Snehal Patil D15A 39

Advanced DevOps Expt No:08

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.

Theory:

Static Application Security Testing (SAST) is a methodology that analyzes source code for security vulnerabilities before the code is compiled, often referred to as white box testing.

Problems SAST Solves:

- **Early Detection**: Identifies vulnerabilities in the early stages of the Software Development Life Cycle (SDLC), allowing developers to fix issues without breaking builds or passing vulnerabilities to production.
- **Real-Time Feedback**: Provides immediate insights while coding, which helps in addressing issues proactively.
- **Visual Guidance**: Offers graphical representations of vulnerabilities, indicating their locations and providing detailed guidance on remediation.

Importance of SAST:

- **Efficiency**: Can analyze 100% of the codebase quickly, scanning millions of lines in minutes, unlike manual reviews that are time-consuming.
- **Scalability**: Addresses the challenge of limited security staff by automating vulnerability detection, identifying critical issues like SQL injection and cross-site scripting with high accuracy.

What is SonarQube?

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. Sonar does static code analysis, which provides a detailed report of bugs, code smells, vulnerabilities, code duplications.

Benefits of SonarQube

- Sustainability - Reduces complexity, possible vulnerabilities, and code duplications, optimising the life of applications.
- Increase productivity - Reduces the scale, cost of maintenance, and risk of the application; as such, it removes the need to spend more time changing the code
- Quality code - Code quality control is an inseparable part of the process of software development.
- Detect Errors - Detects errors in the code and alerts developers to fix them automatically before submitting them for output.

Integrating Jenkins with SonarQube:

Prerequisites:

● Jenkins installed

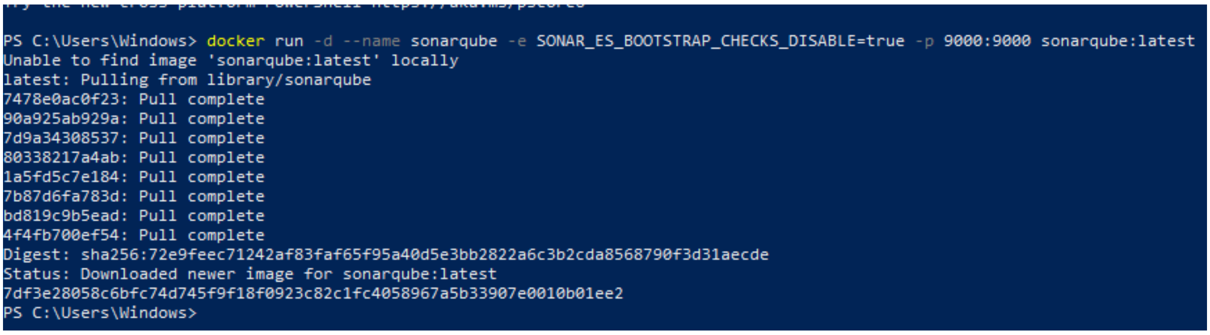
● Docker Installed (for SonarQube)

● SonarQube Docker Image

Steps to create a Jenkins CI/CD Pipeline and use SonarQube to perform

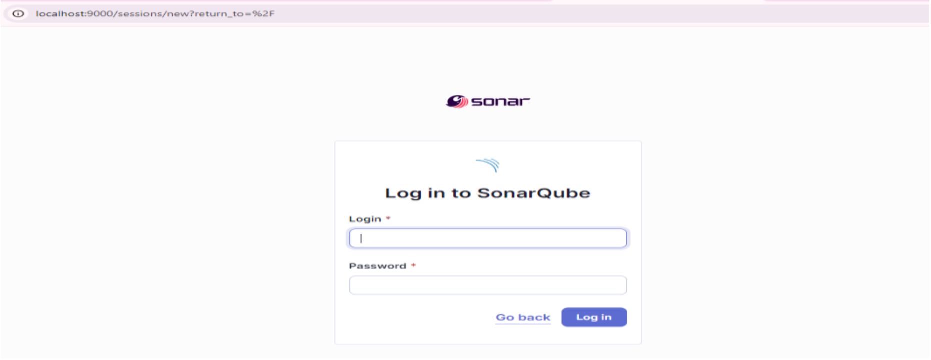
SAST

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command –

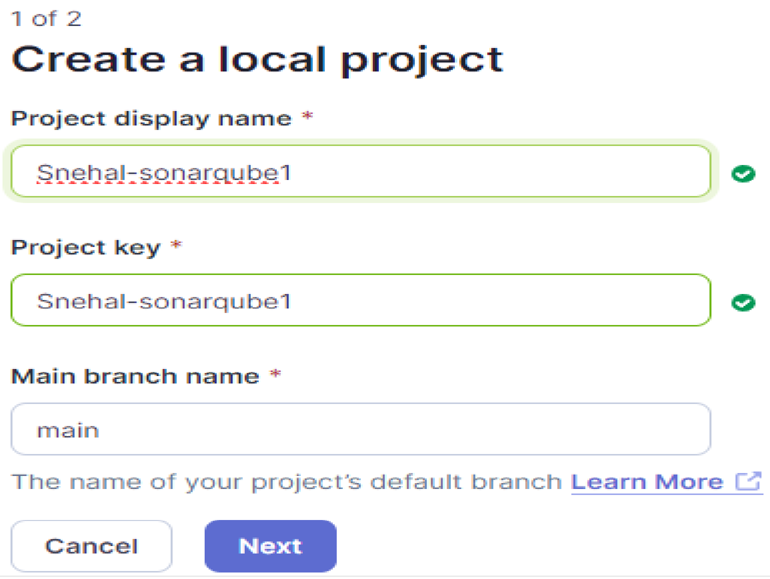


```
PS C:\Users\Windows> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
7df3e28058c6bfc74d745f9f18f0923c82c1fc4058967a5b33907e0010b01ee2
PS C:\Users\Windows>
```

3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.
4. Login to SonarQube using username admin and password admin.



