

### Experiment No 7

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

Step 1: Make sure you download all the Prerequisites required :

- Jenkins installed
- Docker Installed (for SonarQube)
- SonarQube Docker Image

And then run the following command on your window powershell/cmd prompt only once:

```
$ docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true  
-p 9000:9000 Sonarqube:latest
```

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows  
  
PS C:\Users\Govind> 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  
7b87d6fa783d: Pull complete  
bd819c9b5ead: Pull complete  
4f4fb700ef54: Pull complete  
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde  
Status: Downloaded newer image for sonarqube:latest  
f7b8d5f7157e7cb2643b22c497a78368667800258325d55d236330b6b6286f85
```

Step 2: Once the container is up and running, you can check the status of SonarQube at localhost port 9000.

SonarQube is starting



Step 3: Login to SonarQube using username admin and password admin then create a local project in SonarQube with the name sonarqube-test.

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

Step 4: Setup the project to use the global setting and open Jenkins Dashboard.

## 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.

Step 5: Now, Go to Dashboard ->Manage Jenkins -> Plugin Manager and search for SonarQube Scanner under Available plugins for Jenkins and install it.

sona



Name ↓

Enabled

SonarQube Scanner for Jenkins 2.17.2

This plugin allows an easy integration of [SonarQube](#), the open source platform for Continuous Inspection of code quality.

[Report an issue with this plugin](#)



### Plugins



Updates

28



Available plugins



Installed plugins



Advanced settings



Download progress

### Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner

✓ Success

Loading plugin extensions

