Arnav Santosh Sawant   Advance DevOps Exp-8   Roll-no:- D15A/53

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

Static Application Security Testing (SAST)

SAST is a testing methodology that analyzes source code to identify security vulnerabilities, making applications less prone to attacks. This is a form of **white-box testing** since it scans the application before the code is compiled.

Problems SAST Solves:

- **Early Detection:** Identifies vulnerabilities early in the Software Development Life Cycle (SDLC), allowing developers to fix issues before builds are broken or vulnerabilities are passed to the final release.
- **Real-Time Feedback:** Provides immediate feedback to developers as they code, helping them address issues before moving to the next phase.
- **Graphical Representations:** Offers visual aids to help navigate the codebase, pinpointing exact locations of vulnerabilities and providing guidance on how to resolve them.
- **Regular Scanning:** Should be run frequently during daily/monthly builds, code check-ins, or before code releases.

Importance of SAST:

- **Resource Efficiency:** Developers far outnumber security staff, making manual code reviews difficult. SAST tools can analyze the entire codebase efficiently.
- **Speed:** Capable of scanning millions of lines of code within minutes, it identifies critical vulnerabilities like buffer overflows, SQL injections, and cross-site scripting with high accuracy.

CI/CD Pipeline

A **CI/CD pipeline** (Continuous Integration/Continuous Delivery) is the backbone of the DevOps approach, streamlining software releases. It automates tasks such as building code, running tests, and deploying new software versions. The pipeline helps in delivering software quickly and reliably by connecting these tasks in a sequence.

SonarQube

SonarQube is an open-source platform developed by SonarSource for continuous inspection of code quality. It performs static code analysis, providing detailed reports on bugs, code smells, vulnerabilities, and code duplications. SonarQube supports over 25 major programming languages and can be extended using various plugins.

Benefits of SonarQube:

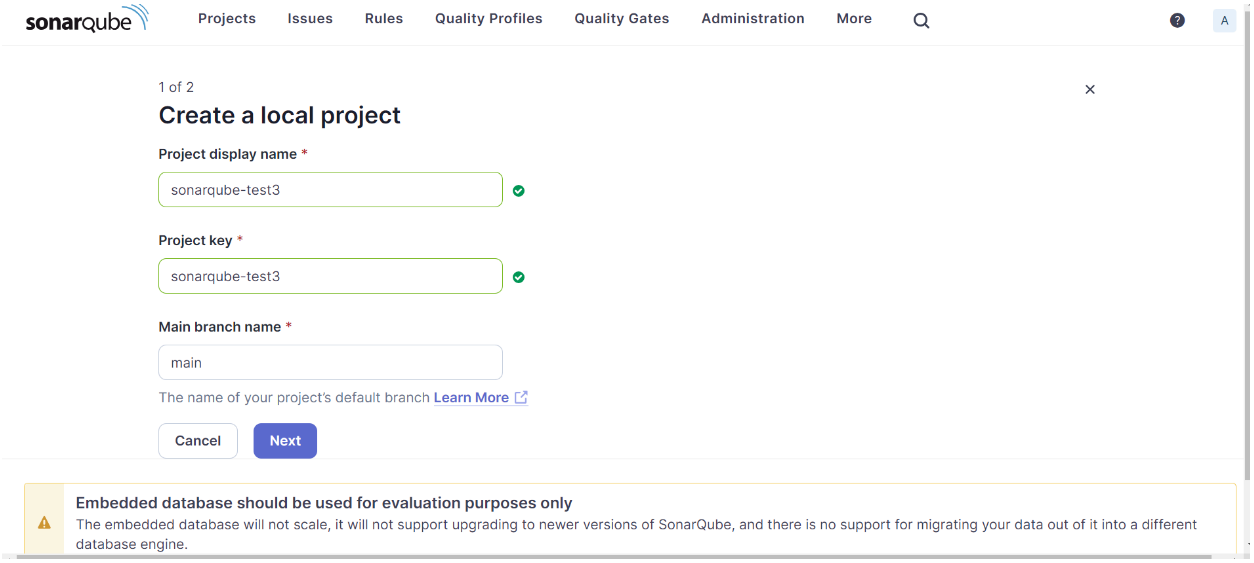
- **Sustainability:** Reduces complexity, vulnerabilities, and code duplications, optimizing the lifespan of an application.
- **Increased Productivity:** Lowers maintenance costs and risks by minimizing the need for extensive code changes.
- **Quality Code:** Ensures code quality is a key part of the software development process.
- **Error Detection:** Automatically detects errors, alerting developers to fix them before they reach the final output.
- **Consistency:** Identifies breaches in code quality standards, improving overall code consistency.
- **Business Scaling:** Supports seamless scaling without restrictions.

Implementation:

Prerequisites:

1. **Jenkins** installed on your machine.
2. **Docker** installed to run SonarQube.
3. **SonarQube** set up using Docker.

