



Docker

Mise en oeuvre et déploiement de conteneurs virtuels

Here is where your formation begins

Le Formateur

Guilian GANSTER

Développeur web & mobile

Freelance depuis 4 ans dans les
technologies du web & mobile.

Formateur depuis 2 ans.

Développeur fullstack React /
React native
Firebase / Strapi

Couteau suisse de l'informatique:
Devops, sysadmin, électronique,
client lourd, ...




Participants

- 01 Quelles sont vos attentes par rapport à cette formation ?
- 02 Les sujets sur React Native que vous voudriez aborder ?
- 03 Les types d'applications que vous développez en React Native ?



Le slack



Vérifiez dans vos email que vous avez
bien reçu le lien d'invitation au
Slack de la formation.

.....





Déroulé de Formation



Horaires & temps de pause

- de 9h à 17h
- Une pause de 20mn le matin, et une pause de 20mn l'après midi.
- A midi, une pause déjeuner de 1h15

- A chaque demi journée, un questionnaire à remplir pour récolter vos retours (équilibre théorie / pratique, notions sur lesquelles revenir, ...)

Notions abordées

J1 - Comprendre Docker

1. Contexte
2. Fonctionnement et terminologie
3. Installation de l'environnement
4. Rappel ligne de commande UNIX
5. Mon premier conteneur + TP
- 6.

J2 - Architecture

7. Les design pattern (HOC, Provider, Reducer, ...)
8. Les contextes
9. Props de rendu, transfert de props & useRef
10. L'async storage
11. La Navigation
12. Optimisation
13. Gestion des erreurs (développement & production)



01

Contexte



compatibilité



PC
Développement
(windows)



PC QA
(mac os)



