Name: Shivpratik Hande **Class/Rolno: D15C-14**

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

- ResourceEfficiency
  : 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. DockerinstalledtorunSonarQube.
3. SonarQubeinstalledviaDocker

1. Set Up Jenkins

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

- Install the necessary plugins:
- SonarQube Scanner Plugin

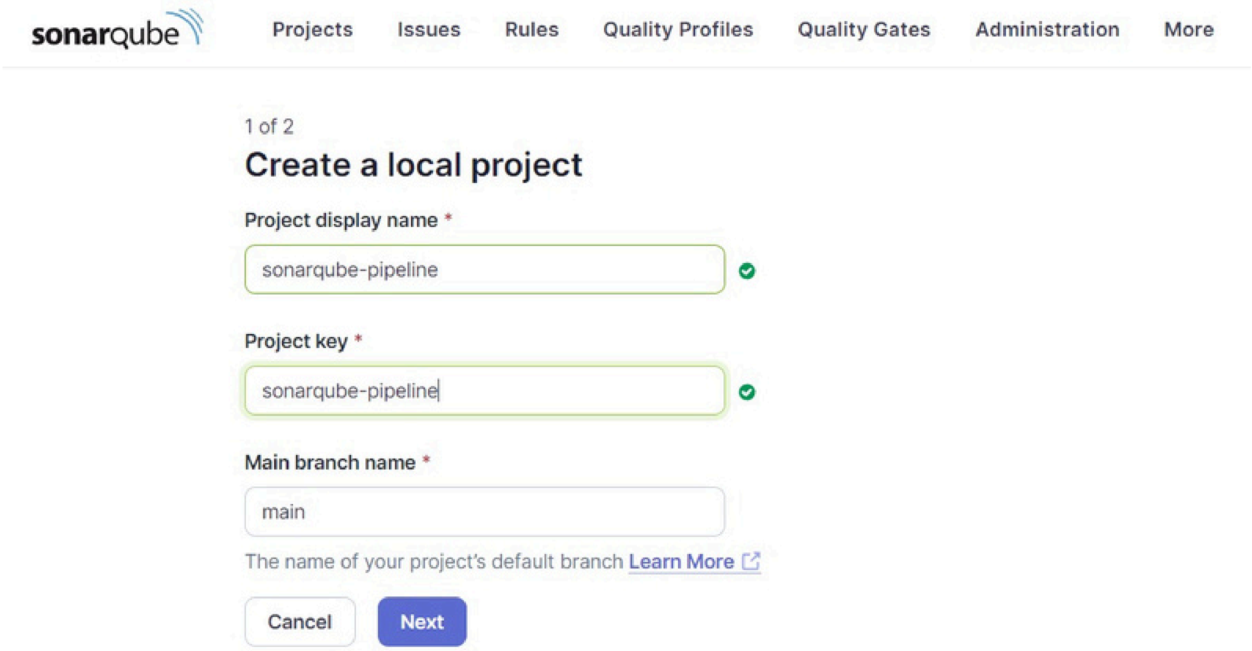
2. Run SonarQube in Docker

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.



