



Vivekanand Education Society's

Institute of Technology

An Autonomous Institute Affiliated to University of Mumbai,, Approved by AICTE & Recognized by Govt. of Maharashtra
Hashu Advani Memorial Complex, Collector Colony, Chembur East, Mumbai - 400074.

Department of Information Technology

A.Y. 2024-25

Advance DevOps Lab

Experiment 08

Aim: To identify and remediate application vulnerabilities earlier and help integrate security in the development process using SAST Techniques.

Roll No.	42
Name	NAIKWADI YASH SHIVDAS
Class	D15B
Subject	Advance DevOps Lab
LO Mapped	LO1: To understand the fundamentals of Cloud Computing and be fully proficient with Cloud based DevOps solution deployment options to meet your business requirements. LO4: To identify and remediate application vulnerabilities earlier and help integrate security in the development process using SAST Techniques.
Grade:	

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:

What is a CI/CD Pipeline?

A Continuous Integration/Continuous Delivery (CI/CD) pipeline automates the processes of building, testing, and delivering software. It allows developers to integrate their code changes frequently and deliver new software versions efficiently. The pipeline includes various steps such as coding, building the application, running tests, and deploying the application to users.

What is Jenkins?

Jenkins is an open-source automation server widely used to facilitate CI/CD pipelines. It automates tasks needed to compile code, run tests, and deploy applications. Jenkins integrates with various tools, making it a popular choice for developers looking to streamline their software development processes.

What is SonarQube?

SonarQube is a tool that performs static analysis of code to assess its quality. It checks the source code for bugs, security vulnerabilities, and code smells (issues that may indicate deeper problems). By providing detailed reports, SonarQube helps developers understand the quality of their code and how to improve it.

Integration of Jenkins and SonarQube:

Integrating Jenkins with SonarQube allows the CI/CD pipeline to automatically analyze code quality during the build process. Whenever developers commit changes, Jenkins triggers a SonarQube scan to detect any issues early. This integration ensures that only high-quality code is deployed, reducing the risk of bugs and vulnerabilities.

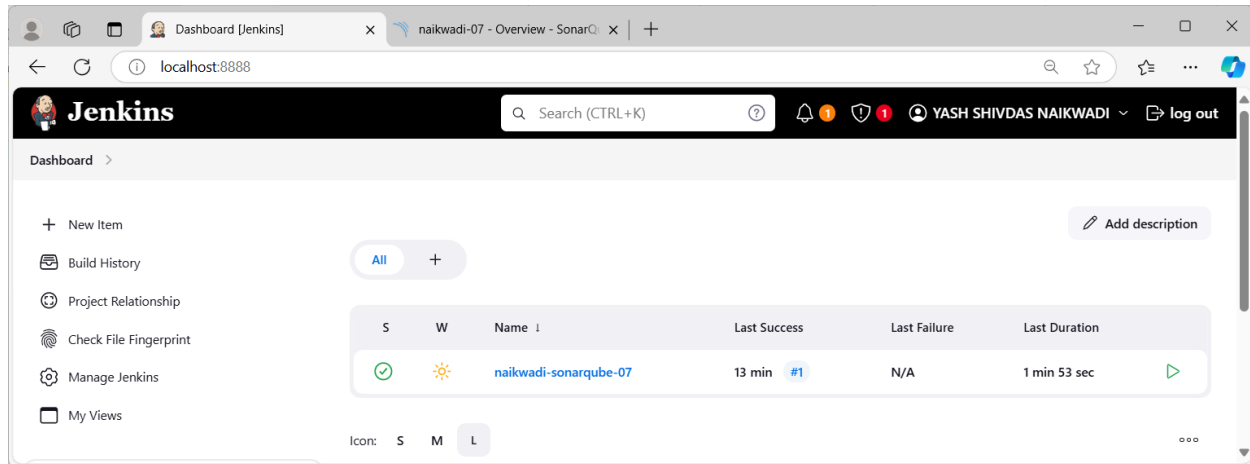
Importance of Code Quality Analysis:

Using SonarQube in the CI/CD pipeline helps developers identify and fix issues before code is deployed. This proactive approach saves time and resources, improves application quality, and enhances security by addressing vulnerabilities early in development.

