

Introduction à l'intégration continue

Julien Lamy

Université de Strasbourg, laboratoire ICube

Intégration continue ?

Qu'est-ce qu'on intègre ?

- Intégration : *action d'assembler des éléments divers afin d'en constituer un tout organique* (TLFI)
- Éléments divers : travaux des développeurs
- Tout organique : application fonctionnelle & stable

Intégration continue ?

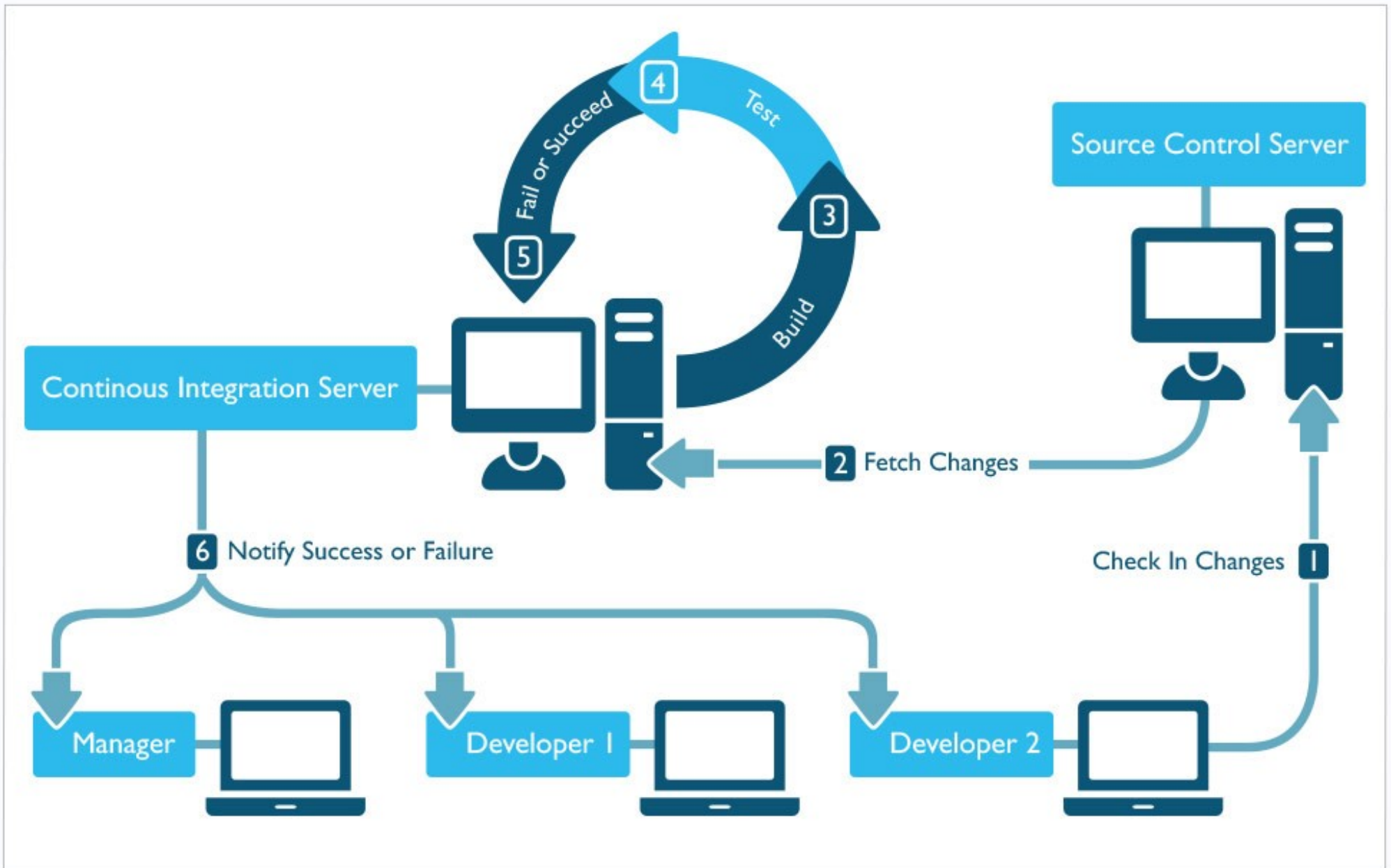
Pourquoi continue ?

- Par opposition à « juste avant la sortie d'une version stable » ou « une fois tous les mois »
- Réalisation régulière du travail d'intégration
 - Sur un intervalle court
e.g. toutes les heures, tous les jours
 - Événementielle : e.g. à chaque commit ou fusion

Intégration continue ?

Pour résumer :
automatisation du processus de développement

- Construction
- Tests
- Packaging
- Déploiement



Avantages

- Détection précoce des bugs et des régressions
- Tests dans des environnements multiples
(32 bits / 64 bits, Linux / Windows / macOS, etc.)
- Tests dans des environnements propres
(notamment grâce aux VM et conteneurs)
- Accès facile aux dernières versions stables
- Documentation du processus de construction

Ingrédients

- Des développeurs et des applications
- Un serveur de contrôle de sources
 - De CVS à Git, en fonction des usages locaux
 - Pratique de cette journée : Git
- Un serveur d'intégration continue

Ingrédients

- Des développeurs et des applications
- Un serveur de contrôle de sources
- Un serveur d'intégration continue
 - Dans le *cloud* : facile pour commencer, mais fonctionnalités limitées ou payantes, problèmes de confidentialité
 - Appveyor, CircleCI, Gitlab-CI, Travis, etc.

Ingrédients

- Des développeurs et des applications
- Un serveur de contrôle de sources
- Un serveur d'intégration continue
 - Auto-hébergé : nécessite des ressources propres (temps et serveurs)
 - Buildbot, *Gitlab-CI*, *Jenkins*