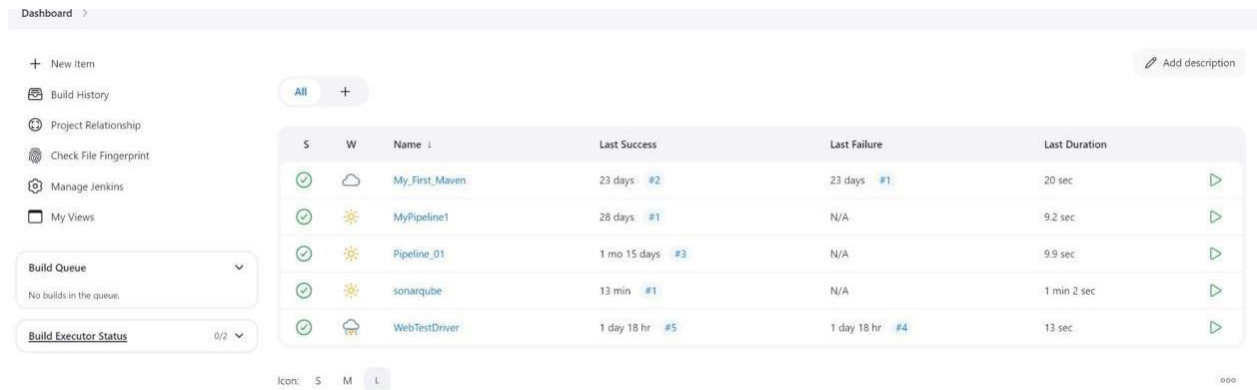


Aim: Create a Jenkins CI/CD Pipeline with SonarQube / GitLab Integration to perform a static analysis of the code to detect bugs, code smells, and security vulnerabilities on a sample Web / Java / Python application.

1. Open up Jenkins Dashboard on localhost:8080.

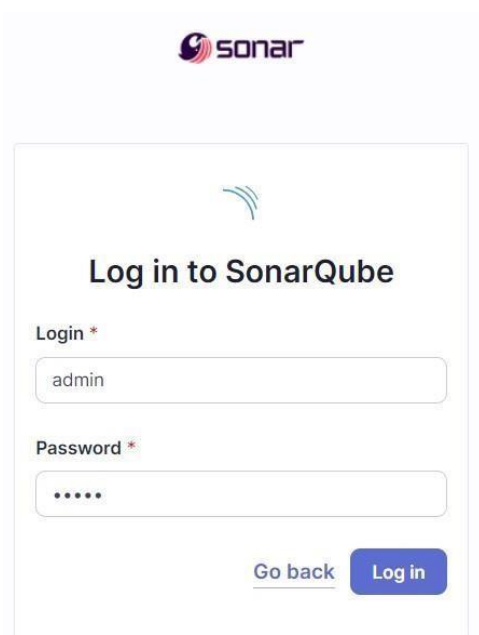


2. Run SonarQube in a Docker container using this command: a] `docker -v` b] `docker pull sonarqube` c] `docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest`

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

C:\Users\adity>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 interface. At the top is the Sonar logo. Below it is a large heading "Log in to SonarQube". Underneath the heading are two input fields: "Login *" with the text "admin" and "Password *" with masked characters. At the bottom right of the form are two buttons: "Go back" (a link) and "Log in" (a blue button).

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

1 of 2

Create a local project

Project display name *

sonarqube-test ✓

Project key *

sonarqube-test ✓

Main branch name *

main

The name of your project's default branch [Learn More](#)

Cancel Next

2 of 2

Set up project for Clean as You Code

This new code definition sets which part of your code will be considered new code. This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology. [Learn more: Defining New Code](#)

Choose the baseline for new code for this project

☒ Use the global setting

Previous version

Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.

☐ Define a specific setting for this project

☐ Previous version

Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.

☐ Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code. Recommended for projects following continuous delivery.

