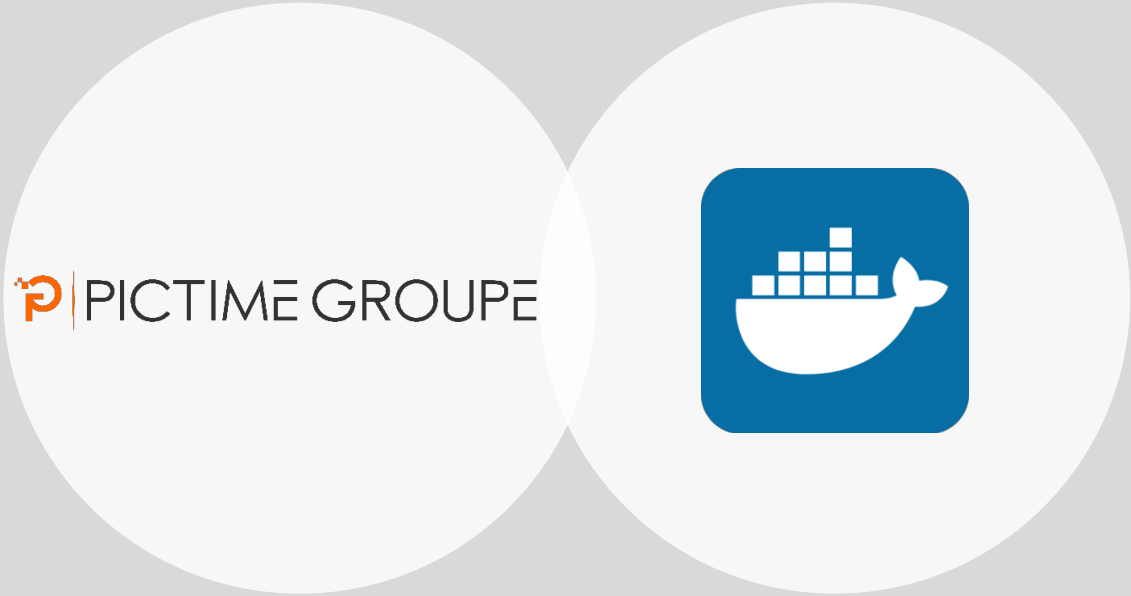


Docker

La base de la base

2 Octobre 2018

1. Pourquoi faire ?

Ça c'était avant

Avant, on n'était jamais certain qu'une application fonctionnerait bien d'une plateforme à une autre. Il fallait gérer les dépendances, les versions des dépendances, les frameworks, leurs versions, etc. **=> Compatibilité entre plateformes**

Avant, il fallait trouver la procédure d'installation correspondant à chaque plateforme à déployer, chercher la configuration qui convient, etc. (sans parler des droits d'admin)

Accessoirement il fallait jouer avec les process de sa machine (personne ne souhaite avoir un Elasticsearch "up" en permanence …) **=> Complexité de chaque plateforme**

Avant pour isoler les applications ou les environnements, on utilisait des VMs, avec la nécessité de charger un second OS et ce que cela implique en charge mémoire et processeur.

=> Cout de l'isolation

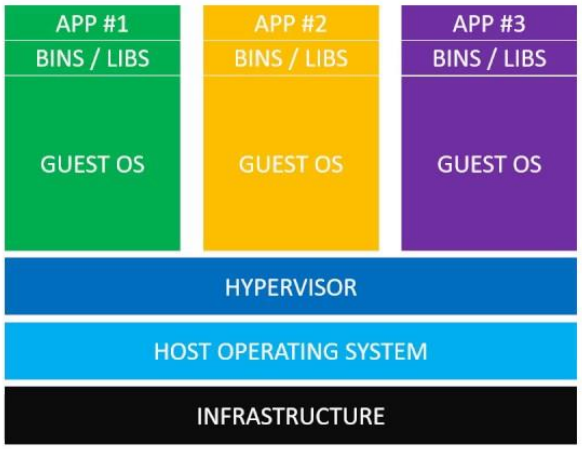
Avec Docker

On économise un OS ! (les conteneurs s'exécutent sur le noyau de la machine hôte)

