Ekimetrics x Datacraft

Quand la baleine part dans les nuages

11/06/2021

datacraft

Ekimetrics.

Data science for business

PARIS | LONDON | NEW YORK | HONG KONG | DUBAI

Agenda.

00. Overview
01. Why Should I care ?
02. DE Overview
03. Let’s go !
04. Training resources
04. Questions ?

What are we going to do ?

MyApp.py

```python
import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!*
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```

Streamlit

docker

Local container

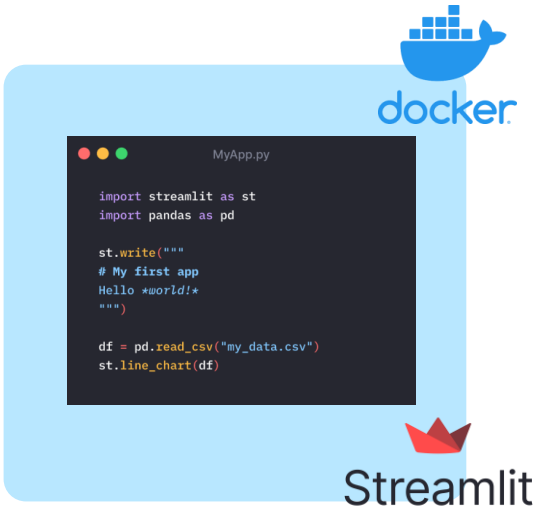
alternatives

plotly

FastAPI

MKDOCS

What are we going to do ?



Local container

Cloud container

alternatives

Google Cloud aws

What do we need ?

Python

docker desktop

Github account

docker hub

Dockerhub account

git

Vs Code

Azure Account

05.
Streamlit

Streamlit : Data App made easy

Streamlit

https://streamlit.io
Created : October 2019

Python library (partly open-source)
Machine Learning & Data Science Apps

Code First & Low code

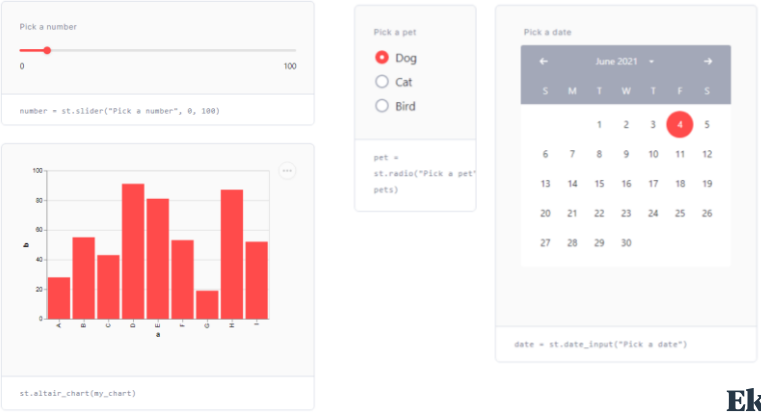
Keeping It Super Simple

MyApp.py

```
import streamlit as st
import pandas as pd

st.write("""
# My first app
Hello *world!*
""")

