**CI/CD pipelines with static code analysis**

A static code analysis tool inspects your codebase through the development cycle, and it's able to identify bugs, vulnerabilities, and compliance issues without actually running the program.

**What is static code analysis?**

Static code analysis is a practice that allows your team to automatically detect potential bugs, security issues, and, more generally, defects in a software's codebase. Thus, we can view static analysis as an additional automated code review process. Let's examine this analogy more in detail.

**Static Code Analysis tools:**

Many commercial and free static code analyzers support a vast plethora of programming languages. One of the most famous is Sonarqube.

Sonarqube is open-source software for continuous inspection of code quality. It performs automatic reviews with static analysis on more than 20 programming languages. It can spot duplicated code, compute code coverage, code complexity, and finds bugs and security vulnerabilities. In addition, it can record metrics history and provides evolution graphs via a dedicated web interface.

**Integrate SonarQube scan into the pipeline**

Once our project is created and configured, we can automatically trigger a code analysis, adding a step in our CD pipeline.

**you can start an analysis by running this command.**

**sonar-scanner**

-Dsonar.projectKey=[PROJECT\_KEY]

-Dsonar.sources=.

-Dsonar.host.url=[LOAD\_BALANCER\_URL]

-Dsonar.login=[LOGIN\_KEY]

-Dsonar.qualitygate.wait=true

-Dsonar.qualitygate.wait=true