1 of 2

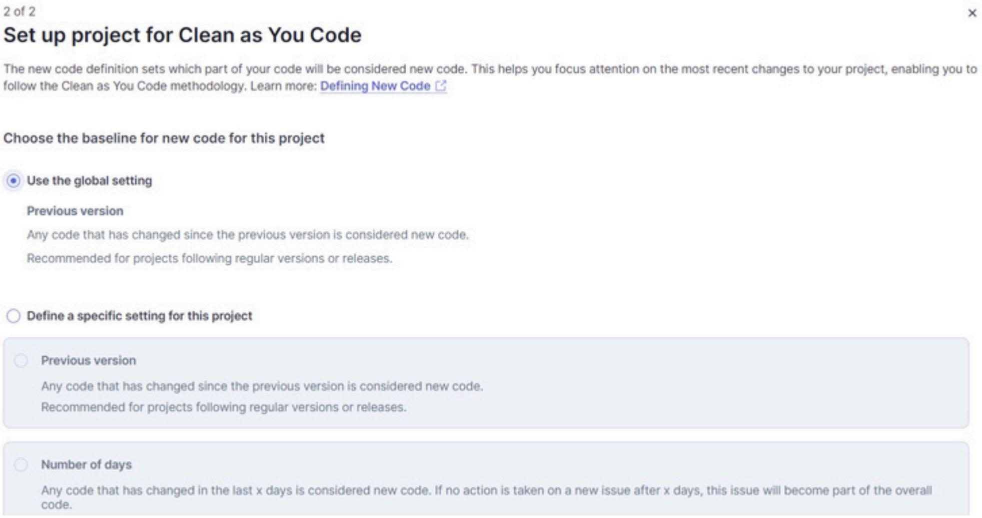
Create a local project

Project display name *
sonarqube-pipeline

Project key *
sonarqube-pipeline

Main branch name *
main

The name of your project's default branch Learn More



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

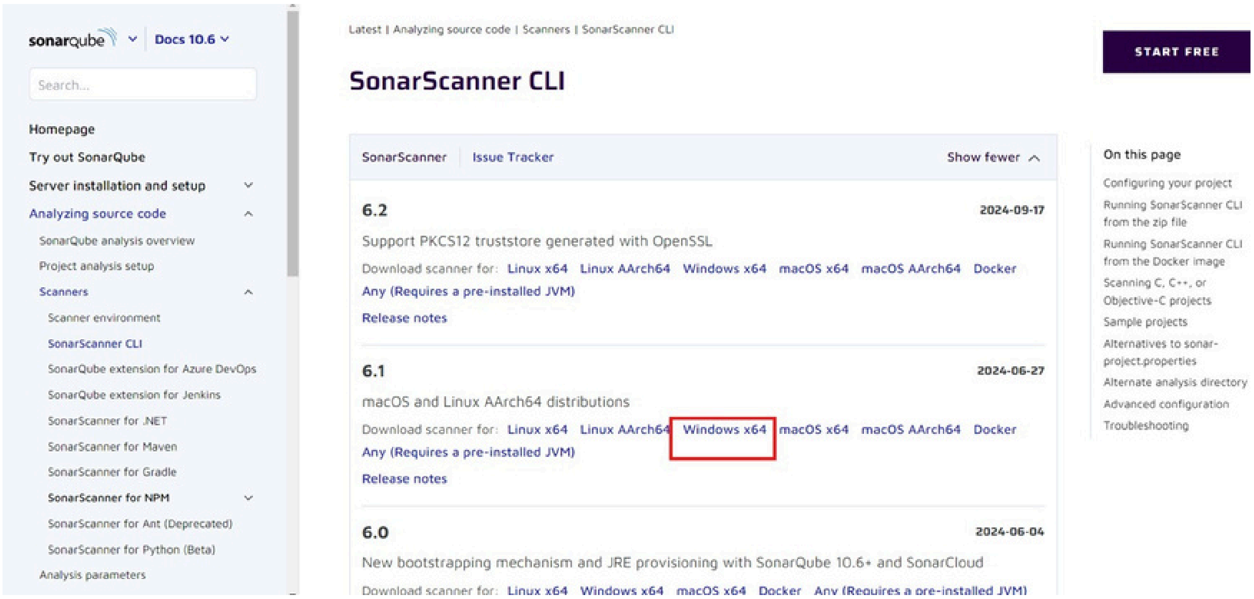
Any code that has changed since the previous version is considered new code.

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.

Step 2: Go to download_sonarscanner to download sonar scanner



sonarqube
Docs 10.6
Search...
Homepage
Try out SonarQube
Server installation and setup
Analyzing source code
SonarQube analysis overview
Project analysis setup
Scanners
Scanner environment
SonarScanner CLI
SonarQube extension for Azure DevOps
SonarQube extension for Jenkins
SonarScanner for .NET
SonarScanner for Maven
SonarScanner for Gradle
SonarScanner for NPM
SonarScanner for Ant (Deprecated)
SonarScanner for Python (Beta)
Analysis parameters
START FREE
On this page
Configuring your project
Running SonarScanner CLI from the zip file
Running SonarScanner CLI from the Docker image
Scanning C, C++, or Objective-C projects
Sample projects
Alternatives to sonar-project.properties
Alternate analysis directory
Advanced configuration
Troubleshooting

Latest | Analyzing source code | Scanners | SonarScanner CLI

SonarScanner CLI

SonarScanner | Issue Tracker — Show fewer

6.2 — 2024-09-17

Support PKCS12 truststore generated with OpenSSL

Download scanner for: Linux x64 Linux AArch64 Windows x64 macOS x64 macOS AArch64 Docker Any (Requires a pre-installed JVM)

Release notes

6.1 — 2024-06-27

macOS and Linux AArch64 distributions

Download scanner for: Linux x64 Linux AArch64 Windows x64 macOS x64 macOS AArch64 Docker Any (Requires a pre-installed JVM)

Release notes

6.0 — 2024-06-04

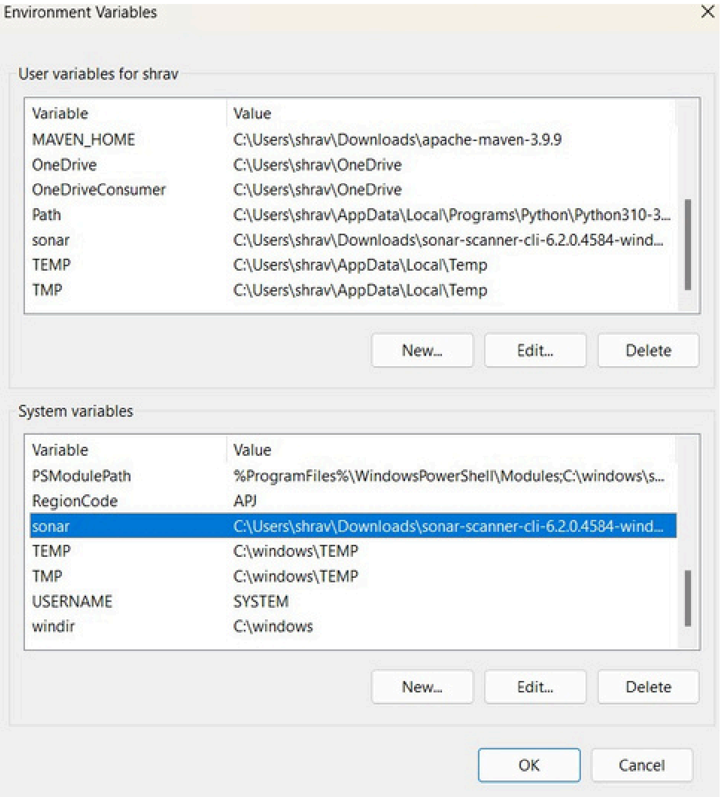
New bootstrapping mechanism and JRE provisioning with SonarQube 10.6+ and SonarCloud

Download scanner for: Linux x64 Windows x64 macOS x64 Docker Any (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.



