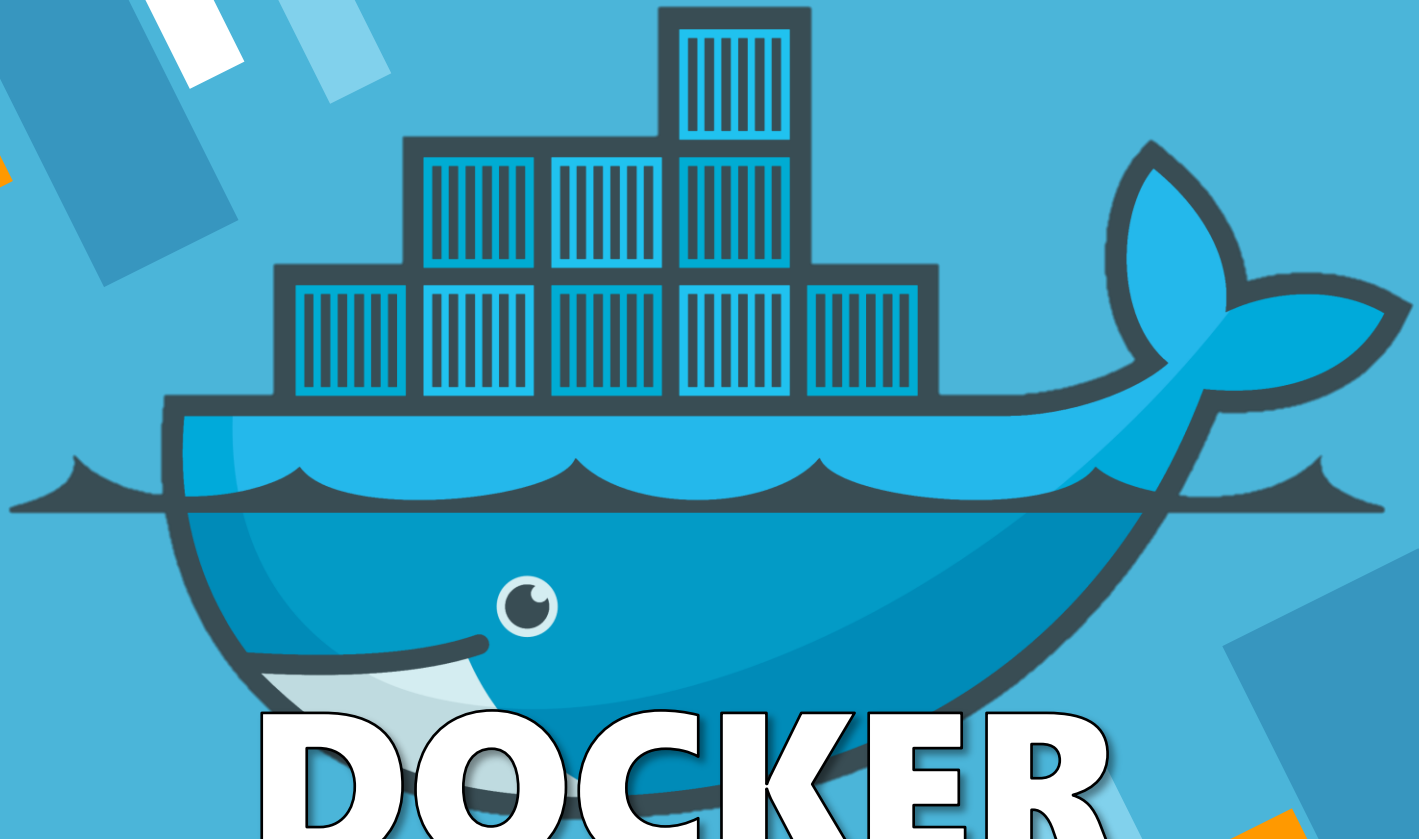


DOCKER

Léo Schirvanian – Josselin Anisset

QU'EST-CE QUE DOCKER ?

- **Lancer des applications dans des conteneurs**

- **Lancer des applications dans des conteneurs**
- **Empaqueter une application et ses dépendances**

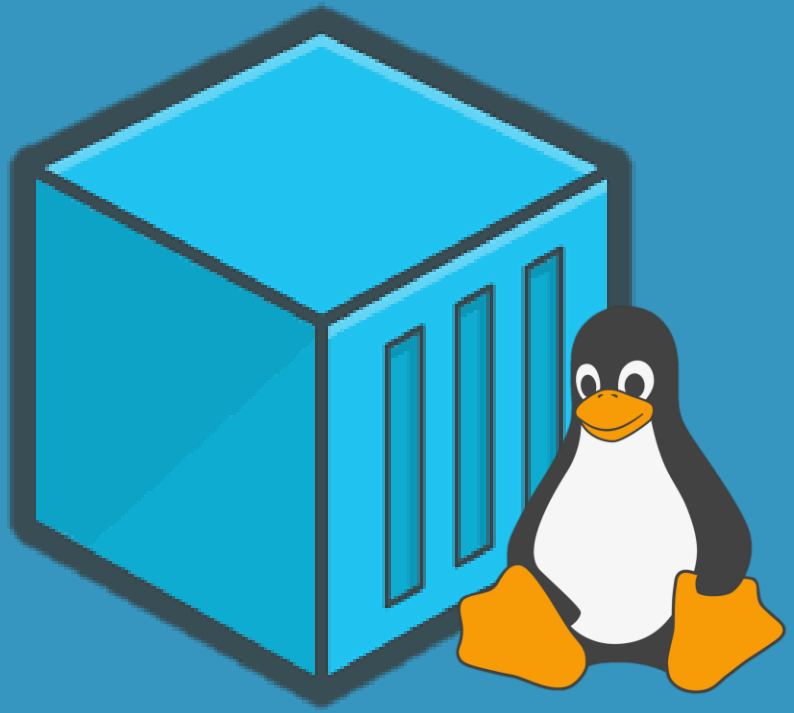
- Lancer des applications dans des conteneurs
- Empaqueter une application et ses dépendances
- Scalable, portable et facilement déployable

- Lancer des applications dans des conteneurs
- Empaqueter une application et ses dépendances
- Scalable, portable et facilement déployable
- Mieux qu'une machine virtuelle

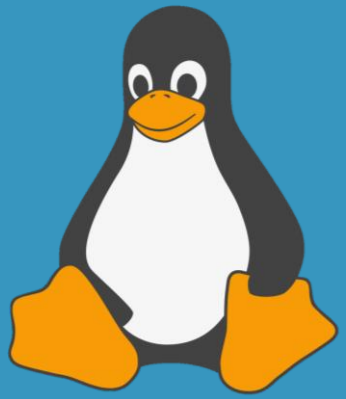
- Lancer des applications dans des conteneurs
- Empaqueter une application et ses dépendances
- Scalable, portable et facilement déployable
- Mieux qu'une machine virtuelle
- Réconcilie développeur et administrateur système

- Lancer des applications dans des conteneurs
- Empaqueter une application et ses dépendances
- Scalable, portable et facilement déployable
- Mieux qu'une machine virtuelle
- Réconcilie développeur et administrateur système
- Excellent environnement de développement

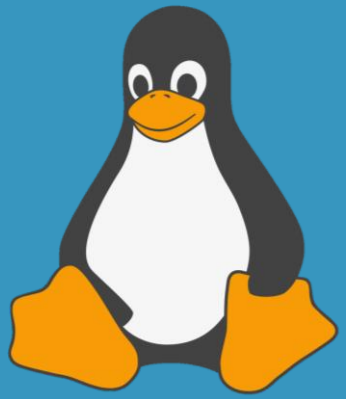
CONTENEUR ?



LXC = Linux Container

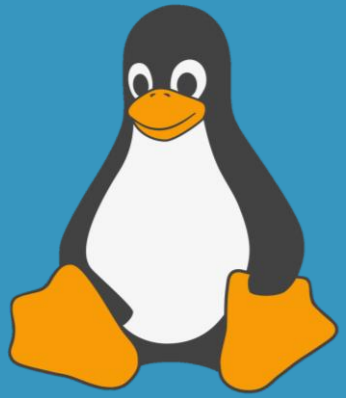


KERNEL LINUX



KERNEL
LINUX

ISOLATION



KERNEL LINUX

ISOLATION

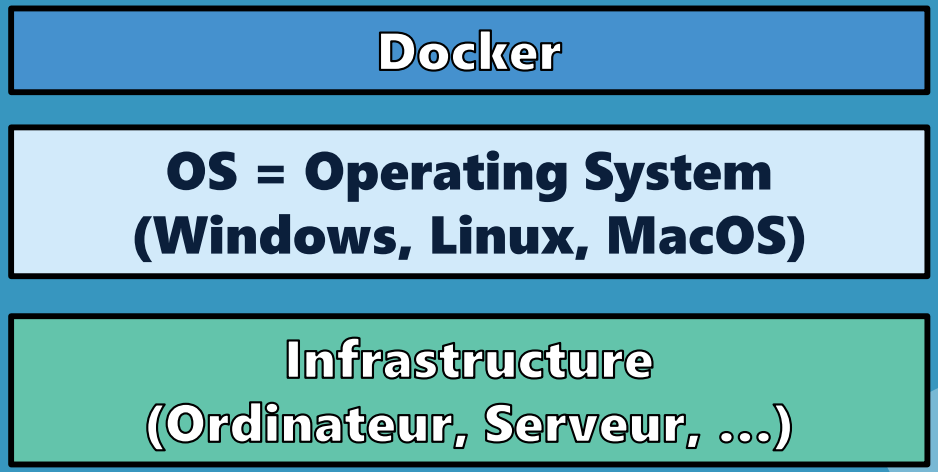
PAS DE DÉPENDANCE

COMMENT ÇA MARCHE ?

