SONARQUBE

1) Introduction

Les codes écrits par les Devs peuvent contenir des variables inutilisées, de la duplication de code ou d'autres codes inutiles, mais cela n'impacte en rien la partie fonctionnelle du livrable. Ces erreurs sont généralement mises en évidence lors de la revue de leur code par le chef de projet technique ou par exemple, un collègue. Cela reste cependant compliqué de détecter toutes les erreurs et les potentielles vulnérabilités, car il peut avoir énormément de lignes de code à relire et cela peut prendre énormément de temps**. On peut utiliser alors un analyseur statique de code comme SonarQube.**

2) Qu'est-ce que SonarQube ?

SonarQube est un outil de révision de code automatique autogéré qui vous aide systématiquement à fournir un code propre. Il s'intègre à votre flux de travail existant et détecte les problèmes dans votre code pour vous aider à effectuer des inspections continues du code de vos projets. L'outil supporte plus de 30 langages de programmation différents et s'intègre à votre pipeline CI et à votre plateforme DevOps pour garantir que votre code répond à des normes de qualité élevées.

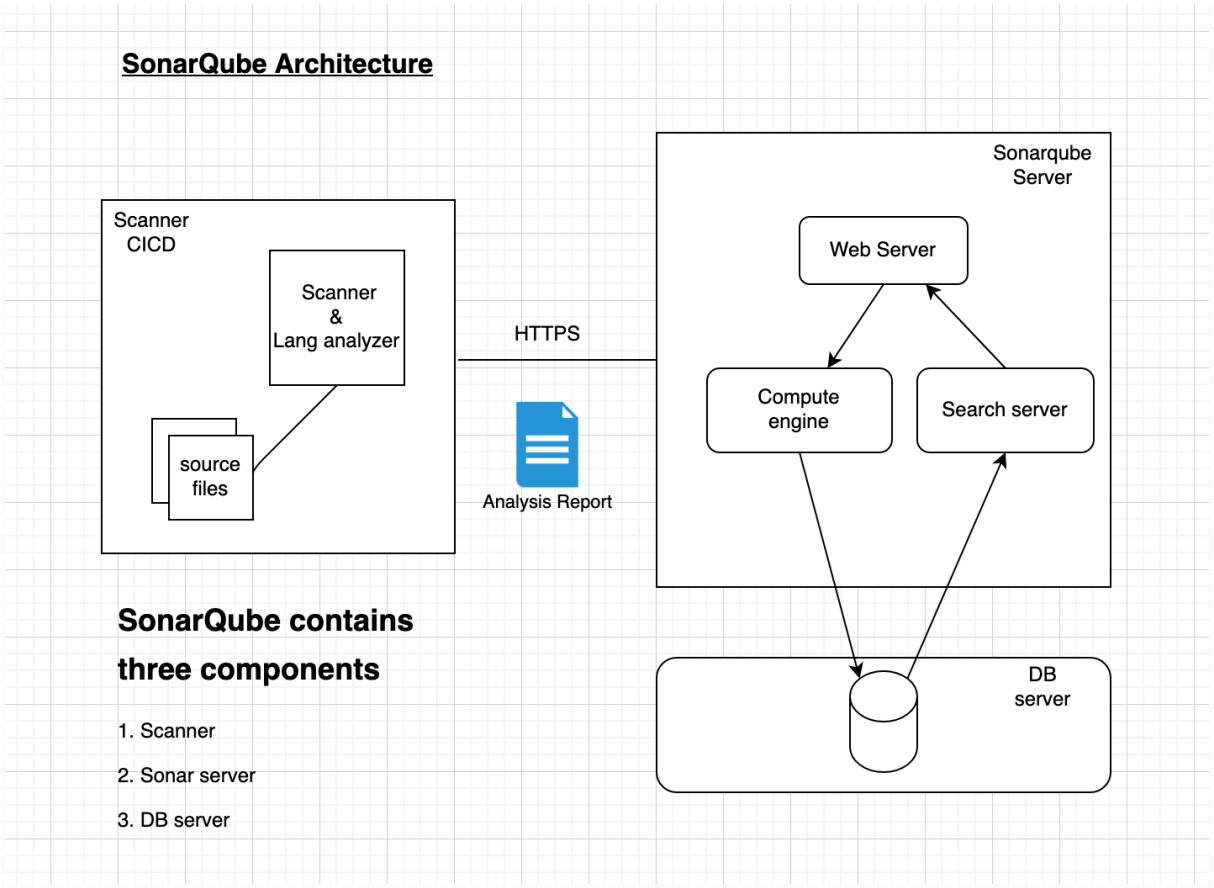
Un code propre est un code fiable, sécurisé, maintenable, lisible et modulaire. Cela s'applique à tout sorte de code : code source, code de test, infrastructure en tant que code, code de collage, scripts, etc.

Ce logiciel permet de vérifier le code source soumis par le développeur dans le but de faire un contrôle de qualité très poussé. Sonar couvre les 7 axes de la qualité d'un code :

- la sécurité et la maintenabilité
- Les potentiels bugs
- Le respect des standards de nommage comme pour le nom des variables
- Le taux de duplication de code présent
- Le taux de couverture du code par les tests unitaires
- La documentation du code
- La complexité du code à comprendre dans le futur par un développeur et qui peut entraîner une erreur à l'avenir