5. Create a manual project in SonarQube with the name sonarqube-test



1 of 2

Create a local project

Project display name *

Snehal-sonarqube1

Project key *

Snehal-sonarqube1

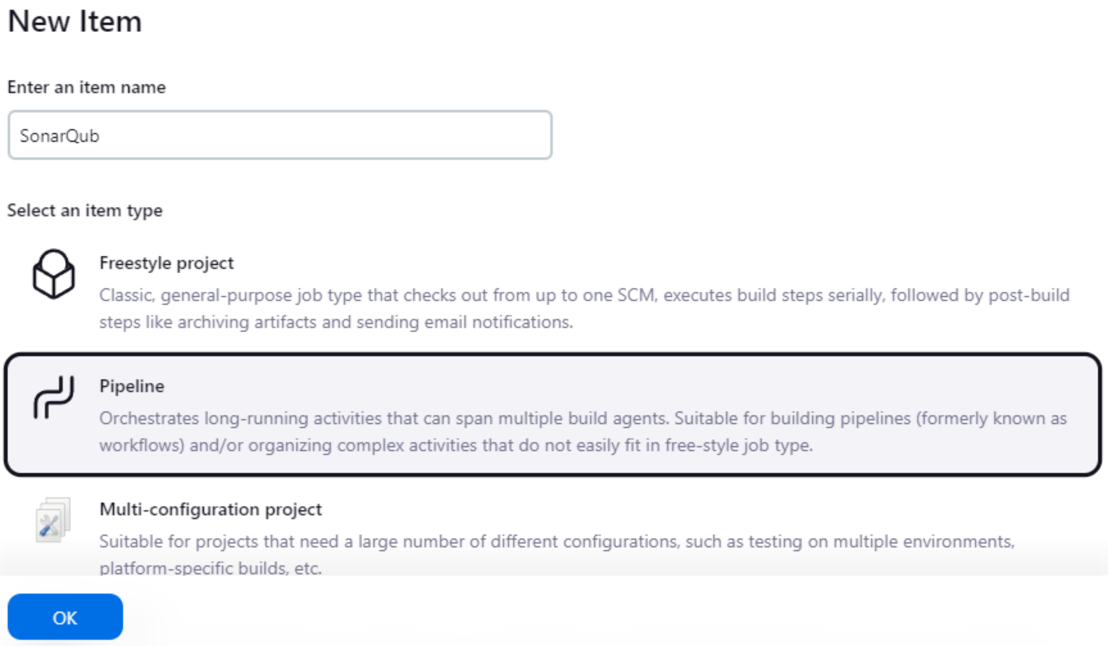
Main branch name *

main

The name of your project's default branch Learn More

5. Setup the project and come back to Jenkins Dashboard.

6. Create a New Item in Jenkins, choose Pipeline.



New Item

Enter an item name

SonarQub

Select an item type

Freestyle project

Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

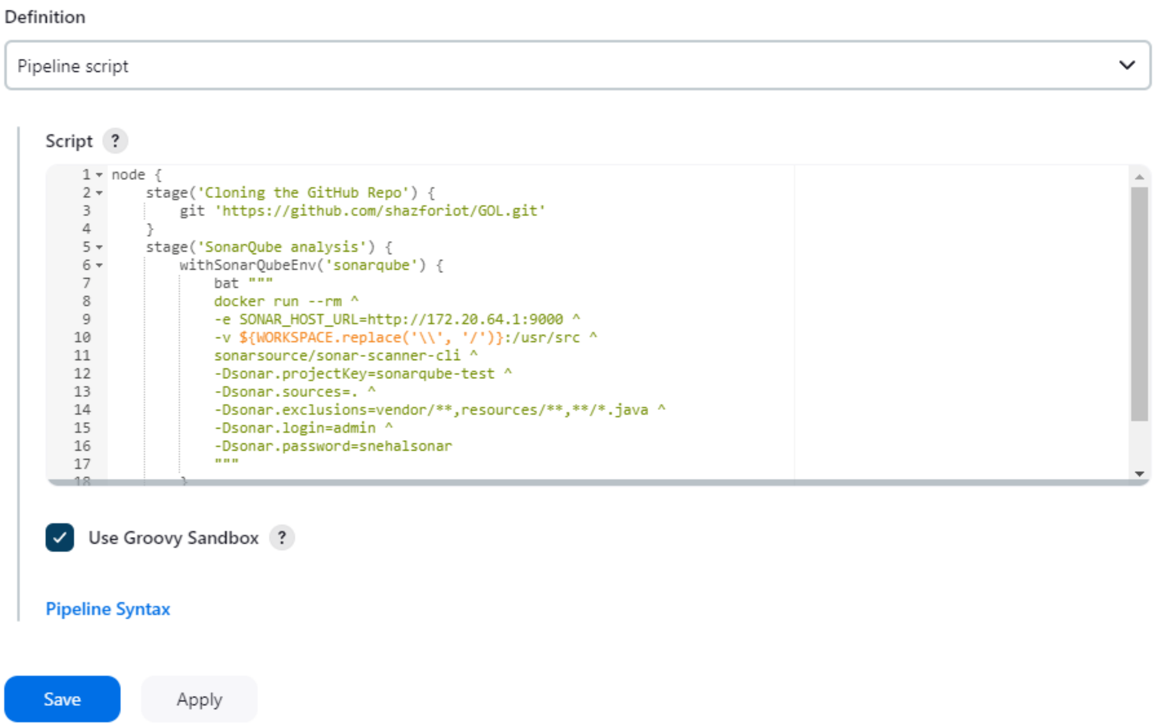
Multi-configuration project

Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

7. Under Pipeline Script, enter the following -

Under Pipeline Script, enter the following -

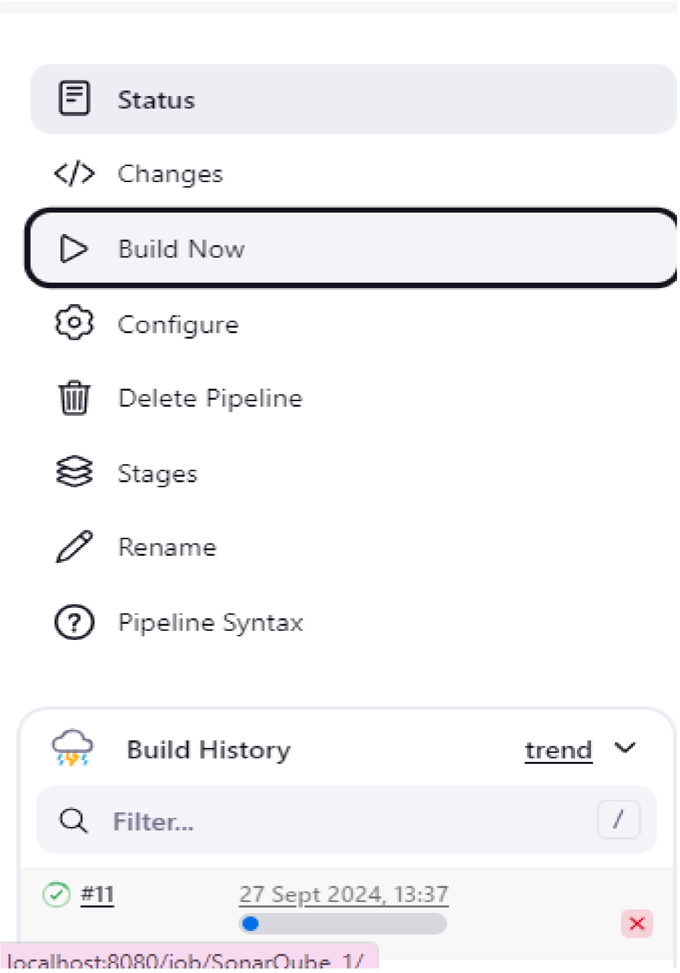
```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
sh "<PATH_TO_SONARQUBE_FOLDER>//bin//sonar-scanner \
-D sonar.login=<SonarQube_USERNAME> \
-D sonar.password=<SonarQube_PASSWORD> \
-D sonar.projectKey=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



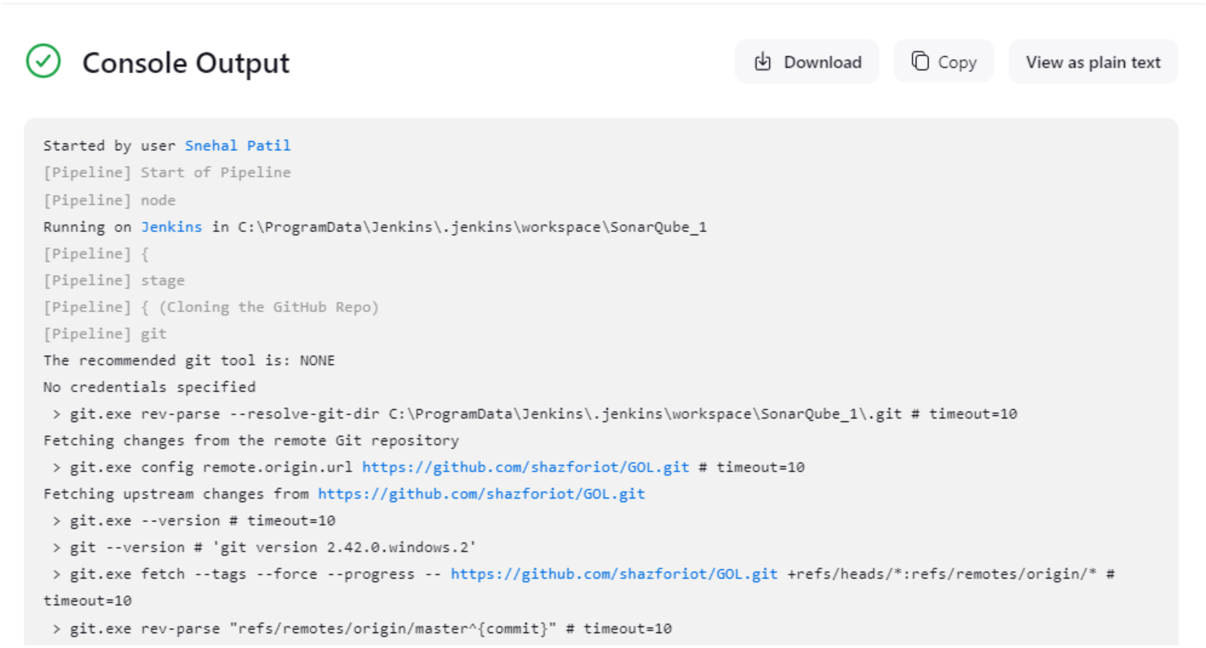
```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
            docker run --rm ^
