2022

Pipeline CI/CD POC Emergency Responder Subsystem



Magalie Morteau Architecte Logiciel MedHead 20/01/2022

Table of contents

1	Obje	ectif	2
2	Des	cription du pipeline	3
	2.1	Les jobs	3
	2.2	Les déclencheurs	3
	2.3	Le job build	4
	2.4	Le job tests	5
	2.5	Le job sonar	6
	2.6	Le job deploy	7
3	Rés	ultats	8
	3.1	Déclenchement sur un Pull Request	8
	3.2	Exécution avec succès du pipeline CI/CD	9
	3.3	Job Build	10
	3.4	Job Tests	11
	3.5	Job Sonar	12
	3.6	Job Deploy	14

1 Objectif

Ce document présente une description de la configuration du **Pipeline CI/CD** du POC ERS (Emergency Responder Subsystem).

On utilise l'outil GitHub Actions intégré dans GitHub pour créer un Workflow nous permettant de gérer le pipeline CI/CD.

Un Workflow étant une procédure automatisée composée d'une ou plusieurs étape. Cette procédure est écrite en YAML.

Pour notre projet, la configuration du Pipeline CI/CD est codée dans le fichier <u>maven.yml</u> en tant que workflow du repository GitHub de l'API.

Depuis Octobre 2021, il est possible de réutiliser des workflows complet.

Cette configuration pourra donc être réutilisée pour d'autres POC.

Liens officiels:

- GitHub Actions: https://docs.github.com/en/actions
- Réutilisation: https://docs.github.com/en/actions/using-workflows/reusing-workflows
- $\bullet \quad \text{Syntaxe}: \underline{\text{https://docs.github.com/en/actions/using-workflows/workflow-syntax-for-github-actions\#github-hosted-runners} \\$

2 Description du pipeline

2.1 Les jobs



Le pipeline sous Github (maven.yml) est configuré pour :

- 1. Compiler le projet
- 2. Exécuter les tests
- 3. Vérifier la qualité avec SonarCloud
- 4. Construire et déployer le fichier jar

L'ensemble du pipeline doit s'exécuter en moins de 3 minutes.

2.2 Les déclencheurs

```
name: CI/CD Pipeline

on:
    #Trigger the workflow on pull request and push from the main branch
pull_request:
    branches: [ main ]
push:
    branches: [ main ]
```

Le pipeline CI/CD se déclenche automatiquement sur les Pull Request et les Push sur la branche principale.

2.3 Le job build

jobs:

```
#build's job
build:
   name: Build
   runs-on: ubuntu-latest
   steps:
   - uses: actions/checkout@v2
   - name: Set up JDK 17
    uses: actions/setup-java@v2.5.0
   with:
        java-version: '17'
        distribution: 'temurin'
        cache: maven
   - name: Build with Maven
   run: mvn -B compile
```

Le job **build**, s'exécute sur une machine virtuelle Linux de GitHub avec la dernière version Ubuntu.

Clone le repo GitHub et charge la version 17 du Java JDK.

Puis compile le code avec la commande « compile » de Maven.

2.4 Le job tests

```
#Test's job
tests:
 #Depends on build's job
 needs: build
 name: Tests
 runs-on: ubuntu-latest
 steps:
  - uses: actions/checkout@v2
  - name: Set up JDK 17
    uses: actions/setup-java@v2.5.0
    with:
      java-version: '17'
      distribution: 'temurin'
      cache: maven
  - name: Run Tests
    run: mvn -B test
```

Le job **tests** ne va s'exécuter que si le job **build** se termine sans erreur.

Pour optimiser les dépendances Maven sont mises dans le cache.

Puis exécute les tests avec la commande « test » de Maven.

2.5 Le job sonar

```
#Sonar's Job
sonar:
 #Depends on build's job
 needs: build
 name: SonarCloud analysis
 runs-on: ubuntu-latest
   - uses: actions/checkout@v2
         fetch-depth: 0
    - name: Set up JDK 17
     uses: actions/setup-java@v2.5.0
       java-version: '17'
       distribution: 'temurin'
       cache: maven
    - name: Cache SonarCloud packages
     uses: actions/cache@v2
     with:
       path: ~/.sonar/cache
       key: ${{ runner.os }}-sonar
       restore-keys: ${{ runner.os }}-sonar
    - name: Analyze with SonarCloud
      run: mvn -B verify org.sonarsource.scanner.maven:sonar-maven-plugin:sonar -Dsonar.projectKey=Mamak2020_poc-api-ers
       GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
       SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

Le job sonar ne va s'exécuter que si le job build se termine sans erreur.

On utilise un token GITHUB_TOKEN pour autoriser Sonar à accéder un repository GitHub et un token SONAR_TOKEN pour autoriser GitHub à accéder au projet Sonar.

Puis vérifie la qualité du code avec la commande « verify » de Maven.

2.6 Le job deploy

