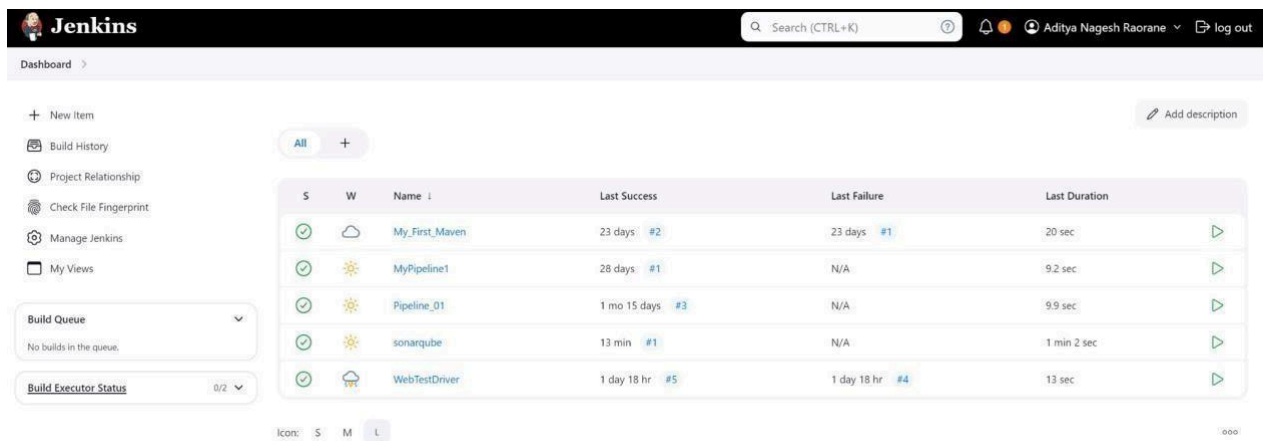


Aim: Create a Jenkins CICD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

1. Open up Jenkins Dashboard on localhost:8080.

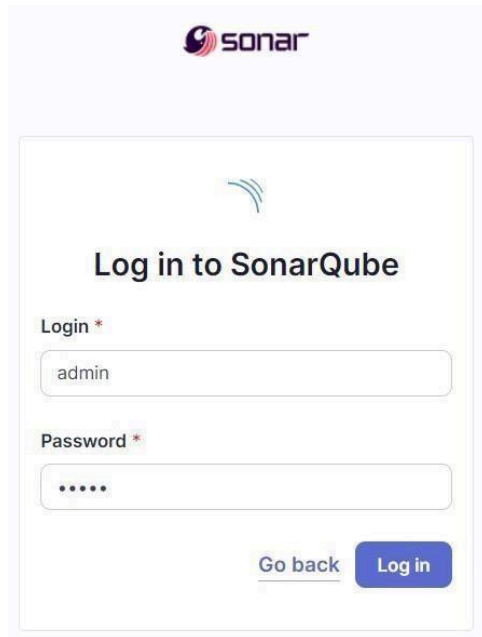


2. Run SonarQube in a Docker container using this command: a] `docker -v` b] `docker pull sonarqube` c] `docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest`

```
C:\Users\Neeraj>docker -v
Docker version 27.0.3, build 7d4bcd8

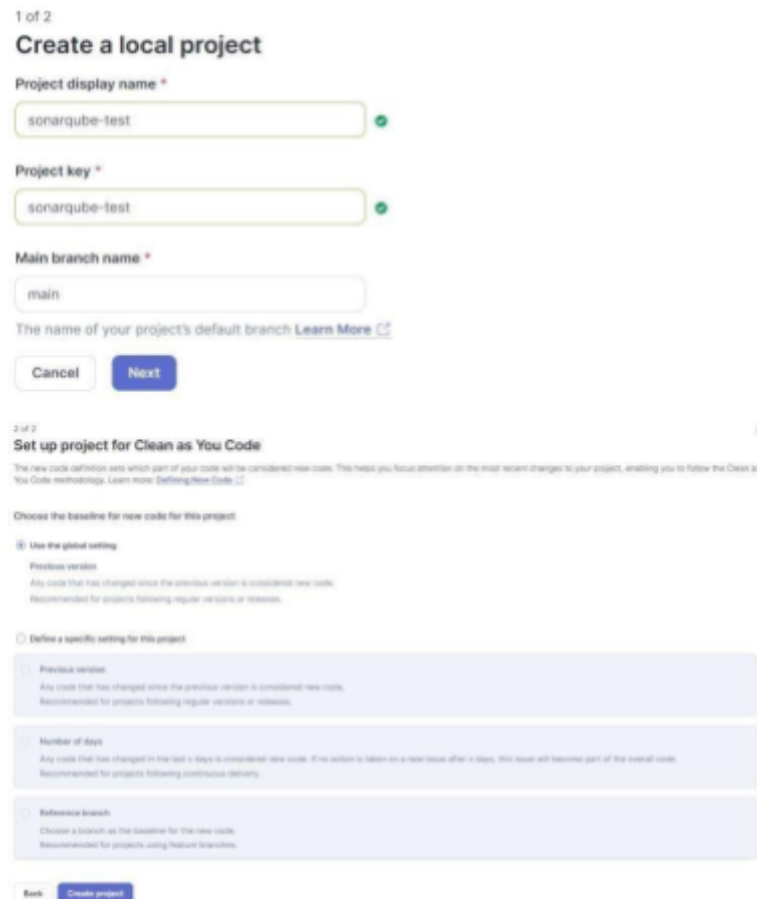
C:\Users\aditya>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecd
Status: Downloaded newer image for sonarqube:latest
4a6e73f4472de892b1ddead1abe77372a85a7b09408cce3a0abd37c5ab6b49a4
```

3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is **“admin”** and the password is **“mus12”**.



The image shows the SonarQube login page. At the top is the Sonar logo. Below it is a large box with the title "Log in to SonarQube". There are two input fields: "Login *" with the value "admin" and "Password *" with masked characters. At the bottom right of the box are two buttons: "Go back" (a link) and "Log in" (a button).

