

Atelier T6.A2 : Industrialisation des tests – Intégration continue et automatisation des tests Installation des postes

Cédric Joffroy
cedric.joffroy@femto-st.fr

Fabrice Ambert
fabrice.ambert@femto-st.fr

05 Septembre 2013

1 Installation de la partie cliente

Cette section détaille l'installation de **Maven** et de **Netbeans** ainsi que la configuration de ceux-ci.

Installation de Maven

Pour installer **Maven** :

- Sur Windows, télécharger la dernière version de **Maven** à l'adresse suivante : <http://maven.apache.org/download.cgi>. Une fois l'archive *zip* téléchargée, décompressée là dans le répertoire désiré (*e.g.* `C:\apache-maven`). Il faut ensuite compléter le *Path* dans les variables d'environnement. Pour cela, il faut aller dans : *Panneau de configuration* → *Système et Sécurité* → *Système*. A ce moment là, cliquer sur *Paramètres système avancés* (cf. Figure 1). Ensuite, il faut cliquer sur le bouton *Variables d'environnement...* (cf. Figure 2). Ajouter au niveau de la variable *Path*, le chemin jusqu'au répertoire *bin* inclus dans le répertoire **Maven**.
- Sur Linux (Debian/Ubuntu), il suffit de faire :

```
sudo apt-get install maven.
```

Le système se charge ensuite de configurer tout ce qu'il faut. Dans le cas d'autres distributions, il faut télécharger **Maven** et ne pas oublier de l'ajouter ensuite dans le *Path* pour que la commande *mvn* soit accessible (encore une fois, le répertoire *bin* doit être ajouté au *path*).
- Sur Mac, **Maven** est disponible de base.

1.1 Configuration de Maven

Pour configurer **Maven** afin que celui-ci utilise **Nexus**, il faut pour cela :

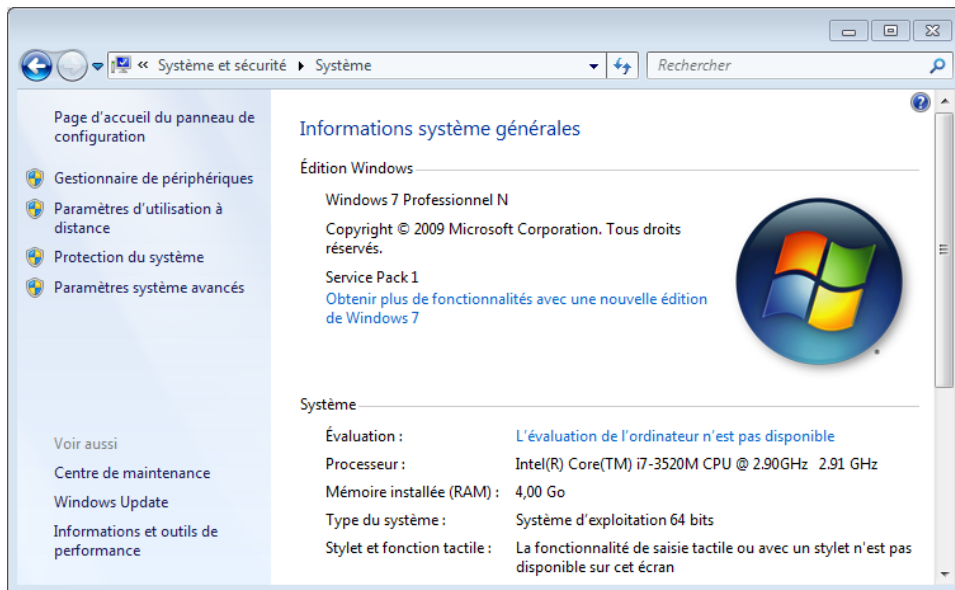


FIGURE 1 – Paramètres systèmes (Windows 7)