Infrastructure
(Ordinateur, Serveur, ...)

OS = Operating System
(Windows, Linux, MacOS)

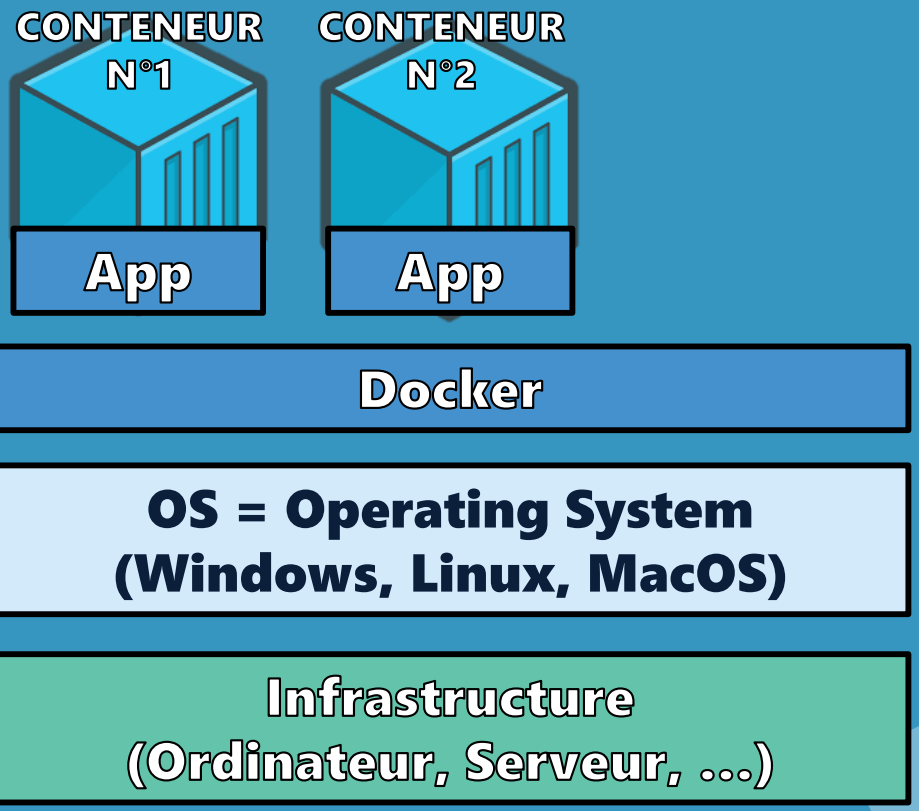
Infrastructure
(Ordinateur, Serveur, ...)

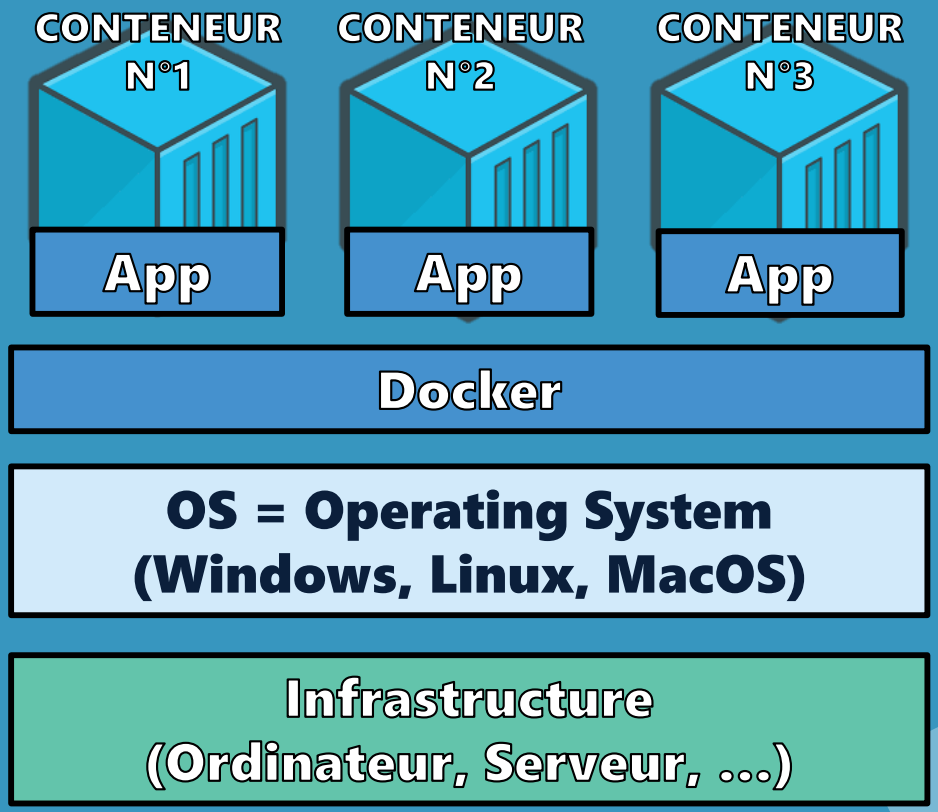

Docker
OS = Operating System
(Windows, Linux, MacOS)
Infrastructure
(Ordinateur, Serveur, ...)

Docker

OS = Operating System
(Windows, Linux, MacOS)

Infrastructure
(Ordinateur, Serveur, ...)


CONTENEUR N°1
CONTENEUR N°2
App
App
Docker
OS = Operating System
(Windows, Linux, MacOS)
Infrastructure
(Ordinateur, Serveur, ...)


CONTENEUR N°1
CONTENEUR N°2
CONTENEUR N°3
App
App
App
Docker
OS = Operating System
(Windows, Linux, MacOS)
Infrastructure
(Ordinateur, Serveur, ...)

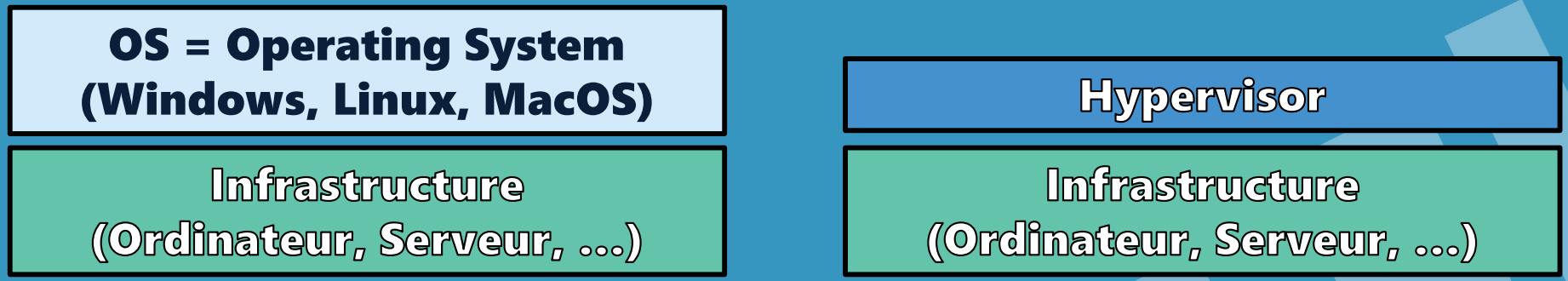
QUELLE DIFFÉRENCE AVEC LES MACHINES VIRTUELLES ?

Infrastructure
(Ordinateur, Serveur, ...)

CONTENEUR

Infrastructure
(Ordinateur, Serveur, ...)

MACHINE VIRTUELLE



CONTENEUR **MACHINE VIRTUELLE**

VM 1

OS

Hypervisor

Infrastructure
(Ordinateur, Serveur, ...)

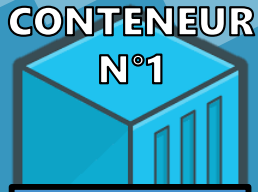
MACHINE VIRTUELLE

Docker

OS = Operating System
(Windows, Linux, MacOS)

Infrastructure
(Ordinateur, Serveur, ...)

CONTENEUR



App

Docker

OS = Operating System
(Windows, Linux, MacOS)

Infrastructure
(Ordinateur, Serveur, ...)

