

Static Code Analysis Procedure

Introduction

Static code analysis is an integral step in software development which must be employed by developers to validate their code before merging with the development branch. In this way, any bugs or security vulnerabilities present in their work can be identified and eliminated at an early stage of development. In view of this, we will be exploring SonarQube, an industry-standard static code analysis tool, in this document and be explaining its implementation in our local machines. This tool will be integrated in our Software Development Lifecycle (SDLC) for both internal and external projects. SonarQube is used in place of Sonar Cloud as we are unable secure permission from A*STAR ITSS for connecting Sonar Cloud plugin with Azure repository.

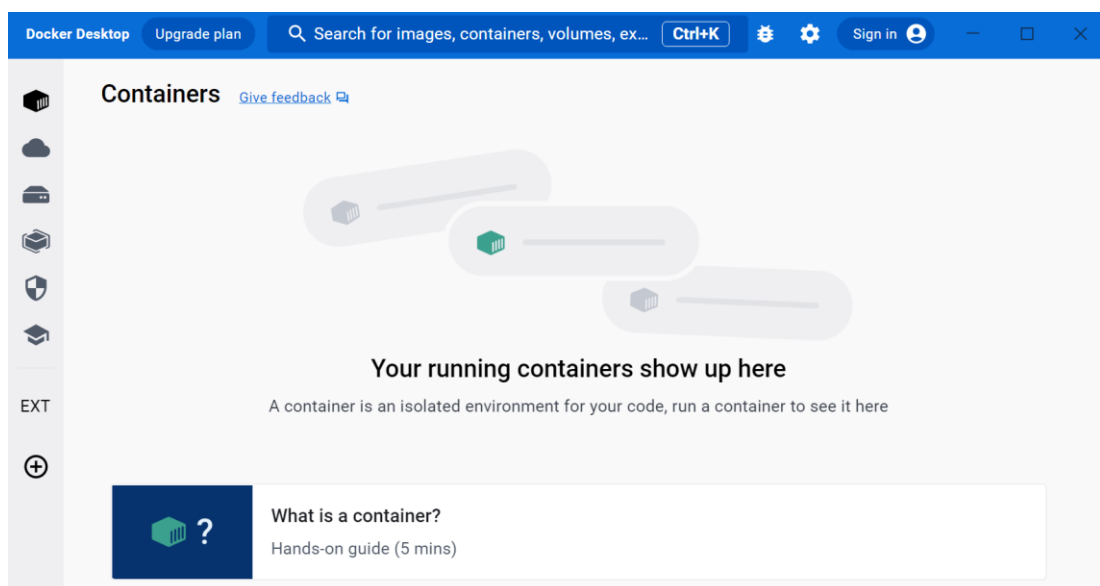
Description

SonarQube (formerly **Sonar**) is an open-source platform developed by SonarSource for continuous inspection of code quality to perform automatic reviews with static analysis of code to detect bugs and code smells on 29 programming languages. SonarQube offers reports on duplicated code, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security recommendations.

Steps for installing and scanning

These steps are meant for installing SonarQube in a developer's local machine with Windows Operating System. Steps may vary for other operating systems. Please note that steps 1 to 6, 14 and 15 are only meant for installing SonarQube the first time.

