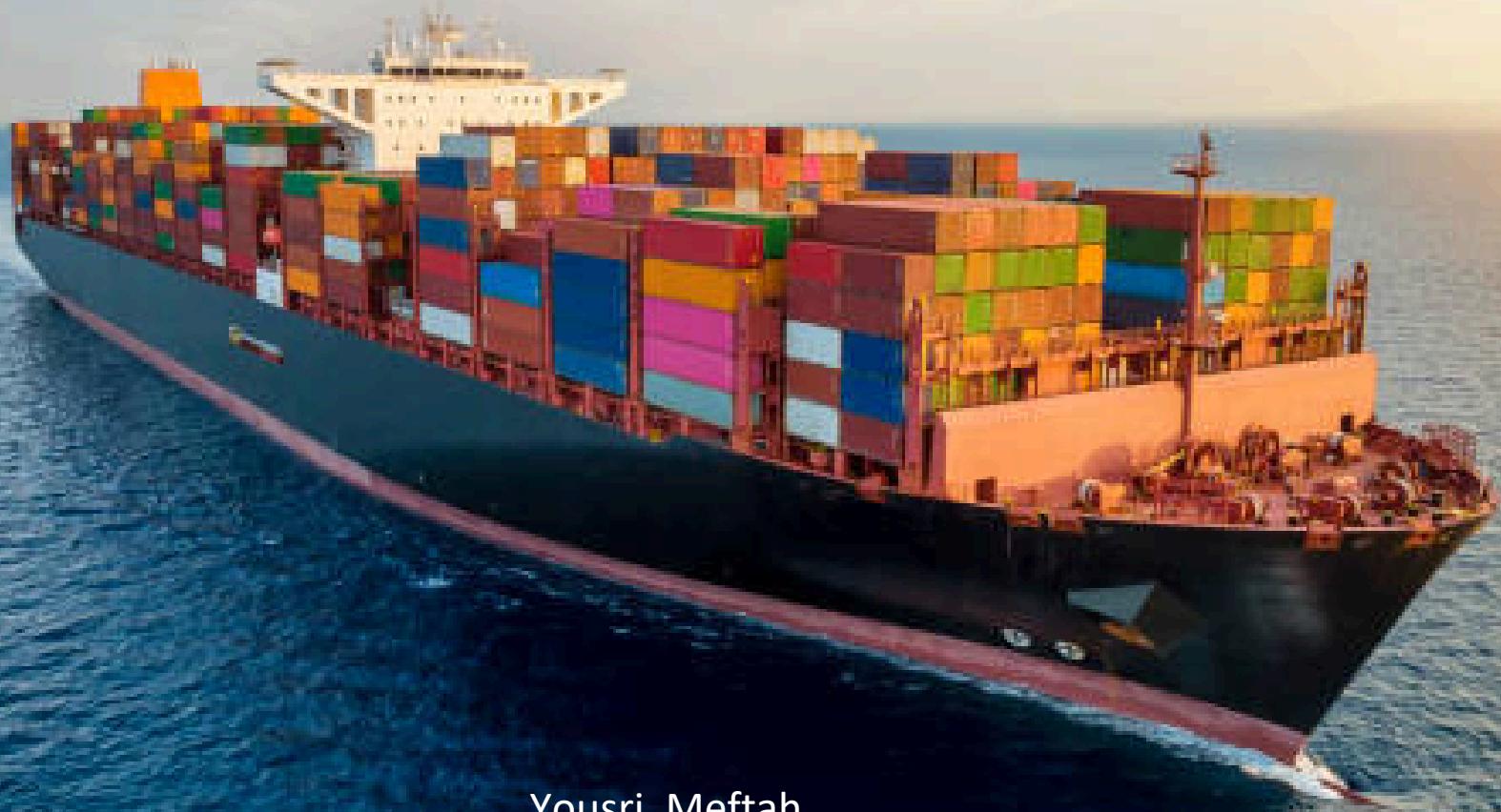


Containers & Docker



Yousri Meftah
02/10/2026

‘Real’ containers

'Real' containers



What is special about containers?



What is special about containers?



What is special about containers?



What is special about containers?



Power of Containers

Power of Containers

- Standard format everybody agrees on

Power of Containers

- Standard format everybody agrees on
- Once you seal the box, it gets shipped “as is”

