Rujuta Medhi D15A-28

EXPERIMENT: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.

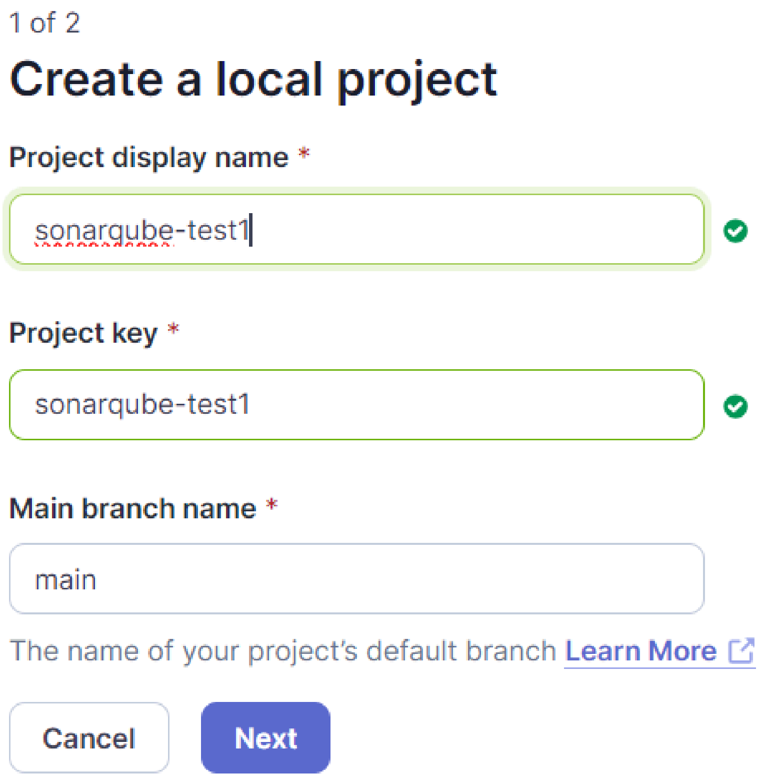
Benefits of SonarQube

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:
- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