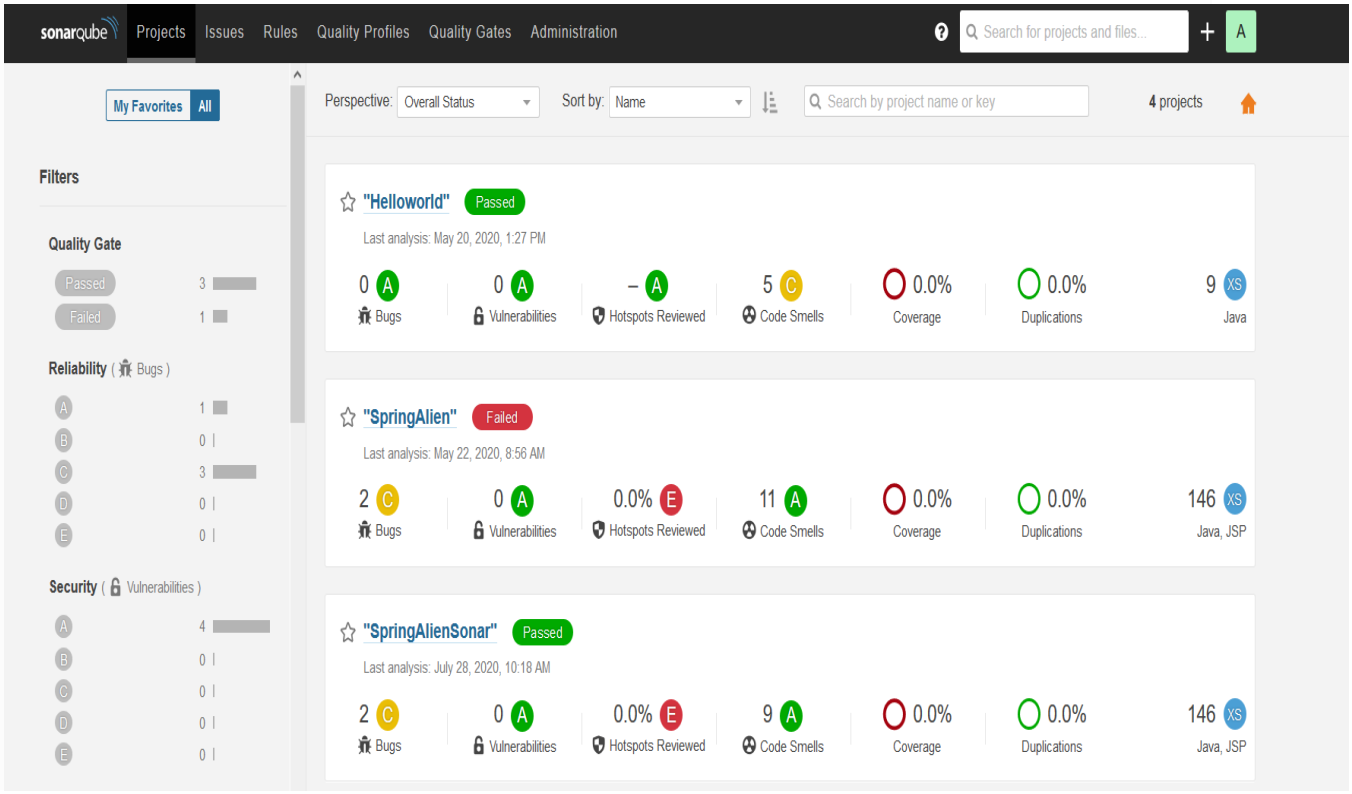
De plus, SonarQube nous rend la tâche facile en agissant comme un gestionnaire de qualité en générant des rapports de qualité qui sont stockés dans la base de données du serveur Sonar. Ces rapports sont fort utiles dans le cadre de projets où la qualité est très attendue par le client et peut être utile lors des présentations.
Il existe d'autres outils d'analyse comme Coverity, veracode, codeScene et raxis

3) Architecture de SonarQube



- **Le SonarScanner** : il est l'analyseur ; il accède au code source, notamment à travers les outils de build comme Maven, Ant et Gradle. Après avoir récupéré le code source, il récupère les rules au niveau du SonarQube Server pour faire l'analyser et renvoie les resultat au server. Le SonarScanner est indépendant du SonarQube Server. Par exemple, ce scanner peut être intégré à Jenkins en installant le plugin correspondant ce qui permettra de lancer l'analyse de SonarQube directement à travers Jenkins lors d'un build.
  On peut voir que SonarQube peut facilement s'intégrer dans le cadre de l'intégration continue.
- **Le SonarQube Server** comporte des règles de programmation liée à un langage particulier parmi ceux proposés par le logiciel. Par la suite, ce serveur comporte également une interface graphique sous forme de tableaux de bord qui offre la possibilité de consulter le détail de l'analyse.
- **Une base de données** où l'on stocke les différents rapports des analyses effectuées par le SonarScanner avant d'être affiché sur l'interface graphique. Cette base de données permet de consulter l'évolution des indicateurs à différentes périodes du projet.

En prod il est préférable d'installer sonarqube sur son propre server et de le séparer de la base de données (MySQL ou PostgreSQL), il ne faut pas utiliser la base de données par défaut (h2).



4) Avantages de SonarQube

- La fiabilité du code (nombre de bugs potentiels trouvés lors de l'analyse)
- Il agit comme un outil de gestion de qualité
- S'inscrit aisément dans la chaîne de le CI
- Améliore la productivité de l'équipe en les poussant à coder de la meilleure manière possible

5) Concepts

Rules : Une norme ou une pratique de codage qui devrait être suivie. Le non-respect des règles de codage peut entraîner des bugs, des vulnérabilités etc.

Profile de Qualité : Un ensemble de règles.

Vulnérabilité : Un problème lié à la sécurité qui représente une porte dérobée pour les attaquants.

Les Security Hotspot sont des zones qui requiert une revue de code manuelle car ce sont des parties de code sensible qui pourraient être considérées comme une vulnérabilité. Le développeur pourra ensuite faire le choix de sécuriser cette zone ou non.