Serveur préprod
Ubuntu S 22.04



Serveur
production
Ubuntu S 20.04





Isolation

MySQL 8

PHP 7

Application 1

MySQL 6

PHP 5

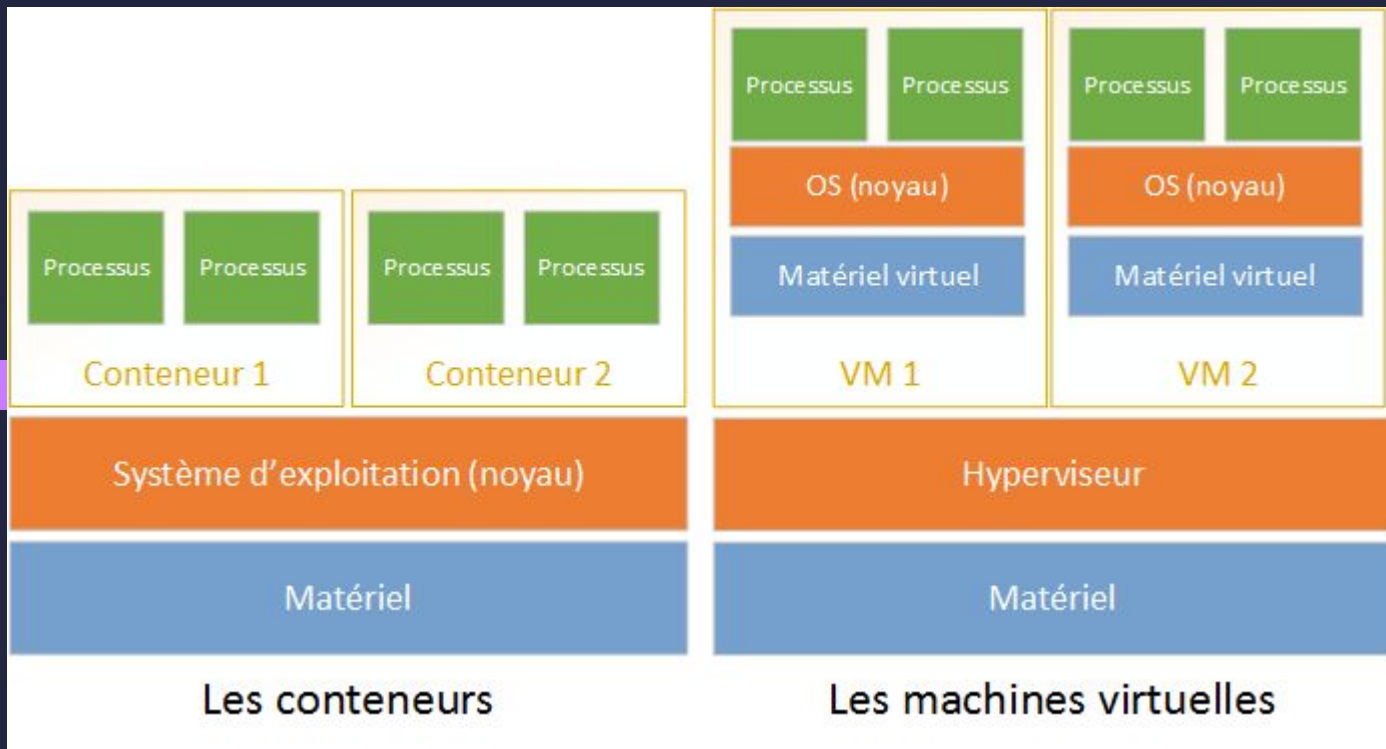
Application 2

...

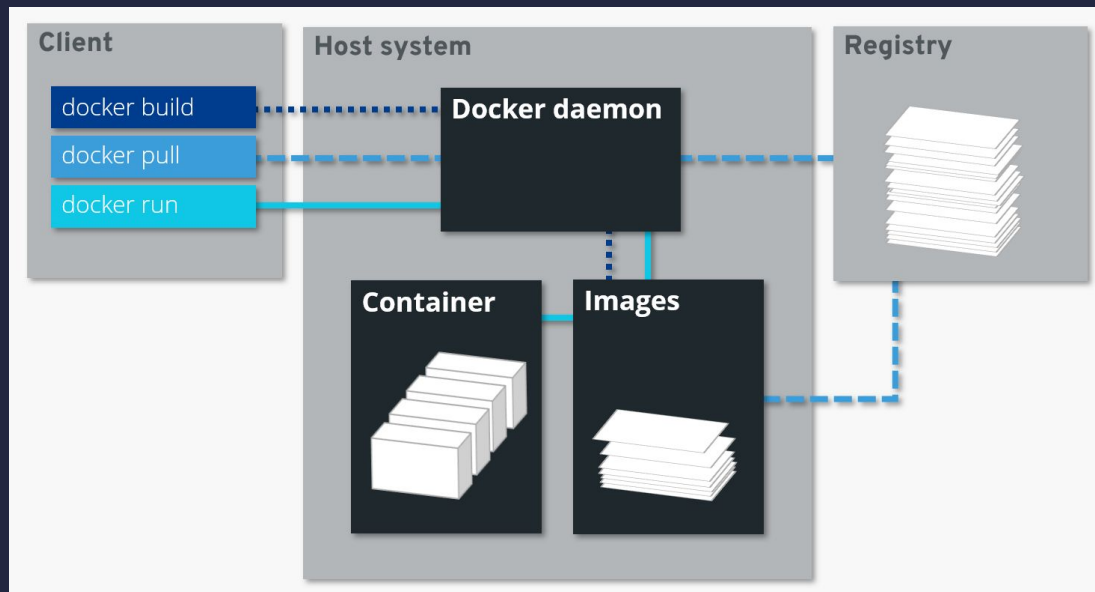




Performances



Simplicité d'utilisation



Votre premier conteneur

01

docker run <image> <...>

Crée un conteneur à partir d'une image

Astuces:

-d = mode détaché (daemon)