1. Login to SonarQube create a manual project in SonarQube with the name sonarqube-test



1 of 2

Create a local project

Project display name *

sonarqube-test1

Project key *

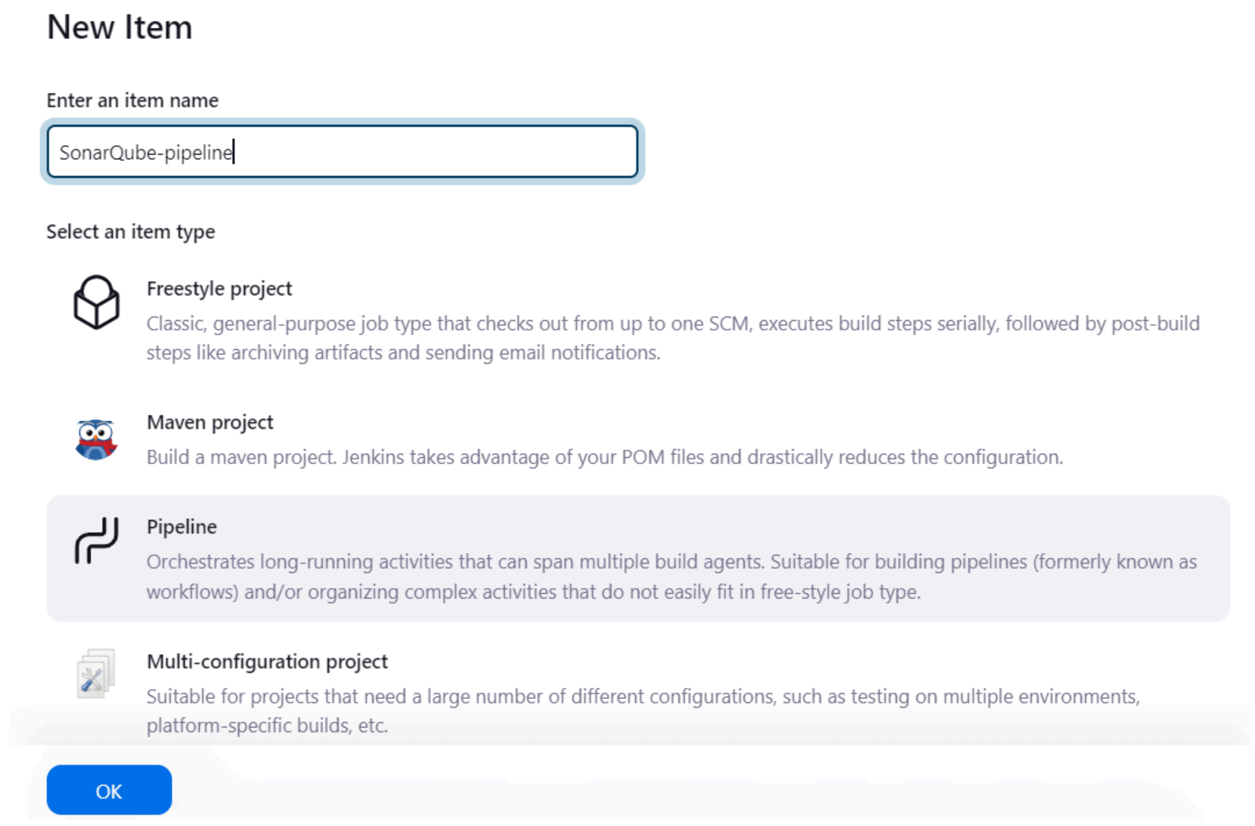
sonarqube-test1

Main branch name *

main

The name of your project's default branch Learn More

2. 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.

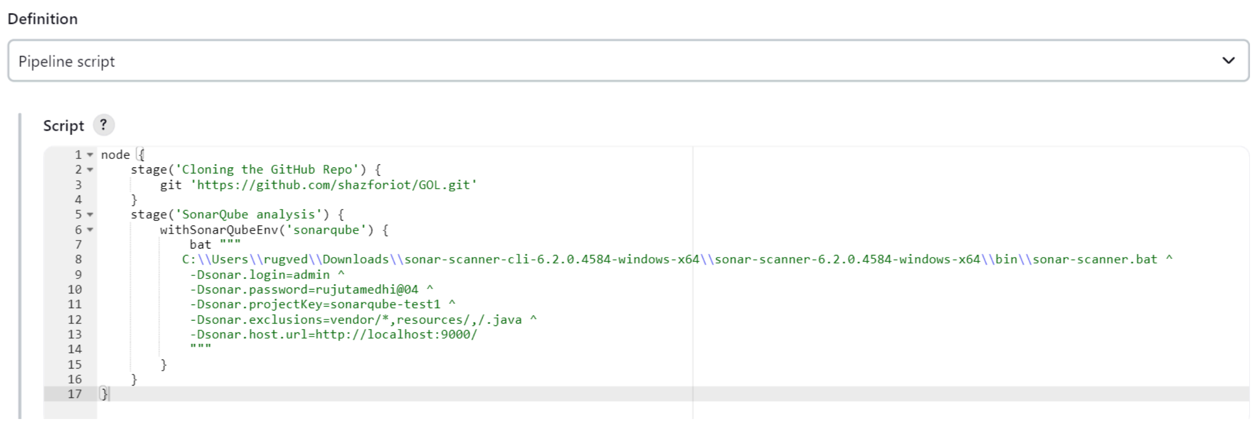
Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

3. Under Pipeline Script, enter the following -

```
node {
stage('Cloning the GitHub Repo') {
git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
sh "<PATH_TO_SONARQUBE_FOLDER>//bin//sonar-scanner \
-D sonar.login=<SonarQube_USERNAME> \
-D sonar.password=<SonarQube_PASSWORD> \
-D sonar.projectKey=<Project_KEY> \
-D sonar.exclusions=vendor/**,resources/**,**/*.java \
-D sonar.host.url=http://127.0.0.1:9000/"
}
}
}
```