Environment Variables

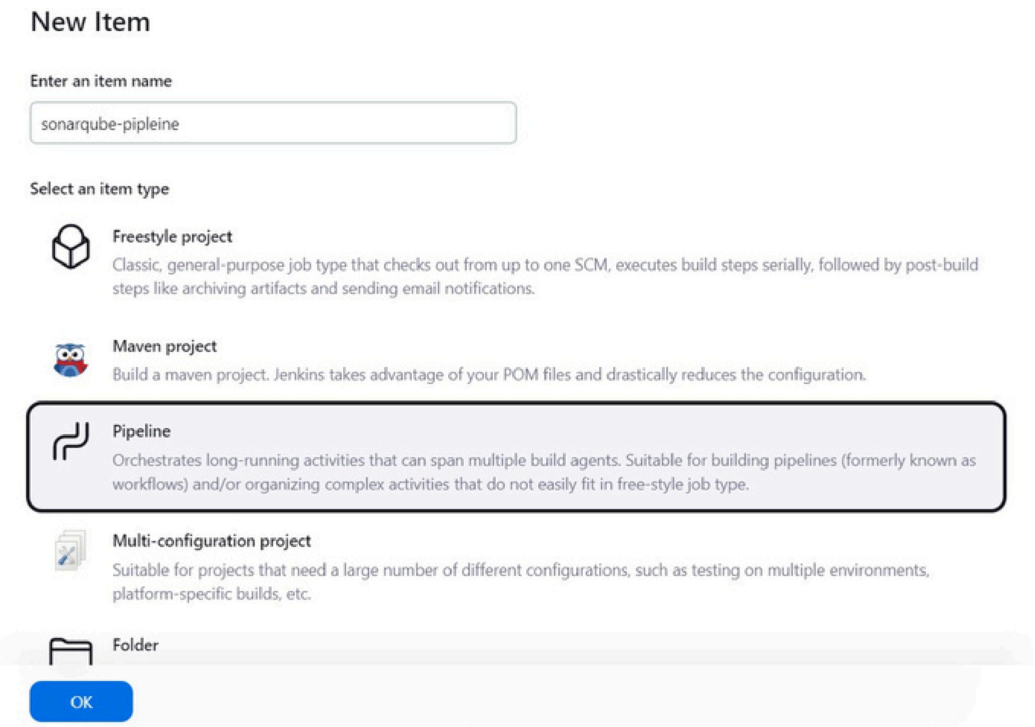
User variables for shrav

Variable	Value
MAVEN_HOME	C:\Users\shrav\Downloads\apache-maven-3.9.9
OneDrive	C:\Users\shrav\OneDrive
OneDriveConsumer	C:\Users\shrav\OneDrive
Path	C:\Users\shrav\AppData\Local\Programs\Python\Python310-3...
sonar	C:\Users\shrav\Downloads\sonar-scanner-cli-6.2.0.4584-wind...
TEMP	C:\Users\shrav\AppData\Local\Temp
TMP	C:\Users\shrav\AppData\Local\Temp

System variables

Variable	Value
PSModulePath	%ProgramFiles%\WindowsPowerShell\Modules;C:\windows\s...
RegionCode	APJ
sonar	C:\Users\shrav\Downloads\sonar-scanner-cli-6.2.0.4584-wind...
TEMP	C:\windows\TEMP
TMP	C:\windows\TEMP
USERNAME	SYSTEM
windir	C:\windows

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



New Item

Enter an item name

sonarqube-pipleine

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.

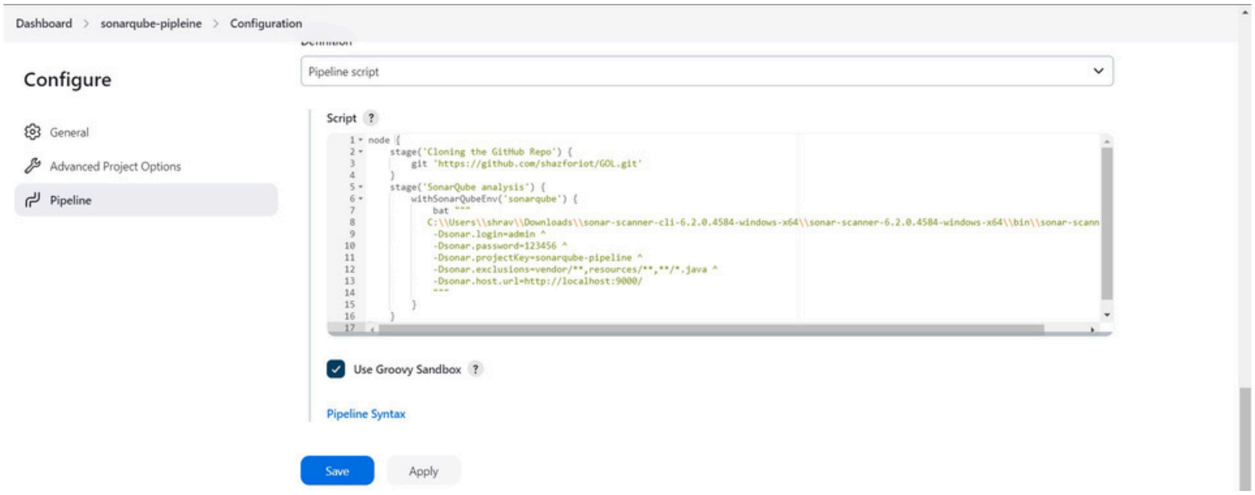
Folder

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/
         """

