

**Advanced Devops  
Experiment 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)** :SAST is a methodology for testing an application's source code to identify security vulnerabilities before the code is compiled. This type of testing, also referred to as white-box testing, helps improve application security by finding weaknesses early in development.

**Problems SAST Solves**

- **Early Detection:** SAST finds vulnerabilities early in the Software Development Life Cycle (SDLC), allowing developers to fix issues without affecting builds or passing vulnerabilities to the final release.
- **Real-Time Feedback:** Developers receive immediate feedback during coding, helping them address security issues before moving to the next stage of development.
- **Graphical Representations:** SAST tools often provide visual aids to help developers navigate the code and identify the exact location of vulnerabilities, offering suggestions for fixes.
- **Regular Scanning:** SAST tools can be configured to scan code regularly, such as during daily builds, code check-ins, or before releases.

**Importance of SAST**

- **Resource Efficiency:** With a larger number of developers than security experts, SAST allows full codebase analysis quickly and efficiently, without relying on manual code reviews.
- **Speed:** SAST tools can analyze millions of lines of code within minutes, detecting critical vulnerabilities such as buffer overflows, SQL injection, and cross-site scripting (XSS) with high accuracy.

## CI/CD Pipeline

A Continuous Integration/Continuous Delivery (CI/CD) pipeline is a sequence of automated tasks designed to build, test, and deploy new software versions rapidly and consistently. It plays a crucial role in DevOps practices, ensuring fast and reliable software releases.

## SonarQube

SonarQube is an open-source platform from SonarSource that performs continuous code quality inspections through static code analysis. It identifies bugs, code smells, security vulnerabilities, and code duplications in a wide range of programming languages. SonarQube is extendable with plugins and integrates seamlessly into CI/CD pipelines.

## Benefits of SonarQube

**Sustainability:** By reducing complexity and vulnerabilities, SonarQube extends the lifespan of applications and helps maintain cleaner code.

**Increased Productivity:** SonarQube minimizes maintenance costs and risks, resulting in fewer code changes and a more stable codebase.

**Quality Code:** Ensures code quality checks are integrated into the development process.

**Error Detection:** Automatically identifies coding errors and alerts developers to resolve them before moving to production.

**Consistency:** Helps maintain consistent code quality by detecting and reporting violations of coding standards.

**Business Scaling:** SonarQube supports scaling as the business grows without any restrictions.

## Implementation:

### Prerequisites

1. Jenkins installed on your machine.
2. Docker installed to run SonarQube.
3. SonarQube installed via Docker

## 1. Set Up Jenkins

- Open Jenkins Dashboard on localhost:8080 or your configured port

- Install the necessary plugins:
- SonarQube Scanner Plugin

## 2. Run SonarQube in Docker

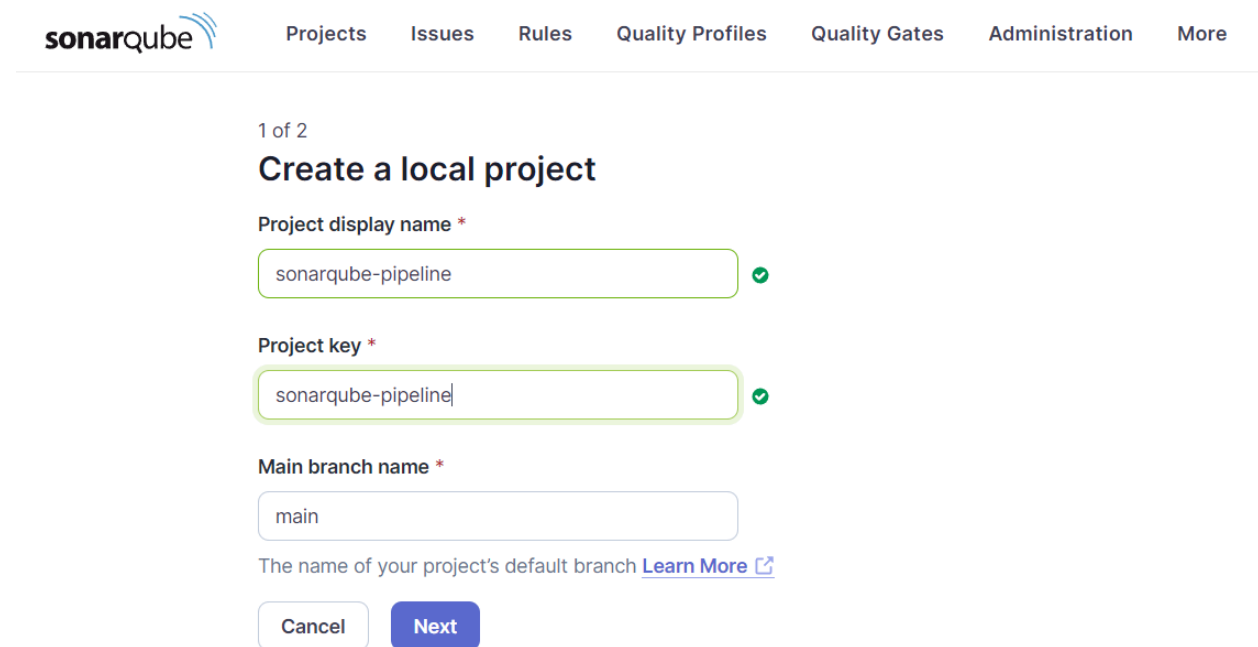
Run the following command to start SonarQube in a Docker container: command

