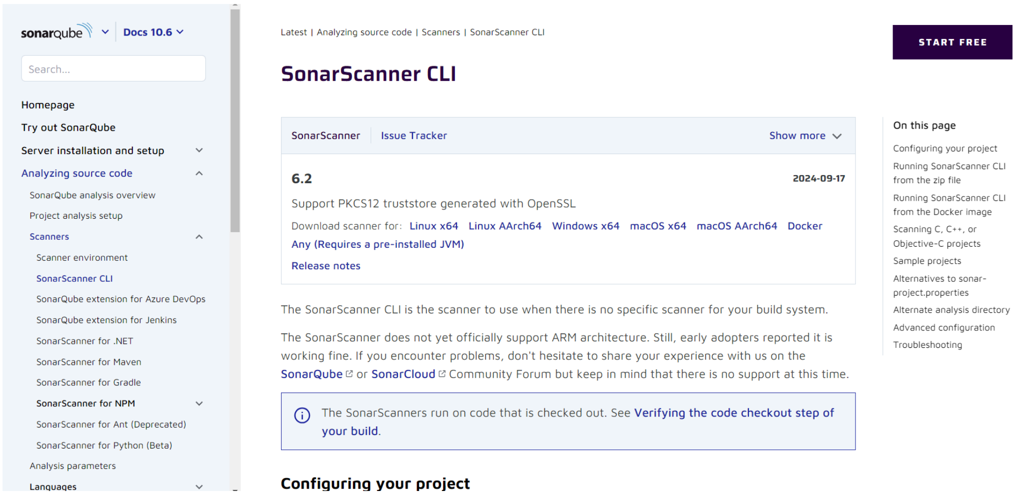
Name:Atharv Nikam **Div:D15C** **Roll No:36**

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.

Prerequisites:

1. **Download Sonar Scanner:**
   Access the SonarQube documentation and download the SonarQube scanner CLI from this link:

https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/


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
Languages
Latest | Analyzing source code | Scanners | SonarScanner CLI
START FREE
SonarScanner
Issue Tracker
Show more
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

SonarScanner CLI

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

The SonarScanner CLI is the scanner to use when there is no specific scanner for your build system.

The SonarScanner does not yet officially support ARM architecture. Still, early adopters reported it is working fine. If you encounter problems, don't hesitate to share your experience with us on the SonarQube or SonarCloud Community Forum but keep in mind that there is no support at this time.

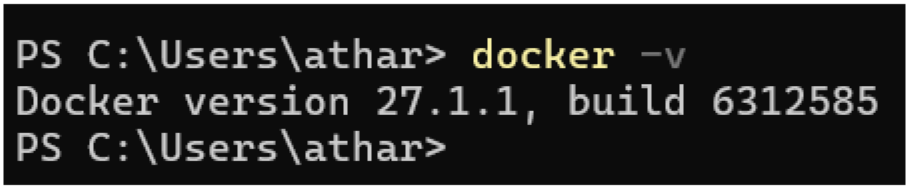
The SonarScanners run on code that is checked out. See Verifying the code checkout step of your build.

Configuring your project

2. After downloading, extract the zip file into a designated folder.

Install Docker:

Run the following command to verify Docker is installed:

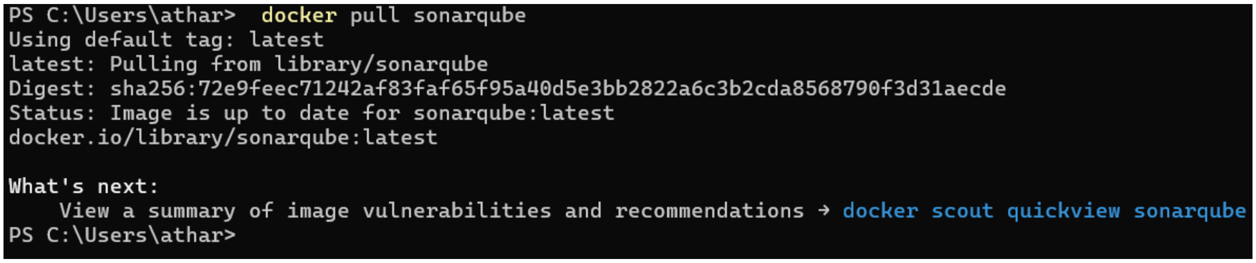


```
PS C:\Users\athar> docker -v
Docker version 27.1.1, build 6312585
PS C:\Users\athar>
```

3 .Pull SonarQube Docker Image:
Install the SonarQube image by executing:

Copy code

```
docker pull sonarqube
```



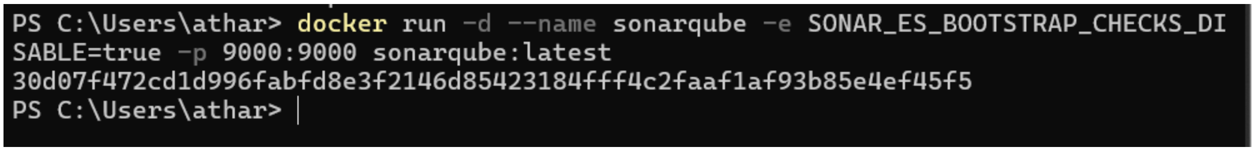
```
PS C:\Users\athar>  docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
PS C:\Users\athar>
```

4. Ensure Jenkins is installed:
Confirm that Jenkins is installed and configured on your system.

Experiment Steps:

Step 1:
Run the SonarQube Docker container by entering the command below:

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



```
PS C:\Users\athar> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DI
SABLE=true -p 9000:9000 sonarqube:latest
30d07f472cd1d996fabfd8e3f2146d85423184fff4c2faaf1af93b85e4ef45f5
PS C:\Users\athar> |
```

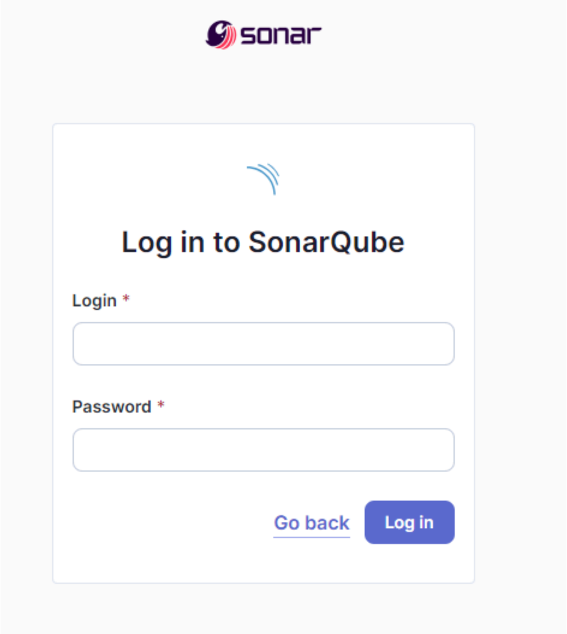
Step 2:

After SonarQube is running, open your browser and go to `http://localhost:9000`.