            -e SONAR_HOST_URL=http://172.20.64.1:9000 ^
            -v ${WORKSPACE.replace('\\', '/')}:/usr/src ^
            sonarsource/sonar-scanner-cli ^
            -Dsonar.projectKey=sonarqube-test ^
            -Dsonar.sources=. ^
            -Dsonar.exclusions=vendor/**,resources/**,**/*.java ^
            -Dsonar.login=admin ^
            -Dsonar.password=snehalsonar
            """
        }
```

It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.

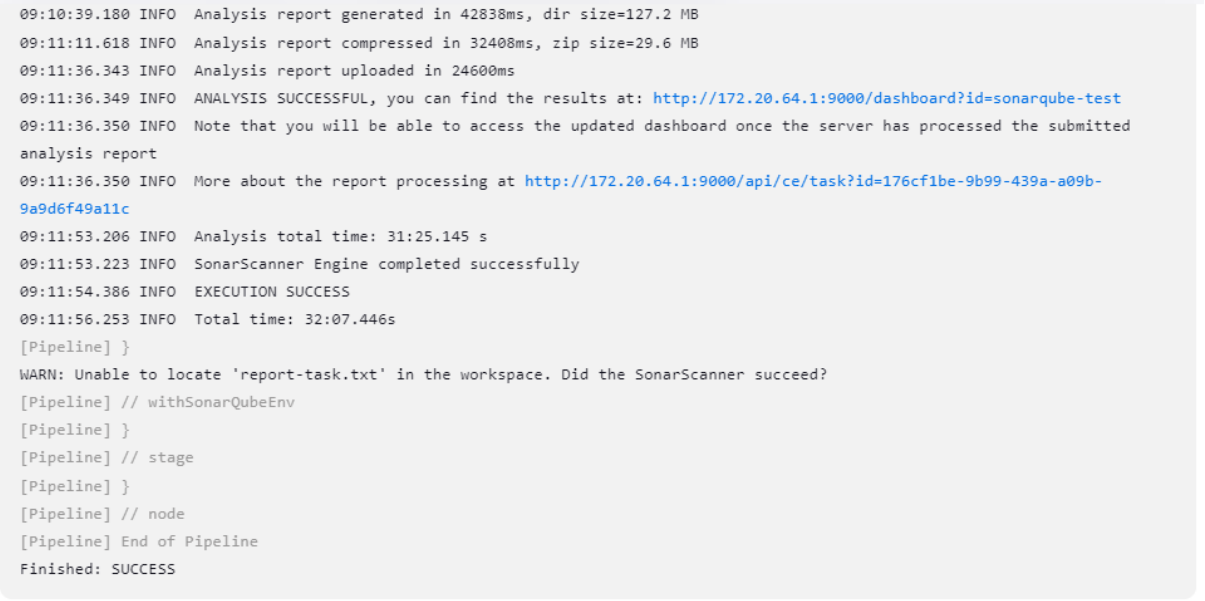
8.Run The Build.



9. Check the console output once the build is complete.

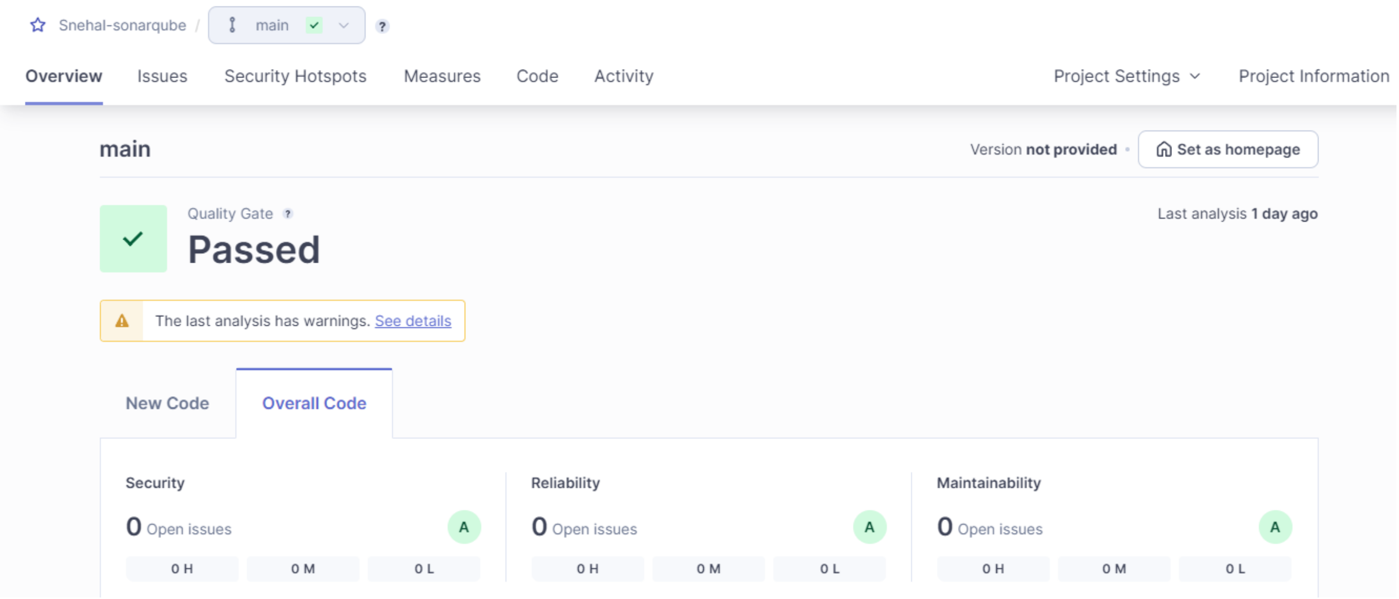


