

Intégration continue avec Jenkins



RAZAFIMAHATRATRA Hajarisena
Docteur en Informatique

Génie Logiciel Avancé – Master I

Année: 2024

Jenkins : qu'est-ce que c'est ?

- Un outil d'intégration continue;
- Open source;
- Origine: Hudson en 2008 (2011 fork Jenkins);
- Permet de répondre au besoin:
 - ✓ Automatisation de tests;
 - ✓ Historisations des builds;
 - ✓ Accessible pour tous;
 - ✓ Contrôle avec SCM.

Fonctionnalités (1/2)

- Support d'outils de build: Suite Apache et script shell;
- Support de systèmes de gestion de versions : CVS, Subversion, ClearCase, Perforce, Starteam, Visual Source Safe, CM Synergy, Bazaar, Mercurial;
- Type de construction : manuel, programmé;
- Modèle de job : l'utilisateur peut définir ses propres constructions sur chaque projet;
- Notification de résultat de la construction;
- Construction en parallèle grâce à la gestion de plusieurs files de construction;

Fonctionnalités (2/2)

- Construction en mode distribué grâce à des agents de construction;
- Intégration avec différents outils de suivi d'incidents ou de bogues;
- Exécution des scripts à distance par SSH;
- Transfert des fichiers par SCP ou FTP;
- Transfert d'artefacts vers un repository;
- Exécution d'outils de contrôles de qualité de code.

Installation de Jenkins

- Télécharger Jenkins :
<https://jenkins.io/index.html>
- Accès à Jenkins via : `http://localhost:8080/`

Configurer un job

- Free-Style;
- Maven;
- Multi-Project.;
- External Job.

Création d'un job

Nom du Projet	Reteest
Description	<div></div>
Aperçu	
<input type="checkbox"/> Supprimer les anciens builds	
<input type="checkbox"/> Batch tasks	
<input type="checkbox"/> Ce build a des paramètres	
<input type="checkbox"/> Désactiver le Build (Aucun nouveau build ne sera exécuté jusqu'à ce que le projet soit réactivé.)	
<input type="checkbox"/> Exécuter des builds simultanément si nécessaire	
JDK	<div>(Valeur par défaut)</div>
<small>Le JDK à utiliser pour ce projet</small>	
Options avancées du projet	
<div>Avancé...</div>	
Gestion de code source	
<input checked="" type="radio"/> Aucune	
<input type="radio"/> CVS	
<input type="radio"/> CVS Projectset	
<input type="radio"/> Git	
<input type="radio"/> Subversion	
Ce qui déclenche le build	
<input type="checkbox"/> Construire à la suite d'autres projets (projets en amont)	
<input type="checkbox"/> Construire périodiquement	
<input type="checkbox"/> Scrutation de l'outil de gestion de version	
Environnements de Build	
<input type="checkbox"/> Provide Configuration files	
<input type="checkbox"/> Copier des fichiers dans le workspace du job avant le build	
Build	
<div>Sauver Apply</div>	

Les plugins

- Installation automatique via Jenkins;
- Possibilité de créer ses propres plugins:
 - ✓ Récupérer les sources sur Maven;
 - ✓ Générer son plugin sous Java + Pom.xml;
 - ✓ Configurer son context.xml.
- Nombreux plugins: (Scripts, LDAP ...);
- Communauté très actives.

Historique des builds

 Historique des builds		tendance 
<input type="text" value="find"/>		
 #5	20 juil. 2016 07:23	
 #4	20 juil. 2016 07:23	
 #3	20 juil. 2016 07:22	
 #2	19 juil. 2016 15:51	
 #1	19 juil. 2016 09:35	

Points forts

- Simple de déployer et configurer;
- Milliers des plugins pour automatiser les tâches;
- Très bonne intégration avec les outils du build;
- Très bonne intégration avec les outils du test;
- Histoire du statut du job et une résumé des changements du code;
- Facile d'ajouter les nouveaux jobs et éditer les jobs existants.

Bénéfices

- Tableau de bord pour comprendre le statut du logiciel;
- Minimisation de l'effort pour mettre en place les processus de développement et les maintenir;
- Ça rendre accessible les informations du build, test, et intégration aux toutes les participants.

Résumé

- Bon candidat pour les projets : facile à déployer mais robuste et complet;
- Suffisamment flexible pour automatiser les tests unitaires, contrôles de qualité, l'intégration des composants, et la gestion des releases.