

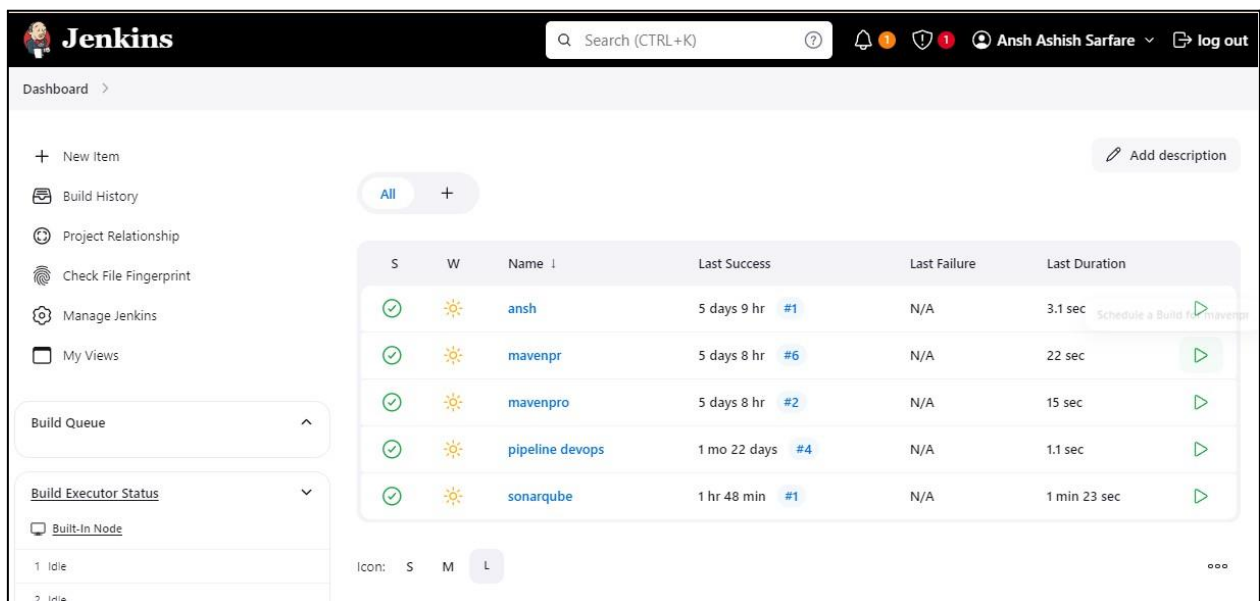
ADVANCE DEVOPS EXP-8

Sanket More

D15A 30

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: Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is atfor you.



Step-2: Run SonarQube in a Docker container using this command :- a]docker -vb]
docker run -d --name sonarqube-test -e
SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9001:9000 sonarqube:latest

```
Windows PowerShell
PS C:\Users\Ansh> docker -v
Docker version 27.0.3, build 7d4bcd8
PS C:\Users\Ansh> docker run -d --name sonarqube-test -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9001:9000 sonarqube:latest
11f77905edf285ee21894f76f48408535f257f304883595be20f493e4c18039f
PS C:\Users\Ansh>
```

Step-3: Once the container is up and running, you can check the status of SonarQube at localhost port 9001. The login id is “admin” and the password is also “ansh16”.

Log in to SonarQube

Login *

Password *


[Go back](#) [Log in](#)

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


1 of 2

Create a local project


Project display name *

Project key *

Main branch name *

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

[Cancel](#) [Next](#)

Step-5: Setup the project and come back to Jenkins Dashboard.

2 of 2

Set up project for Clean as You Code

The 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

Step-6: Create a New Item in Jenkins, choose Pipeline.

 **Jenkins**

Search (CTRL+K)

Ansh Ashish Sarfare

log o


Dashboard > All > New Item

New Item


Enter an item name

sonarqube-test


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.




Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.




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.



Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.



Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

Step-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\Ansh\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0. 4477-
windows-x64\bin\sonar-scanner.bat \
-D sonar.login=admin \
-D sonar.password=ansh16 \
-D sonar.projectKey=sonarqube-test \
-D sonar.exclusions=vendor/**,resources/**,*/*.java \
-D sonar.host.url=http://localhost:9001/"
        }
    }
}
```

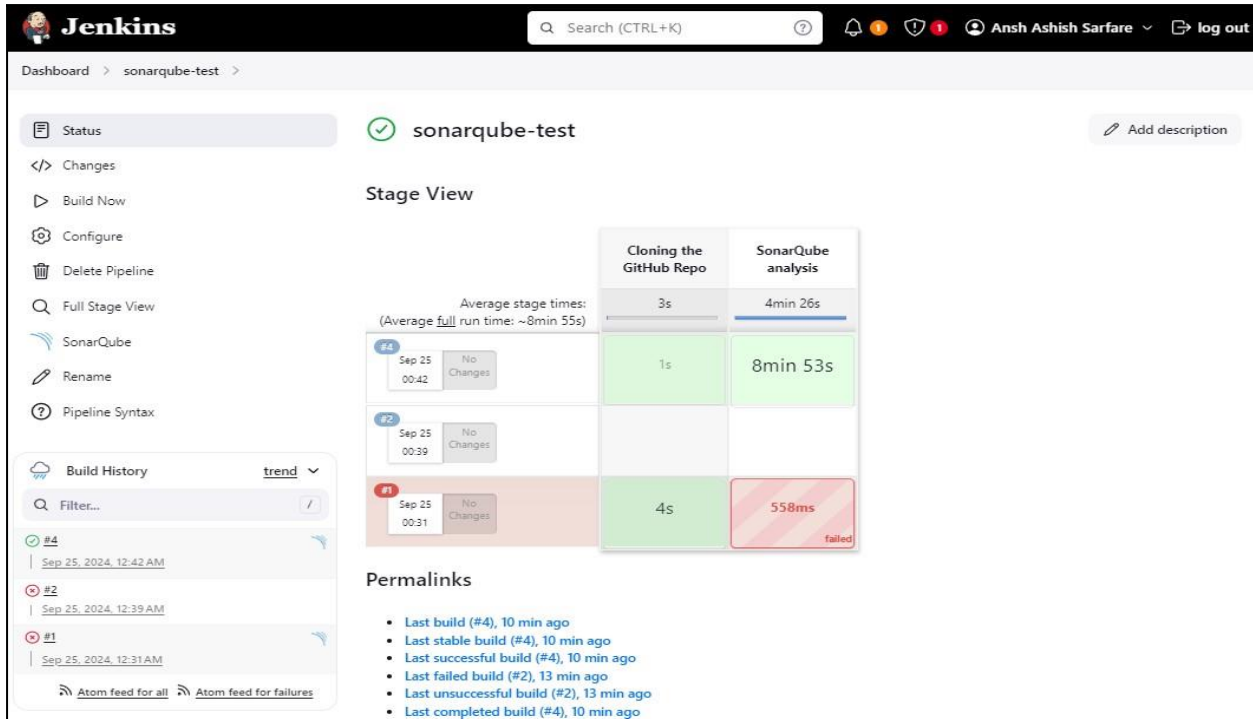
The screenshot shows the Jenkins Pipeline configuration page. The 'Definition' dropdown is set to 'Pipeline script'. The 'Script' section contains a Groovy script for cloning a GitHub repository and running a SonarQube analysis. The script is as follows:

