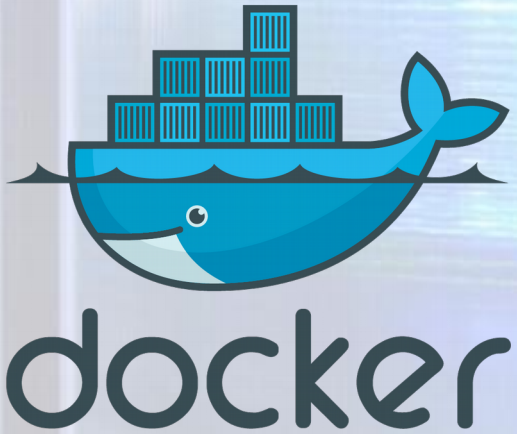
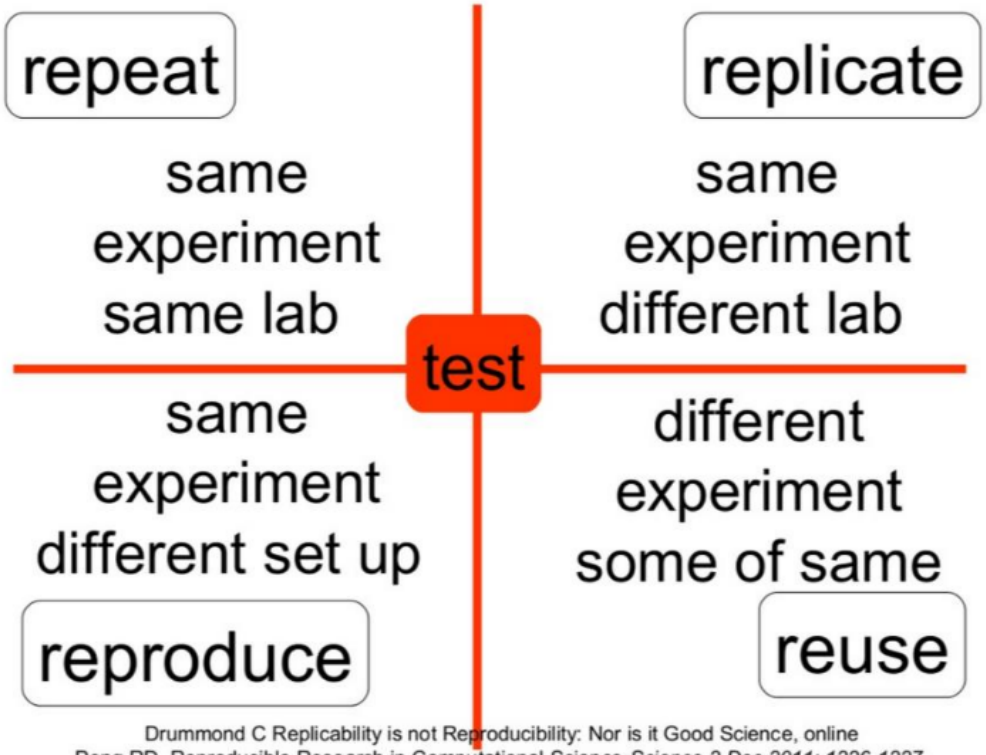
Docker et reproductibilité

Marine Aglave, Coralie Rohmer, Sarah Jelassi, Alice Rogier



Introduction



Drummond C Replicability is not Reproducibility: Nor is it Good Science, online
Peng RD, Reproducible Research in Computational Science *Science 2 Dec 2011: 1226-1227.*

docker

Introduction

Mon environnement est-il supporté ?

Rhaaa !! Une nouvelle manière de déployer !

Ai-je les bonnes dépendances?

Si ça ne marche pas, saurai-je revenir à mon environnement initial ?

Y a-t-il des conflits avec d'autres applications déjà installées ?

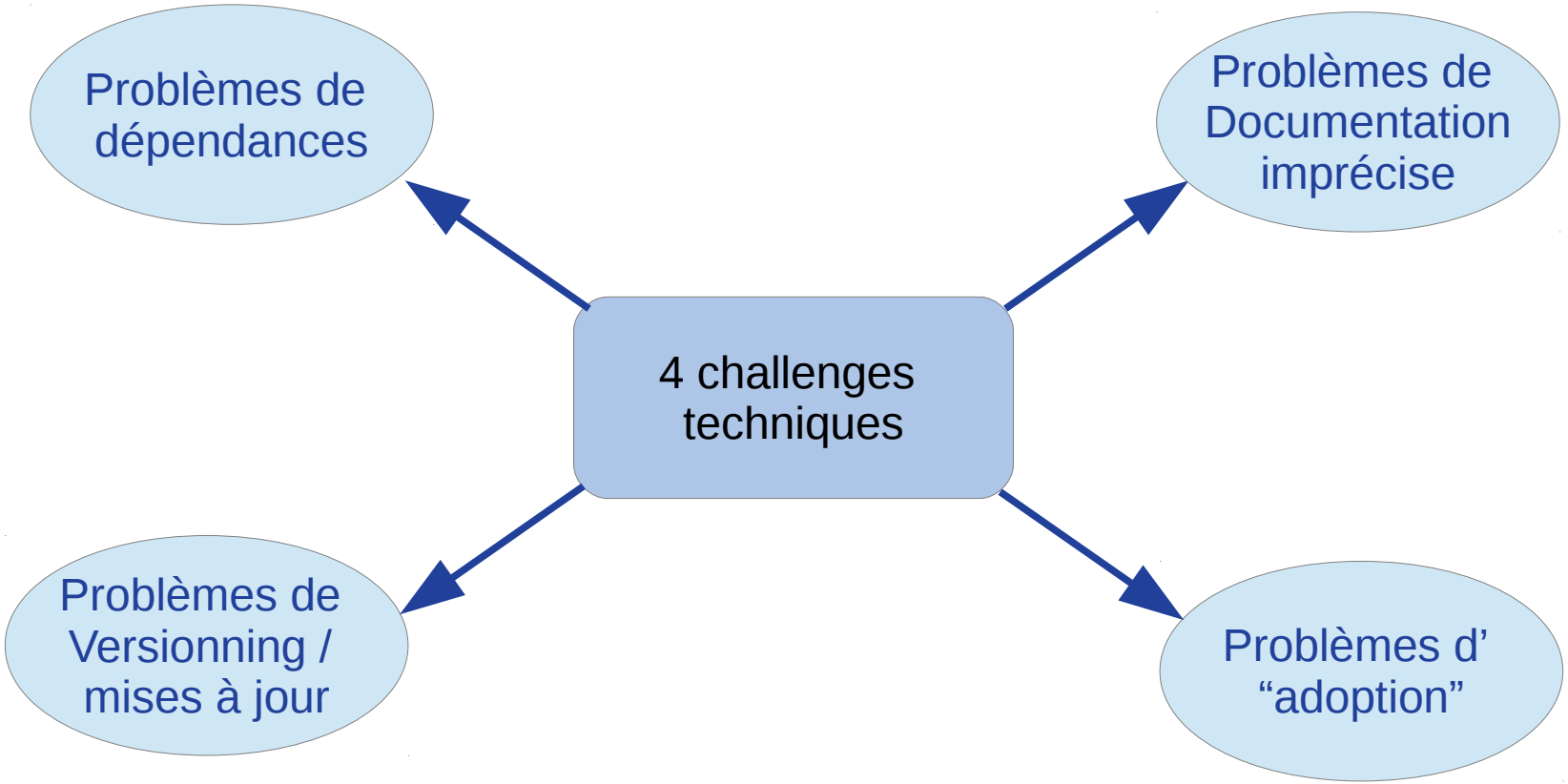
Je veux seulement tester une appli : est-ce que ça vaut le coup de passer tant de temps au déploiement ?

On me remonte un bug. Pourtant, ça marche sur ma machine !

Je ne le sens pas !

