Name: Shejal Tiwari Div/Roll No:D15C/57

AdvanceDevops Experiment 7

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

Integrating Jenkins with SonarQube:

Windows installation
Step 1 Install JDK 1.8
Step 2 download and install jenkins
https://www.blazemeter.com/blog/how-to-install-jenkins-on-windows

Ubuntu installation

https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt on-ubuntu-20-04#installing-the-default-jre-jdk

Step 1 Install JDK 1.8
sudo apt-get install openjdk-8-jre

sudo apt install default-jre

https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu 20-04

Open SSH

Prerequisites:

- Jenkins installed
- Docker Installed (for SonarQube)

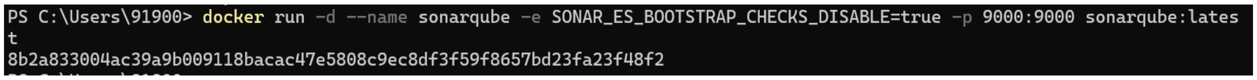
(sudo apt-get install docker-ce=5:20.10.15~3-0~ubuntu-jammy docker-ce-cli=5:20.10.15~3-0~ubuntu-jammy containerd.io docker-compose-plugin)

- SonarQube Docker Image

Steps to integrate Jenkins with SonarQube

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

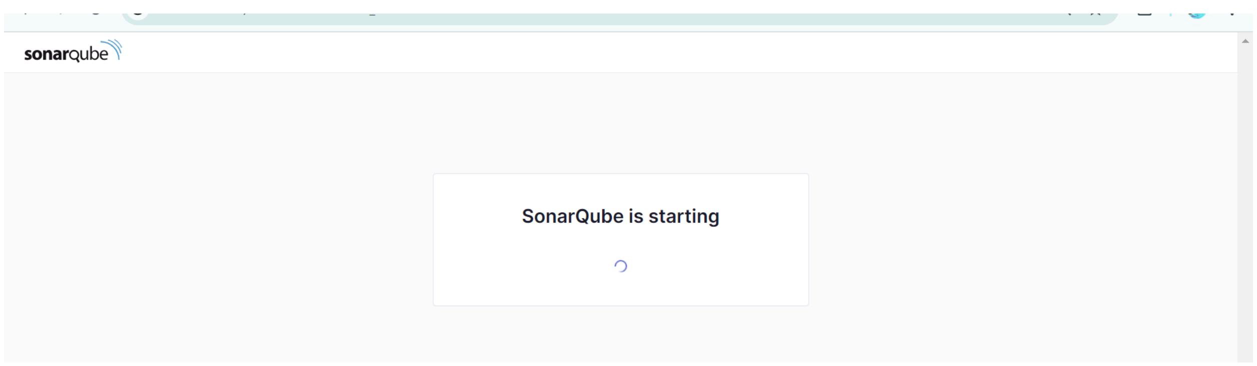
2. Run SonarQube in a Docker container using this command -



```
PS C:\Users\91900> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:lates
t
8b2a833004ac39a9b009118bacac47e5808c9ec8df3f59f8657bd23fa23f48f2
```

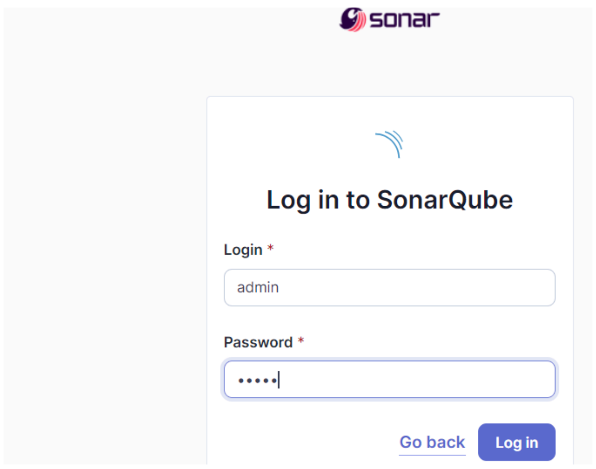
Warning: run below command only once
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

