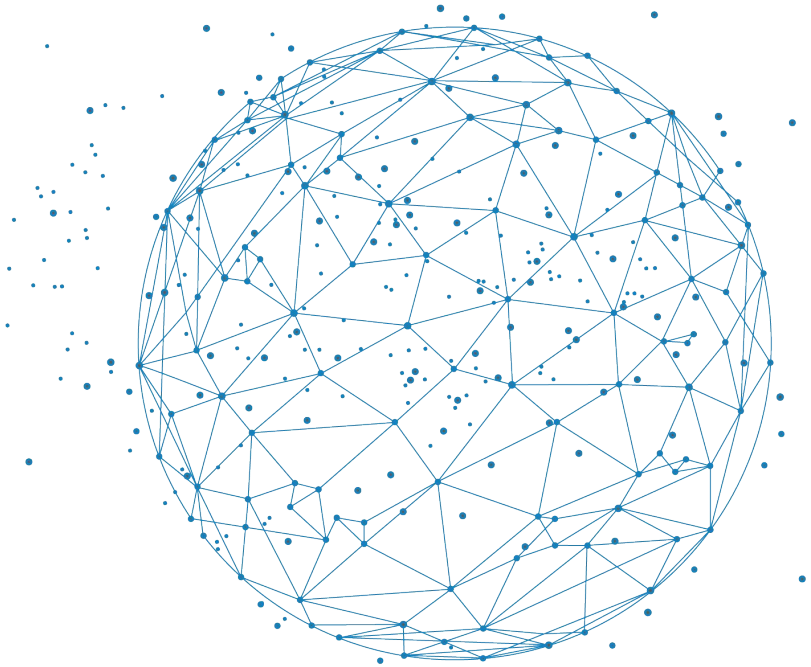


Use case 2 : Encapsulation

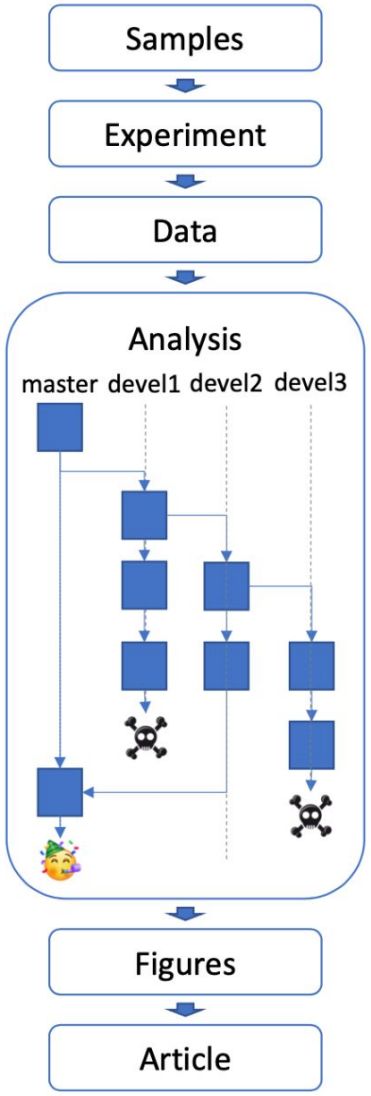
Céline Hernandez, I2BC, 0000-0001-8664-1340

IFB INSTITUT FRANÇAIS DE BIOINFORMATIQUE

elixir FRANCE

cnrs INRAE Inserm cea INVESTIR L'AVENIR

A (not-so-uncommon) nightmare



A (not-so-uncommon) nightmare

Samples

Experiment

Data

Analysis

master devel1 devel2 devel3

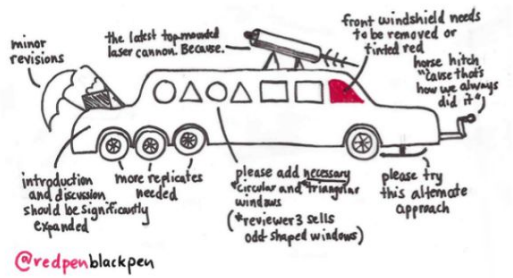
Figures

Article

Submit to a journal

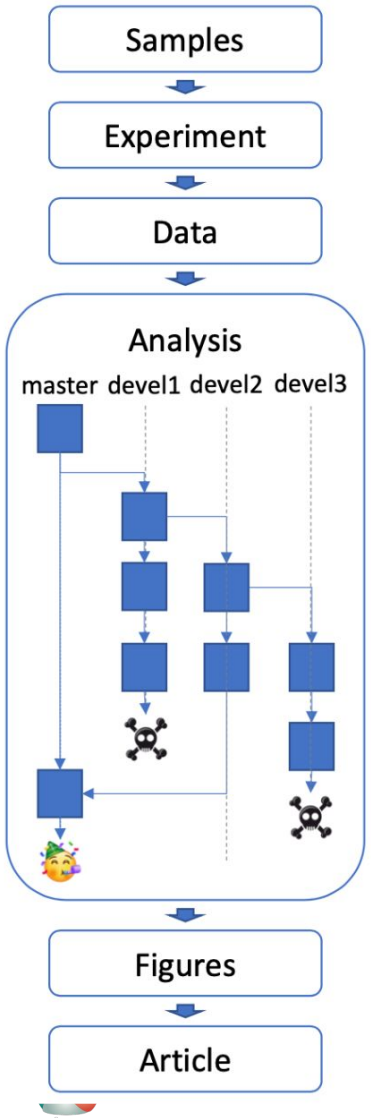
Reviewer 1

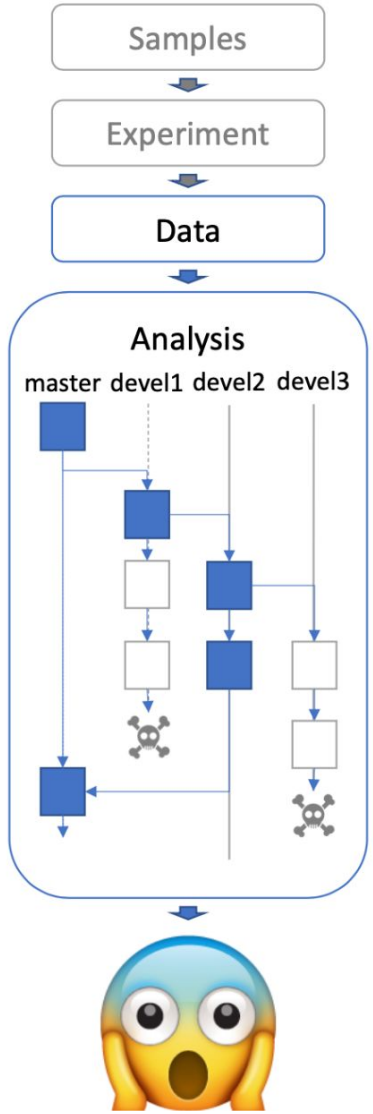
Reviewer 2



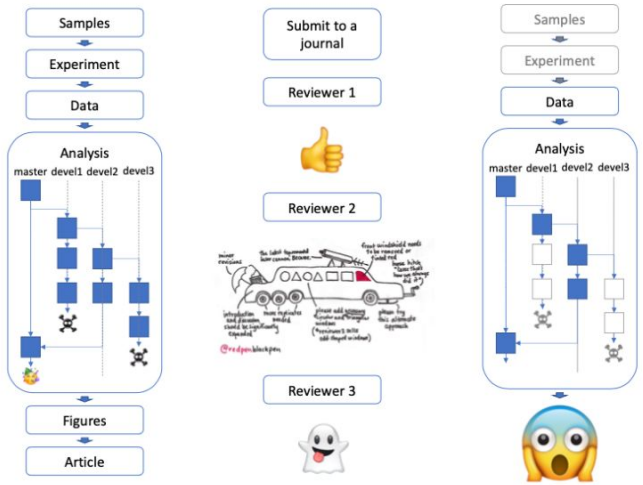
Reviewer 3

A (not-so-uncommon) nightmare





A (not-so-uncommon) nightmare



What changed?

