

FAIR_bioinfo for bioinformaticians

Introduction to the tools of reproducibility in bioinformatics

C. Hernandez¹ T. Denecker¹ J.Sellier² C. Toffano-Nioche¹

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

²Institut 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

A screenshot of a terminal window on an Ubuntu system. The command entered is "sudo apt-get install firefox". The output shows the package being installed: "Reading package lists... Done", "Building dependency tree", "Reading state information... Done", "Suggested packages: fonts-noto", and "The following packages will be upgraded: firefox". It also shows the download progress: "Get: http://security.ubuntu.com/ubuntu/ trusty-security/main firefox 1386 44.8.1~ubutu14.04.1 [42.0 MB]" and "75.0 kB/s 9min 12s".

Encapsulation



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



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

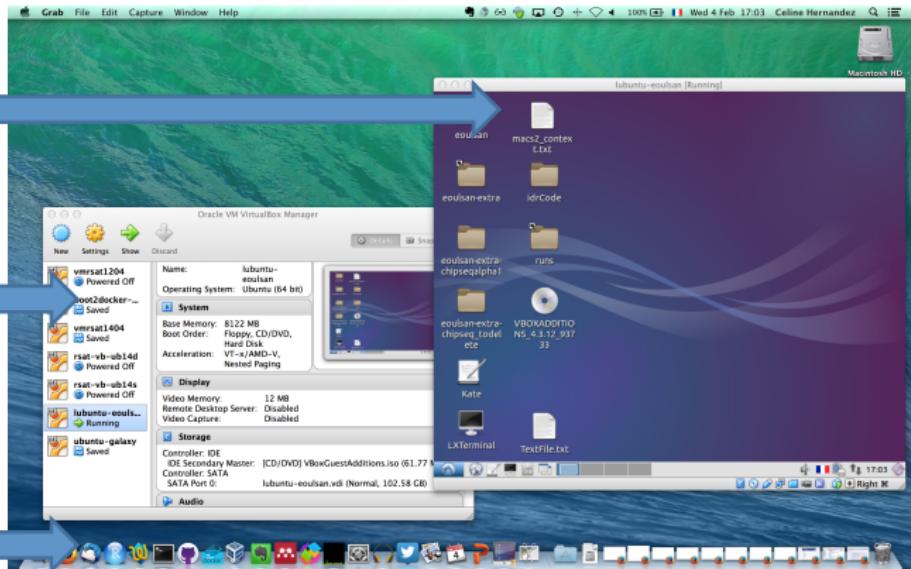


Idea: use virtual machines
Pros:

- 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:

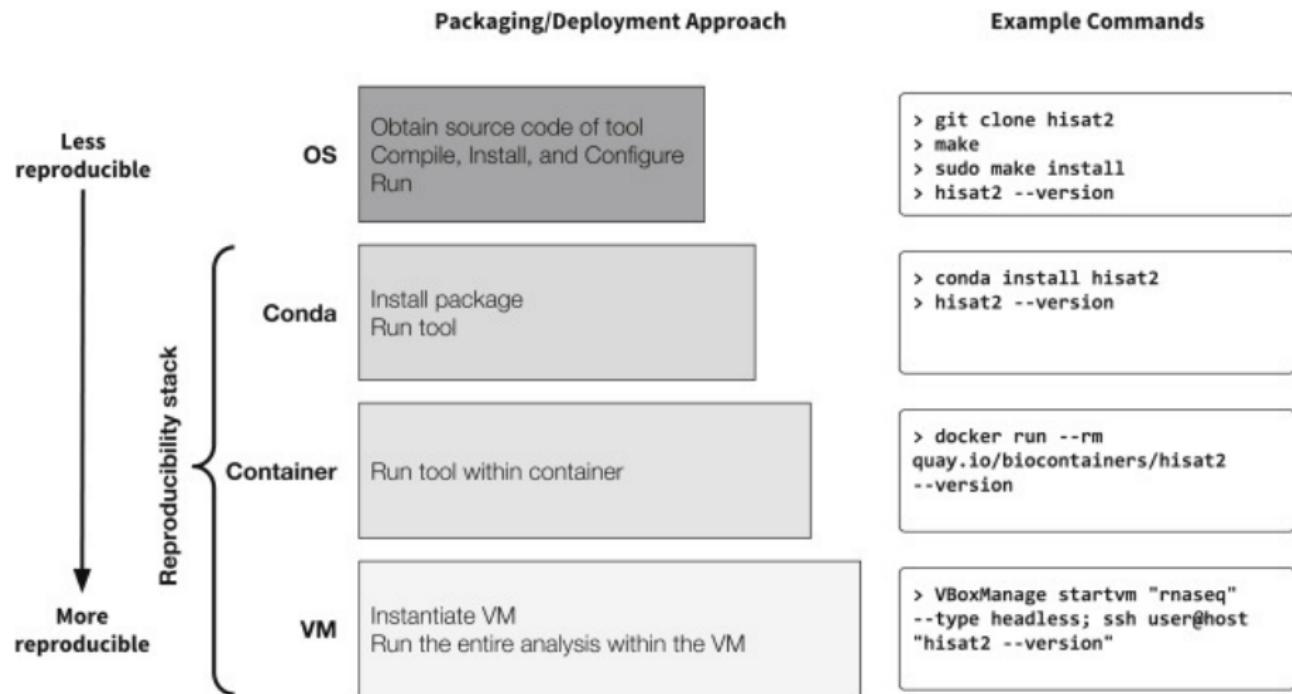
- Redundancy between VMs
- Heavy to set up
- No automation