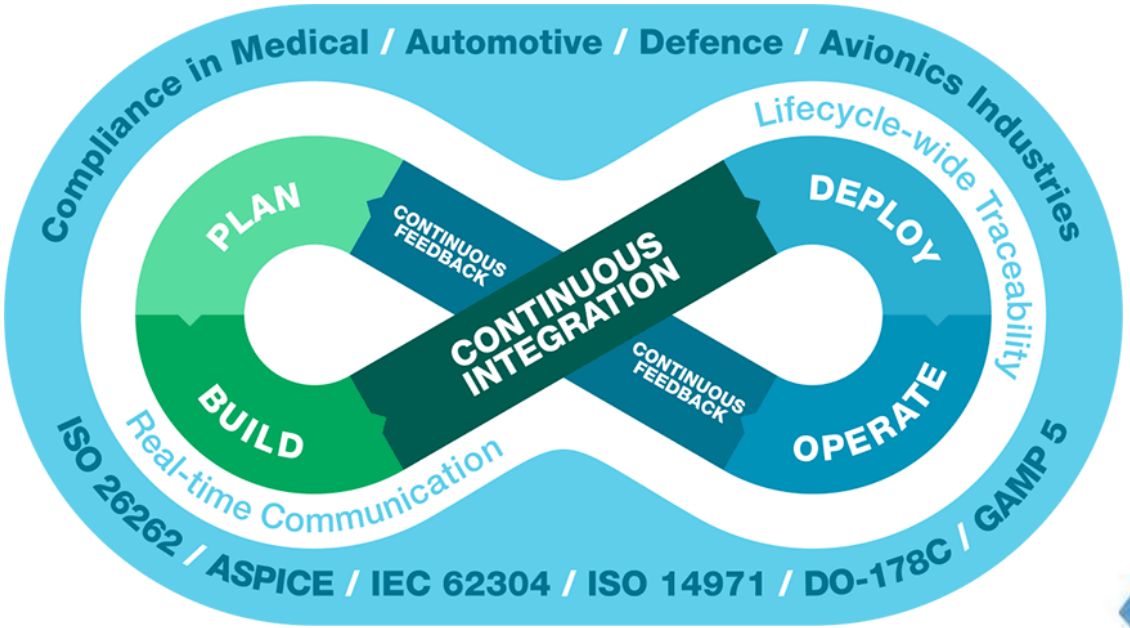
docker

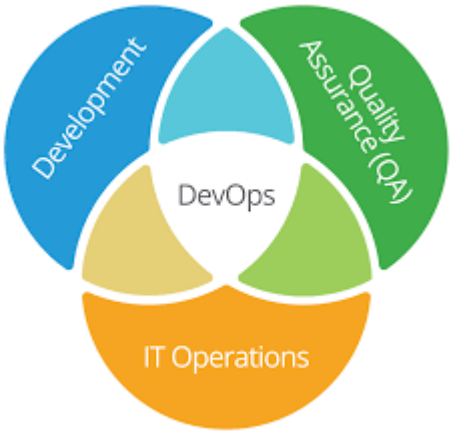
Challenges à relever pour une meilleure reproductibilité des expériences, en bioinformatique

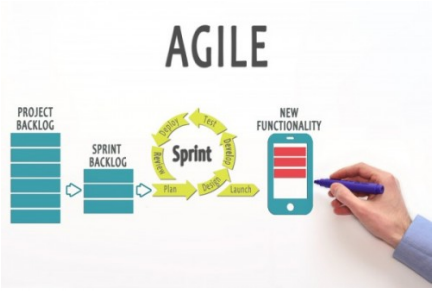


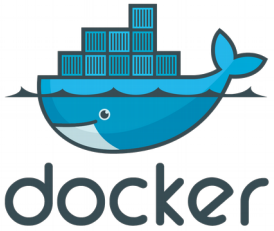
D'après *An introduction to Docker for reproducible research*
Carl Boettiger

Docker : un outil qui s’est développé dans la philosophie DevOps

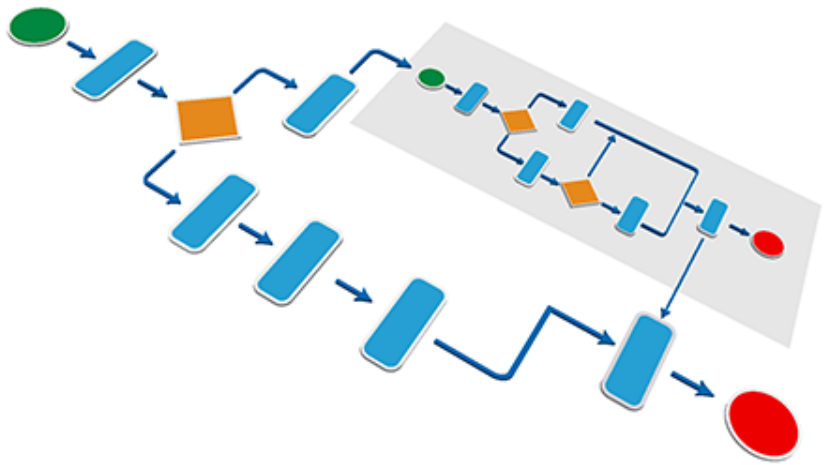






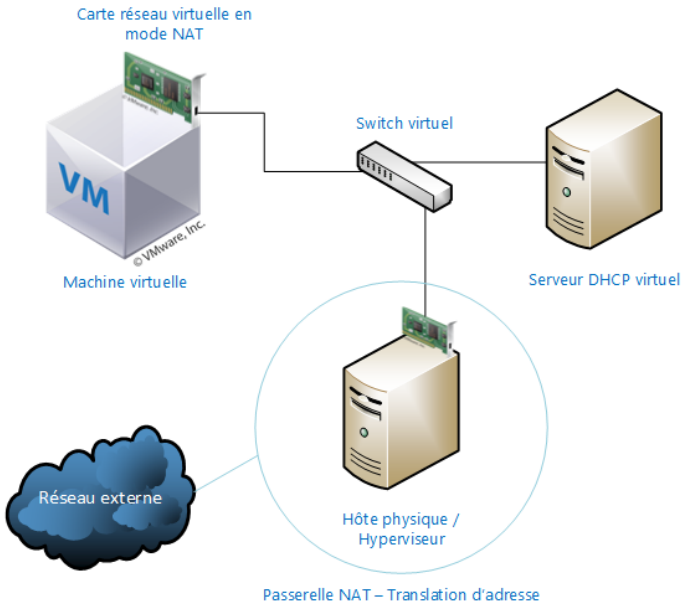


Solutions pré-existantes aux conteneurs



Kepler

SNAKE MAKE



docker

Solutions pré-existantes aux conteneurs : avantages et inconvénients

SYSTEMES DE WORFLOW	MACHINES VIRTUELLES
-communication facile (belle interface graphique) -collaboration aisée entre scientifique	-OS fournit les logiciels nécessaires pour coordonner les différents éléments -évite l'installation fasctidieuse de divers logiciels
-utilisateurs qui peinent à apprendre comment utiliser ces systèmes	-boîte noire donc mal adépté à la reproductibilité -on peine à comprendre ce qu'on fait

D'après *An introduction to Docker for reproducible research*
Carl Boettiger

Solutions pré-existantes aux conteneurs : avantages et inconvénients

SYSTEMES DE WORFLOW	MACHINES VIRTUELLES
Répond au problème de la documantation imprécise	Répond aux problèmes de versionning et de dépendances

D'après *An introduction to Docker for reproducible research*
Carl Boettiger

Solutions pré-existantes aux conteneurs : avantages et inconvénients

SYSTEMES DE WORFLOW	MACHINES VIRTUELLES
Répond au problème de la documantation imprécise	Répond aux problèmes de versionning et de dépendances