      }
   }
}
```

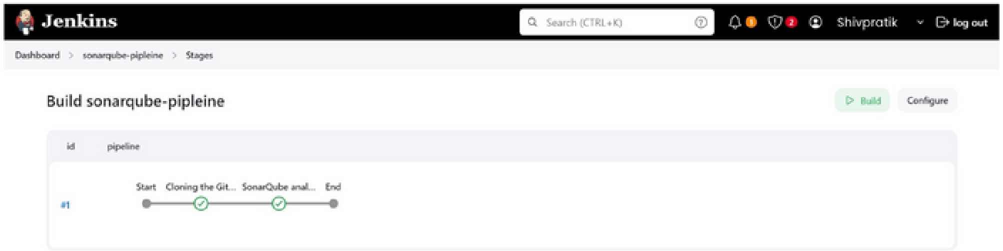


Definition: Pipeline script

```
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-scann
            -Dsonar.login=admin ^
            -Dsonar.password=123456 ^
            -Dsonar.projectKey=sonarqube-pipeline ^
            -Dsonar.exclusions=vendor/**,resources/**,**/*.java ^
            -Dsonar.host.url=http://localhost:9000/
            """
        }
    }
```

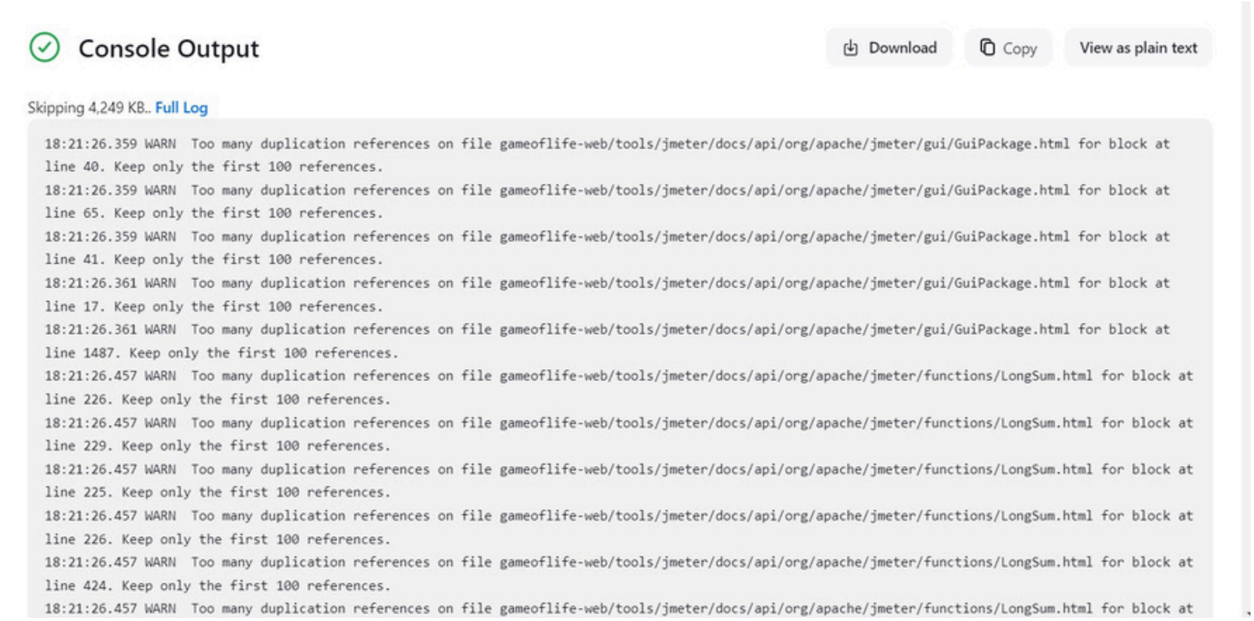
☑ Use Groovy Sandbox ?

Step 4: Save the pipeline and build it.



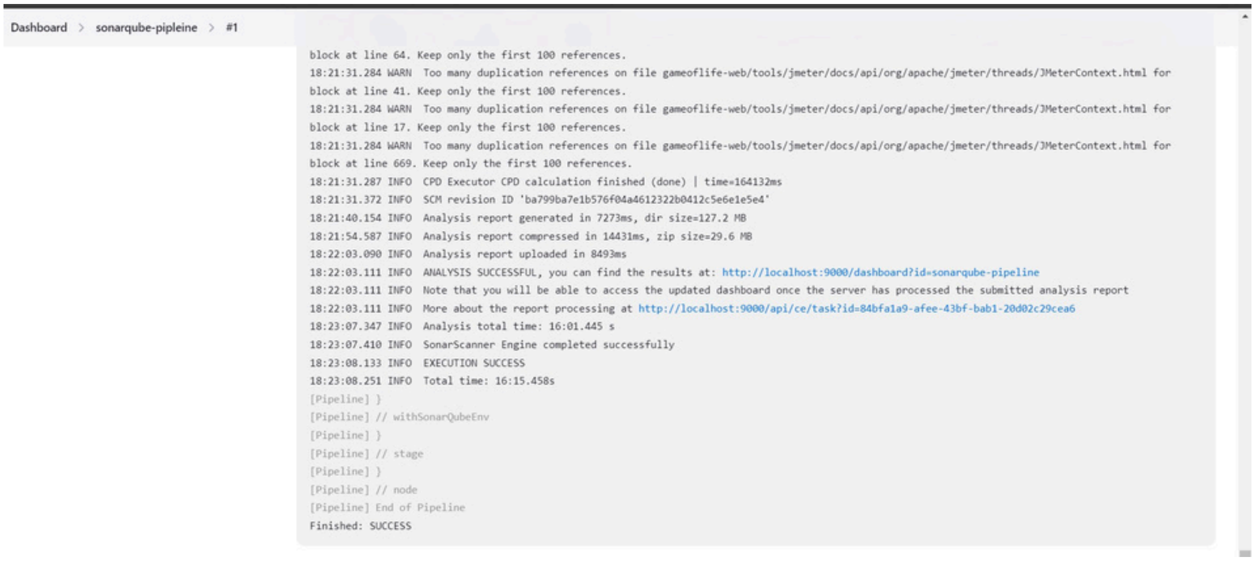
Build sonarqube-pipleine

Console output:



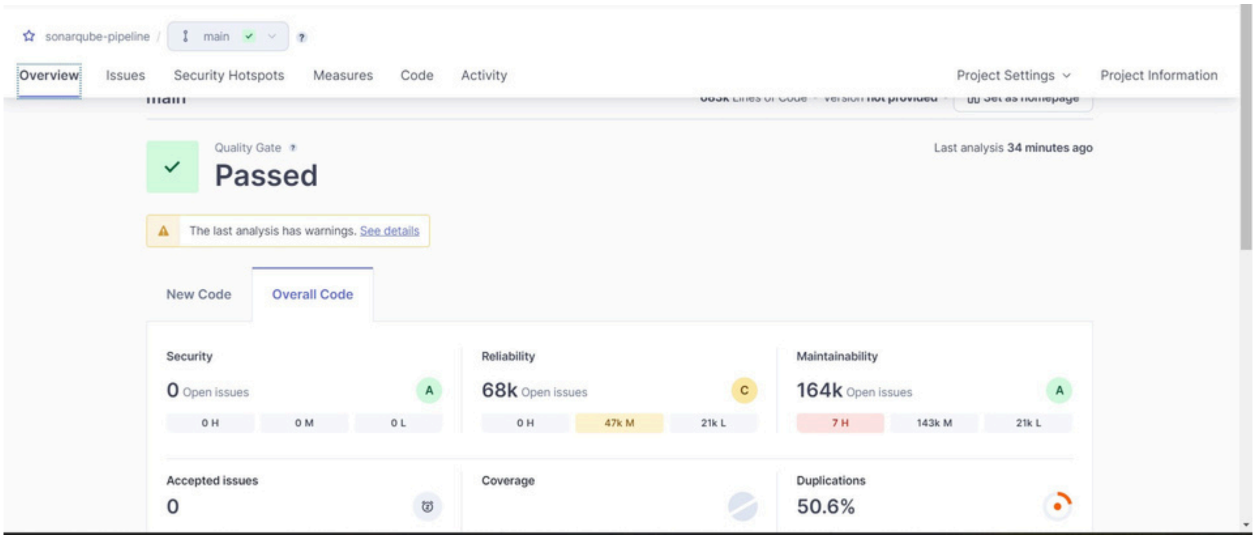
Skipping 4,249 KB.. Full Log