```
1 node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shazforiot/GOL.git'
4   }
5   stage('SonarQube analysis') {
6     withSonarQubeEnv('sonarqube') {
7       bat """
8         C:/Users/Ansh/Downloads/sonar-scanner-cli-6.1.0.4477-windows-x64/sonar-scanner-6.1.0.4477-windows-x64/bin/sonar-scanner.bat ^
9         -D sonar.login=admin ^
10        -D sonar.password=ansh16 ^
11        -D sonar.projectKey=sonarqube-test ^
12        -D sonar.exclusions=vendor/**,resources/**,*/*.java ^
13        -D sonar.host.url=http://127.0.0.1:9001/
14      """
15    }
16  }
17 }
```

Below the script, the 'Use Groovy Sandbox' checkbox is checked. At the bottom, there are 'Save' and 'Apply' buttons.

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

Step-8: Run The Build and check the console output:

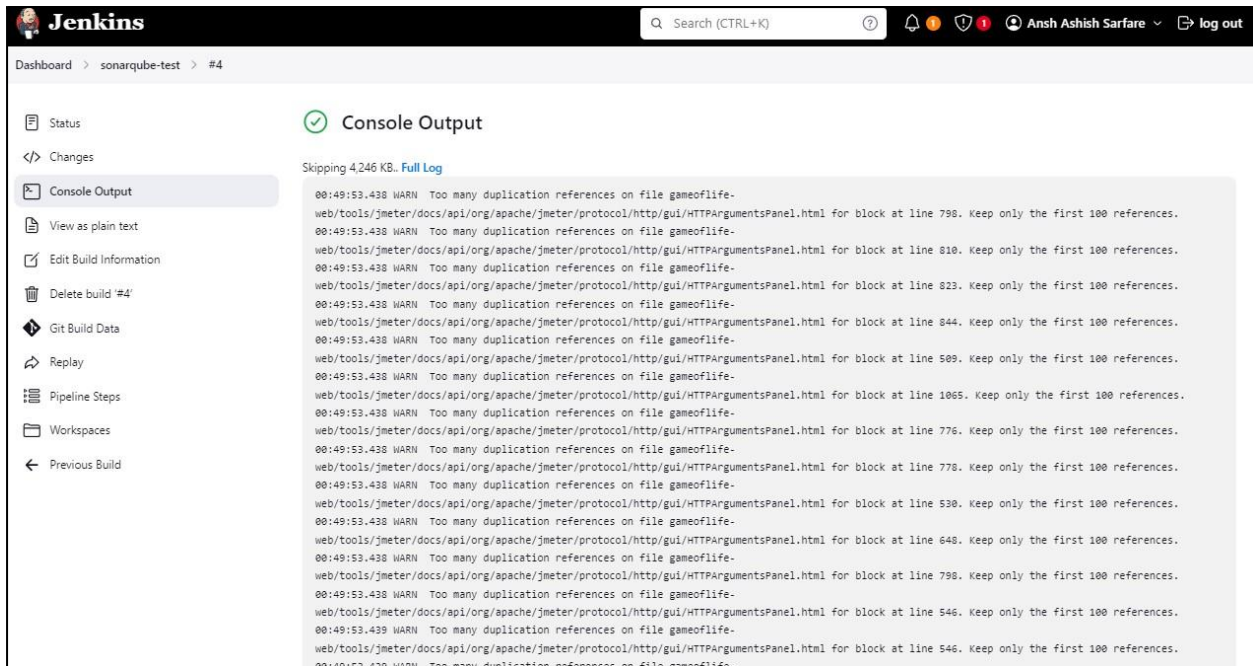


The Jenkins Pipeline View for the 'sonarqube-test' pipeline shows the following details:

- Status:** ✔ sonarqube-test
- Stage View:**

Stage	Cloning the GitHub Repo	SonarQube analysis
#4 (Sep 25, 00:42)	1s	8min 53s
#2 (Sep 25, 00:39)		
#1 (Sep 25, 00:31)	4s	558ms (failed)

Average stage times: 3s (Cloning), 4min 26s (SonarQube analysis). Average full run time: ~8min 55s.
- Permalinks:**
 - Last build (#4), 10 min ago
 - Last stable build (#4), 10 min ago
 - Last successful build (#4), 10 min ago
 - Last failed build (#2), 13 min ago
 - Last unsuccessful build (#2), 13 min ago
 - Last completed build (#4), 10 min ago



The Jenkins Console Output for build #4 shows the following details:

- Status:** ✔ Console Output
- Console Output:**

