



TD1 : Composition avec Docker de services Cloud SaaS, PaaS, et IaaS

1. Prérequis

- 1- Installer sur Docker sur votre machine, ou utilisez une machine préinstallée sur le Cloud.
- 2- Connectez-vous au site <https://hub.docker.com/> et créer votre compte Docker. Il est conseiller de choisir votre nom d'utilisateur de la façon suivante: prénom.nom

2. Création sur votre machine d'un espace de travail Docker

Avant de procéder avec la création de votre premier image Docker, créer sur votre machine ou celle que vous utilisez pour Docker, votre espace de travail Docker. Pour cela, lancer votre outil de ligne de commande, rendez-vous dans l'endroit où vous souhaitez créer votre espace de travail Docker, et exécuter la ligne de commande suivante :

```
sudo mkdir Docker-Images
```

La commande ci-dessus créera un répertoire sur votre machine nommé Docker-images, à utiliser pour vos projets Docker.

3. Création d'un image Docker Apache

Pour créer une image Apache, rendez-vous dans votre espace de travail Docker et créer un nouveau répertoire, nommée Apache-Server-Image, comme espace de travail pour votre projet de création de l'image Apache. Pour cela, exécuter les lignes de commande ci-dessous :

```
sudo mkdir Docker-Images
sudo mkdir Apache-Server-Image
```

Maintenant, vous allez créer un fichier Docker pour la création de votre image Docker Apache. Pour ce faire, rendez-vous dans le répertoire Apache-Server-Image et créer un fichier nommé Dockerfile comme suivant :

```
cd Apache-Server-Image
sudo vim Dockerfile
```

Copiez dans le fichier Dockerfile les instructions ci-dessous, cliquez sur le bouton ESC de votre clavier, puis taper la commande :wq pour enregistrer et quitter votre fichier Dockerfile.

```
FROM ubuntu:latest
MAINTAINER Yehia TAHER <yehia.taher@gmail.com>
RUN apt-get update
RUN apt-get -y upgrade
RUN apt-get -y install apache2
ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
ENV APACHE_PID_FILE /var/run/apache2.pid
EXPOSE 80
CMD /usr/sbin/apache2ctl -D FOREGROUND
```

Maintenant votre fichier Dockerfile contient toutes les instructions nécessaires pour la création de votre image docker. Pour la créer, exécuter la commande ci-dessous :

```
sudo docker build -t ytaher/apache .
# n'oublier pas l'espace suivi d'un point à la fin de votre ligne de commande
# à la place de ytaher, utiliser votre nom d'utilisateur
```

La commande ci-dessus créera une image Docker, nommée Apache et la publiera dans le registre Docker (en local).

Pour vérifier si a été bien créée, exécuter la commande suivante :

```
sudo docker images
```

Votre image récemment créée devrait être affichée dans la liste des images existantes dans le registre Docker de votre machine.

4. Publication d'une image Docker dans Dockerhub

Pour rendre votre image accessible par tout le monde (ou un groupe d'utilisateurs), vous devez la publier dans le registre Dockerhub. Pour cela, connectez-vous d'abord à dockerhub en exécutant la ligne de commande suivante :

```
sudo docker login
```

Puis renseigner votre nom d'utilisateur et votre mot de passe.

Pour publier l'image dans le registre Dockerhub, exécuter la commande suivante :

```
sudo docker push ytaher/apache
# à la place de ytaher, utiliser votre nom d'utilisateur
```

5. Lancer un « Docker container »

L'instanciation de l'image Docker Apache va créer un container Docker déployant un serveur apache. Pour ce faire, exécuter la ligne de commande suivante :

```
sudo docker run -d -p 8080:80 ytaher/apache
```

A l'exécution de cette commande, l'image ytaher/apache sera téléchargée dans le registre Docker (en local) et un conteneur correspondant sera créé. Le conteneur sera accessible via le port 8080 de la machine hôte. Le port 8080 est mappé au port 80 du conteneur. Ainsi, toute requête reçue par la machine hôte sur le port 8080 est directement transmise au port 80 du conteneur.

Vous pouvez créer plusieurs conteneurs Apache sur votre machine en renseignant chaque fois un port différent (8080, 8081, etc.).

6. Connexion à un « Docker container »

Pour vous connecter à un conteneur docker, il faut d'abord récupérer son identifiant en exécutant la commande :

```
sudo docker ps
```

Une fois l'identifiant de la commande est récupéré, connectez-vous au conteneur en exécutant la commande suivante :

```
sudo docker exec -it dba31845d2ba bash # dba31845d2ba ID du conteneur
```

Pour déployer une application Web dans votre conteneur, rendez-vous dans le répertoire var/www/html de votre conteneur en exécutant la commande :

```
cd var/www/html
```

créer un nouveau répertoire pour votre application en exécutant la commande :

```
sudo mkdir mywebapp
```

Puis télécharger les fichiers html de votre application dans le répertoire créer.

Pour tester votre application, pointer votre navigateur Web à l'adresse :
<http://localhost:8080/mywebapp>

7. Crédit d'une image Docker déployant une application Web donné sur un serveur apache

Au lieu de déployer l'application manuellement dans le serveur Apache, créer une nouvelle image docker qui déploie votre application web dans un serveur Apache.

Une fois l'image créée, publiez la dans Docker Hub, puis instancier la en créant un ou plusieurs containers déployant votre application Web.