



# 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.	43
Name	Harsh Pramod Padyal
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:**

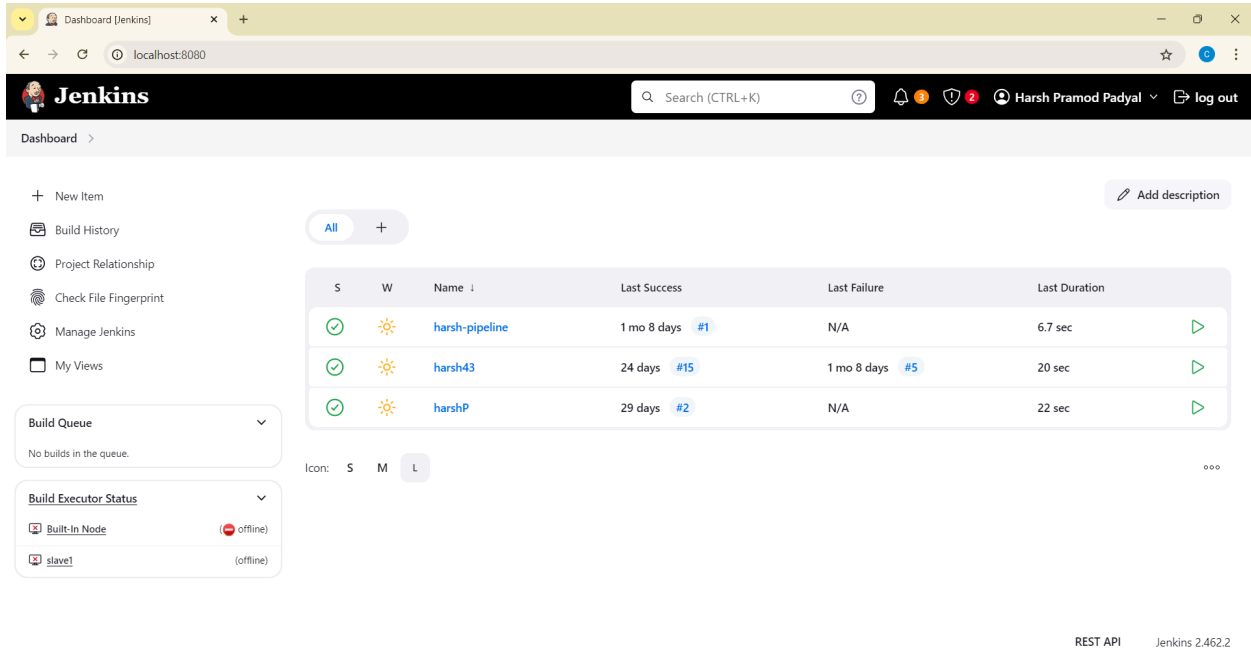
Using SonarQube in the CI/CD pipeline helps developers identify and fix issues before code is deployed. This proactive approach saves time and resources, improves application quality, and enhances security by addressing vulnerabilities early in development.