```
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 798. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 810. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 823. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 844. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 509. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 1065. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 776. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 778. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 530. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 648. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 790. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.  
00:49:53.439 WARN Too many duplication references on file gameoflife-  
web/tools/jmeter/docs/api/org/apache/jmeter/protocol/http/gui/HTTPArgumentsPanel.html for block at line 546. Keep only the first 100 references.  
00:49:53.438 WARN Too many duplication references on file gameoflife-
```

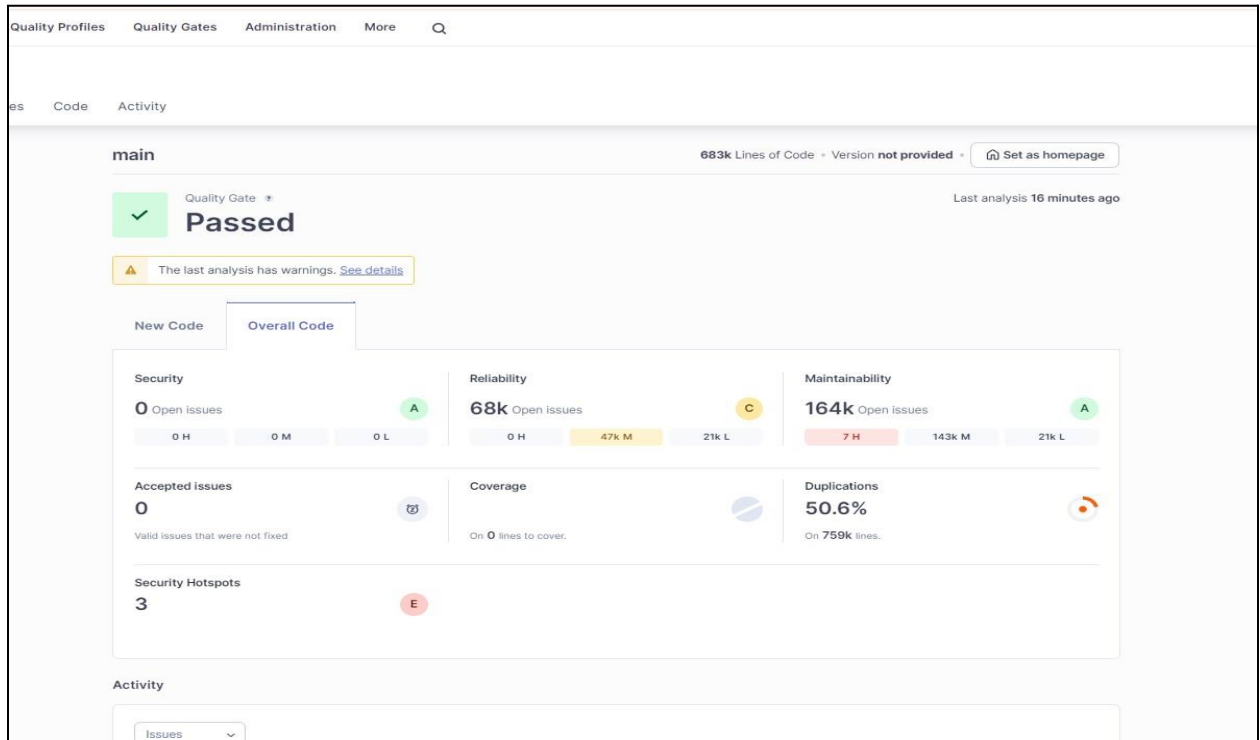
```

