# **How to check code quality using SonarQube**

SonarQube is a powerful tool for continuously inspecting the quality of code. It performs automatic reviews with static analysis of code to detect bugs, code smells, and security vulnerabilities. Here’s a step-by-step guide on how to use SonarQube to check code quality:

**Step 1: Install SonarQube**

1. **Download SonarQube**: Go to the SonarQube downloads page and download the version appropriate for your operating system.
2. **Extract SonarQube**: Unzip the downloaded file to a directory of your choice.
3. **Install Java**: Ensure you have Java (Java Runtime Environment or JDK) installed. SonarQube requires Java 11 or 17.
4. **Start SonarQube**: Navigate to the bin directory inside your SonarQube installation and start SonarQube by running the appropriate script for your OS (StartSonar.bat for Windows, sonar.sh for Unix/Linux/MacOS).

**Step 2: Install and Configure SonarScanner**

SonarScanner is the tool to analyze your project and send the results to SonarQube.

1. **Download SonarScanner**: Get it from the SonarScanner downloads page.
2. **Configure SonarScanner**: After extracting, configure the sonar-scanner.properties file located in the conf directory. At a minimum, you should set the sonar.host.url to point to your SonarQube server.

**Step 3: Create a SonarQube Project**

1. **Access SonarQube**: Open your browser and navigate to http://localhost:9000 (default port).
2. **Log in**: The default credentials are usually admin/admin.
3. **Create a new project**: Go to the "Projects" section and create a new project. You will need to generate a unique token for the project, which will be used later.

**Step 4: Configure Your Project for Analysis**

In your project directory:

1. **Create a sonar-project.properties file**:

properties

Copy code

sonar.projectKey=your\_project\_key

sonar.projectName=Your Project Name

sonar.projectVersion=1.0

sonar.sources=.

sonar.host.url=http://localhost:9000

sonar.login=your\_generated\_token

1. **Customize properties as needed**: Adjust paths and settings according to your project structure.

**Step 5: Run SonarScanner**

1. **Navigate to your project directory**.
2. **Run SonarScanner**: Execute the sonar-scanner command in your terminal.

**Step 6: Review the Analysis Results**

1. **Access the SonarQube Dashboard**: Go to your SonarQube instance in the browser.
2. **Navigate to your project**: View the analysis results which include metrics, issues, and code smells.
3. **Review and address issues**: SonarQube provides detailed information on each issue, including where it is located in the code and how to potentially fix it.

**Continuous Integration (Optional)**

Integrate SonarQube with your CI/CD pipeline for continuous code quality checks. This can be done using plugins for Jenkins, GitHub Actions, GitLab CI, etc.

**Tips for Better Results**

* **Use Quality Profiles**: Customize or select quality profiles that fit your project’s needs.
* **Configure Exclusions**: If there are files or directories you want to exclude from analysis, specify them in your sonar-project.properties.
* **Regular Analysis**: Run SonarQube analysis regularly to keep track of code quality over time.

By following these steps, you can effectively use SonarQube to maintain and improve the quality of your codebase.