
Déployer Laravel avec MySQL et Docker

Guide complet pour déployer une application Laravel avec MySQL en utilisant Docker, incluant Dockerfile, script d'entrée et configuration de la base de données

DevOps **15 min de lecture** **Niveau Intermédiaire**

Document généré le 25/05/2026 à 20h12 · nouv.fr/wiki/deployer-laravel-avec-mysql-et-docker

Sommaire

17 section(s) · 15 min de lecture

1. Structure du projet

2. Dockerfile

3. Script d'entrée (entrypoint.sh)

4. Déploiement de la base de données MySQL

5. Commandes de déploiement

- ↳ Construire l'image Docker
- ↳ Lancer le conteneur Laravel (lié à MySQL)
- ↳ Vérifier les logs

6. Points importants

- ↳ Permissions SSH
- ↳ Configuration de la base de données
- ↳ Variables d'environnement

7. Dépannage

- ↳ Problèmes de permissions
- ↳ Vérifier les logs Laravel
- ↳ Accéder au conteneur
- ↳ Vérifier la connexion MySQL

Ce guide vous explique comment déployer une application Laravel avec MySQL en utilisant Docker, avec un Dockerfile personnalisé et un script d'entrée.

1. Structure du projet

Votre projet doit contenir les fichiers suivants :

- `Dockerfile` - Configuration de l'image Docker
- `entrypoint.sh` - Script d'initialisation du conteneur
- `id_rsa` - Clé SSH privée pour accéder aux dépôts privés

2. Dockerfile

```

# Utiliser l'image officielle PHP 8.2 avec Apache
FROM php:8.2-apache

# Installer les extensions et outils nécessaires
RUN apt-get update && apt-get install -y \
    git \
    unzip \
    libpng-dev \
    libjpeg-dev \
    libfreetype6-dev \
    libzip-dev \
    libicu-dev \
    libonig-dev \
    openssh-client \
    default-mysql-client \
    && docker-php-ext-configure gd --with-freetype --with-jpeg \
    && docker-php-ext-install gd zip intl pdo pdo_mysql mbstring

# Installer Composer
COPY --from=composer:latest /usr/bin/composer /usr/bin/composer

# Configurer les certificats SSH
RUN mkdir -p /root/.ssh
COPY id_rsa /root/.ssh/id_rsa
RUN chmod 600 /root/.ssh/id_rsa && ssh-keyscan github.com >> /root/.ssh/known_hosts

# Définir le répertoire de travail
WORKDIR /var/www/html

# Copier le script d'entrée
COPY entrypoint.sh /usr/local/bin/
RUN chmod +x /usr/local/bin/entrypoint.sh

# Configurer Apache pour pointer vers le dossier public
RUN sed -i 's|DocumentRoot /var/www/html|DocumentRoot /var/www/html/public|g' \
    /etc/apache2/sites-available/000-default.conf
RUN echo "<Directory /var/www/html/public>\n\tOptions Indexes \
FollowSymLinks\n\tAllowOverride All\n\tRequire all granted\n</Directory>" >> \
    /etc/apache2/apache2.conf

# Activer le module rewrite d'Apache
RUN a2enmod rewrite

# Exposer le port 80
EXPOSE 80

# Commande par défaut
CMD ["entrypoint.sh"]

```

📄 Copier

3. Script d'entrée (entrypoint.sh)

```

#!/bin/sh

# Cloner le dépôt privé
git clone git@github.com:Nouvy/mpi-laravel.git .

# Ajuster les permissions du dossier public
chown -R www-data:www-data /var/www/html
chmod -R 755 /var/www/html

