



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.	44
Name	Ganesh Sanjay Pandhre
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 at localhost:8080. The dashboard includes a sidebar with navigation links: New Item, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, and My Views. The main area displays a table of build history for the 'ganesh-pipeline' job.

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	ganesh-pipeline	29 days #1	N/A	7.4 sec
✓	☁	ganesh-sonarqube	1 hr 45 min #3	1 hr 49 min #2	1 min 2 sec
✓	☀	ganesh44	24 days #9	29 days #1	1 min 29 sec
✓	☁	GaneshP	24 days #5	29 days #1	51 sec

Below the build history table, there is a section for 'Build Queue' (No builds in the queue) and 'Build Executor Status'. The 'Build Executor Status' section shows two executors: 'Built-In Node' (1 idle, 2 idle) and 'Ganesh_Agent' (offline).

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

```

C:\Users\ganes>docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
5a141c150a20   sonarqube:latest  "/opt/sonarqube/dock..."  2 hours ago   Up 2 hours   0.0.0.0:9000->9000/tcp    ganesh-sonarqube

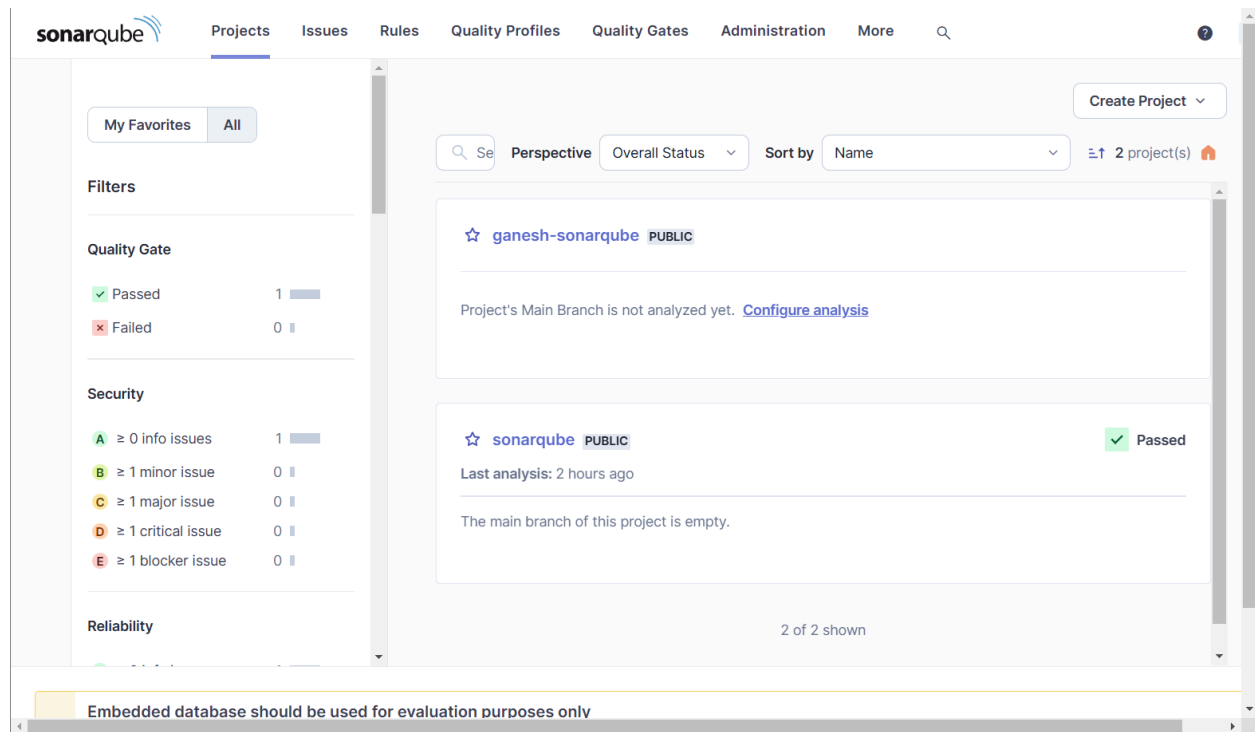
C:\Users\ganes>docker container start ganesh-sonarqube
ganesh-sonarqube

C:\Users\ganes>docker container ls
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
5a141c150a20   sonarqube:latest  "/opt/sonarqube/dock..."  2 hours ago   Up 2 hours   0.0.0.0:9000->9000/tcp    ganesh-sonarqube

C:\Users\ganes>

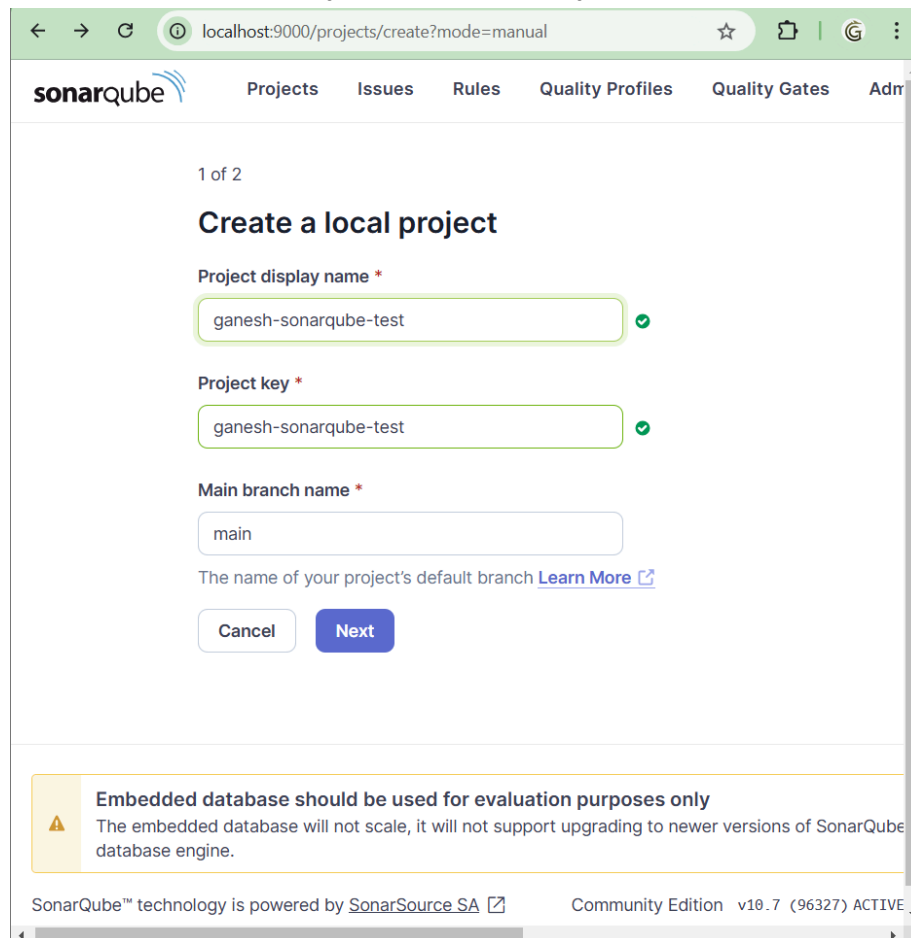
```

