **command options** to:
   - Ignore warnings
   - Ignore NIfTI header content

HINTS:
   - Print the help text again to see all command options for bids/validator
     ```
     docker run --rm bids/validator --help
     ```

```
docker run -v./ds006:/data bids/validator /data --ignoreNiftiHeaders
--ignoreWarnings
```

Apptainer (Singularity)

Apptainer is the container solution for HPCs (e.g. Compute Canada)

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

Instead, **Apptainer** (formerly Singularity) can be used to run containers:

> 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

💡 Singularity and Apptainer commands are nearly identical

Build images with Docker, run them with Apptainer

1. Pull an image from a Docker registry and create a local **S**ingularity**I**mage**F**ile (.sif)

```
$ apptainer pull docker://ghcr.io/apptainer/lolcow
```

can also be Docker Hub

2. Run the Singularity File using `apptainer run`

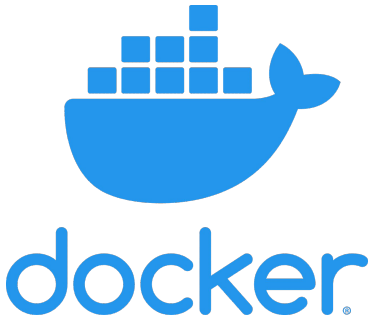
```
$ apptainer run lolcow_latest.sif
 _______________________________
< Mon Aug 16 13:01:55 CDT 2021 >
 -------------------------------
        \   ^__^
         \  (oo)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
```

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

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

Containerization summary

- Docker **images** are **read-only snapshots**, you can find them on Docker Hub
- Images have tags (or versions), that you should specify when pulling
- A Docker **container** is an isolated, **live instance** of an image
- Docker **containers** have their own filesystem, but this is **not persistent**
- With **bind mounts** 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 **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 website is in progress. Is there is something you would like to see here, please submit a new GitHub issue.

```
neurodocker generate docker \
    --pkg-manager apt \
    --base-image neurodebian:bullseye \
    --ants version=2.4.3 \
    --miniconda version=latest conda_install="nipype notebook" \
    --user nonroot
```

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

Additional resources

- Docker docs https://docs.docker.com/get-started/
- Hands-on docker tutorials https://training.play-with-docker.com/, https://www.docker.com/101-tutorial/
- 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