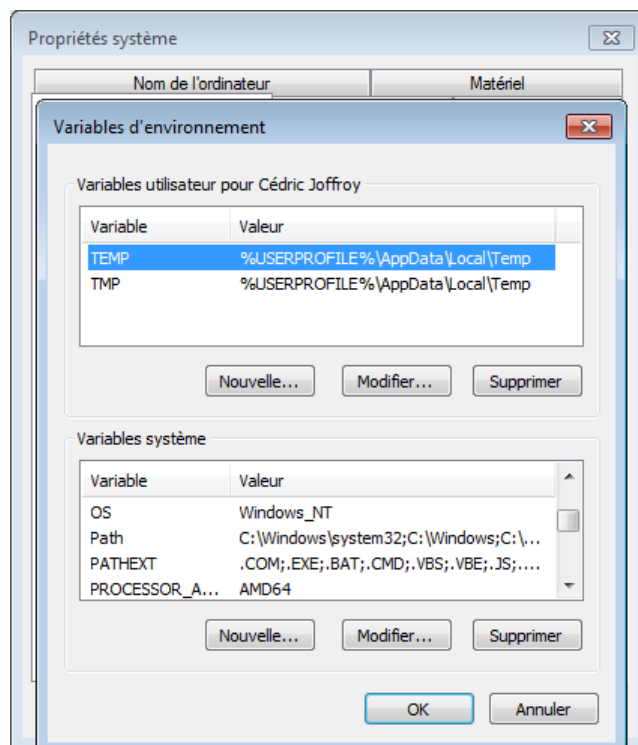


FIGURE 2 – Variables d'environnements (Windows 7)

- Soit créer un fichier `settings.xml` dans le répertoire `.m2` qui se trouve dans votre répertoire personnel;
- Soit modifier le fichier `settings.xml` si celui-ci est déjà présent.

Dans le premier cas, si vous ne possédez pas de répertoire `.m2`, il faut le créer :

- soit dans le répertoire `/home/login` dans le cas de Linux;
- soit dans le `C:\Users\Nom Complet` dans le cas de Windows;
- soit dans le répertoire `/Users/login` dans le cas de Mac.

Une fois le répertoire créé, il faut ajouter le fichier `settings.xml`. Le Listing 18 propose une configuration permettant d'utiliser le Nexus installé pour les *JDev* et permet d'accéder à la fois aux librairies qui possèdent une version *Release* mais également aux librairies qui ne possèdent qu'une version *Snapshot*.

Attention : les identifiants doit être modifié pour correspondre à vos identifiants. Ils suivent la même logique que ce qui a été décrit au début. *INC* est un nombre sur deux chiffres allant de 01 à 30. Votre identifiant vous sera attribué le jour de l'atelier.

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
    http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <servers>
    <server>
      <id>nexus-jdevt6b</id>
      <username>userINC</username>
      <password>c1l@p1INC</password>
    </server>
    <server>
      <id>nexus-jdevt6b-snapshots</id>
      <username>userINC</username>
      <password>c1l@p1INC</password>
    </server>
  </servers>
  <mirrors>
    <mirror>
      <id>nexus-jdevt6b</id>
      <mirrorOf>*</mirrorOf>
      <url>http://194.57.136.189:8083/nexus/content/groups/public/</url>
    </mirror>
  </mirrors>
  <profiles>
    <profile>
      <id>nexus-jdevt6b-snapshots</id>
      <activation<del>activeByDefault</del>>true</activation>
      <repositories>
```