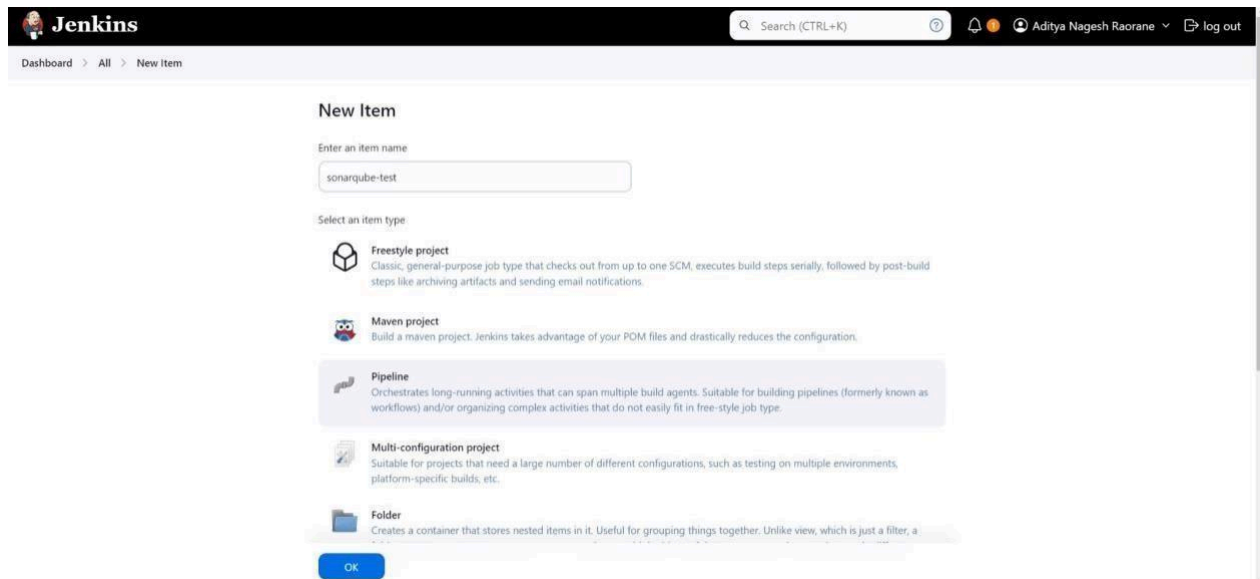
4. Create a local project in SonarQube with the name **sonarqube-test**.



The image shows the "Create a local project" page in SonarQube. It is a two-step process. Step 1 of 2: "Create a local project". It has three input fields: "Project display name *" (sonarqube-test), "Project key *" (sonarqube-test), and "Main branch name *" (main). Below these is a link "The name of your project's default branch Learn More". At the bottom are "Cancel" and "Next" buttons. Step 2 of 2: "Set up project for Clean as You Code". It has a sub-header "Choose the baseline for new code for this project". There are three radio button options: "Use the global setting" (selected), "Previous version", and "Define a specific setting for this project". Under "Define a specific setting for this project", there are three sub-options: "Previous version", "Number of days", and "Reference branch". At the bottom are "Back" and "Create project" buttons.