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



The screenshot shows the SonarQube interface with the 'Projects' tab selected. On the left, there are filters for 'Quality Gate' (Passed: 1, Failed: 0) and 'Security' (0 info, 0 minor, 0 major, 0 critical, 0 blocker issues). The main area displays a list of projects. The first project, 'ganesh-sonarqube', is marked as 'PUBLIC' and 'Passed'. Below it, a message states 'Project's Main Branch is not analyzed yet. [Configure analysis](#)'. The second project, 'sonarqube', is also marked as 'PUBLIC' and 'Passed', with a note 'Last analysis: 2 hours ago' and 'The main branch of this project is empty.' A footer message reads 'Embedded database should be used for evaluation purposes only'.

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



The screenshot shows the 'Create a local project' form in SonarQube. The form has three main sections: 'Project display name', 'Project key', and 'Main branch name'. Each section has a text input field and a green checkmark indicating the input is valid. The 'Project display name' and 'Project key' fields both contain 'ganesh-sonarqube-test'. The 'Main branch name' field contains 'main'. Below the form, there is a 'Cancel' button and a 'Next' button. A footer message reads 'Embedded database should be used for evaluation purposes only'. The bottom of the page shows 'SonarQube™ technology is powered by SonarSource SA' and 'Community Edition v10.7 (96327) ACTIVE'.

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

localhost:9000/projects/create?mode=manual&setnacd=true

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

☐ 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](#)

In the top-right corner of the page, click on your user profile icon (or the initial of your username).

localhost:9000/projects

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

My Favorites All

Filters

Quality Gate

Passed 1
Failed 0

Security

A ≥ 0 Info issues 1
B ≥ 1 minor issue 0
C ≥ 1 major issue 0
D ≥ 1 critical issue 0
E ≥ 1 blocker issue 0

Reliability

Search for projects... Perspective Overall Status Sort by Name

☆ ganesh-sonarqube PUBLIC
Project's Main Branch is not analyzed yet. [Configure analysis](#)

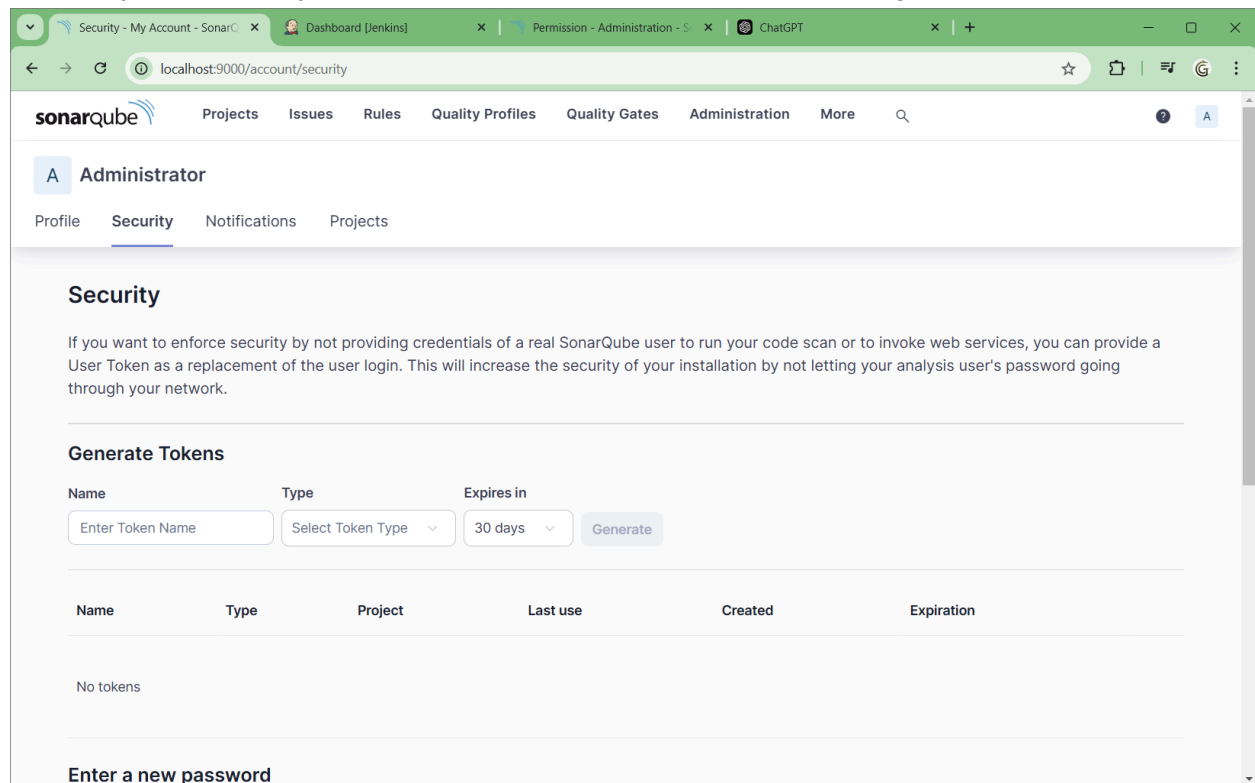
☆ ganesh-sonarqube-test PUBLIC
Project's Main Branch is not analyzed yet.