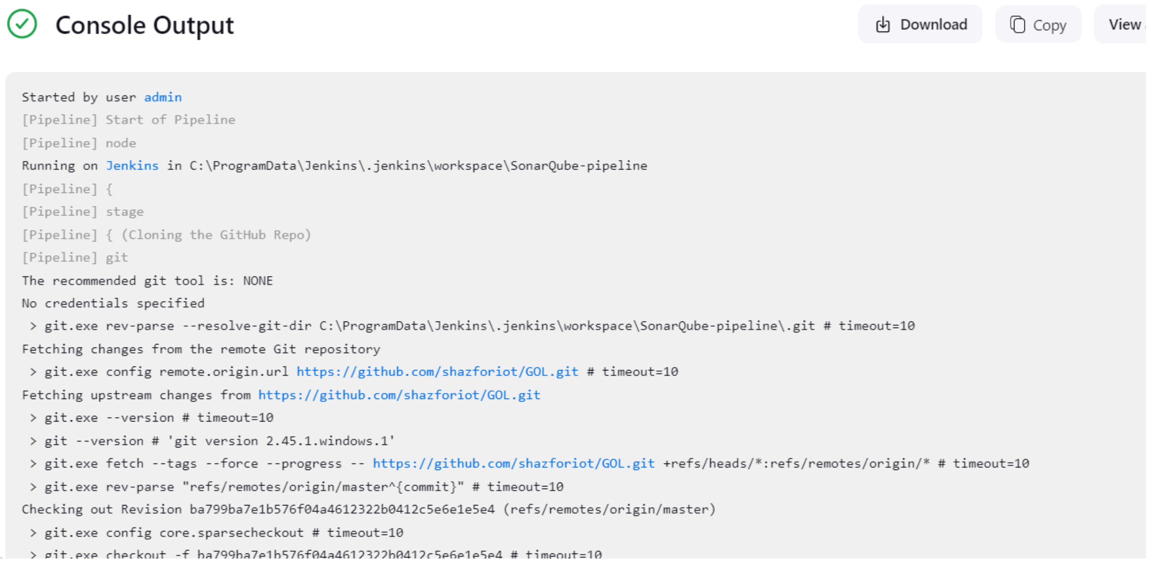
Definition

Pipeline script

Script

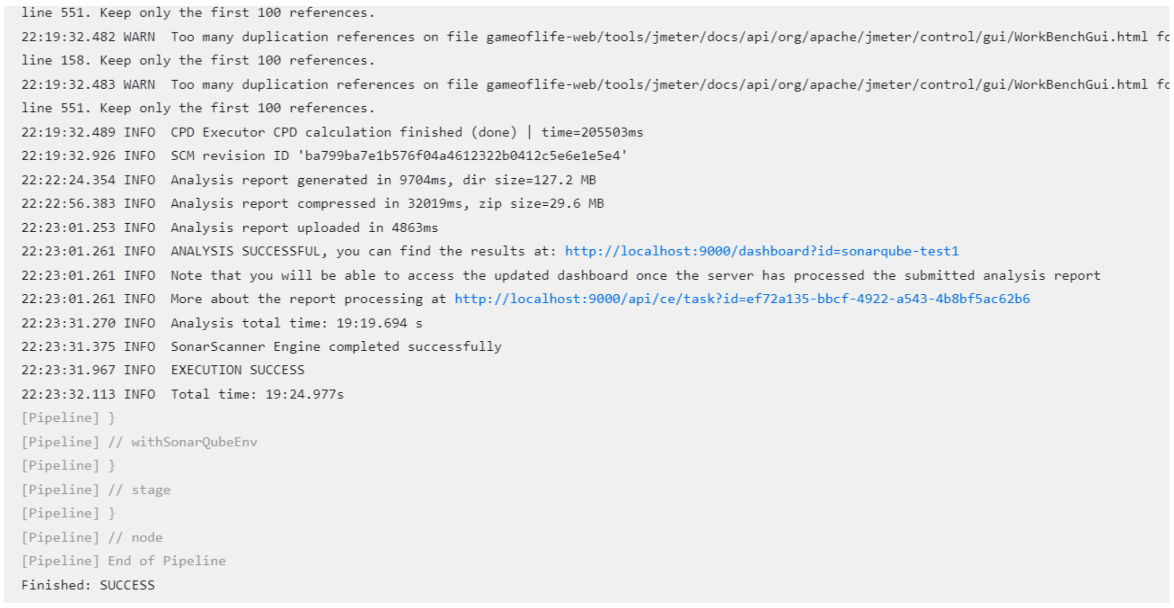
```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }
    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            bat """
           C:\\Users\\rugved\\Downloads\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat ^
            -Dsonar.login=admin ^
            -Dsonar.password=rujutamedhi@04 ^
            -Dsonar.projectKey=sonarqube-test1 ^
            -Dsonar.exclusions=vendor/*,resources/,/.java ^
            -Dsonar.host.url=http://localhost:9000/
            """
        }
    }
}
```

