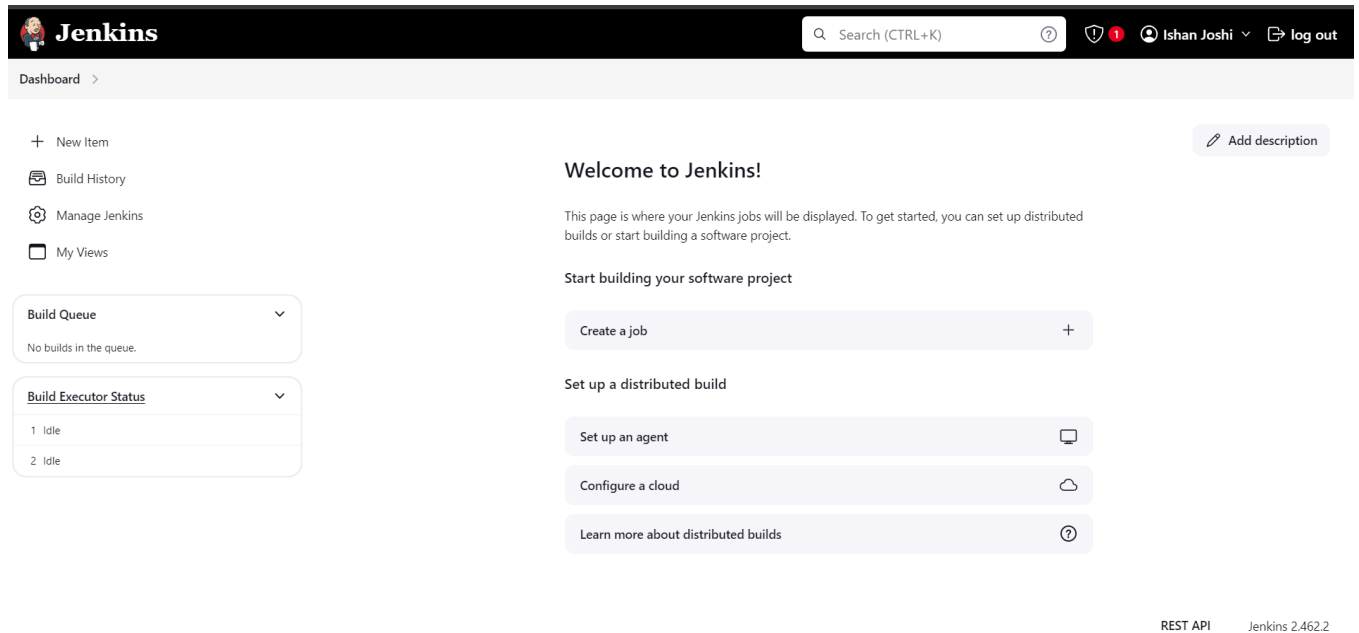


Name-Ishan Kiran Joshi DIV-D15C Roll NO-21 A.Y.-2024-25

Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.