Step-1 - Open your jenkins and create a new project in sonarqube



1 of 2

Create a local project

Project display name *

sonarqube-test3

Project key *

sonarqube-test3

Main branch name *

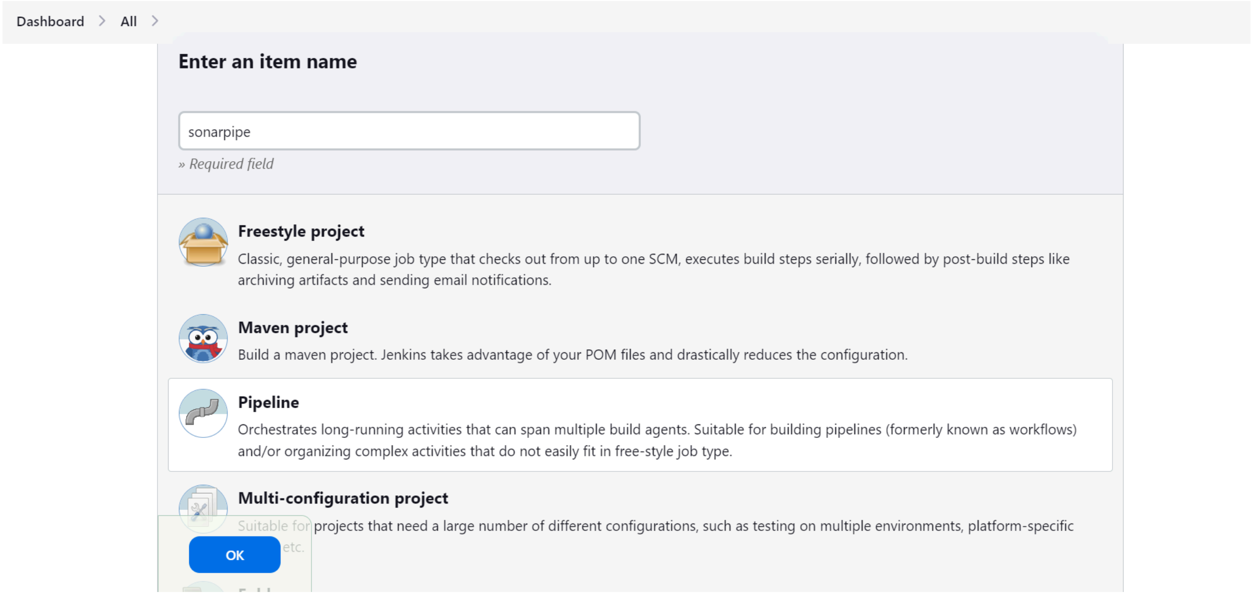
main

The name of your project's default branch Learn More

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.

Step-2 - Create a pipeline



Enter an item name

sonarpipe

» Required field

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.

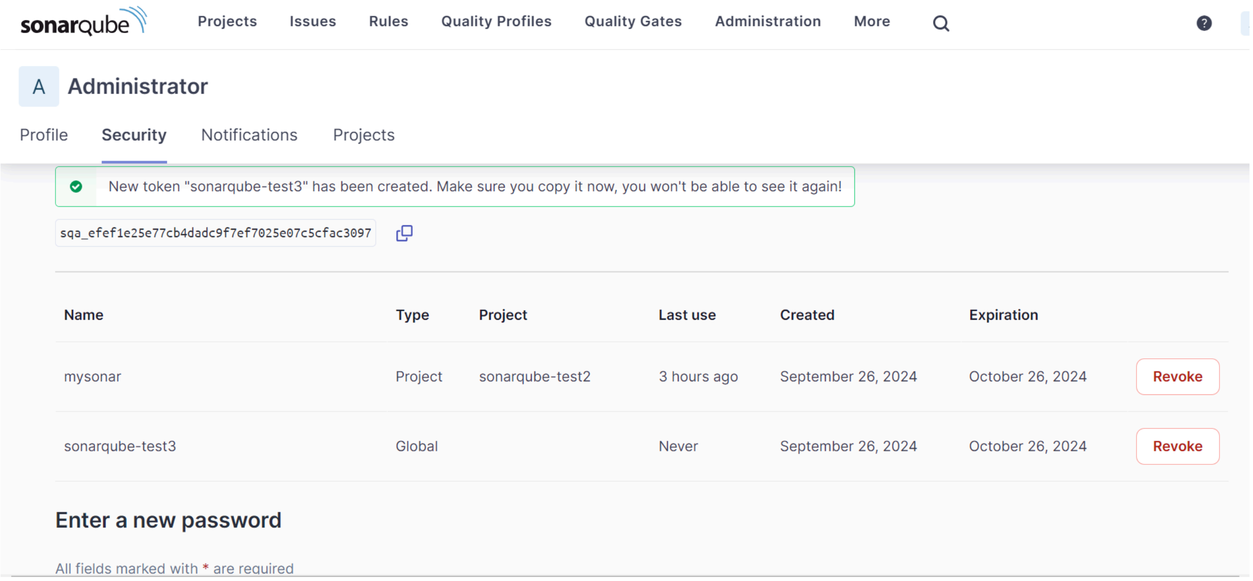
Pipeline

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 etc.

Step-3 - Create a new global token (optional)



A Administrator

Profile Security Notifications Projects

New token "sonarqube-test3" has been created. Make sure you copy it now, you won't be able to see it again!

```
sqa_efef1e25e77cb4dadc9f7ef7025e07c5cfac3097
```

Name	Type	Project	Last use	Created	Expiration	
mysonar	Project	sonarqube-test2	3 hours ago	September 26, 2024	October 26, 2024	Revoke
sonarqube-test3	Global		Never	September 26, 2024	October 26, 2024	Revoke

