

EXP 07:

Prerequisites:

1) Docker

Run docker -v command.

```
Last login: Sat Sep 14 19:35:44 on ttys004
[(base) krushikeshsunilshelar@Krushikeshs-MacBook-Air ~ % docker -v
Docker version 27.0.3, build 7d4bcd8
```

2) Install sonarqube image

Command: docker pull sonarqube

```
[(base) krushikeshsunilshelar@Krushikeshs-MacBook-Air ~ % docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
4be1db8bbbeb: Pull complete
8cc429601029: Pull complete
f3f704211ab9: Pull complete
cbee39a89b4f: Pull complete
5d25eb3700d3: Pull complete
3098d7b8f6ca: Pull complete
b5ea2a39b0fb: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Downloaded newer image for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
```

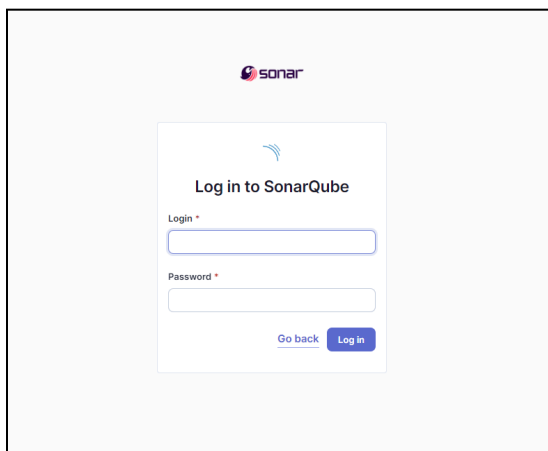
3) Run sonarqube image

docker run -d --name sonarqube -e

SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest


```
(base) krushikeshsunilshelar@Krushikeshs-MacBook-Air ~ % docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
4d9bb0253ea4b9a05f65cbf91cd1f09286915bca2c999daf42dc1088157ba855
```

4) Run localhost:9000



- 5) Login using username="admin", password="admin". It will prompt you to set a new password.

Update your password

 This account should not use the default password.

Enter a new password

All fields marked with * are required

Old Password *


New Password *

Confirm Password *

[Update](#)

- 6) This is the interface. Create a local project with name sonarqube-test.


localhost:9000/projects/create


 Projects Issues Rules Quality Profiles Quality Gates Administration More


How do you want to create your project?


Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.


First, you need to set up a DevOps platform configuration.

 Import from Azure DevOps [Setup](#)

 Import from Bitbucket Cloud [Setup](#)


 Import from Bitbucket Server [Setup](#)

 Import from GitHub [Setup](#)

 Import from GitLab [Setup](#)