Benefits of SonarQube:

- **Sustainability:** SonarQube helps reduce complexity and vulnerabilities, extending the lifespan of applications.
- **Increased Productivity:** It streamlines development by minimizing the effort required for manual code reviews, lowering maintenance costs.
- **Error Detection:** SonarQube automatically alerts developers to errors, allowing them to fix issues before production.
- **Consistency:** The tool sets standards for code quality, ensuring overall improvement across projects.
- **Business Scaling:** SonarQube can evaluate multiple projects at once, supporting organizational growth.
- **Enhanced Developer Skills:** Regular feedback helps developers improve their coding practices and fosters continuous learning.

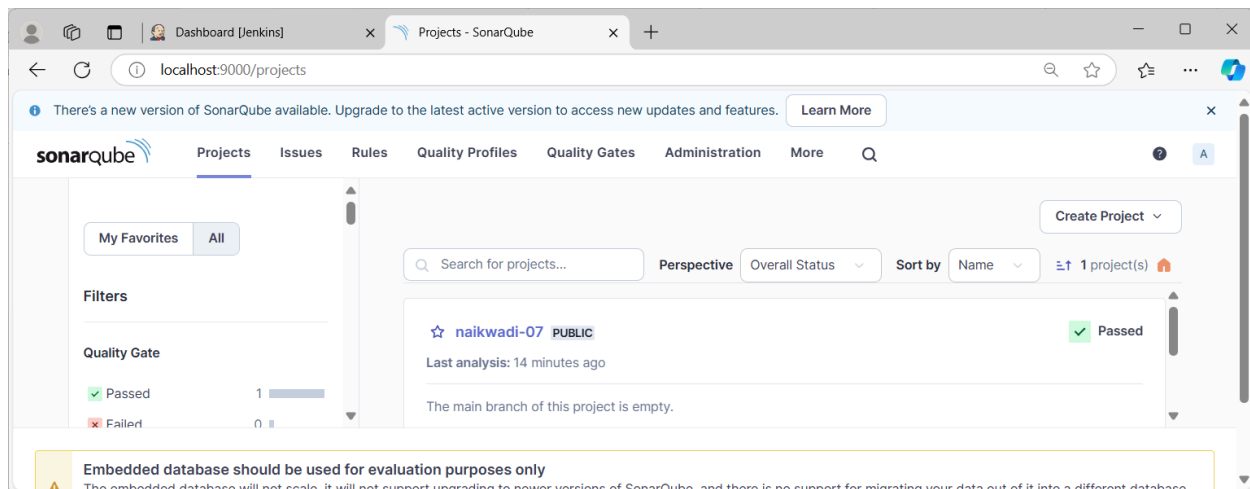
Open Jenkins by going to <http://localhost:8080> in your browser (or use the port you set during installation).