☆ sonarqube PUBLIC Passed

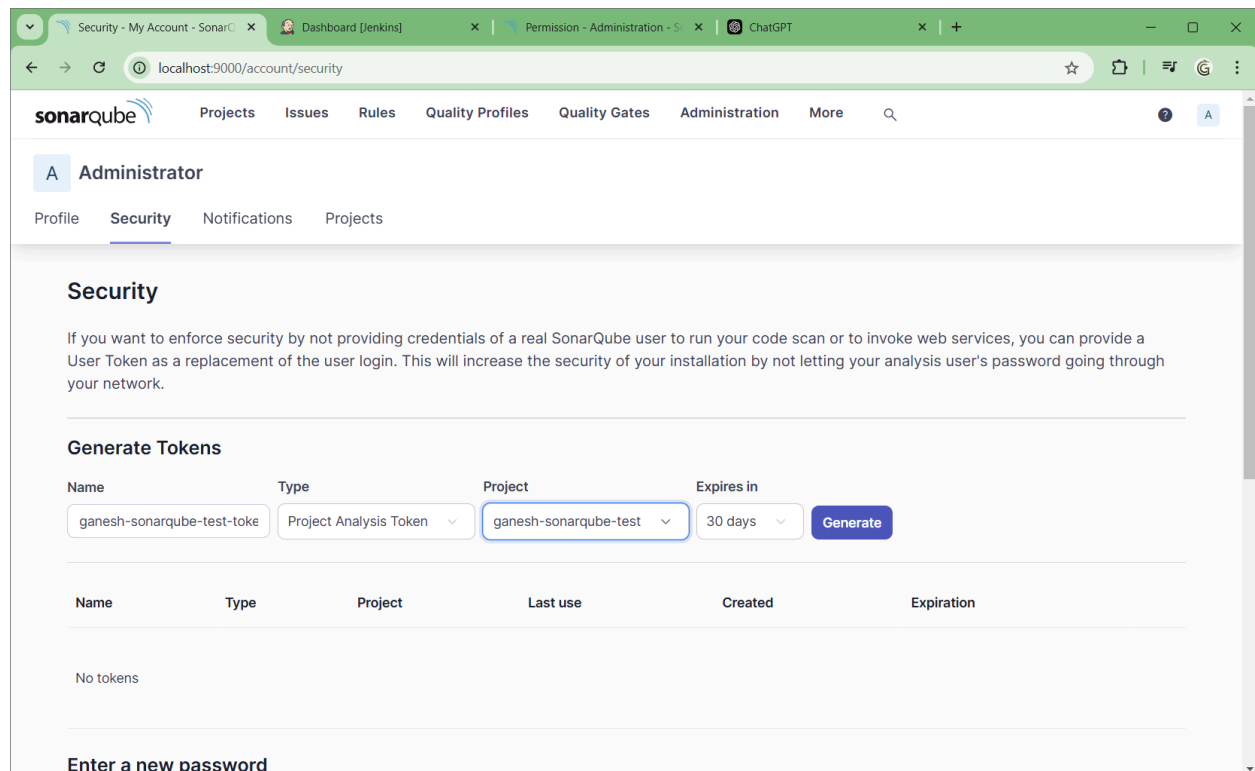
Administrator
My Account
Log out

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.

