

TP N° 8

Intégration continue d'une API Java avec Jenkins

Enoncé du TP

Il s'agit de développer le processus d'intégration continue de deux versions d'une API de calcul matriciel. Le pipeline est représenté dans la figure ci-dessous.

La création et l'exécution du Pipeline doit se faire avec l'interface Blue Ocean.



1 Configuration de Jenkins

Utilisez le fichier *configuration-Jenkins.pdf* pour effectuer toutes les configurations nécessaires.

2 Création et partage du projet

1. Créer un projet Gradle et intégrer le code source et le fichier *build.gradle*.
2. Partager le code source sur le GitHub en ajoutant un fichier *.gitignore* pour ignorer les commits du dossier build.

3.1 Les phases du pipeline

3.1.1 La phase Build

Cette phase est composée des étapes suivantes:

1. Lancement du build Gradle.
2. Génération de la documentation.
3. Archivage du fichier Jar et de la documentation.
4. Archivage des résultats de tests unitaires.
5. Après le build, une notification par mail doit être envoyée au(x) membre(s) du projet dans les deux cas: succès et échec de la phase Build.

3.1.2 La phase Code Analysis

Dans cette phase, analyser la qualité du code (en lançant la commande du Plugin de Gradle) et vérifier l'état de Quality Gates. Si cette dernière est dans l'état *Failed*, le pipeline doit s'arrêter et un mail doit être envoyé au(x) membre(s) du projet.

3.1.3 La phase Test reporting

Utilisez le plugin Cucumber reports de Jenkins pour générer les rapports des tests Cucumber.

3.1.4 La phase Deployment

Lancez la tâche *publish* de Gradle pour déployer le fichier Jar généré dans <https://mymavenrepo.com/>. Les paramètres d'authentification doivent être créés dans Jenkins.

3.1.5 La phase Slack Notification

Utilisez le plugin de Jenkins *Slack Notification* pour envoyer une notification après le déploiement de l'API.