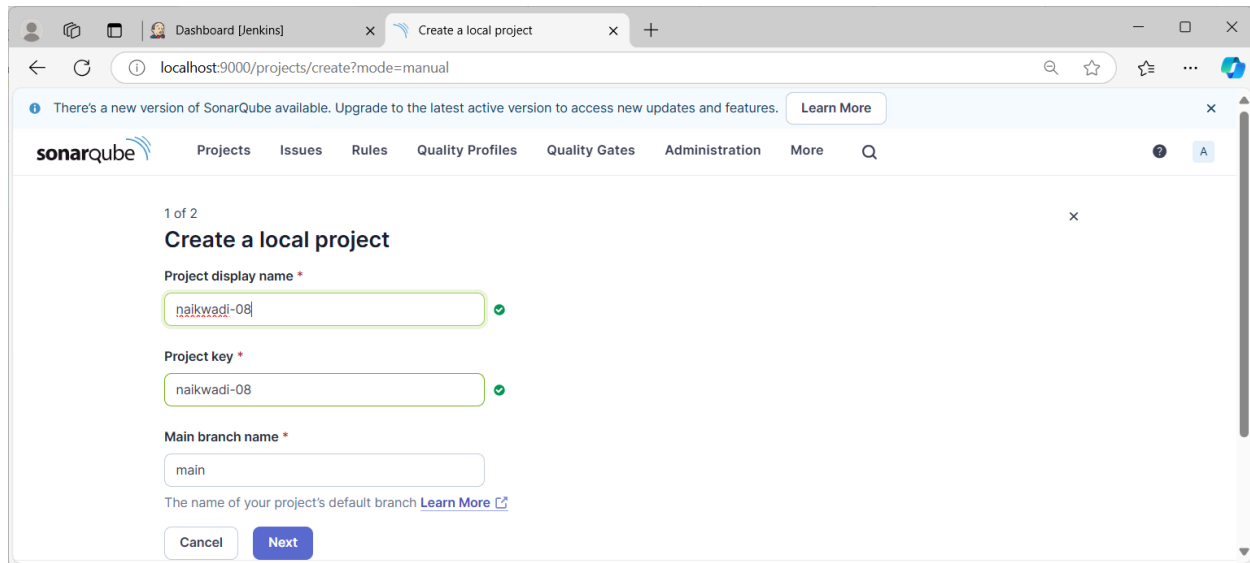
As we have already prepared the docker container of sonarqube in exp 07. We just need to start it again.



Visit <http://localhost:9000> in your browser. If SonarQube is running, you'll see the login page.



Click on "Create new project". Name the project **sonarqube-test**.



1 of 2

Create a local project

Project display name *

naikwadi-08

Project key *

naikwadi-08

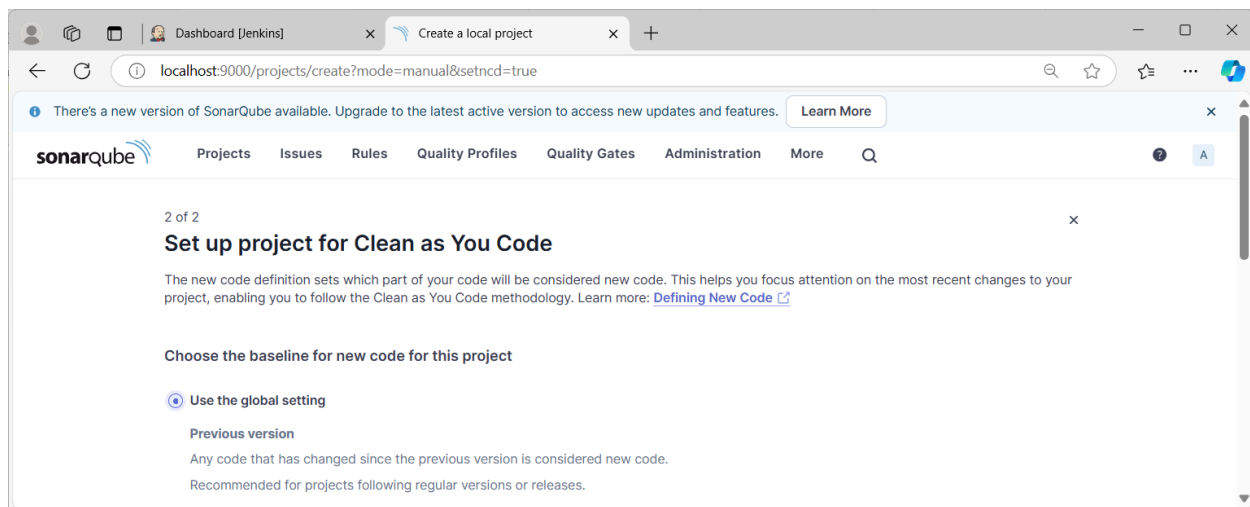
Main branch name *

main

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

Cancel Next

Choose Use the global setting.



