

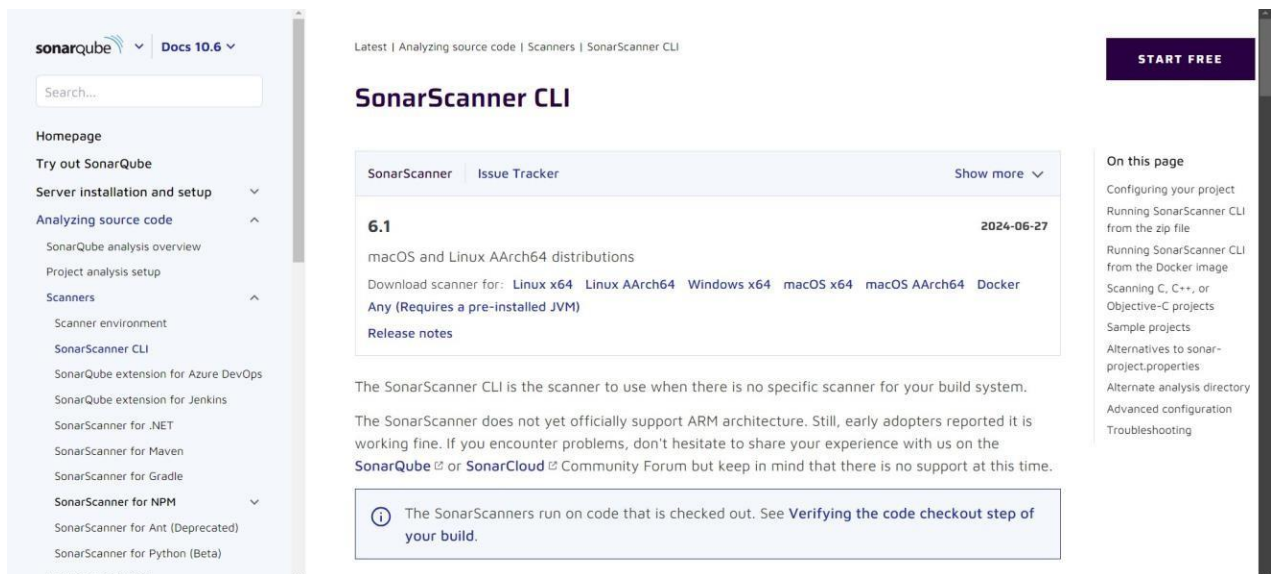
## Experiment No: 8

**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



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.

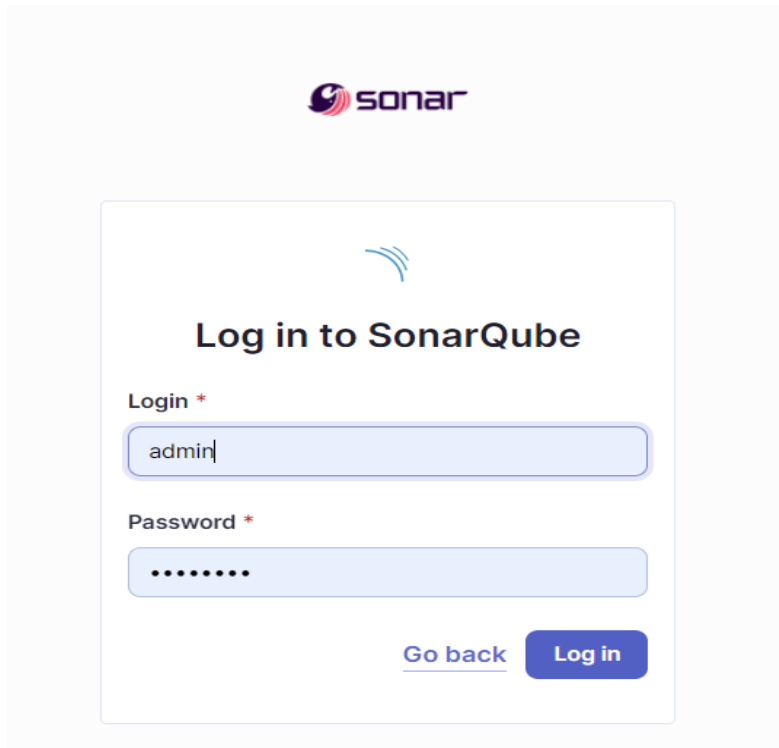
```

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

C:\Users\Student>

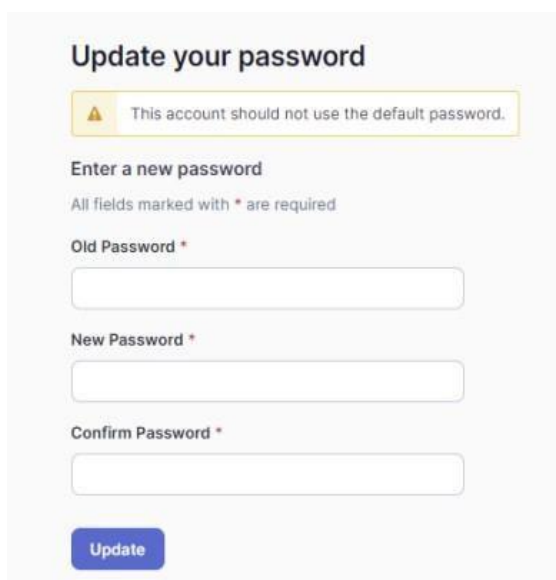
```