:

```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

- Check SonarQube status at <http://localhost:9000>.
- Login with your credentials:

**Step 1:** Log in to sonarqube portal and create a local project.



The screenshot shows the SonarQube web interface. At the top, there is a navigation bar with the SonarQube logo and links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. Below the navigation bar, the page title is '1 of 2 Create a local project'. The form contains three input fields: 'Project display name \*' with the value 'sonarqube-pipeline', 'Project key \*' with the value 'sonarqube-pipeline', and 'Main branch name \*' with the value 'main'. Each of the first two fields has a green checkmark icon to its right. Below the 'Main branch name' field, there is a note: 'The name of your project's default branch' followed by a link 'Learn More' and an external link icon. At the bottom of the form, there are two buttons: 'Cancel' and 'Next'.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

1 of 2

### Create a local project

Project display name \*

sonarqube-pipeline ✓

Project key \*

sonarqube-pipeline ✓

Main branch name \*

main

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

Cancel Next

## 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.

**Step 2:** Go to [download\\_sonarscanner](#) to download sonar scanner

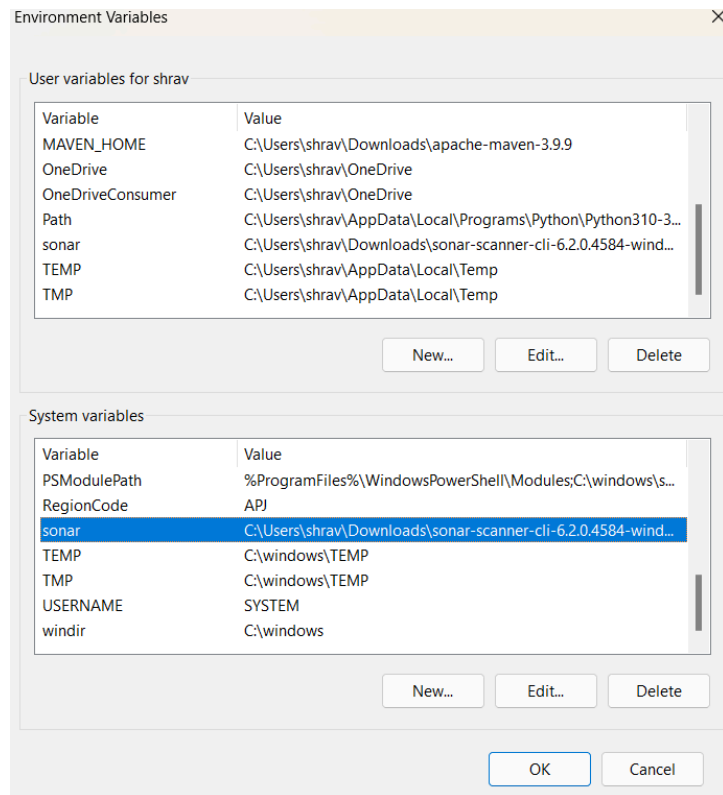
The screenshot shows the SonarScanner CLI download page. The left sidebar contains navigation links for SonarQube 10.6 documentation, including 'Analyzing source code' and 'Scanners'. The main content area displays the 'SonarScanner CLI' section with a table of versions. The table has columns for version number and date. The '6.2' version is the latest, dated '2024-09-17'. Below the version number, it lists download links for various operating systems: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, and Docker. The 'Windows x64' link is highlighted with a red box. Below the download links, there are links for 'Any (Requires a pre-installed JVM)' and 'Release notes'. The table also shows versions 6.1 and 6.0 with their respective dates and download links.

| Version | Date       | Download links                                                                |
|---------|------------|-------------------------------------------------------------------------------|
| 6.2     | 2024-09-17 | Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker       |
| 6.1     | 2024-06-27 | Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, Docker       |
| 6.0     | 2024-06-04 | Linux x64, Windows x64, macOS x64, Docker, Anv (Requires a pre-installed JVM) |

After the download is complete, extract the file and copy the path to bin folder

Go to environment variables, system variables and click on path

Add a new path, paste the path copied earlier.



**Step 3:** Create a New Item in Jenkins, choose Pipeline.

## New Item

Enter an item name

sonarqube-pipeline

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**

OK

## Add pipeline script :

