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.

STEP 1: In your command prompt to ensure whether the docker is installed or not.

```
C:\Users\Dell>docker -v
Docker version 27.1.1, build 6312585
```

STEP 2: Run docker login command and add your username and password for docker.

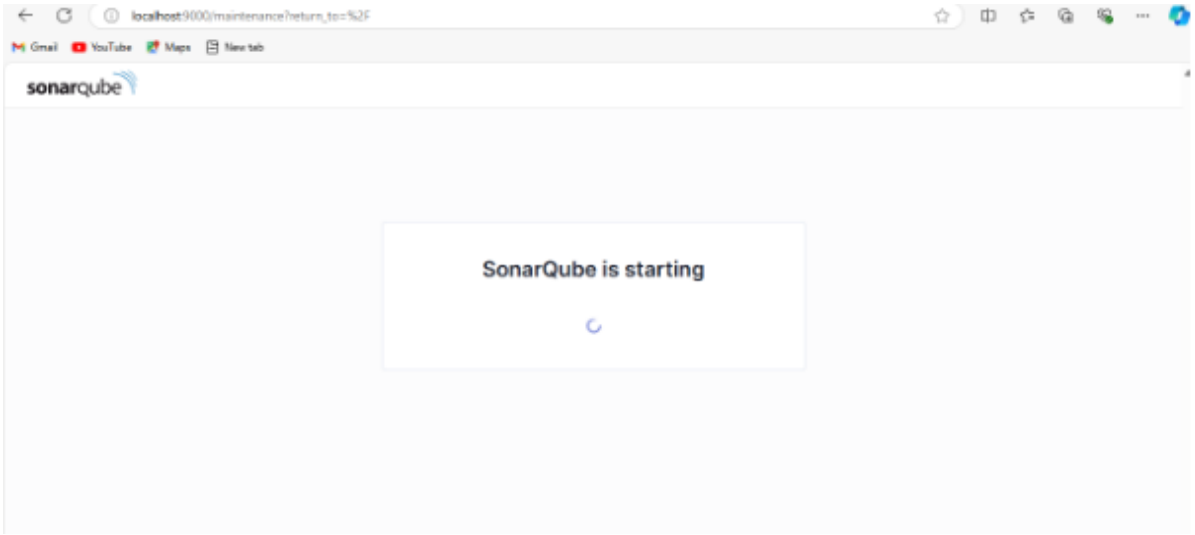
```
C:\Users\Dell>docker pull sonarqube
Using default tag: latest
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31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

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
```

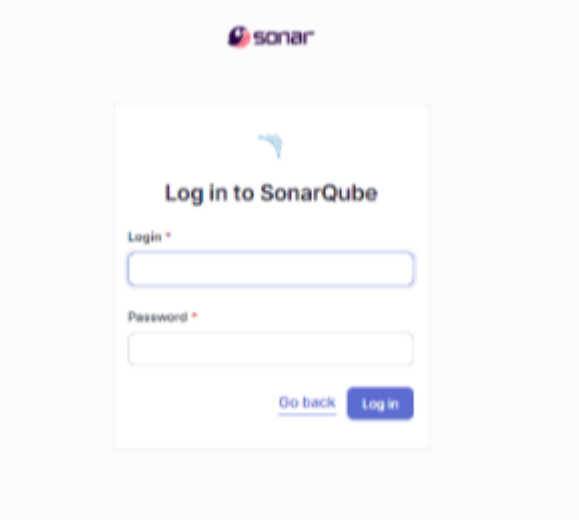
STEP 3: Run docker pull sonarqube command to install sonarqube image without actually installing then.

```
C:\Users\Dell>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
ac1f985dedebc00a642a4c69a502d611389e8f9fa46610febe75aa5021767cab
```

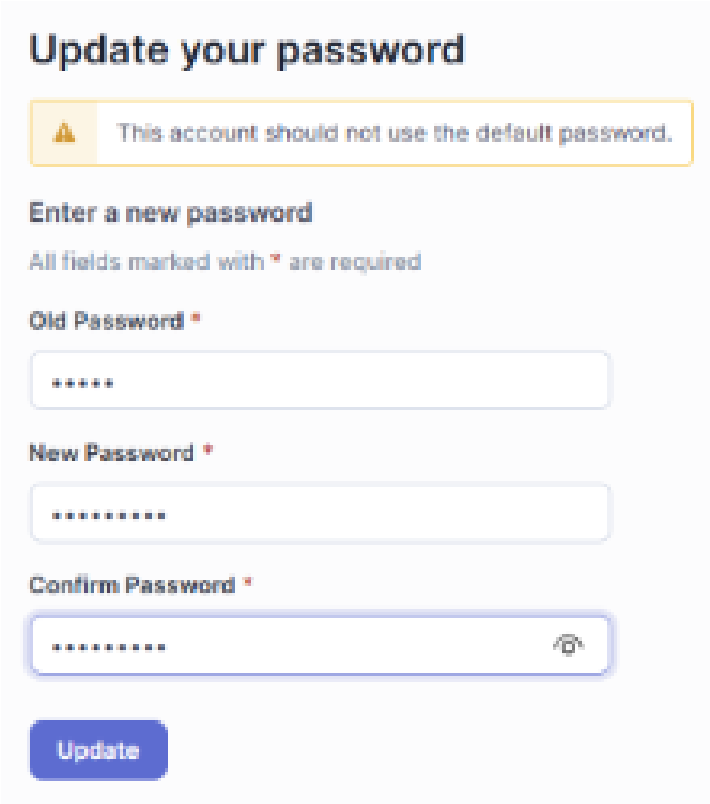
STEP 4: Run docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest Command to run the sonarqube.