df = pd.read_csv("my_data.csv")
st.line_chart(df)
```

Interactive by design

Native sliders, box, calendar and more..

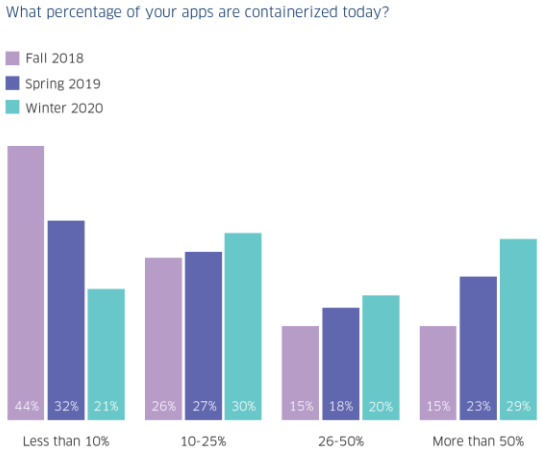


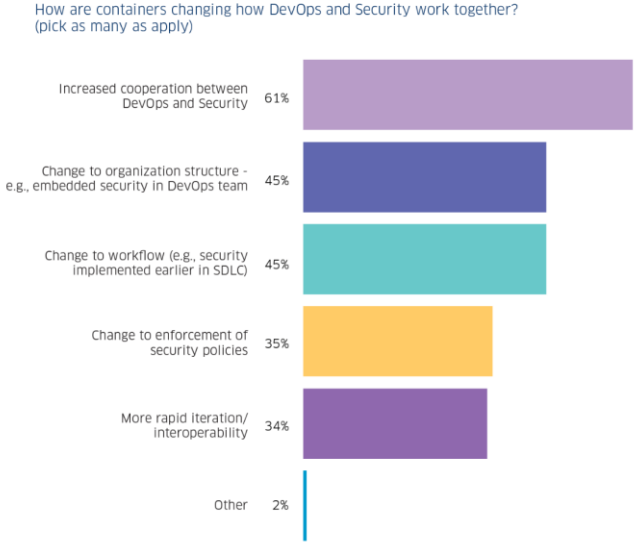
Ekimetrics.

01.
Docker, why should I care ?

Everywhere

"By 2022, more than 75% of global organizations will be running containerized applications in production, up from less than 30% today." –Gartner , June 2020



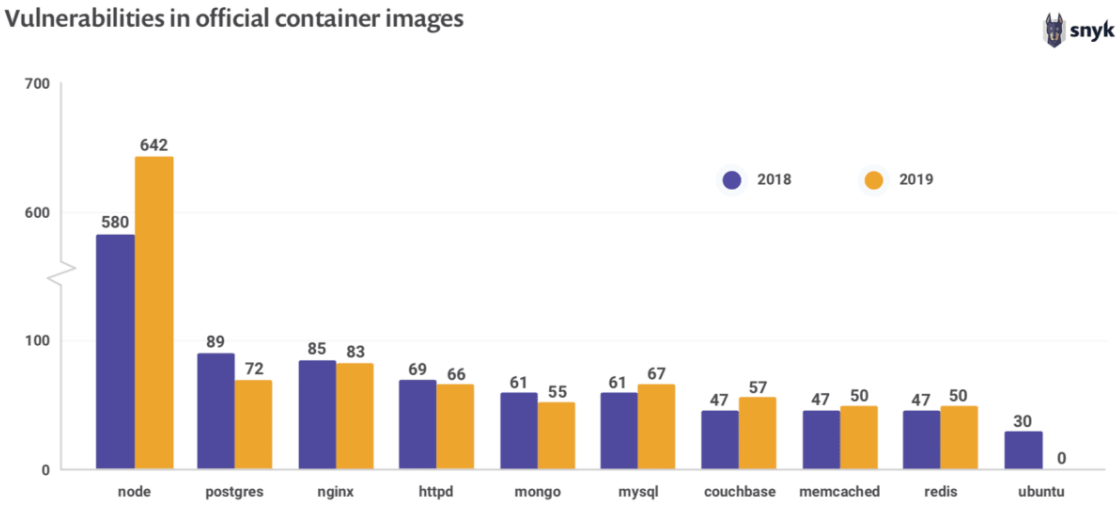


source: stackrox

Ekimetrics.

What about security ?

Decrease of “native” vulnerabilities



Rootless mode

introduced in Docker Engine v19.03 and graduated from experimental in Docker Engine v20.10. in Dec 2020.

Eki.Mission Example

GROUPE BPCE

docker

Dash

Passed Security Audit

Why for Data Science ?

Reproducibilty

Easily isolate and reproduce Data Science scenarios so that your colleagues obtains the same result on their end, whatever laptop they have.

Portability

