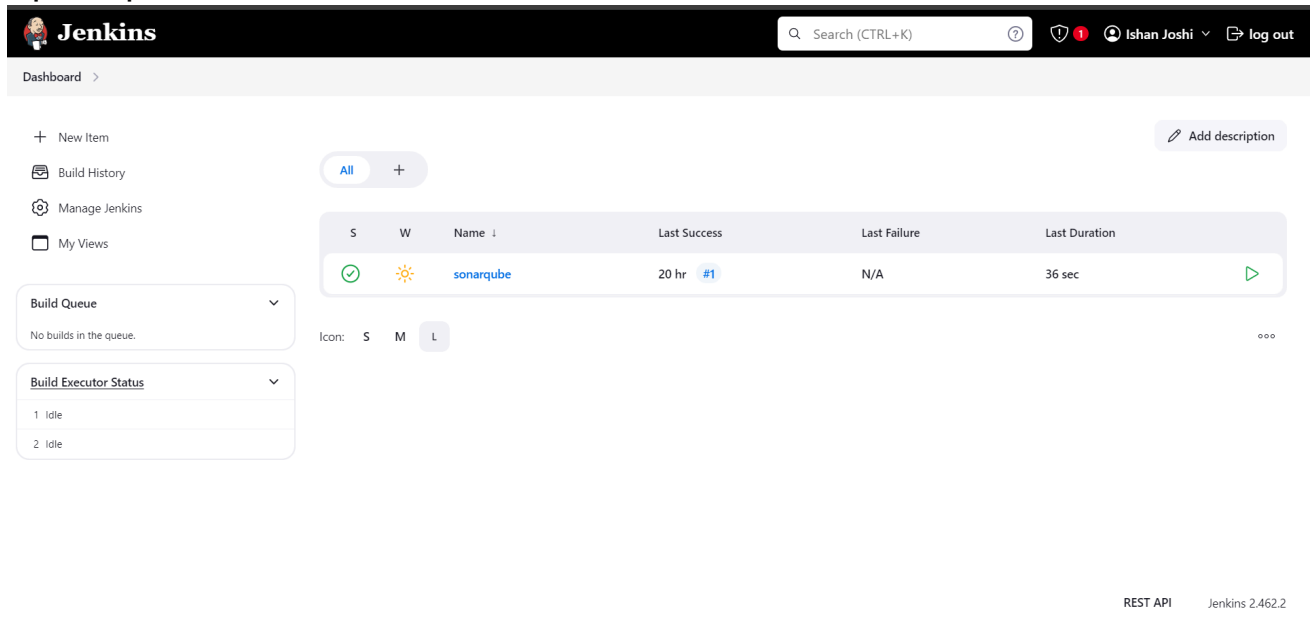


Name-Ishan Kiran Joshi Div-D15C Roll No-21 A.Y.-2024-25

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.



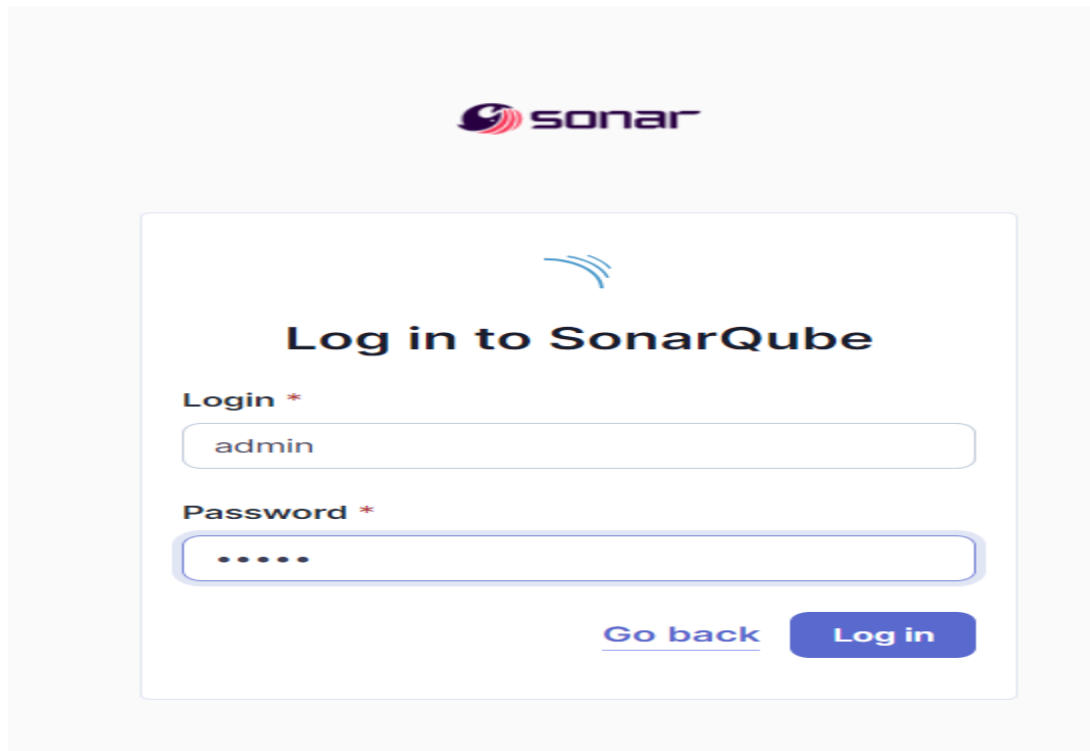
2. Run SonarQube in a Docker container using this command:

- 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\ishan>docker -v
Docker version 27.1.1, build 6312585

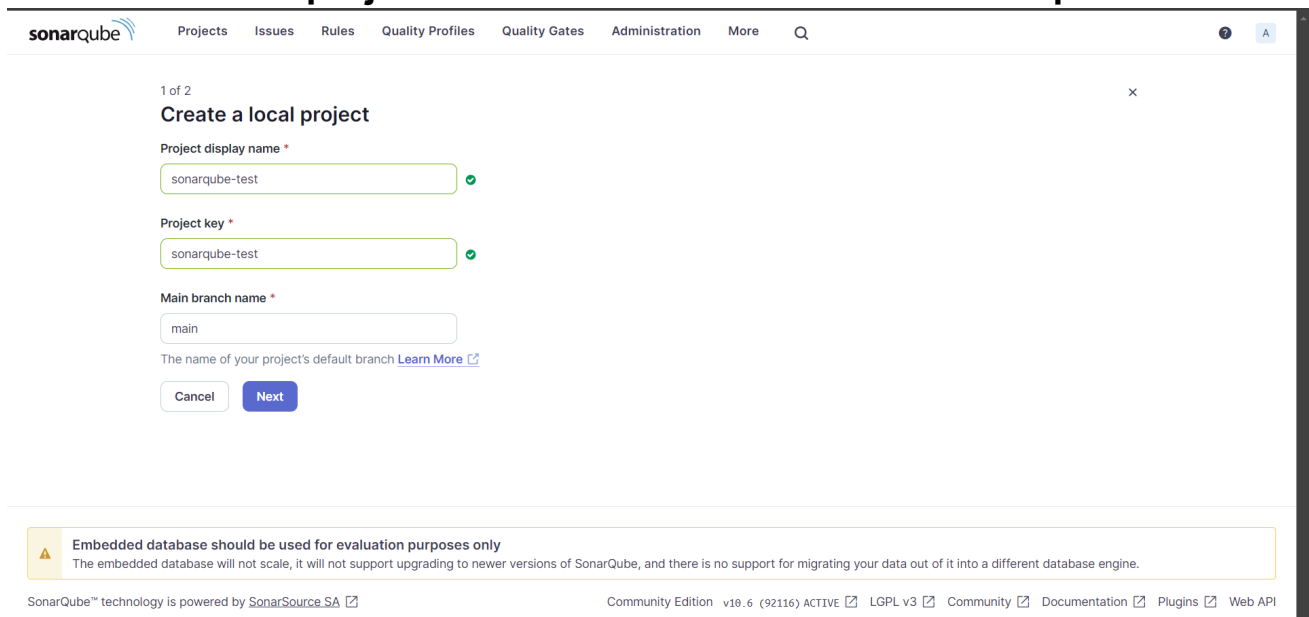
C:\Users\ishan>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d6fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
23c14503da77ee785f6069bfa7f714939ddb794a6b846124594503f6183b4c68
```

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 “**ishan**”.



The image shows the SonarQube login interface. At the top is the Sonar logo. Below it is a large white box with a blue Sonar icon and the text "Log in to SonarQube". Inside this box are two input fields: "Login *" with the value "admin" and "Password *" with masked characters ".....". At the bottom of the box 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**.



The image shows the "Create a local project" form in SonarQube. The form is titled "1 of 2 Create a local project" and has a close button (X). It contains three input fields: "Project display name *" with the value "sonarqube-test" and a green checkmark, "Project key *" with the value "sonarqube-test" and a green checkmark, and "Main branch name *" with the value "main". Below the last field is a link "Learn More" and a note "The name of your project's default branch". At the bottom are "Cancel" and "Next" buttons. A yellow warning box at the bottom states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine." The footer contains the text "SonarQube™ technology is powered by SonarSource SA" and a list of links: "Community Edition v10.6 (92116) ACTIVE", "LGPL v3", "Community", "Documentation", "Plugins", and "Web API".

2 of 2

x

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

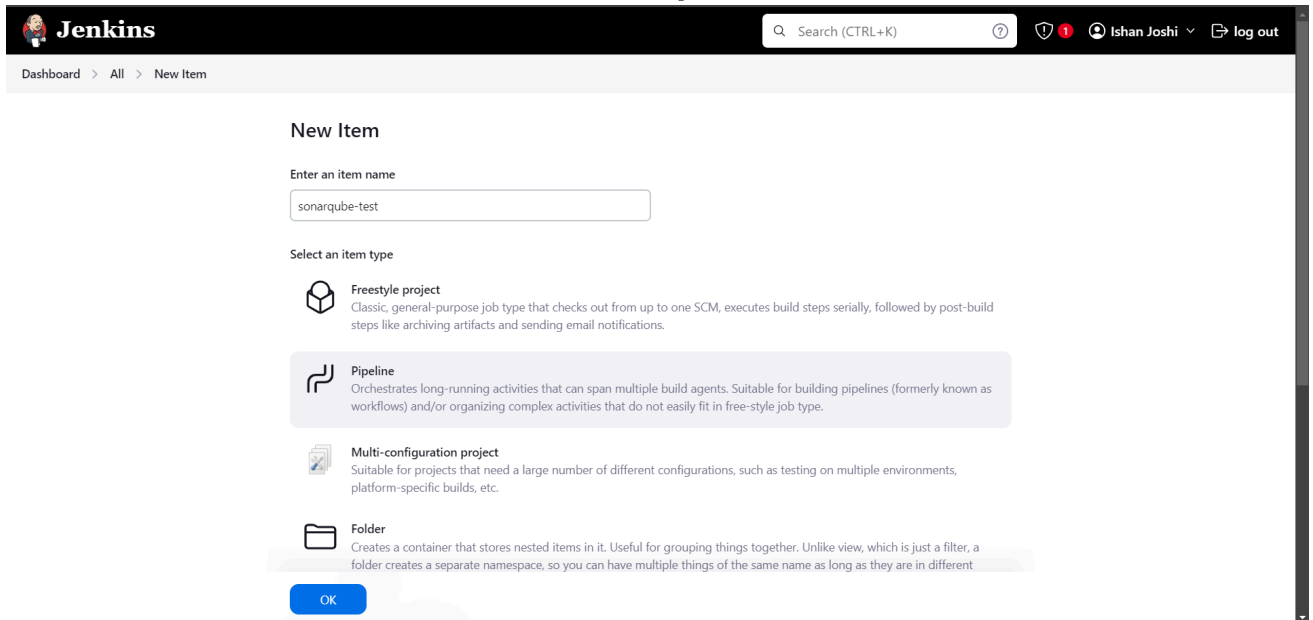
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

☐ 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.

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('sonarqu  
be') {  
            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/"  
        }  
    }  
}
```