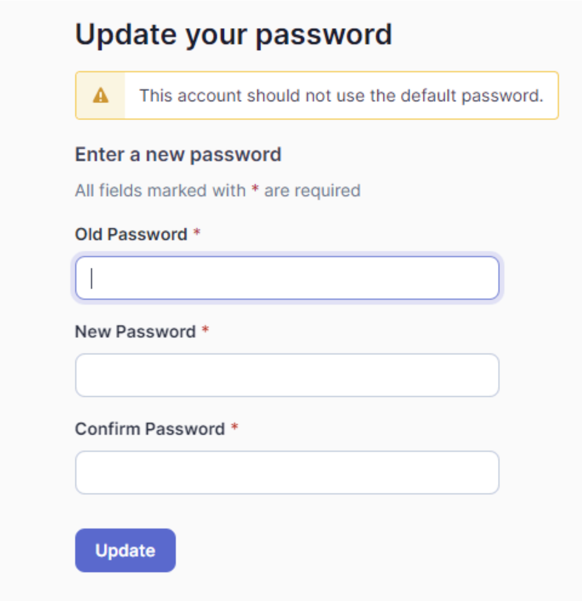
Step 3:
Log in to SonarQube using the default credentials:

```
Username: admin  Password: admin

You will be asked to reset the password after logging in for the first
time. Set a new password and remember it.
```

Name:Atharv Nikam Div:D15C Roll :36



Log in to SonarQube



Update your password

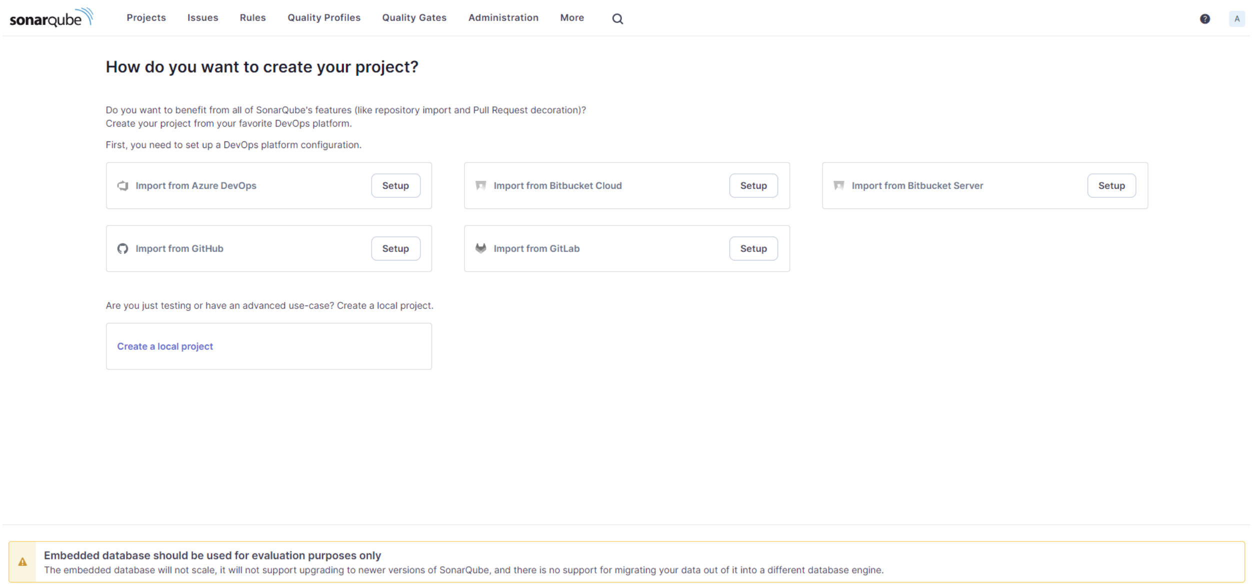
This account should not use the default password.

Enter a new password

All fields marked with * are required

Name:Atharv Nikam Div:D15C Roll :36

Step 4:
On the SonarQube dashboard, click **Create a Local Project**. Provide a project name and a unique project key.



How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.

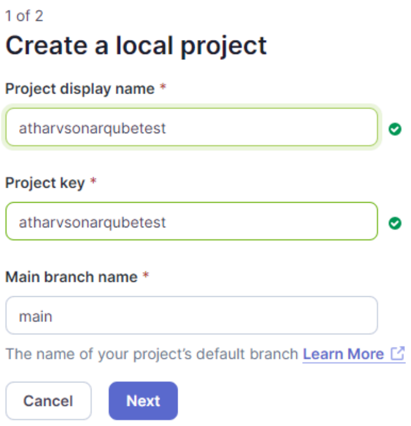
First, you need to set up a DevOps platform configuration.

Import from Azure DevOps Setup
Import from Bitbucket Cloud Setup
Import from Bitbucket Server Setup
Import from GitHub Setup
Import from GitLab Setup

Are you just testing or have an advanced use-case? Create a local project.

Create a local project

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.



1 of 2

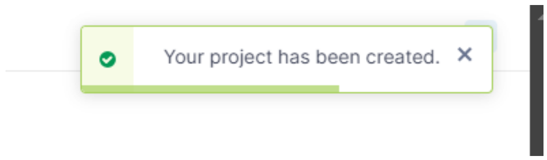
Create a local project

Project display name *
atharvsonarqubetest

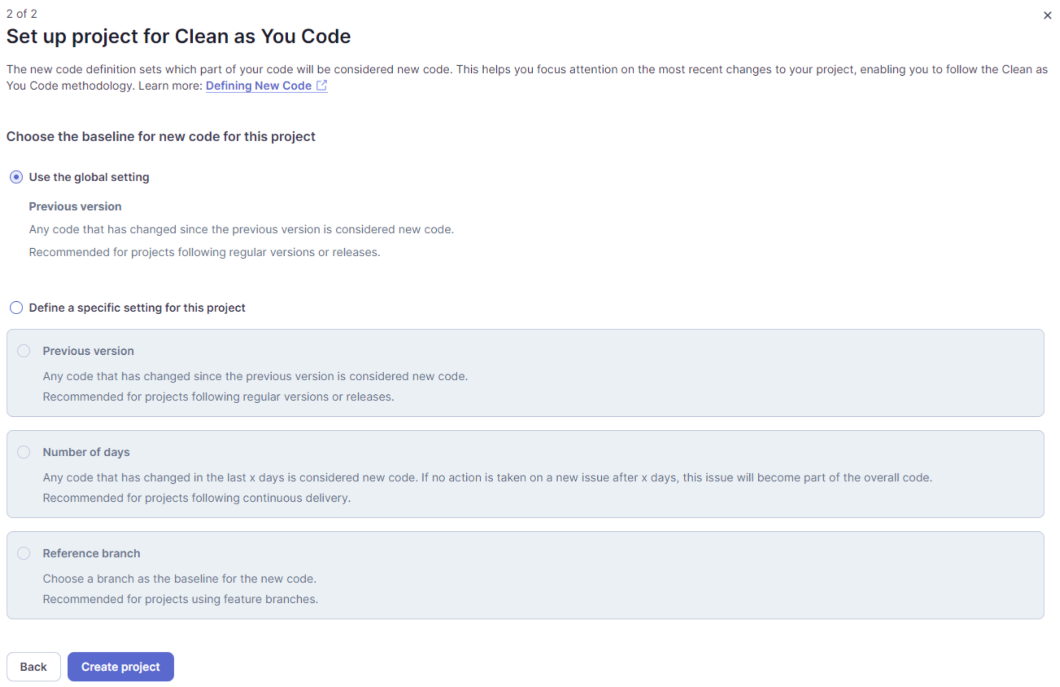
Project key *
atharvsonarqubetest

Main branch name *
main

The name of your project's default branch Learn More



Your project has been created.