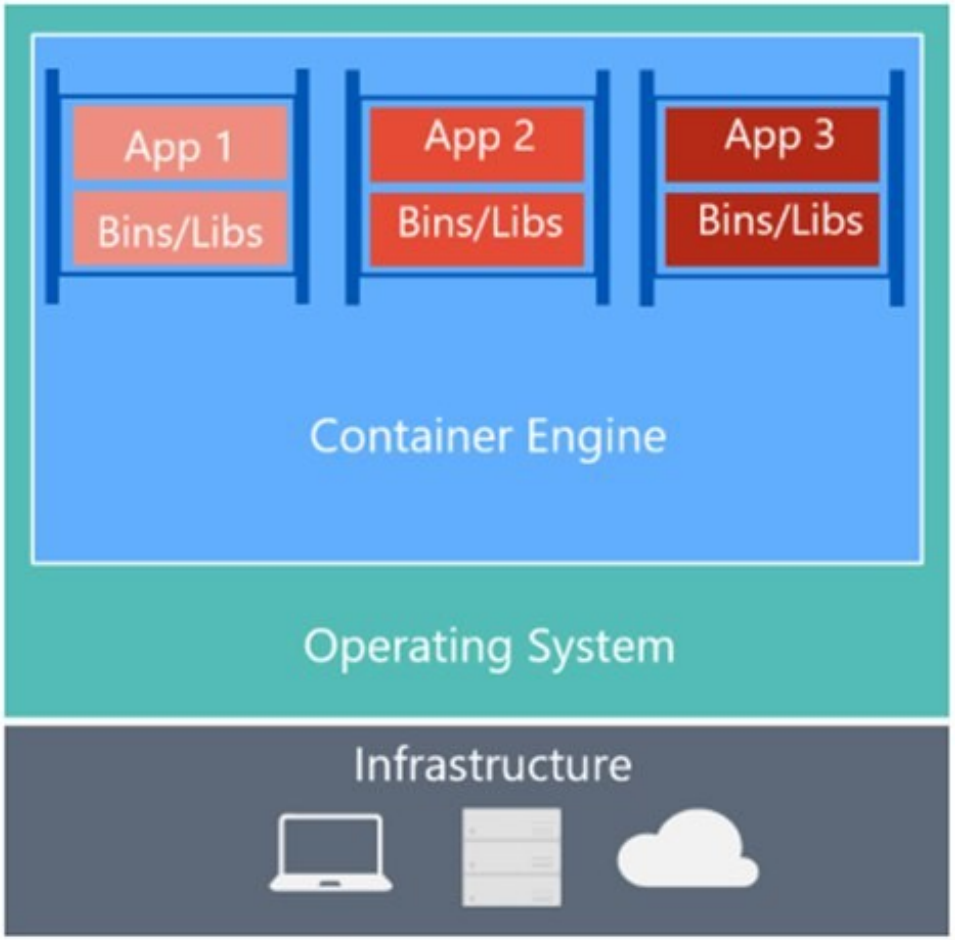
Docker, un système basé sur les conteneurs, peut répondre aux quatre challenges à la fois.

→ Mais qu'est-ce qu'un conteneur ?

D'après *An introduction to Docker for reproducible research*
Carl Boettiger

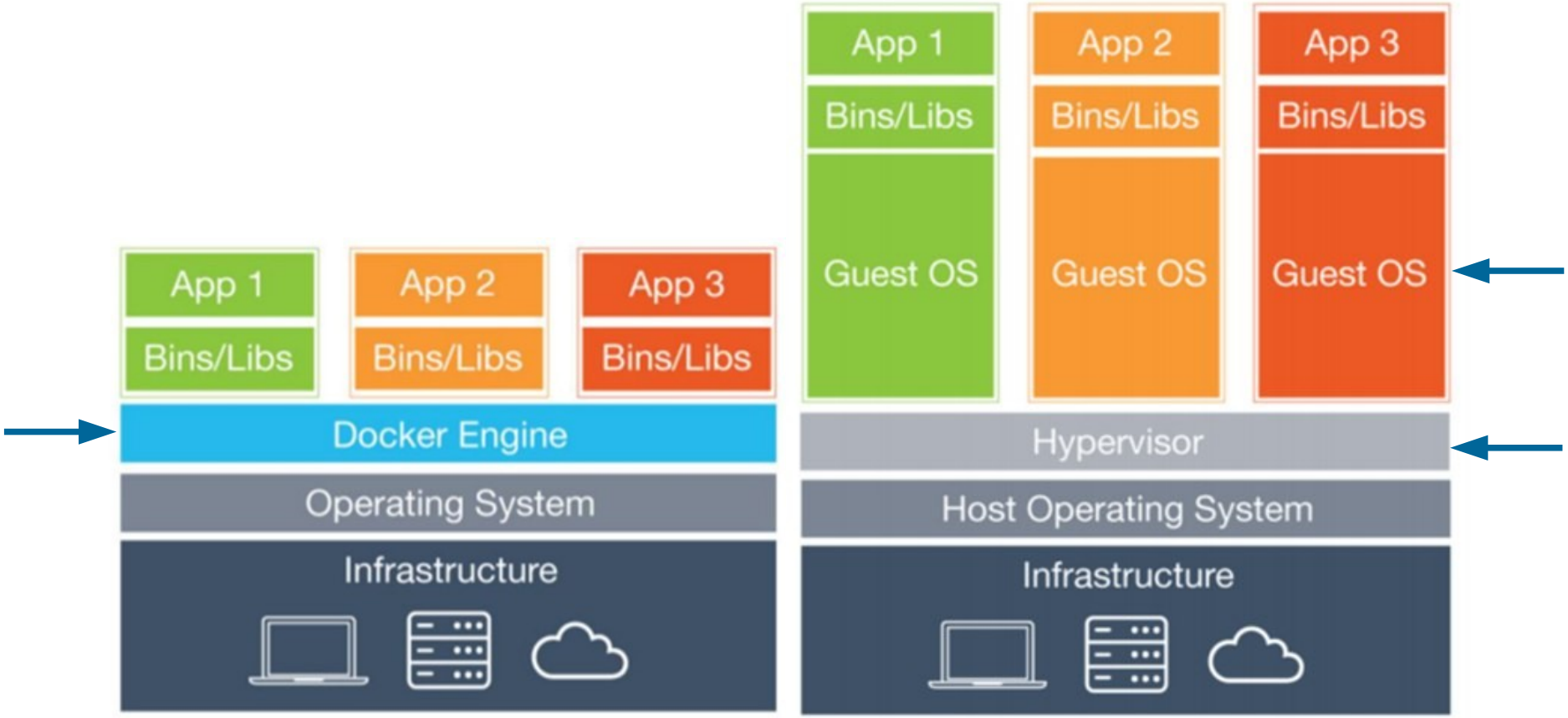
Qu’est ce qu’une image et un conteneur ?

Une image est un template prêt à l’emploi avec des instructions pour la création de conteneurs.



docker

La différence entre un conteneur et une machine virtuelle



Conteneurs **Machine Virtuelle**

La virtualisation par conteneurs : le LinuX Containers (LXC)

Principe : faire tourner des environnements linux dans des conteneurs partageant le même noyau.

LXC repose sur 2 fonctionnalités du noyau linux :

1) fonctionnalité des Cgroups (Control groups):

permettent de limite et d'isoler l'utilisation des ressources.

2) Name space (Cloisonnement des espaces de nommage):

empêche qu'un groupe puisse voir les ressources d'un autre groupe.

