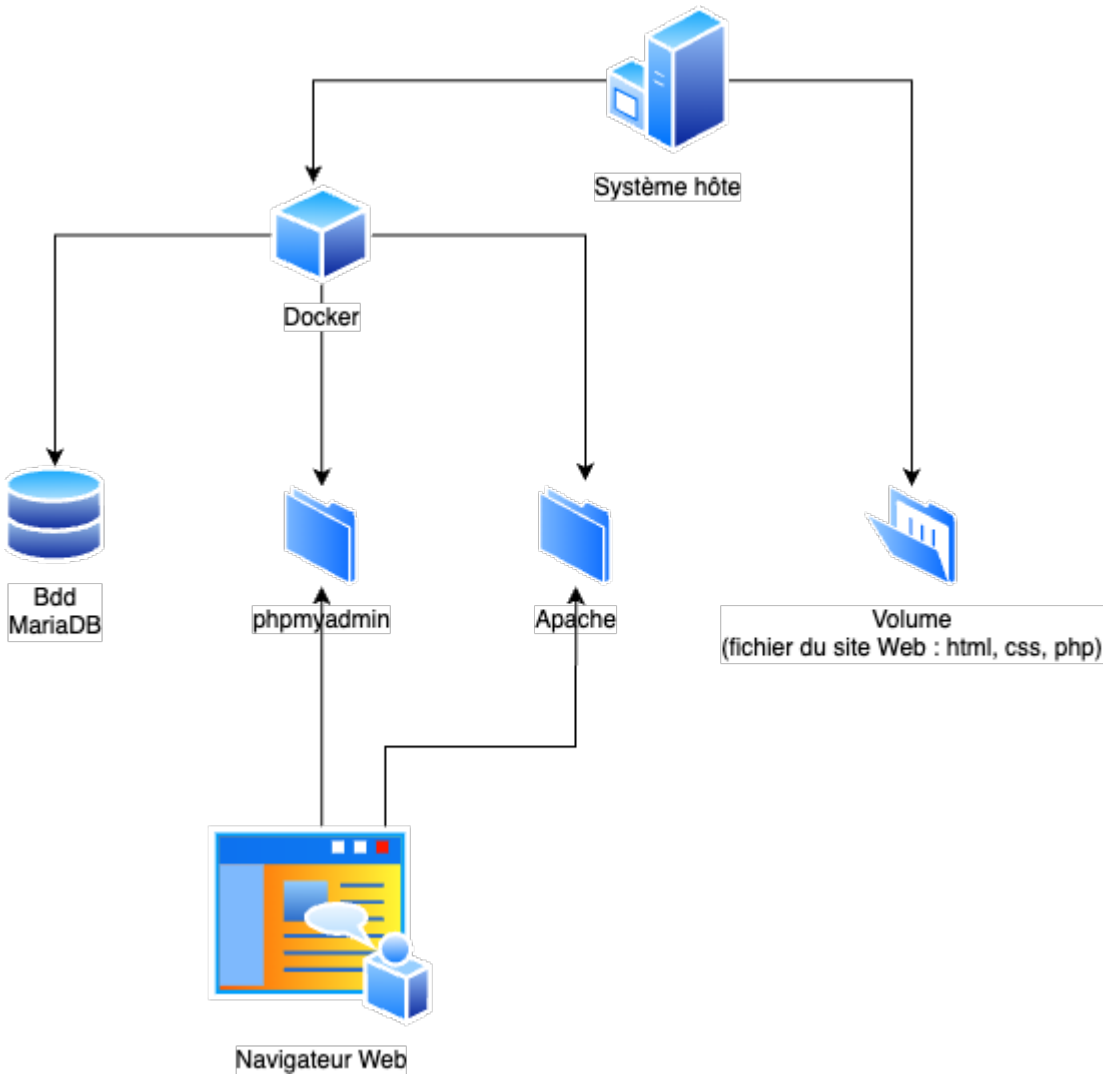
1. Activité : créer un environnement LAMP avec Docker

1.1. Objectif

Réaliser un environnement LAMP avec 3 conteneurs pour le **site Web Nolark** :

- 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.



1.2. 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) : **ctecher/php-mariadb**.

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

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

1.3. Utilisation d'un serveur de base de données conteneurisé

Votre architecture doit maintenant mettre en oeuvre :

- un conteneur Docker créé à partir de l'image **ctecher/php-mariadb** pour le **serveur Web** ;
- un conteneur Docker créé à partir de l'image **mariadb** pour la base de données.

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

- le service MariaDB sera en **écoute** sur le port par défaut **3306** (si le port d'écoute ne change pas, 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.
- Lancement d'un conteneur appelé **servbdd** basé sur l'image **MariaDB** avec comme mot de passe du compte root **passwordmariadb** :
- Code Windows

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

- Code Linux

```
docker run --name servbdd  -d -v /home/user/www/html/siteweb-bdd:/var/lib/mysql -e
MARIADB_ROOT_PASSWORD=passwordmariadb  mariadb
```

1.4. 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 l'opérateur **link** ,
  - en renseignant le **nom du serveur MariaDB** dans la variable d'environnement **PMA_HOST**.

L'opérateur **link** permet à un conteneur d'avoir accès directement au service d'un autre conteneur :

- via le **port** de ce service (par défaut ou redéfini par vous),
- en accédant au conteneur **par le nom** précisé dans l'opérateur **link** : pour cela Docker modifie le fichier **/etc/hosts** pour associer le nom à l'adresse IP du service.