```

<repository>
  <id>nexus-jdevt6b-snapshots</id>
  <url>http://194.57.136.189:8083/nexus/content/groups/public/</url>
  <releases<enabled>true</enabled></releases>
  <snapshots<enabled>true</enabled></snapshots>
</repository>
</repositories>
</profile>
</profiles>
</settings>

```

Listing 1 – Fichier settings.xml pour la connexion à Nexus

1.2 Installation et configuration de Netbeans

Dans le cadre de l’atelier, nous allons utilisé Netbeans 7.3 (ou plus). Durant l’installation, il ne faut pas oublier de cocher le fait d’installer **JUnit** (cf. Figure 3).

Une fois Netbeans installé, il faut ensuite modifié la configuration afin qu’il utilise le **Maven** que vous avez installé précédemment. Pour se faire, il faut aller dans :

- Tools → Option (sur Windows et Linux)
- Netbeans → Préférences (sur Mac)

Ensuite, il faut aller dans l’onglet *Java* puis *Maven*. Enfin, il faut modifier la version de **Maven** utilisé en spécifiant ou sélectionnant la version précédemment installée (cf. Figure 4). Une fois que c’est fait, votre Netbeans est configuré. Il ne reste plus qu’à créer un nouveau projet **Maven** afin de contrôler que tout fonctionne correctement.

2 Installation du serveur d’intégration continue

Le but est l’installation de l’ensemble de la suite d’outils pour l’intégration continue :

- Jenkins
- Nexus
- Maven
- Ant
- Sonar

L’installation a été réalisée sur une Debian (et sur une Ubuntu). La démarche présentée ci-après n’a pas été testée sur d’autres plateformes.

L’installation va être découpée en plusieurs étapes. La configuration des différents outils également

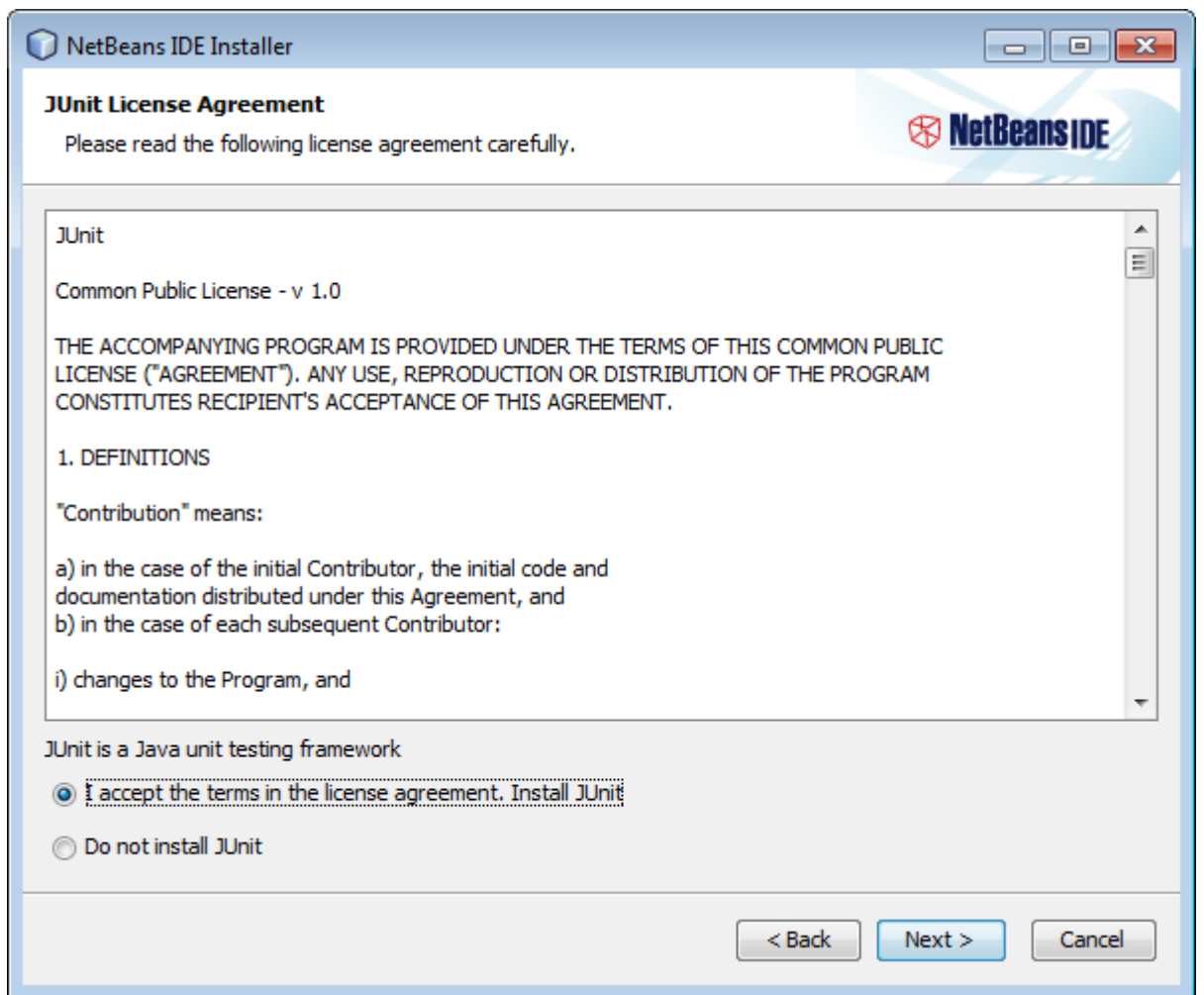


FIGURE 3 – Fenêtre de sélection de JUnit

2.1 Installations préliminaires

2.1.1 Java 6 (ou autre) JDK et JRE

Télécharger la version de java (JDK et JRE) désirée sur le site d'Oracle.

Ensuite, il suffit de le décompresser. On placera le répertoire obtenu dans /opt. Ensuite, et afin de permettre une montée de version de Java par la suite, on crée un lien symbolique sur le répertoire de la version que l'on vient d'installer :