CONTENEUR

VM 1

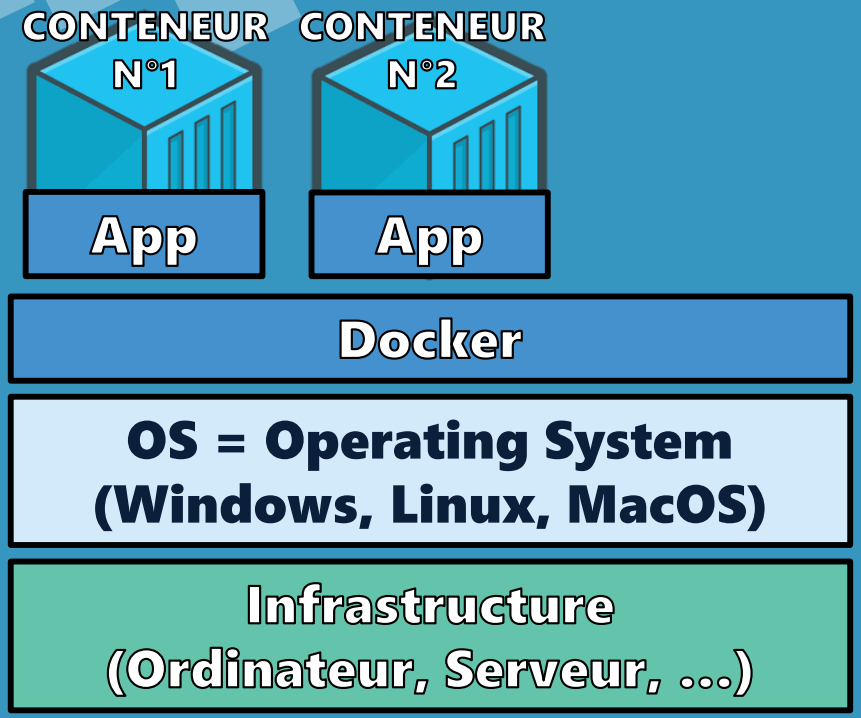
App

OS

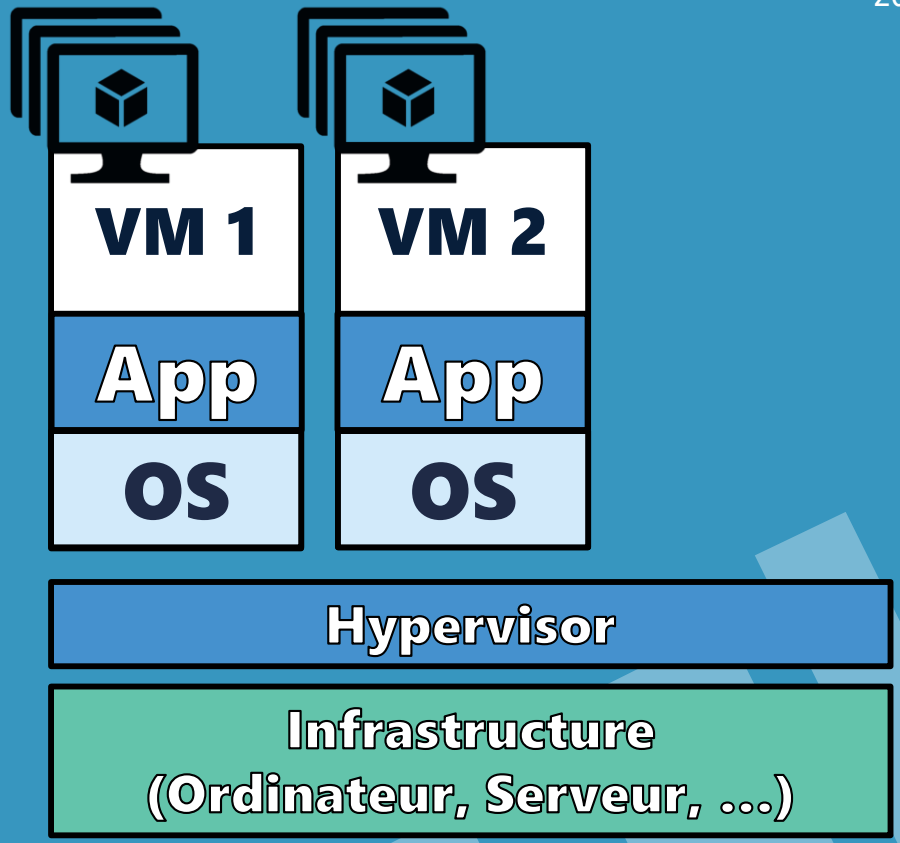
Hypervisor

Infrastructure
(Ordinateur, Serveur, ...)

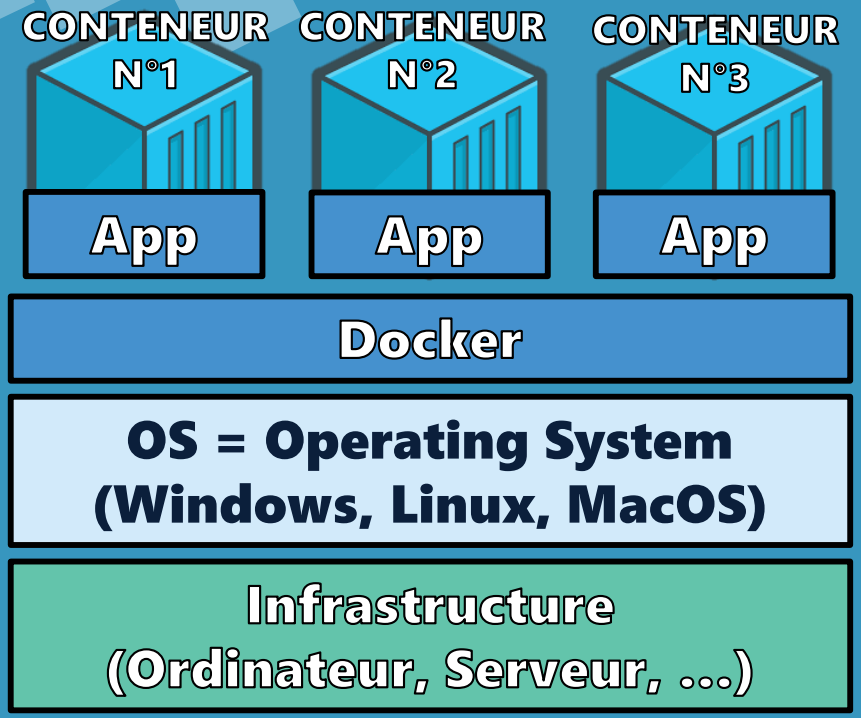
MACHINE VIRTUELLE



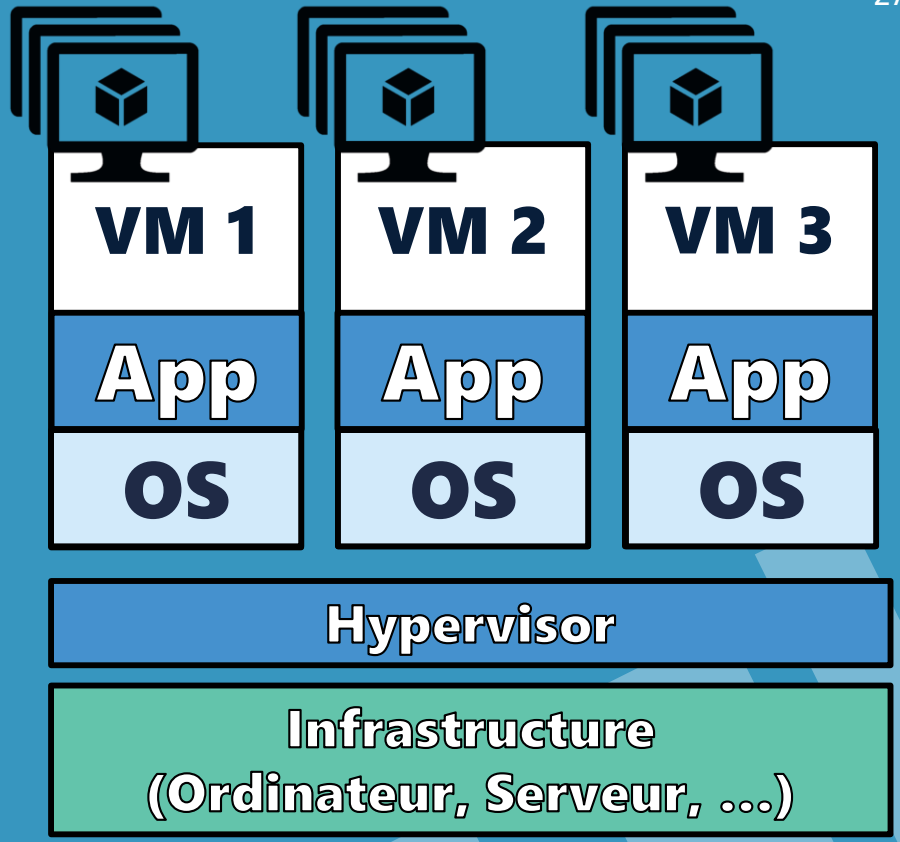
CONTENEUR



MACHINE VIRTUELLE



CONTENEUR



MACHINE VIRTUELLE

POURQUOI MAINTENANT ?

