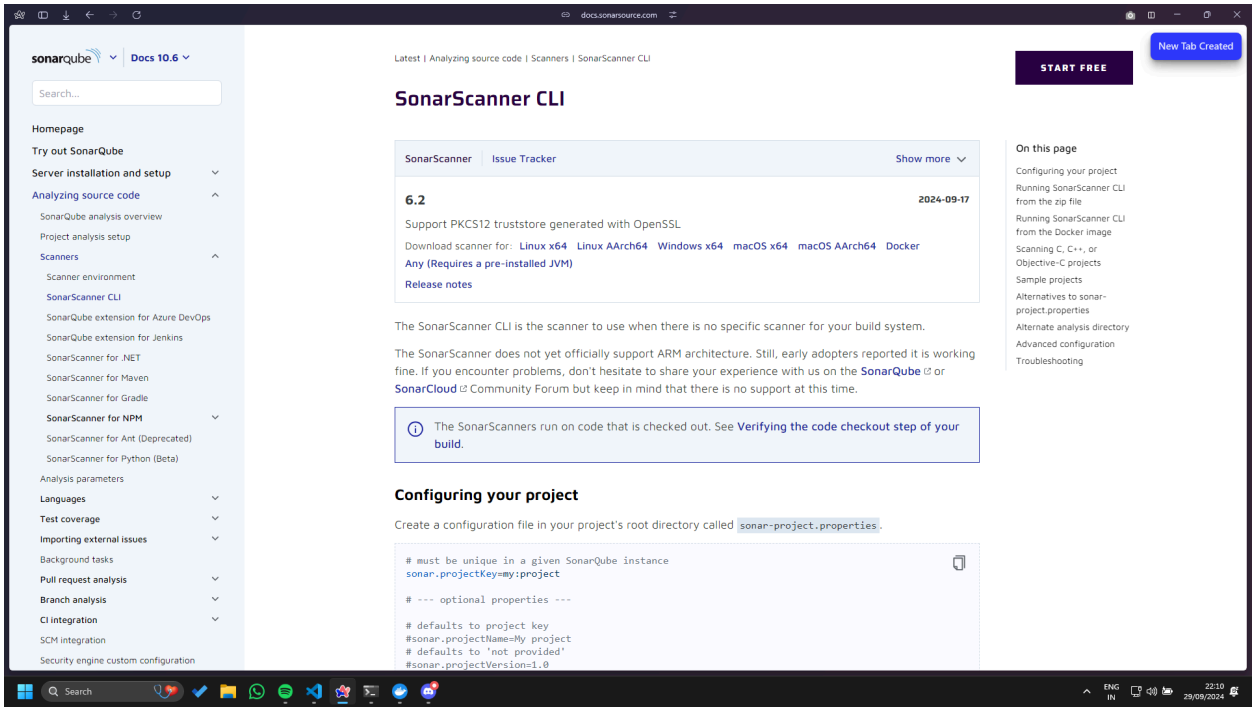
ADITYA LALWANI D15C ROLL NO.30

Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/
Visit this link and download the sonarqube scanner CLI.



Run docker -v command.
Run docker pull sonarqube command

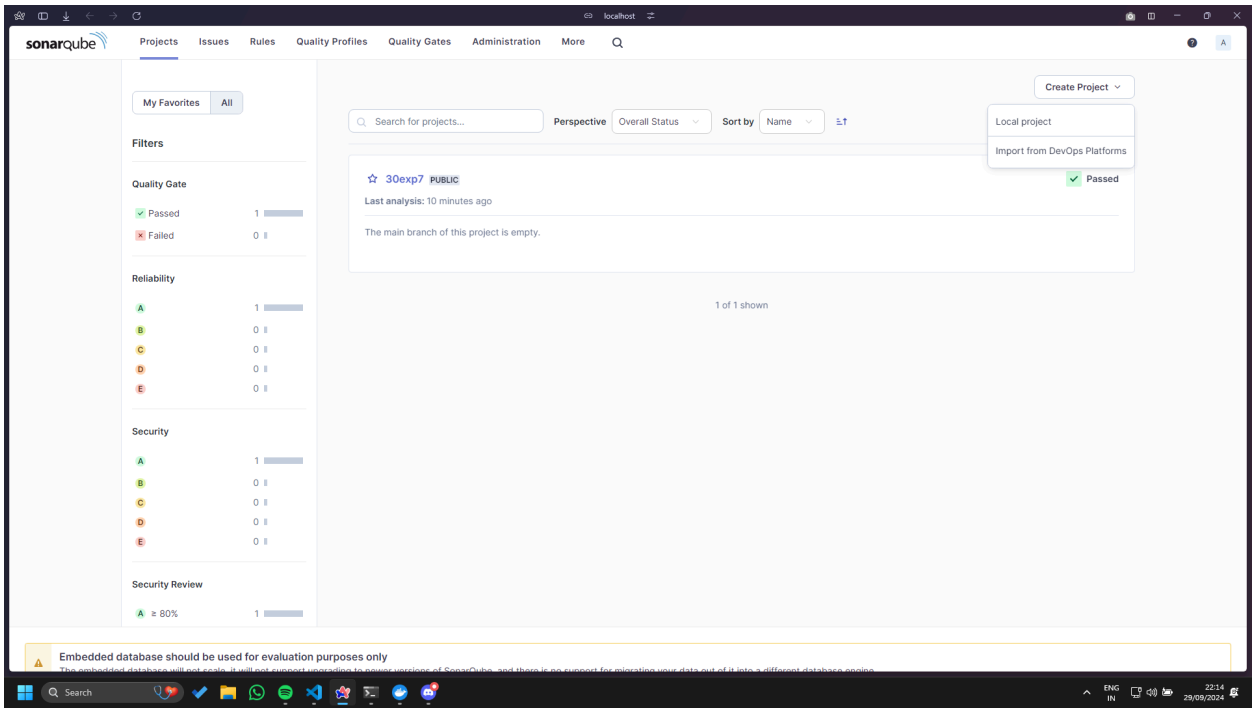
```
C:\Users\Aditya>docker -v
Docker version 27.2.0, build 3ab4256

C:\Users\Aditya>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
4f4fb700ef54: Download complete
bd819c9b5ead: Download complete
80338217a4ab: Download complete
7478e0ac0f23: Download complete
90a925ab929a: Download complete
7b87d6fa783d: Download complete
1a5fd5c7e184: Download complete
7d9a34308537: Download complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest
```