Dashboard > sonarqube-test > Configuration

Configure

General
Advanced Project Options
Pipeline

Pipeline

Definition

Pipeline script

```
1 node {
2   stage('Cloning the GitHub Repo') {
3     git 'https://github.com/shazforiot/GOL.git'
4   }
5
6   stage('SonarQube analysis') {
7     withSonarQubeEnv('sonarqube') {
8       bat -x-
9       C:\Users\Ishan\Downloads\sonar-scanner-cli-6.1.0.4477-windows-x64\sonar-scanner-6.1.0.4477-windows-x64\bin\sonar-
10      -Dsonar.login=admin ^
11      -Dsonar.password=ishan ^
12      -Dsonar.projectkey=sonarqube-test ^
13      -Dsonar.exclusions=vendor/**,resources/**,*/*.java ^
14      -Dsonar.host.url=http://localhost:9000/
15      ---
16    }
17  }
```

☒ Use Groovy Sandbox ?

Pipeline Syntax

Save Apply

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

8. Run The Build.

Jenkins

Search (CTRL+K) ?

Ishan Joshi log out

Dashboard > sonarqube-test >

sonarqube-test

Add description

Permalinks

Status
Changes
Build Now
Configure
Delete Pipeline
Stages
Rename
Pipeline Syntax

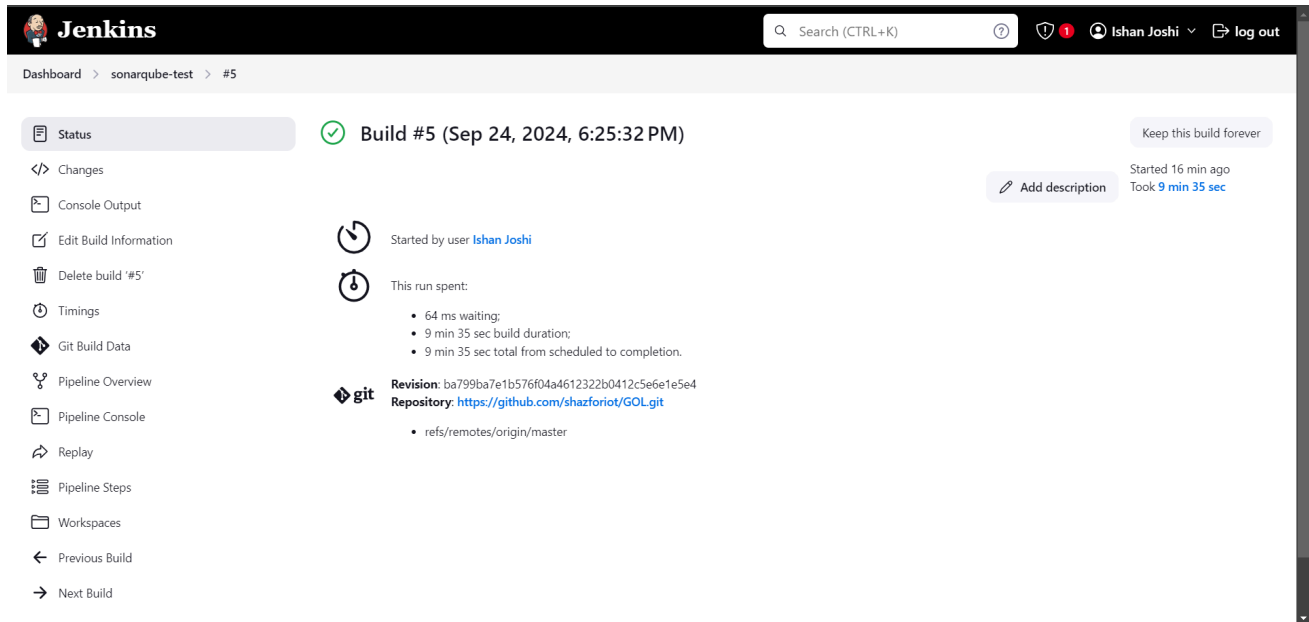
Build History trend

No builds

Atom feed for all Atom feed for failures

REST API Jenkins 2.462.2

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



The Jenkins dashboard shows the status of Build #5 for the 'sonarqube-test' job. The build is successful, indicated by a green checkmark. The build was started by user 'Ishan Joshi' on September 24, 2024, at 6:25:32 PM. The console output shows the build process, including cloning the repository and checking out the revision. The build took 9 minutes and 35 seconds to complete.

Build #5 (Sep 24, 2024, 6:25:32 PM)

Keep this build forever

Started 16 min ago
Took 9 min 35 sec

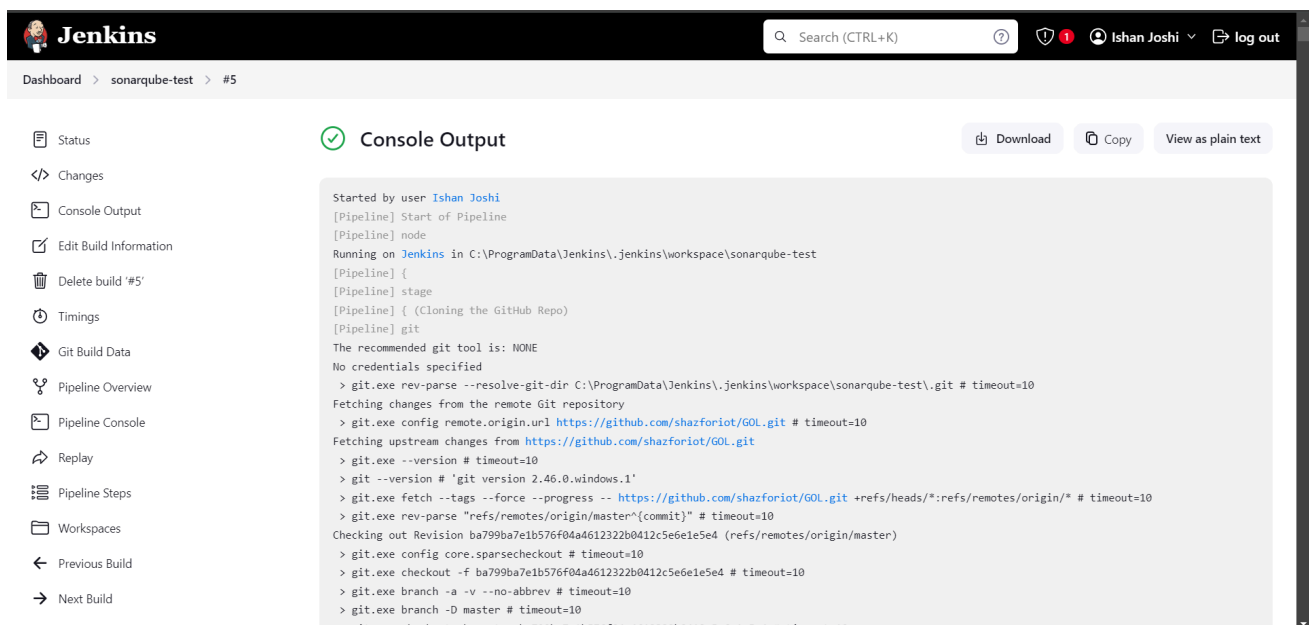
Started by user **Ishan Joshi**

This run spent:

- 64 ms waiting;
- 9 min 35 sec build duration;
- 9 min 35 sec total from scheduled to completion.

