Sonarcloud

TABLE OF Contents

[Introduction 3](#_Toc120610908)

[Sonarcloud 3](#_Toc120610909)

[Why Sonarcloud? 3](#_Toc120610910)

[What is Code Quality? 4](#_Toc120610912)

[Sonarcloud configuration 5](#_Toc120610913)

[For Angular Project 11](#_Toc120610917)

[For .Net Project 14](#_Toc120610918)

[Conclusion 16](#_Toc120610919)

Meena Ramji N

# Introduction

While we all strive to improve the quality of our code, assessing the results of that effort remains a challenge. Sonarcloud provides a standardized way to measure code quality and the progress of your team. It has quickly become the industry standard for code analysis, especially on projects we are involved with. Your team can then access a quick and simple interface to evaluate their code and take informed decisions on how to proceed. Isn’t that exciting?

# Sonarcloud

* Sonarcloud is the leading online service to catch Bugs and Security Vulnerabilities in your Pull Requests and throughout your code repositories. Sonar Cloud follows the concept of [Clean As You Code](https://sonarcloud.io/documentation/improving/clean-as-you-code/). The core idea is that you focus your attention and effort on new code. SonarCloud analyses your code on each new commit and alerts you to any code quality problems and vulnerabilities. This allows you to address the issues right away and ensure that all new code added to the project is always clean.  It is the #11 ranked solution in AST(Application Security Testing) tools.
* SonarCloud is a cloud-based code analysis service designed to detect code quality issues in 29 different programming languages, continuously ensuring the maintainability, reliability and security of your code. Code quality is always good using sonar cloud. Thousands of rules to track down hard-to-find bugs and quality issues.

# why Sonarcloud?

The main feature of Sonarcloud are,

* Super-fast analysis
* Automatic analysis
* Shared, unified configurations
* Sonarlint IDE integration
* Dozens of languages and frameworks
* Native integration with DevOps Platforms

**This whitepaper will help you how to configure Sonar Code-Quality Checks in Github Action and how to map your repository then execute the report.**

## What is Code Quality?

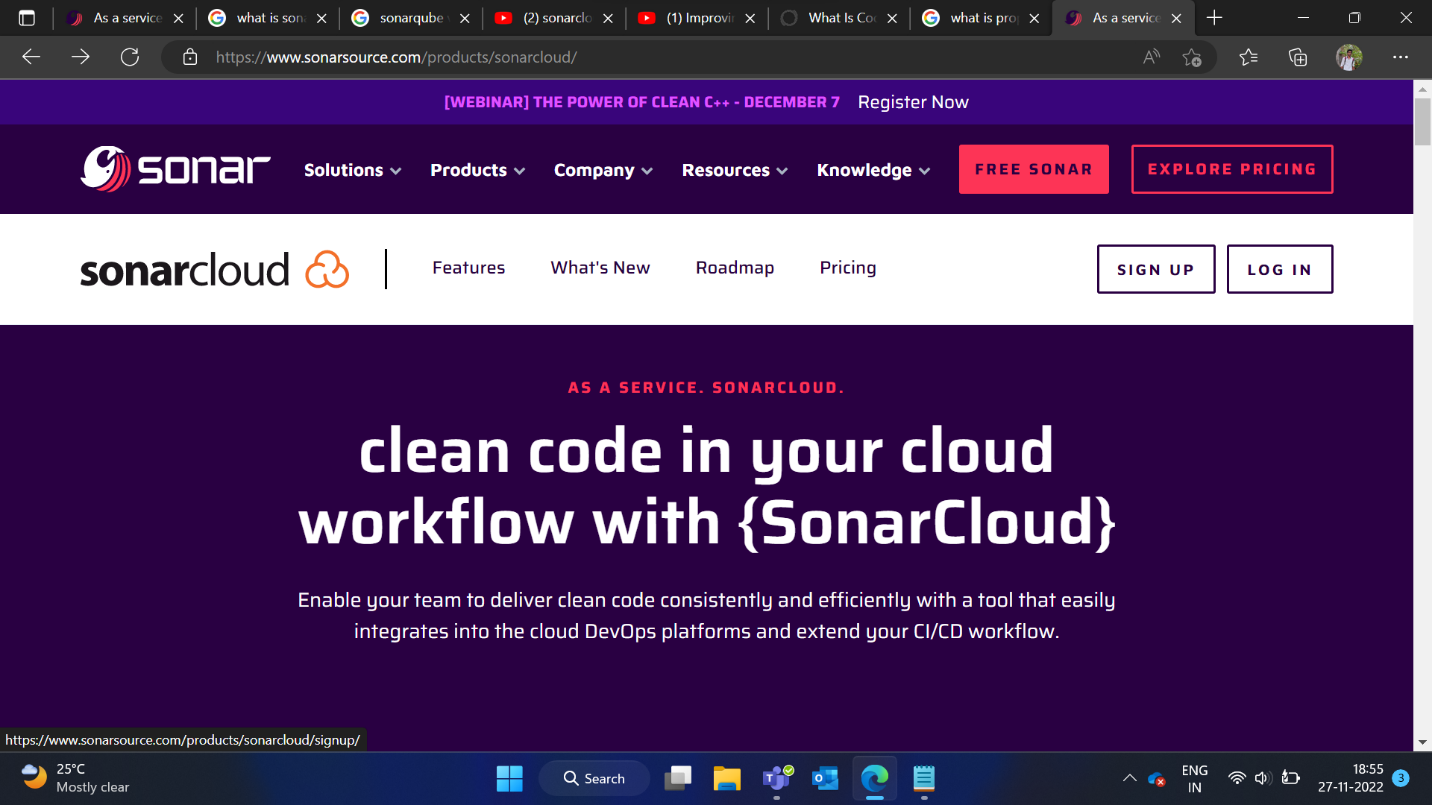
* Quality code defines the code that functions as intended for end-users without any deficiencies.
* Some automated tools can also analyze source code and provide a code quality score based on a number of metrics that measure complexity and functionality.
* Writing clean code - involves a number of best practices during writing the code. By following standard conventions, reducing complexity, improving old code whenever new code is added, and more, developers/testers can eliminate code smells and build a higher-quality, more secure product.