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



SonarQube is starting

4. Login to SonarQube using username *admin* and password *admin*.



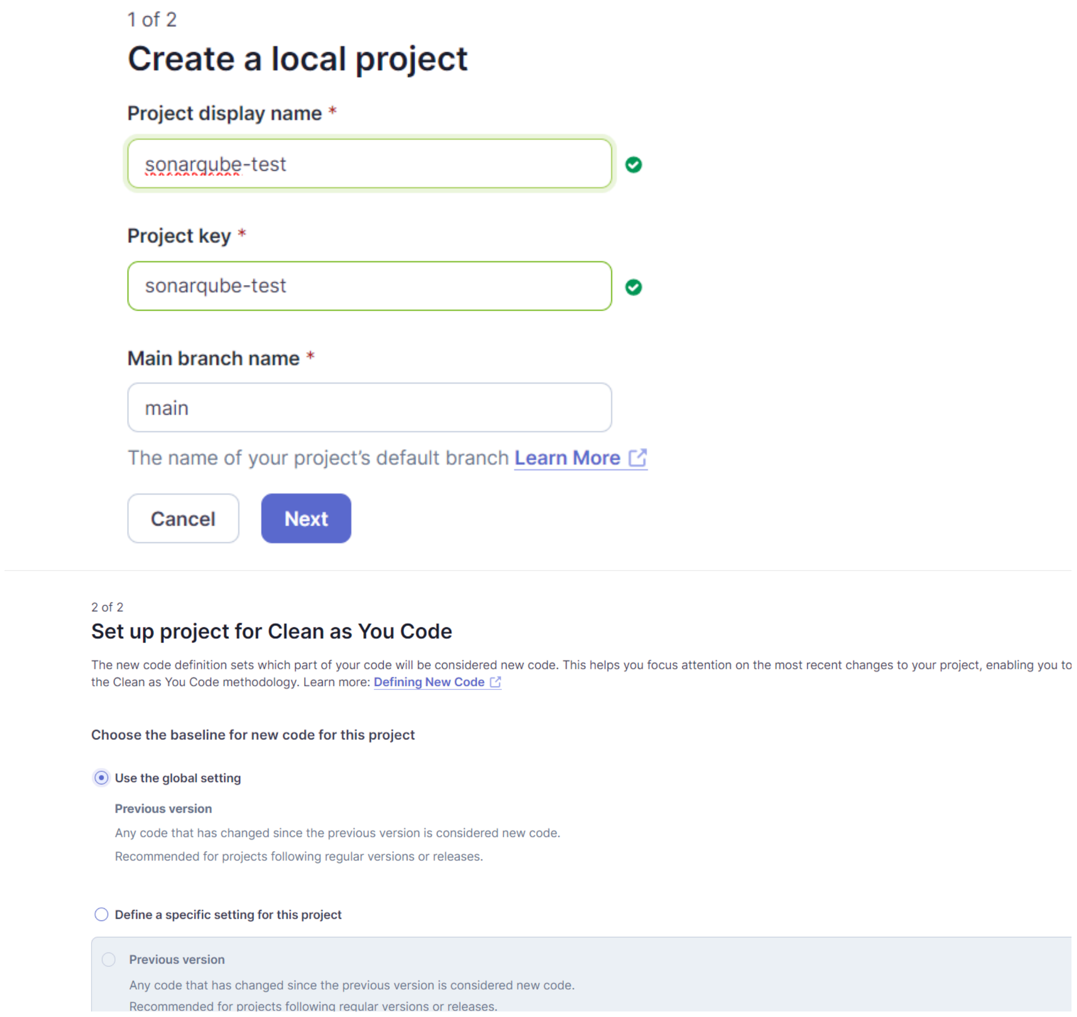
Log in to SonarQube

Login *
admin

Password *
•••••

5. Create a manual project in SonarQube with the name **sonarqube**

Name: Shejal Tiwari Div/Roll No:D15C/57


1 of 2

Create a local project

Project display name *

sonarqube-test

Project key *

sonarqube-test

Main branch name *

main

The name of your project's default branch Learn More

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 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

