

docker®

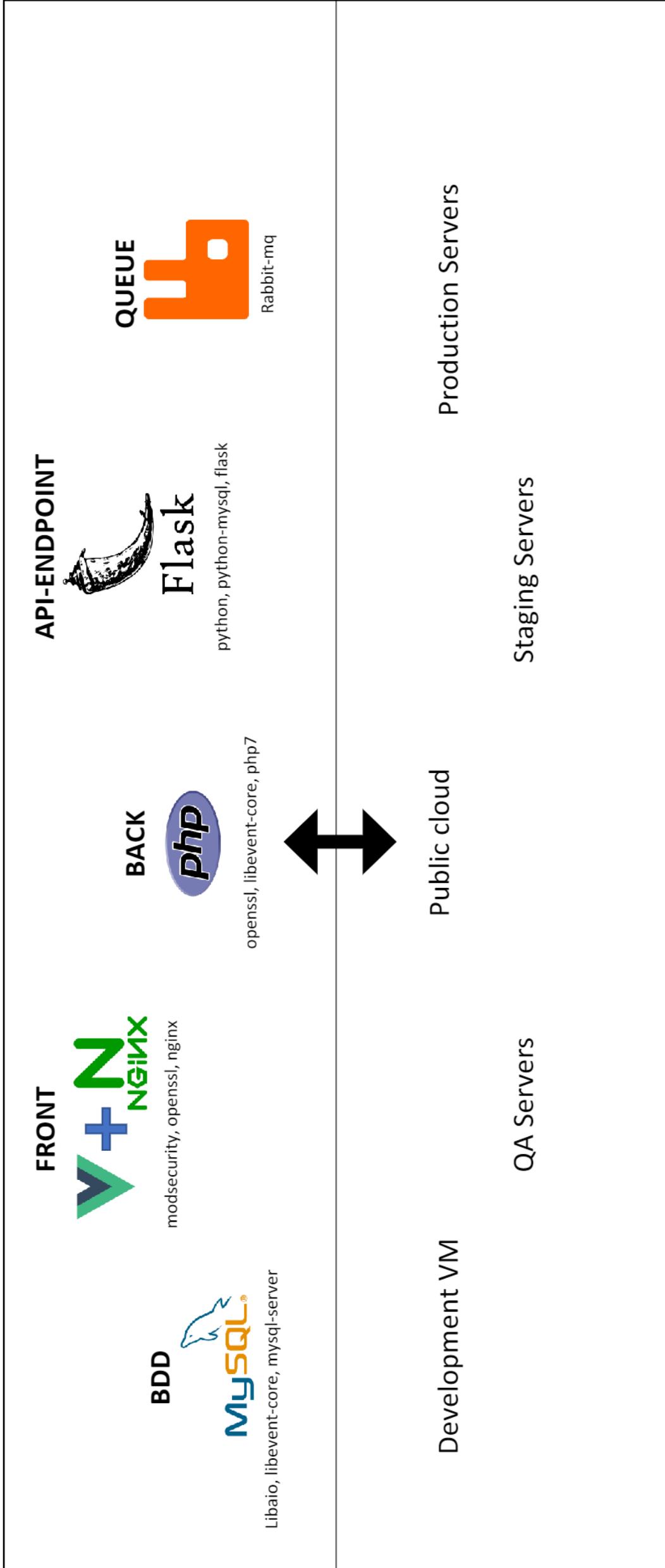
# Introduction à Docker

# Présentation de Docker

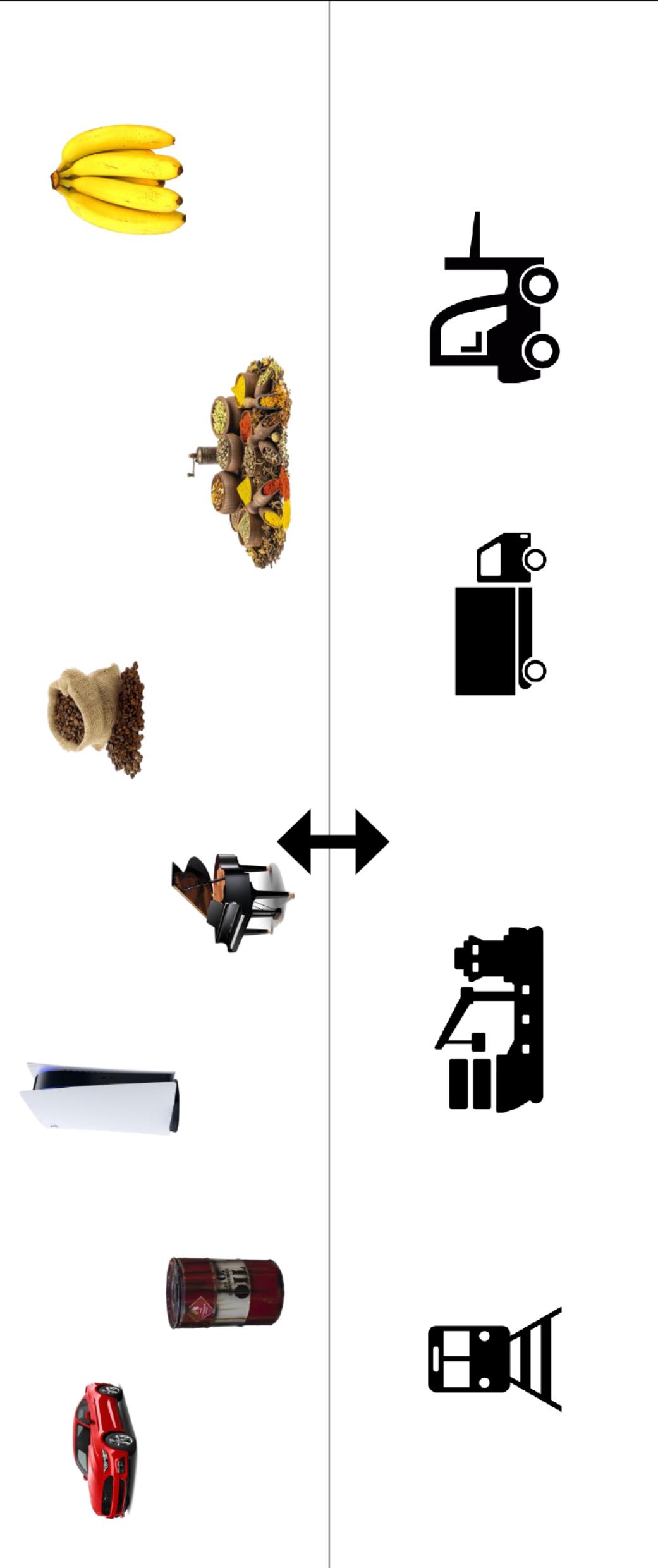
- Logiciel Open source
- Apparu en Mars 2013
  - Slogan « Build, Ship, Run »
- Nouvelle manière d'appréhender la virtualisation

