Q: What is SonarQube?

Ans:

SonarQube is an open-source framework developed by SonarSource for continuous inspection of code quality to conduct automated reviews of 20 + programming languages with static code analysis to identify bugs, code bad smells and security vulnerabilities.

Q: Why to use SonarQube?

Ans:

SonarQube increases productivity by allowing development teams to detect and muzzle duplication and redundancy of code. SonarQube makes it easier for team members to reduce application size, code complexity, time and cost of maintenance, and make code easier to read and understand.

Q: What is difference between SonarQube And SonarLint?

Ans:

SonarLint:

- SonarLint exists only in the IDE (IntelliJ, Visual Studio and Eclipse).
- Its aim is to provide immediate feedback as you type in your code.
- It focuses on what code you add or update for this function.
- SonarLint is an agent that allow us to connect with this SonarQube and execute the analysis remotely.

SonarQube:

- SonarQube is a central server which performs full analysis (triggered by the different SonarQube scanners).

- The purpose is to give your code base a 360 Â° view of the quality. To this end, it periodically analyzes all of the source lines of your project.

Both SonarLint and SonarQube depend on the same analyzers for static source code-most of which are written using SonarSource technology.

Q: Is SonarQube Replacement for Checkstyle, PMD, FindBugs?

Ans:

By default for Java projects, Sonar will run CheckStyle, FindBugs and PMD, as well as a few other "plugins" such as Cobertura . The main added advantage is that it stores the history in a database.
These 3 tools are used by Sonar as plugins and the data from all three of these tools is applied with a value that displays graphs.

Q: What is difference between Sonar Runner and Sonar Scanner?

Ans:

The old name for "Scanner" is "Runner."
All you need to know about the different SonarQube Scanners is available in the Scanners section of the official documentation.
You can use below option,if you are stuck to Java 7:

- SonarQube Runner (sonar-runner) up to version 5.5 of SonarQube
- SonarQube Scanner (sonar-scanner) 2.6.1

Q: What is sonarqube quality profile?

Ans:

Quality Profiles are a core component of SonarQube, since they are where you define a set of rules that, when violated, should raise issues on your codebase (example: methods should not have a Cognitive Complexity higher than 15). Quality Profiles are defined for each language.

What is SonarQube

A:Sonar is a web based code quality analysis tool for Maven based Java projects. It covers a wide area of code quality check points which include: Architecture & Design, Complexity, Duplications, Coding Rules, Potential Bugs, Unit Test etc.

Q: Why use SonarQube ?

A:Sonar covers the 7 sections of code quality

- Architecture and Design
- Unit tests
- Duplicated code
- Potential bugs
- Complex code
- Coding standards
- Comments

Q: What are the advantages of using SonarQube?

A:

- SonarQube is open source
- SonarQube supports for various languages like Java, C#
- SonarQube reports for duplicate code, unit testing, code coverage, code complexity historical
- We can integrate SonarQube with build tools like ant, gradle
- SonarQube has Eclipse plugin like Sonarlint
- SonarQube supports external plugins like plugin for ldap

Q: What are Quality Profiles in SonarQube?

A: The Quality Profiles service is central to SonarQube, since it is where you define your requirements by defining sets of rules (ex: Methods should not have a Cognitive Complexity greater than 15).
Ideally, all projects will be measured with the same profile for any given language, but that's not always practical. For instance, you may find that:

The technological implementation differs from one application to another (for example, different coding rules may apply when building threaded or non-threaded Java applications). You want to ensure stronger requirements on some of your applications (internal frameworks for example).

Q: What are Quality Gates in SonarQube?
A: A quality gate is the best way to enforce a quality policy in your organization. It's there to answer ONE question: can I deliver my project to production today or not? In order to answer this question, you define a set of Boolean conditions based on measure thresholds against which projects are measured. For example:

No new blocker issues Code coverage on new code greater than 80% Etc.
Ideally, all projects will be verified against the same quality gate, but that's not always practical. For instance, you may find that:
Technological implementation differs from one application to another (you might not require the same code coverage on new code for Web or Java applications). You want to ensure stronger requirements on some of your applications (internal frameworks for example). Etc.
Which is why you can define as many quality gates as you wish. Quality Gates are defined and managed in the Quality Gates page found in the top menu.

