

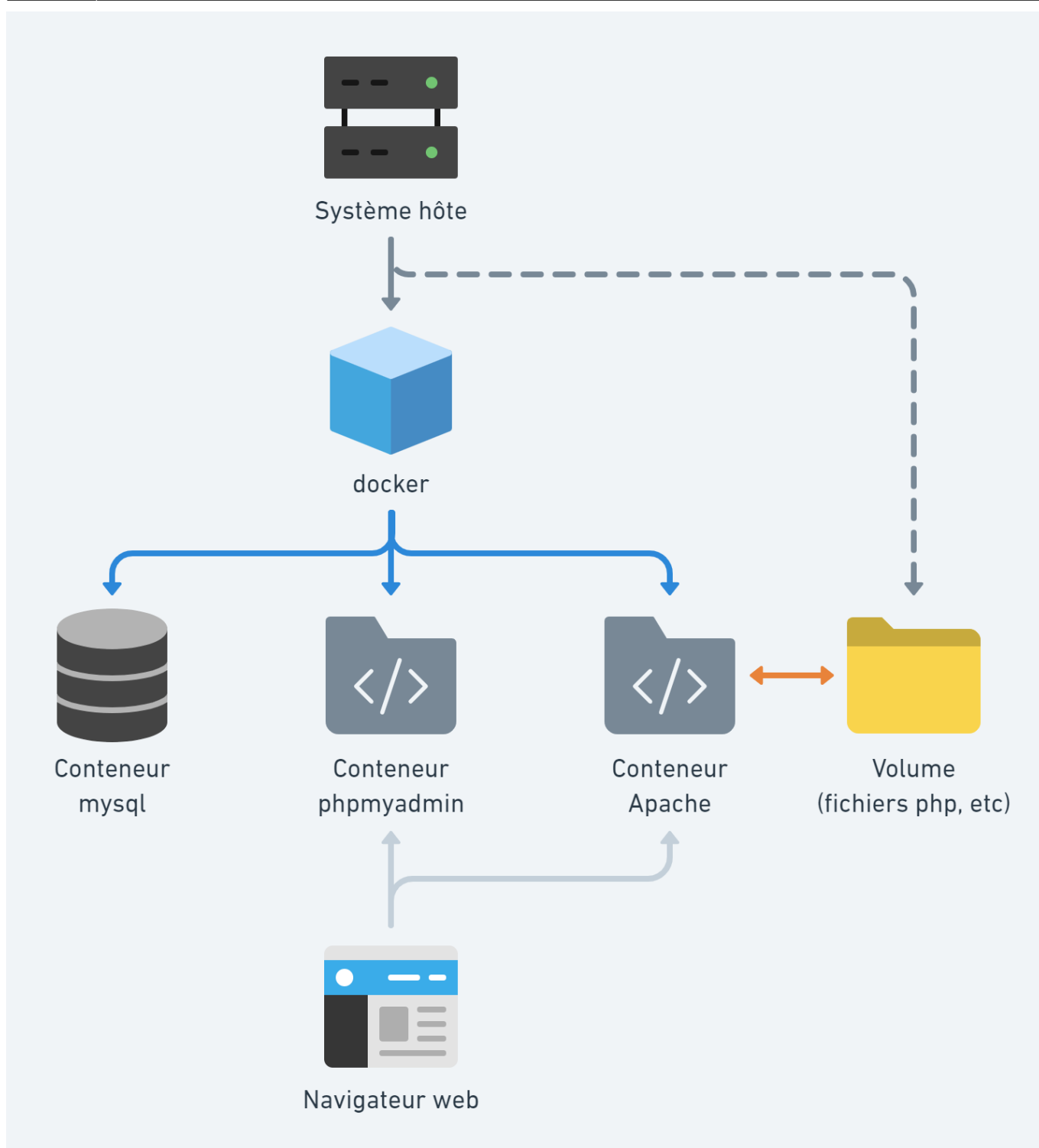
Activité : créer un environnement LAMP avec Docker

Objectif

Réaliser un environnement LAMP avec 3 conteneurs :

- Conteneur Docker **Apache** (serveur Web) ;
- Conteneur Docker **mariadb** (base de données) ;
- Conteneur Docker **phpmyadmin** (site Web d'administration de la base de données).

Les fichiers du site Web et de la base de données seront enregistrés en local et non dans les conteneurs en utilisant le principe des volumes de Docker.



Utilisation d'un serveur Web Apache avec PHP

Créez un conteneur avec les informations suivantes :

- dossier local contenant les **fichiers du site Web** ;
- dossier local pour la **persistance des logs du serveur Apache** ;
- mappage du site Web sur le **port local 8001** ;
- conteneur Docker que vous avez créé (basé sur l'image php:8.1-apache) : **charles/php**.