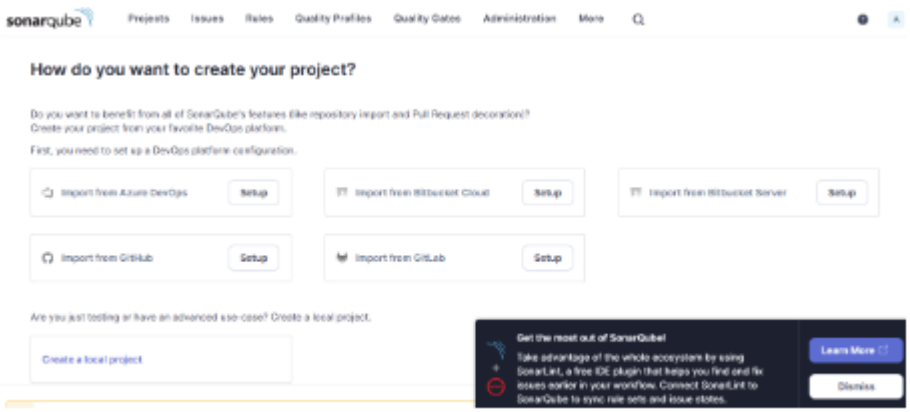
STEP 5: Once the sonarqube is runned go to your web browser and whatever port number u have mentioned in the previous command open that page using localhost:9000.



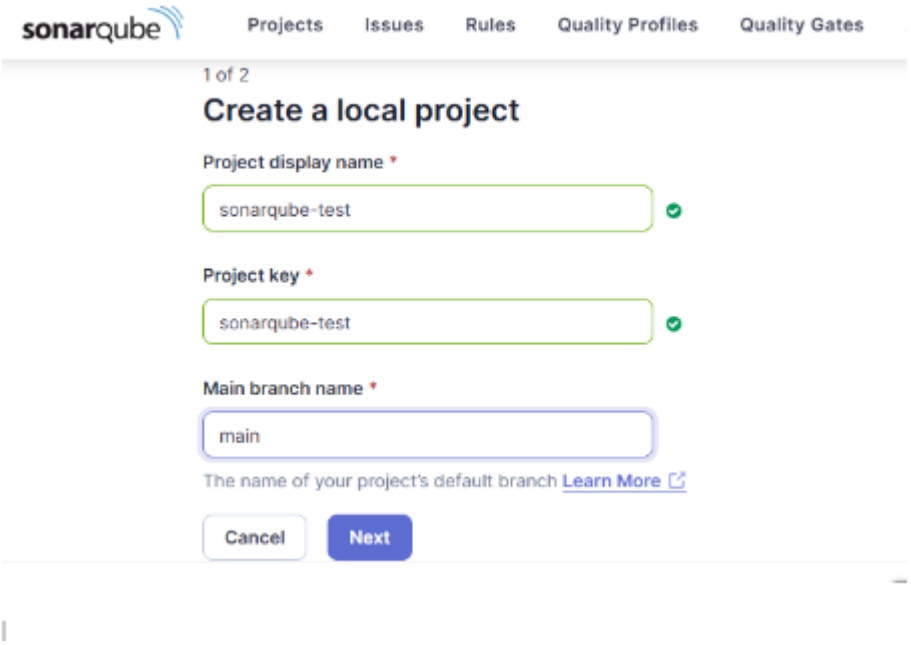
STEP 6: Once sonarqube is started it will redirect you to login page .The login and password for sonarqube is both “admin”.