4. Build the project and check the console output



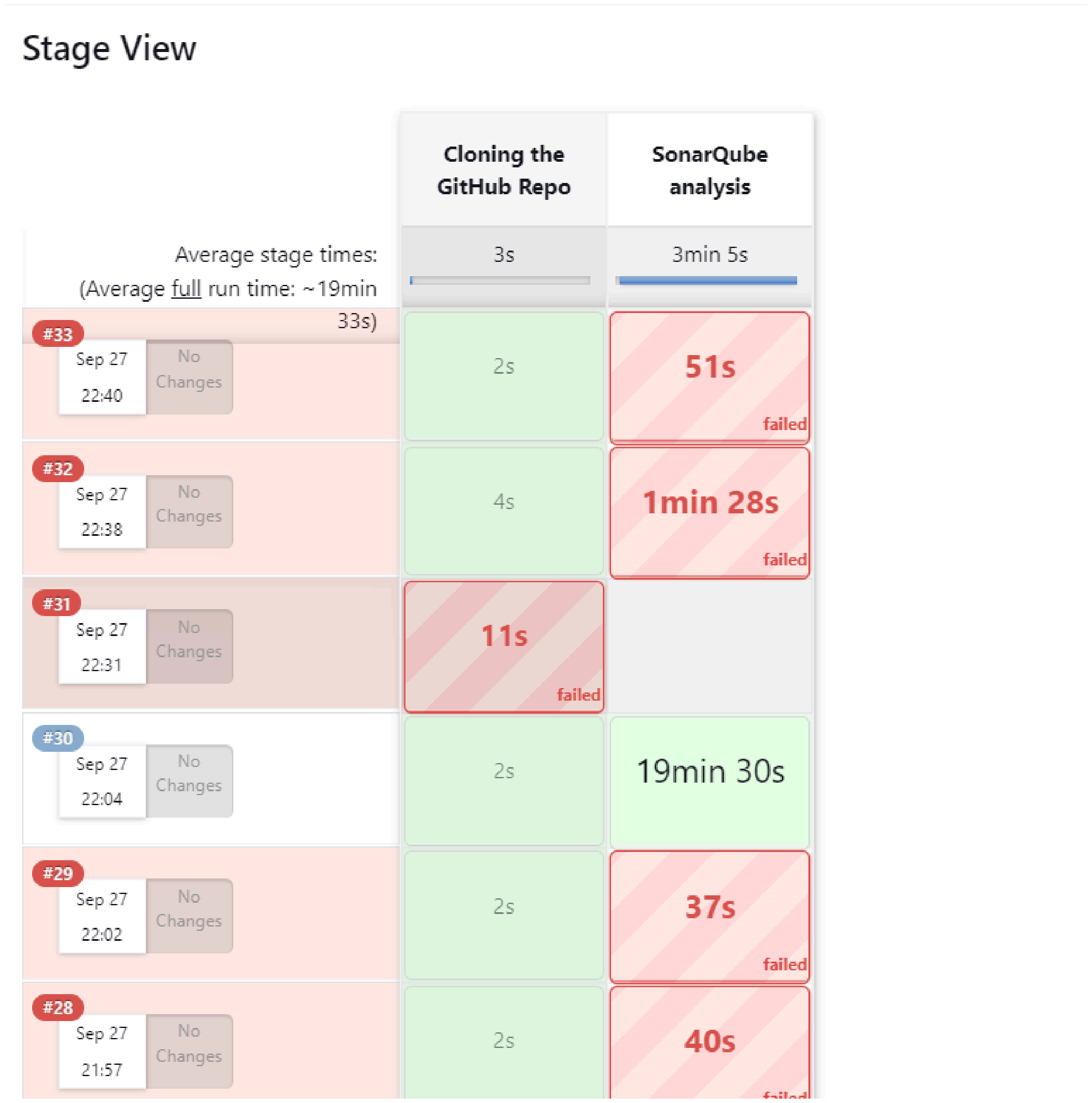
Console Output

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\SonarQube-pipeline
[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-pipeline\.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 2.45.1.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
```