No more friction when moving your PoC out of your laptop when more power is required. No need to recreate your env on the cloud, the container will handle it.

Scalability

With orchestration tools such as Swarm, Kubernetes, Kubeflow... you can scale you container to handle heavy loads.

01.
Some basic theory

Definitions

Dockerfile // The Recipe

Text file containing the set of instructions to build the docker image.

Docker build // The Cooking

Building an image from the Dockerfile.

Docker image // The Dough

A template file resulting of a docker build. If you run it, you create a container.

Docker run // The Baking

Run a command in a new container using a docker image.

Docker Container // The Cake

Set of isolated processes running on its own space on a shared kernel.

Docker client // You

Tool to interact with Docker.

Docker registry // Cake Shelf

Minimal collection of Docker layers

Docker hub // Cake Shop

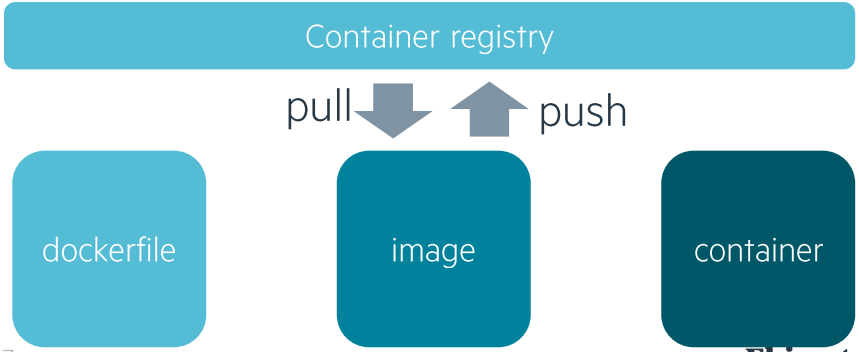
Centralized collection of Docker images, ready to be pulled. See it as GitHub for containers.

