INSTITUT FRANÇAIS DE BIOINFORMATIQUE

The IFB Core Cluster Infrastructure

FAIR Bioinfo 2022

Gildas Le Corguillé & Julien Seiler
IFB Core Cluster taskforce

DOI 10.5281/zenodo.6628340

High Performance Computer

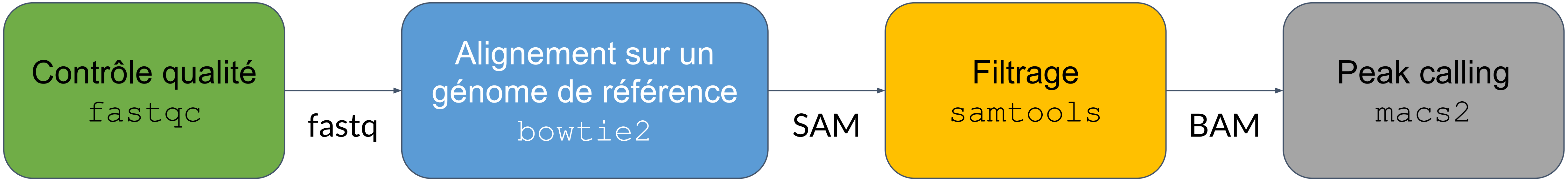
Votre ordinateur peut-il faire de la bioinformatique ?

Un ou deux microprocesseurs
Un microprocesseur est chargé de l'exécution des instructions élémentaires demandées par le logiciel

4 à 8 Go de mémoire vive (RAM)
La mémoire vive est utilisée par le microprocesseur pour traiter les données

≃ 1 To d'espace de stockage
L'espace de stockage est utilisé pour conserver de grandes quantités de données de manière plus permanente

Votre ordinateur peut-il faire de la bioinformatique ?



Contrôle qualité
`fastqc`

fastq

Alignement sur un génome de référence
`bowtie2`

SAM

Filtrage
`samtools`

BAM

Peak calling
`macs2`

1 CPU	10 CPU	1 CPU	1 CPU
8 Go RAM	1 Go RAM	1 Go RAM	8 Go RAM

L’exécution de ce workflow nécessite au minimum toutes les ressources d’un ordinateur de bureau pendant plusieurs heures et ceci seulement pour 1 seul fichier fastq.
Pour faire ce type d’analyse nous avons besoin d’ordinateurs plus puissants !

Du data center au coeur

Rue Louis de Broglie
Rue Louis de Broglie
Rue John Von Neumann
Institut du développement et des...
LIMSI

Le Data Center de l'IDRIS

Un bâtiment conçu pour accueillir des infrastructures informatiques

Du data center au coeur

Groupes froid
Pour refroidir les équipements

Du data center au coeur

Groupe électrogène
Pour garantir l’alimentation électrique

Du data center au coeur

Les armoires de l’IFB
Chaque armoire peut contenir 80 super-ordinateurs

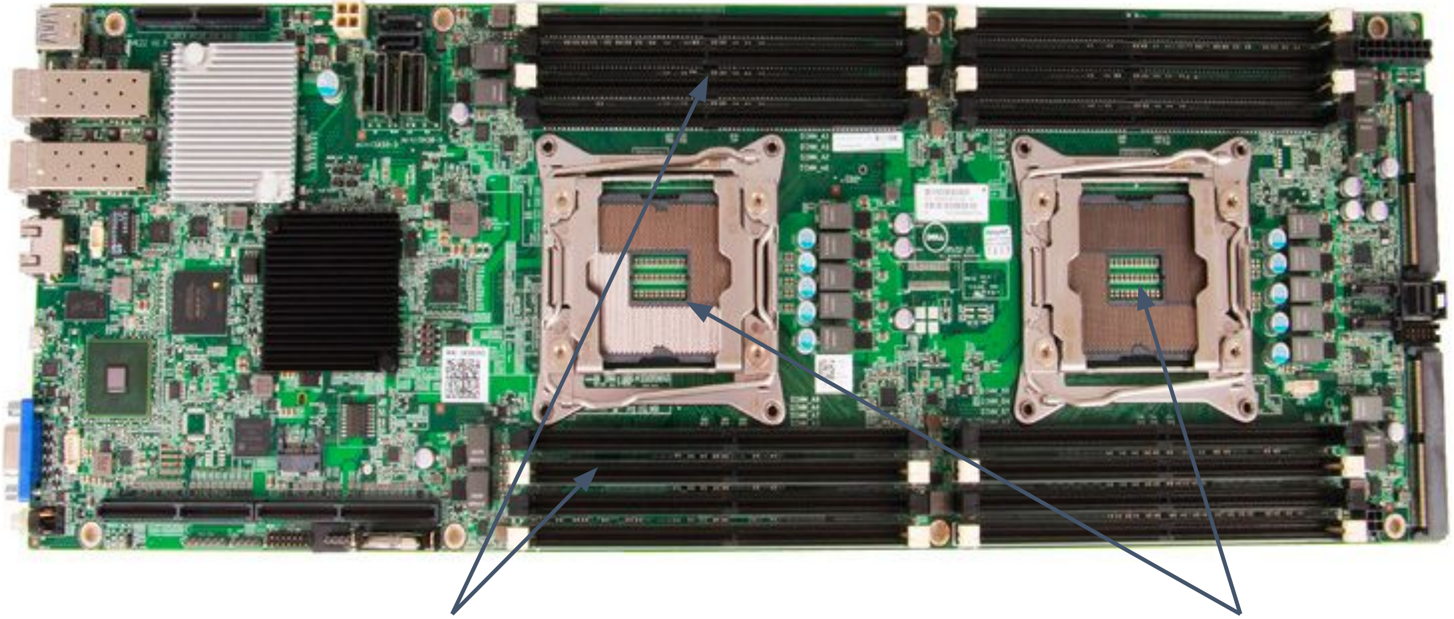
Du data center au coeur

ordinateurs de calcul

Baies de stockage

Du data center au coeur

Un ordinateur ou **noeud** de calcul



Mémoire vive

Supports processeurs

Du data center au coeur

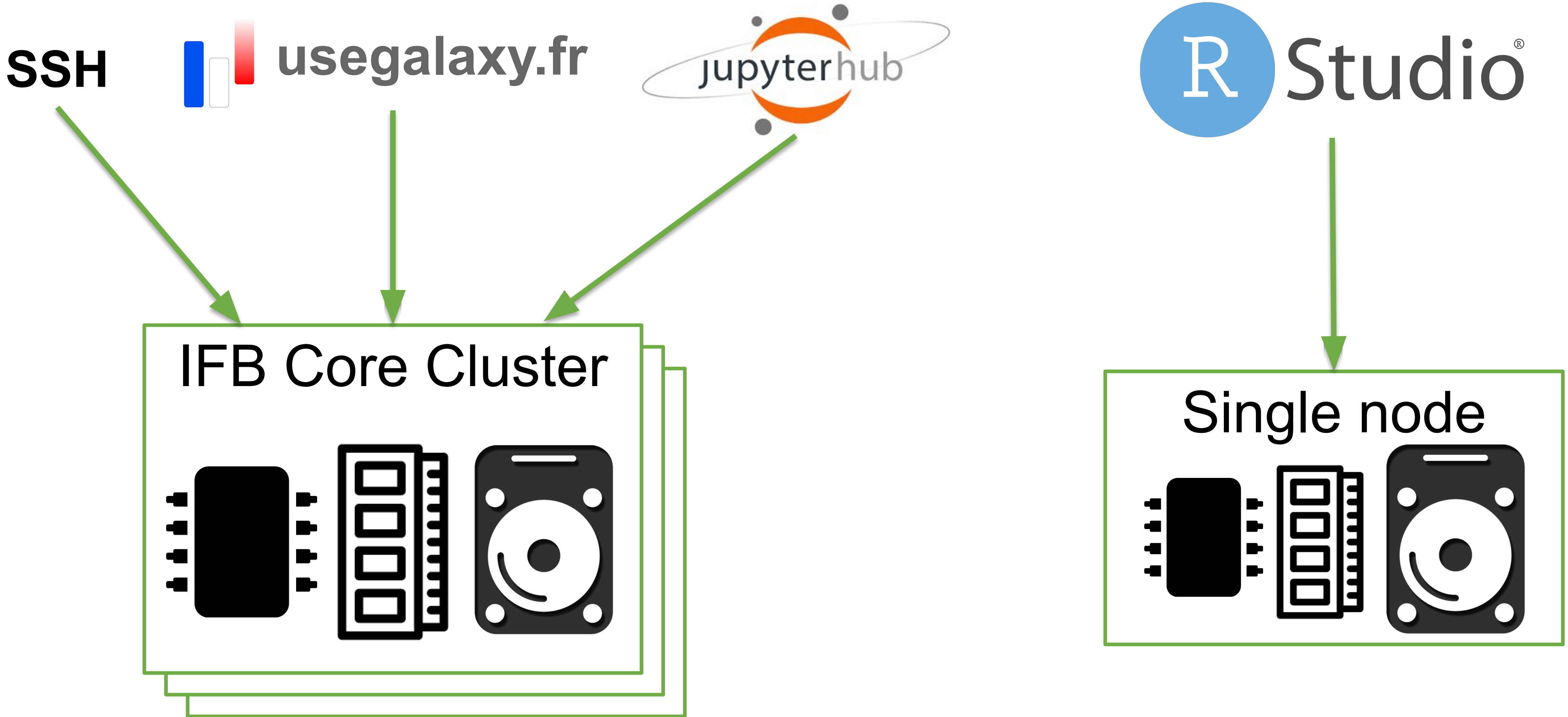
Un microprocesseur

Un microprocesseur contient plusieurs **coeurs**
Chaque coeur se comporte comme un microprocesseur unique.

