Rapport de Projet

Titre : Déploiement d'un environnement de conteneurs avec Docker et Docke r Compose
Groupe : 35
Cours : Système d'exploitation
Professeur : Dr Masengedia Motumbe Pierre
Année académique : 2025-2026

a) Résumé

Ce projet présente la mise en place d'un environnement applicatif basé sur la c onteneurisation à l'aide de **Docker** et **Docker Compose**.
L'objectif principal est de déployer une application multi-services de manière si mple, portable et reproductible.
Grâce aux conteneurs, chaque service fonctionne dans un environnement isol é, ce qui réduit les conflits de dépendances et facilite la maintenance.

L'architecture conçue est composée de plusieurs services, notamment une appl ication web et une base de données.
Docker Compose a été utilisé pour orchestrer ces services, gérer leur communic ation et automatiser leur démarrage.

Ce travail démontre que la conteneurisation constitue aujourd'hui une solution essentielle pour le déploiement moderne des applications, aussi bien en enviro nnement de développement que de production.

b) Abstract

This project focuses on deploying a containerized environment using Docker a nd Docker Compose. The main objective is to build a multi-service application that is portable, reliable, and easy to manage.
Containerization allows each service to run in an isolated environment, reducin g dependency conflicts and improving system stability.

The system architecture includes a web application service and a database ser vice. Docker Compose was used to configure, manage, and run these services t ogether while ensuring proper communication between them.

This project highlights the importance of container technologies as a key solut ion for modern software deployment and infrastructure management.

c) Introduction

Avec l'évolution rapide des technologies informatiques, le déploiement des app

lications est devenu un défi majeur. Les différences entre les environnements d e développement, de test et de production provoquent souvent des erreurs d'ins tallation et des incompatibilités logicielles.

Docker est une technologie de conteneurisation qui permet d'exécuter une appl ication avec toutes ses dépendances dans un conteneur léger et portable. Docker Compose, quant à lui, permet de gérer facilement des applications com posées de plusieurs services.

Dans ce projet, nous utilisons Docker et Docker Compose afin de créer un envir onnement stable, reproductible et simple à déployer.

d) Problématique et objectif

Problématique :

Les méthodes traditionnelles de déploiement des applications présentent plusi eurs difficultés : Incompatibilités entre systèmes, installation complexe des dé pendances, manque de portabilité et la difficulté de maintenance

Objectif principal :

Mettre en place un environnement applicatif conteneurisé capable de fonction ner de manière identique sur différentes machines.

Objectifs spécifiques :

- Comprendre le fonctionnement de Docker
- Créer des images Docker personnalisées
- Orchestrer plusieurs services avec Docker Compose
- Tester le bon fonctionnement de l'environnement

e) Méthodologie

La réalisation du projet s'est faite en suivant les étapes suivantes :

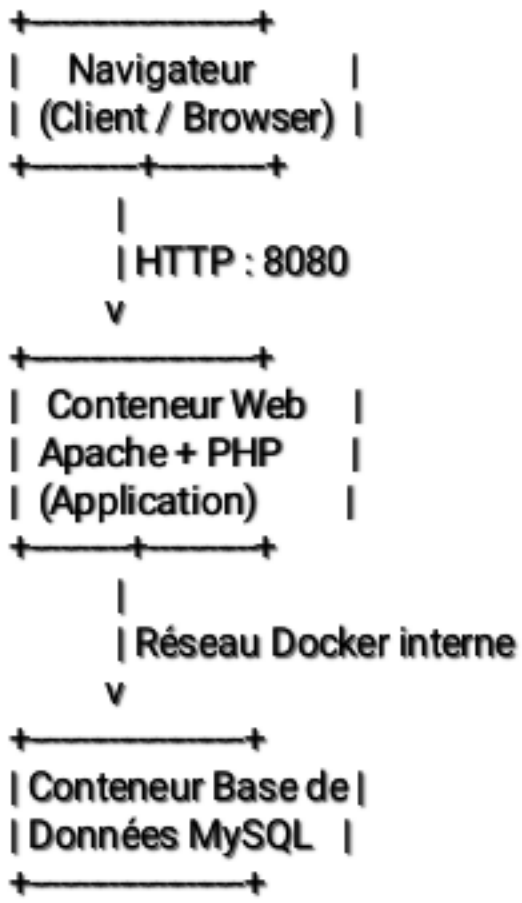
1. Analyse des besoins et identification des services nécessaires
2. Conception de l'architecture du système
3. Création des fichiers Dockerfile pour les images
4. Configuration du fichier docker-compose.yml
5. Déploiement des conteneurs
6. Tests et validation du fonctionnement global

Cette méthodologie progressive a permis une meilleure organisation du travail.

f) Conception du système

L'architecture du système repose sur une approche multi-conteneurs :
Service Application Web : Contient le serveur web et le code de l'application.