**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 page. At the top is the Sonar logo. Below it is a blue icon of three curved lines. The main heading is "Log in to SonarQube". There are two input fields: "Login \*" with the text "admin" and "Password \*" with masked dots. At the bottom right are two buttons: "Go back" (a link) and "Log in" (a blue 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" page. At the top is the heading "Update your password". Below it is a yellow warning box with a triangle icon and the text "This account should not use the default password.". Underneath 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 asks you regarding which code should be considered as the new code for examining it.

The screenshot shows the SonarQube web 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 radio button options: "Use the global setting" (selected), "Define a specific setting for this project", and "Reference branch". Under "Define a specific setting for this project", there are three sub-options: "Previous version", "Number of days", and "Reference branch". Each sub-option has a brief description and a recommendation. 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 the SonarQube logo, version information (Community Edition v10.6 (92116) ACTIVE), and links to LGPL v3, Community, Documentation, Plugins, and Web API.

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

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.

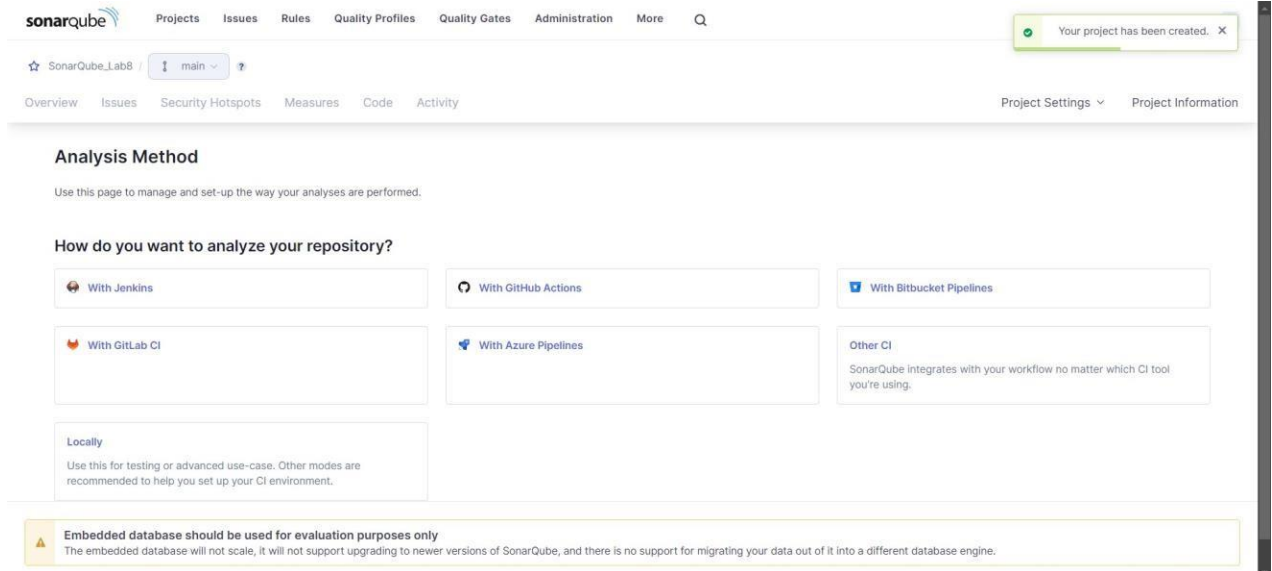
[Back](#) [Create project](#)