La fédération de cluster de l’IFB (NNCR)

Cluster	Localisation du Data center	Coeurs	RAM (Go)	Stockage (To)
IFB Core	**IDRIS - Orsay**	**5 042**	**26 542**	**2 000**
Genotoul	Toulouse	6 128	34 304	3 000
ABiMS	Roscoff	2 608	10 600	2 500
GenOuest	Rennes	1 824	7 500	2 300
Migale	Jouy en Josas	1 084	7 000	350
BiRD	Nantes	560	4 000	500

L’infrastructure **Core Cluster** de l’**IFB**

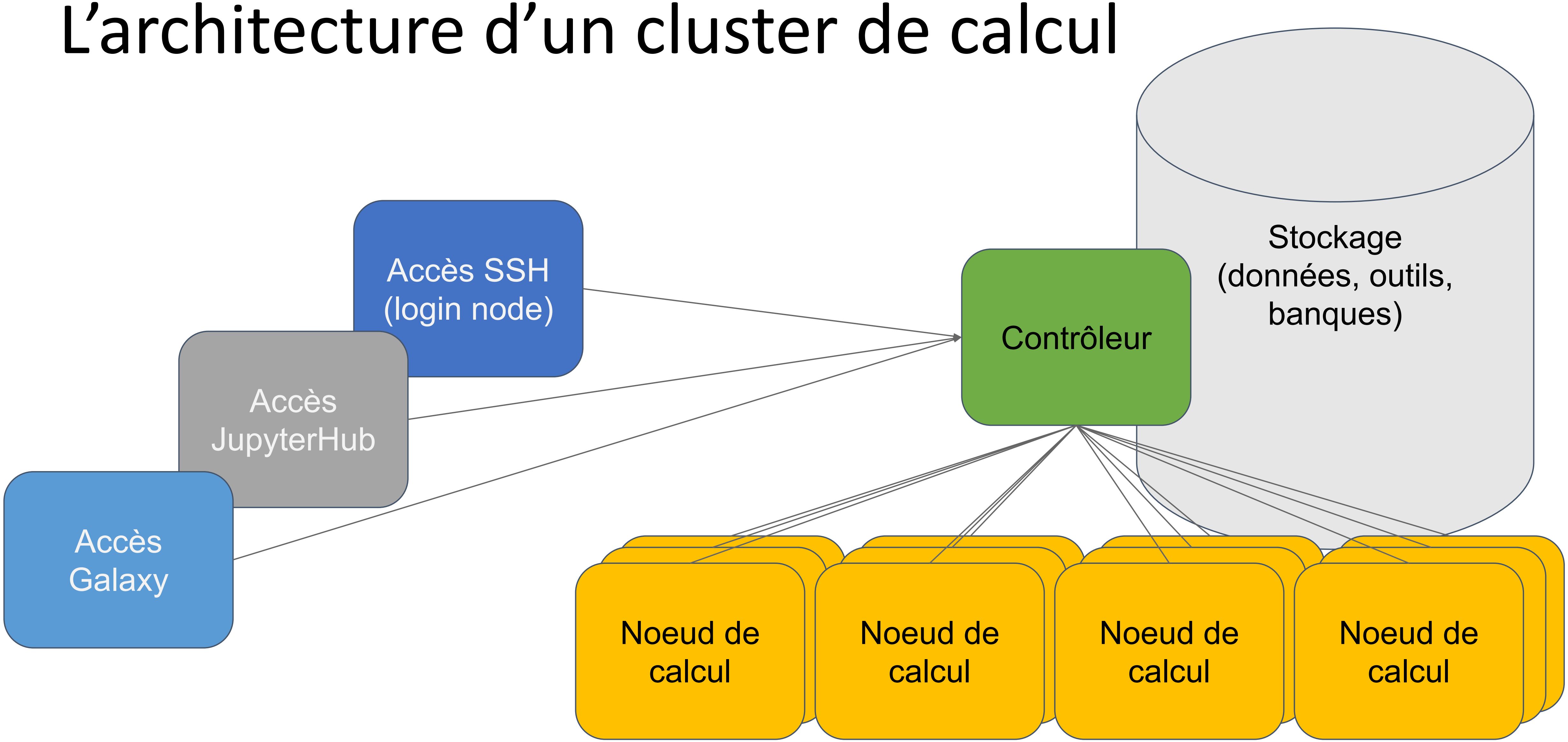


SSH

IFB Core Cluster

Single node

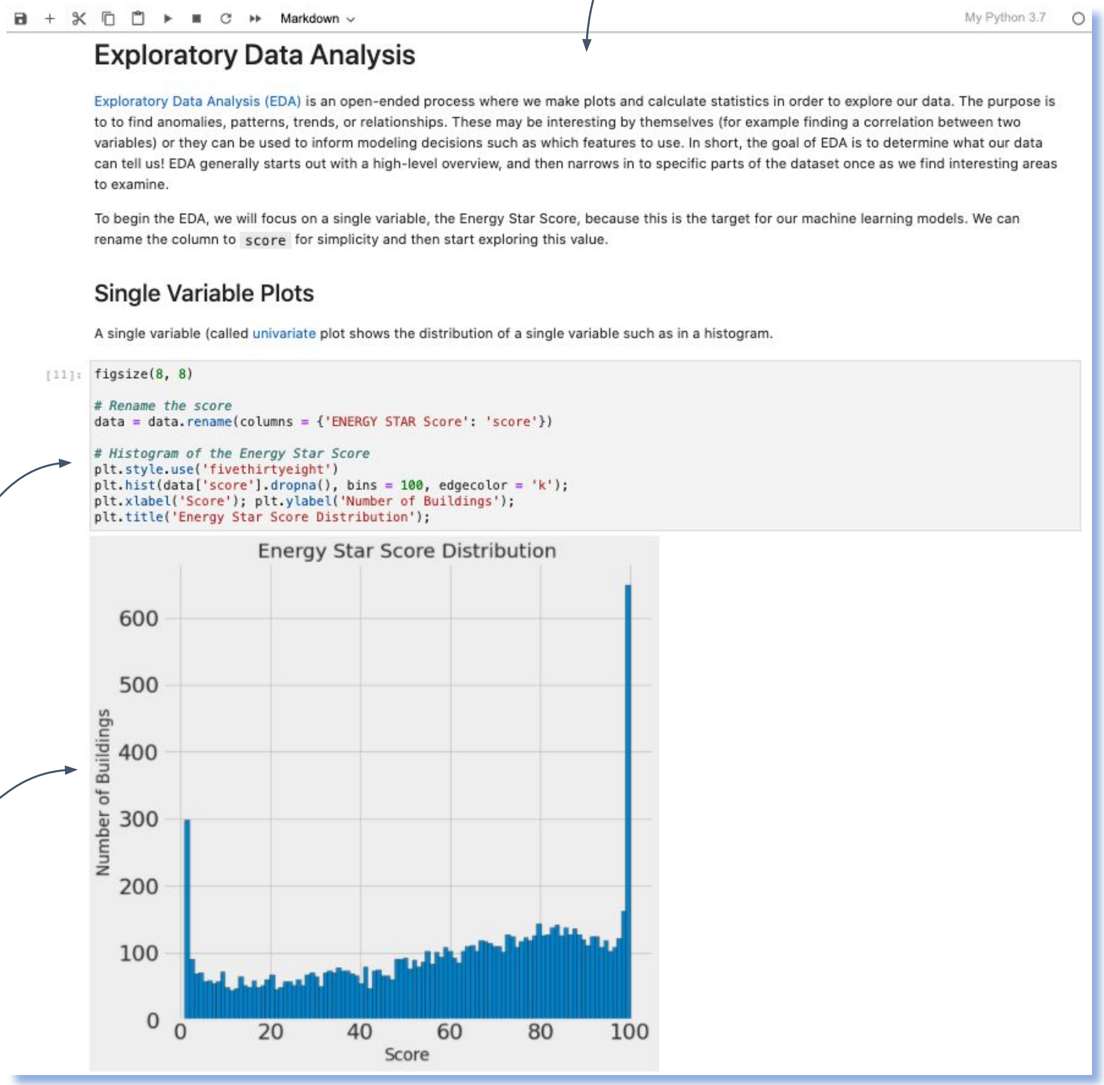
L'architecture d'un cluster de calcul



Jupyter

What is a Jupyter Notebook ?

- Special file with extension .ipynb
- Combination of **Markdown** and **code**
- Code can be executed inside the notebook
- Code Output is integrated directly in the notebook



Exploratory Data Analysis

Exploratory Data Analysis (EDA) is an open-ended process where we make plots and calculate statistics in order to explore our data. The purpose is to to find anomalies, patterns, trends, or relationships. These may be interesting by themselves (for example finding a correlation between two variables) or they can be used to inform modeling decisions such as which features to use. In short, the goal of EDA is to determine what our data can tell us! EDA generally starts out with a high-level overview, and then narrows in to specific parts of the dataset once as we find interesting areas to examine.

To begin the EDA, we will focus on a single variable, the Energy Star Score, because this is the target for our machine learning models. We can rename the column to `score` for simplicity and then start exploring this value.

Single Variable Plots

A single variable (called univariate plot shows the distribution of a single variable such as in a histogram.

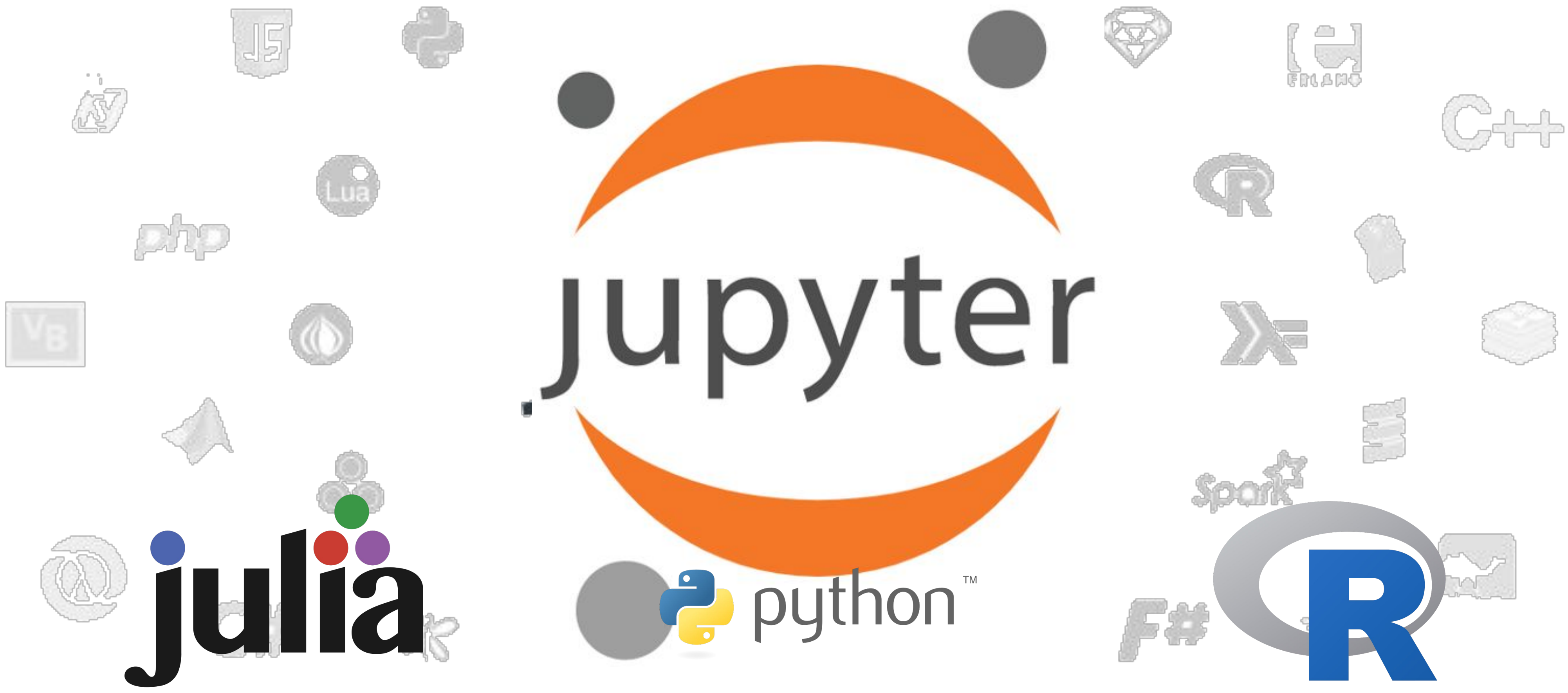
```
[11]: figsize(8, 8)

# Rename the score
data = data.rename(columns = {'ENERGY STAR Score': 'score'})

# Histogram of the Energy Star Score
plt.style.use('fivethirtyeight')
plt.hist(data['score'].dropna(), bins = 100, edgecolor = 'k');
plt.xlabel('Score'); plt.ylabel('Number of Buildings');
plt.title('Energy Star Score Distribution');
```