```
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
```



```
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 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
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-bab1-20d02c29cea6
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



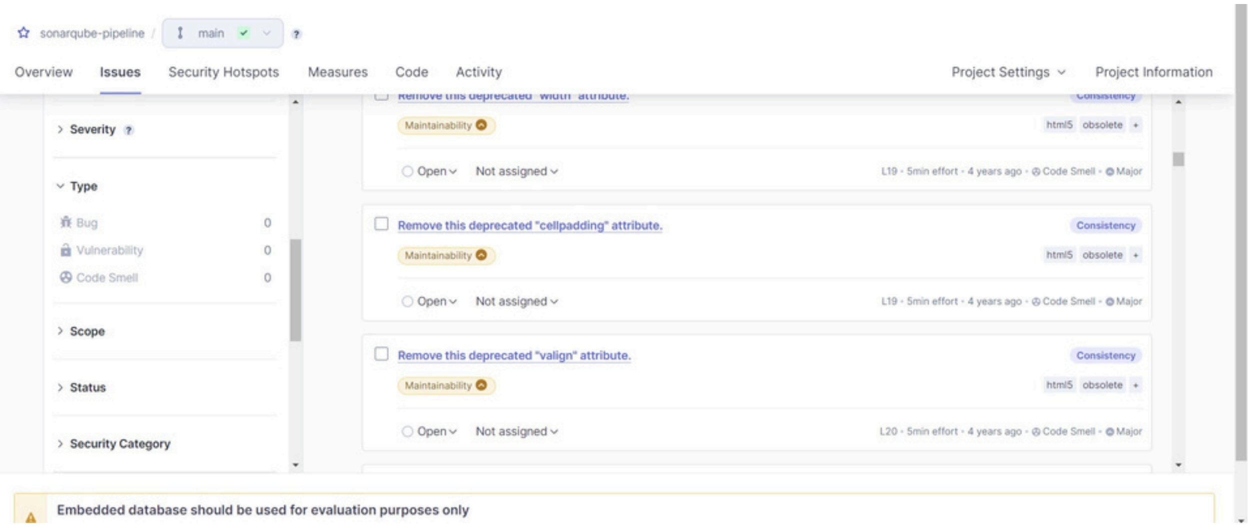
Quality Gate
Passed

Last analysis 34 minutes ago

The last analysis has warnings. See details

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 0	Coverage	Duplications 50.6%