Q: What is role of database in SonarQube?
A: Sonar uses a Derby or H2 as default database. When running Sonar, it says that these databases may only be used for evaluation. We can change this default database and use our custom DB.

Q: How to create reports in SonarQube?
A: To create reports using SonarQube

```
mvn clean install

mvn sonar:sonar -Dsonar.issuesreport.html.enable=true
```

What are rules in SonarQube?

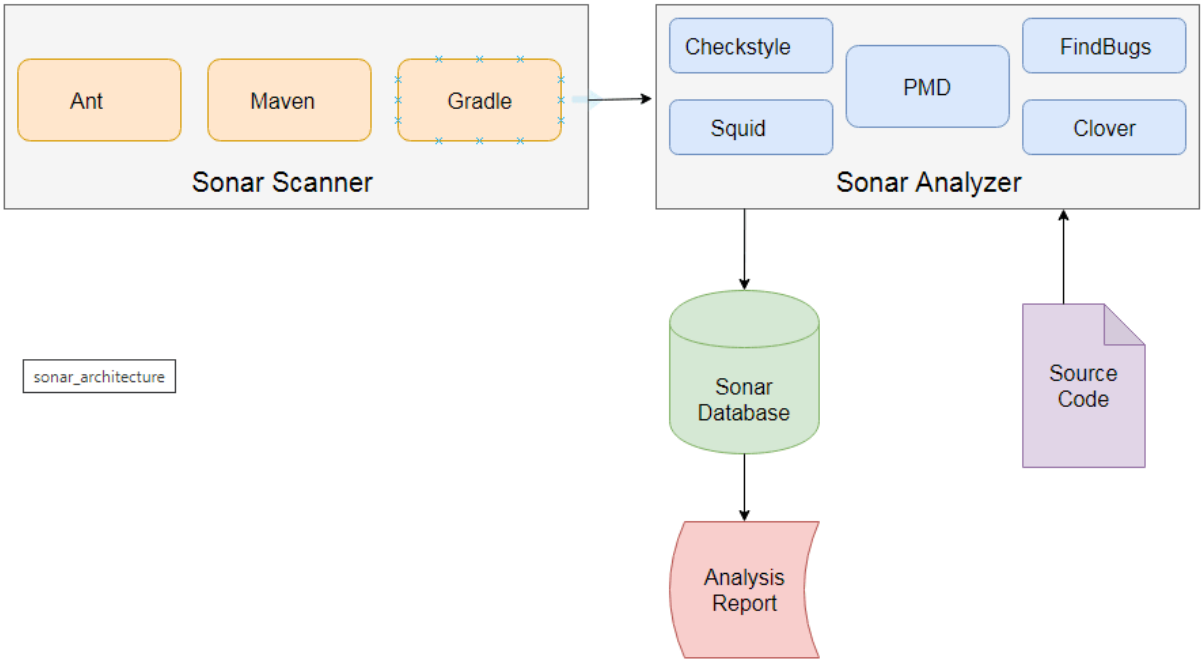
SonarQube executes rules on source code to generate issues.
There are four types of rules: Code Smell (Maintainability domain)
Bug (Reliability domain)
Vulnerability (Security domain)
Security Hotspot (Security domain)

What does SonarQube And SonarLint differ ?

- SonarQube has a server associated with it.
- SonarQube is a central server that processes full analyses which is triggered by the various SonarQube Scanners. Its purpose is to give a 360° vision of the quality of your code base. For this, it analyzes all the source lines of your project on a regular basis.
- SonarQube is a server where you can host your projects and execute analysis
- Sonar lint works more like a plugin.
- SonarLint lives only in the IDE (IntelliJ, Eclipse and Visual Studio). Its purpose is to give instantaneous feedback as you type your code. For this, it concentrates on what code you are adding or updating.
- SonarLint is an agent that allow us to connect with this SonarQube and execute the analysis remotely.

Define Sonar Architecture ?

Here we will have a look at sonarqube architecture.



sonar_architecture