SonarQube 코드 분석 - Overview

이 문서는 SonarQube 코드 분석의 개요를 공유하기 위해 작성되었다.

개요

SonarQube 설치가 완료되면, 코드를 분석 하기 위한 Scanner 설치와 프로젝트 생성 준비가 된 것이다.
사용자는 언어 및 빌드도구에 적합한 Scanner를 설치하고 구성해야 한다. 지원하는 Scanner는 다음과 같다.

- Gradle - SonarScanner for Gradle
- .NET - SonarScanner for .NET
- Maven - use the SonarScanner for Maven
- Jenkins - SonarScanner for Jenkins
- Azure DevOps - SonarQube Extension for Azure DevOps
- Ant - SonarScanner for Ant
- 이외 (CLI) - SonarScanner

SonarQube Integration은 널리 사용되는 DevOps 플랫폼(**GitHub Enterprise 및 GitHub.com, BitBucket Server, Azure Devops Server 및 Azure DevOps Services**)에서 지원된다.

다음 사이트에서 언어별로 SonarQube에서 지원하는 Rule을 확인할 수 있다.

- SonarSource Code Analyzers Rules Explorer

SonarQube 분석이 실행되는 컴퓨터에서 바이러스 백신 스캐너를 실행하면 예기치 않은 동작이 발생할 수 있다.

프로젝트는 첫 번째 분석에서 SonarQube에서 자동으로 생성된다. 그러나 첫 번째 분석 전에 프로젝트에 대한 일부 구성을 설정해야 하는 경우,
관리 옵션 또는 프로젝트 생성 권한이 있는 사용자에게 표시되는 **+** 메뉴 항목을 통해 프로비저닝할 수 있다.

- SonarQube 프로젝트 관리하기 항목 참고

분석은 무엇을 생산하는가?

SonarQube는 버전에 따라 최대 29 개의 다른 언어를 분석 할 수 있다. 이 분석의 결과는 코딩 규칙이 깨진 경우, 품질(Quality) 측정(Measures) 및 문제(Issues) 가 될 것이다.
그러나 분석되는 내용은 언어에 따라 다르다.

- 모든 언어에서 "blame" 데이터는 지원되는 SCM 공급자로부터 자동으로 가져온다.
  - Git 및 SVN은 자동으로 지원되고, 다른 공급자는 추가 플러그인이 필요하다.
- 모든 언어에서 소스 코드의 정적 분석이 수행된다.
  - Java 파일, COBOL 프로그램 등
- 특정 언어의 경우 정적 분석은 컴파일 된 코드에서 수행되어야 한다.
  - (Java의 .class 파일, C #의 파일 .dll 등

모든 파일이 분석되는가?

기본적으로 분석 중에 SonarQube 버전에서 인식하는 파일만 프로젝트에 로드된다. 예를 들어 SonarQube Community Edition을 사용하는 경우, Java 및 JavaScript의 분석을 포함하지만, C ++은 포함하지 않는 않으므로 모든 .java, .js 파일은 로드되지만 .cpp 파일은 무시된다.

Branch 및 Pull request는 연동 여부?

*Developer Edition*에는 프로젝트의 Branch 및 Pull request을 분석하는 기능과 Pull request 분석을 DevOps Platform 인터페이스에 자동으로 보고하는 기능이 추가 된다.

- Community Edition 지원 안함

분석 중에 어떤 일이 발생하나?

분석 중에 서버에서 데이터를 요청하고 분석에 제공된 파일이 분석되며, 결과 데이터는 보고서 형태로 마지막에 서버로 다시 전송된 다음 서버 측에서 비동기적으로 분석된다.

분석 보고서는 큐에 대기되고 순차적으로 처리되므로 분석 로그에 완료가 표시된 후 짧은 기간 동안 업데이트된 값이 SonarQube 프로젝트에 표시되지 않을 수 있다. 그러나 프로젝트 이름 오른쪽의 프로젝트 홈페이지에 아이콘이 추가되기 때문에 무슨 일이 일어나고 있는지 확인 가능하다.

blocked URL

처리가 완료되면 아이콘은 사라지지만, 어떤 이유로 분석 보고서 처리가 실패하면 다음과 같은 아이콘(Failed)으로 변경된다.

blocked URL

자주 묻는 질문(FAQ)

Q. "`java.lang.OutOfMemoryError: GC overhead limit exceeded`."와 함께 분석 오류가 발생.

A. 프로젝트가 너무 크거나 Scanner가 기본 메모리 할당으로 분석하기에는 너무 복잡하다는 것을 의미한다. 이 문제를 해결하려면 분석을 실행하는 프로세스에 더 큰 Heap Memory를 할당해야 한다.
일부 CI 엔진은 필요한 값을 지정하는 입력을 제공할 수 있다. (예: Jenkins 작업에서 Maven 빌드 단계를 사용하여 분석을 실행하는 경우)
그렇지 않으면 Java 옵션을 사용하여 더 높은 값을 설정한다. Java 옵션 설정에 대한 자세한 내용은 환경에 따라 다르므로 여기서는 생략한다.

Q. "Analysis errors out with `PKIX path building failed`" 와 함께 분석 오류가 발생.

A. SonarQube 서버가 HTTPS 및 사용자 지정 SSL 인증서로 구성되어 있으나, 인증서가 Scanner 시스템의 JVM에는 올바르게 구성되지 않아서 발생한 오류이다.
SonarQube 범위에서 벗어난 구성으로, 서버 인증서를 알 수 없어서 제공된 신뢰 저장소(truststore)에서 유효성을 검사할 수 없다.
SonarQube 서버 인증서를 Java 신뢰 저장소에 추가해야 한다. 자세한 내용은 오라클의 설명서 참조.

참조 링크

- Overview | SonarQube Docs