

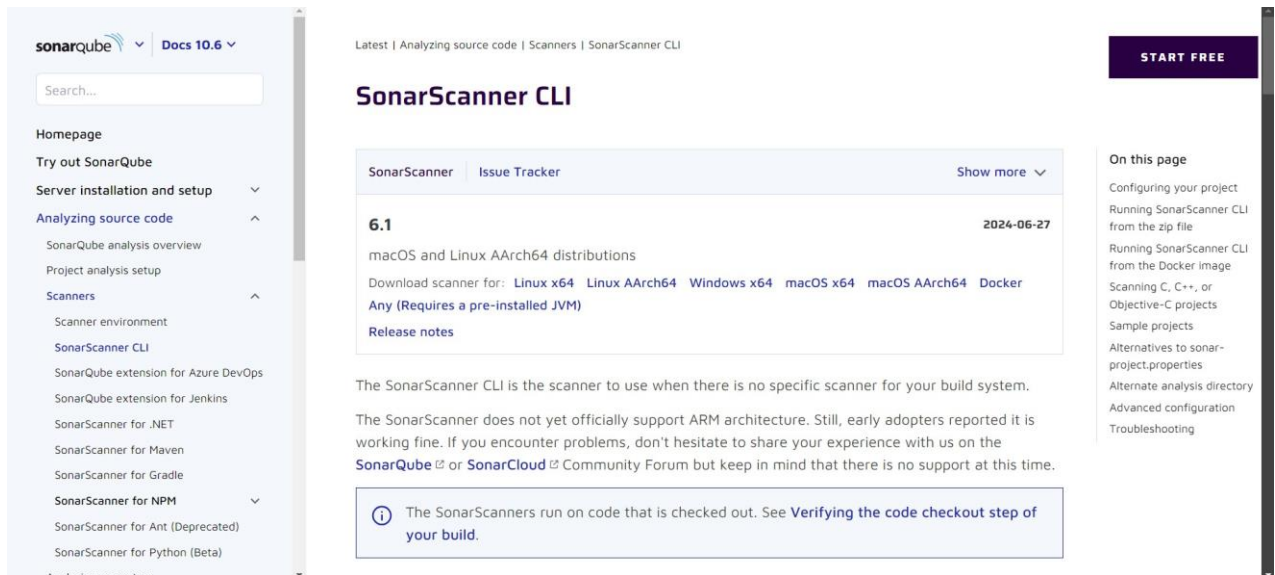
Experiment No: 8

AIM: Create a Jenkins CI/CD 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:

Step 1: Download sonar scanner

<https://docs.sonarsource.com/sonarqube/latest/analyzing-source-code/scanners/sonarscanner/> . Visit this link and download the sonarqube scanner CLI

The screenshot shows the SonarScanner CLI documentation page. On the left is a navigation menu with links like 'Homepage', 'Try out SonarQube', 'Server installation and setup', 'Analyzing source code', and 'Scanners'. The main content area is titled 'SonarScanner CLI' and shows version '6.1' with a release date of '2024-06-27'. It lists download links for various operating systems: Linux x64, Linux AArch64, Windows x64, macOS x64, macOS AArch64, and Docker. A note mentions that the scanner requires a pre-installed JVM. On the right, there is a 'START FREE' button and a list of topics covered on the page, such as 'Configuring your project', 'Running SonarScanner CLI from the zip file', and 'Running SonarScanner CLI from the Docker image'.

Extract the downloaded zip file in a folder.

Step 2: Docker Run **docker -v** command .If docker is not installed so install it

```
C:\Users\praja>docker --version
Docker version 27.0.3, build 7d4bcd8
```

Step 3: Install sonarqube image Command: **docker pull sonarqube**

```
C:\Users\Student>docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59b
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
  View a summary of image vulnerabilities and recommendations → docker

C:\Users\Student>
C:\Users\Student>
```

Step 4: Keep **Jenkins** installed on your system.

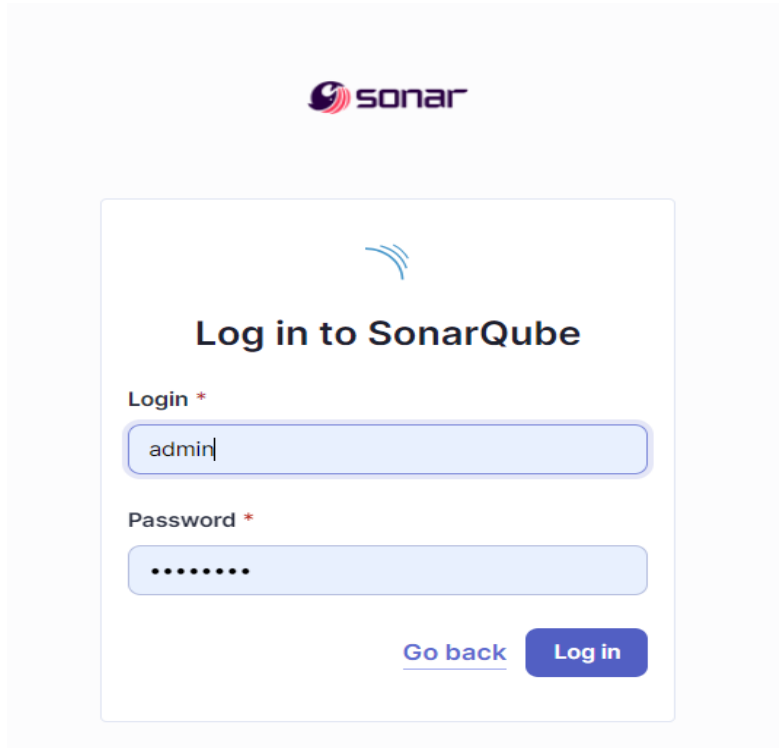
EXPERIMENT STEPS:

Step1: Run SonarQube image **docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest** .This command will run the SonarQube image that was just installed using docker.

```
C:\Users\Student>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
83330c33cd961d8d659f362c5f62c6cd1ff87f31ec99da134350b9b419370561

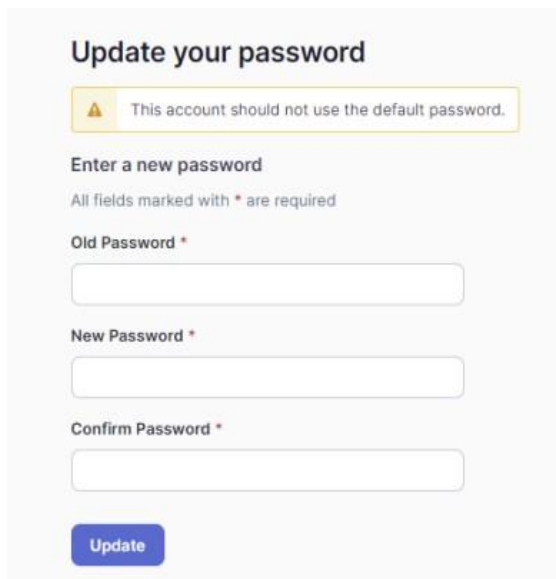
C:\Users\Student>
```

Step 2: Once the SonarQube image is started, you can go to **http://localhost:9000** to find the SonarQube that has started



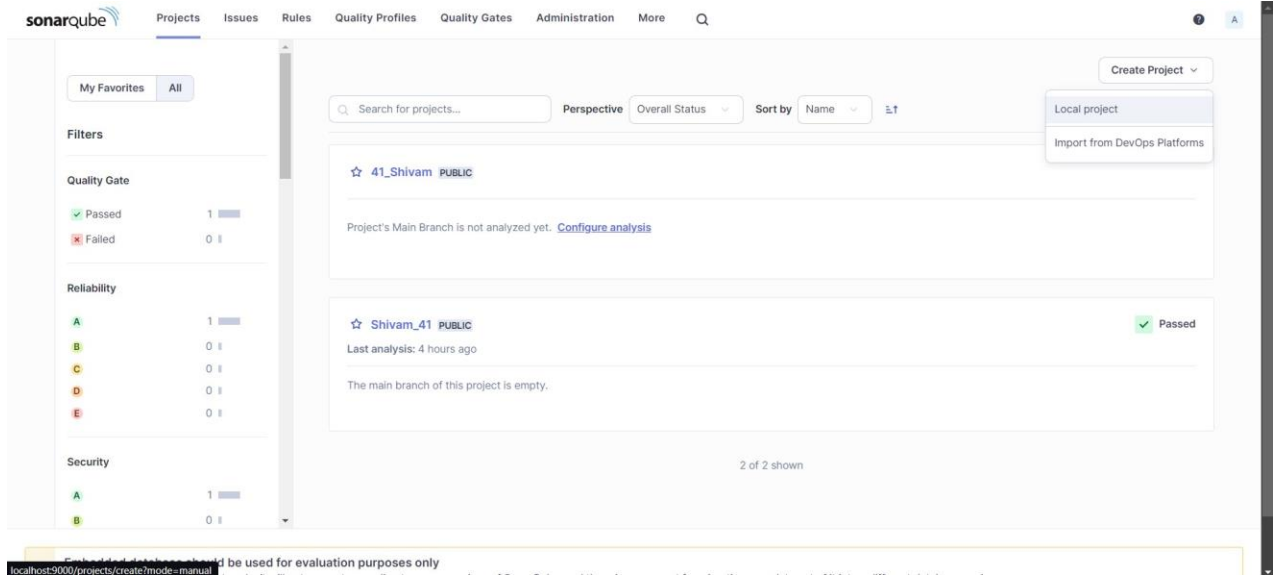
The image shows the SonarQube login interface. At the top is the Sonar logo. Below it is a blue icon representing a SonarQube project. The main heading is "Log in to SonarQube". There are two input fields: "Login *" with the text "admin" and "Password *" with masked characters. Below the fields are two buttons: "Go back" (a link) and "Log in" (a button).

Step 3: On this interface, login with **username = 'admin'** and **password = 'admin'**. Once logged in successfully, SonarQube will ask you to reset this password. Reset it and remember this password.



The image shows the "Update your password" interface. It features a yellow warning box with a triangle icon and the text "This account should not use the default password." Below this is the heading "Enter a new password" and a note "All fields marked with * are required". There are three input fields: "Old Password *" (empty), "New Password *" (empty), and "Confirm Password *" (empty). At the bottom is a blue "Update" button.

Step 4: After changing the password, you will be directed to this screen. Click on **Create a Local Project**. Give the project a display name and project key



Click on Create Project

1 of 2

Create a local project

Project display name *



Project key *



Main branch name *

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

Cancel

Next

Set up the project as required and click on create.

In the Step 2 while creating the project,Sonarqube ask you regarding which code should be considered as the new code for examining it .

The screenshot shows the SonarQube interface for configuring a project. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The main heading is "Set up project for Clean as You Code". Below this, a sub-heading asks to "Choose the baseline for new code for this project". There are three main options: "Use the global setting" (selected), "Define a specific setting for this project" (with sub-options: "Previous version", "Number of days", and "Reference branch"), and "Reference branch" (which is a sub-option of the previous one). The "Previous version" option is described as "Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases." The "Number of days" option is described as "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." The "Reference branch" option is described as "Choose a branch as the baseline for the new code. Recommended for projects using feature branches." At the bottom, there are "Back" and "Create project" buttons. A warning message at the bottom states: "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." The footer includes "SonarQube™ technology is powered by SonarSource SA" and "Community Edition v10.6 (92116) ACTIVE" with links to LGPL v3, Community, Documentation, Plugins, and Web API.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More Q

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.

