

Travaux Pratiques

TP3 Docker - Dockerfile

2022/2023

Notions Abordées :

- Installation de Docker
- Quelques commandes essentielles
- Dockerfile et Construction d'image Docker
- Exécution d'une image Docker et conteneur

1. Installation de Docker

Linux:

Vous devez avoir installé Docker. Dans l'idéal vous travaillerez sous Linux, sur un hôte physique, sans passer par une machine virtuelle. Pour installer Docker, Vous suivrez les étapes d'installation de la documentation officielle propre à votre distribution.

<https://docs.docker.com/get-docker/>

Pour connaître votre distribution, vous pouvez exécuter la commande suivante :

```
cat /etc/*release*
```

Par exemple pour Ubuntu 22.04 :

On commence par la désinstallation des versions antécédentes de Docker (s'il y en a une) :

```
sudo apt-get remove docker docker-engine docker.io containerd runc
```

Pour une désinstallation complète de docker, suivez les étapes suivantes :

Etape 1 : cette commande vous indiquera tous les packages relatifs à Docker, que vous avez installés.

```
dpkg -l | grep -i docker
```

Etape 2 : La commande ci-dessous supprime tout à part les conteneurs, les images, les volumes, et les configurations des utilisateurs.

```
sudo apt-get purge -y docker-engine docker docker.io docker-ce  
sudo apt-get autoremove -y --purge docker-engine docker docker.io docker-ce
```

Etape 3 : Exécutez cette commande pour supprimer tous les conteneurs, images, volumes et configurations des utilisateurs.

```
sudo rm -rf /var/lib/docker /etc/docker
sudo rm /etc/apparmor.d/docker
sudo groupdel docker
sudo rm -rf /var/run/docker.sock
```

Installer Docker à partir d'un script :

```
curl -fsSL https://get.docker.com -o get-docker.sh
sudo sh get-docker.sh
```

Windows

Si vous êtes sous Windows, installez de préférence Docker pour Windows.

<https://docs.docker.com/docker-for-windows/install/>

Mac

Si vous êtes sous Mac, installez de préférence Docker for Mac.

<https://docs.docker.com/docker-for-mac/install/>

A la fin, pour connaître la version de docker installée, exécuter la commande :

```
docker version
```

2. Premier exemple

Pour s'assurer que l'installation s'est bien déroulée, on va exécuter un premier exemple :

Rendez vous sur : -<https://hub.docker.com/> pour chercher une image existante sur DockerHub.
soit par exemple : whalesay.

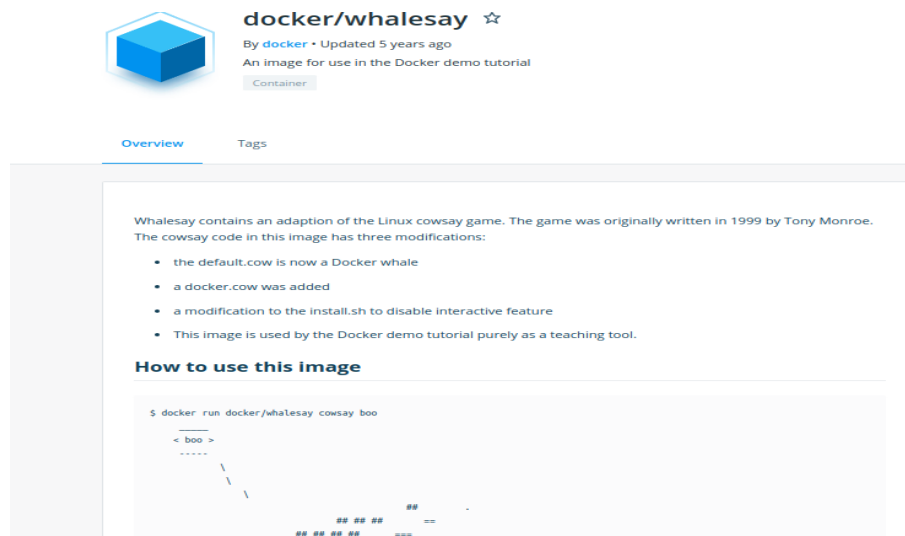


Figure 1:

Sur la ligne de commande, la commande suivante nous permettra d'exécuter et de charger l'image whalesay : `sudo docker run docker/whalesay cowsay hello`

3. Commandes Essentielles

sudo docker run imageName : Pour exécuter notre image.

sudo docker run -pport__host:port__Conteneur : Exécution du conteneur en liant le port du conteneur aux port de la machine

docker run -ti --rm imageName : Permet d'exécuter l'image docker en mode actif

sudo docker images : pour lister les images docker existantes sur votre machine.

sudo docker images -a : pour lister les images docker existantes dans le cache.

sudo docker ps -a : Pour lister tous les conteneurs (l'historique des conteneurs).

sudo docker ps : Pour lister tous les conteneurs actifs

sudo docker stop : Pour arrêter un conteneur

sudo docker start : Pour démarrer un conteneur

sudo docker pull tag_image:version : Télécharger une image à partir de Dockerhub

sudo docker build -t imageName . : Permet de construire une image nommée *imageName* à partir du fichier Dockerfile correspondant dans le dossier courant.

