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.	53
Name	Aryan Deepak Saraf
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.

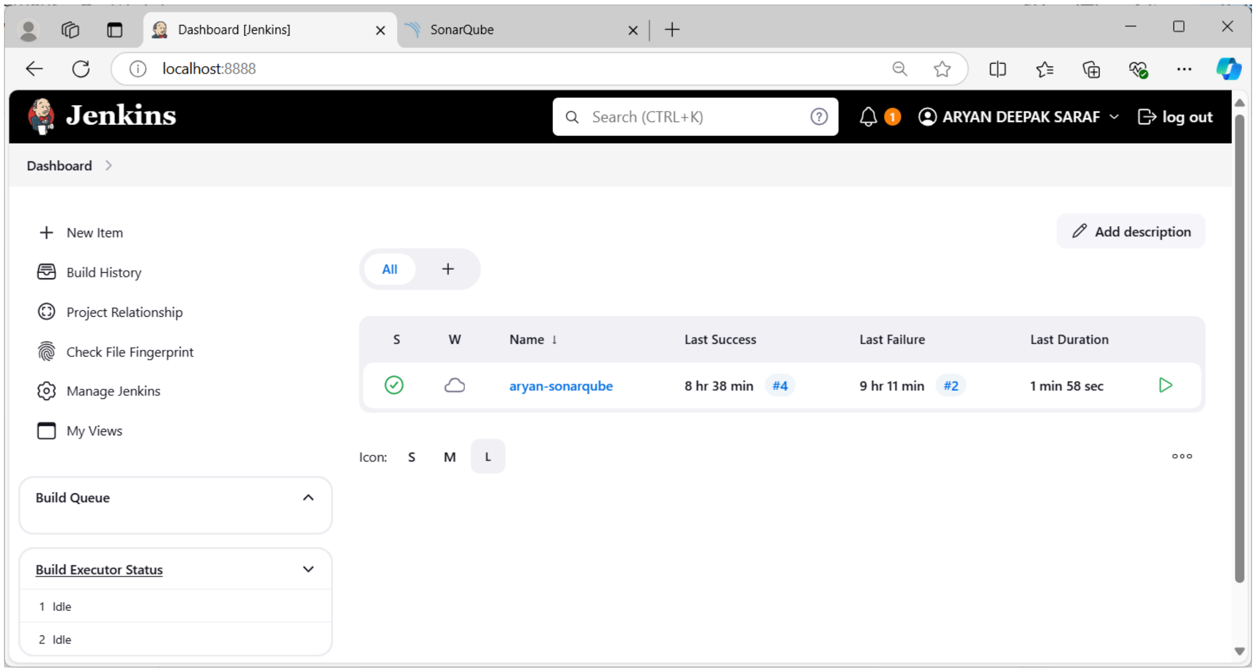
Importance of Code Quality Analysis:
Using SonarQube in the CI/CD pipeline helps developers identify and fix issues before code is deployed. This proactive approach saves time and resources, improves application quality, and enhances security by addressing vulnerabilities early in development.

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.

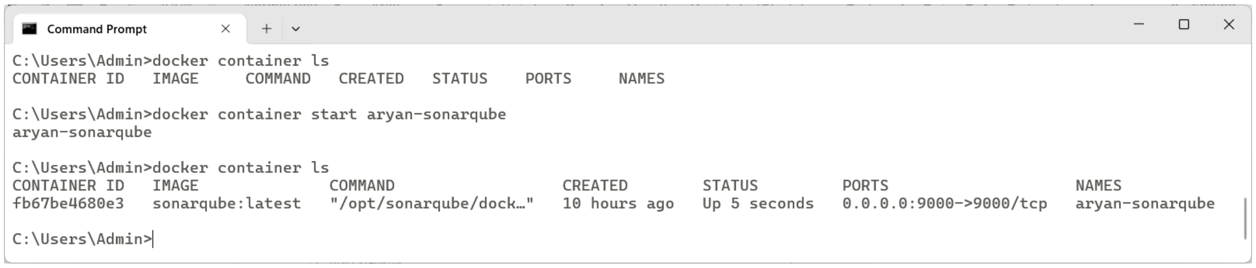
Enhanced Developer Skills: Regular feedback helps developers improve their coding practices and fosters continuous learning.

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



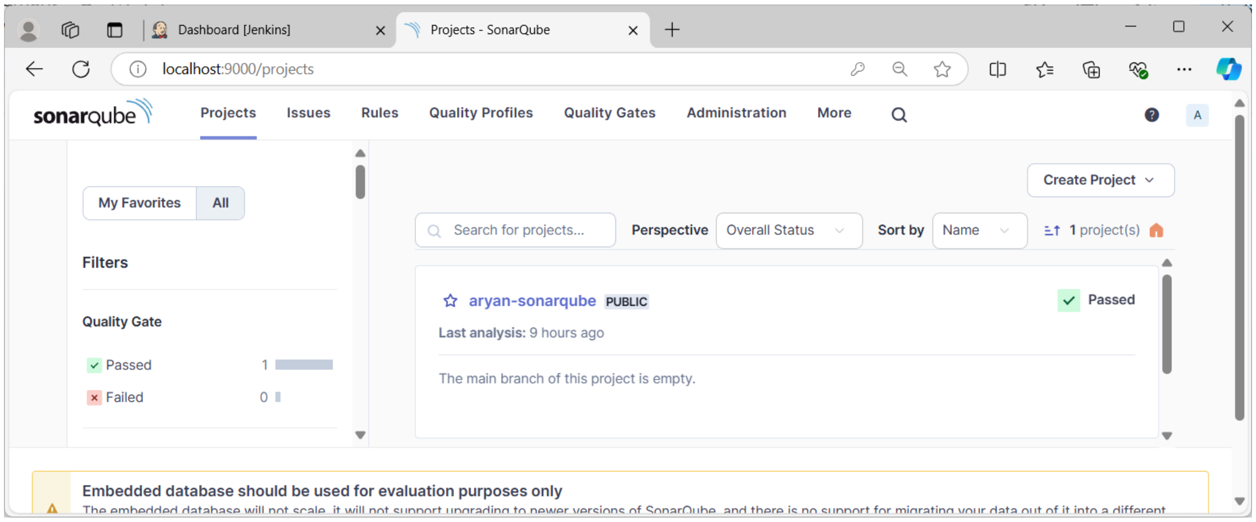
S	W	Name ↓	Last Success	Last Failure	Last Duration
✓		aryan-sonarqube	8 hr 38 min #4	9 hr 11 min #2	1 min 58 sec