**Code quality** is graded according to five core parameters:

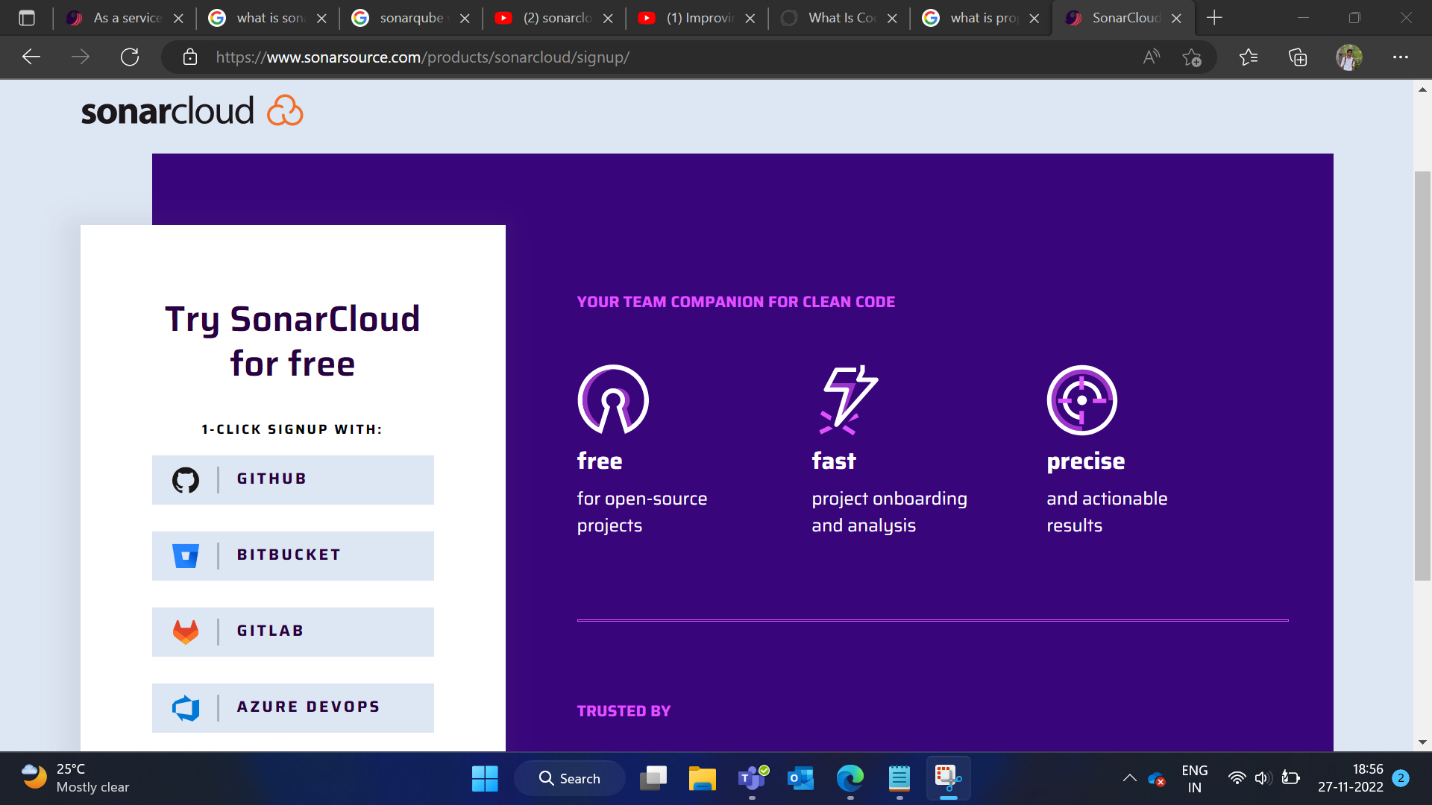
1. **Bugs**: issues that can make your code unreliable
2. **Vulnerabilities**: issues that can put your code at risk of attack
3. **Code smells**: issues that make your code harder to read
4. **Coverage**: how much of your code is covered by tests
5. **Duplications**: how many times a bit of code is repeated across the code base

