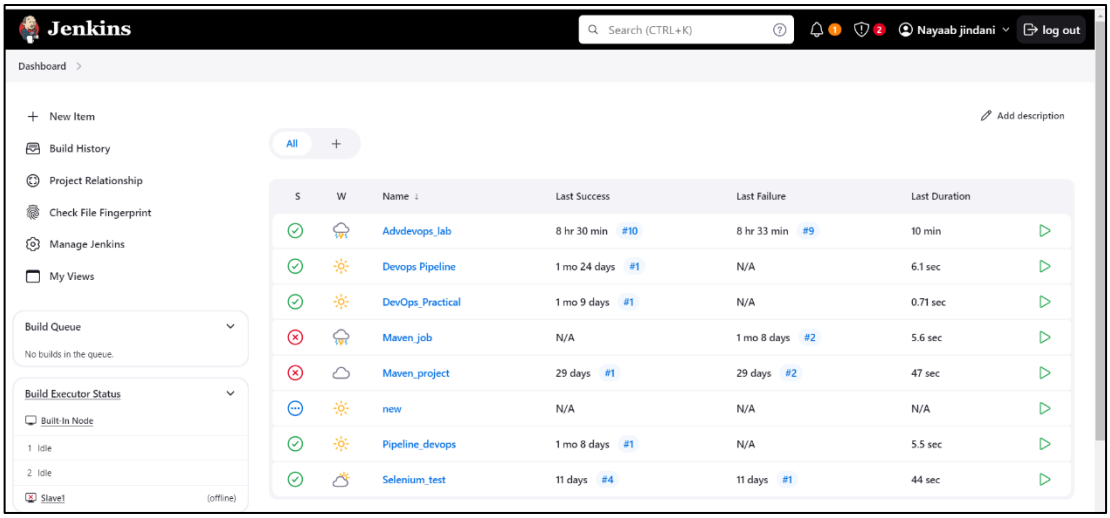
Advance DevOps

Experiment 8

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.

Steps:

1. Open the Jenkins dashboard.



2. First, we will pull the latest version of sonar qube image from the docker hub using the command:
   docker pull sonarqube:latest

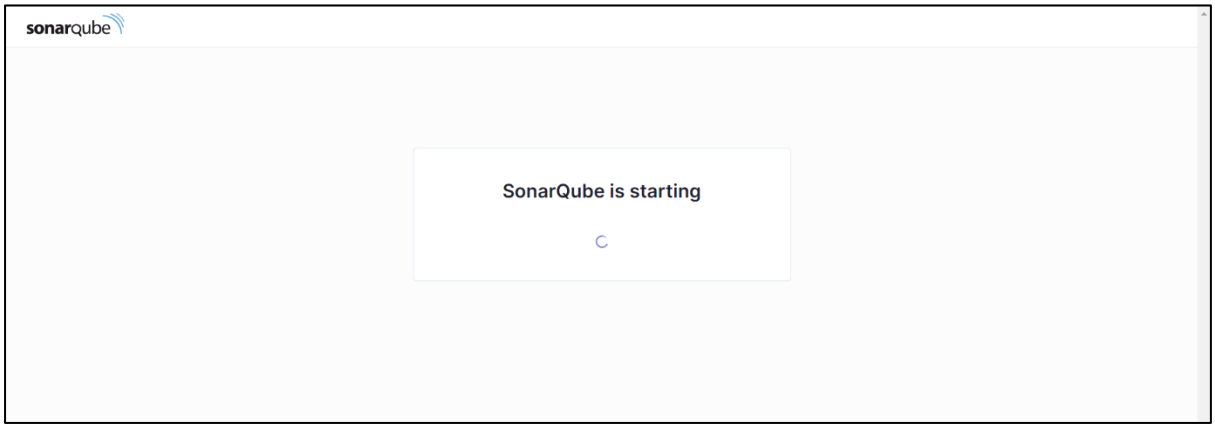
```
docker pull sonarqube:latest
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: 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:latest
```