00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 32. Keep only the first 100 referen
00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 177. Keep only the first 100 referen
00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 180. Keep only the first 100 referen
00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 65. Keep only the first 100 referenc
00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 349. Keep only the first 100 referen
00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 40. Keep only the first 100 referenc
00:49:56.323 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 referenc
00:49:56.323 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRenderer.html for block at line 41. Keep only the first 100 referenc
00:49:56.324 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 referenc
00:49:56.324 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 referen
00:49:56.324 INFO CPO Executor CPO calculation finished (done) | time=94621ms
00:49:56.350 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
00:51:30.402 INFO Analysis report generated in 2893ms, dir size=127.2 MB
00:51:40.652 INFO Analysis report compressed in 10210ms, zip size=29.6 MB
00:51:44.098 INFO Analysis report uploaded in 3444ms
00:51:44.101 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://127.0.0.1:9001/dashboard?id=sonarqube-test
00:51:44.101 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
00:51:44.101 INFO More about the report processing at http://127.0.0.1:9001/api/ce/task?id=22b0b5c1-635d-4c1b-8d62-99d4ce4567b9
00:51:53.341 INFO Analysis total time: 8:44.093 s
00:51:53.349 INFO SonarScanner Engine completed successfully
00:51:54.059 INFO EXECUTION SUCCESS
00:51:54.071 INFO Total time: 8:51.363s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS

```

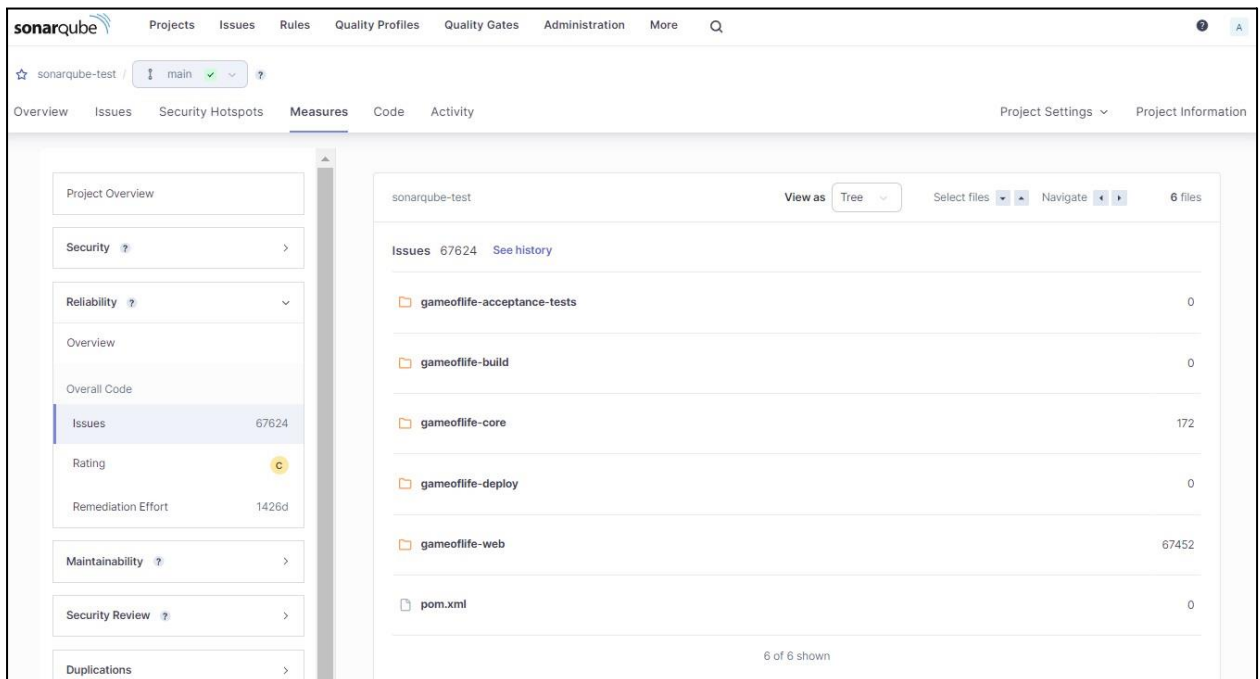
Step-9: After that, check the project in SonarQube.

The screenshot shows the SonarQube web interface. On the left, there's a sidebar with 'My Favorites' and 'All' tabs. Below them are filters for 'Quality Gate', 'Reliability', and 'Security'. The 'Quality Gate' filter shows 'Passed' with a count of 1 and 'Failed' with a count of 0. The 'Reliability' filter shows 'A' with a count of 0, 'B' with a count of 0, 'C' with a count of 1, 'D' with a count of 0, and 'E' with a count of 0. The 'Security' filter is empty. The main area displays the project 'sonarqube-test' in a 'PUBLIC' state, marked as 'Passed'. It shows the last analysis was 15 minutes ago, with 683k lines of code in HTML, XML, etc. Below this, there's a row of metrics: Security (0), Reliability (68k), Maintainability (164k), Hotspots Reviewed (0.0%), Coverage (50.6%), and Duplications (50.6%).