DÉVELOPPEUR

CODE

DÉVELOPPEUR

APPLICATION

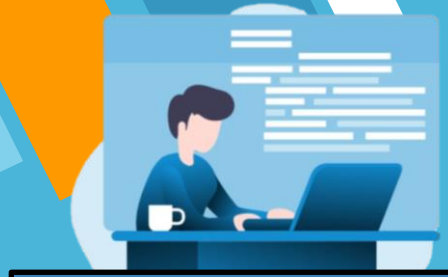
DÉVELOPPEUR

CODE

APPLICATION

DÉPEND

- LIBRAIRIES
- FRAMEWORK
- VERSION
- TECHNOLOGIES



DÉVELOPPEUR

CODE

APPLICATION

DÉPEND

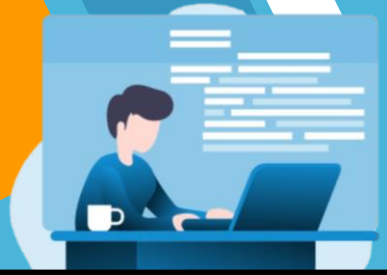
LIBRAIRIES

FRAMEWORK

VERSION

TECHNOLOGIES

ADMINISTRATEUR SYSTÈME



DÉVELOPPEUR

CODE

APPLICATION

DÉPEND

LIBRAIRIES

FRAMEWORK

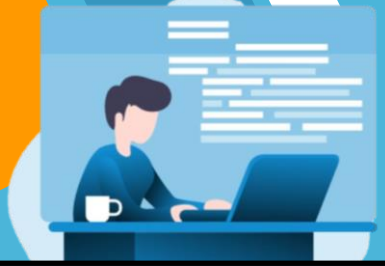
VERSION

TECHNOLOGIES

ADMINISTRATEUR SYSTÈME

INSTALLE

RÉSEAU



DÉVELOPPEUR

CODE

APPLICATION

DÉPEND

LIBRAIRIES

FRAMEWORK

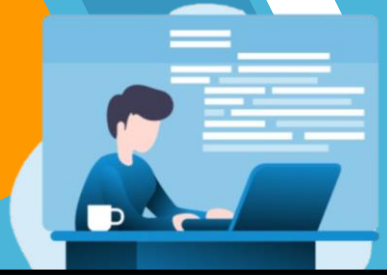
VERSION

TECHNOLOGIES

ADMINISTRATEUR SYSTÈME

INSTALLE

RÉSEAU



DÉVELOPPEUR

CODE

APPLICATION

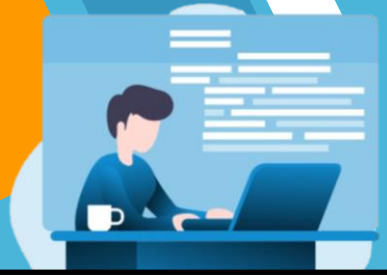
DÉPEND

LIBRAIRIES
FRAMEWORK
VERSION
TECHNOLOGIES

RÉSEAU

INSTALLE

ADMINISTRATEUR SYSTÈME



DÉVELOPPEUR

CODE

APPLICATION

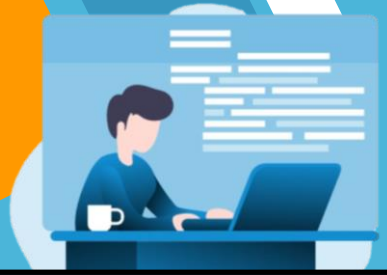
DÉPEND

LIBRAIRIES
FRAMEWORK
VERSION
TECHNOLOGIES

RÉSEAU

INSTALLE

ADMINISTRATEUR SYSTÈME



DÉVELOPPEUR

CODE

APPLICATION

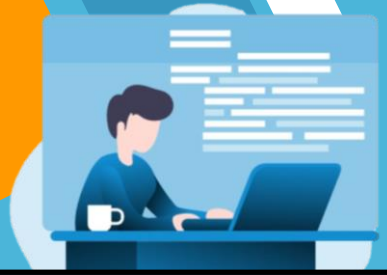
DÉPEND

LIBRAIRIES
FRAMEWORK
VERSION
TECHNOLOGIES

RÉSEAU

INSTALLE

ADMINISTRATEUR SYSTÈME



DÉVELOPPEUR

CODE

APPLICATION

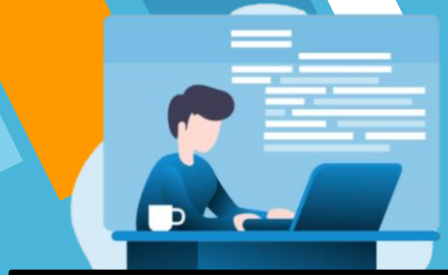
DÉPEND

LIBRAIRIES
FRAMEWORK
VERSION
TECHNOLOGIES

RÉSEAU

INSTALLE

ADMINISTRATEUR SYSTÈME



DÉVELOPPEUR

CODE

APPLICATION

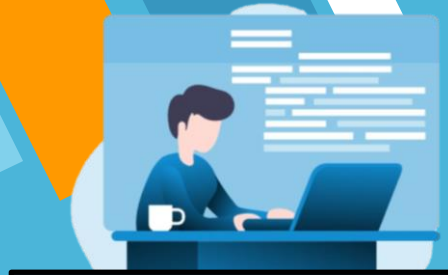
DÉPEND

LIBRAIRIES
FRAMEWORK
VERSION
TECHNOLOGIES

RÉSEAU

INSTALLE

ADMINISTRATEUR SYSTÈME



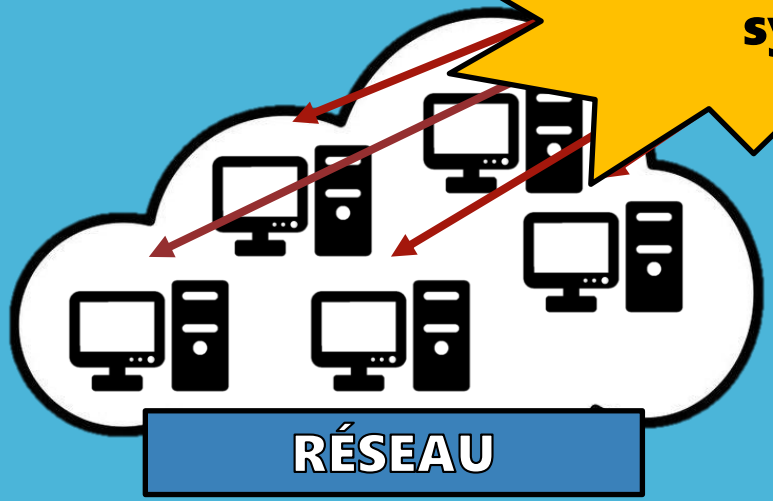
DÉVELOPPEUR

CODE

DÉPEND

LIBRAIRIES
FRAMEWORK
VERSION
TECHNOLOGIES

Conflit développeur vs administrateur système



RÉSEAU

INSTALLE

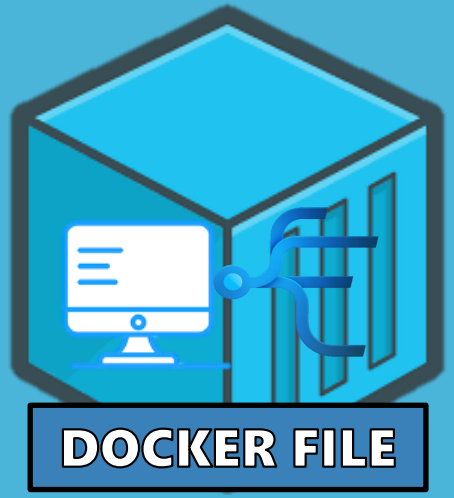
ADMINISTRATEUR SYSTÈME

ET AVEC DOCKER ?