Service Base de données : Assure le stockage et la gestion des données.
Les services communiquent via un réseau Docker interne, garantissant sécurit
é et performance.

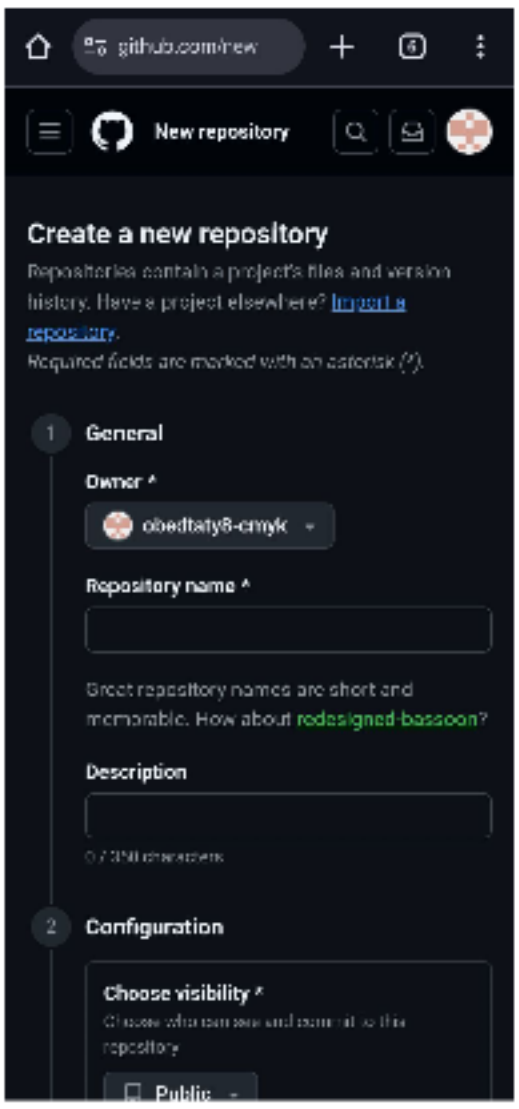


Nous avons procédés à plusieurs étapes pour réalisé ce projet;

Étape 1 : Création du repository GitHub

La première étape du projet a consisté à créer un repository GitHub afin d'hébe
rger l'ensemble des fichiers du projet et faciliter le partage avec l'enseignant.

Image :

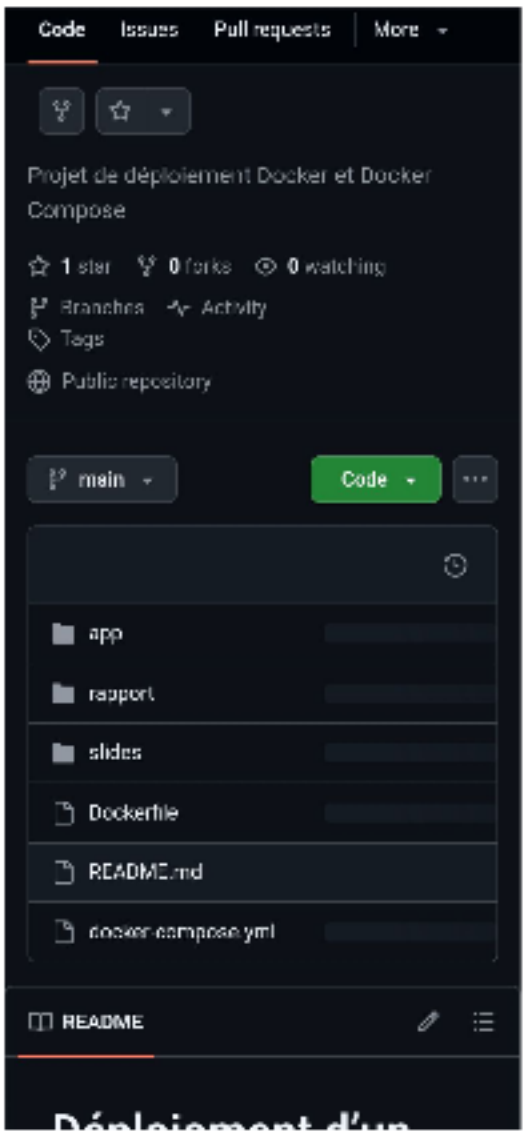


Étape 2 : Création de la structure du projet

Ensuite, nous avons créé la structure du projet contenant les dossiers et fichier

s nécessaires au déploiement de l'application Dockerisée.

