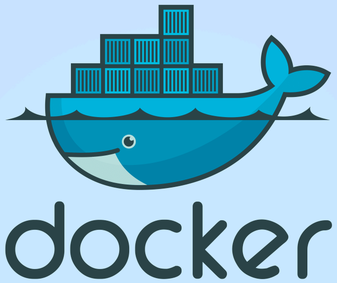


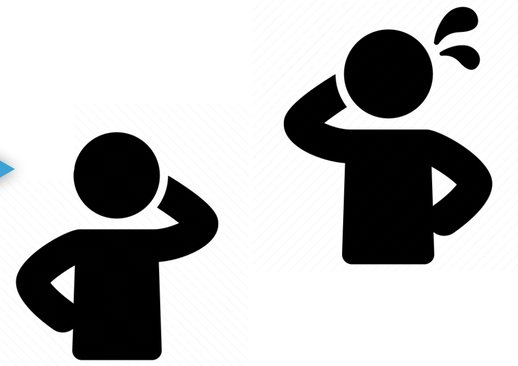
Docker : Introduction and basics

Amine FERDJAOUI
amine.ferdjaoui@sogeti.com

Année universitaire 2025/2026

Why Docker ?

The application works fine on my computer but not on another ?
The application is not rendered the same in your computer and mine ?



- **Some reasons :**
  - Missing/Mismatching libraries
  - Mismatching programming langage versions

Working locally

When you start a project locally you should always work with **venv (or conda)**.

To initialise a virtual environment :

```
$ python -m venv venv
```

To start it :

Unix (mac or linux)

```
$ source venv/bin/activate
```

On windows

```
$ .\venv\Scripts\activate
```

To exit :

```
(venv) $ deactivate
```

Why Docker ?

When you start a project locally you should always work with **venv (or conda)**.

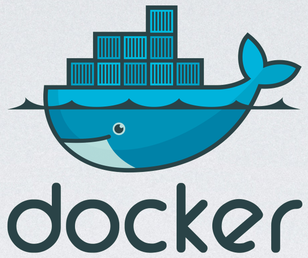
But in the cloud how can you manage and push your solution(s) ?