Enter a new password

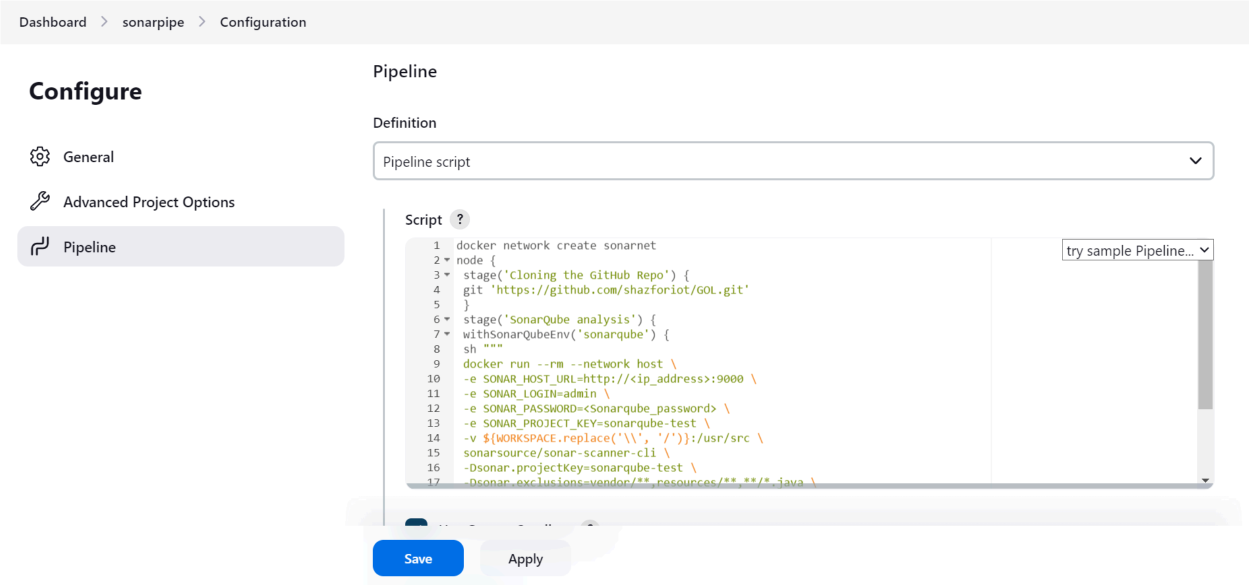
All fields marked with * are required

Step-4 - Once you created that pipeline add the below code in the pipeline script

Use the below line of code and paste it in the pipeline script -

```
 docker network create

 node {

   stage('Cloning the GitHub Repo'){

     git 'https://github.com/shazforiot/GOL.git'

   }

   stage('SonarQube analysis') {

        withSonarQubeEnv('Sona
          rQube-server') { bat """
             “C:\\Users\\Santosh
    Sawant\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-sc
    anner- 6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat” ^
```

```
        -D sonar.login=admin ^

        -D sonar.password=pranav ^

        -D sonar.projectKey=sonarqube-test-project-2 ^

        -D sonar.exclusions=vendor/**,resources/**,**/*.java ^

        -D sonar.host.url=http://localhost:9000/ """
      }

    }

}
```



Pipeline

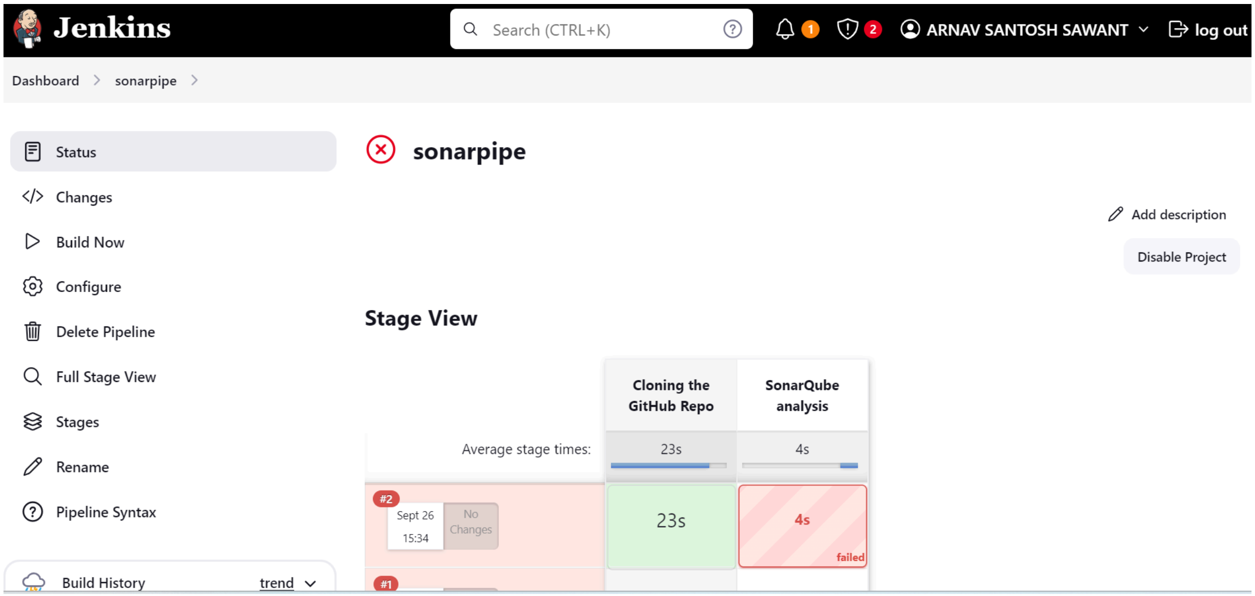
Definition

Pipeline script

Script ?

```
docker network create sonarnet
node {
 stage('Cloning the GitHub Repo') {
 git 'https://github.com/shazforiot/GOL.git'
 }
 stage('SonarQube analysis') {
 withSonarQubeEnv('sonarqube') {
 sh """
 docker run --rm --network host \
 -e SONAR_HOST_URL=http://<ip_address>:9000 \
 -e SONAR_LOGIN=admin \
 -e SONAR_PASSWORD=<Sonarqube_password> \
 -e SONAR_PROJECT_KEY=sonarqube-test \
 -v ${WORKSPACE.replace('\\', '/')}:/usr/src \
 sonarsource/sonar-scanner-cli \
 -Dsonar.projectKey=sonarqube-test \
 -Dsonar.exclusions=vendor/**,resources/**,**/*.java \
```

Step-5 - Run the code and fix the bugs and run again … it takes a lot of time to show success or to actually build



Jenkins

Search (CTRL+K)