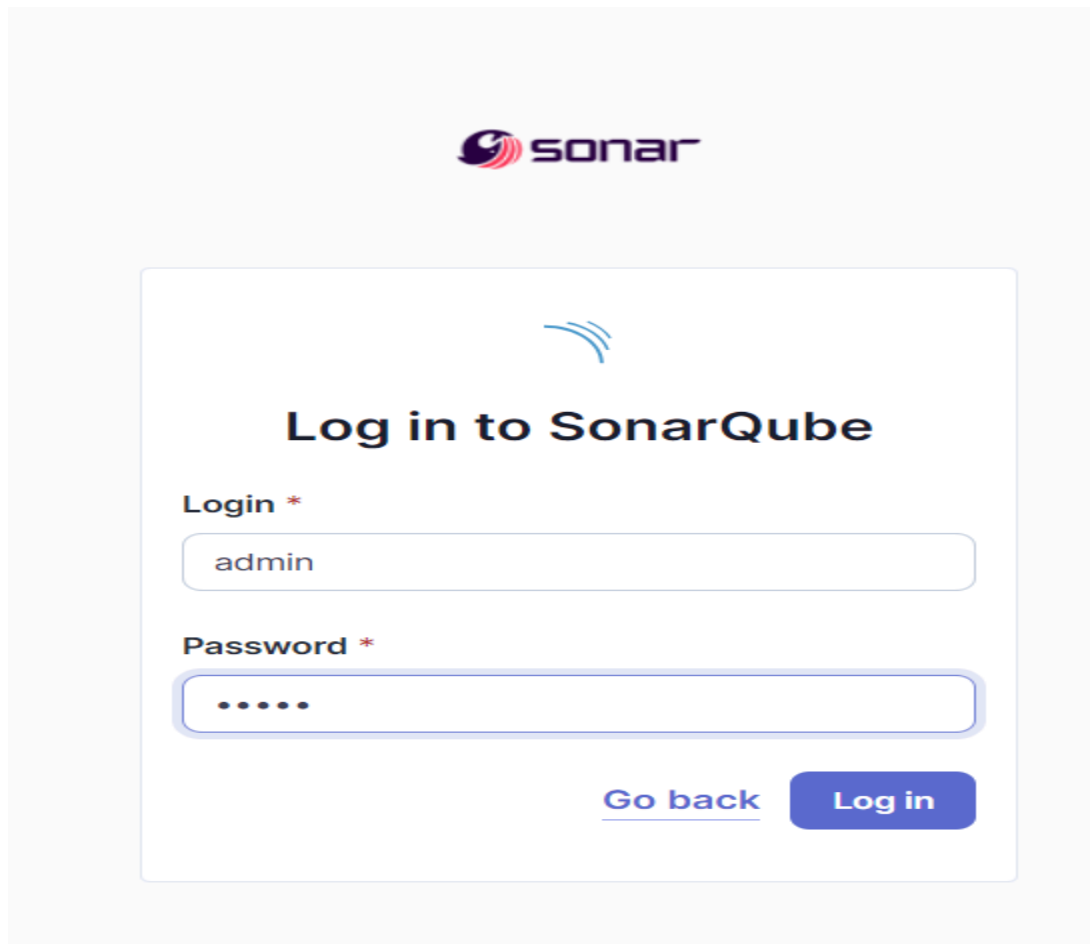


2. Run SonarQube in a Docker container using this command :- a) `docker -v`
b) `docker pull sonarqube`
c) `docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest`


```
C:\Users\ishan>docker -v
Docker version 27.1.1, build 6312585


C:\Users\ishan>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
90a925ab929a: Pull complete
7d9a34308537: Pull complete
80338217a4ab: Pull complete
1a5fd5c7e184: Pull complete
7b87d66fa783d: Pull complete
bd819c9b5ead: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
23c14503da77ee785f6069bfbaff714939ddb794a6b846124594503f6183b4c68
```

3. Once the container is up and running, you can check the status of SonarQube at **localhost port 9000**. The login id is “**admin**” and the password is “**ishan**”.



The image shows the SonarQube login interface. At the top, there is the Sonar logo. Below it, a blue curved line icon is centered above the heading "Log in to SonarQube". The login form consists of two input fields: "Login *" with the text "admin" entered, and "Password *" with five dots representing a masked password. At the bottom right of the form, there is a blue button labeled "Log in" and a blue link labeled "Go back" to its left.






Log in to SonarQube

Login *

Password *

[Go back](#) Log in

4. Create a local project in SonarQube with the name sonarqube



ProjectsIssuesRulesQuality ProfilesQ

1 of 2

Create a local project

Project display name *

sonarqube

Project key *


sonarqube

Main branch name *

main

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

CancelNext



ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMoreQ

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.

5. Setup the project and come back to Jenkins Dashboard. Go to **Manage Jenkins → Plugins** and search for **SonarQube Scanner** in **Available Plugins**

