DOCKER FOR REPRODUCIBLE AND SHAREABLE SCIENCE

June Sallou, june.benvegnu-sallou@irisa.fr

https://github.com/Jnsll/ModelisationScientifique

Some challenges (cf Challenges of SciModelling)

- ✘ **Scalability**
- ✘ Interactivity
- ✘ Collaboration
- ✘ Version control
- ✘ **Reproducibility**

Common challenges

- Needing different versions of the same library for different projects
- Losing track of the required libraries for a specific project
- Requiring a different Python versions
- Setting up projects on a new team member's machine (which could have a different OS as well)
- Automating deployment
- Control of environments (version control)

How to tackle them ?

Docker (containers)

Docker ? What's that ?

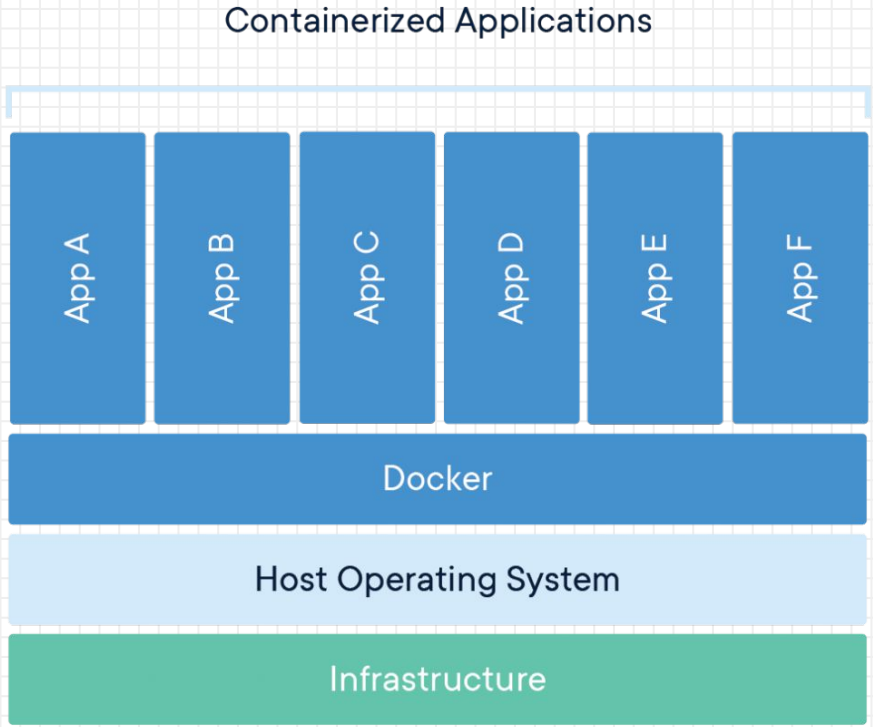
Docker Container is stand-alone software that contains both application code and its dependencies, which can run in any platform smoothly. The motto of Docker is to **build once anywhere and run anywhere**.

Docker

Docker allows us to manage the following dependencies in a single place:

- OS dependencies
- CLI tools dependencies
- Python dependencies

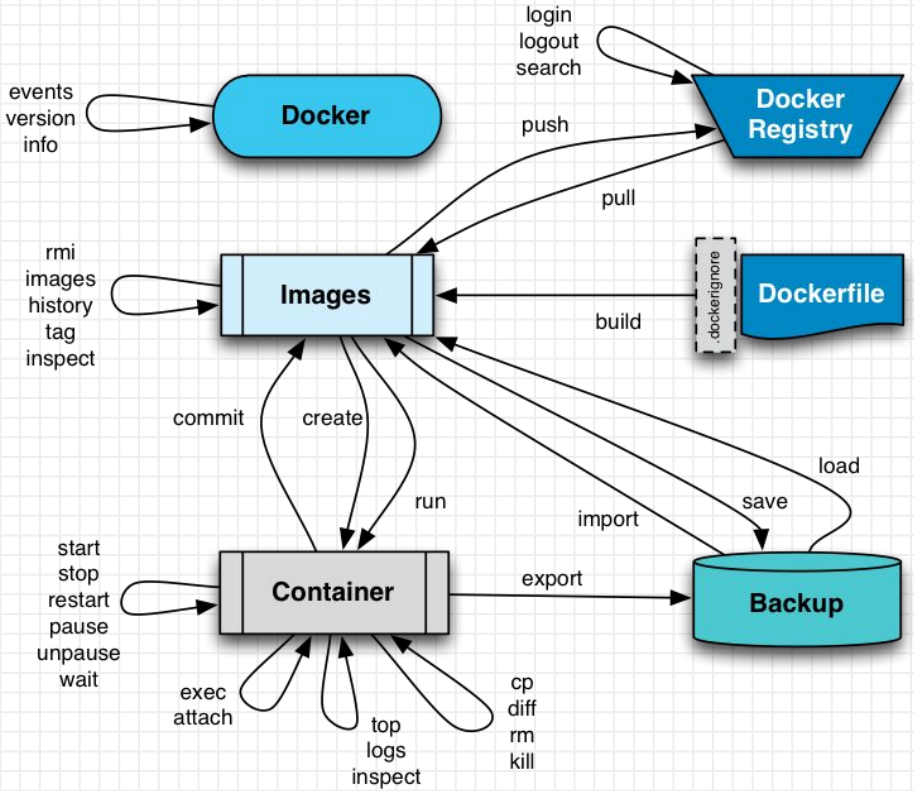
Docker


Containerized Applications
App A
App B
App C
App D
App E
App F
Docker
Host Operating System
Infrastructure

Docker

You can use Docker to create an **image**, run it as a **container,** and ship it anywhere. You can use a container registry service like **Dockerhub** for storing application images, and it integrates with Bitbucket and Github, where you can host **Dockerfile**.

Overview of the commands



Docker Image

> Using a Docker Image from the DockerHub

https://hub.docker.com/

>> docker pull jupyter/datascience-notebook

Create a Docker Image

> Using a Dockerfile

```
FROM ubuntu:xenial
RUN apt update \
 && apt-get install -y python3-pip python3-dev
ADD myFile <LocationInTheContainer>
CMD bash myProcess.sh
```

Create a Docker Image

>> docker image build Dockerfile

>> docker build . -t my_project_image

>> docker images -all

Run a Docker Image

>> docker run -p 8888:8888 <image-name> --name <name>

#Show the containers

>> docker ps -a

Interactive connection to the container

>> docker exec -it <container_name> bash

Stop a Docker container

>> docker stop <Container-name>

Remove a Docker container

>> docker rm <Container-name>

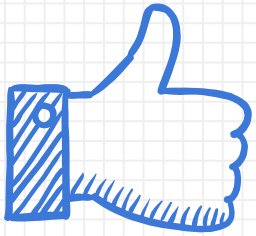
Remove a Docker image

>> docker rmi <image-container-id>

References

- https://blog.datasciencedojo.com/data-science-with-docker-containers/
- https://blog.cnvrg.io/docker-for-machine-learning-and-reproducible-data-science
- https://www.docker.com/products/container-runtime
- https://www.thoughtworks.com/insights/blog/reproducible-work-environments-using-docker

Credits Template : https://www.slidescarnival.com/friar-free-presentation-template/425



THANKS!

Any questions?

Activité 1

10-15 minutes

Installez Docker sur votre machine, à l'aide de ce lien:
https://docs.docker.com/get-docker/

Suivez les différentes instructions (dépend de votre OS)

Sous linux, vous pouvez tester que docker
est correctement installé avec la commande suivante:
"sudo docker --version"

Activité 2

5 minutes

Pull et run le container llesoil/chiffre:latest

Quel chiffre obtenez-vous?

Activité 3

5 minutes

Trois erreurs dans ce Dockerfile empêchent le build de terminer…

Saurez-vous les retrouver?

```
1 FROM debian:9
2 RUN apt update
3  && apt-get install python3-dev python3-pip
4 ADD monScript.py
5 CMD python3 monScript.py
```

Activité 3

5 minutes

```
1 FROM debian:9
2 RUN apt update \
3  && apt-get install -y python3-dev python3-pip
4 ADD monScript.py /
5 CMD python3 monScript.py
```

Activité 4

5 minutes

Build le Dockerfile de l’activité 3

```
1 FROM debian:9
2 RUN apt update \
3  && apt-get install -y python3-dev python3-pip
4 ADD monScript.py /
5 CMD python3 monScript.py
```