Revision: ba799ba7e1b576f04a4612322b0412c5e6e1e5e4
Repository: <https://github.com/shazforiot/GOL.git>

- refs/remotes/origin/master



The Jenkins dashboard shows the console output for Build #5. The output displays the build process, including cloning the repository and checking out the revision. The build took 9 minutes and 35 seconds to complete.

Console Output

Download Copy View as plain text

```
Started by user Ishan Joshi
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube-test
[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-test\.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 2.46.0.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
> git.exe checkout -b master ba799ba7e1b576f04a4612322b0412c5e6e1e5e4 # timeout=10
```

```
18:34:27.439 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRendererer.html
for block at line 75. Keep only the first 100 references.
18:34:27.439 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRendererer.html
for block at line 41. Keep only the first 100 references.
18:34:27.439 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRendererer.html
for block at line 17. Keep only the first 100 references.
18:34:27.439 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRendererer.html
for block at line 296. Keep only the first 100 references.
18:34:27.439 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/util/TextAreaCellRendererer.html
for block at line 75. Keep only the first 100 references.
18:34:27.440 INFO CPD Executor CPD calculation finished (done) | time=151634ms
18:34:27.458 INFO SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
18:34:33.719 INFO Analysis report generated in 4472ms, dir size=127.2 MB
18:34:51.097 INFO Analysis report compressed in 17377ms, zip size=29.6 MB
18:34:51.752 INFO Analysis report uploaded in 655ms
18:34:51.754 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
18:34:51.754 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:34:51.754 INFO More about the report processing at http://localhost:9000/api/ce/task?id=a383e6c0-5738-44c0-9305-9a27b39ce19a
18:35:06.143 INFO Analysis total time: 9:26.800 s
18:35:06.147 INFO SonarScanner Engine completed successfully
18:35:06.832 INFO EXECUTION SUCCESS
18:35:06.862 INFO Total time: 9:29.562s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

10. After that, check the project in SonarQube

The image shows two screenshots of the SonarQube web interface. The top screenshot displays the 'Projects' page, listing two projects: 'sonarqube' and 'sonarqube-test'. Both are in a 'Passed' state. The bottom screenshot provides a detailed view of the 'sonarqube-test' project, showing its overall status as 'Passed' and a list of quality metrics.