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



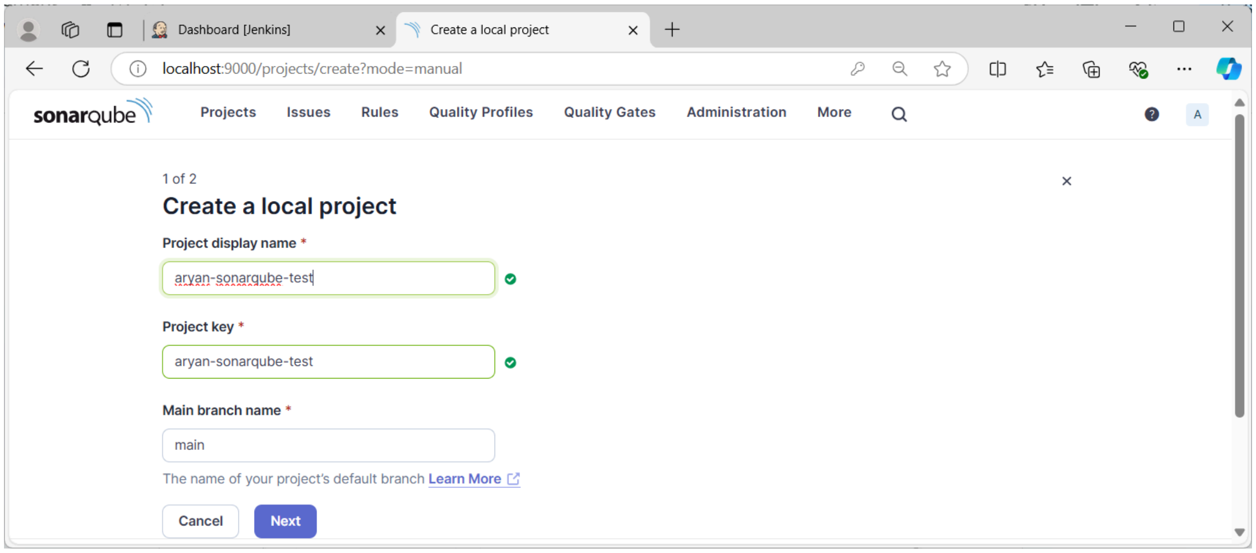
```
C:\Users\Admin>docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

C:\Users\Admin>docker container start aryan-sonarqube
aryan-sonarqube

C:\Users\Admin>docker container ls
CONTAINER ID   IMAGE              COMMAND                  CREATED        STATUS         PORTS                    NAMES
fb67be4680e3   sonarqube:latest   "/opt/sonarqube/dock…"   10 hours ago   Up 5 seconds   0.0.0.0:9000->9000/tcp   aryan-sonarqube

C:\Users\Admin>
```

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



Embedded database should be used for evaluation purposes only

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



localhost:9000/projects/create?mode=manual

1 of 2

Create a local project

Project display name *

aryan-sonarqube-test

Project key *

aryan-sonarqube-test

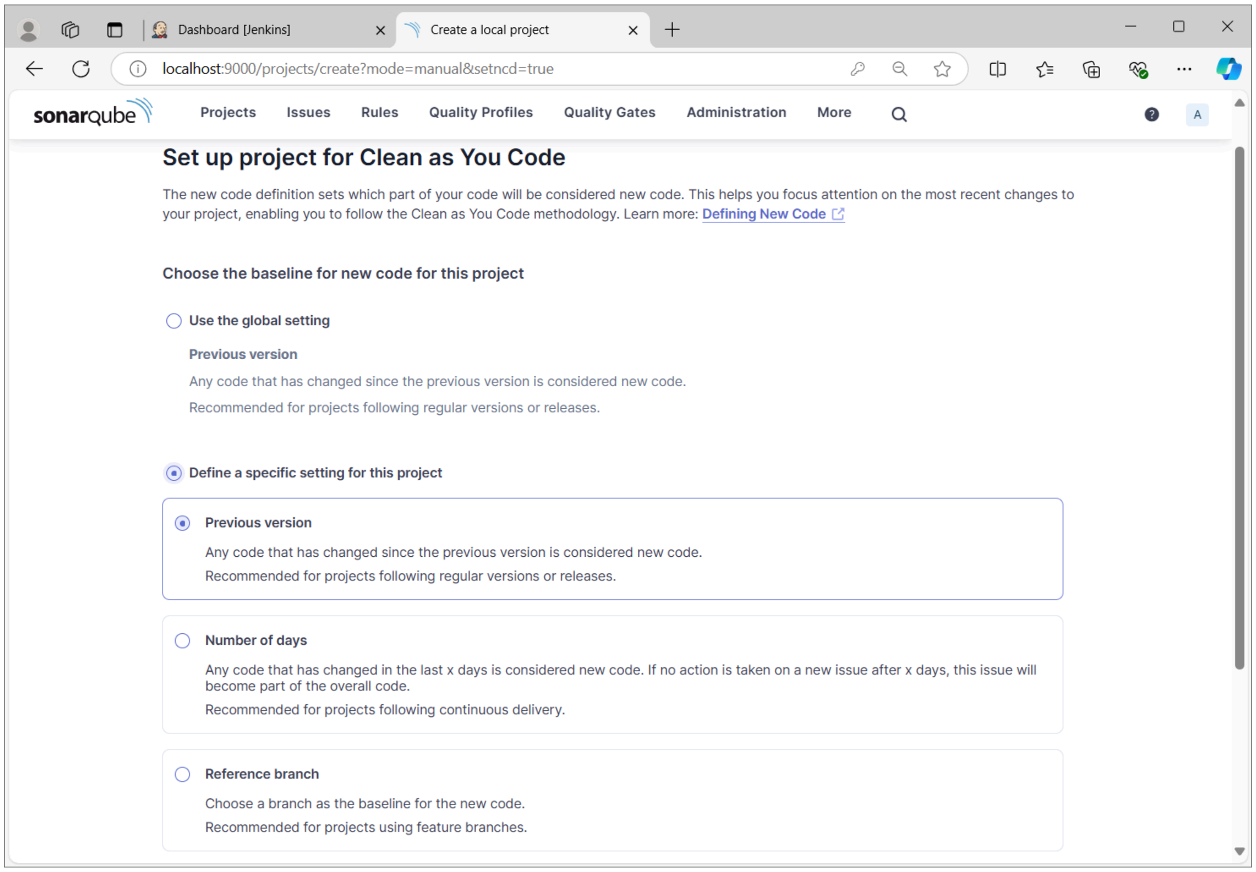
Main branch name *

main

The name of your project's default branch Learn More

Cancel Next

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



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

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.

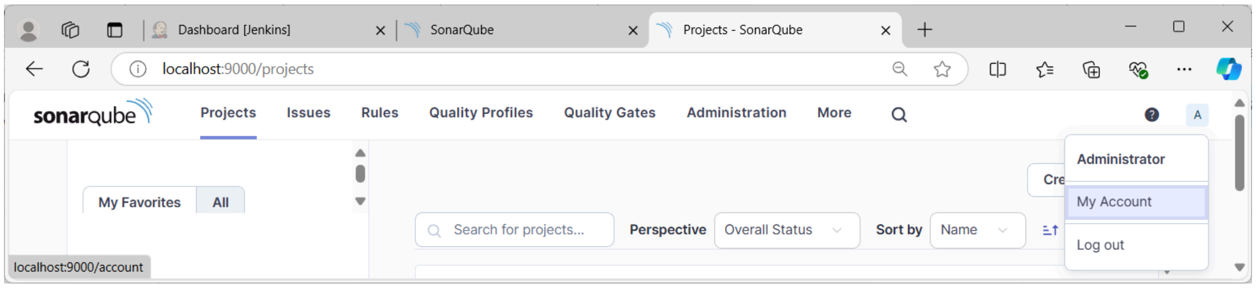
Recommended for projects following continuous delivery.

