Les étapes clés de l'intégration continue

1

Codage

Les développeurs travaillent sur leurs branches de fonctionnalités respectives et commettent régulièrement leurs changements dans le système de contrôle de version.

2

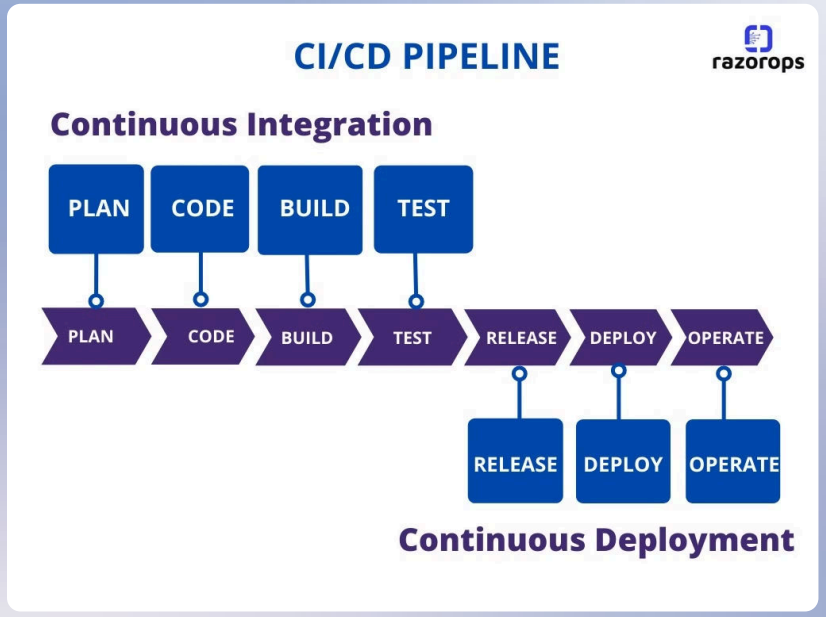
Construction

Le système d'intégration continue compile automatiquement le code à chaque commit, vérifiant ainsi que le logiciel peut être correctement construit.

3

Test

Une suite de tests automatisés est exécutée sur le code compilé, s'assurant que les nouvelles modifications n'ont pas introduit de régressions.



Made with Gamma

Jenkins: L'outil d'intégration continue de référence

Open Source et Extensible

Jenkins est une application open source qui peut être facilement étendue avec une vaste gamme de plugins. Cela permet de l'adapter à des besoins spécifiques et d'automatiser des tâches complexes.

Exécution Distribuée

Jenkins prend en charge une architecture maître-esclave, permettant d'exécuter les tâches de construction et de test sur différentes machines en fonction des besoins.

Pipelines as Code

Jenkins permet de définir les pipelines d'intégration continue sous forme de code, facilitant leur gestion, leur versionnage et leur partage entre équipes.

Made with Gamma

L'analyse de code: un élément clé de l'intégration continue

1 Détection des Vulnérabilités

L'analyse statique du code permet d'identifier les failles de sécurité potentielles avant le déploiement, réduisant les risques d'attaques.

2 Vérification des Bonnes Pratiques

Les outils d'analyse de code vérifient que le code source respecte les conventions et les meilleures pratiques de développement, améliorant ainsi la maintenabilité.

3 Couverture des Tests

L'analyse de la couverture des tests unitaires aide à s'assurer que le code est suffisamment testé avant le déploiement.

4 Détection des Anomalies

Les outils d'analyse de code peuvent détecter les erreurs fonctionnelles et les bogues avant qu'ils ne se manifestent dans l'environnement de production.