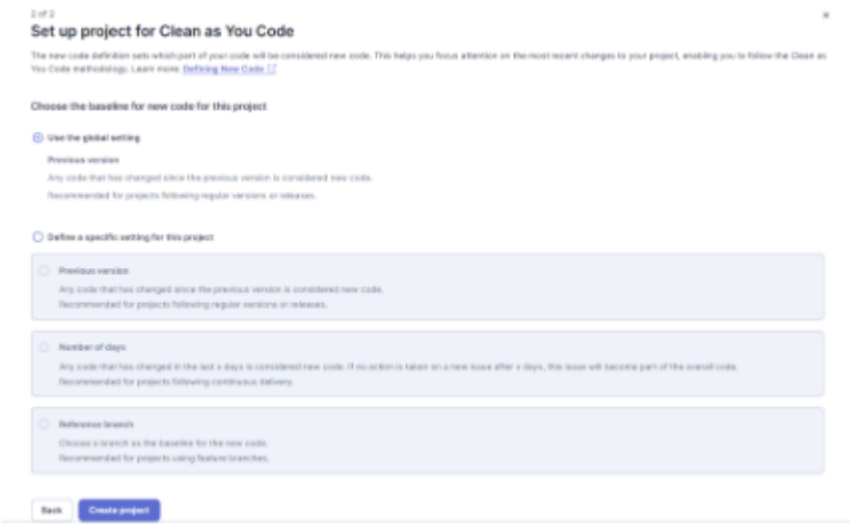
STEP 8: After changing the password, you will be directed to this screen. Click on Create a Local Project.



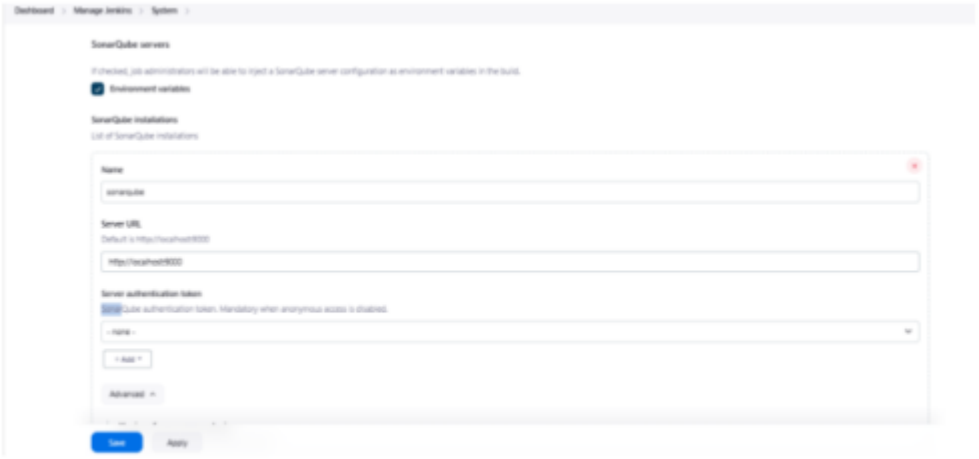
STEP 9: Add name of the project and project key and select the main branch name and click on next.



STEP 10: Set up the project as required and click on create.

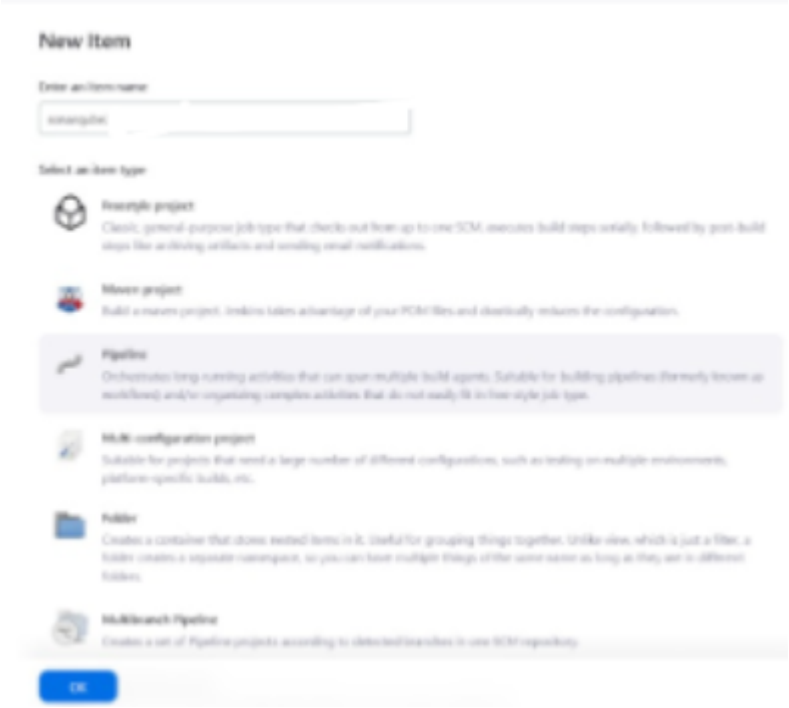


STEP 11:Go to Manage Jenkins and then go to Systems and name to the environment variables then apply the changes and then save them.



STEP 12: In SonarQube Scanner add the latest version then apply the changes and save it.

STEP 13: Go to Jenkins and then create a new item enter the item name and select an item type to “Pipeline” and then click on ok.



