Lab2 – Jenkins CI

Préparation de l'environnement (Ubuntu 18,04)

1. Installer les paquets suivants : git vim gcc build-essential unzip openjdk-11-jdk openjdk-8-jre-headless postgresql
2. Installer et configurer les outils suivants : Jenkins maven nexus3 sonar

Création de pipeline (Jenkins)

1. 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
2. Configurer les plugins de Maven, Nexus et SonarQube
3. 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/danielalejandrohc/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";
                    }
                }
            }
        }
    }
}
```

4.