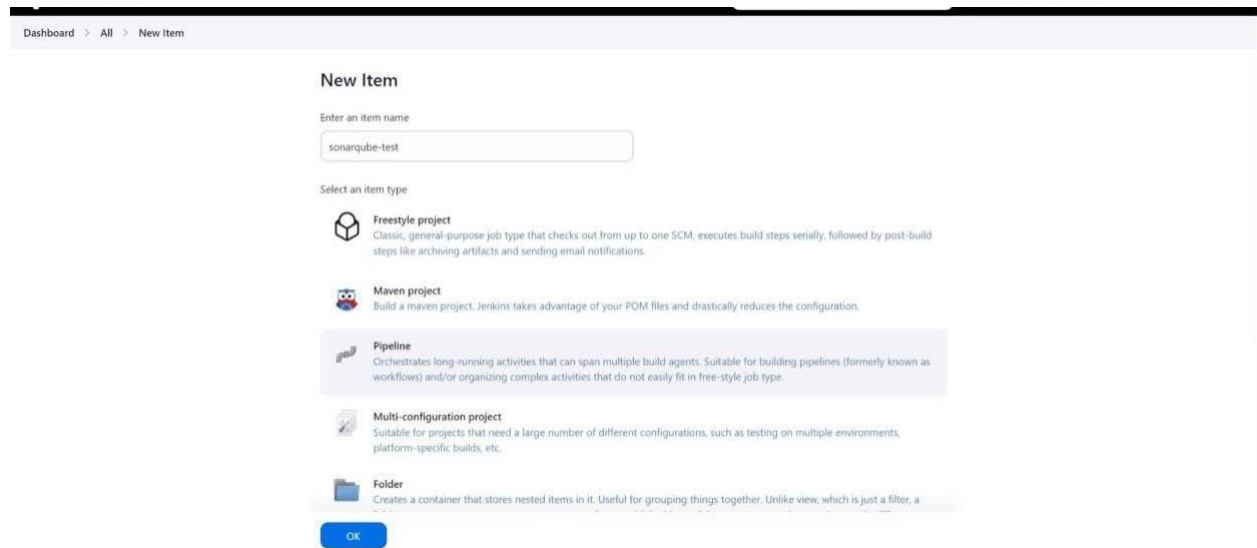
☐ Reference branch

Choose a branch as the baseline for the new code. Recommended for projects using feature branches.

Back Create project

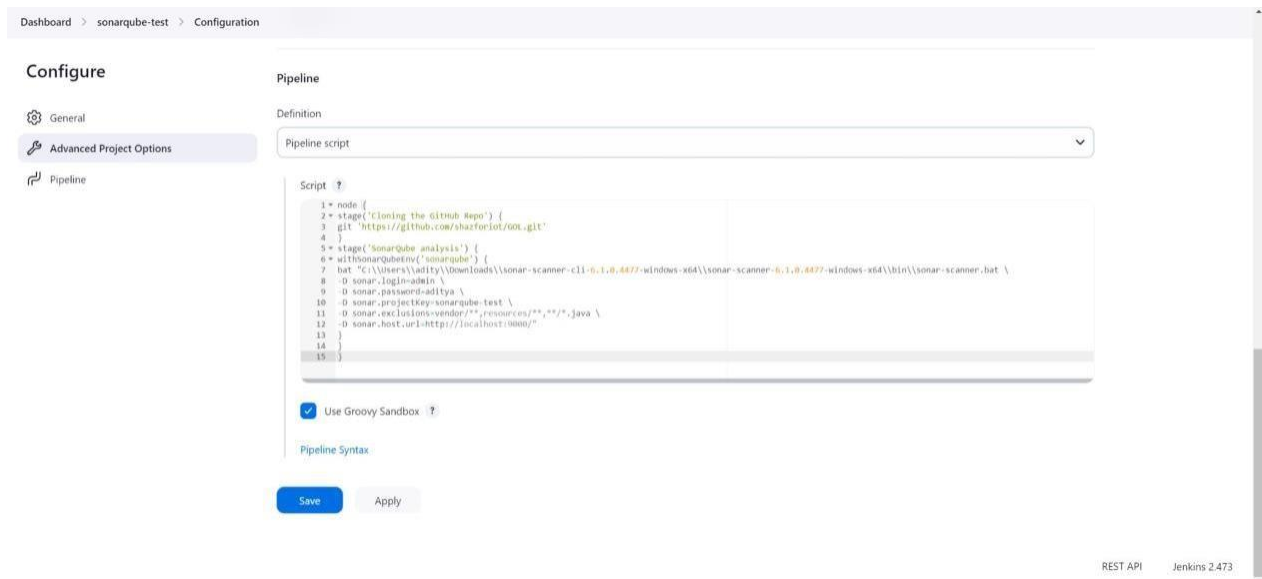
Setup the project and come back to Jenkins Dashboard.

