Créer un lamp avec Docker

LAMP définition

Un système LAMP est une solution d'hebergement web :

- **L**inux, le système d'explotation sur lequel tourne l'hebergement
- **A**pache, le serveur web qui gère les requetes HTTP du client.
- **M**ySQL, le système de gestion de base de données
- **P**HP, le langage de programmation back-end

A eux quatre ils formes l'environnement d'execution d'un back-end.

Il existe également l'appelation WAMP qui désigne la même chose sous Windows plutot que Linux.

Aujourd'hui, grâce à Docker, le système d'exploitation du développeur importe peu. Que vous soyez sous Mac, Linux ou Windows, avec Docker, nous allons avoir une installation et une exploitation uniformisé de notre serveur LAMP.

Docker run

Pour notre LAMP nous avons donc besoin de virtualiser 3 logiciels : apache, mysql et php. Pour plus de confort on souhaite aussi avoir PhpMyAdmin d'installer. Pour ceci nous utiliseront 3 images docker :

- **php:8.2-apache**, l'image officel de PHP contenant PHP et apache. Grace à cette image pas besoin de lancer deux conteneurs différent pour PHP et apache, les deux logiciels seront dans le même conteneur.
- **mysql:latest**, la dernière version de mysql notre système de gestion de base de donnée (SGBD).
- **phpmyadmin:latest**, la dernière version de phpmyadmin

Toute ces images sont disponible sur Docker Hub. La commande `docker run` nous permet de télécharger ces images et lancer les conteneurs associées. Il nous faut donc lancer 3 fois `docker run`.

> **Différence entre image et conteneur**
> Le conteneur est le programme que nous voulont virtualiser, alors qu'une image est simplement

> le patron de conception du conteneur. On crée des conteneurs à partir d'une image. Un conteneur est un programme actif qui est virtualiser dans docker, il est executé sur notre ordinateur, une image quant à elle ne s'execute pas car ce n'est pas un programme.

Commandes Docker pour lancer un LAMP

Voici les commandes à lancer pour mettre en place un LAMP :

```
docker network create lamp-net

docker run -d --name lamp-php --network=lamp-net -p 80:80 php:8.2-apache

docker exec lamp-php docker-php-ext-install pdo
docker exec lamp-php docker-php-ext-install pdo_mysql
docker exec lamp-php docker-php-ext-install mysqli

docker run -d --name lamp-mysql --network=lamp-net -e MYSQL_ROOT_PASSWORD=root mysql

docker run -d --name lamp-pma --network=lamp-net -e PMA_HOST=lamp-mysql -p 8080:80 phpmy

docker start lamp-php
docker start lamp-mysql
docker start lamp-pma
```

Si tout c'est bien passé vous verrez un page "Error Forbidden" sur localhost dans votre navigateur, c'est normal il vous faut rajouter un fichier index.html ou index.php pour définir une page d'accueil.

Pour rajouter et mettre à jour sont site il faut rajouter des fichiers sources dans le container lamp-php, pour ceci utiliser l'extension VSCode Dev Container et ouvrez le container lamp-php dans VSCode.

> **docker: Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:80 -> 0.0.0.0:0: listen tcp 0.0.0.0:80: bind: address already in use.**
> Pour le bon fonctionnement du serveur apache il vous faut **stopper tout serveur apache** qui tournerait deja sur votre PC ou tout service qui utilise le port 80.
> **Pour linux tapez :**
>
> ```
> sudo systemctl stop apache2    # Arrete apache2
> sudo systemctl disable apache2 # Désactive le lancement au démarrage du PC
> ```
>
> **Pour windows** fermez WAMP, XAMPP ou toute installation local de apache.

--name , nommer son conteneur

Le paramètre --name permet de donner un nom à son conteneur. Le nom du conteneur agit comme un nom de domaine.