**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](#)

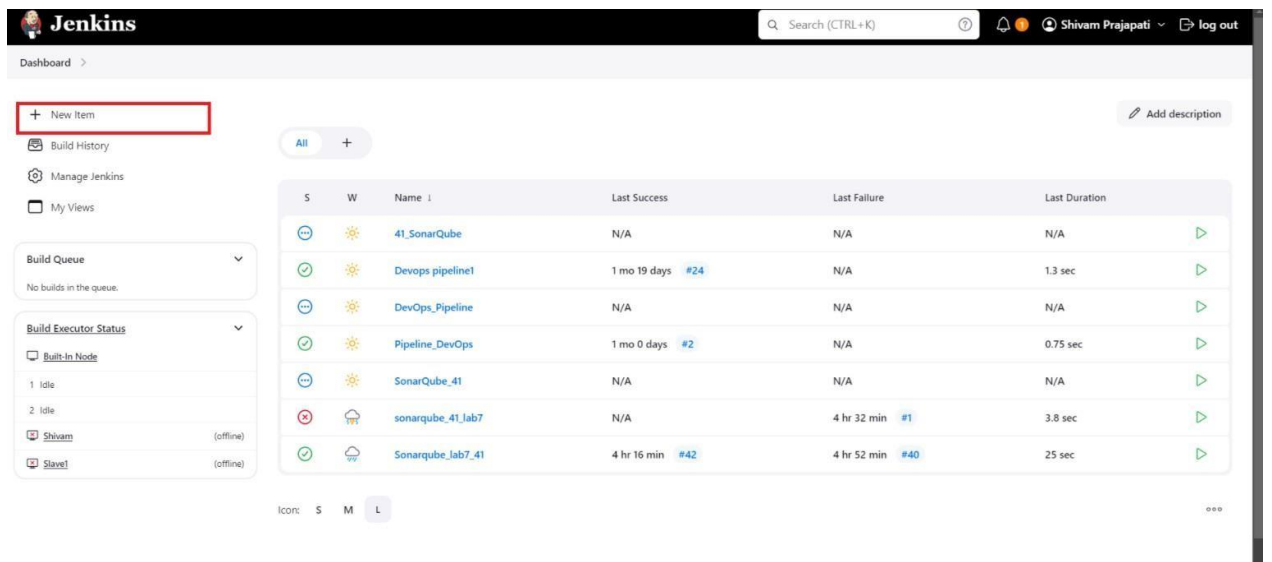
Community Edition v10.6 (92116) ACTIVE [LGPL v3](#) [Community](#) [Documentation](#) [Plugins](#) [Web API](#)

Click on Create



Project is created

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



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

Dashboard > Manage Jenkins > Plugins

## Plugins

Updates 2

Available plugins

Installed plugins

Advanced settings

Install

↻

Install	Name	Released
<input type="checkbox"/>	<b>SonarQube Scanner</b> 2.17.2 <a href="#">External Site/Tool Integrations</a> <a href="#">Build Reports</a> This plugin allows an easy integration of <a href="#">SonarQube</a> , the open source platform for Continuous Inspection of code quality.	6 mo 29 days ago
<input type="checkbox"/>	<b>Sonar Gerrit</b> 388.v9b.f1c3b.e42306 <a href="#">External Site/Tool Integrations</a> This plugin allows to submit issues from <a href="#">SonarQube</a> to <a href="#">Gerrit</a> as comments directly.	3 mo 13 days ago
<input type="checkbox"/>	<b>SonarQube Generic Coverage</b> 1.0 TODO	5 yr 1 mo ago

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

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

ⓘ This property is mandatory.