```
Started by user Snehal Patil
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube_1
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Cloning the GitHub Repo)
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube_1\.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 2.42.0.windows.2'
 > 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
```



```
09:10:39.180 INFO  Analysis report generated in 42838ms, dir size=127.2 MB
09:11:11.618 INFO  Analysis report compressed in 32408ms, zip size=29.6 MB
09:11:36.343 INFO  Analysis report uploaded in 24600ms
09:11:36.349 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://172.20.64.1:9000/dashboard?id=sonarqube-test
09:11:36.350 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted
analysis report
09:11:36.350 INFO  More about the report processing at http://172.20.64.1:9000/api/ce/task?id=176cf1be-9b99-439a-a09b-
9a9d6f49a11c
09:11:53.206 INFO  Analysis total time: 31:25.145 s
09:11:53.223 INFO  SonarScanner Engine completed successfully
09:11:54.386 INFO  EXECUTION SUCCESS
09:11:56.253 INFO  Total time: 32:07.446s
[Pipeline] }
WARN: Unable to locate 'report-task.txt' in the workspace. Did the SonarScanner succeed?
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

10. After that, check the project in SonarQube.



main

Version **not provided**

Quality Gate

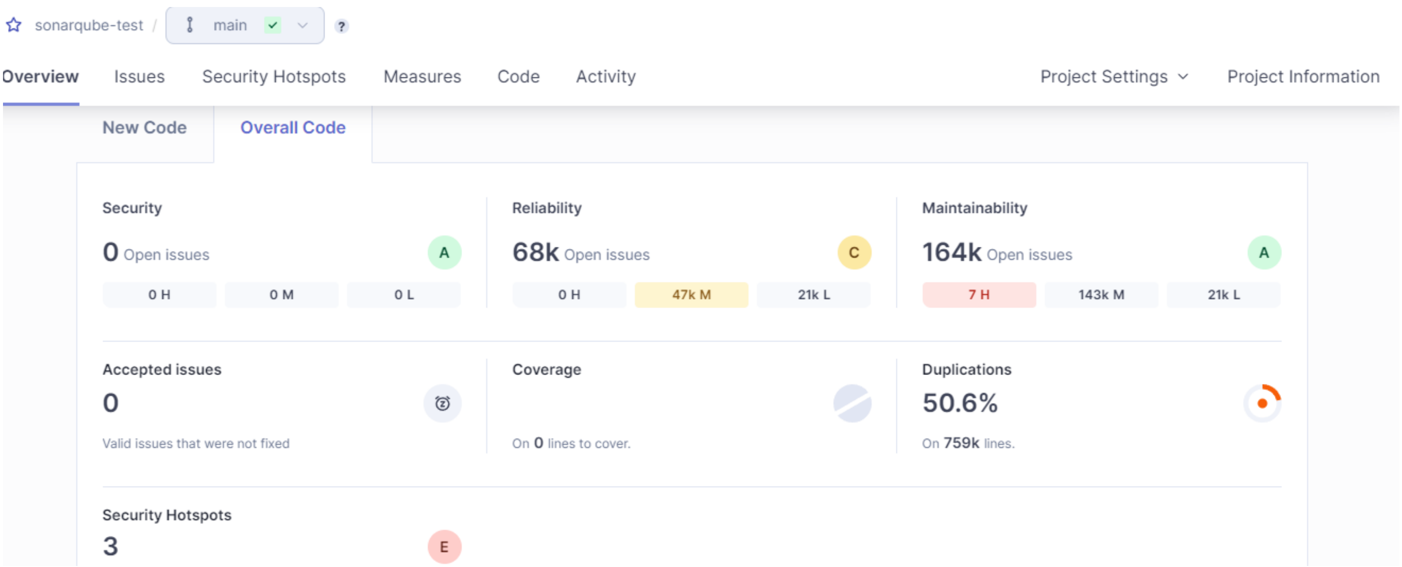
Passed

Last analysis **1 day ago**

The last analysis has warnings. See details

Security	Reliability	Maintainability
0 Open issues A	0 Open issues A	0 Open issues A
0 H · 0 M · 0 L	0 H · 0 M · 0 L	0 H · 0 M · 0 L

Under different tabs, check all different issues with the code.



