FAIR_bioinfo for bioinformaticians

Introduction to the tools of reproducibility in bioinformatics

C. Hernandez[1] T. Denecker[1] J.Sellier[2] C. Toffano-Nioche[1]

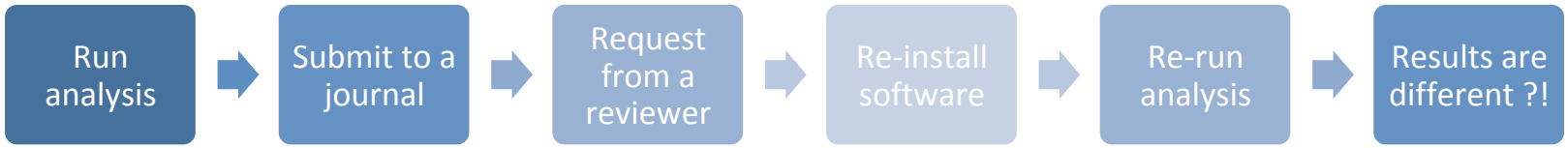
[1]Institute for Integrative Biology of the Cell (I2BC)
UMR 9198, Université Paris-Sud, CNRS, CEA
91190 - Gif-sur-Yvette, France

[2]Institut Français de Bioinformatique
à compléter

Sept. 2020

Introduction

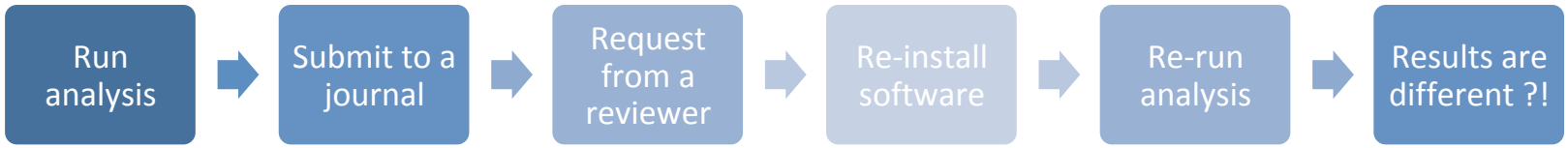
A (not-so-uncommon) nightmare



What changed?

Introduction

A (not-so-uncommon) nightmare



What changed?

- Software version
- Libraries version
- OS version
- ..?

Different levels of encapsulation

Goal : capture the system environment of applications (OS, packages, libraries,. . .) to control their execution.

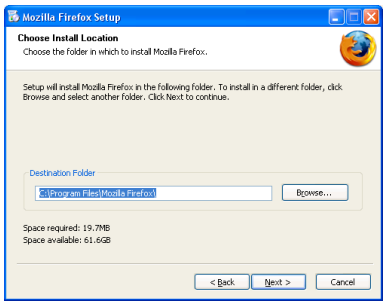
- Hardware virtualisation (virtual machines) 
- OS virtualisation (images and containers) 

- Environment management CONDA

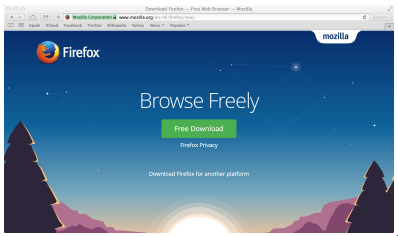
Encapsulation

Let's say we want to install Firefox...

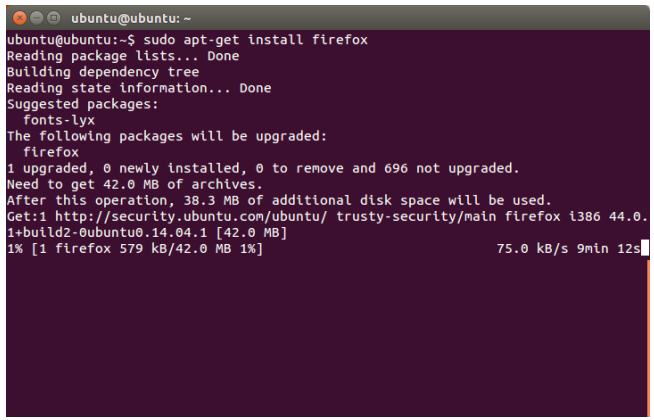
Windows



MacOS



Unix-based



Encapsulation

Windows MacOS Unix-based

We started with a computer using a specific OS...
And inside this environment, we installed a new application.
Applications rely on dependencies, e.g. external libraries.

Encapsulation

Windows MacOS Unix-based

Usually dependencies of different applications don't interfere.
But what if we want to test the latest version of our favourite tool?
There might be conflicts...