( ) Reference branch

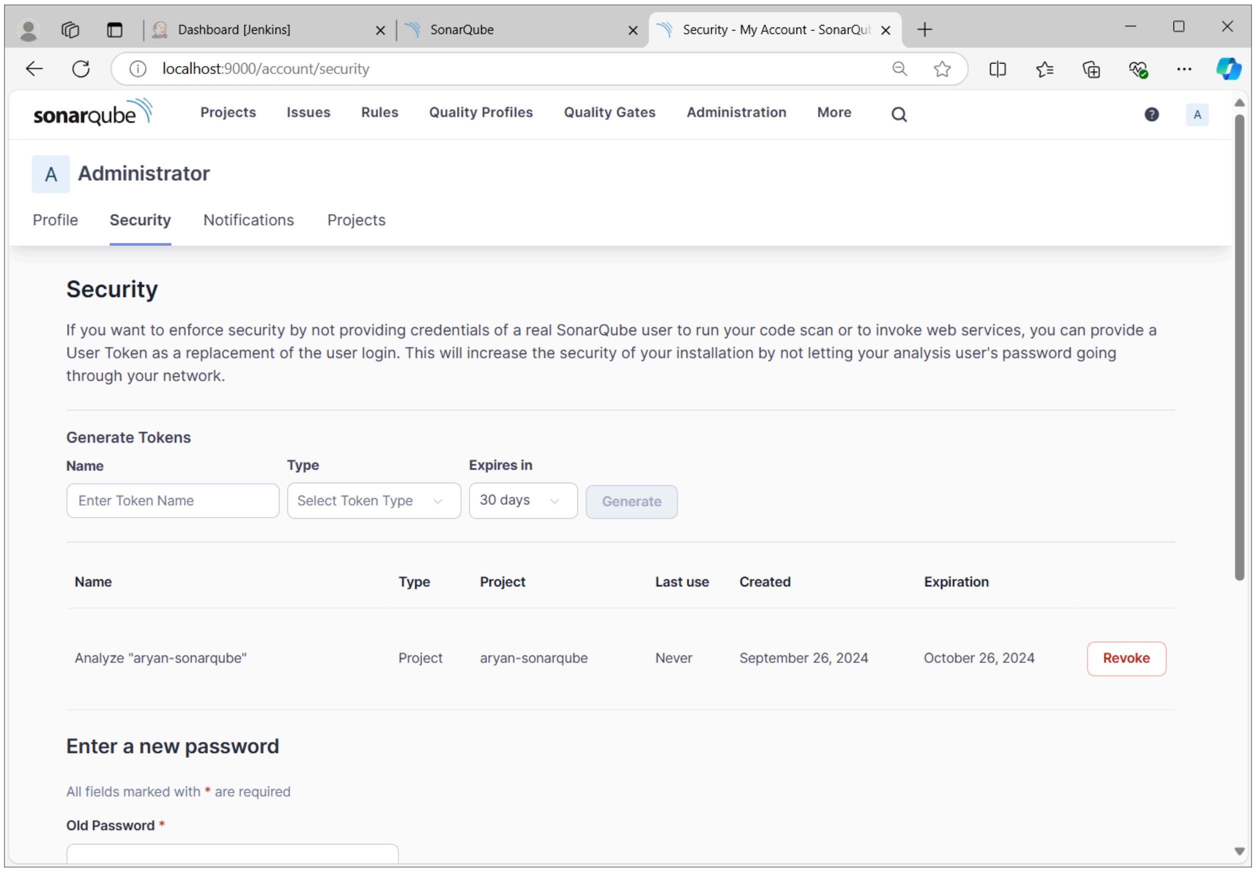
Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

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



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


Dashboard [Jenkins]
SonarQube
Security - My Account - SonarQube
localhost:9000/account/security
Projects Issues Rules Quality Profiles Quality Gates Administration More
Administrator
Profile Security Notifications Projects

Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Expires in	
Enter Token Name	Select Token Type	30 days	Generate

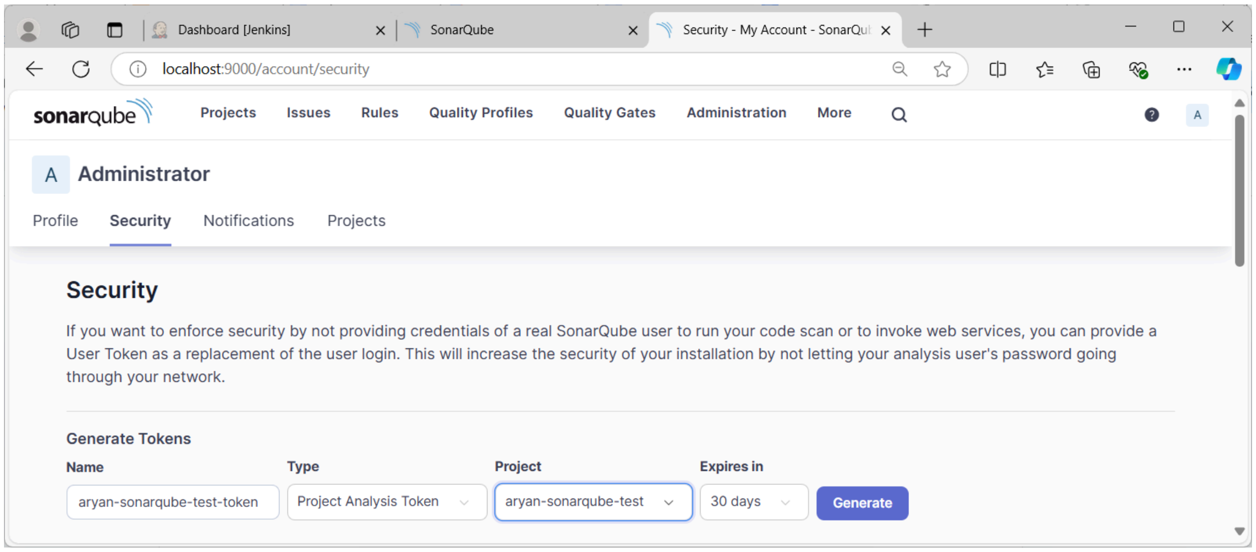
Name	Type	Project	Last use	Created	Expiration	
Analyze "aryan-sonarqube"	Project	aryan-sonarqube	Never	September 26, 2024	October 26, 2024	Revoke

Enter a new password

All fields marked with * are required

Old Password *

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.


Dashboard [Jenkins]
SonarQube
Security - My Account - SonarQube
localhost:9000/account/security
Projects Issues Rules Quality Profiles Quality Gates Administration More
Administrator
Profile Security Notifications Projects

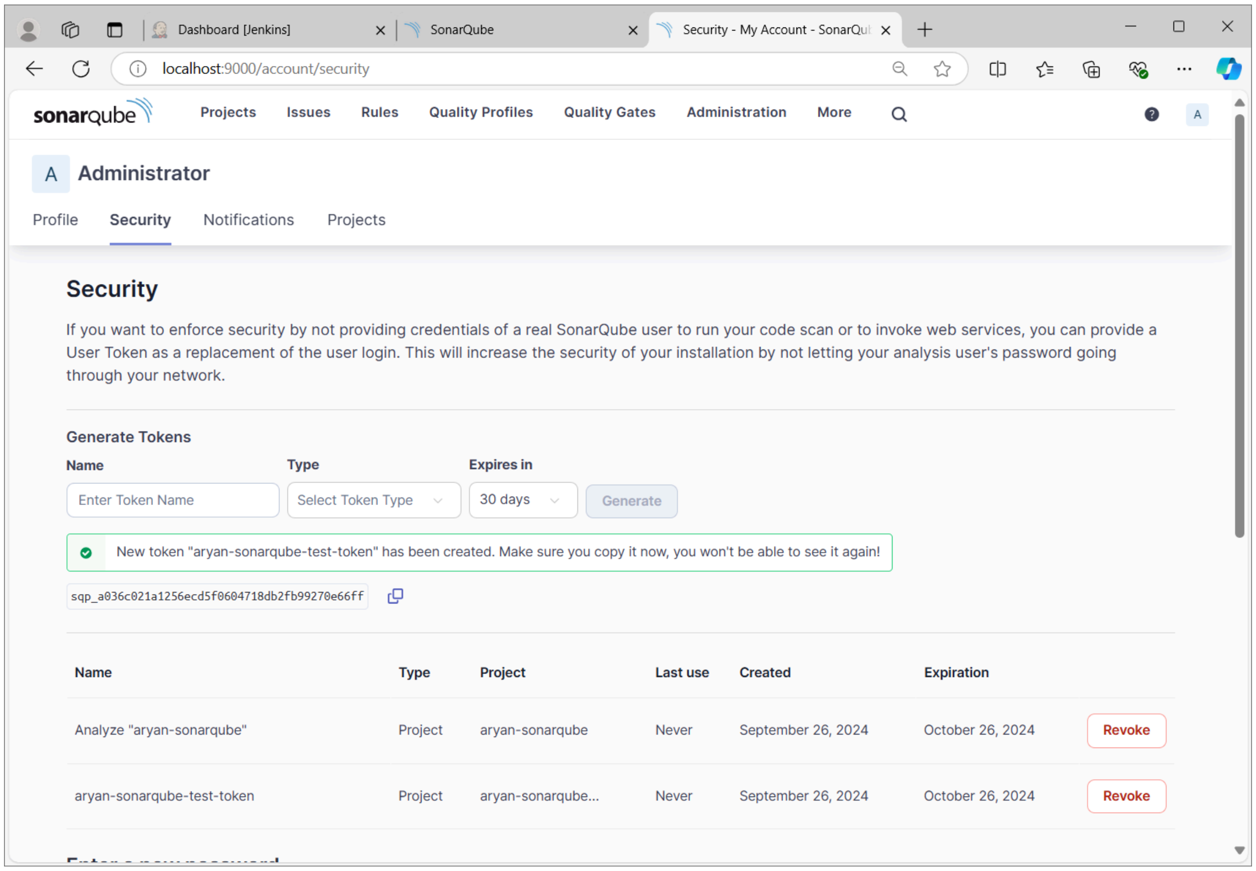
Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Name	Type	Project	Expires in	
aryan-sonarqube-test-token	Project Analysis Token	aryan-sonarqube-test	30 days	Generate