# sonarcloud configuration

* The first thing we will obviously need is to have a SonarCloud account. To get one, just go to their website [https://sonarcloud.io](https://sonarcloud.io/)  click signup if you are not an existing user



* Signup/login to SonarCloud using your GitHub account.



* Next, you have to authorize SonarCloud.

A screenshot of a computer

Description automatically generated with medium confidence

* Now, You can import your GitHub organization for analyzing your projects in SonarCloud by clicking on import button.

Graphical user interface, text, application

Description automatically generated

* I choose my personal organization. SonarCloud will be installed as a GitHub App for that organization then you can grant SonarCloud access to your repository.

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence

* In SonarCloud you can now create an organization.

A screenshot of a computer

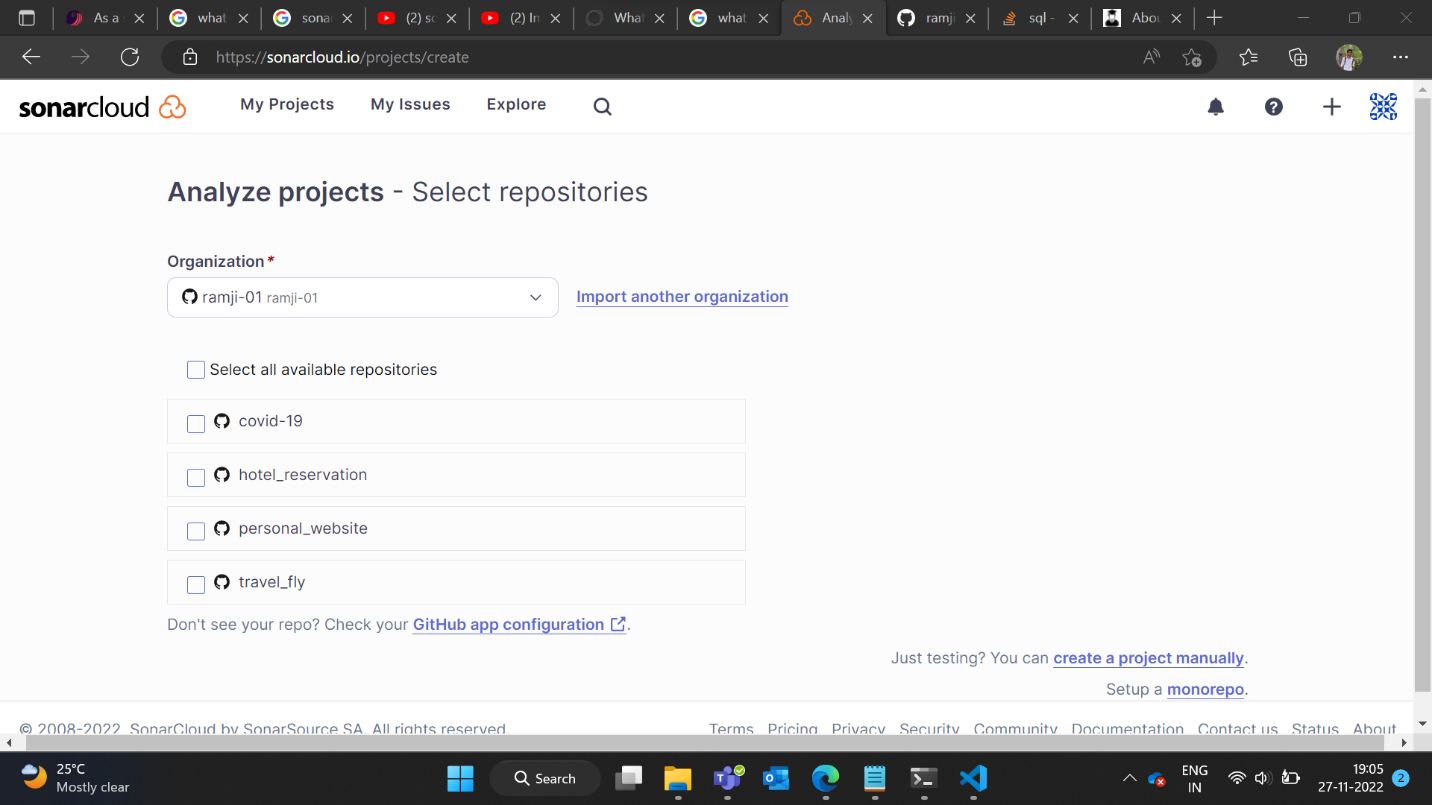
Description automatically generated

* It offers two plans, One is free plan which allows only public repository project to analyze in their website. Another one is paid plan which allows private repository to analyze.

Graphical user interface, text

Description automatically generated

* Select the repository that you want to analyze.



* After clicking the respective project, it asks to choose analysis method, click Github Actions(Recommended)

Graphical user interface, application

Description automatically generated

## STEP 1 : Create a GitHub Secret

* In your GitHub repository, you need to create a token by entering the shown values in below screenshot. So, GitHub can access SonarCloud.

Graphical user interface

Description automatically generated with medium confidence

* Navigate to your github repository, settings -> secrets -> action -> add new secret

Graphical user interface

Description automatically generated with medium confidence