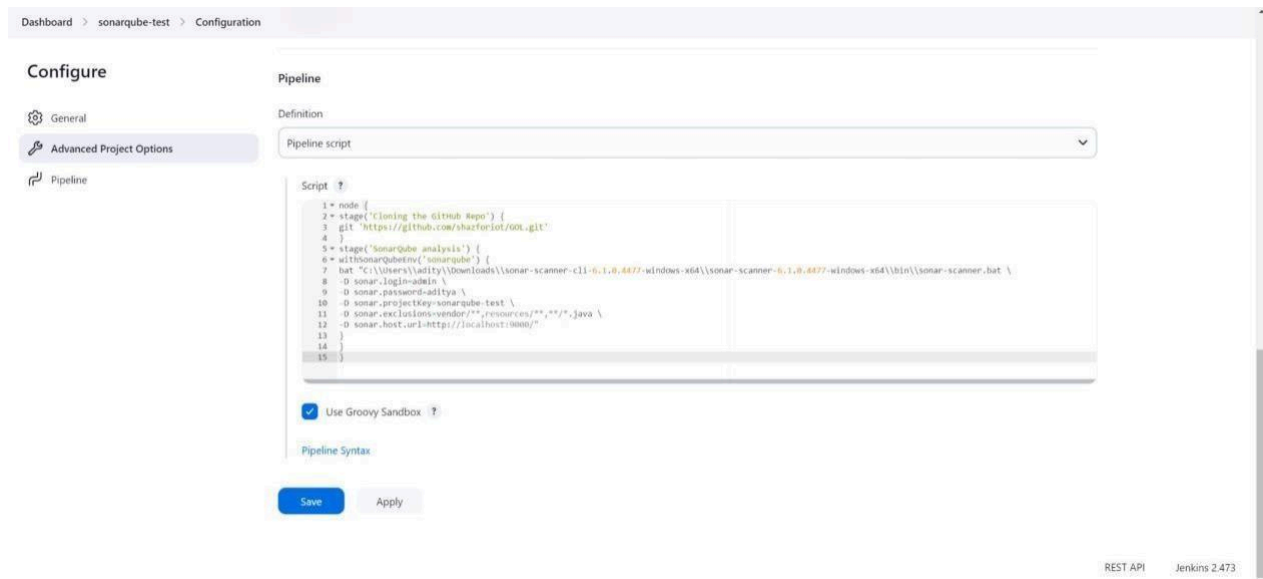
Setup the project and come back to Jenkins Dashboard.

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



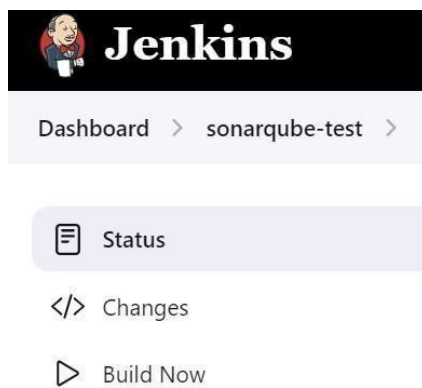
7. Under **Pipeline Script**, enter the following -

```
node { stage('Cloning the GitHub Repo')
    {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') { withSonarQubeEnv('sonarqube')
    { bat
        "C:\\Users\\adity\\Downloads\\sonar-scanner-cli-6.1.0.4477-windows-x64\\sonar-s
        canner-6.1.0.4477-windows-x64\\bin\\sonar-scanner.bat \
        -D sonar.login=<YOUR ID> \
        -D sonar.password=<YOUR PASSWORD> \
        -D sonar.projectKey=<YOUR PROJECT KEY> \
        -D sonar.exclusions=vendor/**,resources/**,**/*.java \
        -D sonar.host.url=http://localhost:9000/"
    }
    }
}
```



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.