1 2 ARNAV SANTOSH SAWANT log out

Dashboard > sonarpipe >

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

Build History trend

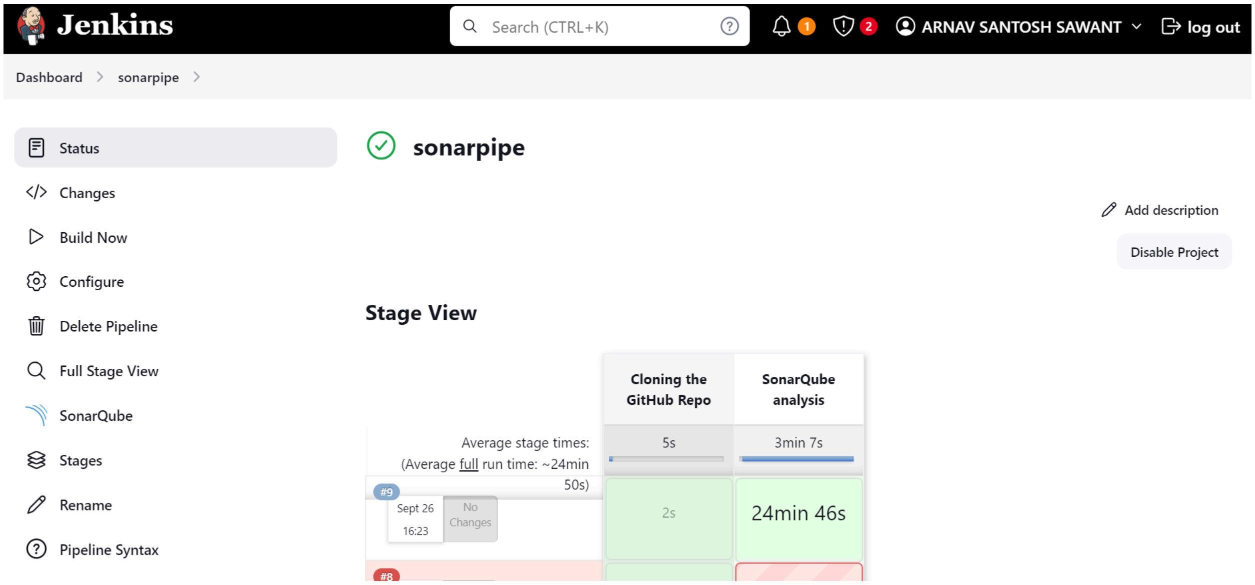
sonarpipe

Add description

Disable Project

Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times:	23s	4s
#2 Sept 26 15:34 No Changes	23s	4s failed
#1		



Jenkins

Search (CTRL+K)