## STEP 2 : Create or Update a .github/workflows/build.yml file

* You can choose your build option with respective technology that you developed a project



* The workflow file, usually declared in .github/workflows/build.yml, looks like:

Text

Description automatically generated

The above code is the base configuration to run a Sonarcloud analysis on our master branch and pull requests.

## STEP 3 : Create a sonar-project.properties file

* Create a configuration file in the root directory of the project and name it sonar-project.properties

Text

Description automatically generated

# For Angular Project

To find your project key and organization in your project, go to your project -> In the left sidebar, click information -> In the right pane, there is a project key and organization key values, you can copy paste those values in your file.

Graphical user interface, text, application, chat or text message

Description automatically generated

Build workflow for my project is shown in the below screenshot,

Text

Description automatically generated

Property file looks like below screenshot,

Graphical user interface, text, application

Description automatically generated

Now after analysing your project it shows up like this,

Graphical user interface, text, application

Description automatically generated

It shows analyzed results values for respective code quality parameters. If you click the bugs, it displays what are the bugs in our code.

Graphical user interface, text, application

Description automatically generated

It also shows a description as why is this an issue in our code and suggests alternative way to fix that bug. Approximate time for solving that bug also displayed.

Graphical user interface, text, application

Description automatically generated

If you click the code smells, it displays issues that make your code harder to read.