Containers vs. VMs

Type 1 Hypervisor

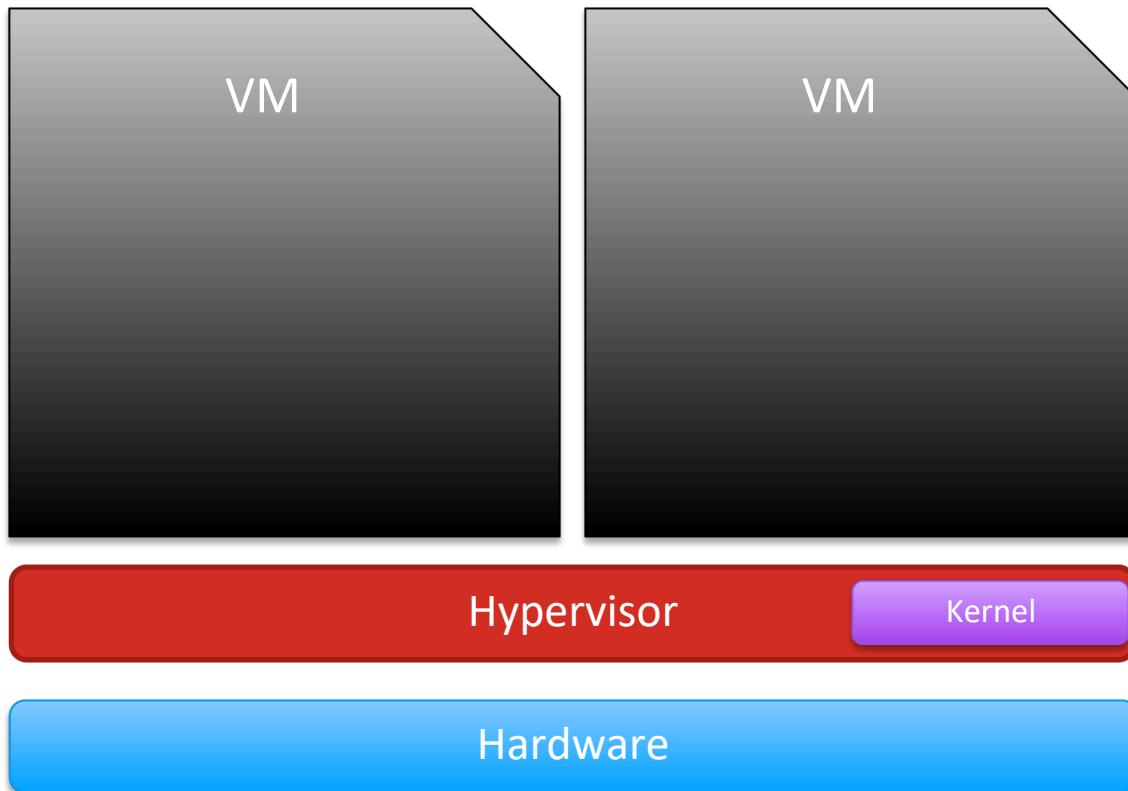
Type 1 Hypervisor

Hardware

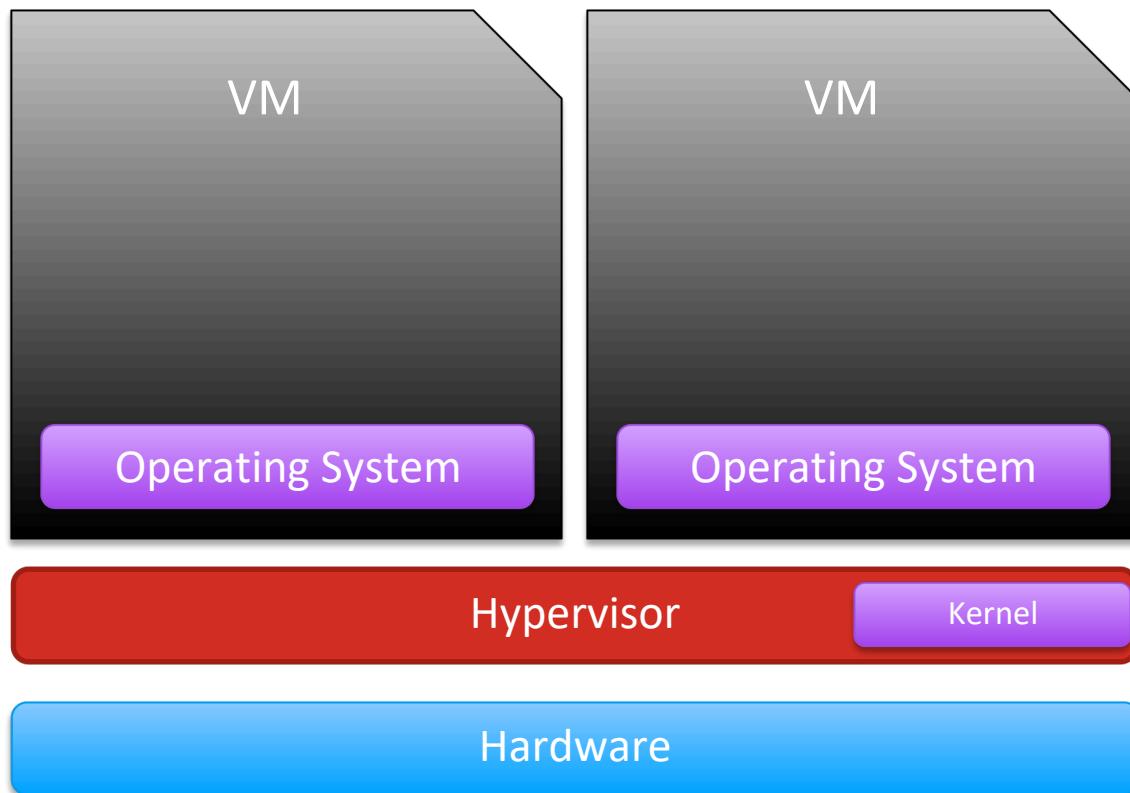
Type 1 Hypervisor



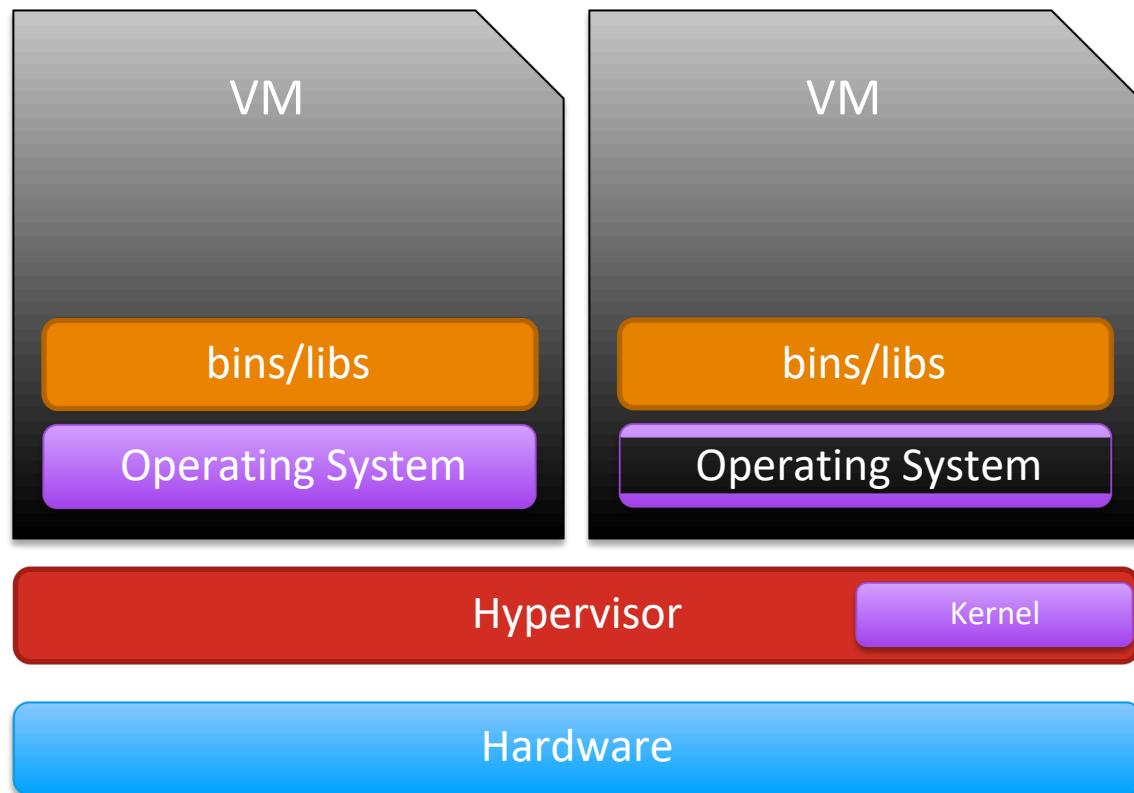
Type 1 Hypervisor



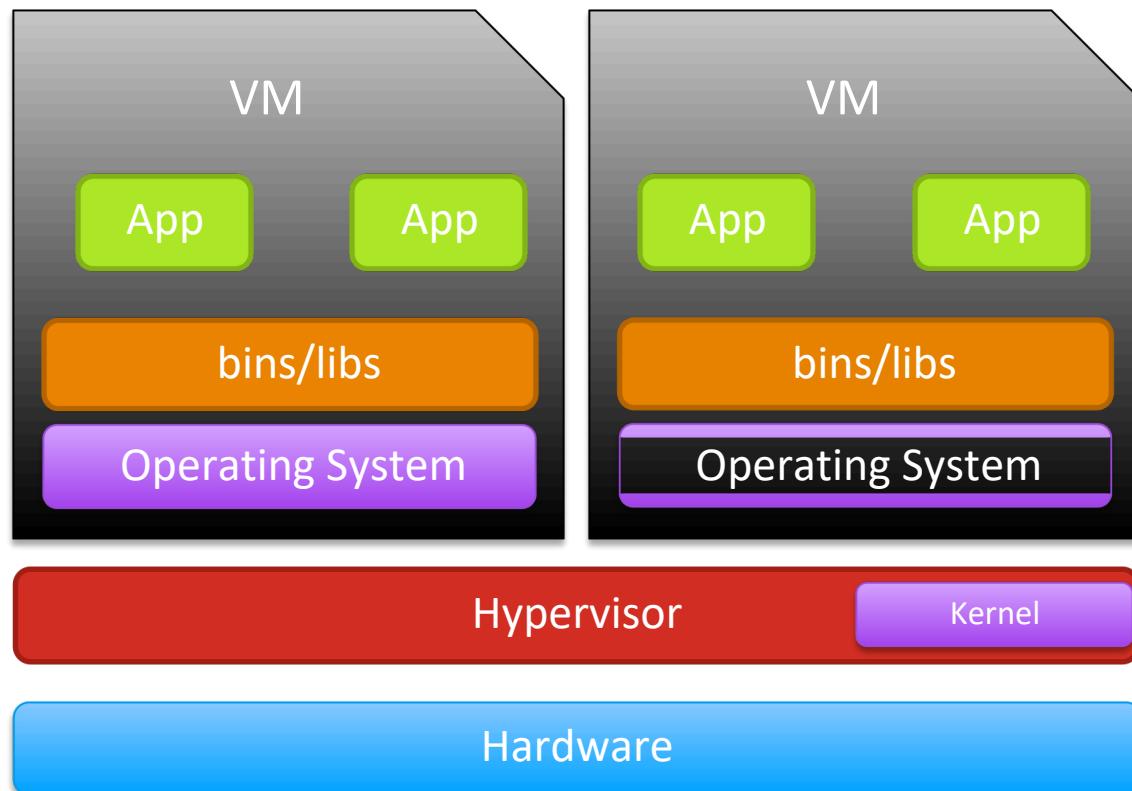
Type 1 Hypervisor



Type 1 Hypervisor



Type 1 Hypervisor



Type 2 Hypervisor

Type 2 Hypervisor

Hardware

Type 2 Hypervisor