```
line 551. Keep only the first 100 references.
22:19:32.482 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/control/gui/WorkBenchGui.html fo
line 158. Keep only the first 100 references.
22:19:32.483 WARN  Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/control/gui/WorkBenchGui.html fo
line 551. Keep only the first 100 references.
22:19:32.489 INFO  CPD Executor CPD calculation finished (done) | time=205503ms
22:19:32.926 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
22:22:24.354 INFO  Analysis report generated in 9704ms, dir size=127.2 MB
22:22:56.383 INFO  Analysis report compressed in 32019ms, zip size=29.6 MB
22:23:01.253 INFO  Analysis report uploaded in 4863ms
22:23:01.261 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test1
22:23:01.261 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
22:23:01.261 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=ef72a135-bbcf-4922-a543-4b8bf5ac62b6
22:23:31.270 INFO  Analysis total time: 19:19.694 s
22:23:31.375 INFO  SonarScanner Engine completed successfully
22:23:31.967 INFO  EXECUTION SUCCESS
22:23:32.113 INFO  Total time: 19:24.977s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

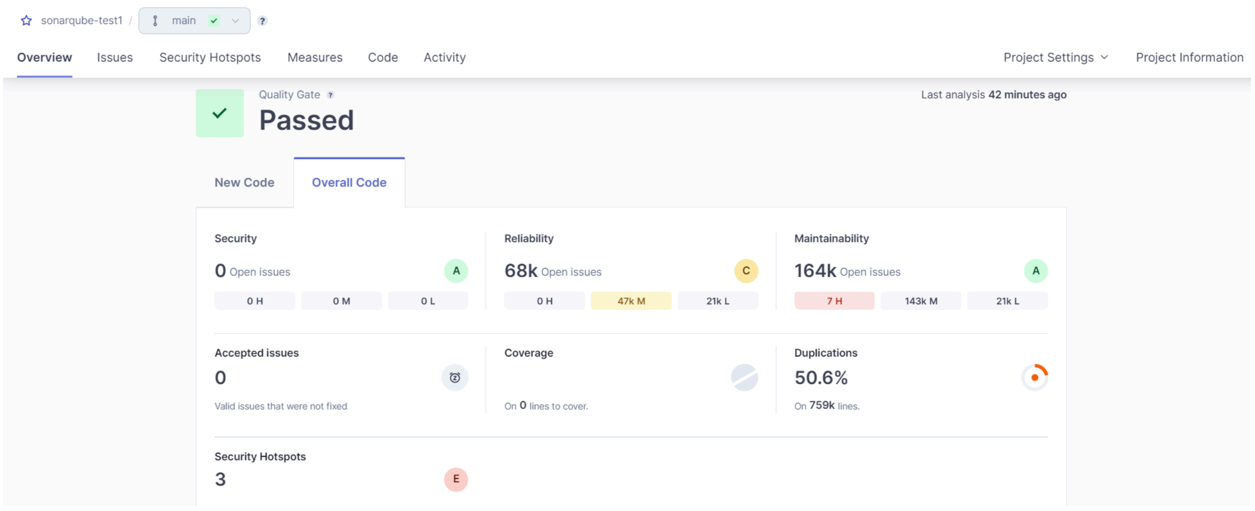
5. After that, check the project in SonarQube



Stage View

	Cloning the GitHub Repo	SonarQube analysis
Average stage times: (Average full run time: ~19min 33s)	3s	3min 5s
#33 Sep 27 22:40 No Changes	2s	51s failed
#32 Sep 27 22:38 No Changes	4s	1min 28s failed
#31 Sep 27 22:31 No Changes	11s failed	
#30 Sep 27 22:04 No Changes	2s	19min 30s
#29 Sep 27 22:02 No Changes	2s	37s failed
#28 Sep 27 21:57 No Changes	2s	40s

6. After that, check the project in SonarQube


sonarqube-test1 / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Quality Gate ?
Passed
Last analysis 42 minutes ago