Les copains??

docker

Docker :

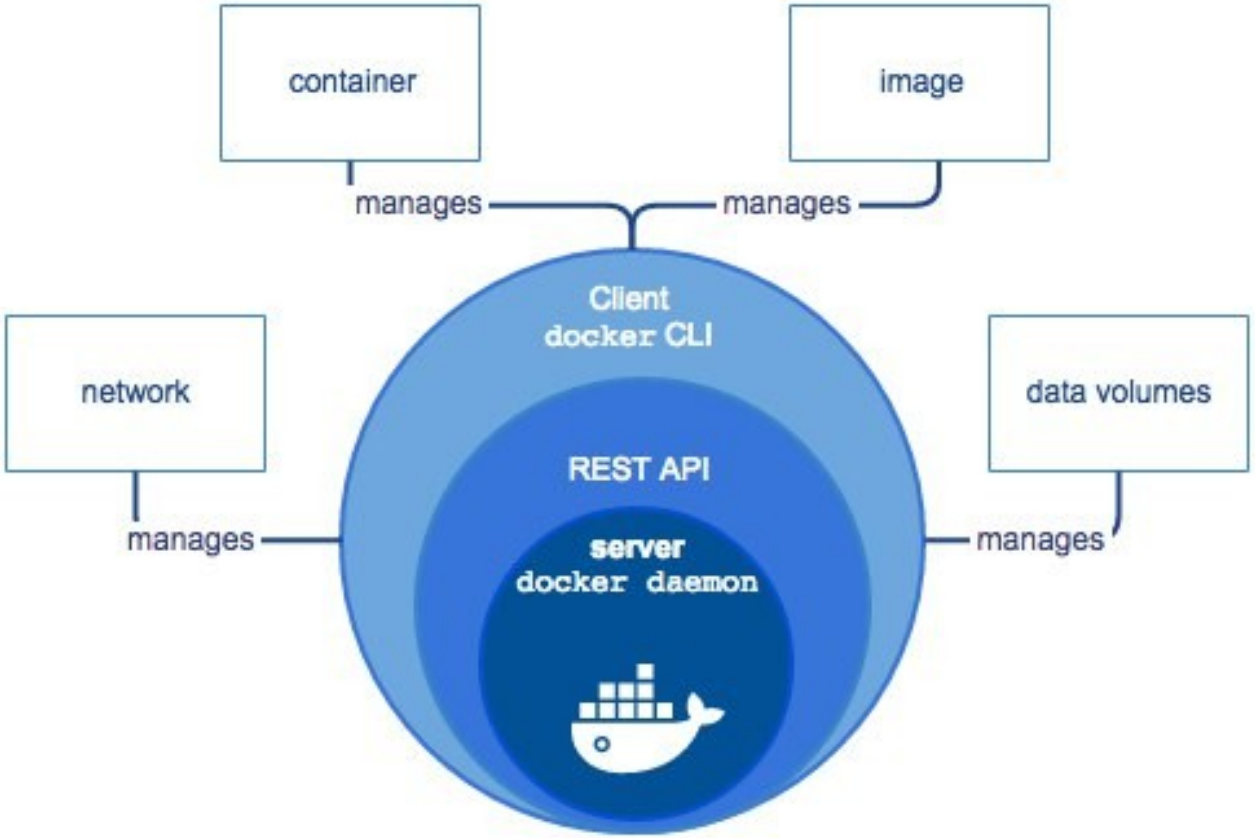
Première sortie le 13 mars 2010.

Docker permet de créer des conteneurs qui vont uniquement contenir des applications avec leurs dépendances.

Docker permet donc d’automatiser le déploiement d’applications en tant que conteneurs portables et autonomes exécutables sur le cloud ou localement.

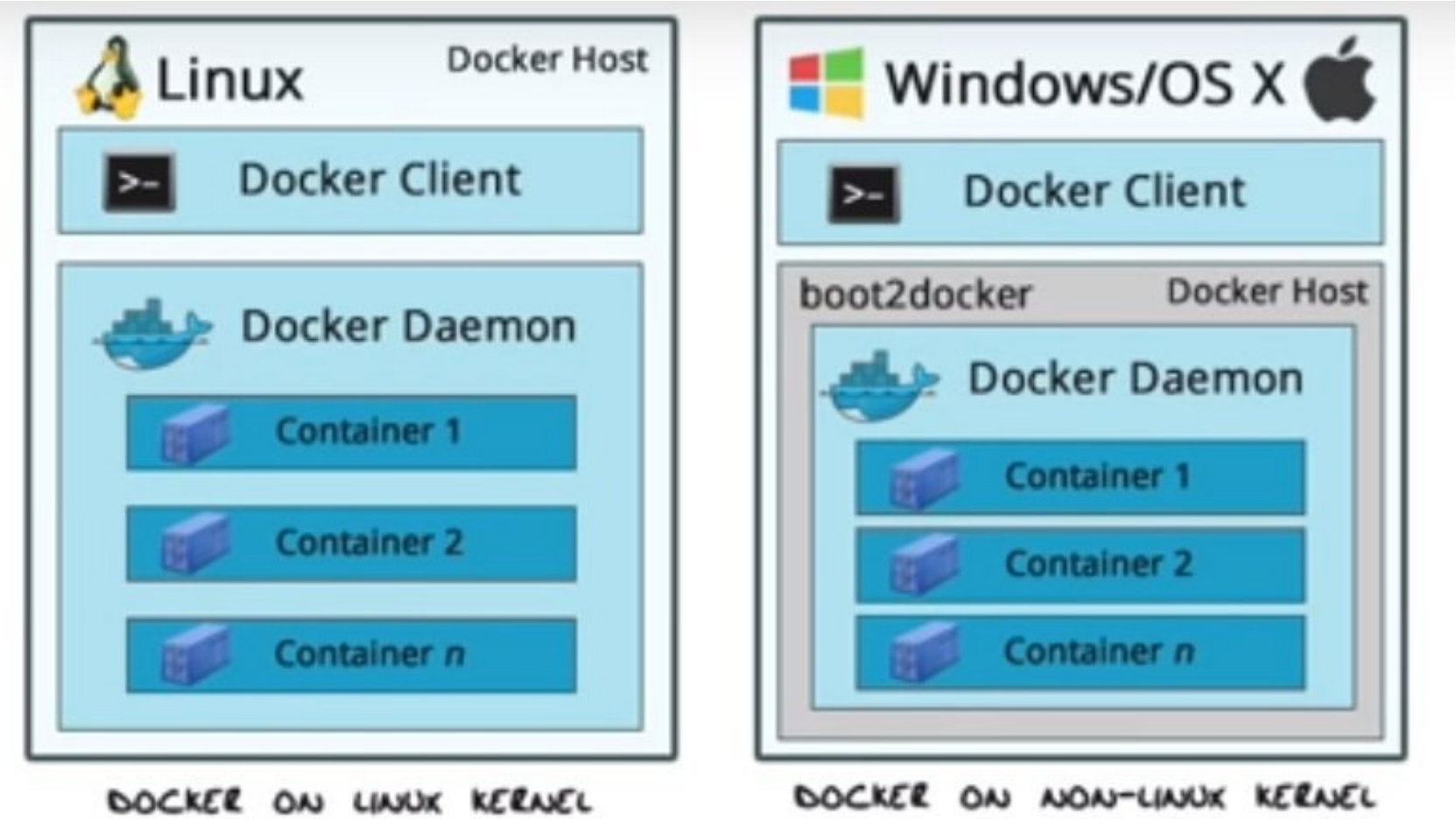
docker

L’architecture de Docker



docker

Docker est aussi disponible sous Windows et Mac OS



docker

Présentation du projet

SUJET DYNAMIQUE MOLECULAIRE
ETUDE DU COMPLEXE sRNP H /ACA

CONTEXTE

Contexte biologique : le projet vise à essayer de comprendre l'impact de certaines mutations dans certains composants du complexe sRNP H/ACA, sachant que l'organisme (l'archaea: Pyrococcus abyssi) vit à température élevée (~90C). Il n'existe pas de structure 3D d'un complexe RNP H/ACA de ce type chez les Eucaryotes et comme pour beaucoup d'études structurales, le système équivalent chez les archaea est utilisé comme modèle. Les complexes sRNP H/ACA catalysent l'isomérisation de l'uridine (U) en pseudo-uridine (ψ); c'est la protéine Cbf5 (celle de plus grande masse moléculaire) qui porte l'activité enzymatique. C'est la modification chimique des ARN la plus abondante dans une cellule (tRNA, rRNA, sRNA, mRNA, etc). Plusieurs rôles structuraux et fonctionnels ont été attribués aux pseudo-uridines notamment dans la stabilisation des double-hélices d'ARN pouvant jouer un rôle dans la stabilisation de certaines structures secondaires de l'ARN, dans la résistance à la dégradation, etc. Les pseudo-uridines sont abondantes dans le ribosome et sont indispensables au fonctionnement du ribosome et donc à la traduction. En dessous d'un certain niveau de modification (induite par un KO des gènes codant pour les ARN guides H/ACA: composant ARN du complexe RNP), le ribosome n'est plus fonctionnel; il existe plusieurs sites de pseudo-uridylations dans le site catalytique du ribosome (site PTC ou peptidyl-transferase center). La protéine est constituée de 5 domaines distincts (**Figure 1**) et nous cherchons à comprendre le rôle de chacun de ces domaines dans l'interaction avec l'ARN.