```
cd /opt
sudo ln -s jdk1.6.0_45 current_jdk
```

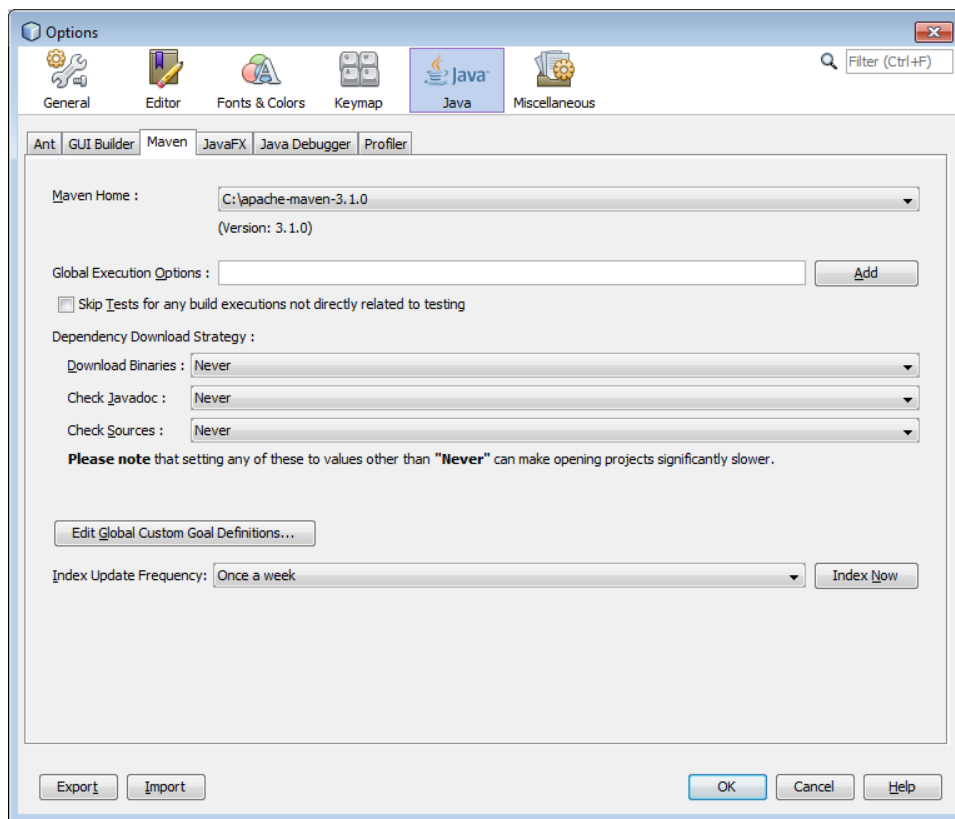


FIGURE 4 – Configuration de Maven dans Netbeans