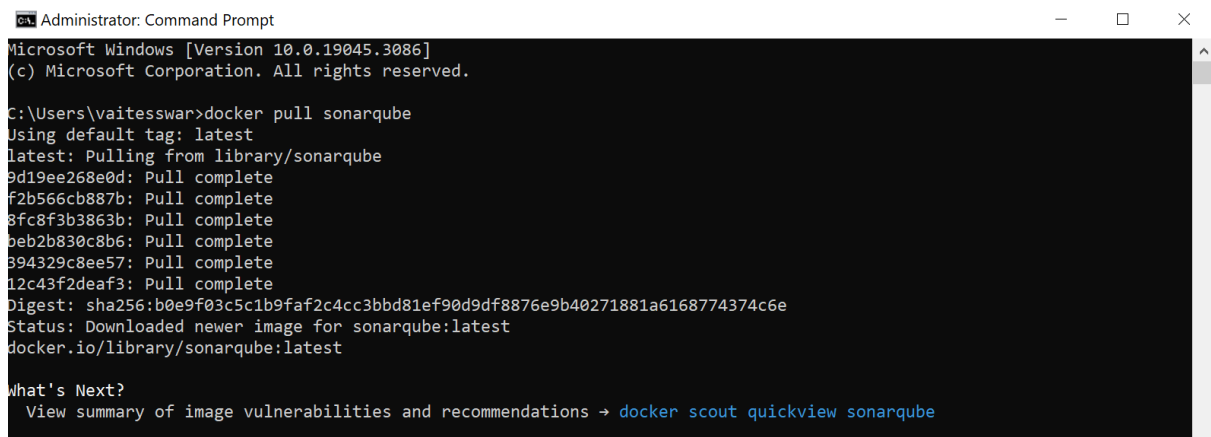
1. Open Docker Desktop



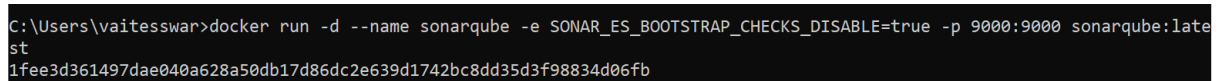
2. Open Command Prompt (run as Administrator)



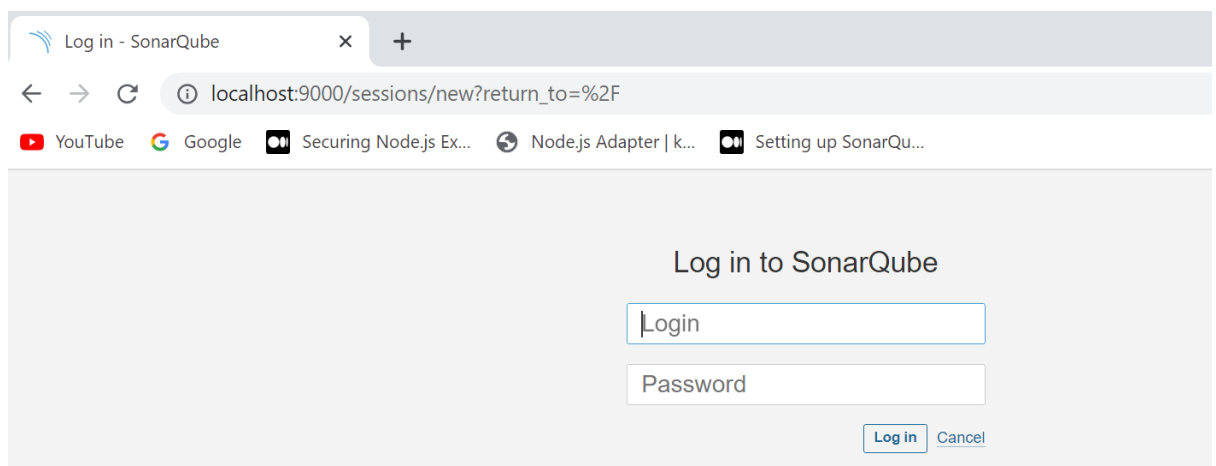
3. Run the command “docker pull sonarqube”



4. Run the command “docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest”



5. Log in to <http://localhost:9000/> (username: admin, password: admin)



6. Change to desired password.

Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required


Old Password *

New Password *

Confirm Password *


[Update](#)

7. Choose “Manually”.

[Projects](#)[Issues](#)[Rules](#)[Quality Profiles](#)[Quality Gates](#)[Administration](#)[More](#)[Q](#)


How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform. First, you need to set up a DevOps platform configuration.