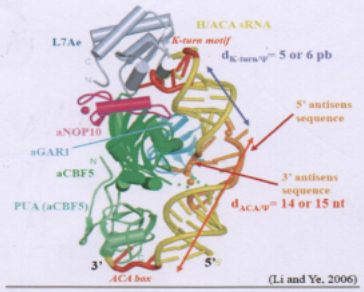


Figure 1 : structure du complexe. Chaque domaine est représenté d'une couleur différente.

Contexte de l'étude : ainsi, deux dynamiques moléculaires de 10ns ont été réalisées afin d'analyser la flexibilité globale et locale des différents composants du complexe RNP H/ACA pour le sauvage et pour des mutants connus pour altérer l'activité. Ici nous analyserons uniquement la dynamique réalisée pour le sauvage. Les temps de contact entre plusieurs résidus de la protéine et certains résidus de la protéine avec l'ARN ont été mesurés au cours de la dynamique moléculaire avec le mutant et ont permis d'identifier des résidus jouant un rôle clé dans l'interaction avec l'ARN (**Figure 2**). On se propose de mesurer les temps de contact entre ces résidus dans la structure sauvage ainsi que la flexibilité des différents domaines de la protéine lors de l'interaction avec l'ARN.

But : **<u>Etudier les changements conformationnels globaux d'une protéine</u>**

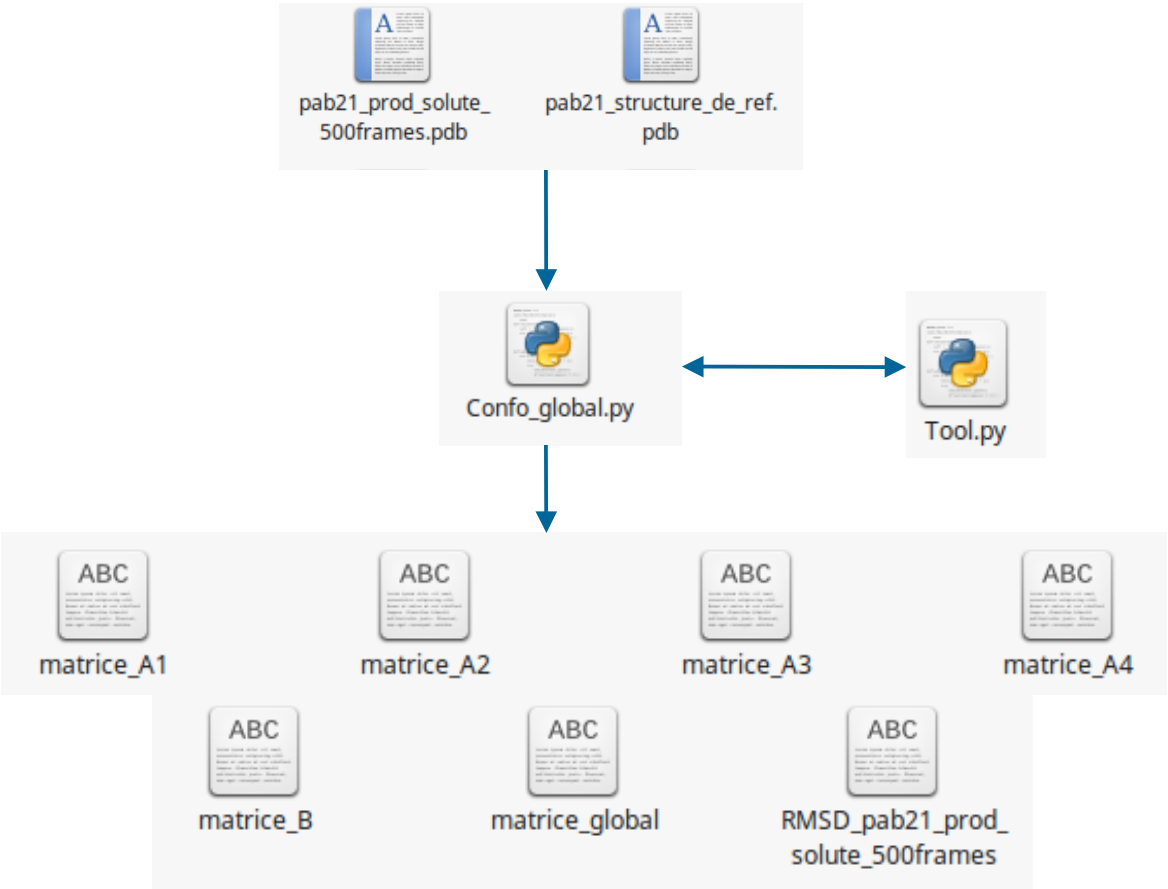
Données:
- 1 fichier avec structure de référence
- 1 fichier avec 500 conformations

Calculs réalisés dans le projet:
- RMSD :
pour chaque domaine séparément ainsi que pour toute la protéine (entre la structure de référence et chacune des conformations)
- RMSD croisés :
pour chaque domaine (entre chacune des conformations)

docker

Présentation du projet

Relation entre les fichiers :



Reprendre le projet

```
coralie@levana:~$ python3 Confo_global.py -ref pab21_structure_de_ref.pdb -conf pab21_prod_solute_500
frames.pdb -domaines: A1,A2,A3,A4,B
Traceback (most recent call last):
  File "Confo_global.py", line 2, in <module>
    import Tool,sys,math,numpy,pylab
ImportError: No module named 'numpy'
coralie@levana:~$
```

Problème: Ne fonctionne plus.

Quel solution pour résoudre ce problème?

docker

Installer Docker

Get Docker CE for Ubuntu | Docker Documentation - Mozilla Firefox

Fichier Édition Affichage Historique Marque-pages Outils Aide

Get Docker CE for Ubuntu | D ✕ Get Started, Part 1: Orientatio ✕

https://docs.docker.com/install/linux/docker-ce/ubuntu/

Most Visited Linux Mint Formation Service d'Authentific... Sysadmin déprimé – ... Liste des groupes BDE 2018 - Google D... Espace « BIMBO (BU... Messages envoyés - ...

docker docs Search the docs Guides Product manuals Glossary Reference Samples Docker v18.09 (current)

Get Docker
- Overview of Docker editions
- Docker CE
  - About Docker CE
  - Cloud
  - Linux
    - CentOS
    - Debian
    - Fedora
    - Ubuntu
    - Binaries
    - Optional Linux post-installation steps
  - MacOS
  - Microsoft Windows

Get Docker CE for Ubuntu

Estimated reading time: 12 minutes

To get started with Docker CE on Ubuntu, make sure you meet the prerequisites, then install Docker.

Prerequisites

Docker EE customers

To install Docker Enterprise Edition (Docker EE), go to Get Docker EE for Ubuntu **instead of this topic**.

To learn more about Docker EE, see Docker Enterprise Edition.

OS requirements

To install Docker CE, you need the 64-bit version of one of these Ubuntu versions:

- Bionic 18.04 (LTS)
- Xenial 16.04 (LTS)
- Trusty 14.04 (LTS)

Docker CE is supported on Ubuntu on `x86_64`, `armhf`, `s390x` (IBM Z), and `ppc64le` (IBM Power) architectures.

