Recap J1 et J2: dockerfiles

Images et Containers

Une image est un fichier executable composé de

- un OS
- configurations de l'environnement
- installations de librairies
- execution de commandes

Un **container** est une **instance** de l'**image**, une execution particulière de l'image.

Dockerfile

Pour une application donnée, on crée un ou plusieurs fichier `Dockerfile` qui spécifie(nt) les étapes de création de l'image.

La convention est de mettre le `Dockerfile` à la racine du projet.

Le Dockerfile est composé des éléments suivants:

- **FROM**: chargement de l'image de base le plus souvent constituée d'un OS ou d'un OS combiné à un package de base (python, Nginx, etc) ou encore d'une image propriétaire
- Installations de packages et librairies avec **RUN** suivi du manager de package de l'OS
- **COPY** des fichiers locaux de développement
- Créations des volumes soient liés au host (Bind Mounts), soit internes a Docker
- Gestions des secrets et variables d'environnements (ENV)
- Lancement de la commande principale qui est exécutée au démarrage du container : CMD ou ENTRYPOINT

et quelques autres commandes pour affiner la configuration

Gestion des containers

On a vu comment

- Faire tourner un container : `docker run <image>`
- lister les containers actifs ou non : `docker ps` et `docker ps -a`
- Accéder à un container : `docker exec -it` ou `docker attach`
- stopper I supprimer un container: `docker stop | rm`
- supprimer une image: `docker stop | rmi`
- supprimer toutes les images et containers 'dangling' (en l'air): `docker system prune -f`

Ainsi que la suite de flags qu'ils faut prendre en compte pour faire tourner un container

- `-p` : mapping des ports host:container
- `-e`: variables d'environnement et `--env-file`: fichier secret
- `-d` : mode détaché
- `-it` : terminal interactif
- `-v` : pour volume: lier un repertoire local avec un repertoire dans le container
- `--name` : nommer le container

et quelques autres.

Au final, une bonne partie de la complexité de la config du container est reléguée au niveau de la ligne de commande avec l'utilisation de nombreux flags

---

Docker hub

Enfin on a vu comment publier une image sur Docker hub