- Package
- Software
- Libraries
- Environment variables

- OS version
- Computer
- ..?

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

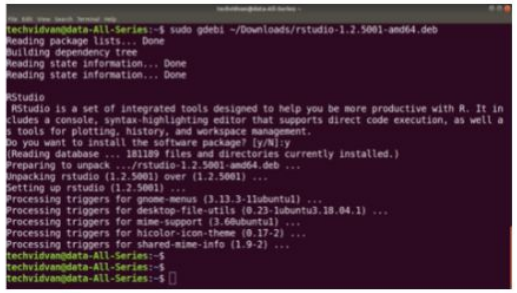
Let's say we want to install RStudio...

Install Rstudio ?

MacOS

Windows

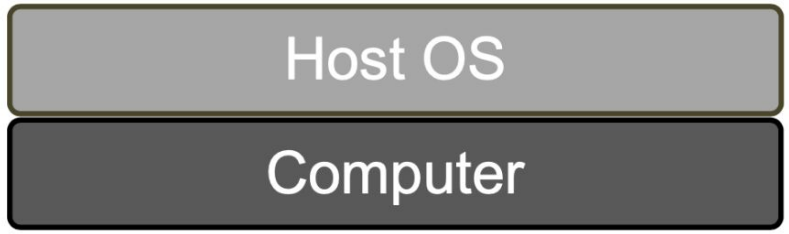
Unix-based



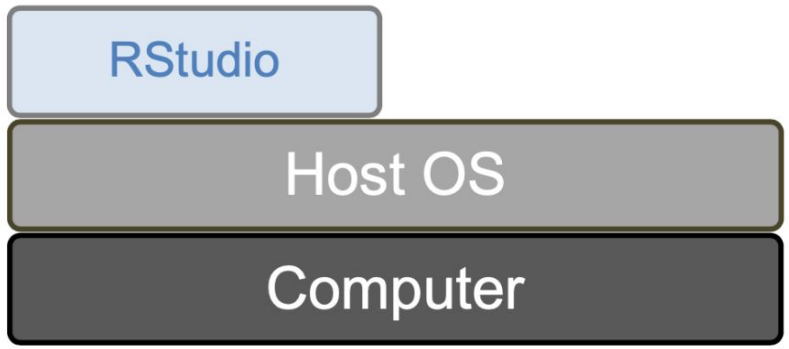
Use Rstudio

Encapsulation

We started with a computer using a specific OS...

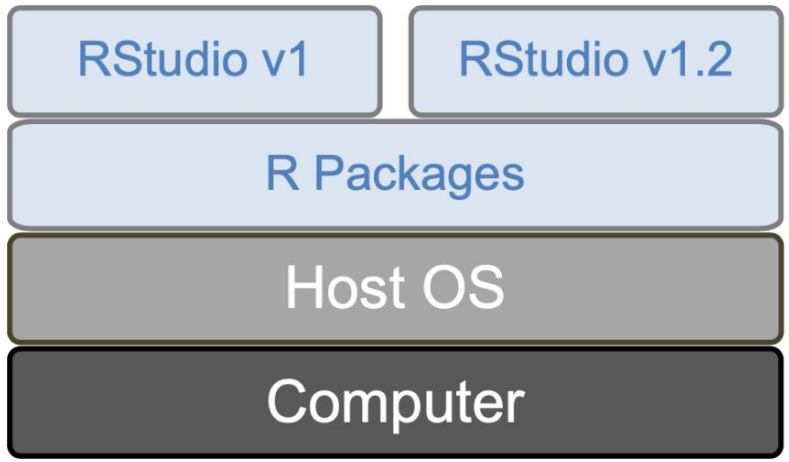


Encapsulation



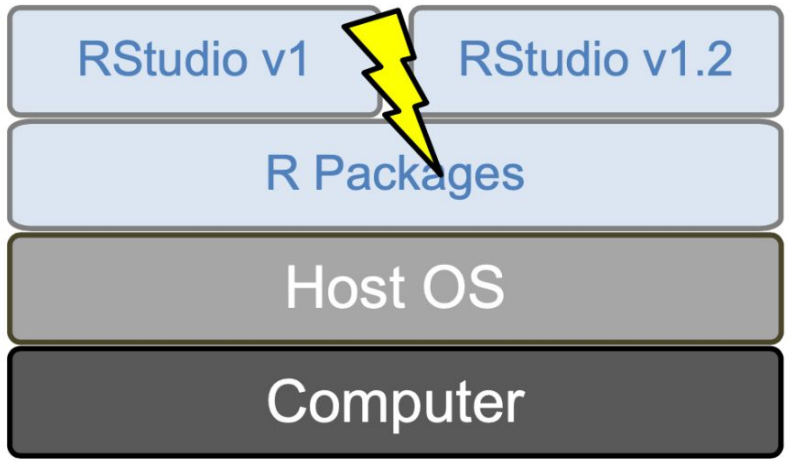
We started with a computer using a specific OS...
And inside this environment, we installed a new application.

Encapsulation



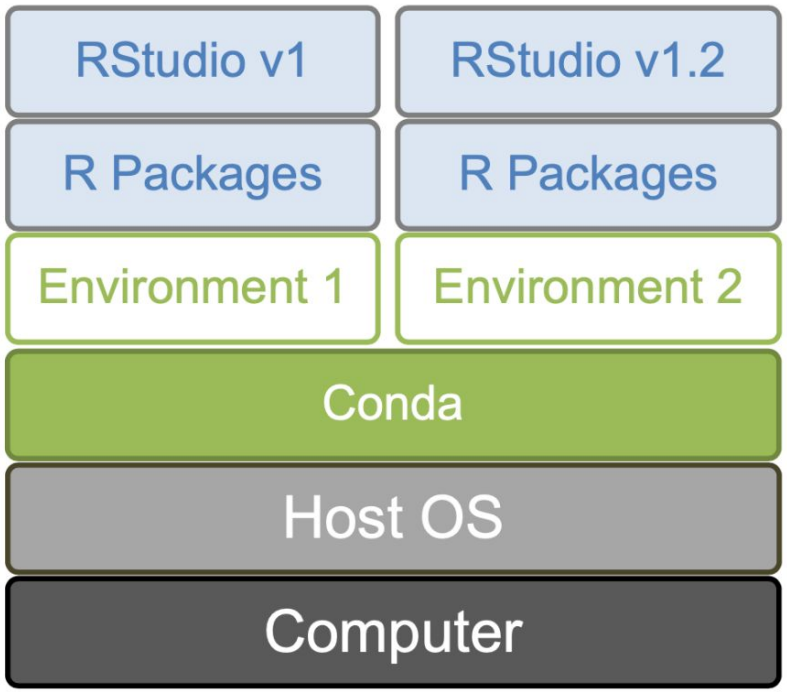
Usually dependencies of different applications don't interfere.
But what if we want to test the latest version of our favourite tool?
There might be conflicts. . .

Encapsulation



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. . .