STEP 14: Under Pipeline script, enter the following:

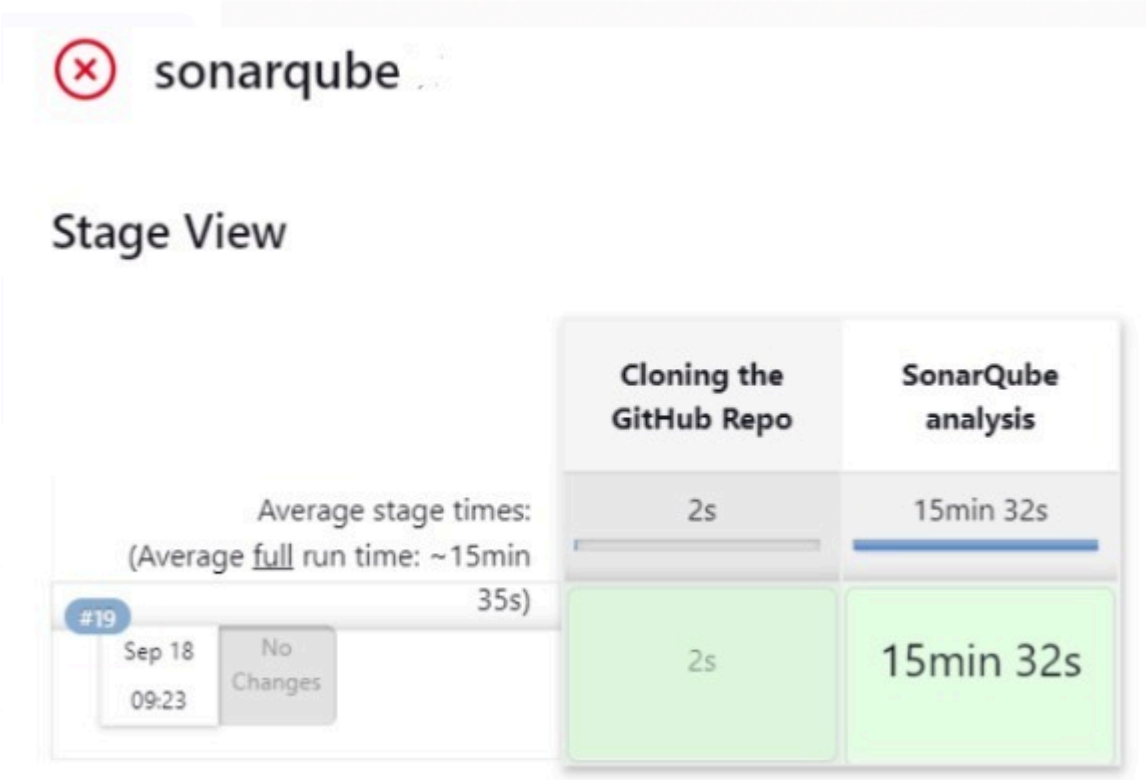
```
node {
        stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
        }

         stage('SonarQube analysis') {
                withSonarQubeEnv('sonarqube lab') {
                bat """
                        <PATH_TO_SONARSCANNER_FOLDER>\\bin\\sonar-scanner.bat
        ^
        -D sonar.login=<SONARQUBE_LOGIN> ^
        -D sonar.password=<SONARQUBE_PASSWORD> ^
        -D sonar.projectKey=<PROJECT_KEY> ^
        -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
        -D sonar.host.url=http://localhost:9000/

        }
    }
}
```

STEP 15: Add the script to the pipeline script.

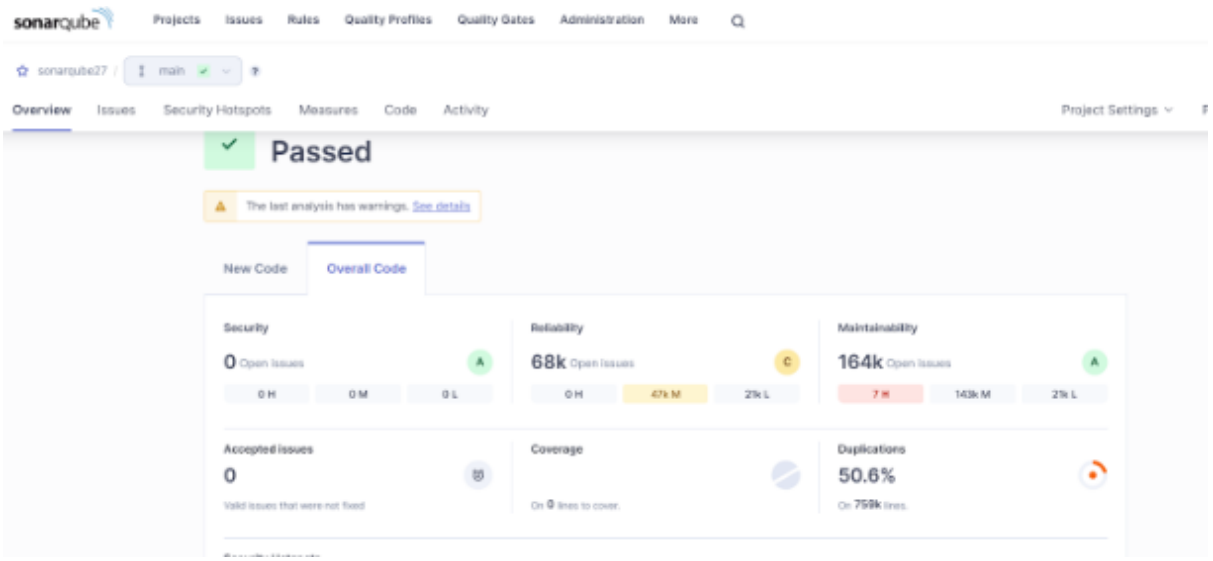
STEP 16: Go to the job you had just built and click on Build Now

