1. Activité : créer une image Docker avec un fichier Dockerfile

1.1. Présentation

Les images Docker disponibles sur Docker Hub, sont construites à partir d'une image de base dans laquelle est ajouté des paquets logiciels et des éléments de configuration et de personnalisation.

Pour cela, ces nouvelles images sont construites à partir d'un fichier appelé **Dockerfile** décrivant toutes les opérations à effectuer.

1.2. Créer une image personnalisée de l'image php

L'image **php:8.1-apache** contient le serveur web **Apache2** et une **configuration de php** avec uniquement les pilotes pour gérer les accès à la base de données **sqlite**.

Pour pouvoir disposer d'une image contenant le **serveur Web Apache** mais aussi **php** configuré pour communiquer avec la base de données **MariaDB**, vous allez créer une **nouvelle image** à partir de l'image **php:8.1-apache** qui contiendra les pilotes nécessaires.

- Créer un ficher appelé **Dockerfile** (sans extension) ;
- Editez ce fichier pour y mettre les instructions suivantes :

Dockerfile

```
FROM php:8.1-apache
RUN apt-get update && apt-get upgrade -y
RUN docker-php-ext-install mysqli && docker-php-ext-enable mysqli
RUN docker-php-ext-install pdo pdo_mysql
```

- Depuis l'invite de commande, le **terminal** pour Linux ou **Powershell** pour Windows, dans le dossier contenant le **fichier Dokerfile**, lancez la commande suivante pour créer :
    - une image **appelée php-mariadb** et **préfixée** par votre pseudo Docker :

```
docker build -t ctecher/php-mariadb .
```

1.3. Test de votre image

Pour pouvoir tester votre nouvelle image **php-mariadb**, vous devez mettre en œuvre :

- un **conteneur MariaDB** appelé **srvbdd**;
- un **script bdd_phptest_sov.sql** de **création** d'une base de données appelée **phptest** ;
- un **conteneur** appelé **testphp** créé à partir de la nouvelle image **php-mariadb** ;
- un **script index.php** permettant de **tester la connexion** à la base de données **phptest**.

bdd_php-test_sov.sql

```
CREATE DATABASE phptest;
USE phptest;
CREATE TABLE produits (
    nom VARCHAR(50)
);
INSERT INTO produits (nom) VALUES ('fraises');
INSERT INTO produits (nom) VALUES ('melons');
```

index.php

```
<?php
$host = 'srvbdd'; // nom du conteneur MariaDB (base de données)
$db   = 'phptest';
$user = 'root';
$pass = 'Sio1234*';

try {
    $pdo = new PDO("mysql:host=$host;dbname=$db;charset=utf8", $user, $pass);
```

```
} catch (Exception $e) {
    die("Erreur connexion : " . $e->getMessage());
}

$stmt = $pdo->query("SELECT nom FROM produits");

echo "<h1>Contenu de la base de test</h1>";
echo "<table border='1'>";
echo "<tr><th>Nom</th></tr>";

while ($row = $stmt->fetch()) {
    echo "<tr><td>" . htmlspecialchars($row['nom']) . "</td></tr>";
}

echo "</table>";
?>
```

1.3.1. Création du conteneur mariadb appelé srvbdd

- création du dossier contenant le script **bdd_php-test_sov.sql** :

```
mkdir init
cd init
```

- copie du fichier **bdd_php-test_sov.sql** dans le dossier init.
- création du conteneur **srvbdd** avec :
    - la définition du mot de passe du compte root,
    - exécution automatique du contenu du fichier **bdd_php-test_sov.sql** :

```
docker run -d \
  --name srvbdd \
  -e MARIADB_ROOT_PASSWORD=Sio1234* \
  -e MARIADB_DATABASE=phptest \
  -p 3306:3306 \
  -v ./init:/docker-entrypoint-initdb.d \
  mariadb
```

Important :

- Tous les scripts présents dans **/docker-entrypoint-initdb.d** ne sont exécutés qu'au premier démarrage (quand /var/lib/mysql est vide).
- Pour exécuter à nouveau des scripts d'initialisation, il faut :
    - soit rajouter lors de la création du conteneur un volume pour le dossier **/var/lib/mysql**, dossier qui va contenir les fichier de la base de données MariaDB. Il faudra alors supprimer le contenu de ce dossier **/var/lib/mysql** pour relancer l'initialisation avec tous les scripts contenus dans le dossier **/docker-entrypoint-initdb.d**;
    - soit supprimer le conteneur et le recréer.

1.3.2. Création du conteneur Web appelé testphp

- création du dossier **www** contenant le script php **index.php** :

```
mkdir www
cd www
```

- copie du fichier **index.php** dans le dossier **www**.
- création du conteneur **testphp** avec :
    - la mappage du dossier **www** avec le dossier de piblicatio **/var/www/html** du conteneyr ;
    - le port réseau de l'hôte de votre choix. Par exemple 8080 ;
    - le lien avec le conteneur **svrbdd** (l'utilisation de ce paramètre **link** sera explicité dans l'activité suivante) ;
    - la nouvelle image que vous devez tester. Ici **ctecher/php-mariadb**

```
docker run -d \
  --name testphp \
  -p 8080:80 \
```

```
  --link srvbdd:srvbdd
  -v ./www:/var/www/html \
  ctecher/php-mariadb
```

1.4. Test de votre image avec un fichier docker-compose.yml

L'utilisation du fichier Docker-compose.yml sera explicité dans une prochaine activité.

- utilisation du dossier **init** contenant le script **bdd_php-test_sov.sql** ;
- utilisation du dossier **wws** contenant le fichier **index.html**.

Docker-compose.yml

```
services:
  # base de données MariaDB
  srvbdd:
    image: mariadb
    volumes:
      - ./init:/docker-entrypoint-initdb.d/
    environment:
      MARIADB_ROOT_PASSWORD: Sio1234*

  # site Web php
  testphp:
    image: ctecher/php-mariadb
    depends_on:
      - srvbdd
    ports:
      - 8080:80
    volumes:
      - ./www:/var/www/html
```

- lancement des conteneurs :

```
docker compose up -d
```

- arrêt des conteneurs :

```
docker compose down
```

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/creerimagedockerfile

Last update: **2026/06/21 14:54**