-name <name> = donne un nom a son container, plus facile a manipuler que les ids

02

docker ps

Liste les conteneurs démarrés

03

docker stop <id>

Stoppe un conteneur

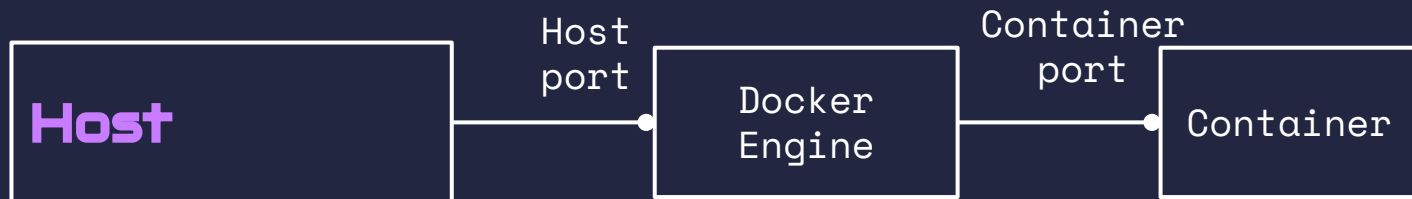
04

docker start <id>

Démarre un conteneur existant

Le port forwarding

```
docker run <image> -p <host port>:<container port>
```



La persistance par les volumes

01

docker volume create <name>

Crée un volume qui persistera même si le conteneur est coupé

02

docker volume ls

Liste les volumes disponibles

03

Docker volume rm <name>

Stoppe un volume

04

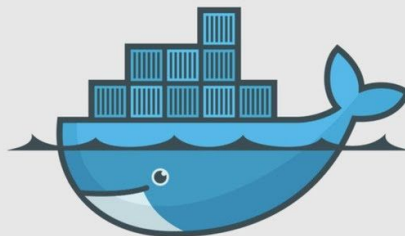
Docker volume inspect

Donne des informations détaillées sur un volume

docker run <image> -v <name>:<container path>

Le registry

Docker Hub



docker



TP

Installer portainer CE
Avec persistance de données
Accessible par le port 80
Redémarre automatiquement au boot /
en cas de plantage

//////////

Executer un script dans un container

01

`docker exec -it <container_id> <path>`

Connecte le terminal au container, de ce fait on peut exécuter des scripts a la main dans la vm

Ctrl+d ou "exit" pour quitter

L'environnement

01

`docker run -e key=value -e key2=value2 <...>`

Ajoute des entrées à l'environnement d'exécution du container. (cf documentation d'image)

02

`docker run --env-file <path> <...>`

Ajoute à l'environnement toutes les clés valeurs du fichier d'env passé en paramètre.

Le linkage

01


`docker run --link <container_name> <...>`

Connecte le terminal au container, de ce fait on peut exécuter des scripts à la main dans la vm



TP

Démarrer un conteneur mysql
Démarrer un conteneur wordpress lié
Accessible sur port 80



Attention aux volumes de persistances
Mysql ne doit pas être exposé

//////////



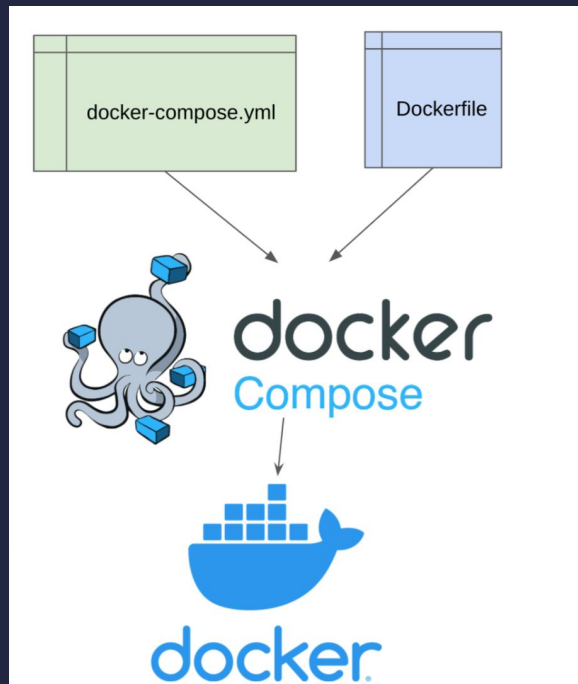
Docker exec

01

`docker exec -it <container_name or id> <command>`

Exécute une commande à l'intérieur d'un container.
Astuce: utiliser "sh" ou "bash" pour se connecter au terminal