4. Application

- (a) Placez vous sur DockerHub
- (b) Choisissez une image (exemple : redis)
- (c) Télécharger une version de cette image : `docker pull redis`
- (d) Lister l'ensemble d'image que vous avez sur votre machine
- (e) Créer un conteneur à partir de cette image : `docker run redis`
- (f) lister les conteneur en cours d'exécution
- (g) Exécuter le conteneur en mode détaché : `docker run -d redis`
- (h) Stopper le conteneur
- (i) Démarrer le conteneur
- (j) Afficher l'historique de tous les conteneurs existants sur votre machine
- (k) Télécharger une deuxième version de redis différentes de la première version
- (l) Exécutez-la parallèlement avec la première
- (m) Qu'est-ce que vous remarquez?
- (n) Liez le port du conteneur redis à un port de la machine

5. Initiation à Dockerfile : Construire votre image Docker

Pour construire votre propre image docker, vous devez d'abord écrire un *Dockerfile*. Dockerfile est un plan "blueprint" pour la création des images docker.

Dans un fichier Dockerfile, vous devez tout d'abord spécifier quelle sera votre première image (base image). Par la suite, vous devez spécifier quelles libraires et quelles package vous devez installer

dans cette particulière "docker image".

Ceci composera une pile d'images construites les unes en dessus des autres, et le résultat sera une image docker finale.

(a) **Choisir une image de base (Docker base image) :**

Dans notre tp, on va choisir l'image de base "alpine", cette image est disponible sur docker hub.

(b) Commençons par exécuter l'image alpine en mode actif :

docker run -ti --rm alpine:latest /bin/sh

Essayons de voir si cette image contient git, vim, wget.

```
safaa@safaa-Inspiron-3542:~/dockerfileTp$ sudo docker run -ti --rm alpine:latest
[sudo] Mot de passe de safaa :
/ # git --version
/bin/sh: git: not found
/ # vim
/bin/sh: vim: not found
/ # wget
BusyBox v1.31.1 () multi-call binary.

Usage: wget [-c|--continue] [--spider] [-q|--quiet] [-O|--output-document FILE]
          [-o|--output-file FILE] [--header 'header: value'] [-Y|--proxy on/off]
          [-P DIR] [-S|--server-response] [-U|--user-agent AGENT] [-T SEC] URL...
```

(c) Maintenant, essayons de créer notre propre image basée sur alpine et on va y ajouter git, vim. Pour ce faire, on va créer un Dockerfile (un fichier texte sans aucune extension).

Notre fichier Dockerfile :

```
FROM alpine:latest
MAINTAINER BadreBOUSALEM b.bousalem@enpc.fr

#Installer git et vim
RUN apk update
RUN apk add git
RUN apk add vim
```

(d) Pour effectuer le build de ce fichier Dockerfile, exécuter l'instruction suivante :

sudo docker build -t Badre09/alpine-smarter .

- Les arguments **-t nom_tag** : permettent de donner un nom à votre image, dans ce cas safaa09/alpine-smarter.
- et le dernier argument indique le dossier qui contient le docker file, dans ce cas, le dossier courant (indiquer par .).

(e) Exécuter l'image docker en mode actif docker run -ti -rm badre09/alpine-smarter:latest

(f) Qu'est-ce que vous remarquez cette fois-ci?

- (g) Essayez d'exécuter le même fichier Dockerfile, Cette fois-ci en regroupant les instructions exécutées avec RUN.

```
FROM alpine:latest
MAINTAINER Badre BOUSALEM b.bousalem@enpc.fr

#Installer git et vim
RUN apk update && \
    apk add git && \
    apk add vim
```

- (h) Qu'est-ce que vous remarquez?

- (i) Maintenant, essayer d'afficher un message avec la commande RUN et un message avec la commande CMD. Ajoutez à votre Dockerfile, les deux instructions suivantes :

```
RUN echo "Hello RUN !"
CMD echo "Hello CMD !!"
```

Lancer le build de votre image et exécutez-la. Qu'est-ce que vous remarquez ?

- (j) Copiez le contenu de votre répertoire ou un fichier spécifique dans un répertoire de votre image avec l'instruction COPY; ce, en ajoutant à votre Dockerfile, l'instruction suivante : *COPY ./fich1 usr/fich1*.

6. Cas d'un Dockerfile pour un programme Java : (construction d'une image Docker)

Soit le programme Java suivant :

```
public class HelloWorld {
    public static void main(String[] args) {

        System.out.println(" Salut, c'est notre première image Docker!!");

    }
}
```

- (a) On veut dockeriser cette application java en créant un Dockerfile. On utilisera alpine comme OS de base.
- Est-ce que l'image alpine contient java / jdk? Sinon, installer jdk-8 ou openjdk8 sur votre image.
 - Toujours dans votre Dockerfile, Introduisez des instruction afin de pouvoir compiler et exécuter votre programme.
 - Construisez votre image avec Dockerfile
 - Exécutez votre Image
- (b) Pouvons-nous implémenter un autre dockerfile qui réalise la même tâche que le dockerfile précédent.

Indication : à partir d'une autre image de base, par exemple jdk8 ou openjdk8.

7. Envoyez votre image sur DockerHub

- Créer un compte sur le dépôt Dockerhub
- à partir de votre ligne de commande, connectez-vous à votre dépôt Dockerhub:
cat /password.txt | docker login -username foo -password-stdin
- Envoyez votre image à votre dépôt Dockerhub `docker push yourhubusername/imagename`.
note : Pour envoyer votre image docker sur votre dépôt Dockerhub elle doit respecter le format suivant : `votre-nom-dockerhub/imagename`