```
node {  
    stage('Cloning the GitHub Repo') {  
        git 'https://github.com/shazforiot/GOL.git'  
    }  
    stage('SonarQube analysis') {  
        withSonarQubeEnv('sonarqube') {  
            bat """
```

```
C:\\Users\\shrav\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat ^
```

```
-Dsonar.login=admin ^
```

```
-Dsonar.password=123456 ^
```

```
-Dsonar.projectKey=sonarqube-pipeline ^
```

```
-Dsonar.exclusions=vendor/**,resources/**,**/*.java ^
```

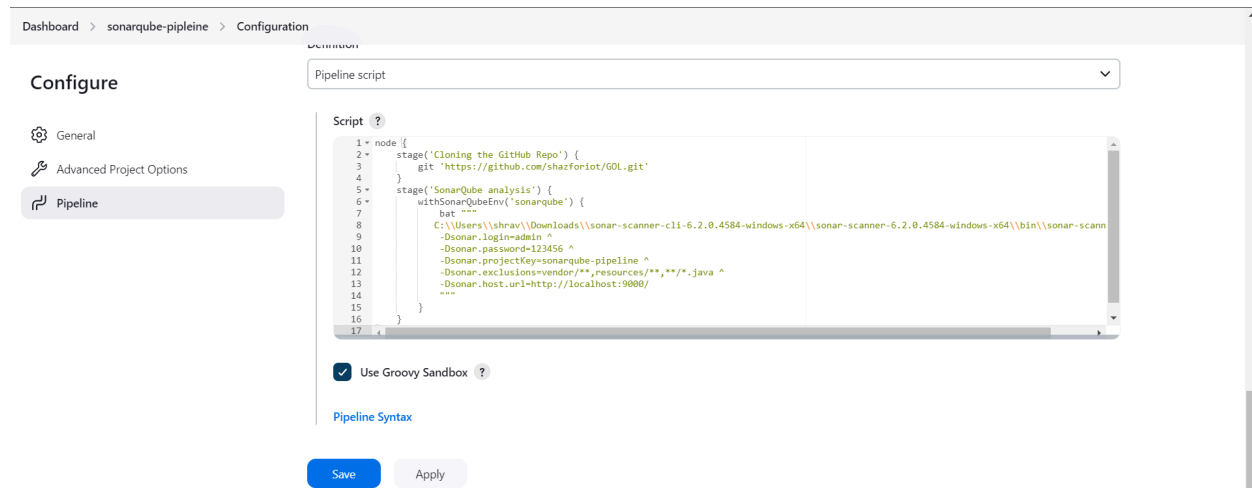
```
-Dsonar.host.url=http://localhost:9000/  
"""
```

