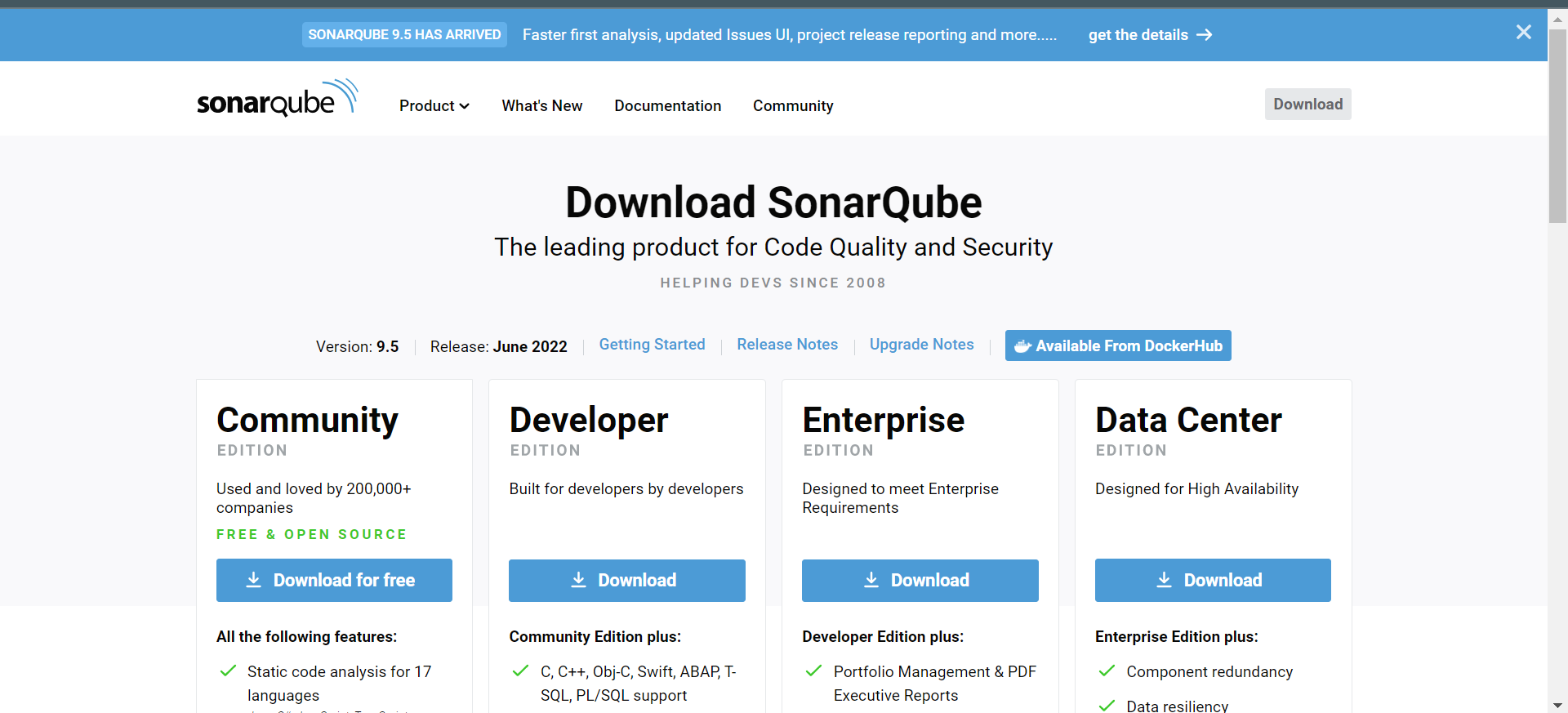
SonarQube with GitHub

The documentation is all about how we can configure SonarQube locally with GitHub using GitHub actions by writing a Yaml file.

Installation of SonarQube and Sonar Scanner in local system.

Step 1: URL for downloading SonarQube <https://www.sonarqube.org/downloads/>

URL for downloading Sonar Scanner <https://docs.sonarqube.org/latest/analysis/scan/sonarscanner/>

