THE UNIVERSITY OF QUEENSLAND
AUSTRALIA
CREATE CHANGE

Software Containers

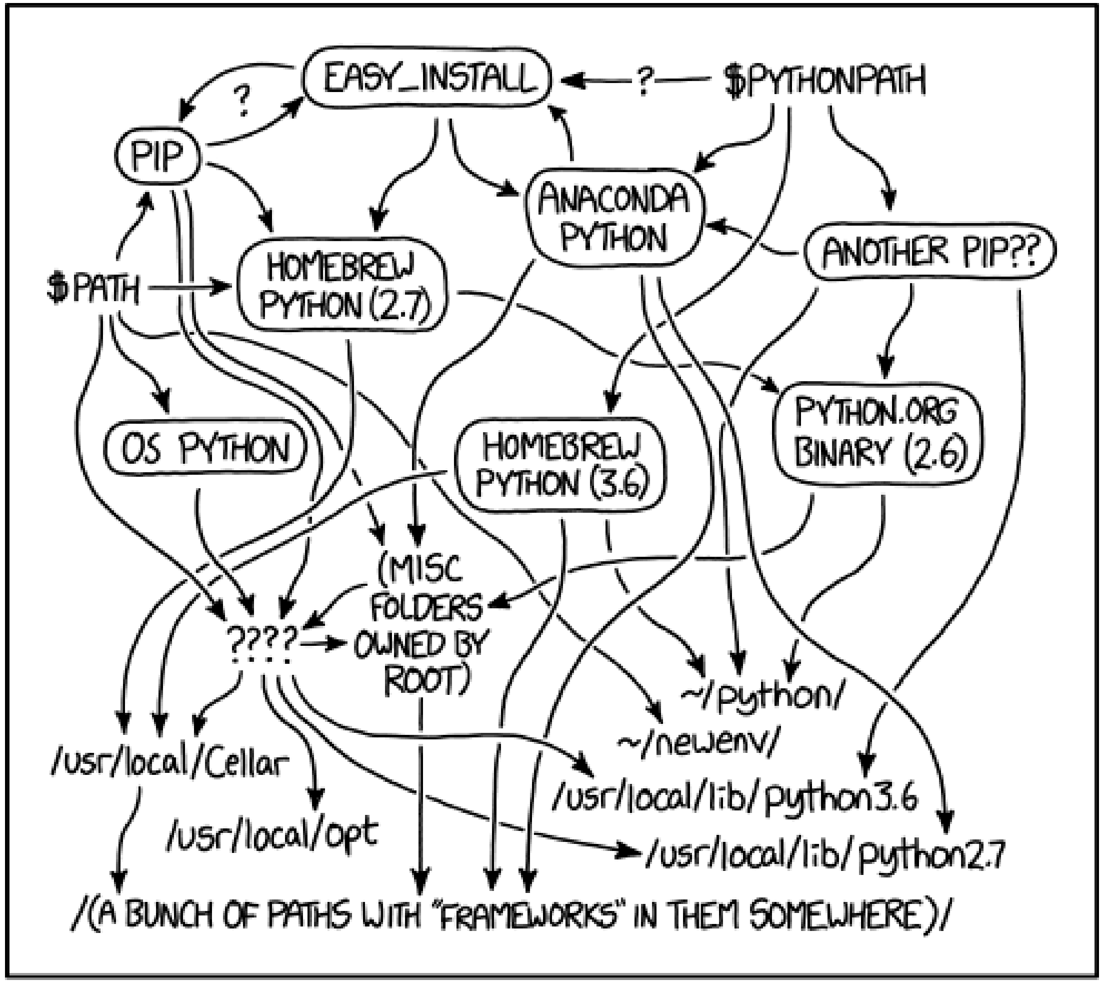
What are they? Why are they useful? How to use them? How to build them?

Steffen Bollmann

School of Information Technology and Electrical Engineering

The University of Queensland

Installing and maintaining software is trivial … right?



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED
THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

Software Containers in a nutshell

Developers needed to move their applications between different computers -> this became especially important with more and more Cloud applications

+ Standardization

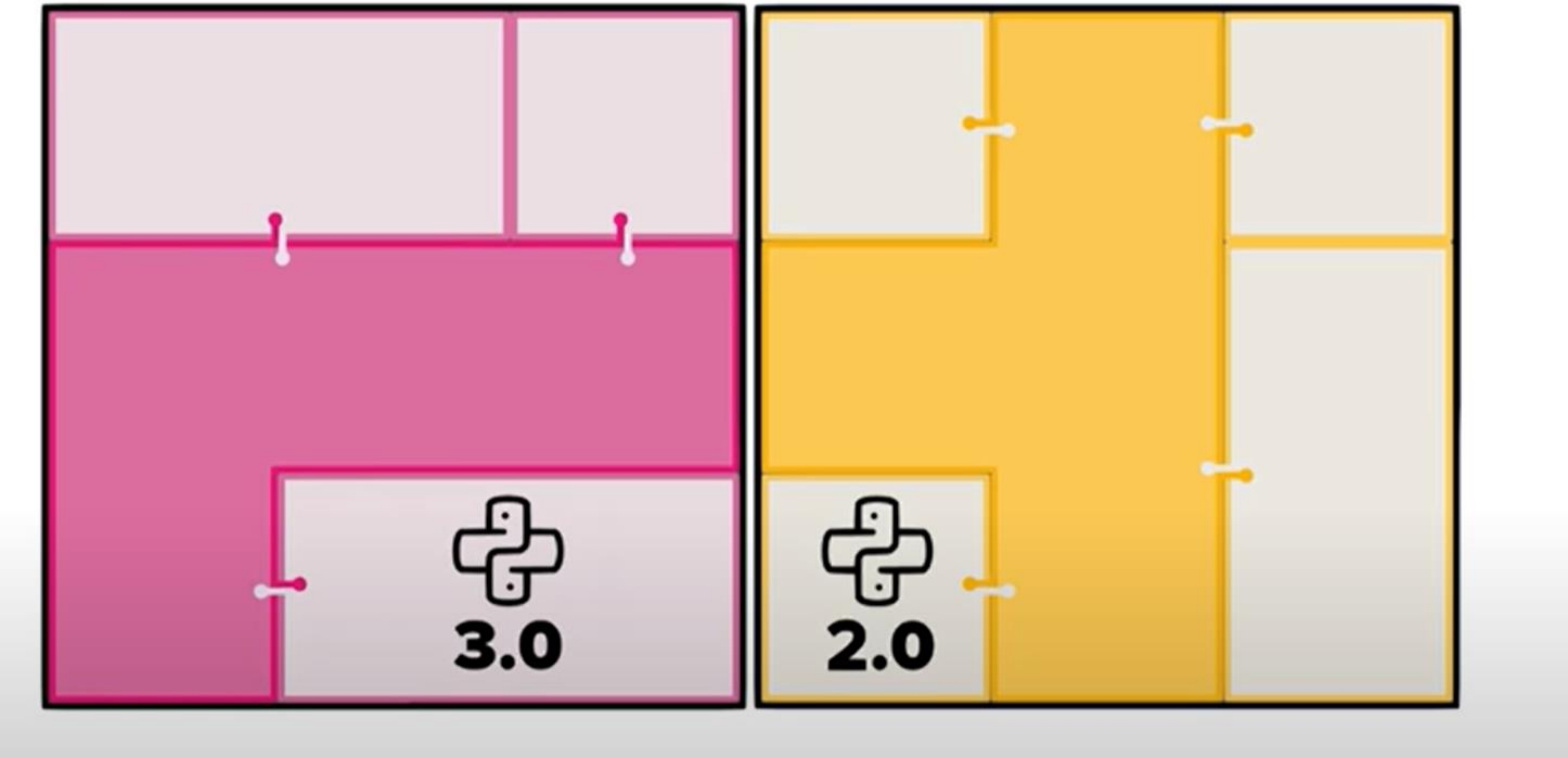
+ Portability

+ Reliability

ARDC Containers in 3 minutes https://www.youtube.com/watch?v=HelrQnm3v4g

Software Containers in a nutshell

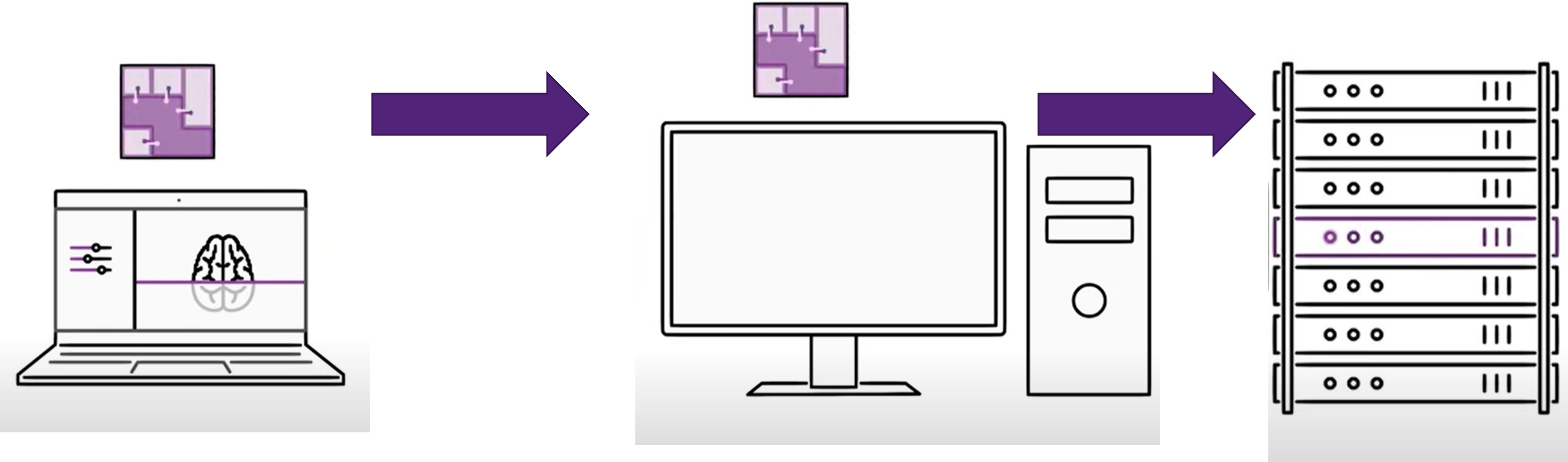
Software containers package the application + dependencies in a standardized format