Encapsulation : hardware virtualisation

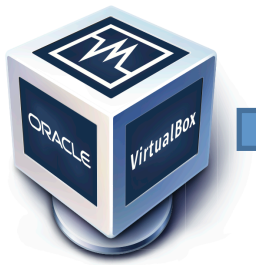
Idea: use virtual machines

Pros:

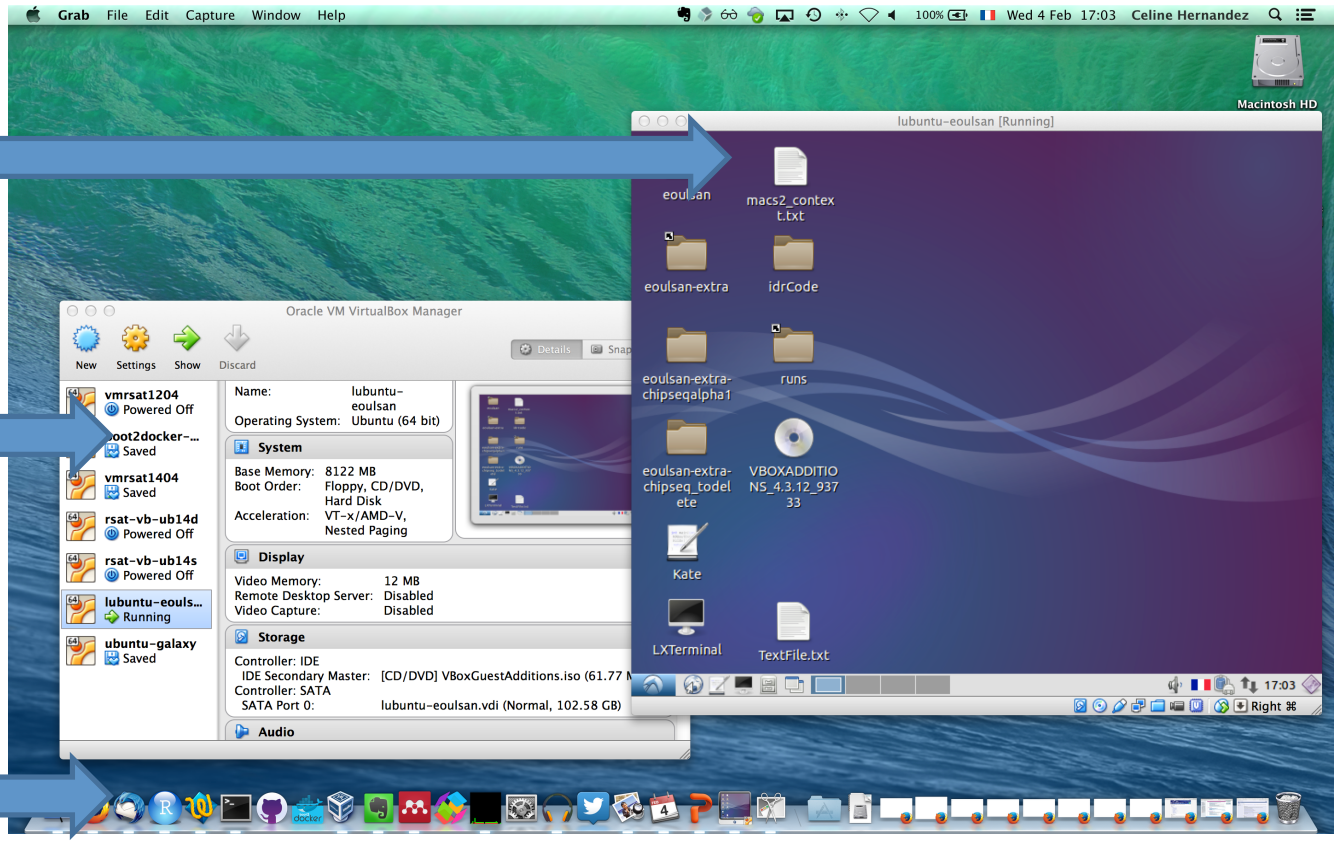
- Each application gets a completely different and independent environment
- Virtual machines can be transferred to another computer (using the same manager)

Encapsulation : hardware virtualisation

Ubuntu



MacOS



Encapsulation : hardware virtualisation

Idea: use virtual machines

Pros: transferable independent environments

Cons:

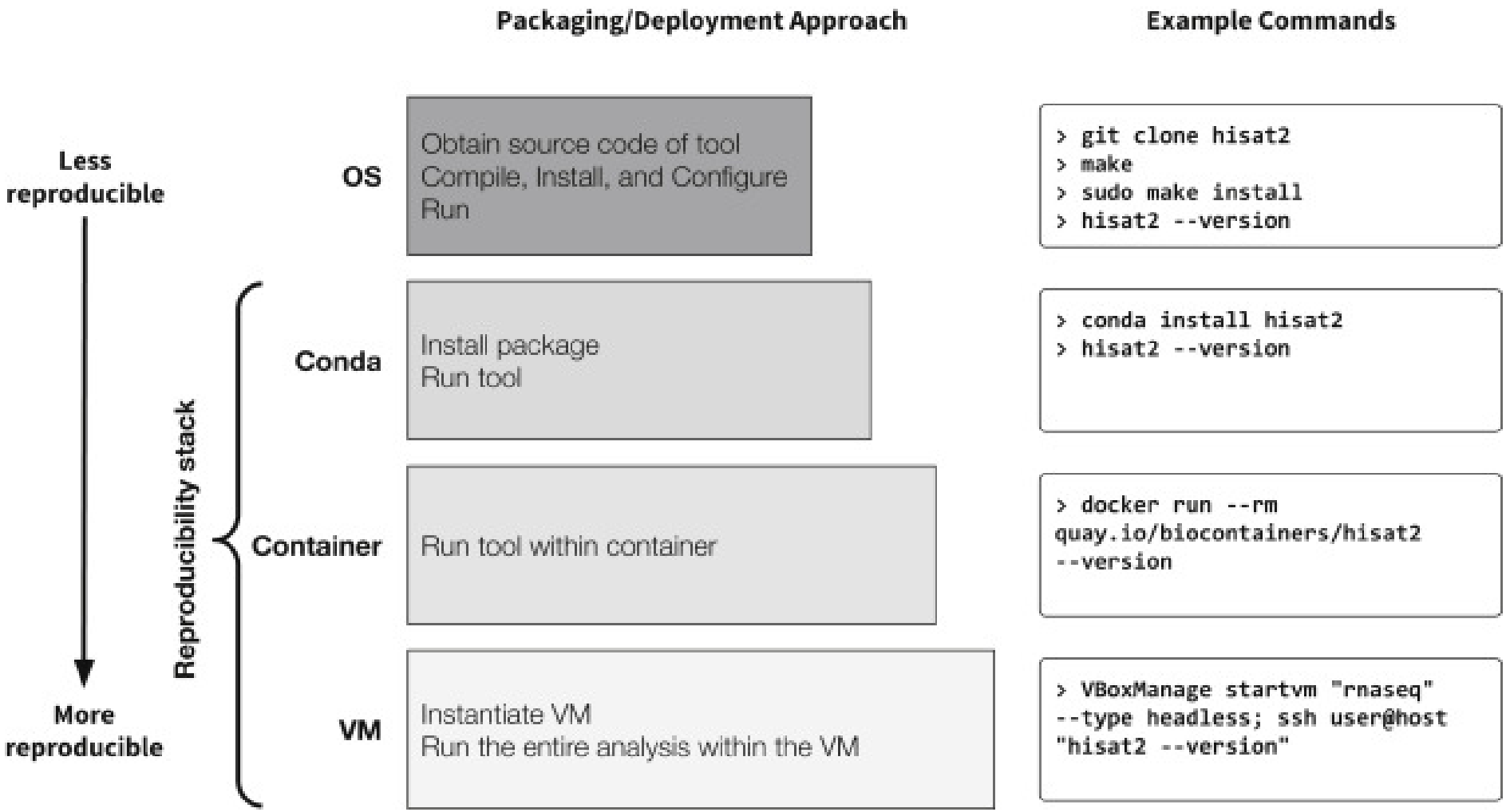
- Redundancy between VMs
- Heavy to set up
- No automation

Encapsulation : OS virtualisation

Windows MacOS Unix-based

Idea: "trick" applications into believing that they are in a different OS than the host's
Avoid redundancy.

Encapsulation : OS virtualisation



Practical Computational Reproducibility in the Life Sciences - Björn Grüning et al (2018)

What is Docker?

Docker is not very “old”

- First commit January 2013
- First version March 2013
- Version 1.0 in June 2014

But its adoption was fast

- Officially packaged in Ubuntu since 2014 (v14.04)

What is Docker?