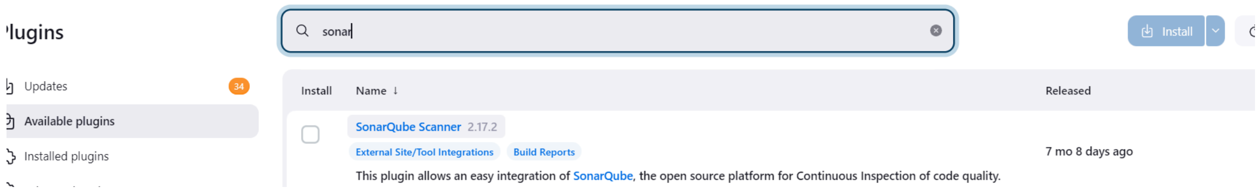
Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

Setup the project and come back to Jenkins Dashboard.

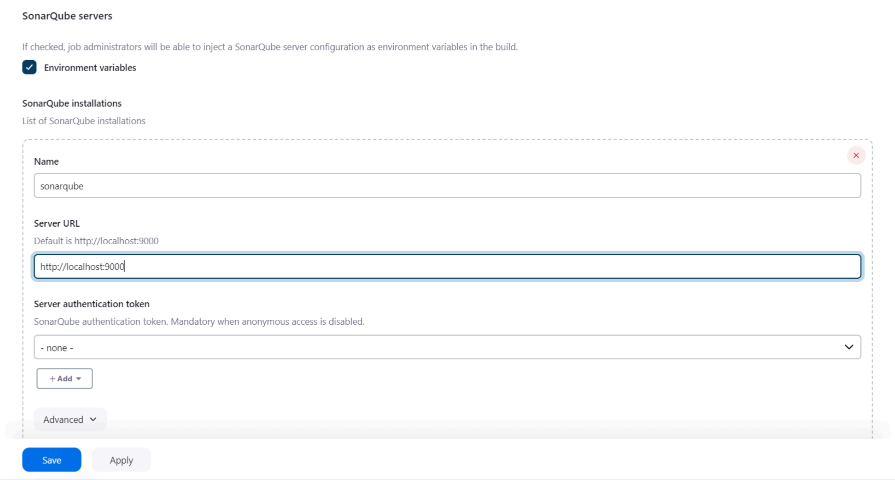
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



Install	Name ↓	Released
☐	SonarQube Scanner 2.17.2 External Site/Tool Integrations · Build Reports This plugin allows an easy integration of SonarQube, the open source platform for Continuous Inspection of code quality.	7 mo 8 days ago

6. Under Jenkins 'Configure System', look for SonarQube Servers and enter the details.

Enter the Server Authentication token if needed.



