#### **1. Mise en place d’un pipeline CI/CD avec GitLab**

**Question 1 :** Décrivez les étapes nécessaires pour initialiser un projet dans GitLab et configurer un fichier .gitlab-ci.yml afin de créer un pipeline CI/CD de base.

**Question 2 :** Expliquez les étapes nécessaires pour créer un fichier .gitlab-ci.yml dans un projet GitLab. Quelles informations doivent être incluses pour définir les étapes du pipeline ?

**Question 3 :** Comment définiriez-vous des étapes de pipeline comme build, test et deploy dans un fichier .gitlab-ci.yml ?

**Question 4 :** Quelle est la structure de base d’un fichier .gitlab-ci.yml et comment définissez-vous les stages (étapes) de votre pipeline CI/CD ?

**Question 5 :** Quel est le rôle de la section script dans un job GitLab ? Donnez un exemple de configuration pour un job de construction qui utilise npm pour installer les dépendances.

**Question 6 :**Donnez un exemple de configuration de job dans un pipeline GitLab pour la phase de construction (build). Quels sont les types de commandes que vous devez inclure dans ce job ?

**Question 7 :** Expliquez comment vous pouvez configurer un job dans GitLab CI/CD pour exécuter des tests unitaires avec un framework comme Mocha ou Jest sur un projet Node.js.

**Question 8 :** Comment pouvez-vous configurer un job de tests dans un fichier .gitlab-ci.yml pour exécuter des tests unitaires sur un projet Node.js utilisant Jest ?

**Question 9 :** Quels sont les différents déclencheurs de pipeline dans GitLab et comment configurez-vous un pipeline qui ne s'exécute que lors d'un push sur la branche develop ?

**Question 10 :** Quels sont les différents types de trigger que vous pouvez utiliser dans GitLab pour exécuter des pipelines CI/CD et comment définiriez-vous un pipeline qui ne s'exécute que sur certaines branches, comme main ou master ?

#### **2. Intégration de SonarQube dans le pipeline GitLab**

**Question 1 :** Quel est le rôle de SonarQube dans un pipeline CI/CD et comment l’intégrer dans un projet GitLab en utilisant le fichier .gitlab-ci.yml ?

**Question 2:** Expliquez le processus pour intégrer SonarQube dans un pipeline CI/CD GitLab. Quelle configuration devez-vous ajouter au fichier .gitlab-ci.yml pour utiliser SonarQube ?

**Question 3 :** Expliquez les étapes à suivre pour connecter SonarQube à GitLab et quelle configuration est nécessaire dans SonarQube pour recevoir les analyses de votre projet.

**Question 4 :** Quel fichier de configuration devez-vous créer pour que SonarQube puisse analyser votre code ? Donnez un exemple de configuration de base.

**Question 5 :** Qu'est-ce qu'un sonar-project.properties et comment l'utiliser dans un projet GitLab ? Montrez un exemple de configuration basique.

**Question 6 :** Quelle est l'importance du token d'authentification dans l'intégration de SonarQube avec GitLab et comment l'ajoutez-vous dans votre pipeline ?

**Question 7 :** Quel rôle joue le token d'authentification de SonarQube et comment l'utiliser dans le fichier .gitlab-ci.yml pour permettre à GitLab d'interagir avec SonarQube ?

**Question 8 :** Que doit contenir le fichier sonar-project.properties pour que SonarQube analyse correctement un projet ? Donnez un exemple de ce fichier.

**Question 9 :** Une fois l'intégration de SonarQube terminée, comment consultez-vous les résultats des analyses de code dans l'interface web de SonarQube ? Quels types de problèmes peuvent être détectés par SonarQube ?

**Question 10 :** Après avoir intégré SonarQube dans votre pipeline, comment consulter les résultats de l'analyse dans l'interface de SonarQube et quels types de problèmes SonarQube identifie-t-il pendant l'analyse ?

**Exercice : Configuration d'un dépôt Git et d'un pipeline CI/CD dans GitLab avec analyse de code**

**Question 1 : Initialiser un dépôt Git local dans votre projet**

Quel est la commande pour initialiser un dépôt Git dans un projet déjà existant ?

**Question 2 : Ajouter tous les fichiers du projet au dépôt Git et effectuer un premier commit**

Quelle commande permet d'ajouter tous les fichiers au dépôt Git et de faire un premier commit ?

**Question 3 : Créer un fichier .gitlab-ci.yml et définir les étapes du pipeline**

Quelle est la syntaxe pour définir les différentes étapes (stages) d'un pipeline CI/CD dans un fichier .gitlab-ci.yml ?

**Question 4 : Ajouter un job pour construire l'application dans la phase de build**

Quelle commande dans le fichier .gitlab-ci.yml permet de définir un job pour la phase de build, avec un script simple de construction ?

**Question 5 : Ajouter un job pour exécuter des tests unitaires avec pytest dans la phase de test**

Comment ajouter un job dans le fichier .gitlab-ci.yml pour exécuter des tests unitaires avec pytest dans la phase de test ?

**Question 6 : Intégrer SonarQube pour analyser la qualité du code dans la phase de test**

Quelle commande dans le fichier .gitlab-ci.yml permet d'intégrer SonarQube pour analyser la qualité du code dans la phase de test ?

**Question 7 : Ajouter un job pour déployer l'application en production seulement sur la branche main**

Quelle est la commande dans le fichier .gitlab-ci.yml pour déployer l'application uniquement sur la branche main ?

**Question 8 : Ajouter un outil de linting (comme ESLint pour JavaScript) dans le pipeline CI/CD**

Quelle est la commande pour ajouter un job de linting avec ESLint dans le fichier .gitlab-ci.yml ?

**Question 9 : Créer un job pour scanner les vulnérabilités des dépendances avec Snyk**

Quelle est la commande dans le fichier .gitlab-ci.yml pour ajouter un job utilisant Snyk pour analyser les vulnérabilités des dépendances ?

**Question 10 : Déployer l'application en staging uniquement si des fichiers dans le répertoire app/ ont été modifiés**

Quelle règle dans le fichier .gitlab-ci.yml permet de déclencher le déploiement en staging uniquement si des fichiers dans le répertoire app/ ont été modifiés ?

### Explication de l'exercice :

Chaque question dans cet exercice se concentre sur une étape clé de la configuration d'un pipeline CI/CD dans GitLab :

1. **Initialisation du dépôt Git** : Comprend les commandes pour initialiser et effectuer un premier commit.
2. **Configuration du fichier .gitlab-ci.yml** : Définition des étapes du pipeline, des jobs de build, de test et de déploiement, ainsi que l'intégration d'outils pour l'analyse de la qualité du code (SonarQube, ESLint, Flake8, etc.).
3. **Déploiement conditionnel** : Configure le déploiement sur des branches spécifiques ou en fonction des modifications apportées à des répertoires spécifiques.
4. **Sécurité et tests** : Intègre des outils comme OWASP Dependency-Check ou Snyk pour l'analyse des vulnérabilités.

Chaque réponse est une commande ou un extrait de configuration GitLab CI/CD qui permet de configurer un pipeline robuste et adapté aux pratiques DevOps.