

Mise en place d'un projet d'intégration continue avec Jenkins

Installation et configuration de Jenkins

Vous pouvez installer Jenkins **en local** ou **sur un serveur distant**. Voici les étapes pour chaque méthode.

Option 1 : Installation locale

1. **Pré-requis :**
 - **Java JDK 11 ou supérieur** (vérifiez avec `java -version`).
 - **Git** (vérifiez avec `git --version`).
 - **Maven** (optionnel mais utile, vérifiez avec `mvn -version`).
2. **Installation sur Windows / macOS / Linux :**
 - Téléchargez la version LTS de Jenkins sur le site officiel : <https://www.jenkins.io/download/>.
 - Installez Jenkins.
 - Démarrez Jenkins en exécutant la commande suivante :
`java -jar jenkins.war`
 - Accédez à Jenkins via l'URL : <http://localhost:8080>.
 - Suivez les instructions de l'assistant d'installation.
3. **Déblocage de Jenkins :**
 - Récupérez le **mot de passe initial** à partir du fichier journal affiché à l'écran.
 - Collez-le dans l'interface web de Jenkins.
4. **Installation des plugins recommandés :**
 - Installez les **plugins recommandés** (Git, Maven, Pipeline, etc.).

Présentation de l'interface utilisateur Jenkins

Une fois Jenkins installé, vous accédez à l'interface utilisateur via <http://localhost:8080>. Voici les principales sections de l'interface Jenkins.

Section	Description
Dashboard	Vue globale des jobs Jenkins et de leur statut.
Manage Jenkins	Section de configuration générale de Jenkins.
New Item	Créez de nouveaux jobs (Freestyle, Pipeline, etc.).
Build Queue	Montre les builds en attente et en cours.
Build History	Liste des builds récents avec le statut (succès, échec, etc.).

Configuration de Jenkins : Avant de créer des jobs, vous devez **ajouter les outils nécessaires**.

Ajouter des plugins

1. Accédez à **Manage Jenkins > Manage Plugins**.
2. Recherchez et installez les plugins suivants :
 - **Git Plugin** : Pour interagir avec des dépôts GitHub/GitLab.

- **Maven Integration Plugin** : Pour utiliser Maven.
- **Pipeline Plugin** : Pour créer des pipelines CI/CD.
- **Email Extension Plugin** (optionnel) : Pour envoyer des notifications.

Ajouter des outils dans "Global Tool Configuration"

1. Accédez à **Manage Jenkins > Global Tool Configuration**.
2. Configurez les outils suivants :
 - **JDK** : Ajoutez le chemin d'installation de Java.
 - **Git** : Indiquez le chemin de l'exécutable `git` (il est généralement détecté automatiquement).
 - **Maven** : Ajoutez le chemin de Maven (ou installez-le via Jenkins).

Configuration de l'environnement de build

Vérifiez que les outils suivants sont configurés dans Jenkins :

- **JDK** : Requis pour exécuter Jenkins et pour compiler le projet.
- **Git** : Requis pour cloner les dépôts.
- **Maven** : Requis pour construire des projets Java (généralement utilisé avec `pom.xml`).

Ajout des outils via "Global Tool Configuration"

1. Accédez à **Manage Jenkins > Global Tool Configuration**.
2. Configurez :
 - **JDK** : Ajoutez l'emplacement de l'installation du JDK.
 - **Git** : Assurez-vous que l'exécutable `git` est bien détecté.
 - **Maven** : Ajoutez le chemin de l'exécutable Maven (ou laissez Jenkins le télécharger automatiquement).

Créer le premier job Jenkins (Créer un projet Freestyle)

1. **New Item** > Entrez le nom du job (par ex. **"test-git"**).
2. **Type de projet** : Choisissez **Freestyle project**.
3. **Configuration du job** :
 - **Source Code Management** : Choisissez **Git** et ajoutez l'URL du dépôt (par ex. <https://github.com/spring-projects/spring-petclinic>).
 - **Build Steps** :
 - **Exécuter des commandes shell** :
 - `mvn clean install`

2.6. Lier Jenkins au dépôt GitHub/GitLab

1. **Configurer l'accès GitHub** :
 - Créez un **Jeton personnel GitHub (Personal Access Token)**.
 - Dans Jenkins, accédez à **Manage Jenkins > Credentials**.
 - Ajoutez les identifiants GitHub.
2. **Configurer le projet Jenkins** :
 - Ajoutez l'URL du dépôt GitHub.
 - Ajoutez les identifiants GitHub.

2.7. Ajouter des étapes de build

Dans la section **Build Steps**, ajoutez des étapes comme :

- **Checkout** : Cloner le dépôt.
 - **Build** : Exécuter la commande `mvn clean install`.
 - **Test** : Exécuter les tests unitaires et d'intégration.
 - **Rapport** : Afficher les rapports de tests JUnit.
-

2.8. Exécuter un job simple

2.8.1. Cloner un dépôt simple

Clonons le dépôt **Spring Petclinic** :

- URL du dépôt : <https://github.com/spring-projects/spring-petclinic>
- **Configurer le job Jenkins** :
 - Allez dans la section **Source Code Management** et ajoutez l'URL du dépôt.

2.8.2. Exécuter des commandes shell

Ajoutez un **Build Step** et exécutez la commande suivante :

```
mvn clean install
```

2.9. Logs et analyse des résultats

Lorsque vous exécutez le job, Jenkins :

- **Clône le dépôt.**
- **Construit le projet** (Maven compile le code).
- **Exécute les tests** (JUnit exécute les tests unitaires).
- **Affiche les logs** de chaque étape.

Analyse des logs

- **Succès** : Si tout est OK, vous verrez une sortie "SUCCESS".
 - **Échec** : Les erreurs de compilation ou de test s'affichent.
-

Exemple complet de configuration

Étape 1 : Créez le job Jenkins

1. Nom du job : **test-git**
2. Type de projet : **Freestyle project**
3. **Source Code Management** :
 - Type : **Git**
 - URL : <https://github.com/spring-projects/spring-petclinic>

4. **Build Triggers :**
 - Cochez **Build periodically** (optionnel) ou **Poll SCM**.
 5. **Build Environment :**
 - Cochez **Delete workspace before build starts**.
 6. **Build Steps :**
 - **Exécuter des commandes shell :**
 - `mvn clean install`
-

Résumé des étapes

Étape	Action
Installation	Installer Jenkins, JDK, Git, Maven.
Configuration	Configurer JDK, Git, Maven dans "Global Tool Configuration".
Plugins	Installer les plugins Git, Maven, Pipeline.
Créer Job	Créer un job Freestyle, lier au dépôt GitHub.
Exécuter Build	Cloner, compiler (Maven), exécuter les tests.

Exercice pratique

1. Installez Jenkins.
2. Configurez **JDK, Maven et Git**.
3. Créez un job Jenkins et liez-le au dépôt **Spring Petclinic**.
4. Configurez une build avec **mvn clean install**.
5. Interprétez les logs Jenkins et identifiez les erreurs éventuelles.