# Déploiement avant Docker

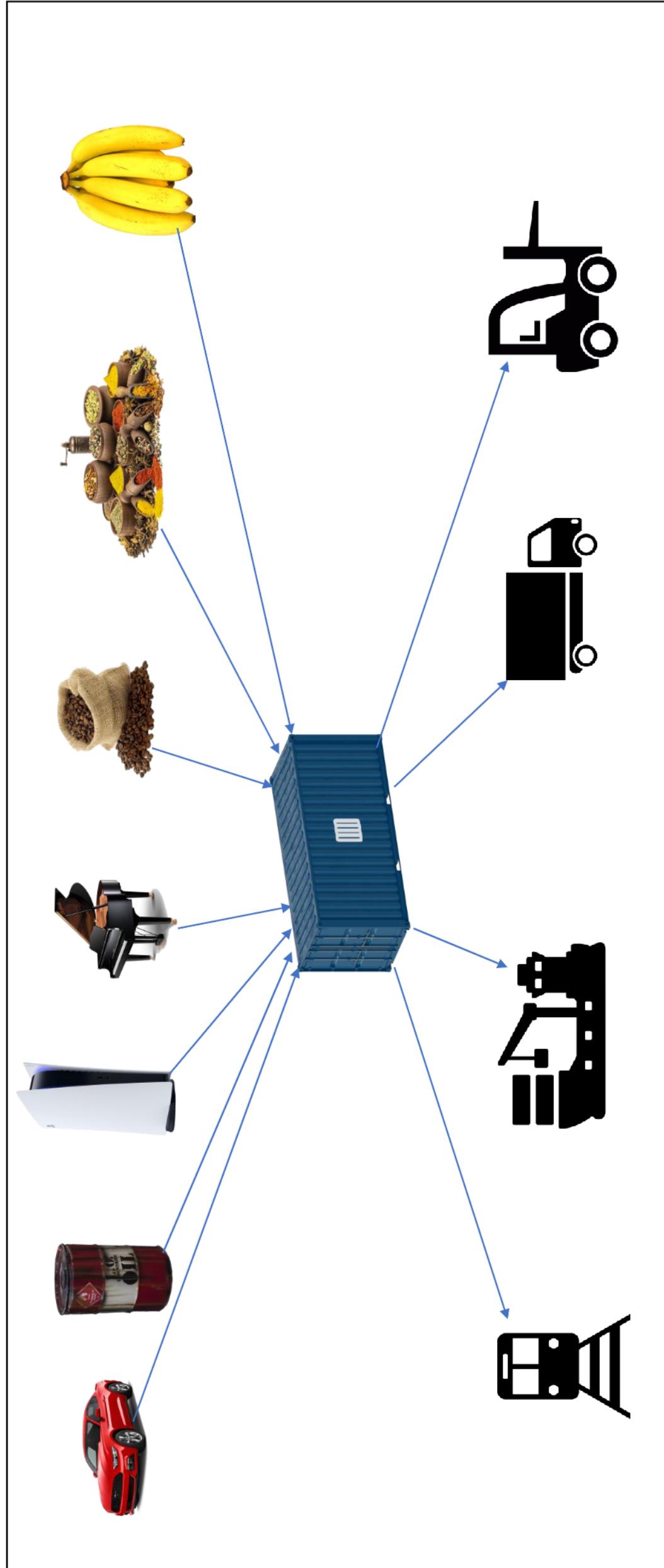
# Mise en production des