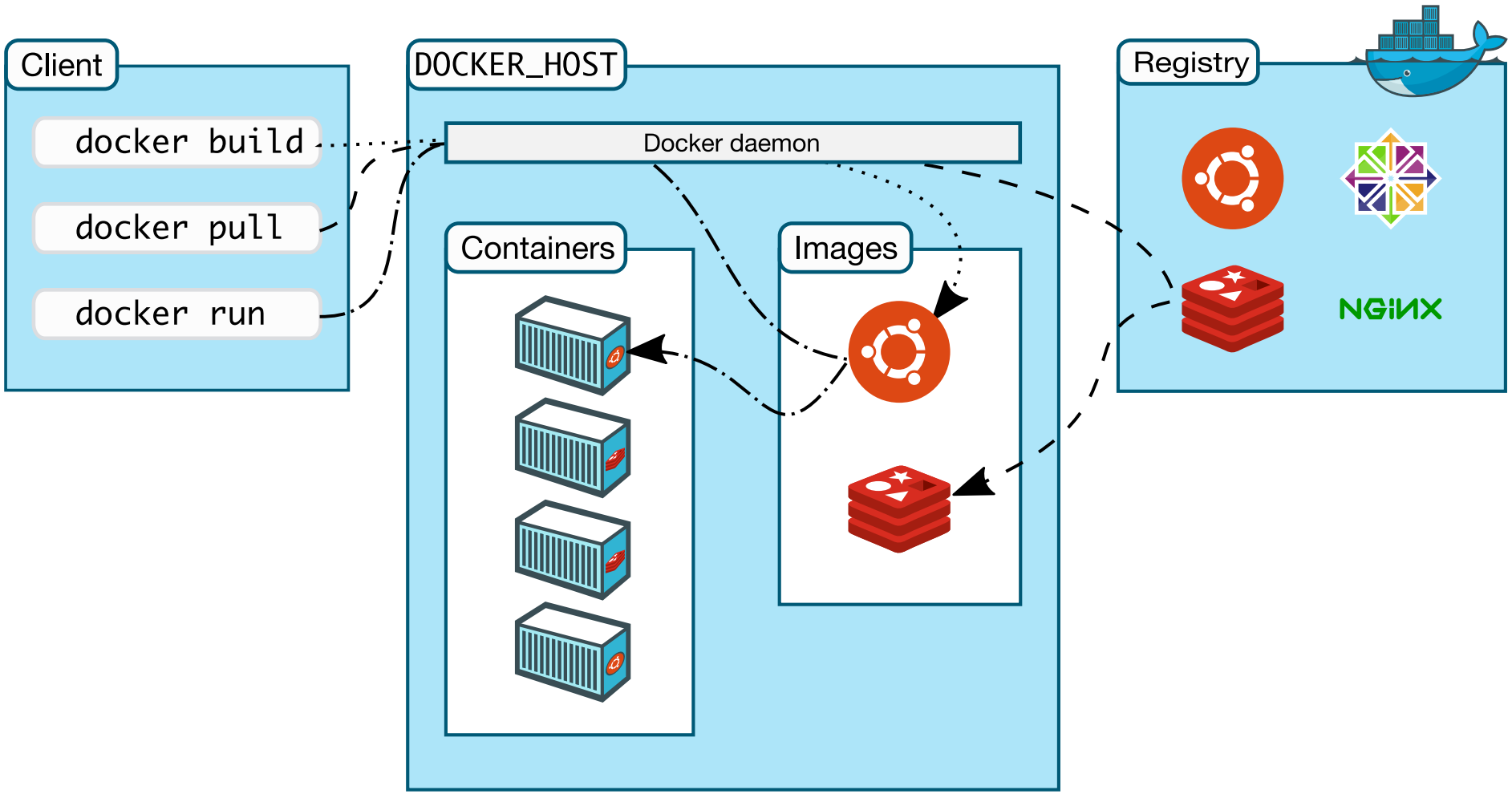
Image

- Set of libraries and functions
- Fixed. Cannot be modified
- Can be stored/shared online
- Can be automatically built

Container

- "Active image"
- Can be modified (interactive)
- Can be turned into an image
- One image, many containers