Encapsulation : OS virtualisation



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



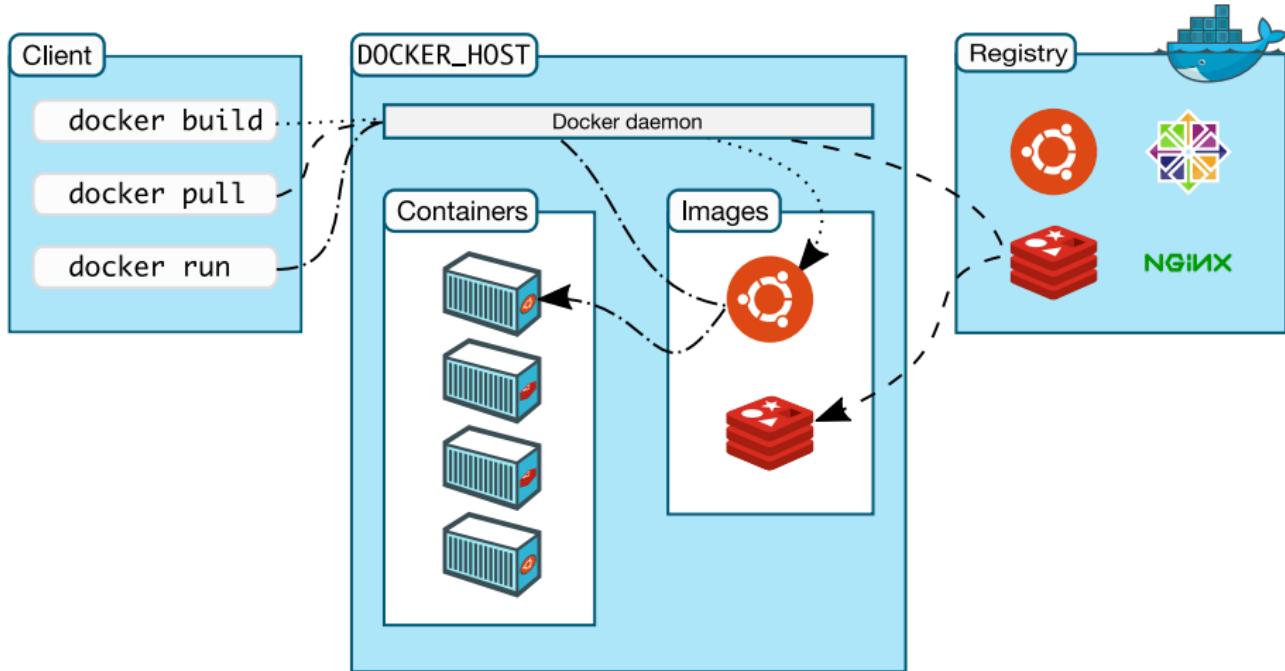
Container



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

- "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

The screenshot shows the DockerHub homepage with the URL <https://hub.docker.com/explore/> in the address bar. A banner at the top says "Docker Store is the new place to discover public Docker content. Check it out →". Below the banner, there's a search bar with a ship icon and a "Search" button. On the right, there are links for "Explore", "Help", "Sign up", and "Sign in". The main content area is titled "Explore Official Repositories". It lists five official Docker repositories:

Repository	Owner	Stars	Pulls	Details
nginx	nginx	5.3K	10M+	DETAILS
redis	redis	3.4K	10M+	DETAILS
busybox	busybox	924	10M+	DETAILS
ubuntu	ubuntu	5.5K	10M+	DETAILS
registry	docker	1.3K	10M+	DETAILS

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



What is Docker?

Usermade images (1/2)

The screenshot shows the Docker Hub user profile for `genomicpariscentre`. On the left, there's a large placeholder image for a profile picture, followed by the user's name, `genomicpariscentre`, and the full name, `Genomic Paris Centre`. Below this, there are links to the user's location (`Paris`), website (`http://genomique.biologie.ens.fr`), and the date they joined (`Joined June 2014`). The main area is titled "Repos" and lists eight Docker images:

Image Name	Description	Stars	Pulls	Actions
<code>genomicpariscentre/star</code>	public automated build	1	1.2K	DETAILS
<code>genomicpariscentre/bcl2fastq</code>	public automated build	0	1.2K	DETAILS
<code>genomicpariscentre/blast2</code>	public automated build	0	765	DETAILS
<code>genomicpariscentre/bcbio-nextgen</code>	public automated build	0	451	DETAILS
<code>genomicpariscentre/fastqc</code>	public automated build	0	404	DETAILS
<code>genomicpariscentre/bowtie2</code>	public automated build	0	308	DETAILS
<code>genomicpariscentre/samtools</code>	public automated build	0	304	DETAILS
<code>genomicpariscentre/eulcaan</code>	public automated build	2	231	DETAILS

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



What is Docker?

Usermade images (2/2)

Be critical!