application



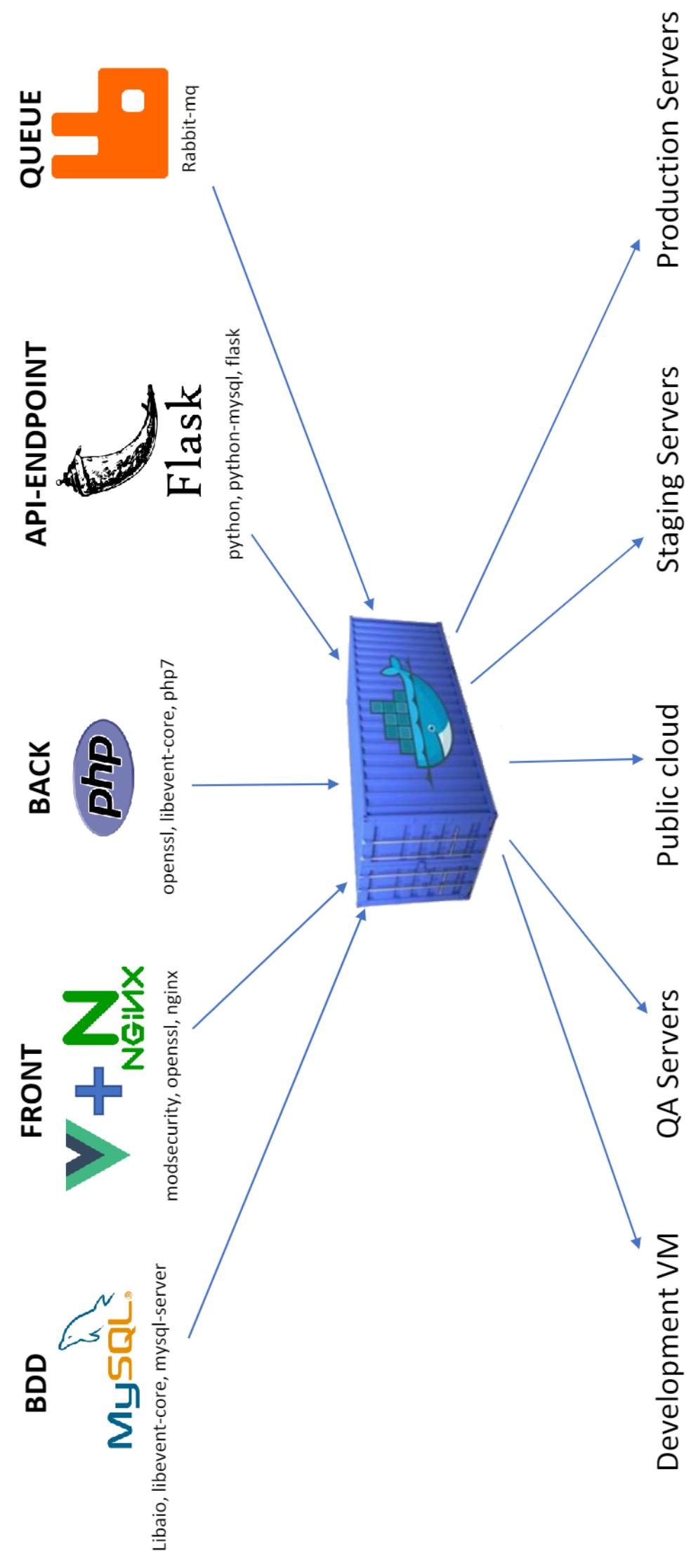
# Parallèle avec l'industrie du transport