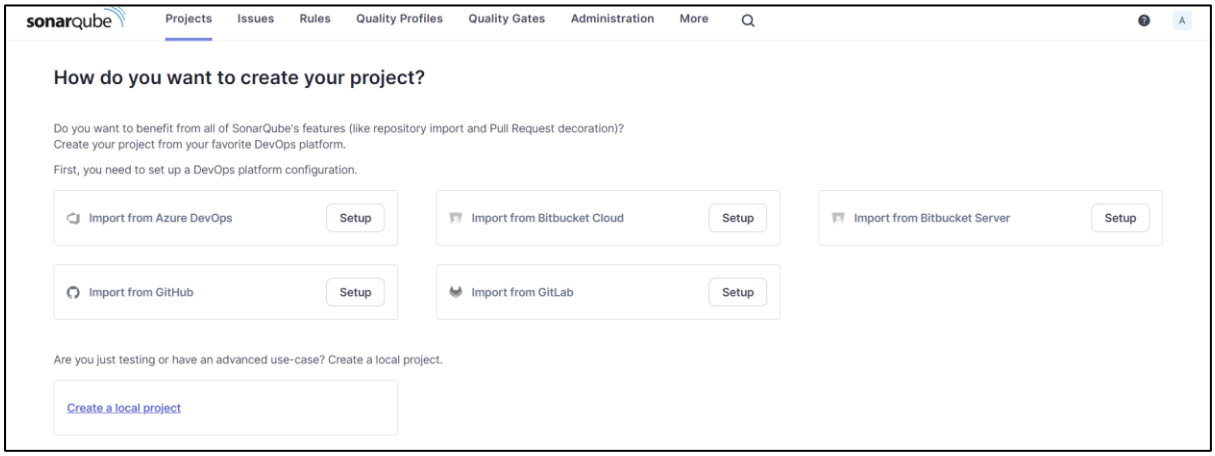
3. Next, we will run SonarQube in a docker container

```
PS C:\Users\Dell> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
67aeea599cfc48e12d50da592eff01d8257f58e6c1bffdf50446066e5f2a8844
```

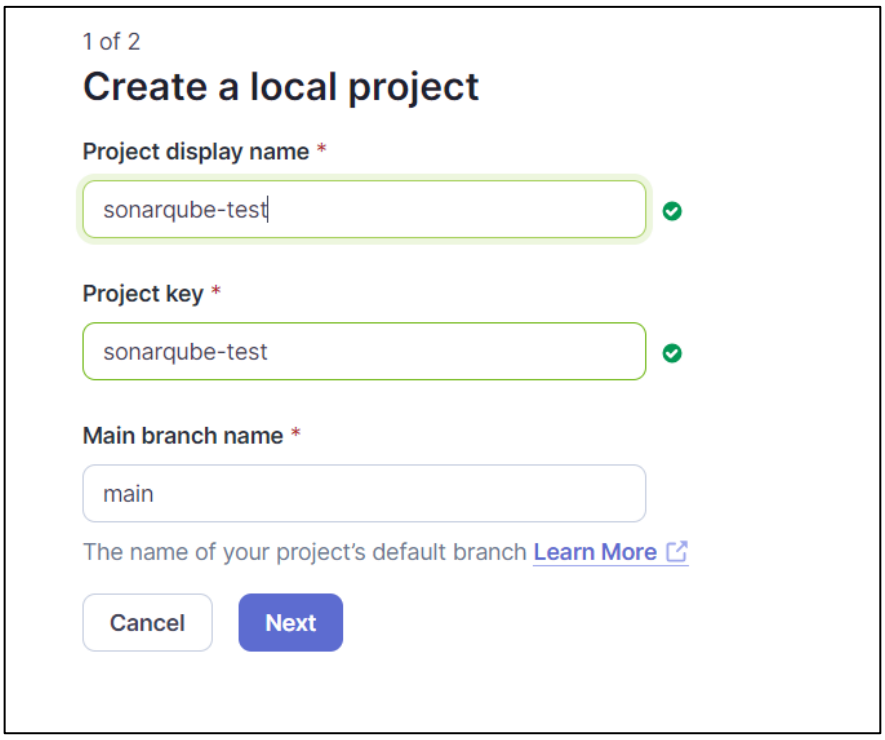
4. Once the container is running, we will check the status of sonar qube on the port 9000. It will show “Sonar qube is starting”