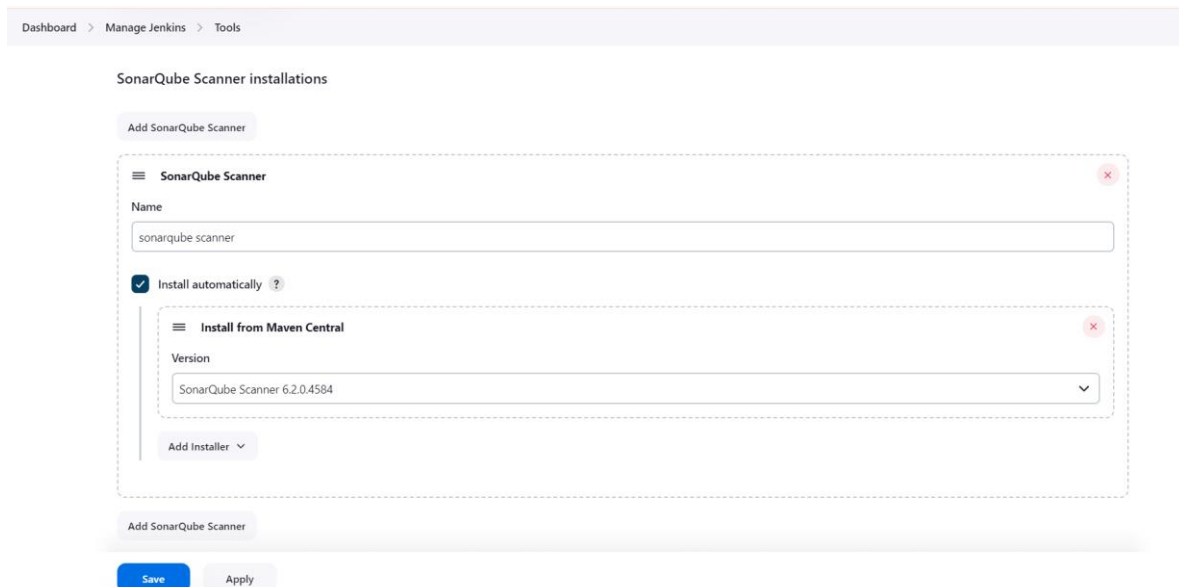
✓ Success

→ [Go back to the top page](#)

(you can start using the installed plugins right away)

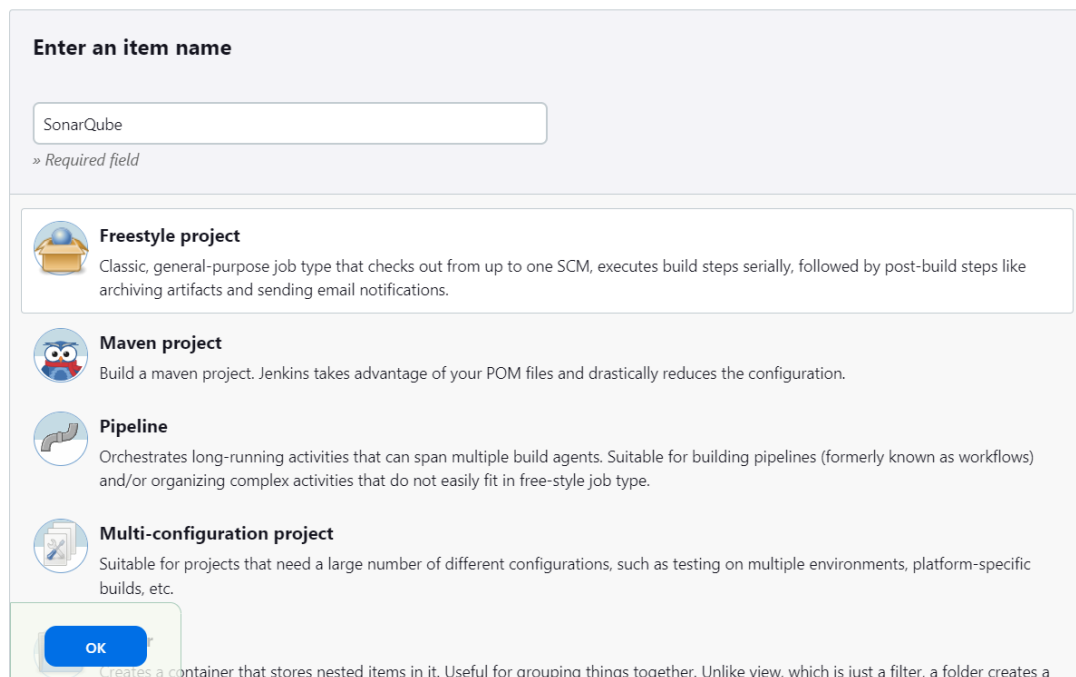
→ ☐ Restart Jenkins when installation is complete and no jobs are running

Step 6: Under Jenkins 'Configure System', look for SonarQube Servers and enter the details then Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.



The screenshot shows the Jenkins 'SonarQube Scanner installations' configuration page. At the top, there is a breadcrumb trail: 'Dashboard > Manage Jenkins > Tools'. The main heading is 'SonarQube Scanner installations'. Below this, there is a button 'Add SonarQube Scanner'. A dashed box contains the configuration for a 'SonarQube Scanner'. Inside this box, there is a 'Name' field with the value 'sonarqube scanner'. Below the name field, there is a checkbox 'Install automatically' which is checked. Underneath, 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 dashed box is a button 'Add Installer'. Below the dashed box, there is another 'Add SonarQube Scanner' button. At the very bottom of the page are two buttons: 'Save' and 'Apply'.

Step 7: Create a New Item in Jenkins and choose a freestyle project.



The screenshot shows the 'Enter an item name' dialog in Jenkins. At the top, the title is 'Enter an item name'. Below the title is a text input field containing the text 'SonarQube'. Below the input field is a small text label '» Required field'. Below the input field, there is a list of project types. Each item in the list has an icon, a title, and a description. The items are: '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.), and 'Multi-configuration project' (Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.). At the bottom of the dialog is a blue button labeled 'OK'.

Step 7: Choose this GitHub repository in Source Code Management.

