

SonarCloud Integration with GitHub Actions

Overview

This document provides a step-by-step guide to integrating **SonarCloud** with **GitHub Actions** for code quality analysis of the *Wednesday Adventures* project. The integration ensures automated SonarCloud scans on code pushes and pull requests.

Key Concepts

What is SonarCloud? SonarCloud is a cloud-based static code analysis tool that helps maintain **code quality** by detecting **bugs, vulnerabilities, and code smells**. It integrates seamlessly with CI/CD pipelines to ensure that every code change meets predefined quality standards.

Why Use SonarCloud in CI/CD?

- **Automated Code Reviews:** Provides immediate feedback on code quality.
- **Security & Reliability:** Identifies potential security vulnerabilities.
- **Maintainability:** Detects duplicated code, complex logic, and poor coding practices.
- **Seamless GitHub Integration:** Reports are visible directly in pull requests.

Requirements

Prerequisites

- **SonarCloud Account** ([Sign up here](#))
- **GitHub Repository** configured for CI/CD
- **GitHub Secrets**
 - **SONAR_TOKEN:** Authentication token for SonarCloud
 - **GITHUB_TOKEN:** Authentication token for GitHub actions

Steps to Set Up SonarCloud Integration

1. Configure SonarCloud Project

- Create a **SonarCloud project** and note the **projectKey** and **organization** name.
- Generate a **SONAR_TOKEN** from SonarCloud and store it in **GitHub Secrets**.

2. GitHub Actions Workflow

This workflow triggers SonarCloud analysis on code pushes and pull requests.

File Location: `.github/workflows/sonarcloud-analysis.yml`

```
name: SonarCloud Analysis
on:
  push:
    branches:
      - 100187927-WA-Jira18-CIPipeline # Trigger analysis on this branch
      - main
  pull_request:
    branches:
      - 100187927-WA-Jira18-CIPipeline # Trigger analysis on PRs to this
branch
      - main
jobs:
  sonarcloud:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v3
      - name: SonarCloud Scan
        uses: SonarSource/sonarqube-scan-action@v4.1.0
        env:
          SONAR_TOKEN: ${ secrets.SONAR_TOKEN }
      - name: Set up Node.js
        uses: actions/setup-node@v3
        with:
```

```

    node-version: '18.19.1'
    cache: 'npm'

- name: Install dependencies (Backend)
  run: |
    cd project-backend
    npm install
- name: Install dependencies (Frontend)
  run: |
    cd frontend
    npm install
- name: Run SonarCloud analysis
  if: always() # Run this step even if previous steps fail
  uses: sonarsource/sonarcloud-github-action@master
  env:
    GITHUB_TOKEN: ${ secrets.GITHUB_TOKEN } # GitHub token for authentication
    SONAR_TOKEN: ${ secrets.SONAR_TOKEN } # SonarCloud token
  with:
    args: >
      -Dsonar.branch.name=main # Analyze this branch
- name: SonarCloud Scan
  run: sonar-scanner -Dsonar.branch.name=main -X

```

3. SonarCloud Configuration File

Create the following `sonar-project.properties` file in the root of your project.

File Location: `sonar-project.properties`

```

# Required metadata
sonar.projectKey=SumathiD20_Wednesday_Adventures
sonar.organization=sumathid20
sonar.host.url=https://sonarcloud.io

# Optional but recommended
sonar.projectName=Wednesday_Adventures
sonar.projectVersion=1.0

# Source code location

```

```
sonar.sources=project-backend,frontend
sonar.exclusions=**/node_modules/**,**/tests/**

# Encoding
sonar.sourceEncoding=UTF-8
```

Explanation of Workflow Steps

Step	Description
Checkout Code	Retrieves the latest repository code.
Set up Node.js	Configures Node.js (version 18.19.1) with npm caching.
Install Dependencies	Runs <code>npm ci</code> for both backend and frontend for consistency.
Run SonarCloud Analysis	Executes SonarCloud analysis and uploads results.
Final SonarCloud Scan	Runs an explicit SonarQube scan with debug logs enabled (<code>-X</code>).

How to Review SonarCloud Reports

1. Go to **SonarCloud Dashboard**: <https://sonarcloud.io>
2. Open your **project dashboard**.
3. View **Code Quality Metrics**: Bugs, Code Smells, Vulnerabilities.
4. Check **Pull Request Annotations** in GitHub for direct feedback.

Troubleshooting

1. SonarCloud Scan Fails

Error: `ERROR: Project not found` **Solution:** Ensure `sonar.projectKey` and `sonar.organization` are correct in `sonar-project.properties`.

2. Missing GitHub Secrets

Error: `SONAR_TOKEN not found` **Solution:** Add `SONAR_TOKEN` as a secret in the GitHub repository settings.

3. Node Module Issues

Error: `npm install fails` **Solution:** Use `npm ci` instead of `npm install` to ensure dependency consistency.

Conclusion

This setup ensures that every code change undergoes **automated quality analysis** in SonarCloud, improving **code reliability** and **maintainability**.