```
sudo ln -s jre1.6.0_45 current_jre
```

Listing 2 – Installation de Java

2.1.2 Maven

Pour installer **Maven**, on s'appuie sur les paquets mis à disposition dans la distribution. Pour se faire, il suffit de taper les commandes suivantes :

```
sudo apt-get install maven
```

Listing 3 – Installation de Maven

2.1.3 Ant

Pour installer **Ant**, on s'appuie sur les paquets mis à disposition dans la distribution. Pour se faire, il suffit de taper les commandes suivantes :

```
sudo apt-get install ant
```

Listing 4 – Installation de Ant

2.1.4 Git et Subversion

Pour installer **Git** et **Subversion**, on s'appuie sur les paquets mis à disposition dans la distribution. Pour se faire, il suffit de taper les commandes suivantes :

```
sudo apt-get install git subversion
```

Listing 5 – Installation de Git et Subversion

2.2 Installation de Tomcat7

Pour installer **Tomcat 7**, on s'appuie sur les paquets mis à disposition dans la distribution. Pour se faire, il suffit de taper les commandes suivantes :

```
sudo apt-get install tomcat7 tomcat7-admin
```

Listing 6 – Installation de Maven

Il faut ensuite modifier le fichier `/etc/tomcat7/tomcat-users.xml` afin de pouvoir créer les rôles d'administrateur. Pour installer **Maven**, on s'appuie sur les paquets mis à disposition dans la distribution. Pour se faire, il suffit de taper les commandes suivantes :

```
<role rolename="admin" />
<role rolename="admin-gui" />
<role rolename="manager" />
<role rolename="manager-gui" />
<user username="tomcat" password="tomcat"
  roles="admin , admin-gui , manager , manager-gui" />
```

Listing 7 – Configuration des utilisateurs Tomcat

Enfin, il faut modifier le fichier de configuration de **Tomcat 7** afin d'ajouter certaines options Java. Le fichier à modifier est le suivant : `/etc/default/tomcat7`.

```
JAVA_OPTS="-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC
-XX:+CMSIncrementalMode -XX:+CMSClassUnloadingEnabled
-XX:+CMSPermGenSweepingEnabled -XX:MaxPermSize=128m"
```

Listing 8 – Fichier de configuration de Tomcat

2.3 Jenkins

Pour installer **Jenkins**, il est tout d'abord nécessaire de couper le serveur **Tomcat** car l'installation de **Jenkins** se fait par défaut sur le même port.

C'est pourquoi il faut exécuter la commande : `sudo /etc/init.d/tomcat7 stop`

Ensuite, il faut ajouter une nouvelle clé.

```
wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key |
sudo apt-key add -
ou
wget http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key
sudo apt-key add jenkins-ci.org.key
```

Listing 9 – Ajout de la clé de dépôt de Jenkins

Puis, il faut modifier le fichier `/etc/apt/sources.list` en ajoutant la ligne suivante :

```
# Depot Jenkins
deb http://pkg.jenkins-ci.org/debian binary/
```

Listing 10 – Modification du fichier sources.list

Après la modification du fichier, il faut mettre à jour les paquets disponibles avec la commande :

```
sudo apt-get update
```

Une fois ceci fait, il suffit d'exécuter la commande suivante :

```
sudo apt-get install jenkins
```

Une fois **Jenkins** installé, il faut stopper celui-ci pour modifier le port sur lequel il s'exécute. Pour se faire, taper la commande suivante :

```
sudo /etc/init.d/jenkins stop
```

Il faut ensuite modifier le fichier `/etc/default/jenkins`