Step-10: Under different tabs, check all different issues with the code.
Code Problems

Code issues:



Consistency:

The screenshot displays the SonarQube web interface for a project named 'sonarqube-test'. The 'Issues' tab is active, showing a list of issues under the 'Consistency' attribute. The left sidebar shows filters for 'Clean Code Attribute' and 'Software Quality'. The main panel lists four issues related to deprecated attributes in an HTML file.

Issue Description	Attribute	Severity	Effort	Age	Smell
Insert a <DOCTYPE> declaration to before this <html> tag.	Consistency	Major	5min	4 years ago	if Bug
Remove this deprecated "width" attribute.	Consistency	Major	5min	4 years ago	Code Smell
Remove this deprecated "align" attribute.	Consistency	Major	5min	4 years ago	Code Smell
Remove this deprecated "align" attribute.	Consistency	Major	5min	4 years ago	Code Smell

Intentionally:

The screenshot displays the SonarQube web interface for the same project, 'sonarqube-test', but with the 'Intentionally' attribute selected. The left sidebar shows filters for 'Clean Code Attribute' and 'Software Quality'. The main panel lists four issues related to image tags and variable quoting in a Dockerfile.

Issue Description	Attribute	Severity	Effort	Age	Smell
Use a specific version tag for the image.	Intentionally	Major	5min	4 years ago	Code Smell
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionally	Major	5min	4 years ago	Code Smell
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionally	Major	5min	4 years ago	Code Smell
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionally	Major	5min	4 years ago	Code Smell

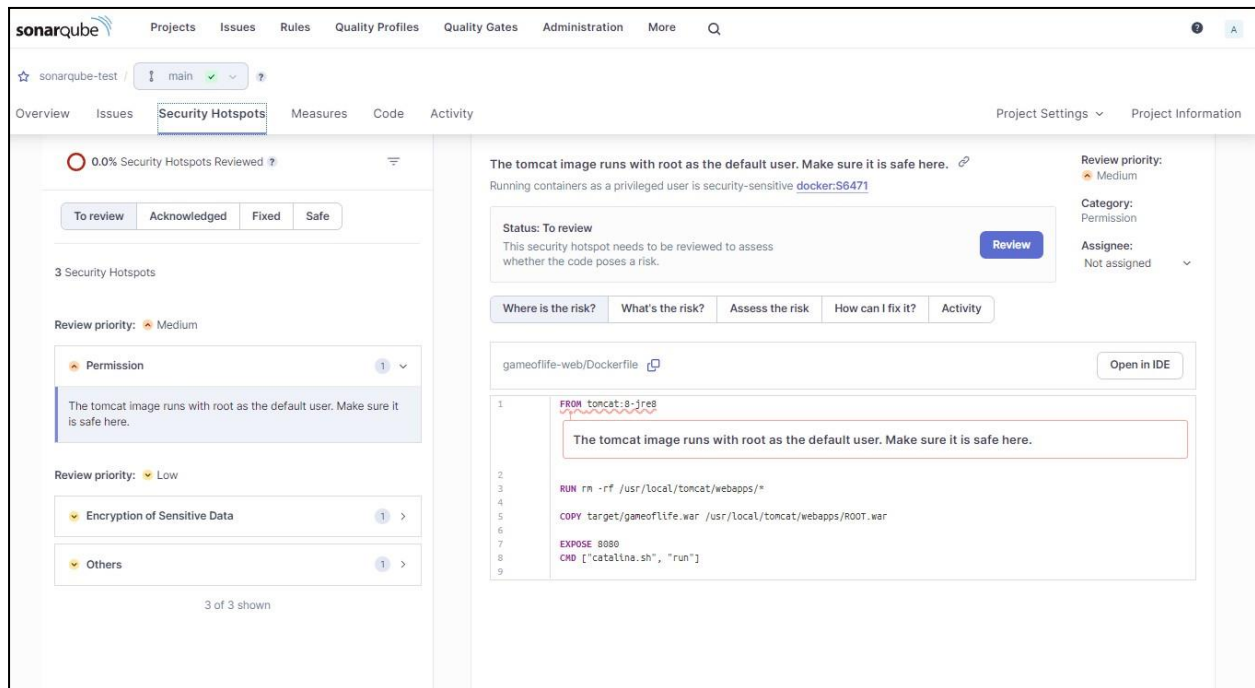
Reliability:

The screenshot displays the SonarQube web interface for a project named 'sonarqube-test'. The 'Issues' tab is active, showing a list of issues categorized by 'Clean Code Attribute' and 'Software Quality'. The 'Clean Code Attribute' section shows 'Consistency' with 54k issues and 'Intentionality' with 14k issues. The 'Software Quality' section shows 'Security' with 0 issues, 'Reliability' with 14k issues, and 'Maintainability' with 15 issues. The 'Intentionality' category is selected, showing a list of issues. The first issue is 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' with a 'Reliability' severity and 'Intentionality' category. The second issue is 'Add "<th>" headers to this "<table>:"' with a 'Reliability' severity and 'Intentionality' category. The third issue is 'Add "lang" and/or "xml:lang" attributes to this "<html>" element' with a 'Reliability' severity and 'Intentionality' category. The fourth issue is 'Add "<th>" headers to this "<table>:"' with a 'Reliability' severity and 'Intentionality' category. The interface also shows a 'Filters' sidebar on the left and a 'Bulk Change' button at the top.