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.

Recommended for projects following continuous delivery.

○ Reference branch

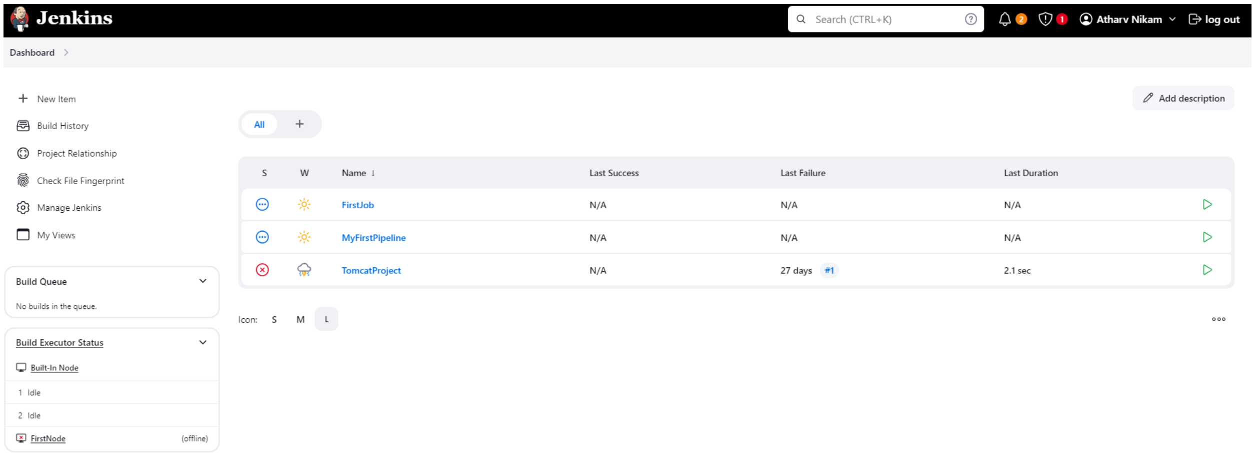
Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

Step 5:

Open Jenkins by navigating to the port on which it is installed:

http://localhost:<port_number>



S	W	Name ↓	Last Success	Last Failure	Last Duration
⋯	☀	FirstJob	N/A	N/A	N/A
⋯	☀	MyFirstPipeline	N/A	N/A	N/A
⊗	☁	TomcatProject	N/A	27 days #1	2.1 sec

Icon: S M L

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

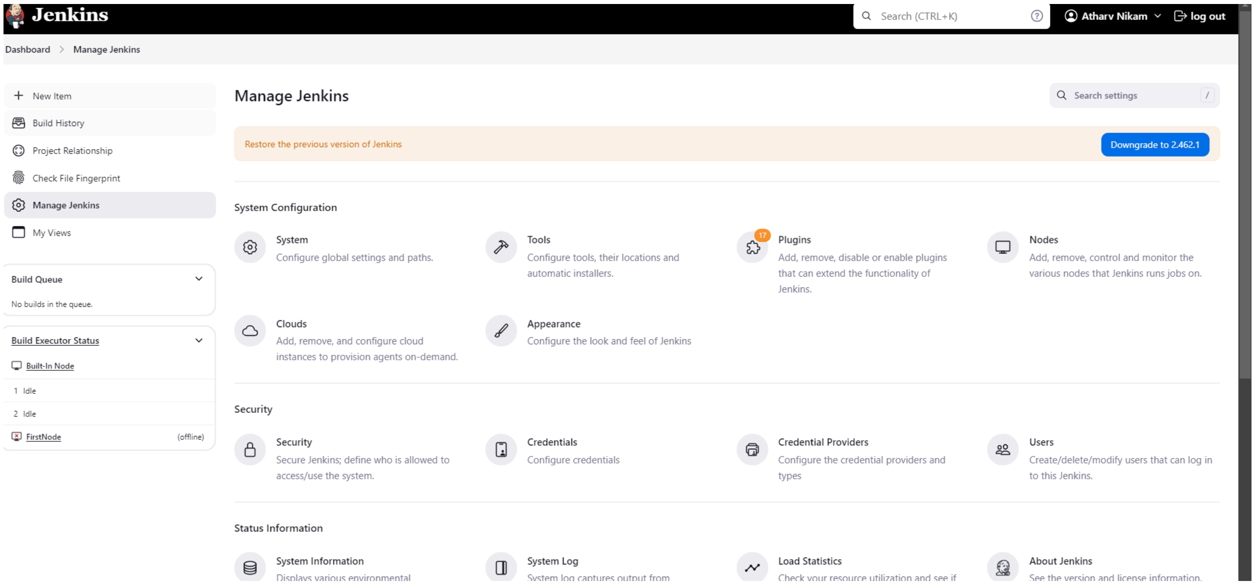
1 Idle

2 Idle

FirstNode (offline)

Name:Atharv Nikam Div:D15C Roll :36

Step 6:
In Jenkins, go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner for Jenkins**. Install the plugin.



Manage Jenkins

Restore the previous version of Jenkins

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds
Add, remove, and configure cloud instances to provision agents on-demand.

Appearance
Configure the look and feel of Jenkins

Security

Security
Secure Jenkins; define who is allowed to access/use the system.

Credentials
Configure credentials

Credential Providers
Configure the credential providers and types

Users
Create/delete/modify users that can log in to this Jenkins.

Status Information

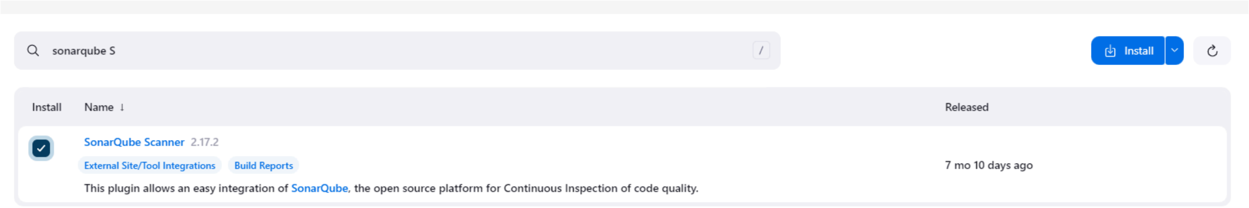
System Information
Displays various environmental

System Log
System log captures output from

Load Statistics
Check your resource utilization and see if

About Jenkins
See the version and license information.