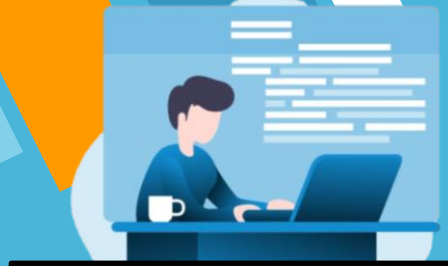
DÉVELOPPEUR

DÉVELOPPEUR

CODE



DOCKER FILE



DÉVELOPPEUR

CODE

DOCKER FILE

Application
+
Dépendances

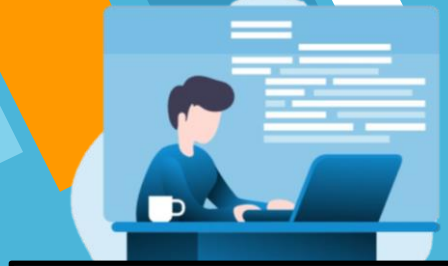
DÉVELOPPEUR

CODE

DOCKER FILE

Application
+
Dépendances

ADMINISTRATEUR SYSTÈME



DÉVELOPPEUR

CODE

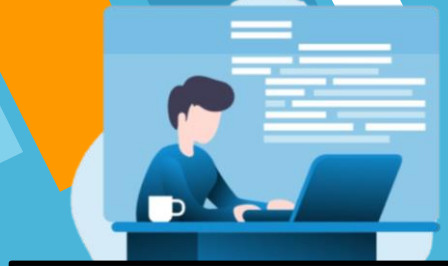
DOCKER FILE

Application
+
Dépendances

ADMINISTRATEUR SYSTÈME

INSTALLE

RÉSEAU



DÉVELOPPEUR

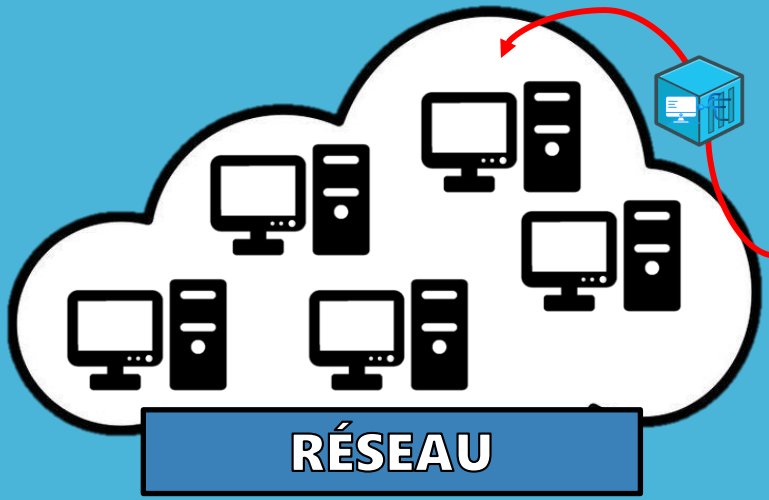
CODE

DOCKER FILE

Application
+
Dépendances

ADMINISTRATEUR SYSTÈME

INSTALLE



RÉSEAU

DÉVELOPPEUR

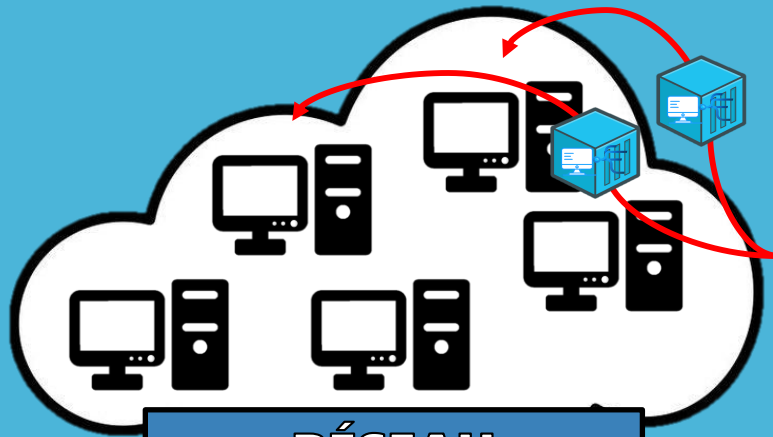
CODE

DOCKER FILE

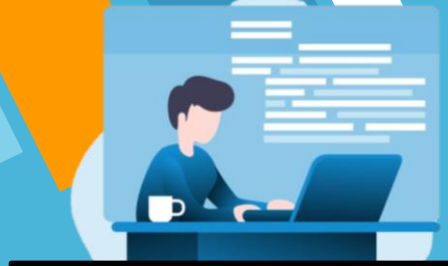
Application
+
Dépendances

ADMINISTRATEUR SYSTÈME

INSTALLE



RÉSEAU



DÉVELOPPEUR

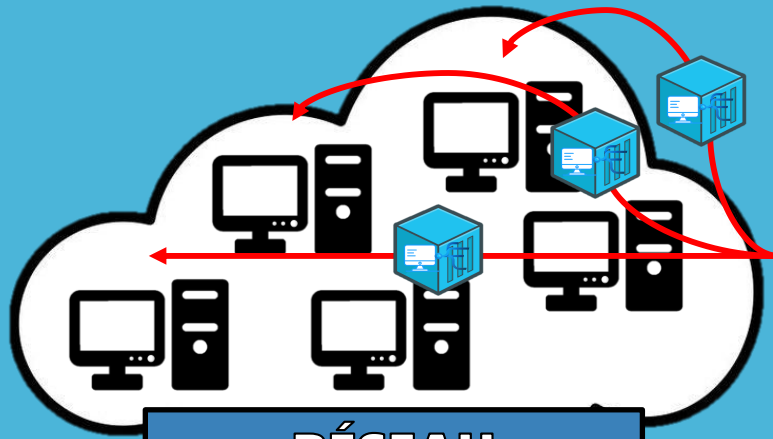
CODE

DOCKER FILE

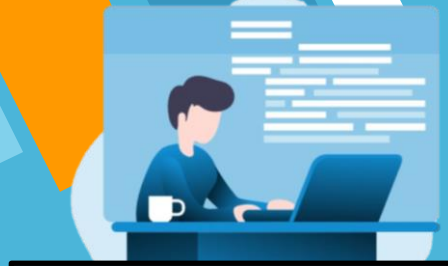
Application
+
Dépendances

ADMINISTRATEUR SYSTÈME

INSTALLE



RÉSEAU



DÉVELOPPEUR

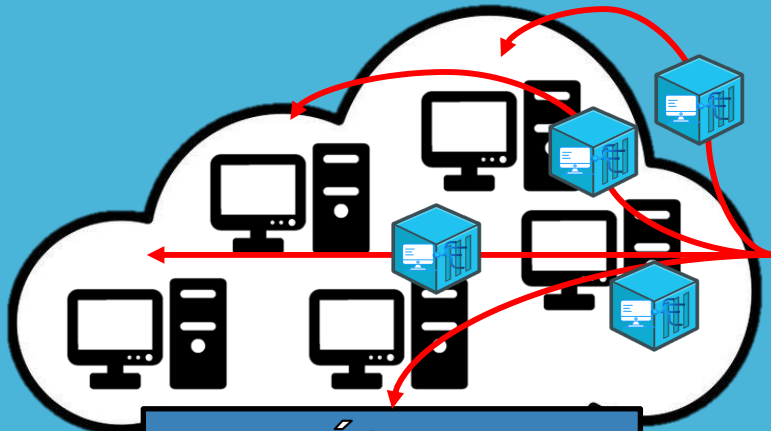
CODE

DOCKER FILE

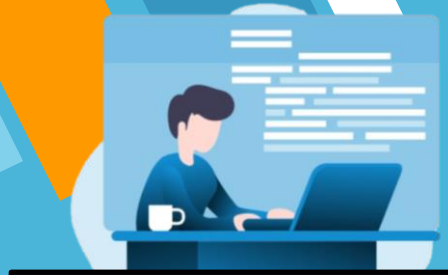
Application
+
Dépendances

ADMINISTRATEUR SYSTÈME

INSTALLE



RÉSEAU

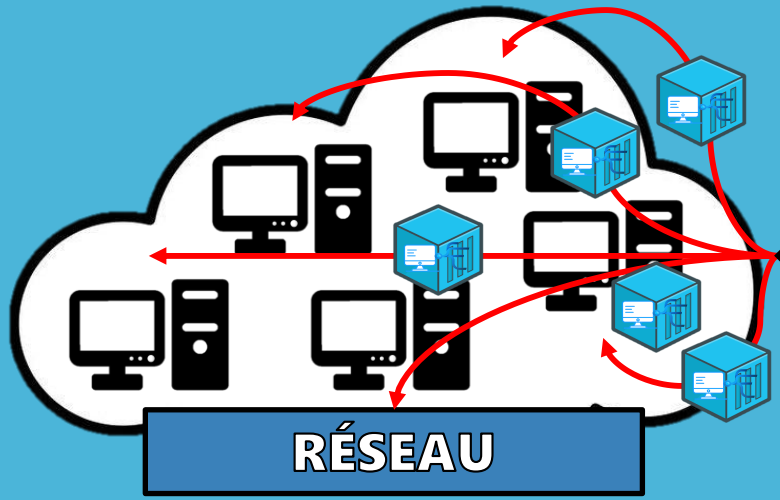


DÉVELOPPEUR

CODE

DOCKER FILE

Application
+
Dépendances

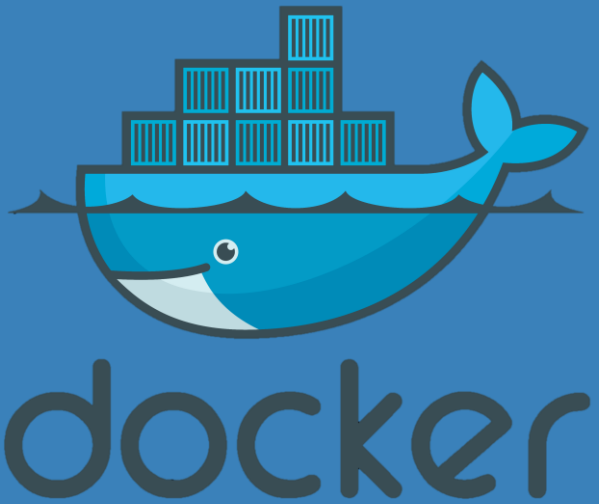


RÉSEAU

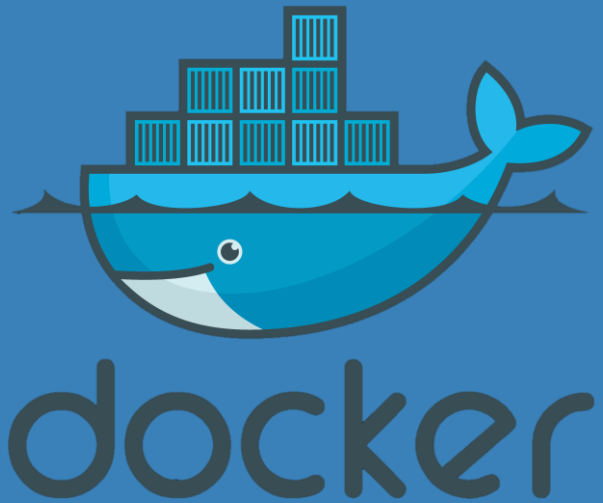
INSTALLE

ADMINISTRATEUR
SYSTÈME

CONCRÈTEMENT COMMENT UTILISER DOCKER SUR UN PROJET ?