```
#HTTP_PORT=8080
HTTP_PORT=8500
```

Listing 11 – Modification du port de Jenkins

Par défaut, le répertoire de travail de **Jenkins** est : `/var/lib/jenkins`. Ce répertoire suit la philosophie Linux sur l'emplacement des différents répertoires, c'est pourquoi nous ne modifions pas le répertoire de travail par défaut (il en est de même pour **Tomcat** : `/var/lib/tomcat7`).

Une fois ceci fait, il est possible de relancer **Jenkins** :

```
sudo /etc/init.d/jenkins start
```

2.4 Installation de Nexus

Selon la version de **Nexus** à installer il est nécessaire de mettre à jour la version de Java (en effet, la dernière version de **Nexus** : 2.6.1-02, requiert Java 7).

Pour l'installation, il suffit de télécharger directement le fichier war qui se trouve sur <http://www.sonatype.org/nexus/go>

Au préalable, il faut stopper **Tomcat** et copier le fichier `nexus.war` dans le répertoire `/var/lib/tomcat7/webapps/`

```
sudo cp nexus.war /var/lib/tomcat7/
```

Ensuite, il faut changer le propriétaire du fichier afin qu'il appartienne à **Tomcat**


```
cd /var/lib/tomcat7/
sudo chown tomcat7:tomcat7 nexus.war
```

Listing 12 – Modification du propriétaire du fichier

Il faut ensuite créer un répertoire de travail afin que Nexus puisse y déposer ses fichiers. On va donc créer un répertoire `nexus` dans le répertoire `/var/lib/` qui aura pour propriétaire `tomcat7`.

```
sudo mkdir /var/lib/nexus
sudo chown tomcat7:tomcat7 /var/lib/nexus
```

Listing 13 – Création du répertoire de travail de Nexus

Enfin, il est maintenant possible de relancer Tomcat.

A ce moment là, un répertoire Nexus va être créé, mais Nexus ne sera pas exécuté. En effet, il est nécessaire d'apporter quelques modifications au sein du fichier de configuration.

Il faut donc stopper tomcat et modifier le fichier

```
/var/lib/tomcat7/webapps/nexus/WEB-INF/plexus.properties
```

```
sudo vim /var/lib/tomcat7/webapps/nexus/WEB-INF/plexus.properties
#nexus-work=${user.home}/sonatype-work/nexus
nexus-work=/var/lib/nexus
```

Listing 14 – Modification du fichier de configuration Nexus

Maintenant que la modification a été faite, il est possible de relancer tomcat. Cette fois-ci Nexus devrait s'exécuter comme il faut.

Pour y aller, il suffit de taper dans un navigateur l'url de Nexus : `http://tomcat_host:8080/nexus`

Par défaut, pour se connecter à Nexus, le login/password administrateur est le suivant : `admin/admin123`

Maintenant, il faut ajouter un nouvel utilisateur que l'on utilisera pour faire les déploiements. Cet utilisateur est lié à Jenkins puisque c'est lui qui se chargera de faire les déploiements au sein de Nexus (cf. Figure 5).

La configuration de Nexus est terminée. Il est temps de repasser sur Jenkins pour finir sa configuration.

2.5 Configuration de Jenkins

Dans un premier temps (et puisque Jenkins est toujours stoppé), nous allons commencer par créer un répertoire `.m2` ainsi qu'un fichier `settings.xml` dans le répertoire `/var/lib/jenkins`.

```
sudo mkdir /var/lib/jenkins/.m2
sudo touch /var/lib/jenkins/.m2/settings.xml
sudo vim /var/lib/jenkins/.m2/settings.xml
```

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
```

First Name	★ Jenkins	?
Last Name	★ Jenkins	?
Email	★ cjoffro2@femto-st.fr	?
Status	★ Active	?
New Password (optional)	?
Confirm Password	?

