Examen Pratique : Jenkins CI/CD - Pipeline Java avec Robot Framework et SonarQube

Durée : 1h30

Exercice 1 :

- Créer un projet Jenkins Freestyle, qui affiche sur la console le texte suivant :
  - (« Hello Jenkins »),
  - puis affiche sur la console la date système.
- Configurez un déclencheur (Trigger) pour que ce job s'exécute automatiquement chaque 15 minutes.
- Configurez un déclencheur pour que ce job s'exécute automatiquement chaque jour à 12 :30 am.
- Configurez un déclencheur pour que ce job s'exécute automatiquement à 17h45 tout les jours de la semaine sauf le samedi et le dimanche.
- Créez un déclencheur de builds à distance en utilisant le Jeton (25092024).

Exercice 2 :

Vous travaillez en tant qu'ingénieur DevOps dans une équipe de développement Java. Votre tâche est de mettre en place un pipeline CI/CD complet dans Jenkins pour le projet Java **SpringPetClinic**[1]. Ce pipeline doit inclure des tests automatisés avec Robot Framework, une analyse statique de la qualité du code avec SonarQube, et l'utilisation d'un build Maven pour compiler et tester l'application.

Le pipeline doit être capable de :

* Cloner le repository contenant le code source Java (que vous devez monter sur votre compte Github).
* Compiler le projet avec Maven.
* Exécuter les tests unitaires.
* Exécuter les tests fonctionnels avec Robot Framework.
* Effectuer une analyse de la qualité du code avec SonarQube.

[1] https://github.com/karoumbr/SpringPetClinic

* Générer un rapport et notifier en cas d'échec.

2. Pipeline détaillé :

- Le pipeline doit inclure les étapes suivantes :

1. Checkout du code : Cloner le code source à partir du repository Git.
2. Analyse statique avec SonarQube : Exécuter une analyse de qualité du code pour identifier les bugs, vulnérabilités et problèmes de maintenabilité.
3. Build et tests unitaires : Utiliser Maven pour compiler le projet et exécuter les tests unitaires.
4. Tests fonctionnels avec Robot Framework : Lancer deux tests fonctionnels (afin de tester la recherche de propriétaires (find owners) et la récupération de la liste des vétérinaires.
5. Génération des rapports : Générer les rapports de test et de qualité du code (JUnit, Robot Framework, SonarQube).
6. Notifications : Envoyer des notifications en cas de succès ou d'échec du pipeline. (bonus)

Document à rendre :

Vous devez préparer un document qui comporte :

- des imprimes écran des résultats des constructions ;
- les commandes en format texte ;
- le code du pipeline.

Le document final (pdf ou word) doit être envoyé sur le compte email : karim.benromdhan@iteam-univ.tn