ARDC Containers in 3 minutes https://www.youtube.com/watch?v=HelrQnm3v4g

Software Containers in a nutshell

Software containers enable the moving of software between different platforms



ARDC Containers in 3 minutes https://www.youtube.com/watch?v=HelrQnm3v4g

Containers are not new …

2005: OpenVZ

2008: Linux-Vserver

2008: LXC

2013: Docker (wrapper for LXC in the beginning, now libcontainer)

-> Docker was the start of the container hype, because it made this technology easy to use for developers

2013: lmctfy (googles container format, now in libcontainer)

2014: rkt

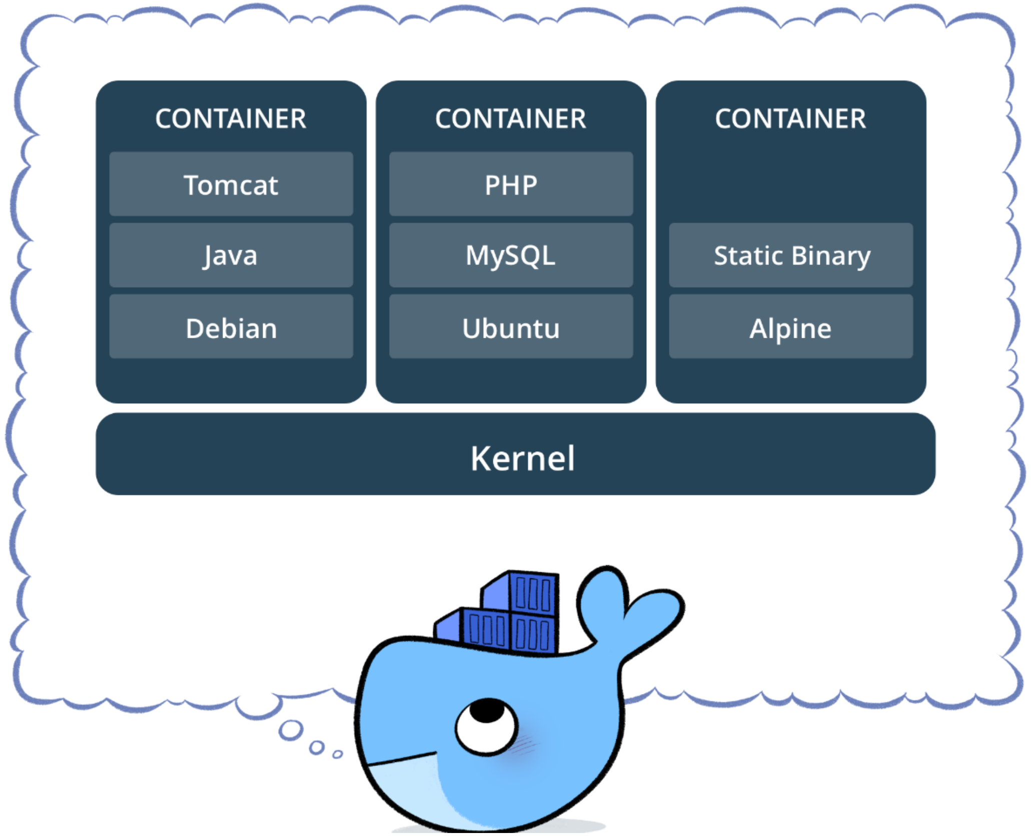
2014: Release of Kubernetes -> Open Source container orchestrator -> almost the whole Internet is based on this today …

2015: runc (open container initiative)

….

What are containers?

- isolate software from its surroundings
- container image includes: code, runtime, system tools, system libraries, settings
- resource management provided by the Linux kernel (namespaces and cgroups)
- recipe = describes what should be in an image
- image = stores everything we need to run
- container = what we launch based on an image



https://www.docker.com/what-container#/package_software

What are Namespaces and Cgroups?

namespaces control what a **process can see**:

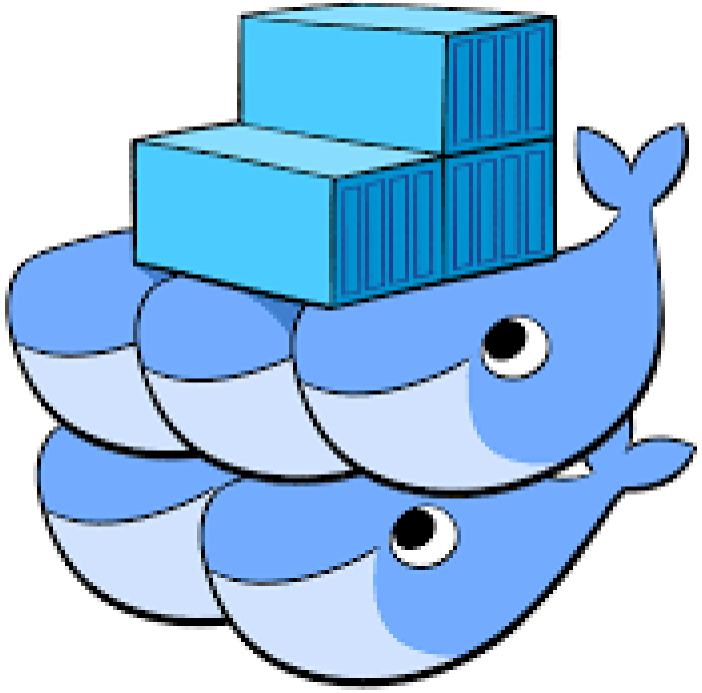
- Linux kernel feature that isolates and virtualizes system resources
- mount, process IDs, hostnames, user IDs, filesystems, Network
- full support for containers in Linux kernel version 3.8 (user namespaces)

control groups (Cgroups) control what a **process can use**:

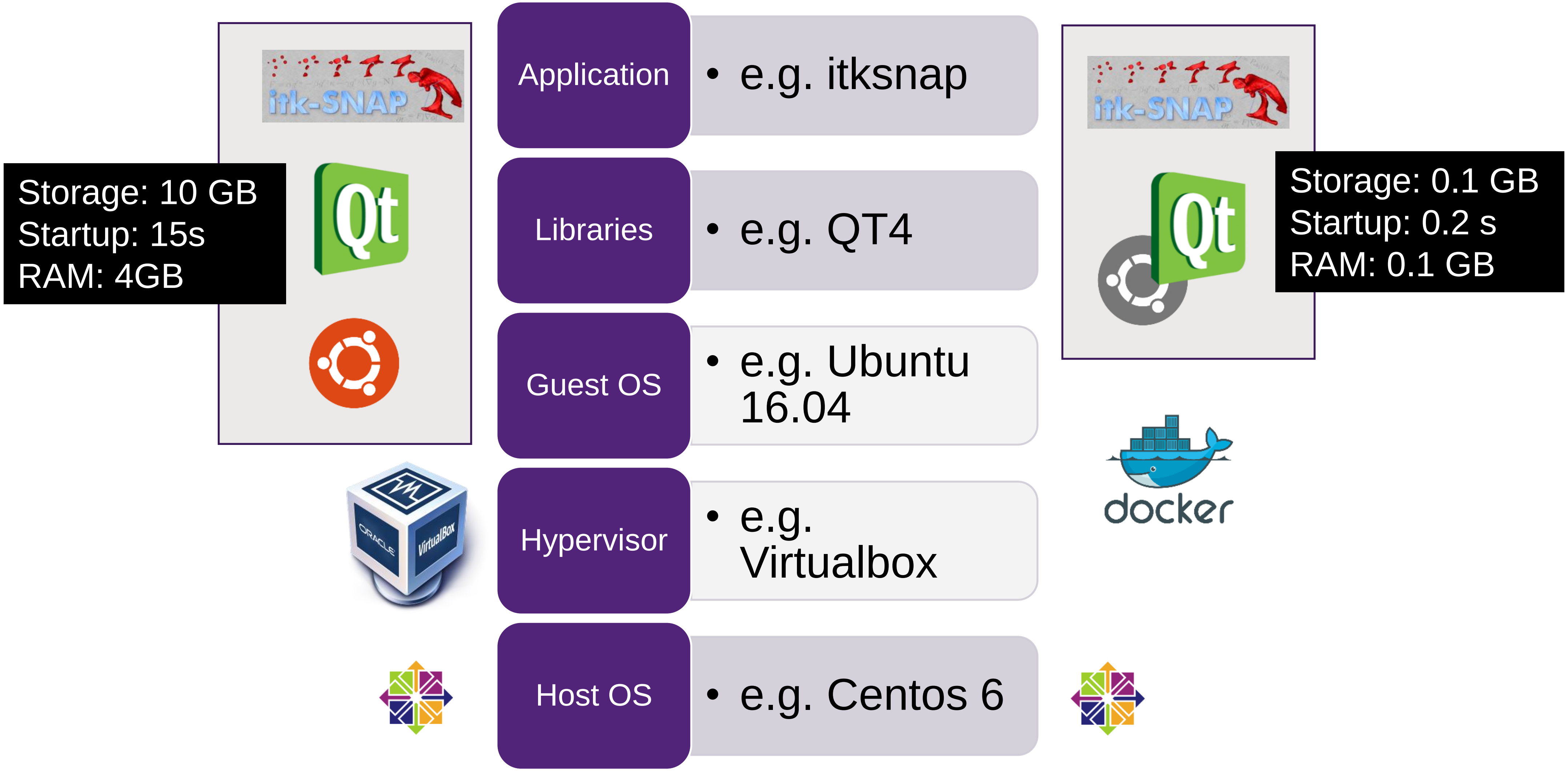
- Linux kernel feature that limits, accounts for, and isolates the resource usage of processes
- Memory, CPU, I/O, network, etc.