Operating System

Hardware

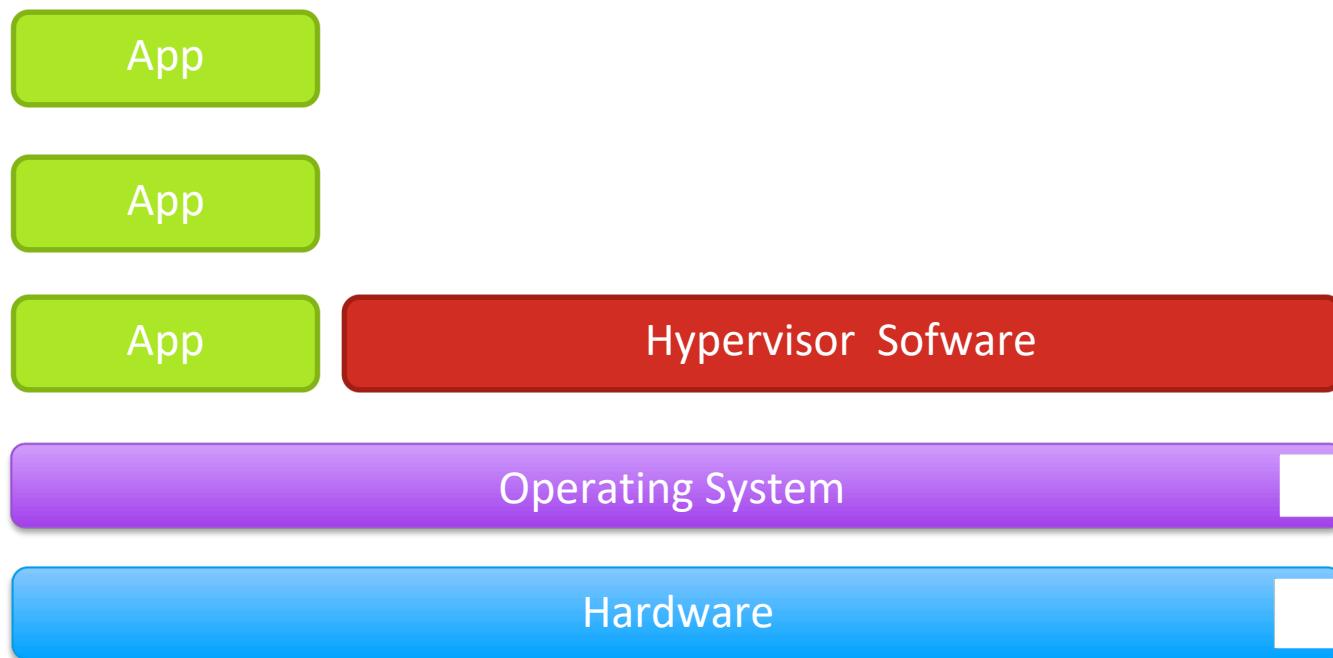
Type 2 Hypervisor

Hypervisor Software

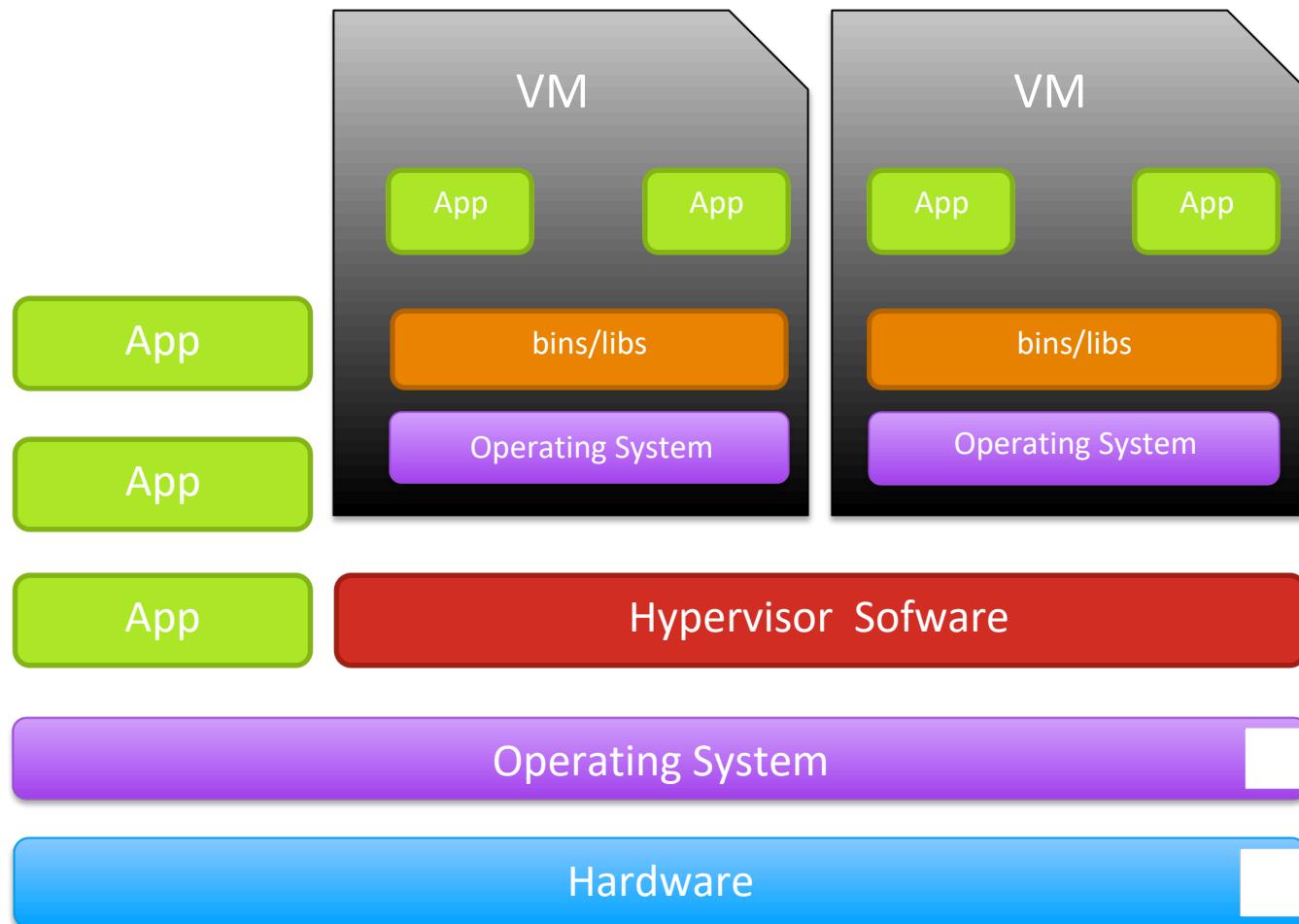
Operating System

Hardware

Type 2 Hypervisor



Type 2 Hypervisor



Linux Containers

Linux Containers

Hardware

Linux Containers

Operating System (Linux)

Hardware

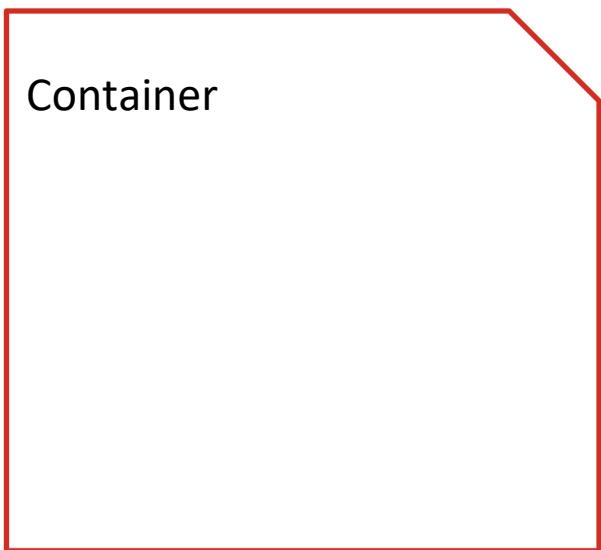
Linux Containers

bins/libs

OperaVng System (Linux)