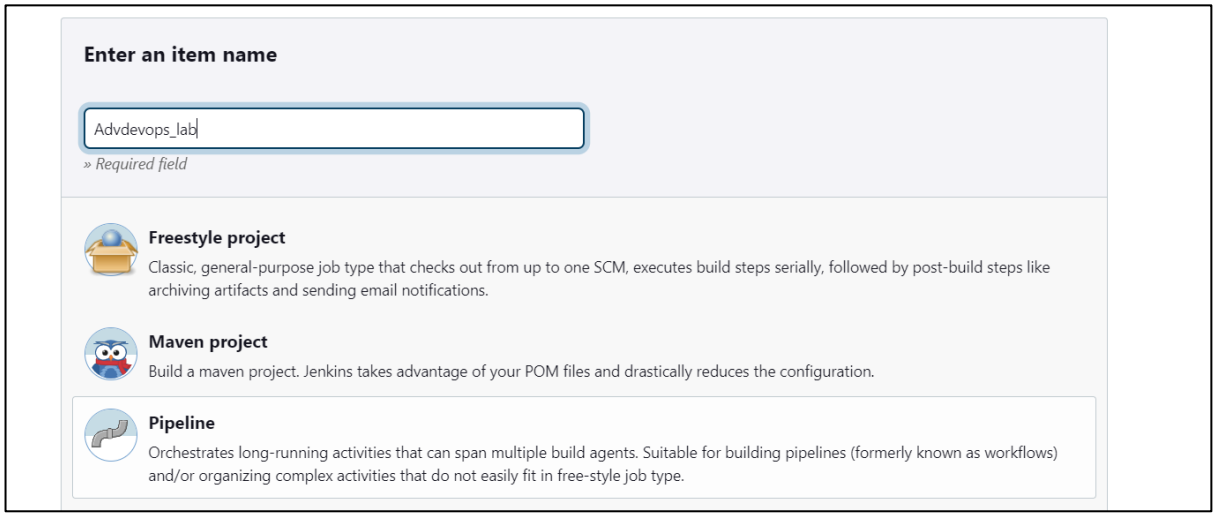
5. Now login to SonarQube using username and password.



6. Click on create a local project option from the dashboard and give a name to the project, click on next and complete the setup.



7. Go to Jenkins dashboard and create a new item by giving a name and select pipeline option.



8. Scroll down to pipeline script and enter the following script:

```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
sh "<PATH_TO_SONARQUBE_FOLDER>//bin//sonar-scanner \
-D sonar.login=<SonarQube_USERNAME> \
-D sonar.password=<SonarQube_PASSWORD> \
-D sonar.projectKey=<Project_KEY> \
-D sonar.exclusions=vendor/**,resources/**,**/*.java \
-D sonar.host.url=http://127.0.0.1:9000/"
}
}
}
```

(Change the path and credentials)