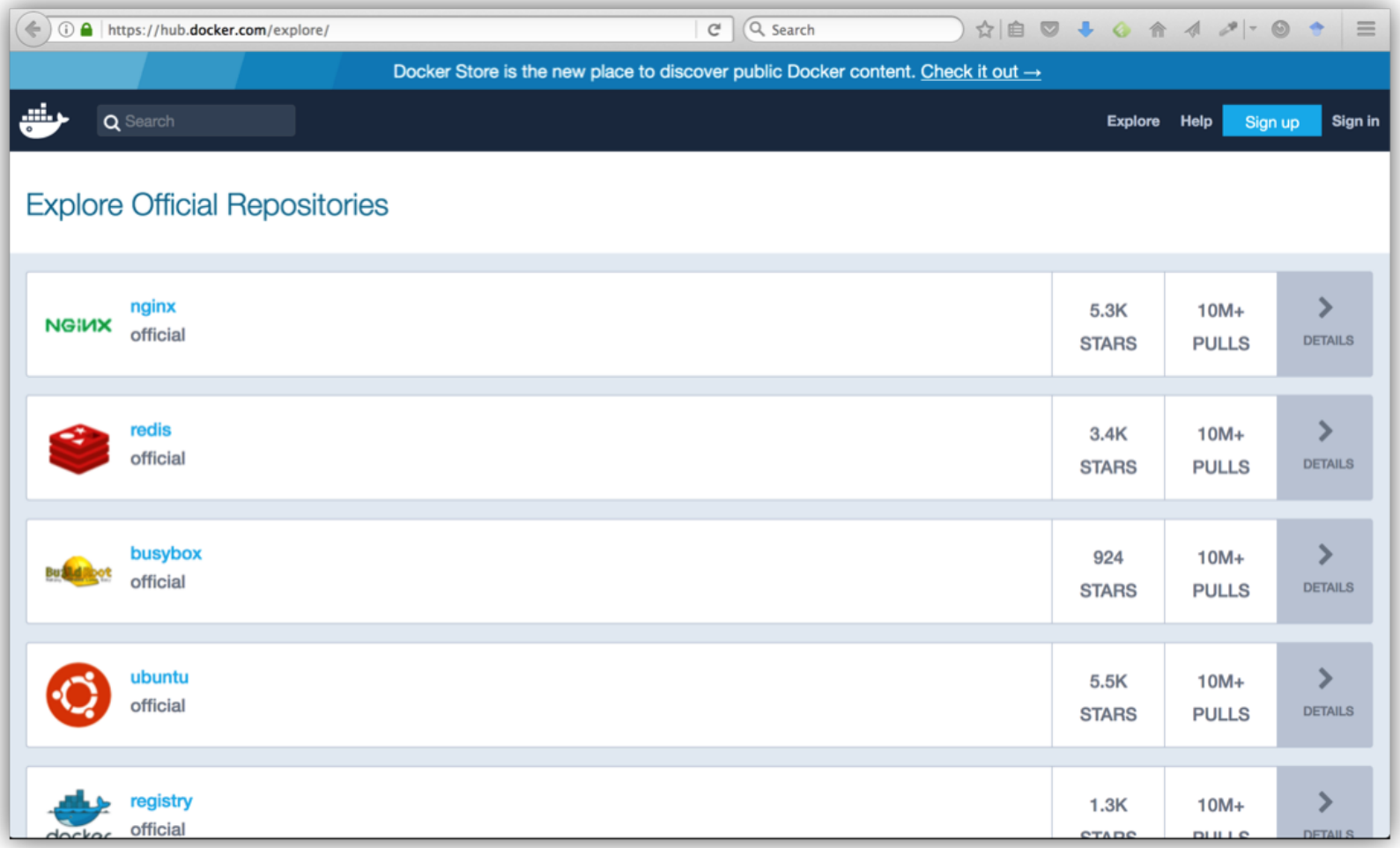
What is Docker?



(https://docs.docker.com/get-started/overview/)

What is Docker?