Hardware

Linux Containers



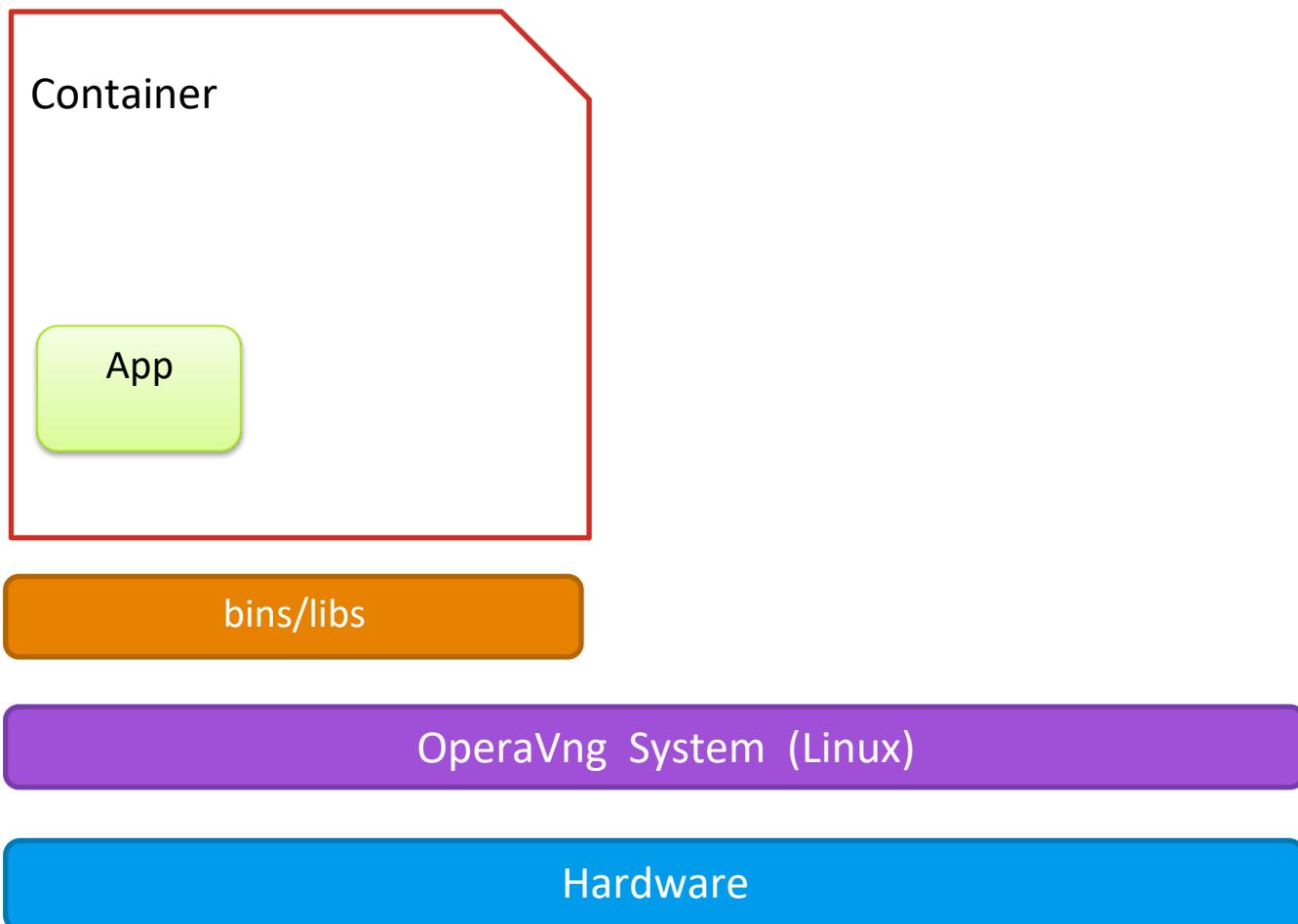
Container

bins/libs

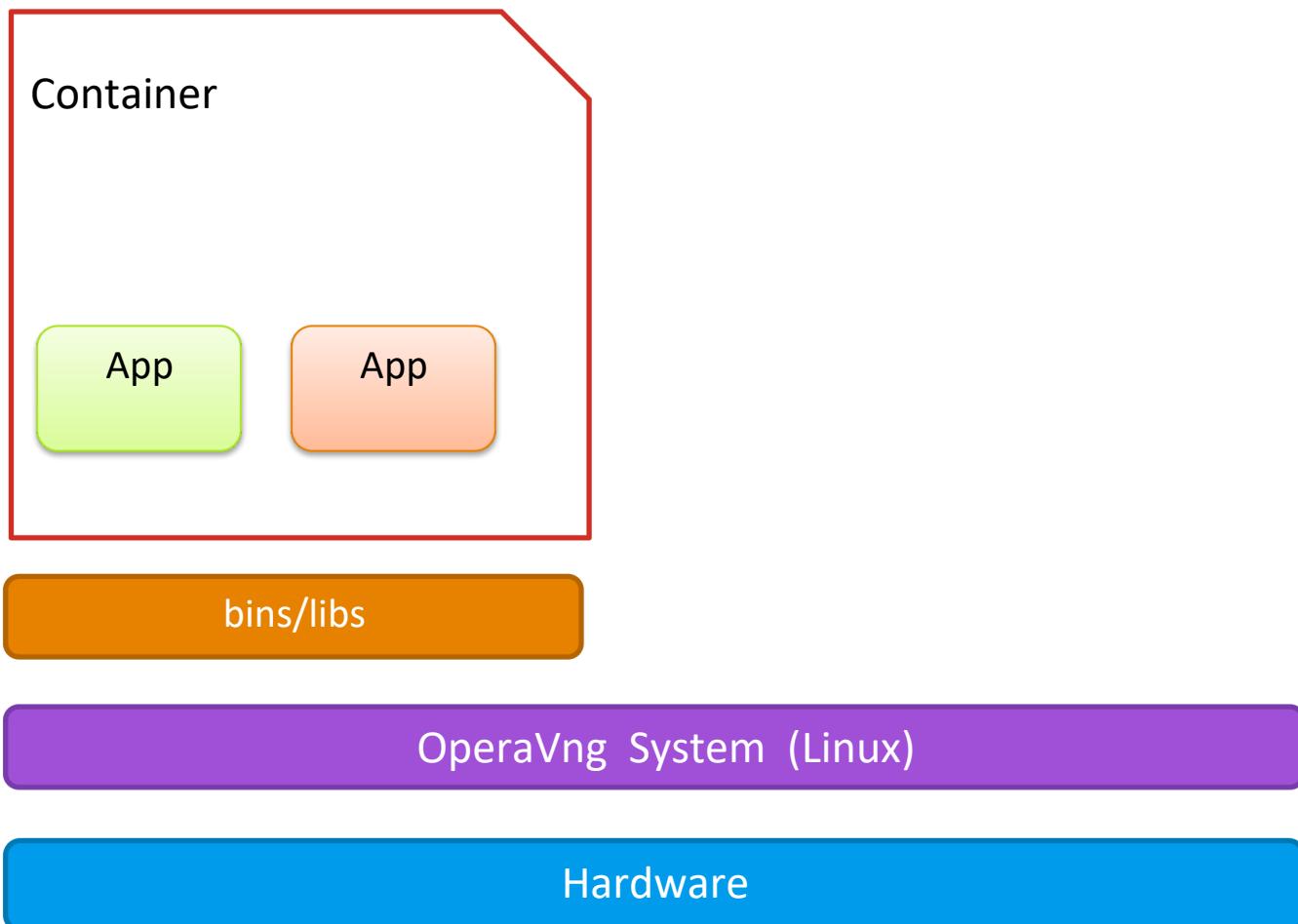
Operating System (Linux)

