

Exemple « Declarative Pipeline » : Déployer une application dans un serveur Tomcat



1. Installation du serveur Tomcat sur une autre Machine Linux

- Téléchargez les packages tomcat depuis <https://tomcat.apache.org/download-80.cgi>

```
[root@worker-node2 ~]# wget https://downloads.apache.org/tomcat/tomcat-8/v8.5.61/bin/apache-tomcat-8.5.61.tar.gz
[root@worker-node2 ~]# tar xvfz apache-tomcat-8.5.61.tar.gz -C /opt
[root@worker-node2 ~]# mv /opt/apache-tomcat-8.5.61/ /opt/tomcat8
```

- donnez les autorisations d'exécution à *startup.sh* et *shutdown.sh* qui sont sous *bin*

```
[root@worker-node2 ~]# chmod +x /opt/tomcat8/bin/startup.sh
[root@worker-node2 ~]# chmod +x /opt/tomcat8/bin/shutdown.sh
```

- Créez des liens pour tomcat *startup.sh* et *shutdown.sh*

```
[root@worker-node2 ~]# ln -s /opt/tomcat8/bin/startup.sh /usr/local/bin/tomcatup
[root@worker-node2 ~]# ln -s /opt/tomcat8/bin/shutdown.sh /usr/local/bin/tomcatdown
```