Markdown cell

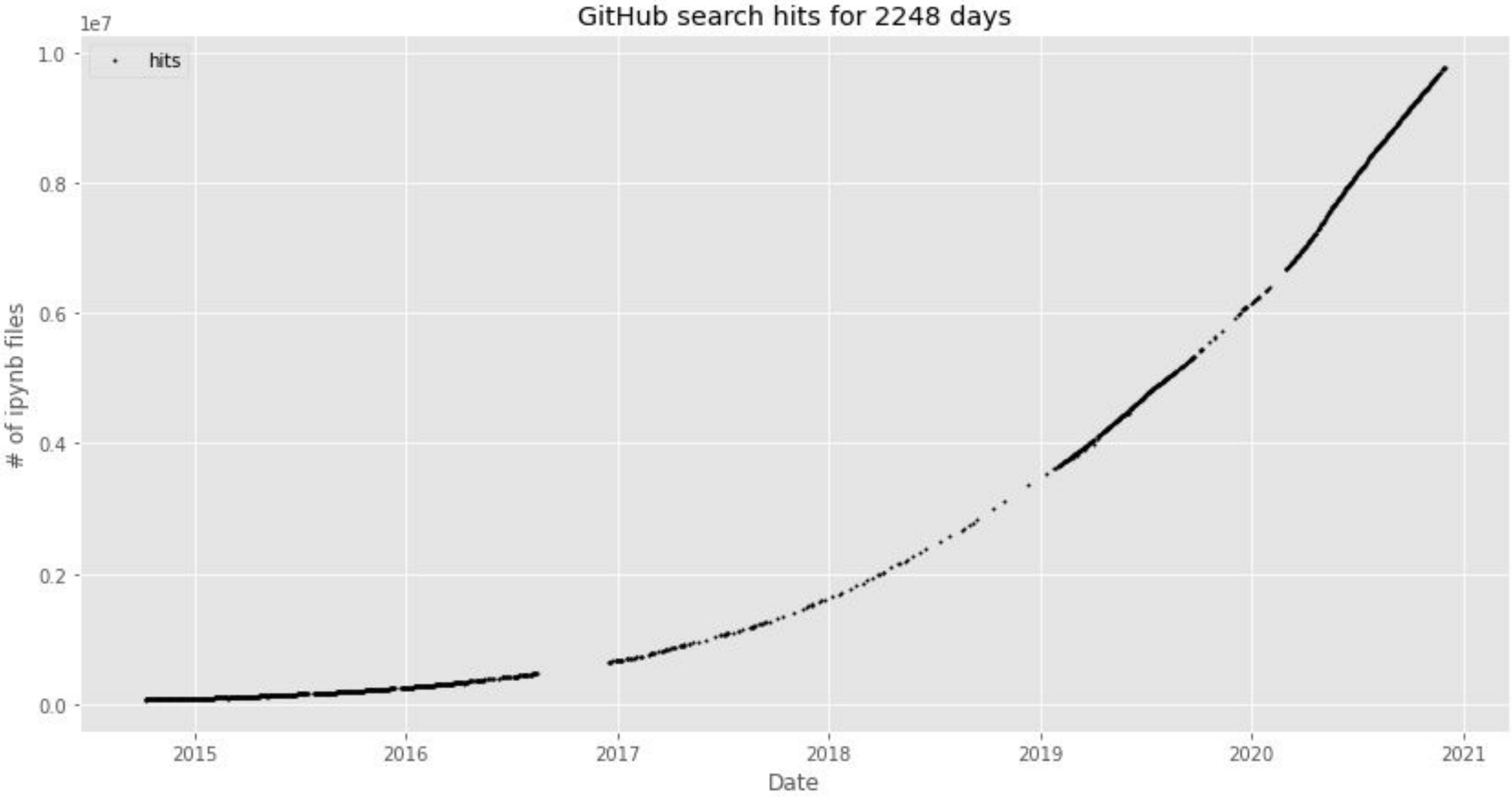
Code cell

Code output

What is a Jupyter Notebook ?



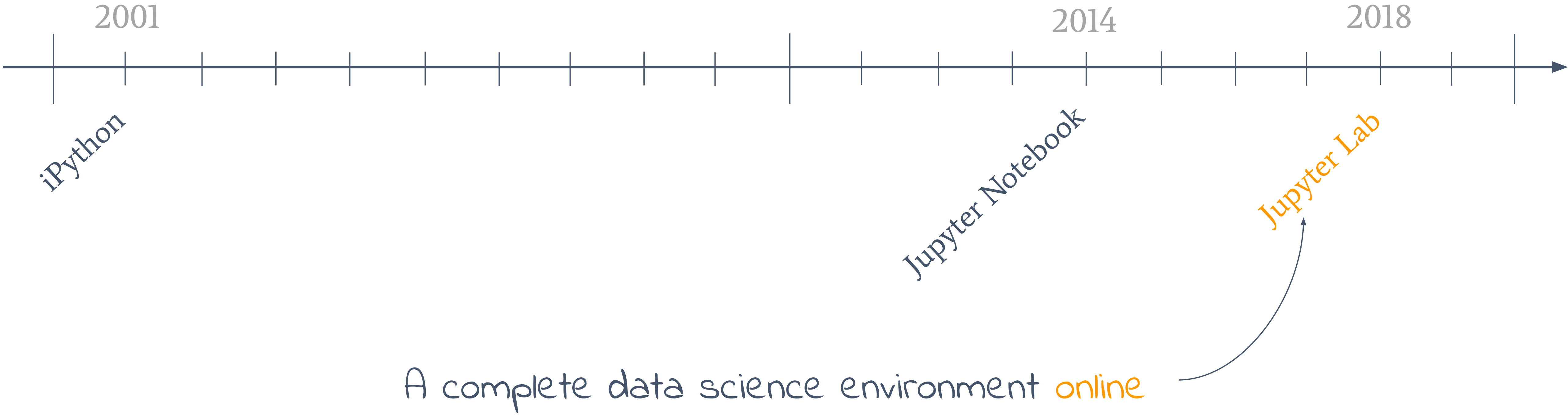
Notebooks are popular



Source : Nbviewer - https://nbviewer.jupyter.org/github/parente/nbestimate/blob/master/estimate.ipynb

What do I need to work on notebooks ?

A notebook environment or notebook server



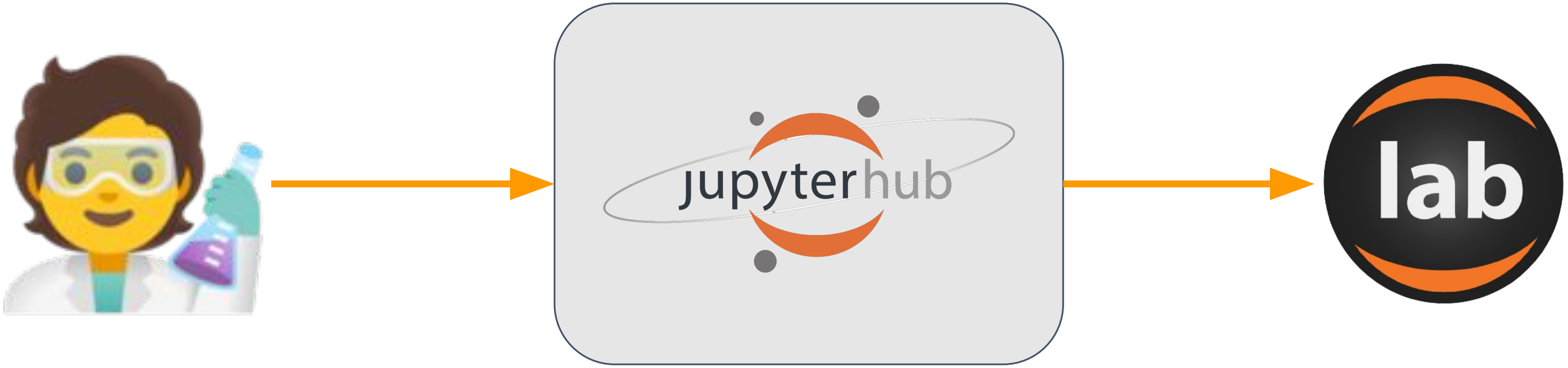
Why should I care about notebooks ?