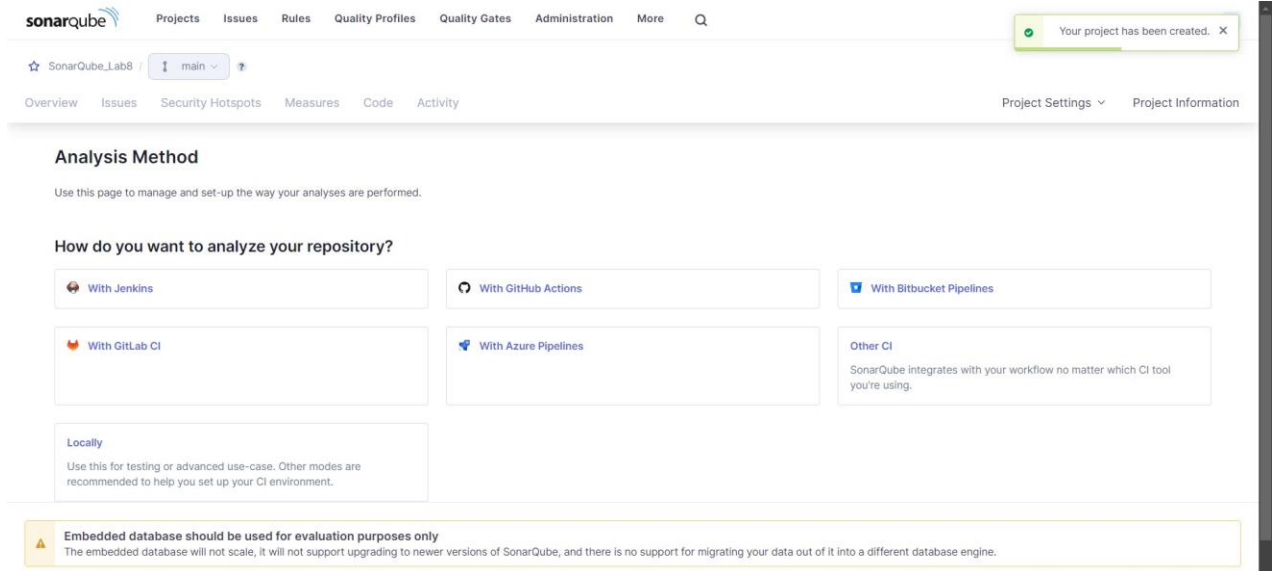
Back Create project

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.

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

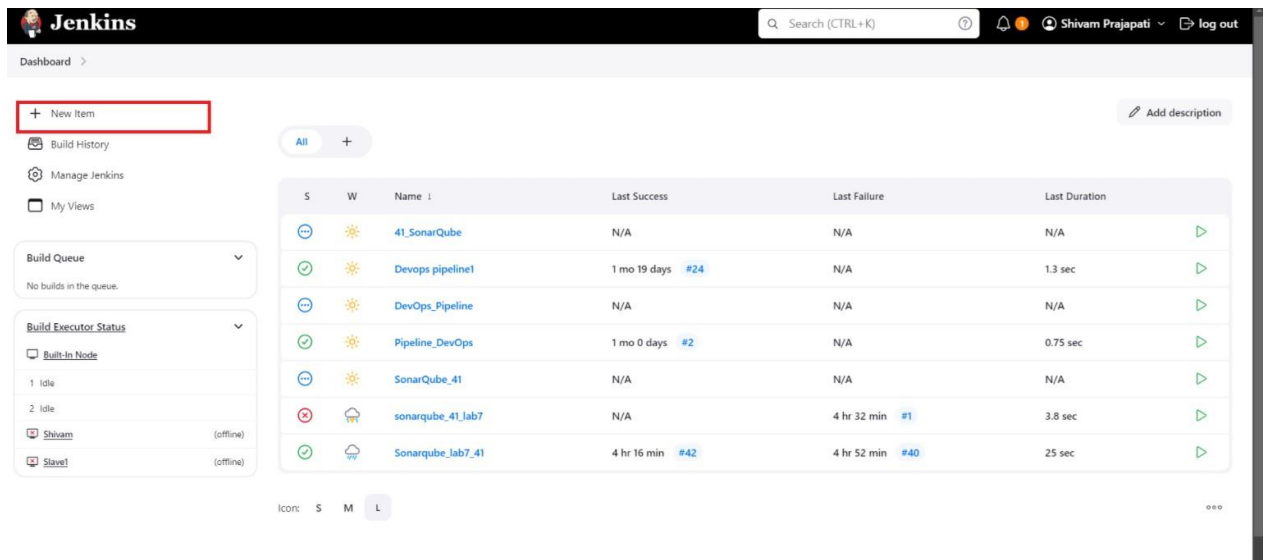
Community Edition v10.6 (92116) ACTIVE LGPL v3 Community Documentation Plugins Web API