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



localhost:9000/account/security

A Administrator

Profile Security Notifications Projects

Security

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

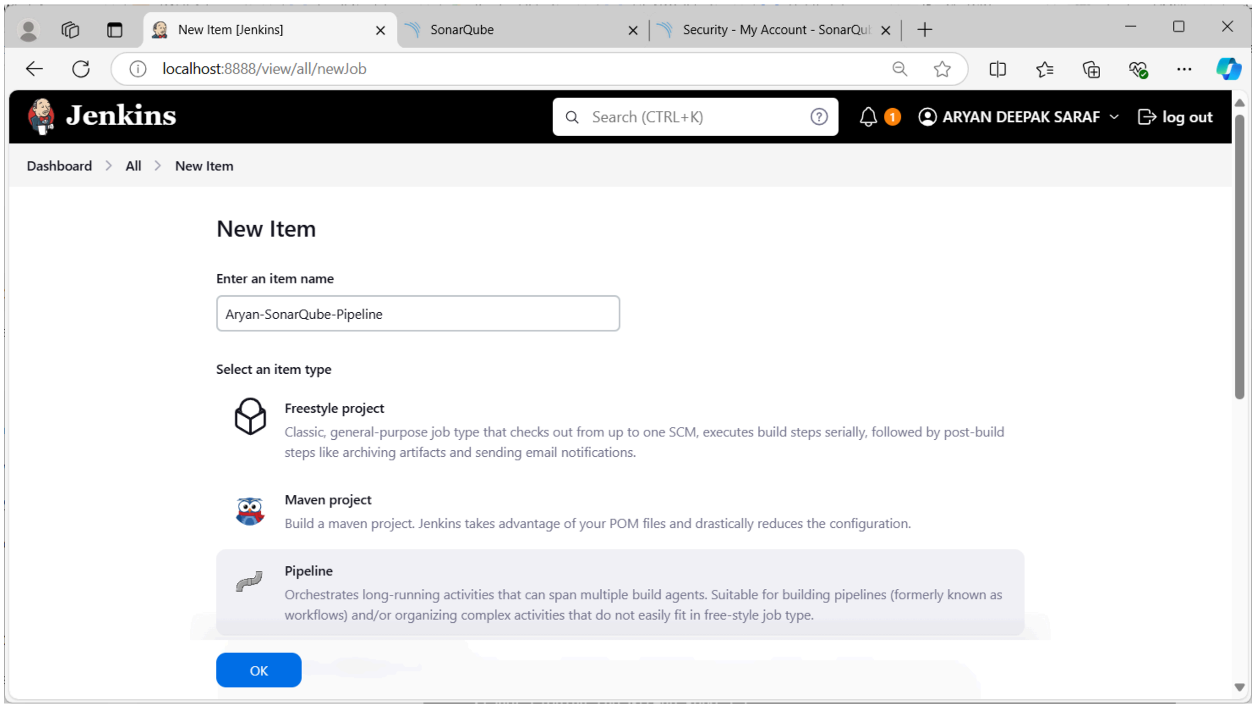
Name: Enter Token Name | Type: Select Token Type | Expires in: 30 days | Generate

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

```
sqp_a036c021a1256ecd5f0604718db2fb99270e66ff
```

Name	Type	Project	Last use	Created	Expiration	
Analyze "aryan-sonarqube"	Project	aryan-sonarqube	Never	September 26, 2024	October 26, 2024	Revoke
aryan-sonarqube-test-token	Project	aryan-sonarqube...	Never	September 26, 2024	October 26, 2024	Revoke

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.



localhost:8888/view/all/newJob

Dashboard > All > New Item

New Item

Enter an item name: Aryan-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.

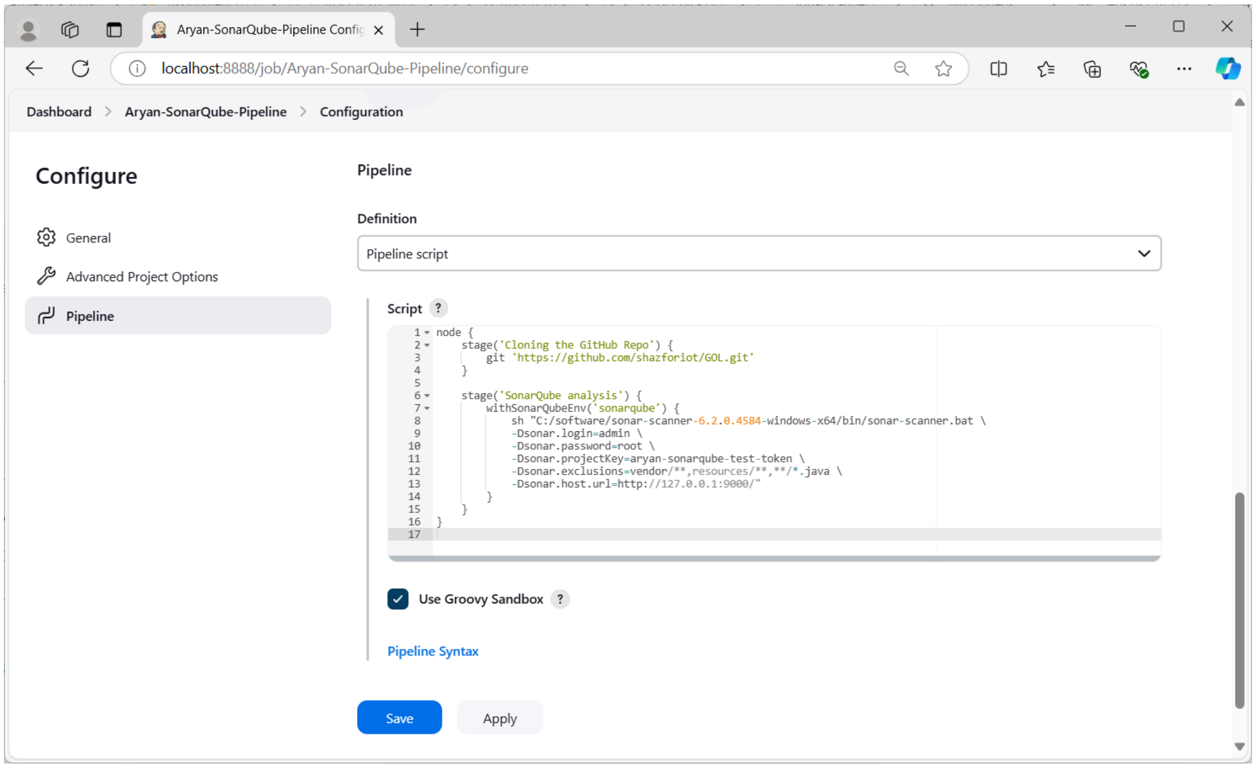
Maven project Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

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.