### **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)



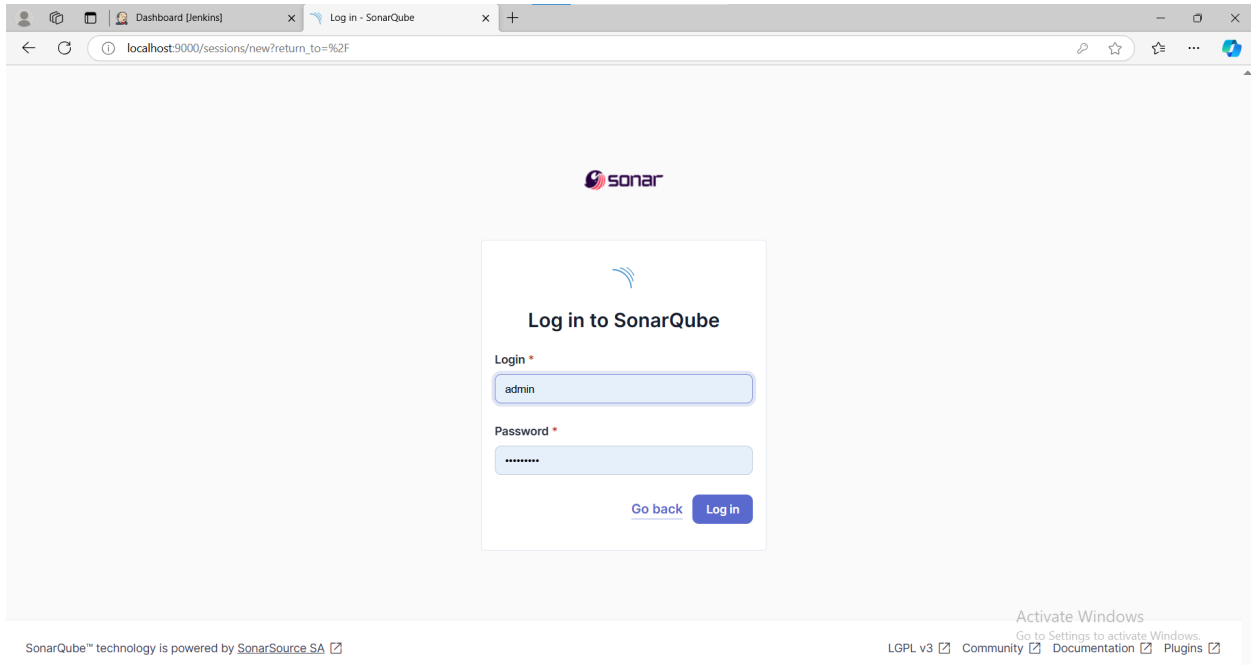
The screenshot shows the Jenkins Dashboard in a web browser. The top navigation bar includes the Jenkins logo, a search bar, and a user profile for 'Harsh Pramod Padyal' with a 'log out' button. The main content area features a sidebar on the left with links to 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The central panel displays a table of build history for the 'harsh-pipeline' project. The table has columns for status (S), warnings (W), name, last success, last failure, and last duration. Three builds are listed: 'harsh-pipeline' (1 mo 8 days, #1, 6.7 sec), 'harsh43' (24 days, #15, 20 sec), and 'harshP' (29 days, #2, 22 sec). Below the table, there are filters for 'Icon: S M L' and a 'Build Queue' section showing 'No builds in the queue.' and a 'Build Executor Status' section showing 'Built-In Node' (offline) and 'slave1' (offline). The bottom right corner indicates 'REST API' and 'Jenkins 2.462.2'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	⚠	harsh-pipeline	1 mo 8 days #1	N/A	6.7 sec
✓	⚠	harsh43	24 days #15	1 mo 8 days #5	20 sec
✓	⚠	harshP	29 days #2	N/A	22 sec

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

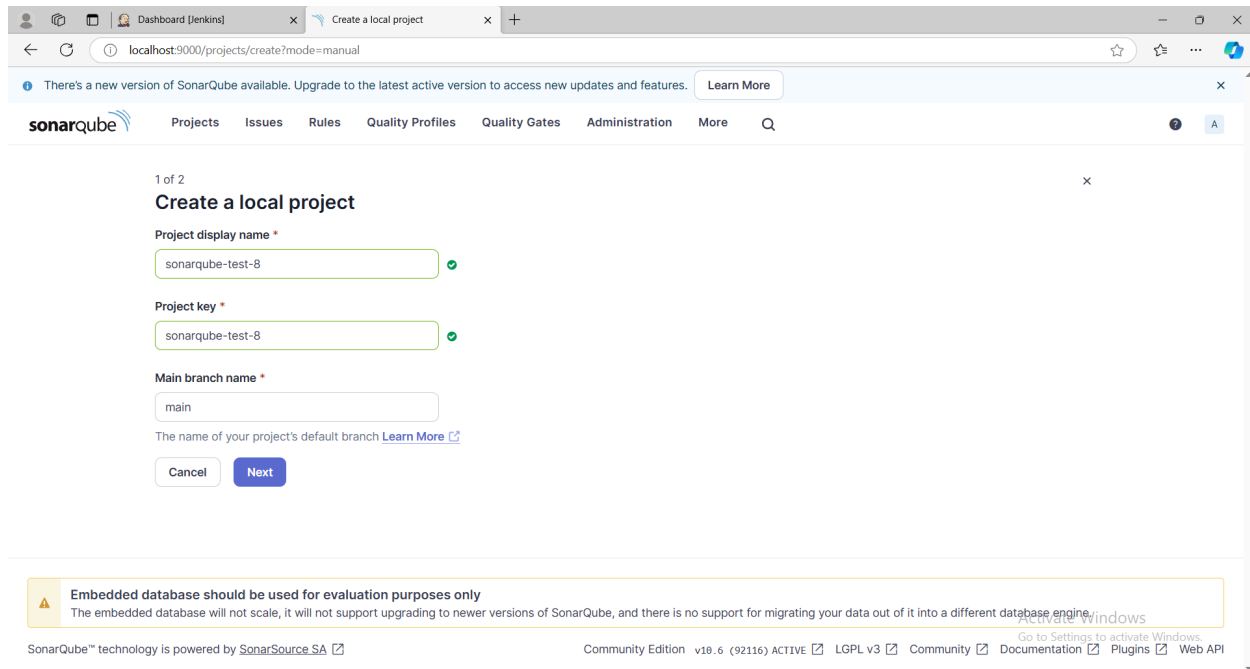
```
C:\Windows\system32>docker start sonarqube  
sonarqube
```