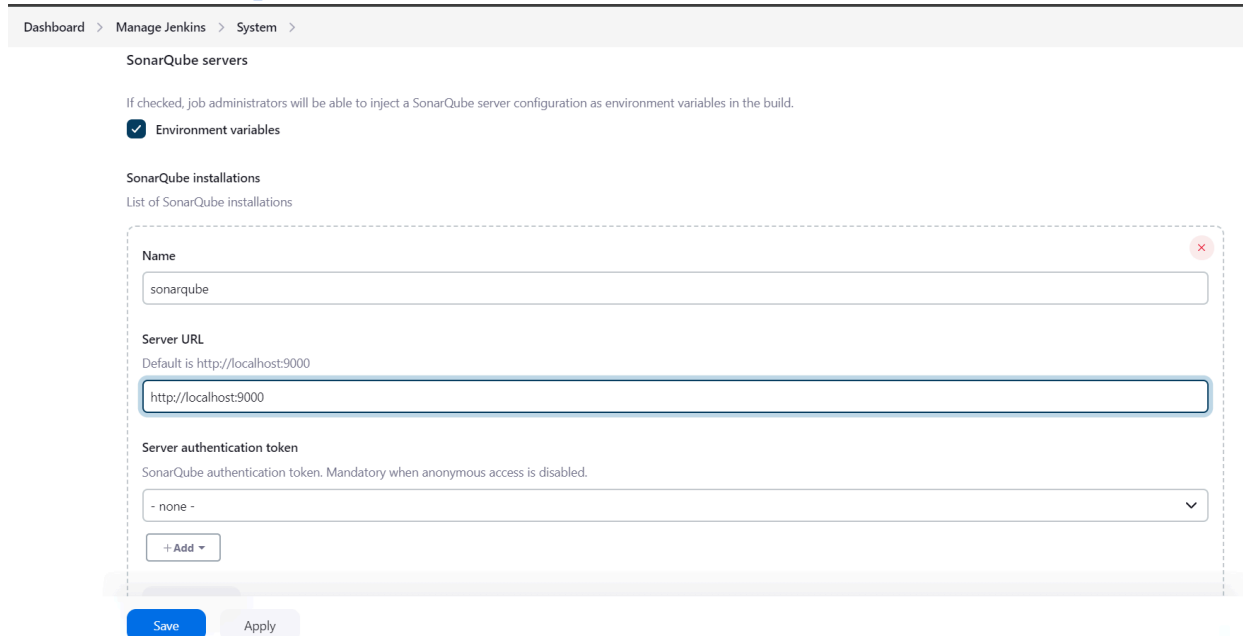


and install it.

6. Under '**Manage Jenkins → System**', look for **SonarQube Servers** and enter these details.

Name : sonarqube

Server URL : <http://localhost:9000>



7. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

Manage Jeknins → Tools → SonarQube Scanner Installation

The screenshot shows the 'SonarQube Scanner installations' configuration page in Jenkins. At the top, there is a breadcrumb trail: 'Dashboard > Manage Jenkins > Tools'. Below this, the page title is 'SonarQube Scanner installations'. There are two 'Add SonarQube Scanner' buttons. The main configuration area is a dashed box containing a 'SonarQube Scanner' section. Inside this section, there is a 'Name' field with the value 'sonarqube'. Below the name field, there is a checkbox labeled 'Install automatically' which is checked. Underneath the checkbox, there is a sub-section titled 'Install from Maven Central' which contains a 'Version' dropdown menu showing 'SonarQube Scanner 6.2.0.4584'. At the bottom of this sub-section is an 'Add Installer' button. Below the main configuration area, there is another 'Add SonarQube Scanner' button. At the very bottom of the page, there are 'Save' and 'Apply' buttons.

8. After the configuration, create a **New Item** in Jenkins, choose a **freestyle project** named **sonarqube**.

The screenshot shows the 'New Item' page in Jenkins. At the top, there is a header bar with the Jenkins logo, a search bar, and user information. Below the header, there is a breadcrumb trail: 'Dashboard > All > New Item'. The main section is titled 'New Item'. It contains a form with two main parts. The first part is 'Enter an item name' with a text input field containing 'sonarqube'. The second part is 'Select an item type' with a list of options. The first option is 'Freestyle project', which is highlighted. The other options are 'Pipeline', 'Multi-configuration project', and 'Folder'. At the bottom of the page, there is an 'OK' button.

9. Choose this GitHub repository in **Source Code Management**.

https://github.com/shazforiot/MSBuild_firstproject.git

It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

The screenshot shows the SonarQube Configuration page for a project named 'sonarqube'. The 'Configuration' tab is active, and the 'Advanced' dropdown is open. On the left, the 'Configure' sidebar shows various settings categories: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Source Code Management' section is selected. It shows two options: 'None' (unselected) and 'Git' (selected). Below this, the 'Repositories' section is expanded, showing a 'Repository URL' field with the value 'https://github.com/shazforiot/MSBuild_firstproject.git' and a 'Credentials' dropdown menu set to '- none -'. There is an '+ Add' button and an 'Advanced' dropdown. At the bottom, there is an 'Add Repository' button and 'Save' and 'Apply' buttons.