1 2 ARNAV SANTOSH SAWANT log out

Dashboard > sonarpipe >

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

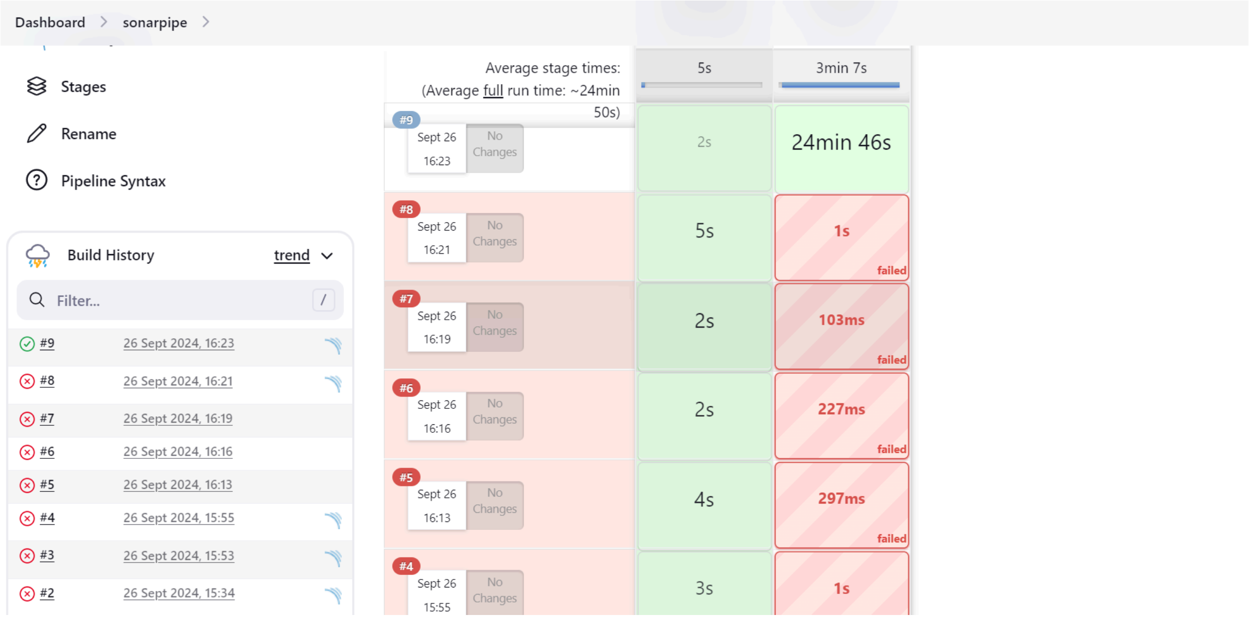
sonarpipe

Add description

Disable Project

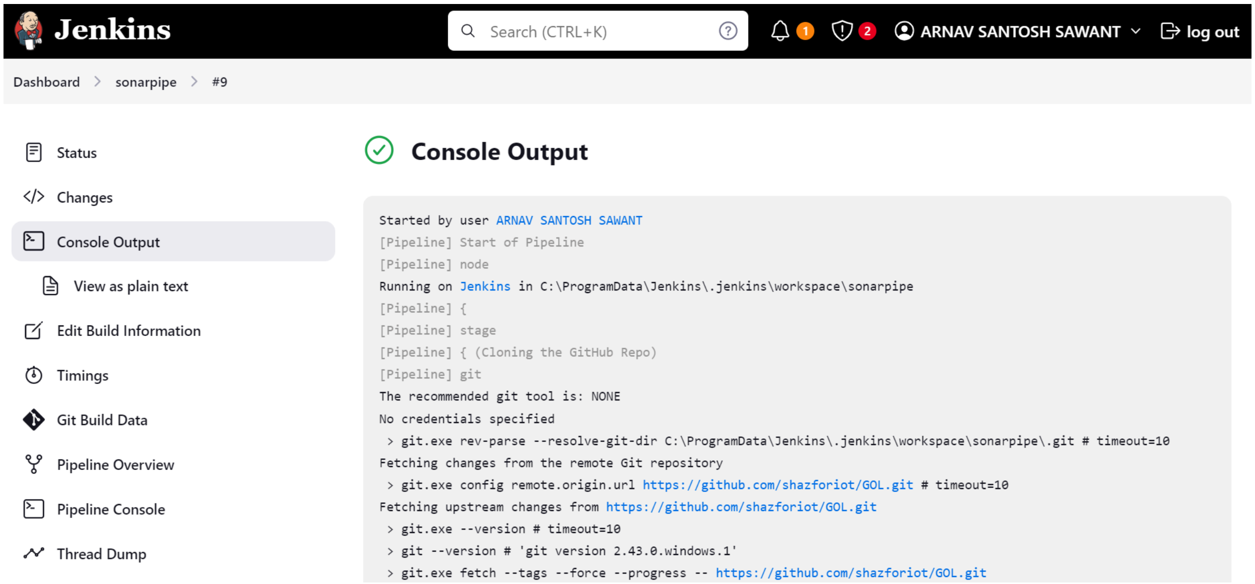
Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~24min 50s)	5s	3min 7s
#9 Sept 26 16:23 No Changes	2s	24min 46s
#8		



Build	Date
#9	26 Sept 2024, 16:23
#8	26 Sept 2024, 16:21
#7	26 Sept 2024, 16:19
#6	26 Sept 2024, 16:16
#5	26 Sept 2024, 16:13
#4	26 Sept 2024, 15:55
#3	26 Sept 2024, 15:53
#2	26 Sept 2024, 15:34

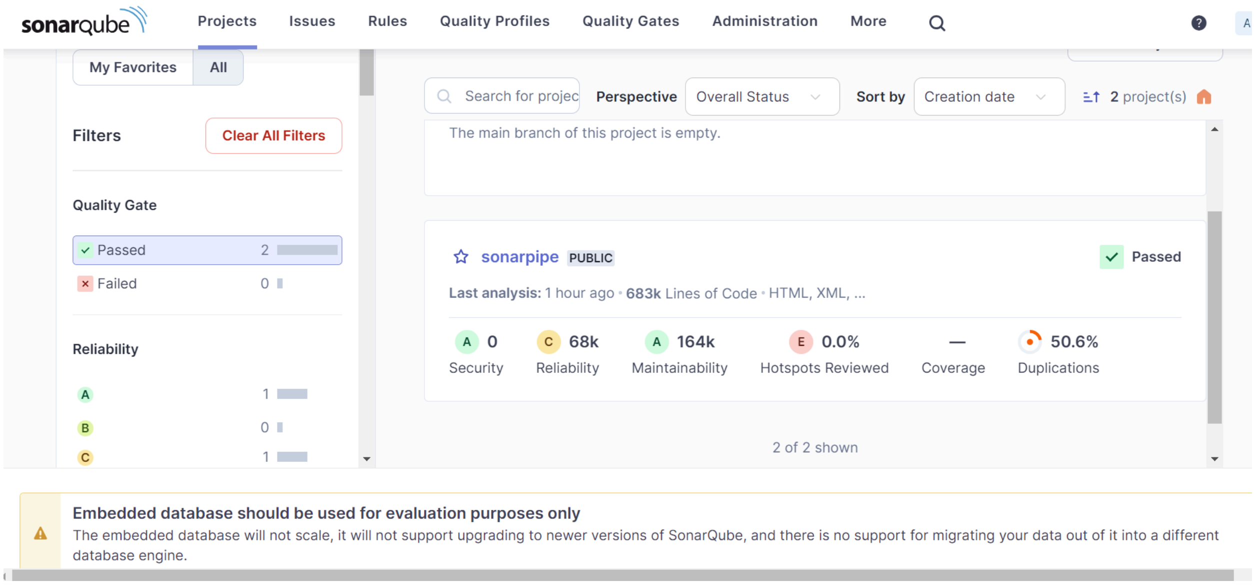
Average stage times: (Average full run time: ~24min 50s)		5s	3min 7s
#9 Sept 26 16:23	No Changes	2s	24min 46s
#8 Sept 26 16:21	No Changes	5s	1s failed
#7 Sept 26 16:19	No Changes	2s	103ms failed
#6 Sept 26 16:16	No Changes	2s	227ms failed
#5 Sept 26 16:13	No Changes	4s	297ms failed
#4 Sept 26 15:55	No Changes	3s	1s



Console Output

```
Started by user ARNAV SANTOSH SAWANT
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\sonarpipe
[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\sonarpipe\.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 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/GOL.git
```

Step-6 - Once ran successfully you can check your project on sonarqube, then particularly analyze that project for issues



Projects · Issues · Rules · Quality Profiles · Quality Gates · Administration · More

My Favorites | All

Filters — Clear All Filters

Quality Gate

✓ Passed	2
✕ Failed	0

Reliability

A	1
B	0
C	1

Search for projec · Perspective: Overall Status · Sort by: Creation date · 2 project(s)

The main branch of this project is empty.