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/"
        }
    }
}
```


Aryan-SonarQube-Pipeline Config
localhost:8888/job/Aryan-SonarQube-Pipeline/configure
Dashboard > Aryan-SonarQube-Pipeline > Configuration
Configure
General
Advanced Project Options
Pipeline
Save
Apply
Pipeline Syntax

Pipeline

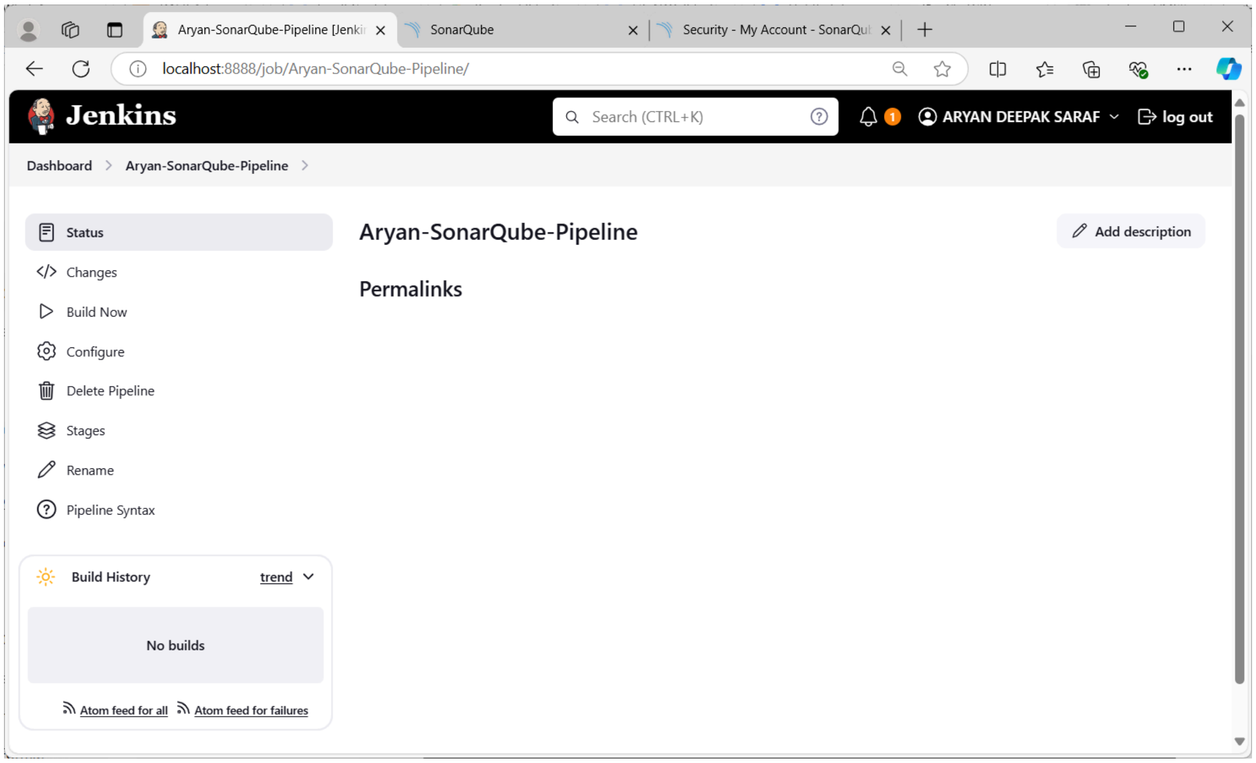
Definition: Pipeline script

Script

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

    stage('SonarQube analysis') {
        withSonarQubeEnv('sonarqube') {
            sh "C:/software/sonar-scanner-6.2.0.4584-windows-x64/bin/sonar-scanner.bat \
            -Dsonar.login=admin \
            -Dsonar.password=root \
            -Dsonar.projectKey=aryan-sonarqube-test-token \
            -Dsonar.exclusions=vendor/**,resources/**,**/*.java \
            -Dsonar.host.url=http://127.0.0.1:9000/"
        }
    }
}
```

[✓] Use Groovy Sandbox

Apply and Save.



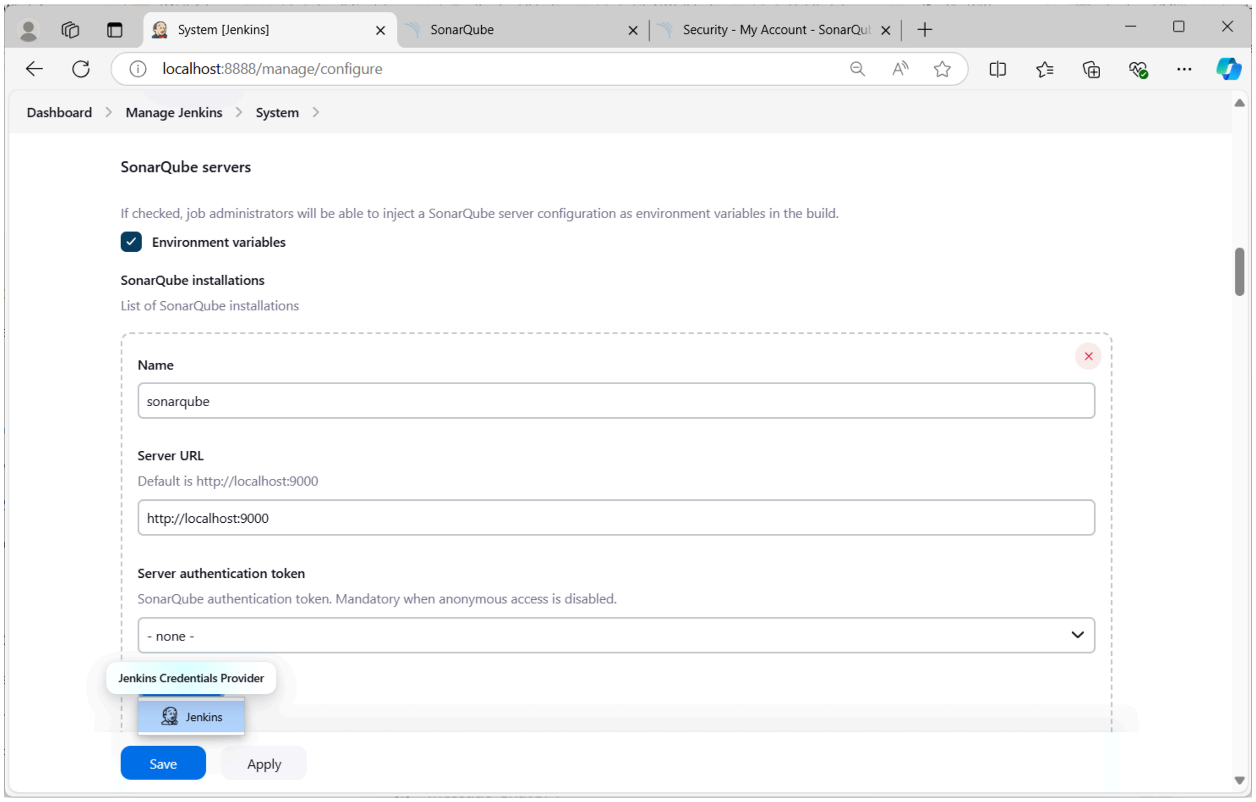
Aryan-SonarQube-Pipeline

Permalinks

No builds

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.