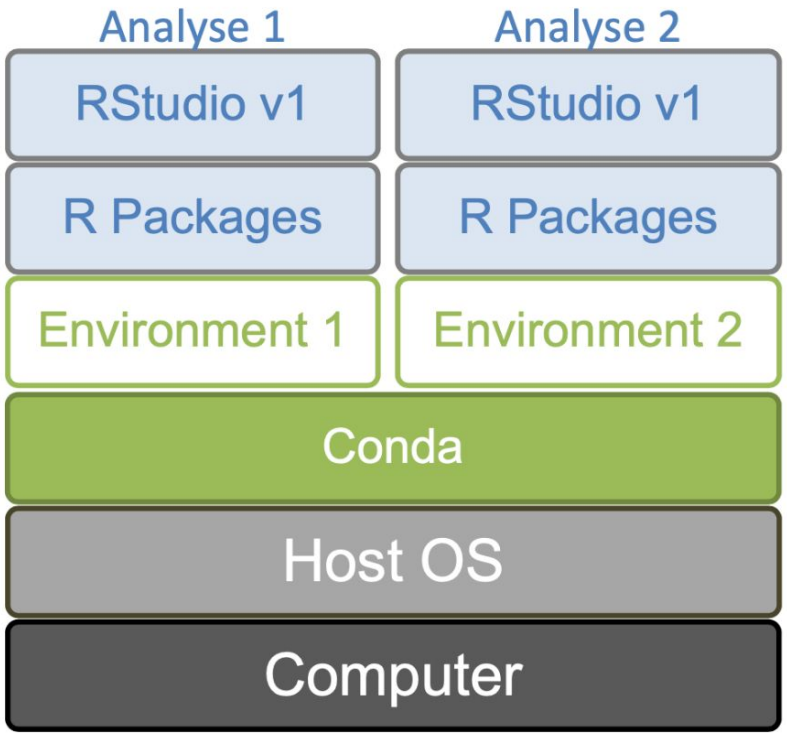
Encapsulation : managing environments



Idea : create separated environments for each application.

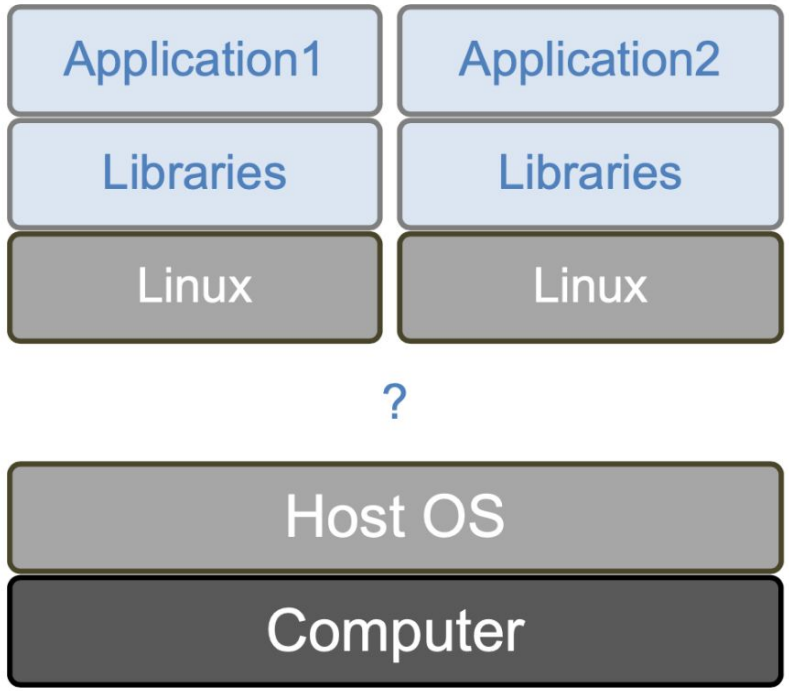
CONDA

Encapsulation : managing environments



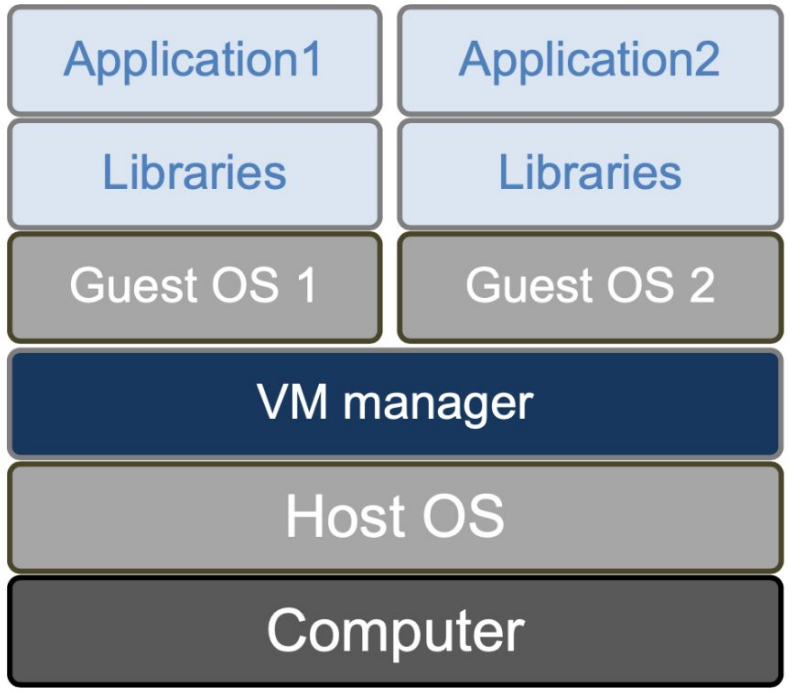
Idea : create separated environments for each application.
More versatile: create a new environment per analysis.

Encapsulation : hardware virtualization



But what if we want to install a software from a different OS?