Role Management Add Remove

- 📁 Nexus Deployment Role
- 📁 Repo: All Repositories (Full Control)

FIGURE 5 – Ajout d'un nouvel utilisateur Jenkins

```

http://maven.apache.org/xsd/settings-1.0.0.xsd">
<servers>
<server>
<id>nexus-ci</id>
<username>jenkins</username>
<password>password</password>
</server>
</servers>
<mirrors>
<mirror>
<id>nexus-ci</id>
<name>Nexus for Continuous Integration</name>
<url>http://localhost:8080/nexus/content/groups/public/</url>
<mirrorOf>*</mirrorOf>
</mirror>
</mirrors>
</settings>

```

Listing 15 – Création du fichier settings.xml pour Jenkins

Ensuite, on change le propriétaire du répertoire et du fichier nouvellement créé (dans l'exemple le propriétaire est *jenkins :nogroup*, mais il est important de vérifier quel est le propriétaire des fichiers qui se trouvent dans */var/lib/jenkins*).

```
sudo chown -R jenkins:nogroup /var/lib/jenkins/.m2
```

Il est maintenant possible de relancer Jenkins

```
sudo /etc/init.d/jenkins start
```

Il faut ensuite configurer l'emplacement d'installation des différents outils (JDK, Maven, Ant). Pour cela, il faut aller dans : *Administrer Jenkins* → *Configurer le système*.

Il faut ensuite installer certains plugins (il est d'ailleurs recommandé de mettre à jour les plugins déjà installés).

JDK

JDK installations

⌵ JDK

Nom

JAVA_HOME

Install automatically ?

Nombre d'installations JDK sur ce système

FIGURE 6 – Configuration de la JDK

Maven

Maven installations

⌵ Maven

Nom

MAVEN_HOME

Install automatically ?

Nombre d'installations Maven sur ce système

FIGURE 7 – Configuration de la Maven

Ant

Ant installations

⌵ Ant

Nom

ANT_HOME

Install automatically ?

Nombre d'installations Ant sur ce système

FIGURE 8 – Configuration de la Ant

Voici la liste des plugins que nous allons installer :

- *M2 Release Plugin*
- *Github Plugin*
- *Git Plugin*
- *Jenkins Sonar Plugin* (pour faire le lien avec Sonar que nous allons installer ensuite, la configuration de celui-ci sera expliqué dans ce qui suit)

Il faut ensuite redémarrer Jenkins pour que les modifications soient prises en compte

2.6 Installation de Sonar

Avant de pouvoir installer Sonar, il faut installer un serveur de base de données. Dans cet exemple, nous allons utiliser PostgreSQL 9.1.

Pour installer PostgreSQL, il suffit de taper la commande suivante :

```
sudo apt-get install postgresql-9.1
```

A la fin de l'installation, le serveur PostgreSQL est démarré.

Il faut ensuite se connecter avec l'utilisateur *postgres* :

```
sudo su - postgres
```

De là, on va aller sur l'éditeur *psql* afin de créer un nouvel utilisateur ainsi qu'une nouvelle base de données :

```
postgres@jdevt6b:~$ psql
postgres=# CREATE USER sonar WITH PASSWORD 'sonar';
CREATE ROLE
postgres=# SELECT username, usesysid FROM PG_USER;
username | usesysid
-----+-----
 postgres |      10
  sonar   |    16384
(2 rows)
postgres=# CREATE DATABASE sonar WITH OWNER sonar ENCODING 'UTF8';
CREATE DATABASE
postgres=# SELECT datname, datdba, encoding FROM pg_database;
 datname | datdba | encoding
-----+-----+-----
 template1 |      10 |      6
 template0 |      10 |      6
 postgres  |      10 |      6
  sonar   |    16384 |      6
(4 rows)
```

Listing 16 – Configuration de la base de données

Enfin *Ctrl+D* pour quitter (et *exit* pour quitter la session de *postgres*)

Ensuite, il faut télécharger Sonar sur le site :<http://www.sonarqube.org/downloads/>

Une fois le fichier téléchargé, il faut le décompresser. On le copie ensuite dans le répertoire */opt/*

```
sudo cp -r sonar-3.7 /opt/sonar
```

