

VIRT1

Benjamin Delacombaz

SI-T1a

16.05.2018

Docker

Configuration d'un hôte Docker et premiers containers

1. Installez une VM Debian
2. Installez Docker avec les commandes suivantes ou suivez la dock suivante
<https://docs.docker.com/install/linux/docker-ce/debian/#set-up-the-repository>:

```
# apt-get update
# apt-get install \apt-transport-https \ca-certificates \curl \gnupg2 \software-
properties-common
# curl -fsSL https://download.docker.com/linux/debian/gpg | apt-key add -
# apt-key fingerprint 0EBFCD88
# add-apt-repository \
"deb [arch=amd64] https://download.docker.com/linux/debian \
$(lsb_release -cs) \
stable"
# apt-get update
# apt-get install docker-ce
# docker run hello-world
```

3. Si cela fonctionne correctement, installez le container officiel ubuntu avec les commandes suivantes:

```
# docker pull ubuntu
# docker images
```

4. Si l'image ubuntu apparait dans la liste, l'image est correctement installée

Premier container

1. Démarrez le containers ubuntu avec la commande suivante:

```
# docker run ubuntu
```

2. Listez les containers actifs (normalement aucun), puis les actifs et inactifs avec les commande suivantes:

```
# docker ps  
# docker ps -a
```

Ils sont tous inactifs, normal car nous ne les utilisons pas (aucun service actif)

3. Redirigez l'entrée standard du container pour ouvrir un pseudo-terminal avec la commande suivante

```
# docker run -ti --name=ubuntu ubuntu /bin/bash
```

4. Controllez la version d'ubuntu avec la commande suivante:

```
# cat /etc/lsb-release
```

```
DISTRIB_ID=Ubuntu  
DISTRIB_RELEASE=18.04  
DISTRIB_CODENAME=bionic  
DISTRIB_DESCRIPTION="Ubuntu 18.04 LTS"
```

5. Ouvrez un second terminal et listez les containers actifs:

```
# docker ps
```

Il y en a un seul, celui que nous avons démarré précédemment.

6. Pour savoir le container ID, il suffit de lancer la commande:

```
# docker ps
```

Mon container id est le suivant: 4054b5ad576c

7. Vérifiez que vous voyez tous les processus de votre container en listant également dans le container avec la commande (cherchez avec la colonne RES):

```
# top
```

8. L'adresse IP de mon container est la suivante: 172.17.0.2

9. L'adresse IP de la passerelle est la suivante: 172.17.0.1

10. Arrêtez votre container en utilisant la commande

```
# docker stop ubuntu
```

11. Demarrez votre container avec la commande suivant:

```
# docker start ubuntu
```

12. Pour vous connectez à votre ubuntu, utilisez la commande suivante:

```
# docker attach ubuntu
```

13. Supprimez votre container avec la commande suivante:

```
# docker rm ubuntu
```

Gestion des containers

1. Créez un nouveau container basé sur l'image ubuntu avec un nom différent:

```
# docker run -ti --name=bob ubuntu /bin/bash
```

2. Inspectez ce qui se passe dans le container avec la commande suivante:

```
# docker logs -f bob
```

3. Inspectez avec le temps ce qui se passe dans le container avec la commande suivante:

```
# docker logs -ft bob
```

4. Pour avoir une information sur les ressources qu'utilise votre container, utilisez la commande suivante:

```
# docker stats bob
```

Gestion des images

1. Listez les images disponible avec la commande suivante:

```
# docker images
```

2. Tirez l'image d'apache avec la commande suivante:

```
# docker pull httpd
```

3. Créez votre propre image qui va vous permettre de lancer un serveur Web apache. Pour cela, il faut définir la méthode de construction du container dans un fichier

```
# mkdir -p ~/Docker/Apache
```

4. Mettez la config suivante dans un fichier nommez Dockerfile:

```
FROM ubuntu:latest
MAINTAINER Votre nom
RUN apt-get -yqq update && apt-get install -yqq apache2

WORKDIR /var/www/html

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_PID_FILE /var/run/apache2.pid
ENV APACHE_RUN_DIR /var/run/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2

RUN mkdir -p $APACHE_RUN_DIR $APACHE_LOCK_DIR $APACHE_LOG_DIR

ENTRYPOINT [ "/usr/sbin/apache2" ]
CMD [ "-D", "FOREGROUND" ]
EXPOSE 80
```

5. Créez l'image avec la commande suivante:

```
# docker build -t="benjamin/apache" .
```

6. Listez les images avec la commande suivante:

```
# docker images
```

Faire tourner des applications avec Docker

1. Créez un container en mode daemon avec la commande suivante:

```
# docker run -d --name=demon ubuntu /bin/bash
```

Il s'est tout de suite arrêté.

2. Créez un container en utilisant la commande suivante:

```
# docker run -d --name=demon ubuntu /bin/sh -c "while true ; do echo hello world ; sleep 1 ; done"
```

hello world s'affiche en boucle

3. Démarrez un container benjamin/apache en mode demon en exposant le port 80 avec la commande suivante:

```
# docker run -d -p 80 --name=apache benjamin/apache
```

4. Pour trouver quel est le port choisi par Docker, il suffit de taper la commande suivante:

```
# docker port apache 80
```

5. Stoppez, détruisez votre machine apache et recréez votre container avec la commande suivante:

```
# docker run -d -p 80:80 --name=apache benjamin/apache
```

6. Créez un répertoire website dans le répertoire Apache précédemment créé

7. Mettez un fichier index.html avec le contenu suivant:

```
<html> Hello world </html>
```

8. Recréez votre container mais avec la commande suivante:

```
# docker run -d -p 80:80 -v ~/Docker/Apache/website:/var/www/html --name=apache benjamin/apache
```