Hardware

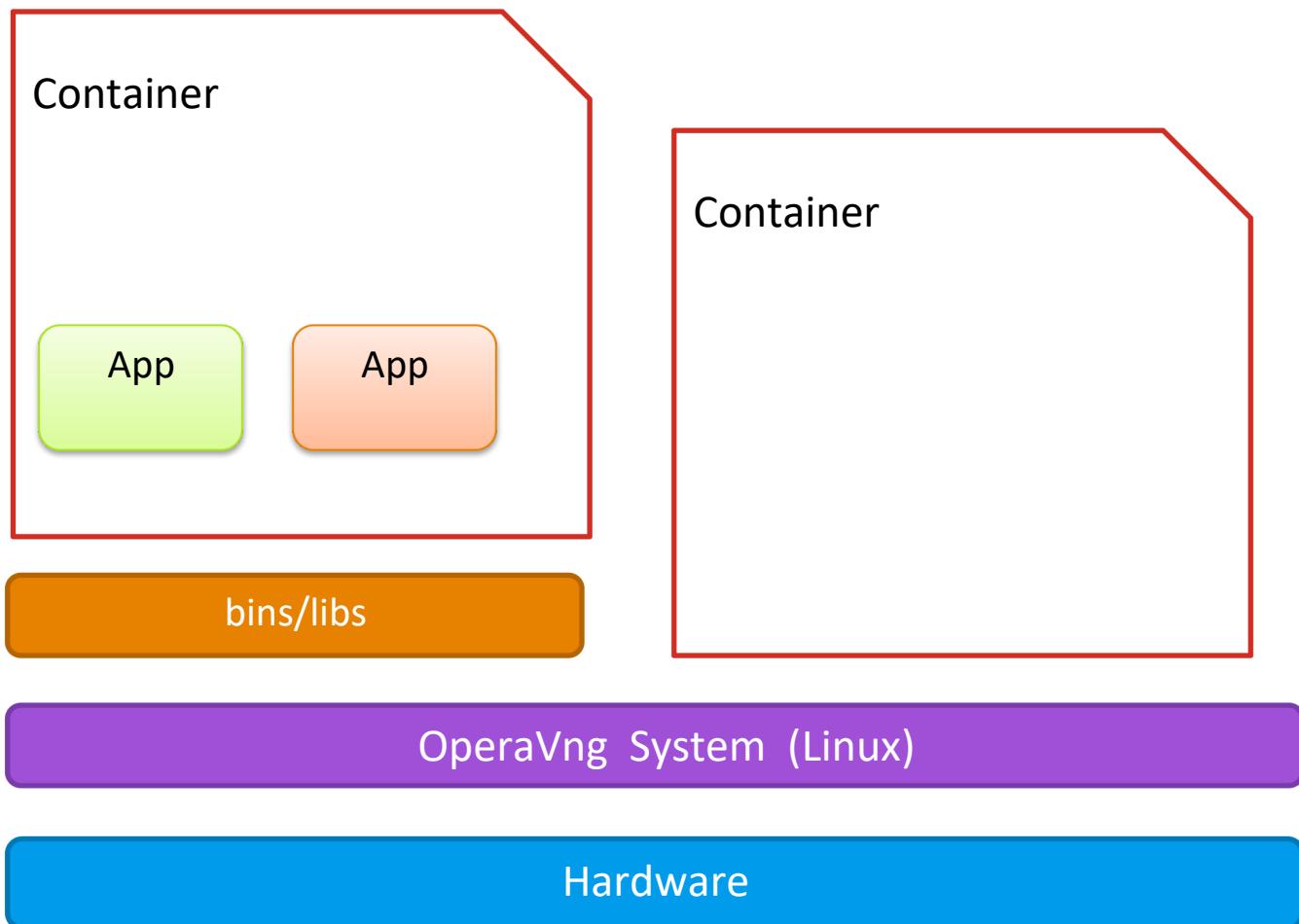
Linux Containers



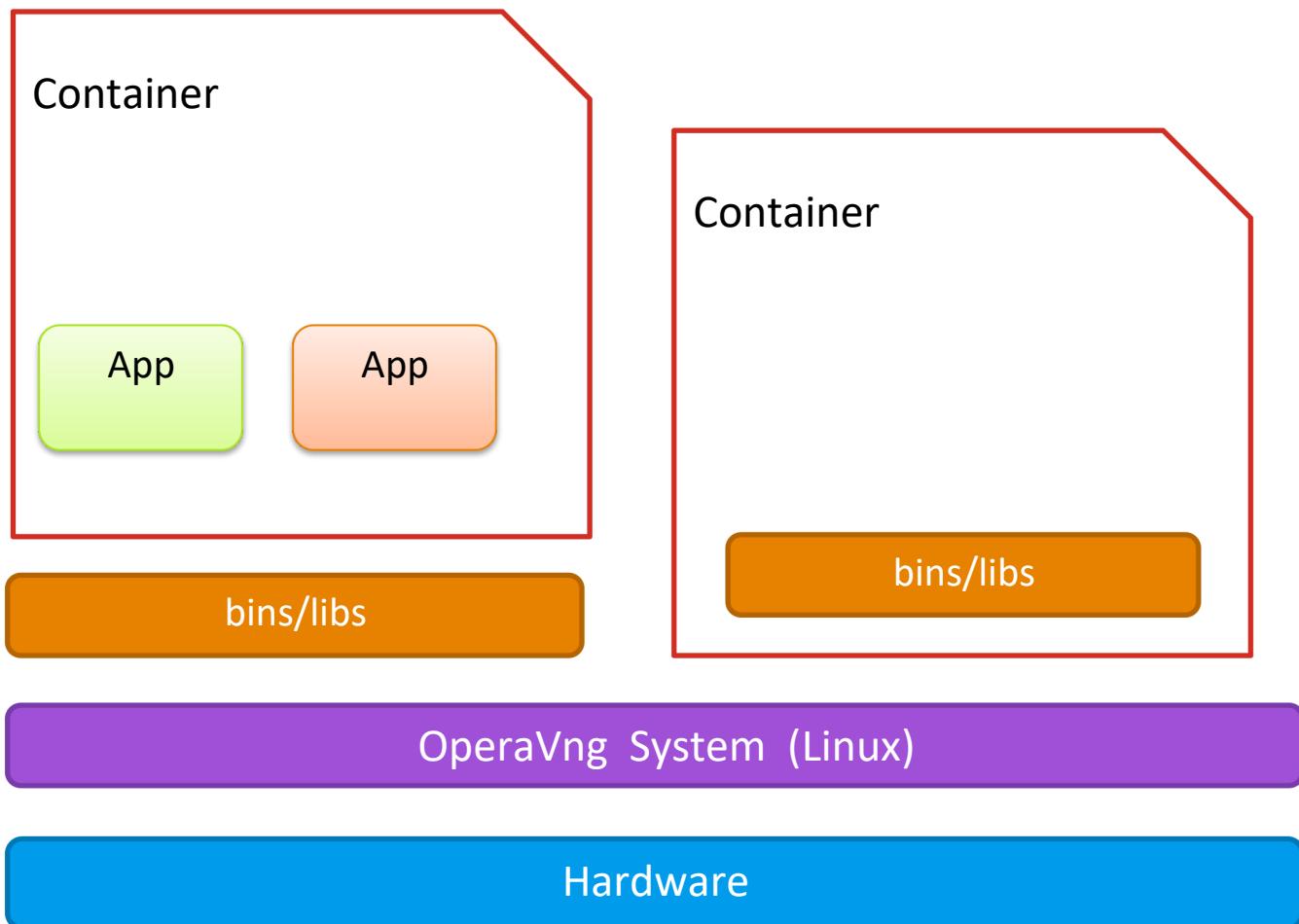
Linux Containers



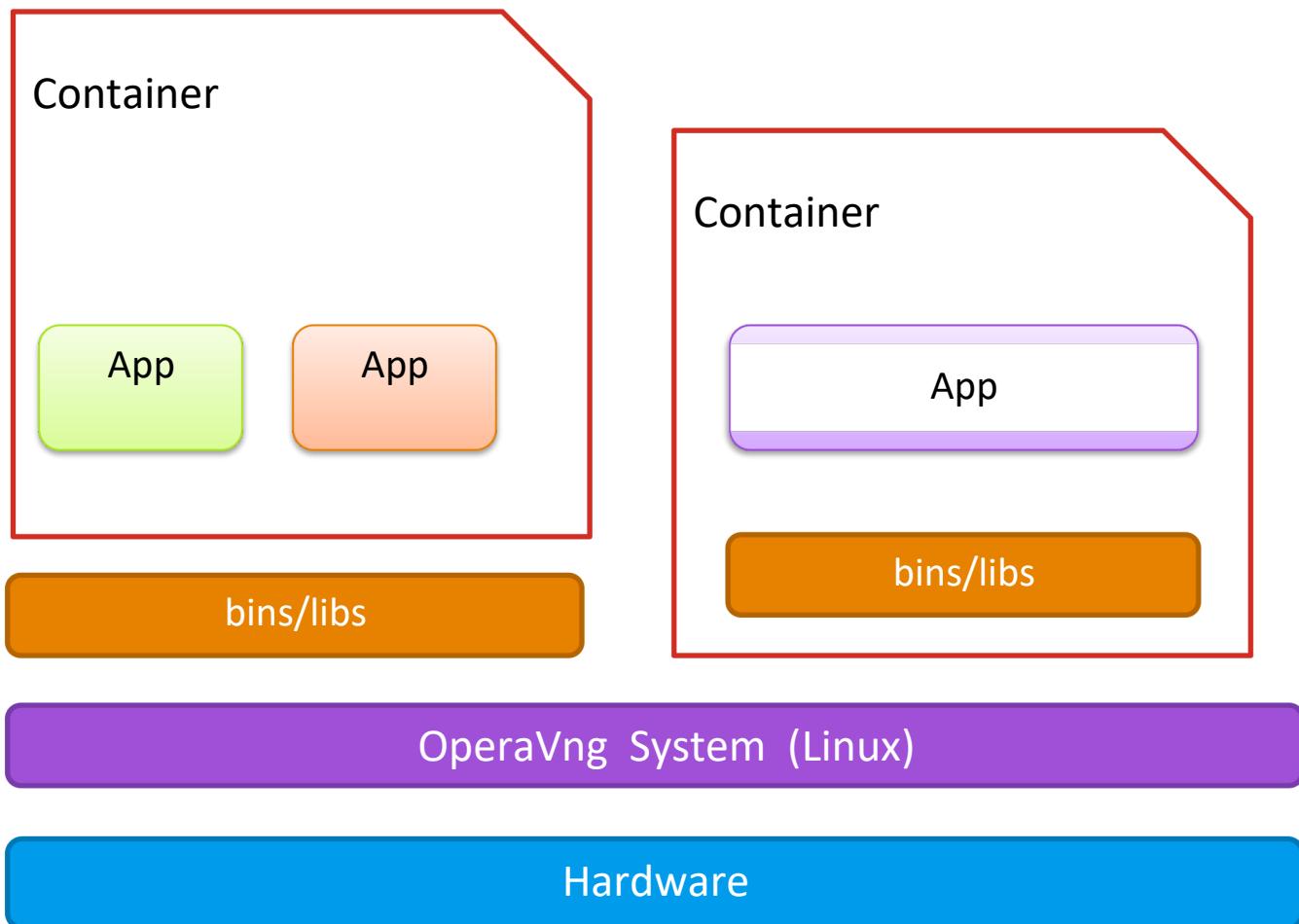
Linux Containers



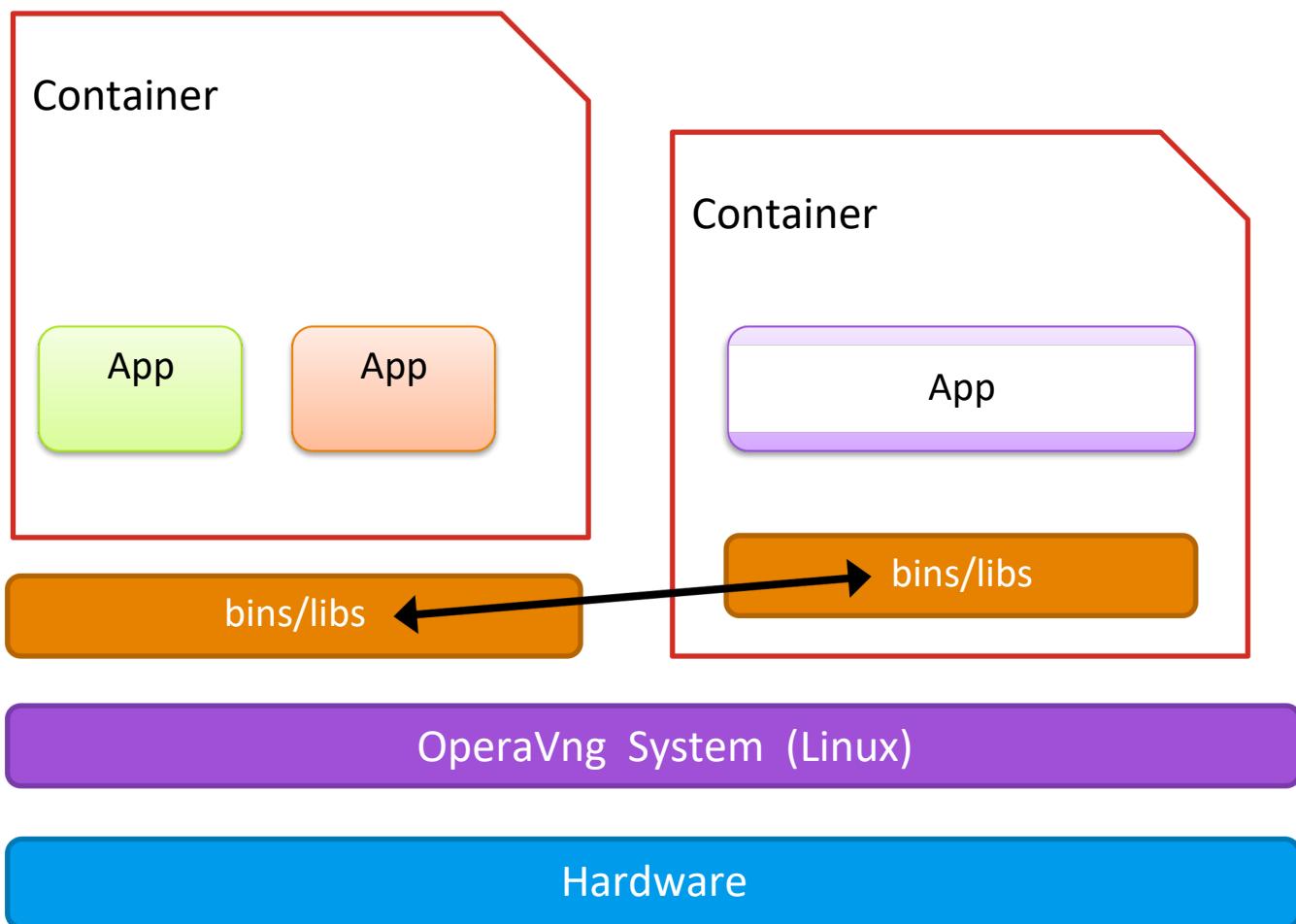
Linux Containers



Linux Containers

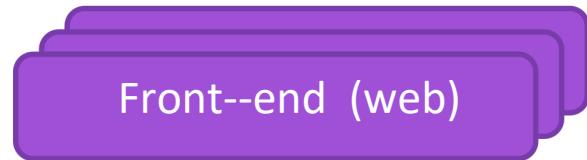


Linux Containers