- Notebook lets you analyze data and write reports in one place
- It supports real time data visualization
- You can easily include interactive section in a notebook
- In line with reproducible science principle

Jupyter Notebook is the Lab Notebook for Data science

What is JupyterHub

JupyterHub is a web application that let you spawn JupyterLab servers on a cluster or cloud infrastructure.



authentication
spawner
proxy

IFB core cluster

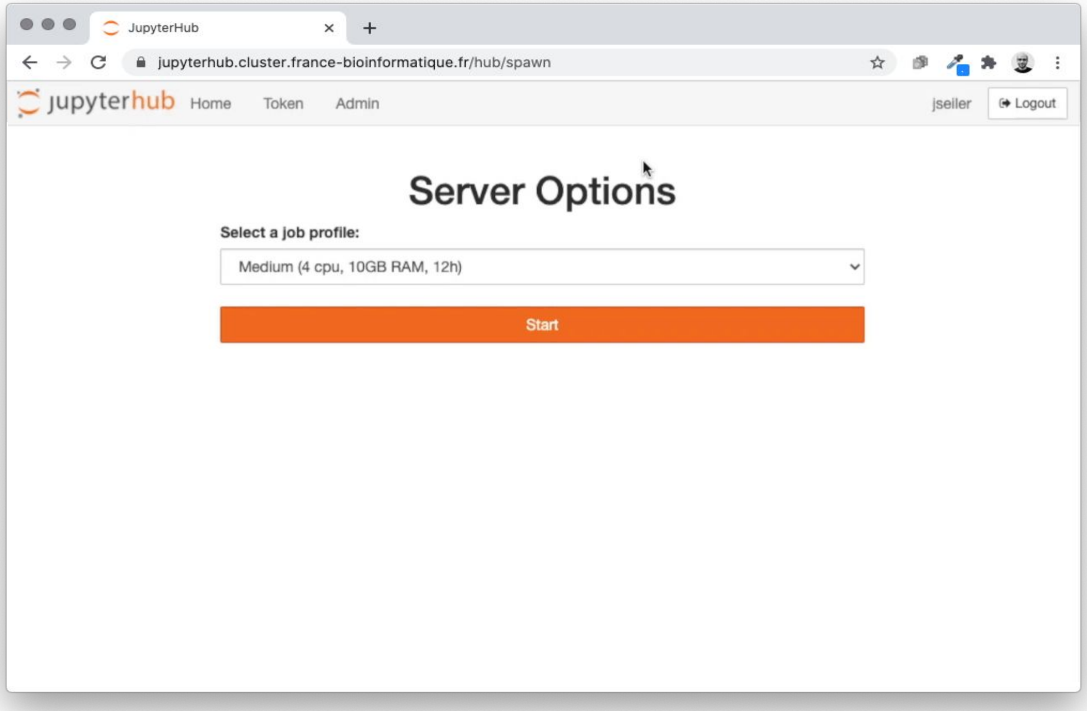
JupyterHub @ IFB

`https://jupyterhub.cluster.france-bioinformatique.fr`

Use your **IFB cluster account** to log in

Spawn JupyterLab server in **SLURM jobs**

Work on the **same storage** as the cluster (ssh)



Server Options

Select a job profile:

Medium (4 cpu, 10GB RAM, 12h)

Demo of notebooks and JupyterLab

Présentation et démonstration ->

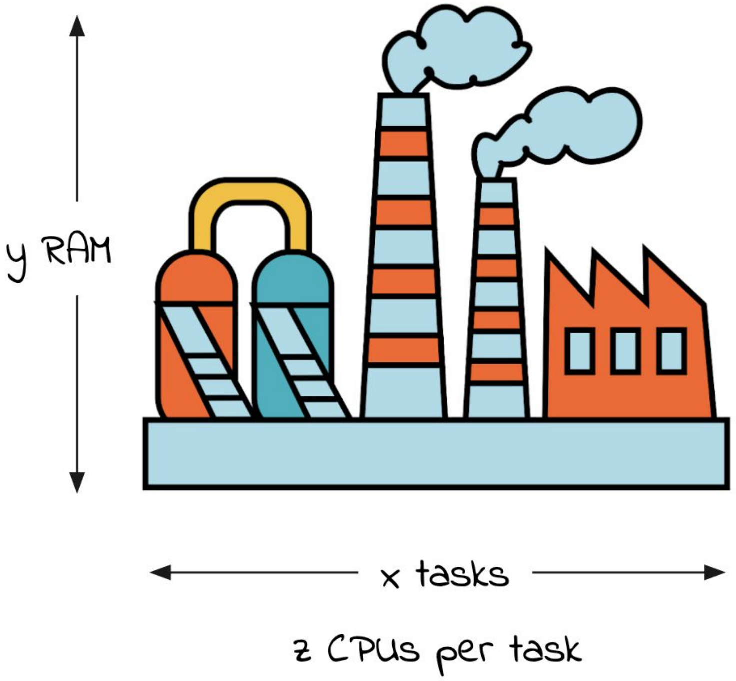
Lean SLURM in a notebook

JupyterLab supports a Bash kernel that let you write notebooks using Bash commands.

The IFB is proposing a SLURM tutorial based on a notebook :

https://gitlab.com/ifb-elixirfr/cluster/tutoriel_slurm

Let's view some best practices to use SLURM the FAIR way.



FAIR Jupyter notebook best practices

- Use Git to follow history of your notebooks (see JupyterLab Git extension)
- Automatically download data from a repository
- Make sure to identify the version of the libraries you are using in your notebooks :
  - Python : watermark or session_info
  - R : sessionInfo

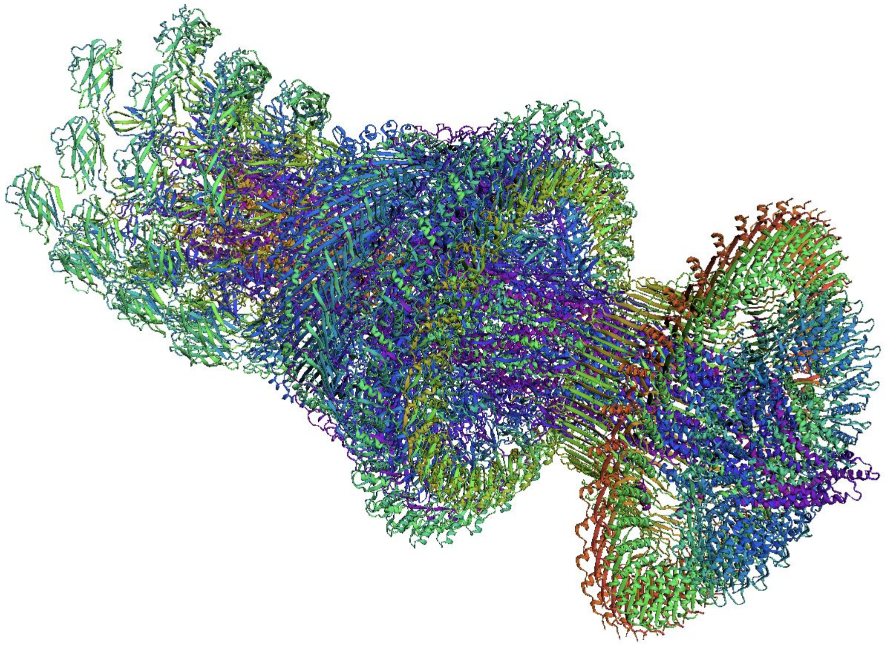
For more tips read :

https://github.com/jupyter-guide/ten-rules-jupyter

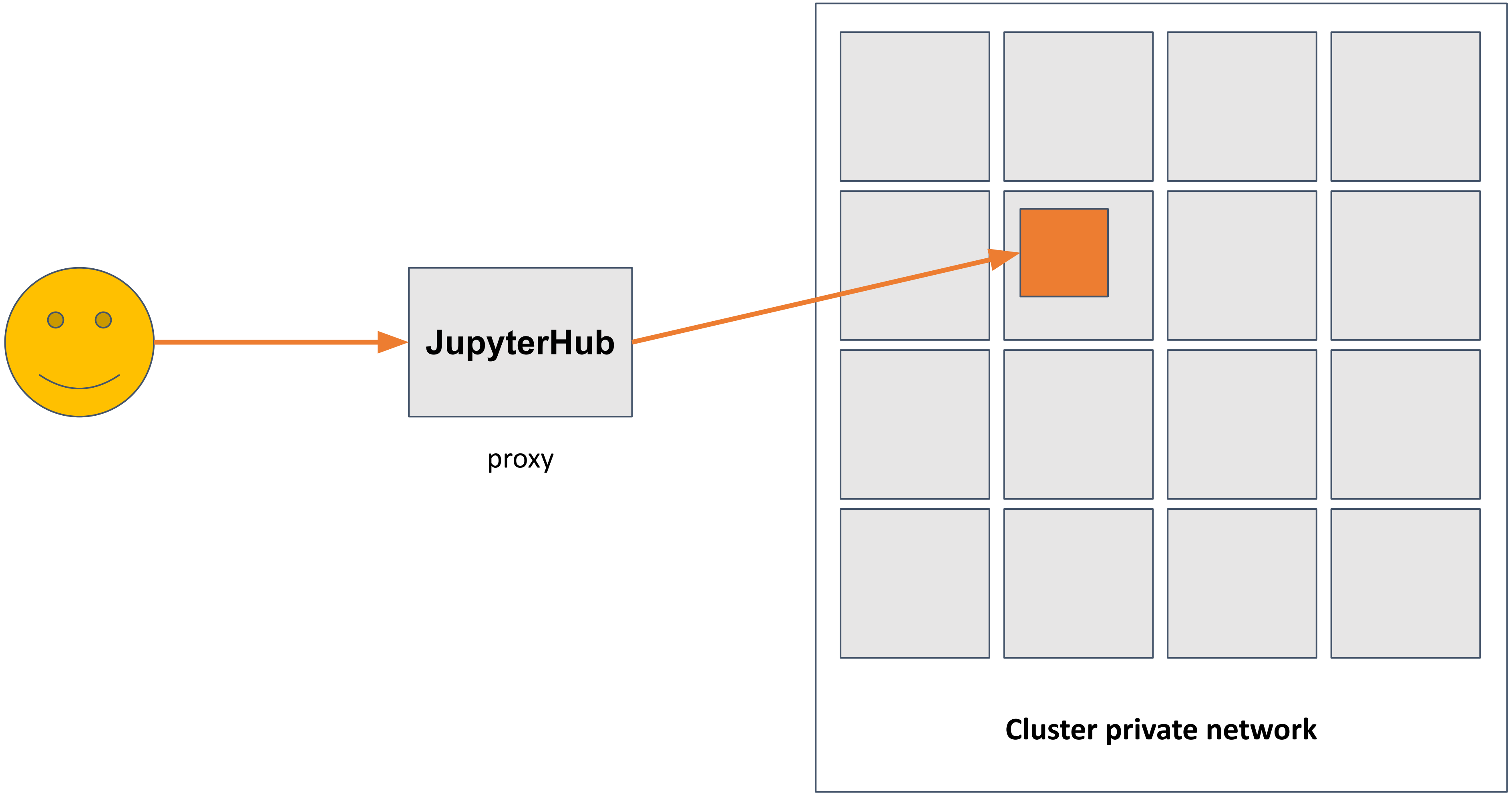
More interactive analysis with notebooks

- Render (dynamic) charts and visualize 3d models
- Train a network with Tensorflow and visualize training logs with Tensorboard

Demo notebooks can be downloaded from
https://gitlab.com/ifb-elixirfr/notebooks/fairbioinfo-demo



Run a Shiny app



Tools

Where is my tools?