Le nom de domaine de mon serveur mysql est : `lamp-mysql` .

```
docker run -d --name lamp-mysql --network=lamp-net -e MYSQL_ROOT_PASSWORD=root mysql
```

Ce qui signifie que pour me connecter avec PDO en php à MySQL la valeur de `host` est `lamp-mysql` .

```
$bdd = new PDO("mysql:host=lamp-mysql;dbname=ma-bdd","root","root");
```

Conteneur en tâche de fond

Le paramètre -d permet simplement de faire tourner un conteneur en tâche de fond.

Grâce à `-d` mon serveur web php apache tourner en tâche de fond sans s'arreter.

```
docker run -d --name lamp-php --network=lamp-net -p 80:80 php:8.2-apache
```

Créer un réseau virtuelle avec `docker network`

Mes container doivent communiquer entre eux pour ceci il faut qu'il soit sur le même réseau. Par défaut un container docker est isolé du reste du monde, je doit donc créer un réseau puis inscrire mes containeeurs dans le réseau lors de leurs créations.

Je crée un nouveau réseau docker nommé lamp-net.

```
docker network create lamp-net
```

Ajouter un containeur au réseau lamp-net

Le paramètre `--network=` permet de rajouter un containeur à un réseau docker, ici je vous veut la rajouter au réseau lamp-net.

```
docker run -d --name lamp-php --network=lamp-net -p 80:80 php:8.2-apache
docker run -d --name lamp-mysql --network=lamp-net -e MYSQL_ROOT_PASSWORD=root mysql
docker run -d --name lamp-pma --network=lamp-net -e PMA_HOST=lamp-mysql -p 8080:80 phpmy
```

Les conteneur lamp-php, lamp-mysql et lamp-pma sont à présent tous dans le réseau lamp-net.

Variable d'environnement

Parfois un conteneur à besoin d'information complémentaire pour fonctionner, par exemple mysql à besoin d'un mot de passe pour l'utilisateur root et phpmyadmin à besoin du nom de domaine du serveur mysql. Ces informations s'appelle des variables d'environnement.

Ces variables sont précisés via le paramètre `-e` .

Syntaxe :

```
docker run  -e NOM_VARIABLE_ENV=value  image-name
```

Exemple, MYSQL_ROOT_PASSWORD de l'image mysql :

La variable d'environnement MYSQL_ROOT_PASSWORD défini le mot de passe root de mysql. Il est primoridal de le définir car vous en avez besoin pour pour connecter à vos BDD avec PHP PDO.

```
docker run -d --name lamp-mysql --network=lamp-net -e MYSQL_ROOT_PASSWORD=root mysql
```

Exemple, PMA_HOST de l'image mysql :

PMA_HOST permet de défini le nom de domaine ou l'adresse ip du serveur mysql auquelle PhpMyAdmin doit se connecter. Le nom de domaine d'un conteneur est défini via le paramètre --name.

```
docker run -d --name lamp-pma --network=lamp-net -e PMA_HOST=lamp-mysql -p 8080:80 phpmy
```

Port binding

Les conteneurs docker sont, par défaut, isolé du PC hôte (votre ordinateur) cela signifie car le navigateur du PC hôte et le serveur apache ne peuvent pas communiquer.

Un serveur HTTP est toujours accessible via le port numéro 80 et c'est aussi vrai pour notre conteneur apache. Il faudrait donc relier le port 80 de l'hôte (localhost:80) avec le port 80 du conteneur apache.

Rien de plus simple grâce au paramètre `-p` .

Ici je relie le port 80 du `localhost` avec le port 80 du container.

```
docker run -d --name lamp-php --network=lamp-net -p 80:80 php:8.2-apache
```

Ce qui signifie que si je tape `localhost:80` dans un navigateur c'est le serveur apache contenerisé qui sera visé.

> Par défaut le navigateur utilise le port 80 pour HTTP et le port 443 pour le HTTPS, il est donc inutile décrire `localhost:80` dans la barre de recherche vous pouvez vous contentez d'écrire `localhost` .