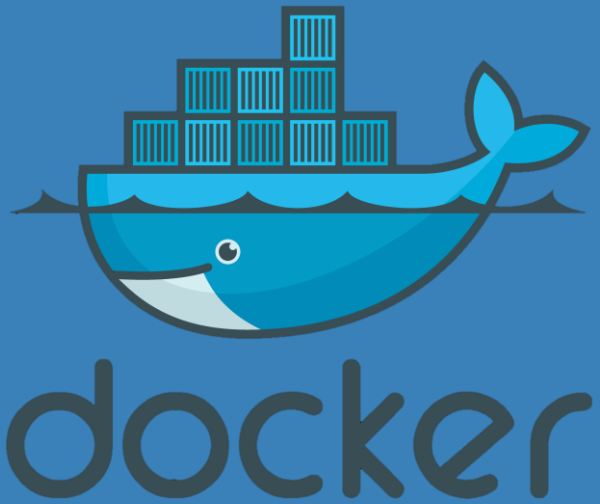

docker

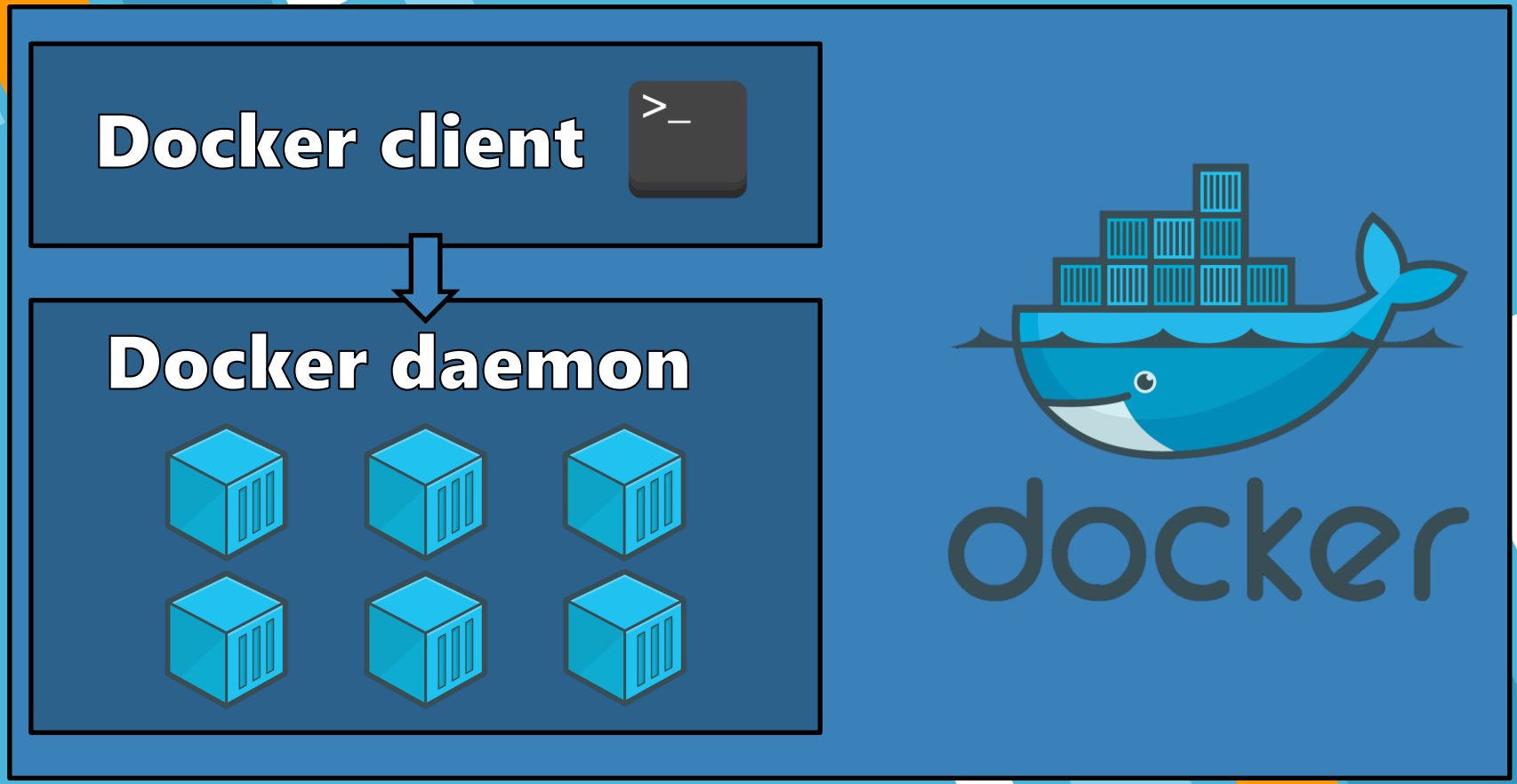
Docker client



Docker client

Docker daemon




Docker client
>_
Docker daemon
docker

Kernel linux

Kernel linux

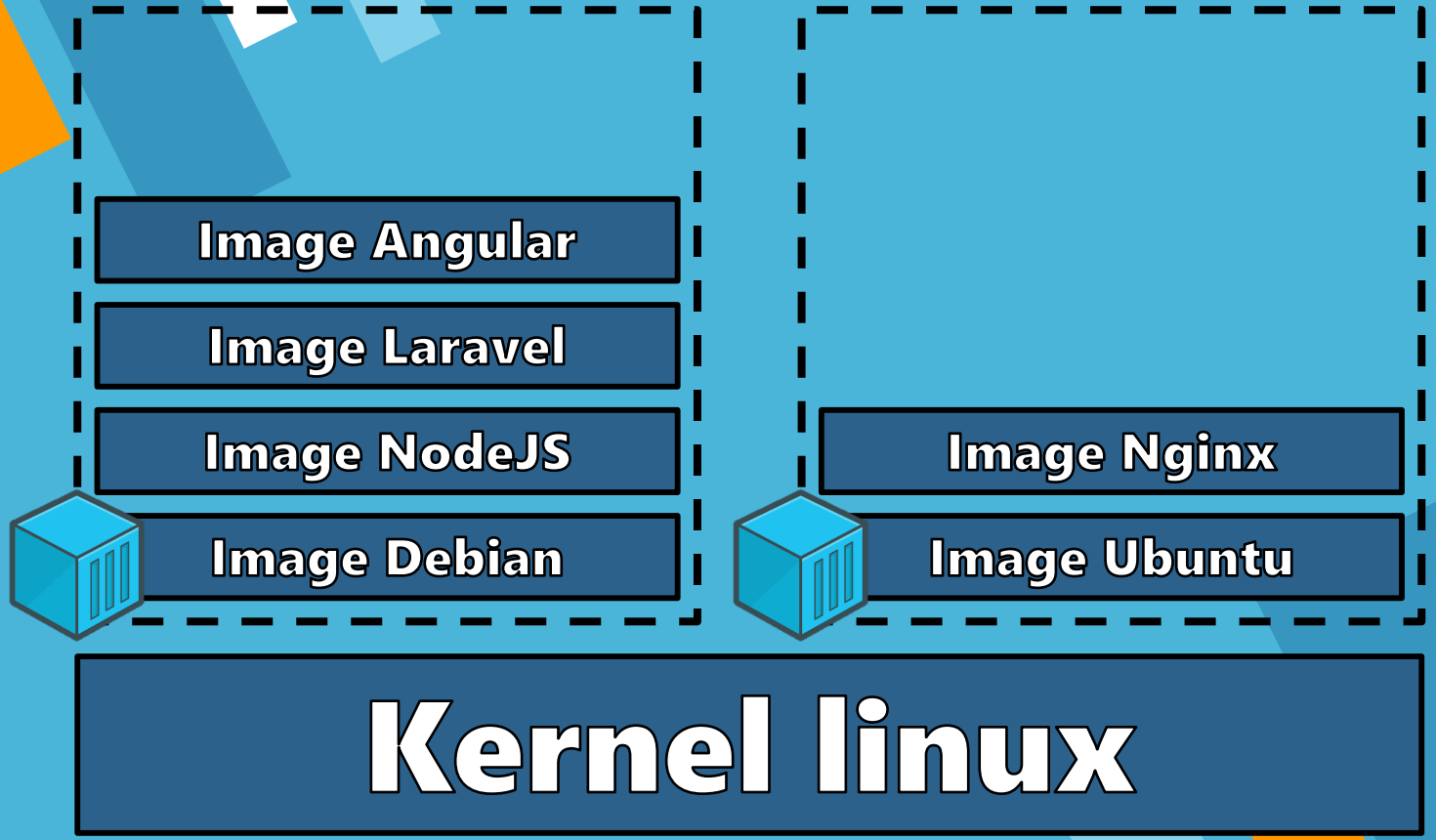
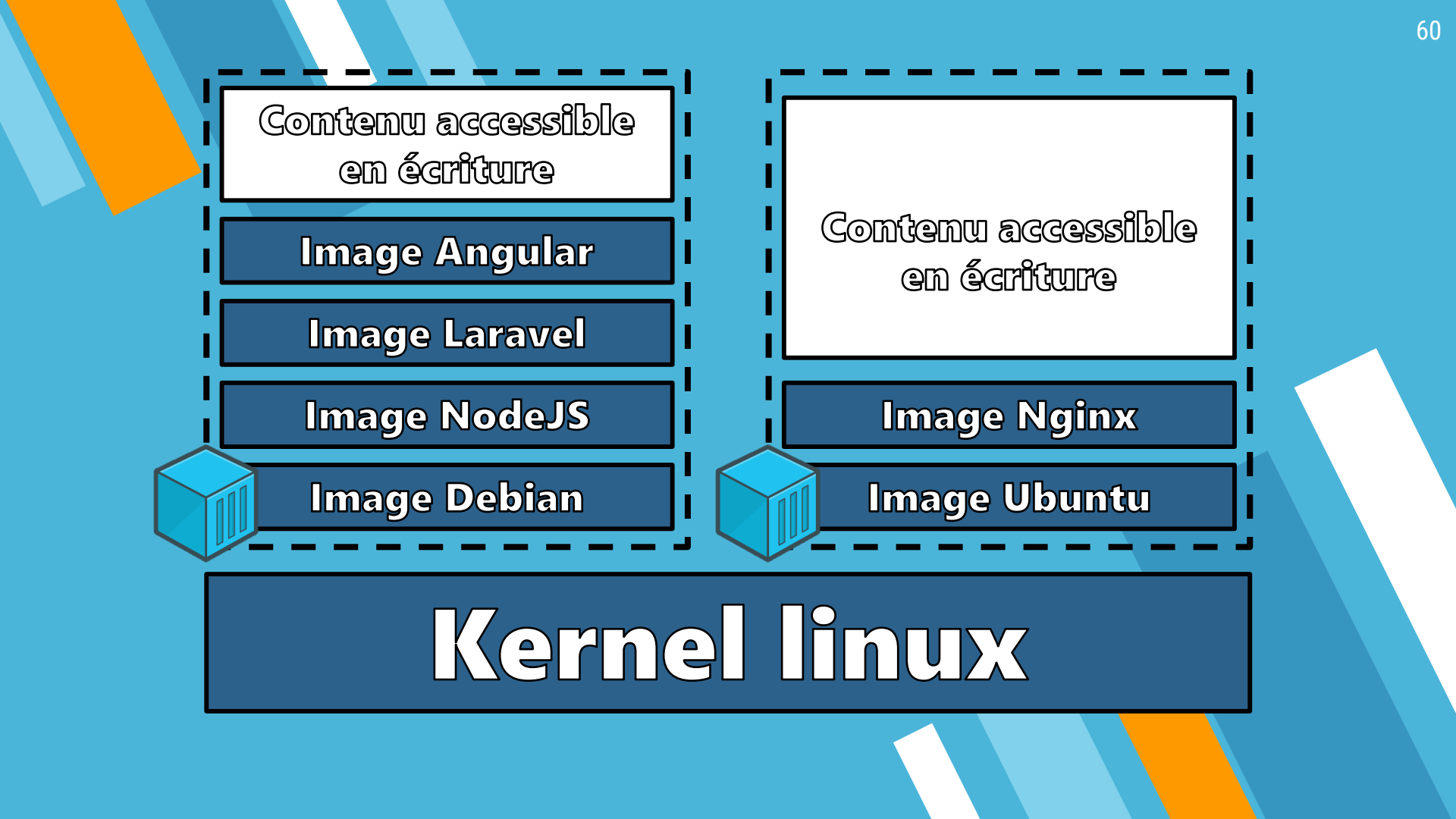
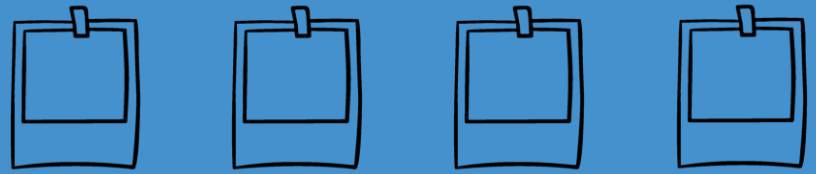