Encapsulation : hardware virtualization

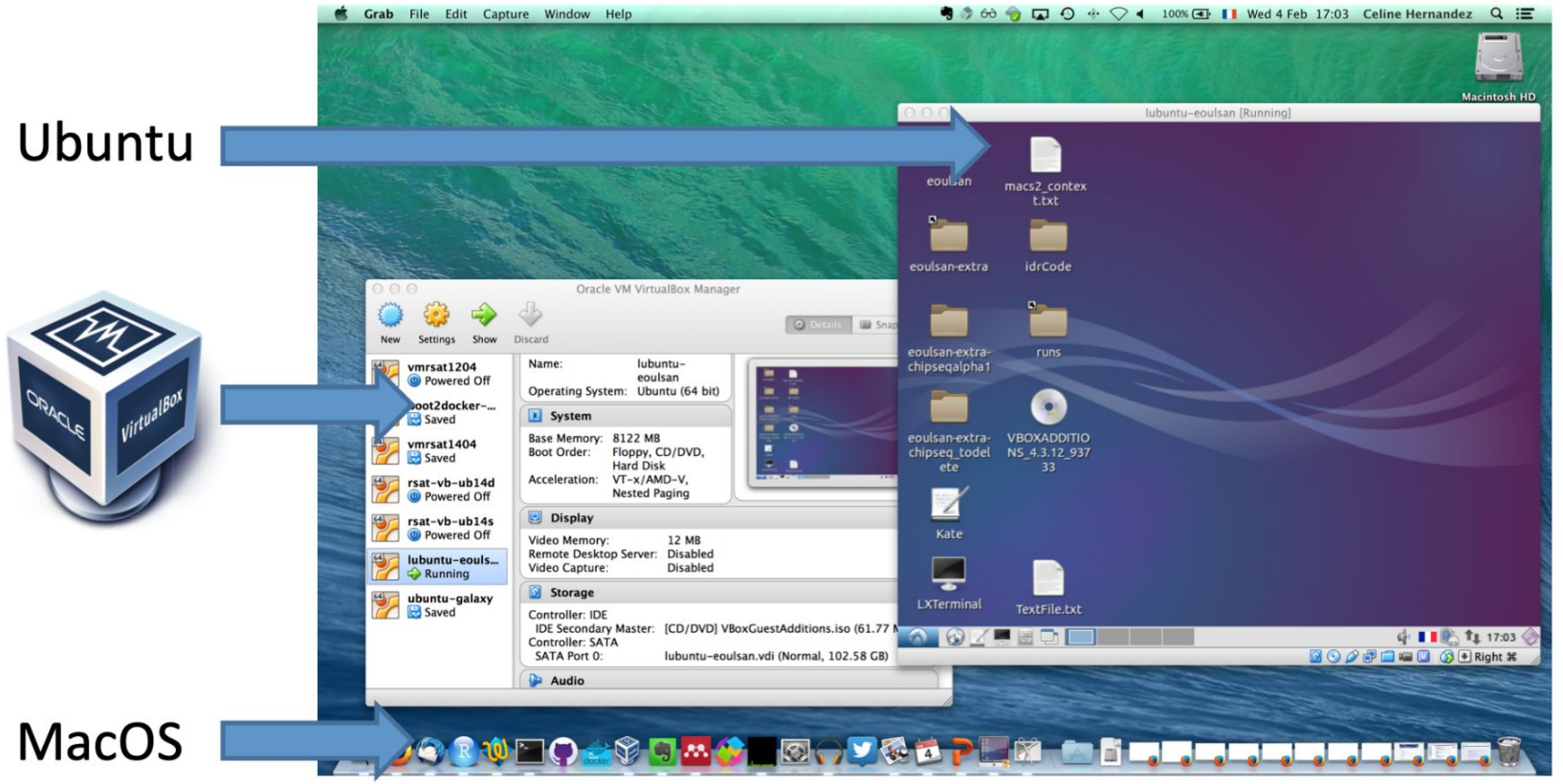


Idea: use virtual machines

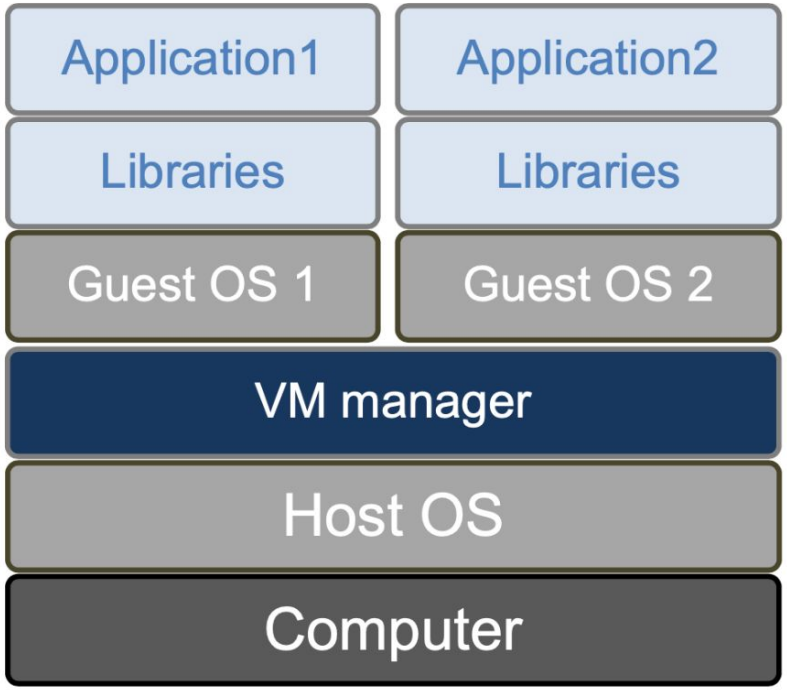
Pros:

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

Encapsulation : hardware virtualization



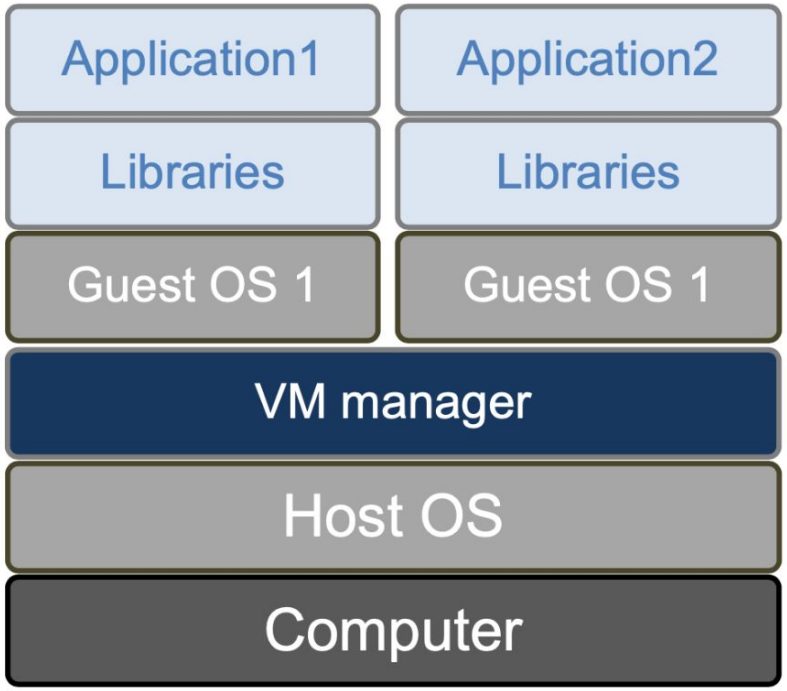
Encapsulation : hardware virtualization



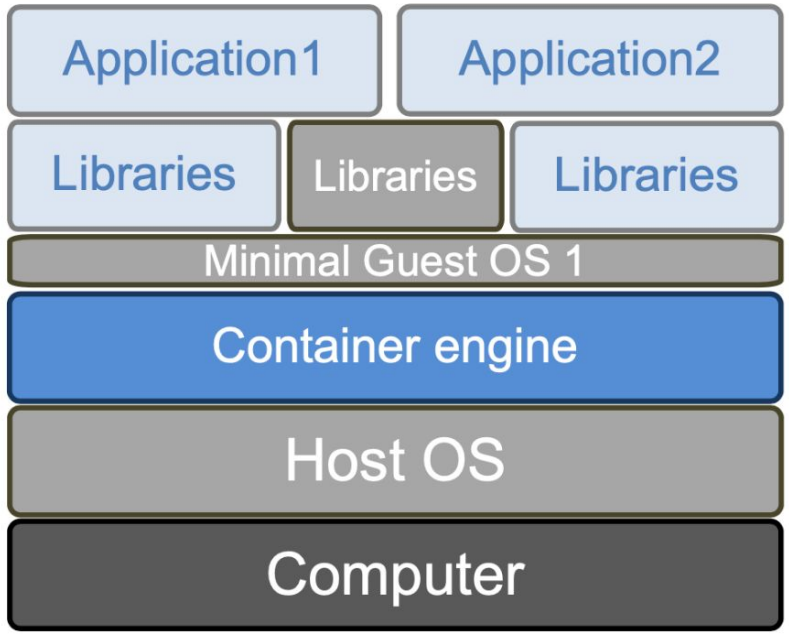
Idea: use virtual machines
Pros: transferable independent environments
Cons:

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

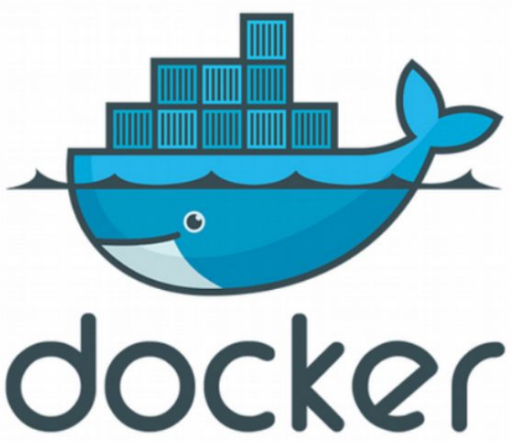
Encapsulation : OS virtualization



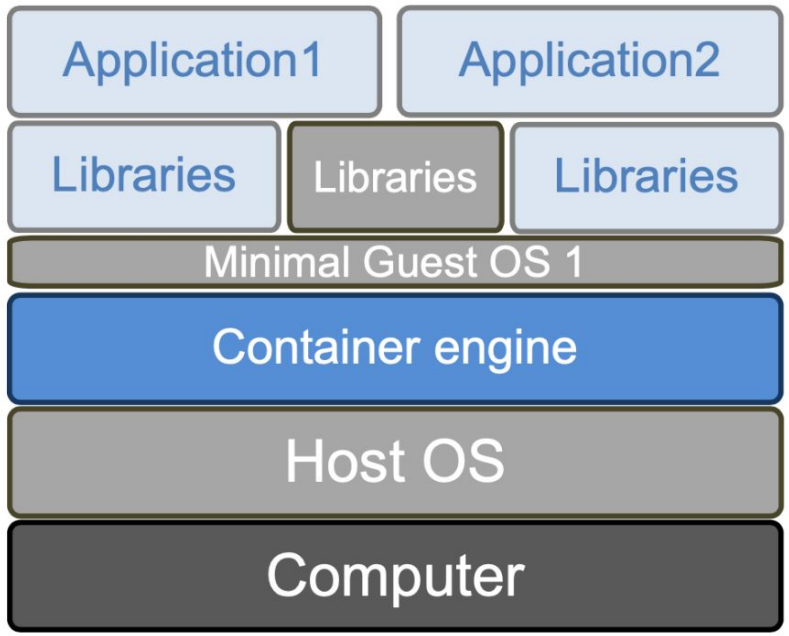
Encapsulation : OS virtualization



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



Encapsulation : OS virtualization

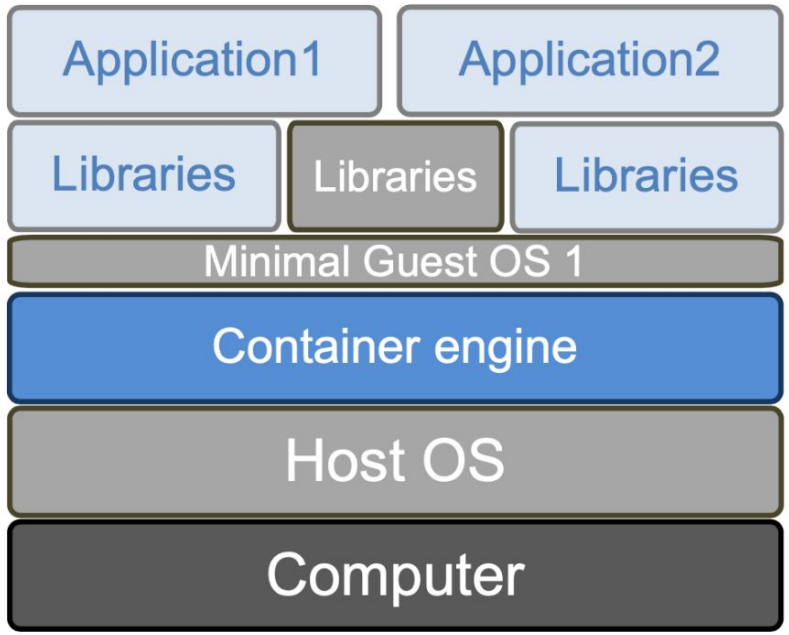


OS virtualisation vs hardware virtualisation

Pros:

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

Encapsulation : OS virtualization



Cons:

- Singularity to use images on a cluster
- 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.

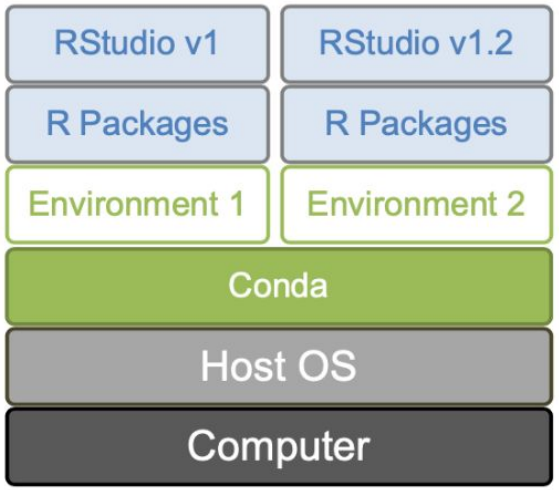
What are the new container image retention limits?

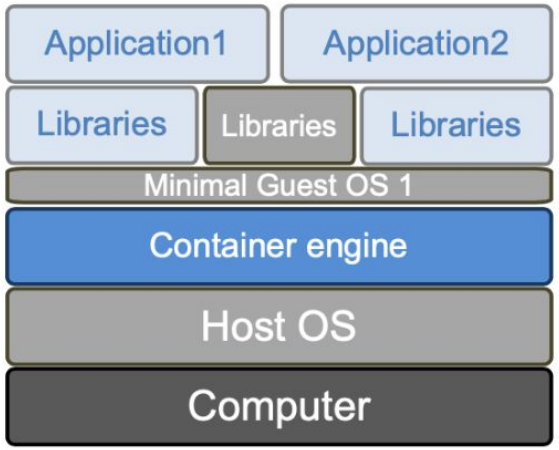
Docker is introducing a container image retention policy which will be enforced starting November 1, 2020. The container image retention policy will apply to the following plans:

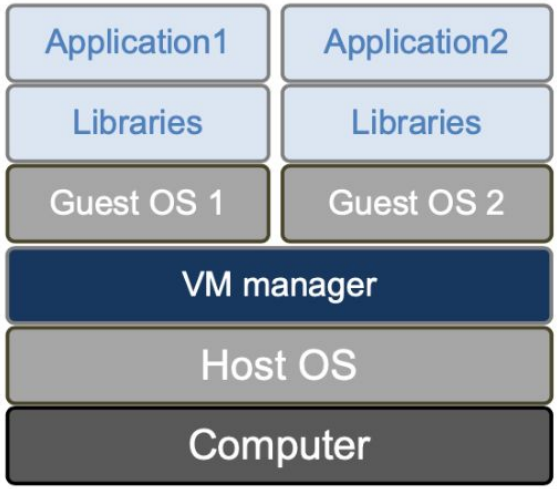
- Free plans will have a 6 month image retention limit
- Pro and Team plans will have unlimited image retention

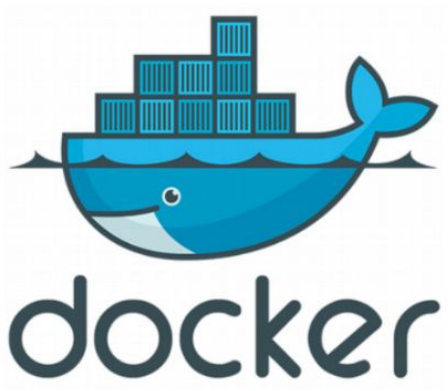
https://www.docker.com/pricing/retentionfaq