Click on Create

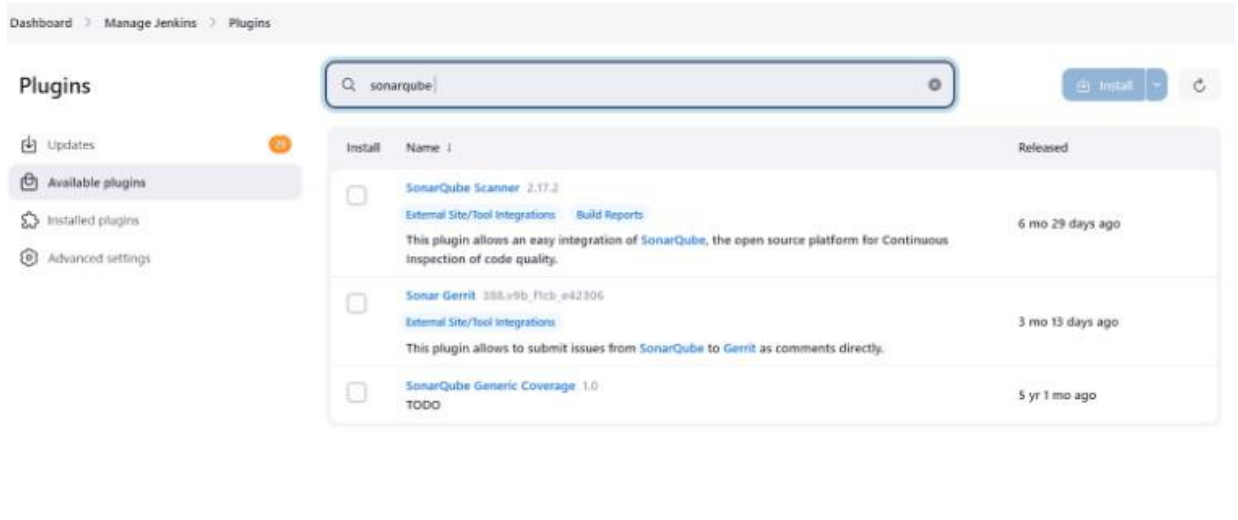


Project is created

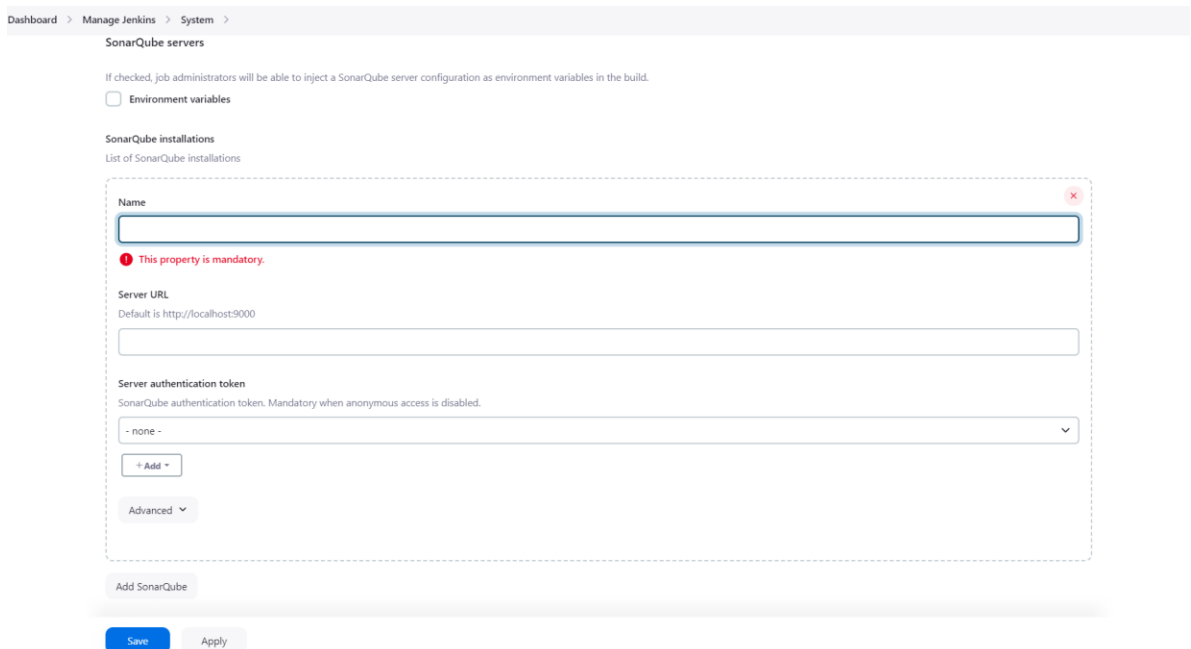
Step 5: Open **Jenkins** on whichever port it is installed. (<http://localhost:8080>). Go to the new item



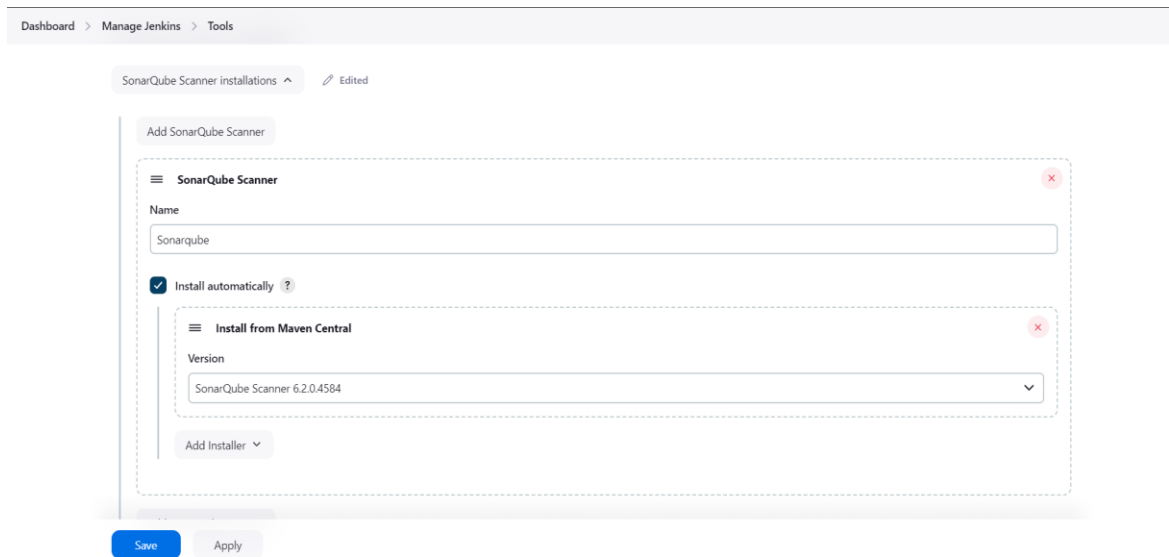
Step 6: Go to manage jenkins →available plugins then Search for **Sonarqube Scanner** for Jenkins and install it