Docker pull // get a cake from the shelf

Like a git clone, get an image from a container registry

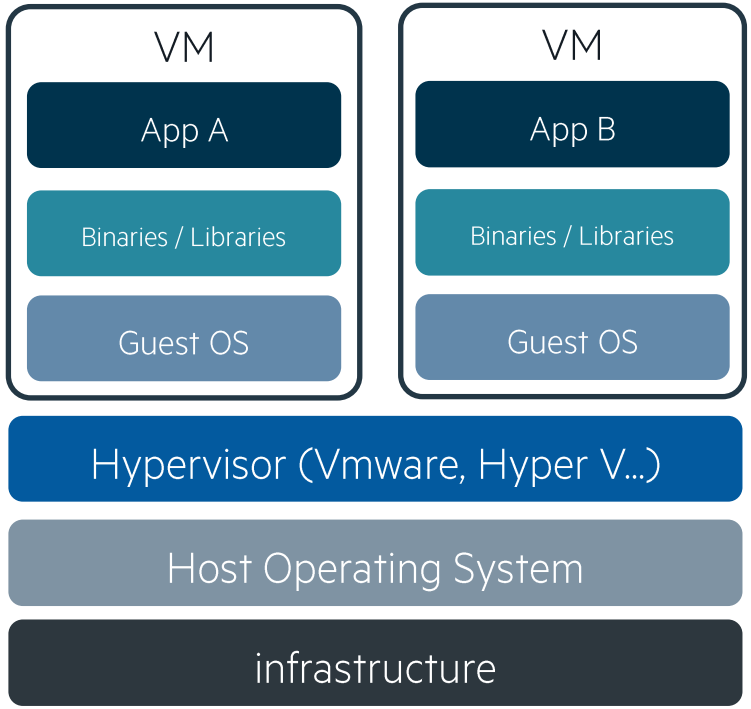
Docker push // put a cake on a shelf

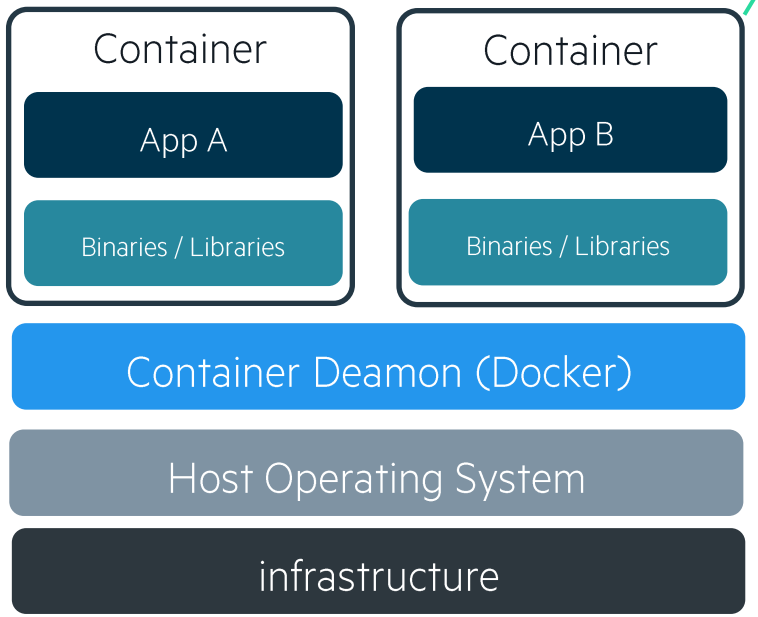
Like a git push, send an image to a container registry



Ekimetrics.

Container ≠ Virtual Machine





source: docker.com

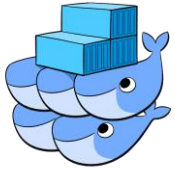
Advantages of containers vs Virtual Machines

Portability

A container is like a self-sufficient package, with all its environment and dependencies shipped together as a single unit and can easily be moved around (on-premise or cloud)

Scalable

Containers can be “orchestrated” by tools such as Docker Swarm or Kubernetes that automate scaling, networking and deployment.



Maintanability

By separating every service in a container, you can easily update a container without impacting the other one

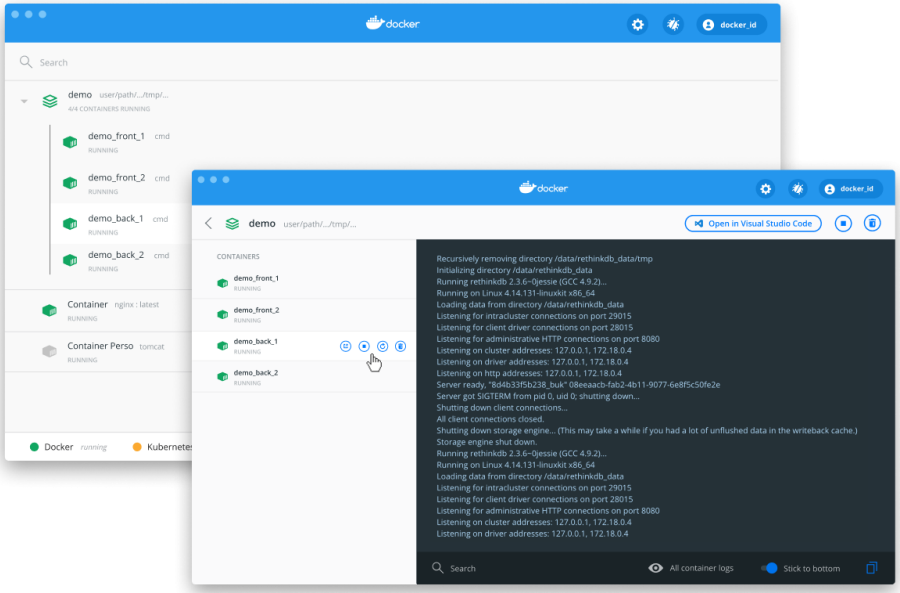
Simple example