Security	Reliability	Maintainability
0 Open issues A	68k Open issues C	164k Open issues A
0 H · 0 M · 0 L	0 H · 47k M · 21k L	7 H · 143k M · 21k L

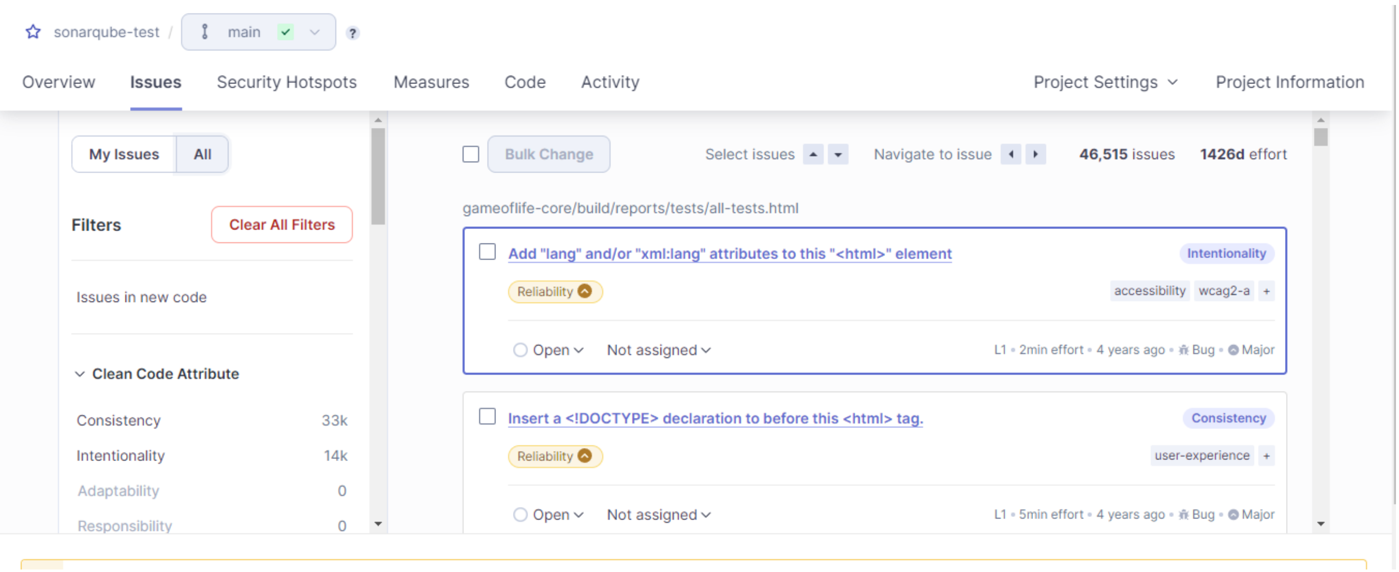
Accepted issues	Coverage	Duplications
0		50.6%
Valid issues that were not fixed	On **0** lines to cover.	On **759k** lines.

Security Hotspots

3 E

11. Code Problems –

Issues:



Clean Code Attribute	
Consistency	33k
Intentionality	14k
Adaptability	0
Responsibility	0

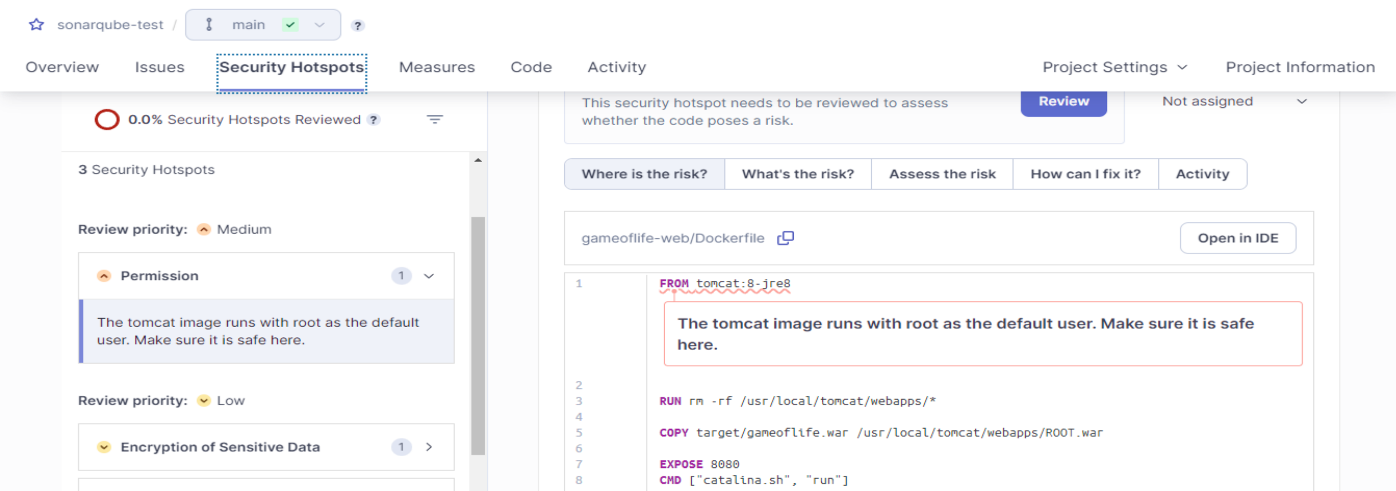
46,515 issues 1426d effort

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

□ Insert a <!DOCTYPE> declaration to before this <html> tag.
Consistency
Reliability
user-experience +
○ Open Not assigned
L1 • 5min effort • 4 years ago • Bug • Major