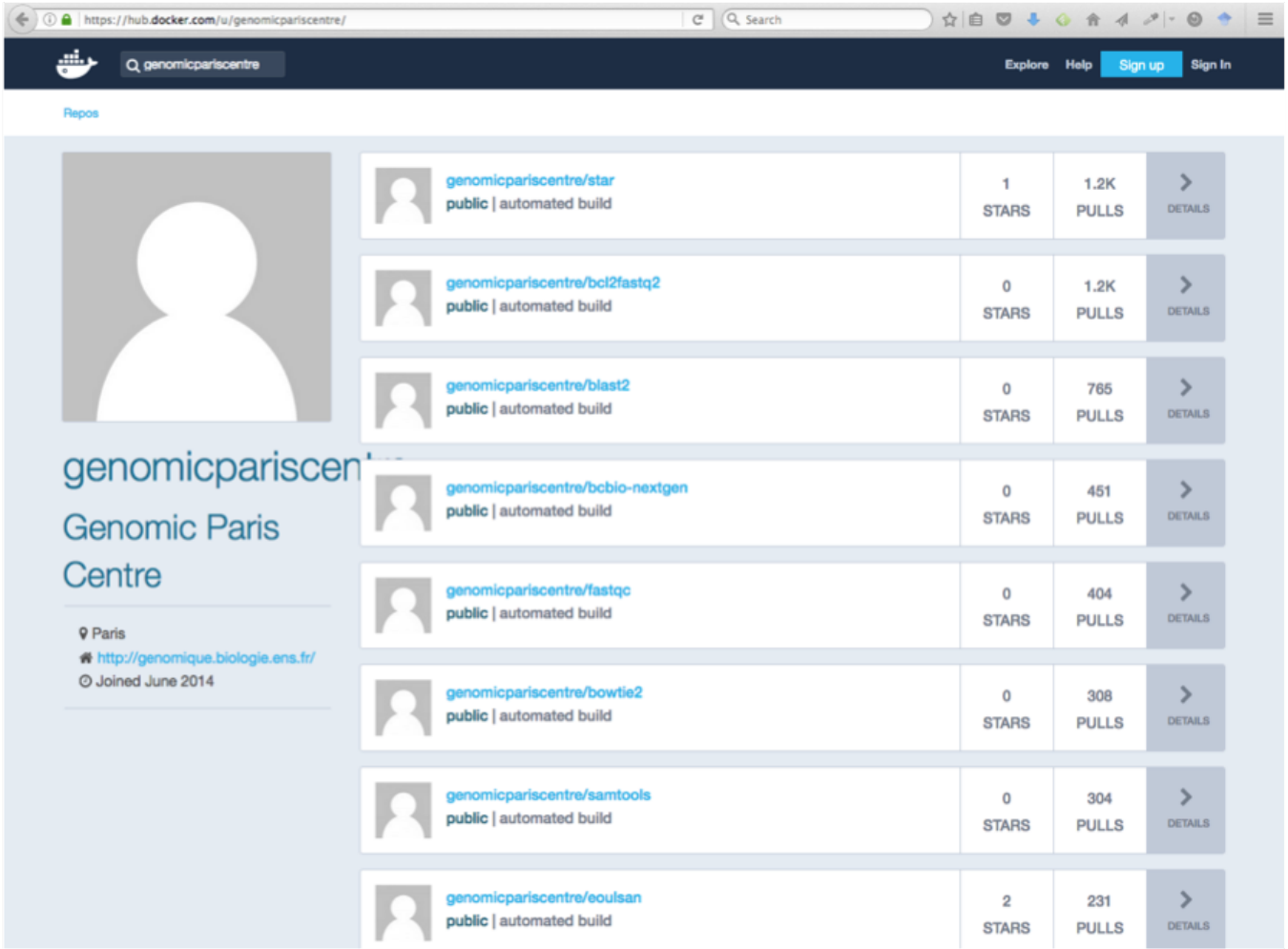
DockerHub



(https://hub.docker.com/)

What is Docker?

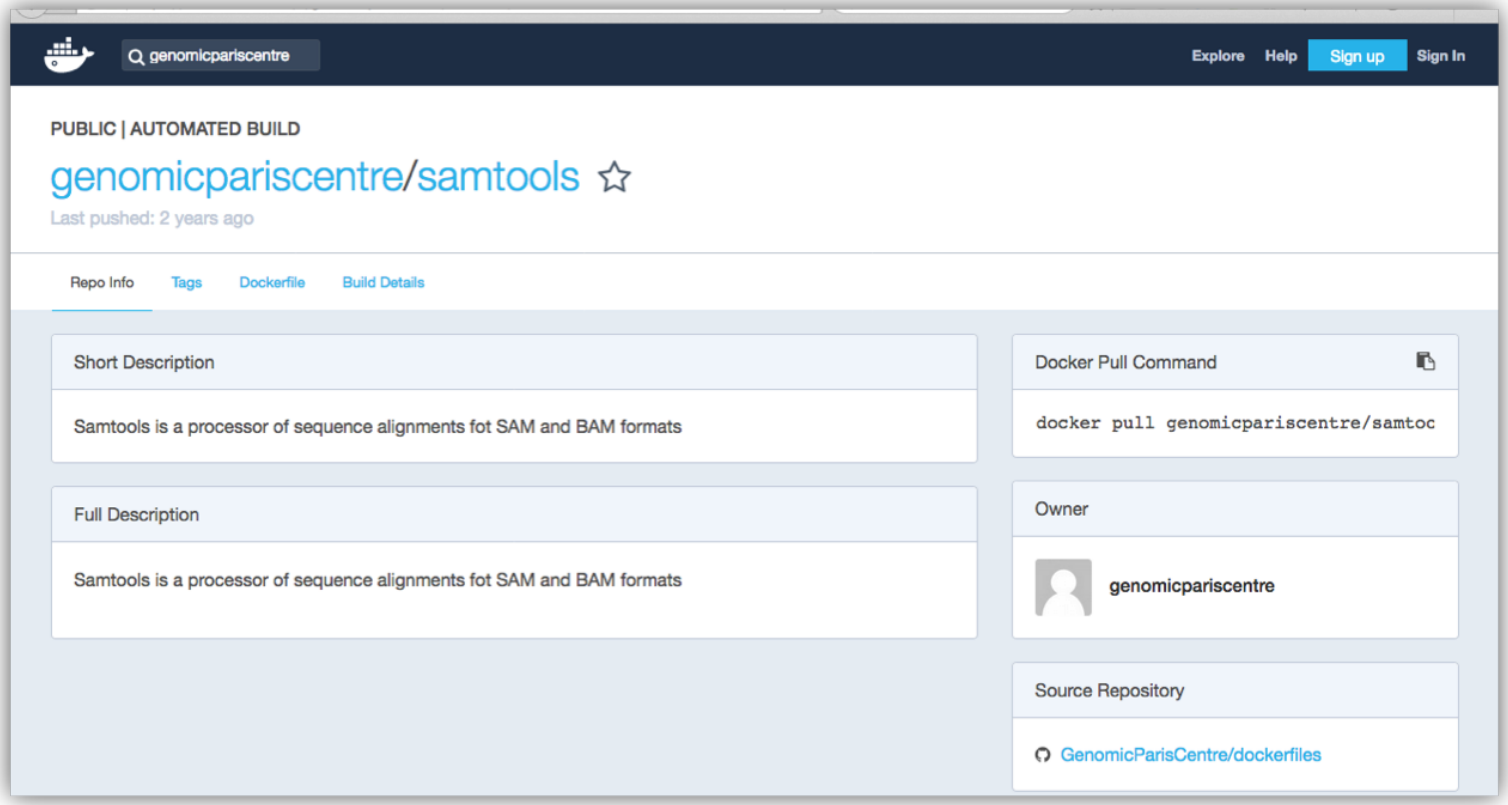
Usermade images (1/2)