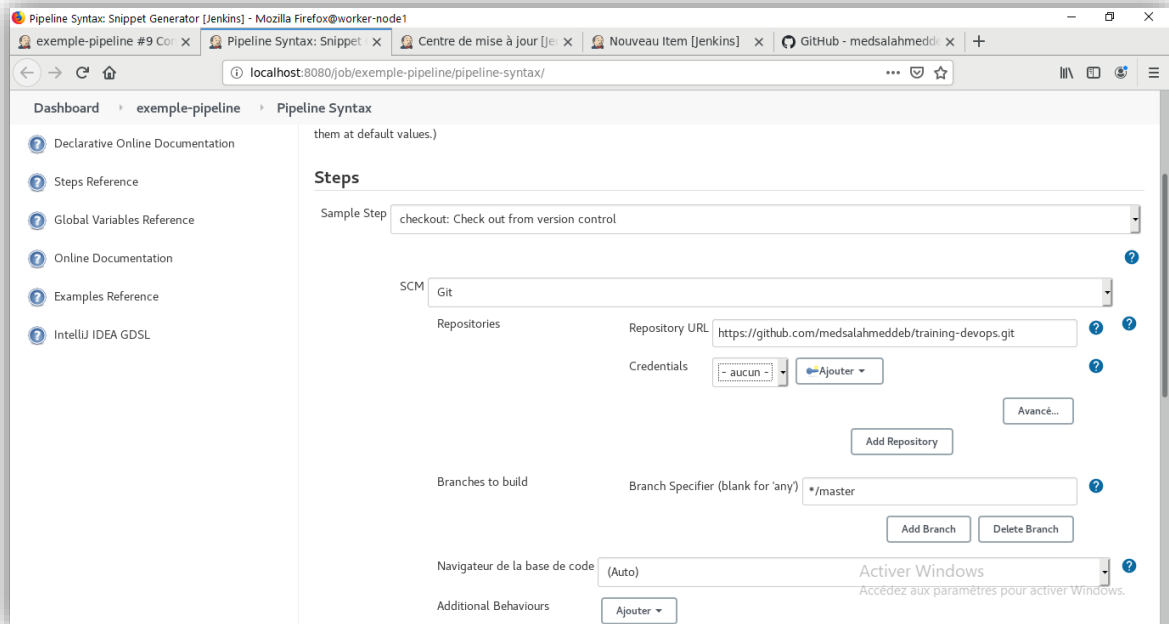
- Démarrez le serveur Tomcat

```
[root@worker-node2 ~]# tomcatup
Using CATALINA_BASE: /opt/tomcat8
Using CATALINA_HOME: /opt/tomcat8
Using CATALINA_TMPDIR: /opt/tomcat8/temp
Using JRE_HOME: /usr
Using CLASSPATH: /opt/tomcat8/bin/bootstrap.jar:/opt/tomcat8/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

2. Partie 1 : Configuration du Git et Maven

- Exécuter le code suivant pour une configuration standard

Ajout du bloc checkout, vérifiez avec l'outil syntaxe



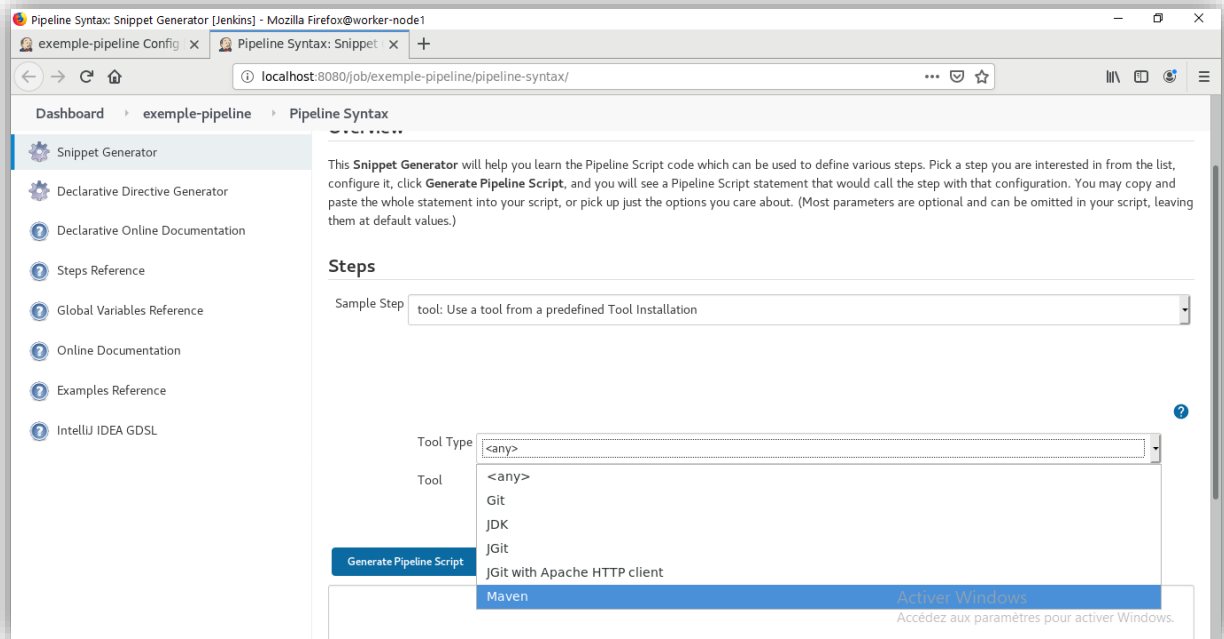
```
node{
    stage('SCM Checout'){
        git branch: 'main', url: 'https://github.com/medsalahmeddeb/web-app-maven'
    }
    stage('Build Package'){
        sh 'mvn package'
    }
}
```

⇒ L'exécution retournera une erreur avec la commande mvn

