# SonarQube

🡪sonar source

🡪 created by java

**SonarQube** is an open-source platform for continuous inspection of code quality. It analyzes code to detect bugs, vulnerabilities, and code smells (areas of suboptimal code structure) and helps developers improve their code’s quality, maintainability, and security. SonarQube supports multiple programming languages and integrates well with CI/CD pipelines, making it a popular tool for maintaining high-quality code in software projects.

**Key Features of SonarQube**

1. **Code Analysis**: SonarQube performs static code analysis, identifying issues like bugs, security vulnerabilities, and code smells, which can lead to maintenance challenges.
2. **Multi-Language Support**: SonarQube supports over 20 programming languages, including Java, Python, JavaScript, C++, and many more, making it suitable for polyglot codebases.
3. **Quality Gates**: SonarQube allows teams to set "quality gates" — thresholds for various metrics, such as code coverage, complexity, and bug counts. If these gates are not met, the code can be flagged, helping enforce quality standards.
4. **Security Analysis**: SonarQube provides security-focused rules to help teams identify vulnerabilities and follow best practices for secure code, such as avoiding injection flaws or misconfigured permissions.
5. **Detailed Reporting and Dashboards**: SonarQube has a comprehensive reporting feature that provides in-depth metrics and visualizations, allowing developers and teams to track quality trends over time.
6. **Integration with CI/CD Pipelines**: SonarQube integrates easily with CI/CD tools like Jenkins, GitLab, and Azure DevOps. This enables automatic code analysis every time code is pushed or a pull request is created, helping catch issues early in the development process.
7. **Developer-Friendly Feedback**: SonarQube gives actionable feedback to developers through annotated code views, indicating precisely where issues are located in the code.

**Example Use Case**

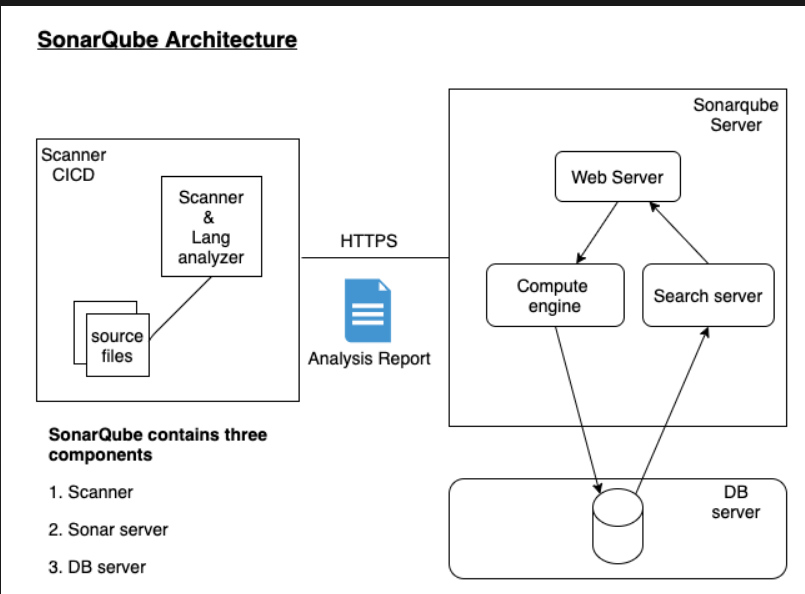
SonarQube is often integrated into a DevOps pipeline to enforce code quality checks automatically before a build is deployed. If the analysis detects issues that fail the quality gate, the build can be stopped, ensuring only high-quality, secure code is promoted to production.

**Why Use SonarQube?**

By automating code reviews and enforcing best practices, SonarQube helps reduce technical debt, improve code readability and maintainability, and reduce the risk of bugs and security issues making it to production.

Dev(write tons of code)🡪sonarqube(code analysis (bugs, security venerability’s, code smells), multi-language support(20+), quality gates(metrices such as code coverage,complexity and bugs etc), developer-friendly etc) 🡪 build the code🡪testing (qa,uat)🡪 production

Sonar scanner: it is used to scanner the code



Sonar is a platform designed to check or inspect the code quality continuously

Scanner 🡪 scan the code and check the code(based on the rules)

By default SonarQube has set of rules for each programming language like java python, go etc(20+)

* Based on this rules will be applied to check the code quality (rules == checklist)

Sonarqube setup(2core cpu, 4 gb ram, 30 gb storage) 🡪 in server (java need to be installed)

**Community Edition**: Best for individuals or teams that only need essential quality checks.

**Developer Edition**: Adds more robust security analysis, making it suitable for small-to-mid-sized teams.

**Enterprise Edition**: A better choice for larger teams that manage multiple projects and need aggregated reporting.

**Data Center Edition**: Ideal for organizations that need high availability, failover, and support for distributed teams.