Server URL

Default is `http://localhost:9000`

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

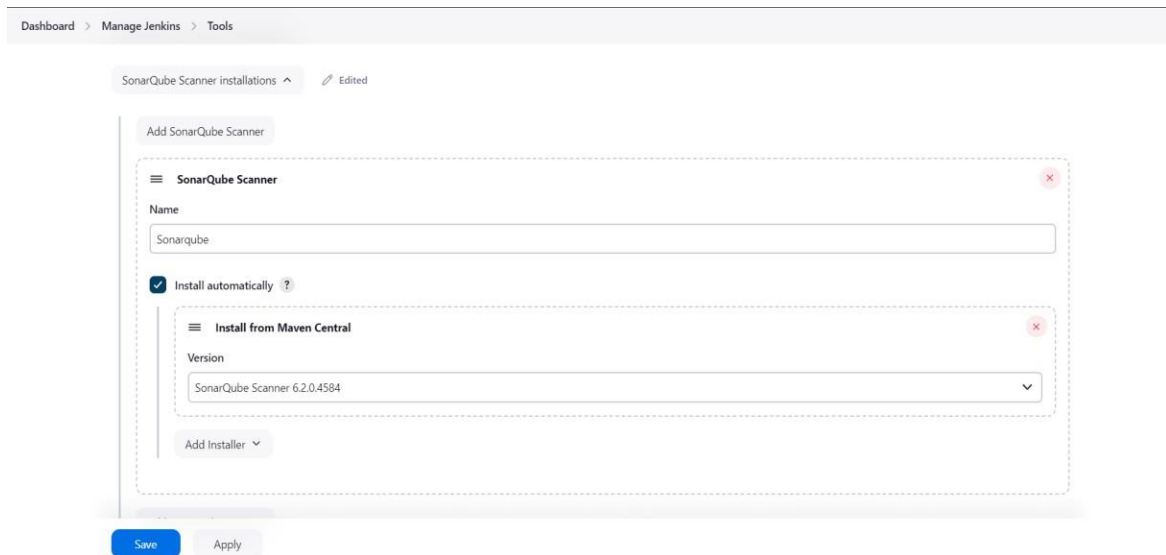
+ Add +

Advanced ▾

Add SonarQube

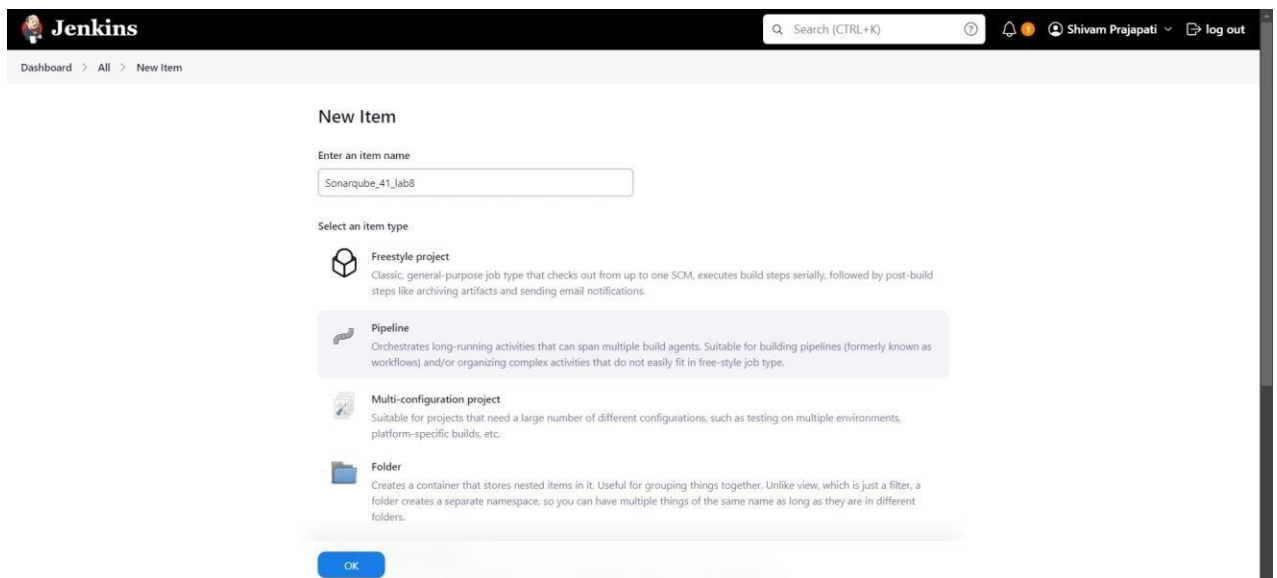
Save Apply

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



The screenshot shows the 'SonarQube Scanner installations' page in Jenkins. The breadcrumb navigation is 'Dashboard > Manage Jenkins > Tools'. The page title is 'SonarQube Scanner installations' with an 'Edited' status. There is a button 'Add SonarQube Scanner'. Below it, a configuration box for 'SonarQube Scanner' is shown. It has a 'Name' field with 'Sonarqube' entered. The 'Install automatically' checkbox is checked. Below this, there is a section 'Install from Maven Central' with a 'Version' dropdown menu showing 'SonarQube Scanner 6.2.0.4584'. At the bottom of the configuration box is an 'Add Installer' button. Below the configuration box are 'Save' and 'Apply' buttons.

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



The screenshot shows the 'New Item' page in Jenkins. The breadcrumb navigation is 'Dashboard > All > New Item'. The page title is 'New Item'. There is a search bar with 'Search (CTRL+K)'. The user is 'Shivam Prajapati'. The 'Enter an item name' field contains 'Sonarqube\_41\_lab8'. Under 'Select an item type', there are four options: '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), '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.), and 'Folder' (Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders). At the bottom is an 'OK' button.

