# Lab 3 : Préparation à la création de projets Java

Après avoir construit un pipeline avec des travaux simulés, nous allons commencer à créer un flux de travail avec un vrai code. Et nous sélectionnerons un projet java qui sera construit avec Maven, un outil de construction basé sur Java. L'exécution de builds Java avec maven nécessiterait des préparations supplémentaires. Maven doit être installé et disponible sur les hôtes Jenkins avec le JDK. On va installer JDK et Maven à partir du tableau de bord Jenkins.

## Configurer JDK

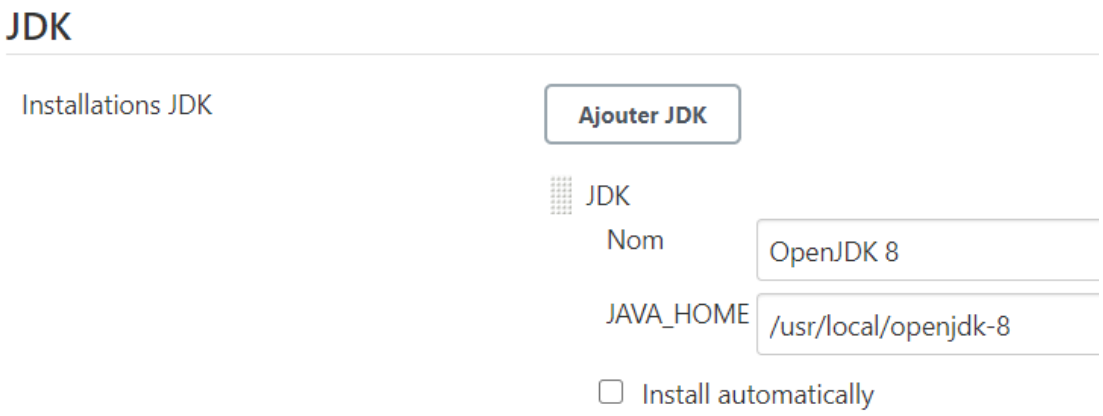
### Option 1: Configuration d'OpenJDk existant

- Dans **Manage Jenkins,** sélectionnez **Global Tool Configuration** .
- Faites défiler jusqu'à la section JDK et cliquez sur **Installations JDK** . Donnez un nom à l'instance de java, par exemple "OpenJDK 8".
- Décochez **Installer automatiquement**
- Fournissez JAVA_HOME
  par exemple
  **/usr/local/openjdk-8**

  (assurez-vous qu'il n'y a pas d'espace vide précédent dans le chemin ci-dessus, lorsque vous le collez sur jenkins)

Si vous souhaitez trouver la valeur exacte de JAVA_HOME, connectez-vous à l'hôte Jenkins et exécutez l'une des commandes suivantes

```
[root@master01 ~]# docker exec -it jenkins /bin/bash
jenkins@56472340602e:/$ echo $JAVA_HOME
/usr/local/openjdk-8
jenkins@56472340602e:/$ env | grep -i java
JAVA_HOME=/usr/local/openjdk-8
JAVA_VERSION=8u242
JAVA_BASE_URL=https://github.com/AdoptOpenJDK/openjdk8-upstream-
binaries/releases/download/jdk8u242-b08/OpenJDK8U-jdk_
JAVA_URL_VERSION=8u242b08
jenkins@56472340602e:/$ exit
```




## Option 2: installation d'Oracle Java

[**Remarque** : vous pouvez ignorer cette étape si vous avez utilisé l'**option 1** ci-dessus]

- Si vous n'avez pas de configuration JDK existante, vous pouvez l'installer à partir de Jenkins lui-même.
- Dans **Gérer Jenkins,** sélectionnez **Configurer le système** .
- Faites défiler jusqu'à la section JDK et cliquez sur **Installations JDK** . Donnez un nom à l'instance de java, par exemple "JDK 8".


- Cochez la case qui mentionne «Installer automatiquement».
- Dans «Installer depuis java.sun.com», sélectionnez la version de java que vous souhaitez installer.
- Acceptez la licence et cliquez sur l'option pour fournir les informations d'identification du compte Oracle.

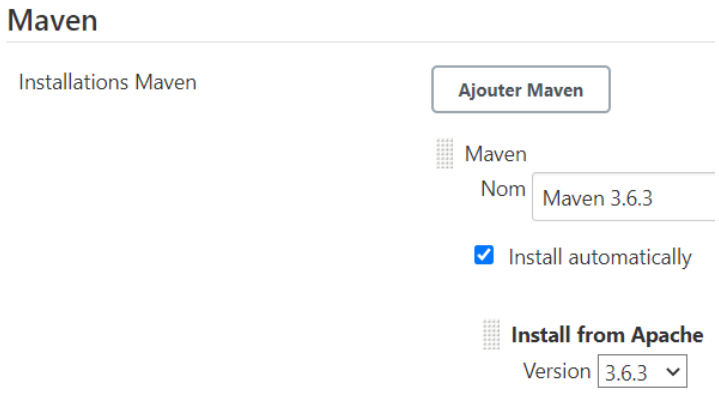
- Vous devrez créer un compte Oracle, ajouter des informations d'identification (email / pass) et faire accepter la licence pour que JDK soit installé en votre nom par Jenkins.
- Cliquez sur le bouton **Appliquer** pour enregistrer les configurations et continuer avec les configurations suivantes sur le même écran.