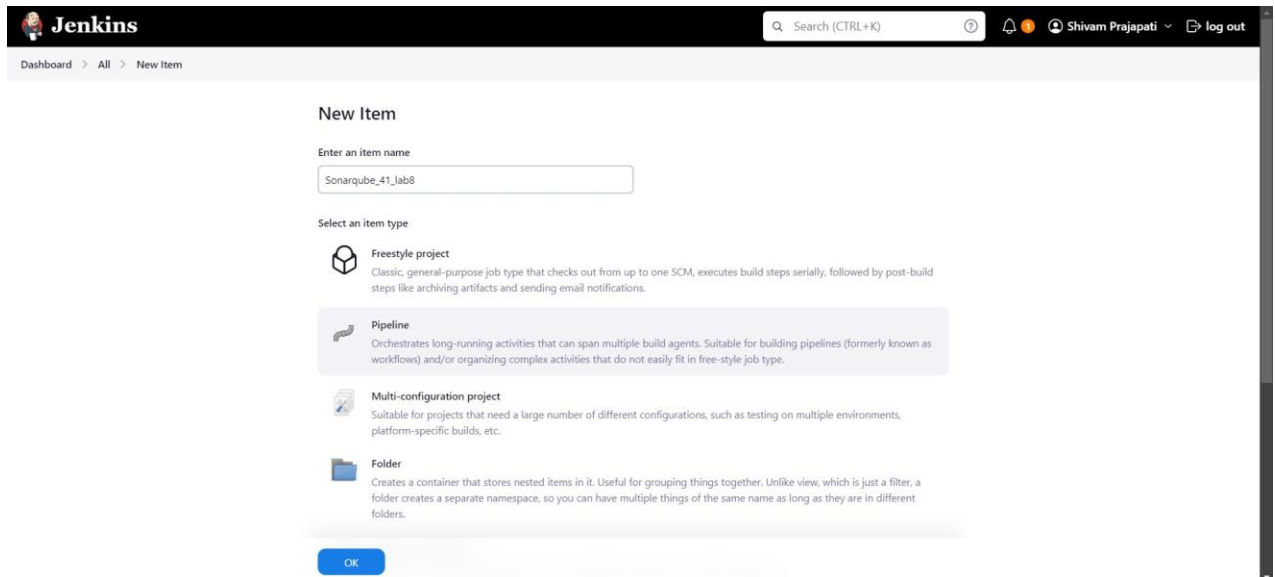
Step 7: Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.



Step 8: Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose to install automatically.



Step 9: After configuring, click on **New Item** and select **Pipeline Project**

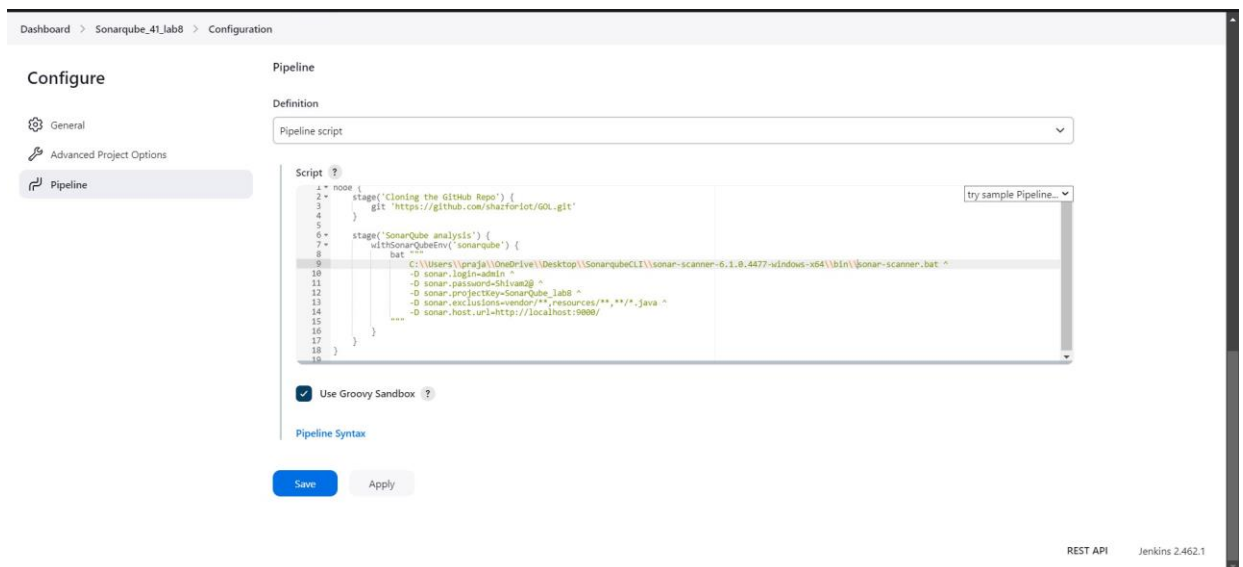


Step 10: Under Pipeline script, enter the following:

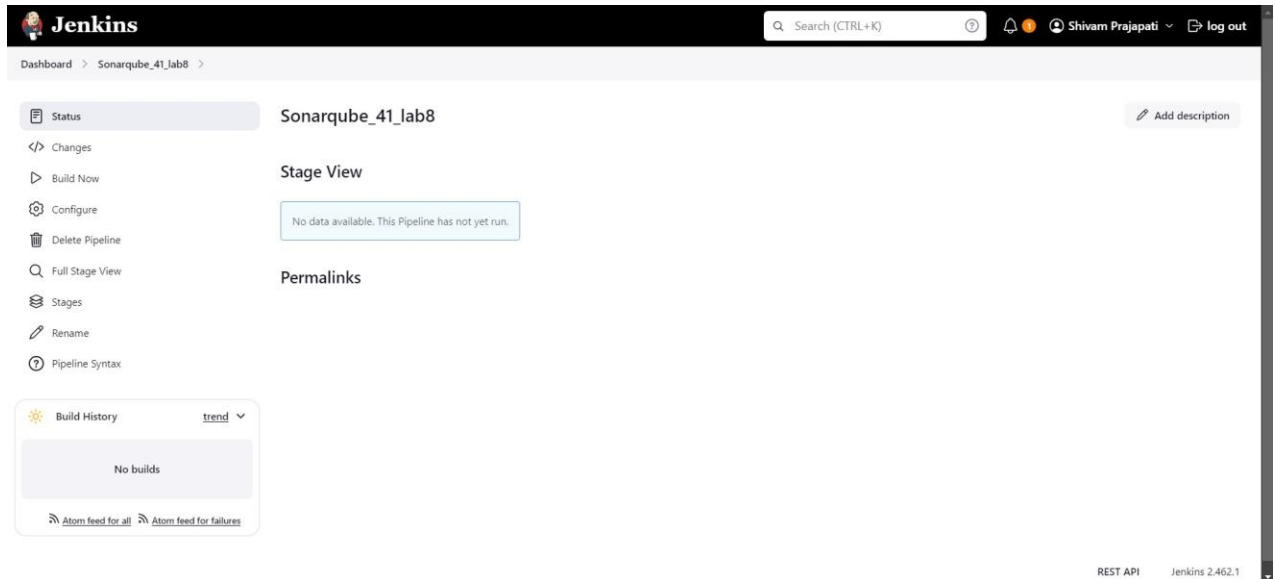
```
node {  
  stage('Cloning the GitHub Repo') {  
    git 'https://github.com/shazforiot/GOL.git'
```



```
}
stage('SonarQube analysis') {
    withSonarQubeEnv('sonarqube') {
        bat """
C:\\Users\\praja\\Downloads\\SonarqubeCLI\\sonar-scanner-6.1.0.4477-windows-x64\\bi
n\\sonar-scanner.bat ^
-D sonar.login=admin ^
-D sonar.password=Shivam2@ ^
-D sonar.projectKey=SonarQube_Lab8 ^
-D sonar.exclusions=vendor/**,resources/**,**/*.java ^
-D sonar.host.url=http://localhost:9000/
"""
    }
}
}
```