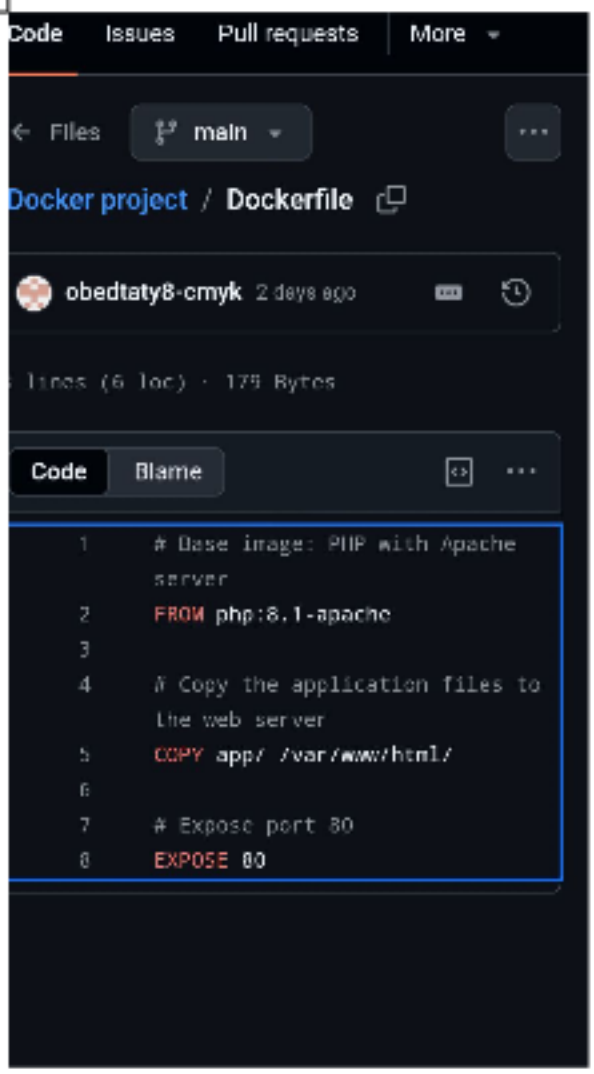
Image :



Étape 3 : Création du fichier Dockerfile

Le fichier Dockerfile permet de définir l'image Docker de l'application web. Il co ntient la configuration du serveur Apache et de PHP.

Image :



Étape 4 : Configuration de docker-compose.yml

Le fichier docker-compose.yml est utilisé pour orchestrer les différents services du projet, notamment l'application web et la base de données MySQL.

Image :

obedtaty8-cmyk avant-hier

22 lines (19 loc) · 330 Bytes

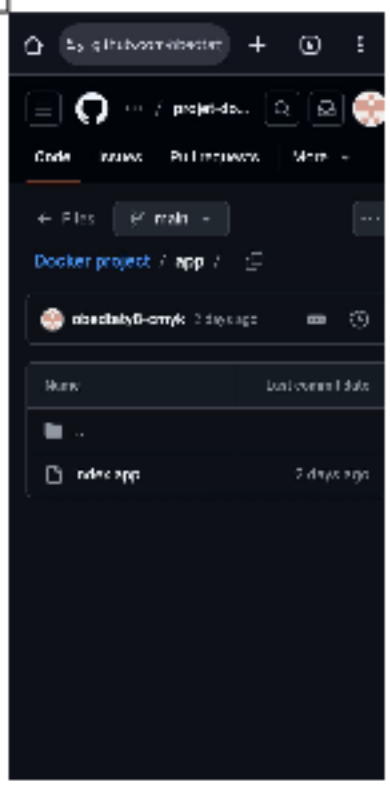
Code Blame

```yaml
version: '3.8'

services:
  web:
    build: .
    container_name: web_app
    ports:
      - "8080:80"
    depends_on:
      - db

  db:
    image: mysql:8.0
    container_name: mysql_db
    environment:
      MYSQL_ROOT_PASSWORD: root
      MYSQL_DATABASE: testdb
    volumes:
      - db_data:/var/lib/mysql

volumes:
  db_data:
```

Étape 5 : Création de l'application web (index.php)

L'application web a été créée dans le dossier app. Le fichier index.php permet d e vérifier le bon fonctionnement du conteneur web.