Code smells:

The screenshot displays the SonarQube web interface for a project named 'sonarqube-test'. The 'Issues' tab is active, showing a list of issues categorized by 'Clean Code Attribute' and 'Software Quality'. The 'Clean Code Attribute' section shows 'Consistency' with 54k issues and 'Intentionality' with 14k issues. The 'Software Quality' section shows 'Security' with 0 issues, 'Reliability' with 14k issues, and 'Maintainability' with 15 issues. The 'Maintainability' category is selected, showing a list of issues. The first issue is 'Use a specific version tag for the image.' with a 'Maintainability' severity and 'Intentionality' category. The second issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with a 'Maintainability' severity and 'Intentionality' category. The third issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with a 'Maintainability' severity and 'Intentionality' category. The fourth issue is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' with a 'Maintainability' severity and 'Intentionality' category. The interface also shows a 'Filters' sidebar on the left and a 'Bulk Change' button at the top.

Security hotspot:



The screenshot shows the SonarQube interface for a project named 'sonarqube-test'. The 'Security Hotspots' tab is active. On the left, a summary shows '0.0% Security Hotspots Reviewed' and a list of hotspots with review priorities: Medium (Permission), Low (Encryption of Sensitive Data), and Others. The main area displays a specific hotspot titled 'The tomcat image runs with root as the default user. Make sure it is safe here.' with a status of 'To review'. Below the title, a code snippet from 'gameoflife-web/Dockerfile' is shown, highlighting the 'FROM tomcat:8-jre8' line. A red box highlights the warning text. The right sidebar shows the review priority as Medium, category as Permission, and assignee as Not assigned.

0.0% Security Hotspots Reviewed

To review Acknowledged Fixed Safe

3 Security Hotspots

Review priority: Medium

Permission

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

Review priority: Low

Encryption of Sensitive Data

Others

3 of 3 shown

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

Status: To review

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

Review

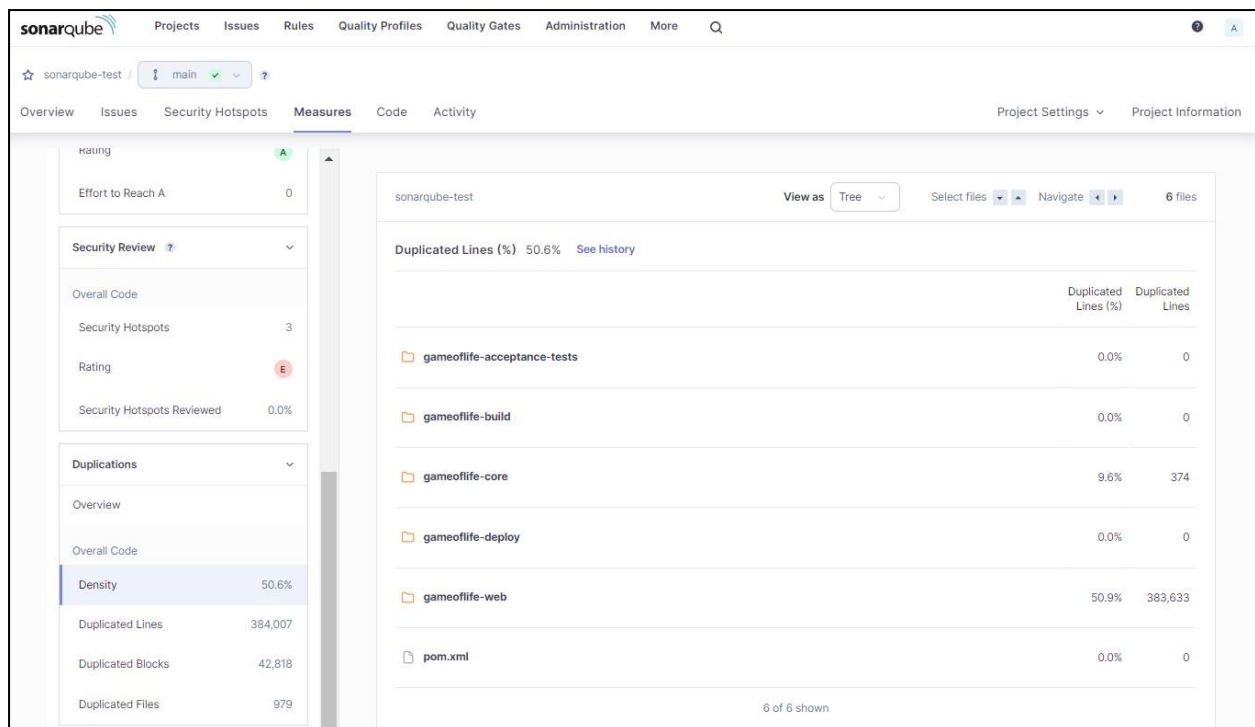
Where is the risk? What's the risk? Assess the risk How can I fix it? Activity

gameoflife-web/Dockerfile

Open in IDE