ADITYA LALWANI D15C ROLL NO.30

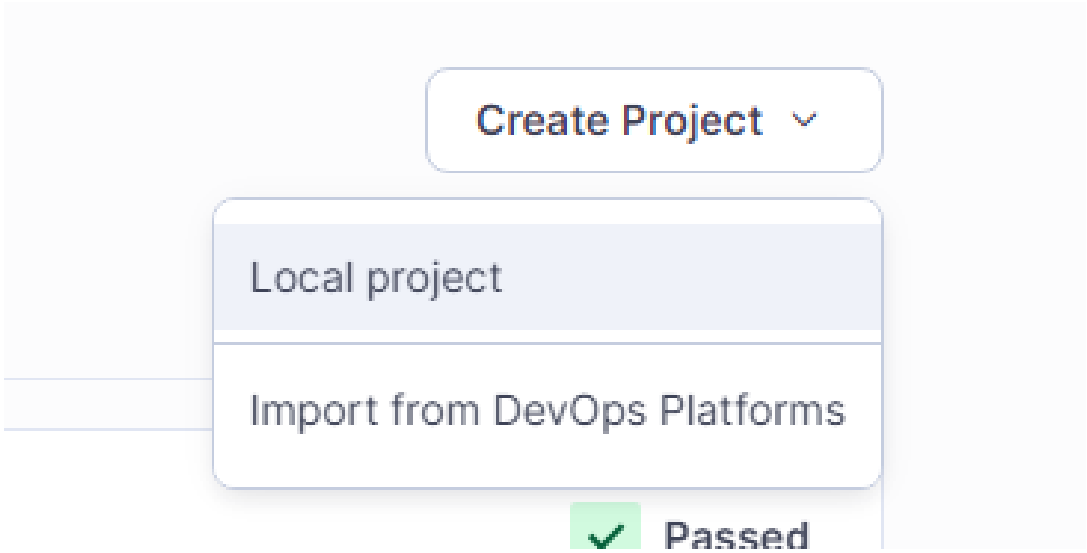
Run SonarQube image docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest This command will run the SonarQube image that was just installed using docker.

```
C:\Users\Aditya>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
102a6e51b67968db1df41bc8c6c3ce068f11506acadcbab2d306470aede30908
```