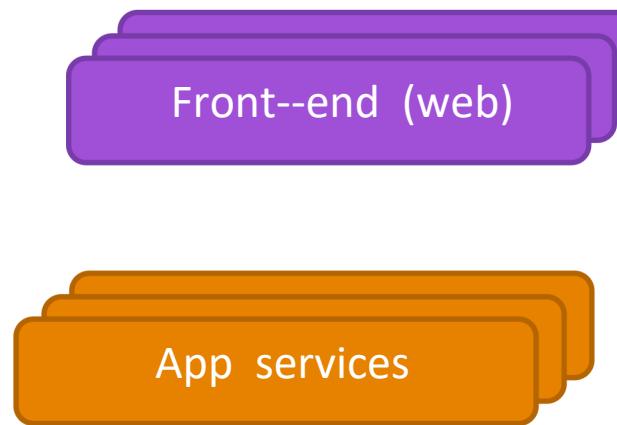


Distributed Application Model

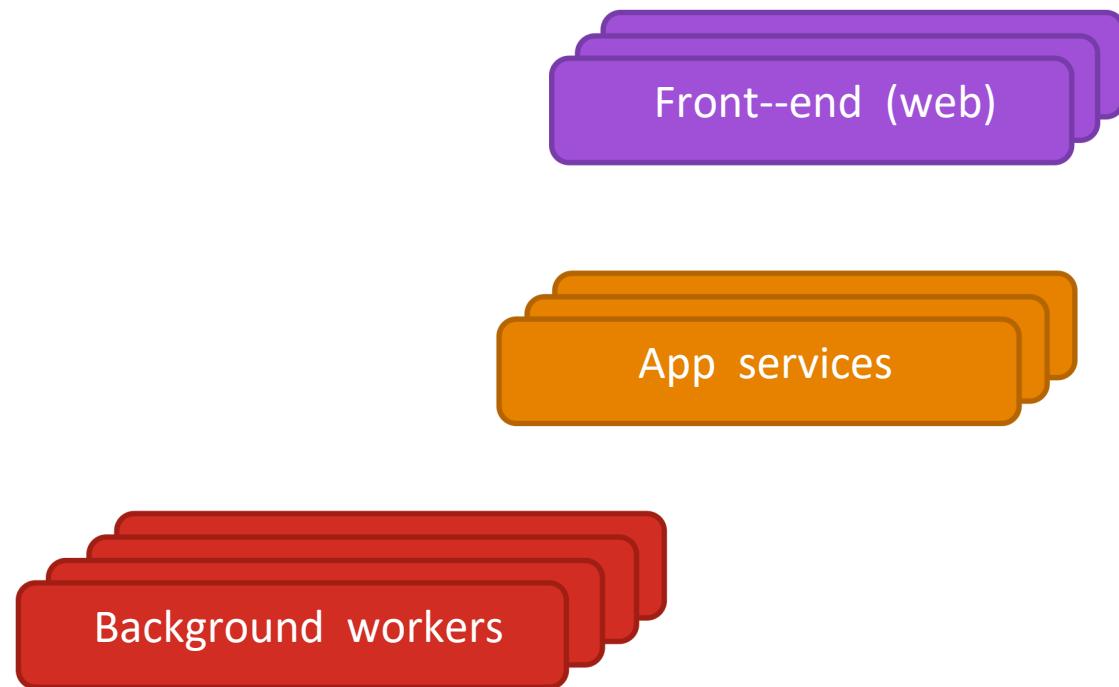
Distributed Application Model



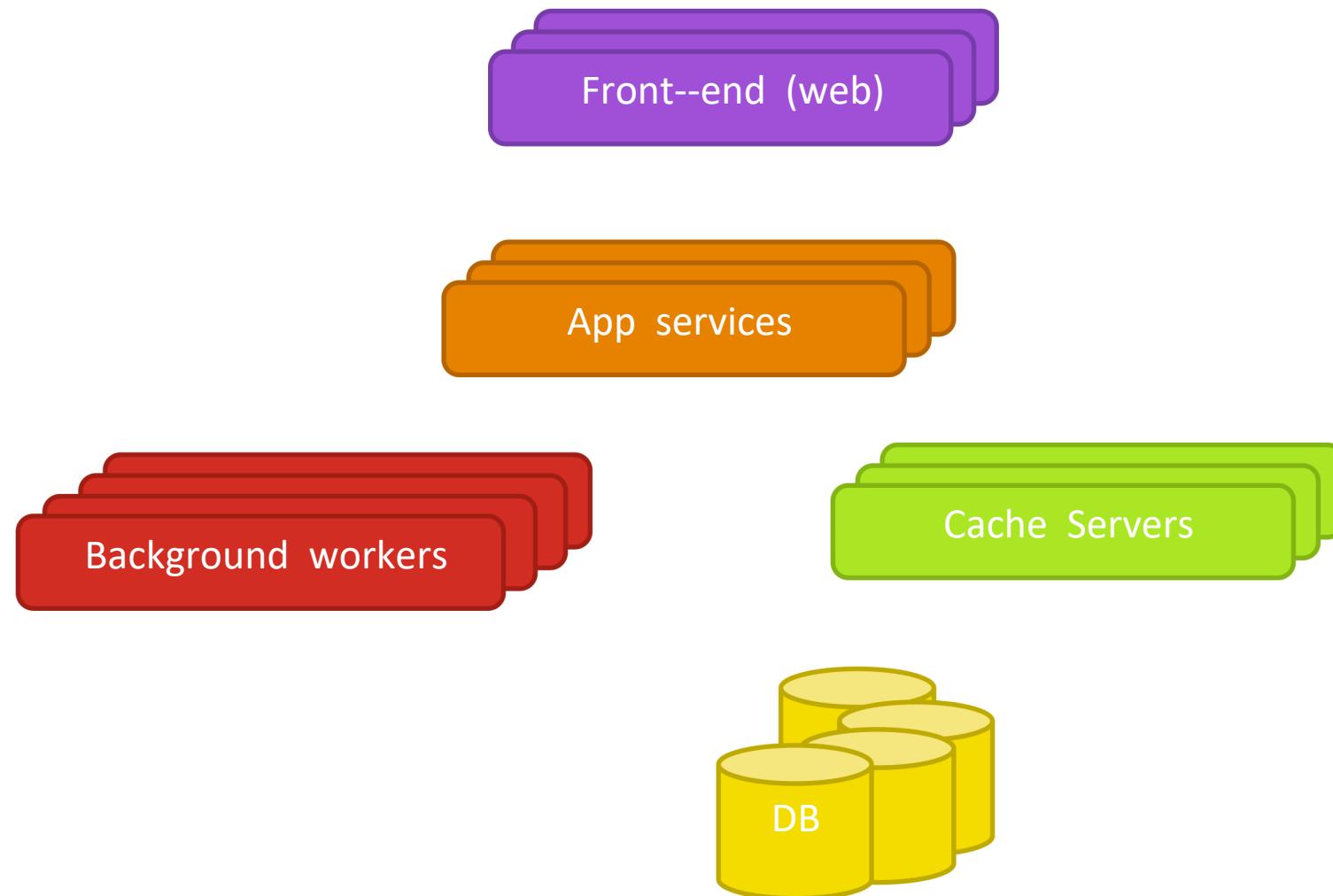
Distributed Application Model



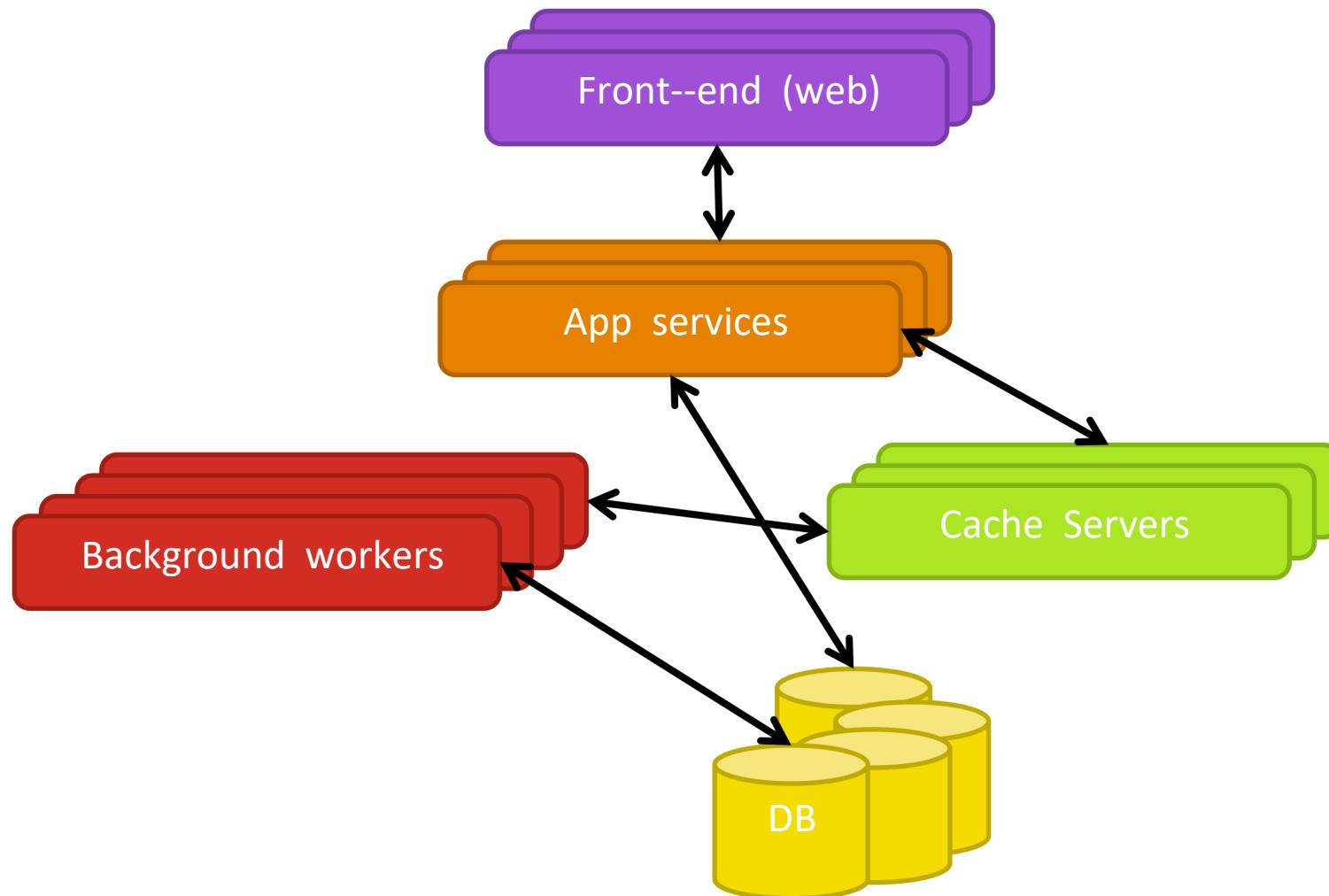
Distributed Application Model



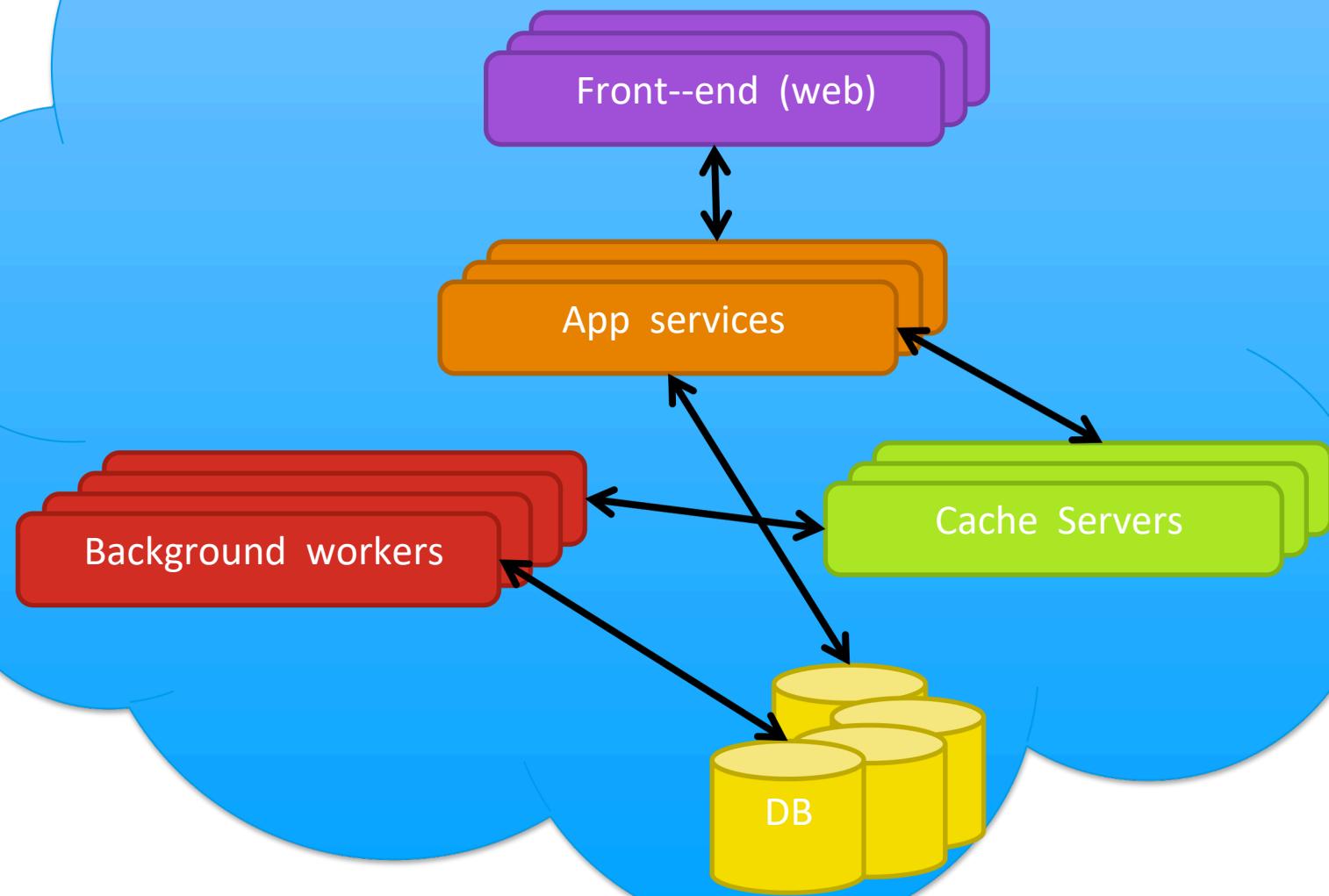
Distributed Application Model



Distributed Application Model



Distributed Application Model



Advantages of containers

Advantages of containers

- Only one kernel runs on the machine

Advantages of containers

- Only one kernel runs on the machine
- No Hypervisor overhead.
 - Apps directly run on CPU
 - Rack space & power & more free CPU cycles