```
Dashboard > sonarqube-test > #1

line 41. Keep only the first 100 references.
21:37:59.929 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at
line 17. Keep only the first 100 references.
21:37:59.929 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.
21:37:59.929 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.
21:37:59.930 INFO CPD Executor CPD calculation finished (done) | time=15336ms
21:37:59.955 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1ef5e4'
21:40:14.276 INFO Analysis report generated in 515ms, dir size=127.2 MB
21:40:35.678 INFO Analysis report compressed in 21588ms, zip size=29.6 MB
21:40:36.178 INFO Analysis report uploaded in 492ms
21:40:36.173 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
21:40:36.173 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:40:36.173 INFO More about the report processing at http://localhost:9000/api/cv/task?id=99fcd1e5-df4e-4f9f-8688-3169741e0856
21:40:53.466 INFO Analysis total time: 14:32.336 s
21:40:53.468 INFO SonarScanner Engine completed successfully
21:40:54.148 INFO EXECUTION SUCCESS
21:40:54.150 INFO Total time: 14:35.645s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.473

10. After that, check the project in SonarQube.

The screenshot displays the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The main content area shows a list of projects, with 'sonarqube-test' highlighted. The project details for 'sonarqube-test' are shown, indicating a 'Passed' status and a last analysis time of 10 minutes ago. The project has 68k Lines of Code, 164k Open Issues, and 50.6% Coverage. The interface also shows a 'Quality Gate' status of 'Passed' and a 'Security' status of '0 Open Issues'.

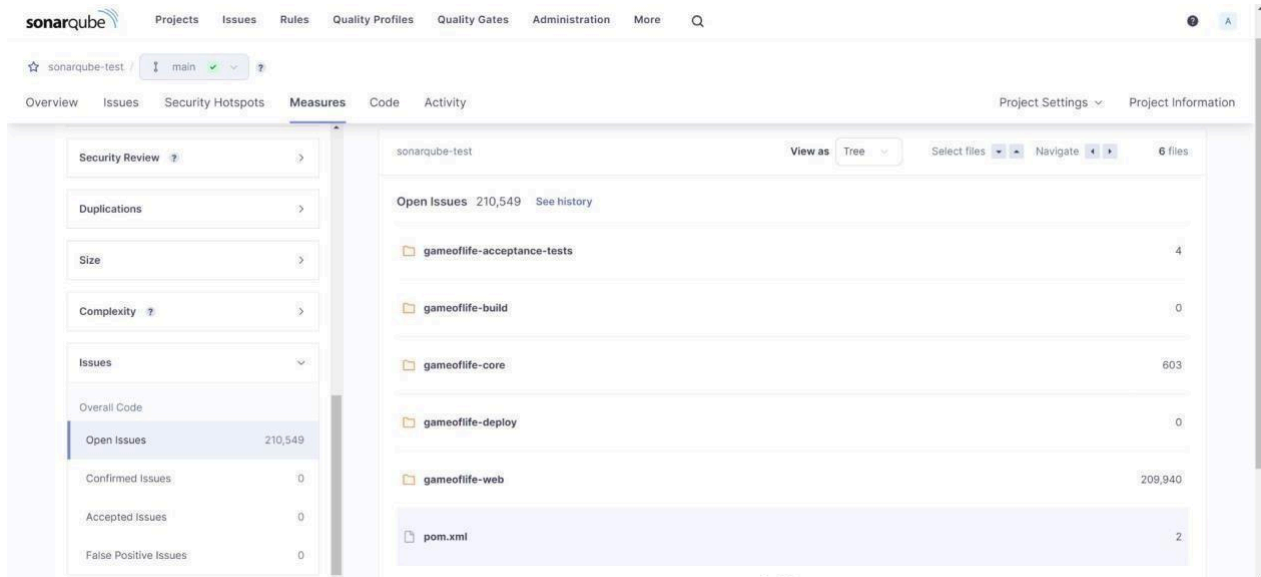
sonarqube-test **Passed**
Last analysis: 10 minutes ago
68k Lines of Code - HTML, XML, ...