**Step 10:** Under Pipeline script, enter the following:

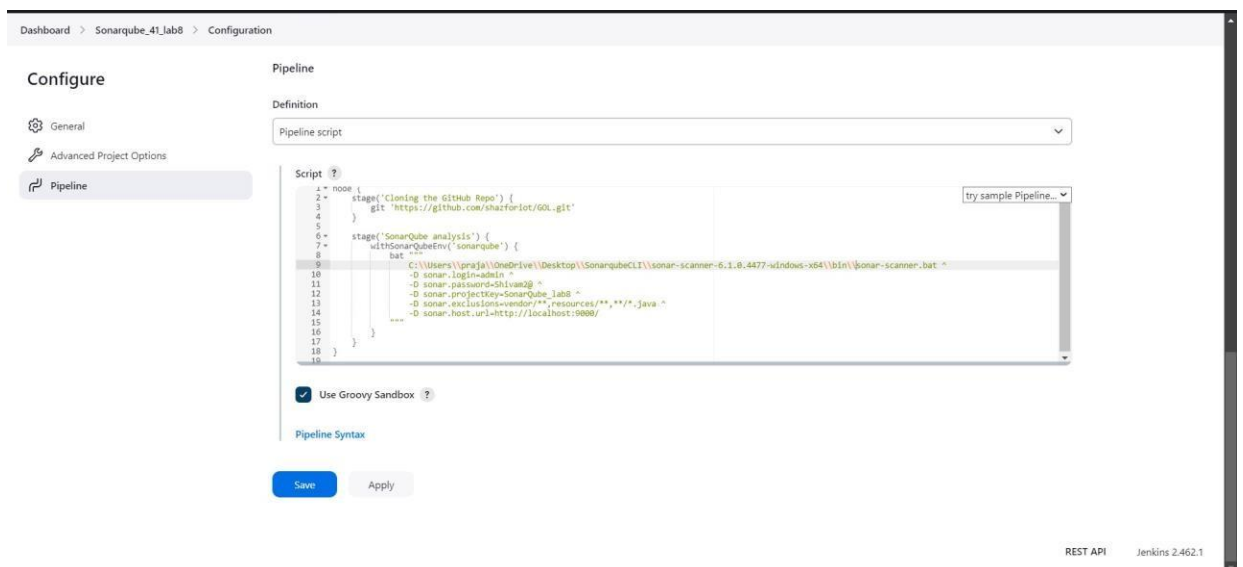
```
node {  
  stage('Cloning the GitHub Repo') {  
    git 'https://github.com/shazforiot/GOL.git'
```



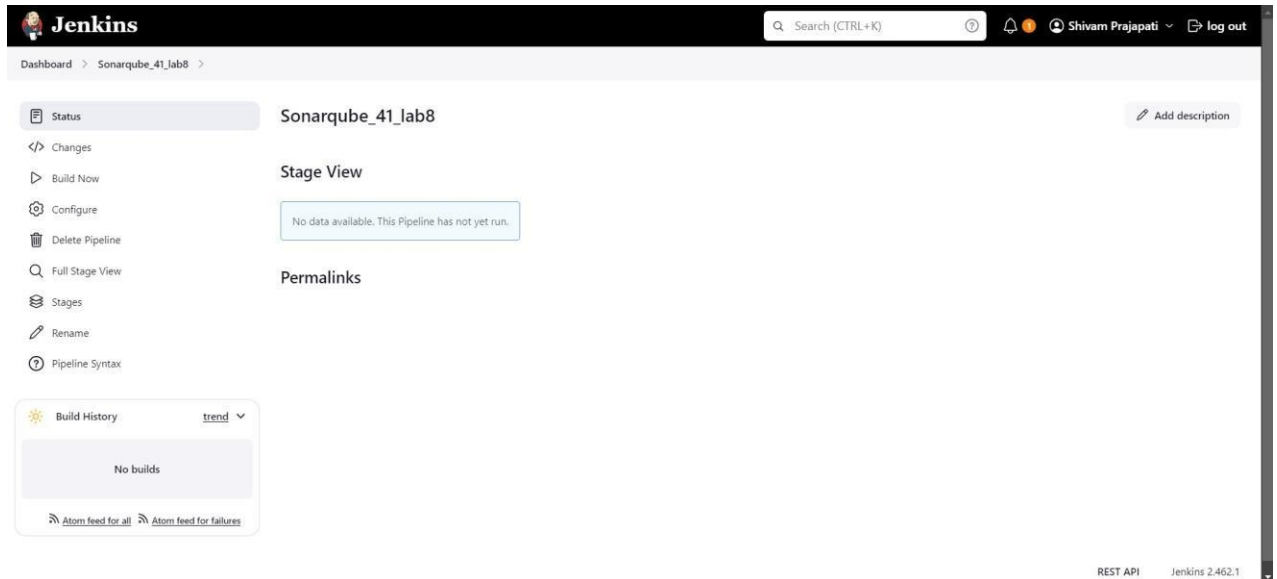
```

}
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/
"""
    }
}
}
}