Image Angular
Image Laravel
Image NodeJS
Image Debian
Image Nginx
Image Ubuntu
Kernel linux


Contenu accessible en écriture
Image Angular
Image Laravel
Image NodeJS
Image Debian
Contenu accessible en écriture
Image Nginx
Image Ubuntu
Kernel linux

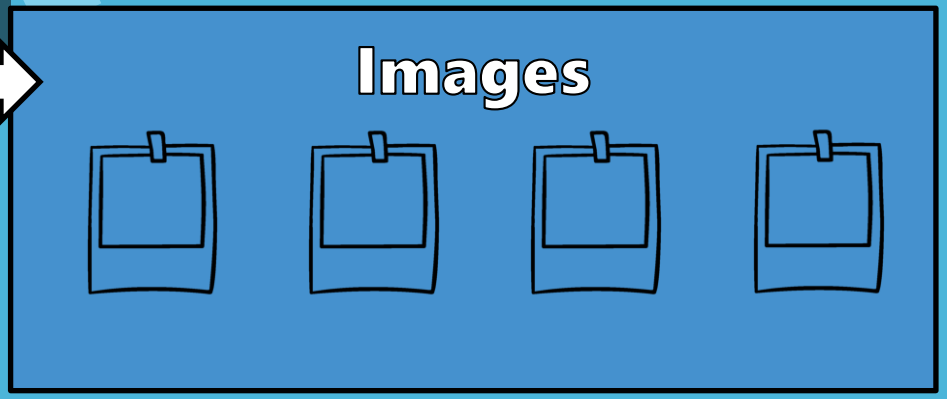
Images

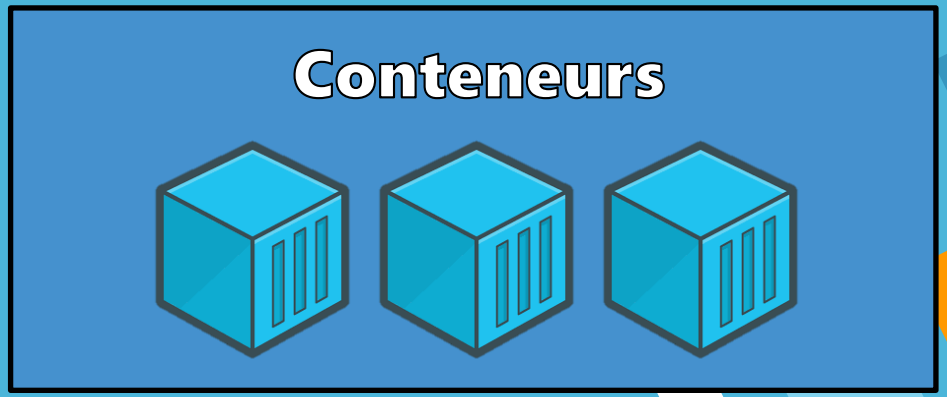


Docker File

Conteneurs

PULL