System [Jenkins]
SonarQube
Security - My Account - SonarQube
localhost:8888/manage/configure
Dashboard > Manage Jenkins > System >
Jenkins Credentials Provider
Jenkins
Save
Apply

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: sonarqube

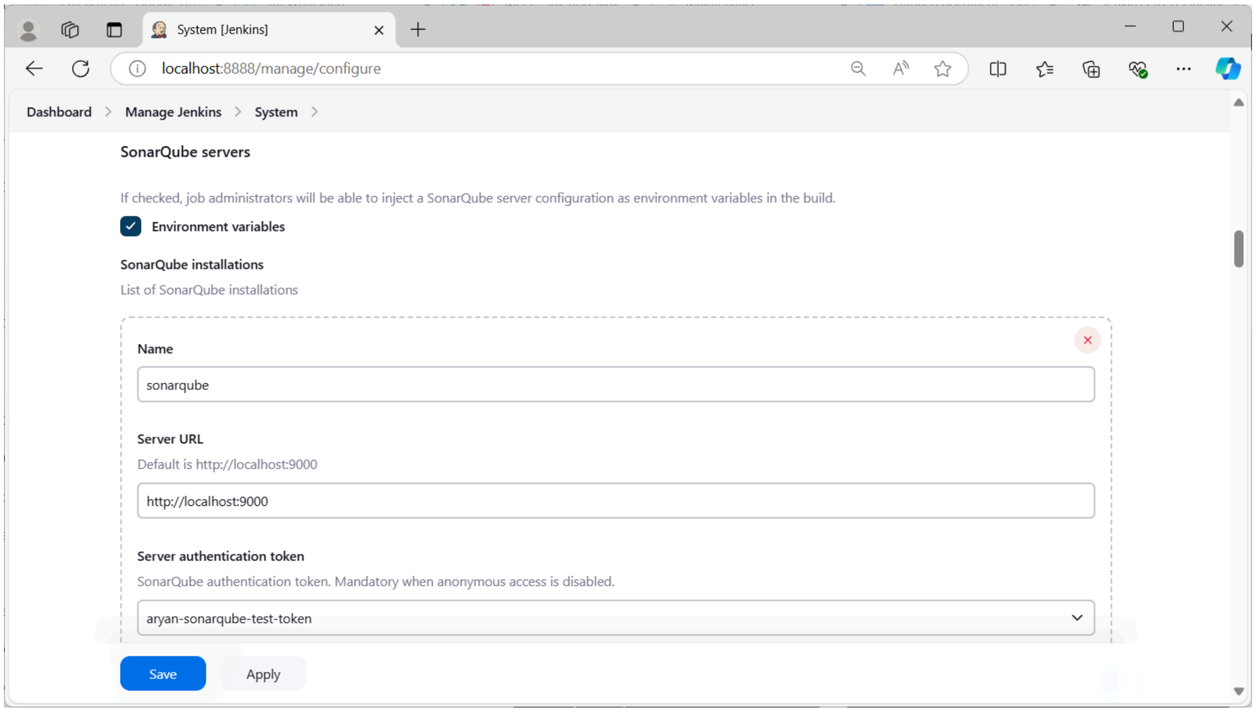
Server URL
Default is http://localhost:9000
http://localhost:9000

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
- none -

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).

Once added, select it from the dropdown menu.


System [Jenkins]
localhost:8888/manage/configure
Dashboard > Manage Jenkins > System >
Save
Apply

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: sonarqube

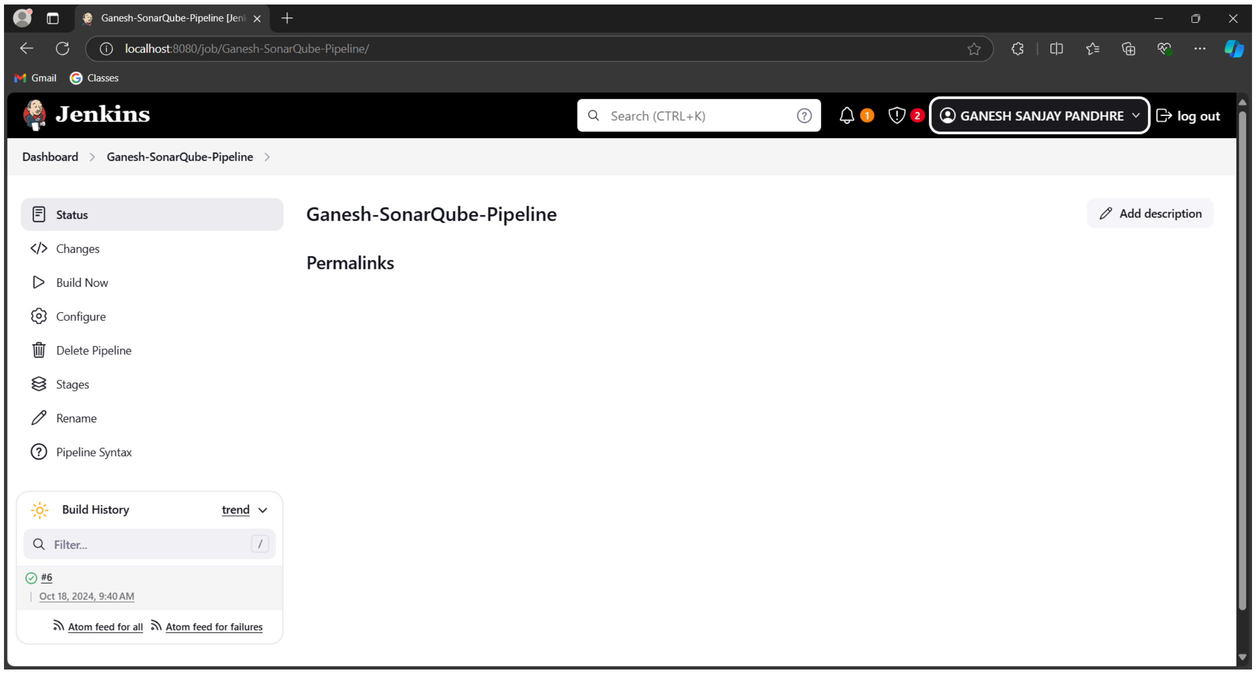
Server URL: Default is http://localhost:9000

http://localhost:9000

Server authentication token: SonarQube authentication token. Mandatory when anonymous access is disabled.

aryan-sonarqube-test-token

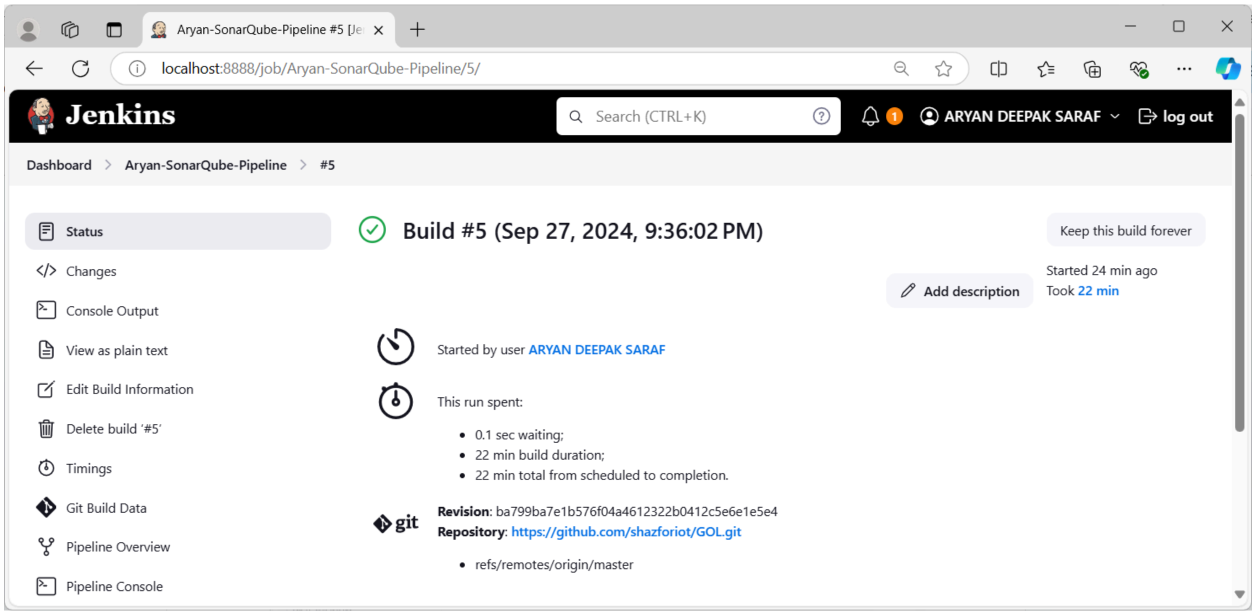
Go back to your pipeline and click Build Now to trigger the build. (The pipeline will clone the GitHub repository and run the SonarQube analysis.)