Le site Web doit être accessible à l'URL : <http://localhost:8001>

Toute page créée ou modifiée dans le dossier local **C:\www\html\siteweb** est publiée dans le site Web.

Utilisation d'un serveur de base de données conteneurisé

Votre architecture doit maintenant mettre en oeuvre :

- un conteneur Docker **php:8.1-apache** pour le **serveur Web** ;
- un conteneur Docker **mariadb** pour la base de données.

Pour cela, vous allez **créer un nouveau réseau Docker**, en plus celui créé par défaut qui est appelé **bridge**.

- Pour visualiser les réseaux créés par Docker utilisez la commande suivante :

```
PS> docker network ls
PS C:\Users\Charles> docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
0357f1f2d2c1        bridge             bridge              local
726b383a30a3        host               host                local
2550a660d6dd        none              null                local
```

- créez un nouveau réseau Docker appelé **lamp** :

```
docker network create lamp
```

Pour le site Web dynamique, l'image officielle Docker de la base de données **MariaDB** sera utilisée avec les informations suivantes :

- le conteneur MariaDB sera dans le réseau Docker **lamp** ;
- le service MariaDB sera en **écoute** sur le port par défaut **3306** (si le port d'écoute ne change pas, m il n'est pas nécessaire de le préciser à la création du conteneur) ;
- Le mot de passe du compte **root** de **MariaDB** est passé au conteneur via la variable d'environnement **MARIADB_ROOT_PASSWORD** ;
- les bases de données du conteneur situé dans **/var/lib/mysql** doivent être externalisées dans un dossier local.

Pour en savoir plus sur l'utilisation du conteneur MariaDB :



- https://hub.docker.com/_/mariadb/

Certaines variables données dans ce lien ne sont plus bonnes.

- Lancement d'un conteneur appelé **servbdd** basé sur l'image **MariaDB**, dans le réseau **lamp** avec comme mot de passe du compte root **passwordmariadb** :

```
docker run --name servbdd --network lamp -d -v c:\www\html\siteweb-  
bdd:/var/lib/mysql -e MARIADB_ROOT_PASSWORD=passwordmariadb mariadb
```

Utilisation d'un conteneur phpmyadmin pour administrer la base de données

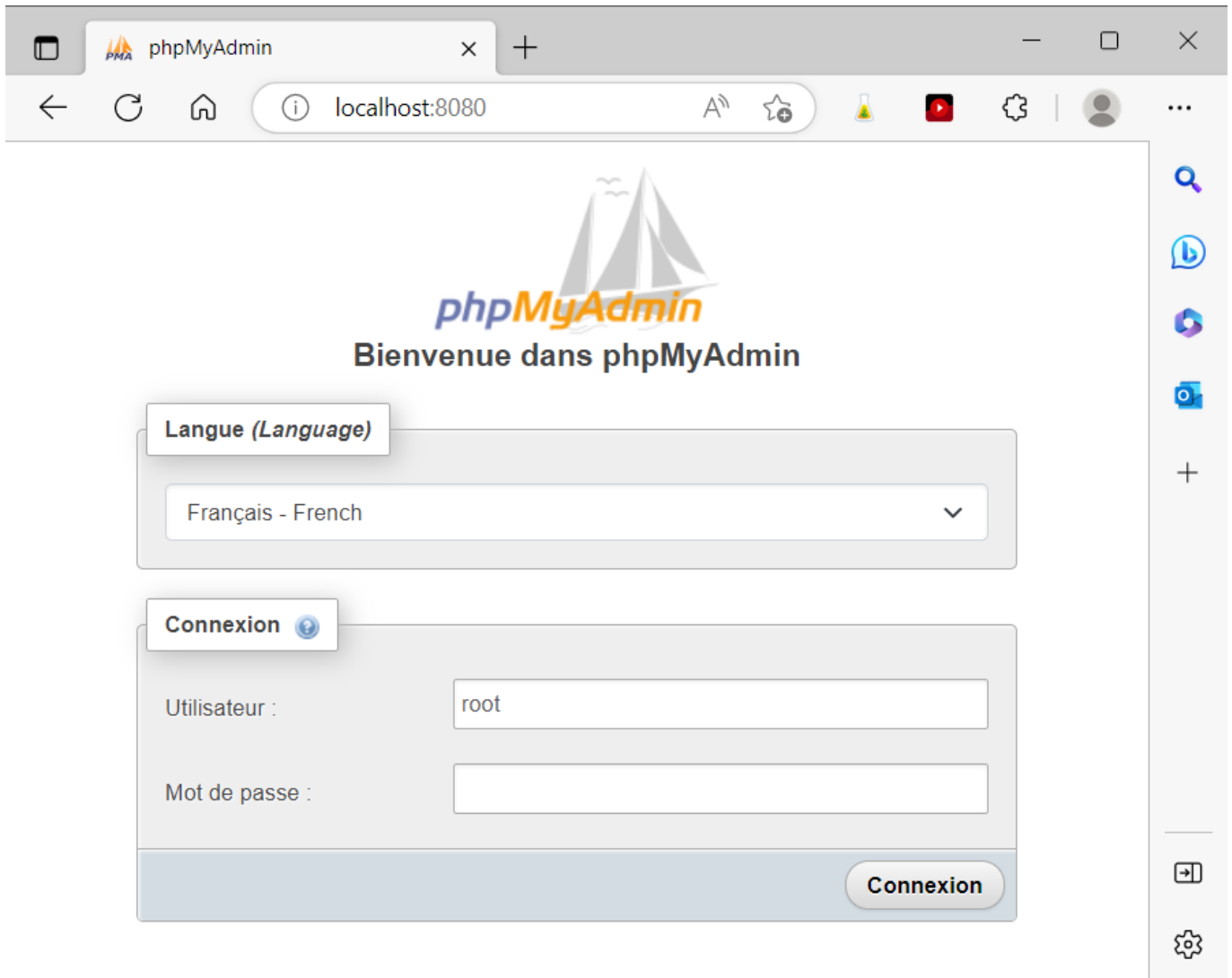
Phpmyadmin est un **site Web écrit avec PHP** qui permet d'**administrer** un serveur **MariaDB**.