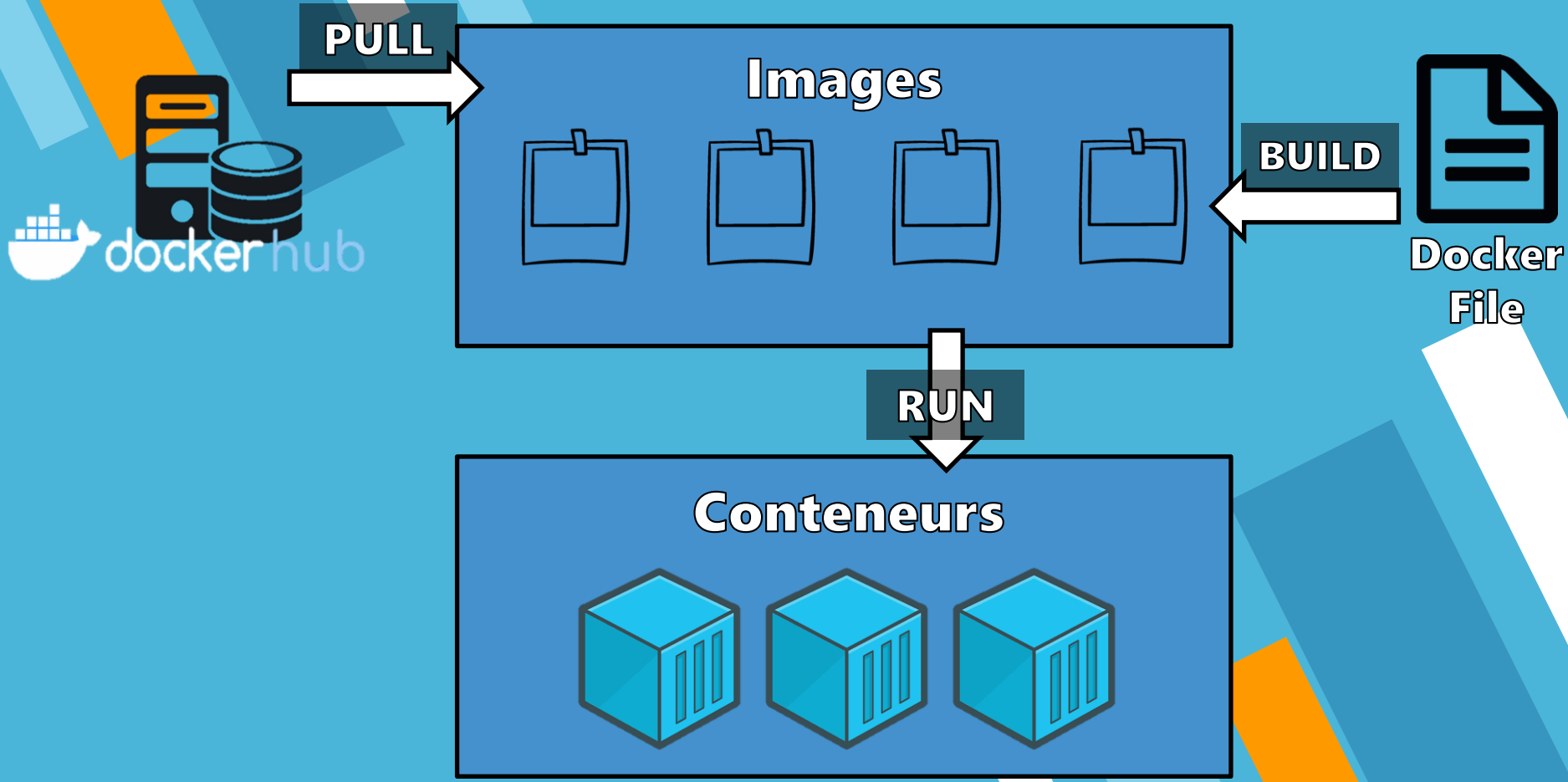
PULL

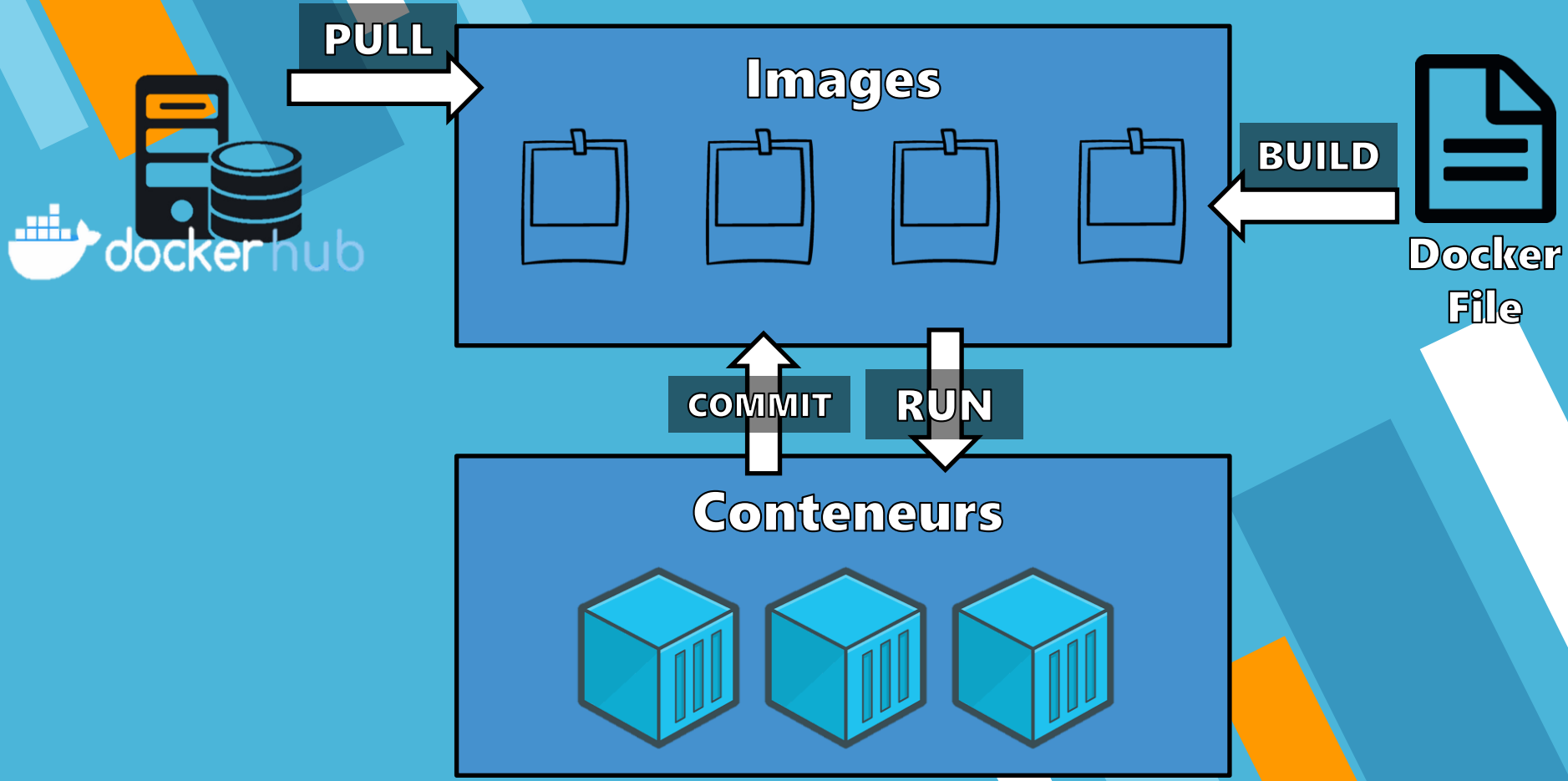
Images

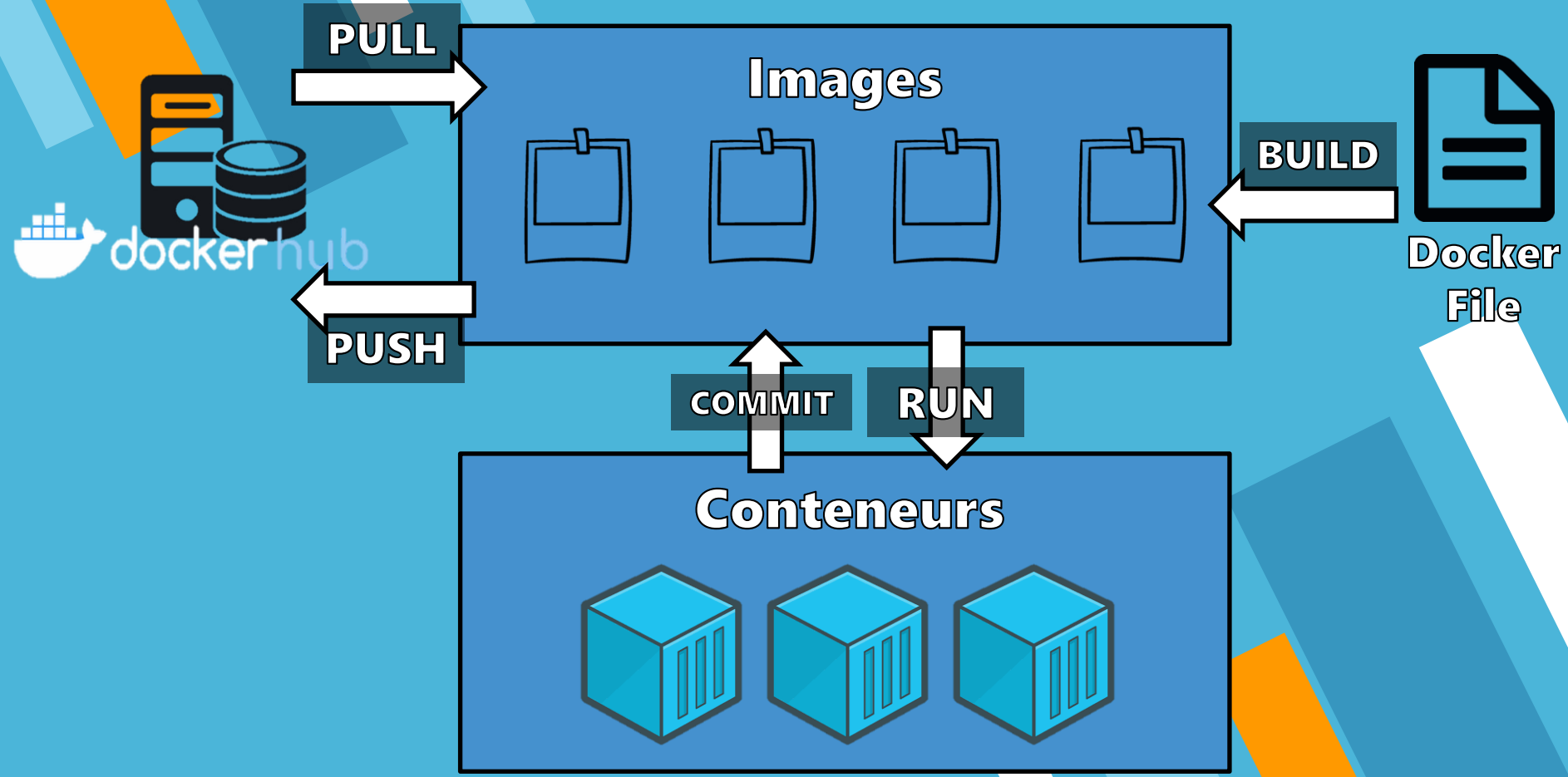
BUILD

Docker File

Conteneurs


PULL
Images
BUILD
Docker
File
docker hub
RUN
Conteneurs


PULL
Images
BUILD
Docker
File
docker hub
COMMIT
RUN
Conteneurs


PULL
Images
BUILD
Docker
File
docker hub
PUSH
COMMIT
RUN
Conteneurs

MERCI DE VOTRE ATTENTION !