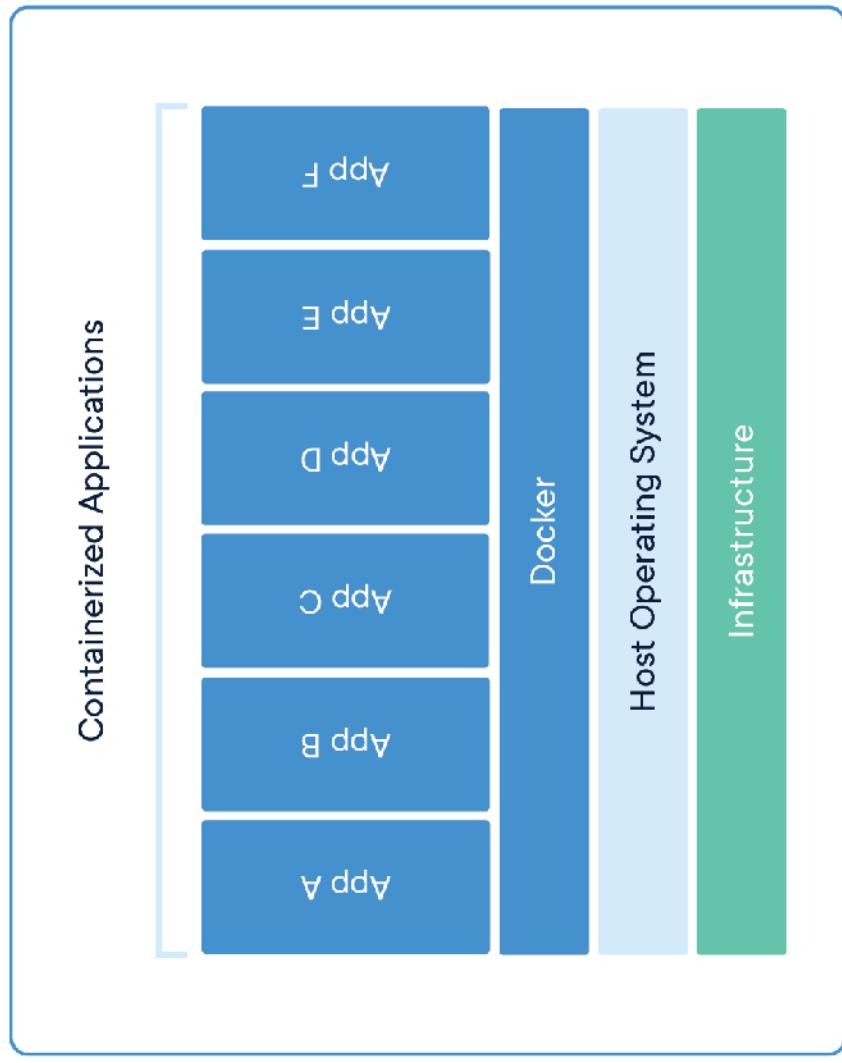
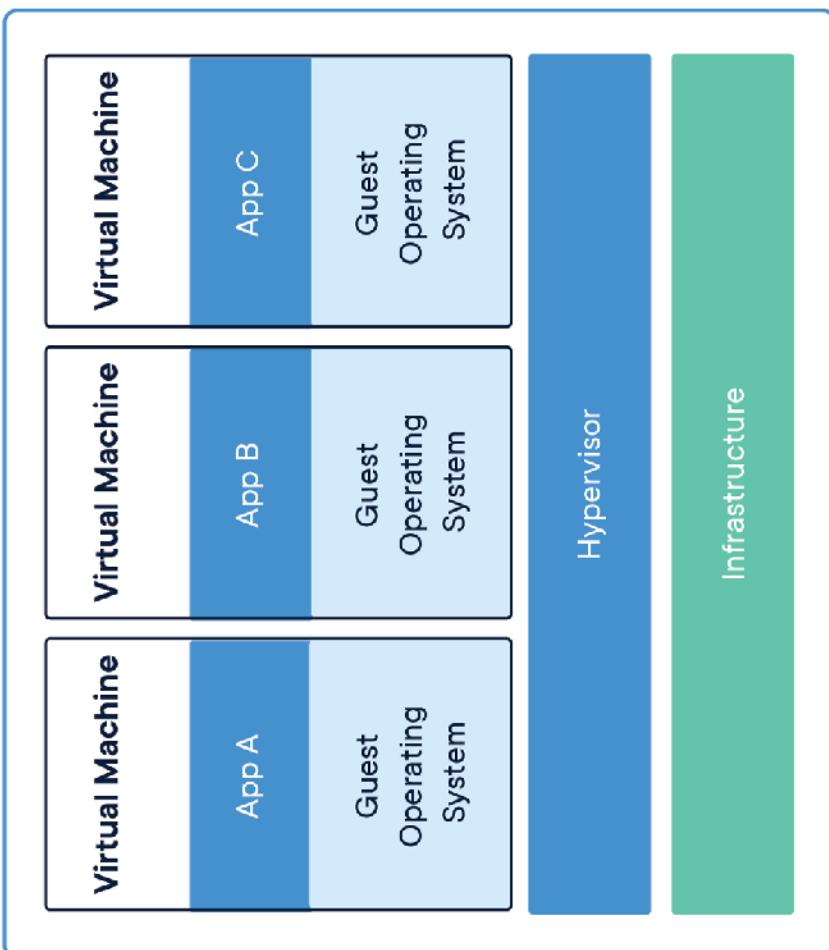
# Uniformisation du transport