```
#Deploy's job
deploy:
  #Depends on Sonar's job
  needs: sonar
  name: Deploy
  runs-on: ubuntu-latest
  steps:
  - uses: actions/checkout@v2
  - name: Set up JDK 17
   uses: actions/setup-java@v2.5.0
   with:
      java-version: '17'
     distribution: 'temurin'
      cache: maven
  - run: mvn -B package -DskipTests --file pom.xml
  - name: Upload JAR
    uses: actions/upload-artifact@v2.3.1
      name: api-ers
      path: target/*.jar
```

Le job **deploy** ne va s'exécuter que si le job **sonar** se termine sans erreur.

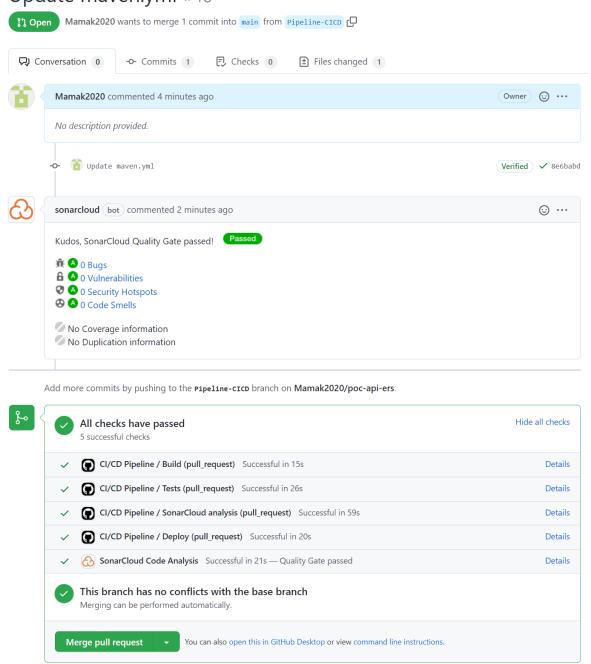
On crée le package Java avec la commande « package » de Maven.

Puis on charge l'artéfact généré (fichier jar compressé) afin qu'il puisse être téléchargé.

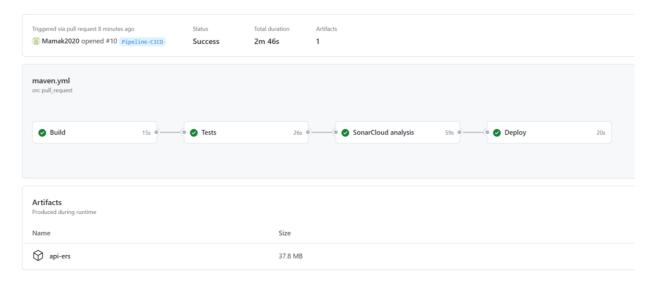
3 Résultats

3.1 Déclenchement sur un Pull Request

Update maven.yml #10

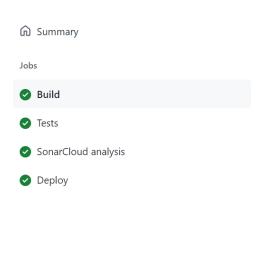


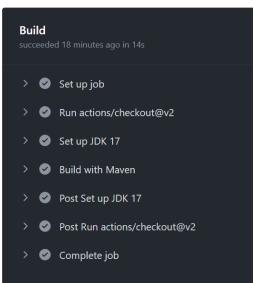
3.2 Exécution avec succès du pipeline CI/CD

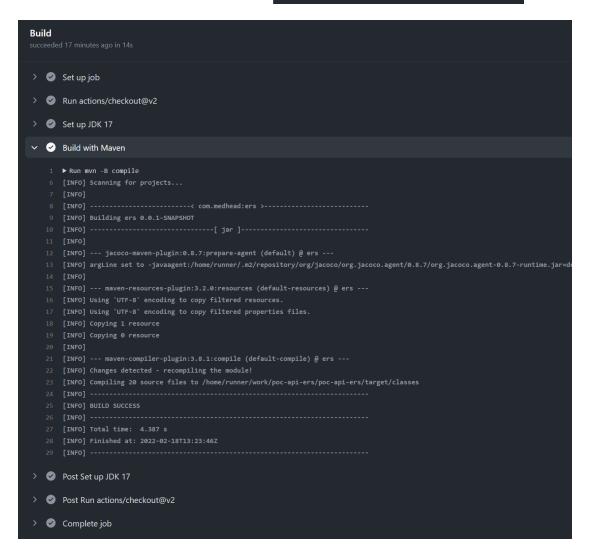


L'ensemble du pipeline CI/CD s'exécute en moins de 3 minutes.

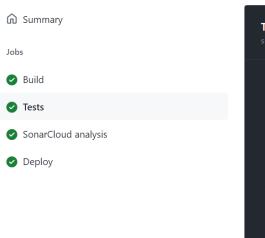
3.3 Job Build

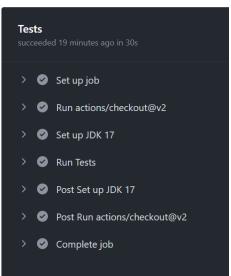






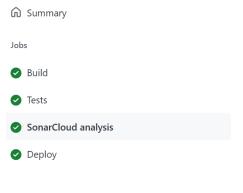
3.4 Job Tests

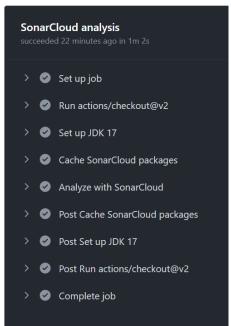


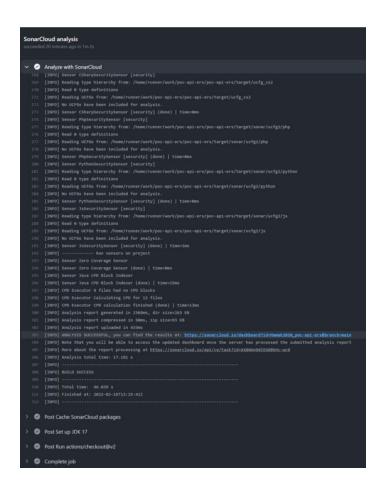


3.5 Job Sonar

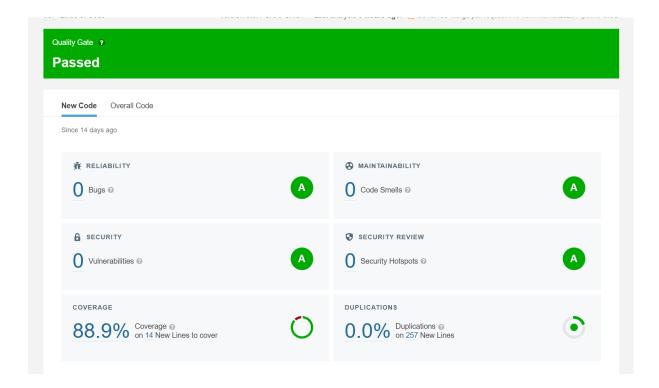
Résultats depuis GitHub







Résultats dans Sonar Cloud



3.6 Job Deploy

Summary
 Jobs
 → Build
 ✓ Tests
 ✓ SonarCloud analysis
 ✓ Deploy

Deploy
succeeded 22 minutes ago in 25s

> Set up job

> Run actions/checkout@v2

> Set up JDK 17

> Run mvn -B package -DskipTests --file pom.xml

> Upload JAR

> Post Set up JDK 17

> Post Run actions/checkout@v2

> Complete job

Package