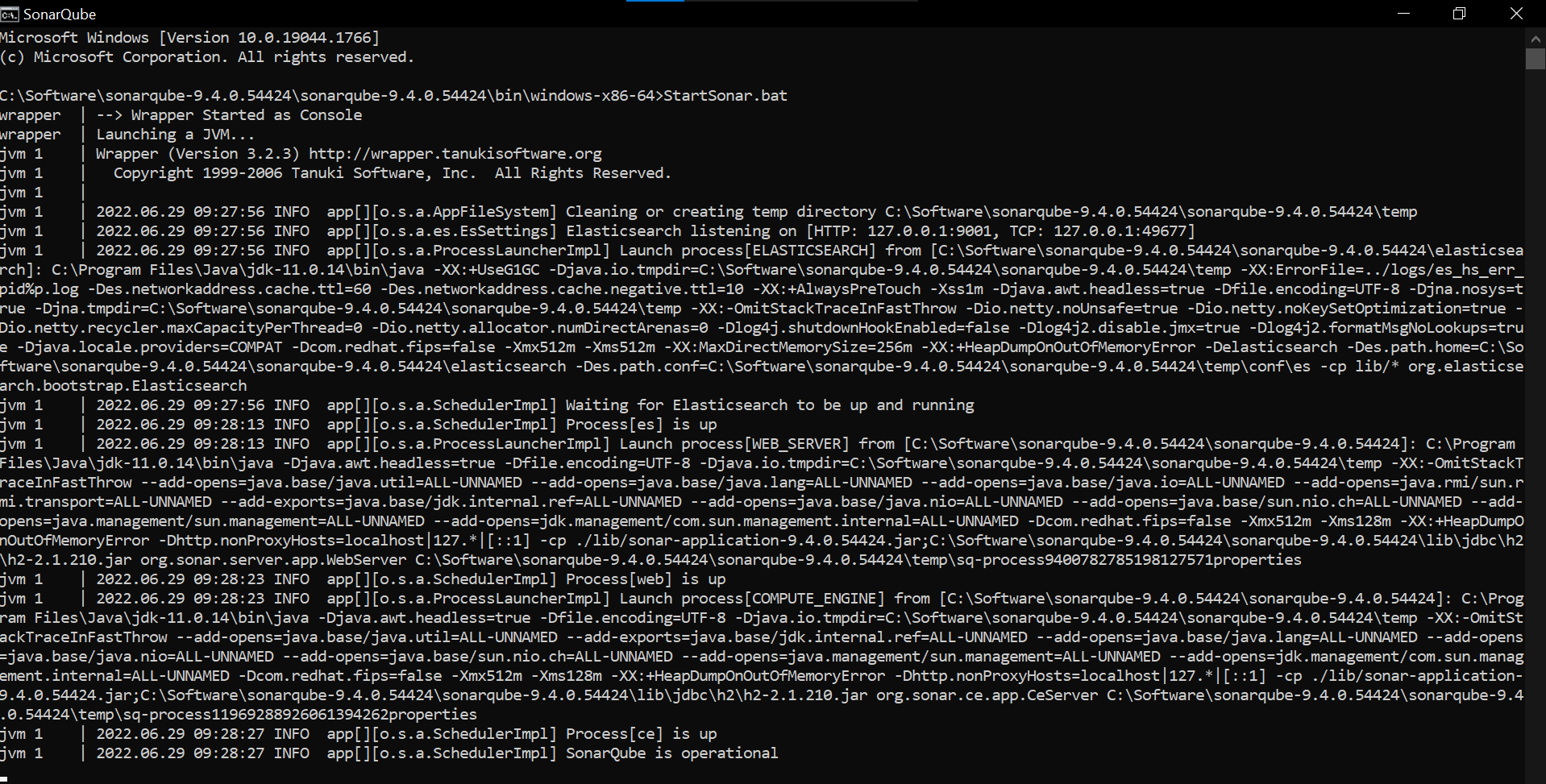


Select the community edition for free and open source and download. It will be downloaded in Zip file extract that zip file store it in hard disk.

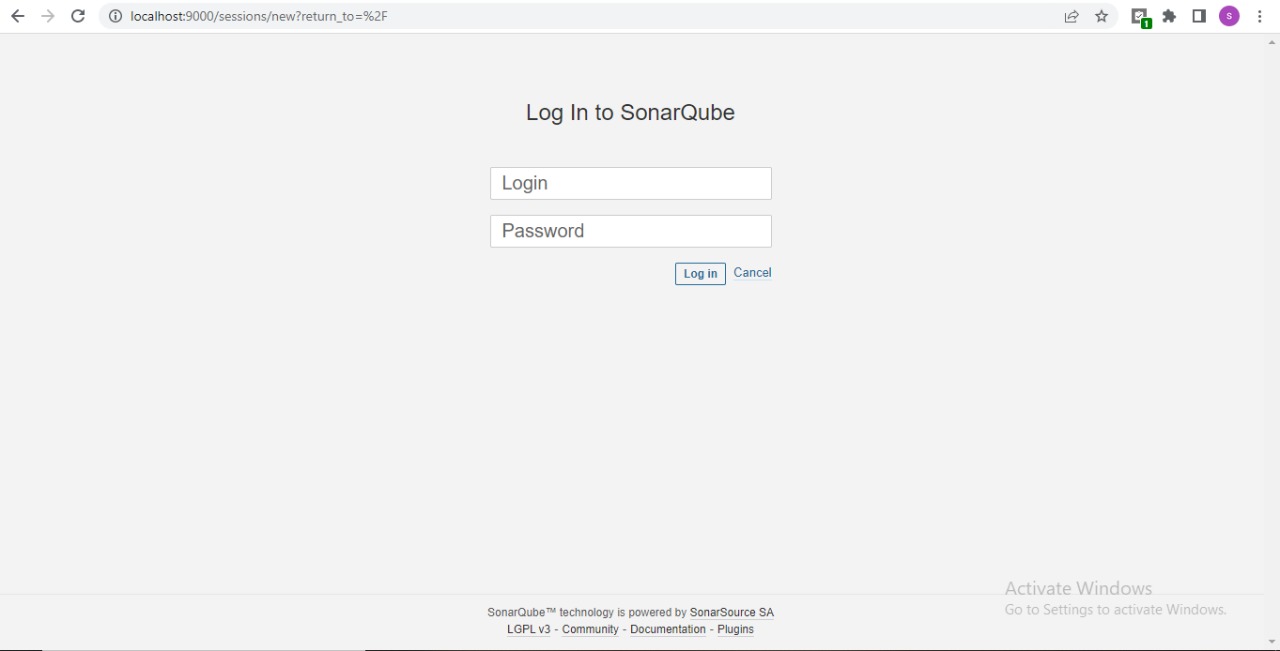
You need to install JDK 11 for SonarQube and set the environmental variables in settings.

Next Go to SonarQube>bin>windows-x86-64>StartSonar.

You can even use Command Prompt for Starting SonarQube by running the command StartSonar.bat in this SonarQube>bin>windows-x86-64 path



Mean while Open the <http://localhost:9000/> port in Chrome once the Sonar is Up the poet will open and ask for SonarQube Credentials.