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](#)

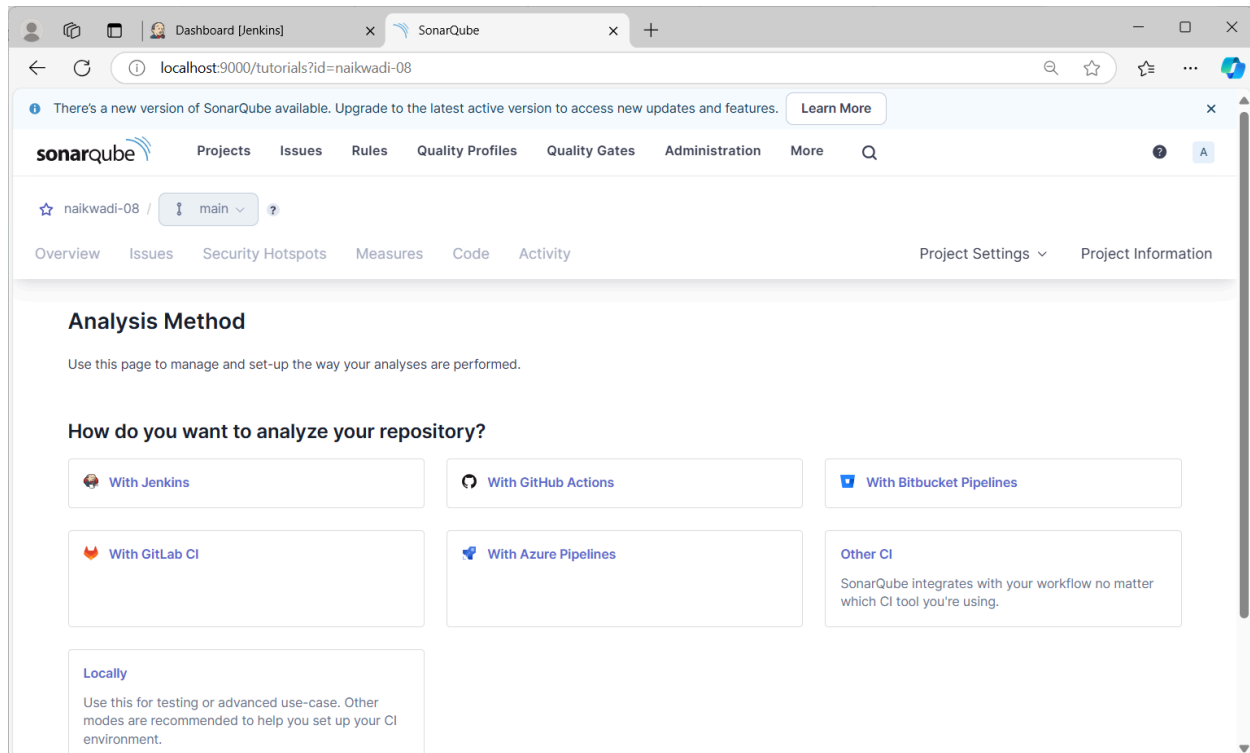
Choose the baseline for new code for this project

☒ 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.



naikwadi-08 / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Analysis Method

Use this page to manage and set-up the way your analyses are performed.

How do you want to analyze your repository?