Encapsulation

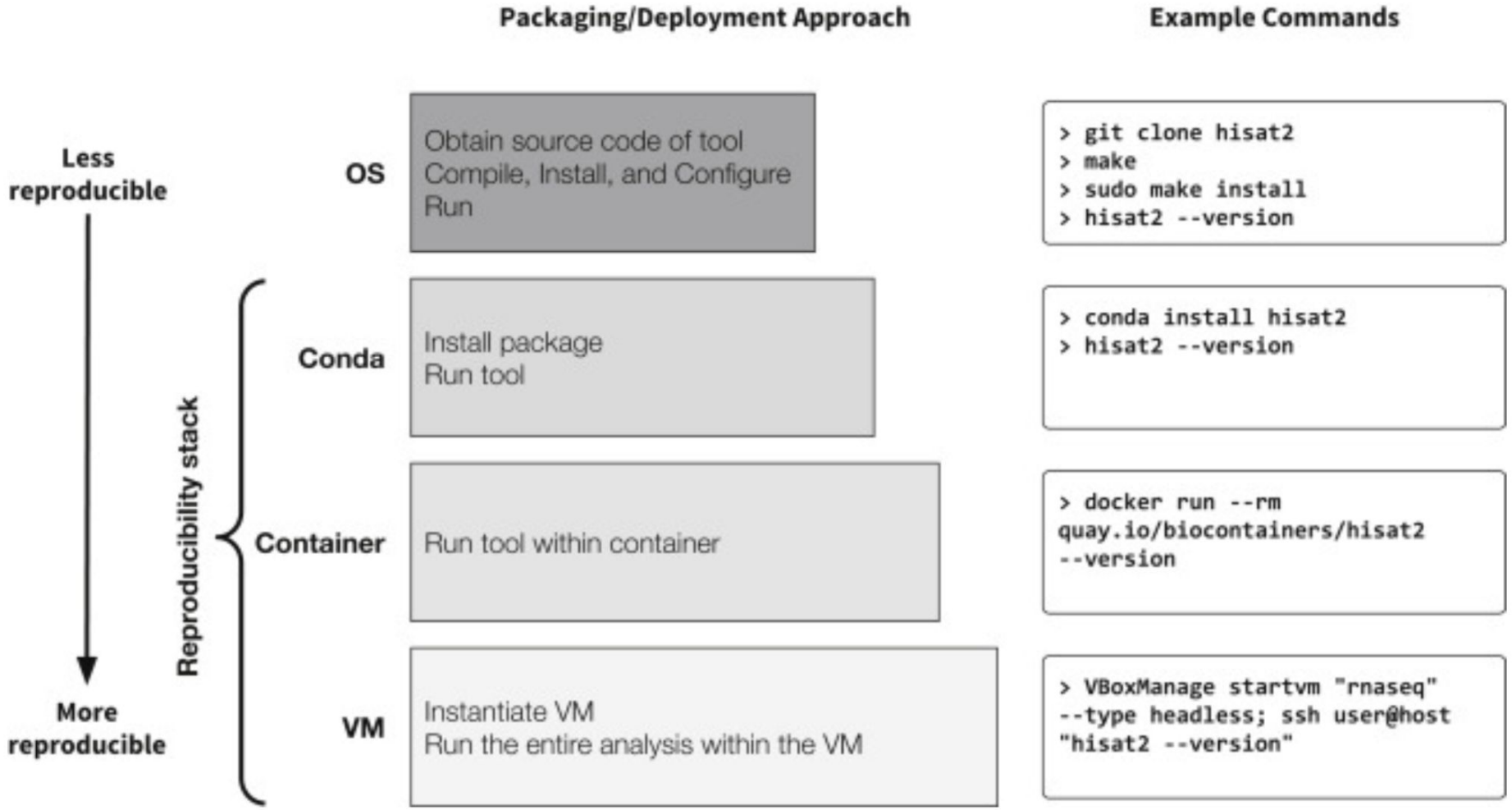




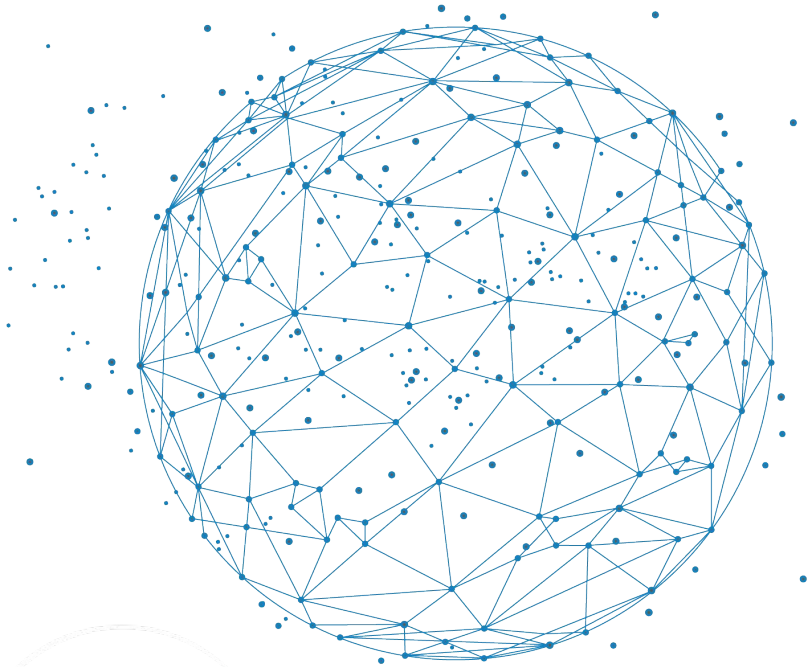




Encapsulation and reproducibility stack



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



Docker

I2BC

IFB
INSTITUT FRANÇAIS DE BIOINFORMATIQUE

elixir FRANCE

cnrs INRAE Inserm cea INVESTIR L'AVENIR

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?