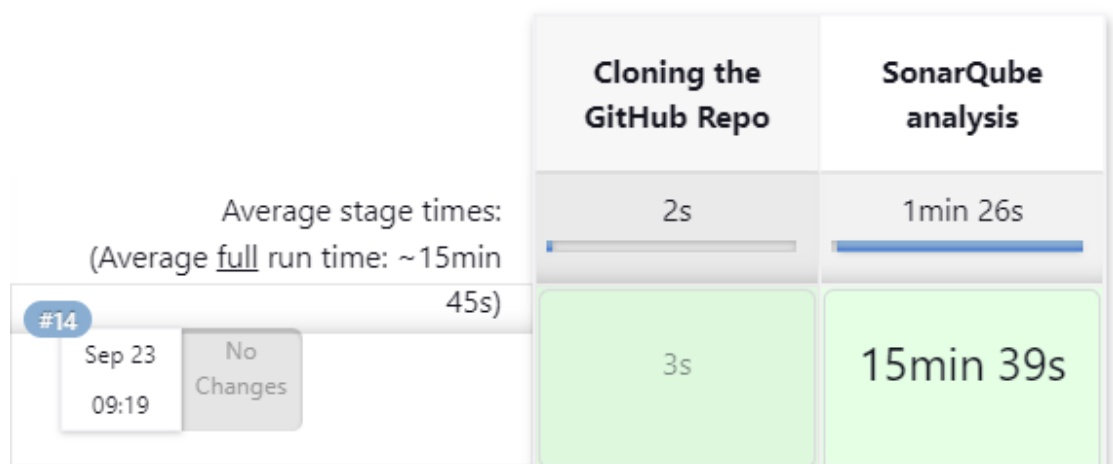
Click on save.



This is a Java sample project with many repetitive sections and coding issues that SonarQube will be able to detect during analysis.

Step 11: Go back to Jenkins. Go to the job you had just built and click on Build Now.

Stage View



The problem was C:\windows\system32 was not there so we need to add in our environment variable .

Now Check the console output once

Jenkins Search (CTRL+K) Shyam Prajapati log out

Dashboard > Sonarqube_lab8_41 > #14

Status
Changes
Console Output
View as plain text
Edit Build Information
Delete build #14
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Replay
Pipeline Steps
Workspaces
Previous Build