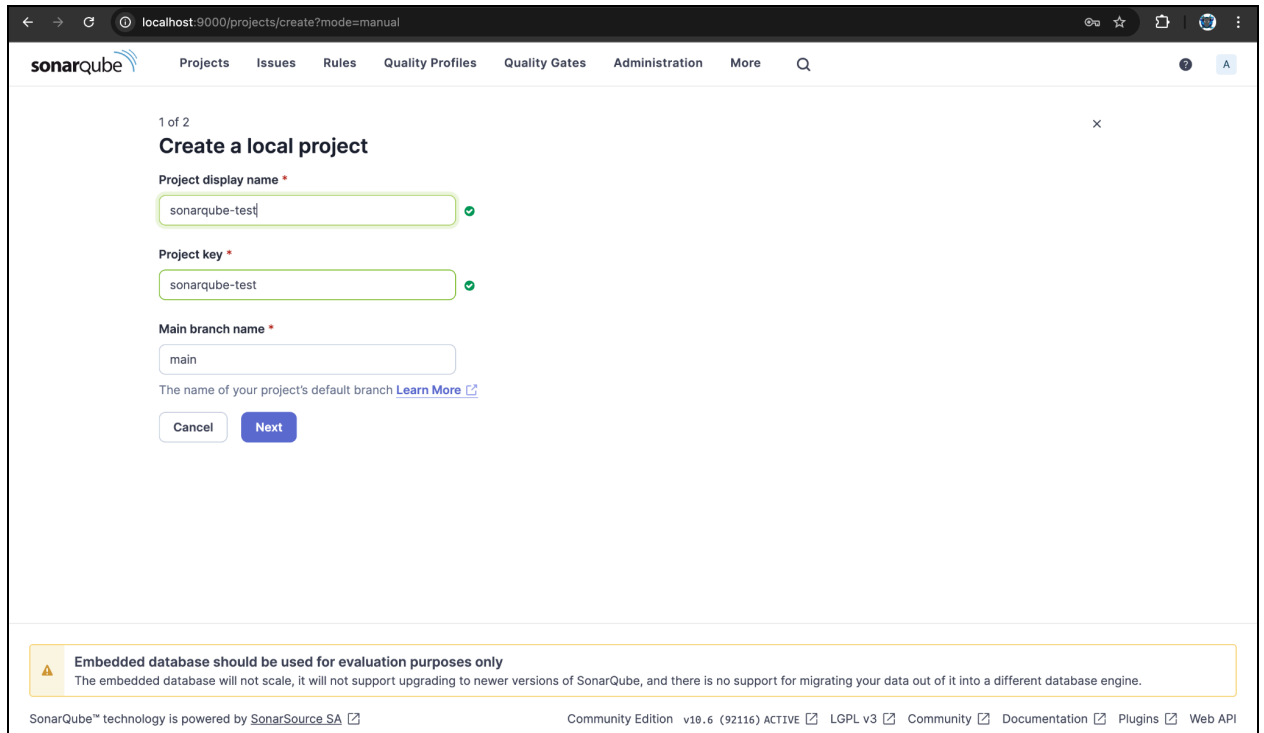
Are you just testing or have an advanced use-case? Create a local project.

[Create a local 