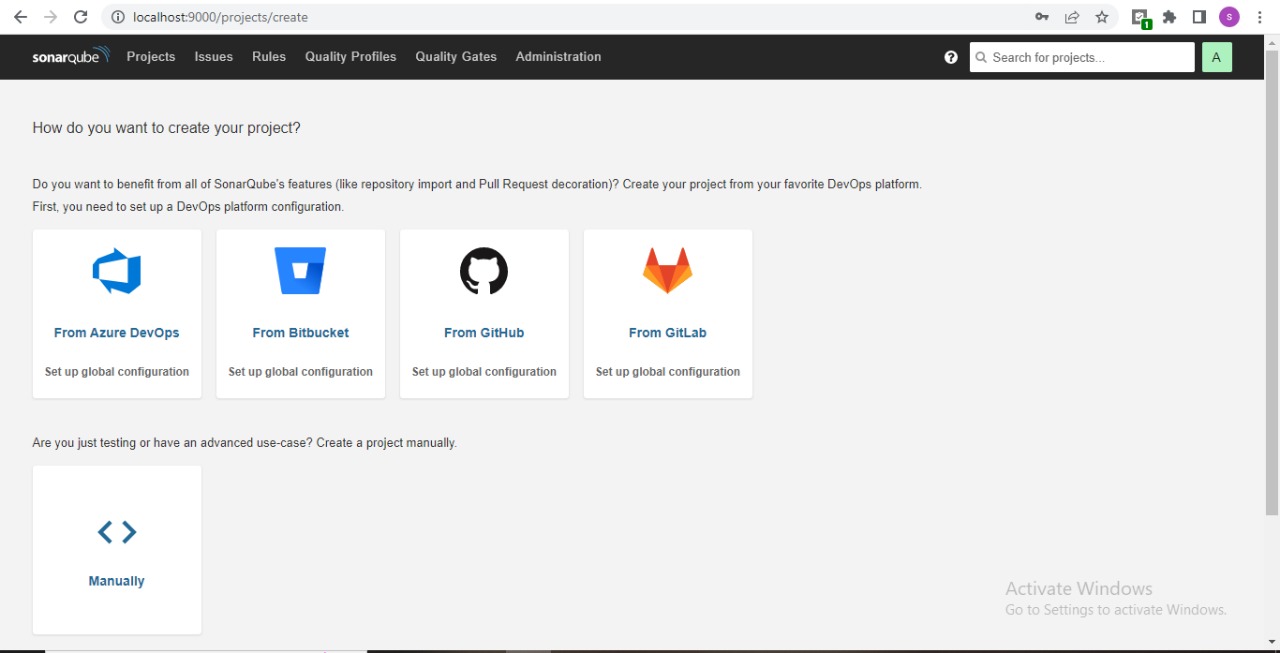
The login credentials are:

Username: admin

Password: admin

Once you login you can change the password.

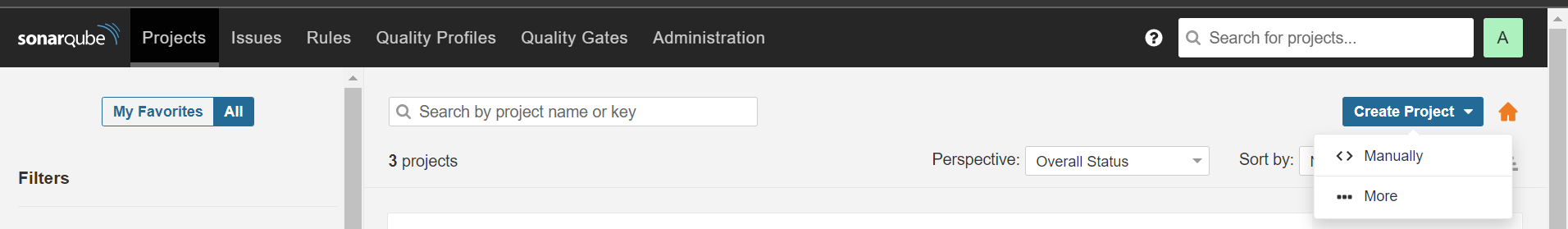
After login You can see SonarQube page



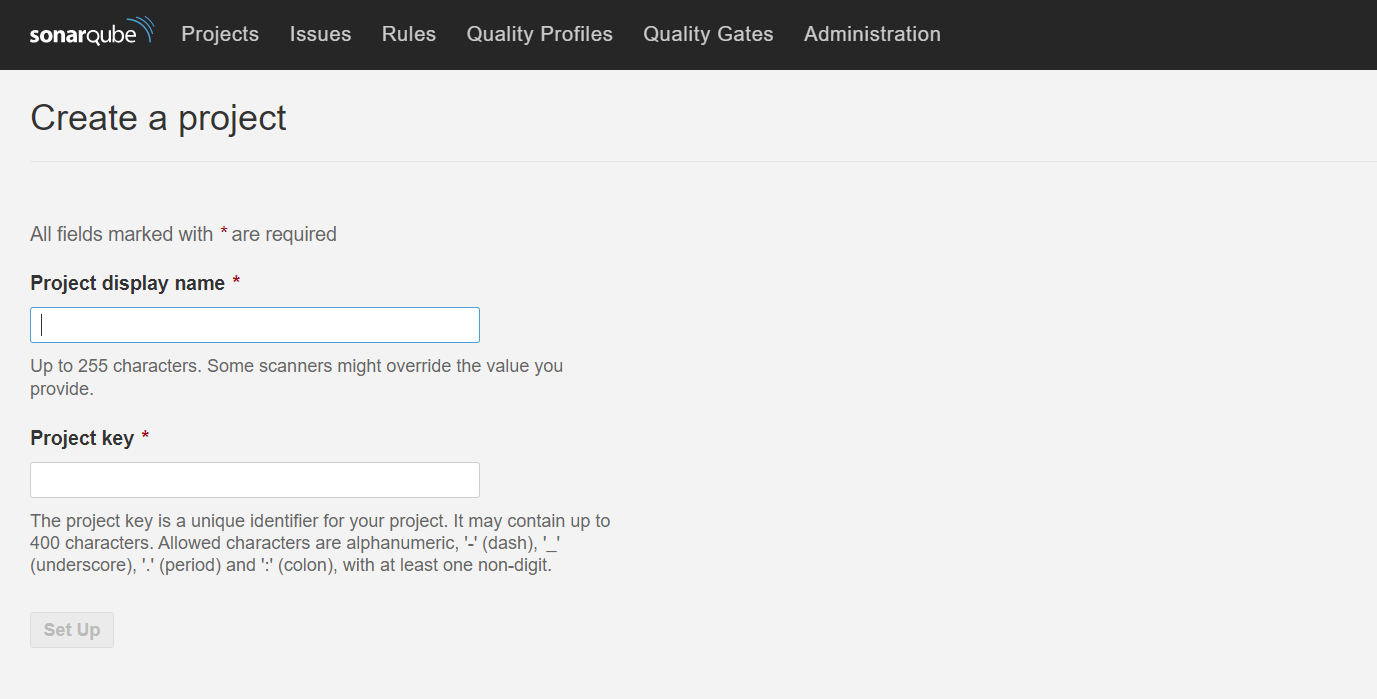
How we can create project manually and connect to Github:

Step 1: Open SonarQube click on project

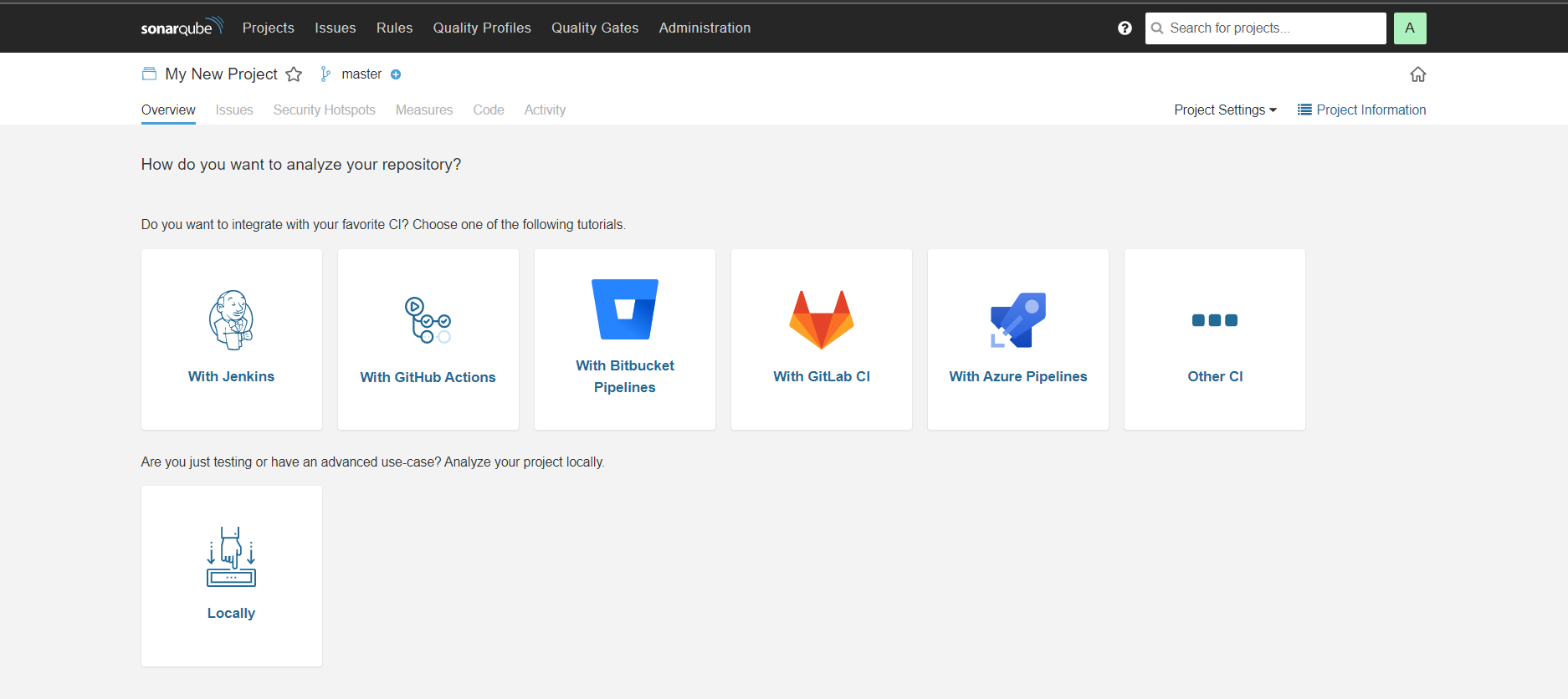
Step 2: select create project>manually.



Step 3: On next page give project name and clock on set up automatically it will create a project key as per project name.