```

```
# Installer les dépendances Composer
composer install

# Définir les droits du dossier
chown -R www-data:www-data /var/www/html
find /var/www/html/storage -type d -exec chmod 755 {} \;

# Définir les droits des fichiers
find /var/www/html/storage -type f -exec chmod 644 {} \;

touch /var/www/html/storage/logs/laravel.log
chown www-data:www-data /var/www/html/storage/logs/laravel.log
chmod 755 /var/www/html/storage/logs/laravel.log

# Remplir le fichier .env
cat <<EOF > .env
APP_NAME=Laravel
APP_ENV=local
APP_KEY=
APP_DEBUG=true
APP_URL=http://localhost

LOG_CHANNEL=stack
LOG_DEPRECATIONS_CHANNEL=null
LOG_LEVEL=debug

DB_CONNECTION=mysql
DB_HOST=mysql
DB_PORT=3306
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=secret

BROADCAST_DRIVER=log
CACHE_DRIVER=file
QUEUE_CONNECTION=sync
SESSION_DRIVER=file
SESSION_LIFETIME=120

MEMCACHED_HOST=127.0.0.1

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_MAILER=smtp
MAIL_HOST=mailpit
MAIL_PORT=1025
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
MAIL_FROM_ADDRESS="hello@example.com"
MAIL_FROM_NAME="${APP_NAME}"

AWS_ACCESS_KEY_ID=
AWS_SECRET_ACCESS_KEY=
AWS_DEFAULT_REGION=us-east-1
AWS_BUCKET=

PUSHER_APP_ID=
PUSHER_APP_KEY=
PUSHER_APP_SECRET=
PUSHER_HOST=
PUSHER_PORT=443
PUSHER_SCHEME=https
```

```
PUSHER_APP_CLUSTER=mt1

MIX_PUSHER_APP_KEY="${PUSHER_APP_KEY}"
MIX_PUSHER_APP_CLUSTER="${PUSHER_APP_CLUSTER}"
EOF

# Générer la clé d'application Laravel
php artisan key:generate

# Création des tables
php artisan migrate

# Ajout des datas de test
php artisan db:seed

# Démarrer Apache
apache2-foreground
```

📄 Copier

4. Déploiement de la base de données MySQL

Avant de déployer l'application Laravel, vous devez d'abord lancer le conteneur MySQL :

```
docker run -d \
  --name db \
  -e MYSQL_ROOT_PASSWORD=secret \
  -p 3306:3306 \
  mysql
```

📄 Copier

Cette commande :

- Lance MySQL en arrière-plan (-d)
- Nomme le conteneur db
- Définit le mot de passe root à secret
- Expose le port 3306

5. Commandes de déploiement

Construire l'image Docker

```
docker build -t laravel-app .
```

📄 Copier

Lancer le conteneur Laravel (lié à MySQL)

```
docker run -d \  
-p 80:80 \  
--name laravel-container \  
--link db:mysql \  
laravel-app
```

📋 Copier

Cette commande :

- Lance Laravel en arrière-plan (-d)
- Expose le port 80
- Nomme le conteneur `laravel-container`
- Lie le conteneur `db` avec l'alias `mysql` (accessible depuis Laravel via `mysql`)

Vérifier les logs

```
docker logs laravel-container
```

📋 Copier

6. Points importants

Permissions SSH

- Assurez-vous que votre clé SSH (`id_rsa`) est présente dans le répertoire
- La clé doit avoir les bonnes permissions (600)
- Le dépôt GitHub doit être accessible avec cette clé

Configuration de la base de données

- Le conteneur MySQL est accessible depuis Laravel via le nom d'hôte `mysql`
- Les paramètres de connexion dans le script `entrypoint.sh` sont déjà configurés pour `mysql`
- Assurez-vous que le conteneur MySQL est démarré avant Laravel

Variables d'environnement

- Ajustez les variables dans le fichier `.env` généré selon votre environnement
- Pour la production, désactivez `APP_DEBUG` et configurez `APP_ENV=production`

7. Dépannage

Problèmes de permissions

```
docker exec -it laravel-container chown -R www-data:www-data /var/www/html
```

📋 Copier

Vérifier les logs Laravel

```
docker exec -it laravel-container tail -f /var/www/html/storage/logs/laravel.log
```

📄 Copier

Accéder au conteneur

```
docker exec -it laravel-container bash
```

📄 Copier

Vérifier la connexion MySQL

```
docker exec -it laravel-container mysql -h mysql -u root -p
```

📄 Copier