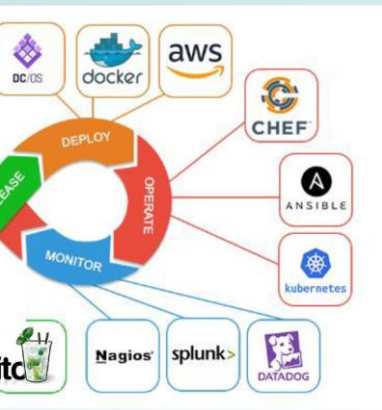


# Atelier de Génie Logiciel

HAI501I  
Cours 5

DevOps

## Docker et la conteneurisation

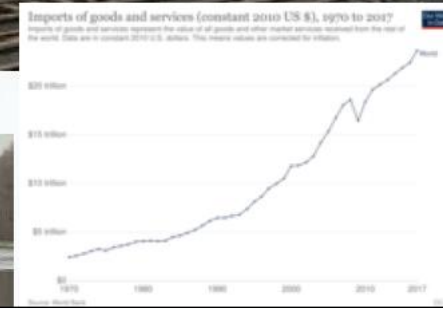


# Conteneur IRL

## Révolution Industrielle & Commerce Mondial

Les dimensions du container 20 pieds sont :

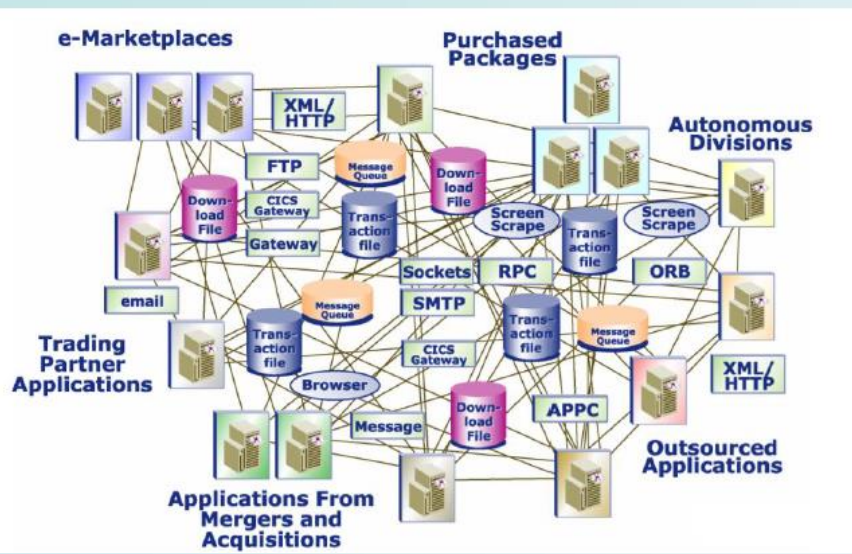
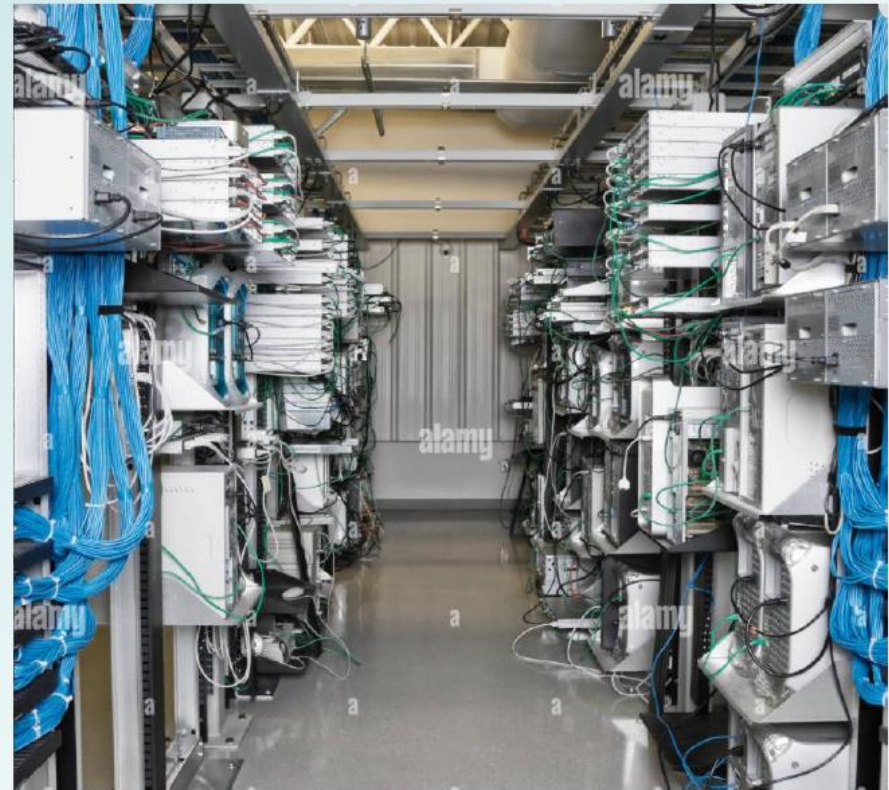
- Longueur : 6m
- Largeur : 2,43m
- Hauteur : 2,59m
- Poids maximum : 1270 kg
- Capacité de stockage : de 30 à 33m<sup>3</sup>