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.

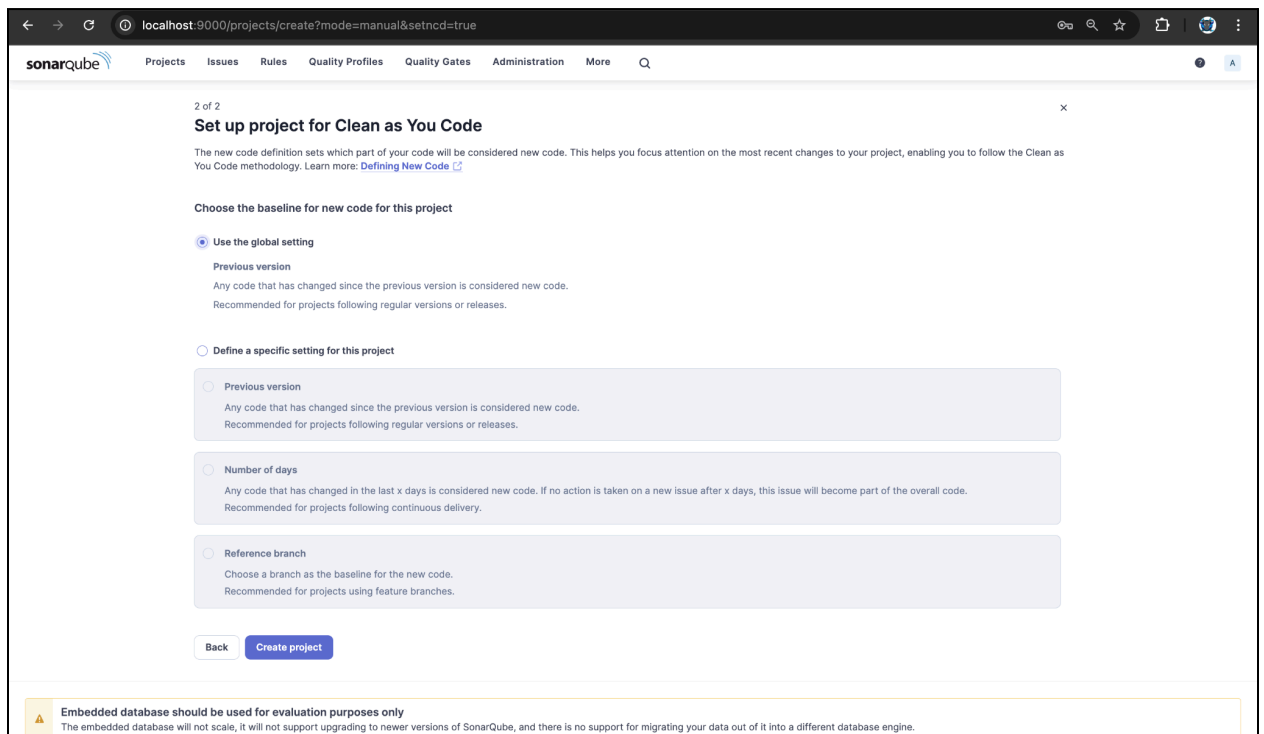
SonarQube™ technology is powered by [SonarSource SA](#)