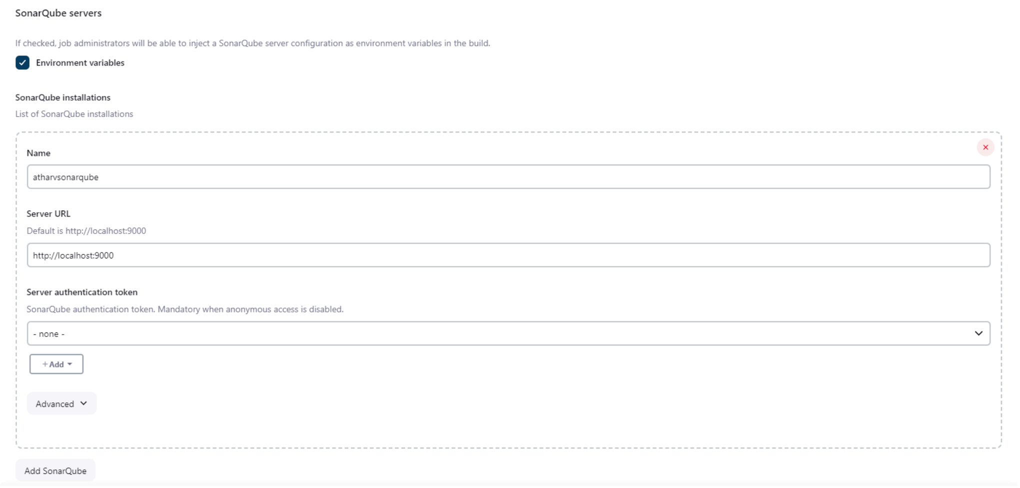
Step 7:
Once installed, head to **Manage Jenkins → System**. Under **SonarQube Servers**, add your SonarQube server, and provide any necessary authentication tokens.



sonarqube S

Install	Name ↓	Released
☑	SonarQube Scanner 2.17.2 External Site/Tool Integrations Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	7 mo 10 days ago

Step 8:
Next, under **Manage Jenkins → Tools**, navigate to **SonarQube Scanner** and configure it to automatically install the latest version.



SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

☑ Environment variables

SonarQube installations

List of SonarQube installations

Name

atharvsonarqube

Server URL

Default is http://localhost:9000

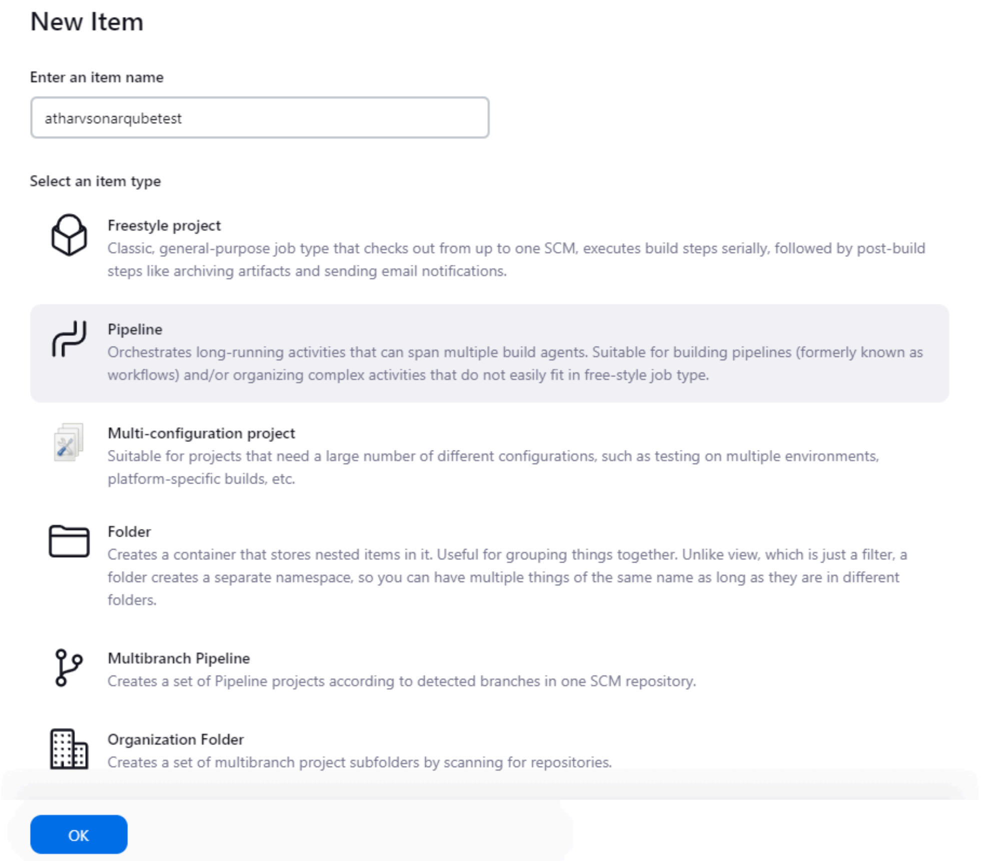
http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

Step 9:
Create a new pipeline item in Jenkins



New Item

Enter an item name

atharvsonarqubetest

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.

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
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

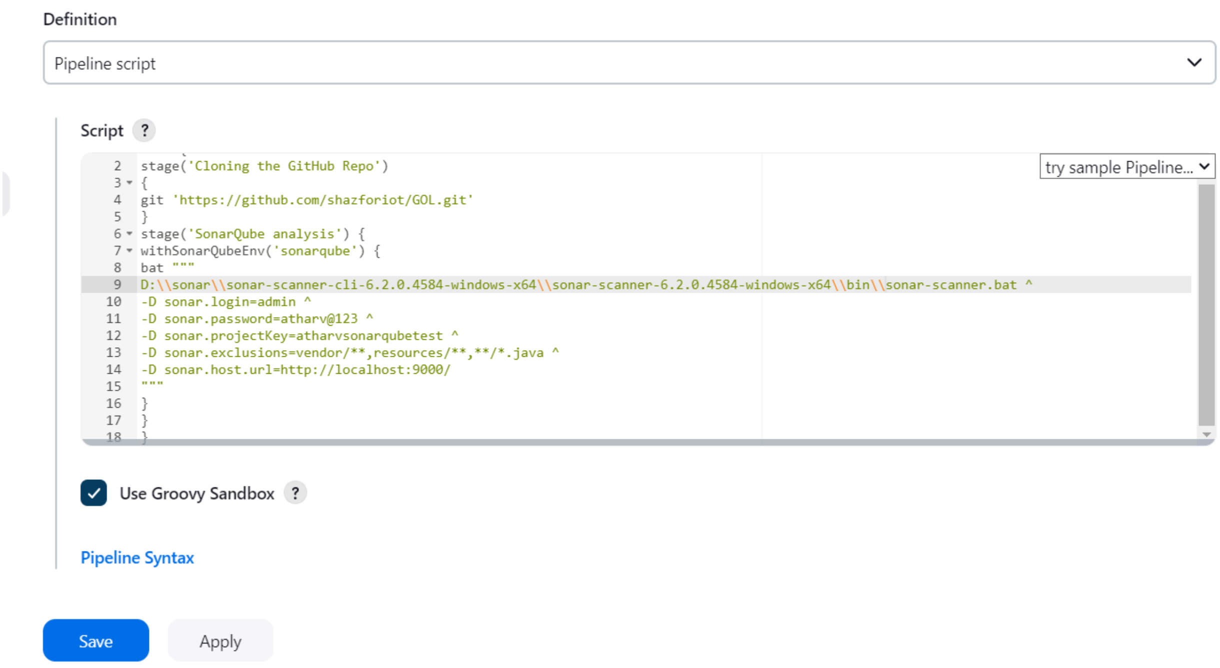
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

Step 10:

In the pipeline script section, input the following:

```
node {
  stage('Cloning the GitHub Repo') {
      git 'https://github.com/shazforiot/GOL.git'
  }

  stage('SonarQube Analysis') {
      withSonarQubeEnv('atharvsonarqube') {
          bat """
          <PATH_TO_SONARSCANNER_FOLDER>\\bin\\sonar-scanner.bat ^
          -D sonar.login=<SONARQUBE_LOGIN> ^
          -D sonar.password=<SONARQUBE_PASSWORD> ^
          -D sonar.projectKey=<PROJECT_KEY> ^
          -D sonar.exclusions=vendor/**,resources/**,**/*.java ^
          -D sonar.host.url=http://localhost:9000/
          """
      }
  }
}
```