Under different tabs, check all different issues with the code



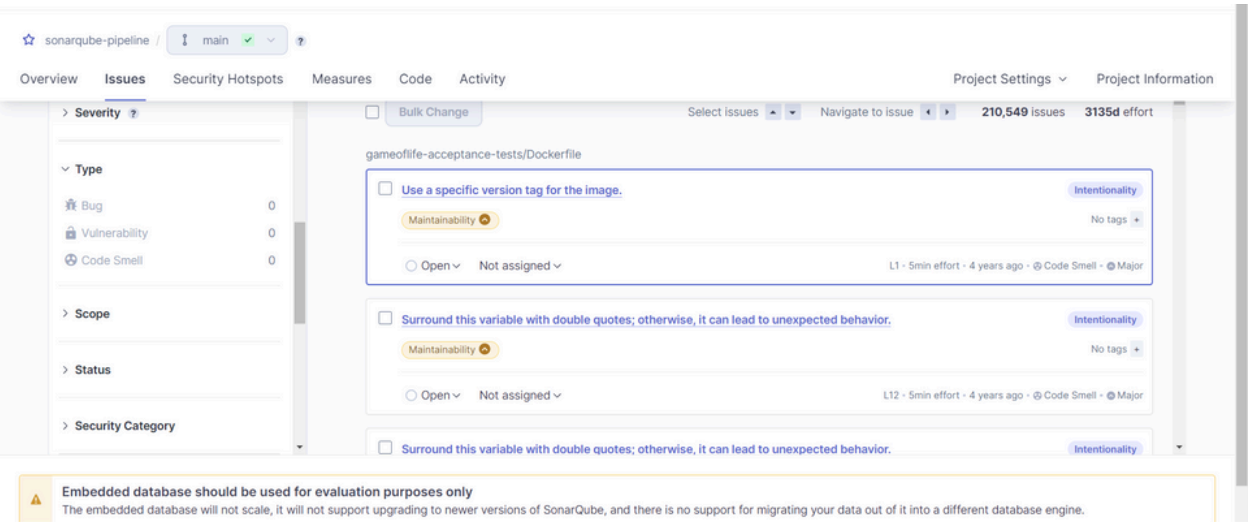
Type	
Bug	0
Vulnerability	0
Code Smell	0

Remove this deprecated "width" attribute. — Consistency — Maintainability — html5 obsolete — Open · Not assigned — L19 · 5min effort · 4 years ago · Code Smell · Major

☐ Remove this deprecated "cellpadding" attribute. — Consistency — Maintainability — html5 obsolete — Open · Not assigned — L19 · 5min effort · 4 years ago · Code Smell · Major

☐ Remove this deprecated "valign" attribute. — Consistency — Maintainability — html5 obsolete — Open · Not assigned — L20 · 5min effort · 4 years ago · Code Smell · Major

Embedded database should be used for evaluation purposes only



Bug	0
Vulnerability	0
Code Smell	0

210,549 issues 3135d effort

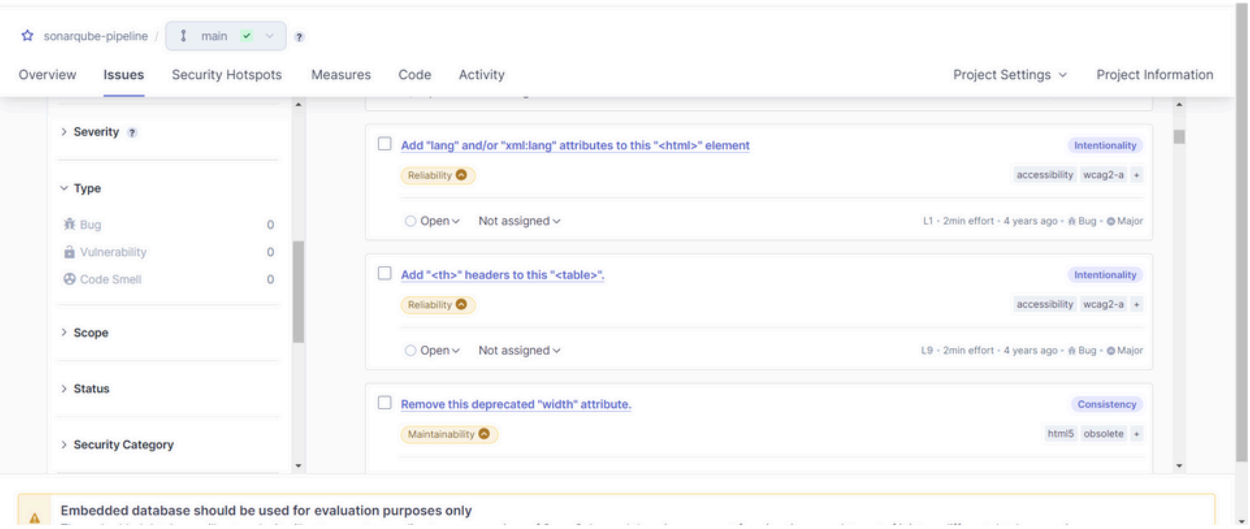
gameoflife-acceptance-tests/Dockerfile

□ 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

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.