```
✓ ✓ Run mvn -B package -DskipTests --file pom.xml
       ▶ Run mvn -B package -DskipTests --file pom.xml
       [INFO] Building ers 0.0.1-SNAPSHOT
   11 [INFO]
       [INFO] argLine set to -javaagent:/home/runner/.m2/repository/org/jacoco/org.jacoco.agent/0.8.7/org.jacoco.agent-0.8
   19 [INFO] Copying 0 resource
   22 [INFO] Changes detected - recompiling the module!
   26 [INFO] Using 'UTF-8' encoding to copy filtered resources.
       [INFO] Copying 2 resources
   30 [INFO] --- maven-compiler-plugin:3.8.1:testCompile (default-testCompile) @ ers ---
       [INFO] Compiling 7 source files to /home/runner/work/poc-api-ers/poc-api-ers/target/test-classes
      [INFO] --- jacoco-maven-plugin:0.8.7:report (report) @ ers ---
   38 [INFO] Skipping JaCoCo execution due to missing execution data file.
   41 [INFO] Building jar: /home/runner/work/poc-api-ers/poc-api-ers/target/ers-0.0.1-SNAPSHOT.jar
       [INFO] --- spring-boot-maven-plugin:2.6.2:repackage (repackage) @ ers ---
```

Upload JAR