

Activité : créer une image Docker avec un fichier Dockerfile

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.

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 fichier 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 Dockerfile**, lancez la commande suivante pour créer :
 - une image **appelée php-mariadb** et **préfixée** par votre pseudo Docker :

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

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_test_sov.sql** de **création d'une base de données de test** ;
- 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 **test**.

bdd_test_sov.sql

```
CREATE DATABASE test;
USE test;
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 = 'test';
$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>";
?>
```

Création du conteneur mariadb appelé srvbdd

- création du dossier contenant le script bdd_test.sql

```
mkdir .\init
cd init
```

- copie du fichier bdd_test.sql dans le dossier .\init

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

Explications :

- -e MARIADBROOTPASSWORD=Sio1234* ⇒ mot de passe root.
- -e MARIADB_DATABASE=test ⇒ base créée automatiquement.
- -v .\init:/docker-entrypoint-initdb.d ⇒ montage du dossier contenant init.sql (exécuté au premier démarrage).
- mariadb ⇒ image officielle 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` et le supprimer ensuite ce dossier

Retour Accueil Docker

- [Docker](#)

From:
[/- Les cours du BTS SIO](#)

Permanent link:
[/doku.php/reseau/docker/creerimagedockerfile](#)

Last update: 2026/06/21 15:26