Lors de la création du conteneur basé sur l'image **phpmyadmin**, il sera nécessaire :

- de définir le port d'écoute local de votre pc : par exemple **8080** ;
- de **faire le lien** avec le conteneur **servbdd** créé à partir de l'image Mariadb :
 - en utilisant le même réseau Docker **lamp**.
 - en renseignant le **nom du serveur MariaDB** dans la variable d'environnement **PMA_HOST**.

```
docker run --name phpmyadmin -d --network lamp -e PMA_HOST=servbdd -p  
8080:80 phpmyadmin
```

En accédant au site de **PhpMyadmin** à l'URL <http://localhost:8080>, vous devez visualiser la page d'accueil de PhpMyAdmin et pouvoir vous authentifier avec le compte **root** et le mot de passe **passwordmariadb** défini précédemment :



Pour en savoir plus sur l'utilisation de l'image phpmyadmin :

- https://hub.docker.com/_/phpmyadmin

Il est également possible de gérer la base de données MariaDB en utilisant le client MariaDB présent dans l'image MariaDB.

Pour cela, lancez un conteneur basé sur l'image MariaDB, en mode interactif :



```
docker run -it --rm --network lamp mariadb mariadb -h servbdd -u root -p
```

Remarques :

- utilisez le même réseau Docker **lamp** ;
- le paramètre **-h** doit être utilisé pour préciser le nom du conteneur **servbdd** contenant le serveur MariaDB.
- la paramètre **-rm** permet de détruire le conteneur dès que vous le quittez.

Architecture complète LAMP

Il ne reste plus maintenant qu'à :

- **récréer le conteneur du site Web** dans le même réseau Docker **lamp** ;
- **créer la base de données** du site web à partir du **script de la sauvegarde de la base** ;
- **créer** dans le serveur MariaDB le **compte utilisé par le site Web** pour accéder à la base de données ;
- Tester que le site est opérationnel.

Activité à faire

Mission 1 :

- **Créez le conteneur Docker du site Web** dans le **même réseau Docker** que celui de la base de données MariaDB.

Mission 2 :



- Prenez connaissance des **paramètres de connexion** nécessaires au site Web pour se connecter à la base de données :
 - **nom du serveur** : host= ? ;
 - **base de données** : dbname= ? ,
 - **nom de compte** : ?
 - **mot de passe** : ? . 'nolarkuser', 'nolarkpwd');
- Utilisez PhpMyAdmin pour :
 - **restaurer** la base de données du site Web ;
 - **créer** le compte de connexion ;
- Testez que le site est **complètement opérationnel**.

Autres ressources

Lien :

<https://www.alsacreations.com/tuto/lire/1836-Docker--decouverte-et-environnement-LAMP-pour-WordPress.html>

Retour Accueil Docker

- [Docker](#)

From:
<https://siocours.lycees.nouvelle-aquitaine.pro/> - **Les cours du BTS SIO**

Permanent link:
<https://siocours.lycees.nouvelle-aquitaine.pro/doku.php/reseau/docker/lampdocker?rev=1677445658>

Last update: **2023/02/26 22:07**

