CI-CD

- GitLab
  - Intégration SonarQube
- Jenkins
- SonarQube

GitLab

https://gitlab.darkmat.fr

Intégration SonarQube

Lors de la création d'un projet, il peut être souhaitable d'intégrer SonarQube afin de déléguer l'analyse du code du projet.

Configurer SonarQube pour GitLab

Vous devez configurer un jeton d'intégration entre GitLab et SonarQube.

Créer un jeton d'accès dans GitLab

1. Connectez-vous à votre instance GitLab (`gitlab.darkmat.fr`).
2. Accédez à vos paramètres de profil (`Profile Settings`) en cliquant sur votre avatar en haut à droite.
3. Dans le menu latéral, sélectionnez **Access Tokens**.
4. Créez un nouveau jeton avec les permissions `api` et `read_user`. Nommez le jeton par exemple `sonarqube-access-token`.
5. Copiez le jeton généré, car vous en aurez besoin plus tard.

Configurer un jeton dans SonarQube

1. Connectez-vous à SonarQube (`sonar.darkmat.fr`) avec un compte administrateur.
2. Accédez à **Administration > Security > Users** et créez un nouvel utilisateur dédié à GitLab, ou utilisez votre propre compte.
3. Créez un **token** d'accès pour cet utilisateur sous **My Account > Security > Tokens**. Donnez-lui un nom descriptif, comme `gitlab-sonarqube-token`.
4. Copiez le jeton généré.

Configurer votre projet GitLab pour l'analyse SonarQube

Vous allez maintenant configurer le fichier `.gitlab-ci.yml` dans votre projet pour intégrer SonarQube.

Ajouter des variables d'environnement dans GitLab

1. Allez dans les **Settings** du projet GitLab que vous voulez intégrer à SonarQube.
2. Sous **CI/CD > Variables**, ajoutez les variables suivantes :
   - `SONAR_HOST_URL` : `http://sonar.darkmat.fr` (ou `https://sonar.darkmat.fr` si vous utilisez HTTPS)
   - `SONAR_TOKEN` : Le jeton d'accès généré depuis SonarQube
   - `CI_PROJECT_KEY` : Un identifiant unique pour le projet, que vous définirez également dans SonarQube

Configurer le fichier `.gitlab-ci.yml`

Ajoutez ou modifiez le fichier `.gitlab-ci.yml` de votre projet GitLab pour inclure un job SonarQube. Voici un exemple basique :

```
stages:
  - build
  - test
  - sonarqube

sonarqube-check:
  stage: sonarqube
  image:
    name: sonarsource/sonar-scanner-cli:latest
    entrypoint: [""]
  script:
    - sonar-scanner
      -Dsonar.projectKey="$CI_PROJECT_KEY"
      -Dsonar.sources=.
      -Dsonar.host.url="$SONAR_HOST_URL"
      -Dsonar.login="$SONAR_TOKEN"
  only:
    - main
```

Configurer le projet dans SonarQube

1. Dans SonarQube, allez dans **Projects > Create Project**.
2. Donnez un nom et un identifiant au projet (`projectKey`). Utilisez le même `projectKey` que celui défini dans le fichier `.gitlab-ci.yml` et les variables CI/CD de GitLab.
3. SonarQube vous fournira un autre token spécifique pour ce projet (si besoin). Vous pouvez l'ajouter dans GitLab, mais généralement, le token utilisateur suffit.

Exécuter une pipeline CI/CD dans GitLab

1. Faites un `commit` et `push` sur votre dépôt GitLab pour déclencher la pipeline CI/CD.
2. GitLab va exécuter l'analyse SonarQube en utilisant le job configuré.
3. Une fois le job terminé, vous pourrez voir les résultats de l'analyse dans SonarQube sous votre projet.

Jenkins

https://jenkins.darkmat.fr

SonarQube

https://sonar.darkmat.fr