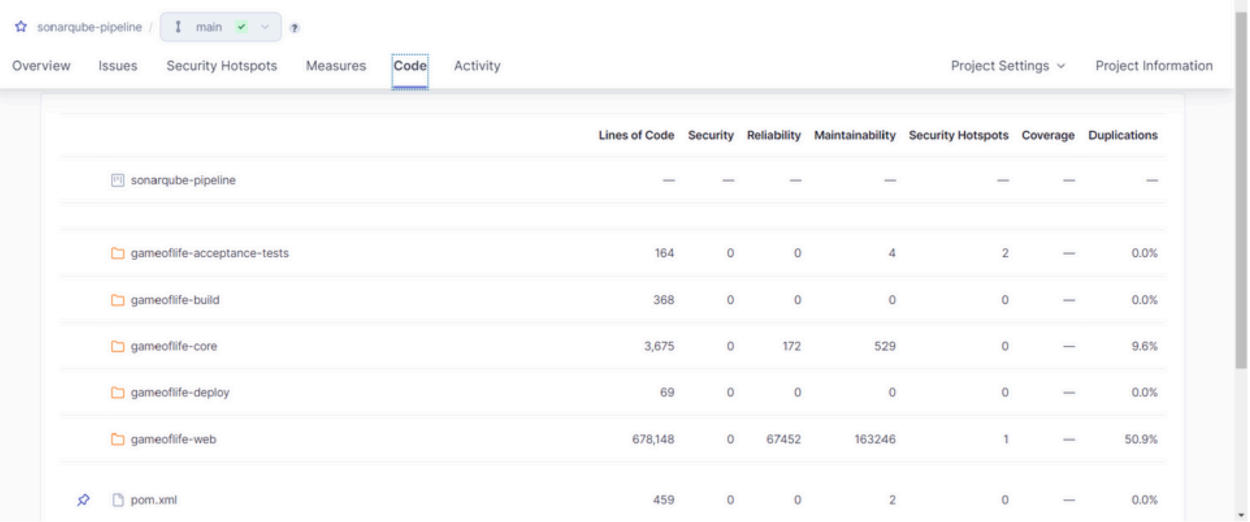
Bug	0
Vulnerability	0
Code Smell	0

□ 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

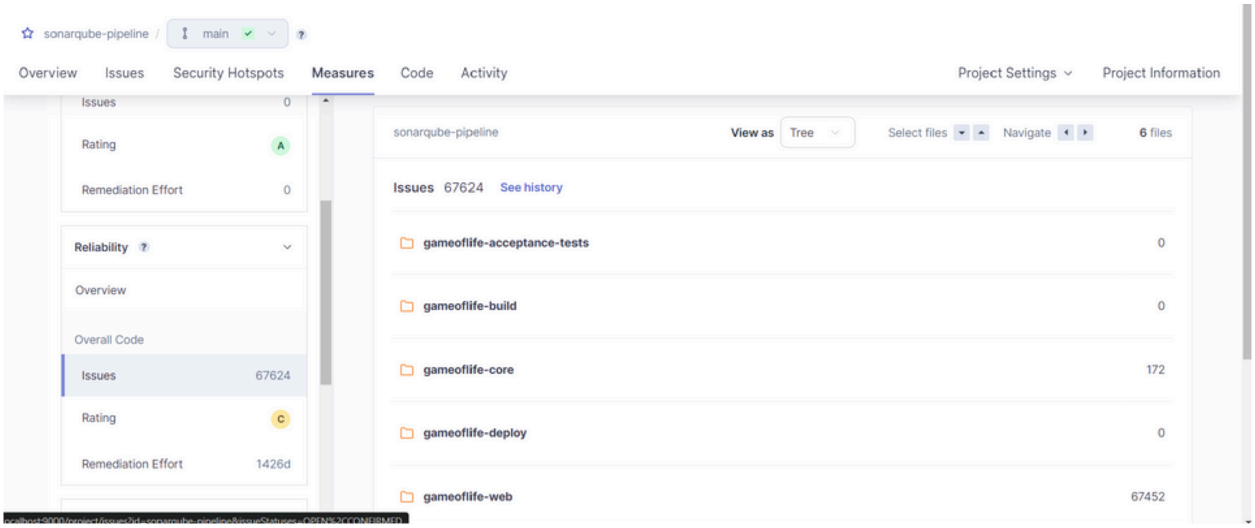
□ Add "<th>" headers to this "<table>". Intentionality
Reliability accessibility wcag2-a +
○ Open Not assigned L9 · 2min effort · 4 years ago · Bug · Major

□ Remove this deprecated "width" attribute. Consistency
Maintainability html5 obsolete +

Embedded database should be used for evaluation purposes only



	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%



Issues	0
Rating	A
Remediation Effort	0

Reliability ?

