Laravel CI/CD Benchmark Project Documentation

# Table of Contents

1. Introduction  
2. Project Requirements  
3. Installation and Setup  
4. Laravel + breeze  
5. Twitter clone code  
6. CI/CD Pipeline Setup  
7. Docker Configuration  
8. Troubleshooting

# 1. Introduction

This project is designed to serve as a benchmark for understanding and implementing Continuous Integration (CI) and Continuous Deployment (CD) practices. It involves developing a Laravel-based application that functions as a simple clone of Twitter, which will help in identifying the best practices, tools, and challenges associated with CI/CD processes. The goal is to automate the entire development, testing, and deployment workflow.  
  
The project also explores Docker to ensure a consistent environment across development and production stages. An optional integration with Kubernetes is considered for managing and scaling the application, though this is not mandatory for the scope of this project.

# 2. Project Requirements

CI/CD Practices  
- Automated Builds: Every commit triggers an automated build process.  
- Automated Tests: Ensure unit and integration tests are executed automatically on every build.  
- Automated Deployment: Once the tests pass, the application is automatically deployed to a staging or production environment.  
  
Docker Usage  
- Consistent Environment: Docker will be used to containerize the application, ensuring that the development and production environments are consistent and reproducible.  
- Docker Compose: Docker Compose will manage the application’s multi-container setup.

# 3. Installation and Setup

Prerequisites  
Ensure that the following dependencies are installed:  
- PHP 8.x or higher  
- Composer  
- MySQL (or the relevant database engine)  
- Node.js (for asset compilation)  
- Docker (for containerization)  
  
  
Steps to Install the Application:  
1. Clone the repository:  
 ```bash  
 git clone https://your-repository-url  
 cd your-project-directory  
 ```  
2. Install PHP and Node.js dependencies:  
 ```bash  
 composer install  
 npm install  
 ```  
3. Set up the environment file:  
 ```bash  
 cp .env.example .env  
 ```  
4. Generate the Laravel application key:  
 ```bash  
 php artisan key:generate  
 ```  
5. Set up the database and configurations:  
- Update the `.env` file for database connection settings.

# 4. Laravel + breeze install

Requirements : Composer (for managing libraries, packages, and dependencies), PHP, XAMPP (for testing the Laravel project).

1.Command for installing a new Laravel project trough composer:

composer create-project laravel/laravel projectnaam

2. Install the Laravel Breeze package trough Composer:

composer require laravel/breeze –dev

3. php artisan breeze:install

php artisan breeze:install

# 5. Twitter clone code

Explanation of the code in the project:

First we made and migrated our tables. We made the table posts using the following command:

php artisan make:migration create\_posts\_table

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijving

Then we update our users table to make a difference between normal users and admins:

php artisan make:migration add\_usertype\_to\_users\_table --table=users

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijving

After that we made a post controller to control the posts (show, edit, store & delete posts)

php artisan make:controller PostController

Afbeelding met tekst, Lettertype, schermopname, Graphics

Automatisch gegenereerde beschrijvingAfbeelding met tekst, Lettertype, schermopname, lijn

Automatisch gegenereerde beschrijving

…..

Afbeelding met tekst, Lettertype, schermopname, ontwerp

Automatisch gegenereerde beschrijvingNext we made our posts pages. The first one in index, on this page will our posts be shown. The second one is create, here you can create new posts. The last one is edit, this page will show a the post you want to edit if you press the edit button.

Afbeelding met tekst, schermopname, Lettertype, lijn

Automatisch gegenereerde beschrijvingAfterwards we made our routes to be able to go to certain pages:

Afbeelding met tekst, schermopname, Lettertype

Automatisch gegenereerde beschrijvingThe last step was to make button(s) on the home page to go to the posts page or the dashboard:

# 6. CI/CD Pipeline Setup The CI/CD pipeline automates testing builds and deployment across different branches. Here is the current configuration of the CI/CD pipeline:

# Key Features:

#### Automated Testing: Tests are executed using PHPUnit/Pest to validate application functionality. Frontend build: Node.js dependencies are installed, and assets are built automatically. Deployment:

Currently, our project is being ran on local servers because we do not have a domain name for deployment, but it can be added when we receive a domain name.