Security: 0 Open Issues, Reliability: 68k Open Issues, Maintainability: 164k Open Issues, Hotspots Reviewed: 0.0%, Coverage: 50.6%, Duplications: 50.6%

sonarqube-test **Passed**
Last analysis: 10 minutes ago
68k Lines of Code - HTML, XML, ...
Security: 0 Open Issues, Reliability: 68k Open Issues, Maintainability: 164k Open Issues, Hotspots Reviewed: 0.0%, Coverage: 50.6%, Duplications: 50.6%

sonarqube-test **Passed**
Last analysis: 10 minutes ago
68k Lines of Code - HTML, XML, ...
Security: 0 Open Issues, Reliability: 68k Open Issues, Maintainability: 164k Open Issues, Hotspots Reviewed: 0.0%, Coverage: 50.6%, Duplications: 50.6%

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

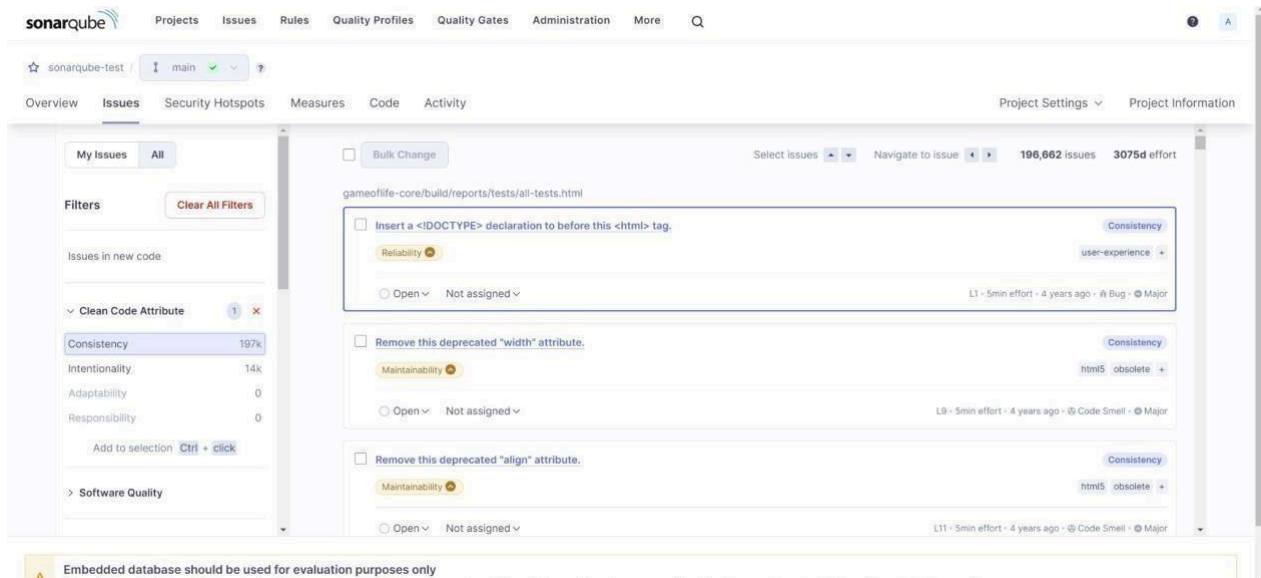
11. Code Problems Open Issues



The screenshot shows the SonarQube interface with the 'Measures' tab selected. The left sidebar displays a tree view of the project structure under 'sonarqube-test'. The main area shows a list of measures for the 'gameoflife' project. The 'Open Issues' measure is highlighted, showing 210,549 issues. The table lists the following measures:

Measure	Value
gameoflife-acceptance-tests	4
gameoflife-build	0
gameoflife-core	603
gameoflife-deploy	0
gameoflife-web	209,940
pom.xml	2

Consistency



The screenshot shows the SonarQube interface with the 'Issues' tab selected. The left sidebar displays a list of filters, including 'Clean Code Attribute' with 197k issues. The main area shows a list of issues for the 'gameoflife-core/build/reports/tests/all-tests.html' file. The 'Consistency' issue is highlighted, showing 196,662 issues and 3075d effort. The table lists the following issues:

Issue	Category	Severity	Effort	Age	Impact
Insert a <!DOCTYPE> declaration to before this <html> tag.	Reliability	Open	5min	4 years ago	Bug - Major
Remove this deprecated "width" attribute.	Maintainability	Open	5min	4 years ago	Code Smell - Major
Remove this deprecated "align" attribute.	Maintainability	Open	5min	4 years ago	Code Smell - Major

Intentionality