```
docker run -dp 80:80 docker/getting-started

docker run -dp 80:80 hello-world
```

Common Docker commands

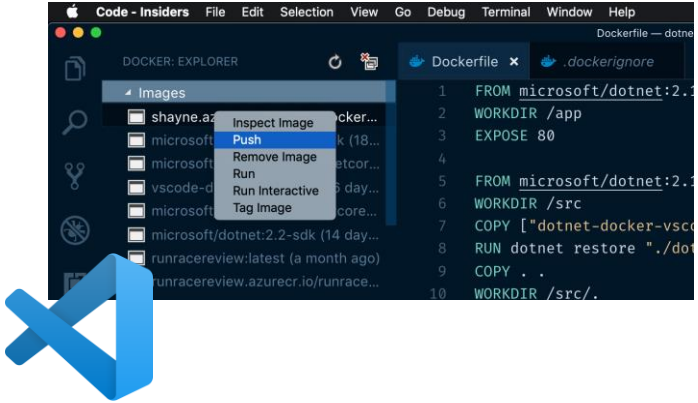
```
docker image ls
docker image rm image_name:version
docker build –t image_name:version

docker container run --name container_name -p 80:80 image_name:version

docker pull image_name:version
docker tag image_name:version axelr/new_name:version
docker push axelr/new_name:version
```

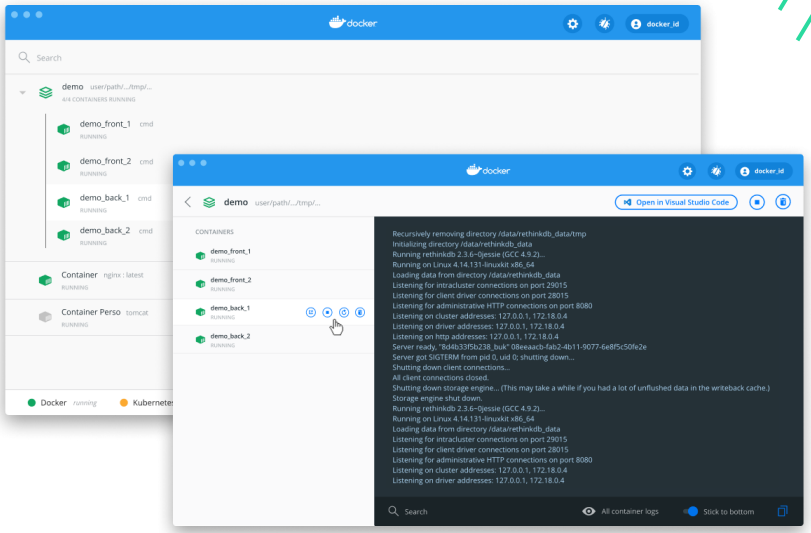
Common Docker commands



Most commands are available with the Docker Desktop UI

Common Docker commands





Most commands are available within VS Code and/or the Docker Desktop UI

Ekimetrics.

05.
Let’s go

Getting started

- git clone git@github.com:EatZeBaby/datacraft-workshop-docker.git
- Start Docker Desktop
- code ./datacraft-workshop-docker

Building our first Dockerfile

Chosing a base image is the first step of a Dockerfile = base linux distribtion

```
FROM python:3.7
```

It's an image with python 3.7 preinstalled

Setting Working directory different from root

```
WORKDIR /app/
```

Installing the requirements for your app in a dedicated command (for caching purpose keep this method)

```
ADD ./requirements.txt /app/requirements.txt
RUN pip3 install -r requirements.txt
```

Copy everything necessary for the app to the image

```
ADD datacraft.py  /app/datacraft.py
ADD css/ /app/css/
ADD utils/ /app/utils/
ADD models/ /app/models/
ADD images/ /app/images/
```

Run the app !