Docker

- started the container hype by providing easy to use packages for Linux, Windows, Mac
- widely adopted and supported by cloud providers, including orchestration of many containers (Docker Swarm)
- Docker technology is open source, but Docker Inc provides commercial applications to make it easy to use (e.g. Docker desktop)



VIRTUAL MACHINES VS CONTAINERS



Storage: 10 GB
Startup: 15s
RAM: 4GB

Application
- e.g. itksnap

Libraries
- e.g. QT4

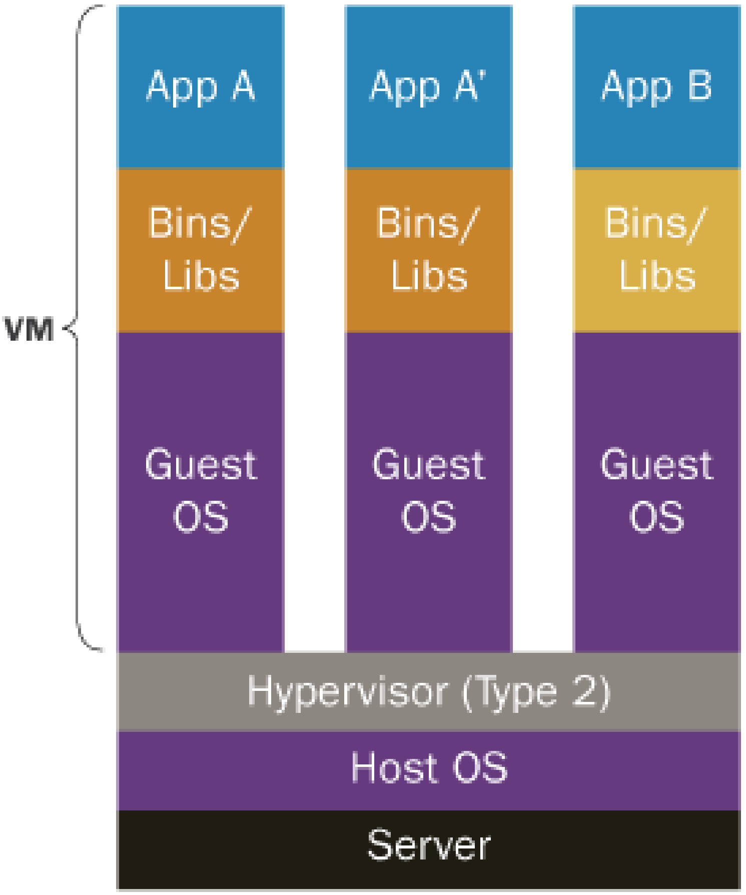
Guest OS
- e.g. Ubuntu 16.04

Hypervisor
- e.g. Virtualbox

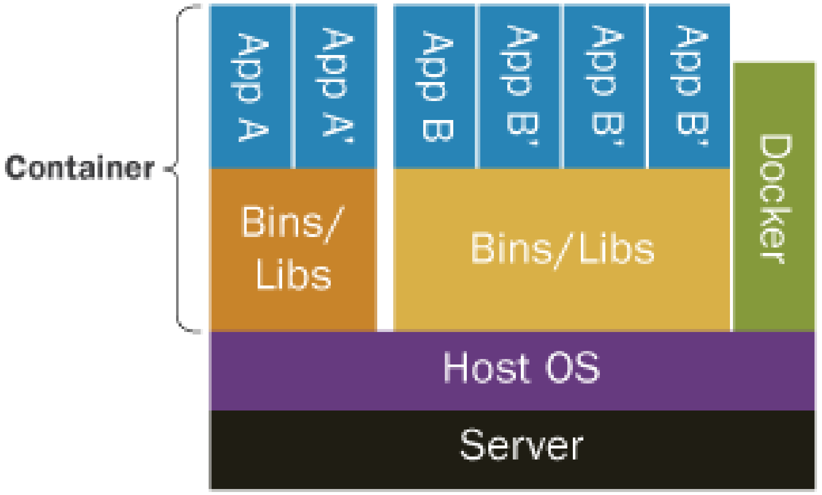
Host OS
- e.g. Centos 6

Storage: 0.1 GB
Startup: 0.2 s
RAM: 0.1 GB

VIRTUAL MACHINES VS CONTAINERS

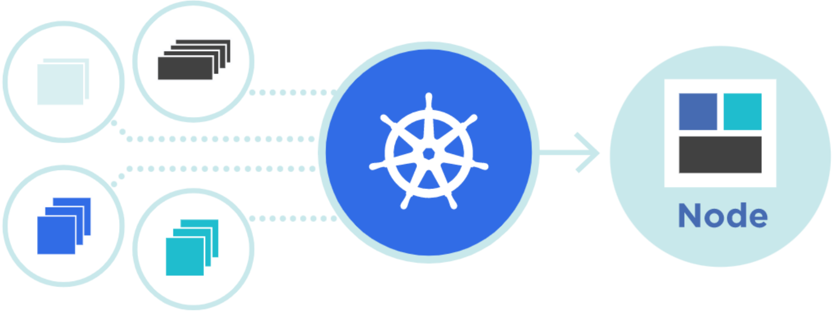


Containers are isolated, but share OS and, where appropriate, bins/libraries



What is Container Orchestration?

- Webservices usually require more than one Container
- Manually configuring network + persistent storage is error prone
- updating containers needs to be efficient and seamless for users

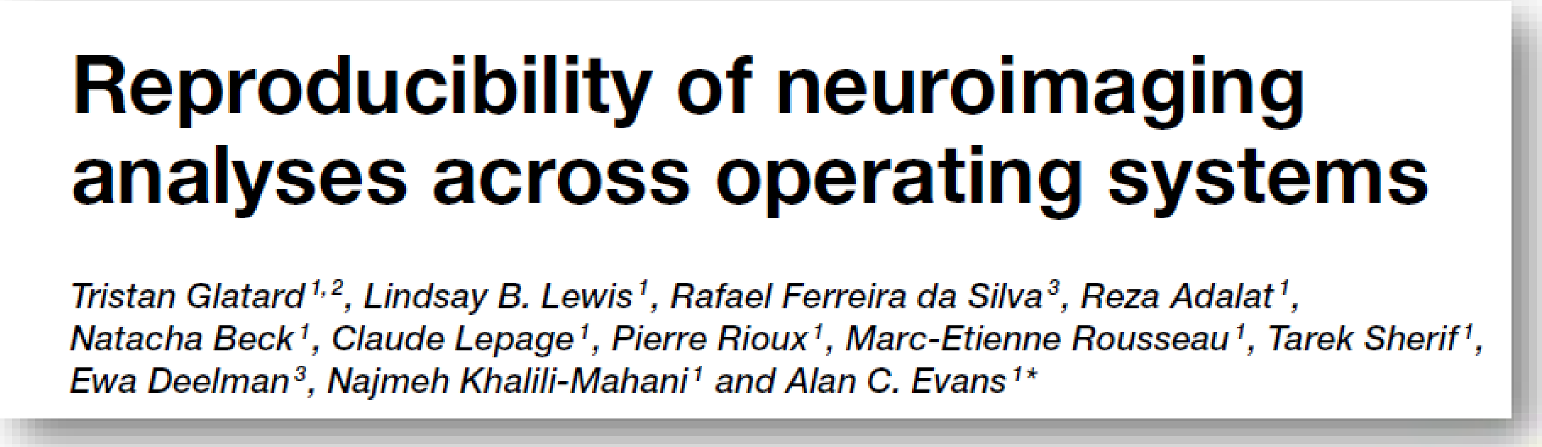


- The solution: **Kubernetes**
  - also known as K8s
  - open-source (developed by google based on their internal Borg project)
  - system for
    - automating deployment
    - scaling
    - management
  - of containerized applications

Questions?

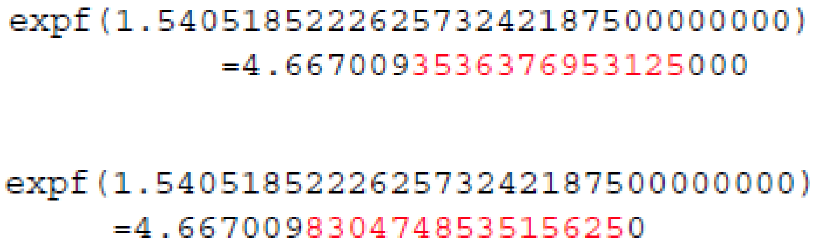
Why are containers useful?

Why are containers useful for science?



Reproducibility of neuroimaging analyses across operating systems

*Tristan Glatard[1,2], Lindsay B. Lewis[1], Rafael Ferreira da Silva[3], Reza Adalat[1], Natacha Beck[1], Claude Lepage[1], Pierre Rioux[1], Marc-Etienne Rousseau[1], Tarek Sherif[1], Ewa Deelman[3], Najmeh Khalili-Mahani[1] and Alan C. Evans[1]**



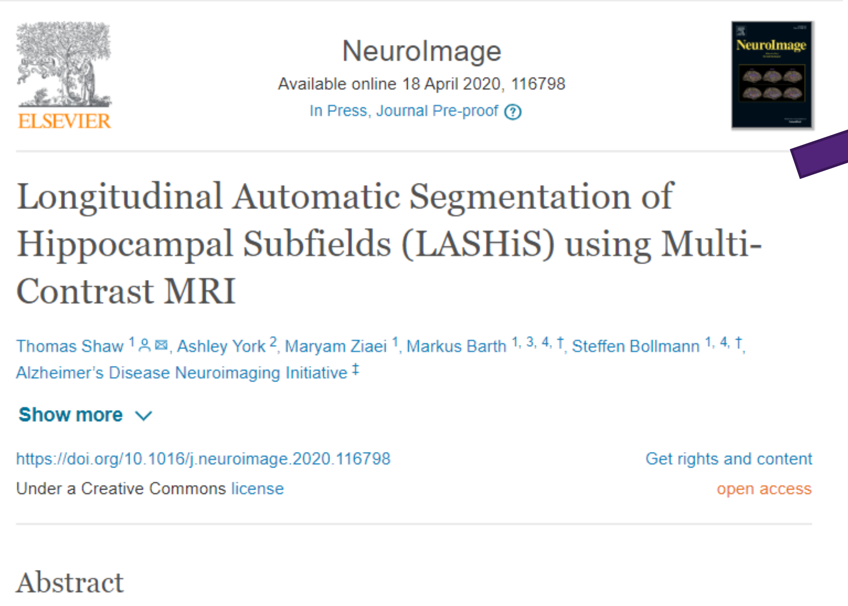
```
expf(1.54051852226257324218750000000)
          =4.66700935363769531250000

expf(1.54051852226257324218750000000)
     =4.667009830474853515625 0
```

- glibc 2.5 vs 2.18 deliver different floating-point results
- leads to significant differences in long pipelines

Why are containers useful for science?