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



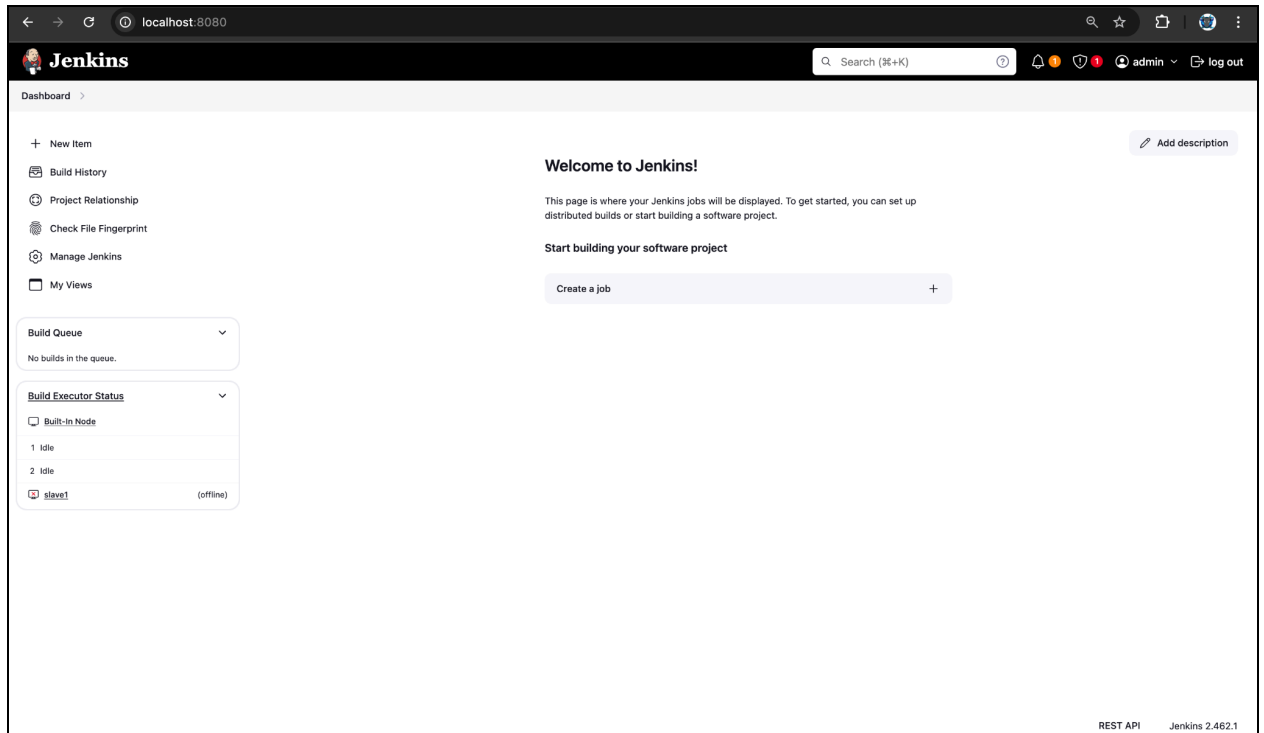
The screenshot shows the SonarQube web interface at localhost:9000/projects/create?mode=manual. The page is titled '1 of 2 Create a local project'. It contains three input fields: 'Project display name' with the value 'sonarqube-test', 'Project key' with the value 'sonarqube-test', and 'Main branch name' with the value 'main'. Each field has a green checkmark icon to its right. Below the inputs are 'Cancel' and 'Next' buttons. A warning banner 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 links for 'Community Edition v10.6 (92116) ACTIVE', 'LGPL v3', 'Community', 'Documentation', 'Plugins', and 'Web API'.