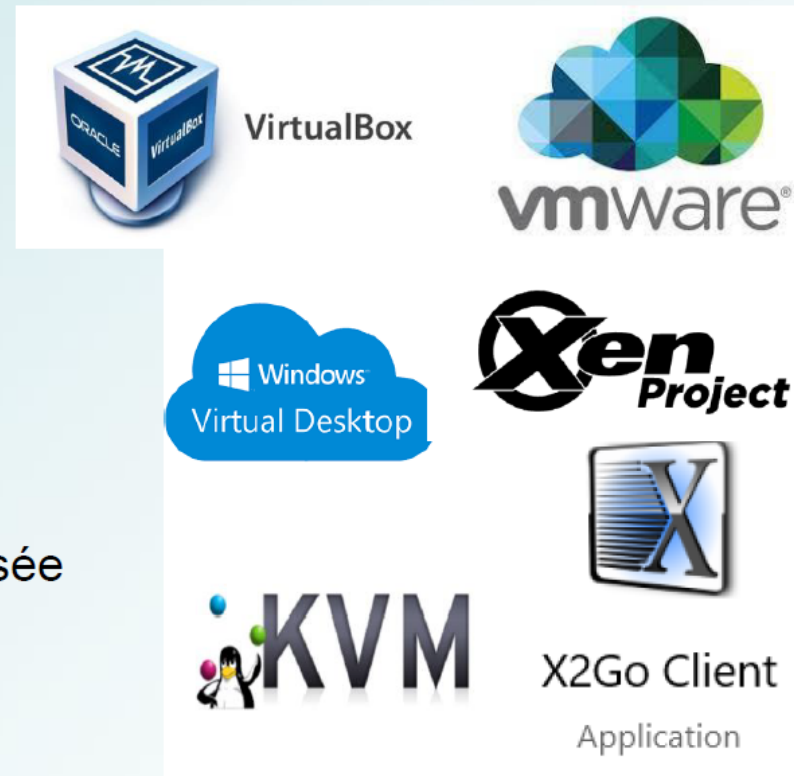
# L'impasse des serveurs / machines et des OS

- Un serveur applicatif pour chaque système d'exploitation
- Un serveur applicatif pour chaque application (sécurité, dimensionnement, version des librairies tierces maîtrisées)
- Un serveur backup en cas de problèmes
- Mutualisation vs QOS



# Machine Virtuelle

- Densité infrastructure
  - 1 seule grosse machine
  - n machines virtuelles
- Déploiement anticipé/facilité (architecture cible)
- Virtualisation Lourde
  - Good
    - Multi OS
    - Isolée
    - Ressource réservée
    - Consommation énergétique optimisée
  - Bad
    - Start lent
    - Ressources réservées