# 7. Docker Configuration

Dockerfile  
The `Dockerfile` will be used to containerize the Laravel application:  
```Dockerfile  
FROM php:8.0-fpm  
  
WORKDIR /var/www  
  
COPY . .  
  
RUN apt-get update && apt-get install -y libpng-dev libjpeg-dev libfreetype6-dev \   
 && docker-php-ext-configure gd --with-freetype --with-jpeg \   
 && docker-php-ext-install gd pdo pdo\_mysql  
  
RUN composer install  
RUN npm install && npm run prod  
  
EXPOSE 80  
CMD ["php-fpm"]  
```  
  
### docker-compose.yml  
This file will define the services used in the project, including the Laravel app, MySQL, and other necessary containers.  
```yaml  
version: '3'  
  
services:  
 app:  
 build:  
 context: .  
 container\_name: twitter-clone-app  
 ports:  
 - "80:80"  
 networks:  
 - app-network  
 volumes:  
 - .:/var/www  
 depends\_on:  
 - mysql  
  
 mysql:  
 image: mysql:5.7  
 container\_name: twitter-clone-mysql  
 environment:  
 MYSQL\_ROOT\_PASSWORD: password  
 MYSQL\_DATABASE: twitter  
 networks:  
 - app-network  
  
networks:  
 app-network:  
 driver: bridge  
```

# 8. Troubleshooting

### Error Handling During the development and deployment of the Laravel CI/CD benchmark project, you may encounter several common errors. Below is a list of errors that can occur, their potential causes, and solutions to resolve them: 1. Port Already Allocated

**Error Message:**

Error response from daemon: Ports are not available: exposing port TCP 0.0.0.0:80 -> 0.0.0.0:0: listen tcp 0.0.0.0:80: bind: An attempt was made to access a socket in a way forbidden by its access permissions.

**Cause:** Another process is already using port 80.

**Solution:**

* Identify and stop the process using port 80 with:
* sudo lsof -i :80

sudo kill -9 <PID>

* Alternatively, modify the Docker configuration to use a different port.

### 2. Table Not Found (SQLSTATE[42S02])

**Error Message:**

SQLSTATE[42S02]: Base table or view not found: 1146 Table 'twitter.sessions' doesn't exist.

**Cause:** Missing sessions table in the database.

**Solution:** Run the following command to create the missing table:

php artisan migrate

### 3. Database Connection Error

**Error Message:**

SQLSTATE[HY000] [2002] php\_network\_getaddresses: getaddrinfo for mysql failed: Temporary failure in name resolution.

**Cause:** Incorrect DB\_HOST configuration or network issues.

**Solution:**

* Verify the DB\_HOST variable in the .env file.
* Ensure the MySQL service is running and reachable.

### 4. Non-Numeric Value Encountered

**Error Message:**

A non-numeric value encountered.

**Cause:** A numeric value is expected but not provided in configuration or code.

**Solution:** Check the .env and docker-compose.yml files for variables that should be numeric, and correct them.

### 5. Port Binding Error (3306)

**Error Message:**

Error response from daemon: driver failed programming external connectivity on endpoint [container-name]-mysql-1: Bind for 0.0.0.0:3306 failed: port is already allocated.

**Cause:** Port 3306 is already in use.

**Solution:**

* Stop the process using port 3306 or change the port mapping in docker-compose.yml.

### 6. Could Not Find Driver (Illuminate\Database\QueryException)

**Description:** System attempts to connect to PostgreSQL while MySQL is intended.

**Solution:** Ensure the .env file specifies the correct database driver:

DB\_CONNECTION=mysql

### 7. Docker Configuration Errors

#### Invalid image Key

**Description:**

services.image must be a mapping

**Solution:** Verify the image key in the docker-compose.yml file is properly configured.

#### Invalid ports Key

**Description:**

services.ports must be a mapping

**Solution:** Correct the ports key syntax in the docker-compose.yml file.

### 8. No Such File or Directory (SQLSTATE[HY000] [2002])

**Description:** Missing MySQL socket file due to incorrect database host.

**Solution:** Ensure DB\_HOST in the .env file points to the correct database host.

### 9. Connection Refused (SQLSTATE[HY000] [2002])

**Description:** MySQL server is unreachable.

**Solution:**

* Verify database credentials in .env.
* Ensure MySQL server is running.

### 10. Artisan Not Found

**Error Message:**

Could not open input file: /var/www/html/artisan

**Cause:** Missing artisan file or empty /var/www/html directory.

**Solution:** Check docker-compose.yml to ensure the Laravel application files are correctly mounted in the container.

### 11. PHP Process Fails to Start

**Error Message:**

php entered FATAL state, too many start retries too quickly

**Cause:** Missing Laravel files or invalid setup.

**Solution:** Fix the missing files issue, then restart supervisord:

supervisorctl reread

supervisorctl update

supervisorctl start php

**Note:** For further assistance, ensure that all error messages are documented and updated in this guide for continuous improvement.