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



The screenshot shows the SonarQube login page in a web browser. The page has a light gray background with the Sonar logo at the top. The main content is a white box with the title 'Log in to SonarQube'. It contains two input fields: 'Login' with the value 'admin' and 'Password' with masked characters. Below the fields are two buttons: 'Go back' and 'Log in'. At the bottom of the page, there is a footer with the text 'SonarQube™ technology is powered by SonarSource SA' and a link to 'Activate Windows'. The browser's address bar shows the URL 'localhost:9000/sessions/new?return\_to=%2F'.

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



1 of 2

### Create a local project

**Project display name \***  
sonarqube-test-8

**Project key \***  
sonarqube-test-8

**Main branch name \***  
main

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

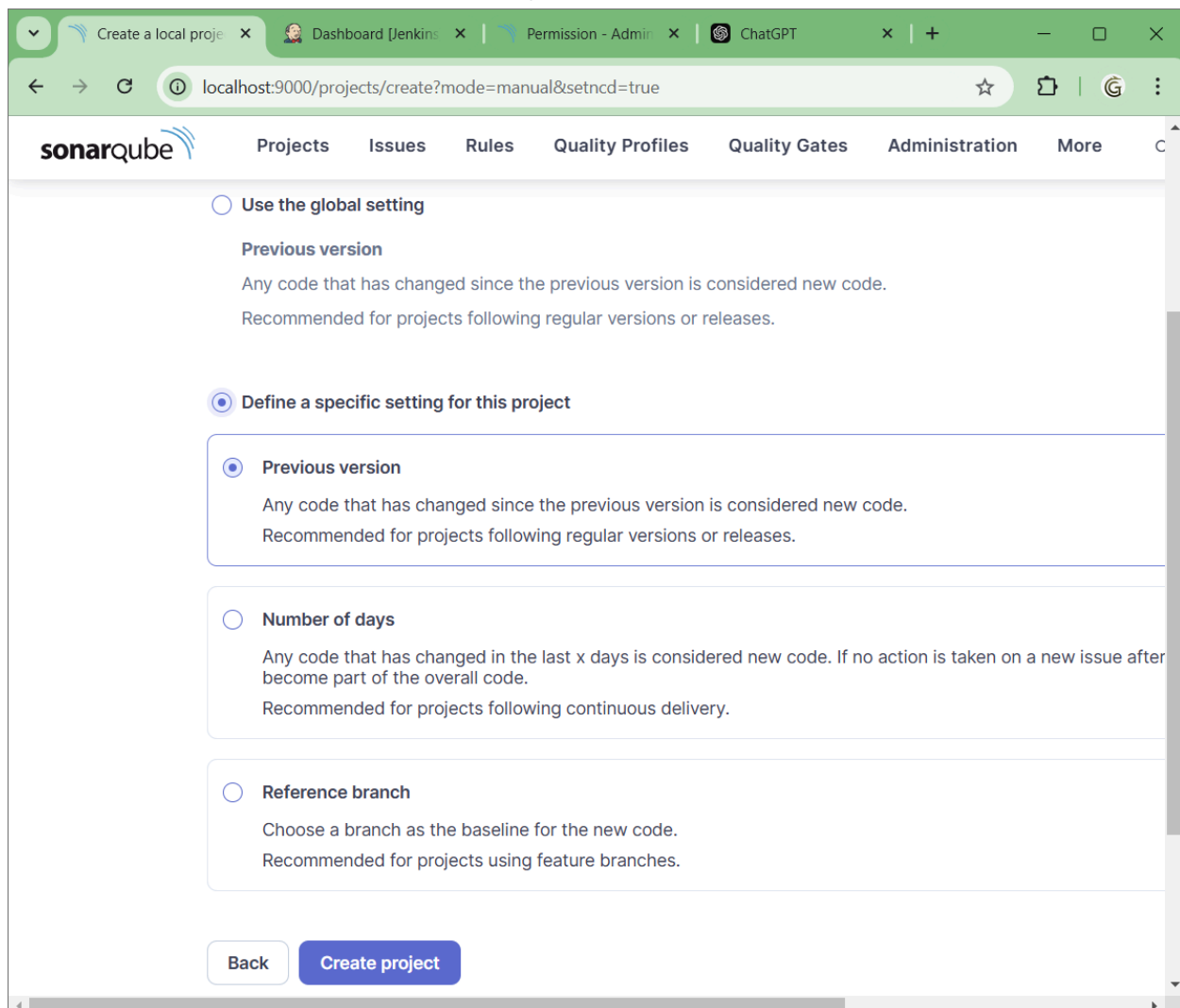
[Cancel](#) [Next](#)