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

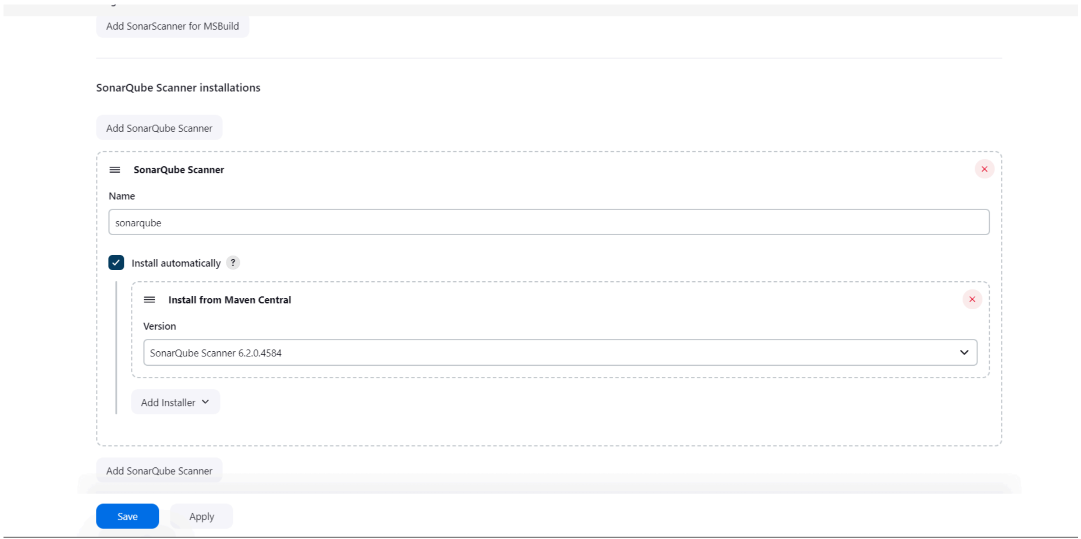
Name: sonarqube

Server URL: http://localhost:9000 (Default is http://localhost:9000)

Server authentication token: - none - (SonarQube authentication token. Mandatory when anonymous access is disabled.)

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

.



SonarQube Scanner installations

SonarQube Scanner

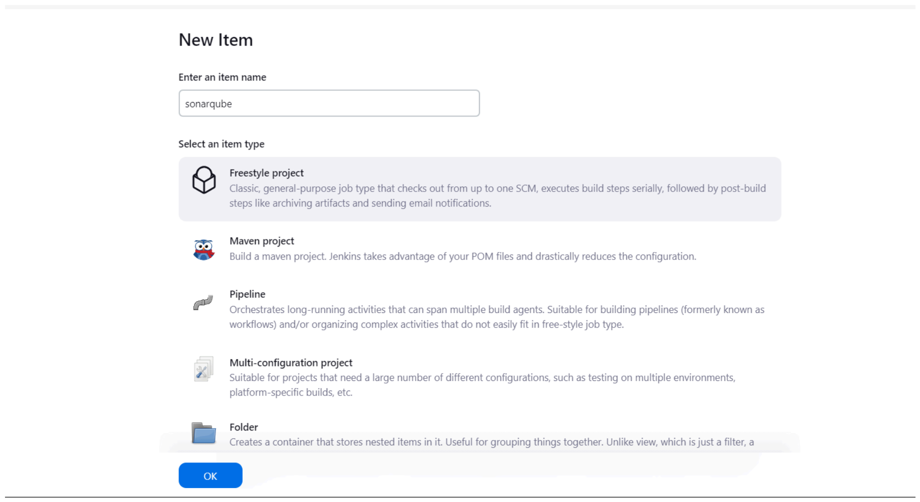
Name: sonarqube

✓ Install automatically

Install from Maven Central

Version: SonarQube Scanner 6.2.0.4584

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


New Item

Enter an item name

sonarqube

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.