From Azure DevOps

Set up global configuration




From Bitbucket Server

Set up global configuration




From Bitbucket Cloud

Set up global configuration



From GitHub


Set up global configuration



From GitLab

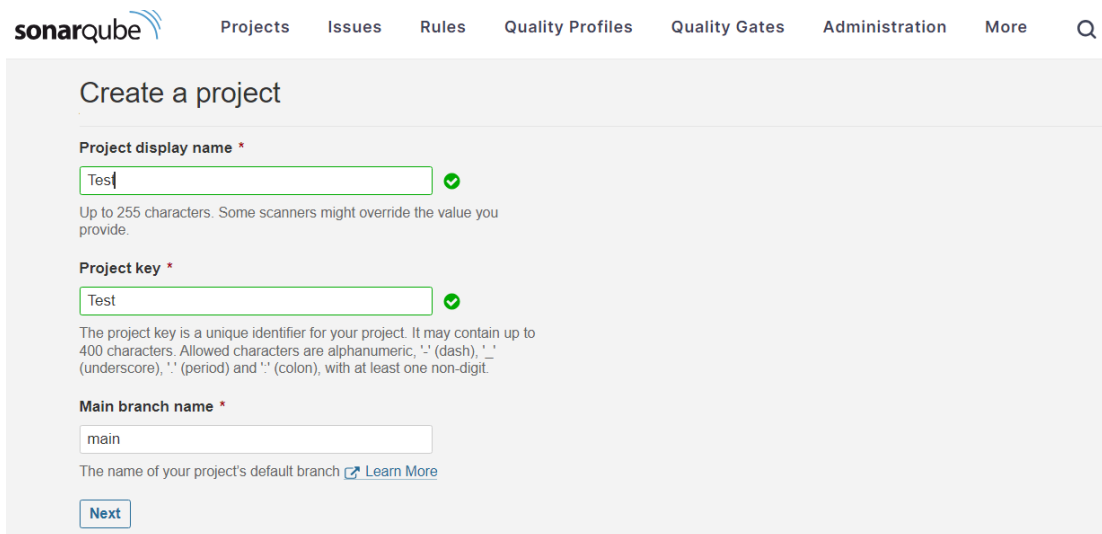
Set up global configuration

Are you just testing or have an advanced use-case? Create a project manually.



Manually

8. Assign a project display name and project key.



The screenshot shows the 'Create a project' form in SonarQube. The form has three main sections: 'Project display name', 'Project key', and 'Main branch name'. Each section has a text input field and a green checkmark icon indicating the value is valid. The 'Project display name' field contains 'Test' and has a tooltip that says 'Up to 255 characters. Some scanners might override the value you provide.' The 'Project key' field also contains 'Test' and has a tooltip that says 'The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.' The 'Main branch name' field contains 'main' and has a tooltip that says 'The name of your project's default branch' with a link to 'Learn More'. At the bottom of the form is a 'Next' button.

Create a project

Project display name *

Test

Up to 255 characters. Some scanners might override the value you provide.