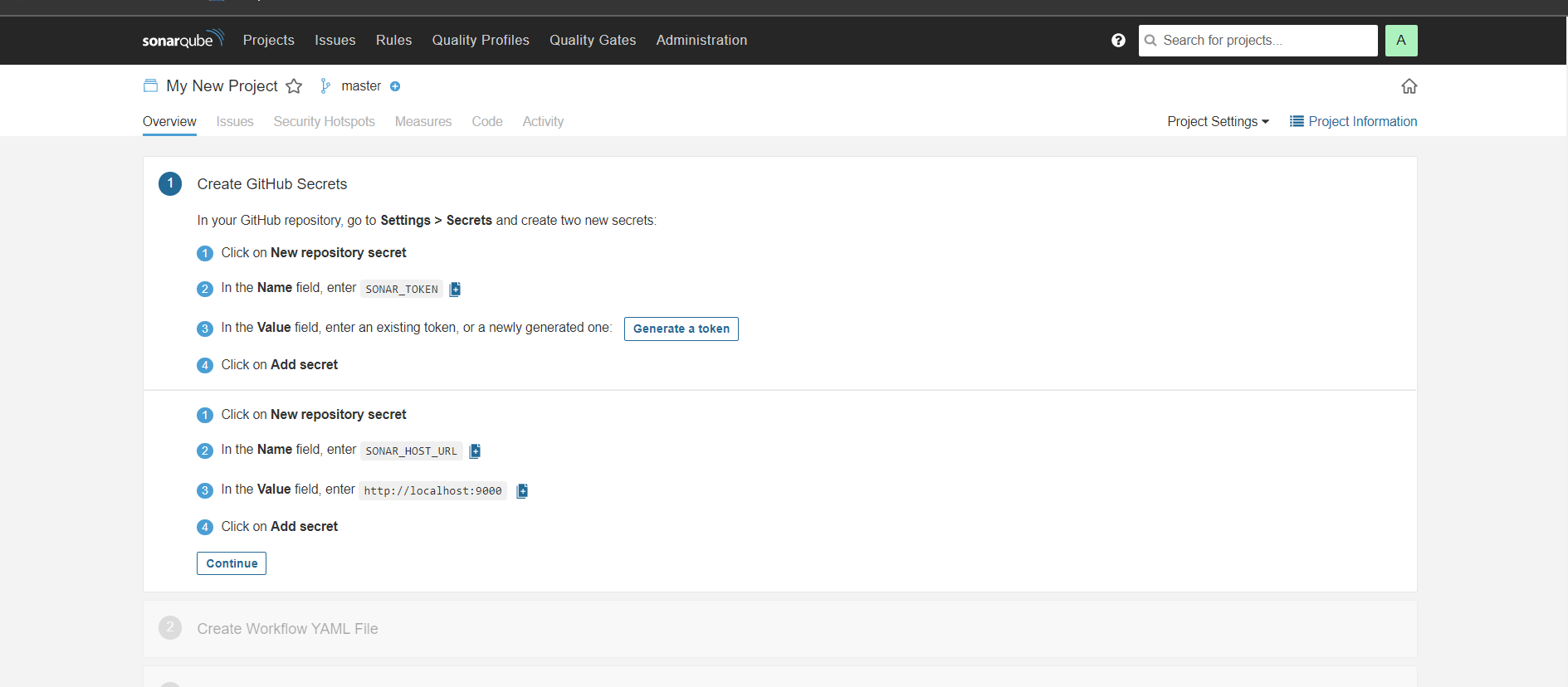


Step 4: Next select using which platform you need to analyze the code.



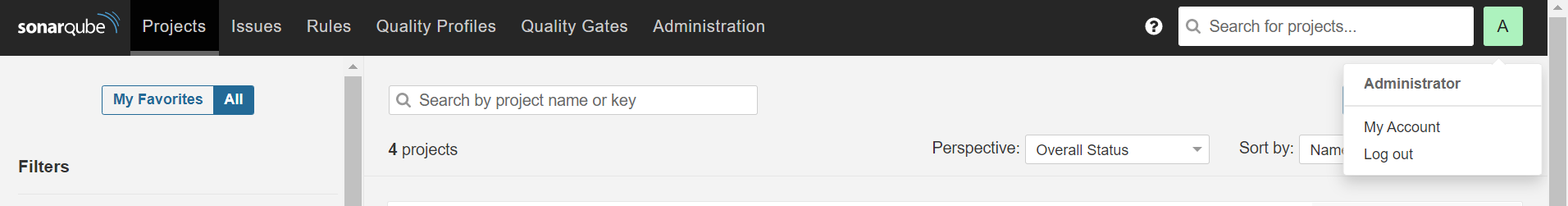
Will be having more option to use as per our project requirement we need to select

Step 5: Click on with GitHub actions. SonarQube will guide you to how to configure with GitHub.Create a New repository secrete and add the token and URL in the secretes.

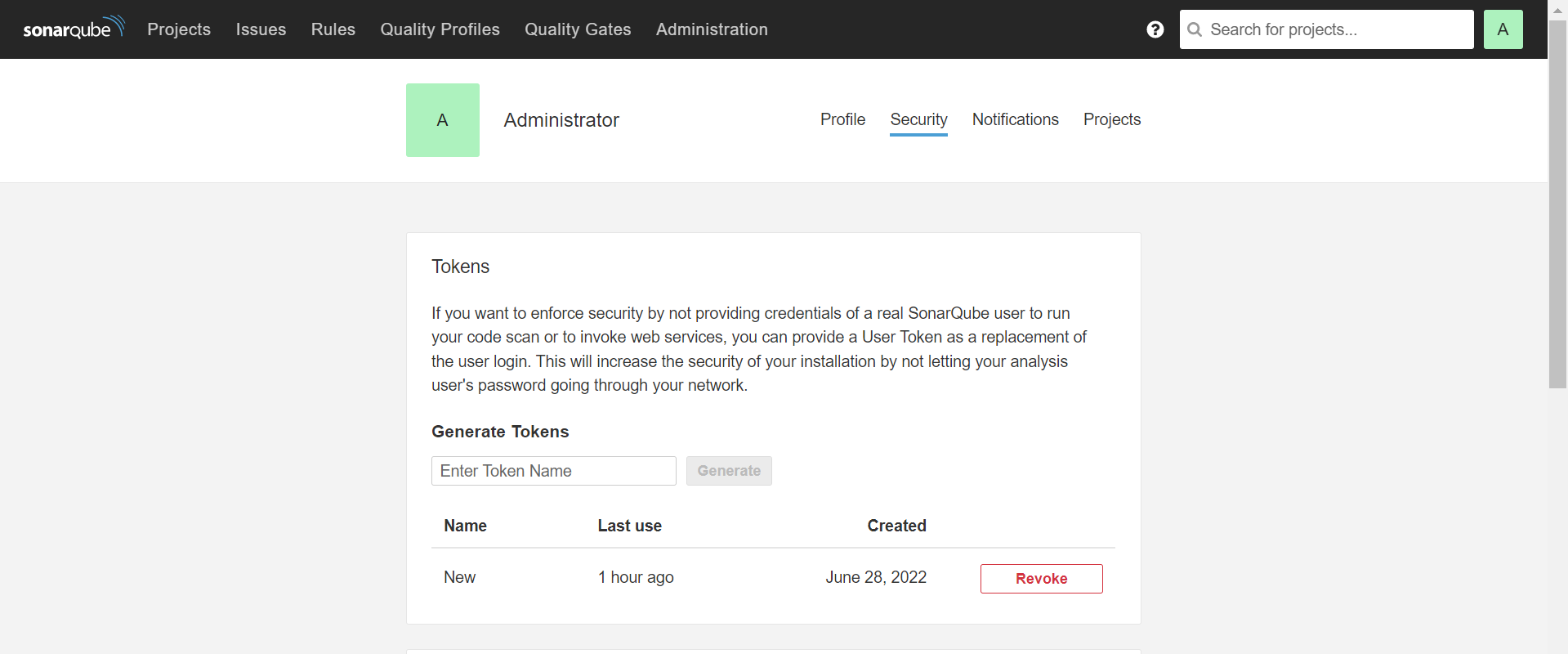


Steps to generate token:

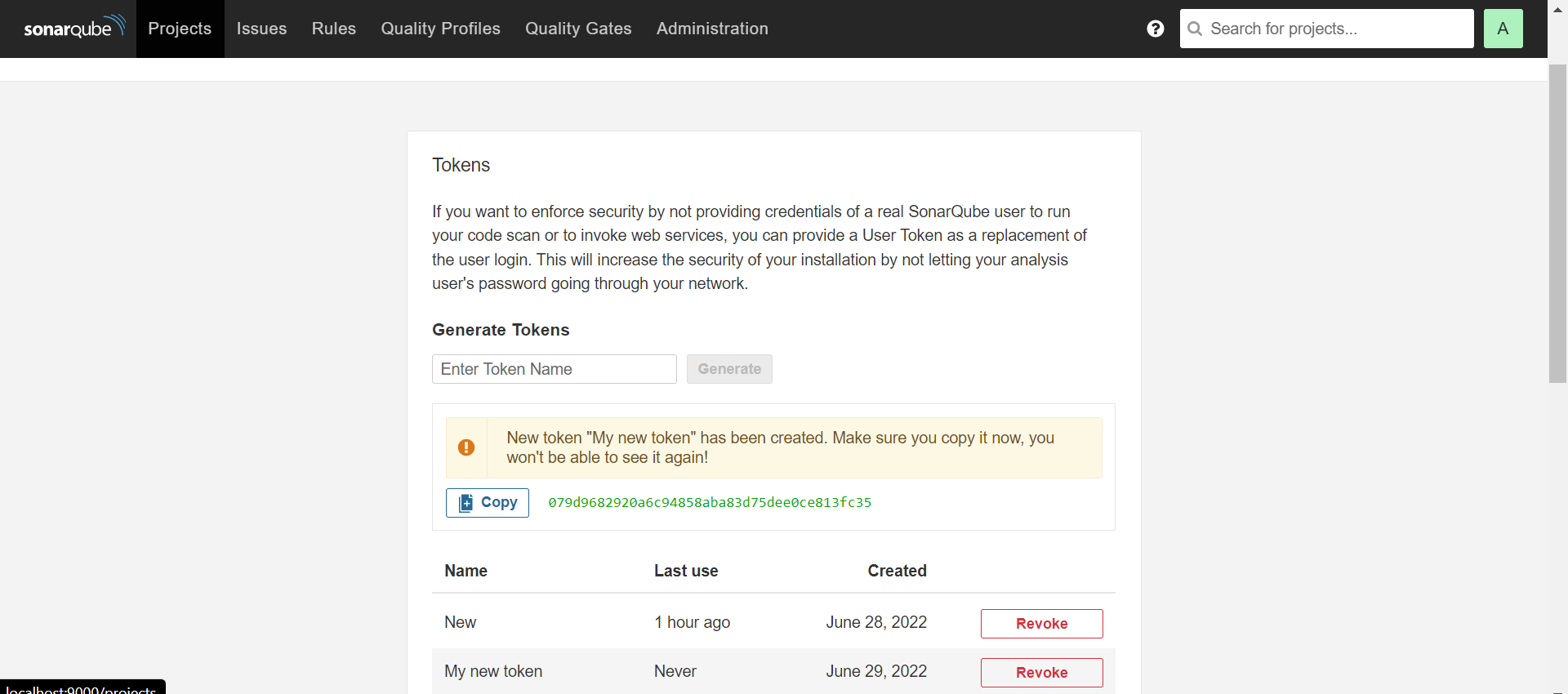
1. Open SonarQube.
2. Click on Profile>My account



1. click on security.



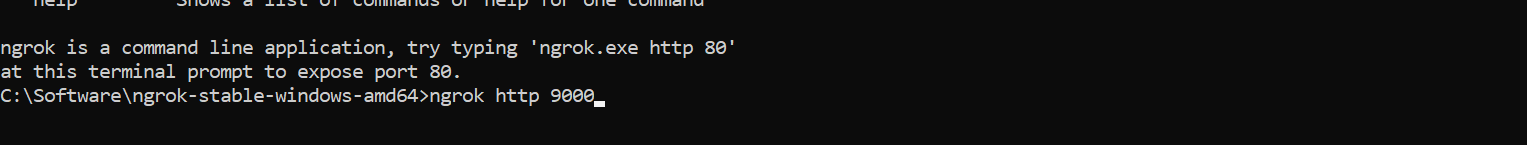
1. Give a name to generate token and click on Generate.



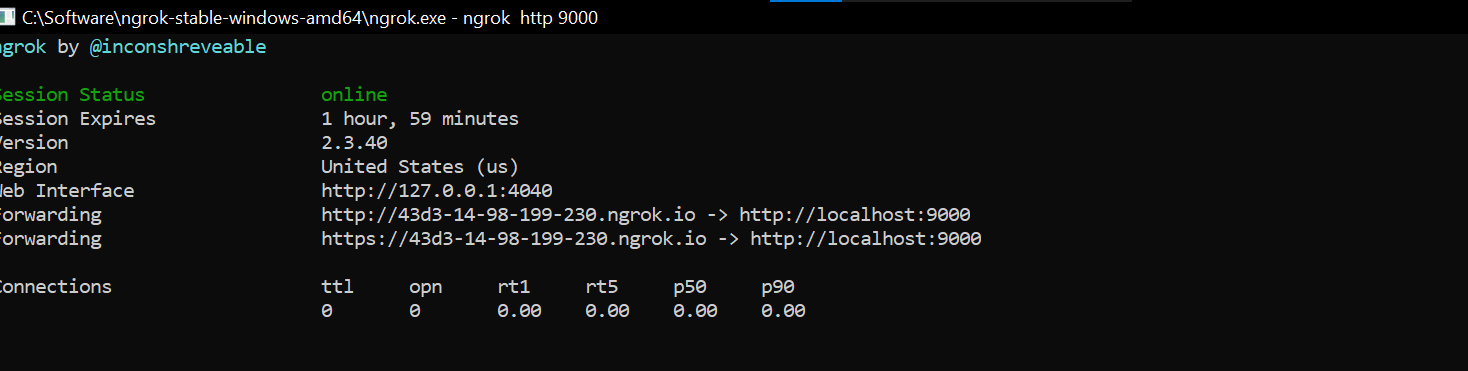
1. Copy the generated token and Paste that token in GitHub secrets.

We are using forwarding tool called ngrok to generate the SonarQube URL then only we will be able to connect to GitHub locally.

1. Install ngrok <https://ngrok.com/download>.
2. Extract the zip file and open ngrock and run the command ngrok http 9000 or ngrok.exe http 9000.