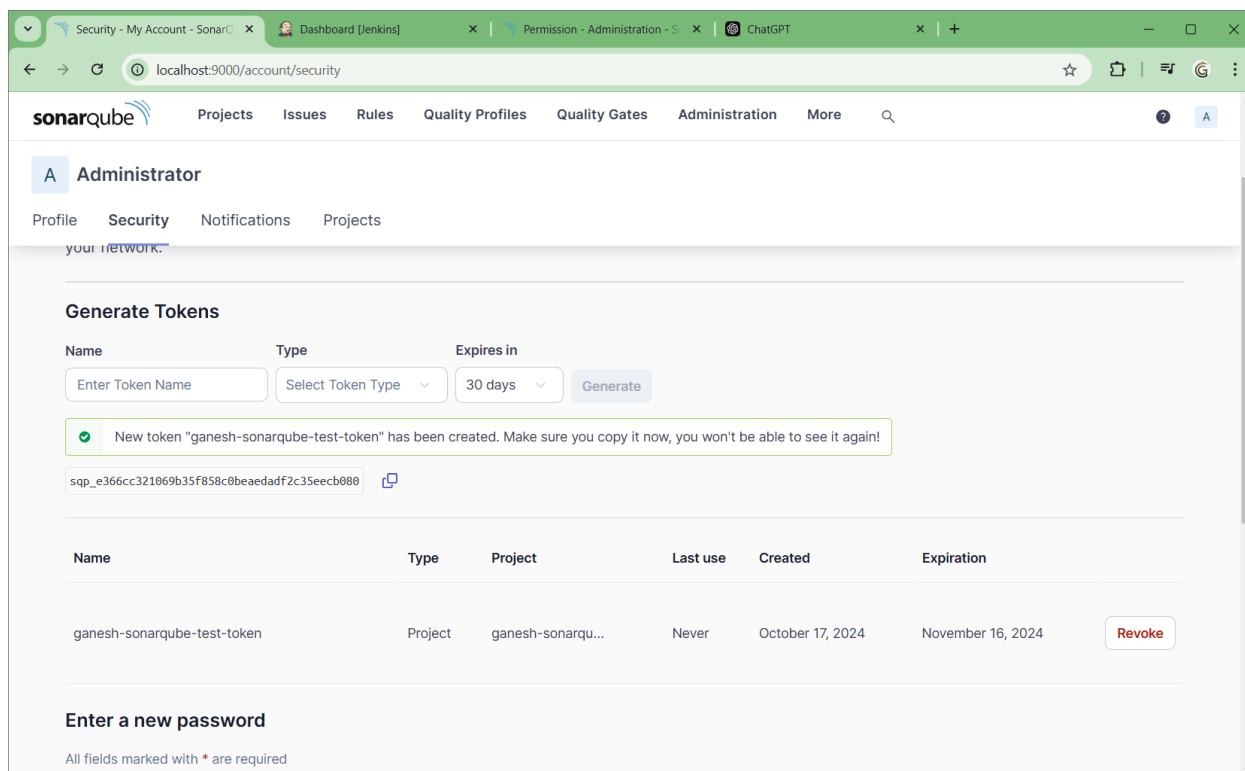
Select My Account. In your account dashboard, click on the **"Security"** tab.



Under "Generate Tokens", you'll see a field to create a new token. Enter a name for your token (e.g., sonarqube-test-token). Choose "Project analysis" as the token type. Click the "Generate" button.



Keep these credentials handy as you'll need them for Jenkins configuration.



The screenshot shows the SonarQube Security - My Account page. The user is logged in as Administrator. The page has tabs for Profile, Security, Notifications, and Projects. The Security tab is active, showing a 'Generate Tokens' section. A new token 'ganesh-sonarqube-test-token' has been created and is displayed in a box. Below the token, there is a table with columns: Name, Type, Project, Last use, Created, and Expiration. The table contains one row for the token 'ganesh-sonarqube-test-token' with Type 'Project', Project 'ganesh-sonarqu...', Last use 'Never', Created 'October 17, 2024', and Expiration 'November 16, 2024'. There is a 'Revoke' button next to the row. Below the table, there is a section 'Enter a new password' with a note 'All fields marked with * are required'.

Security - My Account - SonarQube x Dashboard [Jenkins] x Permission - Administration - S x ChatGPT x +

localhost:9000/account/security

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration More

A Administrator

Profile Security Notifications Projects

your network.