docker-compose





TP

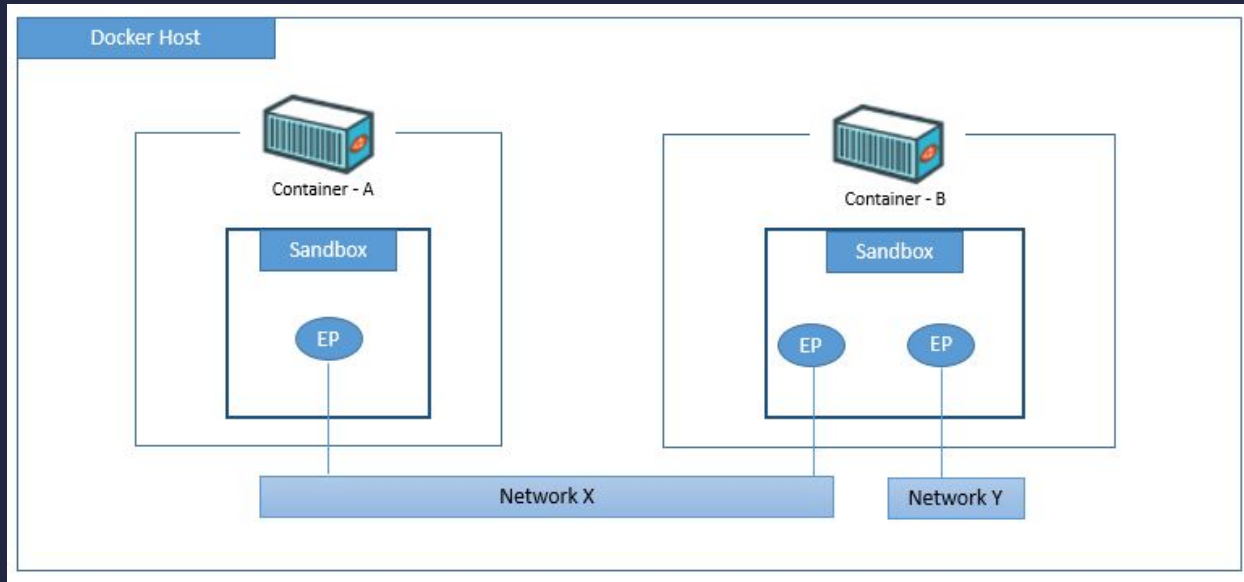


Docker-compose wordpress

• • • • •



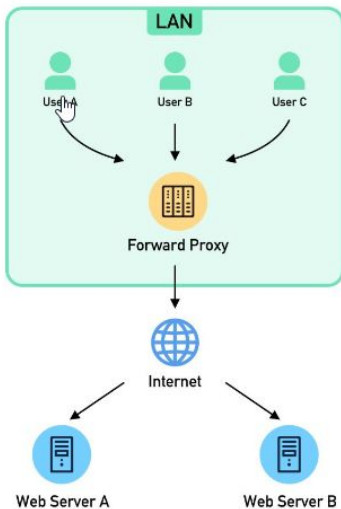
Les networks



Reverse-proxy avec traefik

Forward Proxy

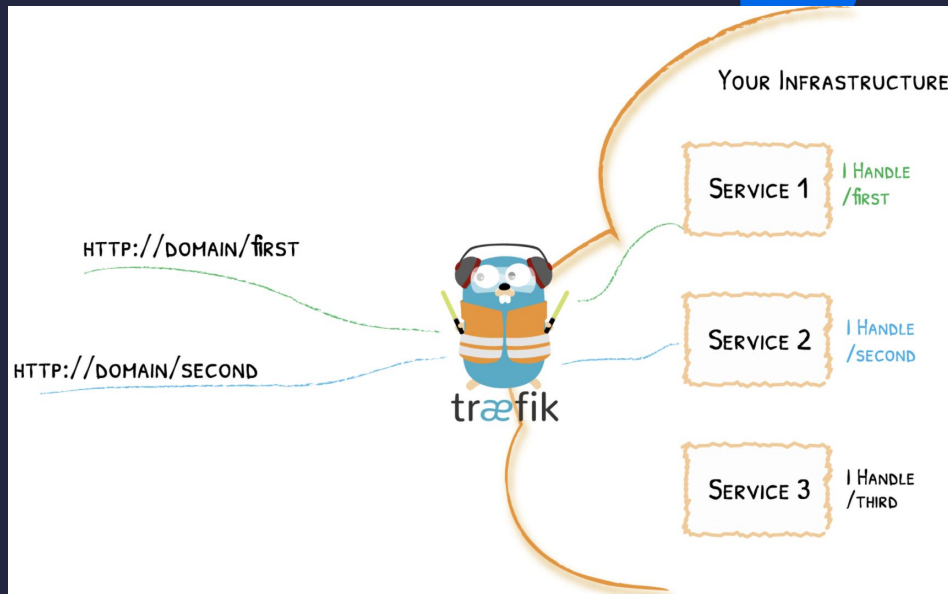
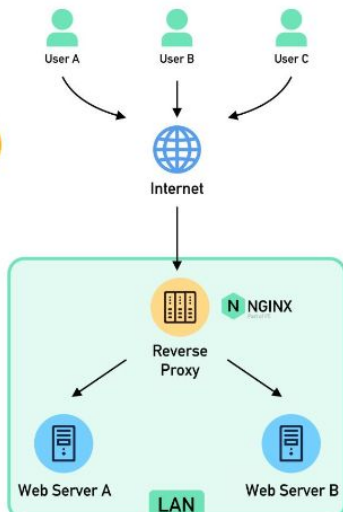
- Avoid browsing restrictions
- Block access to certain content
- Protect user identity online