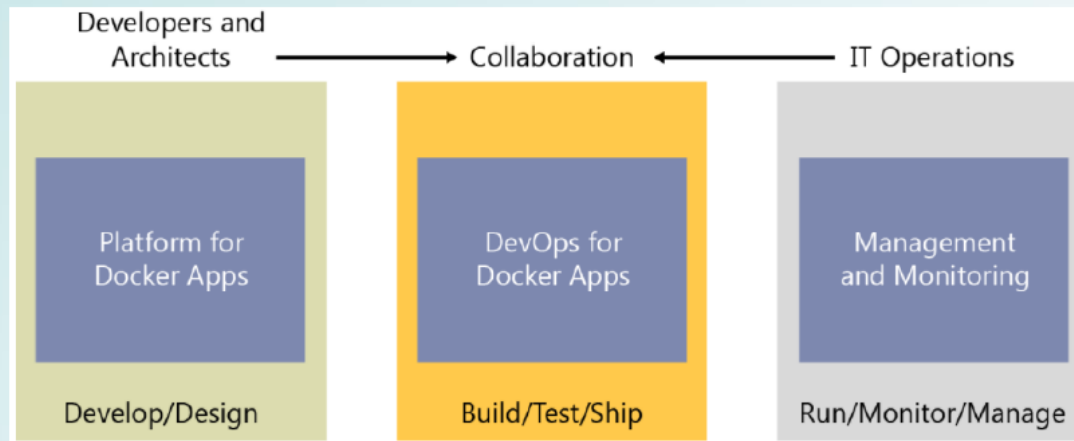
kubernetes

# Conteneur (d'application)



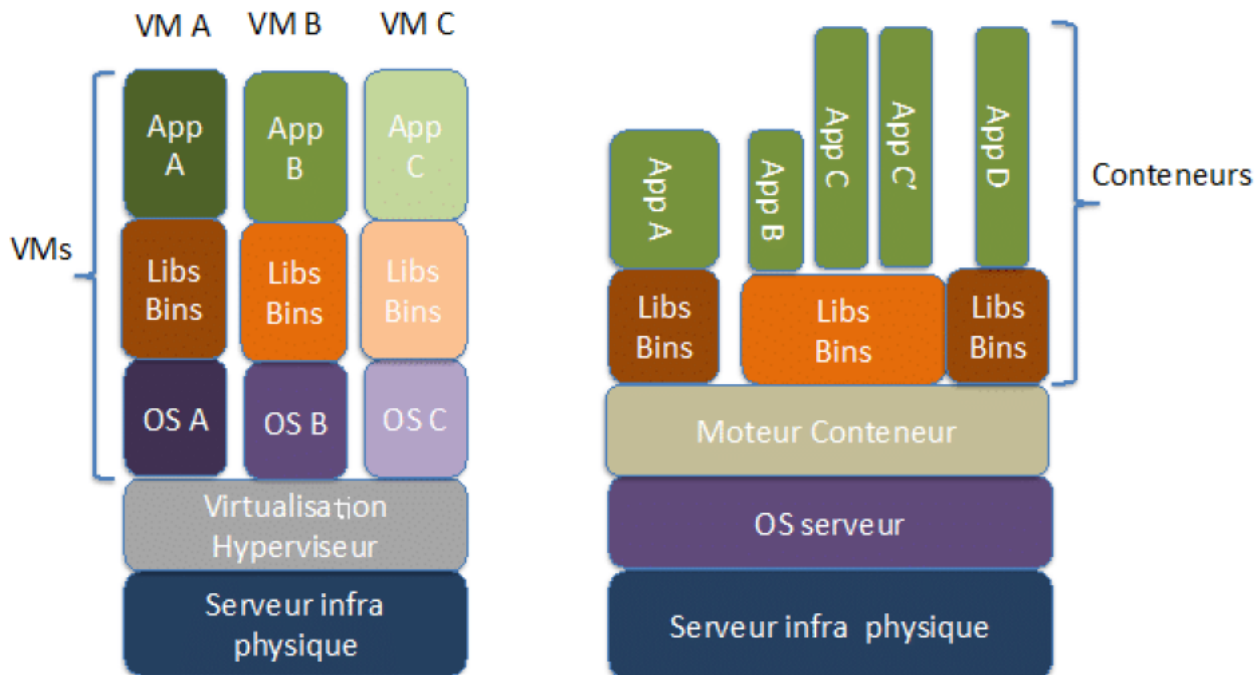
docker

- Virtualisation légère
  - non réservée
- Volatile, permet la scalabilité on-demand
- Permet à une équipe de s'abstraire de l'OS
- Au cœur de la révolution DevOps