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



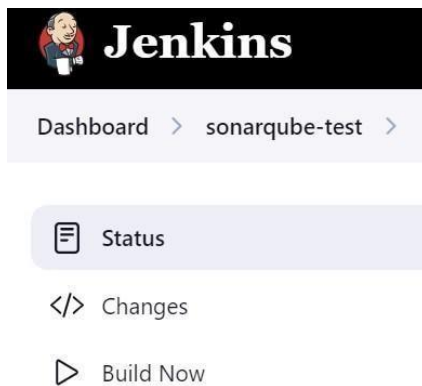
7. Under **Pipeline Script**, enter the following -

```
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=153336ms
21:37:59.955 INFO SCH revision ID 'ba799ba7e1b576f04a612322b0412c5e6e1e5e4'
21:40:14.276 INFO Analysis report generated in 5151ms, dir size=127.2 MB
21:40:35.678 INFO Analysis report compressed in 211388ms, zip size=29.6 MB
21:40:36.170 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/ce/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.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

My Favorites All

Filters

Quality Gate

Passed 2

Failed 0

Reliability

A 1

B 0

C 1

D 0

E 0

Security

A 2

Search for projects... Perspective Overall Status Sort by Name 2 project(s)

☆ sonarqube PUBLIC Passed

Last analysis: 1 hour ago

The main branch of this project is empty.

☆ sonarqube-test PUBLIC Passed

Last analysis: 16 minutes ago - 683k Lines of Code - HTML, XML, ...

0 Security 68k Reliability 164k Maintainability 0.0% Hotspots Reviewed 50.6% Coverage Duplications

2 of 2 shown

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More Q

☆ sonarqube-test / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main 683k Lines of Code - Version not provided - Set as homepage

Quality Gate Passed

The last analysis has warnings. See details

New Code Overall Code

Security 0 Open Issues

Reliability 68k Open Issues

Maintainability 164k Open Issues

Accepted issues 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 for a project named 'sonarqube-test'. The 'Issues' tab is selected, displaying a list of open issues. The left sidebar shows the 'Issues' section with 'Open Issues' highlighted, showing 210,549 issues. The main content area shows a tree view of the project structure with the following files and their issue counts:

File	Issues
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 for a project named 'sonarqube-test'. The 'Issues' tab is selected, displaying a list of consistency issues. The left sidebar shows the 'Issues' section with 'Consistency' highlighted, showing 197k issues. The main content area shows a list of consistency issues with the following details:

Issue	Category	Severity	Effort	Age	Tags
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 shows the SonarQube web interface for a project named 'sonarqube-test'. The 'Issues' tab is selected, and the 'Intentionality' quality attribute is chosen from the left sidebar. The sidebar also shows 'Clean Code Attribute' with a count of 1 and 'Software Quality' with a count of 14k. The main panel displays a list of issues under the 'gameoflife-acceptance-tests/Dockerfile' file. The issues are categorized by 'Intentionality' and 'Maintainability'. The first issue is 'Use a specific version tag for the image.' with a severity of 'Major' and a count of 13,887. The second issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with a severity of 'Major' and a count of 59d effort. The third issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with a severity of 'Major' and a count of 59d effort. A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only'.

Code Smells

The screenshot shows the SonarQube web interface for a project named 'sonarqube-test'. The 'Issues' tab is selected, and the 'Code Smell' quality attribute is chosen from the left sidebar. The sidebar also shows 'Severity' with counts for High (0), Medium (0), and Low (253), and 'Type' with counts for Bug (14k), Vulnerability (0), and Code Smell (253). The main panel displays a list of issues under the 'gameoflife-web/tools/meter/printable_docs/building.html' file. The issues are categorized by 'Intentionality' and 'Reliability'. The first issue is 'Add an "alt" attribute to this image.' with a severity of 'Minor' and a count of 253. The second issue is 'Add an "alt" attribute to this image.' with a severity of 'Minor' and a count of 253. The third issue is 'Add an "alt" attribute to this image.' with a severity of 'Minor' and a count of 253. A warning message at the bottom states: 'Embedded database should be used for evaluation purposes only'.

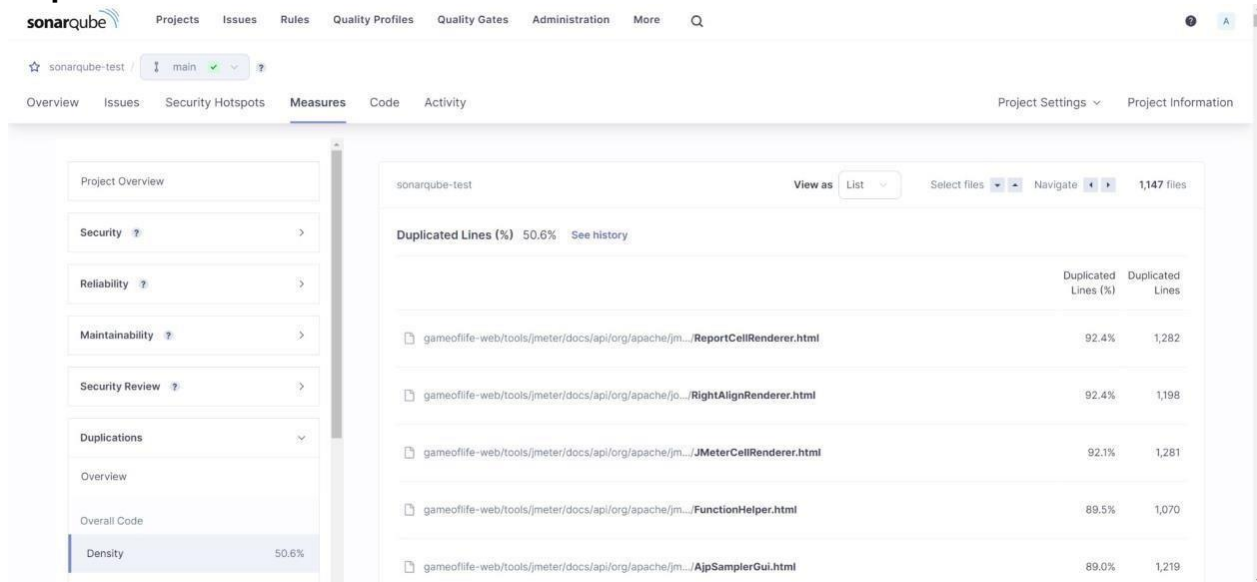
Bugs