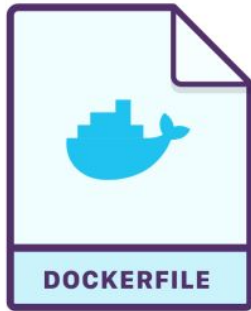
VS

Reverse Proxy

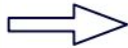
- Load balancing
- Protect from DDos attacks
- Cache static content
- Encrypt and decrypt SSL communications



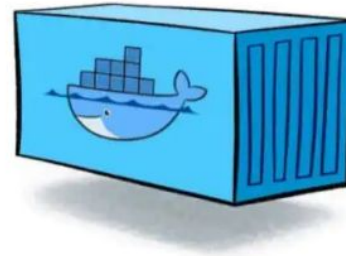
Dockerfile



Docker file



Docker Image



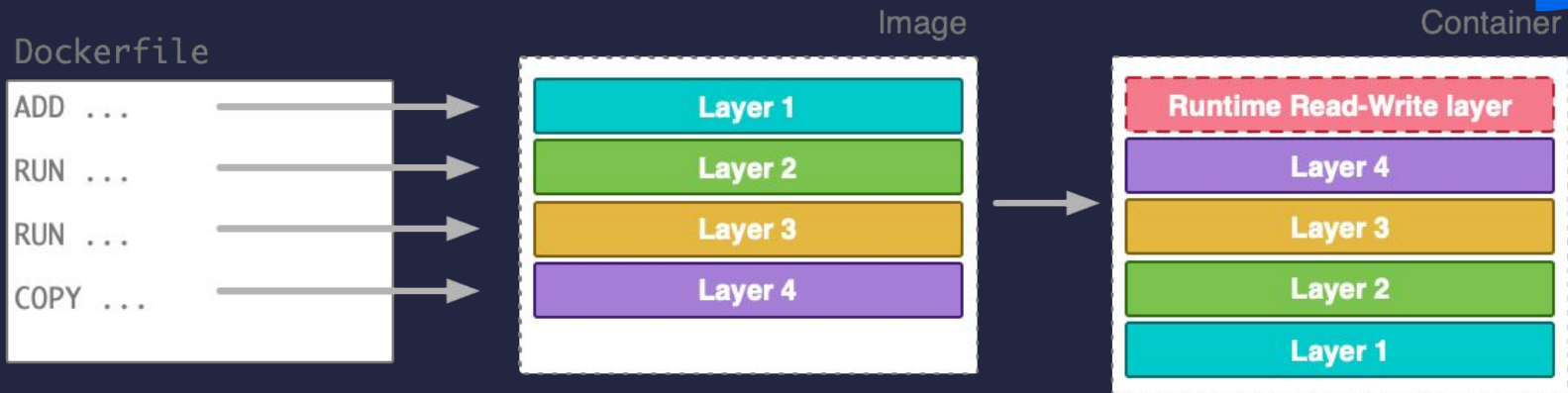
Docker Container

Dockerfile

Builder main commands

command	description
<code>FROM</code> <i>image scratch</i>	base image for the build
<code>MAINTAINER</code> <i>email</i>	name of the maintainer (metadata)
<code>COPY</code> <i>path dst</i>	copy <i>path</i> from the context into the container at location <i>dst</i>
<code>ADD</code> <i>src dst</i>	same as <code>COPY</code> but untar archives and accepts http urls
<code>RUN</code> <i>args...</i>	run an arbitrary command inside the container
<code>USER</code> <i>name</i>	set the default username
<code>WORKDIR</code> <i>path</i>	set the default working directory
<code>CMD</code> <i>args...</i>	set the default command
<code>ENV</code> <i>name value</i>	set an environment variable