```



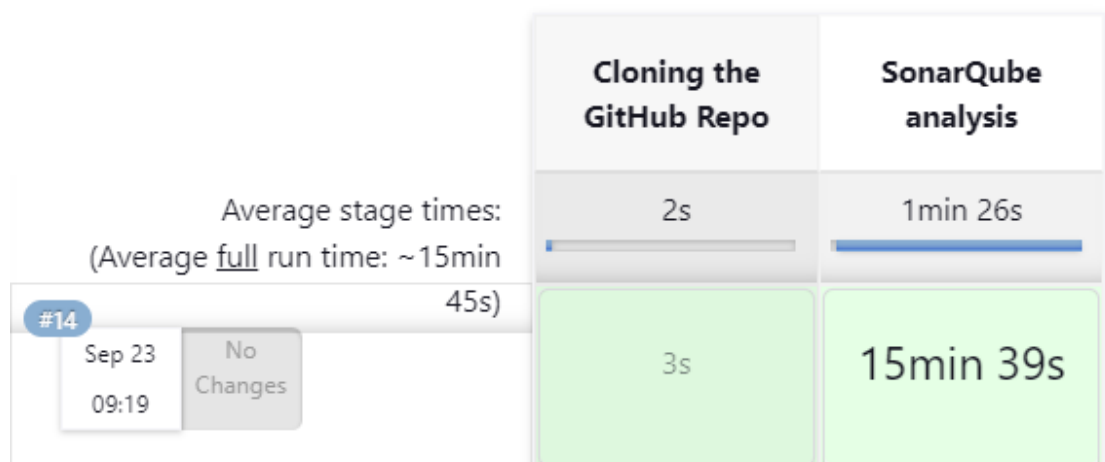
Click on save.



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

The screenshot displays the Jenkins web interface for a build named 'Sonarqube\_lab8\_41' at step '#14'. The 'Console Output' tab is selected, showing a log of messages. The log begins with a warning about too many duplication references on several files, which is repeated multiple times. The build then proceeds with various informational messages, including the completion of the CPD Executor, the generation of an analysis report, and the successful completion of the SonarScanner Engine. The build concludes with a 'Finished: SUCCESS' status. The bottom right corner of the interface shows 'REST API' and 'Jenkins 2.462.1'.

```
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/api/org/apache/jmeter/NavDriver.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/api/org/apache/jmeter/NavDriver.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/api/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/api/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/api/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/api/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/api/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/api/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/api/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/api/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/api/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/api/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/api/org/apache/jmeter/samplers/StatisticalSampleSender.html for block at line

Keep only the first 100 references.
09:31:43.178 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/org/apache/jmeter/assertions/gui/package-summary.html for block at line 48. Keep only the first 100 references.
09:31:43.194 WARN Too many duplication references on file gameoflife-web/tools/jmeter/docs/api/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/api/org/apache/jmeter/protocol/mail/sampler/package-summary.html for block at line 48. 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 'ba799ba7eb576f04a4612322b0412c5e6a1e5d4'
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/cx/task?id=7dfe78a1-793d-47f9-bf86-4636bb755c69
09:34:45.065 INFO Analysis total time: 15:27.329 s
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

REST API Jenkins 2.462.1
```

Successfully BUIL



# UNDER ISSUES:

## 1) Consistency

The screenshot shows the SonarQube web interface for a project named 'SonarQube\_Lab8'. The 'Issues' tab is selected, and the 'Consistency' filter is applied. The left sidebar shows a list of filters, including 'Clean Code Attribute' with a count of 197k and 'Software Quality' with a count of 14k. The main area displays three issues related to HTML attributes: 'Insert a <!DOCTYPE> declaration to before this <html> tag.', 'Remove this deprecated "width" attribute.', and 'Remove this deprecated "align" attribute.'. Each issue is categorized as 'Consistency' and 'Reliability'.