The screenshot displays the SonarQube web interface for a project named 'sonarqube-test'. The 'Issues' tab is active, showing a list of 13,887 issues with a total effort of 59d. The left sidebar contains filters for 'My Issues' (All), 'Clean Code Attribute' (1), and 'Software Quality'. The main content area shows three issues related to 'gameoflife-acceptance-tests/Dockerfile'. Each issue is an 'Intentionality' type, categorized as 'Maintainability', and is marked as 'Open' and 'Not assigned'. The issues are: 'Use a specific version tag for the image.' (L1), 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (L12), and another identical issue (L12). A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only'.

Code Smells

The screenshot displays the SonarQube web interface for the same project 'sonarqube-test'. The 'Issues' tab is active, showing a list of 253 issues with a total effort of 2d 5h. The left sidebar contains filters for 'Severity' (High: 0, Medium: 0, Low: 253), 'Type' (Bug: 14k, Vulnerability: 0, Code Smell: 253), and 'Scope'. The main content area shows three issues related to 'gameoflife-web/tools/jmeter/printable_docs'. Each issue is a 'Code Smell' type, categorized as 'Reliability', and is marked as 'Open' and 'Not assigned'. The issues are: 'Add an "alt" attribute to this image.' (L29), 'Add an "alt" attribute to this image.' (L31), and 'Add an "alt" attribute to this image.' (L31). A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only'.

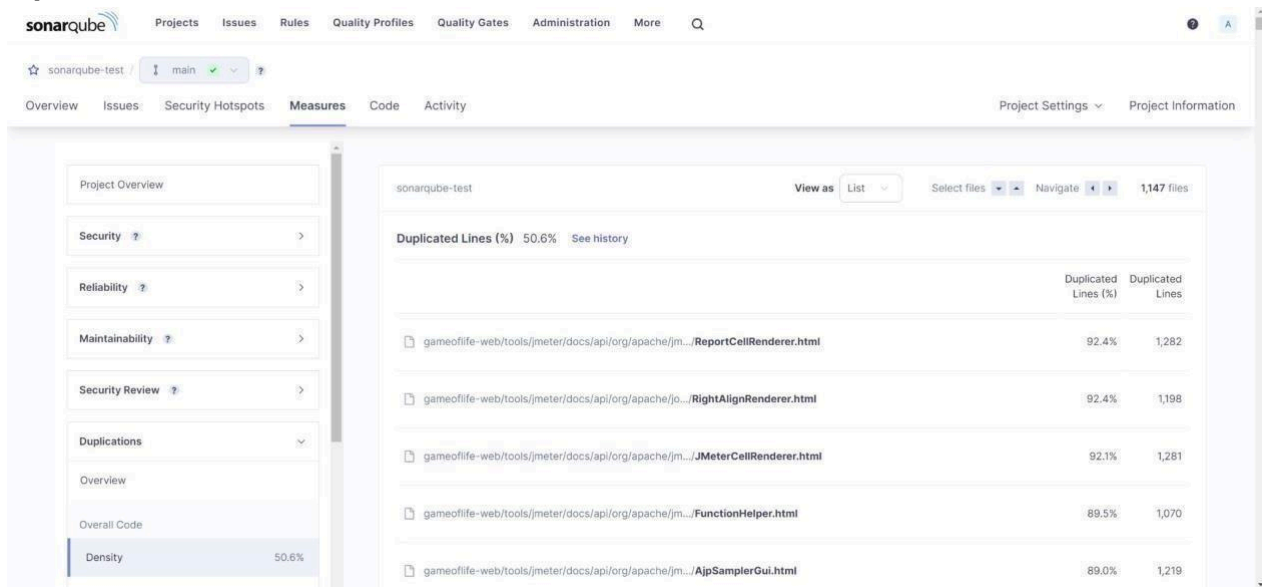
Bugs

The screenshot shows the SonarQube interface with the 'Issues' tab selected. The left sidebar displays filters for Severity (High, Medium, Low) and Type (Bug, Vulnerability, Code Smell). The main area shows a list of issues, including 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' and 'Add "<th>" headers to this "<table>"'. The issues are categorized by 'Intentionality' and 'Reliability'. The bottom of the page features a warning message: 'Embedded database should be used for evaluation purposes only'.

