

Docker

Docker c'est quoi ?

Docker désigne plusieurs éléments, à savoir un projet d'une communauté Open Source, les outils issus de ce projet Open Source, l'entreprise Docker Inc. qui constitue le principal soutien de ce projet, ainsi que les outils que l'entreprise prend officiellement en charge. Le fait que ce terme soit utilisé aussi bien pour désigner les technologies que l'entreprise peut prêter à confusion.

Grâce à Docker, les conteneurs deviennent des machines virtuelles très légères et modulaires qui vous offrent une grande flexibilité pour créer, déployer, copier des conteneurs et les déplacer d'un environnement à un autre. Vos applications sont ainsi optimisées pour le cloud.

Source: www.redhat.com

1-L'installation de Docker :

Première solution : <https://docs.docker.com/get-started/get-docker/>

Deuxième solution : Suivre les commandes :

<https://www.ionos.fr/digitalguide/serveur/configuration/installer-docker-sur-debian-12/>

2- Les Images et les conteneurs :

On peut installer des images Docker depuis le [docker hub](https://hub.docker.com/).

Commandes :

docker ps permet d'afficher la liste des conteneurs en cours d'exécution .

***docker ps -a** permet d'afficher la liste des conteneurs en cours d'exécution et arrêtés .

docker images permet d'afficher toutes les images de la machine locale.

Comment arrêter un container ?

Pour arrêter un container il nous faut utiliser la commande :

```
docker stop <nom_du_conteneur>
```

Comment supprimer un container ?

Pour supprimer un container il nous faut utiliser la commande :

```
docker rm <nom_du_conteneur>
```

Comment supprimer une image ?

Pour supprimer une image il nous faut utiliser la commande :

```
docker rmi <nom de l'image>
```

Comment installer l'image de Apache ?

Pour installer l'image de Apache il faut utiliser la commande :

```
docker pull httpd
```

Comment lancer le container (en mode détaché) avec l'image apache en mappant le port 8081 du host vers le port 80 du container ?

Pour cela il nous faut utilisé la commande :

```
docker run -d -p 8081:80 --name mon-apache httpd
```

Pour test il faut depuis un navigateur web rentré : <http://IP:8081>

3.1-Création de votre image :

Créer un répertoire et se situer dedans :

```
mkdir lab0  
cd lab0
```

Créer un 'DockerFile' :

```
nano Dockerfile
```

On y glisse a l'intérieur :

```
----- DÉBUT COUCHE OS -----  
FROM debian:stable-slim  
----- FIN COUCHE OS -----  
#MÉTADONNÉES DE L'IMAGE  
MAINTAINER BTS SIO2 SISR  
#----- DÉBUT COUCHE Installation -----  
RUN apt-get update \
```

```
&& apt-get install -y vim git htop mc tree \  
&& apt-get clean \  
&& rm -rf /var/lib/apt/lists/* /tmp/* /var/tmp/*
```

Ensuite on doit construire l'image :

```
docker build -t btssio:v1.0 .
```

A noter pour voir l'image ainsi que sa taille il nous faut utiliser la commande :

```
docker images
```

```
root@debian-template:~# docker images
```

IMAGE	ID	DISK USAGE	CONTENT SIZE	EXTRA
btssio:v1.0	8c60ea0708d3	384MB	91.1MB	U

Pour lancer l'image :

```
docker run -tid --name sirs btssio:v1.0
```

Pour vérifier si l'image est 'up':

```
docker ps -a
```

```
root@debian-template:~/lab0# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e3f367f11a67	btssio:v1.0	"bash"	6 days ago	Exited (255) About an hour ago		sirs

Pour une image moins lourde (Alpine) :

```
docker build -t btssio:alpine .
```

```
docker images | grep btssio
```

```
docker run -tid --name sirs-alpine btssio:alpine
```

3.2- Orchestrer des conteneurs :

Créer le répertoire projet puis se rendre dans celui-ci :

```
mkdir projet
```

```
cd projet
```

Créer les fichiers :

```
nano DockerFile
```

```
nano requirements.txt
```

```
nano app.py
```

```
nano docker-compose.yml
```