Generate Tokens

Name Type Expires in

Enter Token Name Select Token Type 30 days Generate

✓ New token "ganesh-sonarqube-test-token" has been created. Make sure you copy it now, you won't be able to see it again!

sqp_e366cc321069b35f858c0beaedaf2c35eecb080

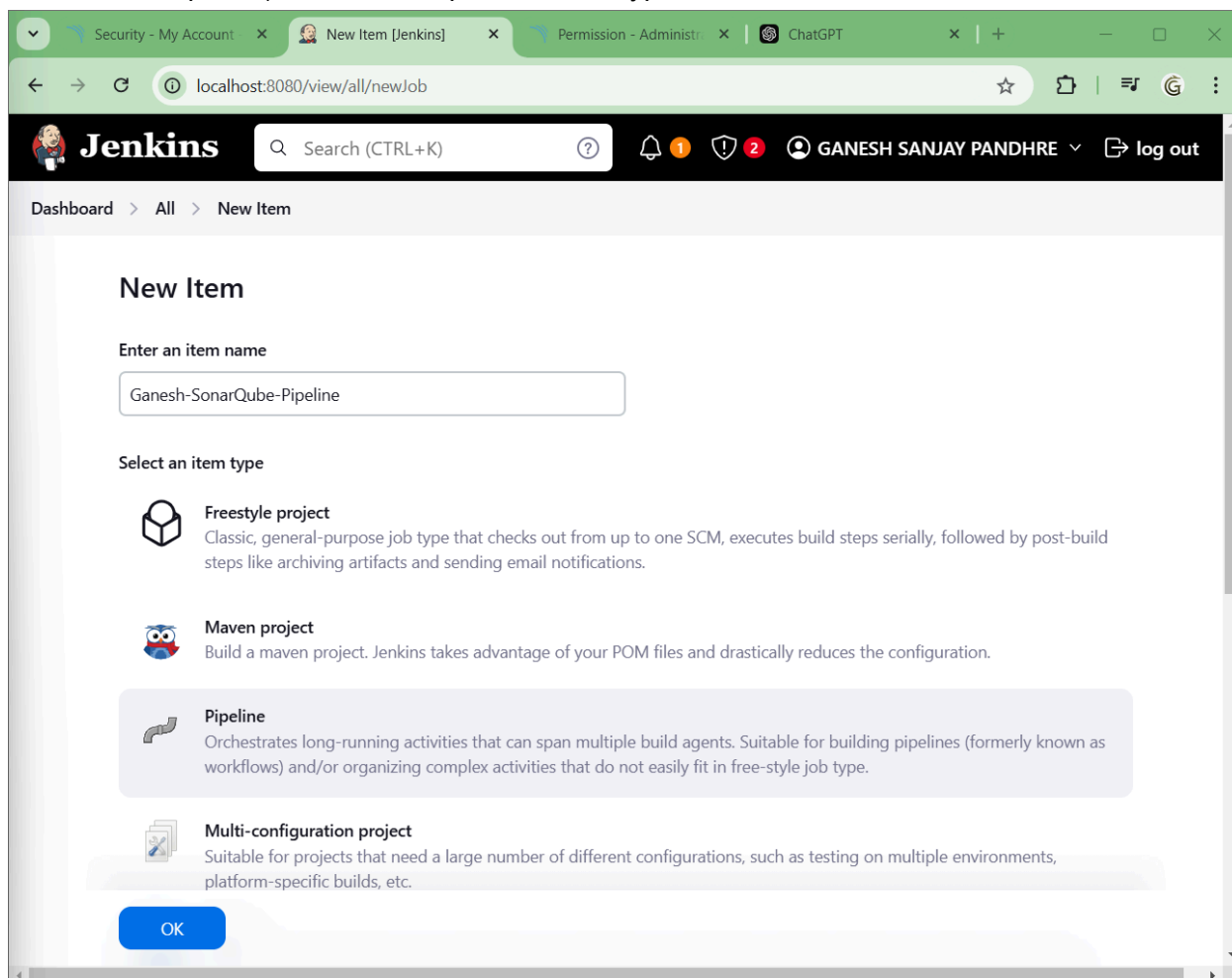
Name	Type	Project	Last use	Created	Expiration
ganesh-sonarqube-test-token	Project	ganesh-sonarqu...	Never	October 17, 2024	November 16, 2024

Revoke

Enter a new password

All fields marked with * are required

Go back to Jenkins and click on "New Item" in the top left corner. Enter a name (e.g., SonarQube-Pipeline) and select Pipeline as the type. Click OK to create the item.