Go to http://localhost:9000



Create a new local project



1 of 2

Create a local project

Project display name *

30exp8

Project key *

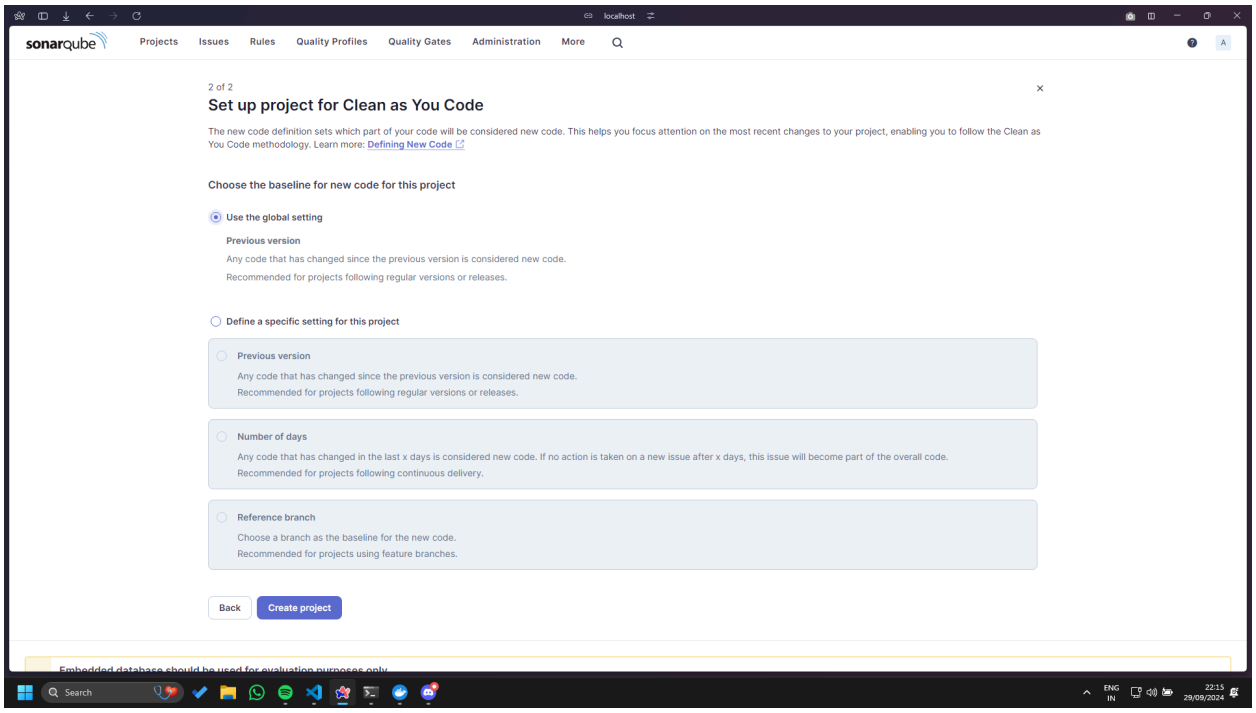
30exp8

Main branch name *

main

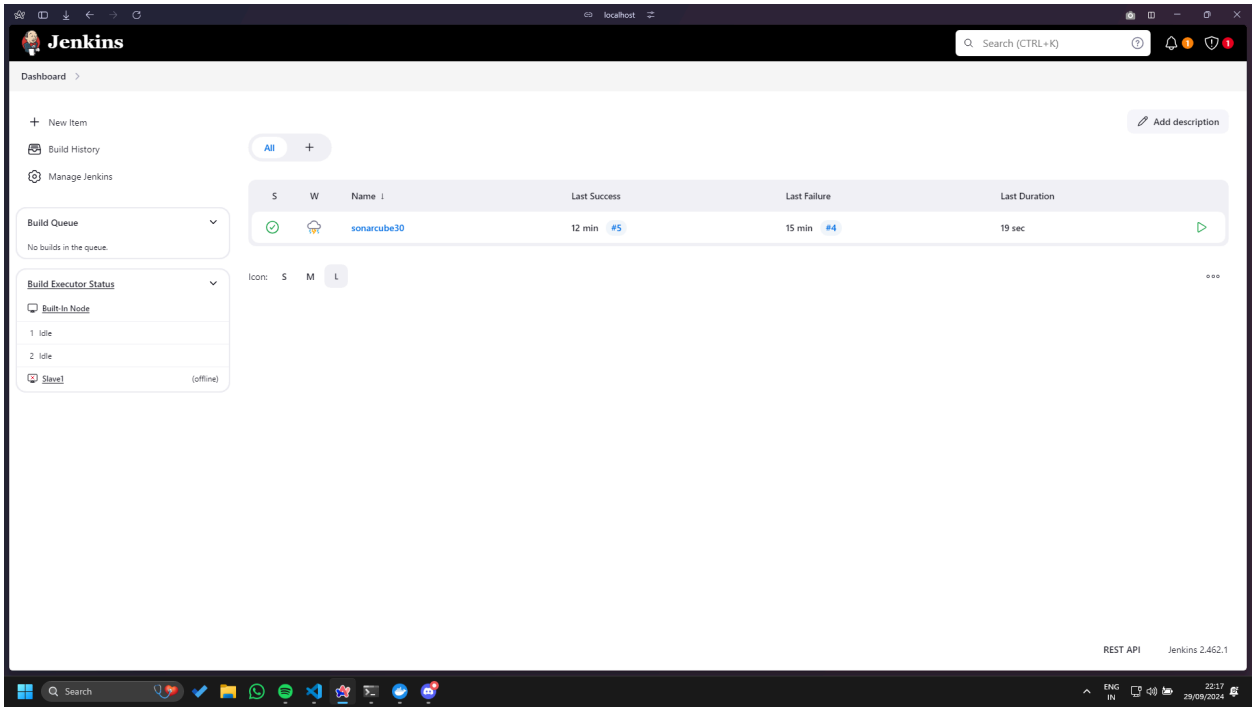
The name of your project's default branch Learn More