☐ With Jenkins

☐ With GitHub Actions

☒ With Bitbucket Pipelines

☐ With GitLab CI

☐ With Azure Pipelines

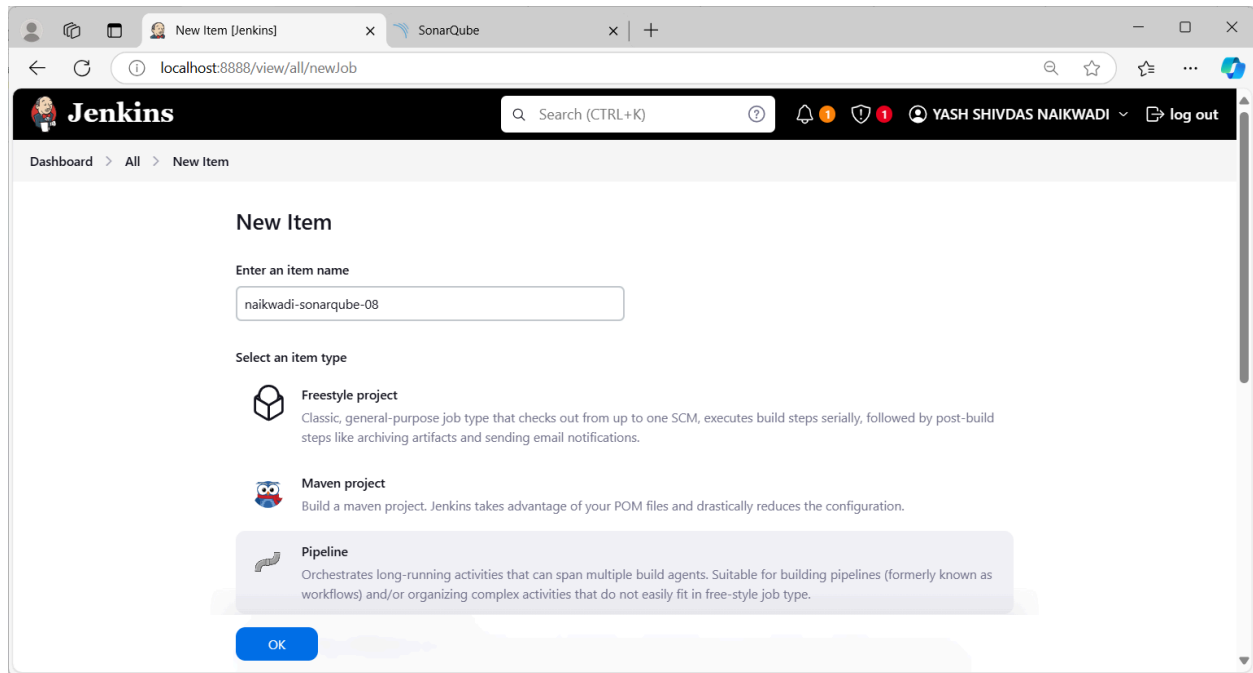
☐ Other CI

SonarQube integrates with your workflow no matter which CI tool you're using.

☐ Locally

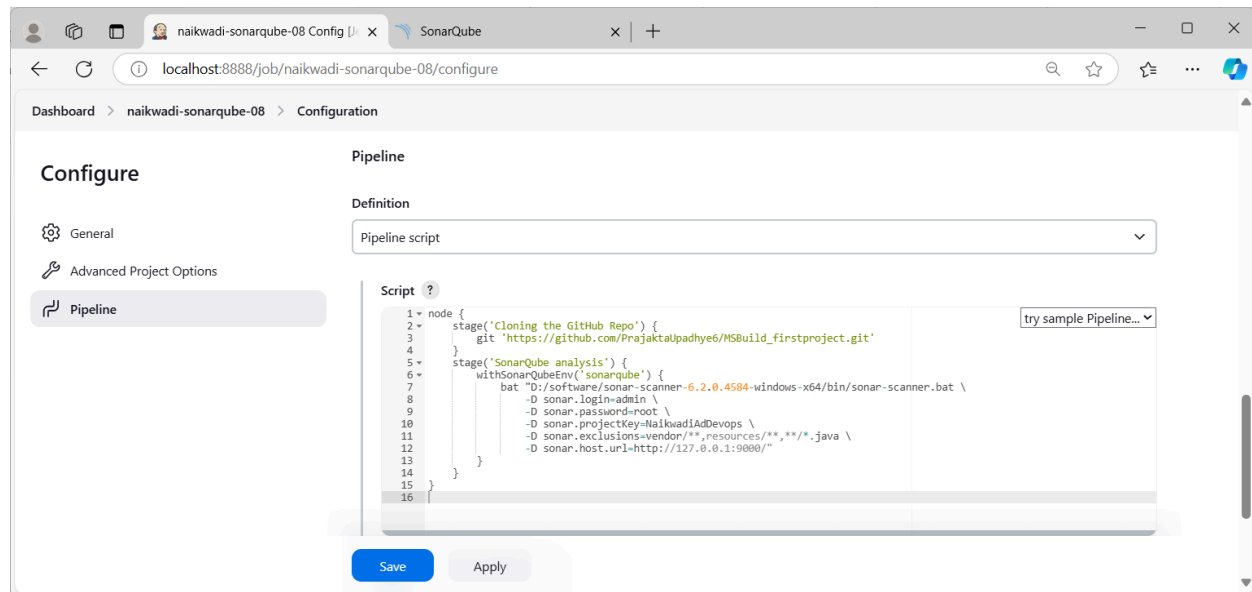
Use this for testing or advanced use-case. Other modes are recommended to help you set up your CI environment.