[https://github.com/shazforiot/MSBuild\\_firstproject.git](https://github.com/shazforiot/MSBuild_firstproject.git)

Under Build ->Execute SonarQube Scanner, enter these Analysis

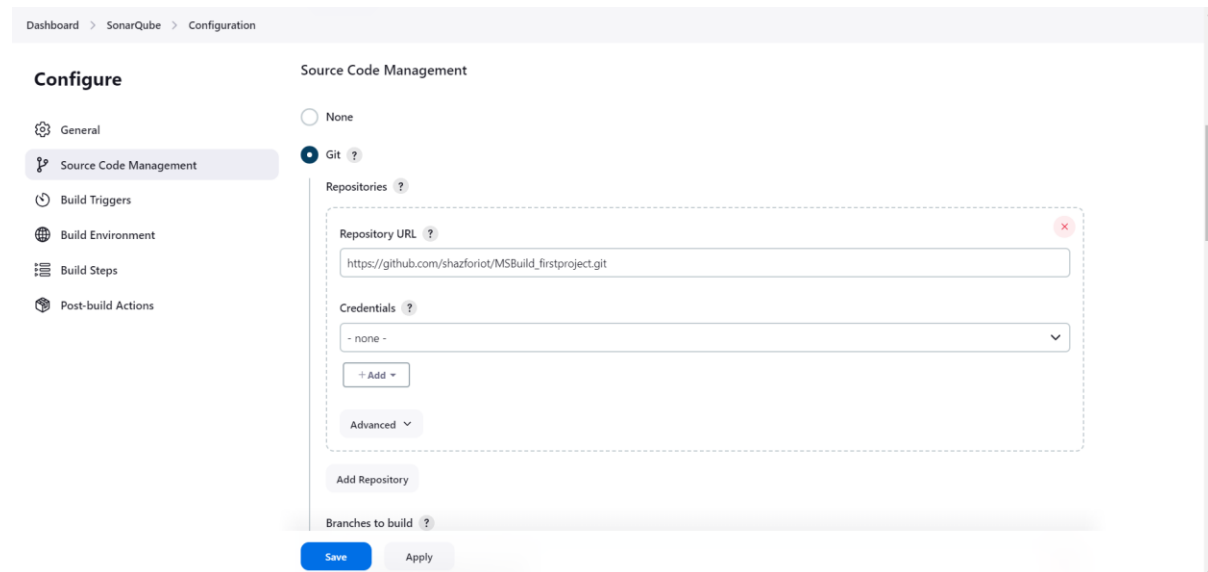
properties. Mention the SonarQube Project Key, Login, Password, and Host URL.

sonar.projectKey=sonarqube-test

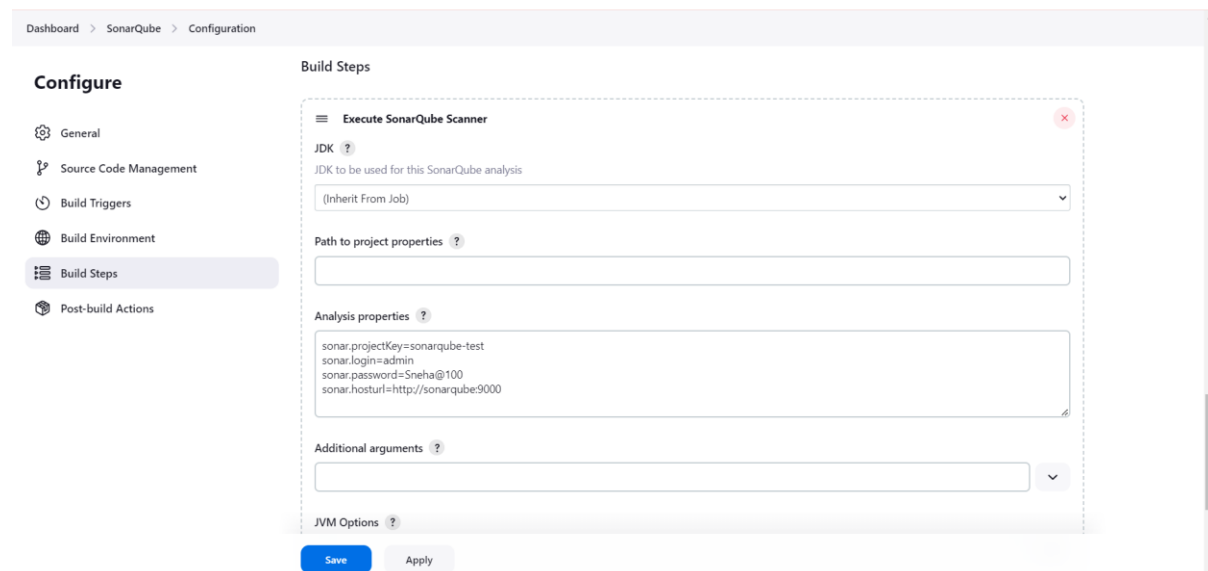
sonar.login=admin

sonar.password=Sneha@100

sonar.hosturl=<http://localhost:9000/>



The screenshot shows the 'Source Code Management' configuration page in SonarQube. The left sidebar has 'Configure' at the top, followed by 'General', 'Source Code Management' (selected), 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The main area is titled 'Source Code Management' and has two radio buttons: 'None' and 'Git' (selected). Below the 'Git' button is a 'Repositories' section with a 'Repository URL' field containing 'https://github.com/shazforiot/MSBuild\_firstproject.git' and a 'Credentials' dropdown menu set to '- none -'. There is an 'Add Repository' button and an 'Advanced' dropdown. At the bottom, there is a 'Branches to build' field and 'Save' and 'Apply' buttons.