Reliability

The screenshot shows the SonarQube interface with the 'Issues' tab selected. The left sidebar displays filters for Clean Code Attribute (Consistency, Intentionality, Adaptability, Responsibility) and Software Quality (Security, Reliability, Maintainability). The main area shows a list of issues, including 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' and 'Add "<th>" headers to this "<table>"'. The issues are categorized by 'Intentionality' and 'Reliability'. The bottom of the page features a warning message: 'Embedded database should be used for evaluation purposes only'.

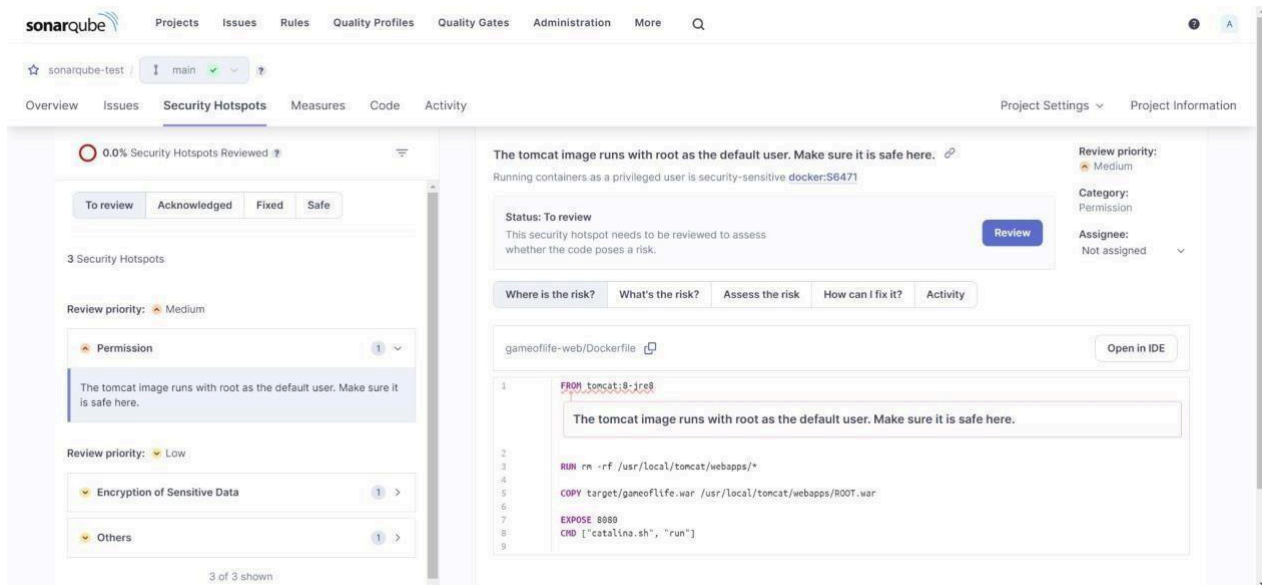
Duplicates



The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Measures' tab is selected, and the 'Density' measure is highlighted in the left sidebar, showing a value of 50.6%. The main area displays a table of duplicated lines. The table has columns for file names, duplicated lines percentage, and duplicated lines count. The files listed are:

File Name	Duplicated Lines (%)	Duplicated Lines
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../ReportCellRenderer.html	92.4%	1,282
gameoflife-web/tools/jmeter/docs/api/org/apache/jo.../RightAlignRenderer.html	92.4%	1,198
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../JMeterCellRenderer.html	92.1%	1,281