```
"""
```

```
}
```

```
}
```

```
}
```



## Step 4: Save the pipeline and build it.

The screenshot shows the Jenkins dashboard for the 'sonarqube-pipeline'. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains various pipeline management options. The main content area displays the 'sonarqube-pipeline' status as 'Success' with a green checkmark. Below this, the 'Stage View' shows a timeline of stages: 'Cloning the GitHub Repo' (16s) and 'SonarQube analysis' (16min 25s). The 'Permalinks' section provides links to the last build, stable build, successful build, and completed build. A 'Build History' widget is also visible on the left.

**Jenkins** Search (CTRL+K) Shrivani Anil Patil

Dashboard > sonarqube-pipeline >

**Status** sonarqube-pipeline Add

Changes  
Build Now  
Configure  
Delete Pipeline  
Full Stage View  
SonarQube  
Stages  
Rename  
Pipeline Syntax

**Stage View**

Average stage times:  
(Average full run time: ~16min 44s)

| Stage                   | Duration  |
|-------------------------|-----------|
| Cloning the GitHub Repo | 16s       |
| SonarQube analysis      | 16min 25s |

**Permalinks**

- Last build (#1), 15 hr ago
- Last stable build (#1), 15 hr ago
- Last successful build (#1), 15 hr ago
- Last completed build (#1), 15 hr ago

**Build History** trend

Filter... /

#1

**Jenkins** Search (CTRL+K) Shrivani Anil Patil log out

Dashboard > sonarqube-pipeline > Stages

**Build sonarqube-pipeline** Build Configure

id pipeline

#1 Start Cloning the Git... SonarQube anal... End

## Console output:



### Console Output

[Download](#)[Copy](#)[View as plain text](#)

Skipping 4,249 KB.. [Full Log](#)

```
18:21:26.359 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 40. Keep only the first 100 references.
18:21:26.359 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 65. Keep only the first 100 references.
18:21:26.359 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 41. Keep only the first 100 references.
18:21:26.361 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 17. Keep only the first 100 references.
18:21:26.361 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/gui/GuiPackage.html for block at line 1487. Keep only the first 100 references.
18:21:26.457 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 226. Keep only the first 100 references.
18:21:26.457 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 229. Keep only the first 100 references.
18:21:26.457 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 225. Keep only the first 100 references.
18:21:26.457 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 226. Keep only the first 100 references.
18:21:26.457 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at line 424. Keep only the first 100 references.
18:21:26.457 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/functions/LongSum.html for block at
```

Dashboard > sonarqube-pipeline > #1

```
block at line 64. Keep only the first 100 references.
18:21:31.284 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/threads/JMeterContext.html for block at line 41. Keep only the first 100 references.
18:21:31.284 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/threads/JMeterContext.html for block at line 17. Keep only the first 100 references.
18:21:31.284 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/threads/JMeterContext.html for block at line 669. Keep only the first 100 references.
18:21:31.287 INFO CPD Executor CPD calculation finished (done) | time=164132ms
18:21:31.372 INFO SCM revision ID 'ba799ba7e1b576f04d612322b0412c5e6e1e5e4'
18:21:40.154 INFO Analysis report generated in 7273ms, dir size=127.2 MB
18:21:54.587 INFO Analysis report compressed in 14431ms, zip size=29.6 MB
18:22:03.090 INFO Analysis report uploaded in 8493ms
18:22:03.111 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-pipeline
18:22:03.111 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:22:03.111 INFO More about the report processing at http://localhost:9000/api/ce/task?id=84bfa1a9-afee-43bf-bab1-28d02c29cea6
18:23:07.347 INFO Analysis total time: 16:01.445 s
18:23:07.410 INFO SonarScanner Engine completed successfully
18:23:08.133 INFO EXECUTION SUCCESS
18:23:08.251 INFO Total time: 16:15.458s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



## Step 5: After that, check the project in SonarQube

The screenshot shows the SonarQube Overview page for the 'sonarqube-pipeline' project on the 'main' branch. The Quality Gate is 'Passed' (indicated by a green checkmark). A warning message states: 'The last analysis has warnings. [See details](#)'. Below this, there are tabs for 'New Code' and 'Overall Code'. The 'Overall Code' tab is active, showing a summary of issues across different categories:

| Category        | Open Issues | High (H) | Medium (M) | Low (L) |
|-----------------|-------------|----------|------------|---------|
| Security        | 0           | 0        | 0          | 0       |
| Reliability     | 68k         | 0        | 47k        | 21k     |
| Maintainability | 164k        | 7        | 143k       | 21k     |

Additional metrics shown include: Accepted issues: 0, Coverage: 50.6%, and Duplications: 50.6%.