Security hotspots:



0.0% Security Hotspots Reviewed

This security hotspot needs to be reviewed to assess whether the code poses a risk.

3 Security Hotspots

Review priority: Medium

Permission 1

The tomcat image runs with root as the default user. Make sure it is safe here.

Review priority: Low

Encryption of Sensitive Data 1

gameoflife-web/Dockerfile

```
FROM tomcat:8-jre8
```

The tomcat image runs with root as the default user. Make sure it is safe here.

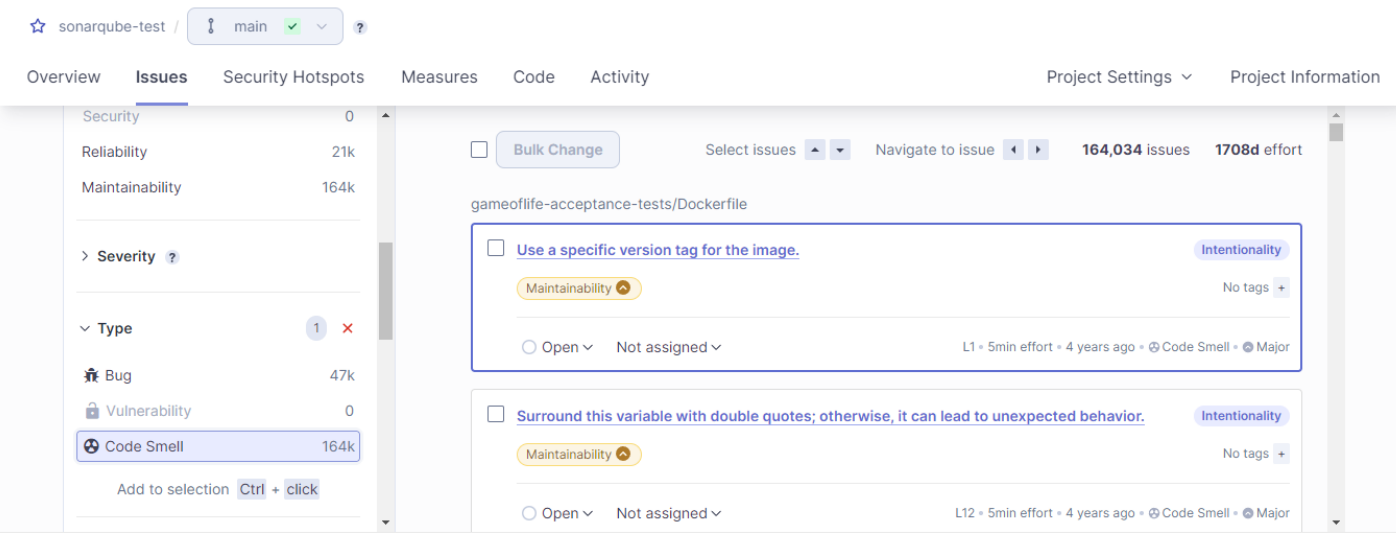
```

RUN rm -rf /usr/local/tomcat/webapps/*

COPY target/gameoflife.war /usr/local/tomcat/webapps/ROOT.war

EXPOSE 8080
CMD ["catalina.sh", "run"]
```

Codesmells:



