

Support de TP  
**DevOps**

# Chapitre 7 : Vérification de la qualité du code

## SOMMAIRE

I.	Installation de SonarQube .....	2
II.	Amélioration de la qualité.....	2
III.	Intégration avec Jenkins .....	2

# I. Installation de SonarQube

1. Modifier la stratégie de démarrage du **Nexus** en désactivant le démarrage automatique.
2. Au niveau du dossier utilisateur de la machine Linux, créer un script permettant d'arrêter Nexus nommé **nexus-down** et un autre permettant de démarrer Nexus appelé **nexus-up**.
3. En utilisant la documentation appropriée, récupérer la dernière image de **SonarQube** et créer un container nommé **sonarqube**.
4. Créer les deux scripts **sonarqube-up** et **sonarqube-down**.
5. Arrêter Nexus et lancer SonarQube. En utilisant la documentation appropriée, se synchroniser avec les logs du conteneur sonarqube et vérifier qu'il a été bien démarré.
6. Se connecter à l'interface graphique, modifier le mot de passe et vérifier qu'il n'y a aucun projet analysé.
7. En utilisant la documentation appropriée, créer un **jeton** d'analyse globale.
8. Lancer l'analyse sur le projet **timesheet** et vérifier au niveau de SonarQube que le projet a été analysé.

# II. Amélioration de la qualité

9. Chaque étudiant doit appartenir à un groupe. Un chef doit être désigné.
10. Le chef du groupe doit créer un dépôt nommé sous la forme **timesheet-classe-groupe-n** où n est le numéro du groupe. Il doit par la suite donner accès à ces collègues (**Collaborators >> Settings**).
11. Créer une branche nommée sous la forme **nomprenom-classe-groupe-n**.
12. Chaque étudiant choisit un problème de maintenabilité parmi les problèmes détectés lors de l'analyse. Il doit le corriger.
13. Envoyer le code de votre branche dans le dépôt central du groupe.
14. Créer un **Pull Request** de fusion au chef et ensemble créer ou modifier la branche **master**.

# III. Intégration avec Jenkins

15. Au niveau de **Jenkins**, créer un Job pipeline nommé sous la forme **Chap7-SonarQube-nomprenom-classe**. Le Job récupère le code source de la branche master à partir de GitHub, puis lance le nettoyage, compilation, tests, contrôle qualité (4 stages au total nommés : **GIT**, **COMPILATION**, **TESTS** et **SONARQUBE**).

16. A partir de l'interface de SonarQube et au niveau de l'onglet **Activity**, vérifier que le nombre de problèmes de maintenabilité a été réduit.