# Hyperviseur vs Conteneur

- D'un point de vue applicatif, c'est la même architecture logicielle
- Conteneur transportable : création d'image distribuable



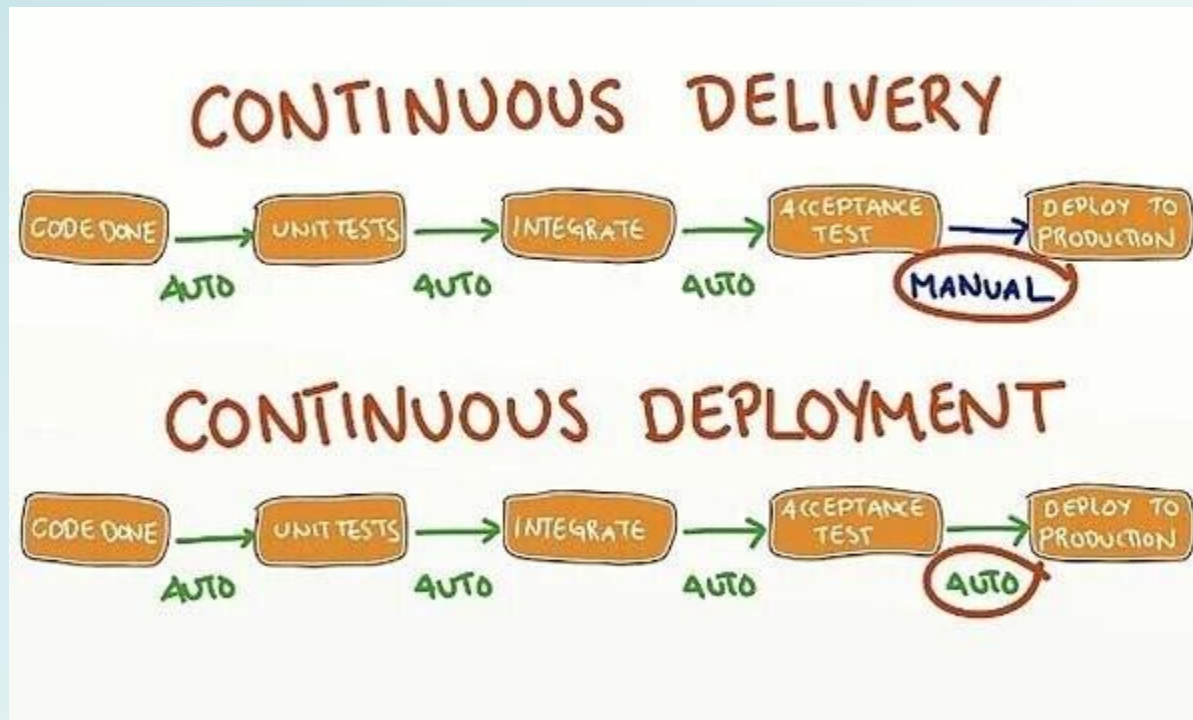
ou les 2 !



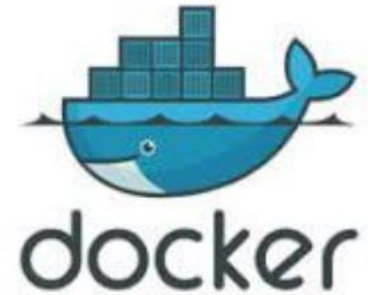


# CI/CD

Un maillon de plus dans la boucle CI



# Docker



- Platform as a Service (PaaS) v2
  - PaaS v1 (Heroku)  
*simple* espace de déploiement cloud de site web
- Immutabilité
  - On ne modifie pas
  - On redéploie une nouvelle image
  - Une philosophie de déploiement
- Reproductibilité
  - *ça marche sur ma machine (sic)*
  - Ça marchera chez vous !