```
+ mvn package
/var/lib/jenkins/workspace/exemple-pipeline@tmp/durable-60e801f3/script.sh:
ligne1: mvn : commande introuvable
```

- Modifier le chemin de la commande mvn pour spécifier le chemin complet du binaire, accéder à la page de Syntaxe de Pipeline pour avoir de l'aide puis chercher une exemple de l'utilisation du Tools Maven

"Declarative Pipeline" : Déployer une application dans un serveur Tomcat

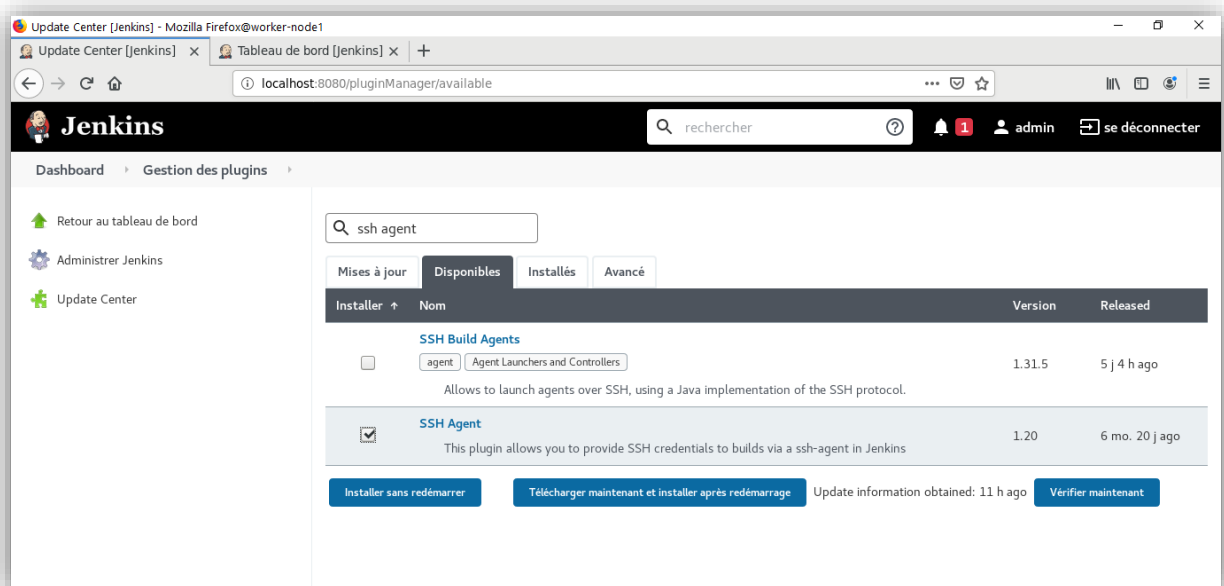


⇒ Changez le code pour corriger l'erreur du Maven:

```
stage('Build Package'){
    def mvnHome = tool name: 'Maven3', type: 'maven'
    sh "${mvnHome}/bin/mvn package"
}
```

3. Installation du plugin SSH Agent sur Jenkins

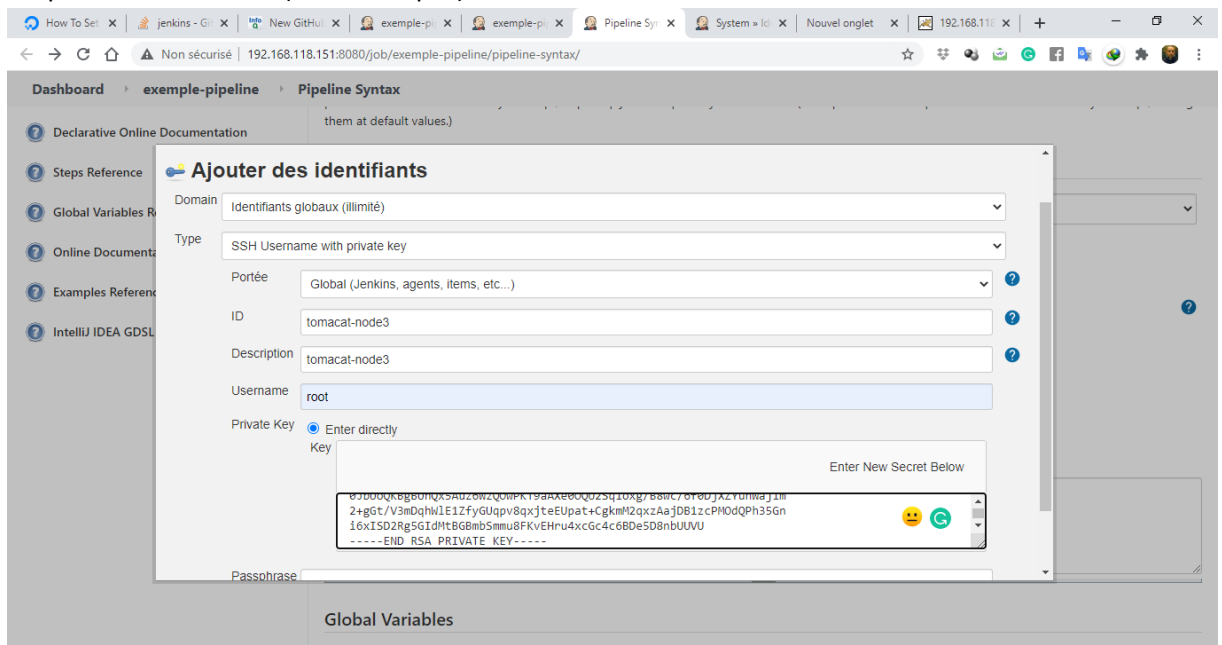
Le plugin SSH Agent est nécessaire pour exécuter des requêtes ssh/scp sur des autres systèmes distants. On va l'utiliser pour copier le package war généré par Maven lors du build



- Ajout d'un 3iem stage dans le Pipeline pour livrer le package générer sur le serveur Tomcat déployé. Dans cet exemple nous allons utiliser le plugin ssh agent pour utiliser des commandes systèmes pour la copie du fichier.
 1. Créer un nouveau stage pour le déploiement dans Tomcat