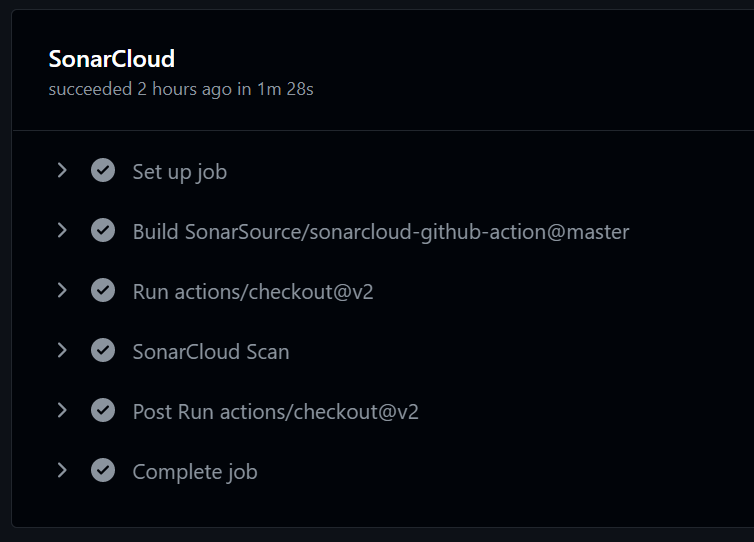
Graphical user interface, text, application, email

Description automatically generated

If you have done any changes in your code, you can track the pipeline action workflow(build.yml) in action tab in your repository

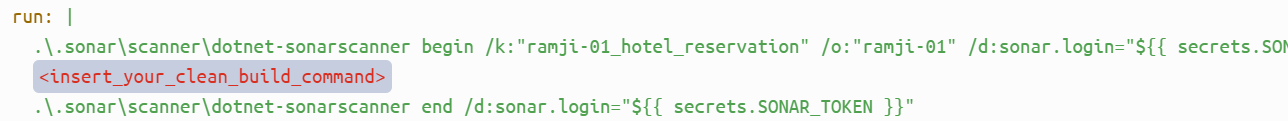
A screenshot of a computer

Description automatically generated with medium confidence



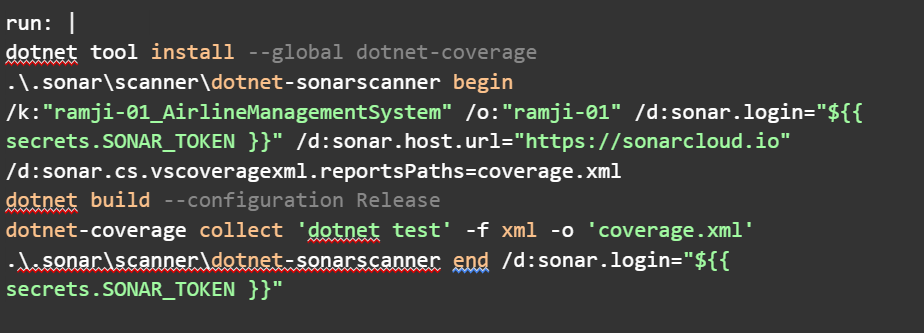
# For .Net Project

* For .net, it is also same process but in workflow file we insert our clean build command,



* SonarCloud supports .NET test coverage tools by dotnet-coverage Code Coverage and other tools are also available. Lets see how it shows the coverage by this tool. To use [dotnet-coverage](https://docs.microsoft.com/en-us/dotnet/core/additional-tools/dotnet-coverage), you must install it as a global dotnet tool

Using this tool, your build script would look something like this:



We use the -f xml parameter to specify that the output format is in XML.

Now after analysing your project it shows up like this,

Graphical user interface, application

Description automatically generated

It shows analyzed results values for respective code quality parameters. It displays 40.0% coverage as this percentage code in my project is covered by tests. If you click the coverage, it shows how much code is covered by each file and if you select any file you can see what codes are covered/not covered.

If you click the coverage parameters, it shows how much of your code is covered by tests.

Graphical user interface, text, application

Description automatically generated

Green marker means this code is fully covered by tests.

Graphical user interface, text, application, email

Description automatically generated

Red marker means this code is not covered by tests.

# Conclusion

To analyze our code, we've mapped our repository to Sonarcloud. So, the overall impression is that SonarCloud does provide useful insights on how well your team is progressing towards a more secure, more maintainable code base and, as such, is a very valuable addition to a developer’s toolbox.