9. Now run the build.

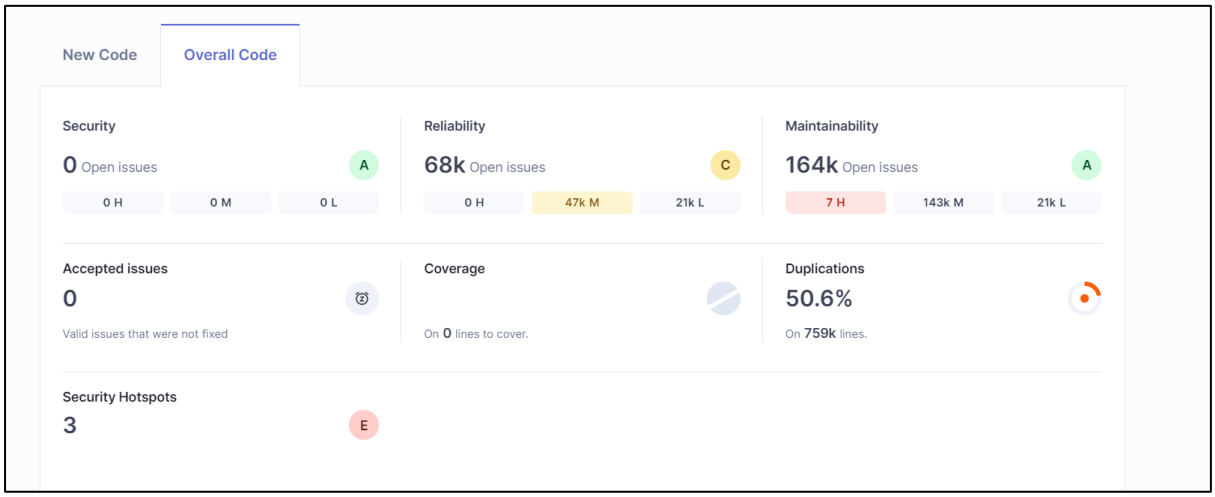
Console Output

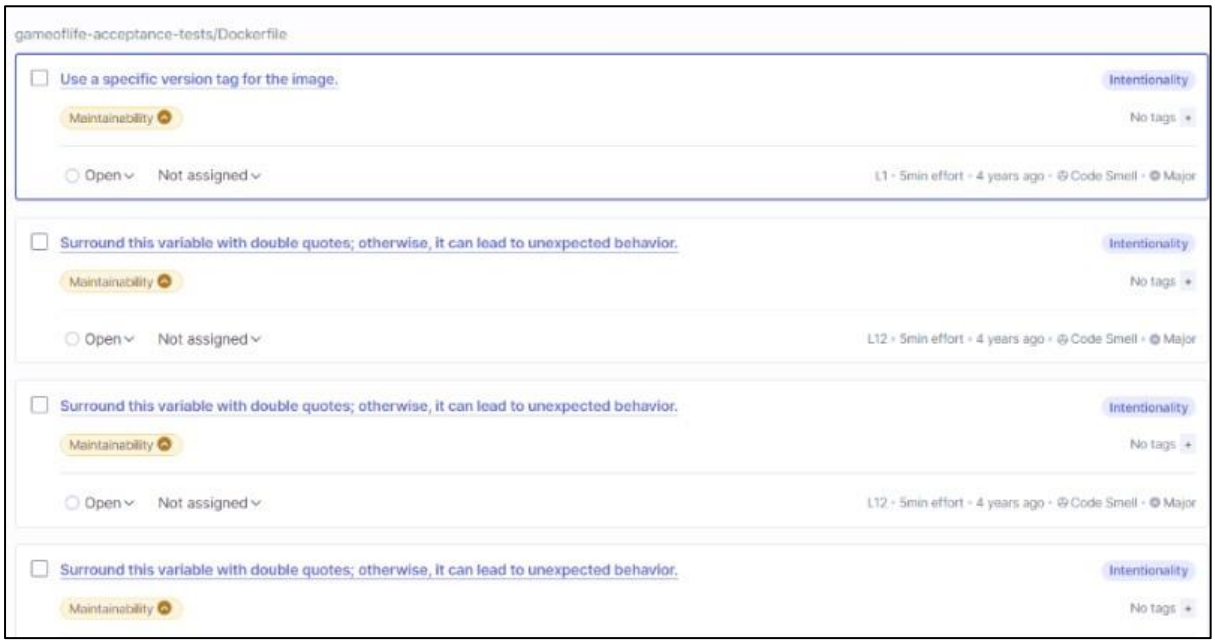
```
Started by user Nayaab jindani
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\Advdevops_lab
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\Advdevops_lab\.git # timeout=10
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/GOL.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/GOL.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.45.2.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe checkout -f ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
 > git.exe branch -a -v --no-abbrev # timeout=10
 > git.exe branch -D master # timeout=10
```