Made with Gamma

Gérer les dépendances avec les dépôts logiciels

Maven

Dépôt pour les dépendances Java basées sur Maven

APT

Dépôt pour les paquets Debian et Ubuntu

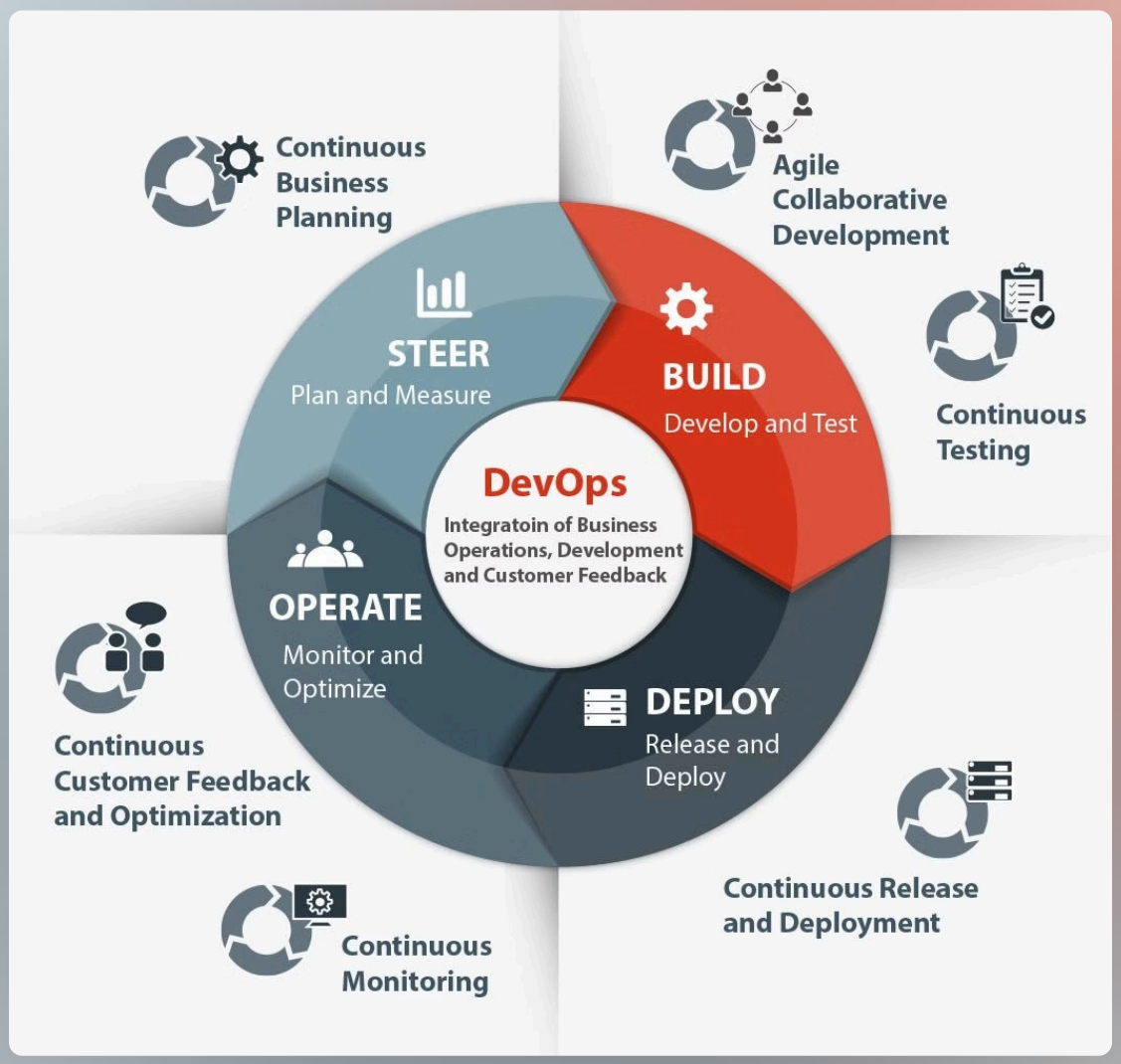
NPM

Dépôt pour les paquets JavaScript

NuGet

Dépôt pour les paquets .NET

Made with Gamma



SonarQube: Analyse de code pour détecter les vulnérabilités et les erreurs fonctionnelles

SonarQube est un outil d'analyse de code qui permet de détecter les vulnérabilités et les erreurs fonctionnelles dans votre logiciel. Il offre une analyse statique approfondie du code source, en identifiant les problèmes de sécurité, les bugs et les violations des bonnes pratiques de codage.

Les fonctionnalités de SonarQube

SonarQube propose les fonctionnalités suivantes :

- Détection des vulnérabilités de sécurité
- Identification des erreurs fonctionnelles
- Analyse de la qualité du code
- Contrôle de la couverture de code
- Évaluation de la dette technique
- Reporting détaillé sur les problèmes identifiés

Intégration de SonarQube dans le processus d'intégration continue

SonarQube peut être intégré dans votre processus d'intégration continue, permettant ainsi une analyse régulière du code et une amélioration continue de la qualité du logiciel. Il peut être configuré pour s'exécuter automatiquement à chaque validation de code, fournissant ainsi des retours instantanés aux développeurs.