Definition

Pipeline script

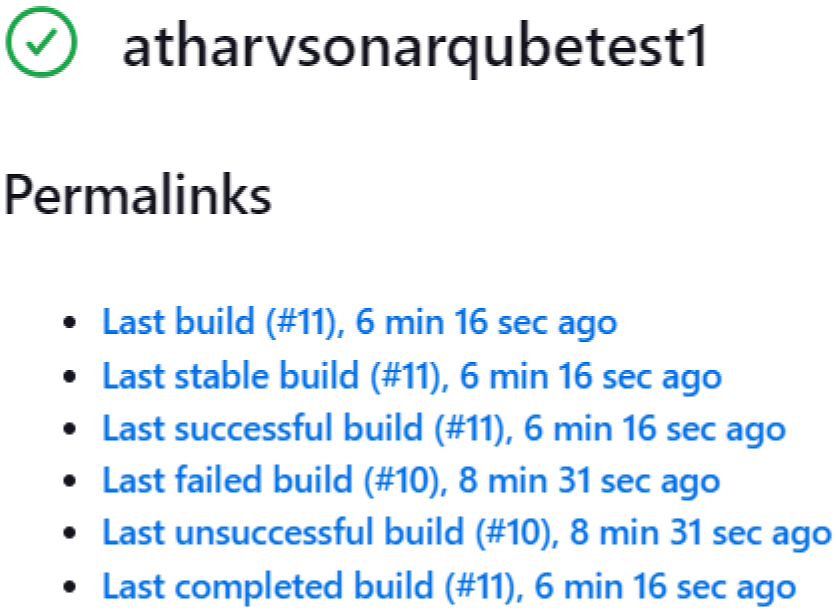
Script

```
stage('Cloning the GitHub Repo')
{
git 'https://github.com/shazforiot/GOL.git'
}
stage('SonarQube analysis') {
withSonarQubeEnv('sonarqube') {
bat """
D:\\sonar\\sonar-scanner-cli-6.2.0.4584-windows-x64\\sonar-scanner-6.2.0.4584-windows-x64\\bin\\sonar-scanner.bat ^
-D sonar.login=admin ^
-D sonar.password=atharv@123 ^
-D sonar.projectKey=atharvsonarqubetest ^
-D sonar.exclusions=vendor/**,resources/**,**/*.java ^
-D sonar.host.url=http://localhost:9000/
"""
}
}
}
```

☑ Use Groovy Sandbox

This script clones a sample Java project from GitHub, which has several issues that SonarQube will detect.

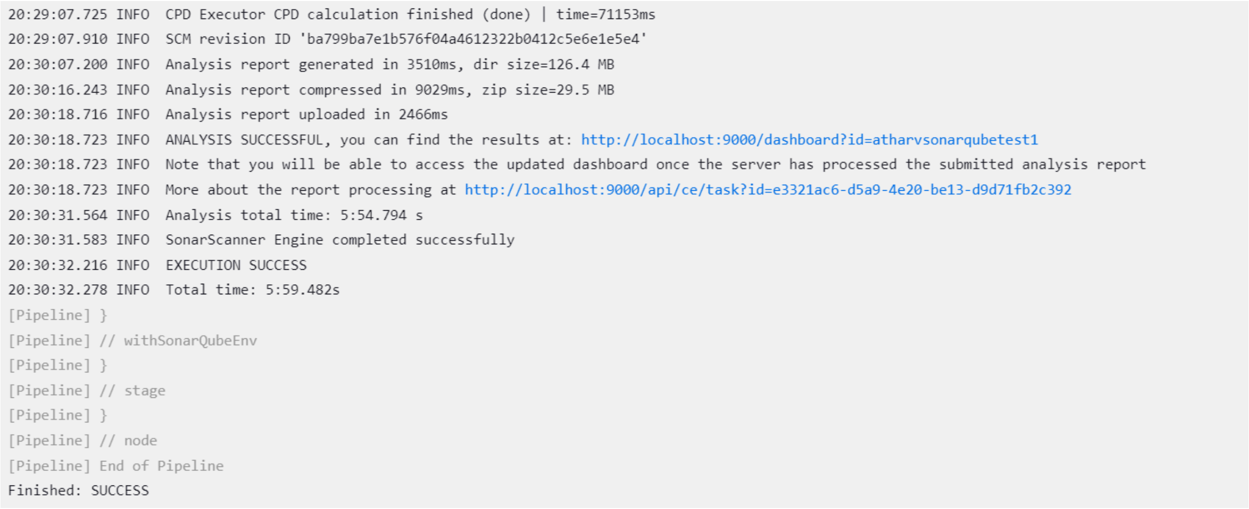
Step 11:
Go back to Jenkins, select the job you just created, and click **Build Now** to run the pipeline.