# Hello Word

```
$ 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/>

```
$ docker images hello-world
```

REPOSITORY	TAG	IMAGE ID	SIZE
hello-world	latest	feb5d9fea6a5	13256

# Statefull/StateLess

- Stateless / StateFull
  - State Full = se rallumer dans l'état précédent (data management)
  - State Less = toujours les mêmes actions réalisées de manière indépendantes de l'historique (traitement requête http), aucun état nécessaire
- 1 conteneur = 1 processus
  - LAMP = 3 conteneurs (Apache, MySQL, PHP)
  - docker-compose pour assembler tous les composants (*stack*) via un fichier Manifest

# SETUP

- Un fichier 'batch' de configuration (DockerFile)

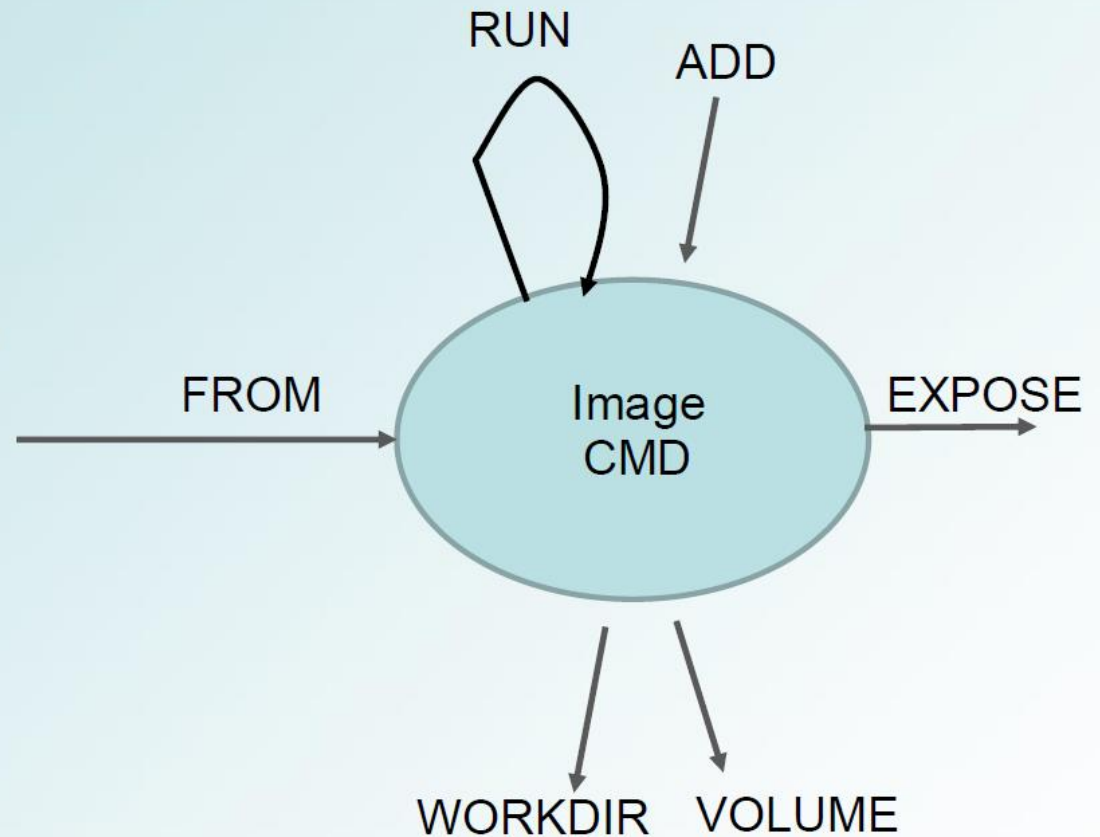
```
# syntax=docker/dockerfile:1
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run"]
```

- Un fichier 'd'assemblage' (docker-compose.yml)
- Un Build