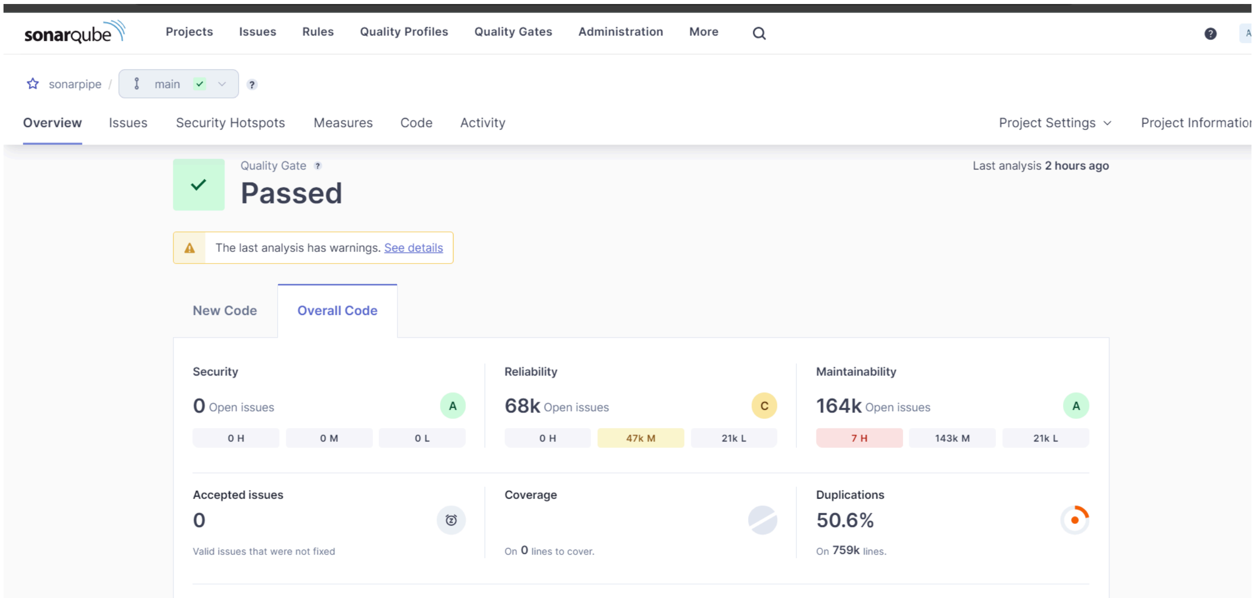
☆ sonarpipe PUBLIC — ✓ Passed

Last analysis: 1 hour ago • 683k Lines of Code • HTML, XML, ...

A 0	C 68k	A 164k	E 0.0%	—	50.6%
Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications

2 of 2 shown

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.



Projects · Issues · Rules · Quality Profiles · Quality Gates · Administration · More

☆ sonarpipe / main ✓

Overview · Issues · Security Hotspots · Measures · Code · Activity — Project Settings · Project Informatio

Quality Gate ✓ **Passed** — Last analysis 2 hours ago

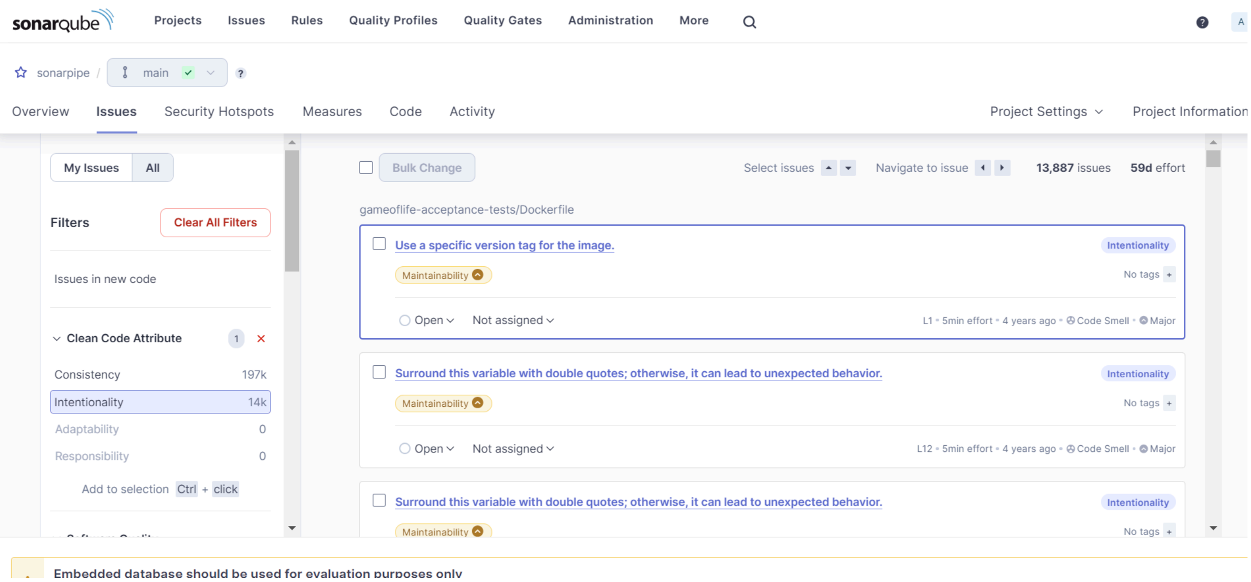
The last analysis has warnings. See details

New Code | Overall Code

Security	Reliability	Maintainability
0 Open issues — A	68k Open issues — C	164k Open issues — A
0 H · 0 M · 0 L	0 H · 47k M · 21k L	7 H · 143k M · 21k L

Accepted issues	Coverage	Duplications
0		50.6%
Valid issues that were not fixed	On 0 lines to cover.	On 759k lines.

intentionality-



Issues in new code

Clean Code Attribute 1 ✕

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Add to selection Ctrl + click