Cancel Next



ADITYA LALWANI D15C ROLL NO.30

Open jenkins

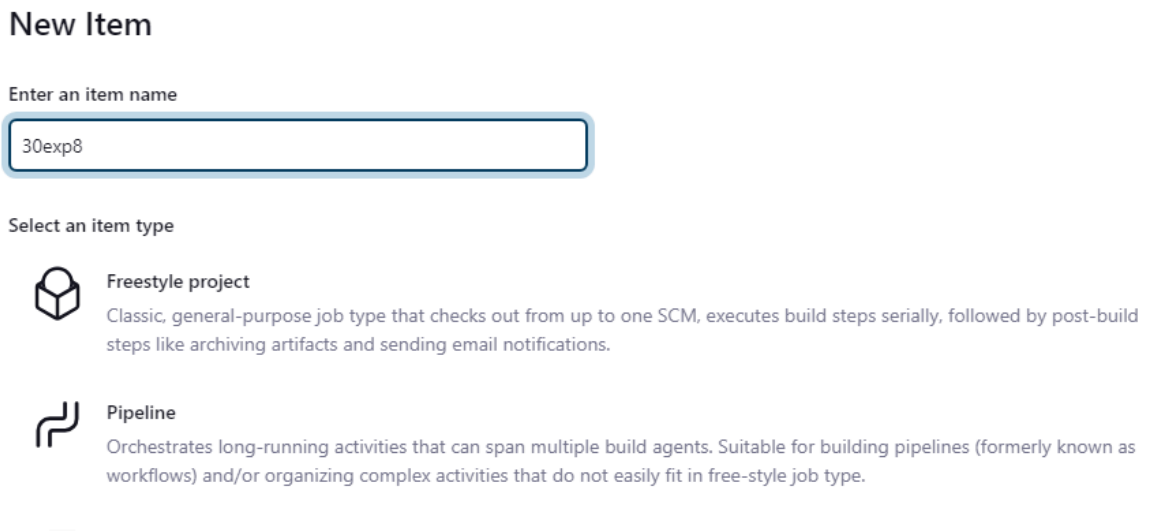


Go to manage jenkins → Search for Sonarqube Scanner for Jenkins and install it.

) Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.

Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose install automatically.

After configuration, create a New Item → choose a pipeline project.

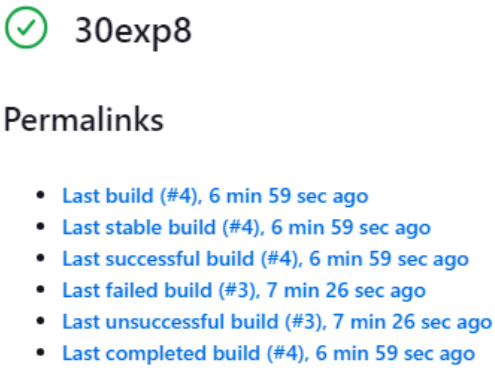


Under Pipeline script, enter the following:

```
node {
   stage('Cloning the GitHub Repo') {
      git 'https://github.com/shazforiot/GOL.git'
   }
   stage('SonarQube analysis') {
      withSonarQubeEnv('sonarqube lab') {
         bat """

C:\\Users\\Aditya\\Downloads\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat ^
         -D sonar.login=admin ^
         -D sonar.password=aditya@123 ^
         -D sonar.projectKey=30exp8 ^
```

```
            -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
            -D sonar.host.url=http://localhost:9000/
            """
        }
    }
}
```

Go to the job you had just built and click on Build Now.

