

Séance 1

jeudi 10 novembre 2022 09:18

Commandes utilisées :

docker pull docker/whalesay : récupération d'image d'un conteneur "whalesay"

docker run docker/whalesay cowsay Hello Juliette ! : démarre le conteneur whalesay et appelle le process "cowsay" avec comme paramètre "Hello Juliette !"

docker container ls : liste tous les conteneurs qui tournent

docker container ls -a : liste tous les conteneurs

--rm : supprime le conteneur dès qu'il s'éteint (bonne pratique)

docker run --rm -it docker/whalesay sh : Ouvrir un terminal sur le conteneur (et le ferme à la fin)

docker run busybox sh -c "while true; do \$(echo date);sleep 1;done" : affiche la date toutes les secondes

docker logs determined_banach : récupérer les logs du conteneur nommé determined_banach

docker rm -f determined_banach : supprimer un conteneur (même si il tourne grâce au -f)

docker container prune : supprimer tous les conteneurs éteints

NB : On utilise ici des conteneurs à partir d'images disponibles en ligne

Run une image => crée un conteneur

Créer une image à partir d'un fichier Dockerfile :

docker build --tag first-test . (dans le bon répertoire)

On peut ensuite run notre image : **docker run first-test**

En l'occurrence, le fichier Dockerfile contenait un echo

L'image second-test fait le petit train

Un peu d'explication sur le contenu du fichier Dockerfile :

FROM ... : démarre ton travail depuis ...

ENV : crée une variable d'environnement

RUN : création de l'image : c'est sa préparation

CMD : config pour dire à l'image quoi faire au démarrage