# Mise en production des application

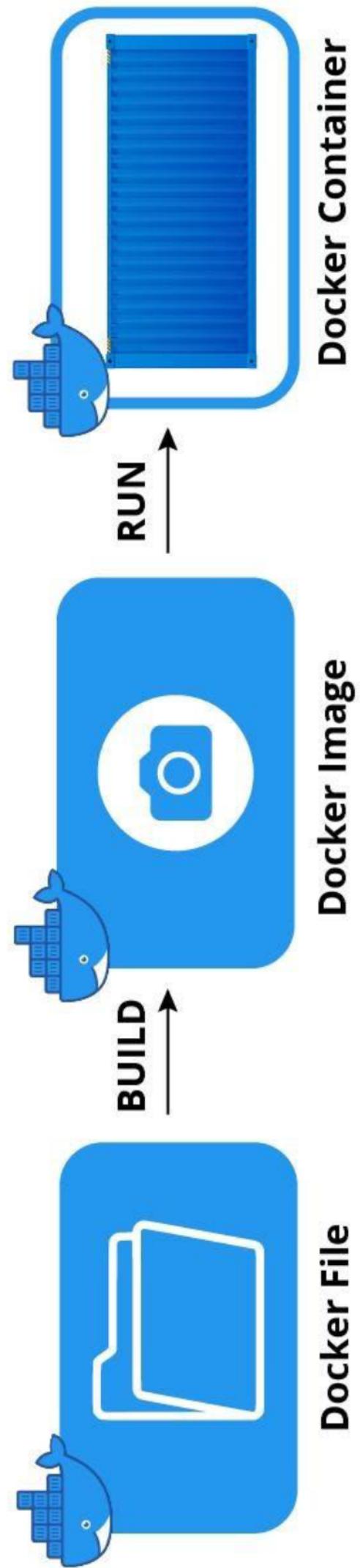


# Évolution de la virtualisation



# Fonctionnement et principes de Docker

# Les éléments



# Image Docker

Définition : C'est une image modèle à partir de laquelle on va pouvoir instancier des containers Docker.  
Elle va contenir une application ou un processus ainsi que ses dépendances pour pouvoir exécuter l'application de manière optimale.