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../FunctionHelper.html	89.5%	1,070
gameoflife-web/tools/jmeter/docs/api/org/apache/jm.../AjpSamplerGui.html	89.0%	1,219

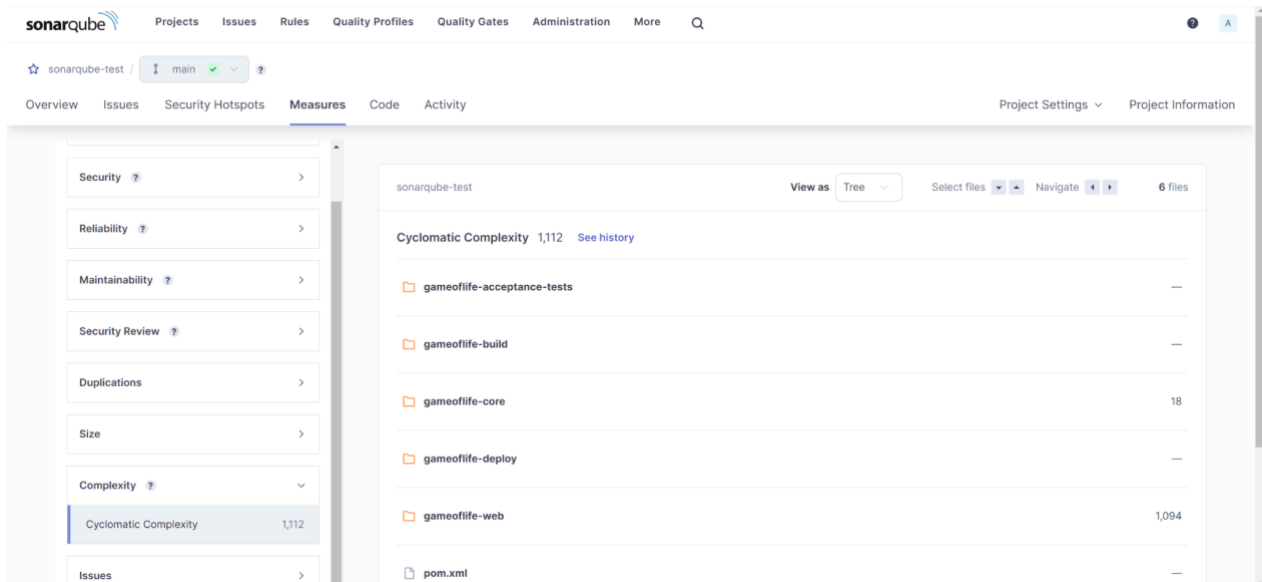
Security Hotspot



The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Security Hotspots' tab is selected. The left sidebar shows a summary of 3 security hotspots, with a review priority of Medium. The main area displays a detailed view of a security hotspot. The title is 'The tomcat image runs with root as the default user. Make sure it is safe here.' The status is 'To review'. The description states: 'This security hotspot needs to be reviewed to assess whether the code poses a risk.' The code snippet shown is:

```
1 FROM tomcat:8-jre8
2
3 RUN rm -rf /usr/local/tomcat/webapps/*
4
5 COPY target/gameoflife.war /usr/local/tomcat/webapps/ROOT.war
6
7 EXPOSE 8080
8 CMD ["catalina.sh", "run"]
9
```

Cyclomatic Complexity



In this way, we have created a CI/CD Pipeline with Jenkins and integrated it with SonarQube to find issues in the code like bugs, code smells, duplicates, cyclomatic complexities, etc.

Conclusion:

In this experiment, we performed a static analysis of the code to detect bugs, code smells, and security vulnerabilities on our sample Java application.