**Enoncé Chapitre 2**

1. Créer un deuxième microservice appelé notification et rajouter le à votre espace de travail Intellij
2. Faites communiquer les deux microservices utilisant les API REST
3. Installer Jenkins sur votre machine
4. Créer une pipeline Jenkins qui retire le code de votre projet depuis le repo Github, le build génère un artifact
5. Brancher l’outil d’analyse de code SonarQube avec Jenkins et inclure le test d’analyse dans votre pipeline

**Solution**

1. Rajouter un nouveau module notification au projet sur Intellij
2. Utiliser REST Template pour faire communiquer les deux micro services (Voir la méthode ProductController. addNewProduct())
3. Suivre le lien : <https://www.jenkins.io/doc/book/installing/windows> pour l’installation de Jenkins. Sinon le lien de téléchargement est : <https://www.jenkins.io/download/#downloading-jenkins>
4. Code de la pipeline pour le build du projet:

pipeline {

agent any

stages {

stage("Checkout Code") {

steps {

git branch: 'main', url: 'https://github.com/aymendr/formation\_devOps\_prise\_en\_mains'

}

}

stage("Build") {

steps {

bat 'mvn clean install'

}

}

}

}

1. Installer SonarQube avec DockerCompose:

version: "3"

services:

sonarqube:

image: sonarqube:community

depends\_on:

- db

environment:

SONAR\_JDBC\_URL: jdbc:postgresql://db:5432/sonar

SONAR\_JDBC\_USERNAME: sonar

SONAR\_JDBC\_PASSWORD: sonar

volumes:

- sonarqube\_data:/opt/sonarqube/data

- sonarqube\_extensions:/opt/sonarqube/extensions

- sonarqube\_logs:/opt/sonarqube/logs

ports:

- "9000:9000"

db:

image: postgres:12

environment:

POSTGRES\_USER: sonar

POSTGRES\_PASSWORD: sonar

volumes:

- postgresql:/var/lib/postgresql

- postgresql\_data:/var/lib/postgresql/data

volumes:

sonarqube\_data:

sonarqube\_extensions:

sonarqube\_logs:

postgresql:

postgresql\_data:

* Installation classique de SonarQube:

Sonarqube installer: <https://docs.sonarsource.com/sonarqube/latest/setup-and-upgrade/install-the-server/installing-sonarqube-from-zip-file/>

Démarrer l’installer, terminer l’installation et ouvrir le navigateur sur l’adresse : <http://localhost:9000/>

* Installer le plugin Sonarqube Scanner dans Jenkins
* Manage Jenkins → Système → Sonarqube.Remplir les informations avec l’url du serveur
* Authentication Token: Aller sur Sonarqube → My Account → Security → Générer un token de type Global Analysis Token, copier le Token, revenir sur Jenkins, rajouter un credential de type secret, coller le Token, donner un Id
* Rajouter le plugin maven avec la configuration nécessaire depuis : <https://docs.sonarsource.com/sonarqube/9.8/analyzing-source-code/scanners/sonarscanner-for-maven/>
* Créer un pipeline Jenkins et rajouter une step d’analyse sonarqube, Exp:

pipeline {

agent any

stages {

stage("Checkout Code") {

steps {

git branch: 'main', url: 'https://github.com/aymendr/formation\_devOps\_prise\_en\_mains'

}

}

stage("Build") {

steps {

bat 'mvn clean install'

}

}

stage("SonarQube Analyse") {

steps {

withSonarQubeEnv('sonarqube') {

bat 'mvn sonar:sonar'

}

}

}

}

}