Ganesh-SonarQube-Pipeline

Permalinks

#6 Oct 18, 2024, 9:40 AM



Build #5 (Sep 27, 2024, 9:36:02 PM)

Started 24 min ago
Took 22 min

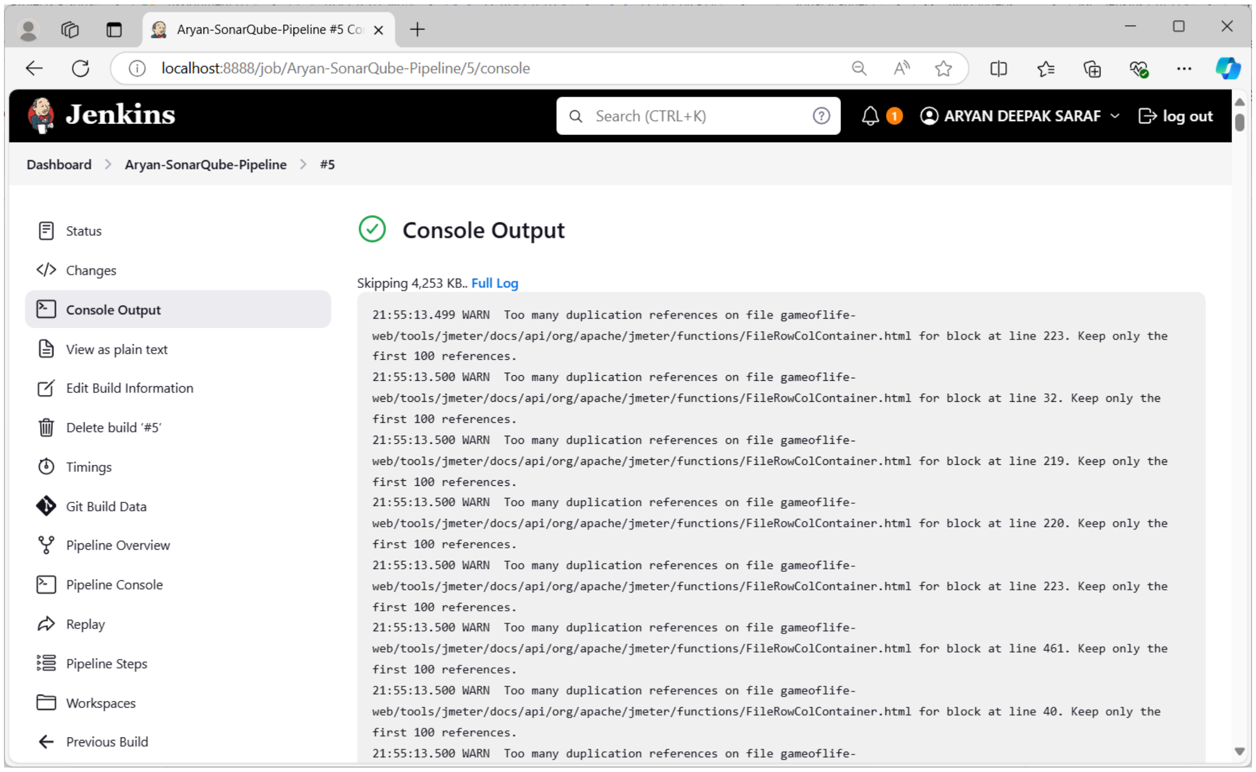
Started by user ARYAN DEEPAK SARAF

This run spent:

- 0.1 sec waiting;
- 22 min build duration;
- 22 min total from scheduled to completion.

Revision: ba799ba7e1b576f04a4612322b0412c5e6e1e5e4
Repository: https://github.com/shazforiot/GOL.git

- refs/remotes/origin/master



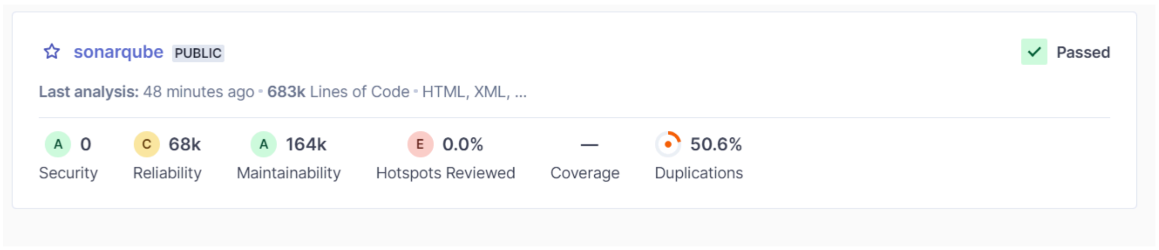
Console Output

Skipping 4,253 KB.. Full Log

```
21:55:13.499 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 223. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 32. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 219. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 220. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 223. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 461. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
web/tools/jmeter/docs/api/org/apache/jmeter/functions/FileRowColContainer.html for block at line 40. Keep only the
first 100 references.
21:55:13.500 WARN  Too many duplication references on file gameoflife-
```

Go back to SonarQube at http://localhost:9000. Open the sonarqube-test project you created earlier.
Check different tabs for issues like:

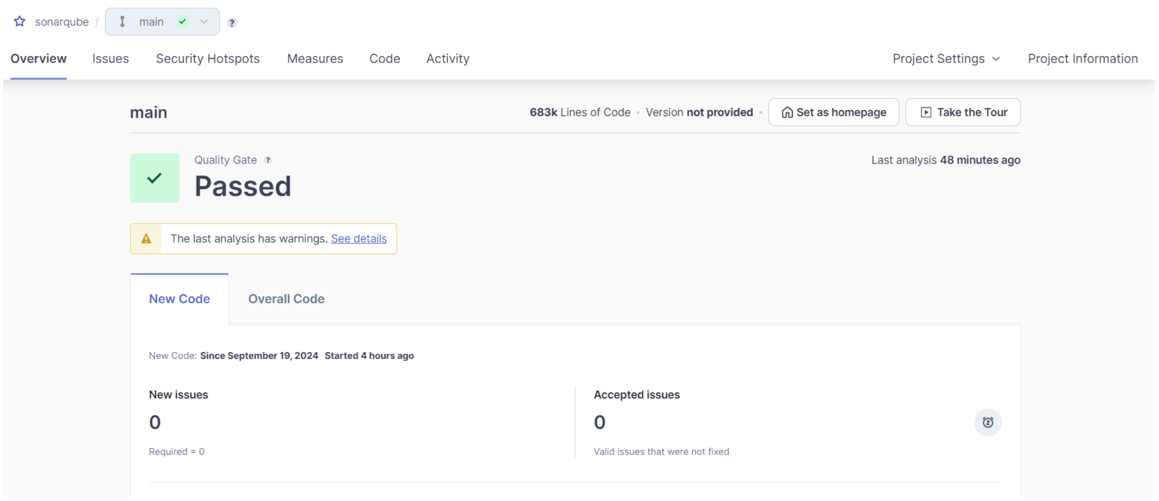
- Bugs and Code Smells: These indicate potential problems in the code.
- Unfinished TODOs: Unresolved items in the code.
- Duplicates: Repeated code blocks.
- Cyclomatic Complexity: Measure of how complex the code is.