**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.

SonarQube™ technology is powered by [SonarSource SA](#)

Community Edition v10.6 (92116) ACTIVE [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

Choose **Define a specific setting for this project**. Choose **Previous version** and proceed.



☐ 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 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.

[Back](#) [Create project](#)

SonarQube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationHelp

AdDevopsLab8main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject 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

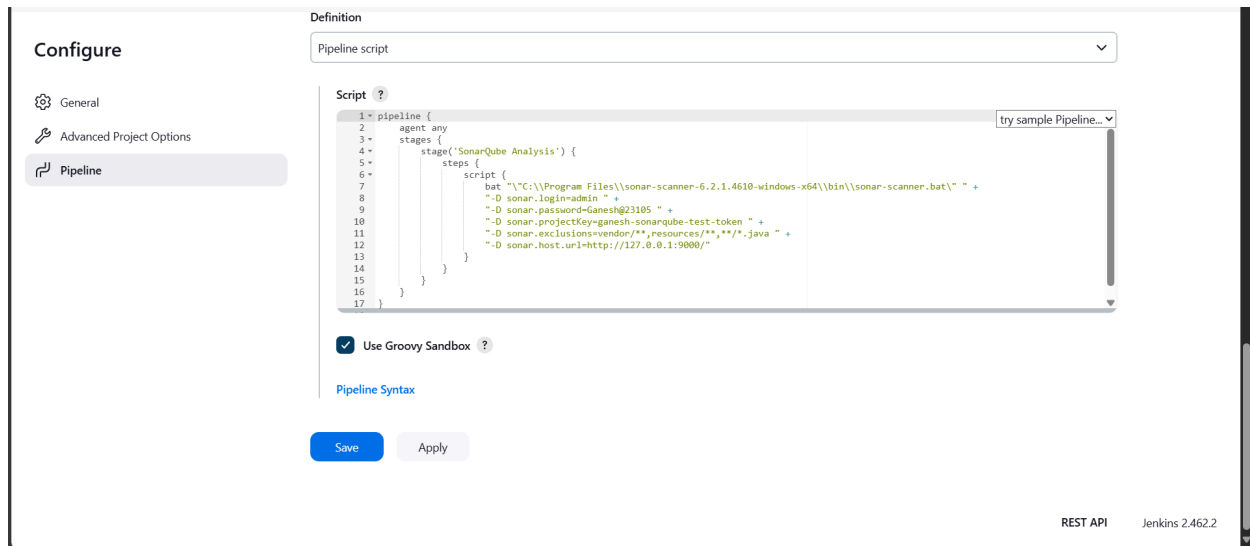
Other CI  
SonarQube integrates with your workflow no matter which CI tool you're using.

