BobApp

CI / CD

Version	Date	Auteur	Description
1.0	01/04/2024	Gilles BERNARD	Version initiale

Table des matières

1. Contexte

Ce document présente l'automatisation de tâches permettant de gérer l'application BobApp.

Ces tâches sont intégrées dans un pipeline CI/CD via des Github Actions.

SonarCloud permet d'analyser la qualité du code.

Les containers Docker sont déployés sur Docker Hub.

2. Pipeline CI/CD

Un pipeline CI/CD s'exécute lors de chaque pull-request sur le repository Github.

Le pipeline se trouve dans le fichier `.github/workflows/ci-cd.yml` situé à la racine du projet.

Le pipeline est découpée en jobs :

- build, test et génération des rapports de couverture de code,
- analyse de la qualité de code,
- déploiement.

Ces jobs sont découpés en steps.

2.1 Build, test et couverture de code

Deux jobs sont dédiés à cette tâche : un pour le back-end, un autre pour le front-end.

Le but de ces jobs est de compiler, tester et générer les rapports de couverture de code.

2.1.1 Back-end

Le job du back-end se nomme `backend-build-test`.

Ses steps sont :

- vérification du repository,
- installation de Java 11,
- installation des dépendances,
- exécution des tests,
- génération de la couverture de code et affichage dans un commentaire de la pull-request,
- upload du rapport de couverture,

- affichage du lien de la couverture dans un autre commentaire.

2.1.2 Front-end

Le job du front-end se nomme frontend-build-test.

Ses steps sont :

- vérification du repository,
- installation de Node.js,
- installation des dépendances,
- exécution des tests,
- génération de la couverture de code,
- upload du rapport de couverture,
- affichage de la couverture dans un commentaire,
- affichage du lien de la couverture dans un autre commentaire.

2.2 Analyse de la qualité de code

Deux jobs sont dédiés à cette tâche : un pour le back-end, un autre pour le front-end.

Le but de ces jobs est de lancer un scan Sonar afin d'analyser la qualité du code.

Le résultat de ces analyses est hébergé sur SonarCloud :

- lien du back-end
- lien du front-end

2.2.1 Back-end

Le job du back-end se nomme backend-quality.

Ce job a besoin que le job backend-build-test soit terminé et réussi.

Ses steps sont :

- vérification du repository,
- installation de Java 17,
- mise en cache des packages SonarCloud,
- mise en cache des packages Maven,
- exécution de Maven verify : validate, compile, test, package,
- exécution du scanner Sonar.

2.2.2 Front-end

Le job du front-end se nomme frontend-quality.

Ce job a besoin que le job frontend-build-test soit terminé et réussi.

Ses steps sont :

- vérification du repository,
- création du répertoire coverage qui contiendra le rapport de couverture,
- download du rapport de couverture dans le répertoire coverage,
- exécution du scanner Sonar.

2.3 Déploiement

Le job de déploiement se nomme deploy.

Le but de ce job est de déployer les images des containers back-end et front-end sur Docker Hub : lien du repository.

Ce job a besoin que les jobs backend-quality et frontend-quality soient terminés et réussis.

Ses steps sont :

- vérification du répertoire,
- installation de la machine virtuelle Qemu,
- installation du plugin Docker Buildx,
- connexion à Docker Hub,
- déploiement de l'image Docker du back-end sur Docker Hub,
- déploiement de l'image Docker du front-end sur Docker Hub.

3. KPI proposés

3.1 KPI de couverture de code

Deux KPI de couverture de code sont paramétrés :

- le taux minimum de couverture de code du projet : 80 %,
- et le taux minimum de couverture de code de fichiers modifiés : 60 %.

3.2 KPI de qualité de code

Les KPI de qualité de code suivants sont activés par défaut par Sonar :

- aucun nouveau problème n'est introduit,
- tous les nouveaux points sensibles liés à la sécurité sont examinés,
- la couverture de test du nouveau code est supérieure ou égale à 80 %,
- la duplication dans le nouveau code est inférieure ou égale à 3,0%.

4. Analyses

4.1 Analyse des métriques

La qualité de code est analysée par SonarCloud et présentée via différents métriques.

4.1.1 Coté back-end

Les métriques du back-end sont disponible ici.

Il en ressort que :

- la couverture de code par les tests est faible : 38,8 %,
- 2 points sensibles de sécurité sont présents,
-

4.1.2 Coté front-end

4.2 Analyse des avis utilisateurs