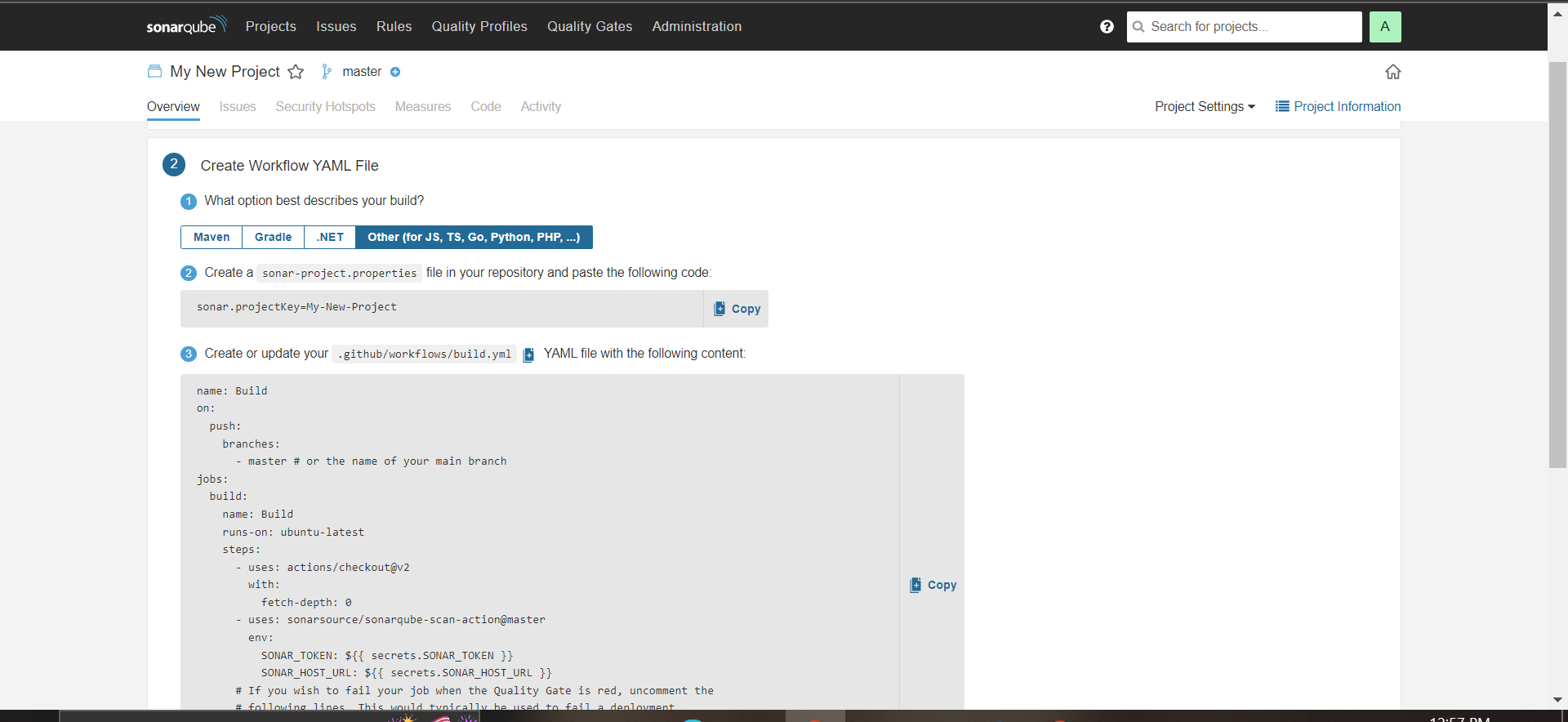
1. After running this command you will find new page with the forwarding URL



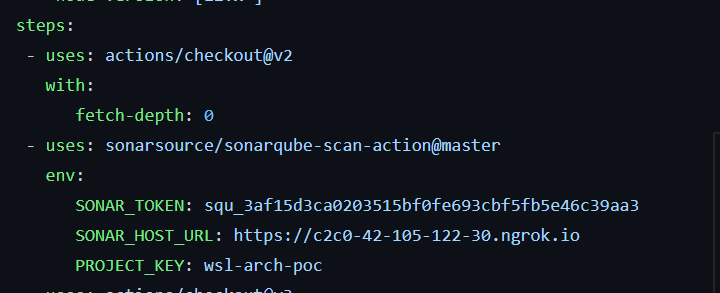
1. Copy the ngrok URL and paste it in GitHub secrets for each and every 2 hours the URL will be keep changing so its better to keep it has hardcode and then need to change the URL for every two hours..

After adding the repository secrets click on continue.

Step 6: Create a Workflow YAML file in the GitHub actions as shown in SonarQube.