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

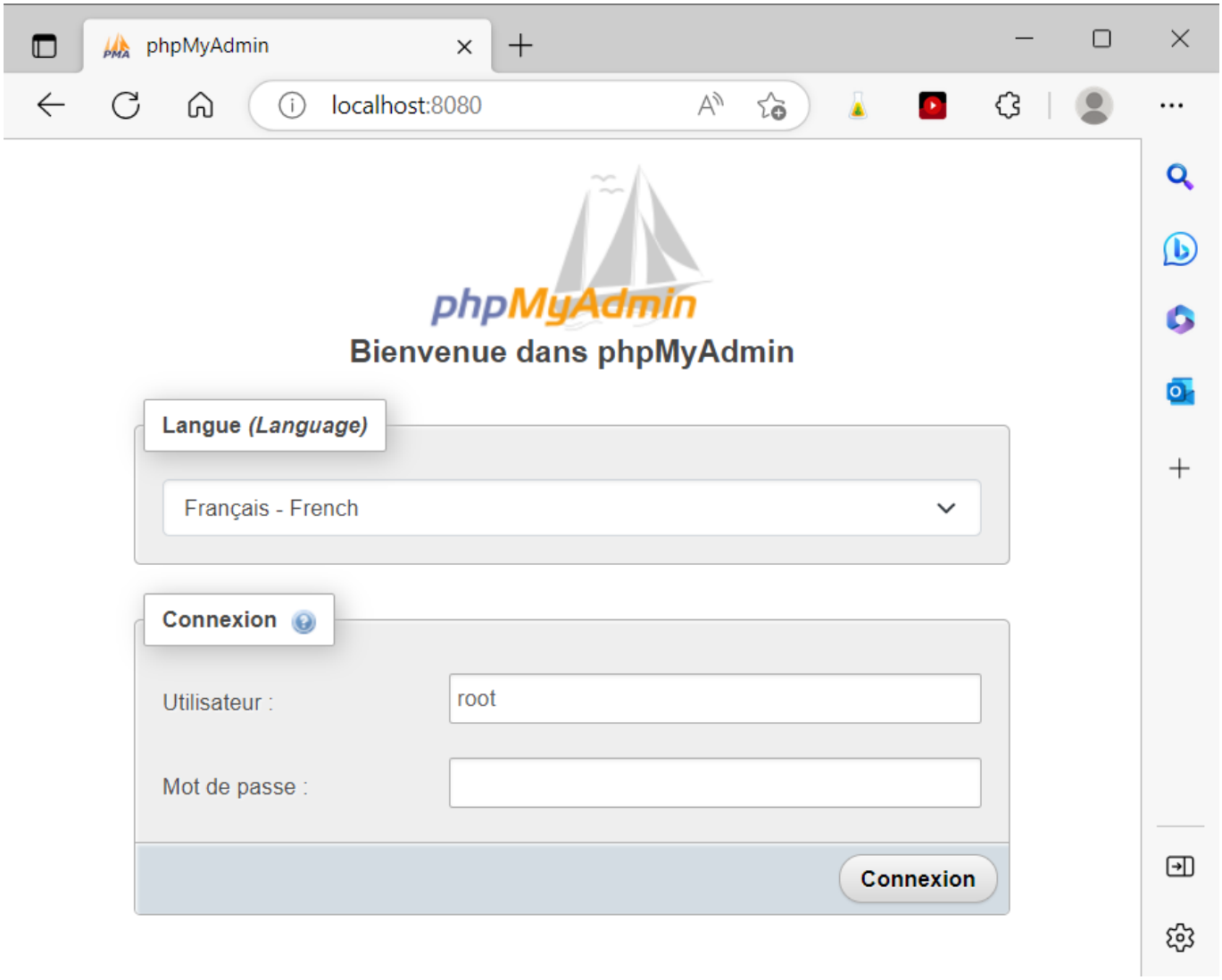
1.5. test

L'opérateur **link** prend en argument le nom_du_conteneur_d_origine:alias_de_ce_conteneur où **alias_de_ce_conteneur** est un nom d'alias libre qui peut être le même que celui du conteneur d'origine.

1.6. Titre

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 **pswordmariadb** défini précédemment.

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 :



1.7. Architecture complète LAMP

Il ne reste plus maintenant qu'à :

- **créer la base de données** du site web à partir du **script de la sauvegarde de la base** ;
- **récréer le conteneur du site Web** en le liant au conteneur **srvbdd** ;
- Tester que le site est opérationnel.

1.8. Activité à faire

Mission 1 :

- Utilisez le site **PhpMyAdmin** pour restaurer la base de données du site Web à partir du **script** de sauvegarde de la base de données.

Mission 2 :

- **Créez** le conteneur Docker du site Web en utilisant l'image que vous avez créée pour php ou bien l'image **ctecher/php** et en le liant au conteneur **servbdd** .
- Testez que le site est **complètement opérationnel**.

Vous pouvez prendre connaissance des **paramètres de connexion** nécessaires au site Web pour se connecter à la base de données dans l'un des fichiers du site Web.

1.9. Autres ressources

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

1.10. Mémento Docker

2. 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=1734451163

Last update: **2024/12/17 15:59**