The screenshot shows the SonarQube web interface with the 'Issues' tab selected. The left sidebar displays filters for Severity (High, Medium, Low) and Type (Bug, Vulnerability, Code Smell). The 'Bug' type is selected, showing 14k issues. The main panel 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. A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only'.

Reliability

The screenshot shows the SonarQube web 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 'Reliability' attribute is selected, showing 14k issues. The main panel 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. A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only'.

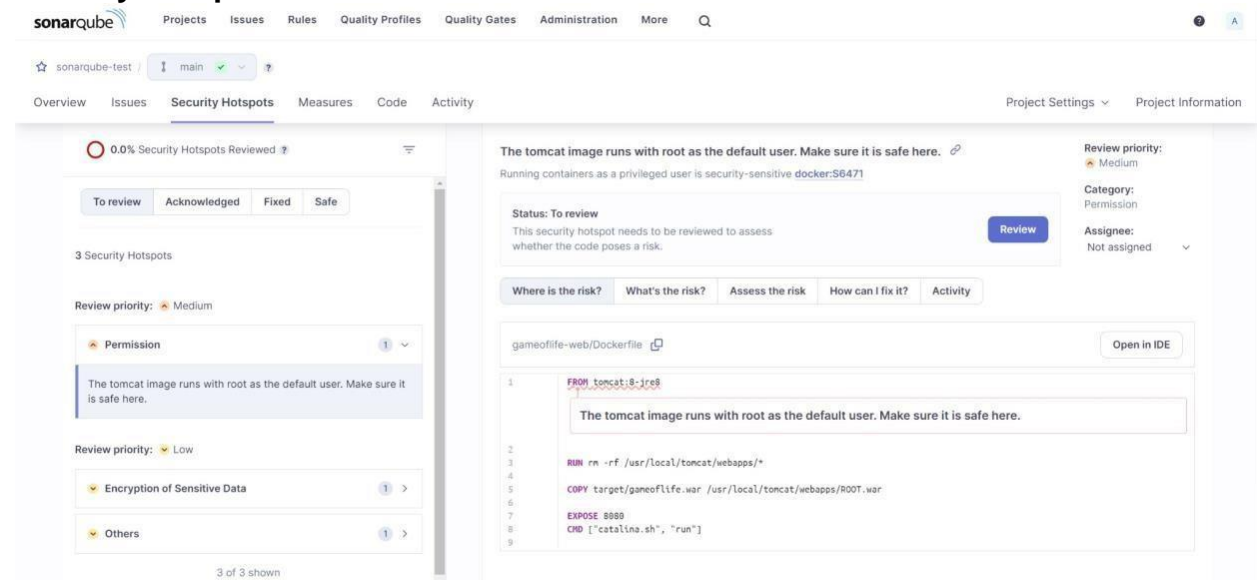
Duplicates



The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Measures' tab is active, displaying a table of duplicated lines. The table has columns for the file path, duplicated lines percentage, and duplicated lines count. The overall duplicated lines percentage for the project is 50.6%.

File Path	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/jm.../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



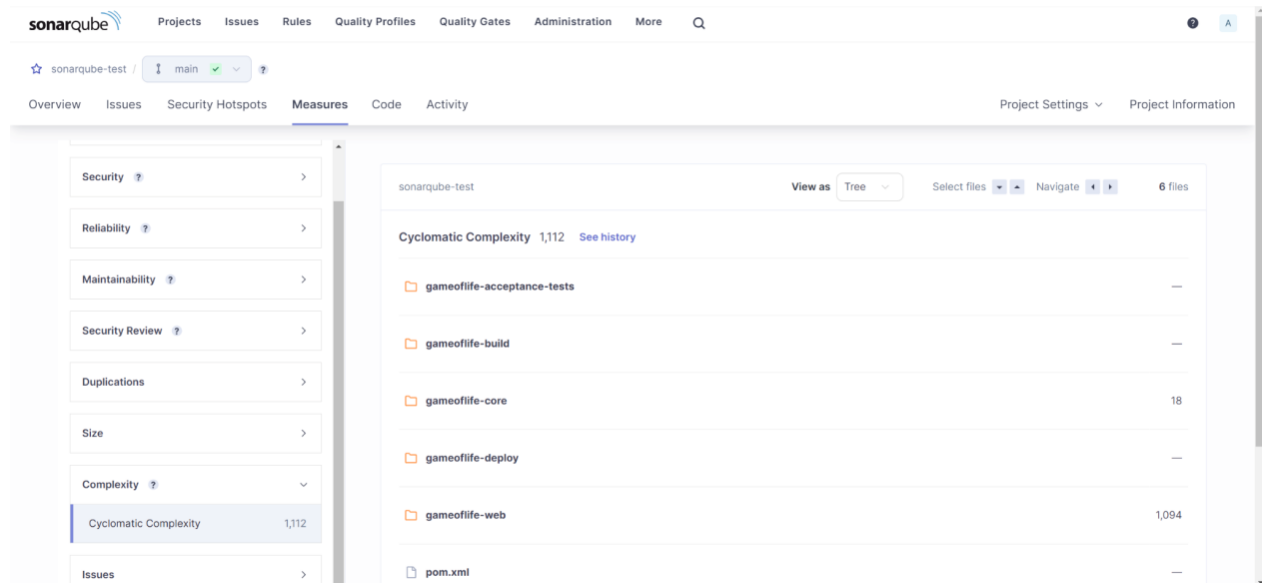
The screenshot shows the SonarQube interface for the 'sonarqube-test' project, specifically the 'Security Hotspots' tab. A security hotspot is displayed with a 'To review' status. The hotspot description is: 'The tomcat image runs with root as the default user. Make sure it is safe here.' The review priority is 'Medium'. The category is 'Permission'. The assignee is 'Not assigned'. The hotspot is located in the file 'gameoflife-web/Dockerfile'. The code snippet shows the 'FROM' instruction for the tomcat image.

Security Hotspot Details:

- Status:** To review
- Description:** The tomcat image runs with root as the default user. Make sure it is safe here.
- Review priority:** Medium
- Category:** Permission
- Assignee:** Not assigned
- Where is the risk?** gameoflife-web/Dockerfile
- What's the risk?** The tomcat image runs with root as the default user. Make sure it is safe here.
- Assess the risk**
- How can I fix it?**
- Activity**

```
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.