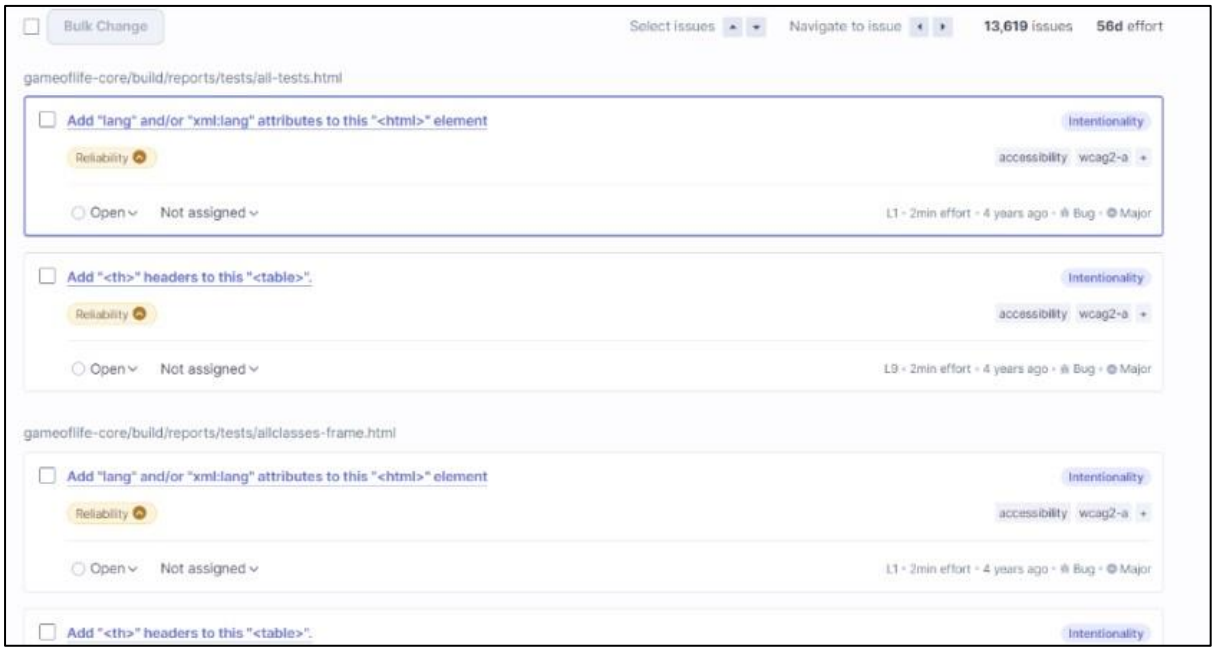
```
for block at line 17. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 296. Keep only the first 100 references.
23:13:58.632 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html
for block at line 75. Keep only the first 100 references.
23:13:58.632 INFO  CPD Executor CPD calculation finished (done) | time=94361ms
23:13:58.695 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
23:15:46.177 INFO  Analysis report generated in 14542ms, dir size=127.2 MB
23:15:55.734 INFO  Analysis report compressed in 9547ms, zip size=29.6 MB
23:15:59.127 INFO  Analysis report uploaded in 3391ms
23:15:59.132 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9000/dashboard?id=sonarqube-test
23:15:59.132 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
23:15:59.132 INFO  More about the report processing at http://127.0.0.1:9000/api/ce/task?id=fbad731f-dcba-45c3-bfdd-b2ed2fec3a9e