Security	0
Reliability	21k
Maintainability	164k

Type	1
Bug	47k
Vulnerability	0
Code Smell	164k

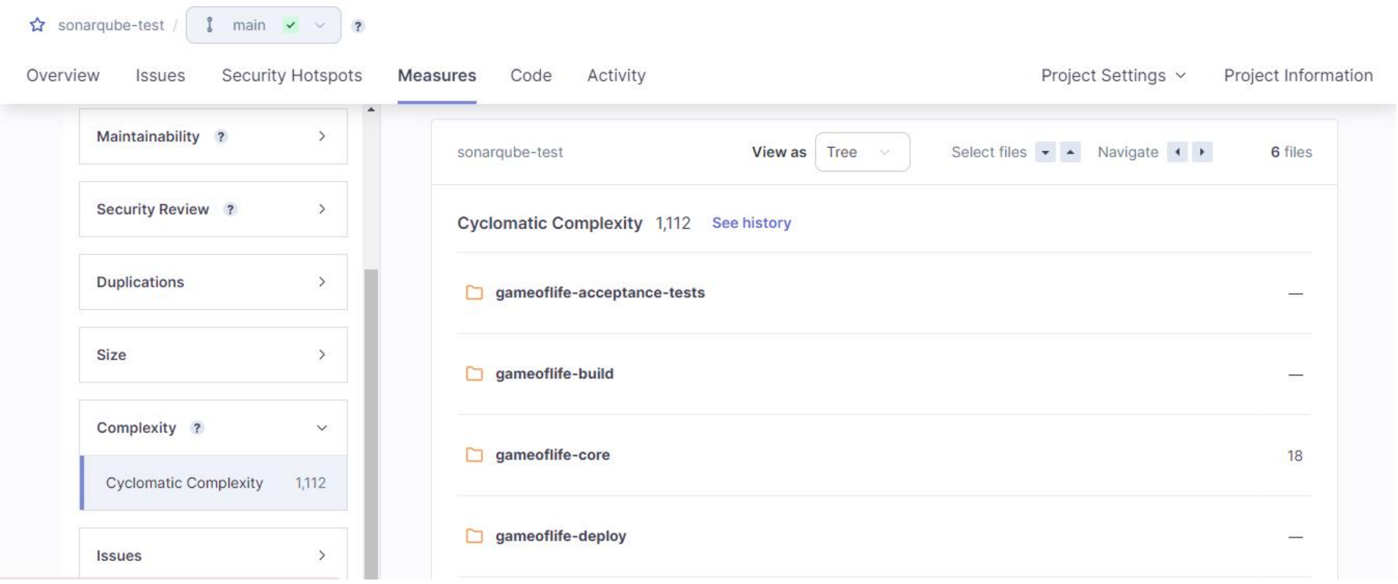
164,034 issues 1708d effort

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

Complexity:

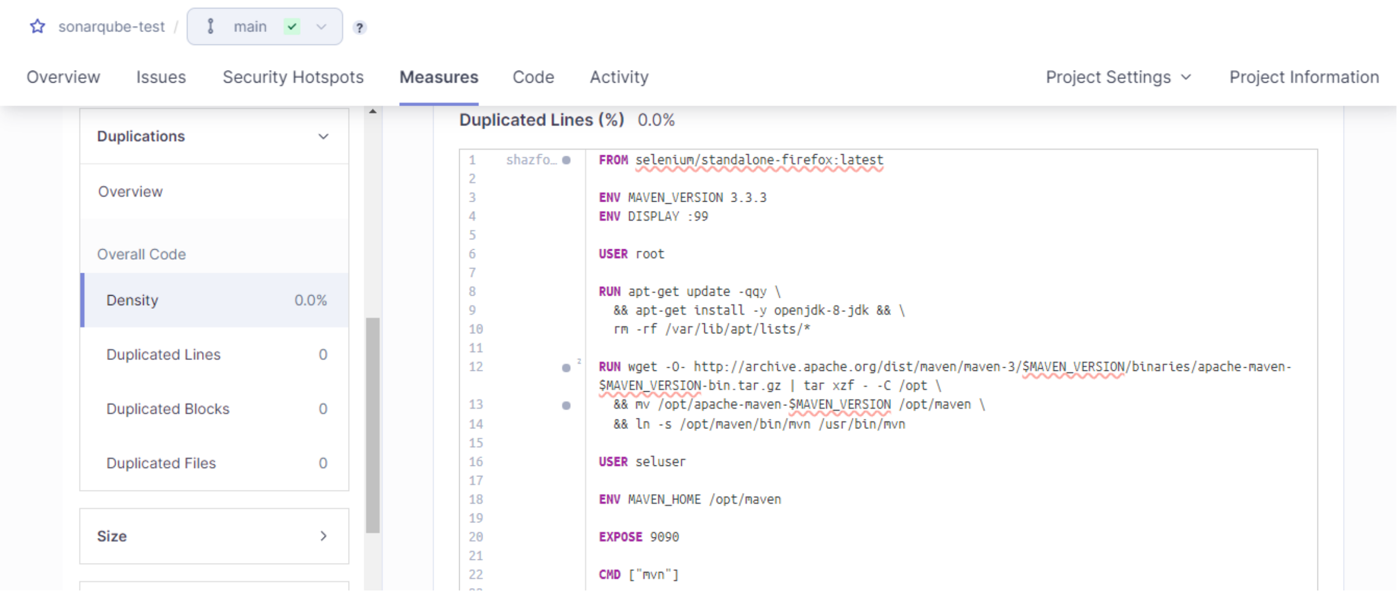


Cyclomatic Complexity 1,112

sonarqube-test — 6 files

Cyclomatic Complexity	1,112
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—

Duplications:



Duplicated Lines (%) 0.0%

Density	0.0%
Duplicated Lines	0
Duplicated Blocks	0
Duplicated Files	0

```
FROM selenium/standalone-firefox:latest

ENV MAVEN_VERSION 3.3.3
ENV DISPLAY :99

USER root

RUN apt-get update -qqy \
  && apt-get install -y openjdk-8-jdk && \
  rm -rf /var/lib/apt/lists/*

RUN wget -O- http://archive.apache.org/dist/maven/maven-3/$MAVEN_VERSION/binaries/apache-maven-$MAVEN_VERSION-bin.tar.gz | tar xzf - -C /opt \
  && mv /opt/apache-maven-$MAVEN_VERSION /opt/maven \
  && ln -s /opt/maven/bin/mvn /usr/bin/mvn

USER seluser

ENV MAVEN_HOME /opt/maven

EXPOSE 9090

CMD ["mvn"]
```