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

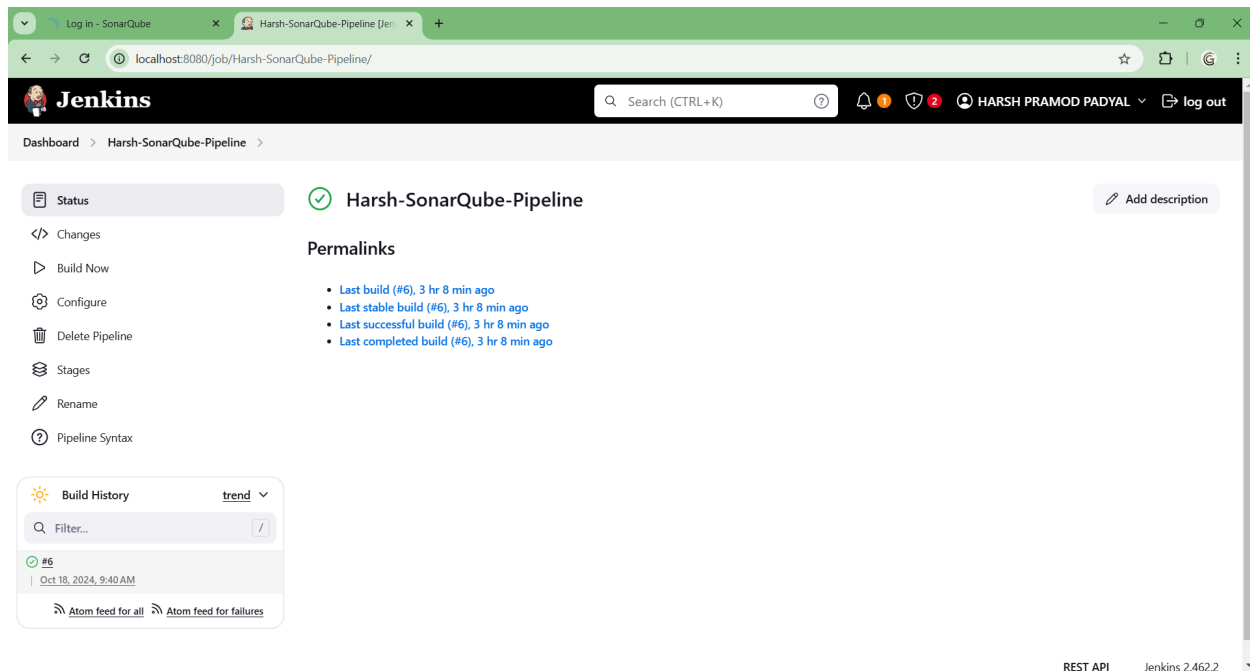
```
node {
    stage('Cloning the GitHub Repo') {
        git 'https://github.com/shazforiot/GOL.git'
    }

    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            sh "/path/to/sonar-scanner/bin/sonar-scanner \
-Dsonar.login=<SonarQube_USERNAME> \
-Dsonar.password=<SonarQube_PASSWORD> \
-Dsonar.projectKey=<Project_KEY> \
"
```

```
-Dsonar.exclusions=vendor/**,resources/**,**/*.java \  
-Dsonar.host.url=http://127.0.0.1:9000/"  
}  
}  
}
```



Apply and Save.



Go to Manage Jenkins > Configure System.