✓ atharvsonarqubetest1

Permalinks

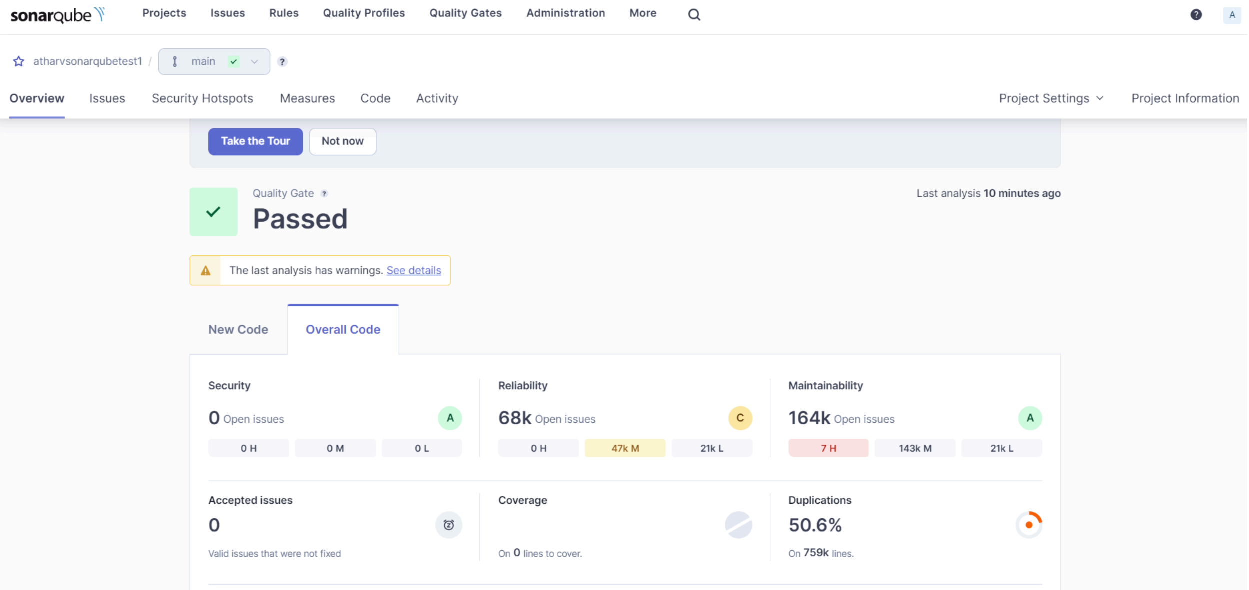
- Last build (#11), 6 min 16 sec ago
- Last stable build (#11), 6 min 16 sec ago
- Last successful build (#11), 6 min 16 sec ago
- Last failed build (#10), 8 min 31 sec ago
- Last unsuccessful build (#10), 8 min 31 sec ago
- Last completed build (#11), 6 min 16 sec ago



```
20:29:07.725 INFO  CPD Executor CPD calculation finished (done) | time=71153ms
20:29:07.910 INFO  SCM revision ID 'ba799ba7e1b576f04a4612322b0412c5e6e1e5e4'
20:30:07.200 INFO  Analysis report generated in 3510ms, dir size=126.4 MB
20:30:16.243 INFO  Analysis report compressed in 9029ms, zip size=29.5 MB
20:30:18.716 INFO  Analysis report uploaded in 2466ms
20:30:18.723 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=atharvsonarqubetest1
20:30:18.723 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
20:30:18.723 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=e3321ac6-d5a9-4e20-be13-d9d71fb2c392
20:30:31.564 INFO  Analysis total time: 5:54.794 s
20:30:31.583 INFO  SonarScanner Engine completed successfully
20:30:32.216 INFO  EXECUTION SUCCESS
20:30:32.278 INFO  Total time: 5:59.482s
[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Name:Atharv Nikam Div:D15C Roll :36

Step 12:
Once the build is complete, return to SonarQube to view the analysis of your project. Check for bugs, code smells, duplications, and other metrics related to the quality of your code.



Quality Gate
Passed

Last analysis 10 minutes ago

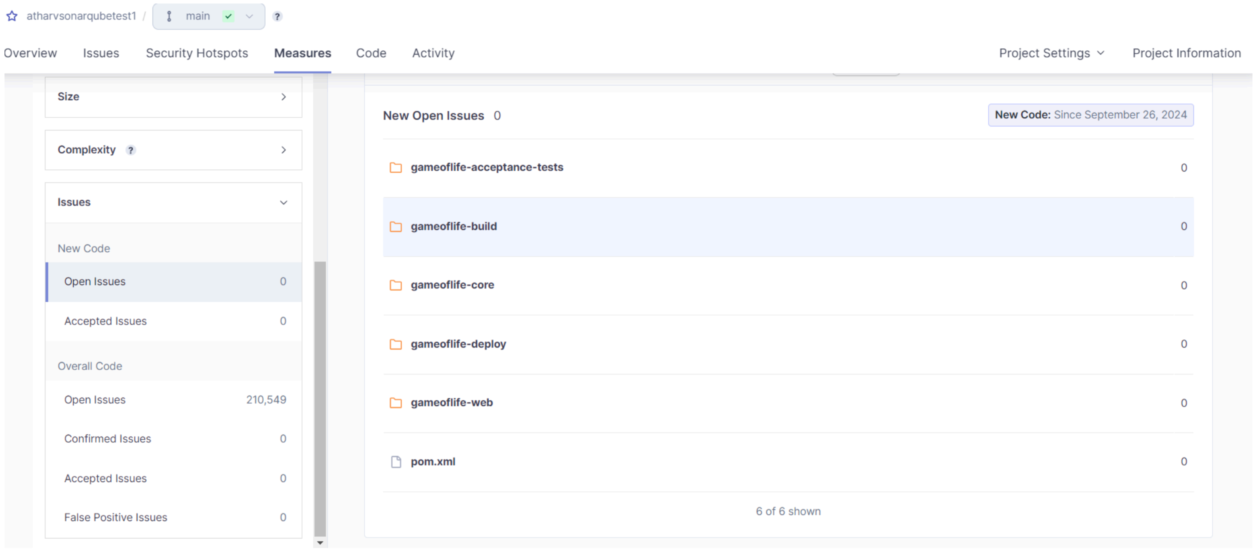
The last analysis has warnings. See details

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

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

- Code Problems



New Code
Open Issues 0
Accepted Issues 0

Overall Code
Open Issues 210,549
Confirmed Issues 0
Accepted Issues 0
False Positive Issues 0

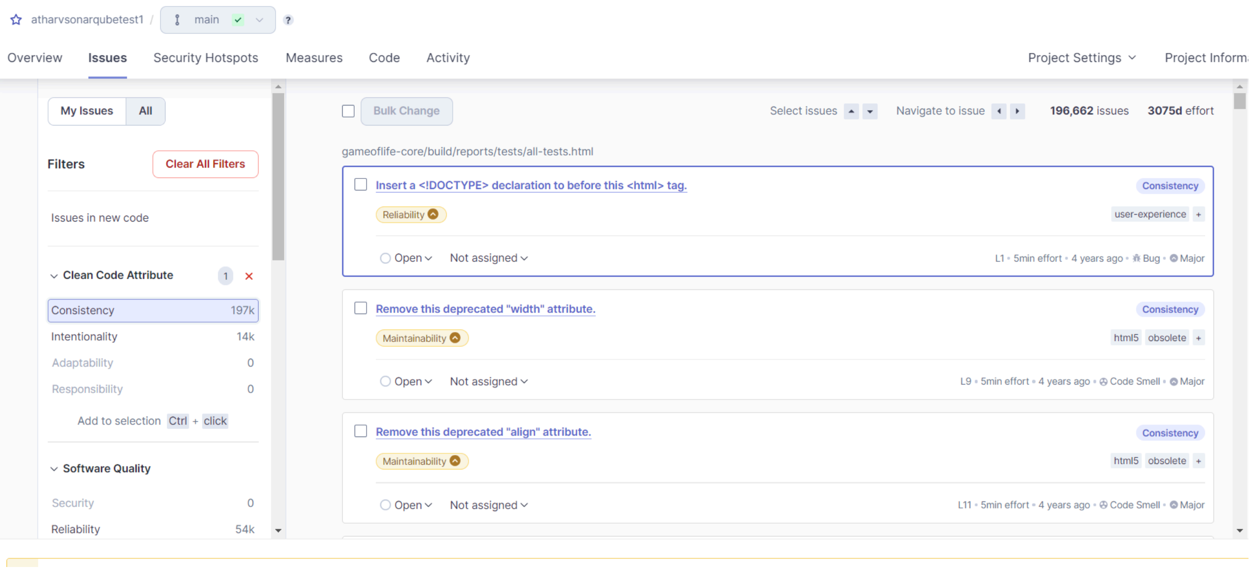
New Open Issues 0 | New Code: Since September 26, 2024

gameoflife-acceptance-tests	0
gameoflife-build	0
gameoflife-core	0
gameoflife-deploy	0
gameoflife-web	0
pom.xml	0