```
CMD ['streamlit', 'run' ,'/app/datacraft.py']
```

Build & Run your first container

- docker build -t dockercraft .
- docker run -d -p 80:8501 dockercraft
- Go to http://localhost/

- Update your code

- Refresh your web page

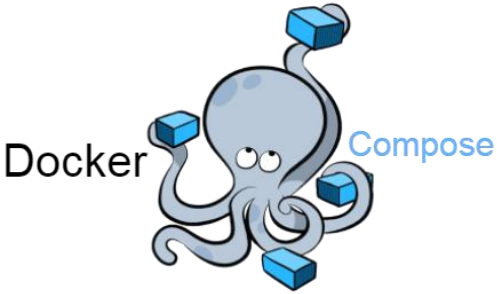
- What's wrong ? ☺

Docker Volumes

- The docker container is isolated from our source code.
- Docker volumes allows to create a link between local files and container files

- docker volume create --name app_volum
- docker run –p 8050:8050

Docker Compose




```yaml
version: "3.8"
services:


  app:
    build:
      context: .
    container_name: "datacraft-docker"
    ports:
      - "8050:8050"
    volumes:
      - ./:/app
    env_file:
       - .env
```

- YAML file containing docker containers and multi container configuration.
- One off command « docker compose up » to start containers with many config options
  port
  env variables
  Volumes
  ....

Let’s move onto the cloud

Push it to Docker Hub

- docker login
- docker tag dockercraft axreki/dockercraft:1
- docker push axreki/dockercraft:1

Azure set up

- Free Trial
- Create a resource group « datacraft-workshop-docker »
- Quickstart center
- Create a container-based web app
- Chose Free Tier for plan

Automating the deployment CI/CD

Automate build on Docker Hub = CI

git commit –am ”commit message”

git tag -a v1.2 -m "version 1.2”

git push origin v1.2

Automated Builds

Autobuild triggers a new build with every **git push** to your source code repository. Learn More.

EatZeBaby/datacraft-workshop-docker | Use Docker Hub's infrastructure | Autotests: Off

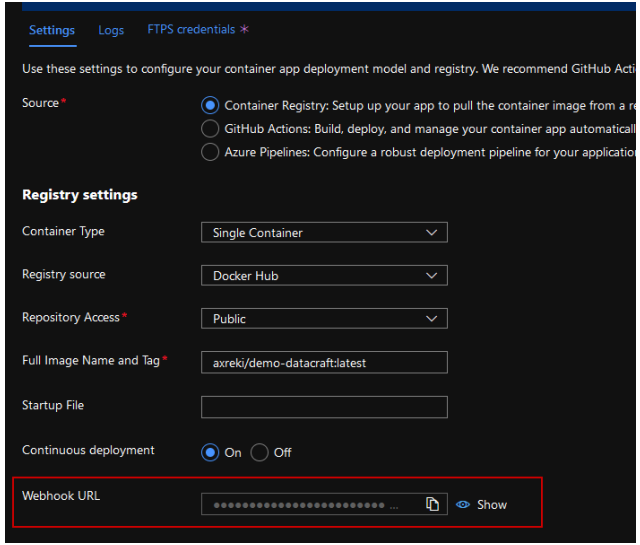
Docker Tag	Source	Latest Build Status	Autobuild	Build caching
version-{\1}	/^v([0-9.]+)$/	SUCCESS	✓	✓

Automating the deployment CI/CD

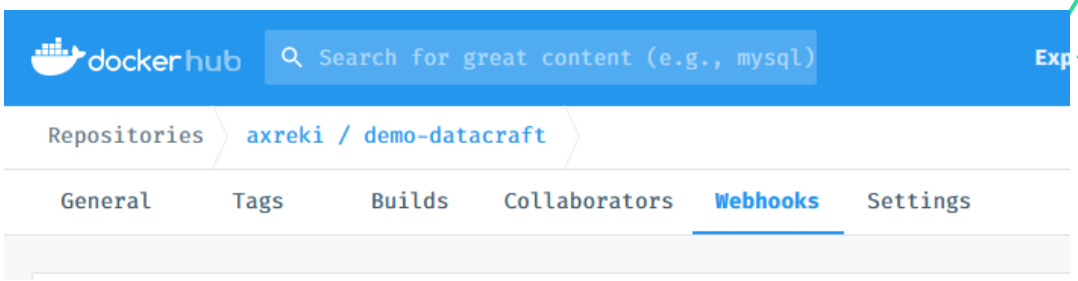
Automate deployment on Azure = CD

Get webhook url from Azure

Go to Azure > Web App > Container Settings > settings



Add a webhook to Docker Hub



Ekimetrics.

Automating the deployment CI/CD

Github Actions

Get webhook url from Azure

Go to Azure > Web App > Container Settings > settings

05.
Optimization

Pro Tip : Reuse of top unmodified layer

Dockerfile