```
$ docker-compose up
```

# Dockerfile

- FROM source
- RUN exécute
- ADD ajoute
- WORKDIR home
- EXPOSE port
- VOLUME disque
- CMD run initial



```
# syntax=docker/dockerfile:1
FROM python:3.7-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
COPY . .
CMD ["flask", "run"]
```

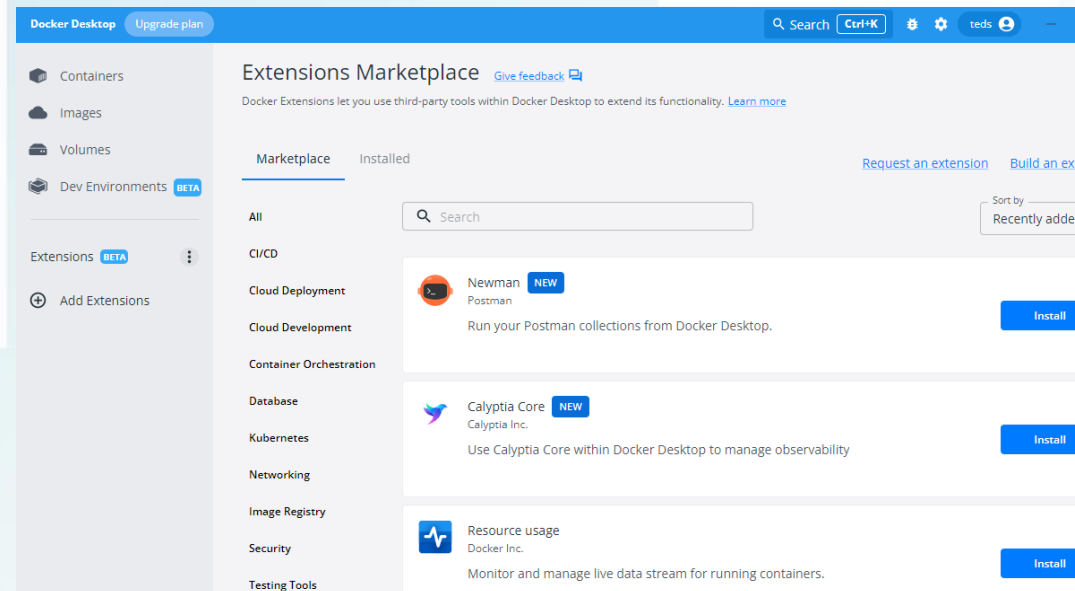
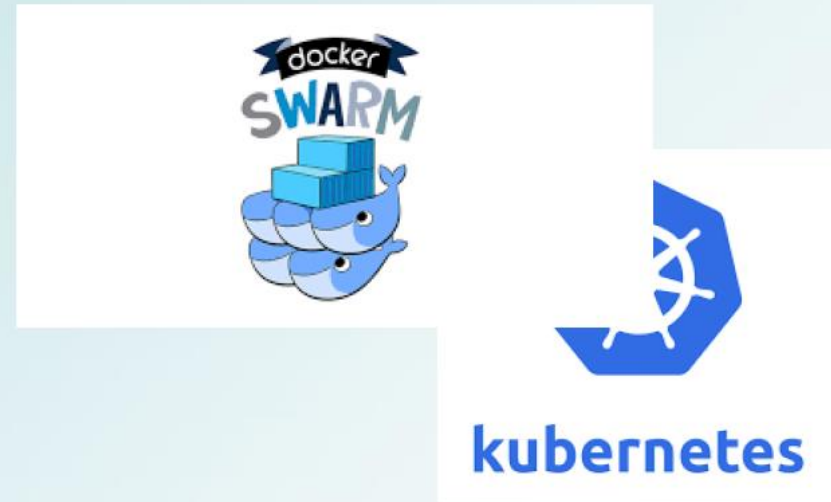
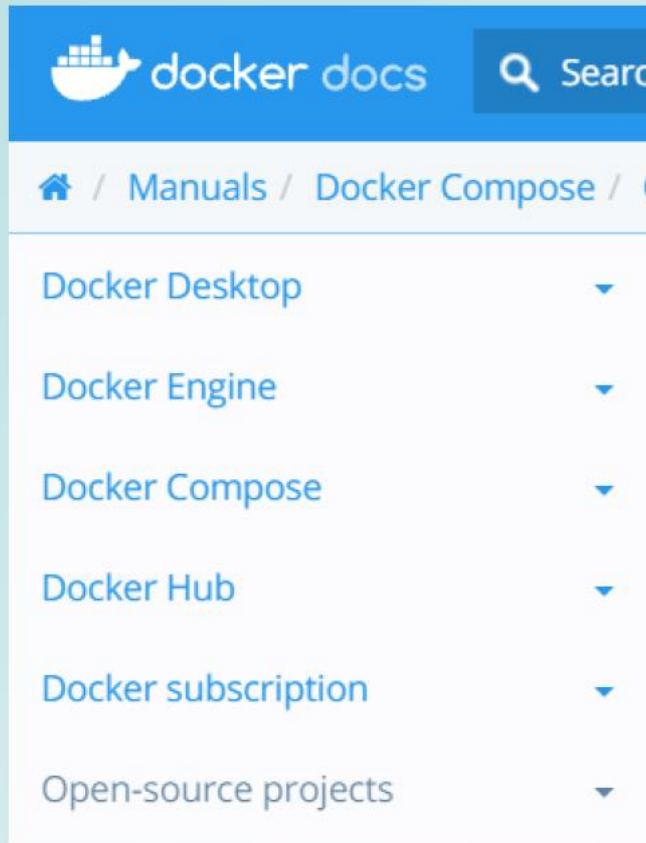


# docker-compose.yml

- Définir des services
- Les lier entre eux
  - origine,
  - dépendance,
  - variables d'environnement,
  - statut

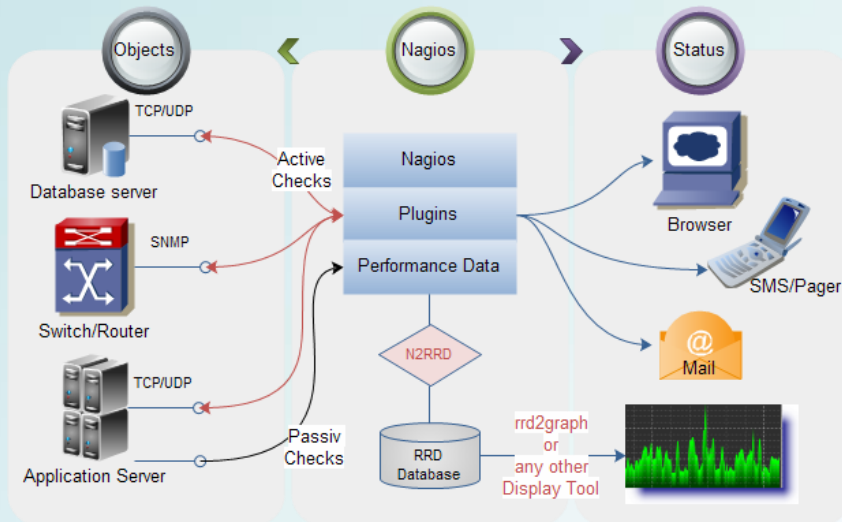
```
1 version: '3'
2 services:
3   mysql:
4     image: mysql:5.7
5     volumes:
6       - db_data:/var/lib/mysql
7     restart: always
8     environment:
9       MYSQL_ROOT_PASSWORD: monPassword
10      MYSQL_DATABASE: ghost
11      MYSQL_USER: ghostuser
12      MYSQL_PASSWORD: ocrpassword
13
14   ghost:
15     depends_on:
16       - mysql
