

Conteneurisation

Docker est le logiciel de référence.

Il y a un seul noyau (Windows ou Linux) sur lequel on peut fixer des containers.

Les images Docker sont transformées en conteneur. On peut orchestrer des conteneurs (les superposer).

Docker fonctionne sur le principe de workflow :

- Build : création des images
- Ship : stockage des images
- Run : déploiement des images, liaison des conteneurs

TP 1

II – Un docker tout simplement

Q1) Les différents modes possèdent des points positifs suivants dans l'ordre de listage :

- Facile pour les tâches simples
- Permet d'économiser de l'énergie et de la puissance de calcul
- Permet de disposer de toutes les fonctionnalités du container

Afin d'installer docker, nous utilisons les commandes suivantes :

- **sudo apt update**
- **sudo apt upgrade**
- **sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg2**
- **curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -**
- **sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian \$(lsb_release -cs) stable"**
- **sudo apt update**
- **sudo apt install docker-ce**

III – Option 1 : Un docker, une tâche

Q2) On exécute la commande : **sudo docker run debian date**

En sortie, la machine pull debian de la librairie et renvoie : **Fri Apr 1 08 :24 :23 UTC 2022**

Q3) On exécute la commande : **sudo docker run debian top**

Nous obtenons une erreur.

Q4) L'erreur nous indique que le conteneur s'exécute de manière ponctuelle, avec une seule commande.

Q5) Nous exécutons la commande : **sudo docker container ls -all**

Nous obtenons le résultat suivant

```
etudiant@debian-stretch:~$ sudo docker container ls -all
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
8ef8b05bbdc3       debian             "top"              6 minutes ago
Created            trusting_darwin
```

Cela nous indique qu'un seul conteneur a été créé.

IV- Option 2 : Une tâche de fond

Q6) Nous exécutons la commande : **sudo docker container run --detach --name monCtDb -e MYSQL_ROOT_PASSWORD=vitrygtr mysql:latest**

Le résultat obtenu est le suivant

```
etudiant@debian-stretch:~$ sudo docker container run --detach --name monCtDb -e
MYSQL_ROOT_PASSWORD=vitrygtr mysql:latest
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
f003217c5aae: Pull complete
65d94f01a09f: Pull complete
43d78aaa6078: Pull complete
a0f91ffbfd69: Pull complete
59ee9e07e12f: Pull complete
04d82978082c: Pull complete
70f46ebb971a: Pull complete
db6ea71d471d: Pull complete
c2920c795b25: Pull complete
26c3bdf75ff5: Pull complete
9ec1f1f78b0e: Pull complete
4607fa685ac6: Pull complete
Digest: sha256:1c75ba7716c6f73fc106dacedfdcf13f934ea8c161c8b3b3e4618bcd5fbcf19
Status: Downloaded newer image for mysql:latest
3ced72e38774c4160d51431e7ccf4803ed633e98ba9cef455ce422069f5d8896
```

Nous en concluons que la machine a bien été créée.

Q7) Les arguments ont les utilités suivantes :

- **--detach** : permet d'exécuter en tâche de fond
- **--name monCtDb** : renomme le conteneur en « monCtDb »
- **-e** : permet de modifier les variables d'environnement
- **MYSQL_ROOT_PASSWORD=vitrygtr** : permet de modifier le mdp en « vitrygtr »
- **mysql:latest** : permet d'utiliser la dernière version de mysql

Q8) Les commandes ont les utilités suivantes :

- **docker container logs** : permet d'afficher les logs d'un conteneur
- **docker container top** : rend compte des processus en cours d'un conteneur
- **docker container exec** : exécute une commande sur le conteneur actuel

IV- Option 3 : Le mode interactif

Q10) Voici les arguments et leur utilité :

- **--interactive** : Laisse le stdin ouvert
- **--tty** : Alloue un pseudo-tty
- **--rm** : Automatiquement enlève le conteneur quand il est fermé

Q11) Cette fois-ci la version est ubuntu. Cela veut dire que le noyau utilisé est le même.

V- DockerFile

Q12) Nous devons d'abord build l'image. Le fichier requirements.txt ne contient que « flask ». Nous utilisons ensuite la commande : **sudo docker build -t /home/etudiant/Hello_Flask/**

Q13) Nous avons à entrer la commande : **sudo docker run 5000 :8080 -d -p Hello_Flask**

VI- Docker Hub

N'ayant pas pu venir au TP2 pour raisons médicales, je n'ai pas pu finir ce TP. Cependant, afin de rattraper cela, je vais restituer ici ma compréhension de ce même TP.