



DOCKER

LAMP STACK

DOCUMENTATION COMPLÈTE STACK LAMP AVEC DOCKER

Objectif

Cette documentation guide à travers les étapes de création d'une stack LAMP en utilisant Docker, depuis l'installation de Docker jusqu'à la configuration d'un réseau personnalisé, la création d'images personnalisées pour PHP/Apache, et le lancement de conteneurs pour MySQL, PHP/Apache, et PHPMyAdmin.

Prérequis

- Docker installé sur votre système.
- Connaissances de base de Docker et des commandes Unix/Linux.

Installation de Docker

- Linux : Suivez les instructions officielles sur [Get Docker](#).
- Windows : Téléchargez Docker Desktop depuis [Docker Hub](#).

Configuration de l'Environnement de travail

1-Création d'un réseau Docker

Pour isoler notre stack LAMP, on crée un réseau Docker personnalisé

> **docker network create --driver bridge --subnet=172.28.0.0/16 lamp-network**

Cette commande crée un réseau nommé lamp-network avec un sous-réseau 172.28.0.0/16.

2-Conteneur MySQL

- Création du Volume pour la Persistance des Données

> **docker volume create mysql-volume**

- Lancement du Conteneur MySQL

> **docker run --name mysql-lamp -e MYSQL_ROOT_PASSWORD=mysecret -v mysql-volume:/var/lib/mysql --network=lamp-network -d mysql:5.7**

2-Conteneur PHP avec Apache

- Préparation du Dockerfile

Créez un fichier nommé Dockerfile-php dans un nouveau dossier. Le Dockerfile doit contenir :

FROM php:7.4-apache

COPY src/ /var/www/html/

EXPOSE 80

- Construction de l'Image

Dans le répertoire contenant le Dockerfile-php :

> **docker build -t php-apache:latest .**

- Lancement du Conteneur PHP/Apache

> **docker run --name php-apache --network=lamp-network -p 8080:80 -d php-apache:latest**

| *3-Conteneur PHPMyAdmin*

> **docker run --name php-apache --network=lamp-network -p 8080:80 -d php-apache:latest**

| *Accès et Gestion*

- PHP/Apache : Accédez à votre application PHP via <http://localhost:8080>.
- PHPMyAdmin : Gérez MySQL à <http://localhost:8081> avec root comme utilisateur et mot de passe.

| *Gestion des Conteneurs*

- Vérification de l'État des Conteneurs

> **docker ps**

- Arrêter un Conteneur

> **docker stop <nom_conteneur>**

- Démarrer tous les Conteneurs

> **docker start php-apache myadmin mysql-lamp**

- Redémarrer un Conteneur

> **docker restart** <nom_conteneur>

- Supprimer un Conteneur

> **docker rm** <nom_conteneur>

Commandes Docker Utiles

- Afficher les logs d'un conteneur pour suivre ce qui se passe dans vos conteneurs.

> **docker logs** <nom_conteneur>

- Exécuter une Commande dans un Conteneur déjà en cours d'exécution:

> **docker exec -it** <nom_conteneur> <commande>

Par exemple, pour accéder au shell bash d'un conteneur :

> **docker exec -it php-apache /bin/bash**

- Voir les Ressources Utilisées par les Conteneurs

Pour surveiller l'utilisation des ressources par vos conteneurs, comme la CPU, la mémoire, etc. :

> **docker stats**

- Réseau Docker : Inspecter et Dépanner

Lister les réseaux : **docker network ls**

Inspecter un réseau pour voir quels conteneurs y sont connectés et leurs adresses IP :

> **docker network inspect <nom_du_réseau>**

- **Nettoyer les Ressources Inutilisées**

Au fil du temps, Docker peut accumuler des images, des conteneurs, des volumes, et des réseaux inutilisés qui prennent de l'espace disque. Pour nettoyer :

> **docker system prune**

Pour une suppression plus spécifique, comme uniquement les conteneurs inutilisés :

> **docker container prune**

Ou seulement les images inutilisées :

> **docker image prune**

- **Vérification**

> Lister les conteneurs: **docker ps -a**

> Lister les réseaux : **docker network ls**

> Lister les images: **docker images**

Interagir avec la base de données MySQL

Grâce aux recherches approfondies, à l'analyse minutieuse, aux projections réalistes et aux plans rigoureux exposés dans ce document, nous sommes confiants dans notre stratégie et impatients d'atteindre nos objectifs !

> **docker exec -it mysql-lamp mysql -uroot -p**

Exemples de Commandes MySQL

Voici quelques commandes MySQL de base une fois connecté :

- **Voir toutes les bases de données :**

- sqlCopy code
- `SHOW DATABASES;`
- Créer une nouvelle base de données :
- sqlCopy code
- `CREATE DATABASE ma_base_de_donnees;`
- Utiliser une base de données spécifique :
- sqlCopy code
- `USE ma_base_de_donnees;`
- Créer une nouvelle table :
- sqlCopy code
- `CREATE TABLE ma_table (id INT AUTO_INCREMENT PRIMARY KEY, nom VARCHAR(255), valeur INT);`
- Insérer des données dans une table :
- sqlCopy code
- `INSERT INTO ma_table (nom, valeur) VALUES ('exemple', 123);`
- Lire des données :
- sqlCopy code
- `SELECT * FROM ma_table;`