Set up the project as required and click on create.

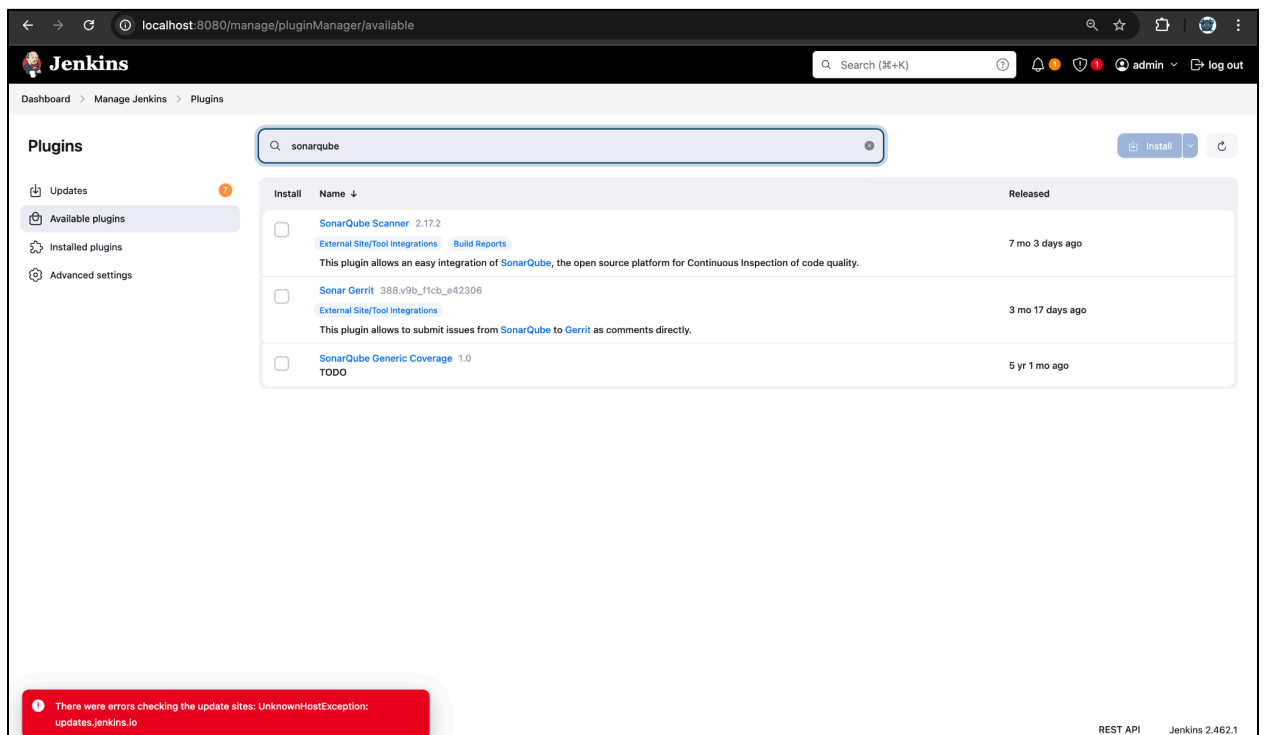


The screenshot shows the SonarQube web interface at localhost:9000/projects/create?mode=manual&setncd=true. The page is titled '2 of 2 Set up project for Clean as You Code'. It explains that the new code definition sets which part of the code will be considered new code. Under 'Choose the baseline for new code for this project', the 'Use the global setting' option is selected. Below it, three options are listed: 'Previous version' (recommended for regular versions or releases), 'Number of days' (recommended for continuous delivery), and 'Reference branch' (recommended for feature branches). At the bottom are 'Back' and 'Create project' buttons. The same warning banner about the embedded database is present at the bottom.

7) Open Jenkins dashboard using localhost on whichever port it is hosted.



8) Go to manage jenkins → Search for Sonarqube Scanner for Jenkins and install it.



- 9) Now, go to Manage Jenkins → System. Under Sonarqube servers, add a server. Add server authentication token if needed.

The screenshot shows the 'SonarQube servers' configuration page in Jenkins. At the top, there is a checkbox for 'Environment variables' which is checked. Below this, the 'SonarQube installations' section is visible, containing a list of installations. The first installation is named 'sonarqube'. Its 'Server URL' is set to 'http://localhost:9000'. The 'Server authentication token' is set to '- none -'. There are buttons for '+ Add', 'Advanced', and 'Add SonarQube' at the bottom.

- 10) Go to Manage Jenkins → Tools. Go to SonarQube scanner, choose the latest configuration and choose install automatically.


The screenshot shows the 'SonarQube Scanner installations' configuration page in Jenkins. At the top, there is a button for 'Add SonarQube Scanner'. Below this, the 'SonarQube Scanner' section is visible, containing a list of installations. The first installation is named 'sonarqube-scanner'. It has a checkbox for 'Install automatically' which is checked. Below this, there is a section for 'Install from Maven Central' with a dropdown menu for 'Version' set to 'SonarQube Scanner 6.2.0.4584'. There are buttons for 'Add Installer' and 'Add SonarQube Scanner' at the bottom.


- 11) After configuration, create a New Item → choose a freestyle project.

New Item

Enter an item name

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.

OK

12) Use this github repository in Source Code Management.

https://github.com/shazforiot/MSBuild_firstproject

It is a sample hello-world project with no vulnerabilities.

← → ↻ 📄 localhost:8080/job/sonarqube-51/configure

Dashboard > sonarqube-51 > Configuration

Advanced ▾

Configure

⚙️ General

🔗 Source Code Management

🕒 Build Triggers

🌐 Build Environment

📋 Build Steps

🔧 Post-build Actions

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

- none - ▾

+ Add ▾

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Add Branch

Repository browser ?

(Auto) ▾

Additional Behaviours

Add ▾

13) Under Build Steps, enter Sonarqube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

localhost:8080/job/sonarqube-51/configure

Dashboard > sonarqube-51 > Configuration

Configure

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

☐ Use secret text(s) or file(s) ?

☐ Prepare SonarQube Scanner environment ?

Build Steps

Execute SonarQube Scanner

JDK ?
JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

Analysis properties ?

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

Additional arguments ?

JVM Options ?

Add build step

14) Go to http://localhost:9000/<user_name>/permissions and allow Execute Permissions to admin user.

