

Notebook

June 14, 2025

1 Documentation du Projet WordPress Docker

1.1 Introduction

Ce document présente en détail la configuration et le déploiement d'un environnement WordPress utilisant Docker. Le projet permet de mettre en place rapidement une instance WordPress avec une base de données MySQL et un outil d'administration phpMyAdmin, le tout orchestré via Docker Compose.

1.1.1 Objectifs du Projet

- Création d'un environnement de développement WordPress isolé
- Utilisation de conteneurs Docker pour une portabilité maximale
- Configuration automatisée de la base de données MySQL
- Interface d'administration via phpMyAdmin
- Facilité de sauvegarde et restauration

1.2 Structure du Projet

```
wordpress-docker-new/  
  docker-compose.yml      # Configuration des services Docker  
  .env                    # Variables d'environnement  
  .gitignore              # Configuration Git  
  database-dumps/         # Dossier pour les sauvegardes de BDD  
  mysql-init/             # Scripts d'initialisation MySQL  
  wordpress_files/        # Fichiers WordPress  
  wp-content/             # Thèmes et plugins WordPress  
  scripts/  
    export-database.sh     # Script de sauvegarde BDD  
    export-existing-site.sh # Script d'export du site  
    fix-permissions.sh     # Script de correction des permissions  
    restore-backup.sh      # Script de restauration
```

1.3 Configuration des Services

1.3.1 Service MySQL (Base de données)

```
db:  
  image: mysql:8.0  
  container_name: wordpress_db
```

```
environment:
  MYSQL_DATABASE: wordpress
  MYSQL_USER: wordpress
  MYSQL_PASSWORD: wordpress_password
  MYSQL_ROOT_PASSWORD: root_password
```

Cette configuration: - Utilise MySQL 8.0 - Crée une base de données 'wordpress' - Configure les accès utilisateur - Inclut un healthcheck pour assurer la disponibilité

1.3.2 Service WordPress

```
wordpress:
  image: wordpress:latest
  container_name: wordpress_site
  ports:
    - "8080:80"
  environment:
    WORDPRESS_DB_HOST: db:3306
    WORDPRESS_DB_USER: wordpress
    WORDPRESS_DB_PASSWORD: wordpress_password
```

Points clés: - Utilise l'image officielle WordPress - Port 8080 pour l'accès web - Configuration automatique de la connexion à la BDD

1.3.3 Service phpMyAdmin

```
phpmyadmin:
  image: phpmyadmin:latest
  container_name: wordpress_phpmyadmin
  ports:
    - "8081:80"
```

- Interface d'administration de la BDD
- Accessible sur le port 8081

1.4 Réseaux et Volumes

1.4.1 Réseaux Docker

```
networks:
  wordpress_network:
    driver: bridge
```

Le réseau `wordpress_network` permet: - L'isolation des services - La communication sécurisée entre conteneurs - La résolution DNS automatique

1.4.2 Volumes Docker

```
volumes:
  wordpress_data:
  db_data:
```

Les volumes persistent: - Les données WordPress - Les données MySQL - Les configurations personnalisées

1.5 Démarrage et Gestion

1.5.1 Commandes Principales

1. Démarrage des services:

```
docker-compose up -d
```

2. Vérification de l'état:

```
docker-compose ps
```

3. Consultation des logs:

```
docker-compose logs
```

4. Arrêt des services:

```
docker-compose down
```

1.6 Accès aux Services

1.6.1 WordPress

- URL: `http://localhost:8080`
- Configuration initiale via l'assistant d'installation
- Personnalisation via le tableau de bord WordPress

1.6.2 phpMyAdmin

- URL: `http://localhost:8081`
- Utilisateur: `root`
- Mot de passe: `root_password`

1.6.3 Informations de Connexion

- Base de données: `wordpress`
- Utilisateur BDD: `wordpress`
- Mot de passe BDD: `wordpress_password`
- Hôte BDD: `db`

1.7 Scripts Utilitaires

1.7.1 Sauvegarde de la Base de Données

```
./export-database.sh
```

- Crée une sauvegarde dans le dossier `database-dumps/`

1.7.2 Export du Site Existant

`./export-existing-site.sh`

- Sauvegarde les fichiers WordPress et la BDD

1.7.3 Correction des Permissions

`./fix-permissions.sh`

- Ajuste les droits d'accès aux fichiers

1.7.4 Restauration d'une Sauvegarde

`./restore-backup.sh`

- Restaure une sauvegarde précédente

```
[ ]: # Installation des dépendances pour la conversion en PDF
!pip install nbconvert
!pip install weasyprint

# Conversion du notebook en PDF
from nbconvert import PDFExporter
import nbformat

# Charger le notebook
with open('documentation_projet.ipynb') as f:
    nb = nbformat.read(f, as_version=4)

# Configurer l'exporteur PDF
exporter = PDFExporter()

# Exporter en PDF
pdf_data, resources = exporter.from_notebook_node(nb)

# Sauvegarder le PDF
with open('Documentation_Projet_WordPress_Docker.pdf', 'wb') as f:
    f.write(pdf_data)
```

1.8 Conclusion

Ce projet fournit une solution complète pour le déploiement de WordPress avec Docker. Les principaux avantages sont:

- Installation rapide et automatisée
- Environnement isolé et sécurisé
- Facilité de sauvegarde et restauration
- Configuration flexible via variables d'environnement
- Outils d'administration intégrés

Pour toute question ou amélioration, n'hésitez pas à consulter la documentation officielle de Docker et WordPress.