Advantages of containers

- Only one kernel runs on the machine
- No Hypervisor overhead.
 - Apps directly run on CPU
 - Rack space & power & more free CPU cycles
- Run different versions of OS on same server
 - Run different OSes on the same server

Advantages of containers

- Only one kernel runs on the machine
- No Hypervisor overhead.
 - Apps directly run on CPU
 - Rack space & power & more free CPU cycles
- Run different versions of OS on same server
 - Run different OSes on the same server
- Process isolation & security

Use of Linux Containers

Use of Linux Containers

- **Borg**

- Google's "secret sauce"
- Everything runs in containers
- Not open source

Use of Linux Containers

- **Borg**

- Google's "secret sauce"
- Everything runs in containers
- Not open source



Use of Linux Containers

- **Borg**

- Google's "secret sauce"
- Everything runs in containers
- Not open source



- **Mesos**

- Developed by Twiger
- Open sourced

Use of Linux Containers

- **Borg**

- Google's "secret sauce"
- Everything runs in containers
- Not open source



- **Mesos**

- Developed by Twiger
- Open sourced



Use of Linux Containers

- **Borg**

- Google's "secret sauce"
- Everything runs in containers
- Not open source

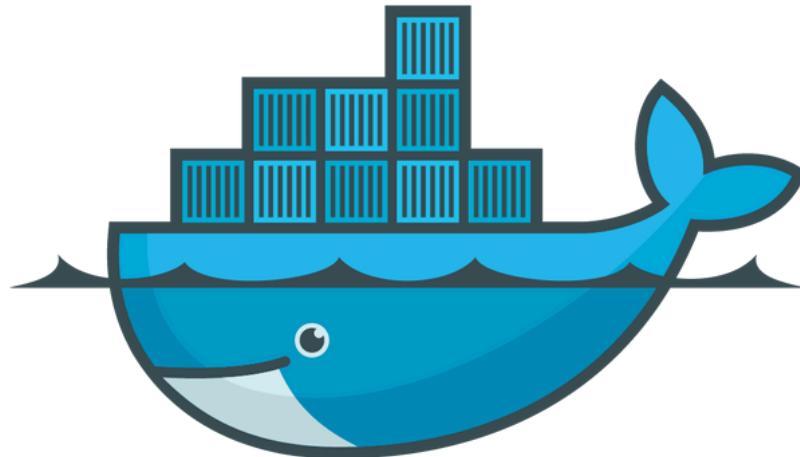


"IT'S A WAY OF STITCHING TOGETHER A
COLLECTION OF MACHINES INTO, BASICALLY,
A BIG COMPUTER."

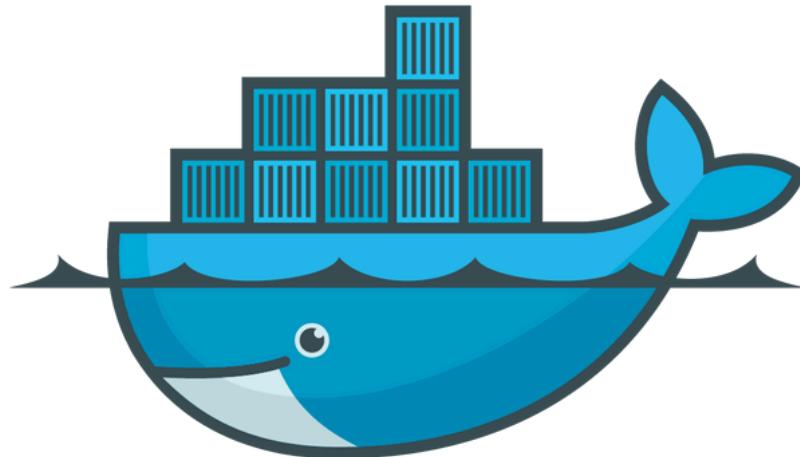
- **Mesos**

- Developed by Twiger
- Open sourced



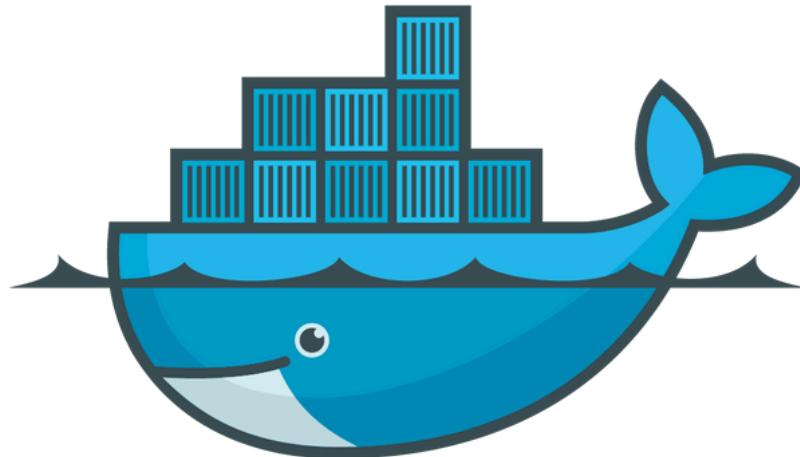


docker



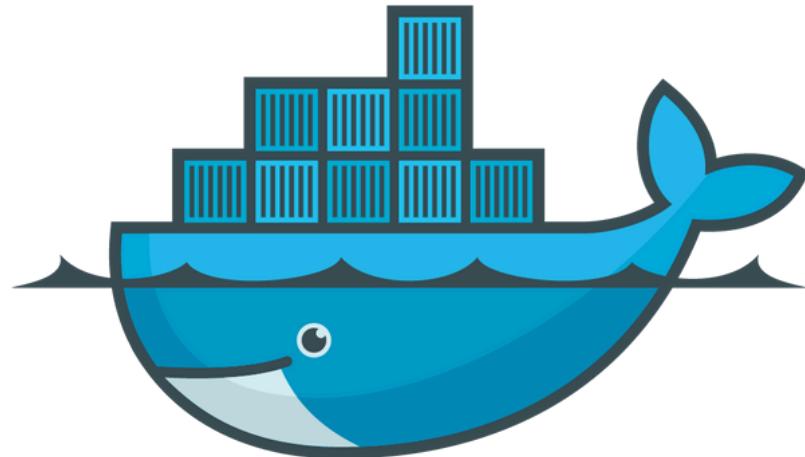
docker

...is a Linux container engine



docker

...is Open Source



docker

...is also a company.

Why is Docker special?

Why is Docker special?

- The real value is not technology.

Why is Docker special?

- The real value is not technology.
- It's “getting people to agree on something”.

Why is Docker special?

- The real value is not technology.
- It's “getting people to agree on something”.
 - Packaging and delivery method.

Why is Docker special?

- The real value is not technology.
- It's “getting people to agree on something”.
 - Packaging and delivery method.



Let's build a Docker image

```
FROM node:lts
ARG NODE_ENV=production
ENV NODE_ENV $NODE_ENV
WORKDIR /code
COPY package.json package-lock.json /code/
RUN npm ci
COPY . /code
CMD npm start
```

Base image

Variable set during build

Environnent variable

Working directory

Copy files into image

Command executed during build

Copy folder into image

Default command that will be
executed during a docker run

The diagram illustrates the structure of a Dockerfile. It shows various Docker instructions grouped by their purpose. Orange curly braces connect each instruction to its corresponding label on the left side of the image.

- Base image:** Points to the `FROM node:lts` instruction.
- Variable set during build:** Points to the `ARG NODE_ENV=production` instruction.
- Environnent variable:** Points to the `ENV NODE_ENV $NODE_ENV` instruction.
- Working directory:** Points to the `WORKDIR /code` instruction.
- Copy files into image:** Points to the `COPY package.json package-lock.json /code/` instruction.
- Command executed during build:** Points to the `RUN npm ci` instruction.
- Copy folder into image:** Points to the `COPY . /code` instruction.
- Default command that will be executed during a docker run:** Points to the `CMD npm start` instruction.

Registry Docker

- Hosts Docker Images
- Many online registry available:
 - Docker Hub
 - Azure ACR
 - AWS ECR
 - Github Registry
 - ...
- Many self-hosted solutions:
 - Docker Registry
 - Nexus
 - Artifactory
 - ...