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

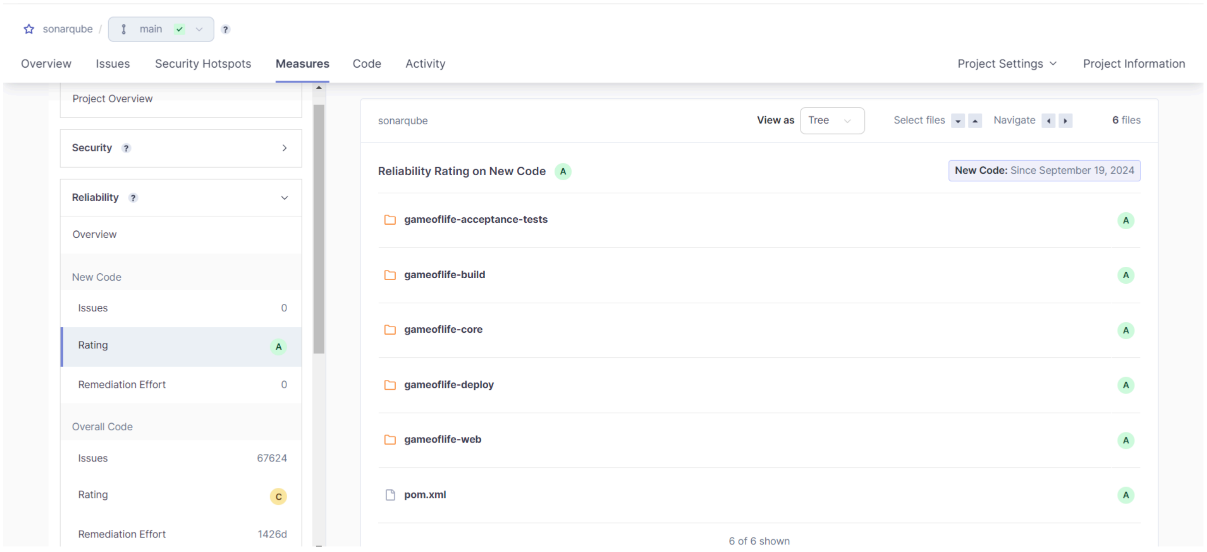
The last analysis has warnings. See details

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

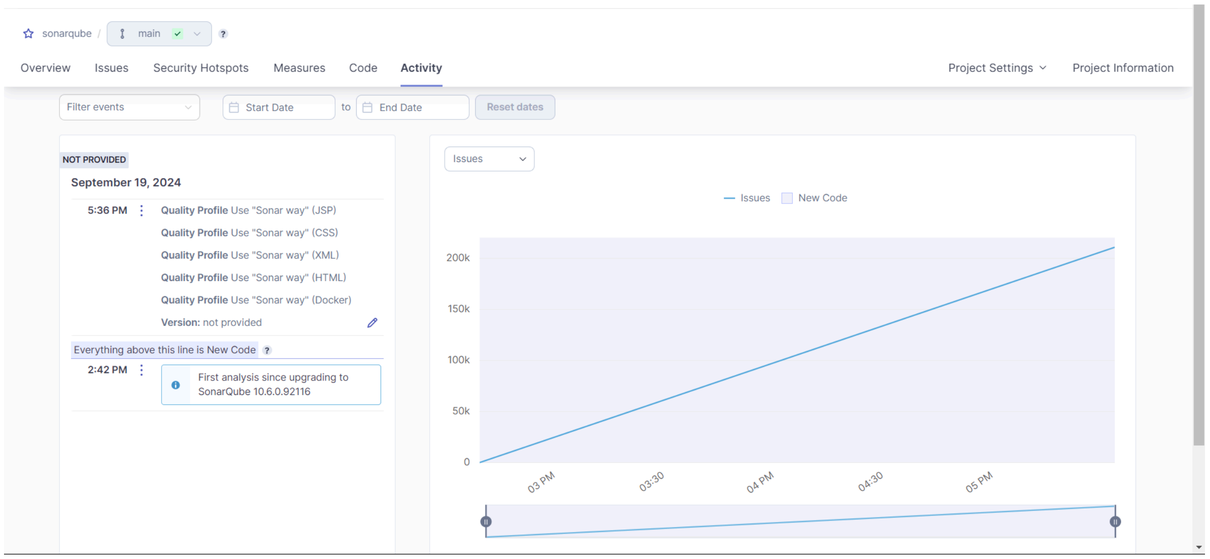
Issues	67624
Rating	C
Remediation Effort	1426d

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



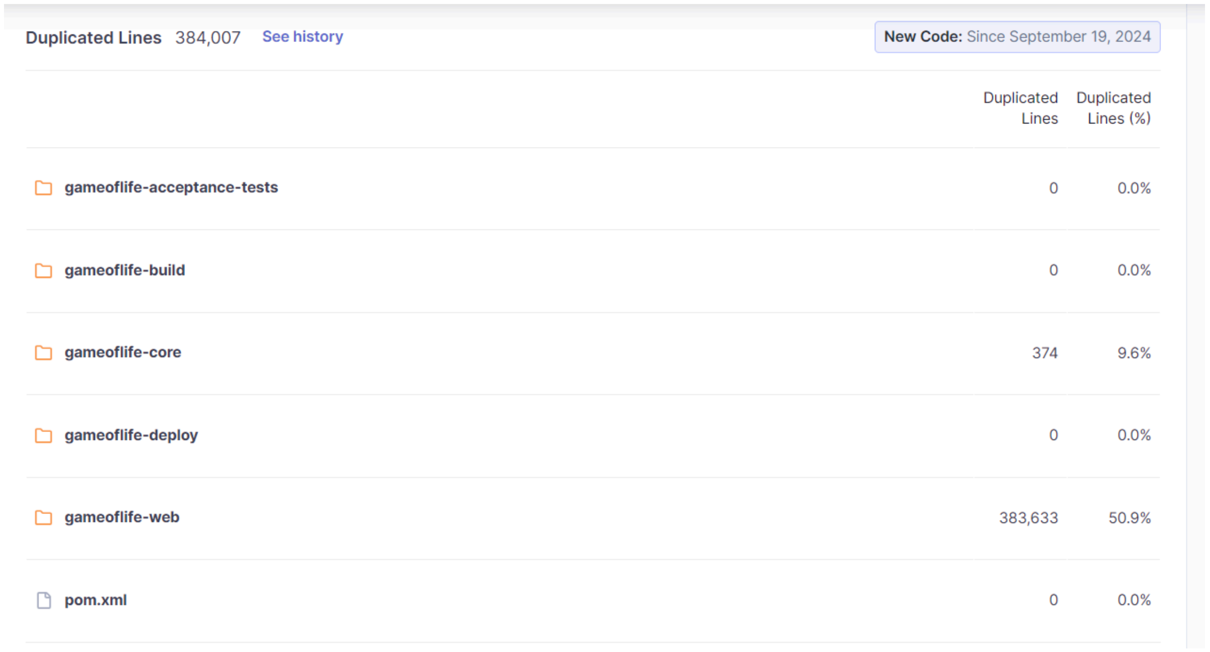
NOT PROVIDED

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



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.