- **Example: Sharing a reproducible pipeline including the software and the data!**



NeuroImage
Available online 18 April 2020, 116798
In Press, Journal Pre-proof ?

ELSEVIER

Longitudinal Automatic Segmentation of Hippocampal Subfields (LASHiS) using Multi-Contrast MRI

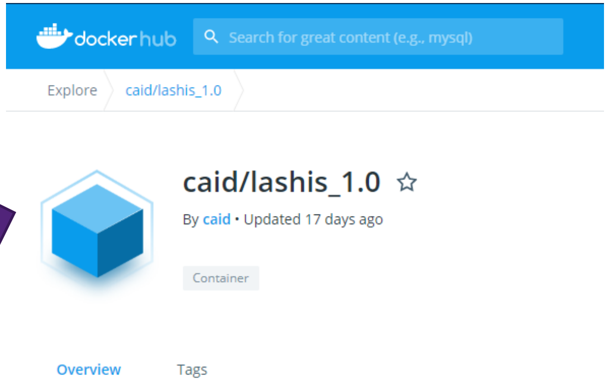
Thomas Shaw [1], Ashley York [2], Maryam Ziaei [1], Markus Barth [1, 3, 4, †], Steffen Bollmann [1, 4, †], Alzheimer's Disease Neuroimaging Initiative [‡]

Show more

https://doi.org/10.1016/j.neuroimage.2020.116798 Get rights and content

Under a Creative Commons license open access

Abstract



docker hub Search for great content (e.g., mysql)

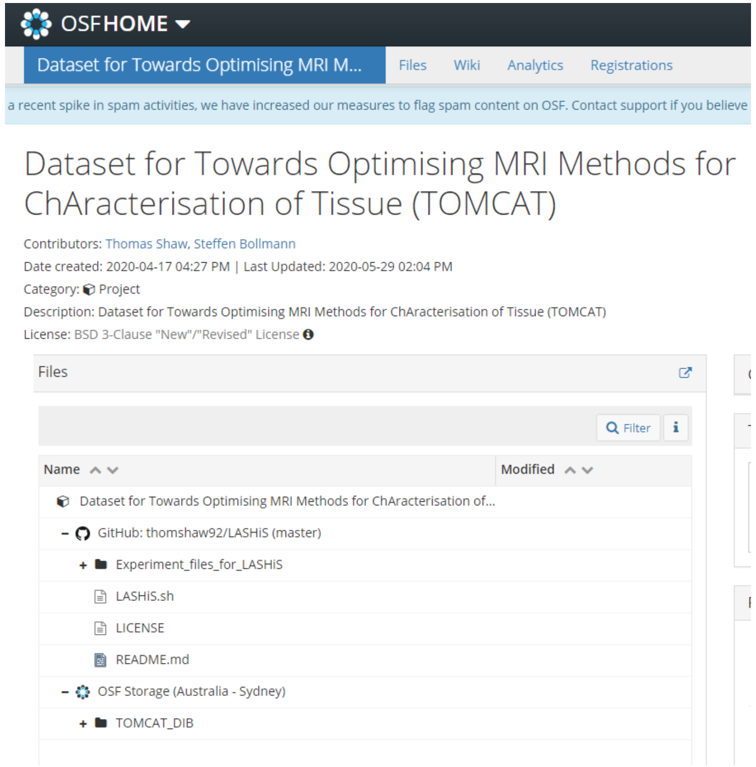
Explore caid/lashis_1.0

caid/lashis_1.0 ☆

By caid • Updated 17 days ago

Container

Overview Tags



OSFHOME ▾

Dataset for Towards Optimising MRI M... Files Wiki Analytics Registrations

a recent spike in spam activities, we have increased our measures to flag spam content on OSF. Contact support if you believe

Dataset for Towards Optimising MRI Methods for ChAracterisation of Tissue (TOMCAT)

Contributors: Thomas Shaw, Steffen Bollmann
Date created: 2020-04-17 04:27 PM | Last Updated: 2020-05-29 02:04 PM
Category: Project
Description: Dataset for Towards Optimising MRI Methods for ChAracterisation of Tissue (TOMCAT)
License: BSD 3-Clause "New"/"Revised" License

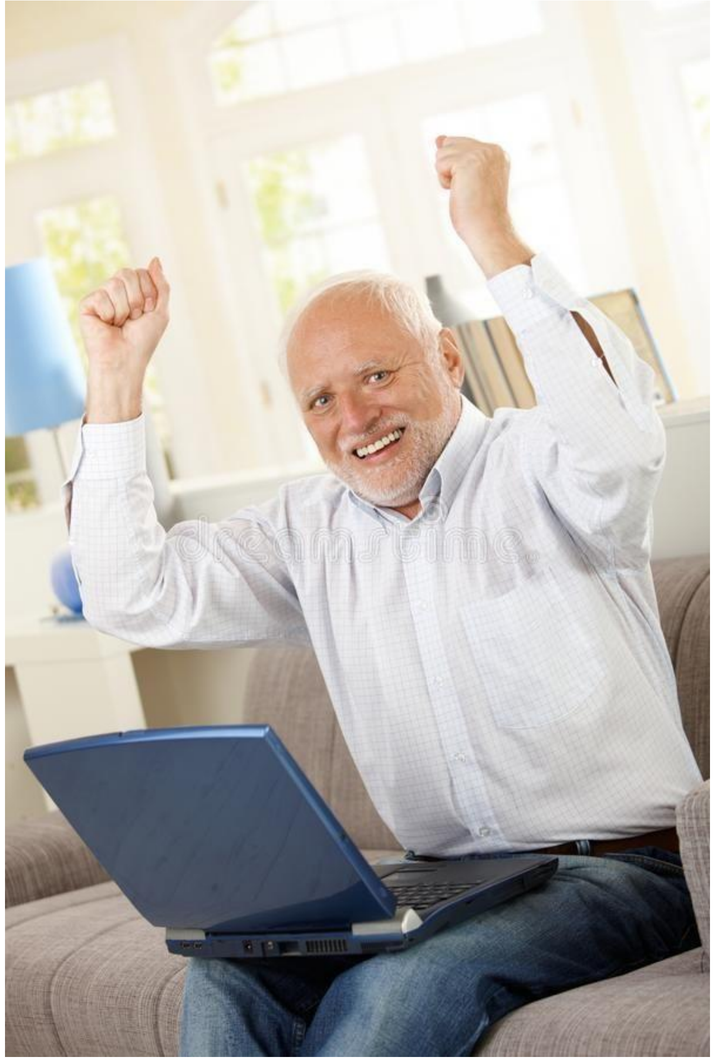
Files

Filter

Name	Modified
Dataset for Towards Optimising MRI Methods for ChAracterisation of...	
− GitHub: thomshaw92/LASHiS (master)	
+ Experiment_files_for_LASHiS	
LASHiS.sh	
LICENSE	
README.md	
− OSF Storage (Australia - Sydney)	
+ TOMCAT_DIB	

Benefits of Software containers

- Longitudinal stability of software pipeline (e.g., upgrade of Ubuntu 16.04 breaks softwarere relying on libpng12 …)
- Portable
- Isolated – things that happen in the container should stay in the container by default (ephemeral!)
- Ease of use
- Software Development environments in Containers make it easy to onboard new project members



Challenges when using containers

- Versioning of containers comes with no guarantees – images may not exist tomorrow, hubs disappear
- Black box – how was it made? (e.g. Container that's based on another Container, that's based on ….
- Security is not automatically better -> important to use latest software versions and regular patches + official distribution images as baselines

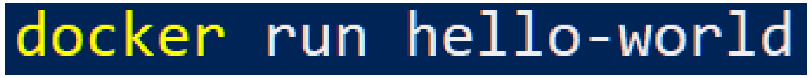
Questions?

How to build containers

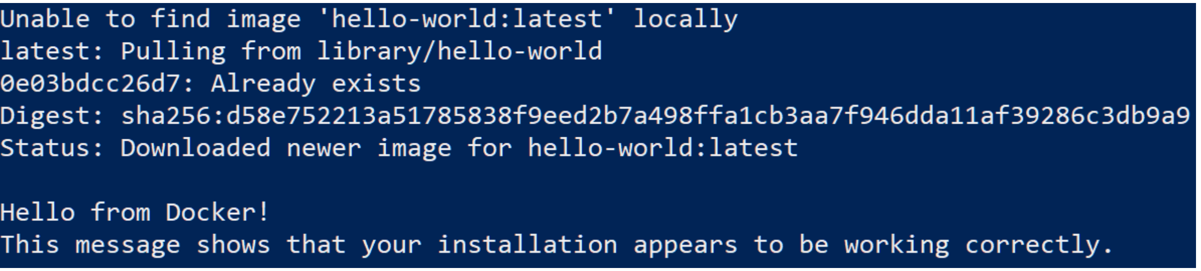
Hello world of Docker – Hands on

If docker installed on your computer:

Windows PowerShell



```
docker run hello-world
```

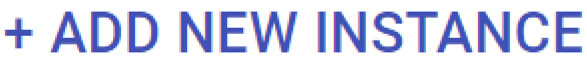


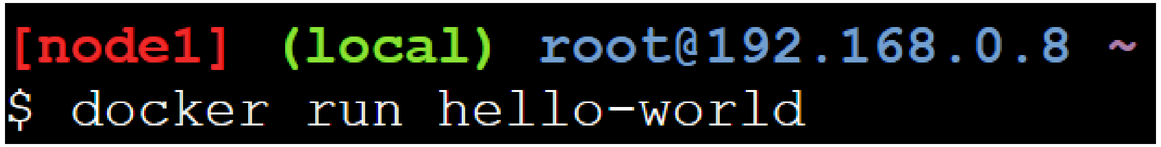
```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Already exists
Digest: sha256:d58e752213a51785838f9eed2b7a498ffa1cb3aa7f946dda11af39286c3db9a9
Status: Downloaded newer image for hello-world:latest

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

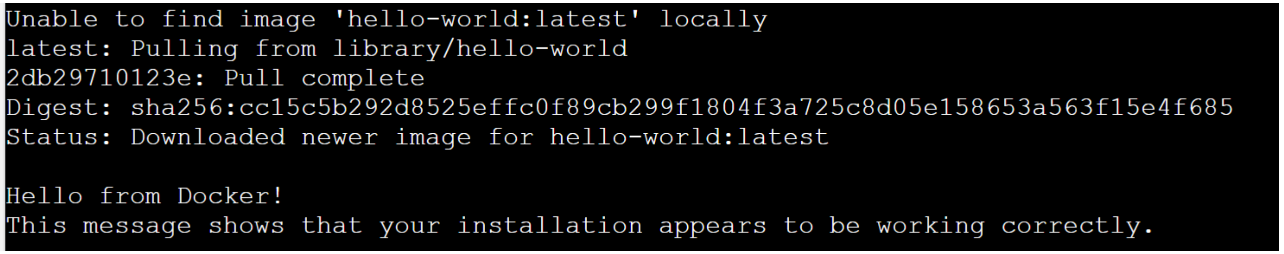
If docker not installed:

https://labs.play-with-docker.com/





```
[node1] (local) root@192.168.0.8 ~
$ docker run hello-world
```

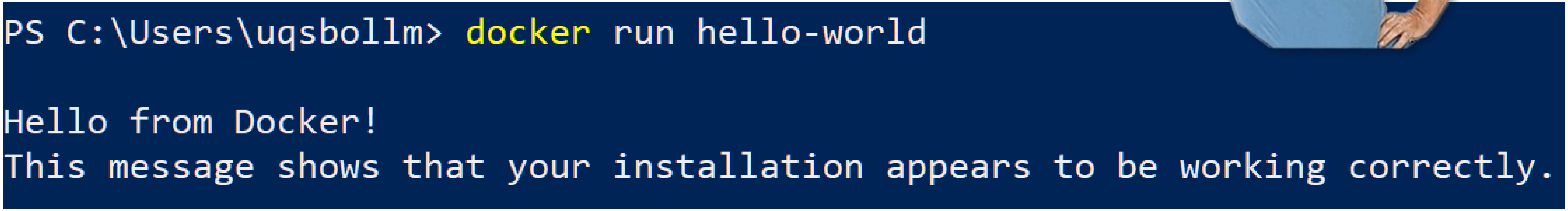


```
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
2db29710123e: Pull complete
Digest: sha256:cc15c5b292d8525effc0f89cb299f1804f3a725c8d05e158653a563f15e4f685
Status: Downloaded newer image for hello-world:latest

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

Hello world of Docker – Hands on

Docker caches the images locally ☺ So, running it again, will be faster!

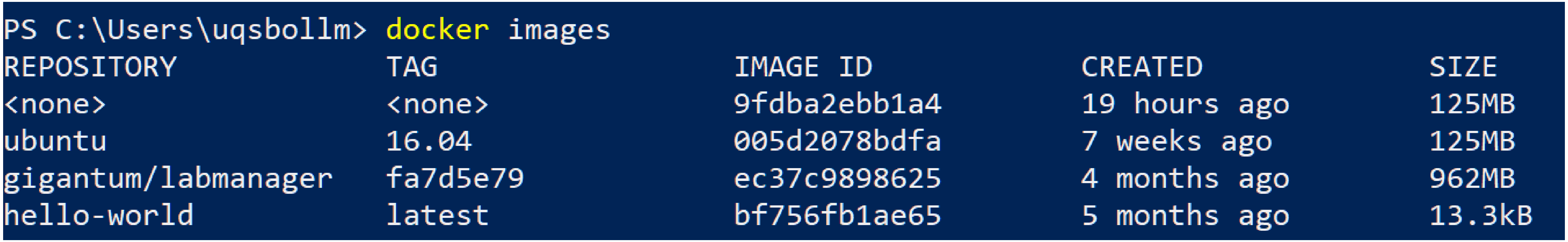


```
PS C:\Users\uqsbollm> docker run hello-world

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

Show downloaded images

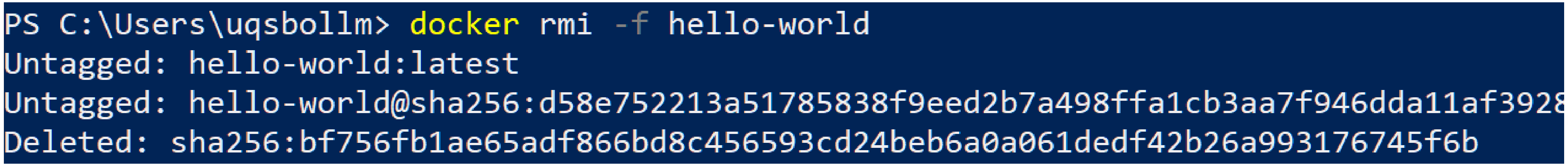
The caching of images can fill up your hard drive ...



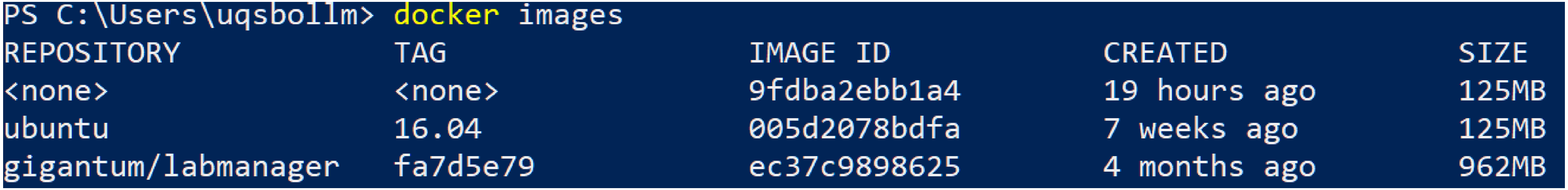
```
PS C:\Users\uqsbollm> docker images
REPOSITORY             TAG           IMAGE ID        CREATED          SIZE
<none>                 <none>        9fdba2ebb1a4    19 hours ago     125MB
ubuntu                 16.04         005d2078bdfa    7 weeks ago      125MB
gigantum/labmanager    fa7d5e79      ec37c9898625    4 months ago     962MB
hello-world            latest        bf756fb1ae65    5 months ago     13.3kB
```

in windows all docker images are stored in a single hyper-v virtual machine disk at:
C:\ProgramData\DockerDesktop\vm-data\DockerDesktop.vhdx

Clean up docker images



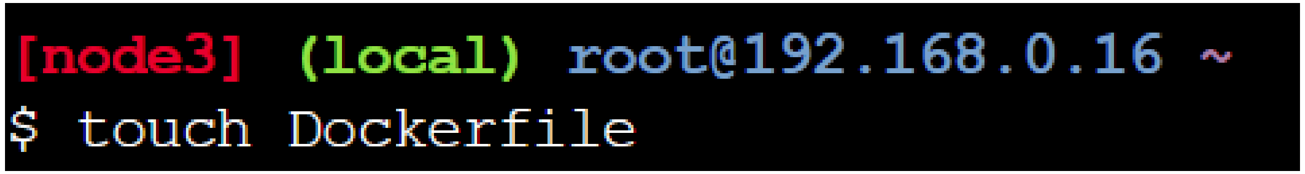
```
PS C:\Users\uqsbollm> docker rmi -f hello-world
Untagged: hello-world:latest
Untagged: hello-world@sha256:d58e752213a51785838f9eed2b7a498ffa1cb3aa7f946dda11af392
Deleted: sha256:bf756fb1ae65adf866bd8c456593cd24beb6a0a061dedf42b26a993176745f6b
```



```
PS C:\Users\uqsbollm> docker images
REPOSITORY             TAG                 IMAGE ID            CREATED             SIZE
<none>                 <none>              9fdba2ebb1a4        19 hours ago        125MB
ubuntu                 16.04               005d2078bdfa        7 weeks ago         125MB
gigantum/labmanager    fa7d5e79            ec37c9898625        4 months ago        962MB
```

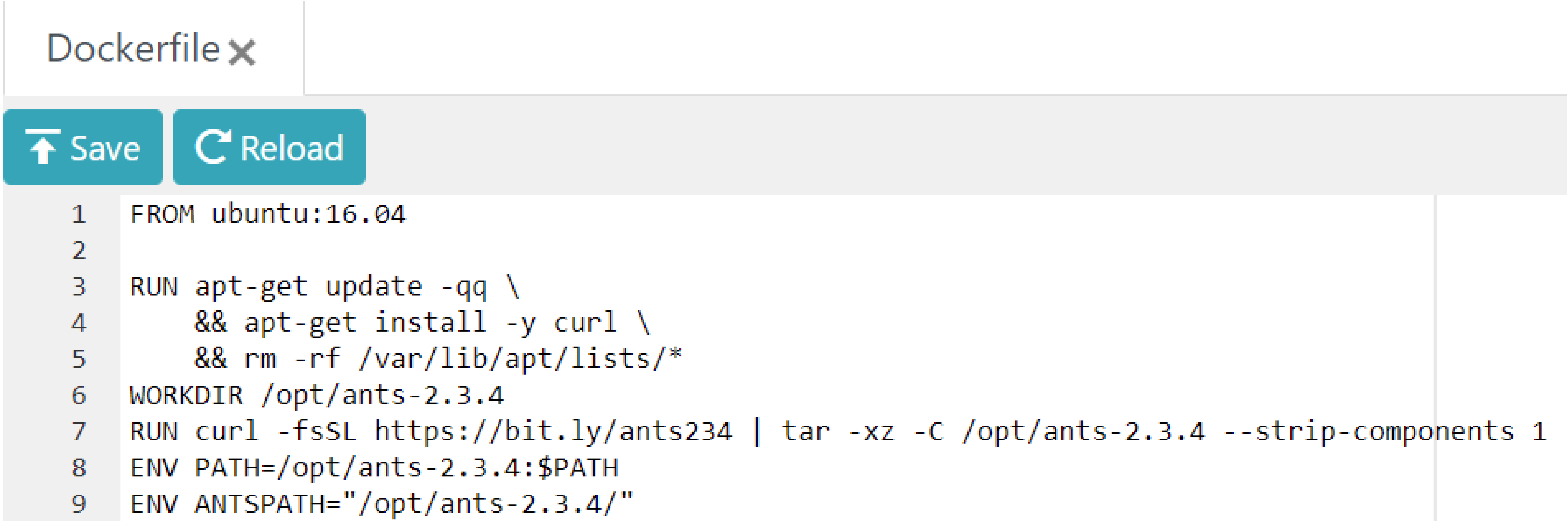
sometimes it can help to remove ALL images, this can be done using: **docker rmi $(docker images -q) --force**

Let’s write our own docker file ☺