# Configurer Maven

Les configurations Maven sont similaires à JDK ci-dessus, c'est en fait plus simple que JDK.

Pour que Maven soit installé automatiquement,

- Cliquez sur **Installations Maven** depuis la page Configurations des systèmes Jenkins.
- Donnez un nom à l'instance de maven en cours d'installation, par exemple "Maven 3.6.3".
- Cochez la case "Installer automatiquement". Cliquez sur enregistrer.




Maven et JDK seront automatiquement installés lorsque vous créez un projet avec java build.

**Remarque:** JDK et Maven ne sont pas installés immédiatement après avoir fourni ces configurations. Jenkins appelle les procédures pour les installer lorsque vous créez un Job qui utilise JDK/Maven.

# Création d'une tâche de build pour un projet Java

Dans ce Lab, nous allons créer un job pour construire/compiler un exemple d'application Java avec maven.

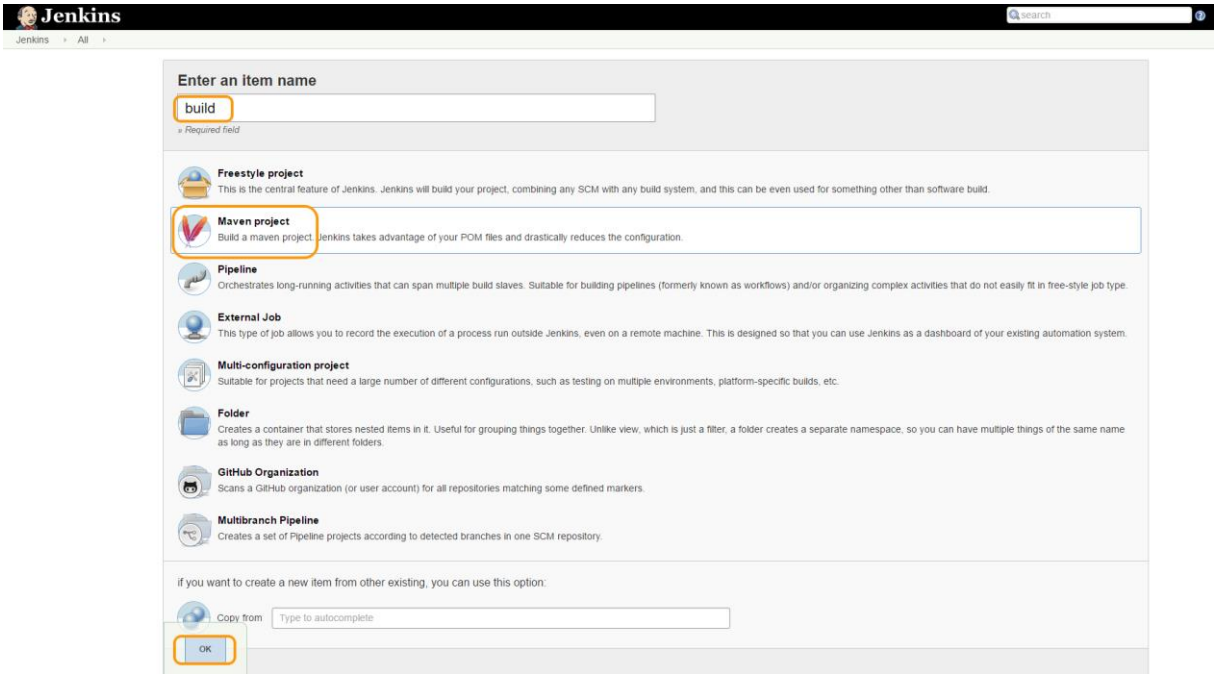
## Créer un projet Maven

Avant de commencer à créer notre tâche de build, nous devons installer le plugin **maven-integration** .


Pour créer un projet de build,

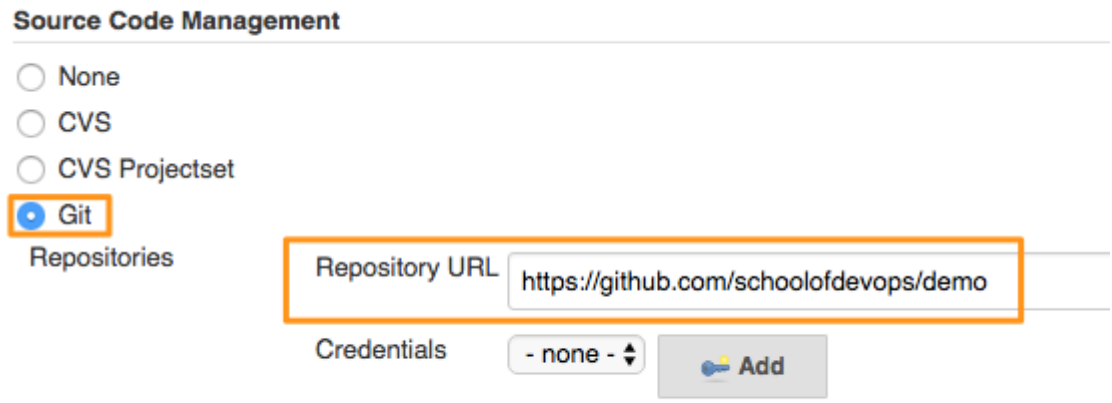
- Dans Nouvel élément, sélectionnez **Projet Maven** et donnez-lui un nom, par exemple «build».