Project key *

Test

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

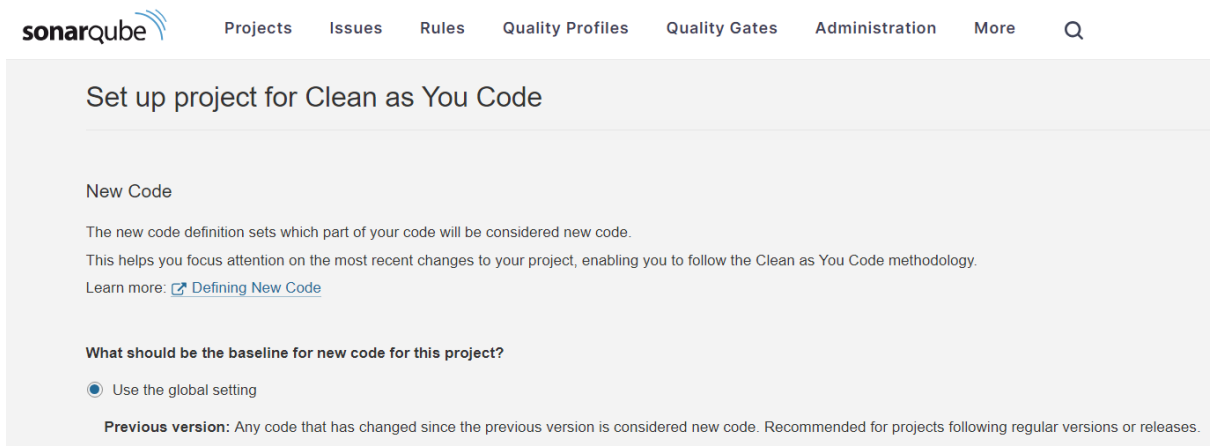
Main branch name *

main

The name of your project's default branch [Learn More](#)

[Next](#)

9. Select “Use the global setting” option.



The screenshot shows the 'Set up project for Clean as You Code' form in SonarQube. The form has a section titled 'New Code' with a description: 'The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)'. Below this is a section titled 'What should be the baseline for new code for this project?' with two radio button options: 'Use the global setting' (which is selected) and 'Define a specific setting for this project'. Below the radio buttons is a text block for the 'Previous version' option: 'Previous version: Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.'

Set up project for Clean as You Code

New Code

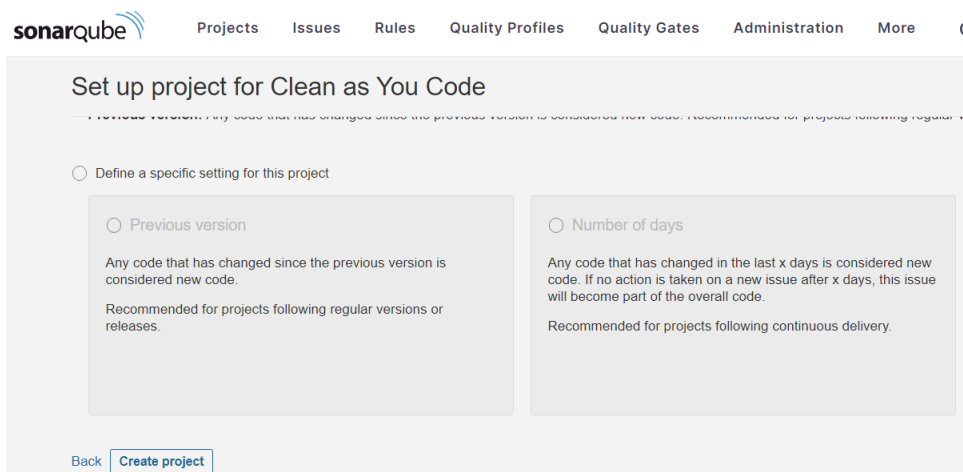
The new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. Learn more: [Defining New Code](#)

What should be the baseline for new code for this project?

☒ Use the global setting

Previous version: Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.

10. Click “Create project” button at bottom of the page.



The screenshot shows the 'Set up project for Clean as You Code' form in SonarQube, similar to the previous one, but with the 'Define a specific setting for this project' radio button selected. Below the radio buttons are two columns of options. The left column has a radio button for 'Previous version' with a description: 'Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.' The right column has a radio button for 'Number of days' with a description: 'Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code. Recommended for projects following continuous delivery.' At the bottom of the form are two buttons: 'Back' and 'Create project'.

Set up project for Clean as You Code

☐ Define a specific setting for this project

☐ Previous version

Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.

☐ Number of days


Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code. Recommended for projects following continuous delivery.


[Back](#) [Create project](#)


11. Choose “Manually”.


How do you want to analyze your repository?


Do you want to integrate with your favorite CI? Choose one of the following tutorials.



With Jenkins


With GitHub Actions



With Bitbucket Pipelines


With GitLab CI


With Azure Pipelines


Other CI

Are you just testing or have an advanced use-case? Analyze your project locally.