localhost:9000/admin/permissions

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration

Configuration Security Projects System Marketplace

Global Permissions

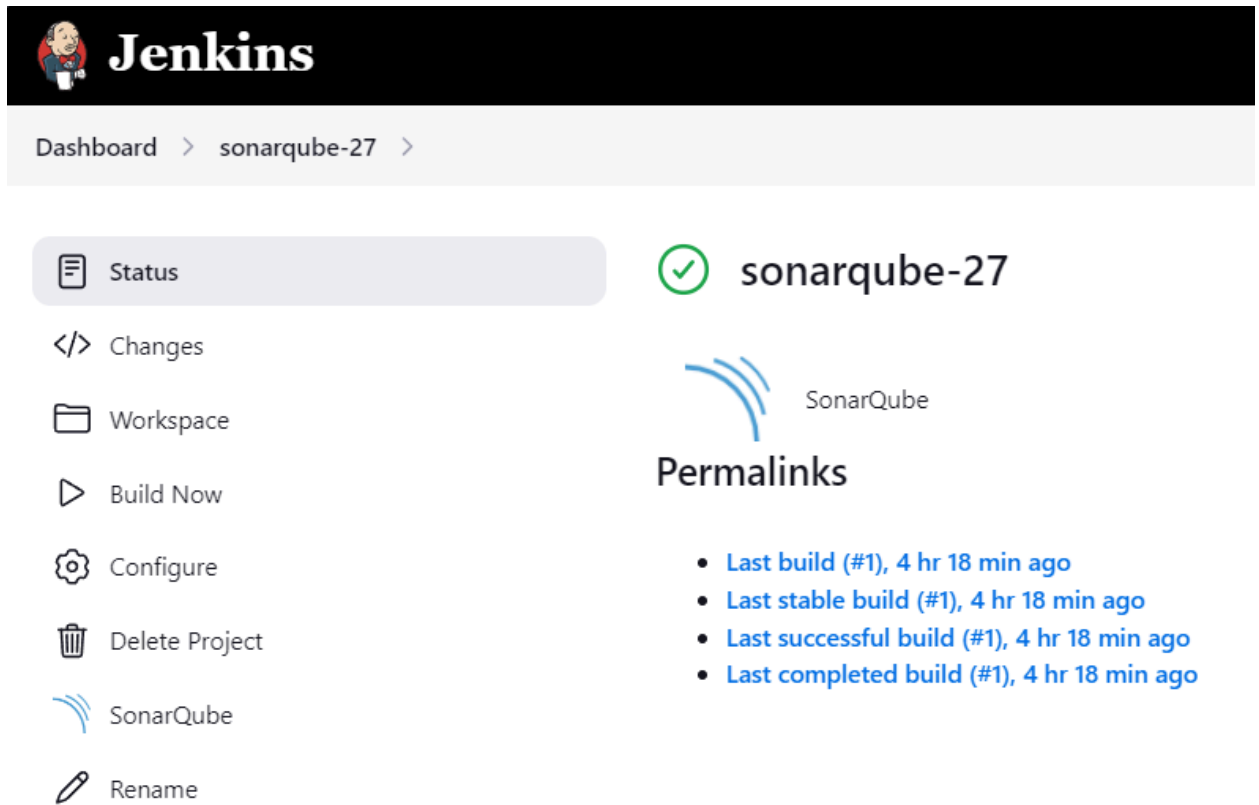
Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

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 type="checkbox"/> Projects

4 of 4 shown


15) Run the Jenkins build.





The image shows the Jenkins web interface. At the top is a black header with the Jenkins logo and name. Below it is a breadcrumb trail: Dashboard > sonarqube-27 >. On the left is a sidebar with icons and labels for Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main area on the right shows a green checkmark icon next to the project name 'sonarqube-27'. Below this is the SonarQube logo and the heading 'Permalinks'. A list of four links follows, all indicating a build completed 4 hours and 18 minutes ago.


Jenkins


Dashboard > sonarqube-27 >


 Status


 Changes


 Workspace


 Build Now


 Configure

 Delete Project

 SonarQube

 Rename

 **sonarqube-27**

 SonarQube

Permalinks

- [Last build \(#1\), 4 hr 18 min ago](#)
- [Last stable build \(#1\), 4 hr 18 min ago](#)
- [Last successful build \(#1\), 4 hr 18 min ago](#)
- [Last completed build \(#1\), 4 hr 18 min ago](#)