SonarQube Projects List

Project Name	Status	Last Analysis
sonarqube	Passed	21 hours ago
sonarqube-test	Passed	19 minutes ago

SonarQube Project Details: sonarqube-test

Quality Gate: Passed

Reliability: A

Maintainability: A

Security: 0 Open Issues

Reliability: 68k Open Issues

Maintainability: 164k Open Issues

Coverage: 50.6%

Duplications: 50.6%

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

11. Code Problems - Open

The screenshot shows the SonarQube interface for the 'sonarqube-test' project. The 'Measures' tab is selected, displaying a list of components and their associated 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 components and their issue counts:

Component	Open Issues
gameoflife-acceptance-tests	4
gameoflife-build	0
gameoflife-core	603
gameoflife-deploy	0
gameoflife-web	209,940

Issues

The screenshot shows the SonarQube interface for the 'sonarqube-test' project, specifically the 'Issues' tab. The left sidebar shows the 'Issues' section with 'Issues' highlighted. The main content area shows a list of issues with the following details:

Issue	Severity	Effort	Age	Category	Tags	
Use a specific version tag for the image.	Intentionality	L1	5min effort	4 years ago	Code Smell	Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality	L12	5min effort	4 years ago	Code Smell	Major
Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.	Intentionality					

Consistency

The screenshot shows the SonarQube web interface with the 'Issues' tab selected. The left sidebar displays a filter for 'Clean Code Attribute' with 'Consistency' selected, showing 197k issues. The main panel shows a list of issues under the path 'gameoflife-core/build/reports/tests/all-tests.html'. Three issues are visible, all categorized as 'Consistency' with a 'Reliability' attribute. The first issue is 'Insert a <!DOCTYPE> declaration to before this <html> tag.' (L1, 5min effort, 4 years ago, @ Bug - @ Major). The second is 'Remove this deprecated "width" attribute.' (L9, 5min effort, 4 years ago, @ Code Smell - @ Major). The third is 'Remove this deprecated "align" attribute.' (html5, obsolete). A yellow warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrades to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

Intentionality

