Les Networks

Nous n'allons pas avoir besoin seulement de **apache** avec **PHP**. Mais aussi de **MySQL** et il pourrait nous être utile d'avoir **PHPMyAdmin**.

Pour lier plusieurs conteneur entre eux, nous allons devoir créer un "**network**". Car par défaut, même si l'on rend accessible depuis l'extérieur un conteneur, il reste isolé des autres dans la machine virtuelle. On pourra indiquer à un conteneur qu'il fait parti du même network qu'un autre. Tout autre conteneur utilisant ce même network pourront alors communiquer ensemble.

Pour créer un network, on utilisera :

```
docker network create firstNetwork
```

Avec "*firstNetwork*" le nom du network. D'autres commandes peuvent être utile :

```
# Pour voir tous les networks :
docker network ls
# Pour supprimer un network :
docker network rm nomDuNetwork
```

Pour connecter un conteneur au network, nous pouvons :

```
# ajouter cette option au run d'un conteneur
--network nomDuNetwork
# ou connecter un conteneur déjà existant au network :
docker network connect nomDuNetwork nomDuConteneur
```

Mettons cela en pratique:

Installer MySQL

Pour installer MySQL, nous avons besoin d'une commande avec quelques paramètres en plus :

```
docker run --name firstMySQL --network firstNetwork -p 3307:3306 -e
MARIADB_ROOT_PASSWORD=root -d mariadb
```

Voyons les nouveautés qui apparaissent ici :

- **-e** sert à indiquer des variables d'environnement qui aideront à faire tourner l'application.
- **MARIADB_ROOT_PASSWORD=** est la variable d'environnement qui permet d'indiquer le mot de passe de l'utilisateur par défaut **root**

Nous avons maintenant une base de donnée MySQL qui fonctionne.

Si on souhaite se connecter à notre BDD avec une invite de commande, il nous suffira de nous connecter en invite de commande à notre conteneur.

Pour nous connecter à la BDD avec un outil extérieur à la VM (VS code, MySQL Workbench...) on utilisera "localhost" (ou 127.0.0.1) comme adresse, et le port de la VM (ici 3307).

Pour nous connecter depuis un conteneur sur le même network, nous devrons utiliser le port du conteneur (ici 3306). Et au choix, soit le nom que l'on a donné au conteneur, soit son adresse IP, pour obtenir cette IP, nous pouvons taper la commande suivante :

```
docker inspect nomDuConteneur
```

vous trouverez l'information dans "NetworkSettings/Networks/nomDuNetwork/IPAddress".

Il est aussi possible de taper directement :

```
docker inspect -f '{{range.NetworkSettings.Networks}}{{.IPAddress}}{{end}}'
nomDuConteneur
```

Ce qui nous donnera directement l'information voulue.

Si on souhaite faire un import dans la BDD. Il nous faudra avoir accès au fichier à importer, pour cela nous allons devoir le mettre dans le conteneur.

Le plus simple lorsque l'on a juste un seul fichier à déplacer comme ceci, est de passer par la commande de copie de docker:

```
# De l'ordinateur vers le conteneur:
docker cp PathToFile nomDuConteneur:PathToDestination
# Depuis le container vers l'ordinateur:
docker cp nomDuConteneur:PathToFile PathToDestination
```

PHPMyAdmin

Les invites de commandes sont bien pratique, mais si on peut avoir une interface pour travailler avec, ce ne sera que plus plaisant:

```
docker run --name phpmyadmin --network nomDuNetwork -e PMA_HOST=IP -d -p 8087:80
phpmyadmin
```

- **PMA_HOST=** se fera suivre par l'adresse IP du conteneur de MySQL.
- **IP** remplacez cela par l'IP ou le nom du conteneur ayant la BDD.

Nous pourrons donc nous connecter grâce à l'utilisateur "root" et le mot de passe paramétré précédemment, ici "root" aussi.