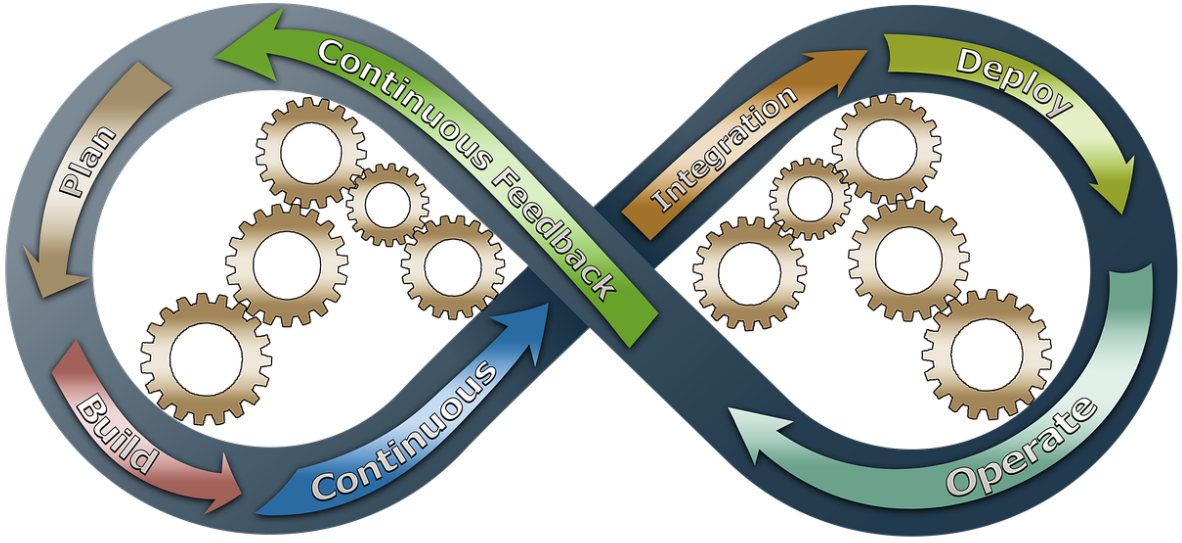

CI / CD
Introduction

Plan

Plan

CI/CD *is* Continous Integration/Delivery

CI/CD : automatisation de tâches déportées sur serveurs tiers

Objectifs

- compilation, build, release, . . . Tout ce qui peut être scripté
- effectuer les tests dans un environnement prédéfini
- évaluer la qualité du code
- mesurer la couverture des tests
- vérifier les vulnérabilités
- déployer le code validé en pré-production
- tester en pré-prod, puis déployer en production
- . . .

Plan

CI *is* Continous Integration

Exigences

- Contrôle de version pour tous les fichiers de code et de configuration ⇒ **maîtrise de Git**
- ⇒ Les *commits pushés* déclenchent les scripts (pipeline)

Avantages

- gain de temps : les tâches sont déportées
- tests automatisés : détection des problèmes au plus tôt
- ⇒ le développement reste focalisé
- facilite la revue de code en équipe

Plan

CD *is* Continous Delivery / Deployment

C **Delivery** : même principe pour la livraison / le déploiement

Exigences

- contrôle de version (encore) : tous les fichiers de configuration
- livraison automatisée et fiable, e.g. par images Docker
- un environnement intermédiaire pour tester les nouvelles versions du logiciel $\Rightarrow$ ***Staging***

Avantages

- Livraison plus rapide des nouvelles fonctionnalités et maj
- Amélioration de la fiabilité et de la qualité des versions logicielles
- Restauration plus facile des modifications de code si nécessaire
- Collaboration accrue entre les équipes de développement et d'exploitation

CD *is* Continous Delivery / Deployment

C **Deployment** : déploiement direct aux utilisateurs finaux

Exigences

- une suite de tests de haute qualité en pré-prod
- une documentation qui maintient le même rythme que la production

Avantages

- risque réduit d'erreur humaine dans le déploiement
- pas besoin d'interrompre le développement
- les versions sont plus faciles à corriger et moins risquées
- des améliorations sont apportées en continu
- augmentations de qualité clairement définies pour les clients

Plan

DevOps CI/CD pipeline : principales étapes

Source

- Un pipeline est généralement déclenché par un référentiel de code source ou par le résultat d'autres pipelines.

Build

- Construction des artefacts de l'application pour déploiement dans un environnement de production prédéfini

Test

- Validation du code, puis du comportement du logiciel en situation de production

Deploy

- Si le build et les tests sont validés : déploiement ou livraison de la version aux clients.