# Container

Un container est un processus isolé du reste du système de l'hôte, qui fonctionne dans un environnement qui embarque toutes les dépendances du processus pour lui permettre de fonctionner correctement.

```
$ 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/>

# Dockerfile

Définition : C'est le fichier qui va servir à Docker à créer une Docker à partir des instructions présentent dans le Dockerfile. L'image Docker sera donc composée de plusieurs couche (layer), une couche par instruction.

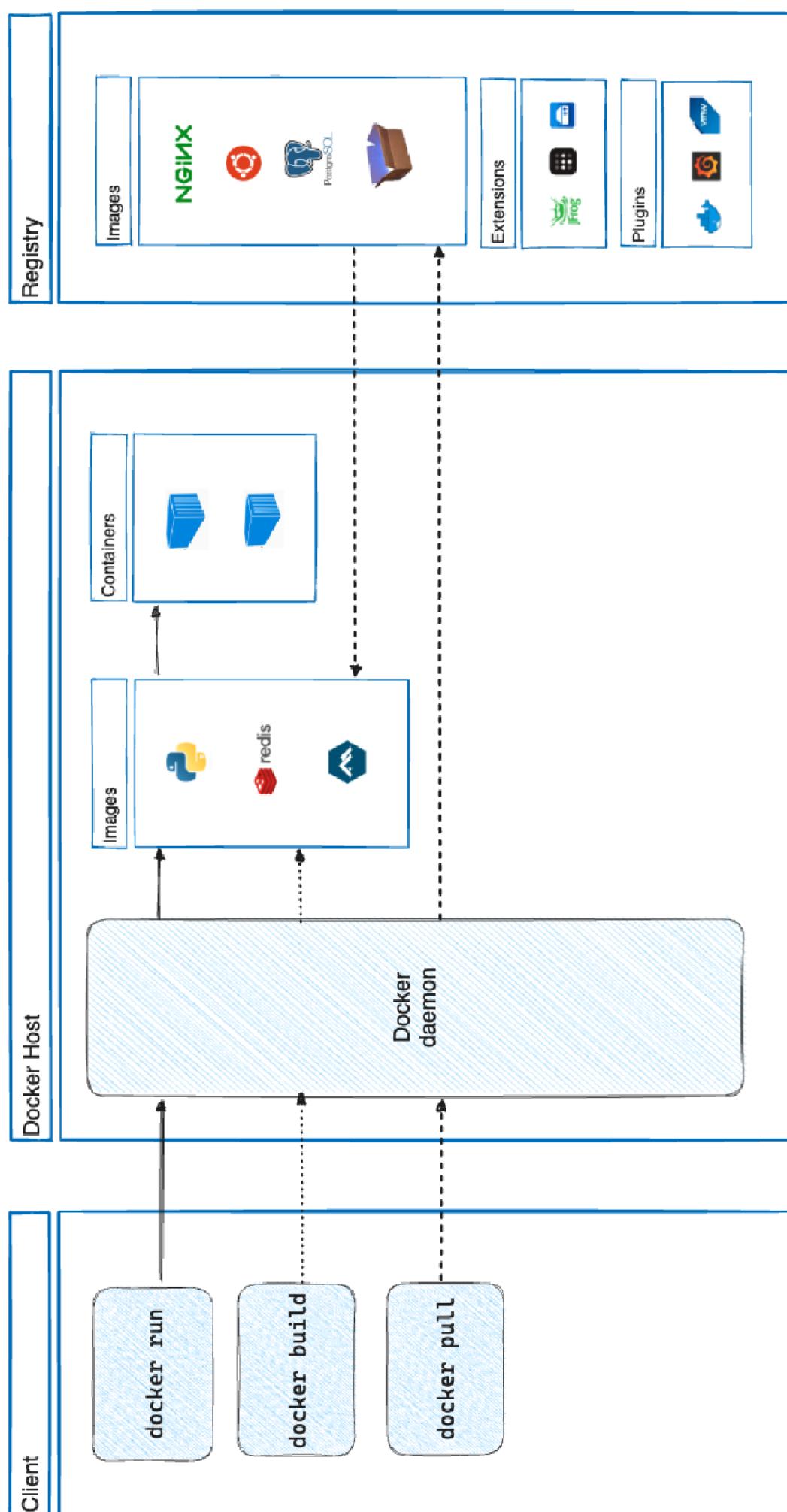
```
FROM ubuntu:20.04

RUN apt-get install nginx
RUN echo "Hello World !!" > /usr/share/nginx/html/index.html