**Remarque** : Si vous ne voyez pas l' option **Projet Maven** sur la page de création de travail, installez le **plugin d'intégration Maven à** partir du gestionnaire de plugins.

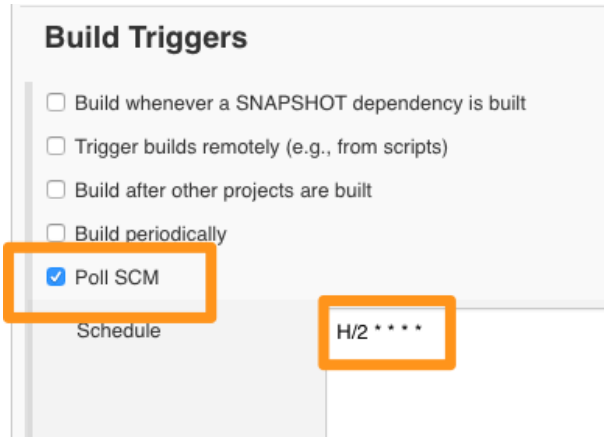


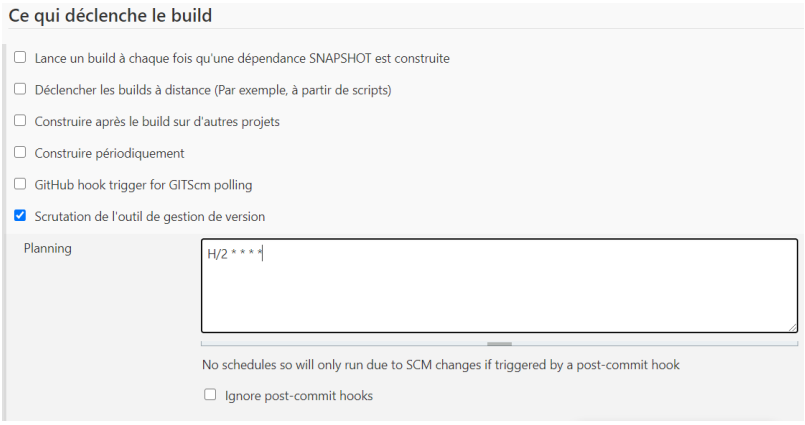

- À partir de l'écran de configuration, faites défiler jusqu'à Gestion du code source, sélectionnez GIT et fournissez l'URL du référentiel.

par exemple https://github.com/eliesjebri/demo

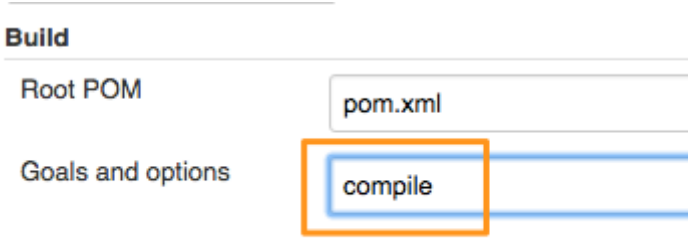



Dans **Build Triggers,** sélectionnez **Poll SCM** . Permet de le configurer pour interroger toutes les 5 minutes en utilisant le calendrier suivant `H/2 * * * *`




- Faites défiler jusqu'à l'étape Build et vous devriez voir Root POM sélectionné car c'est un projet Maven. Dans la section Objectifs et options, indiquez la **compilation** comme objectif.




En plus de la compilation, voici quelques cibles d'un projet Maven.

1. validate
2. compile - compile source code
3. test - unit tests
4. package - build jar/war
5. integration-test
6. verify
7. install
8. deploy

Enregistrez le travail et cliquez sur **Lancer un build**. Voici un extrait de la sortie de la tâche de build.

```
[INFO] Compiling 1 source file to
/var/jenkins_home/workspace/build/target/classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 10.216 s
[INFO] Finished at: 2016-04-27T17:11:30+00:00
[INFO] Final Memory: 19M/240M
[INFO] ------------------------------------------------------------------------
Waiting for Jenkins to finish collecting data
[JENKINS] Archiving /var/jenkins_home/workspace/build/pom.xml to
com.example.app/maven-app/3.0-release/maven-app-3.0-release.pom
channel stopped
Finished: SUCCESS
```