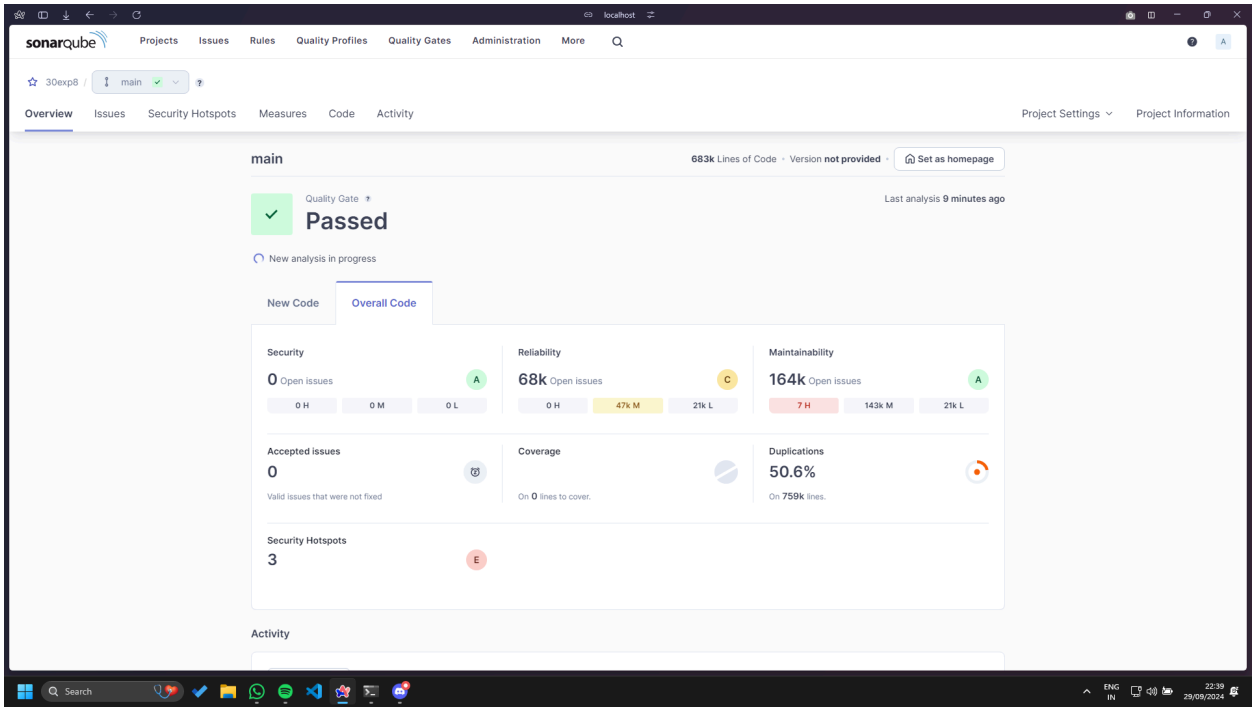


ADITYA LALWANI D15C ROLL NO.30

Check the console output once

```
22:36:56.460 INFO  Analysis total time: 6:26.571 s
22:36:56.477 INFO  SonarScanner Engine completed successfully
22:36:57.109 INFO  EXECUTION SUCCESS
22:36:57.193 INFO  Total time: 6:30.536s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Once the build is complete, go back to SonarQube and check the project linked.