The screenshot shows the 'Build Steps' configuration page in SonarQube. The left sidebar is the same as the previous screenshot, with 'Build Steps' selected. The main area is titled 'Build Steps' and has a section 'Execute SonarQube Scanner'. This section contains a 'JDK' dropdown menu set to '(Inherit From Job)', a 'Path to project properties' field, and an 'Analysis properties' field containing the following text: 'sonar.projectKey=sonarqube-test', 'sonar.login=admin', 'sonar.password=Sneha@100', and 'sonar.hosturl=http://sonarqube:9000'. There is also an 'Additional arguments' dropdown menu and a 'JVM Options' field. At the bottom, there are 'Save' and 'Apply' buttons.

Step 8: Go to <http://localhost:9000/> and enter your previously created username and password then Go to Permissions and grant the Admin user Execute Permissions.

**sonarqube** Projects Issues Rules Quality Profiles Quality Gates Administration More Q

**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 ?
A Administrator admin	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input type="checkbox"/> Projects

## Step 9: Now Build and Run.

**Jenkins** 🔍 Search (CTRL+K) 🔔 1 🔒 2 👤 Sneha Patra ↕ log out

Dashboard > SonarQube >

Status

</> Changes

Workspace

▶ Build Now

⚙️ Configure

🗑️ Delete Project

🔌 SonarQube

✎ Rename

Build History trend ▾

🔍 Filter... /

🟢 #2  
Sep 25, 2024, 6:35 PM

🔴 #1  
Sep 25, 2024, 6:15 PM

🟢 **SonarQube**

🔌 SonarQube

**Permalinks**

- Last build (#2), 13 min ago
- Last stable build (#2), 13 min ago
- Last successful build (#2), 13 min ago
- Last failed build (#1), 32 min ago
- Last unsuccessful build (#1), 32 min ago
- Last completed build (#2), 13 min ago

✎ Add description

Disable Project

**Jenkins** 🔍 Search (CTRL+K) 🔔 1 🔒 2 👤 Sneha Patra ↕ log out

Dashboard > SonarQube > #2

Status

</> Changes

📄 Console Output

✎ Edit Build Information

🗑️ Delete build '#2'

🕒 Timings

🔌 Git Build Data

⬅ Previous Build

🟢 **#2 (Sep 25, 2024, 6:35:00 PM)**

Keep this build forever

</> No changes.

🕒 Started by user [Sneha Patra](#)

🕒 This run spent:

- 5 ms waiting;
- 1 min 16 sec build duration;
- 1 min 16 sec total from scheduled to completion.

🔌 git

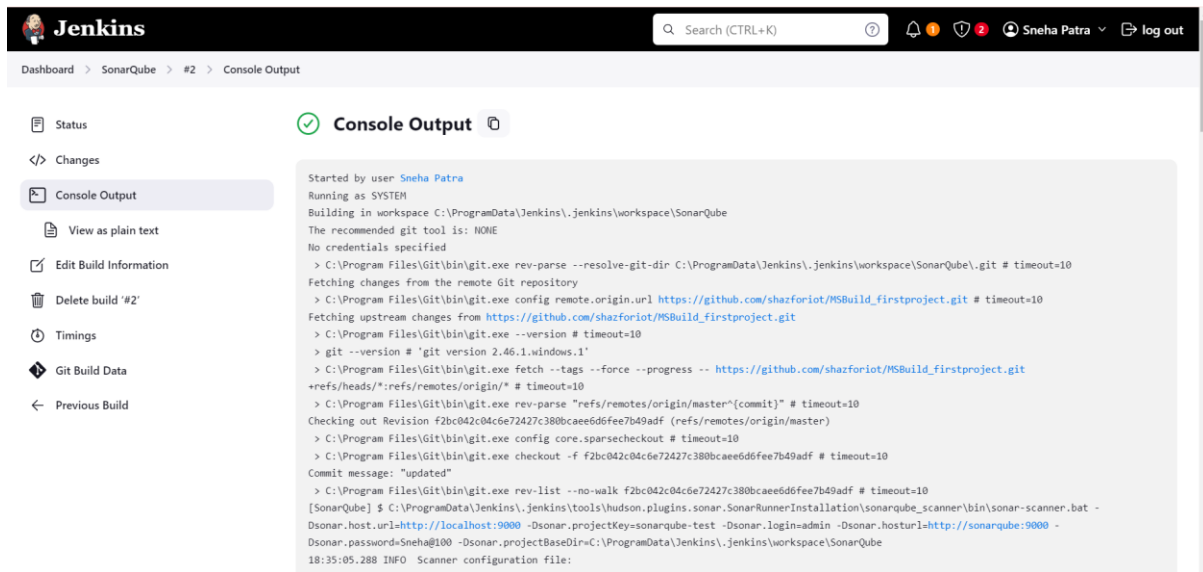
Revision: f2bc042c04c6e72427c380bcaee6d6fee7b49adf  
Repository: [https://github.com/shazforiot/MSBuild\\_firstproject.git](https://github.com/shazforiot/MSBuild_firstproject.git)

- refs/remotes/origin/master

✎ Add description

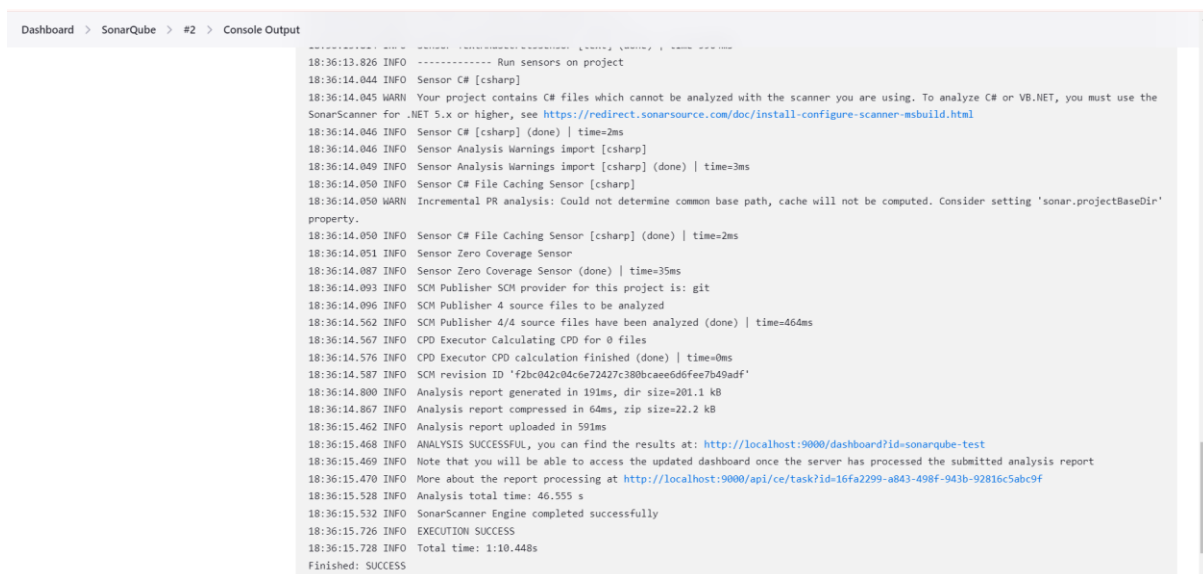
Started 14 min ago  
Took **1 min 16 sec**