The screenshot shows the SonarQube web interface with the 'Issues' tab selected. The left sidebar displays a filter for 'Clean Code Attribute' with 'Intentionality' selected, showing 14k issues. The main panel shows a list of issues under the path 'gameoflife-acceptance-tests/Dockerfile'. Three issues are visible, all categorized as 'Intentionality' with a 'Maintainability' attribute. The first issue is 'Use a specific version tag for the image.' (L1, 5min effort, 4 years ago, @ Code Smell - @ Major). The second is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (L12, 5min effort, 4 years ago, @ Code Smell - @ Major). The third is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (No tags). A yellow warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrades to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

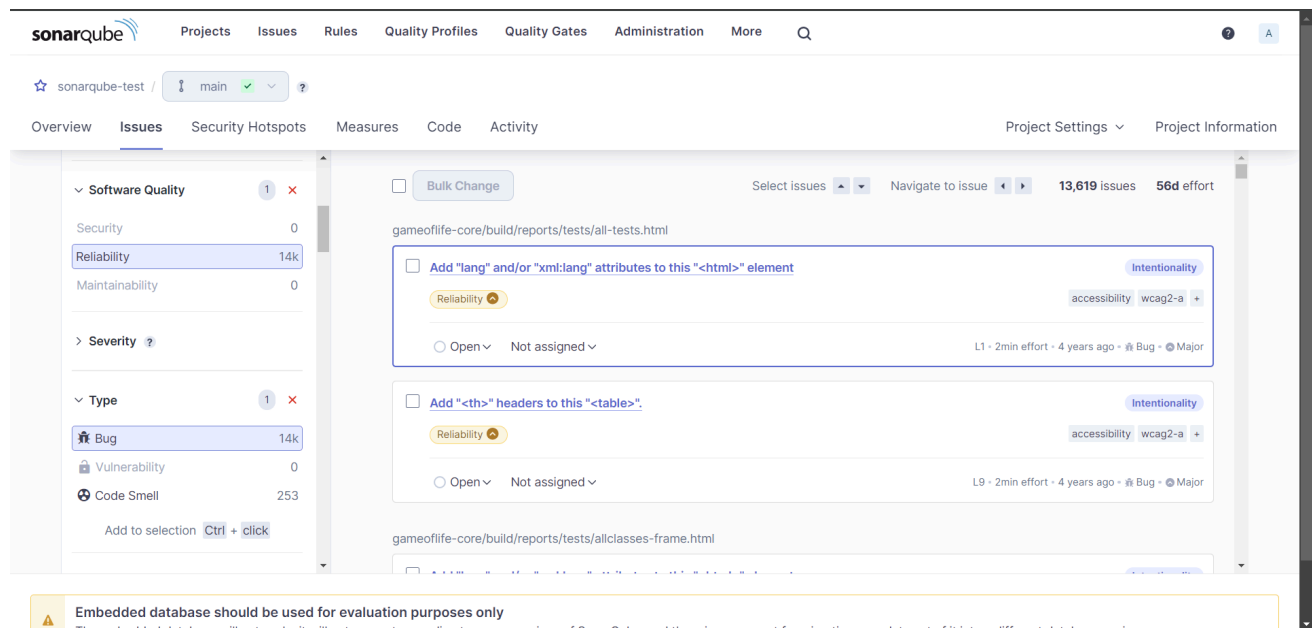
Code Smells

The screenshot shows the SonarQube web interface with the 'Issues' tab selected. The left sidebar displays a filter for 'Software Quality' with 'Code Smell' selected, showing 268 issues. The main panel shows a list of issues under the path 'gameoflife-acceptance-tests/Dockerfile'. Three issues are visible, all categorized as 'Intentionality' with a 'Maintainability' attribute. The first issue is 'Use a specific version tag for the image.' (L1, 5min effort, 4 years ago, @ Code Smell - @ Major). The second is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (L12, 5min effort, 4 years ago, @ Code Smell - @ Major). The third is 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.' (No tags). A yellow warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrades to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

Bugs

The screenshot shows the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The main header shows the project name 'sonarqube-test' and a dropdown menu for 'main'. The left sidebar contains a filter for 'Bug' with 14k results. The main content area displays two issues related to HTML attributes. 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. Both issues are marked as 'Open' and 'Not assigned'.

Reliability

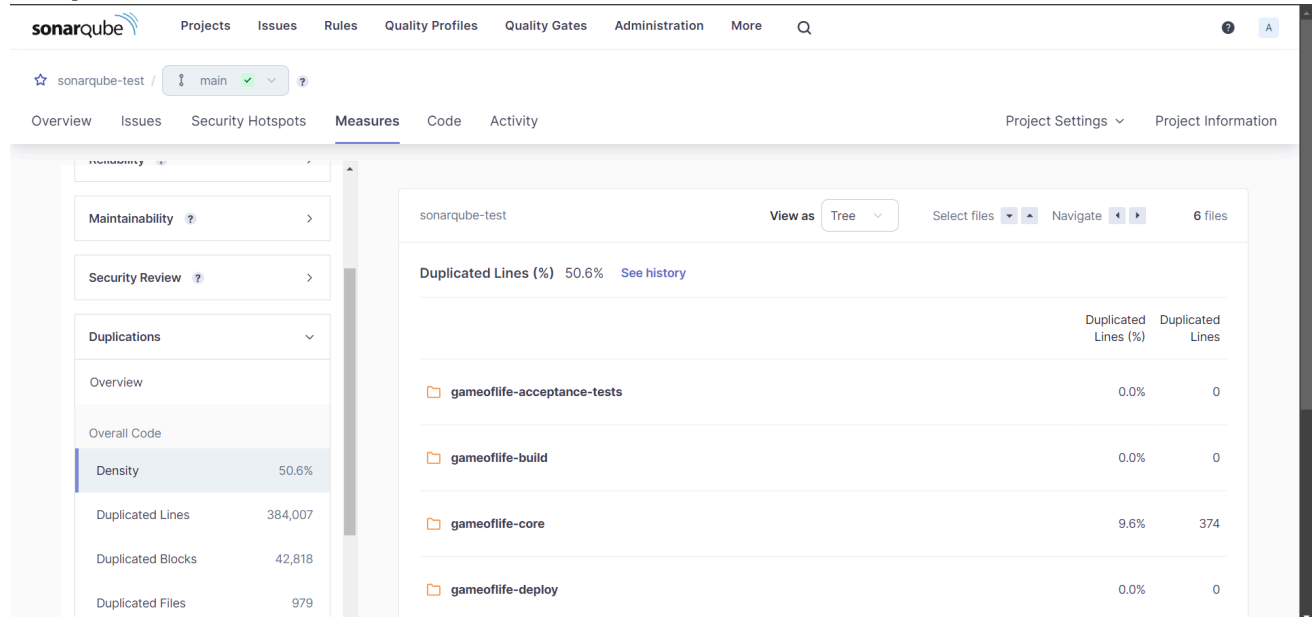


The screenshot shows the SonarQube interface for the 'sonarqube-test' project, specifically the 'Issues' tab. The left sidebar displays a filter for 'Reliability' with 14k issues. The main content area shows two issues related to XML attributes. A yellow banner at the bottom states: 'Embedded database should be used for evaluation purposes only. The embedded database will not scale, it will not support upgrades to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.'

Issues List:

Issue Key	Message	Severity	Effort	Age	Type
L1	Add "lang" and/or "xml:lang" attributes to this "<html>" element	Reliability	2min	4 years ago	Bug - Major