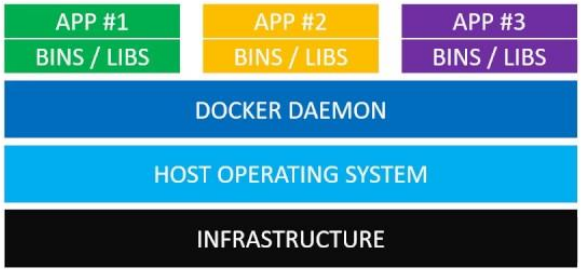
Les applications sont isolées : par défaut aucune communication n'est possible entre elles ou avec la machine hôte.

Instancier, allumer, éteindre ou supprimer un conteneur se fait le plus souvent en une seule ligne de commande.

L'image docker, une fois créée, contient tout ce qu'il faut pour l'application : dépendances, framework, configuration, etc.



Virtual Machines

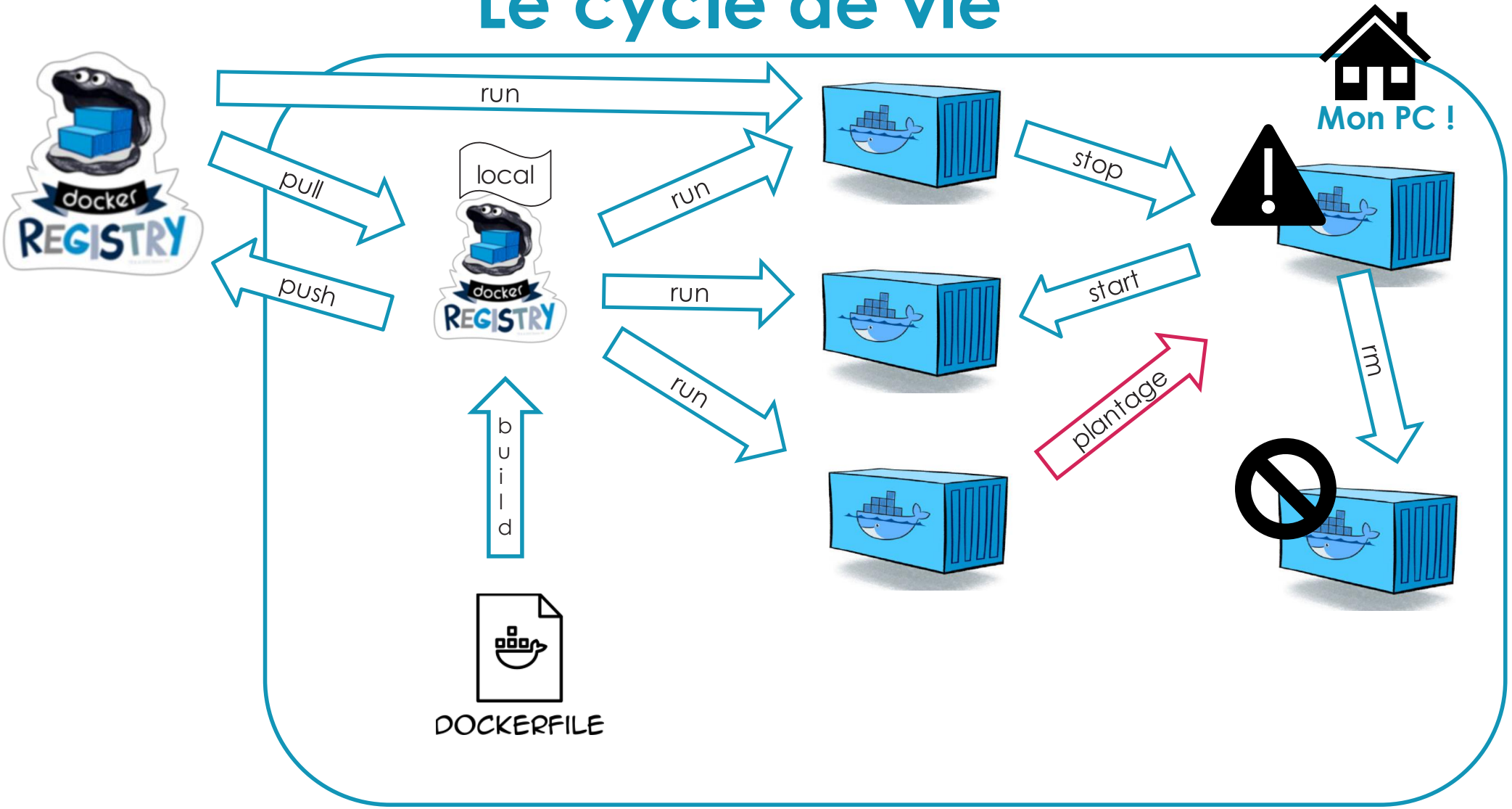


Docker Containers

2. Et donc, comment ça marche ?

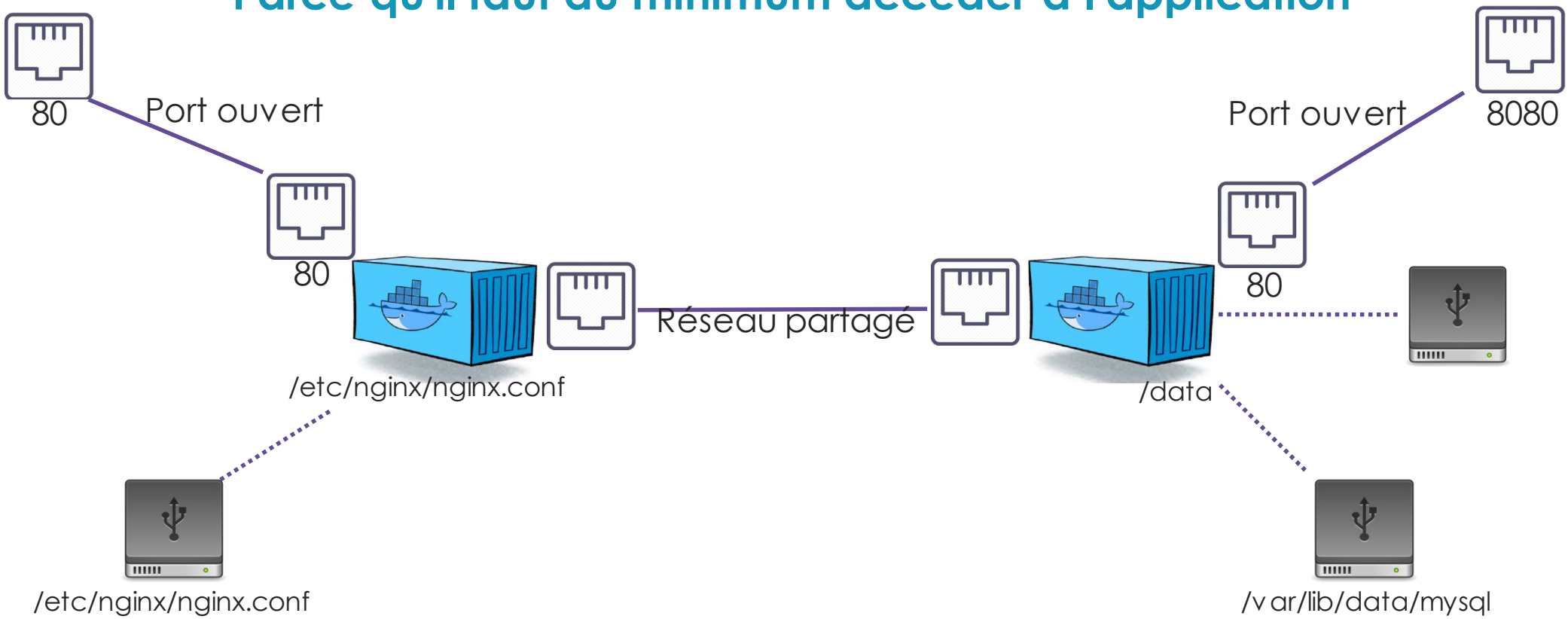
Image ? Conteneur ?