(https://hub.docker.com/u/genomicpariscentre/)

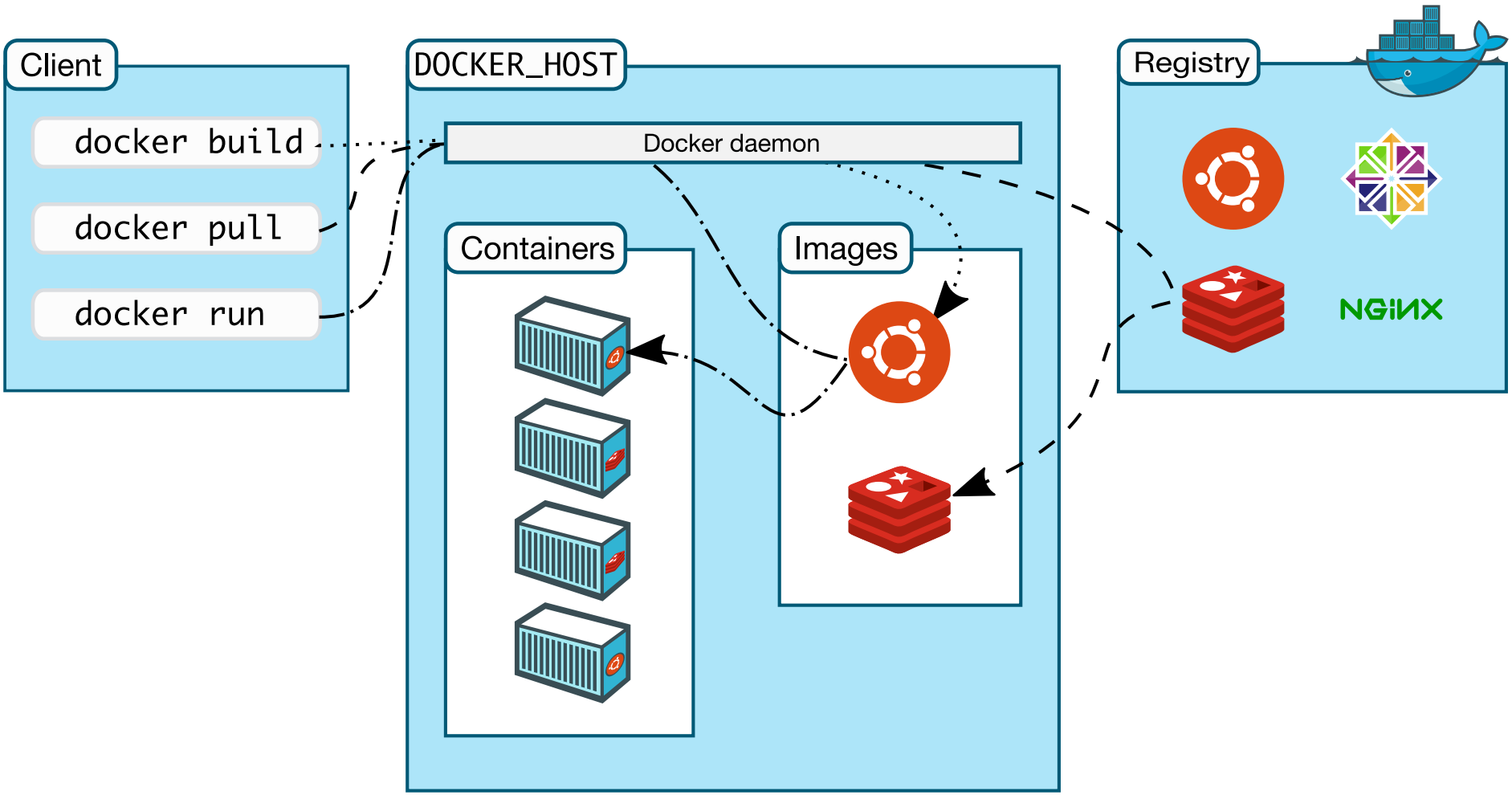
What is Docker?