## Step 10: Check the console output.



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information (Sneha Patra). The breadcrumb trail is Dashboard > SonarQube > #2 > Console Output. On the left sidebar, the 'Console Output' tab is selected. The main content area displays the console output for build #2, which includes the following text:

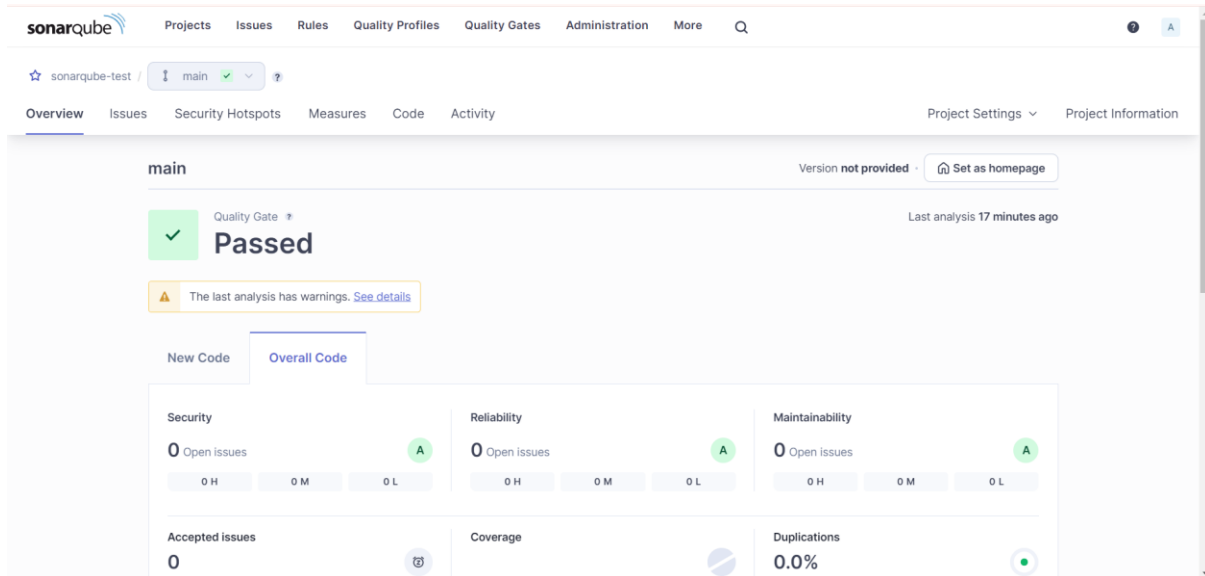
```
Started by user Sneha Patra
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
The recommended git tool is: NONE
No credentials specified
> C:\Program Files\Git\bin\git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> C:\Program Files\Git\bin\git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> C:\Program Files\Git\bin\git.exe fetch --tags --progress -- https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
> git --version # 'git version 2.46.1.windows.1'
> C:\Program Files\Git\bin\git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
> C:\Program Files\Git\bin\git.exe rev-parse --refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> C:\Program Files\Git\bin\git.exe config core.sparsecheckout # timeout=10
> C:\Program Files\Git\bin\git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
> C:\Program Files\Git\bin\git.exe rev-list --no-walk f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
[SonarQube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube-scanner\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube-test -Dsonar.login=admin -Dsonar.hosturl=http://sonarqube:9000 -
Dsonar.password=Sneha@100 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
18:35:05.288 INFO Scanner configuration file:
```



The screenshot continues the console output from the previous build. It shows the execution of various sensors and the final analysis results. The output includes the following text:

```
18:36:13.826 INFO ----- Run sensors on project
18:36:14.044 INFO Sensor C# [csharp]
18:36:14.045 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
18:36:14.046 INFO Sensor C# [csharp] (done) | time=2ms
18:36:14.046 INFO Sensor Analysis Warnings import [csharp]
18:36:14.049 INFO Sensor Analysis Warnings import [csharp] (done) | time=3ms
18:36:14.050 INFO Sensor C# File Caching Sensor [csharp]
18:36:14.050 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir'
property.
18:36:14.050 INFO Sensor C# File Caching Sensor [csharp] (done) | time=2ms
18:36:14.051 INFO Sensor Zero Coverage Sensor
18:36:14.087 INFO Sensor Zero Coverage Sensor (done) | time=35ms
18:36:14.093 INFO SCM Publisher SCM provider for this project is: git
18:36:14.096 INFO SCM Publisher 4 source files to be analyzed
18:36:14.562 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=464ms
18:36:14.567 INFO CPD Executor Calculating CPD for 0 files
18:36:14.576 INFO CPD Executor CPD calculation finished (done) | time=0ms
18:36:14.587 INFO SCM revision ID 'f2bc042c04c6e72427c380bcae6d6fee7b49adf'
18:36:14.800 INFO Analysis report generated in 191ms, dir size=201.1 kB
18:36:14.867 INFO Analysis report compressed in 64ms, zip size=22.2 kB
18:36:15.462 INFO Analysis report uploaded in 591ms
18:36:15.468 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
18:36:15.469 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
18:36:15.470 INFO More about the report processing at http://localhost:9000/api/ce/task?id=16fa2299-a843-496f-943b-92816c5ab9f
18:36:15.528 INFO Analysis total time: 46.555 s
18:36:15.532 INFO SonarScanner Engine completed successfully
18:36:15.726 INFO EXECUTION SUCCESS
18:36:15.728 INFO Total time: 1:10.448s
Finished: SUCCESS
```

Step 11: Once the build is successful, check the project in SonarQube. In this way, we have integrated Jenkins with SonarQube for SAST.



## Conclusion:

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