EXPOSE 80,443

CMD ["nginx", "-g", "daemon off;"]
```

# Fonctionnement



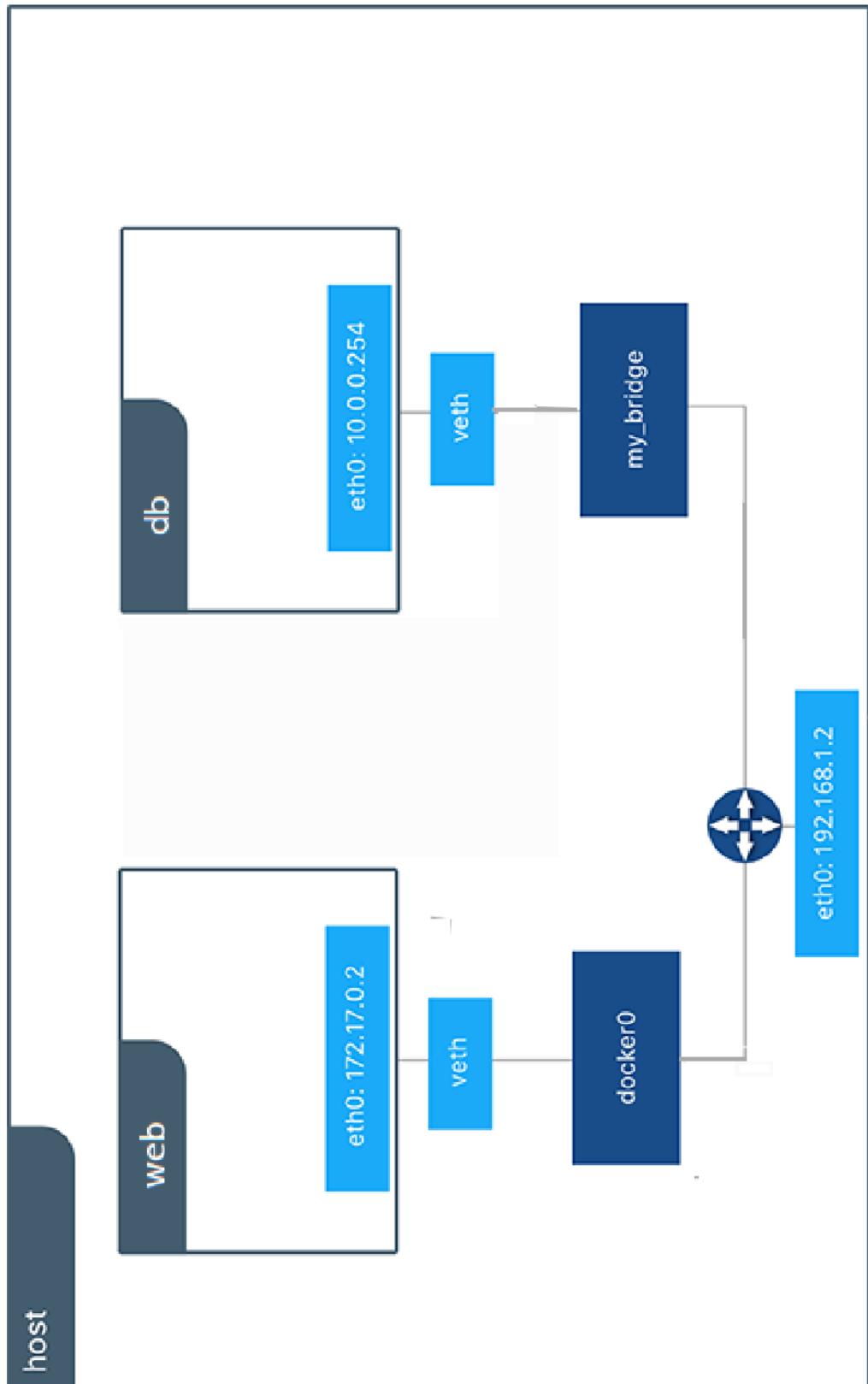
# Volume Docker

Par définition un container docker est STATELESS, cela veut dire qu'il ne contient aucune donnée. Cependant pour Certaine application, un container docker nécessite d'avoir des données dites persistantes. C'est pour cela qu'il existe les volumes. Il en existe 2 sortes, les volumes « nommés » (volume) et les montages (bind mount)

```
docker run -d --name devtest -v data:/app nginx:latest
```

```
docker run -d --name devtest -v /home:/app nginx:latest
```

# Docker Network



# Image Docker : construction

Builder votre image docker va consister à donner votre fichier Dockerfile à votre daemon docker pour qu'il exécute les instructions décrivent à l'intérieur dans le but de créer une image docker que vous pourrez lancer. Le flag « -t » va être utilisé pour tagguer votre image en lui donnant un nom (ici « hello-world-nginx ») ainsi qu'un numéro de Version (ici « 1.0 »).

```
docker build -t hello-world-nginx:1.0 .
```