10. Under **Build-> Execute SonarQube Scanner**, enter these **Analysis Properties**. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

sonar.projectKey=sonarqube

sonar.login=admin

sonar.password=aditya

sonar.sources=.

sonar.host.url=http://localhost:9000

Dashboard > sonarqube > Configuration

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

Execute SonarQube Scanner

JDK ?
JDK to be used for this SonarQube analysis
(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=sonarqube
sonar.login=admin
sonar.password=ishan
sonar.sources=.
sonar.host.url=http://localhost:9000
```

Additional arguments ?

JVM Options ?

Save Apply

11. Go to <http://localhost:9000/admin/permissions> and allow Execute Permissions to the Admin user.

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration

Configuration Security Projects System Marketplace


Global Permissions




Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

All Users Groups Search for users or groups...


	Administer System ?	Administer ?	Execute Analysis ?	Create ?
sonar-administrators System administrators	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
sonar-users Every authenticated user automatically belongs to this group	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects
Anyone DEPRECATED Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects
A Administrator admin	<input checked="" type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects


12. Run The Build and check the console output.


 **Jenkins**


Search (CTRL+K) ?  1  Ishan Joshi  log out


Dashboard > sonarqube > #1


 Status


 Changes

 Console Output


 Edit Build Information

 Delete build '#1'


 Timings


 Git Build Data

✓ #1 (Sep 23, 2024, 9:18:22 PM)


 Add description

Keep this build forever


 Started by user [Ishan Joshi](#)

 This run spent:

- 81 ms waiting;
- 36 sec build duration;
- 36 sec total from scheduled to completion.


 **Revision:** f2bc042c04c6e72427c380bcae6d6fee7b49adf
Repository: https://github.com/shazforiot/MSBuild_firstproject.git




- refs/remotes/origin/master

 No changes.


Started 57 sec ago
Took 36 sec


REST API Jenkins 2.462.2


 **Jenkins**


Search (CTRL+K) ?  1  Ishan Joshi  log out


Dashboard > sonarqube > #1 > Console Output


 Status


 Changes

 Console Output


 Edit Build Information


 Delete build '#1'

 Timings

 Git Build Data

✓ Console Output

 Download

 Copy

View as plain text

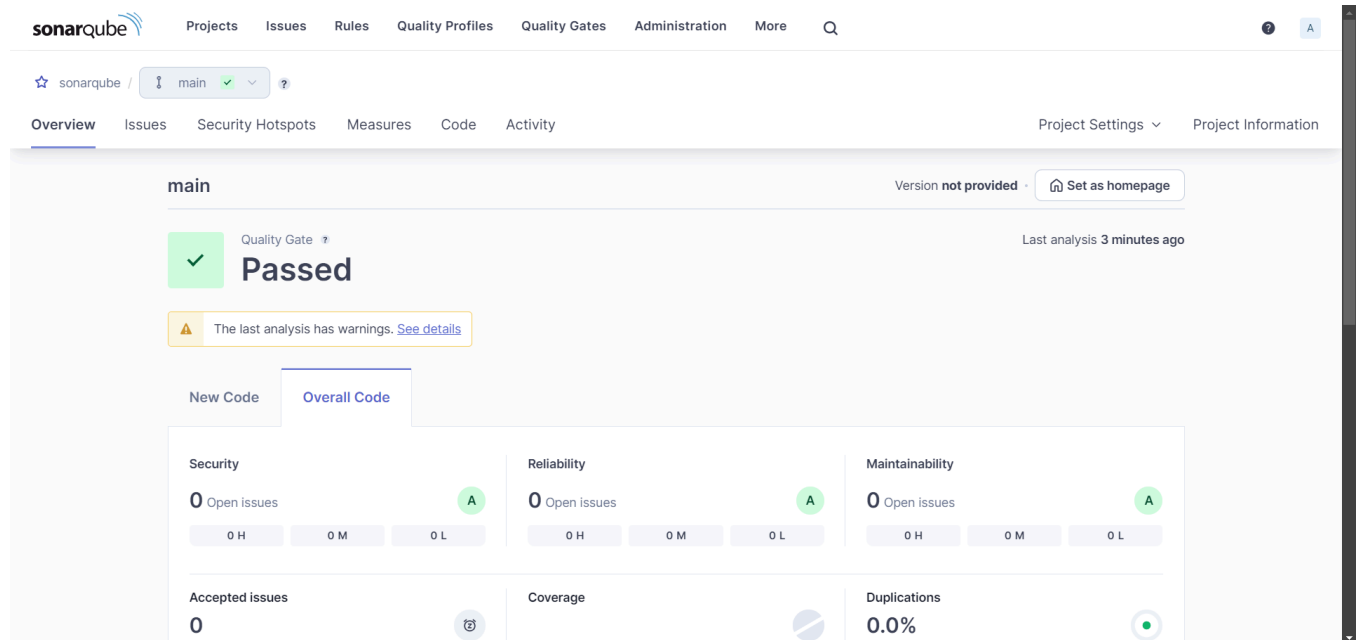
```
Started by user Ishan Joshi
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\sonarqube
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe init C:\ProgramData\Jenkins\jenkins\workspace\sonarqube # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.46.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/master^(commit)" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
Unpacking https://repo1.maven.org/maven2/org/sonarsource/scanner/cli/sonar-scanner-cli/6.2.0.4584/sonar-scanner-cli-6.2.0.4584.zip to
C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube on Jenkins
[sonarqube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube\bin\sonar-scanner.bat -
```