23:16:05.629 INFO  Analysis total time: 10:30.120 s
23:16:05.636 INFO  SonarScanner Engine completed successfully
23:16:06.248 INFO  EXECUTION SUCCESS
23:16:06.273 INFO  Total time: 10:47.728s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

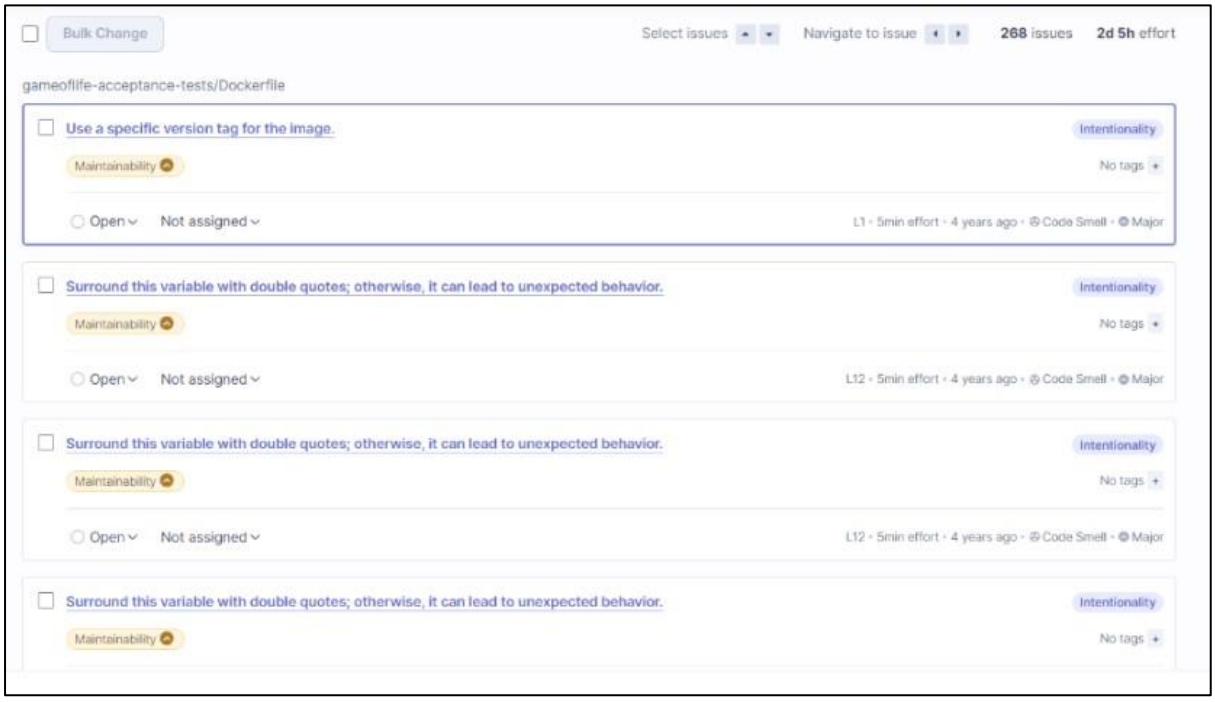
The build is successful.

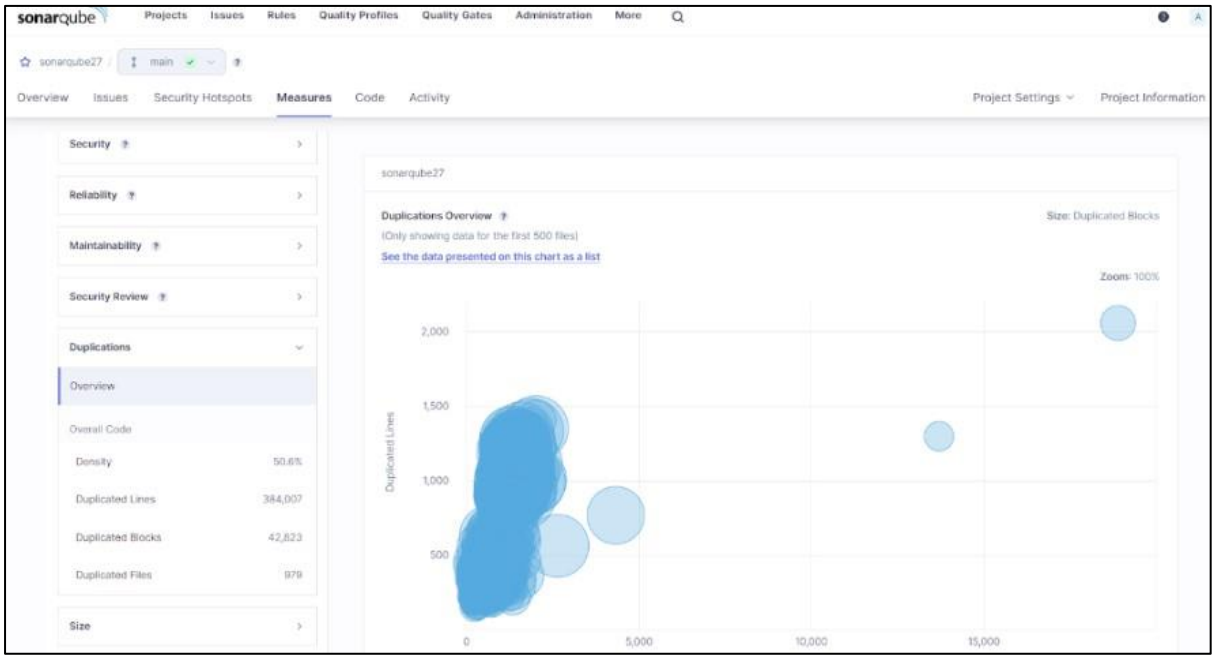
10. Go to sonar qube and check the different errors, code problems, bugs present in the code.






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



Conclusion:

In this experiment we created a Jenkins CICD Pipeline to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample code. It is to be checked whether the sonar scanner plugin is installed in Jenkins or not and also provide the correct path and credentials in the pipeline script or else it leads to the failure of the build.