Console Output

Skipping 4,250 KB. [Full Log](#)

```
09:31:37.340 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/NewDriver.html for block at line 189. Keep only the first 100 references.
09:31:37.340 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/NewDriver.html for block at line 192. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 204. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 207. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 203. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 204. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 354. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 17. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 203. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 204. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 207. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 356. Keep only the first 100 references.
09:31:37.464 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line 356. Keep only the first 100 references.

Keep only the first 100 references.
09:31:43.178 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/assertions/gui/package-summary.html for block at line 40. Keep only the first 100 references.
09:31:43.194 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/protocol/mail/sampler/package-summary.html for block at line 39. Keep only the first 100 references.
09:31:43.194 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/apl/org/apache/jmeter/protocol/mail/sampler/package-summary.html for block at line 40. Keep only the first 100 references.
09:31:43.194 INFO CPD Executor CPD calculation finished (done) | time=170824ms
09:31:43.213 INFO SCM revision ID 'ba799ba7eb576f04a461222b0412c5e6a1e5e4'
09:34:16.341 INFO Analysis report generated in 4552ms, dir size=127.2 MB
09:34:33.584 INFO Analysis report compressed in 17210ms, zip size=29.6 MB
09:34:34.392 INFO Analysis report uploaded in 807ms
09:34:34.395 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=SonarQube_lab8
09:34:34.395 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
09:34:34.395 INFO More about the report processing at http://localhost:9000/api/ci/task?id=7dfe78a1-793d-47f9-bf86-4636bb755c69
09:34:45.070 INFO SonarScanner Engine completed successfully
09:34:45.929 INFO EXECUTION SUCCESS
09:34:45.932 INFO Total time: 15:36.252s

[Pipeline] }
[Pipeline] // withSonarQubeEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Dashboard > Sonarqube_lab8_41 > #14

REST API Jenkins 2.462.1

Successfully BUILD

Step 12: After the build is finished, return to SonarQube and review the linked project in detail.

The top screenshot shows the SonarQube 'Projects' overview page. The left sidebar contains filters for Quality Gate (Passed: 2, Failed: 0), Reliability (A: 1, B: 0, C: 1, D: 0, E: 0), and Security (A: 2, B: 0). The main area lists projects, with 'Shivam_41' and 'SonarQube_Lab8' visible. 'SonarQube_Lab8' is highlighted, showing a 'Passed' status and analysis details: 683k Lines of Code, HTML, XML, etc. The bottom screenshot shows the detailed view of the 'SonarQube_Lab8' project. The 'main' branch is selected, and the project is marked as 'Passed'. The 'Quality Gate' is 'Passed'. The 'Overall Code' tab is active, showing metrics for Security (0 Open Issues), Reliability (68k Open Issues), Maintainability (164k Open Issues), Accepted Issues (0), Coverage (0 lines to cover), and Duplications (50.6%).

Under different options on the navbar , we can check all the issues with the code.

UNDER ISSUES:

1) Consistency

The screenshot displays the SonarQube web interface for a project named 'SonarQube_Lab8'. The 'Issues' tab is active, showing a list of issues under the 'Consistency' category. The left sidebar shows filters for 'Clean Code Attribute' with 'Consistency' at 197%, 'Intentionality' at 14k, 'Adaptability' at 0, and 'Responsibility' at 0. The main panel lists three issues:

- Insert a <DOCTYPE> declaration to before this <html> tag.** (Consistency, Reliability, user-experience, L1, 5min effort, 4 years ago, Bug, Major)
- Remove this deprecated "width" attribute.** (Consistency, Maintainability, html5, obsolete, L9, 5min effort, 4 years ago, Code Smell, Major)
- Remove this deprecated "align" attribute.** (Consistency, Maintainability, html5, obsolete)

A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only'.