```
FROM python:3.7
STEP 1.0
STEP 2.0
STEP 3.0
STEP 4.0
```

First build

```
FROM python:3.7
STEP 1.0
STEP 2.0
STEP 3.0
STEP 4.0
```

Step 2 modified

```
FROM python:3.7
STEP 1.0
STEP 2.1
STEP 3.0
STEP 4.0
```

Step 4 modified

```
FROM python:3.7
STEP 1.0
STEP 2.0
STEP 3.0
STEP 4.0
```

build

reused cache

Ekimetrics.

Pro Tip : Shared Layers

App A

```
FROM python:3.7
STEP 1.0
STEP 2.A
STEP 3.A
```

App B

```
FROM python:3.7
STEP 1.0
STEP 2.B
STEP 3.B
STEP 4.B
```

Reused cache

DÉCONSEILLÉ CAR 5 APPELS = 5 LAYERS

```
1  RUN apt-get update
2  RUN apt-get install -y apt-transport-https
3  RUN curl https://packages.microsoft.com/config/debian/9/prod.list > /etc/apt/sources.list.d/ms
4  RUN apt-get update
5  RUN apt-get install msodbcsql17 unixodbc-dev -y
```

1 SEUL LAYER

```
RUN apt-get update && \
    apt-get install -y apt-transport-https && \
    curl https://packages.microsoft.com/keys/microsoft.asc | apt-key add - && \
    curl https://packages.microsoft.com/config/debian/9/prod.list > /etc/apt/sources.list.d/ms
    apt-get update && \
    ACCEPT_EULA=Y apt-get install msodbcsql17 unixodbc-dev -y
```

EFFET BONUS : GARANTIE D'AVOIR UN APT-GET UPDATE TOUJOURS À JOUR

RÉDUCTION DOCKER SIZE

```
1  FROM python:3.7
2  WORKDIR /app
3  COPY Pipfile* ./
4  RUN pip install pipenv
5  RUN pipenv install --system --deploy
6  COPY src .
7  CMD ["python", "streamlit_index.py"]
```

1GO+

```
1  FROM python:3.7-slim
2
3  WORKDIR /app
4
5  # both files are explicitly required!
6  COPY Pipfile Pipfile.lock ./
7
8  RUN pip install pipenv && \
9    apt-get update && \
10   apt-get install -y --no-install-recommends gcc python3-dev libssl-dev && \
11   pipenv install --deploy --system && \
12   apt-get remove -y gcc python3-dev libssl-dev && \
13   apt-get autoremove -y && \
14   pip uninstall pipenv -y
15
16 COPY app ./
17
18 CMD ["python", "streamlit_index.py"]
```

~250mo

- Base image allégée (python-slim)
- unification des commandes RUN
- nettoyage des package uniquement nécessaire au build

Comparaison des tailles d'image python

```
axel.richier@EKI-PC00053  ~
> docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
python              3.8                 b4b9bf31ec03        23 hours ago        882MB
python              3.8.3-slim          9d84edf35a0a        6 months ago        165MB
python              3.8.3               7f5b6ccd03e9        6 months ago        934MB
python              3.8.3-alpine        8ecf5a48c789        6 months ago        78.9MB
```

05.
Learn more

Ekimetrics.

Learning resources

```
$> docker run -dp 80:80 docker/getting-started
```

Play with Docker [No install required]

Containerized Python Development

3h YouTube Course

Not Only Docker

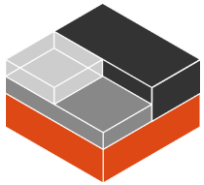
CoreOS rkt

Mesos Containerizer

LXC Linux Containers

OpenVZ

containerd



Ekimetrics.

Questions ?