Application

Python 3.8
TF 2.0.1
Sklearn 1.1
Coclust 0.2

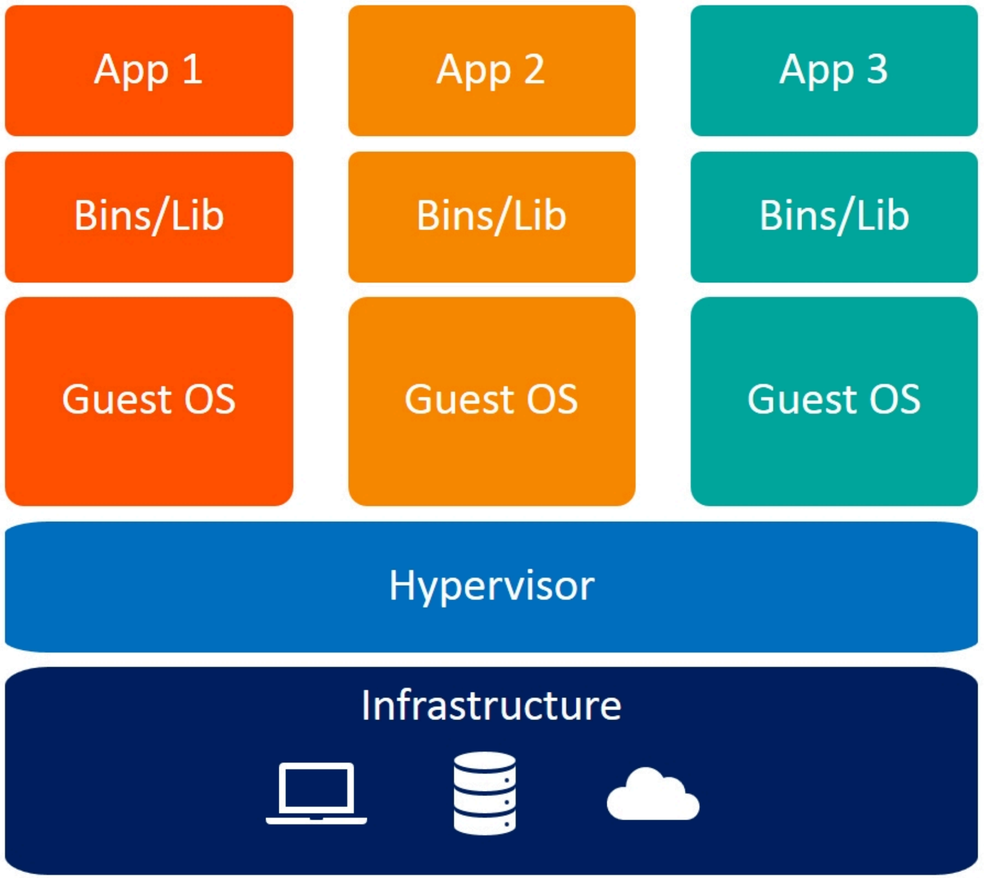
Application 2

Python 3.6
TF 1.5
Sklearn 0.8
Coclust 0.2

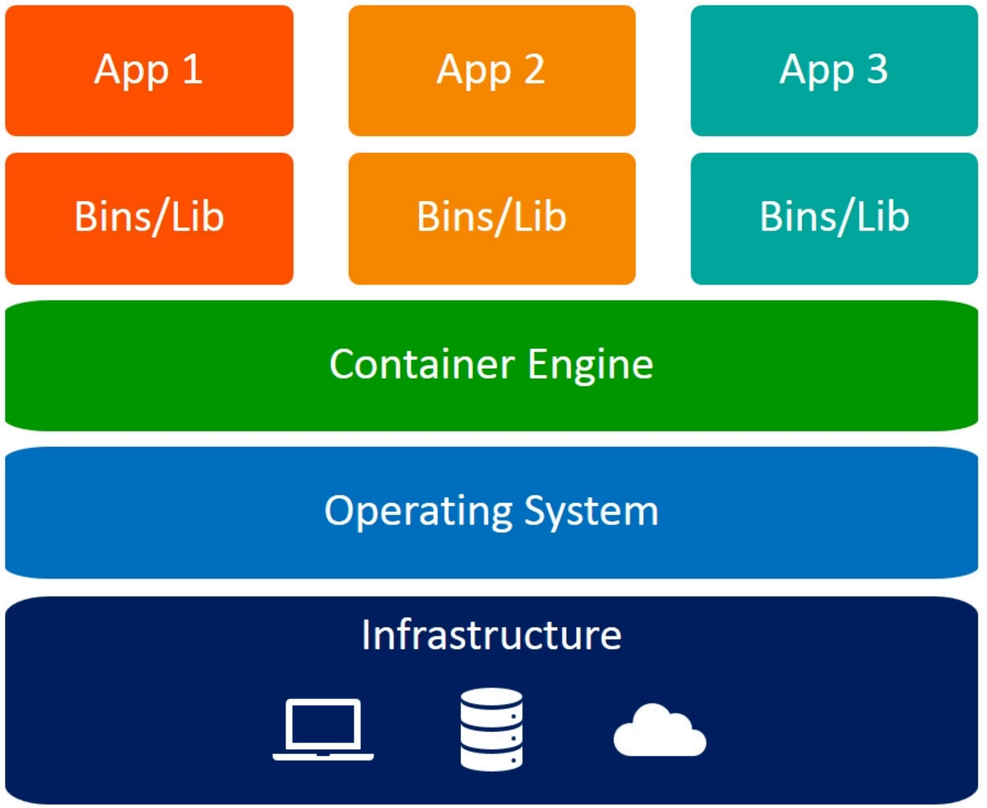


- Docker is a platform for Building, running and shipping apps.
- It is the easiest solution to deploy applications (ML models) to production.
- Each application runs in an isolated environment (no need for any installation).
- We can remove all dependencies with one click.

Docker vs Virtual machines



Virtual Machines



(or docker)

Containers

Docker vs Virtual machines

VMs

Need a separated OS

Very slow

Need lot of ressources

Docker containers

Use host's OS

Light and fast

Can run multiple isolated applications

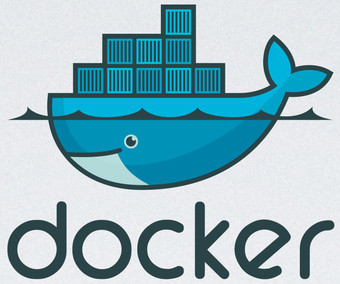
Don't consume lot of memory

Deployment and testing (once it works you can run it anywhere)

Installation

- Go to docs.docker.com/get-docker and create an account and ***download from Docker Hub.***
- Make sure the docker logo appears in status bar each time you want to use docker.
- Once docker installed and launched, test it using :

```
$ docker --version
```



DOCKER IMAGES AND CONTAINERS

How does it work ?