13,887 issues 59d effort

gameoflife-acceptance-tests/Dockerfile

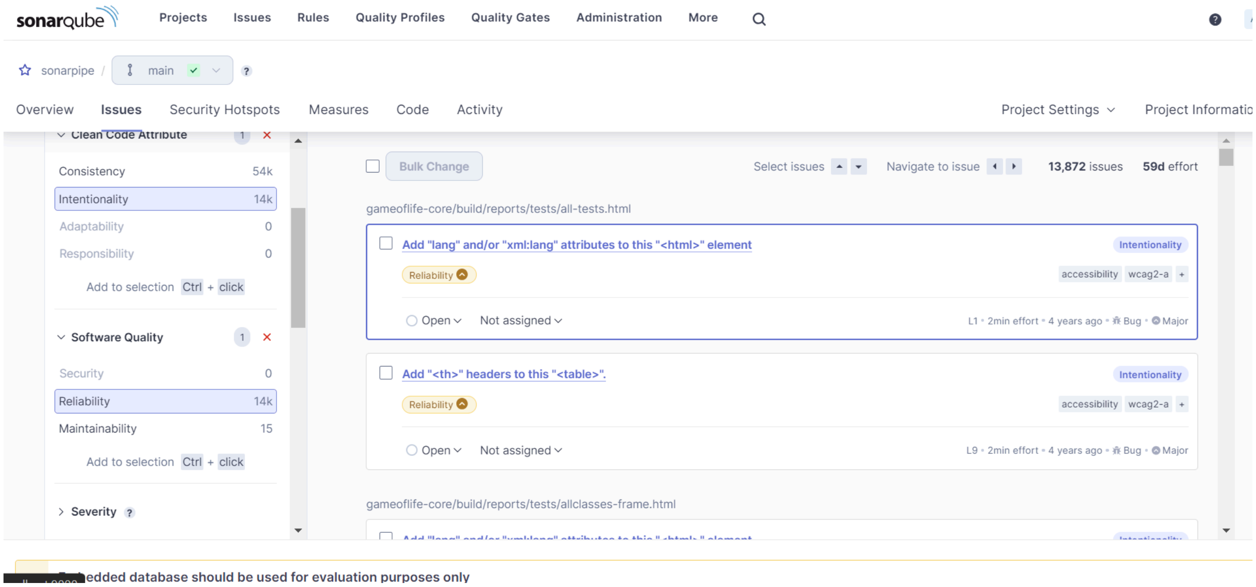
□ Use a specific version tag for the image. Intentionality
Maintainability No tags +
○ Open Not assigned L1 • 5min effort • 4 years ago • Code Smell • Major

□ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability No tags +
○ Open Not assigned L12 • 5min effort • 4 years ago • Code Smell • Major

□ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability No tags +

Embedded database should be used for evaluation purposes only

reliability-



Clean Code Attribute 1 ✕

Consistency	54k
Intentionality	14k
Adaptability	0
Responsibility	0

Add to selection Ctrl + click

Software Quality 1 ✕

Security	0
Reliability	14k
Maintainability	15

Add to selection Ctrl + click

Severity ?

13,872 issues 59d effort

gameoflife-core/build/reports/tests/all-tests.html

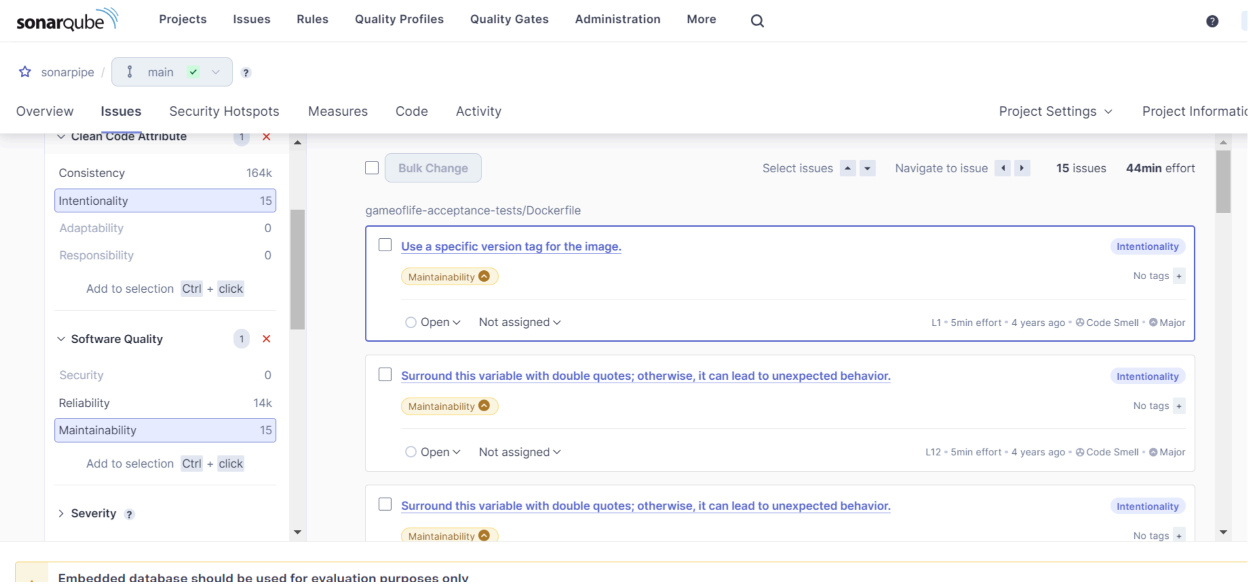
□ Add "lang" and/or "xml:lang" attributes to this "<html>" element Intentionality
Reliability accessibility wcag2-a +
○ Open Not assigned L1 • 2min effort • 4 years ago • Bug • Major

□ Add "<th>" headers to this "<table>". Intentionality
Reliability accessibility wcag2-a +
○ Open Not assigned L9 • 2min effort • 4 years ago • Bug • Major

gameoflife-core/build/reports/tests/allclasses-frame.html

Embedded database should be used for evaluation purposes only

maintainability-



Clean Code Attribute 1 ✕

Consistency	164k
Intentionality	15
Adaptability	0
Responsibility	0

Add to selection Ctrl + click

Software Quality 1 ✕

Security	0
Reliability	14k
Maintainability	15

Add to selection Ctrl + click

Severity ?