Artist: Daniel Nyari

Conda - usage

Installation of miniconda (only once)

```
$ wget https://repo.anaconda.com/miniconda/Miniconda3-py39_4.9.2-Linux-x86_64.sh
$ bash Miniconda3-py39_4.9.2-Linux-x86_64.sh -b -p ~/miniconda3
$ conda config --add channels bioconda; conda config --add channels conda-forge
```

Search for a package

```
$ conda search fastqc==0.11.9
```

or https://anaconda.org/search?q=fastqc

Create an environment for a tool (recommended)

```
$ conda create -n fastqc-0.11.9 fastqc==0.11.9
```

Load a conda environment and use

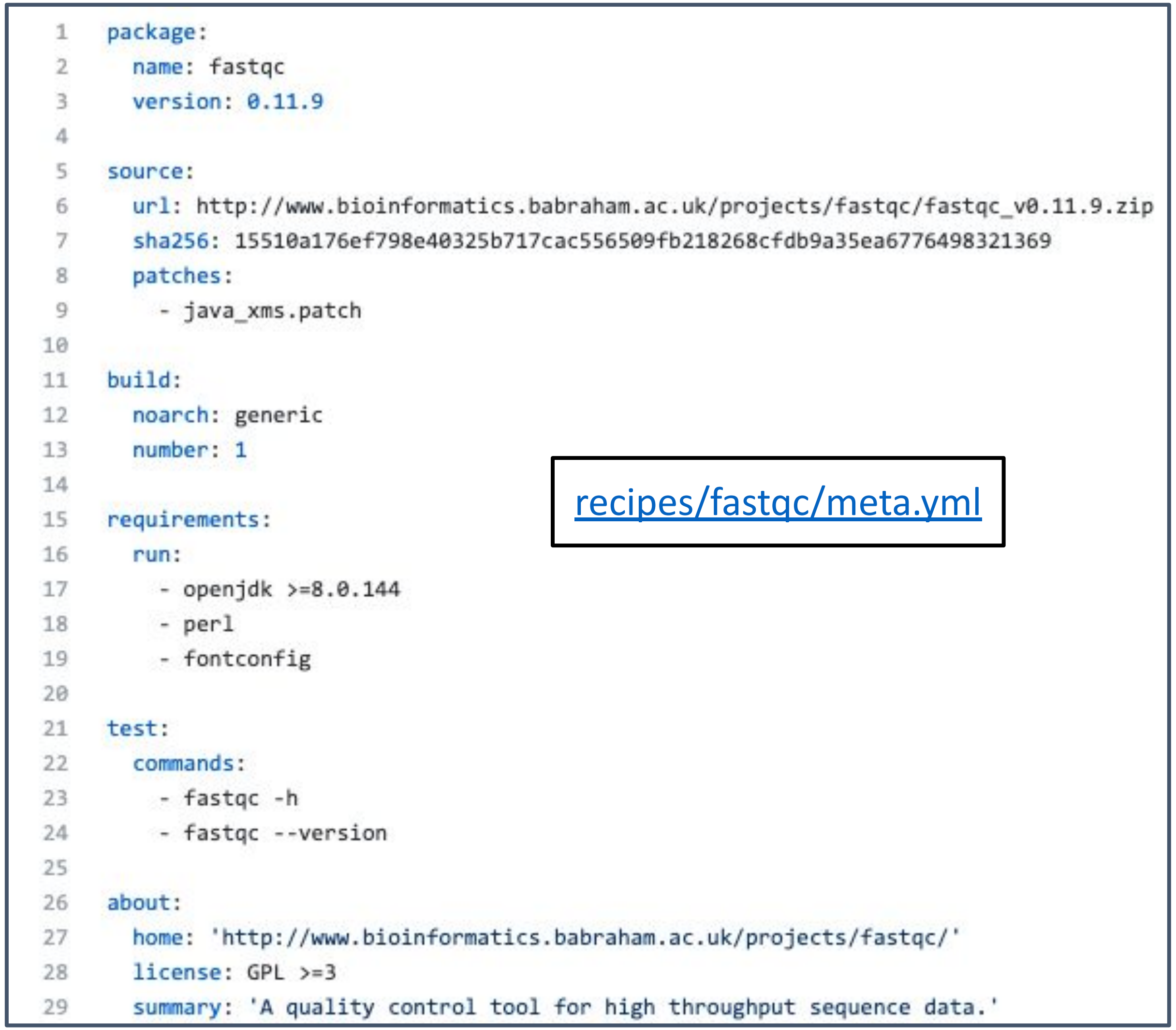
```
$ conda activate fastqc-0.11.9
$ fastqc --version
FastQC v0.11.9
```

ℹ Conda packages are provided by a central repository hosted by a compagny called Anaconda.org

Conda - building

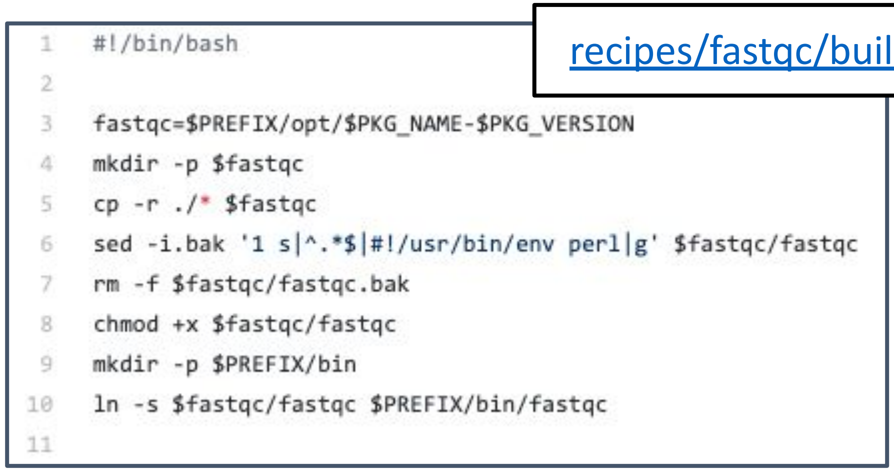
BIOCONDA®

Conda packaging consists of 2 files



```yaml
package:
  name: fastqc
  version: 0.11.9

source:
  url: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip
  sha256: 15510a176ef798e40325b717cac556509fb218268cfdb9a35ea6776498321369
  patches:
    - java_xms.patch

build:
  noarch: generic
  number: 1

requirements:
  run:
    - openjdk >=8.0.144
    - perl
    - fontconfig

test:
  commands:
    - fastqc -h
    - fastqc --version

about:
  home: 'http://www.bioinformatics.babraham.ac.uk/projects/fastqc/'
  license: GPL >=3
  summary: 'A quality control tool for high throughput sequence data.'
```

recipes/fastqc/meta.yml



```bash
#!/bin/bash

fastqc=$PREFIX/opt/$PKG_NAME-$PKG_VERSION
mkdir -p $fastqc
cp -r ./* $fastqc
sed -i.bak '1 s|^.*$|#!/usr/bin/env perl|g' $fastqc/fastqc
rm -f $fastqc/fastqc.bak
chmod +x $fastqc/fastqc
mkdir -p $PREFIX/bin
ln -s $fastqc/fastqc $PREFIX/bin/fastqc

```

recipes/fastqc/build.sh

```
$ # To build and test locally
$ conda build .
```

ℹ Consider to contribute to the Bioconda community/channel
https://bioconda.github.io/