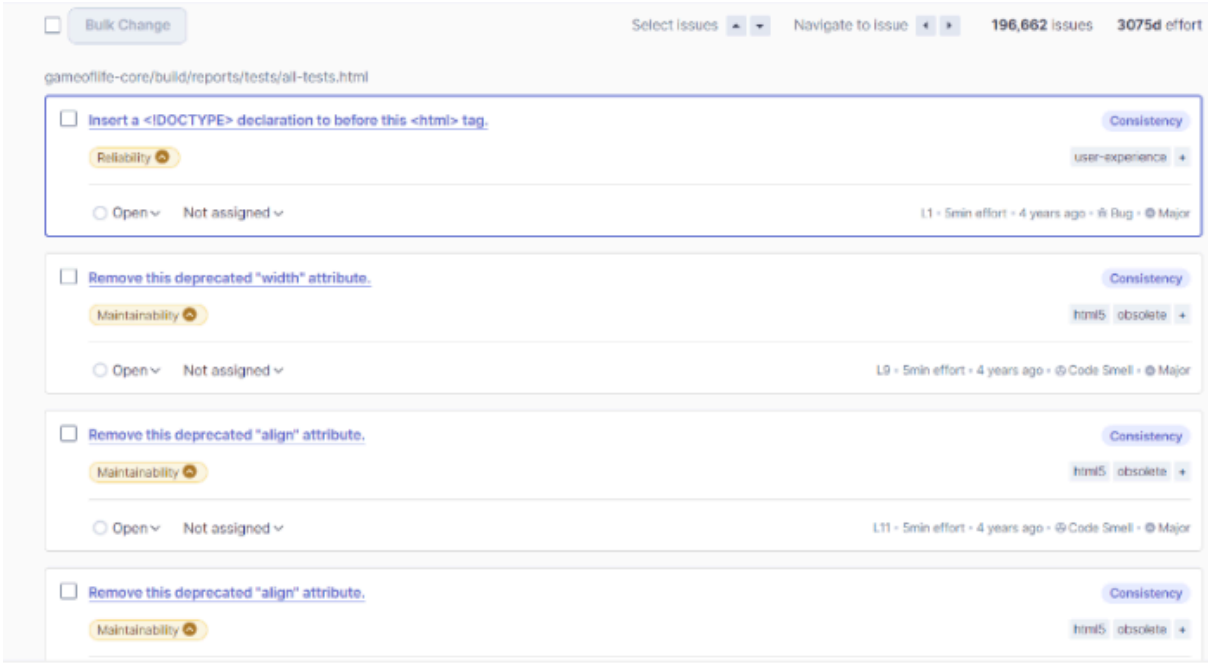


STEP 17: Check the console output

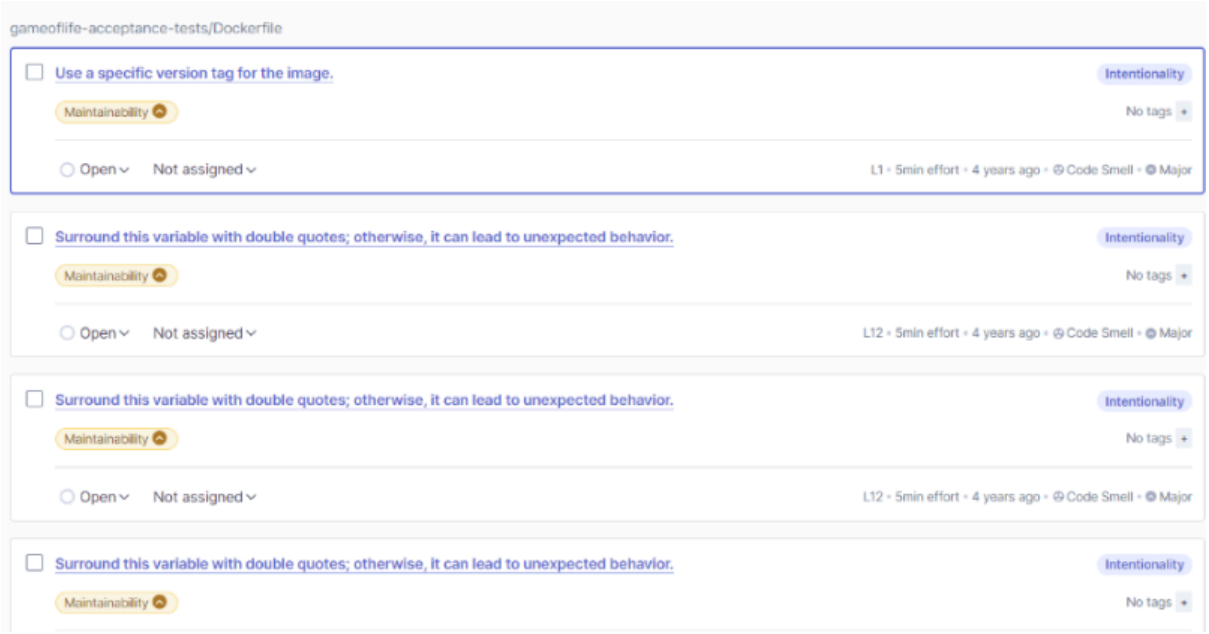
```
09:36:08.069 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
09:38:20.750 INFO  Analysis report generated in 4390ms, dir size=127.2 MB
09:38:45.857 INFO  Analysis report compressed in 25089ms, zip size=29.6 MB
09:38:46.532 INFO  Analysis report uploaded in 675ms
09:38:46.533 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube27
09:38:46.533 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
09:38:46.533 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=46576333-cbde-4277-89d7-471ee554da32
09:39:00.038 INFO  Analysis total time: 15:24.256 s
09:39:00.041 INFO  SonarScanner Engine completed successfully
09:39:00.810 INFO  EXECUTION SUCCESS
09:39:00.811 INFO  Total time: 15:29.301s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

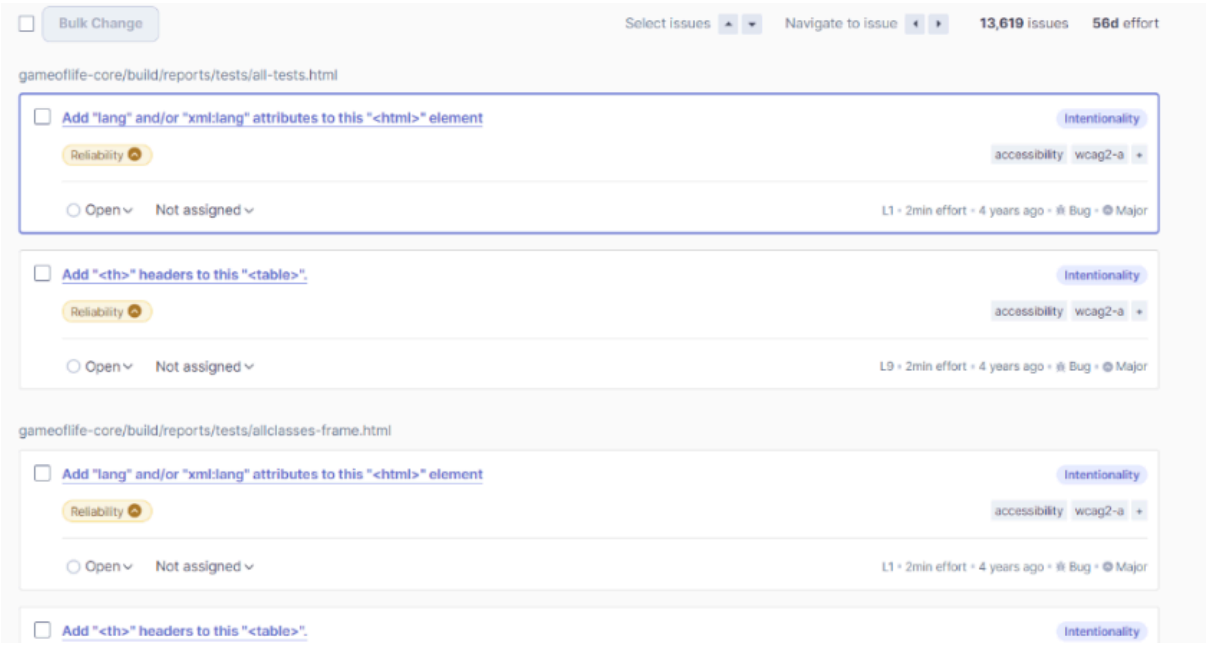
STEP 18: Go to SonarQube and check the project linked.