Usermade images (2/2)
Be critical!



(https://hub.docker.com/r/genomicpariscentre/samtools/)

What is Docker?



(https://docs.docker.com/get-started/overview/)

What is Docker?

Other commands :

- docker images : list images available locally
- docker ps : status of containers
- docker rm : delete a container
- docker rmi : delete an image
- ...

(More details during the practical session.)

Encapsulation : OS virtualisation

OS virtualisation vs hardware virtualisation

Pros:

- Speed
  - Installation is faster
  - No boot time
- Lightweight
  - Minimal base OS
  - Minimal libraries and application set
- Easy sharing of applications

Encapsulation : OS virtualisation

Windows MacOS Unix-based

Cons:

- Needs root access (Singularity)
- Changes of policies of the Docker company

Docker policy

Update of the Docker Image retention policy (13/08/2020)

What is a container image retention limit and how does it affect my account?

Image retention is based on the activity of each individual image stored within a user account. If an image has not either been pulled or pushed in the amount of time specified in your subscription plan, the image will be tagged “inactive.” Any images that are tagged as “inactive” will be scheduled for deletion. Only accounts that are on the **Free** individual or organization plans will be subject to image retention limits. A new dashboard will also be available in Docker Hub that offers the ability to view the status of all of your container images.

What are the new container image retention limits?

Docker is introducing a container image retention policy which will be enforced starting November 1, 2020. The container image retention policy will apply to the following plans:

- Free plans will have a 6 month image retention limit
- Pro and Team plans will have unlimited image retention

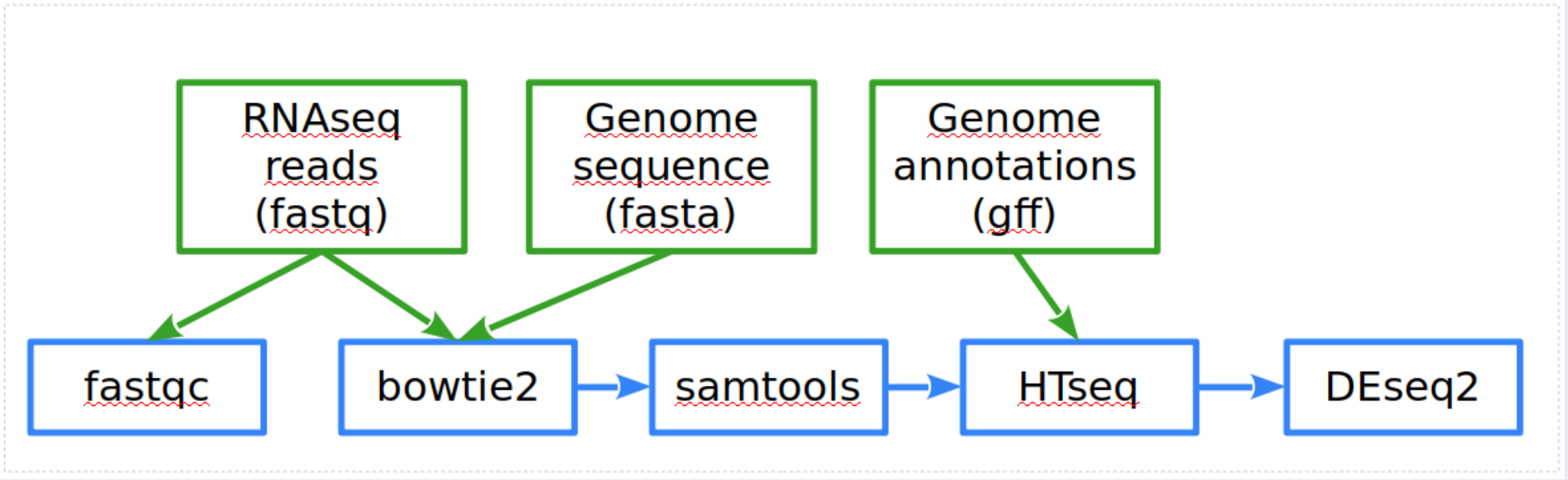
https://www.docker.com/pricing/retentionfaq

Practical session

Practical session : Docker and Samtools.
See companion document.

Practical session

Analysis workflow



green=input, blue=tool

fastqc	control quality of the input reads
bowtie2	reads mapping on the genome sequence
samtools	mapped reads selection & formatting
HTseq	count table of mapped reads on genes (annotations)
DEseq2	statistical analysis: genes list having differential expression

Practical session

Savoir FAIRe

- (Installation de Docker)
- Learn the structure of a Docker command
- Pull a pre-defined image available on the DockerHub
- Start a container
- Bonus: build a Dockerfile