Locally

12. Generate token and click “Continue”.

Analyze your project

We initialized your project on SonarQube, now it's up to you to launch analyses!


1

Provide a token

Generate a project token

Token name ⓘ **Expires in**

Analyze "Test" 30 days Generate

 Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your [user account](#). See the [documentation](#) for more information.

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

2

Run analysis on your project

13. Choose the necessary options.

2

Run analysis on your project

What option best describes your build?

Maven

Gradle

.NET

Other (for JS, TS, Go, Python, PHP, ...)

What is your OS?

Linux

Windows

macOS

Download and unzip the Scanner for Windows

Visit the [official documentation of the Scanner](#) to download the latest version, and add the `bin` directory to the `%PATH%` environment variable

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner.bat -D"sonar.projectKey=Test" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_481339cc64125c4a9f3407e9146e12df4fb6804e"
```

 Copy

14. Download Sonar Scanner (.zip file) from “<https://docs.sonarqube.org/10.1/analyzing-source-code/scanners/sonarscanner/>” and extract it.

SonarScanner

By [SonarSource](#) | GNU LGPL 3 | [Issue Tracker](#)

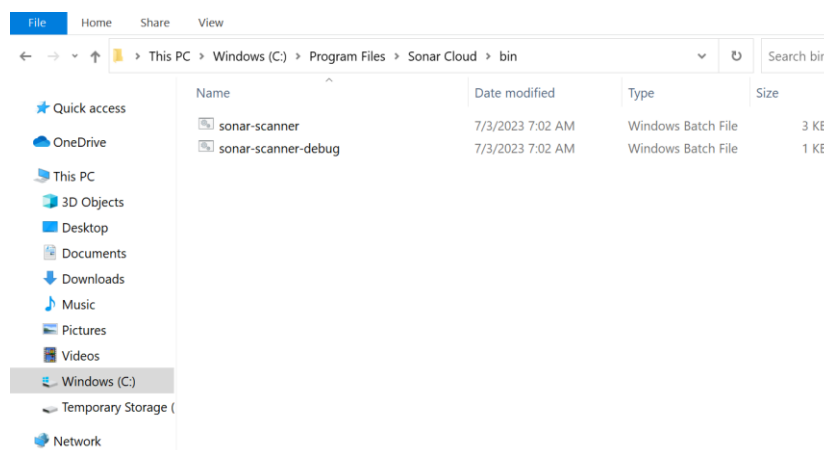
4.8 [Show more versions](#)

2022-02-06

Update embedded JRE 11 to the latest, bug fixes

[Linux 64-bit](#) [Windows 64-bit](#) [Mac OS X 64-bit](#) [Docker](#) [Any \(Requires a pre-installed JVM\)](#) [Release notes](#)

15. Transfer the extracted folder to “Program Files” and rename it to the desired new folder name. In this case, it is renamed as “Sonar Cloud”.



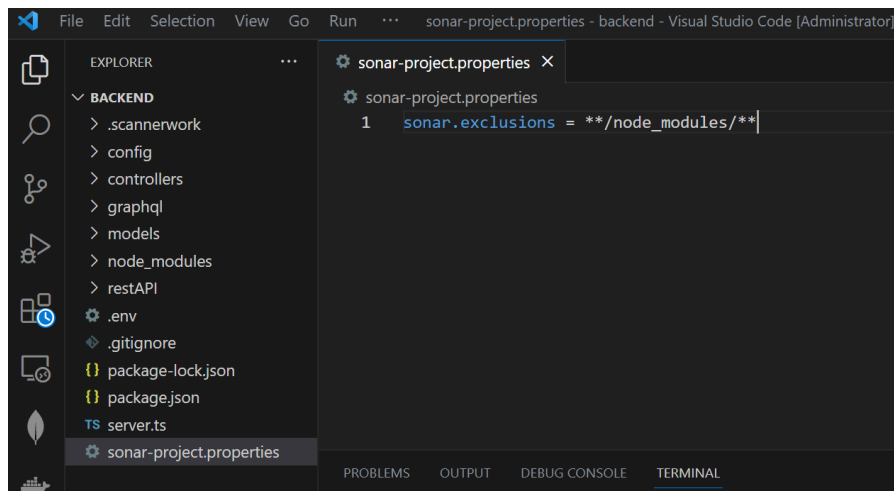
16. Go to command prompt and set the path to the sonar-scanner.bat file as follows which is found in bin folder. The command is “set PATH=%PATH%;”**<bin folder path>**”.

