

Lab2 – Jenkins CI

Préparation de l'environnement (Ubuntu 18,04)

1. Démarrer une machine linux Ubuntu18.04 avec au moins 8 Go de RAM et 2CPU.
2. Mettre à jour la liste des paquets : *sudo apt update*
3. Installer les paquets suivants : *git vim gcc build-essential unzip openjdk-11-jdk openjdk-8-jre-headless postgresql nginx*

Installation et configuration de Jenkins

1. Installer la dernière version de Jenkins, en exécutant les commandes suivantes :
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins.io/debian-stable binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt update
sudo apt install -y jenkins
2. Vérifier que le service jenkins tourne :
systemctl status jenkins
3. Connecter sur l'interface web de Jenkins :
URL : http://localhost:8080
4. Une fois authentifier, installer les plugins suivants :
 - *artifact-promotion*
 - *Maven Artifact ChoiceListProvider (Nexus)*
 - *Maven Release Plug-in Plug-in*
 - *Nexus Artifact Uploader*
 - *Nexus Platform Plugin*
 - *Pipeline Utility Steps*
 - *Pipeline: GitHub Groovy Libraries*
 - *SonarQube Scanner for Jenkins*
5. Configurer les plugins de Maven, Nexus et SonarQube
6. Créer puis exécuter le pipeline suivant :

```
pipeline {  
  agent {  
    label "master"  
  }  
  tools {
```

```
// Note: this should match with the tool name configured in your jenkins instance
(JENKINS_URL/configureTools/)
  maven "Maven 3.6.0"
}
environment {
  // This can be nexus3 or nexus2
  NEXUS_VERSION = "nexus3"
  // This can be http or https
  NEXUS_PROTOCOL = "http"
  // Where your Nexus is running
  NEXUS_URL = "127.0.0.1:8081"
  // Repository where we will upload the artifact
  NEXUS_REPOSITORY = "maven-releases"
  // Jenkins credential id to authenticate to Nexus OSS
  NEXUS_CREDENTIAL_ID = "jenkins"
}
stages {
  stage("clone code") {
    steps {
      script {
        // Let's clone the source
        git 'https://github.com/danielalejandroh/cargotracker.git';
      }
    }
  }
  stage("mvn build") {
    steps {
      script {
        // If you are using Windows then you should use "bat" step
        // Since unit testing is out of the scope we skip them
        sh "mvn package -DskipTests=true"
      }
    }
  }
}
```

```

    }
}
stage("publish to nexus") {
    steps {
        script {
            // Read POM xml file using 'readMavenPom' step , this step 'readMavenPom' is
included in: https://plugins.jenkins.io/pipeline-utility-steps
            pom = readMavenPom file: "pom.xml";
            // Find built artifact under target folder
            filesByGlob = findFiles(glob: "target/*.${pom.packaging}");
            // Print some info from the artifact found
            echo "${filesByGlob[0].name} ${filesByGlob[0].path} ${filesByGlob[0].directory} $
{filesByGlob[0].length} ${filesByGlob[0].lastModified}"
            // Extract the path from the File found
            artifactPath = filesByGlob[0].path;
            // Assign to a boolean response verifying If the artifact name exists
            artifactExists = fileExists artifactPath;
            if(artifactExists) {
                echo "**** File: ${artifactPath}, group: ${pom.groupId}, packaging: $
{pom.packaging}, version ${pom.version}";
                nexusArtifactUploader(
                    nexusVersion: NEXUS_VERSION,
                    protocol: NEXUS_PROTOCOL,
                    nexusUrl: NEXUS_URL,
                    groupId: pom.groupId,
                    version: pom.version,
                    repository: NEXUS_REPOSITORY,
                    credentialsId: NEXUS_CREDENTIAL_ID,

                    artifacts: [
                        // Artifact generated such as .jar, .ear and .war files.
                        [artifactId: pom.artifactId,

```

```
        classifier: "",
        file: artifactPath,
        type: pom.packaging],
    // Lets upload the pom.xml file for additional
    // information for Transitive dependencies
    [artifactId: pom.artifactId,
     classifier: "",
     file: "pom.xml",
     type: "pom"]
    ]
);
} else {
    error "**** File: ${artifactPath}, could not be found";
}
}
}
}
}
```

Installation et configuration de Nexus

- ## 7. Installer la dernière version de Maven :

```
sudo apt install maven
```

- 8. Installer Nexus en exécutant les commandes suivantes :**

```
cd /opt
```

```
sudo wget -O nexus.tar.gz https://download.sonatype.com/nexus/3/latest-unix.tar.gz
```

```
sudo tar xzvf nexus.tar.gz
```

```
sudo mv nexus-3.19.1-01/ nexus
```

9. Créer un utilisateur nexus, et le configurer comme propriétaire des dossiers nexus :

- ## 10. sudo useradd -m nexus

```
sudo chown -R /opt/nexus
```

```
sudo chown -R /opt/sonatype-work
```

Ajouter la ligne suivante au fichier `/opt/nexus/bin/nexus.rc` :

```
run as user="nexus"
```

11. Démarrer nexus en tant qu'utilisateur nexus :

/opt/nexus/bin/nexus start

Si le port TCP/8081 est ouvert, connecter sur l'interface web de nexus :

URL : <http://localhost:8081>

Installation et configuration de Sonarqube

12. Installer et configurer Sonarqube en exécutant les commandes suivantes :

cd /opt

sudo wget -O sonar.zip https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.0.zip

sudo unzip sonar.zip

sudo mv sonarqube-8.0/ sonar

useradd -m sonar

chown -R sonar.sonar /opt/sonar

sudo echo "sonar.jdbc.username=sonar" >> /opt/sonar/conf/sonar.properties

sudo echo "sonar.jdbc.password=sonar" >> /opt/sonar/conf/sonar.properties

sudo echo "sonar.jdbc.url=jdbc:postgresql://localhost/sonar?" >>

/opt/sonar/conf/sonar.properties

sudo echo "sonar.web.host=0.0.0.0" >> /opt/sonar/conf/sonar.properties

sudo echo "sonar.search.javaOpts=-Xms512m -Xmx512m" >>

/opt/sonar/conf/sonar.properties

sudo useradd sonar

sudo echo "RUN_AS_USER=sonar" >> /opt/sonar/conf/sonar.properties

sudo sysctl -w vm.max_map_count=262144

13. Démarrer SonarQube en tant qu'utilisateur sonar :

/opt/sonar/bin/linux-x86/sonar.sh start

14. Vérifier que sonar a bien démarré et que les ports TCP 9000 et 9001 sont ouverts

Si c'est le cas, ouvrir l'interface web de SonarQube :

<http://localhost:9000>