Dockerfile - layers



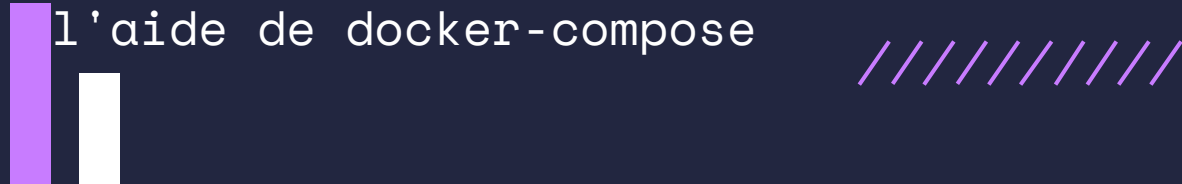


TP

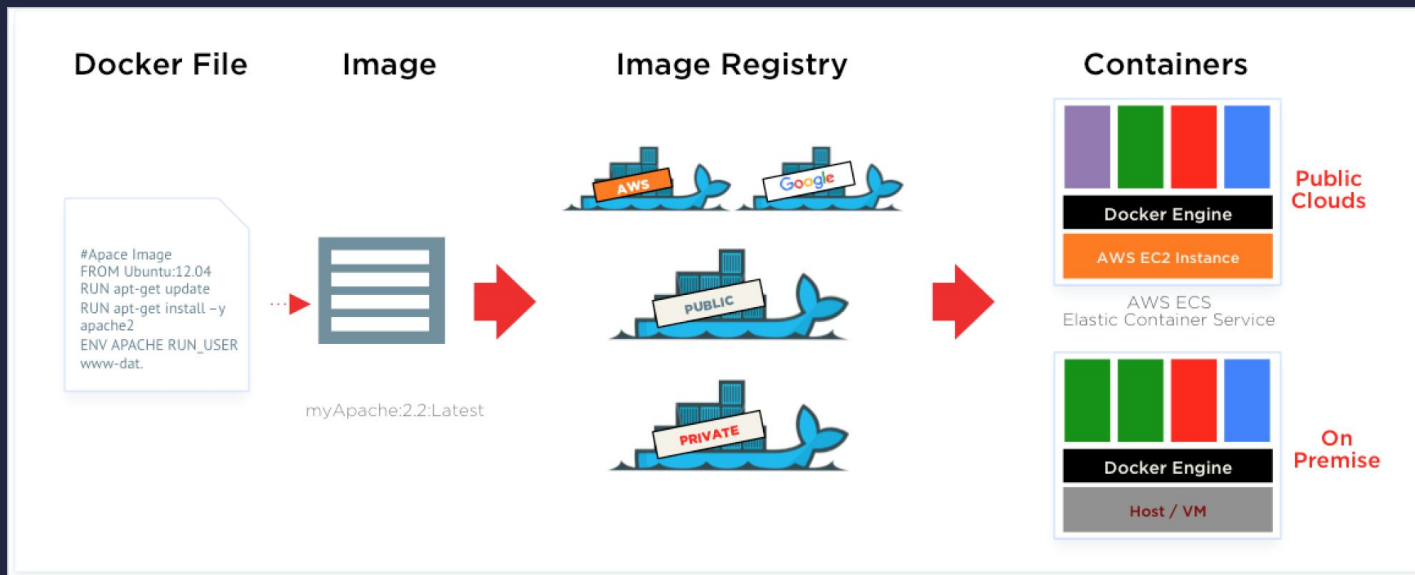
Créer un DockerFile FROM nginx:latest
Permettant le hosting d'un site
statique