2) Reliability

The screenshot displays the SonarQube web interface for the same project, 'SonarQube_Lab8', with the 'Issues' tab active. The left sidebar shows filters for 'Software Quality' with 'Reliability' at 14k and 'Maintainability' at 15. The main panel lists three issues:

- Add "lang" and/or "xml:lang" attributes to this "<html>" element** (Intentionality, Reliability, accessibility, wcag2-a, L1, 2min effort, 4 years ago, Bug, Major)
- Add "<th>" headers to this "<table>"** (Intentionality, Reliability, accessibility, wcag2-a, L9, 2min effort, 4 years ago, Bug, Major)
- Add "lang" and/or "xml:lang" attributes to this "<html>" element** (Intentionality, Reliability, accessibility, wcag2-a)

A warning banner at the bottom states: 'Embedded database should be used for evaluation purposes only'.

3) Maintainability

The screenshot shows the SonarQube web interface for a project named 'SonarQube_Lab8'. The 'Issues' tab is selected, and the 'Maintainability' category is chosen from the left sidebar. The main panel displays three issues related to Dockerfile code. Each issue has a description, a 'Maintainability' icon, and a 'Severity' of 'Intentionality'. The issues are:

- Use a specific version tag for the image. (L1 - 5min effort - 4 years ago - @ Code Smell - @ Major)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (L12 - 5min effort - 4 years ago - @ Code Smell - @ Major)
- Surround this variable with double quotes; otherwise, it can lead to unexpected behavior. (L12 - 5min effort - 4 years ago - @ Code Smell - @ Major)

A warning message at the bottom states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale. It will not support migration to newer versions of SonarQube, and there is no support for migration your data out of it into a different database engine."

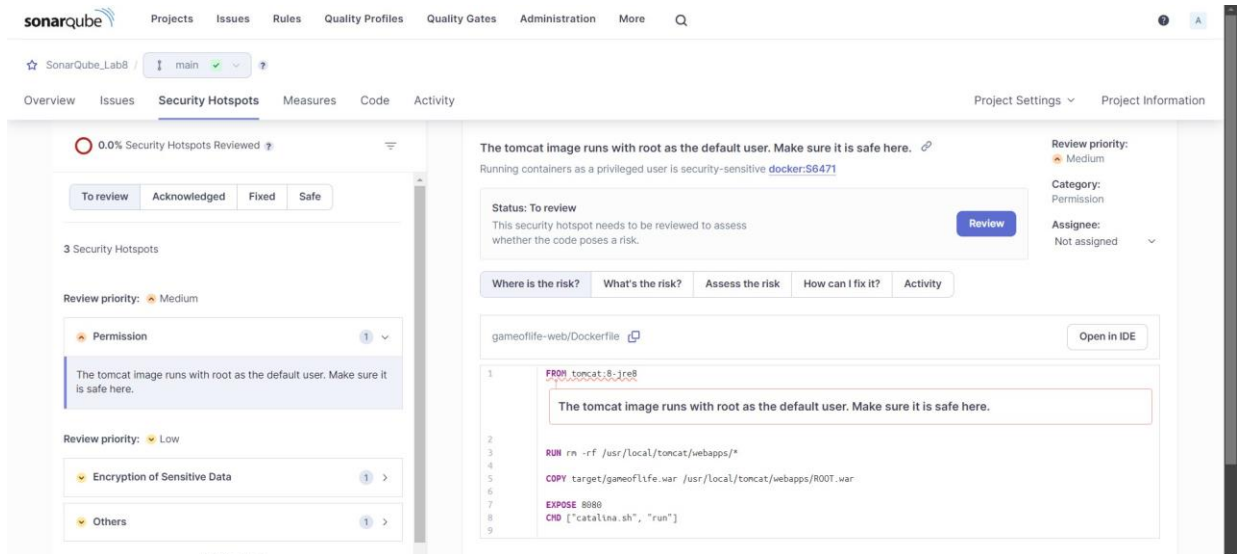
4) Severity

The screenshot shows the SonarQube web interface for the same project 'SonarQube_Lab8'. The 'Issues' tab is selected, and the 'Severity' category is chosen from the left sidebar. The main panel displays three issues related to HTML code. Each issue has a description, a 'Severity' icon, and a 'Severity' of 'Intentionality'. The issues are:

- Add the "let", "const" or "var" keyword to this declaration of "prop" to make it explicit. (L31 - 2min effort - 4 years ago - @ Code Smell - @ Blocker)
- Add the "let", "const" or "var" keyword to this declaration of "prop" to make it explicit. (L31 - 2min effort - 4 years ago - @ Code Smell - @ Blocker)
- Add the "let", "const" or "var" keyword to this declaration of "prop" to make it explicit. (L31 - 2min effort - 4 years ago - @ Code Smell - @ Blocker)

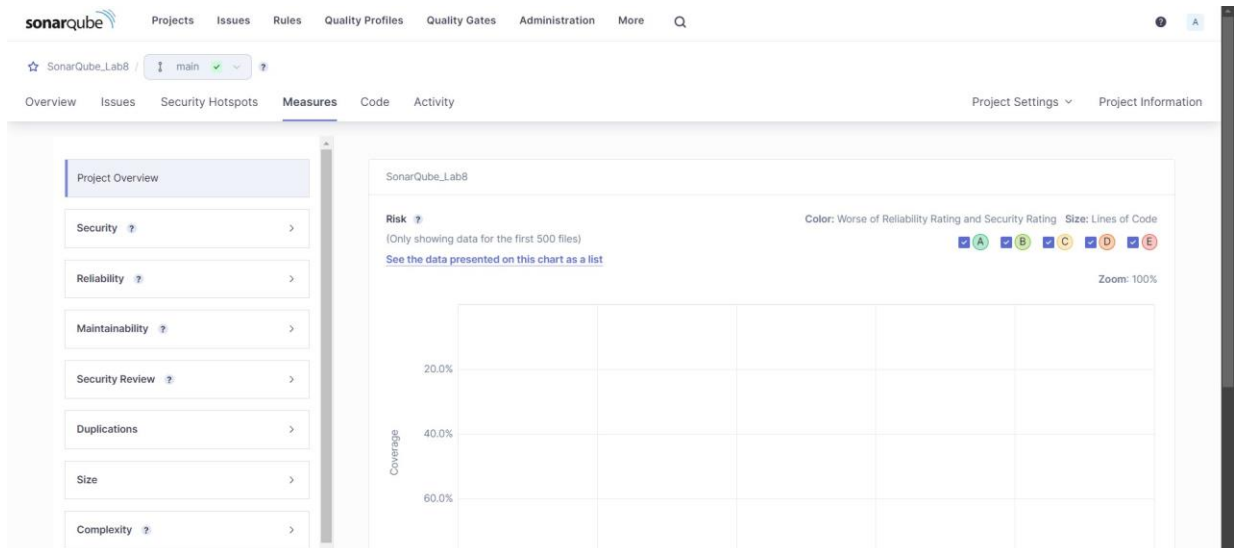
A warning message at the bottom states: "Embedded database should be used for evaluation purposes only. The embedded database will not scale. It will not support migration to newer versions of SonarQube, and there is no support for migration your data out of it into a different database engine."

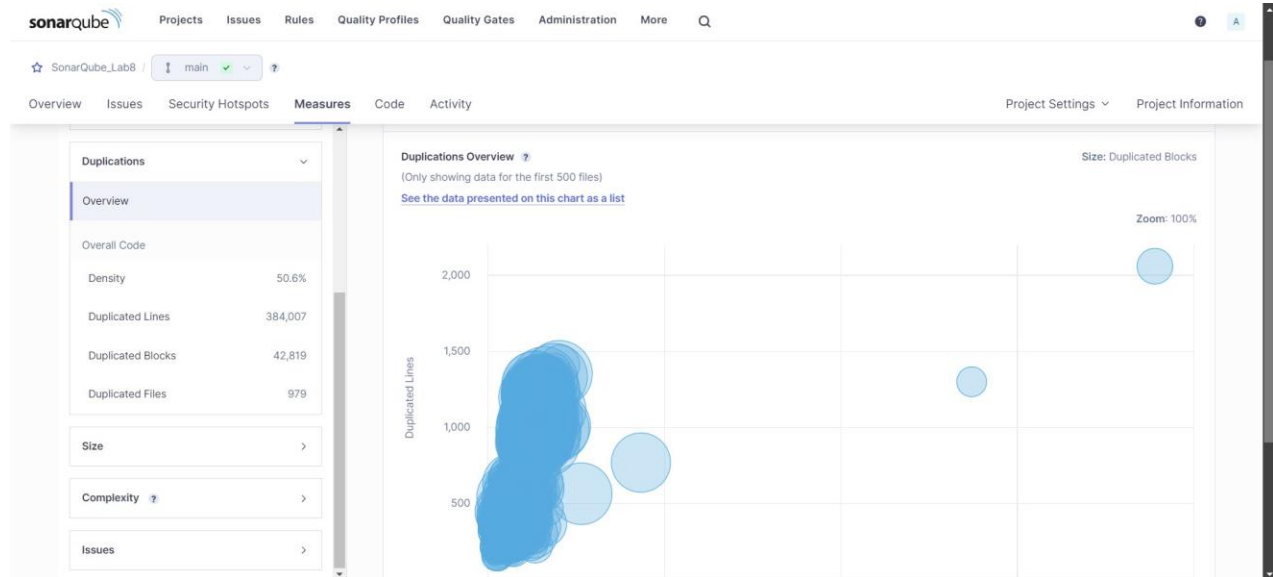
UNDER SECURITY HOTSPOT:



The screenshot shows the SonarQube interface for the 'SonarQube_Lab8' project. The 'Security Hotspots' tab is active, displaying a summary of 3 hotspots with a 0.0% review rate. The left sidebar lists hotspots by review priority: Medium (1), Low (1), and Others (1). The main panel shows a detailed view of a hotspot titled 'The tomcat image runs with root as the default user. Make sure it is safe here.' The status is 'To review', and the category is 'Permission'. The hotspot is located in 'gameoflife-web/Dockerfile'. The code snippet shows a Dockerfile with a 'FROM' statement using 'tomcat:8-jre8' and a 'RUN' statement that sets the user to 'root'. The review priority is 'Medium', and the assignee is 'Not assigned'.

UNDER MEASURES:





CONCLUSION:

In this experiment, we demonstrated how to perform static code analysis using Jenkins CI/CD Pipeline with SonarQube integration. We created a pipeline project with a specific script that contains all the instructions necessary to run SonarQube analysis. After configuring Jenkins appropriately, we built the project. The analyzed code in this experiment had several issues, such as errors, bugs, and duplications, all of which were detected and displayed in the linked SonarQube project.