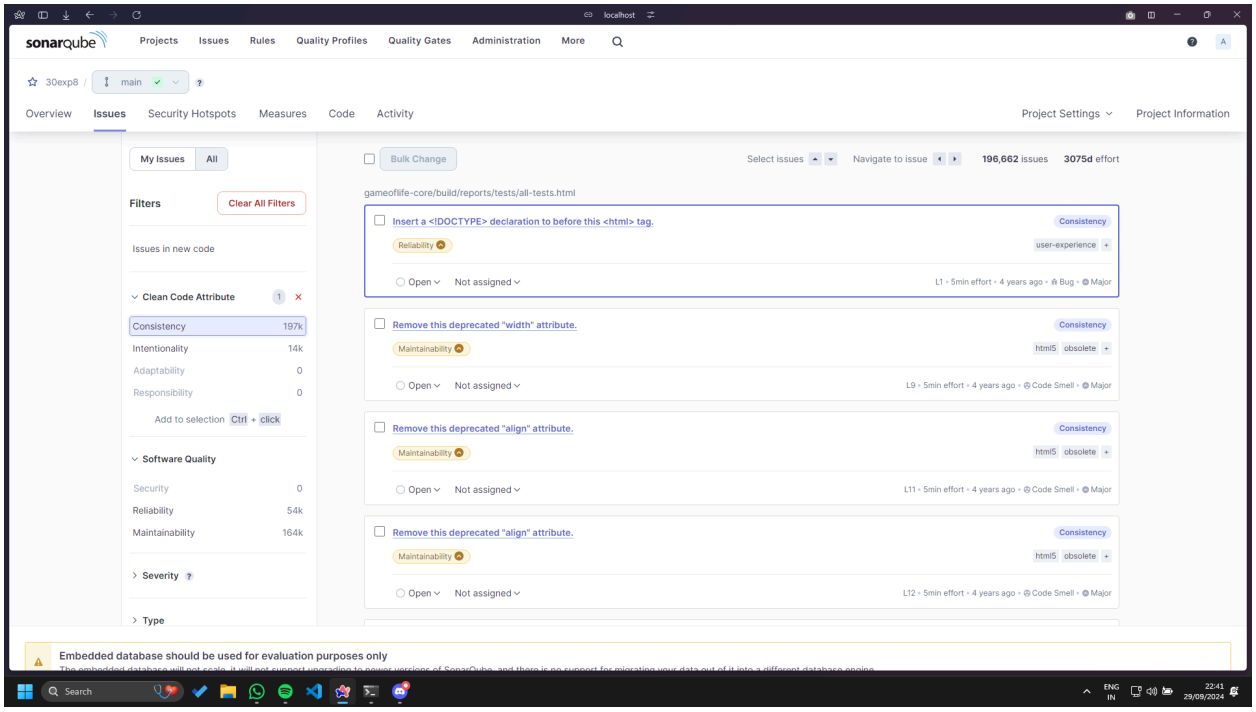


Once the build is complete, go back to SonarQube and check the project linked. Under different tabs, check all the issues with the code.

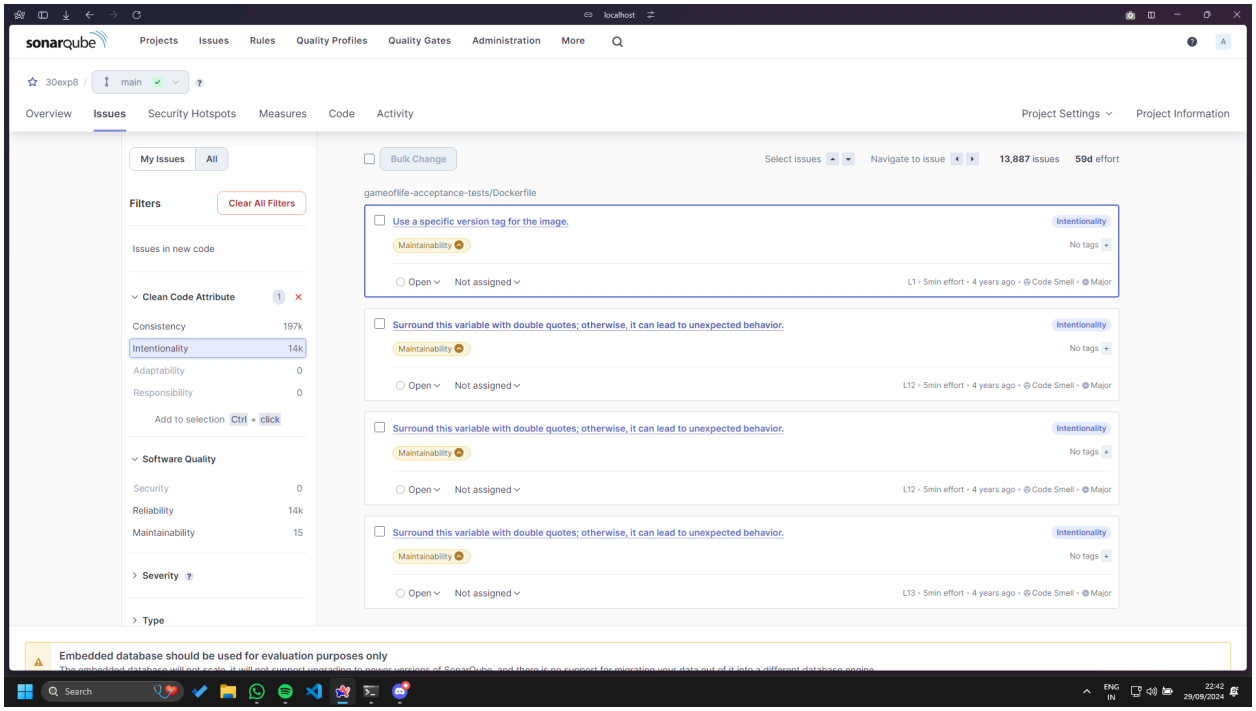
- Code Problems

ADITYA LALWANI D15C ROLL NO.30

- Consistency

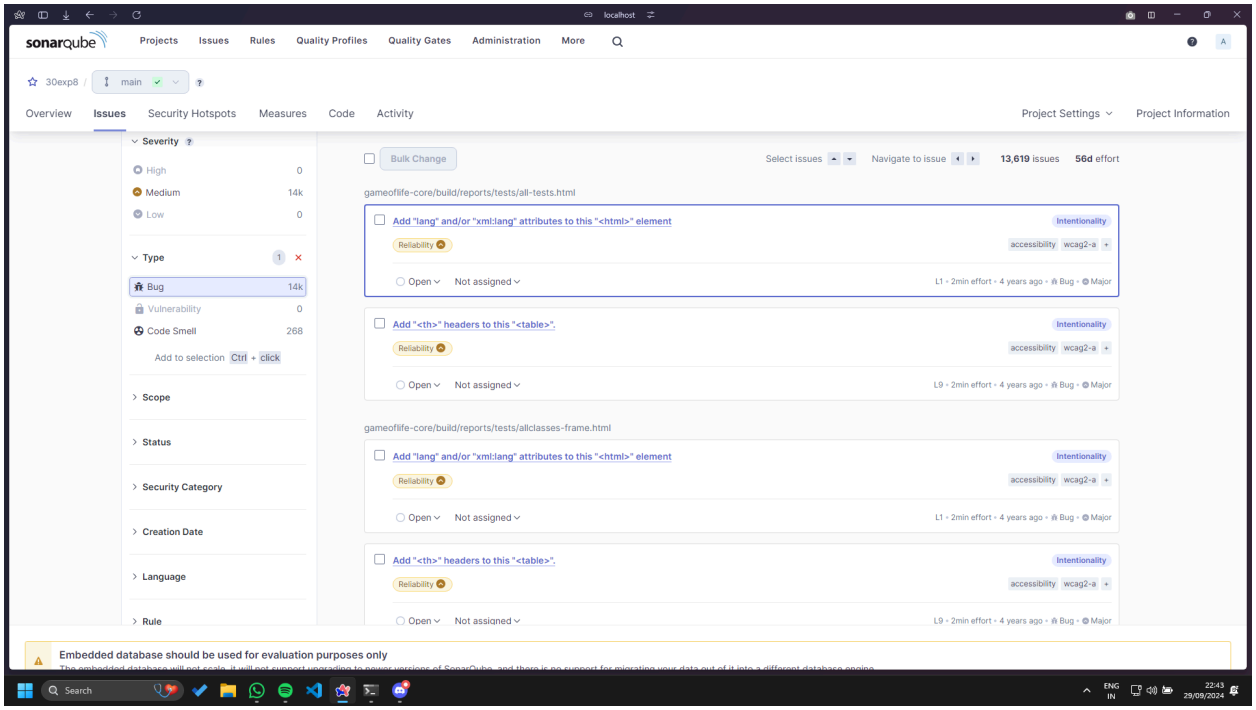


- Intentionality