```
stage('Deploimeent Dans Tomcat'){  
}
```

2. Accéder à la page de syntaxe de pipeline pour préparer l'exemple d'accès sur une machine distante.
- Il faut sélectionner SSH agent
 - Préparer un crédenial (si n'existe pas)



- Générer le code
- Ajouter le code dans le stage du Pipeline

```
sshagent(['rootNode3']) {  
    // some block  
}
```

- Ajouter les instructions suivantes
 - Changer le nom du package (pour éviter les builds ou autres label du package)

```
sh "mv target/web-app-maven*.war target/myweb.war"
```

- Copiez le package dans le serveur tomcat

```
sh "scp -o StrictHostKeyChecking=no target/myweb.war  
root@192.168.118.152:/opt/tomcat8/webapps/"
```

- Redémarrerez serveur tomcat

```
sh "ssh root@192.168.118.152 /opt/tomcat8/bin/shutdown.sh"
sh "ssh root@192.168.118.152 /opt/tomcat8/bin/startup.sh"
```

Pipeline final :

```
node{
    stage('SCM Checout'){
        git branch: 'main', url: 'https://github.com/medsalahmeddeb/web-app-maven'
    }

    stage('Build Package'){
        def mvnHome = tool name: 'Maven3', type: 'maven'
        sh "${mvnHome}/bin/mvn clean package"
    }

    stage('Deploimeent Dans Tomcat'){
        sshagent(['rootNode3']) {
            sh "mv target/web-app-maven*.war target/myweb.war"
            sh "scp -o StrictHostKeyChecking=no target/myweb.war root@192.168.118.152:/opt/tomcat8/webapps/"
            sh "ssh root@192.168.118.152 /opt/tomcat8/bin/shutdown.sh"
            sh "ssh root@192.168.118.152 /opt/tomcat8/bin/startup.sh"
        }
    }
}
```

- Ajoutez une notification par email et par slack sur le Pipeline.

4. Avec Pipelines déclaratives :

```
pipeline {
  agent any

  stages {
    stage('SCM'){
      steps{
        git branch: 'main',
          url: 'https://gitlab.com/meddeb/demo-webapp'
      }
    }
    stage('Build') {
      tools {
        maven 'Maven3'
      }
      steps {
        sh 'mvn package'
      }
    }
    stage('Deploy'){
      steps{
        deploy adapters: [tomcat8(credentialsId: 'tomcat-deployer',
          path: '',
          url: 'http://151.106.12.181:8081')],
          contextPath: null,
          war: '**/*.war'
      }
    }
  }
}
```