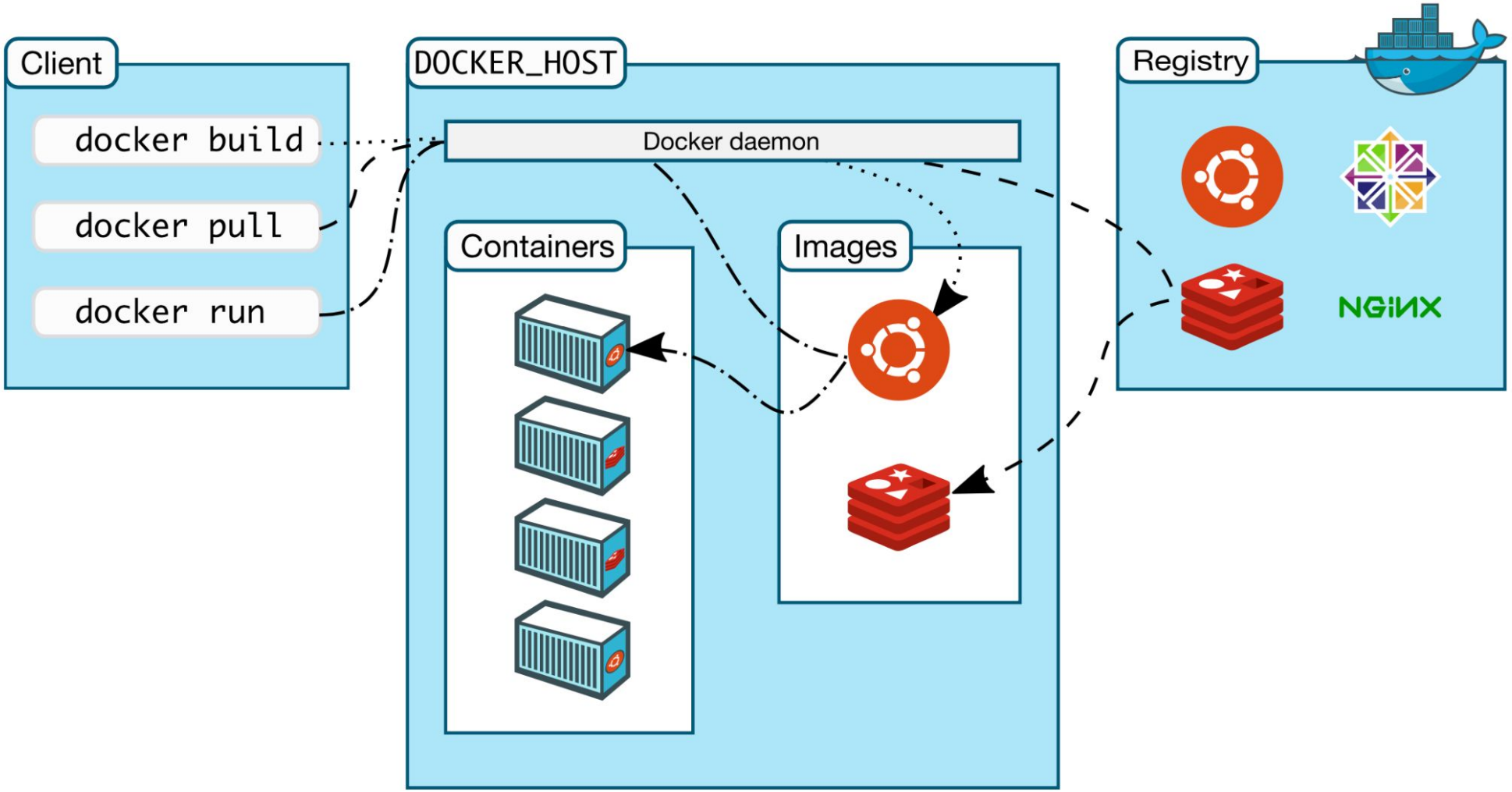
Image

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

Container

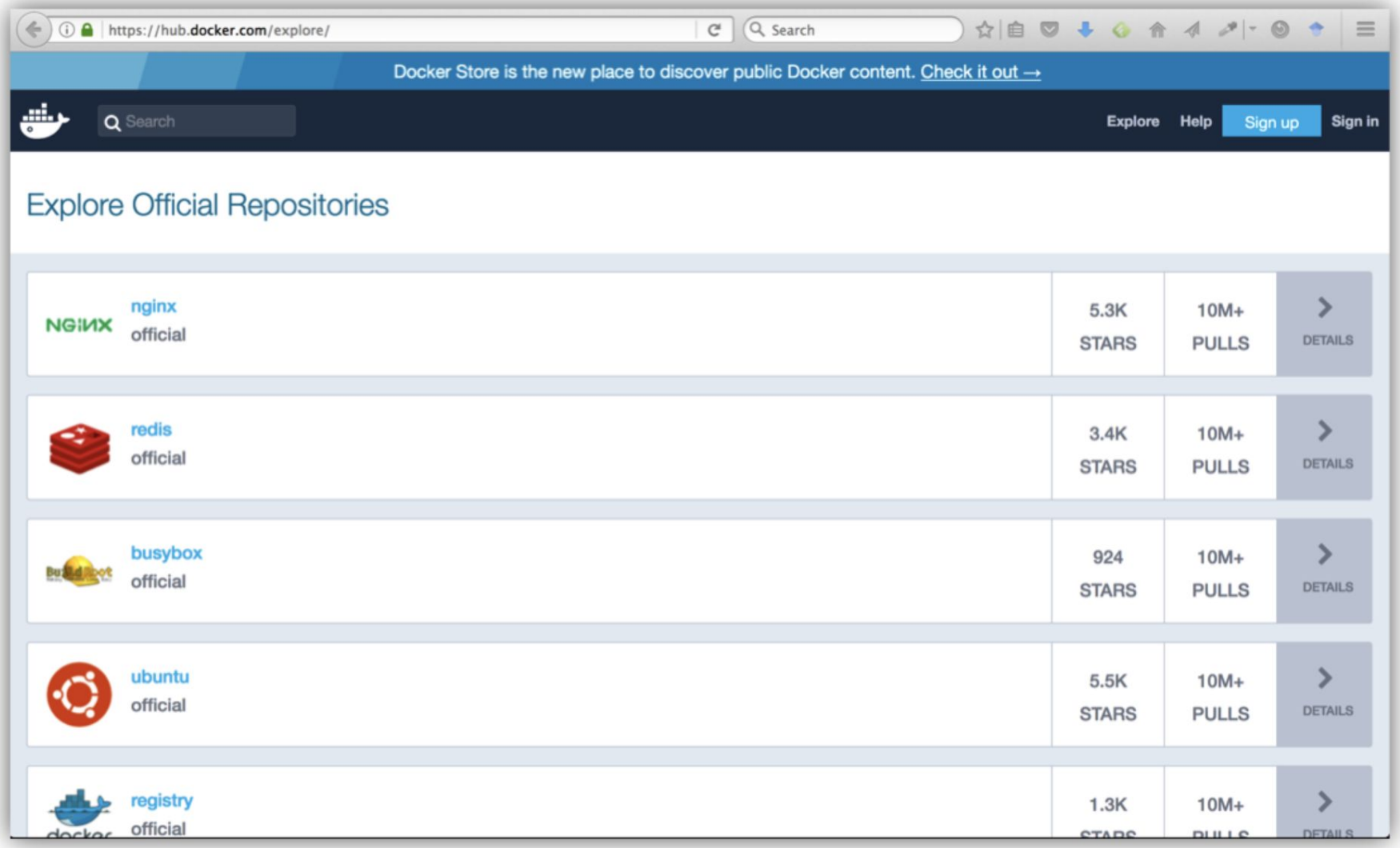
- "Active image"
- Can be modified (interactive)
- Can be turned into an image
- One image, many containers

What is Docker?



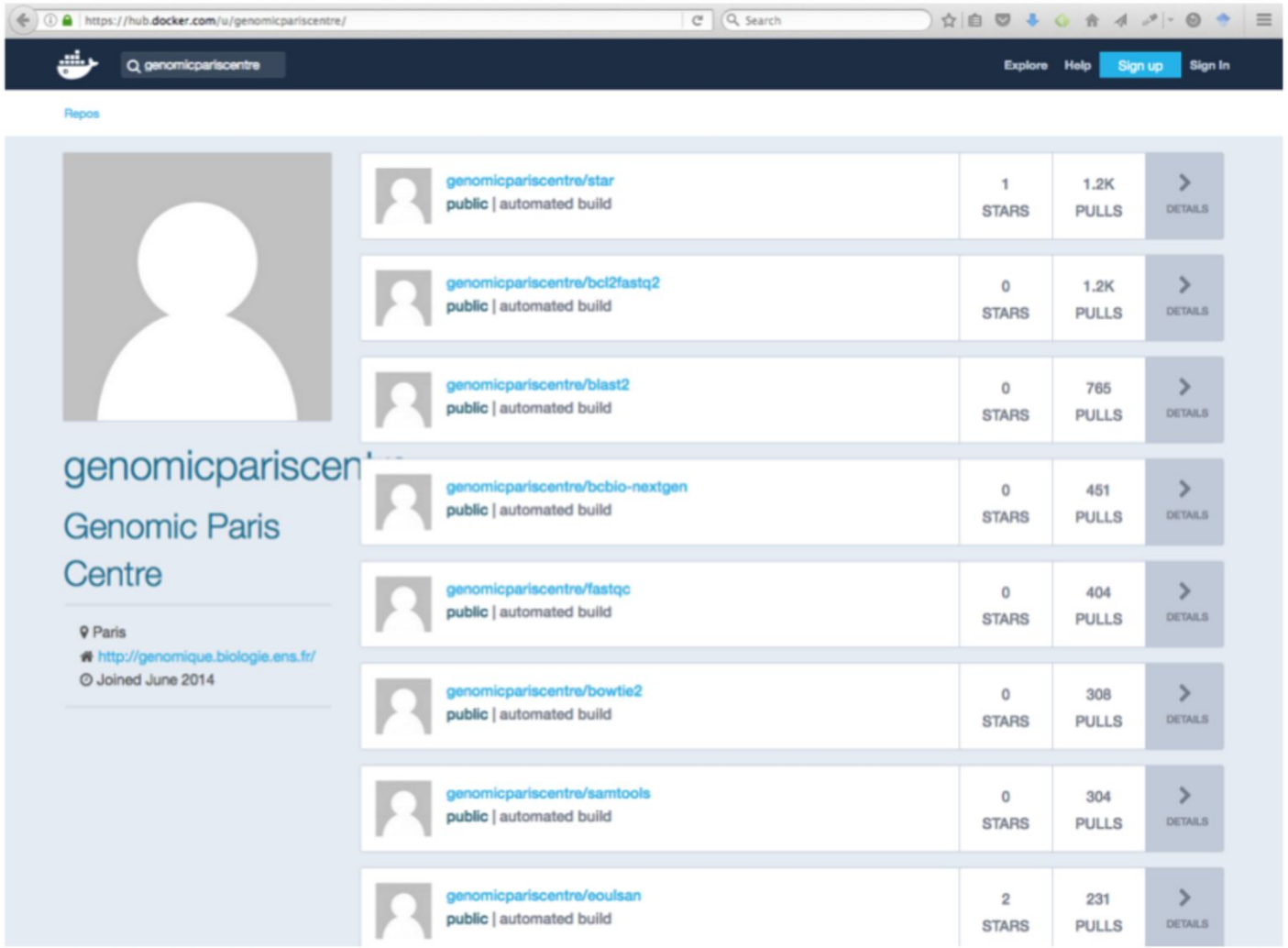
(https://docs.docker.com/get-started/overview/)

What is Docker?