In Jenkins create a pipeline here named “SonarQube”



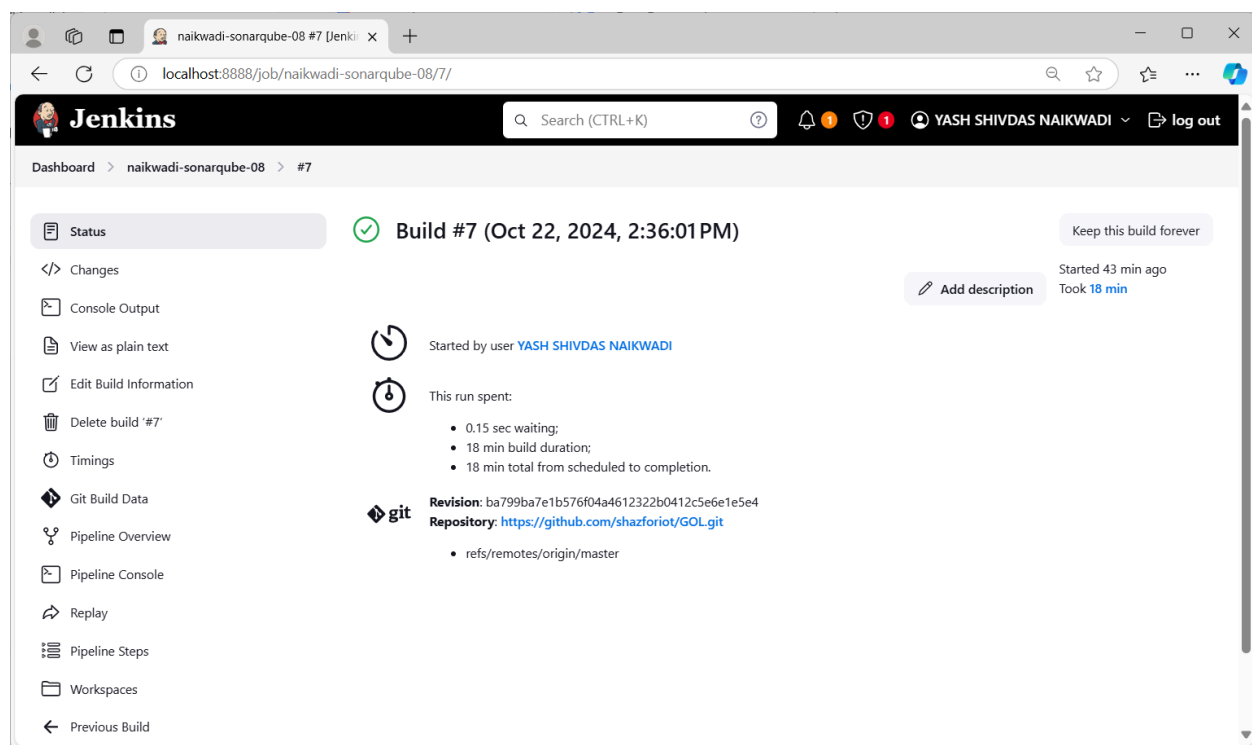
Enter the following in pipeline script:

```
node {  
  stage('Cloning the GitHub Repo') {  
    git 'https://github.com/PrajaktaUpadhye6/MSBuild_firstproject.git'  
  }  
  stage('SonarQube analysis') {  
    withSonarQubeEnv('sonarqube') {  
      bat "D://software//sonar-scanner-6.2.0.4584-windows-x64//bin//sonar-scanner.bat \  
        -D sonar.login=admin \  
        -D sonar.password=root \  
        -D sonar.projectKey=NaikwadiAdDevops \  
        -D sonar.exclusions=vendor/**,resources/**,**/*.java \  
        -D sonar.host.url=http://127.0.0.1:9000/"  
    }  
  }  
}
```