Image :

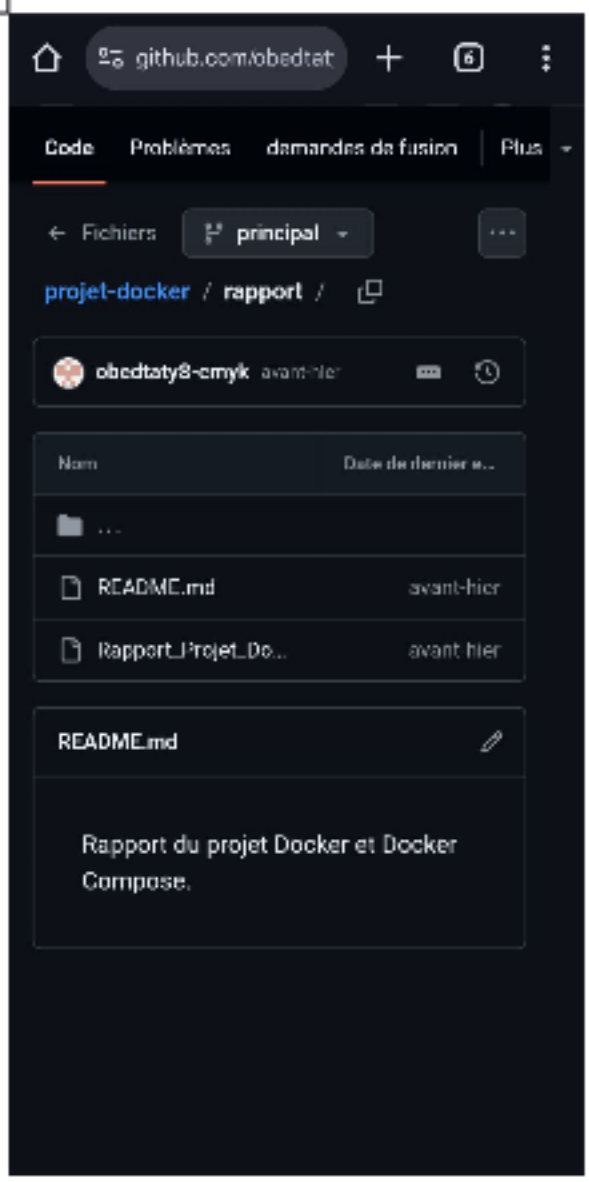


Étape 6: Accès à l'application dans le navigateur

Après le lancement des conteneurs, l'application est accessible via un navigate ur web à l'adresse suivante: https://github.com/obedtaty8-cmyk/projet-docker /tree/main/app

Étape 7 : Ajout du rapport et des diapositives sur GitHub

Enfin, le rapport PDF et les diapositives ont été ajoutés dans le repository GitH ub afin de permettre leur consultation directe



g) Implémentation ou configuration

Dockerfile : Crée une image personnalisée contenant le système de base, les d épendances nécessaires et le code source de l'application.

docker-compose.yml : Définit les services à lancer, les ports exposés, les volu mes pour la persistance des données et les variables d'environnement.

Commandes utilisées :

```
docker build -t monapp .
docker-compose up -d
docker ps
```

h) Test et Résultats

Après le déploiement : L'application était accessible via le navigateur, les conte neurs démarraient correctement, les données étaient conservées grâce aux vol umes et les services redémarraient sans perte de configuration

Les résultats confirment que l'environnement est fonctionnel et stable.

i) Analyse Critique

Points forts : Portabilité élevée, installation rapide, isolation des services et fac ilité de maintenance

Limites : Consommation de ressources sur machines peu puissantes, complex ité initiale pour les débutants et configuration avancée requise pour la producti on à grande échelle

j) Conclusion

Ce projet a permis de comprendre l'importance de la conteneurisation dans le déploiement des applications modernes.
Docker et Docker Compose offrent une solution efficace pour gérer des environ nements complexes de manière simple et fiable.

La conteneurisation améliore la portabilité, réduit les erreurs et facilite le travail collaboratif. Elle représente une compétence essentielle pour les informaticien s d'aujourd'hui.

k) Bibliographie

Documentation officielle Docker, https://docs.docker.com

Documentation Docker Compose, https://docs.docker.com/compose

Nigel Poulton, Docker Deep Dive

Articles techniques sur la conteneurisation (Medium, Dev.to)

Tutoriels vidéo sur Docker et DevOps