Docker - usage

Search for a Docker a image or https://hub.docker.com/r/biocontainers/fastqc

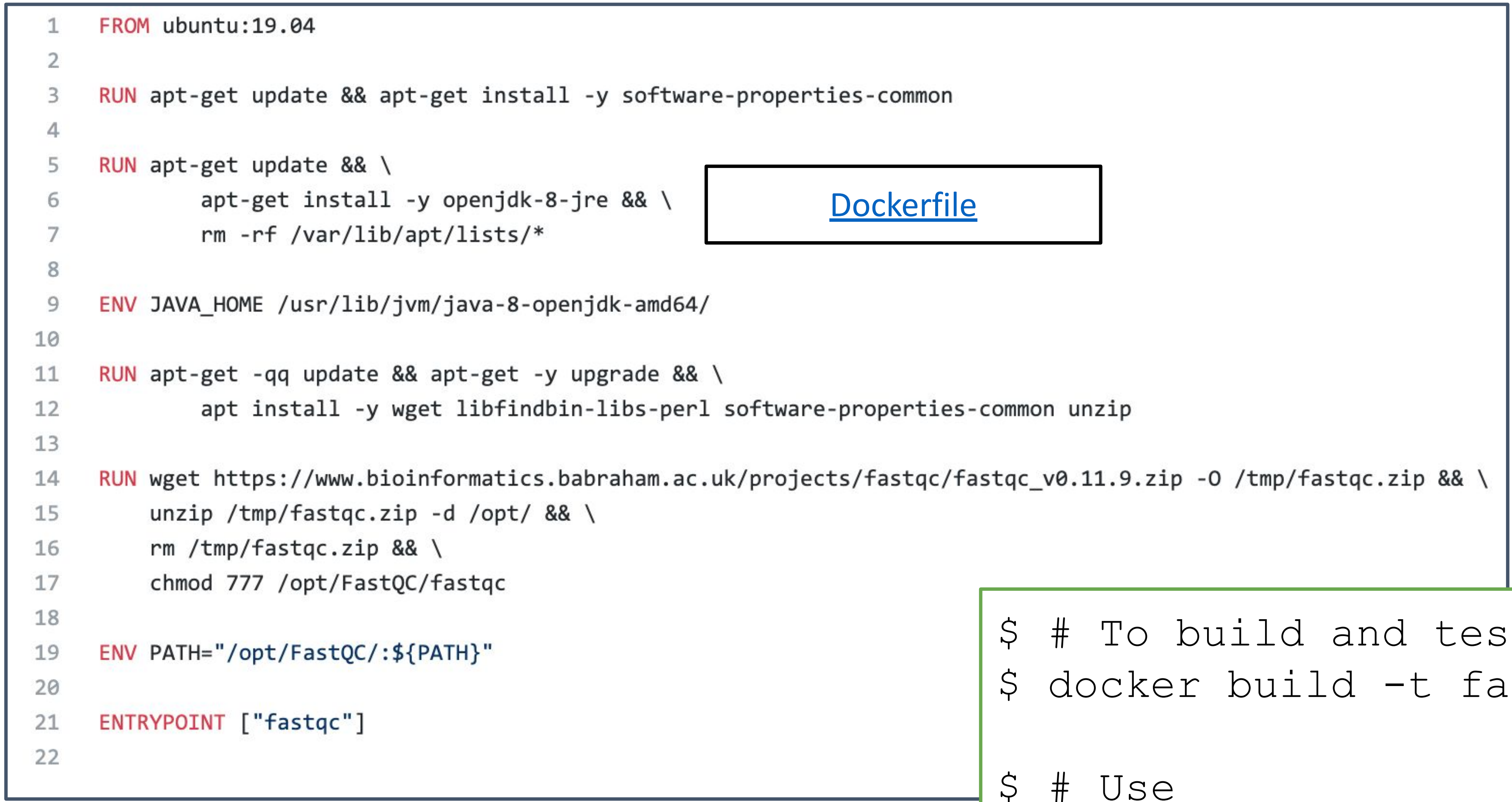
```
$ docker search fastqc
NAME                   DESCRIPTION     STARS     OFFICIAL    AUTOMATED
biocontainers/fastqc   fastqc          3                      [OK]
```

Pull and Run

```
$ docker run biocontainers/fastqc:v0.11.9_cv8 fastqc --version
[...]
$ FastQC v0.11.9
```

ℹ Docker isn't reliable in the context of an HPC infrastructure because of the need of the Docker daemon

Docker - building



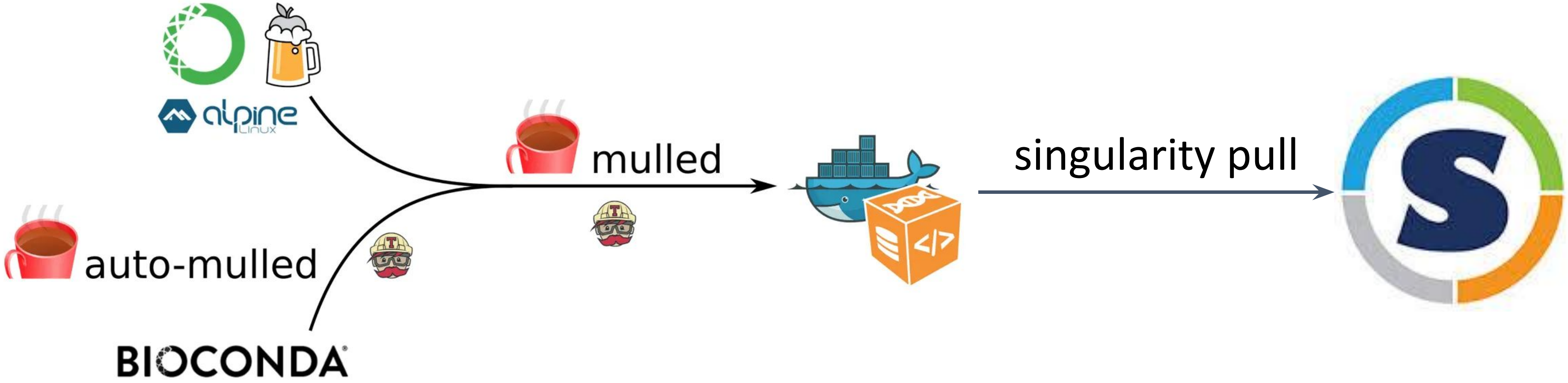
```
FROM ubuntu:19.04

RUN apt-get update && apt-get install -y software-properties-common

RUN apt-get update && \
        apt-get install -y openjdk-8-jre && \
        rm -rf /var/lib/apt/lists/*

ENV JAVA_HOME /usr/lib/jvm/java-8-openjdk-amd64/

RUN apt-get -qq update && apt-get -y upgrade && \
        apt install -y wget libfindbin-libs-perl software-properties-common unzip

RUN wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip -O /tmp/fastqc.zip && \
    unzip /tmp/fastqc.zip -d /opt/ && \
    rm /tmp/fastqc.zip && \
    chmod 777 /opt/FastQC/fastqc

ENV PATH="/opt/FastQC/:${PATH}"

ENTRYPOINT ["fastqc"]

```

Dockerfile

```
$ # To build and test locally
$ docker build -t fastqc-0.11.9

$ # Use
$ docker run fastqc-0.11.9 fasqtc --version
$ FastQC v0.11.9
```

Conda 2 Docker 2 Singularity

BioContainer



Process this full example:

https://ifb-elixirfr.gitlab.io/cluster/doc/singularity/#a-full-example

Singularity - usage

Search for a Docker a image

https://hub.docker.com/r/biocontainers/fastqc

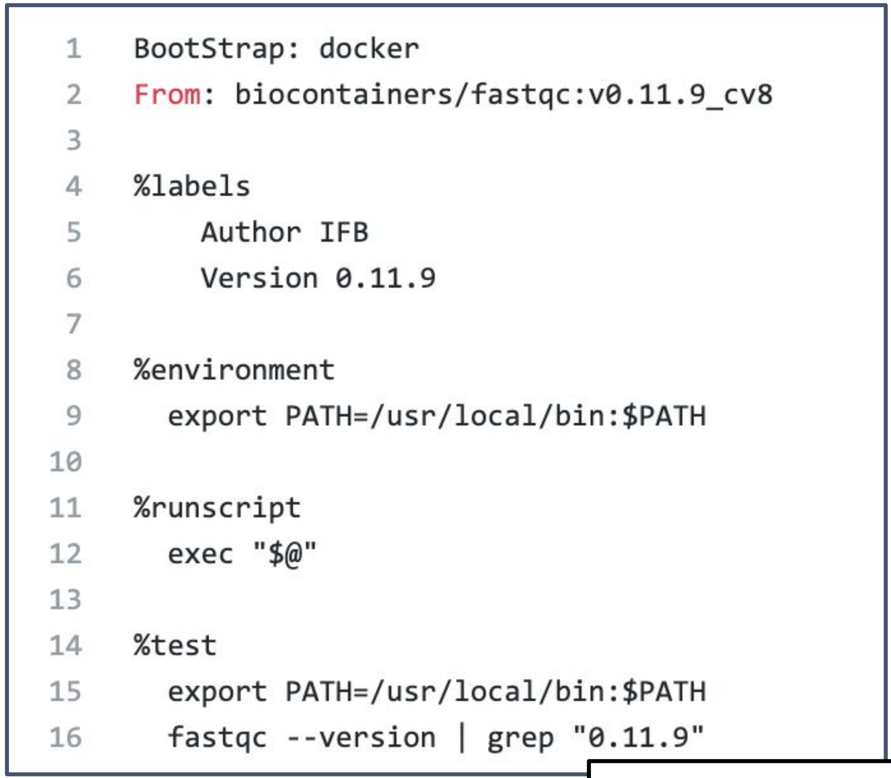
Pull an image

```
$ singularity pull docker://biocontainers/fastqc:v0.11.9_cv8
$ ls -l fastqc_v0.11.9_cv8.sif
-rwxr-xr-x 1 foo bar 297582592 Jun 22 18:11 fastqc_v0.11.9_cv8.sif
```

Use

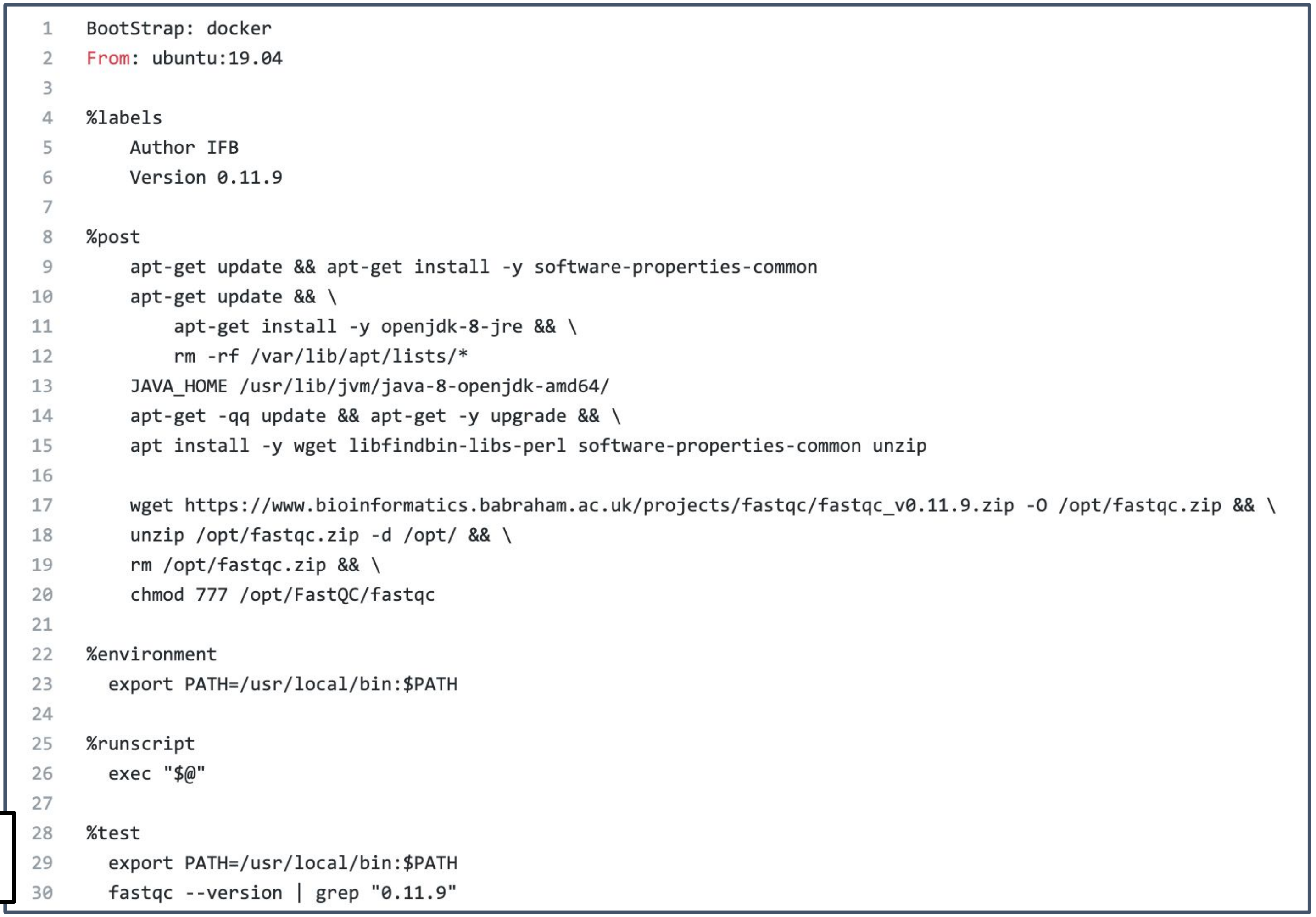
```
$ ./fastqc_v0.11.9_cv8.sif fastqc --version
$ FastQC v0.11.9
```