□ Bulk Change Select issues Navigate to issue **15** issues **44min** effort

gameoflife-acceptance-tests/Dockerfile

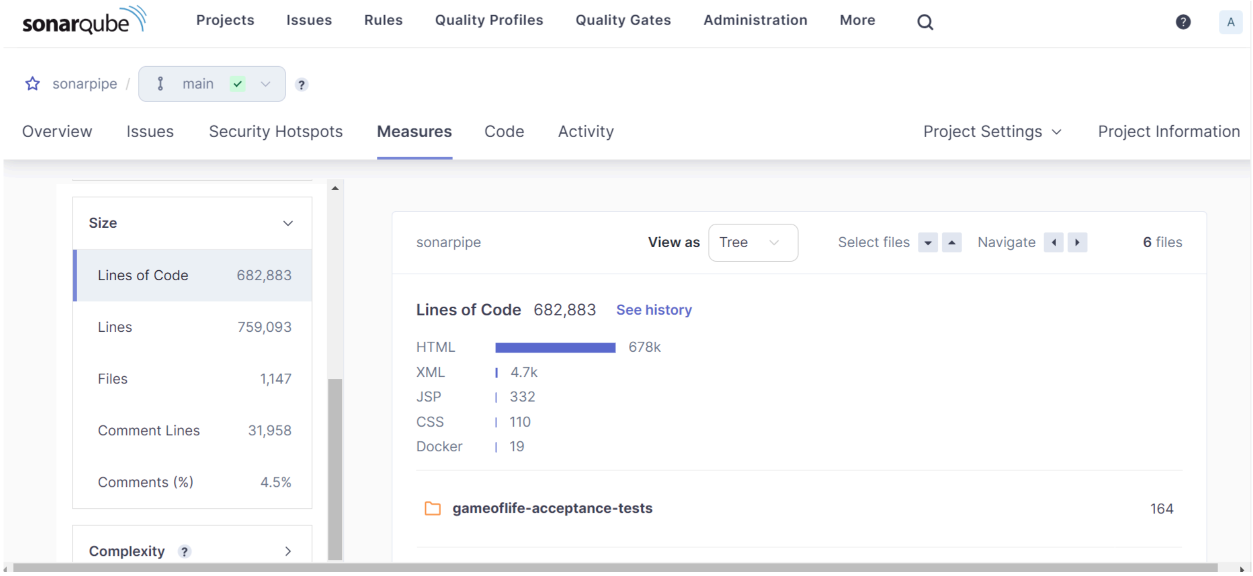
□ Use a specific version tag for the image. Intentionality
Maintainability No tags +
○ Open Not assigned L1 • 5min effort • 4 years ago • Code Smell • Major

□ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability No tags +
○ Open Not assigned L12 • 5min effort • 4 years ago • Code Smell • Major

□ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability No tags +

Embedded database should be used for evaluation purposes only

Lines of code-



Size

Lines of Code	682,883
Lines	759,093
Files	1,147
Comment Lines	31,958
Comments (%)	4.5%

Complexity ?

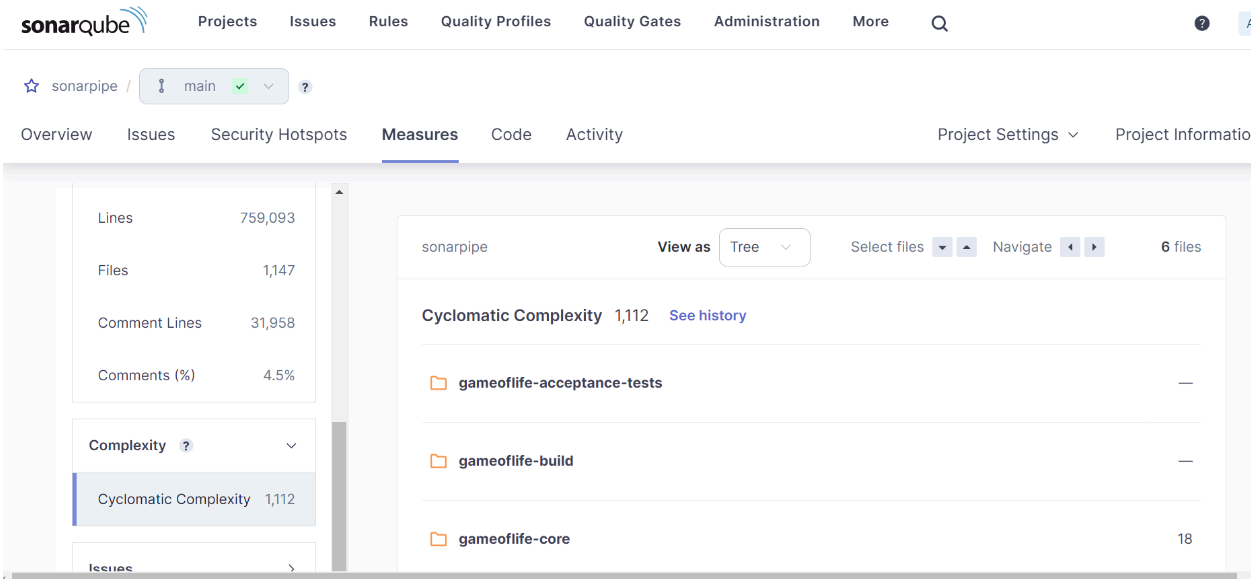
sonarpipe View as Tree Select files Navigate **6** files

Lines of Code 682,883 See history

HTML	678k
XML	4.7k
JSP	332
CSS	110
Docker	19

gameoflife-acceptance-tests 164

Cyclomatic complexity-



Lines	759,093
Files	1,147
Comment Lines	31,958
Comments (%)	4.5%
Cyclomatic Complexity	1,112

sonarpipe — 6 files

Cyclomatic Complexity 1,112

gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18

Overall after this you can analyze your project entirely by visiting various tabs in the sonarqube projects section.