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

The screenshot shows the SonarQube Issues page for the 'sonarqube-pipeline' project on the 'main' branch. The 'Issues' tab is selected. On the left, there is a sidebar with filters for Severity, Type, Scope, Status, and Security Category. The main area displays a list of issues, each with a checkbox, a description, a severity level, and a status. The issues listed are:

- ☐ [Remove this deprecated "width" attribute.](#) (Maintainability, html5, obsolete, Major)
- ☐ [Remove this deprecated "cellpadding" attribute.](#) (Maintainability, html5, obsolete, Major)
- ☐ [Remove this deprecated "valign" attribute.](#) (Maintainability, html5, obsolete, Major)

At the bottom, there is a warning message: 'Embedded database should be used for evaluation purposes only'.

sonarqube-pipeline / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

Severity

Type

Scope

Status

Security Category

gameoflife-acceptance-tests/Dockerfile

Use a specific version tag for the image.

Maintainability

OpenNot assigned

L1 · 5min effort · 4 years ago · Code Smell · Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

Maintainability

OpenNot assigned

L12 · 5min effort · 4 years ago · Code Smell · Major

Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.

Maintainability

OpenNot assigned

L12 · 5min effort · 4 years ago · Code Smell · Major

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.

sonarqube-pipeline / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

Severity

Type

Scope

Status

Security Category

Add "lang" and/or "xml:lang" attributes to this "<html>" element

Reliability

OpenNot assigned

L1 · 2min effort · 4 years ago · Bug · Major

Add "<th>" headers to this "<table>".

Reliability

OpenNot assigned

L9 · 2min effort · 4 years ago · Bug · Major

Remove this deprecated "width" attribute.

Maintainability

OpenNot assigned

html5 · obsolete · Consistency

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.

sonarqube-pipeline / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

|                             | Lines of Code | Security | Reliability | Maintainability | Security Hotspots | Coverage | Duplications |
|-----------------------------|---------------|----------|-------------|-----------------|-------------------|----------|--------------|
| sonarqube-pipeline          | —             | —        | —           | —               | —                 | —        | —            |
| gameoflife-acceptance-tests | 164           | 0        | 0           | 4               | 2                 | —        | 0.0%         |
| gameoflife-build            | 368           | 0        | 0           | 0               | 0                 | —        | 0.0%         |
| gameoflife-core             | 3,675         | 0        | 172         | 529             | 0                 | —        | 9.6%         |
| gameoflife-deploy           | 69            | 0        | 0           | 0               | 0                 | —        | 0.0%         |
| gameoflife-web              | 678,148       | 0        | 67452       | 163246          | 1                 | —        | 50.9%        |
| pom.xml                     | 459           | 0        | 0           | 2               | 0                 | —        | 0.0%         |

sonarqube-pipeline / main

Overview Issues Security Hotspots **Measures** Code Activity

Project Settings Project Information

Issues 0

Rating A

Remediation Effort 0

Reliability ?

Overview

Overall Code

Issues 67624

Rating C

Remediation Effort 1426d

sonarqube-pipeline View as Tree Select files Navigate 6 files

Issues 67624 See history

gameoflife-acceptance-tests 0

gameoflife-build 0

gameoflife-core 172

gameoflife-deploy 0

gameoflife-web 67452

sonarqube-pipeline / main

Overview Issues Security Hotspots **Measures** Code Activity

Project Settings Project Information

Security ?

Reliability ?

Maintainability ?

Security Review ?

Duplications

Size

Lines of Code 682,883

Lines 759,093

sonarqube-pipeline View as Tree Select files Navigate 6 files

Lines of Code 682,883 See history

HTML 678k

XML 4.7k

JSP 332

CSS 110

Docker 19

gameoflife-acceptance-tests 164

gameoflife-build 368

gameoflife-core 3,675

sonarqube-pipeline / main

Overview Issues Security Hotspots **Measures** Code Activity

Project Settings Project Information

Project Overview

Security ?

Overview

Overall Code

Issues 0

Rating A

Remediation Effort 0

Reliability ?

Maintainability ?

Security Overview ?

(Only showing data for the first 500 files)

See the data presented on this chart as a list

Color: Security Rating Size: Vulnerabilities

Zoom: 103% Reset

Security Remediation Effort