```
Dashboard > sonarqube > #1 > Console Output
21:18:36.004 INFO Sensor TEXTARISESECRETSSensor [LEXI] (done) | time=1000ms
21:18:56.871 INFO ----- Run sensors on project
21:18:57.082 INFO Sensor C# [csharp]
21:18:57.082 WARN Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the
SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
21:18:57.082 INFO Sensor C# [csharp] (done) | time=0ms
21:18:57.082 INFO Sensor Analysis Warnings import [csharp]
21:18:57.088 INFO Sensor Analysis Warnings import [csharp] (done) | time=6ms
21:18:57.088 INFO Sensor C# File Caching Sensor [csharp]
21:18:57.088 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir'
property.
21:18:57.089 INFO Sensor C# File Caching Sensor [csharp] (done) | time=0ms
21:18:57.089 INFO Sensor Zero Coverage Sensor
21:18:57.099 INFO Sensor Zero Coverage Sensor (done) | time=11ms
21:18:57.099 INFO SCM Publisher SCM provider for this project is: git
21:18:57.108 INFO SCM Publisher 4 source files to be analyzed
21:18:57.578 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=463ms
21:18:57.579 INFO CPD Executor Calculating CPD for 0 files
21:18:57.579 INFO CPD Executor CPD calculation finished (done) | time=0ms
21:18:57.588 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adff'
21:18:57.812 INFO Analysis report generated in 175ms, dir size=201.0 kB
21:18:57.878 INFO Analysis report compressed in 66ms, zip size=22.4 kB
21:18:58.055 INFO Analysis report uploaded in 174ms
21:18:58.058 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
21:18:58.058 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
21:18:58.058 INFO More about the report processing at http://localhost:9000/api/ce/task?id=60f78a61-aefb-4623-aa2e-0601bb2f5292
21:18:58.060 INFO Analysis total time: 16.021 s
21:18:58.060 INFO SonarScanner Engine completed successfully
21:18:58.150 INFO EXECUTION SUCCESS
21:18:58.151 INFO Total time: 25.472s
Finished: SUCCESS
```

13. Once the build is complete, check the project in SonarQube.

The screenshot displays the SonarQube web interface. The top navigation bar includes links for Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, and More. The main content area shows a list of projects, with 'sonarqube' selected. The project status is 'Passed', and the last analysis was completed 2 minutes ago. The main branch is empty. On the left sidebar, there are filters for Quality Gate (Passed: 1, Failed: 0) and Reliability (A: 1, B: 0, C: 0, D: 0, E: 0). 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.'



By following these steps, we have successfully integrated Jenkins with SonarQube for performing SAST.

Conclusion:

In this integration process, we have successfully connected Jenkins with SonarQube for Static Application Security Testing (SAST), emphasizing the importance of proactive code analysis. This setup automates the detection of security vulnerabilities and code quality issues throughout the development cycle. By implementing SAST, teams can identify potential flaws early, leading to more secure and reliable applications. Ultimately, this integration fosters a culture of continuous improvement, ensuring adherence to security best practices in software development.