Une fois le répertoire copié, il faut modifier le fichier *sonar.properties* qui se trouve dans le répertoire *conf* (dans notre cas */opt/sonar/conf/sonar* verb+.properties+). Dans ce fichier, nous allons faire la configuration de l'accès à la base de données. De fait, comme nous avons comme login/password de sonar : **sonar/sonar**, il n'y aura que deux choses à faire : commenter l'activation par défaut et décommenter l'accès à la base de données PostgreSQL.

```

sudo vim sonar.properties
# Comment the following line to deactivate the default
#embedded database.
#sonar.jdbc.url: jdbc:h2:tcp://localhost:9092/sonar
#----- PostgreSQL 8.x/9.x
# Comment the embedded database and uncomment the
#following property to use PostgreSQL
sonar.jdbc.url: jdbc:postgresql://localhost/sonar

```

Listing 17 – Modification du fichier de configuration de Sonar

Il faut ensuite aller dans le répertoire `war (/opt/sonar/war)`, pour exécuter le fichier `build-war.sh` afin de générer le fichier `sonar.war` qui sera déployé dans Tomcat.

Avant de déployer le fichier obtenu dans Tomcat, nous allons tout d’abord stopper Tomcat

```
sudo /etc/init.d/tomcat7 stop
```

On change le propriétaire du répertoire `sonar` pour le faire appartenir à l’utilisateur `tomcat`.

```
sudo chown -R tomcat7:tomcat7 /opt/sonar
```

Puis, on copie le fichier obtenu dans le répertoire des `webapps` :

```
sudo cp sonar.war /var/lib/tomcat7/webapps
```

Attention de bien changer le propriétaire du fichier

```
sudo chown tomcat7:tomcat7 /var/lib/tomcat7/webapps/sonar.war
```

Maintenant, il est possible de relancer Tomcat.

```
sudo /etc/init.d/tomcat7 start
```

Si tout s’est bien passé, un ensemble de tables a été créé. Il est possible de vérifier cela en exécutant les commandes suivantes :

```

psql -d sonar -h localhost -U sonar
SELECT table_name FROM information_schema.tables
WHERE table_schema = 'public';

```

Listing 18 – Contrôle de l’installation de Sonar

Vous devriez voir apparaître une cinquantaine de tables.

2.7 Configuration de Jenkins (Bis)

Maintenant que `Sonar` est installé, il est nécessaire de configurer `Jenkins` afin que celui-ci connaisse l’emplacement de celui-ci.

Il faut aller dans : *Administrer Jenkins* → *Configurer le système*

Sonar

Installations de Sonar

Nom	<input type="text" value="Sonar"/>
Désactiver	<input type="checkbox"/>
URL du serveur	<input type="text" value="http://localhost:8080/sonar"/>
Login du compte Sonar	<input type="text" value="Per défaut à http://localhost:9000"/>
Mot de passe du compte Sonar	<input type="text" value="Compte Sonar utilisé pour l'analyse. Requis depuis Sonar 3.4 si l'accès anonyme est désactivé."/>
URL de la base de données	<input type="text" value="jdbc:postgresql://localhost/sonar"/>
Nom d'utilisateur BDD	<input type="text" value="Ne pas renseigner si vous utilisez la base embarquée."/>
Mot de passe BDD	<input type="text" value="Per défaut à sonar."/>
Driver BDD	<input type="text" value="Per défaut à sonar."/>
Version du sonar-maven-plugin	<input type="text" value="org.postgresql.Driver"/>
Propriétés additionnelles	<input type="text" value="Ne pas renseigner si vous utilisez la base embarquée."/>
	<input type="text" value="Si non spécifié, sonar:sonar sera utilisé."/>
	<input type="text" value="Propriétés additionnelles fournies à l'exécutable mvn (exemple : -Dsome.property=some.value)."/>
Conditions d'exécution (seulement pour les projets utilisant Sonar en action à la suite du build)	
Ignorer si déclenché par un changement dans l'outil de gestion de version	<input type="checkbox"/>
Ignorer si déclenché par un projet amont	<input type="checkbox"/>
Ignorer si une variable d'environnement est définie à true	<input type="text" value=""/>

FIGURE 9 – Configuration de Sonar dans Jenkins