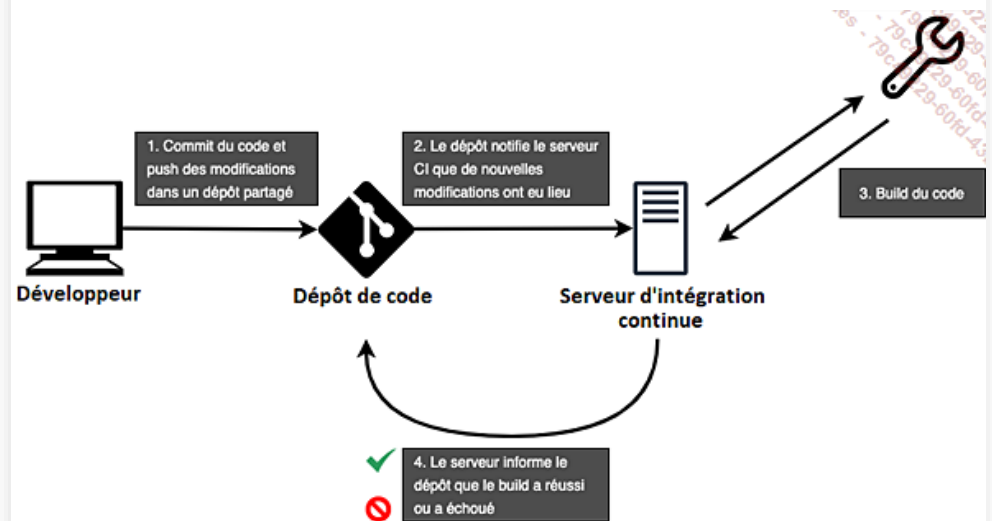


# Intégration continue (CI)

- Pratiques de développement avec dépôt partagé
- Automatisation des builds et tests

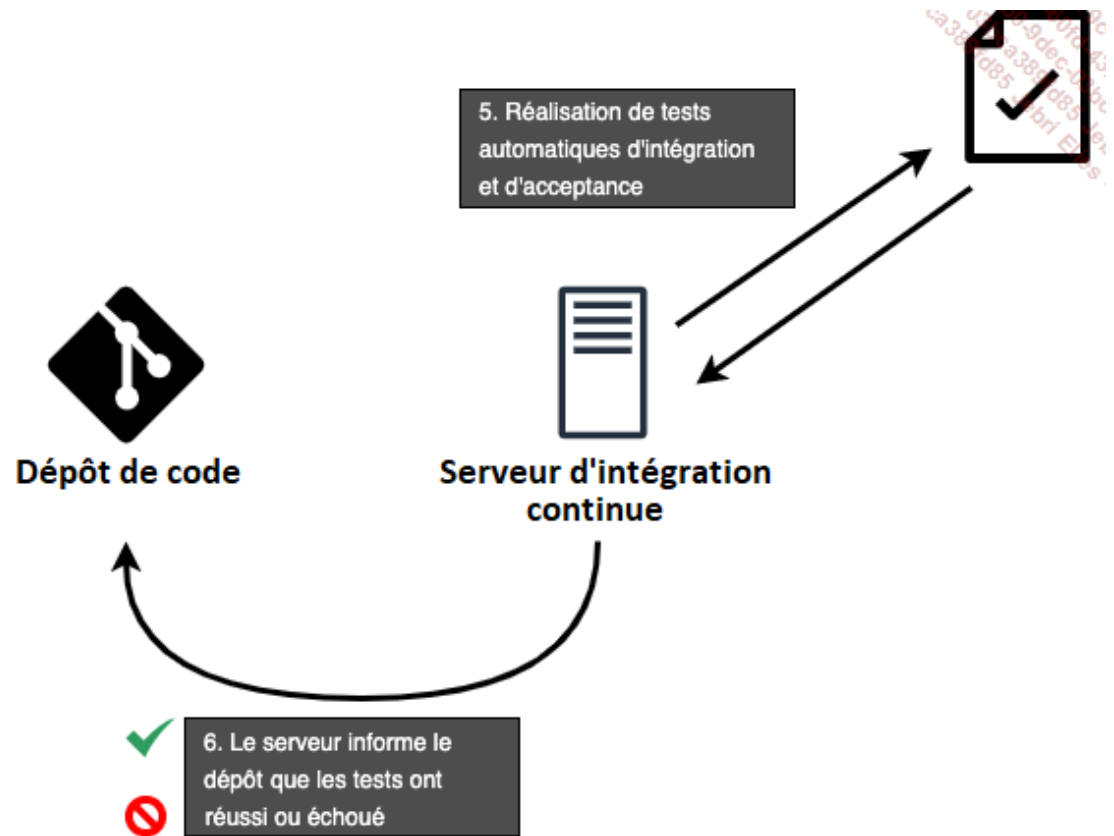
# Phase de Build

- Commit et push du code
- Notification au serveur CI
- Compilation et génération de binaires



# Phases de Tests

- **Tests fonctionnels :**  
unitaires, API, intégration, système
- **Tests non fonctionnels :**  
performance, sécurité, compatibilité



# Prérequis CI : Dépôt de code

- Utilisation de GitHub/GitLab
- Collaboration via versioning
- Documentation officielle pour installation

# Prérequis CI : Standard de code

## Qu'est-ce qu'un standard de code ?

- Ensemble de règles et conventions que les développeurs s'engagent à respecter lors de l'écriture du code.
- Ces règles portent sur :
  - **Style et lisibilité** : indentation, longueur des lignes, nommage des variables/fonctions/classes.
  - **Structure** : organisation des fichiers, hiérarchie des modules, commentaires.
  - **Pratiques de sécurité** : éviter les mots de passe en clair, respecter les guidelines de sécurité du langage.
  - **Compatibilité** : alignement avec les versions de langage/framework supportées.

**L'objectif** : avoir un code homogène, compréhensible, maintenable et sécurisé, quel que soit le développeur.

# Prérequis CI : Standard de code

## Pourquoi est-ce important ?

- Lisibilité et collaboration
  - Tous les développeurs peuvent comprendre le code écrit par un collègue.
  - Réduit le « bus factor » : si un dev part, le projet continue sans blocage.
- Réduction des bugs et des vulnérabilités
  - Des règles comme « pas de variables globales », « éviter les eval() », réduisent les risques de sécurité.
- Automatisation DevSecOps
  - Intégrer le respect du standard de code dans la CI/CD permet de détecter automatiquement les écarts.
- Formation et onboarding
  - Les nouveaux arrivants montent plus vite en compétence si les règles sont claires et suivies.

# Prérequis CI : Standard de code

## Exemples concrets de standards

- Python :
  - PEP8 (indentation, nommage, longueur de lignes  $\leq 79$  caractères).
  - Outils : Flake8, Black (formateur automatique).
- JavaScript / TypeScript :
  - ESLint, Prettier.
- Java :
  - Google Java Style Guide, Checkstyle.
- C#/.NET :
  - StyleCop, conventions Microsoft.

Automatiser la vérification via CI

# Prérequis CI : Serveur d'intégration continue

- Outils : GitLab CI, Jenkins, Travis-CI
- Installer GitLab Runner (Docker/Linux)