```
C:\Users\vaitesswar>set PATH=%PATH%; "C:\Program Files\Sonar Cloud\bin"
```

17. Change the location in command prompt to project folder (cd <PROJECT_PATH>).

```
C:\Users\vaitesswar>cd C:\Users\vaitesswar\Desktop\patientDoctor_application\backend
```

18. Add sonar-project.properties in project folder and enter "sonar.exclusions = **/node_modules/**". This ensures that node modules are **not included** for scanning.



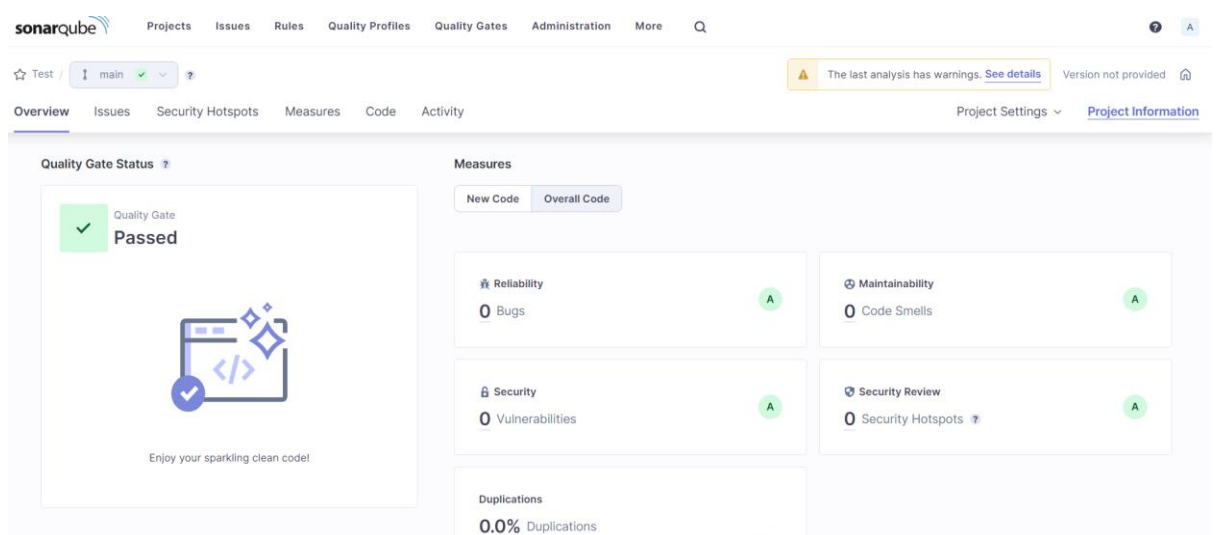
19. Run the command given in SonarQube page with token in command prompt. An example is shown below.

Execute the Scanner

Running a SonarQube analysis is straightforward. You just need to execute the following commands in your project's folder.

```
sonar-scanner.bat -D"sonar.projectKey=Test" -D"sonar.sources=." -D"sonar.host.url=http://localhost:9000" -D"sonar.token=sqp_481339cc64" Copy
```

20. Upon completion, the SonarQube page will automatically refresh to display the results.



21. If issues are found by the tool, the developer has to rectify them based on the corrections suggested by the tool until all issues are rectified. An example is illustrated below.