The screenshot shows the Jenkins 'New Item' page. The user is logged in as GANESH SANJAY PANDHRE. The page has a search bar and a 'log out' button. The 'New Item' section is active, showing a form to enter an item name and select an item type. The item name 'Ganesh-SonarQube-Pipeline' is entered. The item type 'Pipeline' is selected. The 'Pipeline' option is highlighted in a blue box. Below the 'Pipeline' option, there is a description: '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.' Below the 'Pipeline' option, there is a 'Multi-configuration project' option with a description: 'Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.' Below the 'Multi-configuration project' option, there is an 'OK' button.

Security - My Account x New Item [Jenkins] x Permission - Administration - S x ChatGPT x +

localhost:8080/view/all/newJob

Jenkins Search (CTRL+K) ?


Dashboard > All > New Item


New Item


Enter an item name


Ganesh-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.

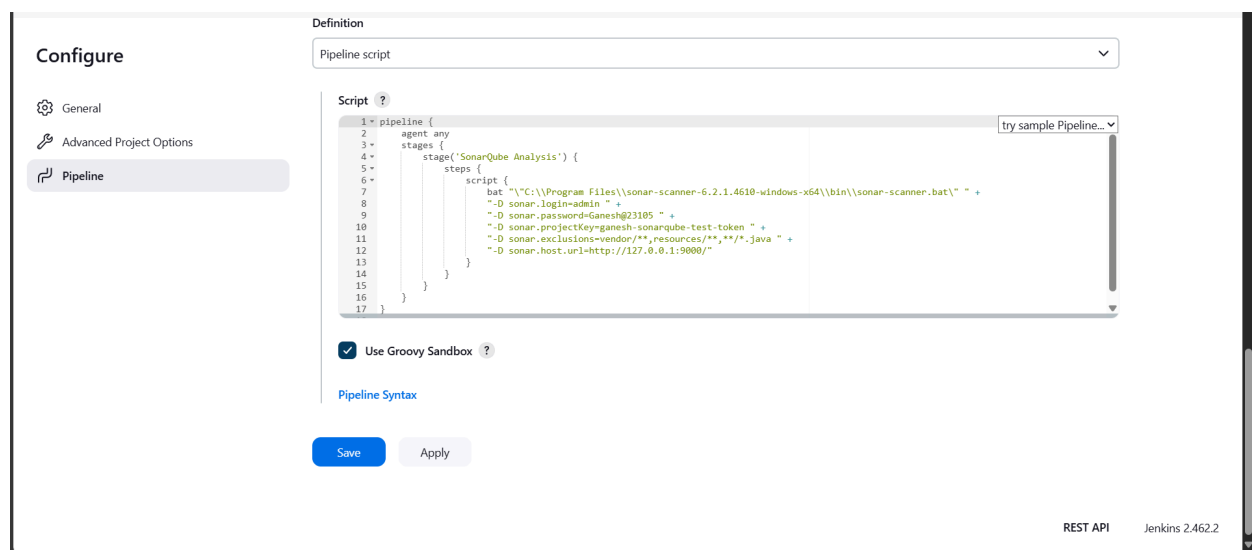
 **Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