Le cycle de vie



"Briser" l'isolation

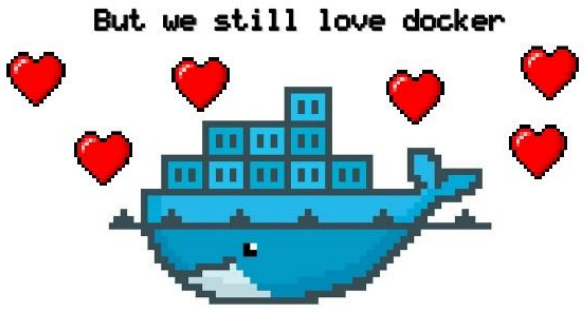
Parce qu'il faut au minimum accéder à l'application



Quelques contraintes

Linux ! (docker "windows" utilise une VM linux)

Un seul process maître par conteneur. (en théorie)



3. Image et registry

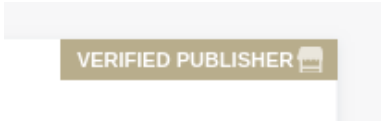
Le docker hub

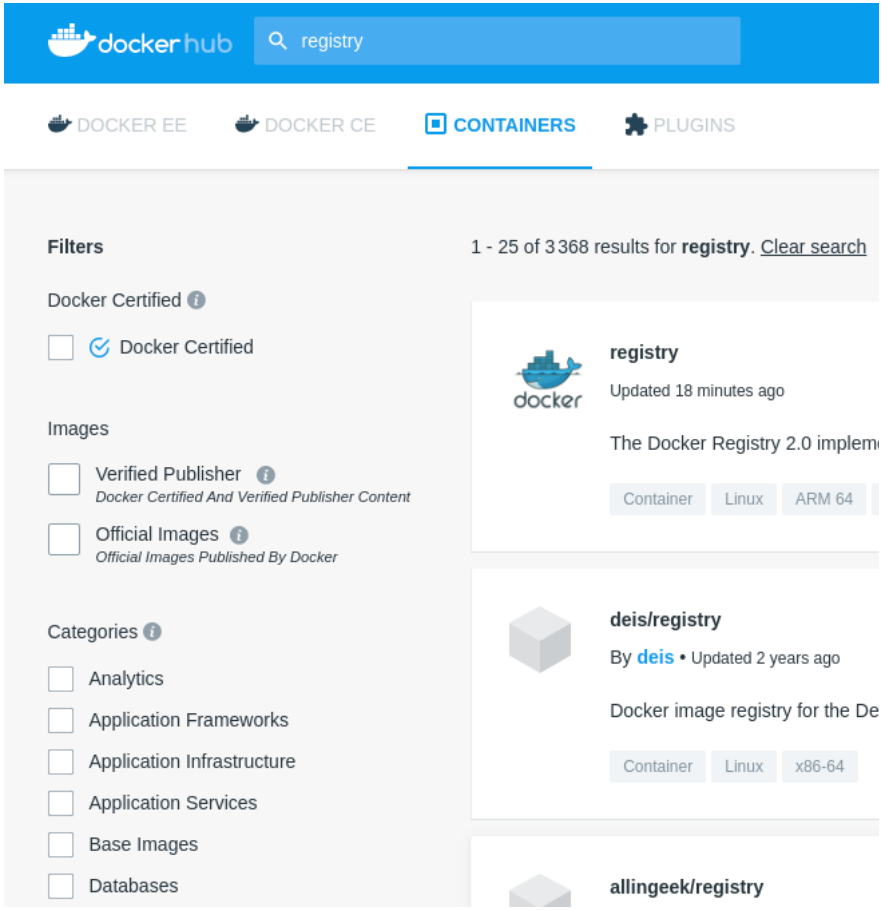
Le registry public

https://hub.docker.com/

3 types d'images :

- "officielles docker"
- "éditeurs certifiés"
- autres





L'image : Comment la nommer ?

[Adresse du registry]/[espace de nom]/**image**:[tag]

Ces 3 images ci sont peut-être différentes :

- mon-registry-docker-prive.chez-moi.com/database/mongodb:4.1.3
- docker-registry.pictime-groupe-integ.com/database/mongodb:4.1.3
- docker-registry.pictime-groupe-integ.com/incubator/mongodb:4.1.3

Adresse du registry : par défaut celle du docker-hub
Espace de nom : par défaut "library"
Tag : par défaut "latest"

registry.hub.docker.com/library/nginx:lastest == nginx

3. Et donc, comment ça marche ?

(pour de vrai)

Un exemple simple

```
$ ### Création d'un fichier index.htm
$ mkdir /tmp/nginx-html/ && echo '<html><p>Ça marche !<p></html>' > /tmp/nginx-html/index.htm
$
$ ### lancement d'un container nginx
$ docker run -d --name nginx \
    -v /tmp/nginx-html:/usr/share/nginx/html:ro \
    -p 1234:80 \
nginx
71ae539a82a44507673f9cdfe18fc92704963a16dc14e30145e9429bfdb18225
$
$ ### test
$ curl localhost:1234
<html><p>Ça marche !<p></html>
$
$ ### nettoyage
$ docker stop nginx && docker rm nginx
$ rm -rf /tmp/nginx-html/
```

Les commandes de base