> **`ppc64le` and `s390x` limitations**: Packages for IBM Z and Power architectures are only

Edit this page
Request docs changes
Get support

On this page:

- Prerequisites
  - Docker EE customers
  - OS requirements
  - Uninstall old versions
  - Supported storage drivers
    - Extra steps for aufs
- Install Docker CE
  - Install using the repository
    - Set up the repository
    - Install Docker CE
    - Upgrade Docker CE

docker

Installer Docker

set up repository:
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable"

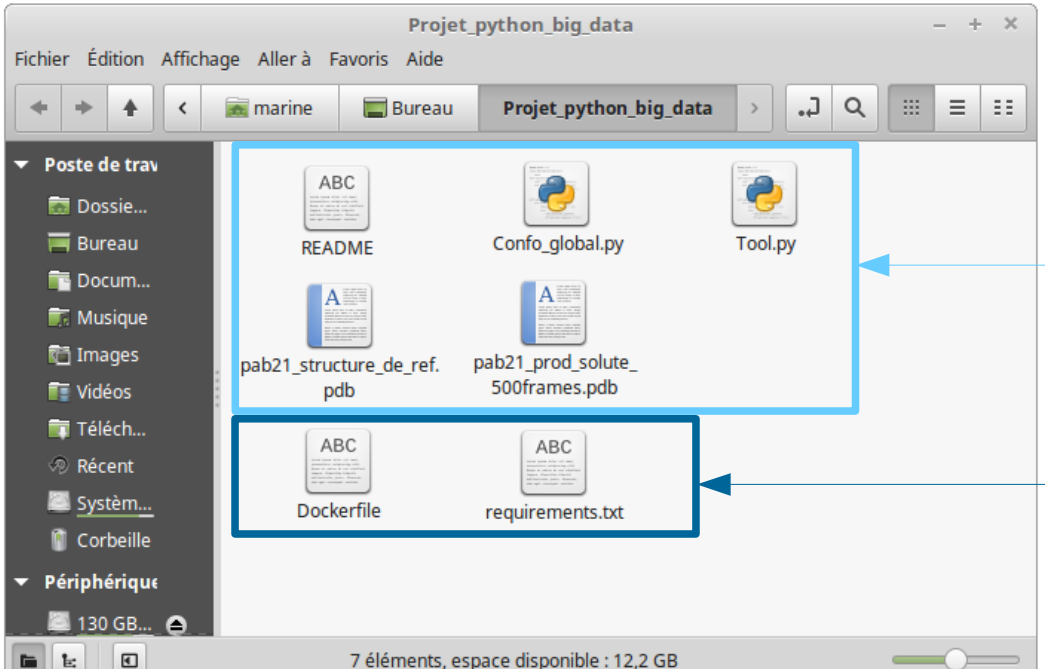
install docker:
sudo apt-get install docker-ce

vérifier l'installation:
sudo docker container run hello-world

update docker:
sudo apt-get update

```
coralie@levana:~$ sudo docker run hello-world

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

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Construction Image : Le DockerFile



Notre Projet

Nécessaires à la construction de l'image

Construction Image : Le DockerFile

Dockerfile (~/Bureau/Projet_python_big_data)

Fichier Édition Affichage Rechercher Outils Documents Aide

Ouvrir Enregistrer Undo

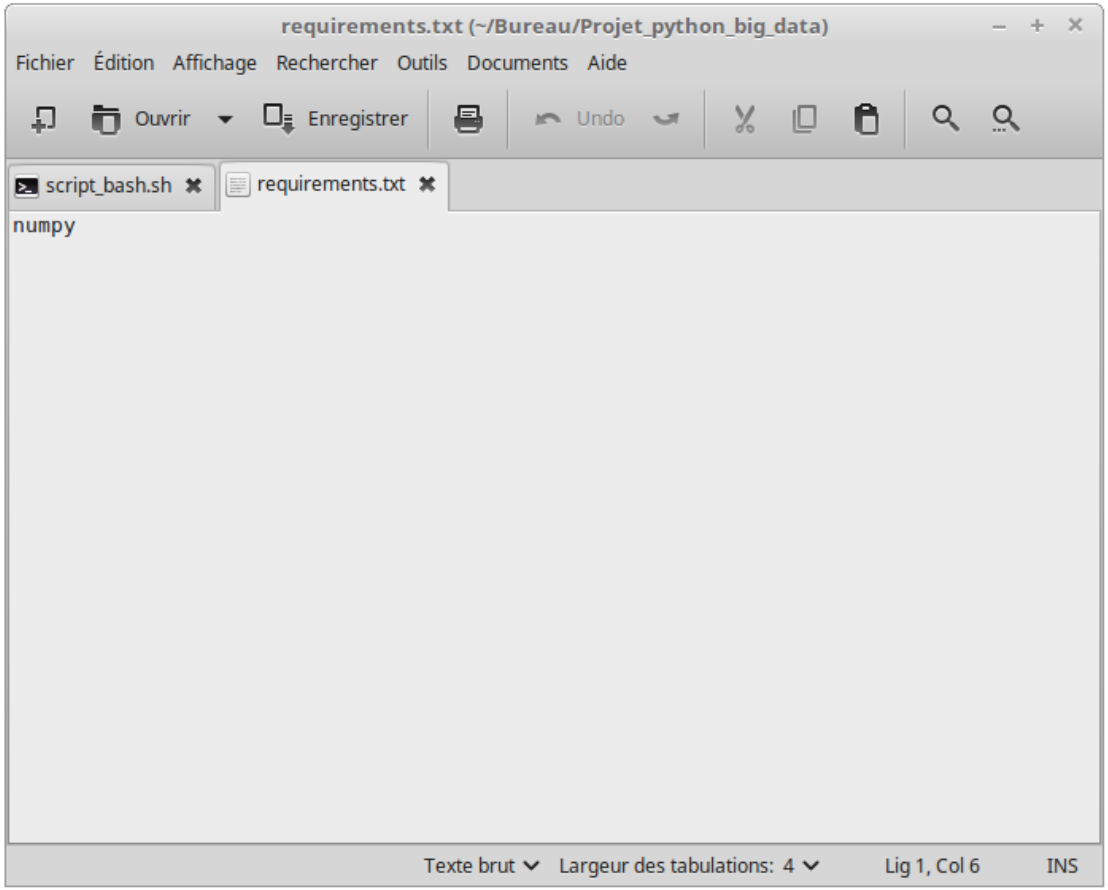
Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:3-slim

# Copy the current directory contents into the container at /Projet_python_big_data
COPY . /Projet_python_big_data

# Set the working directory to /Projet_python_big_data
WORKDIR /Projet_python_big_data

# Run to build the image
RUN pip install --trusted-host pypi.python.org -r requirements.txt
RUN mkdir results

# Run Confo_global.py when the container launches
CMD ["python3", "Confo_global.py", "-ref", "pab21_structure_de_ref.pdb", "-conf",
"pab21_prod_solute_500frames.pdb", "-domaines", "A1,A2,A3,A4,B"]
```

Texte brut Largeur des tabulations: 4 Lig 15, Col 147 INS

docker

Construction Image : Les biliothèques python



Construction Image : Le DockerFile

Dockerfile (~/Bureau/Projet_python_big_data)

Fichier Édition Affichage Rechercher Outils Documents Aide

Ouvrir Enregistrer Undo

Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:3-slim

# Copy the current directory contents into the container at /Projet_python_big_data
COPY . /Projet_python_big_data

# Set the working directory to /Projet_python_big_data
WORKDIR /Projet_python_big_data

# Run to build the image
RUN pip install --trusted-host pypi.python.org -r requirements.txt
RUN mkdir results

# Run Confo_global.py when the container launches
CMD ["python3", "Confo_global.py", "-ref", "pab21_structure_de_ref.pdb", "-conf",
"pab21_prod_solute_500frames.pdb", "-domaines", "A1,A2,A3,A4,B"]
```

Texte brut Largeur des tabulations: 4 Lig 15, Col 147 INS

docker

Construction Image : Le DockerFile

Pré-requis: 1) installer Docker
2) ouvrir un terminal à partir du dossier de projet

Mot clef pour construire

Flag du nom

Appel à Docker

Nom

Dockerfile (~/Bureau/Projet_python_big_data)

Fichier Édition Affichage Rechercher Outils Documents Aide

Ouvrir Enregistrer Undo

Dockerfile

```
# Use an official Python runtime as a parent image
FROM python:3-slim

# Copy the current directory contents into the container at /Projet_python_big_data
COPY . /Projet_python_big_data

# Set the working directory to /Projet_python_big_data
WORKDIR /Projet_python_big_data

# Run to build the image
RUN pip install --trusted-host pypi.python.org -r requirements.txt
RUN mkdir results

# Run Confo_global.py when the container launches
CMD ["python3", "Confo_global.py", "-ref", "pab21_structure_de_ref.pdb", "-conf",
"pab21_prod_solute_500frames.pdb", "-domaines", "A1,A2,A3,A4,B"]
```

Texte brut Largeur des tabulations: 4 Lig 15, Col 147 INS

marine@marine-pcfixe ~/Bureau/Projet_python_big_data

Fichier Édition Affichage Rechercher Terminal Aide