L9	Add "<th>" headers to this "<table>".	Reliability	2min	4 years ago	Bug - Major

Duplicates

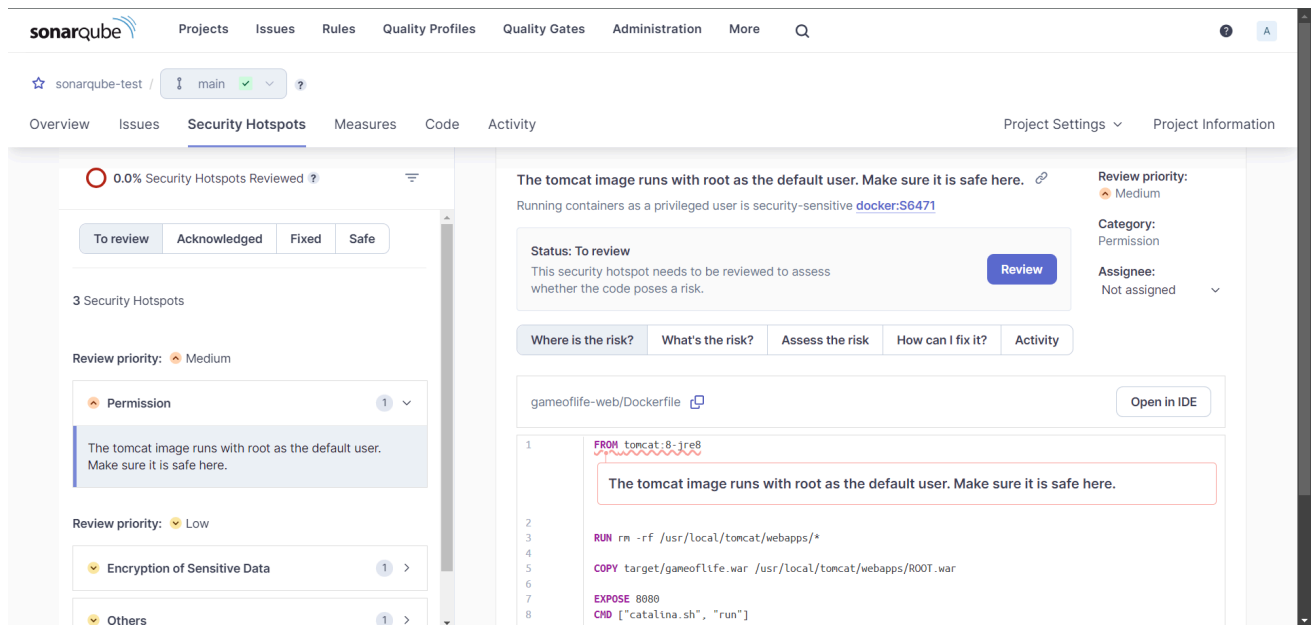


The screenshot shows the SonarQube interface for the 'sonarqube-test' project, specifically the 'Measures' tab. The left sidebar shows the 'Density' measure with a value of 50.6%. The main content area displays a table of duplicated lines across different modules.

Duplicated Lines Summary:

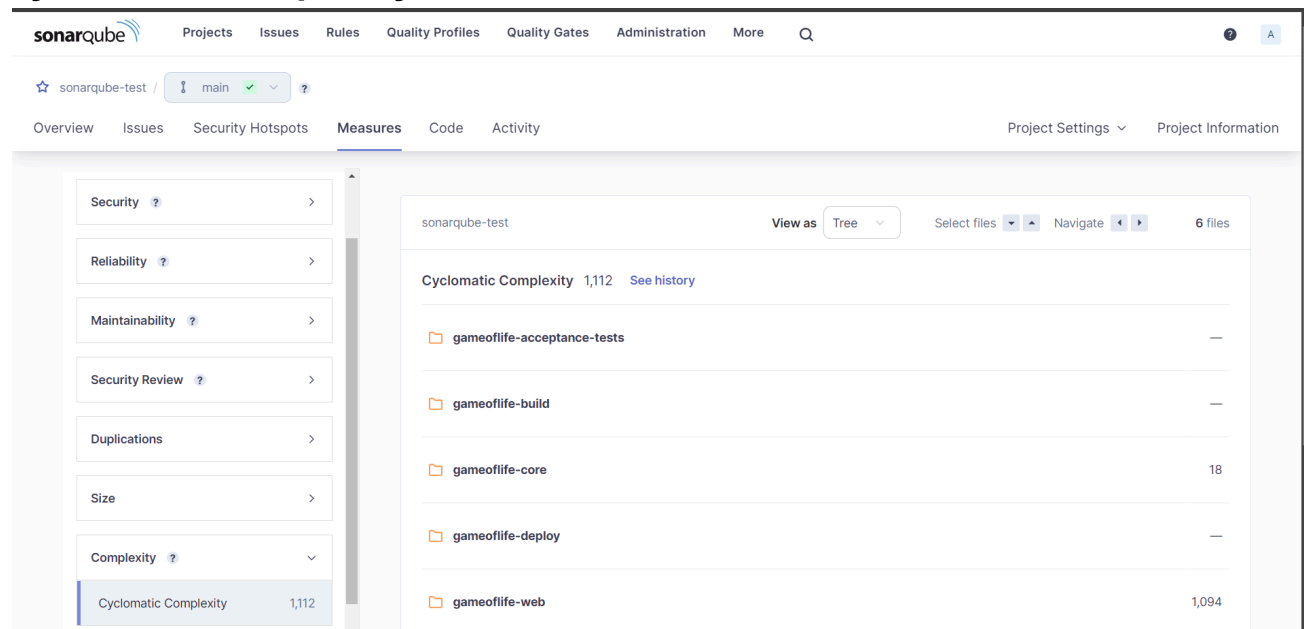
Module	Duplicated Lines (%)	Duplicated Lines
gameoflife-acceptance-tests	0.0%	0
gameoflife-build	0.0%	0
gameoflife-core	9.6%	374
gameoflife-deploy	0.0%	0

Security Hotspot



The screenshot shows the SonarQube interface for a project named 'sonarqube-test'. The 'Security Hotspots' tab is active, displaying a summary of 3 security hotspots. The first hotspot, titled 'Permission', has a 'Medium' review priority and a description: 'The tomcat image runs with root as the default user. Make sure it is safe here.' The second hotspot, 'Encryption of Sensitive Data', has a 'Low' review priority. The third hotspot is under 'Others'. On the right, a detailed view of the 'Permission' hotspot is shown, including a 'Review' button and a 'Where is the risk?' tab. The code snippet for 'gameoflife-web/Dockerfile' is displayed, showing a 'FROM tomcat:8-jre8' line highlighted with a red box and the warning message: 'The tomcat image runs with root as the default user. Make sure it is safe here.'

Cyclomatic Complexity



The screenshot shows the SonarQube interface for the same project, 'sonarqube-test', with the 'Measures' tab active. The left sidebar lists various quality measures, with 'Cyclomatic Complexity' selected. The main area displays the 'Cyclomatic Complexity' measure with a value of 1,112 and a 'See history' link. Below this, a table lists the complexity values for different components of the project:

Component	Cyclomatic Complexity
gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094

We have established a CI/CD pipeline using Jenkins and connected it with SonarQube to identify problems in the code, including bugs, code smells, duplicates, cyclomatic complexities, and more.

Conclusion:

In this experiment, we conducted a static code analysis on our sample Java application. This analysis aimed to identify various issues, including bugs, code smells, and security vulnerabilities. By leveraging tools like SonarQube, we gained insights into the code's quality and potential risks. The findings highlight areas for improvement and ensure better code maintainability. Overall, this process enhances the reliability of our application.