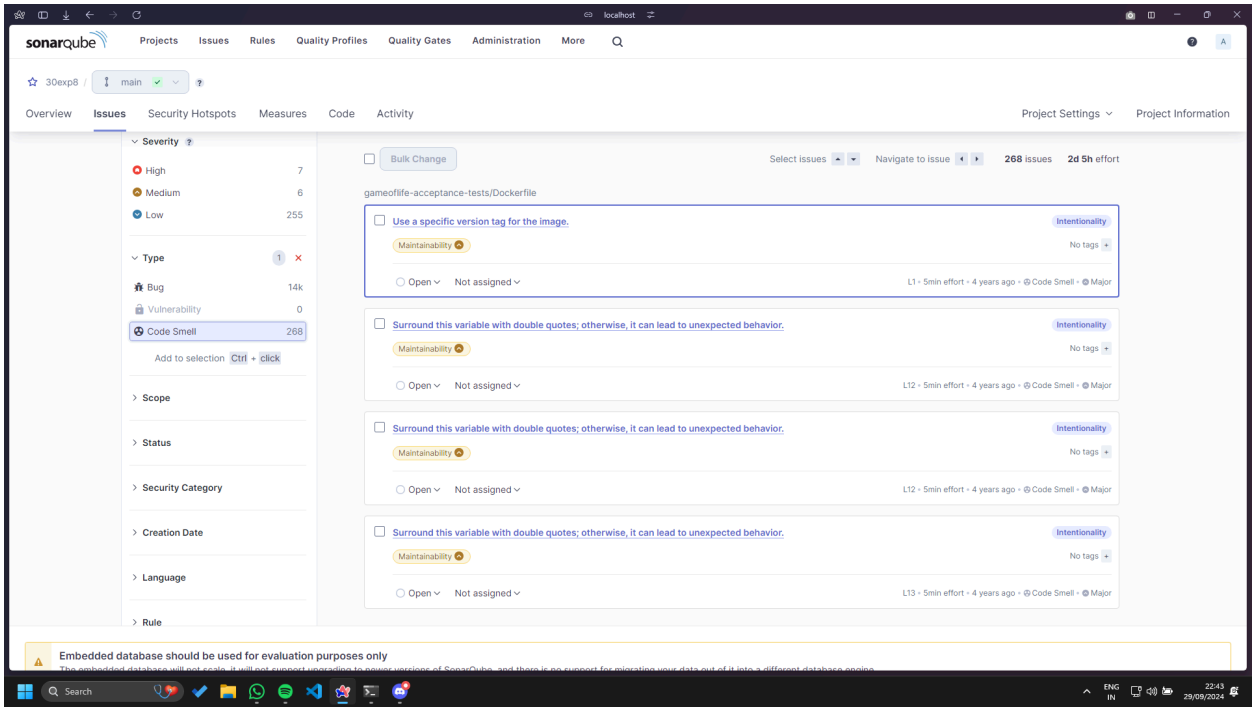


- Bugs

ADITYA LALWANI D15C ROLL NO.30

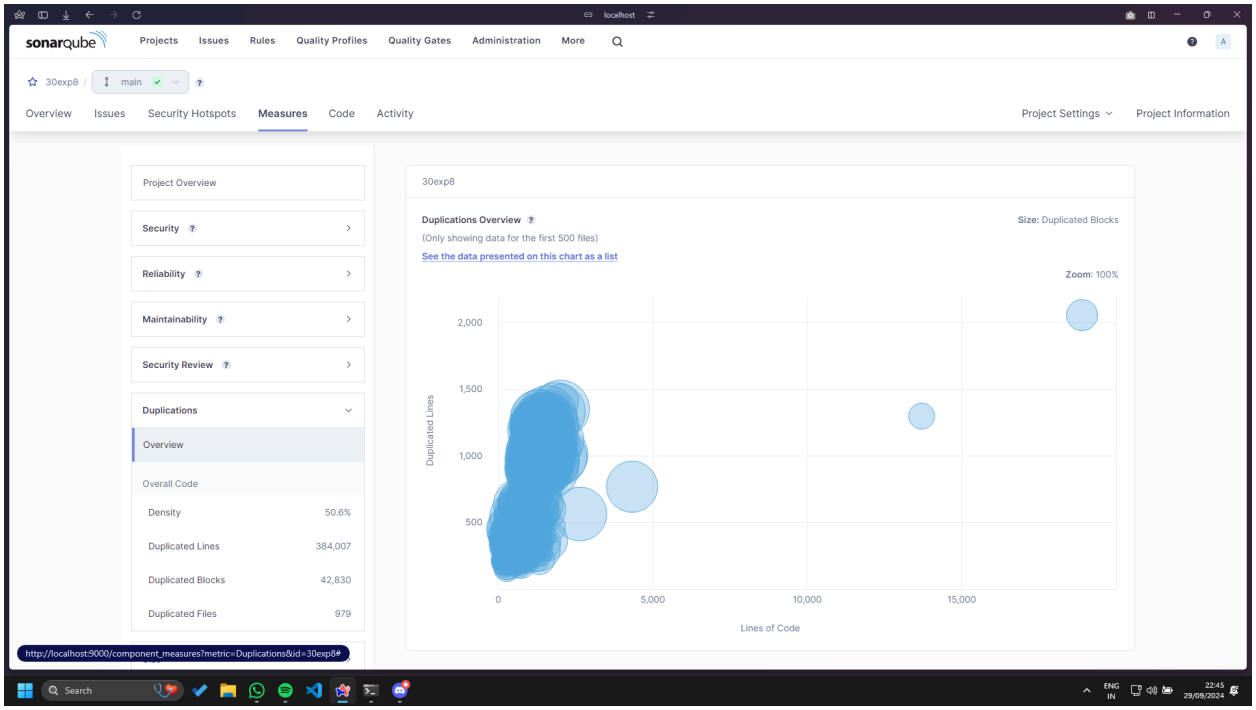


- Code Smells

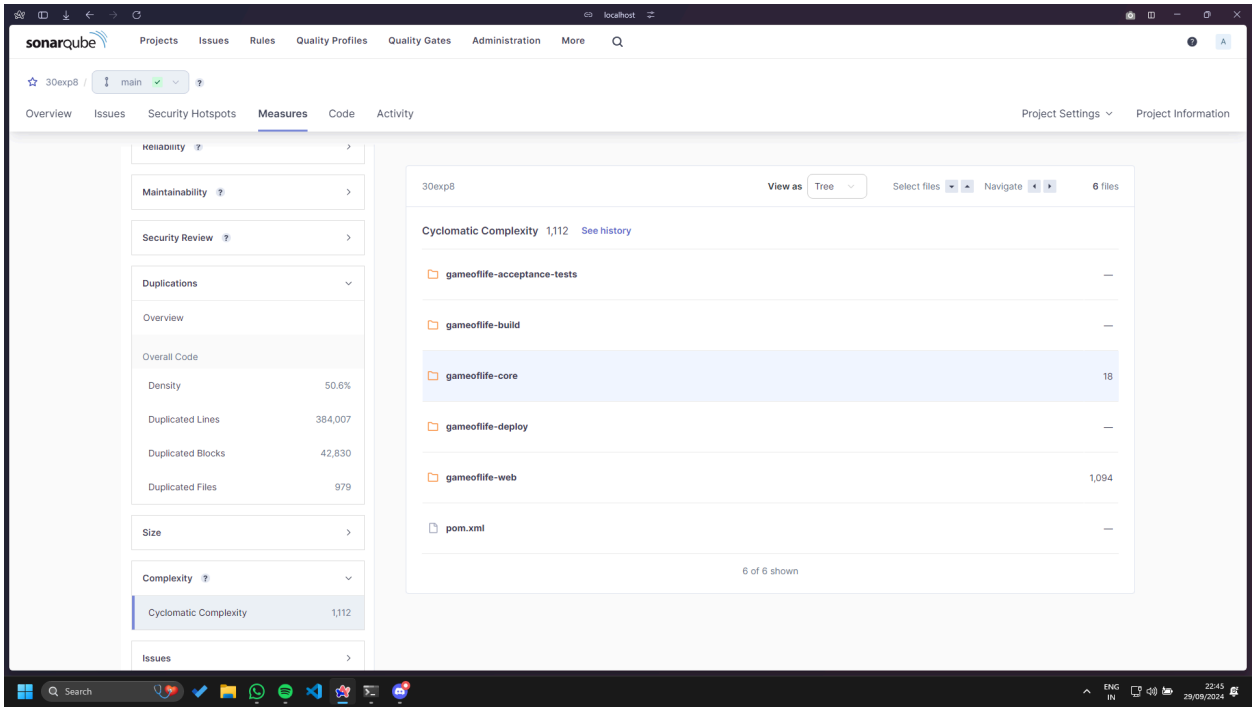


- Duplications

ADITYA LALWANI D15C ROLL NO.30



- Cyclomatic Complexities



Conclusion: In this experiment, we have learned how to perform static analysis of a code using Jenkins CI/CD Pipeline with SonarQune analysis. A pipeline project is to be created which is given a pipeline script. This script contains all the information needed for the project to run the SonarQube analysis. After the necessary configurations are made on jenkins, the Jenkins project is built. The code provided in this experiment contains lots of error, bugs, duplications which can be checked on the SonarQube project linked with this build