17     image: mon_image_docker
18     ports:
19       - "8080:80"
20     restart: always
21     environment:
22       NODE_ENV: production
23
24 volumes:
25   db_data: {}
```

# Une offre intégrée



# Bouclons la boucle

- KUBERNETES
  - Supervise le déploiement, la montée en charge et mise en œuvre d'images Docker sur un cluster de machines ou de serveurs.
- NAGIOS

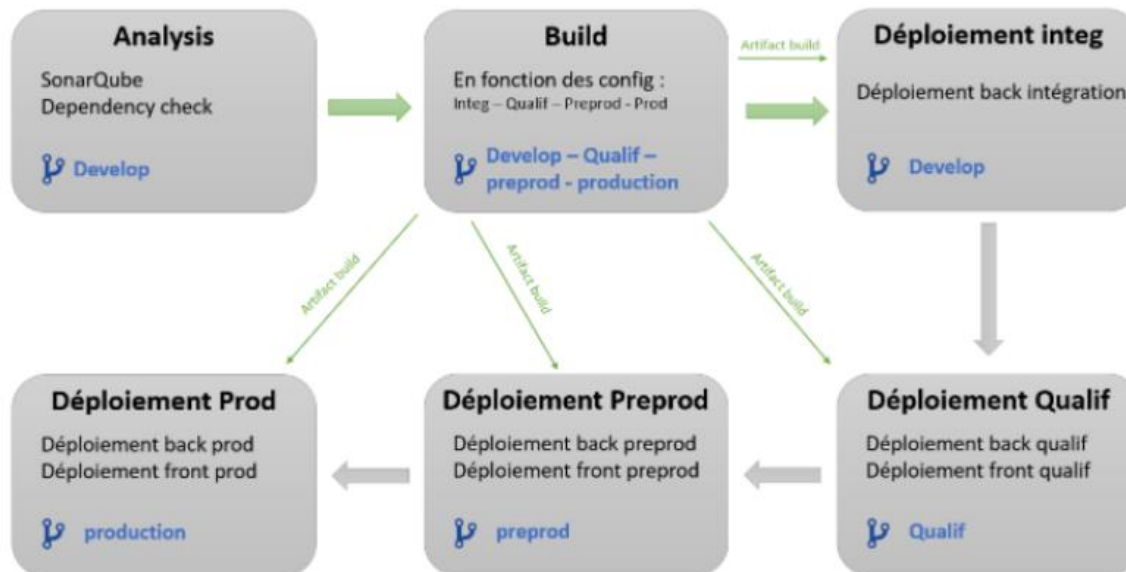


# A suivre : SOPRA (le DevOps en entreprise)

## GitLab CI/CD

Build et déploiement

— Permet de chaîner le build et le déploiement des sources







Accueil

## Consultation des MCC's

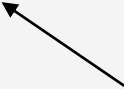
de 1 - 1 sur 1

Code de l'UE

HAI501I

Nom de l'UE

Parcours

UE	Session 1	Session 2		
<b>HAI501I</b> Ateliers de génie logiciel <b>Clementine Nebut</b> <i>Licence</i> Semestre impair ECTS : <b>5</b>	Contôle continu:			
	1	TP Noté	25%	1 h 30
	1	Oral	25%	Non applicable
	1	Écrit devoir maison	25%	Non applicable
	1	Q.C.M	25%	1 h 00
	Contrôle terminal: Contrôle TP / Terrain: <b>Le contrôle continu intégral est utilisé pour cette UE</b>			
Controle Continu: Controle Terminal: Controle TP / Terrain: Informations recalcul : sur les 4 notes de CC, on enlève la moins bonne et la meilleure et on fait la moyenne des deux restantes.				
				
QCM le 29/11				



QCM le 29/11

# Oral1 + Oral 2

- EN BINOME
  - Inscrivez vous (bientôt) sur Moodle
  - Seules les personnes inscrites seront acceptées à l'oral
- Oral 1
  - Du 6 au 10 décembre, durant le créneau de TP
  - **Nous choisissons** un TP parmi TP2 à TP8
  - Vous nous l'exposez en 5 MINUTES
  - On vous pose quelques questions
- Oral 2
  - Du 13 au 16 décembre, durant le créneau de TP
  - **Vous présentez environ 5 slides à l'oral** (parmi ceux que vous avez rendu sur Moodle) **sur** un sujet choisi parmi la liste fournie
  - Vous nous l'expliquez en 5 minutes
  - On vous pose quelques questions