6 of 6 shown

Name:Atharv Nikam Div:D15C Roll :36

- Consistency



☆ atharvsonarqubetest1 / main ✓ ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Inform

My Issues All

Filters Clear All Filters

Issues in new code

Clean Code Attribute 1 ×

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Add to selection Ctrl + click

Software Quality

Security	0
Reliability	54k

□ Bulk Change Select issues Navigate to issue **196,662** issues **3075d** effort

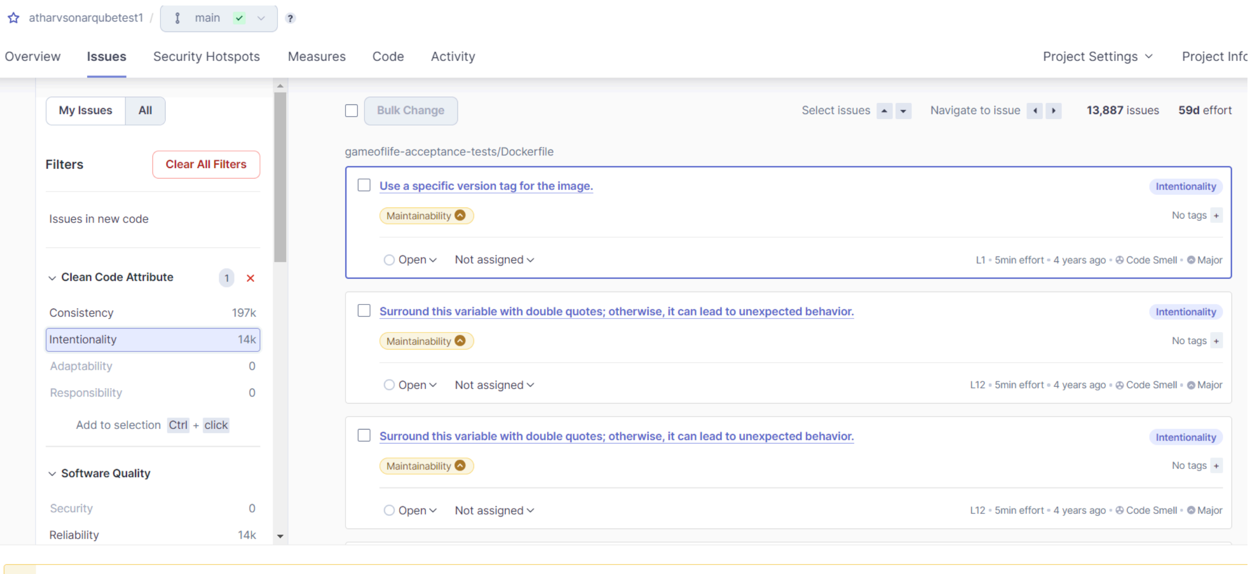
gameoflife-core/build/reports/tests/all-tests.html

□ Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency
Reliability · user-experience +
○ Open Not assigned · L1 • 5min effort • 4 years ago • Bug • Major

□ Remove this deprecated "width" attribute. Consistency
Maintainability · html5 obsolete +
○ Open Not assigned · L9 • 5min effort • 4 years ago • Code Smell • Major

□ Remove this deprecated "align" attribute. Consistency
Maintainability · html5 obsolete +
○ Open Not assigned · L11 • 5min effort • 4 years ago • Code Smell • Major

- Intentionality



☆ atharvsonarqubetest1 / main ✓ ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Info

My Issues All

Filters Clear All Filters

Issues in new code

Clean Code Attribute 1 ×

Consistency	197k
Intentionality	14k
Adaptability	0
Responsibility	0

Add to selection Ctrl + click

Software Quality

Security	0
Reliability	14k

□ Bulk Change Select issues Navigate to issue **13,887** issues **59d** effort

gameoflife-acceptance-tests/Dockerfile

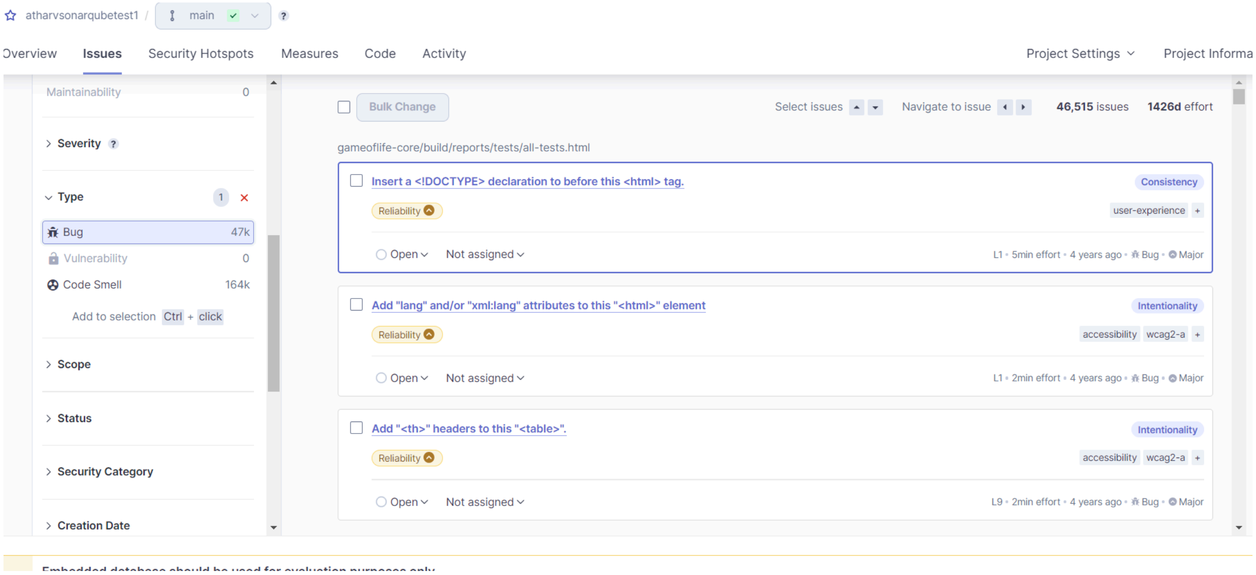
□ Use a specific version tag for the image. Intentionality
Maintainability · No tags +
○ Open Not assigned · L1 • 5min effort • 4 years ago • Code Smell • Major

□ 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 +
○ Open Not assigned · L12 • 5min effort • 4 years ago • Code Smell • Major

Name:Atharv Nikam Div:D15C Roll :36

- Bugs



46,515 issues **1426d** effort

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

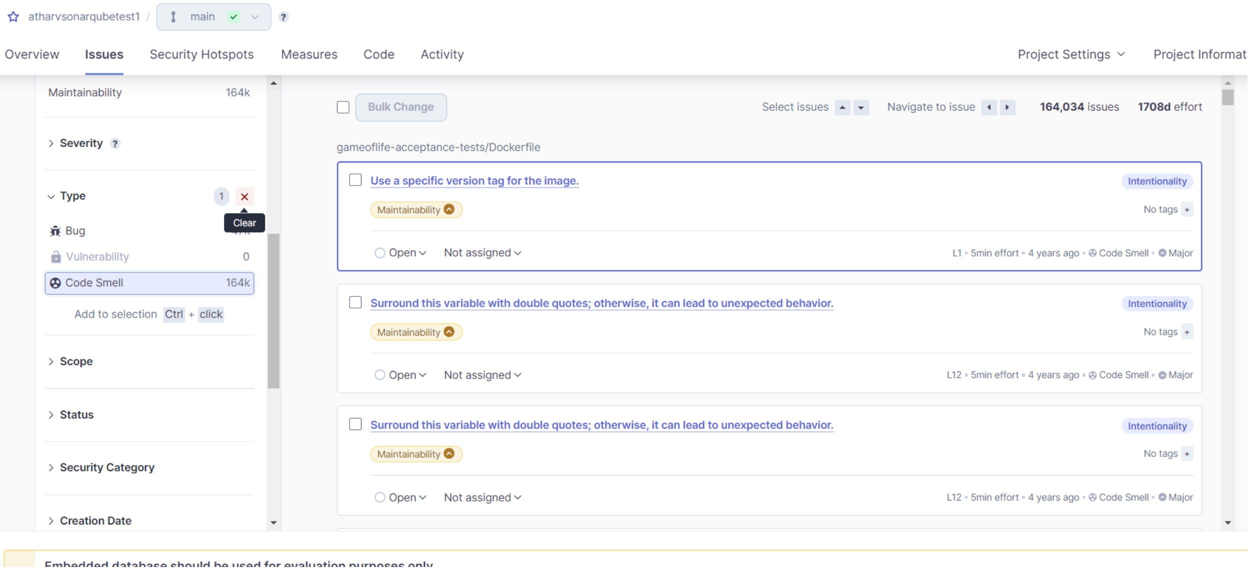
□ Insert a <!DOCTYPE> declaration to before this <html> tag. Consistency