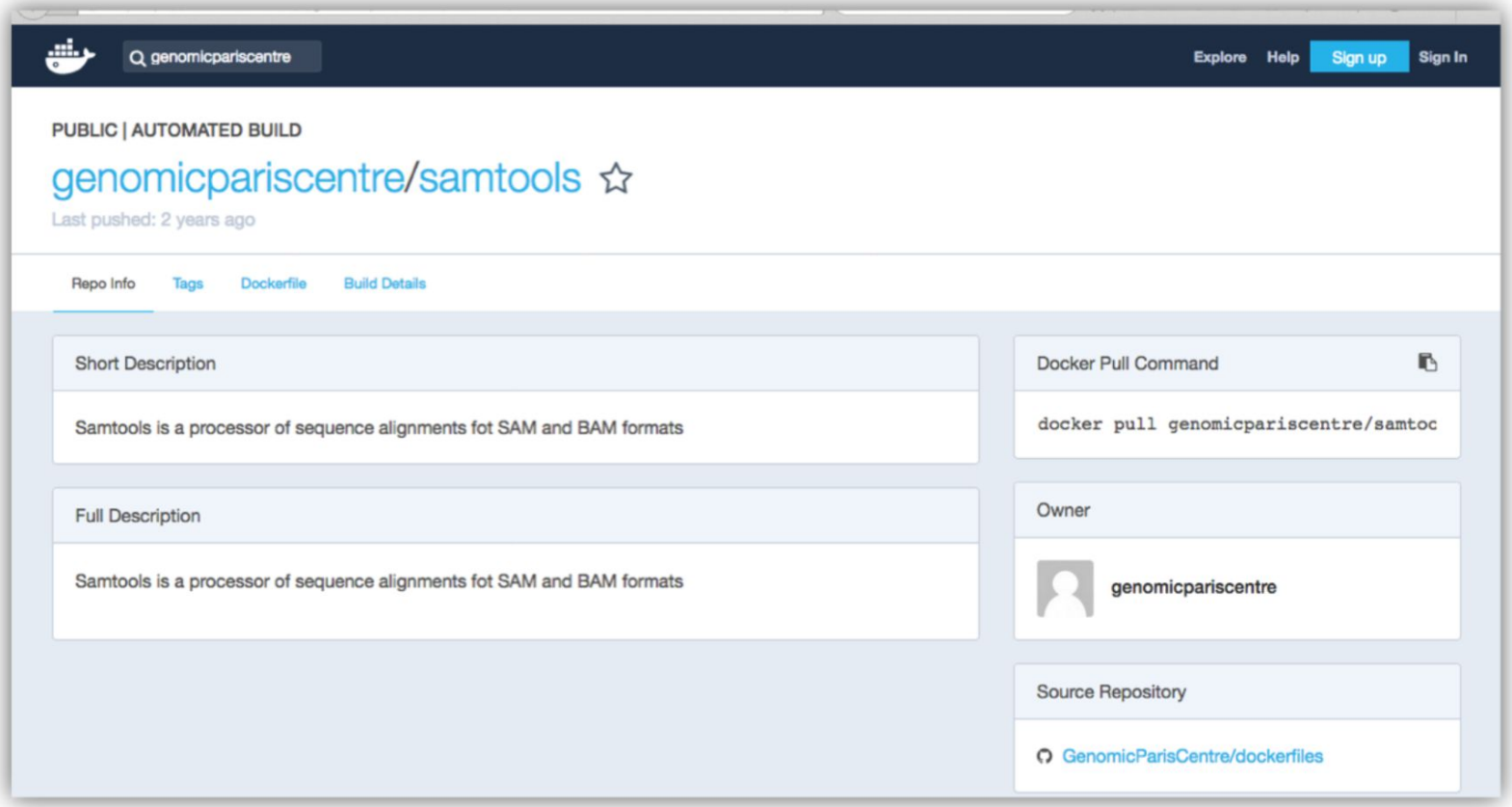
(https://hub.docker.com/)

Usermade images (1/2)



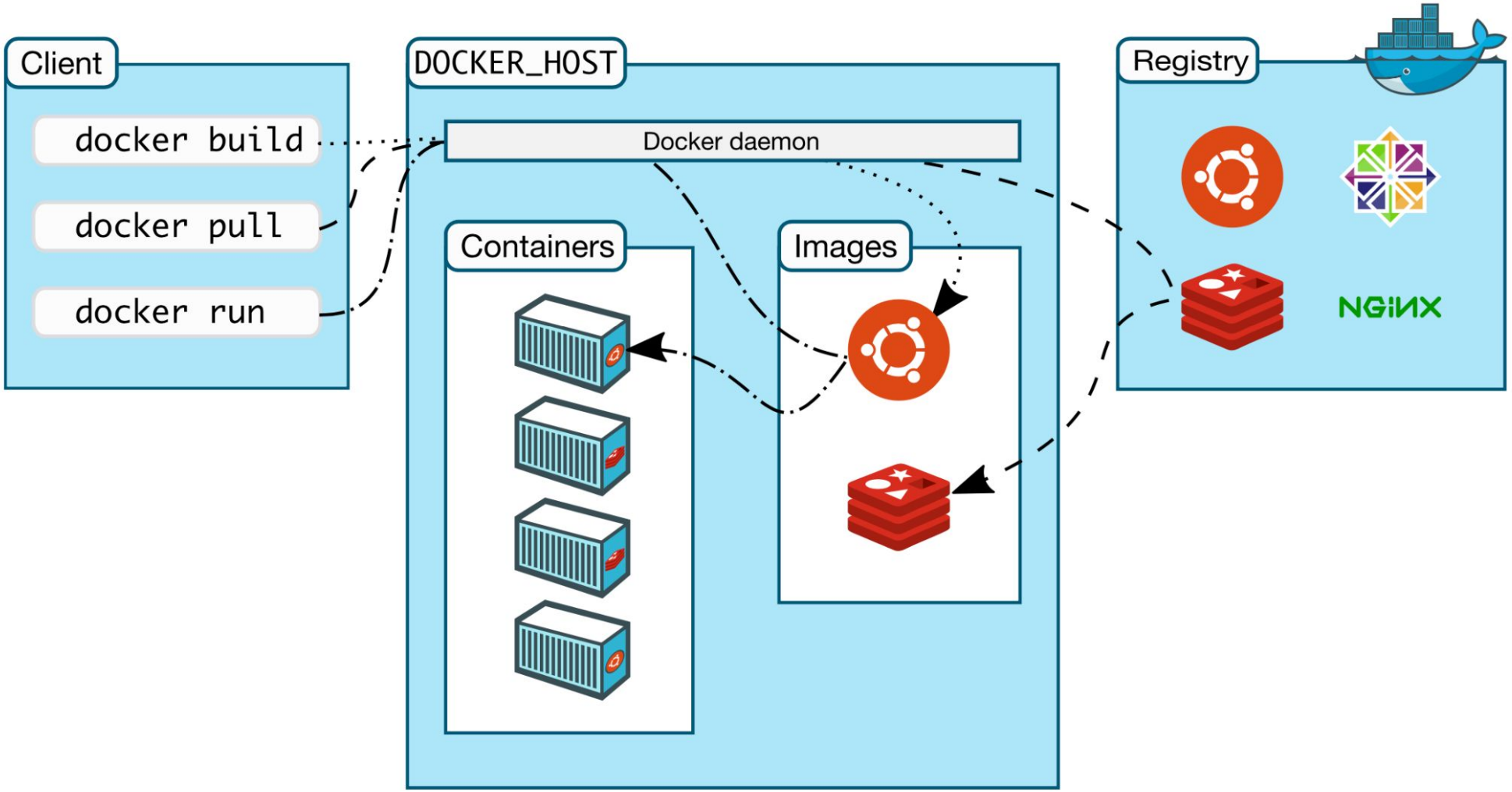
(https://hub.docker.com/u/genomicpariscentre/)

Usermade images (2/2)
Be critical!



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

What is Docker?



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

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.)