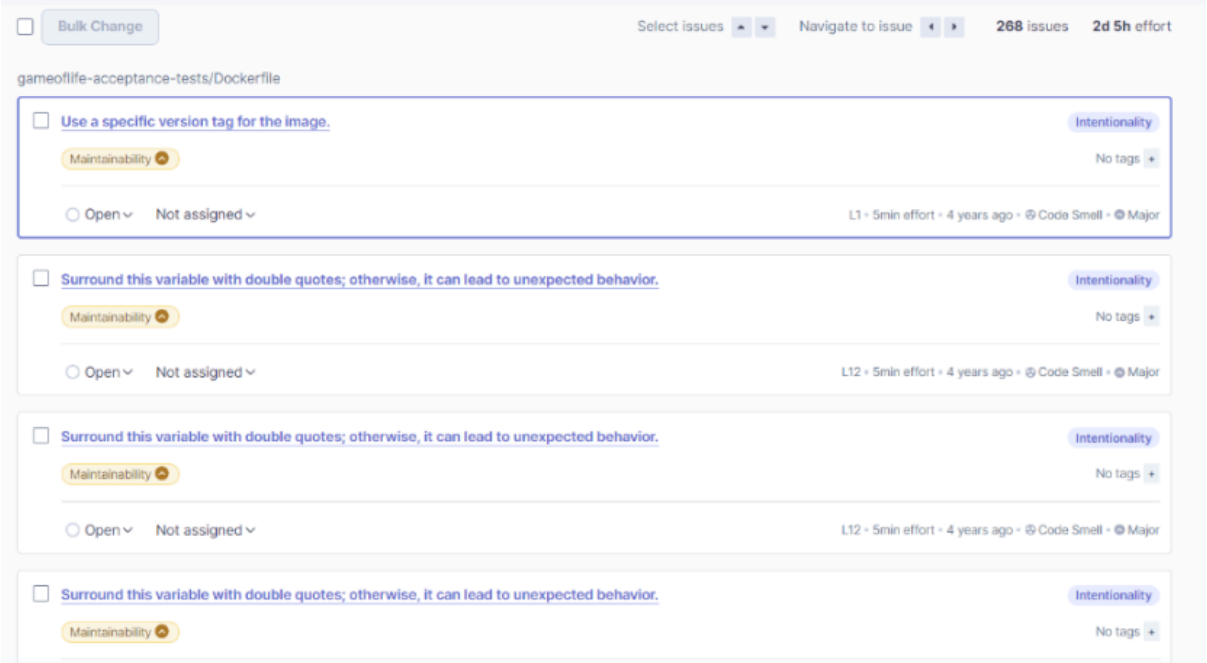


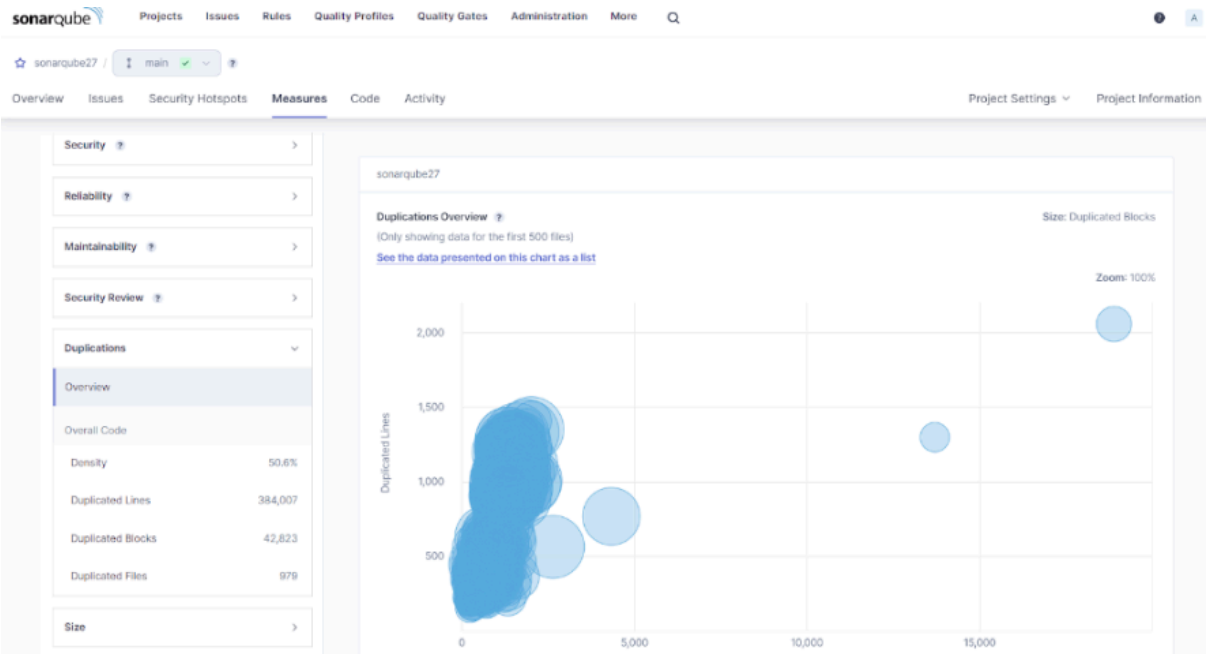


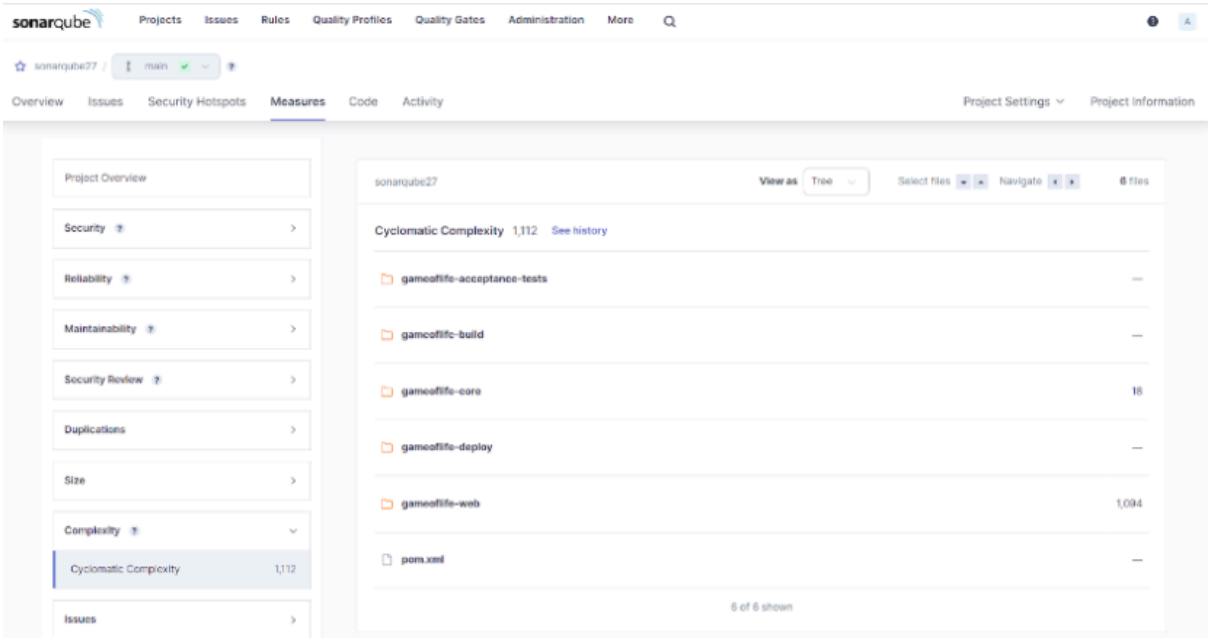
STEP 19: Check all the issues with the code.


gameoflife-acceptance-tests/Dockerfile
Use a specific version tag for the image.
Intentionality
Maintainability
No tags
Open
Not assigned
L1 • 5min effort • 4 years ago • Code Smell • Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 • 5min effort • 4 years ago • Code Smell • Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 • 5min effort • 4 years ago • Code Smell • Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags


Bulk Change
Select issues
Navigate to issue
13,619 issues
56d effort
gameoflife-core/build/reports/tests/all-tests.html
Add "lang" and/or "xml:lang" attributes to this "<html>" element
Intentionality
Reliability
accessibility
wcag2-a
Open
Not assigned
L1 • 2min effort • 4 years ago • Bug • Major
Add "<th>" headers to this "<table>".
Intentionality
Reliability
accessibility
wcag2-a
Open
Not assigned