New Code | Overall Code

Security
0 Open issues A
0 H 0 M 0 L

Reliability
68k Open issues C
0 H 47k M 21k L

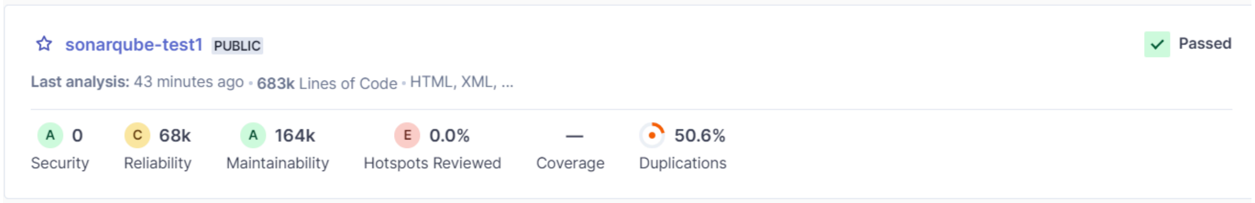
Maintainability
164k Open issues A
7 H 143k M 21k L

Accepted issues
0
Valid issues that were not fixed

Coverage
On 0 lines to cover.

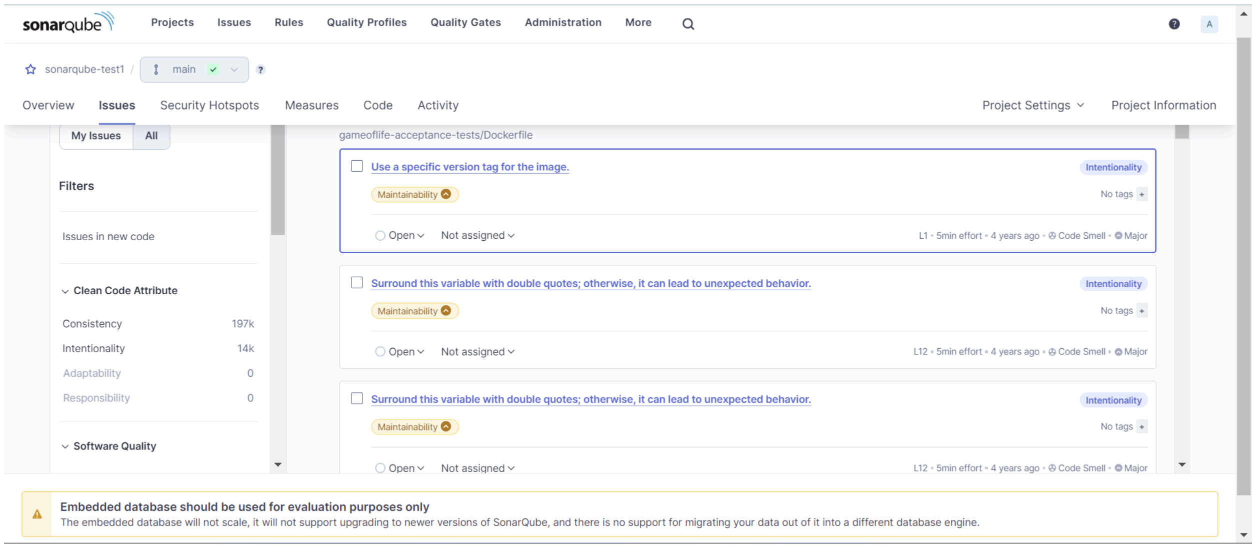
Duplications
50.6%
On 759k lines.