Embedded database should be used for evaluation purposes only

## 2) Reliability

The screenshot shows the SonarQube web interface for the same project, 'SonarQube\_Lab8'. The 'Issues' tab is selected, and the 'Reliability' filter is applied. The left sidebar shows a list of filters, including 'Software Quality' with a count of 14k and 'Reliability' with a count of 15. The main area displays three issues related to HTML attributes: 'Add "lang" and/or "xml:lang" attributes to this <html> element', 'Add <th> headers to this <table>', and 'Add "lang" and/or "xml:lang" attributes to this <html> element'. Each issue is categorized as 'Reliability' and 'Intentionality'.

Embedded database should be used for evaluation purposes only

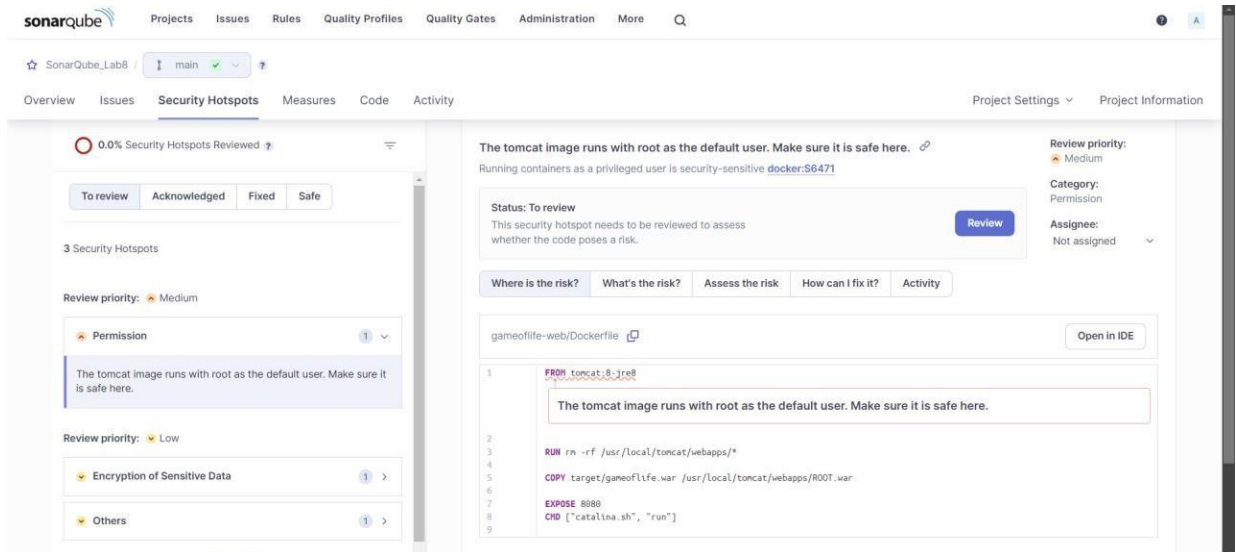
### 3) Maintainability

The screenshot shows the SonarQube web interface for a project named 'SonarQube\_Lab8'. The 'Issues' tab is selected, and the 'Maintainability' quality gate is highlighted in the left sidebar, showing 15 issues. The main panel displays three maintainability issues related to Dockerfile code. Each issue is a code smell of 'Major' severity, labeled 'Intentionality', and has a '44min effort' associated with it. The issues are: 'Use a specific version tag for the image.', 'Surround this variable with double quotes; otherwise, it can lead to unexpected behavior.', and another identical one. Each issue has a 'Maintainability' icon and a 'No tags' dropdown. At the bottom, a yellow warning banner 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