Singularity - build



```
BootStrap: docker
From: biocontainers/fastqc:v0.11.9_cv8

%labels
    Author IFB
    Version 0.11.9

%environment
  export PATH=/usr/local/bin:$PATH

%runscript
  exec "$@"

%test
  export PATH=/usr/local/bin:$PATH
  fastqc --version | grep "0.11.9"
```

OR



```
BootStrap: docker
From: ubuntu:19.04

%labels
    Author IFB
    Version 0.11.9

%post
    apt-get update && apt-get install -y software-properties-common
    apt-get update && \
        apt-get install -y openjdk-8-jre && \
        rm -rf /var/lib/apt/lists/*
    JAVA_HOME /usr/lib/jvm/java-8-openjdk-amd64/
    apt-get -qq update && apt-get -y upgrade && \
    apt install -y wget libfindbin-libs-perl software-properties-common unzip

    wget https://www.bioinformatics.babraham.ac.uk/projects/fastqc/fastqc_v0.11.9.zip -O /opt/fastqc.zip && \
    unzip /opt/fastqc.zip -d /opt/ && \
    rm /opt/fastqc.zip && \
    chmod 777 /opt/FastQC/fastqc

%environment
  export PATH=/usr/local/bin:$PATH

%runscript
  exec "$@"

%test
  export PATH=/usr/local/bin:$PATH
  fastqc --version | grep "0.11.9"
```

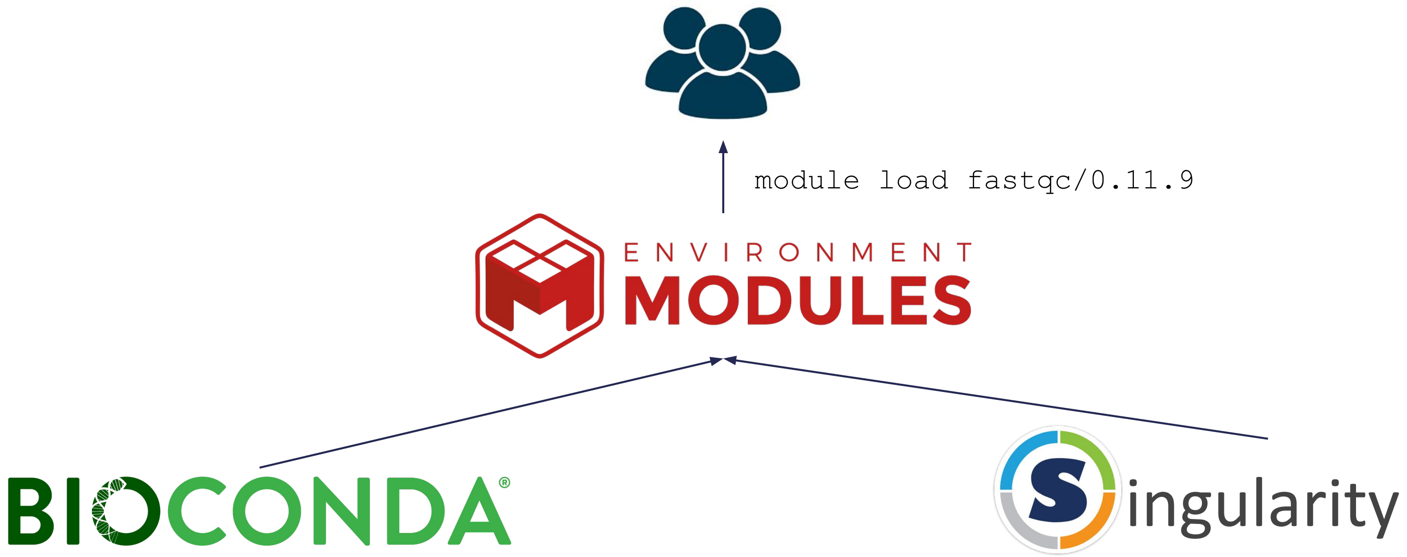
image.def

Conda or Docker or Singularity?

	PROS	CONS
CONDA	● Light during the installation ● Not need to be root ● Sharing repository ● The Alpha exported in Docker	● They are issue to install “old” packages 2 or 3 years after their creation (dependencies have changed) ● A lot of tiny files ● No isolation - security issue
docker	● Portable ● Sharing repository ● Can be ate by Singularity ● Come with the OS	● Not compatible with the HPC infrastructure ● Rather heavy to install, need root grants ○ Need a centralized daemon ● Some security issues/concerns
S	● Compatible with HPC since it’s execute as a binary ● Compatible with Docker image format ● Come with the OS	● Don’t provide the same layer system as Docker ○ So heavy on the filesystem ● No stable shared repository ● It’s a deadlock that can’t be exported ● Not well integrated on MacOSX

Module - usage at IFB

2 technologies - 1 user interface



```
module load fastqc/0.11.9
```

by default

if a licence must be accepted
if not in Bioconda and hard to integrate or urgent
if a Docker container exists

Module - usage

Why do we need to "load" tools ?

- Each tools need its environment (binaries, libraries, documentation, special variables)
- Each tools has its own dependencies.
- It is not possible to coexist all tools in the same environment.
- Reproducibility does matter: some user might need different versions of the same tool
- At the IFB, the cluster community is installing all tools required by the users.

All tool deployment are based on Conda packages or Singularity images :

To get access to a tool, you need to load it into your environment using a special tool called `module`.

ENVIRONMENT MODULES

Module - usage

Loading, listing, switching, unloading

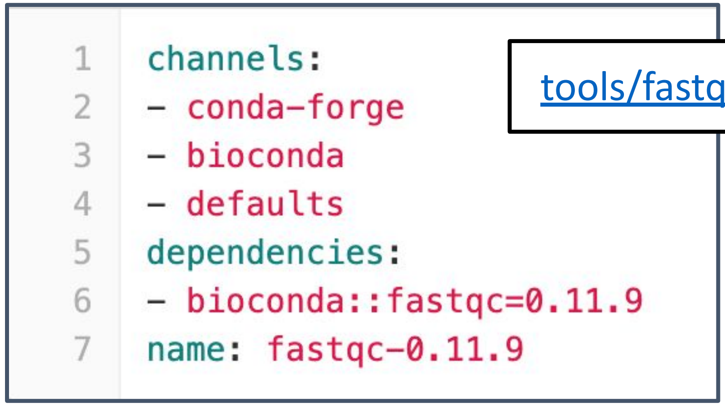
```
module avail         # List the modules available (477 in June 2021)
module avail fastqc  # List the versions available for a tool

module load fastqc   # Load latest version available on the cluster
module load fastqc/0.11.9 multiqc/1.10.1 # Load software
module list          # List tools currently loaded in your environment

module switch fastqc/0.11.7   # Replace current version

module unload blast           # Unload blast from your environment
module purge                  # Unload all tools
```