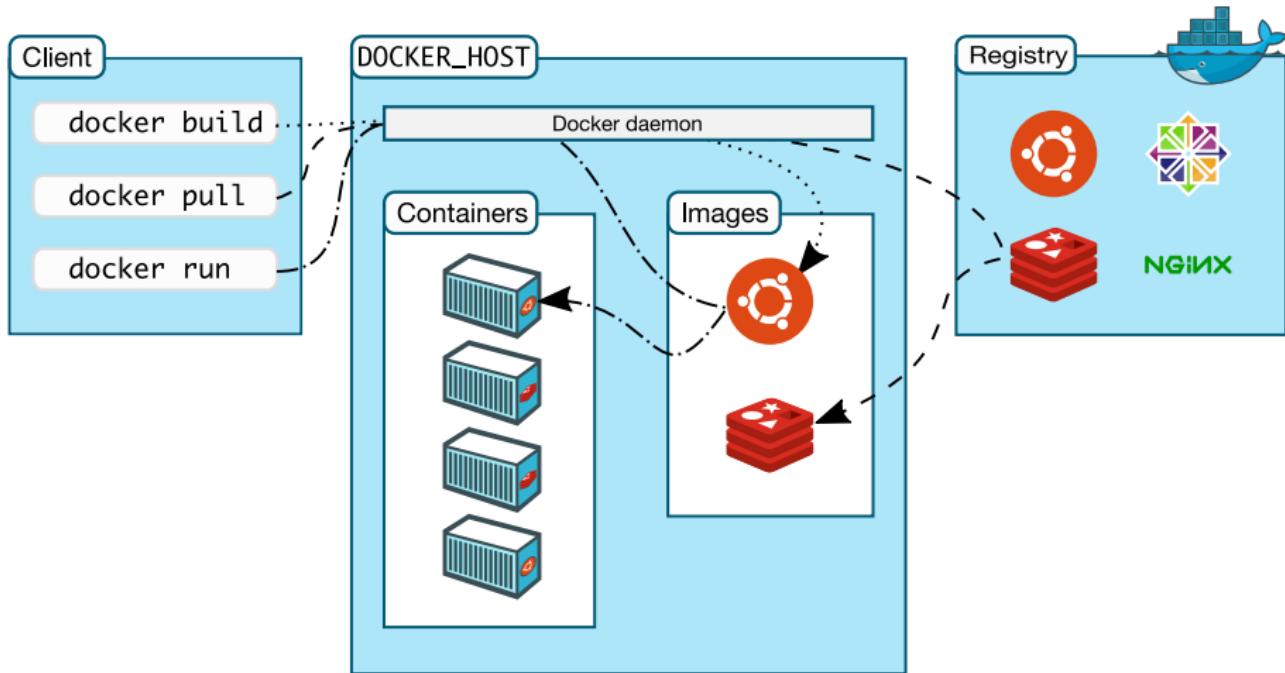
The screenshot shows the Docker Hub interface for the repository `genomicpariscentre/samtools`. The page title is "genomicpariscentre/samtools" with a star icon indicating it's unstarred. Below the title, it says "Last pushed: 2 years ago". There are four tabs at the top: "Repo Info" (selected), "Tags", "Dockerfile", and "Build Details".

Repo Info	Tags	Dockerfile	Build Details
<p>Short Description</p> <p>Samtools is a processor of sequence alignments for SAM and BAM formats</p>	<p>Docker Pull Command</p> <pre>docker pull genomicpariscentre/samtools</pre>		
<p>Full Description</p> <p>Samtools is a processor of sequence alignments for SAM and BAM formats</p>	<p>Owner</p>  genomicpariscentre		
	<p>Source Repository</p> GenomicParisCentre/dockerfiles		

(<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



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

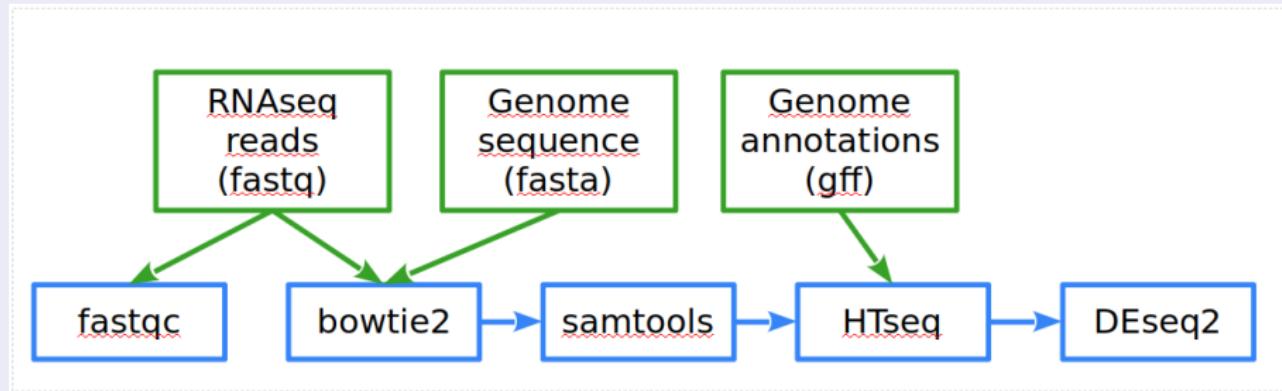
<https://www.docker.com/pricing/retentionfaq>

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