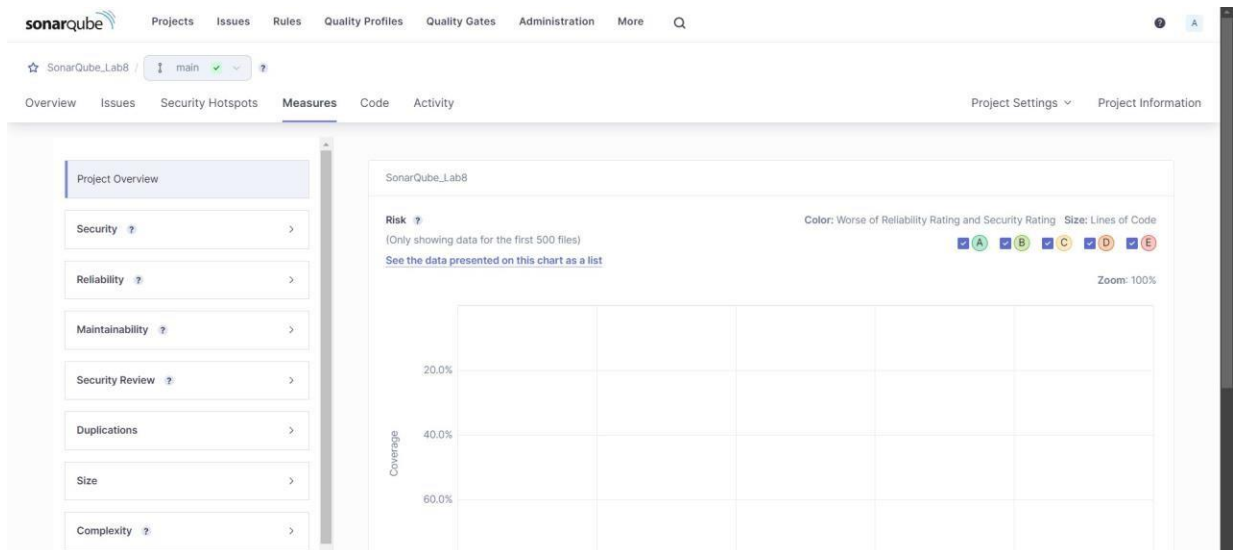
The screenshot shows the SonarQube web interface for the same project 'SonarQube\_Lab8'. The 'Issues' tab is selected, and the 'Severity' quality gate is highlighted in the left sidebar, showing 7 issues. The main panel displays three severity issues related to HTML code. Each issue is a code smell of 'Major' severity, labeled 'Intentionality', and has a '14min effort' associated with it. The issues are: 'Add the "let", "const" or "var" keyword to this declaration of "prop" to make it explicit.', 'Add the "let", "const" or "var" keyword to this declaration of "prop" to make it explicit.', and another identical one. Each issue has a 'Severity' icon and a 'No tags' dropdown. At the bottom, a yellow warning banner 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.'

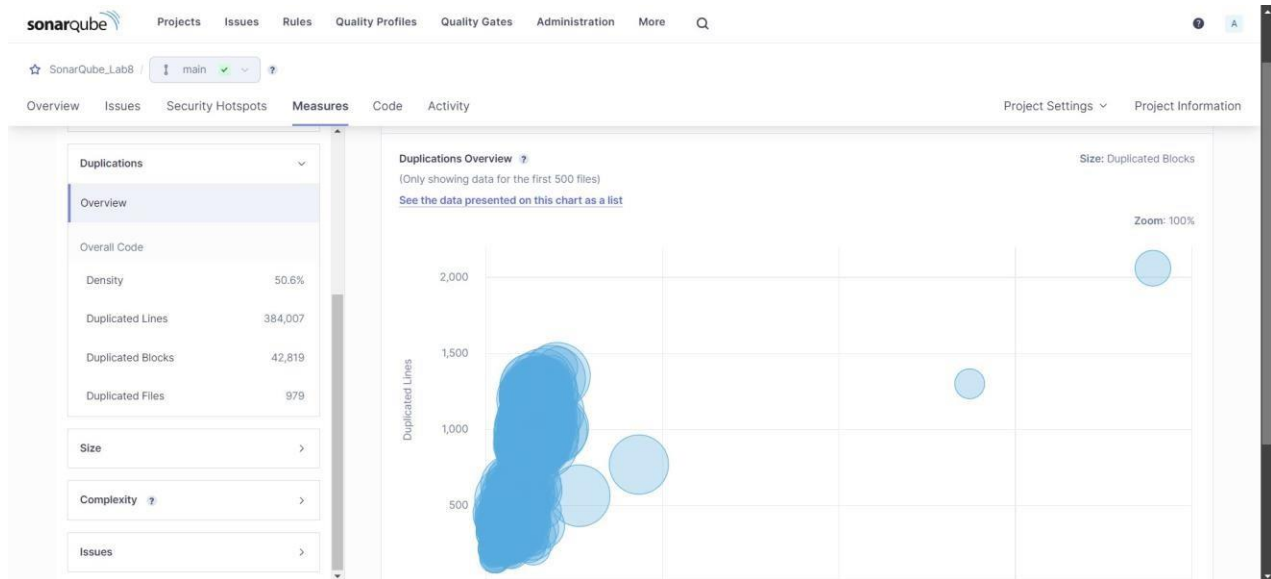
## UNDER SECURITY HOTSPOT:



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. A specific hotspot is highlighted with the title 'The tomcat image runs with root as the default user. Make sure it is safe here.' The hotspot is categorized as 'Permission' with a 'Medium' review priority. The associated code is shown in a Dockerfile snippet, specifically the 'FROM tomcat:8-jre8' line. The interface includes tabs for 'To review', 'Acknowledged', 'Fixed', and 'Safe', and a 'Review' button for the selected hotspot.

## 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.