1 - être capable de créer un conteneur et de gérer des images (container) docker (pull, start, stop, rm, rmi)

La première étape est d’installer docker depuis docker desktop

Une image contenant texte

Description générée automatiquement

Ensuite créer le premier conteneur et le lancer grâce à la commande docker run -d -p 1000:1000 --name test -e MYSQL\_RANDOM\_ROOT\_PASSWORD=yes mysql:latest

Argument -d pour lancé le container en arrière-plan

Argument -p pour donner le port d’écoute

Argument --name pour donner au container, le nom désiré.

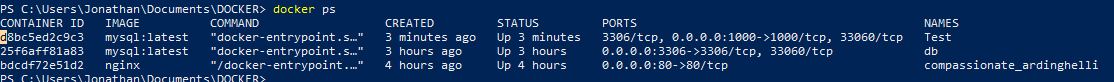
Argument -e pour donner la distribution donnée au container.

Le type de distribution:latest pour la version de la distrib

Dans l’exemple : Mon container s’appellera test.



On voit que le container a été créer avec sous ma ligne de commande sont ID unique.

Suite à la création, il est possible de vérifier que le container est lancé

Dans cette capture nous avons :

* La commande : docker ps qui permet de lister tous les containers de lancé
* Le Container ID qui est l’indentité de celui-ci, il est unique
* L’image installé sur le container : Ici Mysql en dernière version ou NGINX
* Depuis combien de temps la VM est créer, et si elle est lancée ou non et depuis combien de temps.
* Le port d’ecoute lié
* Et le nom du container.

Il est ensuite possible d’agir sur ses container la grâce à différente commande :

* Stopper et redémarrer le container : docker Start / Stop Nom\_du\_Container

****

À la suite de l’arrêt, le container n’est plus présent dans les containers lancés en arrière-plan

Pour relancé le Container : docker start Nom\_du\_container

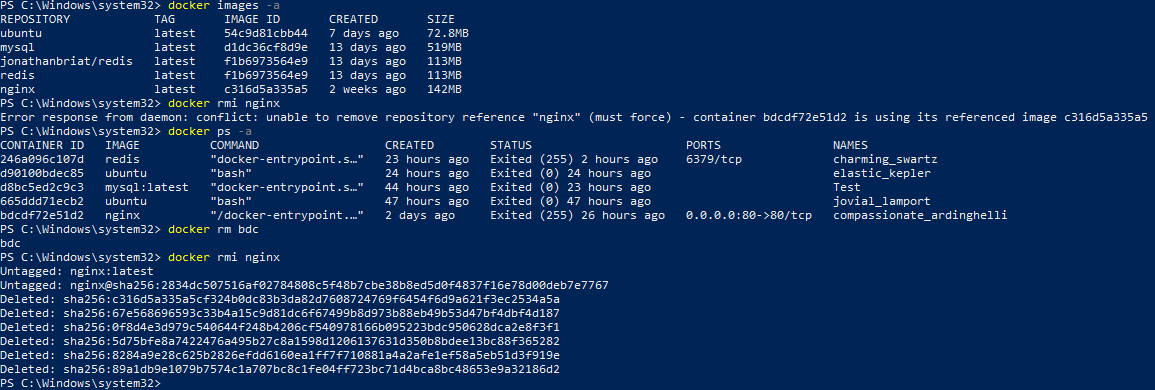
Pour la suppression complète d’un container : docker rm CONTAINER\_ID

Une image contenant texte

Description générée automatiquement

On voit ici que le container mysql :latest 25f n’est plus présent.

Pour supprimé une image : docker rmi NOM\_DE\_I’IMAGE



Impossible de supprimer une image si elle est en cours d’utilisation sur un container

Suppression du container en question et suppression de l’image.

Accéder au Shell d’un conteneur

Une image contenant texte

Description générée automatiquement

Le container doit être lancer

Il est important de mettre que l’on veut accéder au Bash après la commande docker exec -ti NOM\_DU\_CONTAINER /accès au bash

Pour avec le monitoring des container lancés

Avec docker stats, nous avons comme le gestionnaire de tache windows avec l’utilisation du CPU, la mémoire, le nom de la machine et son ID

Une image contenant table

Description générée automatiquement