Overview

Overall Code

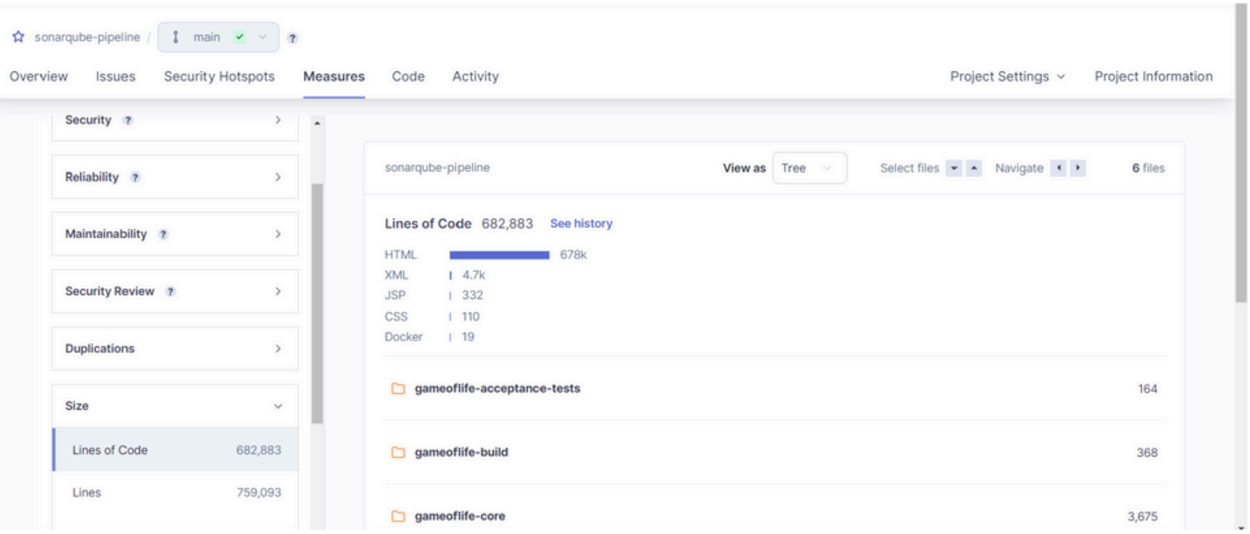
Issues	67624
Rating	C
Remediation Effort	1426d

sonarqube-pipeline — 6 files

Issues 67624 See history

gameoflife-acceptance-tests	0
gameoflife-build	0
gameoflife-core	172
gameoflife-deploy	0
gameoflife-web	67452

localhost:9000/project/issues?id=sonarqube-pipeline&issueStatuses=OPEN%2CCONFIRMED



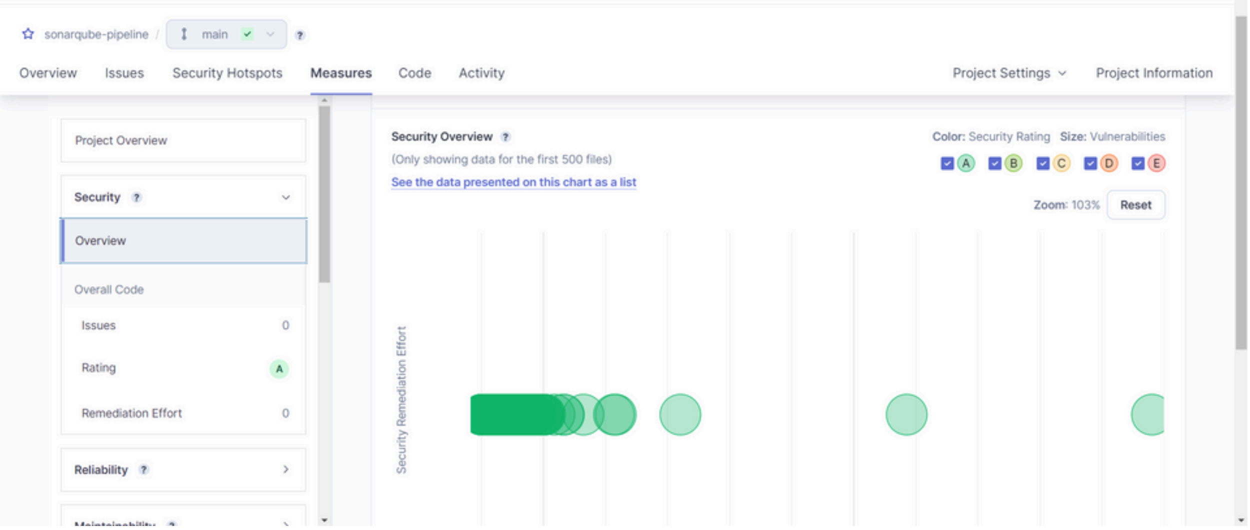
Lines of Code	682,883
Lines	759,093

sonarqube-pipeline — 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



Overall Code

Issues	0
Rating	A
Remediation Effort	0

Security Overview ?

(Only showing data for the first 500 files)

See the data presented on this chart as a list

Conclusion :

Creating a CI/CD pipeline in Jenkins integrated with SonarQube or GitLab for static analysis is a powerful strategy for enhancing the quality and security of your applications. By implementing this pipeline for a sample web application in Java or Python, you can automate the detection of bugs, code smells, and security vulnerabilities.