```
marine@marine-pcfixe ~/Bureau/Projet_python_big_data $ sudo docker build -t projetpython .
[sudo] password for marine:
Sending build context to Docker daemon  392.5MB
Step 1/6 : FROM python:3-slim
 ---> d53702345644
Step 2/6 : COPY . /Projet_python_big_data
 ---> 7abbc38ca4f6
Step 3/6 : WORKDIR /Projet_python_big_data
 ---> Running in dcd9c89966c6
Removing intermediate container dcd9c89966c6
 ---> 4749eb57c77b
Step 4/6 : RUN pip install --trusted-host pypi.python.org -r requirements.txt
 ---> Running in faead86b07f3
Collecting numpy (from -r requirements.txt (line 1))
  Downloading https://files.pythonhosted.org/packages/38/39/f73e104d44f19a6203e786d5204532e2
14443ea2954917b27f3229e7639b/numpy-1.15.4-cp37-cp37m-manylinux1_x86_64.whl (13.8MB)
Installing collected packages: numpy
Successfully installed numpy-1.15.4
Removing intermediate container faead86b07f3
 ---> 971d2c5f5eb3
Step 5/6 : RUN mkdir results
 ---> Running in db22e36ca911
Removing intermediate container db22e36ca911
 ---> a168b5065eb0
Step 6/6 : CMD ["python3", "Confo_global.py", "-ref", "pab21_structure_de_ref.pdb", "-conf",
 "pab21_prod_solute_500frames.pdb", "-domaines", "A1,A2,A3,A4,B"]
 ---> Running in a18c39ff41ac
Removing intermediate container a18c39ff41ac
 ---> 6fdb353ab520
Successfully built 6fdb353ab520
Successfully tagged projetpython:latest
marine@marine-pcfixe ~/Bureau/Projet_python_big_data $
```

Le partage : Upload de l'image sur le hub

Pré-requis: s'être connecté avec son compte sur le Hub ("sudo docker login"):

Mot clef pour tagguer

Nom de l'image en local

Nom d'utilisateur

Nom de l'image sur le hub

Version

Mot clef pour uploader

marine@marine-pcfixe ~/Bureau/Projet_python_big_data

Fichier Édition Affichage Rechercher Terminal Aide

```
marine@marine-pcfixe ~/Bureau/Projet_python_big_data $ sudo docker tag projetpython ami2b/projetpython:latest
marine@marine-pcfixe ~/Bureau/Projet_python_big_data $ sudo docker push ami2b/projetpython:latest
The push refers to repository [docker.io/ami2b/projetpython]
fa682a7045d6: Pushed
1eb086f71600: Pushing [===========>                                        ]   17.7MB/77.73MB
a1f0335725bf: Pushing [========>                                           ]  65.17MB/392.5MB
e12a840389c5: Mounted from ami2b/projet_python
53cd71e6dcb4: Mounted from ami2b/projet_python
ca4d1ce0e66c: Mounted from ami2b/projet_python
03a6b6877a9b: Mounted from ami2b/projet_python
ef68f6734aa4: Mounted from ami2b/projet_python
```

Screen terminal

Utiliser l'image

Exécuter le script dans le container:

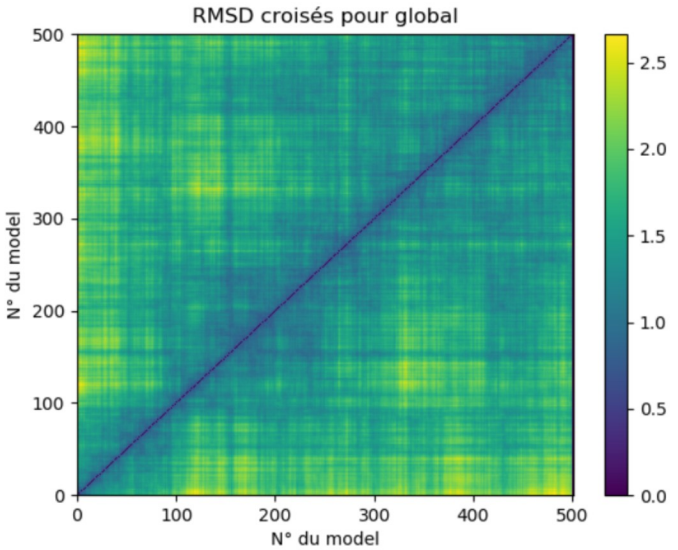
```
coralie@levana:~$ sudo docker run -it ami2b/projetpython:latest
Unable to find image 'ami2b/projetpython:latest' locally
latest: Pulling from ami2b/projetpython
a5a6f2f73cd8: Already exists
3a6fba040982: Already exists
2bbbf89a5531: Already exists
2f40a0d89689: Already exists
46b856c7fde9: Already exists
7cbb8978734e: Pull complete
8c40434aca45: Pull complete
48161482604b: Pull complete
Digest: sha256:a222a5ce6d3479b0e050628673ec7ae02764756ddbb6bb8a4505aa5013ff2c4b
Status: Downloaded newer image for ami2b/projetpython:latest
coralie@levana:~$
```

Récupérer les résultats:

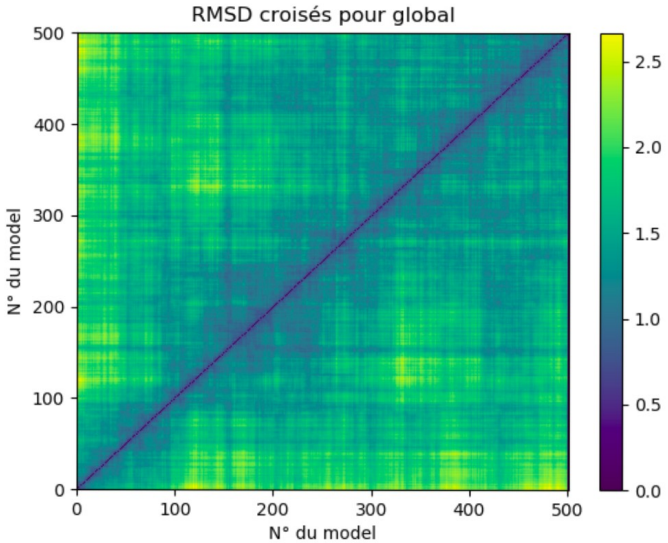
```
coralie@levana:~/BigData$ sudo docker container ls -a
CONTAINER ID        IMAGE                        COMMAND                  CREATED             STATUS
b0beb66e5ff1        ami2b/projetpython:latest    "python3 Confo_glo..."   17 hours ago        Exited (0) 17 hours ago
877e8cceefcf        ami2b/projetpython:latest    "python3 Confo_glo..."   22 hours ago        Exited (0) 18 hours ago
0a8580a92dc4        ami2b/projet_python:latest   "python3 Confo_glo..."   22 hours ago        Created
fa389c281fc9        ami2b/projet_python:latest   "python3 Confo_glo..."   3 days ago          Exited (0) 3 days ago
ace36144856b        hello-world                  "/hello"                 2 weeks ago         Exited (0) 2 weeks ago
coralie@levana:~/BigData$ ls
coralie@levana:~/BigData$ sudo docker cp b0beb66e5ff1:./Projet_python_big_data/results /home/coralie/BigData
coralie@levana:~/BigData$ ls
results
coralie@levana:~/BigData$
```

Reproductibilité des résultats

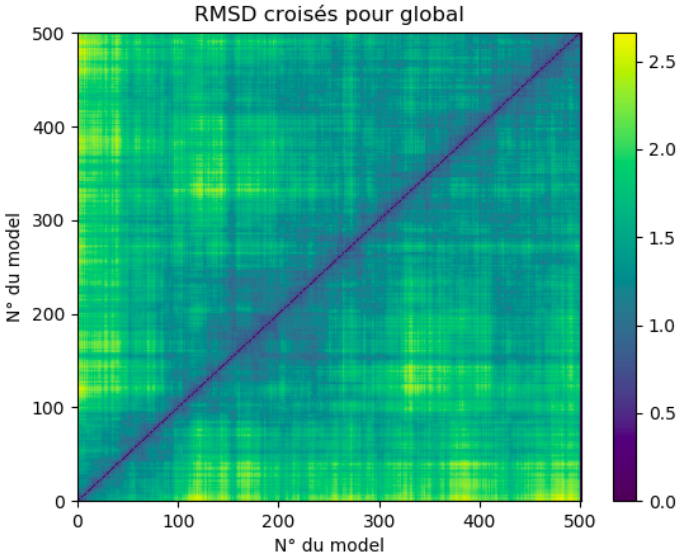
Heat map du projet initial



Obtenu via Docker par Marine



Obtenu via Docker par Coralie



Résultats identiques

Script bash

On peut faire tourner docker dans un script bash :
donc automatisation de l'utilisation de docker possible

script_bash.sh (~/Bureau/Projet_python_big_data)

Fichier Édition Affichage Rechercher Outils Documents Aide

Ouvrir Enregistrer Undo

script_bash.sh

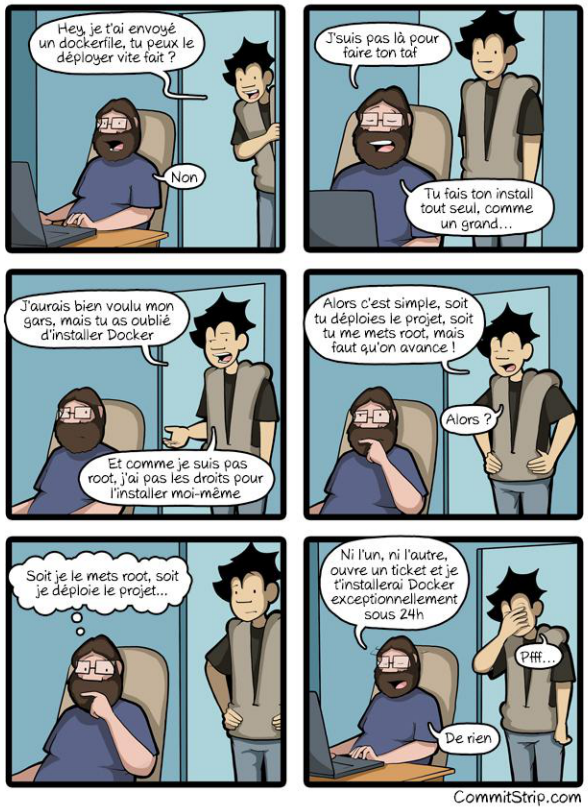
```
#lancement de l'image ami2b/projetpython:latest (en provenance du Hub) dans un conteneur
sudo docker run ami2b/projetpython:latest