Made with Gamma

Nexus: le gestionnaire de dépôts universel

Centralisé et Sécurisé

Nexus offre une solution de dépôt centralisée et sécurisée pour gérer tous les artefacts logiciels d'une organisation.

Prise en Charge Multi-Format

Nexus prend en charge une grande variété de formats de packages, tels que Maven, Docker, npm, NuGet, etc.

Intégration avec Jenkins

Nexus s'intègre parfaitement avec Jenkins pour automatiser le déploiement des artefacts produits par l'intégration continue.

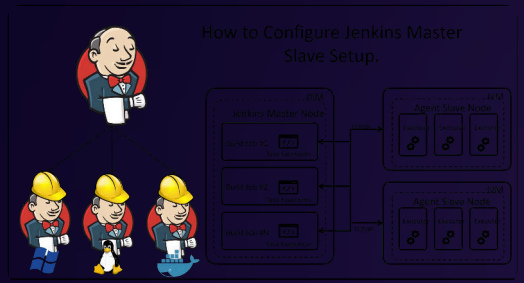
Gestion des Versions

Nexus permet de gérer et de tracer les différentes versions des artefacts logiciels, facilitant la reproductibilité des déploiements.

Made with Gamma

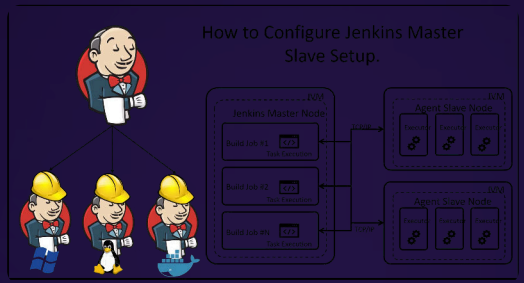
Jenkins et les nœuds esclaves: l'exécution distribuée

Maître Jenkins



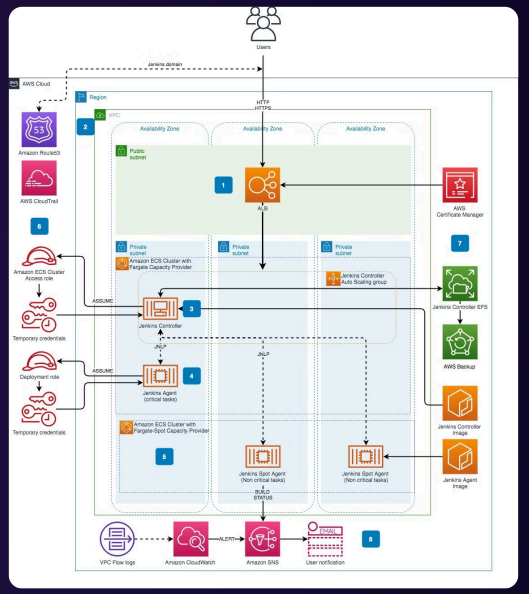
Le serveur Jenkins principal coordonne les tâches d'intégration continue et de déploiement.

Nœuds Esclaves



Les nœuds esclaves exécutent les tâches de construction, de test et de déploiement sur différentes plateformes.

Répartition de Charge



Jenkins distribue les tâches entre les nœuds esclaves, optimisant l'utilisation des ressources.

Made with Gamma

Sécurité et Contrôle d'Accès dans Jenkins

Authentification	Jenkins prend en charge plusieurs méthodes d'authentification, telles que la base de données intégrée, LDAP ou SAML.
Autorisation	Le système de gestion des autorisations de Jenkins permet de définir des rôles et des permissions granulaires pour les utilisateurs.
Sécurité des Tâches	Les autorisations sur les tâches Jenkins peuvent être configurées pour contrôler les actions autorisées (construction, configuration, suppression, etc.).
Sécurité des Informations d'Identification	Jenkins gère de manière sécurisée les informations d'identification requises pour accéder aux systèmes externes (VCS, cloud, etc.).

Made with Gamma