

Le projet consiste à créer les chaines CI/CD d'une application de gestion des événements en lignes. Ci-dessous, le diagramme de classes :

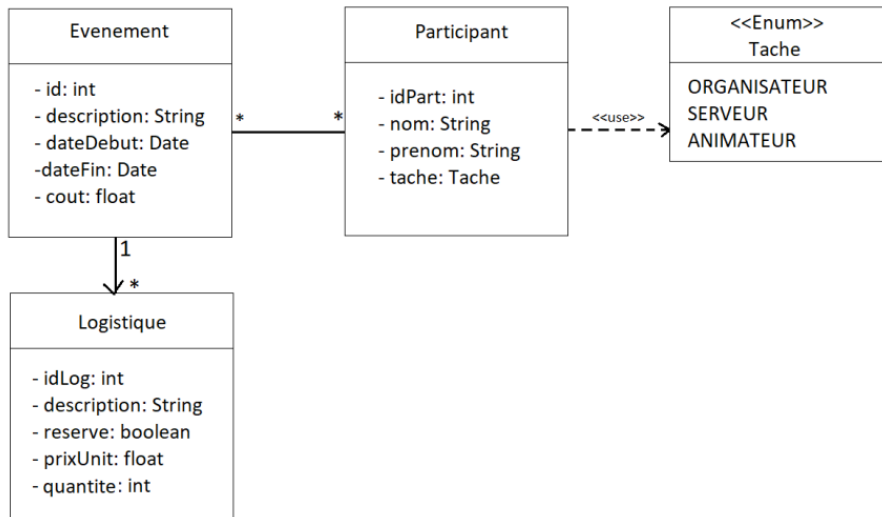


Figure 1. Diagramme de classes de gestions des événements

Le code source de cette application se trouve sur ce lien :

https://drive.google.com/file/d/1hNWvzt7R_YQjFX-3zNSKjnm4i6vMFnfY/view?usp=drive_link

TRAVAIL À FAIRE

1. Créer l'architecture suivante pour réaliser les chaines CI/CD

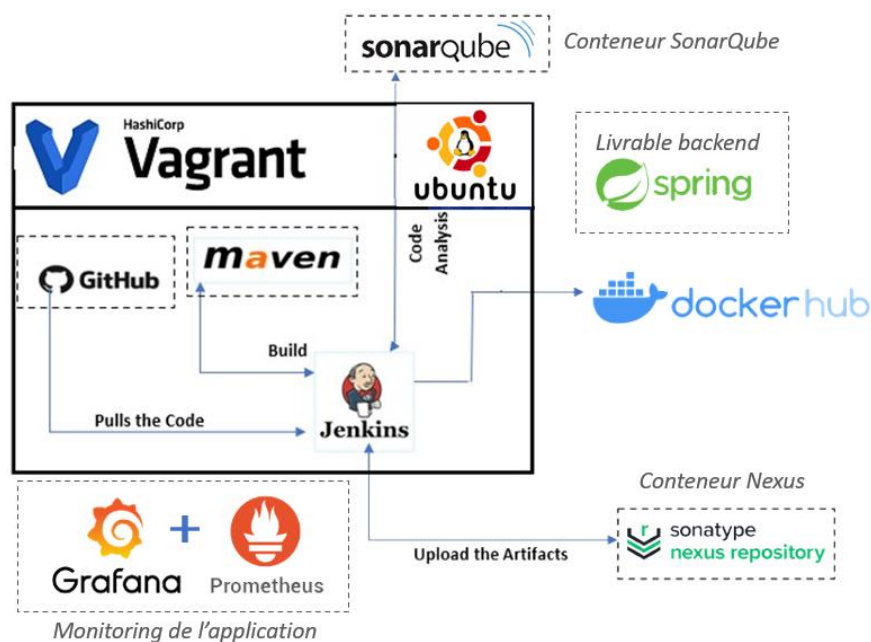


Figure 2. Architecture de la solution finale

2. Implémenter les tests unitaires (En utilisant Mock).
3. Créer un script qui va être automatisé avec l'outil « Jenkins » qui comporte les parties suivantes :
 - Récupération du projet de Git.
 - Compilation du projet.
 - Lancement des tests unitaires automatiques (JUnit).
 - Lancement des tests de qualité de code (SonarQube).
 - Préparation de la version à distribuer.
 - Mise en place de la version à distribuer (Nexus).
 - Création de l'image qui contient le livrable Spring à partir du fichier DockerFile.
 - Dépôt de l'image créée sur DockerHub.
 - Lancement simultané de l'image contenant le livrable Spring Boot et l'image MySQL en utilisant docker-compose.
4. Corriger les problèmes dégagés par l'outil Sonar.
5. Tester les services de l'application avec Postman ou Swagger. Un simple ajout d'une ligne suivie d'un Get de l'élément ajouté est suffisant.
6. Superviser les différents conteneurs en utilisant Grafana et Prometheus.

REMARQUES

1. Chaque étudiant devra ajouter son enseignant au répertoire git (via le mail esprit de l'enseignant).
2. Lancer (à nouveau) le pipeline complet avec l'une des façons suivantes :
 - a. Un simple push du code sur le GitHub lancera le pipeline d'une façon automatique (La solution idéale)
 - b. Un lancement à travers l'utilisation d'un script (url ou autre), périodique ou manuel (La solution par défaut)
3. Des questions sur les différents outils et notions de cours (DevOps en général+ intégration continue + livraison continue) seront posées à chaque étudiant.

Bon courage