P\_Todo\_App CI/CD



Lucas Simões Pólvora & Lucas Lordon

MID2/GRP2D

Venne – ETML

06.11.24 – 08.01.25

Cédric Schaffter

Table des matières

[Introduction 1](#_Toc181795429)

[Planification initiale 1](#_Toc181795430)

[Adaptation de l’environnement 1](#_Toc181795431)

[Workflow CI/CD 1](#_Toc181795432)

[Schéma du Workflow 1](#_Toc181795433)

[Rapport de déploiement 1](#_Toc181795434)

[Schémas CI/CD Github Action 1](#_Toc181795435)

[Utilisation de l’IA 1](#_Toc181795436)

# Introduction

Dans le cadre de ce rapport de projet, nous explorerons le CI/CD mis en place par notre équipe pour une application déjà faite par M. Matthieu Meylan. L’application est un site web avec login pour faire une « todo list ».

Le CI/CD est une pratique qui permet de développer tout en ayant toujours un livrable. Cette pratique est de plus en plus indispensable dans le monde professionnel afin de pouvoir automatiser toutes les tâches répétitives, comme déployer dans le serveur, faire les tests, etc….

Notre CI/CD nous permettra de déployer continuellement notre application dans le Docker Hub.

# Planification initiale

Lors de ce projet, nous avons opté pour un [GitHub Project](https://github.com/users/LucasLordon/projects/2). Dans ce GitHub Project, vous retrouverez notre Kanban avec les différentes tâches et leur statut.

# Adaptation de l’environnement

## Local

En local, les seules étapes à réaliser sont de :

* Télécharger l’application en local
* Aller sur le fichier vite.config.ts dans le frontend et changer l’adresse qui est dans le proxy en « http://127.0.0.1:3000 »
* Ouvrir 2 CMDs : un sur le dossier du frontend et l’autre sur le dossier du backend.
* Exécuter la commande « npm i » sur les CMD
* Ensuite dans le CMD du backend : npm start
* Et le CMD du frontend : npm run dev

Après avoir effectuer ces étapes, l’application devrait être opérationnelle.

# Workflow CI/CD

## Schéma du Workflow

# Rapport de déploiement

# Schémas CI/CD Github Action

# Utilisation de l’IA