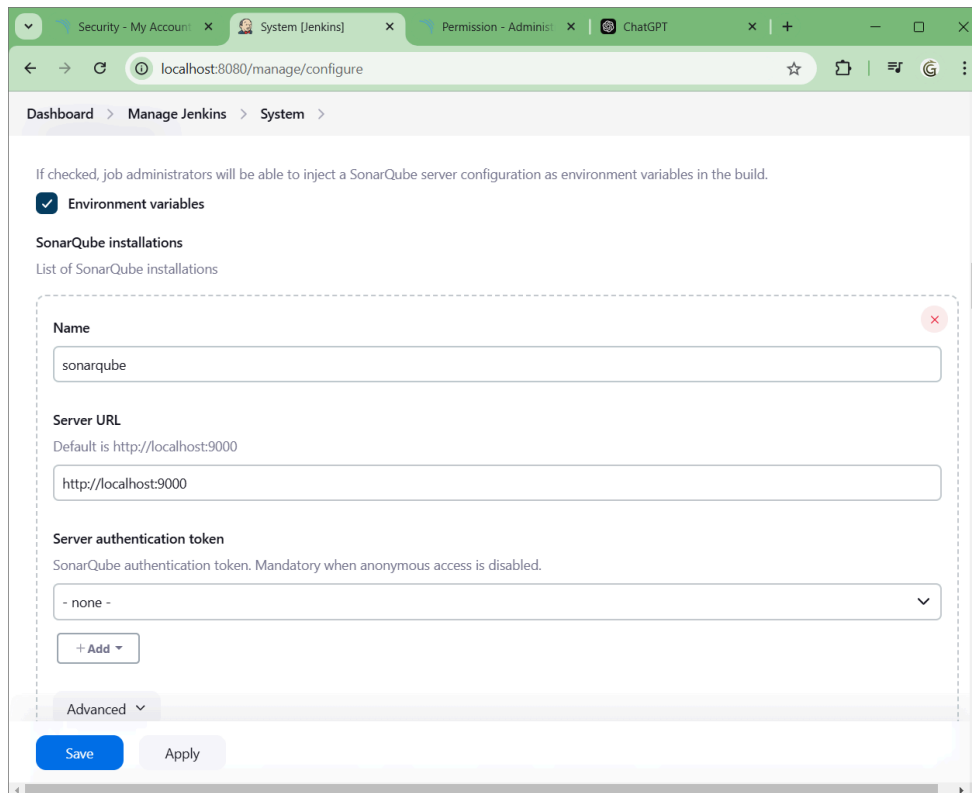
Scroll down to the SonarQube servers section.

Add a new SonarQube server:

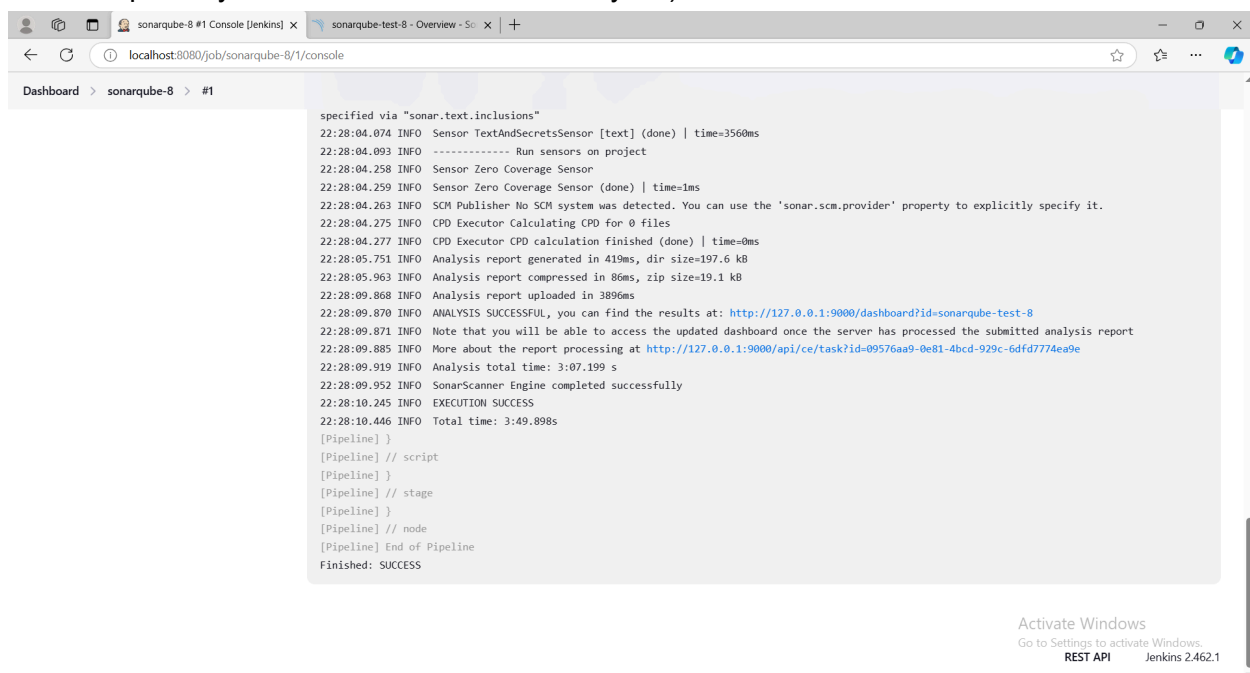
Provide the URL: <http://localhost:9000>