L9 • 2min effort • 4 years ago • Bug • Major
gameoflife-core/build/reports/tests/allclasses-frame.html
Add "lang" and/or "xml:lang" attributes to this "<html>" element
Intentionality
Reliability
accessibility
wcag2-a
Open
Not assigned
L1 • 2min effort • 4 years ago • Bug • Major
Add "<th>" headers to this "<table>".
Intentionality


Bulk Change
Select issues
Navigate to issue
268 issues
2d 5h effort
gameoflife-acceptance-tests/Dockerfile
Use a specific version tag for the image.
Intentionality
Maintainability
No tags
Open
Not assigned
L1 • 5min effort • 4 years ago • Code Smell • Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 • 5min effort • 4 years ago • Code Smell • Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags
Open
Not assigned
L12 • 5min effort • 4 years ago • Code Smell • Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.
Intentionality
Maintainability
No tags


sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
More
sonarqube27 / main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Security
Reliability
Maintainability
Security Review
Duplications
Overview
Overall Code
Density 50.6%
Duplicated Lines 384,007
Duplicated Blocks 42,823
Duplicated Files 979
Size
sonarqube27
Duplications Overview
(Only showing data for the first 500 files)
See the data presented on this chart as a list
Size: Duplicated Blocks
Zoom: 100%
Duplicated Lines
2,000
1,500
1,000
500
0
5,000
10,000
15,000



Conclusion: We established a Jenkins CI/CD pipeline with SonarQube to perform static analysis on a sample web/Java/Python application. This pipeline automated the detection of bugs, code smells, and security vulnerabilities. During the process, we faced issues with SonarQube server connectivity and Jenkins job configurations. These were resolved by adjusting network settings and refining the pipeline script. The integration successfully enhanced our code quality and security by providing continuous and automated feedback throughout the development lifecycle.