It is a java sample project which has a lot of repetitions and issues that will be detected by SonarQube.

Build and run:



Go back to SonarQube at <http://localhost:9000>. Open the sonarqube-test project you created earlier.

Check different tabs for issues like:

- Bugs and Code Smells: These indicate potential problems in the code.
- Unfinished TODOs: Unresolved items in the code.
- Duplicates: Repeated code blocks.
- Cyclomatic Complexity: Measure of how complex the code is.

☆ sonarqube PUBLIC

✓ Passed

Last analysis: 48 minutes ago • 683k Lines of Code • HTML, XML, ...

A 0

C 68k

A 164k

E 0.0%

—

50.6%

SecurityReliabilityMaintainabilityHotspots ReviewedCoverageDuplications

☆ sonarqube / main ✓ ?

OverviewIssuesSecurity HotspotsMeasuresCodeActivityProject SettingsProject Information

main

683k Lines of Code • Version not provided • Set as homepageTake the Tour

Quality Gate

✓ Passed

Last analysis 48 minutes ago

⚠ The last analysis has warnings. See details

New Code

Overall Code

New Code: Since September 19, 2024 Started 4 hours ago

New issues

0

Required = 0

Accepted issues

0

Valid issues that were not fixed

☆ sonarqube / main ✓ ?

OverviewIssuesSecurity HotspotsMeasuresCodeActivityProject SettingsProject Information

Project Overview

Security

Reliability

Overview

New Code

Issues0

RatingA

Remediation Effort0

Overall Code

Issues67624

RatingC

Remediation Effort1426d

sonarqube

View asTree

Select files

Navigate

6 files

Reliability Rating on New Code A

New Code: Since September 19, 2024

gameoflife-acceptance-testsA

gameoflife-buildA

gameoflife-coreA

gameoflife-deployA

gameoflife-webA

pom.xmlA

6 of 6 shown

☆ sonarqube / main ✓ ?

OverviewIssuesSecurity HotspotsMeasuresCodeActivityProject SettingsProject Information

Filter events

Start Date

to

End Date

Reset dates

NOT PROVIDED

September 19, 2024


5:36 PM : Quality Profile Use "Sonar way" (JSP)
Quality Profile Use "Sonar way" (CSS)
Quality Profile Use "Sonar way" (XML)
Quality Profile Use "Sonar way" (HTML)
Quality Profile Use "Sonar way" (Docker)
Version: not provided







Everything above this line is New Code

2:42 PM : First analysis since upgrading to SonarQube 10.6.0.92116

Issues

IssuesNew Code



Duplicated Lines 384,007 See history		New Code: Since September 19, 2024	
		Duplicated Lines	Duplicated Lines (%)
 gameoflife-acceptance-tests		0	0.0%
 gameoflife-build		0	0.0%
 gameoflife-core		374	9.6%
 gameoflife-deploy		0	0.0%
 gameoflife-web		383,633	50.9%
 pom.xml		0	0.0%

Conclusion:

Integrating Jenkins with SonarQube in a CI/CD pipeline allows developers to automatically analyze code for bugs and security vulnerabilities during the development process. This helps ensure that only high-quality code is delivered, making applications more secure and reliable.