```
$ docker ps
CONTAINER ID        IMAGE         COMMAND                  CREATED             STATUS                  PORTS                   NAMES
796c28c281a9        nginx         "nginx -g 'daemon ..."   3 seconds ago       Up 2 seconds            0.0.0.0:1234->80/tcp    nginx
$
$ # docker logs [CONTAINER ID|NAMES]
$ docker logs nginx
… logs du conteneur …
$ # docker stop [CONTAINER ID|NAMES]
$ docker stop nginx
nginx
$
$ docker ps --all
CONTAINER ID        IMAGE          COMMAND                  CREATED             STATUS                     PORTS                   NAMES
796c28c281a9        nginx          "nginx -g 'daemon ..."   12 seconds ago      Exited (0) 7 seconds ago                           nginx
$
$ # docker rm [CONTAINER ID|NAMES]
$ docker rm 796c28c281a9
796c28c281a9
$
$ # les plus importantes !!!
$ docker --help
… liste des commandes …
$ # docker [command] --help
… aide de la commande …
```

La commande "run"

Les options les plus courantes

--name : Permet de nommer le conteneur. S'il n'y en a pas, un nom aléatoire est généré.

-d : "detached"

-P : "Publish" expose les ports réseaux déclarés par le conteneur (dans son image). Chacun vers un port aléatoire de l'hôte.

-p XX : expose le port XX du conteneur vers un port aléatoire de l'hôte

-p YY:XX : expose le port XX du conteneur vers le port YY de l'hôte

--volume | -v : branche un répertoire de la machine hôte vers un répertoire du conteneur.

-v /rep/de/hote:/rep/de/conteneur

-e : permet de renseigner une variable d'environnement à l'intérieur du conteneur.

-e "MA_VARIABLE=valeur"