Check the console Output

Console Output

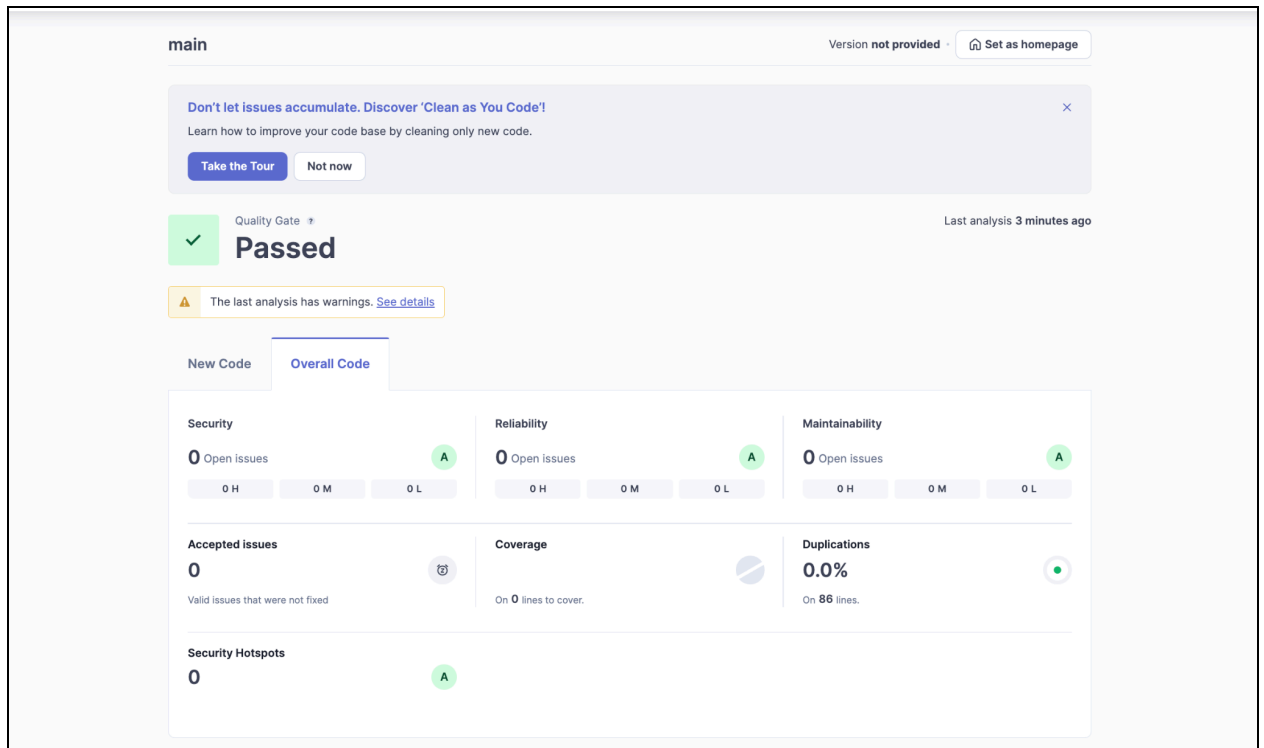
DownloadCopyView as plain text

```
Started by user admin
Running as SYSTEM
Building on the built-in node in workspace /Users/krushikeshsnilshelar/.jenkins/workspace/sonarqube-51
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/shazforiot/MSBuild_firstproject
> git init /Users/krushikeshsnilshelar/.jenkins/workspace/sonarqube-51 # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject
> git --version # timeout=10
> git --version # 'git version 2.39.3 (Apple Git-146)'
> git fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f f2bc042c04c6e72427c380bcaee6d6fee7b49adf # 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
/Users/krushikeshsnilshelar/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqube-scanner on Jenkins
[sonarqube-51] $ /Users/krushikeshsnilshelar/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqube-scanner/bin/sonar-scanner -Dsonar.host.url=http://localhost:9000 -
Dsonar.projectkey=sonarqube-test -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=Rushi@1418 -
Dsonar.projectBaseDir=/Users/krushikeshsnilshelar/.jenkins/workspace/sonarqube-51
10:15:43.472 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
10:15:43.475 INFO Scanner configuration file: /Users/krushikeshsnilshelar/.jenkins/tools/hudson.plugins.sonar.SonarRunnerInstallation/sonarqube-scanner/conf/sonar-scanner.properties
10:15:43.476 INFO Project root configuration file: NONE
10:15:43.486 INFO SonarScanner CLI 6.2.0.4584
10:15:43.486 INFO Java 17.0.8 Oracle Corporation (64-bit)
10:15:43.487 INFO Mac OS X 14.6.1 aarch64
10:15:43.525 INFO User cache: /Users/krushikeshsnilshelar/.sonar/cache
10:15:44.070 INFO JRE provisioning: os[macos], arch[arm64]
10:15:47.173 INFO Communicating with SonarQube Server 10.6.0.92116
10:15:47.342 INFO Starting SonarScanner Engine...
10:15:47.342 INFO Java 17.0.11 Eclipse Adoptium (64-bit)
10:15:47.691 INFO Load global settings

10:16:13.003 INFO Sensor IaC Docker Sensor [iac] (done) | time=11ms
10:16:13.003 INFO Sensor TextAndSecretsSensor [text]
10:16:13.003 INFO Available processors: 8
10:16:13.003 INFO Using 8 threads for analysis.
10:16:13.164 INFO The property "sonar.tests" is not set. To improve the analysis accuracy, we categorize a file as a test file if any of the following is true:
    * The filename starts with "test"
    * The filename contains "test." or "tests."
    * Any directory in the file path is named: "doc", "docs", "test" or "tests"
    * Any directory in the file path has a name ending in "test" or "tests"

10:16:13.182 INFO Using git CLI to retrieve untracked files
10:16:13.196 INFO Analyzing language associated files and files included via "sonar.text.inclusions" that are tracked by git
10:16:13.219 INFO 14 source files to be analyzed
10:16:13.347 INFO 14/14 source files have been analyzed
10:16:13.347 INFO Sensor TextAndSecretsSensor [text] (done) | time=344ms
10:16:13.349 INFO Run sensors on project
10:16:13.372 INFO Sensor C# [csharp]
10:16:13.372 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
10:16:13.372 INFO Sensor C# [csharp] (done) | time=1ms
10:16:13.372 INFO Sensor Analysis Warnings Import [csharp]
10:16:13.372 INFO Sensor Analysis Warnings Import [csharp] (done) | time=0ms
10:16:13.372 INFO Sensor C# File Caching Sensor [csharp]
10:16:13.372 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
10:16:13.372 INFO Sensor C# File Caching Sensor [csharp] (done) | time=0ms
10:16:13.372 INFO Sensor Zero Coverage Sensor
10:16:13.374 INFO Sensor Zero Coverage Sensor (done) | time=2ms
10:16:13.379 INFO CPD Executor Calculating CPD for 0 files
10:16:13.379 INFO CPD Executor CPD calculation finished (done) | time=0ms
10:16:13.380 INFO SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
10:16:13.411 INFO Analysis report generated in 27ms, dir size=200.4 kB
10:16:13.421 INFO Analysis report compressed in 9ms, zip size=22.1 kB
10:16:13.600 INFO Analysis report uploaded in 179ms
10:16:13.601 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
10:16:13.601 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
10:16:13.601 INFO More about the report processing at http://localhost:9000/api/ce/task?id=5f495a8e-aea7-4beb-9bd3-e3faa2fd4347
10:16:13.607 INFO Analysis total time: 10.105 s
10:16:13.607 INFO SonarScanner Engine completed successfully
10:16:13.653 INFO EXECUTION SUCCESS
10:16:13.655 INFO Total time: 12.133s
Finished: SUCCESS
```

16) Once the build is complete, check project on SonarQube



Conclusion:

In this experiment, we have understood the importance of SAST and have successfully integrated Jenkins with SonarQube for Static Analysis and Code Testing.