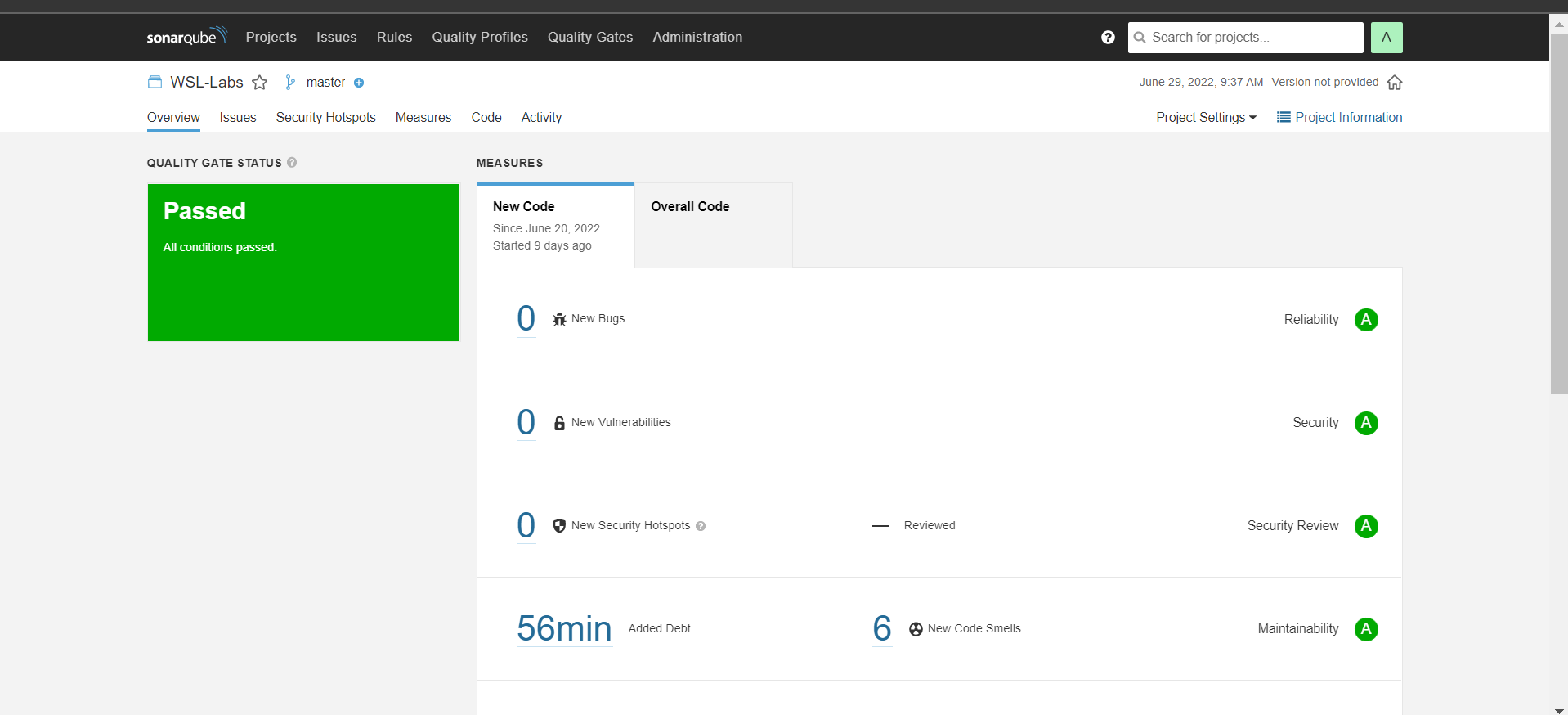


Step 7: Creating workflow or adding sonar code in existing workflow in GitHub action .



Step 8: After adding YAML file click on finish the result.

Step 9: Run the workflow then if the workflow run is successful you will see the code analysis in SonarQube.



Step 10: If the code is passed you can see the quality of code and if any issues present in that code even you can see that in SonarQube.

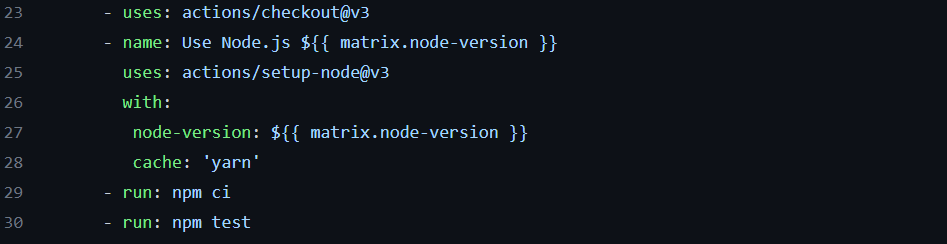
How to delete project in SonarQube.

Go to Project >Project Setting>Deletion>Delete.

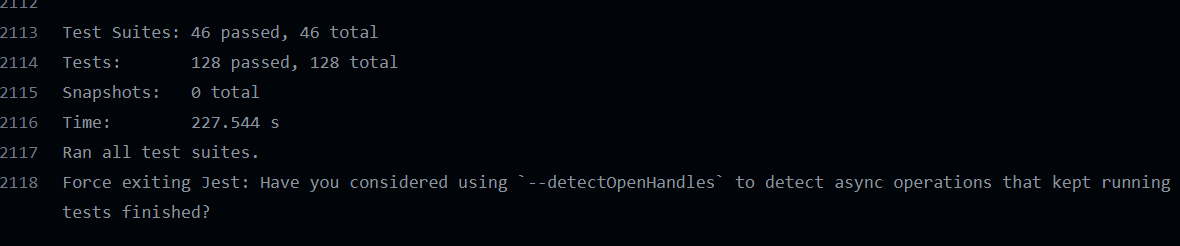
This is we can configure our SonarQube with GitHub using GitHub actions.

How to configure Jest test in GitHub using Github actions:

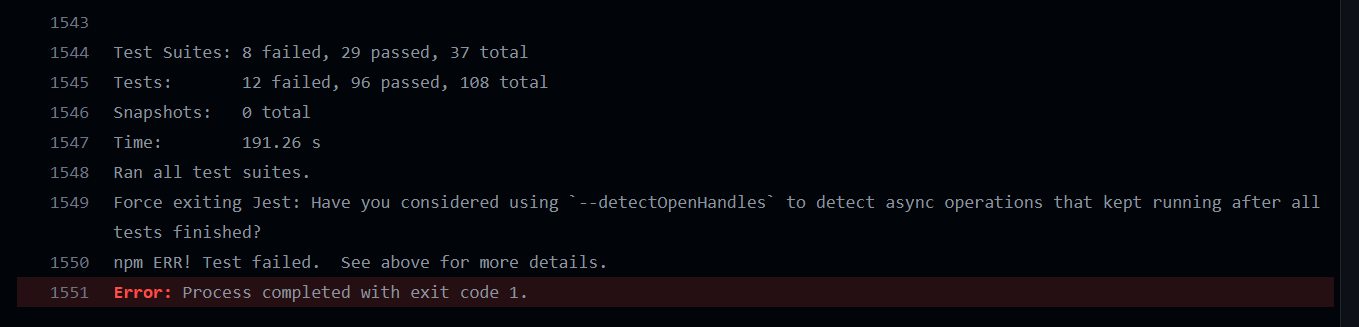
Step 1: Create a new workflow or add the jest code in existing workflow.



Step 2: Run the workflow in GitHub actions if it is successful you can see the test result in actions only.



If the test is failed then it will throw error and not deploy the code in cloud.



By using this code we can achieve integration of SonarQube and Jest with Github.