```
[node3] (local) root@192.168.0.16 ~
$ touch Dockerfile
```

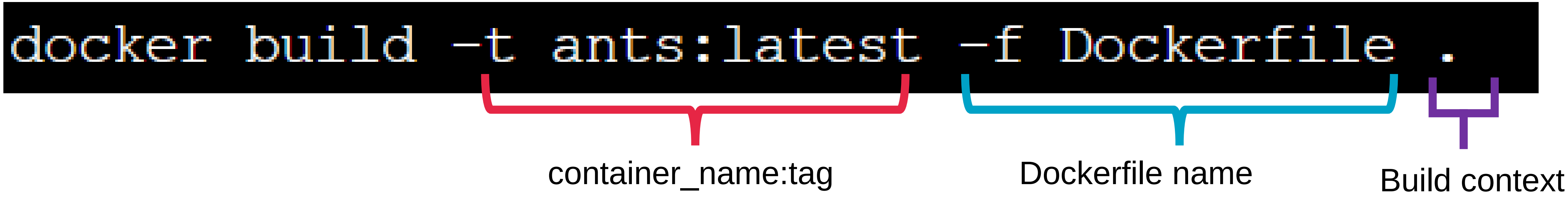
Our Dockerfile:



```
FROM ubuntu:16.04

RUN apt-get update -qq \
    && apt-get install -y curl \
    && rm -rf /var/lib/apt/lists/*
WORKDIR /opt/ants-2.3.4
RUN curl -fsSL https://bit.ly/ants234 | tar -xz -C /opt/ants-2.3.4 --strip-components 1
ENV PATH=/opt/ants-2.3.4:$PATH
ENV ANTSPATH="/opt/ants-2.3.4/"
```

Let’s build our own container

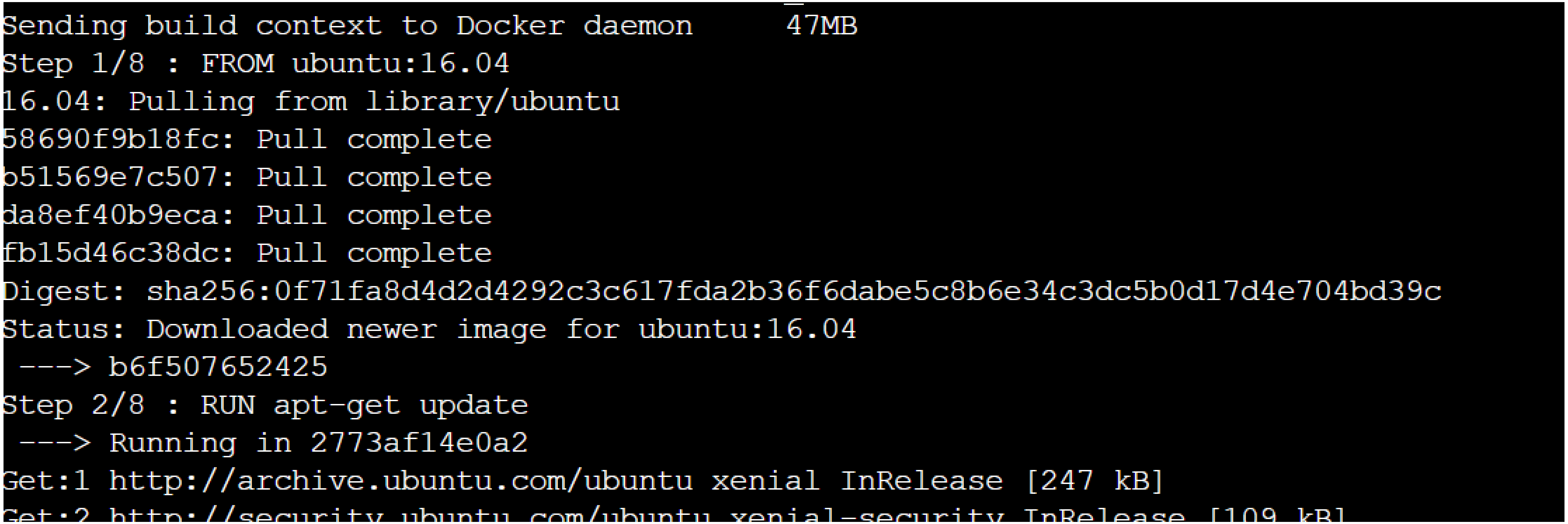


```
docker build -t ants:latest -f Dockerfile .
```

container_name:tag

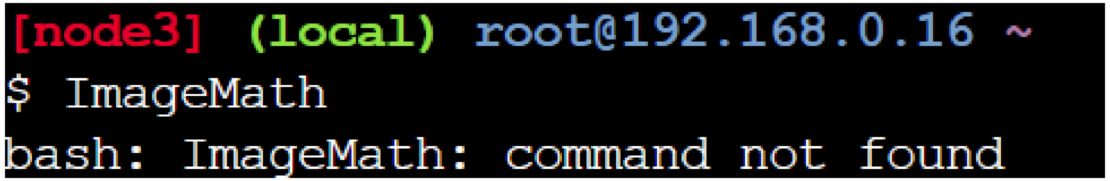
Dockerfile name

Build context

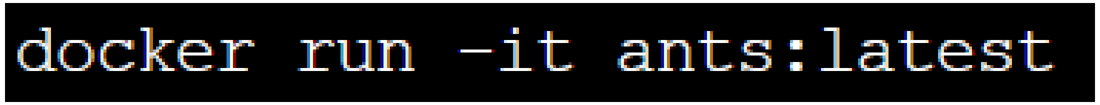


```
Sending build context to Docker daemon     47MB
Step 1/8 : FROM ubuntu:16.04
16.04: Pulling from library/ubuntu
58690f9b18fc: Pull complete
b51569e7c507: Pull complete
da8ef40b9eca: Pull complete
fb15d46c38dc: Pull complete
Digest: sha256:0f71fa8d4d2d4292c3c617fda2b36f6dabe5c8b6e34c3dc5b0d17d4e704bd39c
Status: Downloaded newer image for ubuntu:16.04
 ---> b6f507652425
Step 2/8 : RUN apt-get update
 ---> Running in 2773af14e0a2
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
```

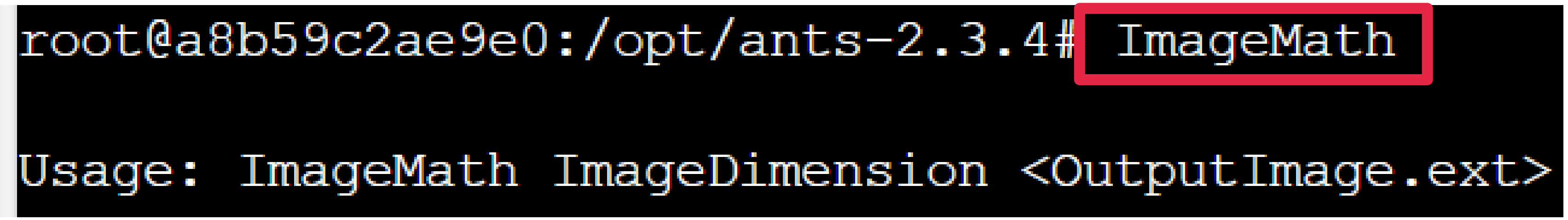
Let’s run our own container ☺



```
[node3] (local) root@192.168.0.16 ~
$ ImageMath
bash: ImageMath: command not found
```



```
docker run -it ants:latest
```



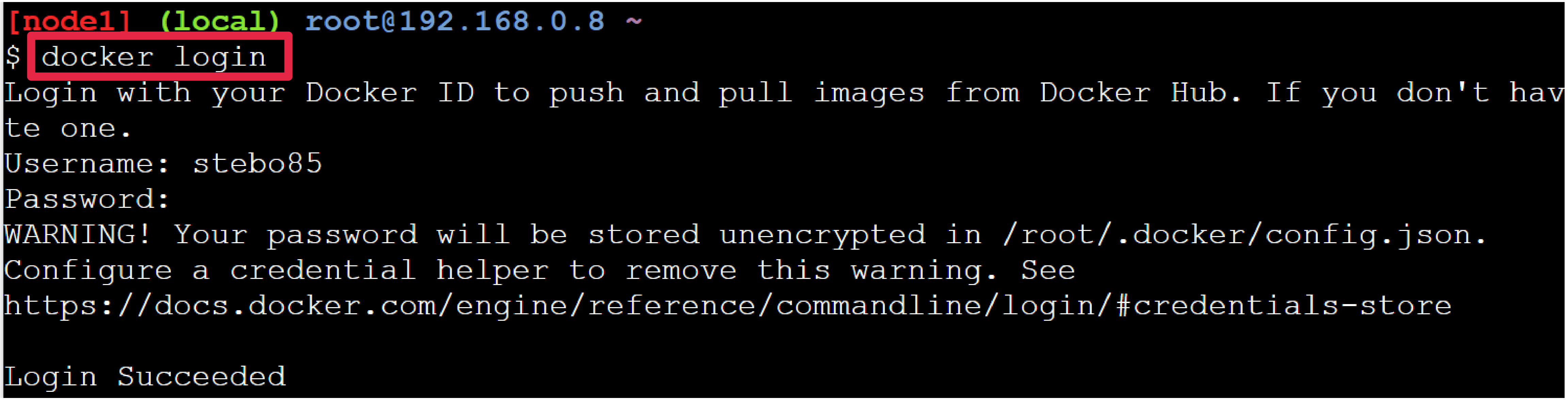
```
root@a8b59c2ae9e0:/opt/ants-2.3.4# ImageMath

