

# TP: Docker

Nous avons maintenant fait le tour des VMs, concentrons nous sur les conteneurs. Vous aurez besoin de Docker et avez soit la possibilité de l'installer sur votre OS (Linux, OSX, dans certains cas Windows) voir <https://docs.docker.com/get-docker/> ou bien vous pouvez utiliser la VM que vous avez récupérée lors de l'exercice 2, elle contient Docker. Voici une liste des commandes de base Docker avec une brève explication de chaque commande, vous êtes encouragés à les essayer :

- **docker version** : Affiche la version de Docker installée sur votre système, ainsi que les informations sur le client et le serveur Docker.
- **docker info** : Affiche des informations détaillées sur la configuration de Docker, telles que le nombre de conteneurs en cours d'exécution, d'images disponibles, etc.
- **docker pull [nom\_image]** : Télécharge une image Docker depuis Docker Hub ou un autre registre d'images. Par exemple, `docker pull nginx` téléchargera l'image Nginx.
- **docker images** : Affiche la liste des images Docker téléchargées localement sur votre système.
- **docker run [options] [nom\_image]** : Crée et lance un conteneur à partir de l'image spécifiée. Par exemple, `docker run -d -p 8080:80 nginx` lancera un conteneur Nginx en arrière-plan sur le port 8080 de l'hôte.
- **docker ps** : Affiche la liste des conteneurs en cours d'exécution sur votre système.
- **docker ps -a** : Affiche la liste de tous les conteneurs, y compris ceux qui ne sont pas en cours d'exécution.
- **docker stop [ID\_conteneur ou nom\_conteneur]** : Arrête un conteneur en cours d'exécution. Vous pouvez spécifier soit l'ID du conteneur, soit son nom.
- **docker rm [ID\_conteneur ou nom\_conteneur]** : Supprime un conteneur arrêté de votre système. Vous pouvez spécifier soit l'ID du conteneur, soit son nom.
- **docker rmi [nom\_image]** : Supprime une image Docker de votre système. Vous devez d'abord supprimer tous les conteneurs associés à cette image.
- **docker exec -it [ID\_conteneur ou nom\_conteneur] [commande]** : Exécute une commande à l'intérieur d'un conteneur en cours d'exécution. L'option `-it` permet une interaction en mode tty.

## Exercice 1: Mon premier conteneur

- 1) Lancez un conteneur debian:  
**docker run -ti --rm debian**
  - a) quel utilisateur êtes-vous dans le conteneur (commande `whoami`) ?
  - b) Que contient le système de fichier / ?
  - c) Est-il différent de celui de votre machine ?
- 2) Maintenant lancez:  
**docker run -ti --rm rancher/cowsay cowsay "Hello Docker"**
  - a) Expliquez les différents arguments
  - b) Que représente **rancher/cowsay** (voir <https://hub.docker.com/r/rancher/cowsay>)
  - c) Que fait la commande **cowsay "Salut Linux"**
  - d) Expliquez où se situe le binaire `cowsay` ?

- e) Quelles sont les différentes étapes lors du lancement du conteneur (en considérant que le conteneur n'est pas présent) ?
- f) Pourquoi la commande est plus rapide lors du second lancement ?

## Exercice 2: Mon premier serveur Web

- 1) Lancez la commande suivante:

**docker run -ti -p8080:80 nginx**

- a) Expliquez l'effet des différents arguments
  - b) Que voyez-vous en vous connectant sur <http://localhost:8080> (en ayant le conteneur lancé)
    - i) Note si vous êtes dans la VM vous pouvez utiliser la commande **curl** <http://localhost:8080> pour voir le contenu de la page
  - c) Trouvez le nom de votre conteneur en lançant:
 

**docker ps**
  - d) Connectez vous dans votre conteneur avec:
 

**docker exec -ti [NOM CONTENEUR] bash**
  - e) Quelle est le chemin de la racine du serveur WEB (regardez la configuration de NGINX)
 

**cat /etc/nginx/conf.d/default.conf**
  - f) Si vous vous rendez dans ce dossier et exécutez la commande suivante:
 

**echo "<h1>PAWNED</h1>" > index.html**

    - i) Le contenu de <http://localhost:8080> a-t-il changé ?
    - ii) Vous pouvez quitter le shell dans le conteneur avec CTRL+D ou exit
- 2) Nous voulons maintenant exposer notre propre site web, pour ce faire nous voulons remplacer le dossier racine du serveur web en insérant notre dossier à la place. Pour cela nous allons utiliser un volume (option -v) la syntaxe est:
 

**docker run -ti -p8080:80 -v SOURCE:DESTINATION nginx**

 Cela va insérer le dossier SOURCE à DESTINATION dans le conteneur
    - a) Lancez le conteneur en remplaçant le site par celui dans ./ex2/html du dépôt du TP
    - b) Quel est le mot de passe de la page secrète ?

## Exercice 3: Redis et Script

- 1) lancer un serveur Redis ([https://hub.docker.com/\\_/redis](https://hub.docker.com/_/redis)) en exposant le port 6379 du conteneur sur le port 6379 local
  - a) Quelle commande utilisez-vous ?
  - b) Qu'est ce que redis ?
  - c) Connectez-vous au redis avec la commande
 

**nc localhost 6379**
  - d) Effectuez des commandes simples:
 

SET lol 123

GET lol
- 2) Réalisez un script python qui compte le nombre d'occurrence dans guerre et paix Tome 1 (voir ./ex6/gp.txt). Pour ce faire, nous utiliserons le Redis pour compter avec la commande HINCR (<https://redis.io/commands/hincrby/>). Pour terminer on affichera le compte final (voir <https://redis.io/commands/hkeys/>)