Enter your authentication token (from SonarQube).

Select "Add" next to the Server authentication token.




Go back to your pipeline and click Build Now to trigger the build. (The pipeline will clone the GitHub repository and run the SonarQube analysis.)



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%

Security


Reliability

Maintainability

Hotspots Reviewed

Coverage

Duplications

 sonarqube / main ✓ ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main

683k Lines of Code • Version not provided • Set as homepage Take 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 ✓ ?

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

Project Overview

Security

Reliability

Overview

New Code

Issues 0

Rating A

Remediation Effort 0

Overall Code

Issues 67624

Rating C

Remediation Effort 1426d

sonarqube

View as Tree

Select files

Navigate

6 files

Reliability Rating on New Code A

New Code: Since September 19, 2024

gameoflife-acceptance-tests

A

gameoflife-build

A

gameoflife-core

A

gameoflife-deploy

A

gameoflife-web

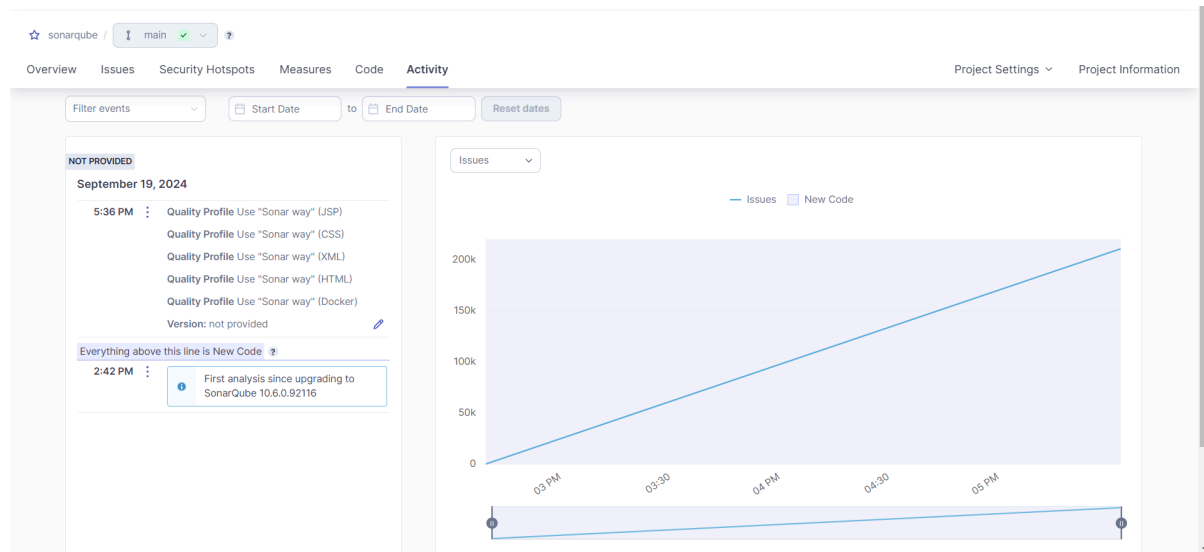
A

pom.xml

A

6 of 6 shown





Duplicated Lines 384,007 <a href="#">See history</a>		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.