Reliability user-experience +
○ Open Not assigned L1 • 5min effort • 4 years ago • Bug • Major

□ Add "lang" and/or "xml:lang" attributes to this "<html>" element Intentionality
Reliability accessibility wcag2-a +
○ 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

Embedded database should be used for evaluation purposes only

- Code Smells



164,034 issues **1708d** 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

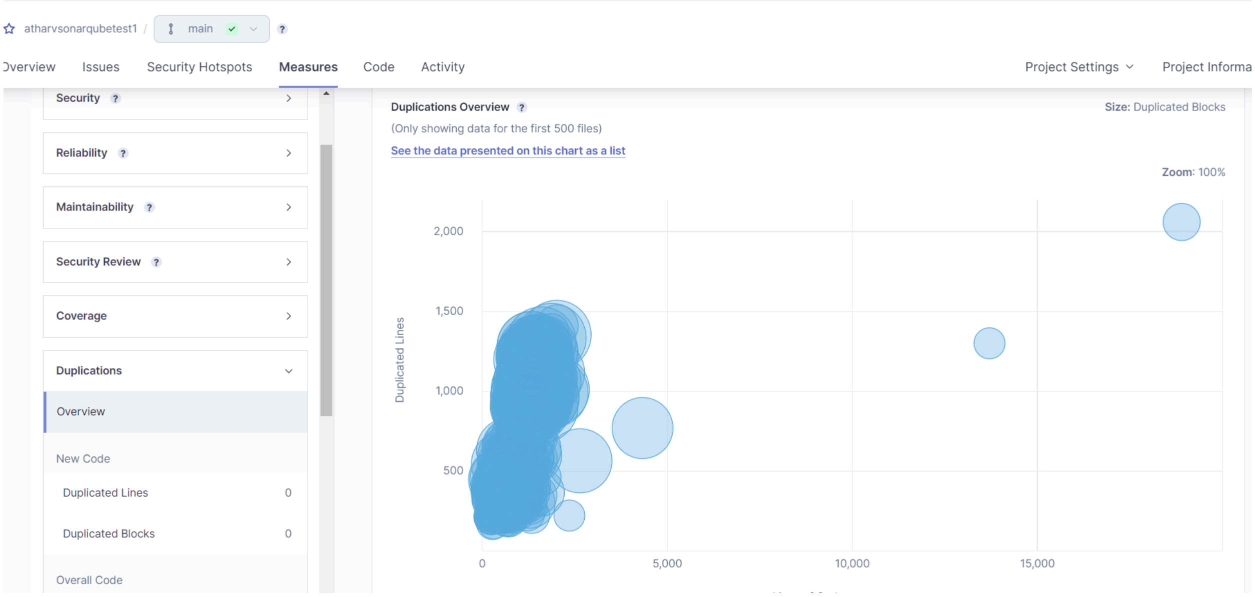
□ Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. Intentionality
Maintainability No tags +
○ Open Not assigned L12 • 5min effort • 4 years ago • Code Smell • Major

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

Embedded database should be used for evaluation purposes only

Name:Atharv Nikam Div:D15C Roll :36

- Duplications



New Code

Duplicated Lines 0

Duplicated Blocks 0

Overall Code

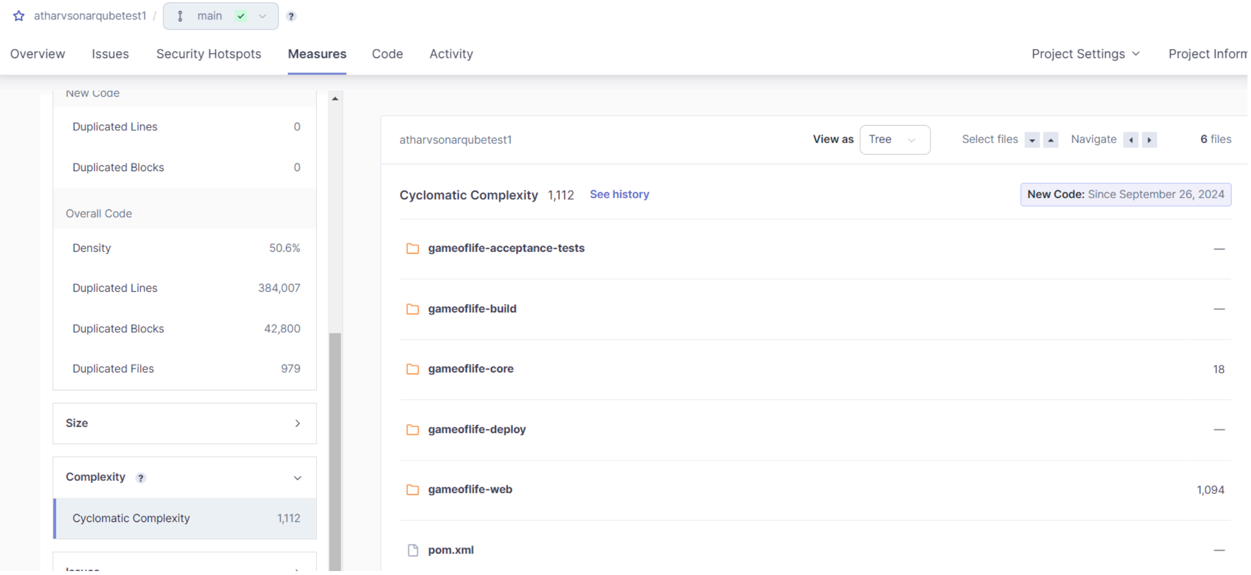
Duplications Overview

(Only showing data for the first 500 files)

See the data presented on this chart as a list

Size: Duplicated Blocks

- Cyclomatic Complexities



New Code	
Duplicated Lines	0
Duplicated Blocks	0
Overall Code	
Density	50.6%
Duplicated Lines	384,007
Duplicated Blocks	42,800
Duplicated Files	979
Cyclomatic Complexity	1,112

atharvsonarqubetest1 — 6 files

Cyclomatic Complexity 1,112

New Code: Since September 26, 2024

gameoflife-acceptance-tests	—
gameoflife-build	—
gameoflife-core	18
gameoflife-deploy	—
gameoflife-web	1,094
pom.xml	—

Conclusion:

This experiment allowed us to integrate Jenkins and SonarQube to set up a CI/CD pipeline capable of performing static analysis on Java code. Through this process, we automated the detection of common code issues such as bugs, code smells, and duplications. By leveraging Docker for SonarQube and the Jenkins pipeline, we streamlined the code scanning process, ensuring any issues were highlighted during the build phase. This integration demonstrates the importance of automated code quality checks in a continuous delivery environment