Security Hotspots
3 E


sonarqube-test1 PUBLIC — Passed

Last analysis: 43 minutes 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


sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

sonarqube-test1 / main ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

My Issues | All

Filters

Issues in new code

Clean Code Attribute

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Software Quality

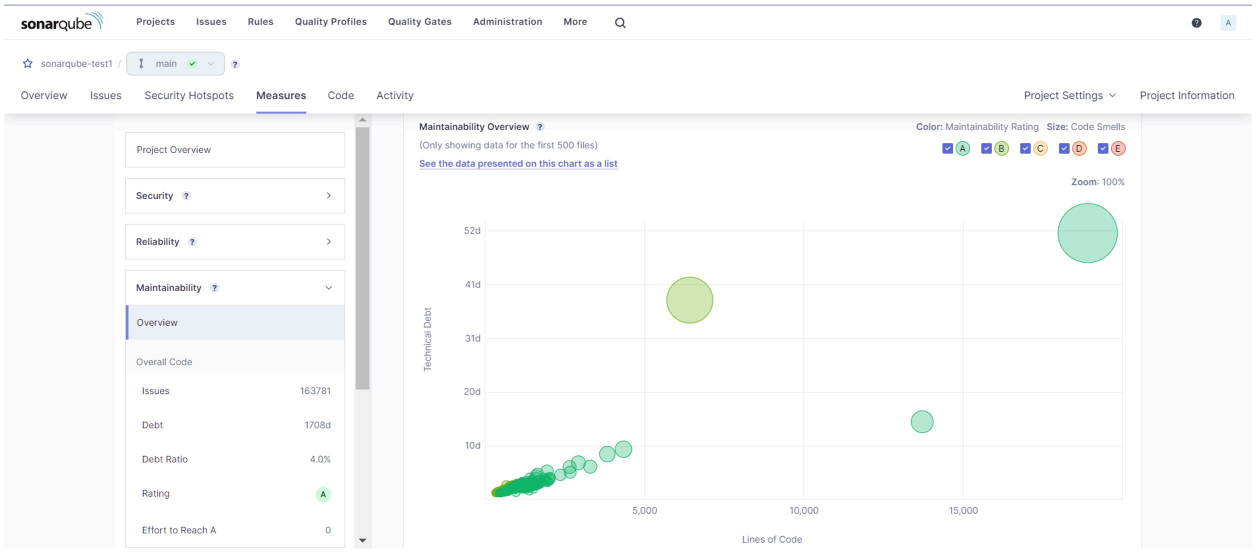
gameoflife-acceptance-tests/Dockerfile

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 +
Open Not 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.



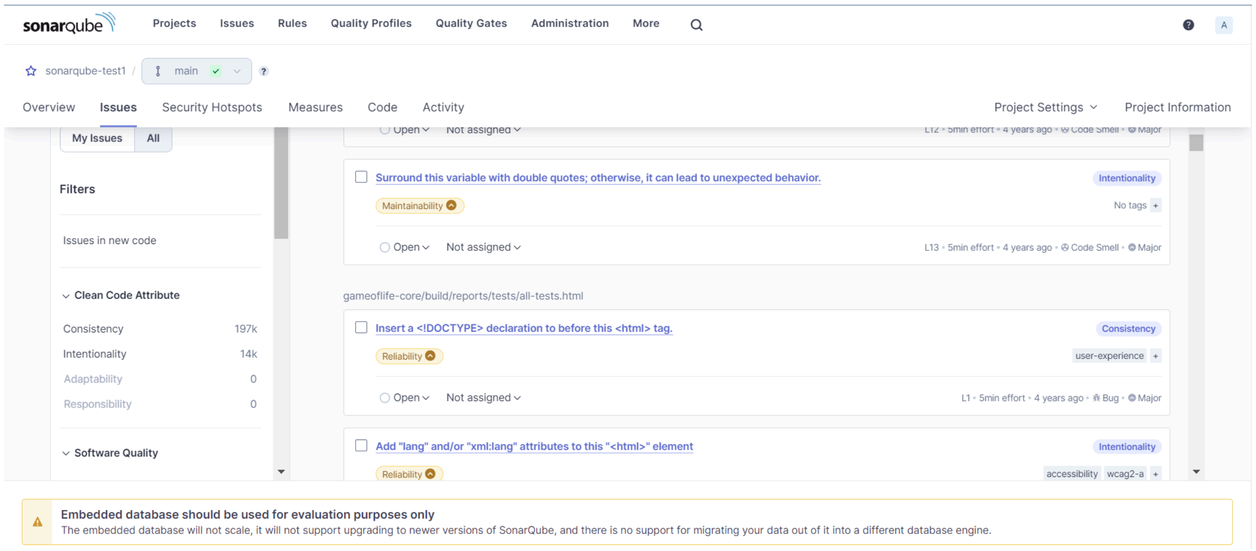
Overall Code	
Issues	163781
Debt	1708d
Debt Ratio	4.0%
Rating	A
Effort to Reach A	0

Maintainability Overview

Color: Maintainability Rating **Size:** Code Smells

(Only showing data for the first 500 files)

See the data presented on this chart as a list



Clean Code Attribute

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Software Quality

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 +

Open Not assigned L13 • 5min effort • 4 years ago • Code Smell • Major

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

Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency

Reliability user-experience +

Open Not assigned L1 • 5min effort • 4 years ago • Bug • Major

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

Reliability accessibility wcag2-a +

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.