Executer votre nouvelle image à
l'aide de docker-compose



Envoi vers le registre



healthcheck



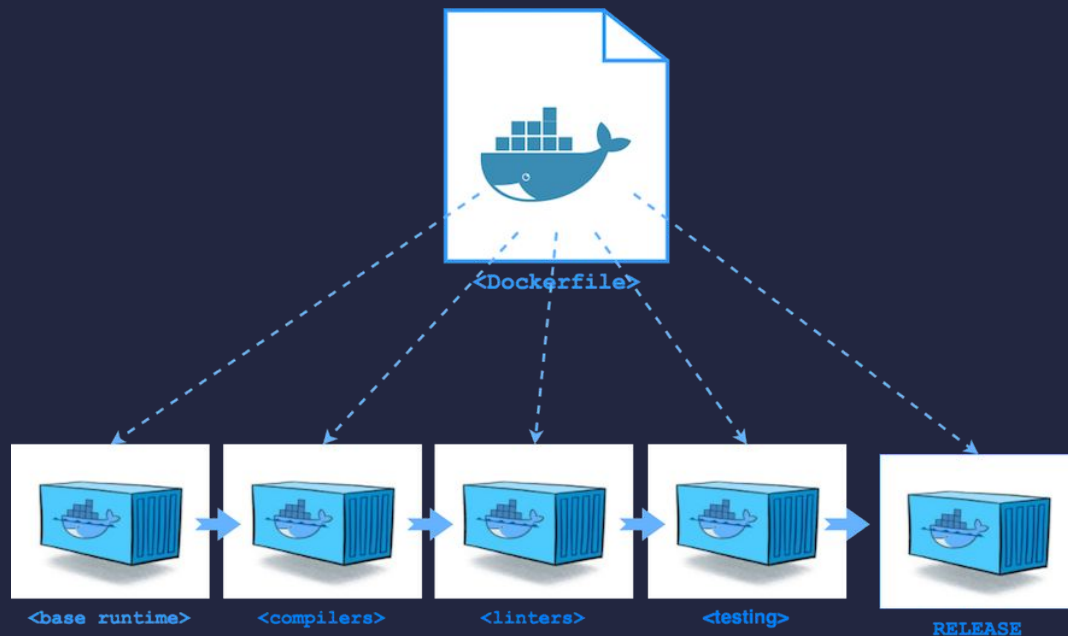
+ Health Check

Docker Compose

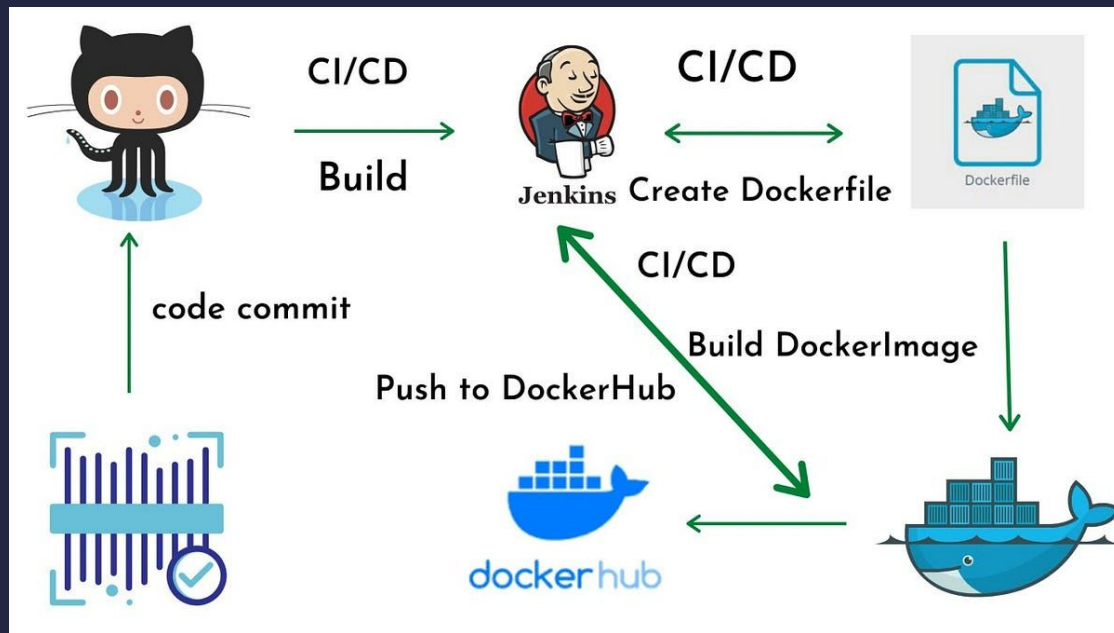
Monitoring



Dockerfile avec build intermédiaire



Intégration dans un pipeline de CI



Bonne pratiques sécurité



Introduction Swarm / Kubernetes