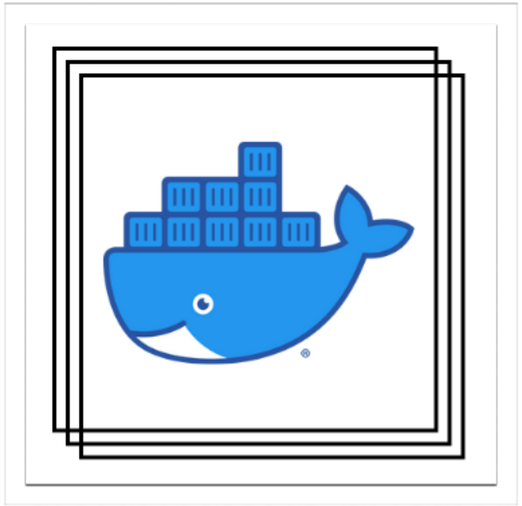
Application

YOLO V8
Flask

Dockerfile

Build

Docker image
(Snapshot)



- Development environment (eg. python)
- Copy of the files
- Libraries
- Commande(s) to be executed

How does it work ?

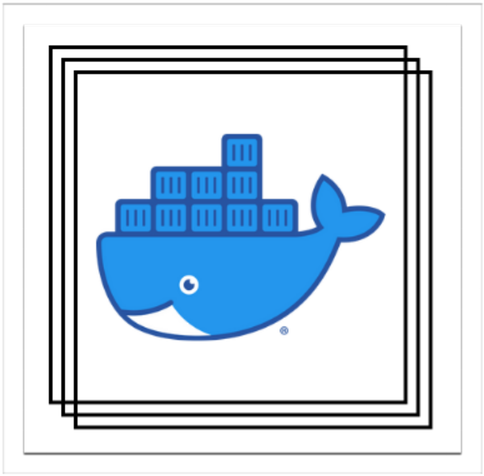
Application

YOLO V8
Flask

Dockerfile

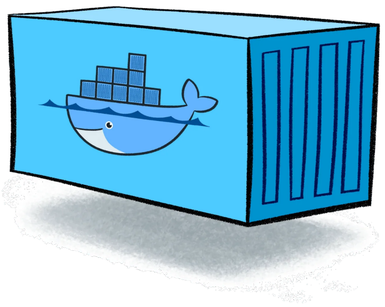
Build

Docker image

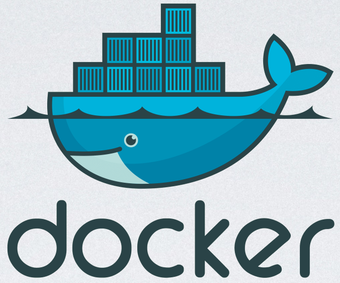


Run

docker container
(Isolated environment)



Runs its own software, binaries, and configurations



DOCKERFILE

https://docs.docker.com/engine/reference/builder/

Dockerfile

- Start from a base image with the command : **FROM**
- Base images can be Linux, or langages (installed on top of a linux) published on docker hub (See hub.docker.com) based on different versions of linux. Alpine (or slim) is preferable for docker images (Cf. https://www.alpinelinux.org/about/)
- Copy files into the image using the command **COPY**
- Execute a command using the command **CMD**

Dockerfile

https://github.com/AmineFrj/flask-backend

Dockerfile

```
1 FROM python:3.8-alpine #or lts-alpine or latest-alpine or slim
2 COPY . /app
3 WORKDIR /app #the commands will be executed within this directory (like cd command)
4 RUN pip install -r requirements.txt
5 CMD python app.py
```

Nb. Try to always specify the version of the base image inside Dockerfile

Build and run an image

```
$ docker build -t <image_name>:<tag> . # -t flag is used to tag the image
$ docker run --name <container_name> <image_name>:<tag> # --name flag is used to name the container
```

```
$ docker images #or docker image ls
$ docker ps # to list the running containers
$ docker ps -a    # to list all containers
$ docker ps --help
```

```
$ docker build -t mon-premier-docker .
$ docker images
$ docker ps -a
$ docker run -p 5000:5000 mon-premier-docker
```