OK

In the pipeline script section, you will define stages like cloning the GitHub repository and running the SonarQube analysis. Use the following script:

```
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.

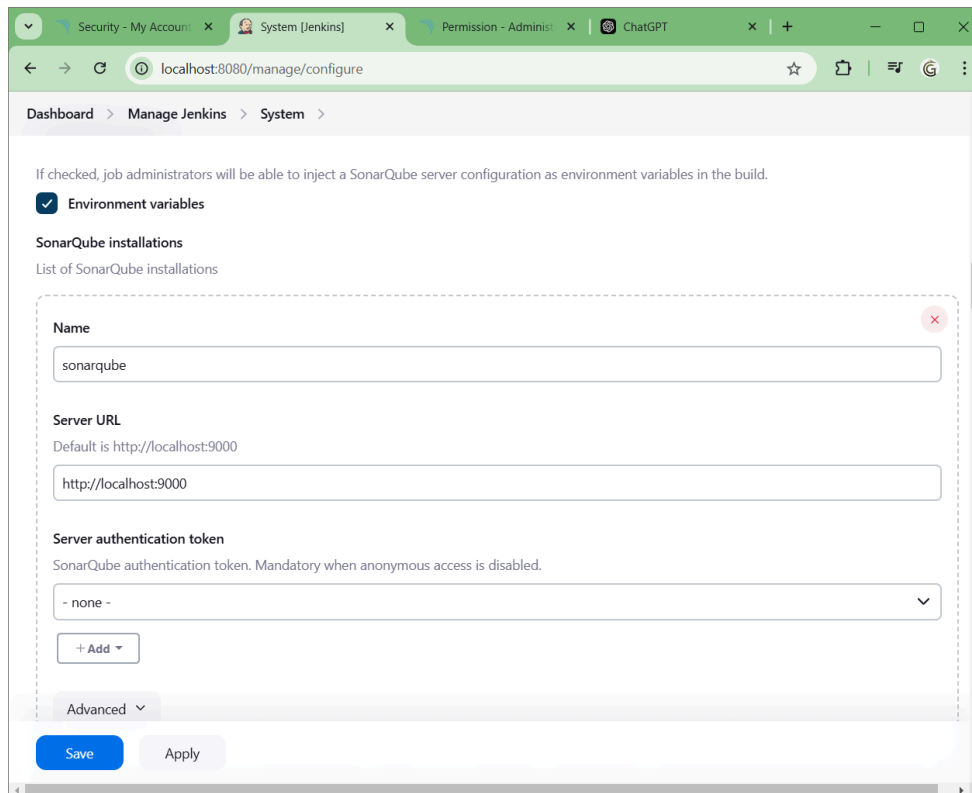
Scroll down to the SonarQube servers section.

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.



The screenshot shows the Jenkins 'System' configuration page. The 'Environment variables' checkbox is checked. Under 'SonarQube installations', there is a list of installations. A new installation is being added with the following details:

- Name:** sonarqube
- Server URL:** http://localhost:9000
- Server authentication token:** - none -

At the bottom, there are 'Save' and 'Apply' buttons.

In the popup, select Jenkins > Secret text.

Enter your SonarQube authentication token (the token you generated in SonarQube).

Save it with a recognizable name (e.g., sonarqube-token).

Jenkins Credentials Provider: Jenkins

Add Credentials

Domain

Global credentials (unrestricted)

Kind

Secret text

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

Secret

.....

ID ?

ganesh-sonarqube-test-token

Description ?

Once added, select it from the dropdown menu.

Dashboard > Manage Jenkins > System >

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

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

+ Add

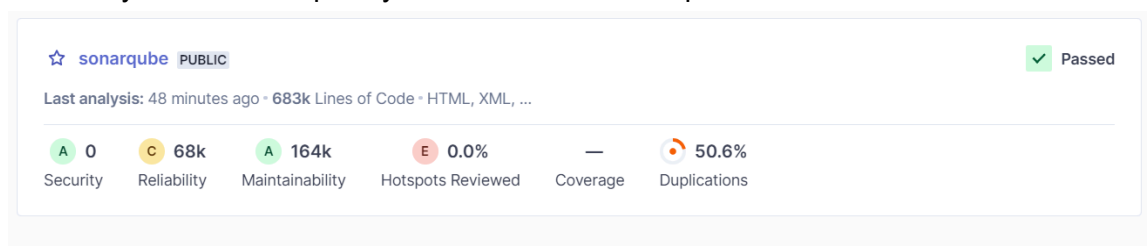
Save Apply

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 / main 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 main 683k Lines of Code Version not provided Set as homepage Take the Tour

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

sonarqube / main main 683k Lines of Code Version not provided Set as homepage Take the Tour

Overview Issues Security Hotspots Measures Code **Activity** Project Settings Project 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 New Code

Issues

200k
150k
100k
50k
0

03 PM 03:30 04 PM 04:30 05 PM

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.