Usage: ImageMath ImageDimension <OutputImage.ext>
```

Let’s share our image with the rest of the world ☺

First, exit the container with CTRL-D or CTRL-C

Then we need to login to Dockerhub:

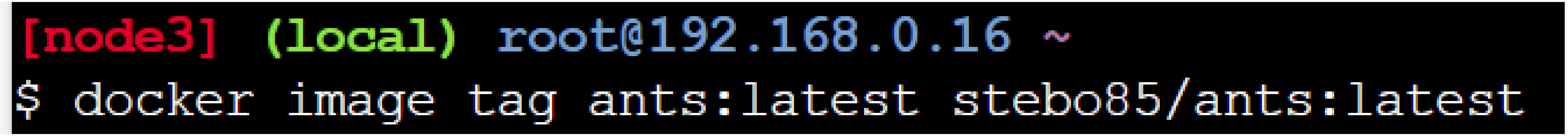


```
[node1] (local) root@192.168.0.8 ~
$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't hav
te one.
Username: stebo85
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

Let’s share our image with the rest of the world ☺

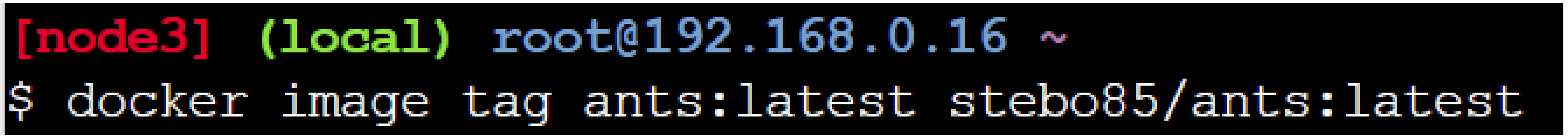
Then we need to tag the image with our docker username:



```
[node3] (local) root@192.168.0.16 ~
$ docker image tag ants:latest stebo85/ants:latest
```

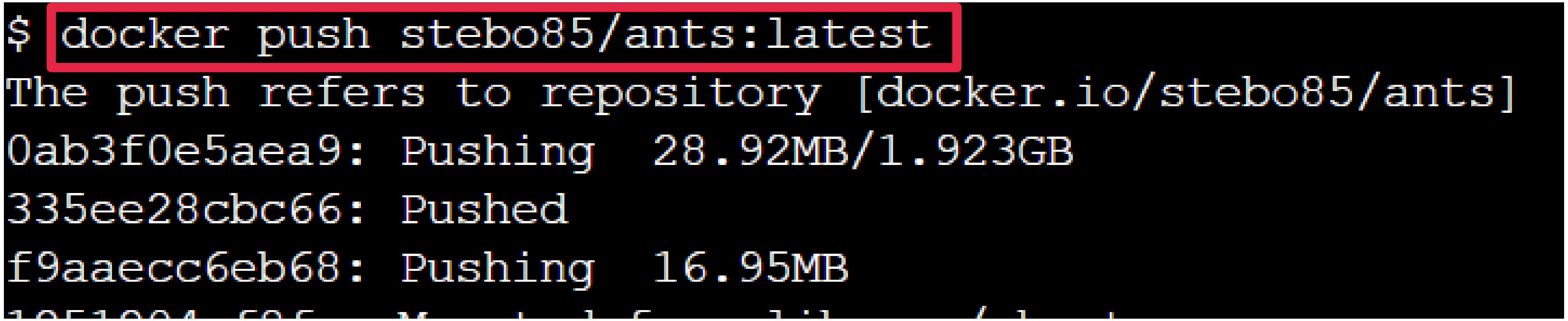
Let’s share our image with the rest of the world ☺

Then we need to tag the image with our docker username:



```
[node3] (local) root@192.168.0.16 ~
$ docker image tag ants:latest stebo85/ants:latest
```

Then we can push:

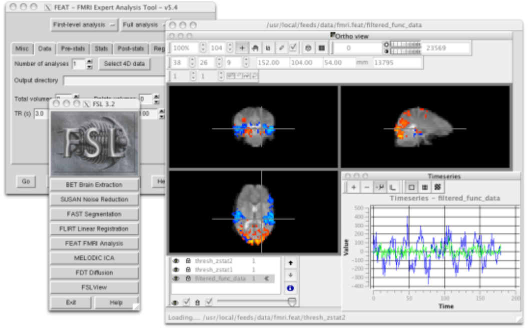


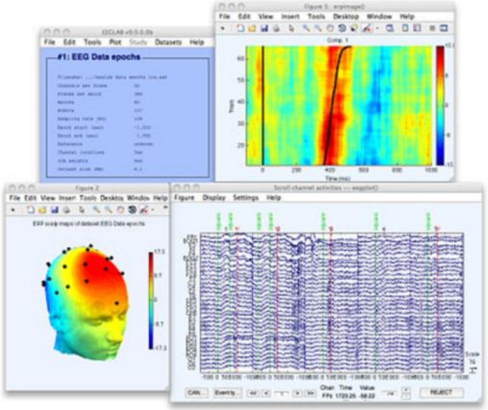
```
$ docker push stebo85/ants:latest
The push refers to repository [docker.io/stebo85/ants]
0ab3f0e5aea9: Pushing  28.92MB/1.923GB
335ee28cbc66: Pushed
f9aaecc6eb68: Pushing  16.95MB
```

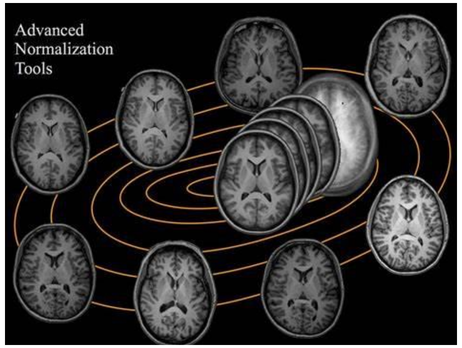
NeuroDesk - an example for the power of containerization in Science

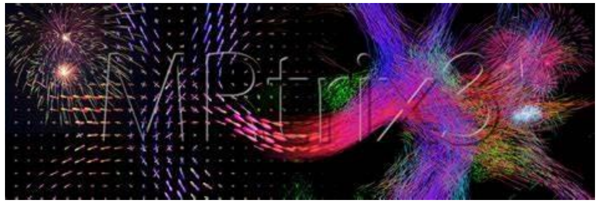
http://neurodesk.github.io/

Large ecosystem of scientific software …







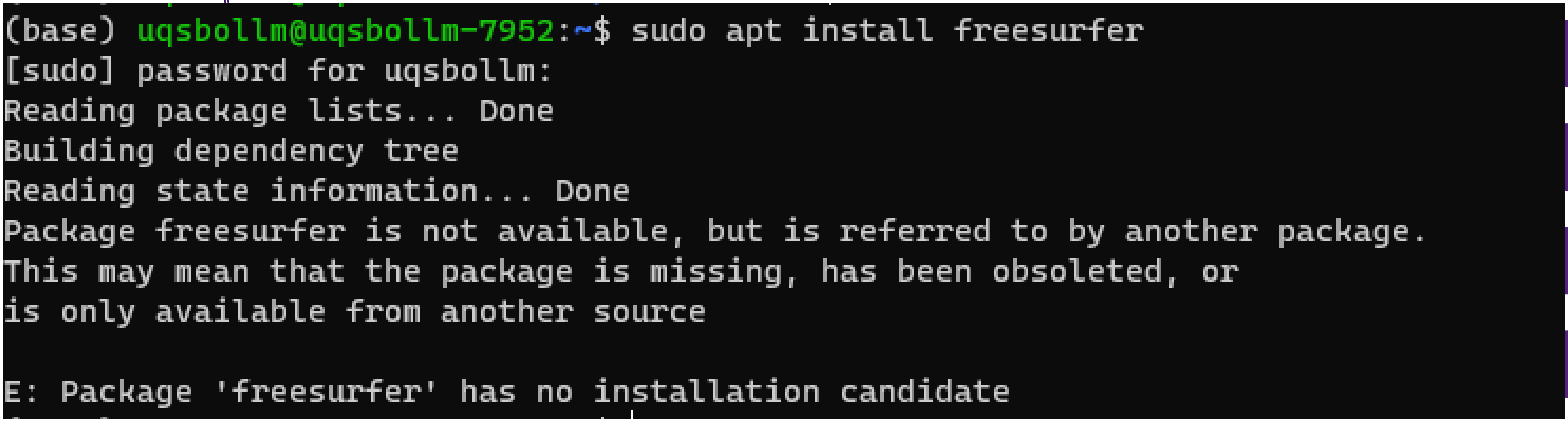


Scientific software can be challenging for researchers:

- Most tools require Linux

Scientific software can be challenging for researchers:

- Most tools require Linux
- Tools are not available in standard package systems

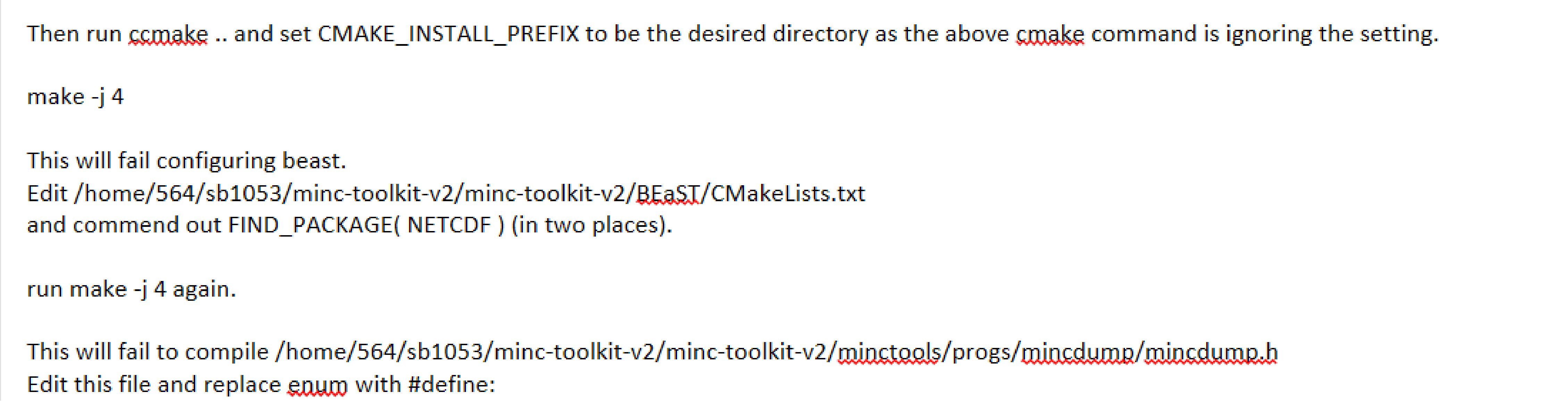


```
(base) uqsbollm@uqsbollm-7952:~$ sudo apt install freesurfer
[sudo] password for uqsbollm:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package freesurfer is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'freesurfer' has no installation candidate
```

Scientific software can be challenging for researchers:

- Most tools require Linux
- Tools are not available in standard package systems
- Compiling from source often a nightmare



Then run ccmake .. and set CMAKE_INSTALL_PREFIX to be the desired directory as the above cmake command is ignoring the setting.

make -j 4

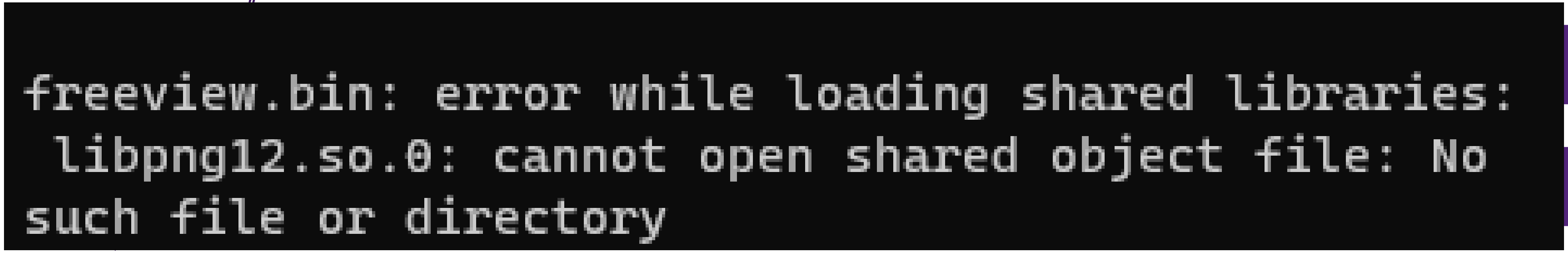
This will fail configuring beast.
Edit /home/564/sb1053/minc-toolkit-v2/minc-toolkit-v2/BEaST/CMakeLists.txt
and commend out FIND_PACKAGE(NETCDF) (in two places).

run make -j 4 again.

This will fail to compile /home/564/sb1053/minc-toolkit-v2/minc-toolkit-v2/minctools/progs/mincdump/mincdump.h
Edit this file and replace enum with #define:

Scientific software can be challenging for researchers:

- Most tools require Linux
- Tools are not available in standard package systems
- Compiling from source often a nightmare
- Conflicting dependencies



```
freeview.bin: error while loading shared libraries:
 libpng12.so.0: cannot open shared object file: No
such file or directory
```

Scientific software can be challenging for researchers:

- Most tools require Linux
- Tools are not available in standard package systems
- Compiling from source often a nightmare
- Conflicting dependencies
- Reinstalling tools on different platforms takes time

Scientific software can be challenging for researchers:

- Most tools require Linux
- Tools are not available in standard package systems
- Compiling from source often a nightmare
- Conflicting dependencies
- Reinstalling tools on different platforms takes time
- Differing results between software versions

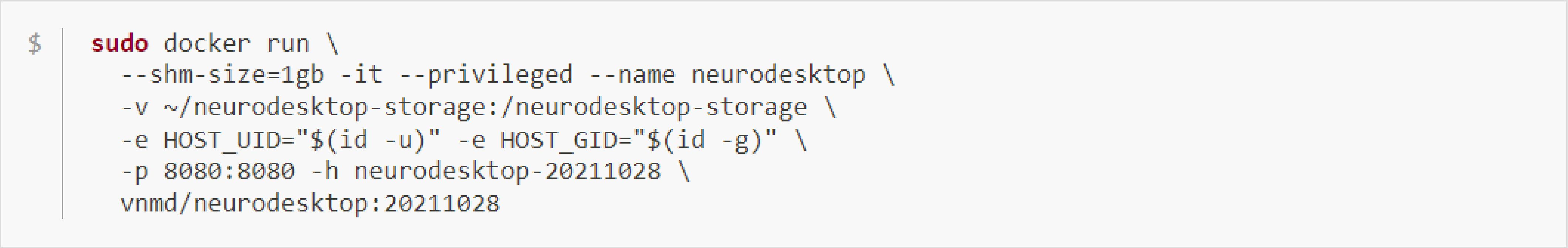
Let’s test it ☺

NeuroDesktop is a full Linux desktop inside a docker container!

First we need to create a new instance:

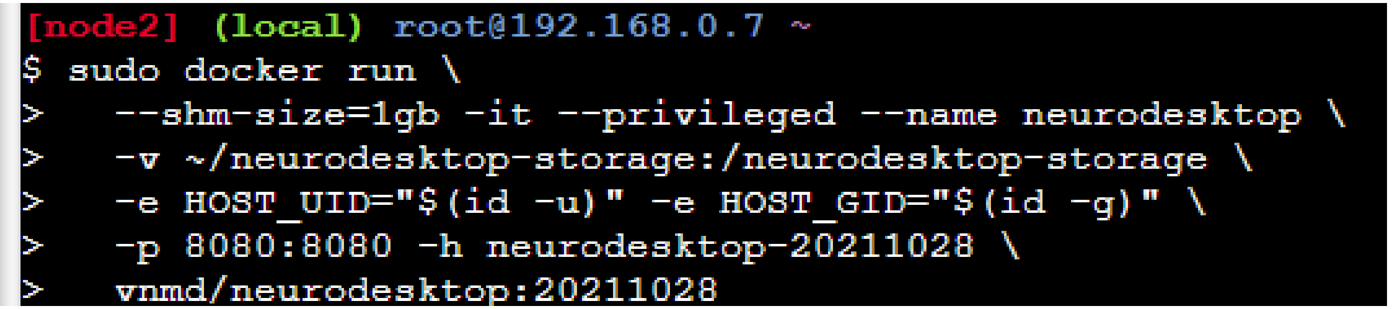
+ ADD NEW INSTANCE

Then copy the run command from neurodesk.github.io -> linux:



```
$ sudo docker run \
    --shm-size=1gb -it --privileged --name neurodesktop \
    -v ~/neurodesktop-storage:/neurodesktop-storage \
    -e HOST_UID="$(id -u)" -e HOST_GID="$(id -g)" \
    -p 8080:8080 -h neurodesktop-20211028 \
    vnmd/neurodesktop:20211028
```

and paste in terminal (CTRL-SHIFT-V):



```
[node2] (local) root@192.168.0.7 ~
$ sudo docker run \
>     --shm-size=1gb -it --privileged --name neurodesktop \
>     -v ~/neurodesktop-storage:/neurodesktop-storage \
>     -e HOST_UID="$(id -u)" -e HOST_GID="$(id -g)" \
>     -p 8080:8080 -h neurodesktop-20211028 \
>     vnmd/neurodesktop:20211028
```

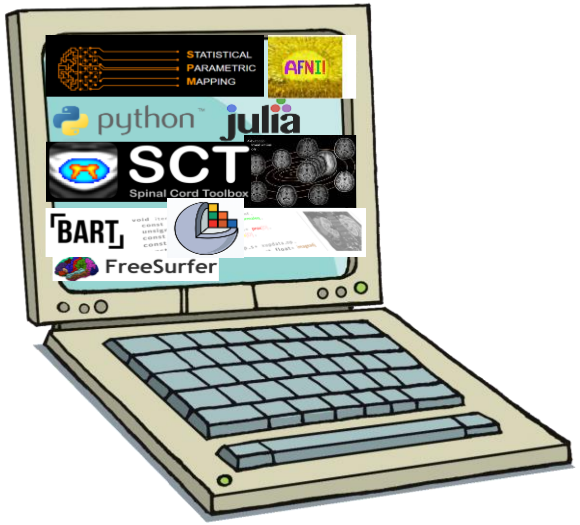
NeuroDesktop - **http://neurodesk.github.io/**

- Everything in a 1.25 GB docker container ☺
- Based on Apache Guacamole (Browser interface!)
- Only dependency is Docker
- >200GB of neuro-imaging software is delivered in unpacked singularity containers on demand via CVMFS
- Data connected via cloud storage or local directory

A complete suite of neuroimaging tools …

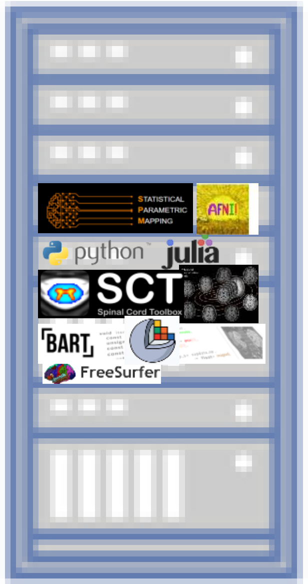
… reproducible, reusable, findable, interoperable, scalable, citable, shareable, accessible ….

… on your notebook



… on your lab workstation

… on the university's high performance cluster



… on any cloud provider

… bringing the data analysis environment to the data using software containers - even behind hospital firewalls … :)

Questions?