```
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
```

Duplicates:



The screenshot shows the SonarQube interface for a project named 'sonarqube-test'. The 'Measures' tab is active. On the left, a summary shows 'Rating: A' and 'Effort to Reach A: 0'. The main area displays a table of duplicated lines across various files. The table has columns for 'Duplicated Lines (%)' and 'Duplicated Lines'. The files listed are 'gameoflife-acceptance-tests', 'gameoflife-build', 'gameoflife-core', 'gameoflife-deploy', 'gameoflife-web', and 'pom.xml'. The 'gameoflife-web' file has the highest percentage of duplicated lines at 50.9%.

Rating: A

Effort to Reach A: 0

Security Review

Overall Code

Security Hotspots: 3

Rating: E

Security Hotspots Reviewed: 0.0%

Duplications

Overview

Overall Code

Density: 50.6%

Duplicated Lines: 384,007

Duplicated Blocks: 42,818

Duplicated Files: 979

sonarqube-test

View as: Tree

Select files: 6 files

Duplicated Lines (%) 50.6% See history

	Duplicated Lines (%)	Duplicated Lines
gameoflife-acceptance-tests	0.0%	0
gameoflife-build	0.0%	0
gameoflife-core	9.6%	374
gameoflife-deploy	0.0%	0
gameoflife-web	50.9%	383,633
pom.xml	0.0%	0

6 of 6 shown

Size:

The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Measures' tab is active, displaying various size-related metrics. On the left, a sidebar shows 'Lines of Code' as 682,883. The main content area shows a breakdown of lines of code by language: HTML (678k), XML (4.7k), JSP (332), CSS (110), and Docker (19). Below this, a table lists files and their sizes: gameoflife-acceptance-tests (164), gameoflife-build (368), gameoflife-core (3,675), gameoflife-deploy (69), gameoflife-web (678,148), and pom.xml (459).

Language	Lines of Code
HTML	678k
XML	4.7k
JSP	332
CSS	110
Docker	19

File	Size
gameoflife-acceptance-tests	164
gameoflife-build	368
gameoflife-core	3,675
gameoflife-deploy	69
gameoflife-web	678,148
pom.xml	459

Complexity:

The screenshot shows the SonarQube interface for the 'sonarqube-test' project, focusing on complexity metrics. The 'Measures' tab is active, displaying 'Cyclomatic Complexity' as 1,112. The main content area shows a breakdown of cyclomatic complexity by file: gameoflife-acceptance-tests (—), gameoflife-build (—), gameoflife-core (18), gameoflife-deploy (—), gameoflife-web (1,094), and pom.xml (—).

File	Cyclomatic Complexity
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—