#récupération du nom du conteneur de l'image ami2b/projetpython:latest
var=`sudo docker container ls -a | cut -d\  -f 1 | sed -n "2 p"`

#récupération du dossier results (du conteneur) sur dans le dossier Bigdata (en local)
sudo docker cp $var:./Projet_python_big_data/results /home/coralie/Bigdata
```

sh Largeur des tabulations: 4 Lig 7, Col 76 INS

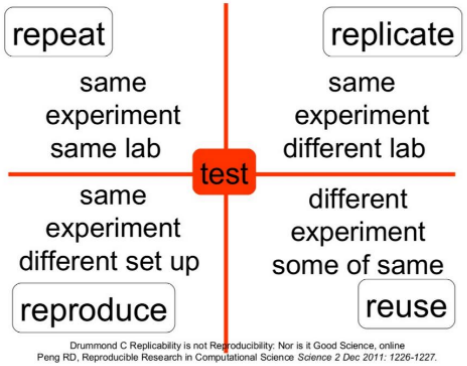
Retour d’expérience

- Léger
- Installation hyper facile
- Tutoriel de départ bien expliqué
- Facile à utiliser pour des scripts simples
- Portabilité
- Reproductibilité

- Obligé d’être sudo tout le temps
- Beaucoup utilisé pour la conception d’application web, et peu d’autres exemples



La PUISSANCE de Docker



- Repeat
  - *Redo*: exact same context
  - Same workflow, execution setting, environement
  - Same *output*

  →Aim = proof for reviewers ☺

Source: D'après le cours d'Intégration de données et BigData de Sarah Cohen-Boulakia

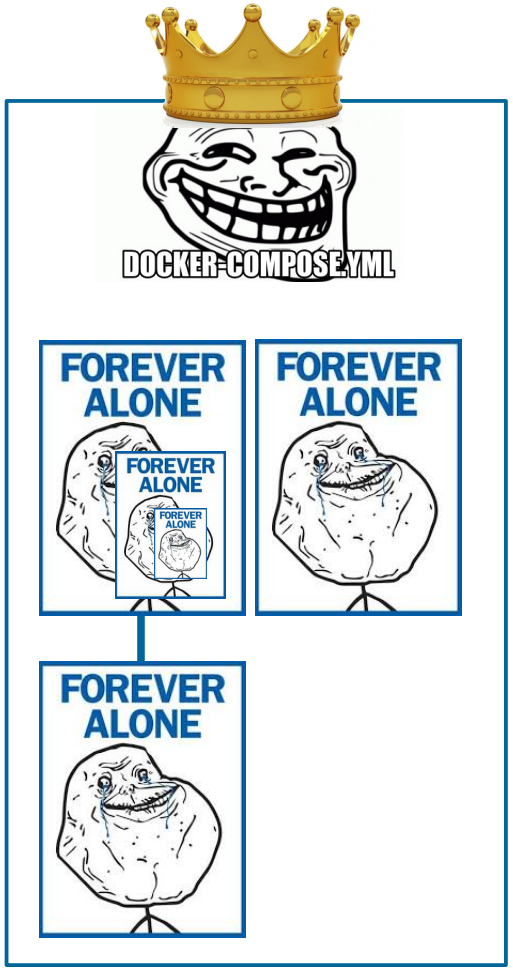
Fichier docker-compose.yml :

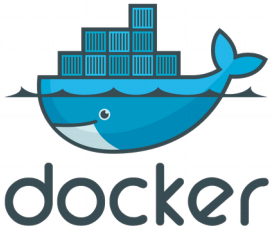
- En parallèle
- En séries

} - avec la même image ou non
- partage des données

Conteneur-ception

Docker Swarm :
Utilisation de plusieurs machines et partage des ressources, gestion des pics de charge.

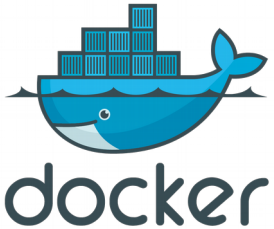


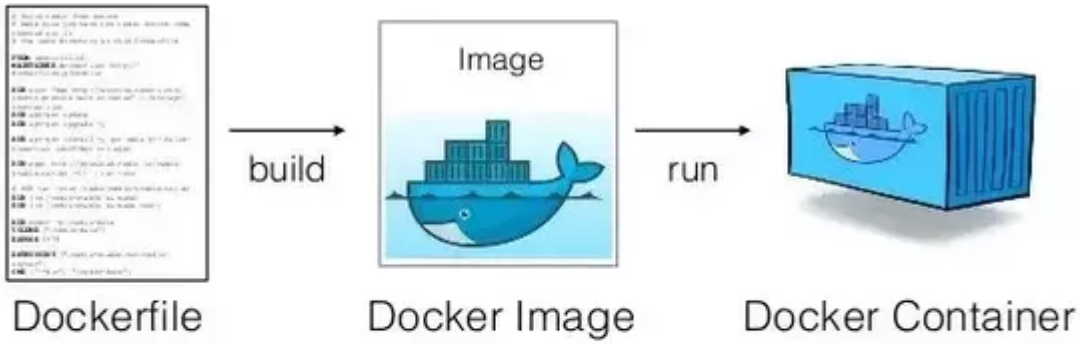


Autour de Docker

BioContainer

High Performance Computing



Merci de votre attention !

Single Container Docker Workflow