Module - build at IFB

Institut Français de Bioinformatique › Cluster › tools

tools

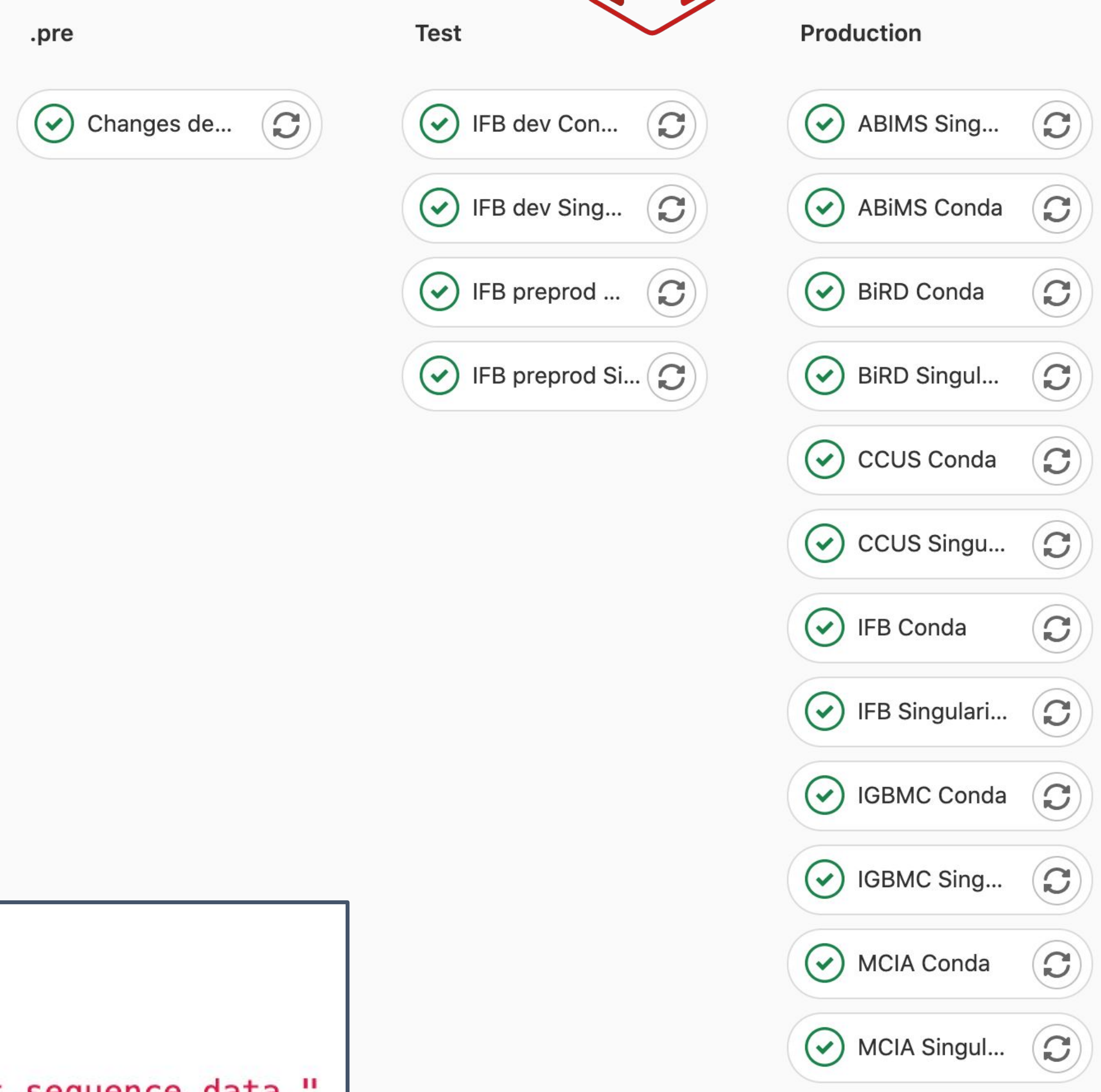
Project ID: 15693267



tools/fastqc/0.11.9/meta.yml

```
channels:
- conda-forge
- bioconda
- defaults
dependencies:
- bioconda::fastqc=0.11.9
name: fastqc-0.11.9
```

🤖 @ifbot



.pre	Test	Production
✓ Changes de...	✓ IFB dev Con...	✓ ABIMS Sing...
	✓ IFB dev Sing...	✓ ABiMS Conda
	✓ IFB preprod ...	✓ BiRD Conda
	✓ IFB preprod Si...	✓ BiRD Singul...
		✓ CCUS Conda
		✓ CCUS Singu...
		✓ IFB Conda
		✓ IFB Singulari...
		✓ IGBMC Conda
		✓ IGBMC Sing...
		✓ MCIA Conda
		✓ MCIA Singul...

tools/fastqc/0.11.9/meta.yml

```
deployment: conda

about:
  description: "A quality control tool for high throughput sequence data."
  url: http://api.anaconda.org/packages/bioconda/fastqc
```

TP - Snakemake over SLURM

TP - Snakemake over SLURM

Exercice 1: connect to the cluster through JupyterHub

- Go to https://jupyterhub.cluster.france-bioinformatique.fr
- Start a small JupyterLab server with 1 CPU and 1 GB of RAM
- Start a Terminal (from the JupyterLab launcher)

TP - Snakemake over SLURM

Exercice 2: Get your environment ready

- Download the workflow
- Download your input data
- Load the snakemake module and all required tools

TP - Snakemake over SLURM

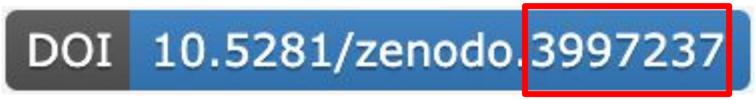
Exercice 2: Get your environment ready

- Download the workflow
- Download your input data

The data used for the snakemake tutorial are available on Zenodo :

Go to zenodo.org

Search for


DOI 10.5281/zenodo.3997237

Copy/paste the download link

TP - Snakemake over SLURM

Exercice 2: Get your environment ready

- Download the workflow
- Download your input data

Download the snakemake workflow and data archive :

```
$ git clone https://github.com/clairetn/FAIR_smk.git
$ cd FAIR_smk

$ module load zenodo_get/1.3.2
$ zenodo_get 10.5281/zenodo.3997237
$ tar -xvzf FAIR_Bioinfo_data.tar.gz
```

TP - Snakemake over SLURM

Exercice 3: Run snakemake

- Run your workflow using `--cluster` mode
- Run your workflow using `--drmaa` mode

TP - Snakemake over SLURM

Exercice 3: Run snakemake

- Run your workflow using `--cluster` mode

```
module load snakemake

snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*

snakemake --cluster "sbatch" --jobs=3 --cores=3 --use-conda -s ex1_o8.smk
```

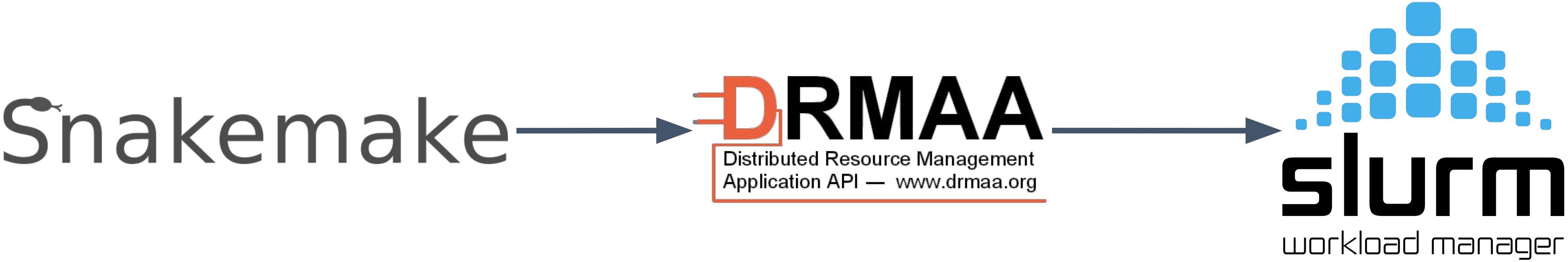
Drawbacks : no control on workflow execution (you can’t stop it)

TP - Snakemake over SLURM

Exercice 3: Run snakemake

- Run your workflow using `--cluster` mode
- Run your workflow using `--drmaa` mode

Distributed **R**esource **M**anagement **A**pplication **API**


Snakemake
DRMAA
Distributed Resource Management
Application API — www.drmaa.org
slurm
workload manager

TP - Snakemake over SLURM

Exercice 3: Run snakemake

- Run your workflow using `--cluster` mode
- Run your workflow using `--drmaa` mode

```
module load snakemake

snakemake --drmaa --use-conda --jobs=3 -s ex1_o8.smk
```

TP - Snakemake over SLURM --use-conda

CONDA

```
rule fastqc:
[...]
  conda:
    "envs/fastqc-0.11.9.yml"
  container:
     "docker://biocontainers/fastqc:v0.11.9_cv8"
  envmodules:
    "fastqc/0.11.9"
  shell: "fastqc --outdir FastQC/ {input} 1>{log.std} 2>{log.err}"
```

```
module purge; module load snakemake conda
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
time snakemake --drmaa --jobs=3 -s ex1_o8.smk --use-conda
```

TP - Snakemake over SLURM --use-singularity

```
rule fastqc:
[...]
  conda:
    "envs/fastqc-0.11.9.yml"
  container:
     "docker://biocontainers/fastqc:v0.11.9_cv8"
  envmodules:
    "fastqc/0.11.9"
  shell: "fastqc --outdir FastQC/ {input} 1>{log.std} 2>{log.err}"
```

```
module purge; module load snakemake singularity
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
time snakemake --drmaa --jobs=3 -s ex1_o8.smk --use-singularity
```

TP - Snakemake over SLURM --use-envmodule

ENVIRONMENT MODULES

```
rule fastqc:
[...]
  conda:
    "envs/fastqc-0.11.9.yml"
  container:
     "docker://biocontainers/fastqc:v0.11.9_cv8"
  envmodules:
    "fastqc/0.11.9"
  shell: "fastqc --outdir FastQC/ {input} 1>{log.std} 2>{log.err}"
```

```
module purge; module load snakemake
snakemake -c 1 -s ex1_o8.smk --delete-all-output; rm -rf multiqc_*
time snakemake --drmaa --jobs=3 -s ex1_o8.smk --use-envmodule
```

Useful links

Request an account:
https://my.cluster.france-bioinformatique.fr

Community support:
https://community.france-bioinformatique.fr/

Learn SLURM in 5 minutes:
https://asciinema.org/a/275233

IFB Core Cluster Documentation
https://ifb-elixirfr.gitlab.io/cluster/doc/

BONUS

The **IFB** **C**ore **C**luster **I**nfrastructure

- Infrastructure administration is automated using Continuous Integration technologies :

ANSIBLE git GitLab IFBot

- Most IFB Core Cluster repositories are **open to contribution**
  - Help us manage the cluster infrastructure
  - Deploy bioinformatics software (conda, singularity, etc.)
  - Deploy new services

What’~~s~~ was new on the IFB NNCR Cluster(s) ?

David BENABEN [1,2] , Nicole CHARRIÈRE [3] , David CHRISTIANY [3] , François GERBES [3,6] , Jean-Christophe HAESSIG [4] ,
Didier LABORIE [5] , Gildas LE CORGUILLÉ [6*] , Olivier SALLOU [7] , Julien SEILER [4*] and Guillaume SEITH [4]

[1] CBiB, Université de Bordeaux, 142 rue Léo Saignat, 33076 Bordeaux, France
[2] INRAE, UMR 1332, Biologie du Fruit et Pathologie, CS20032 Villenave d'Ornon, France
[3] IFB/Institut Français de Bioinformatique, CNRS UMS 3601, IFB-Core, Génoscope, 91057, Évry, France
[4] CNRS, INSERM, IGBMC, 1 rue Laurent Fries, 67404, Illkirch, France
[5] GenoToul-Bioinfo, INRAE, 24 chemin de Borde-Rouge, Auzeville, 31326 Castenet-Tolosan, France
[6] Sorbonne Université/CNRS, FR2424, ABiMS, Station Biologique, 29680, Roscoff, France
[7] IRISA/Université Rennes 1, 263 Avenue Général Leclerc, 35000 Rennes, France

* Corresponding Authors: lecorguille@sb-roscoff.fr, julien.seiler@igbmc.fr

Montpellier
JOBIM
2020

Link