

Chapitre 4 : CI/CD avec GitLab

👤 Owner	👤 louis reynouard
☰ Tags	
🕒 Created time	@February 15, 2024 4:23 PM

Théorie

Principes de CI/CD

Grands Acteurs et Logiciels pour le CI/CD

Introduction à GitLab CI/CD

Configuration d'un Pipeline CI/CD

TP - Création d'un Pipeline CI/CD dans GitLab pour l'Application

L'adoption des pratiques d'Intégration Continue (CI) et de Déploiement Continu (CD) est essentielle pour automatiser le processus de développement et de déploiement des logiciels, permettant ainsi des mises à jour rapides et fiables des applications. Ces pratiques visent à augmenter la vitesse de livraison des applications, à améliorer leur qualité et à renforcer la collaboration au sein des équipes de développement.

Ce chapitre explore les principes fondamentaux de CI/CD, présente GitLab CI/CD et guide à travers la configuration d'un pipeline CI/CD.

Théorie

Principes de CI/CD

- **Intégration Continue (CI)** : La CI est une pratique de développement logiciel où les membres de l'équipe intègrent leur travail fréquemment, généralement plusieurs fois par jour. Chaque intégration est automatiquement vérifiée par la création (build) et les tests pour détecter les erreurs le plus rapidement possible.
- **Déploiement Continu (CD)** : La CD étend la CI en automatisant la livraison de l'application à l'environnement de test ou de production après le processus de build, permettant ainsi un déploiement rapide et sûr des nouvelles versions de l'application.

Grands Acteurs et Logiciels pour le CI/CD

Plusieurs outils et plateformes offrent des solutions d'intégration et de déploiement continus. Voici quelques-uns des plus populaires :

1. **Jenkins** : C'est un outil open-source automatisant les aspects de l'intégration continue et du déploiement continu. Jenkins est hautement configurable grâce à son système de plugins, permettant l'intégration avec une large variété d'outils de développement, de test et de déploiement.
2. **GitLab CI/CD** : GitLab offre une solution CI/CD intégrée à sa plateforme de gestion de version et de code source. Elle permet d'exécuter des pipelines d'intégration et de déploiement directement à partir des dépôts GitLab.
3. **GitHub Actions** : GitHub Actions permet l'automatisation des workflows de développement directement dans les dépôts GitHub. Les actions peuvent être utilisées pour automatiser les tests, les builds, les déploiements, et plus encore.
4. **CircleCI** : CircleCI est une plateforme CI/CD en tant que service (SaaS) qui automatise le pipeline de développement pour les équipes de logiciels. Elle est connue pour sa rapidité et sa facilité d'intégration avec d'autres services.
5. **Travis CI** : Travis CI est un service d'intégration continue utilisé pour construire et tester des projets hébergés sur GitHub. Travis CI automatise le processus de test et de déploiement de votre application.
6. **Bamboo** : Développé par Atlassian, Bamboo est un serveur d'intégration continue et de déploiement continu qui offre une visibilité sur les releases de logiciels, ainsi que des fonctionnalités de déploiement automatisé.
7. **TeamCity** : Un outil d'intégration continue et de gestion de build de JetBrains. TeamCity est reconnu pour sa convivialité et sa richesse en fonctionnalités, offrant une intégration étroite avec diverses outils et environnements de développement.

Introduction à GitLab CI/CD

GitLab offre une solution CI/CD native qui est intégrée dans chaque projet GitLab. Elle utilise un fichier YAML (`.gitlab-ci.yml`) pour définir le pipeline CI/CD, permettant l'exécution de scripts dans différentes étapes et environnements. L'exécution de ce pipeline est gérée par les Runners qui sont des agents qui

exécutent les jobs de votre pipeline. Ils peuvent être installés sur différents serveurs et environnements, assurant ainsi la flexibilité et l'adaptabilité du pipeline aux besoins spécifiques du projet.

Configuration d'un Pipeline CI/CD

- **Définition du Pipeline** : Le pipeline est défini dans le fichier `.gitlab-ci.yml` à la racine de votre projet. Il spécifie les étapes du pipeline, les jobs à exécuter, les scripts, les conditions de déclenchement, et les environnements.
- **Étapes Communes d'un Pipeline** :
 - **Build** : Compilation ou construction de l'application.
 - **Test** : Exécution des tests automatisés pour s'assurer de la qualité du code.
 - **Deploy** : Déploiement de l'application dans les environnements de staging ou de production.

TP - Création d'un Pipeline CI/CD dans GitLab pour l'Application



Finissez la partie 4 du TP