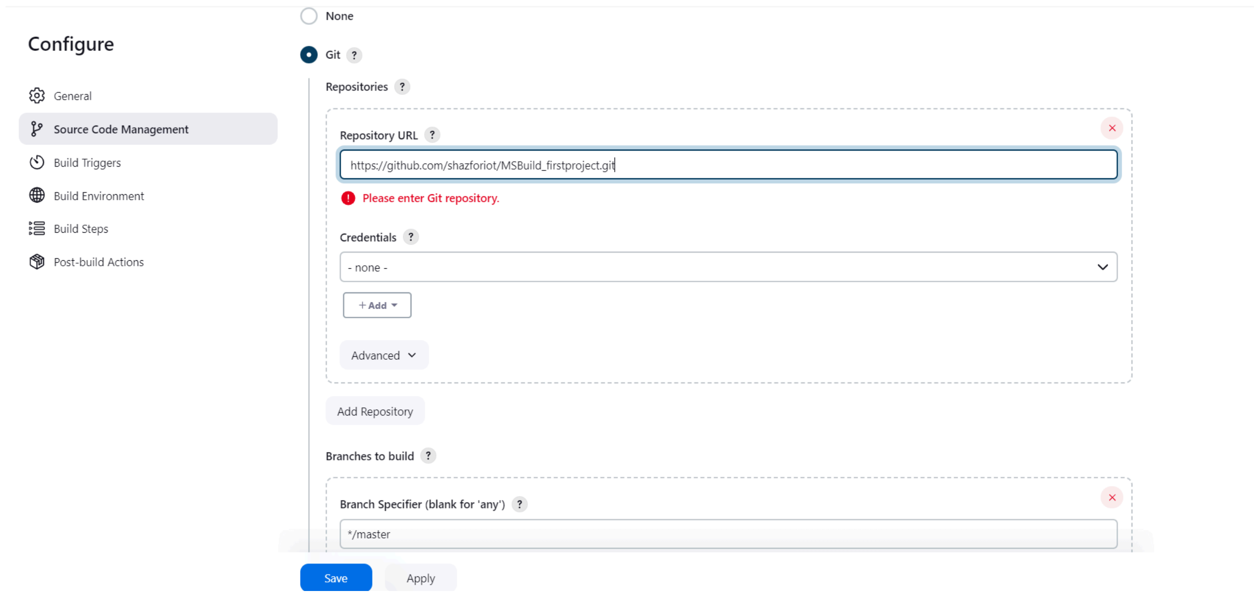
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

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



None
[✓ Git]

Repositories

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

Please enter Git repository.

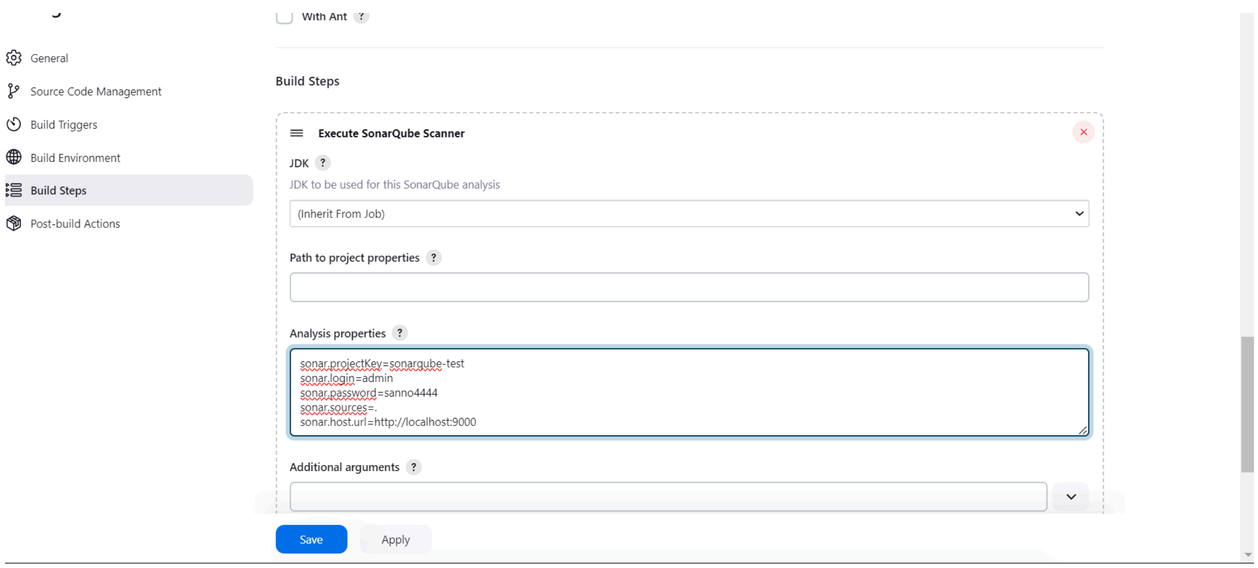
Credentials
- none -

Branches to build

Branch Specifier (blank for 'any')
*/master

the integration.

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



Build Steps

Execute SonarQube Scanner

JDK

JDK to be used for this SonarQube analysis

(Inherit From Job)

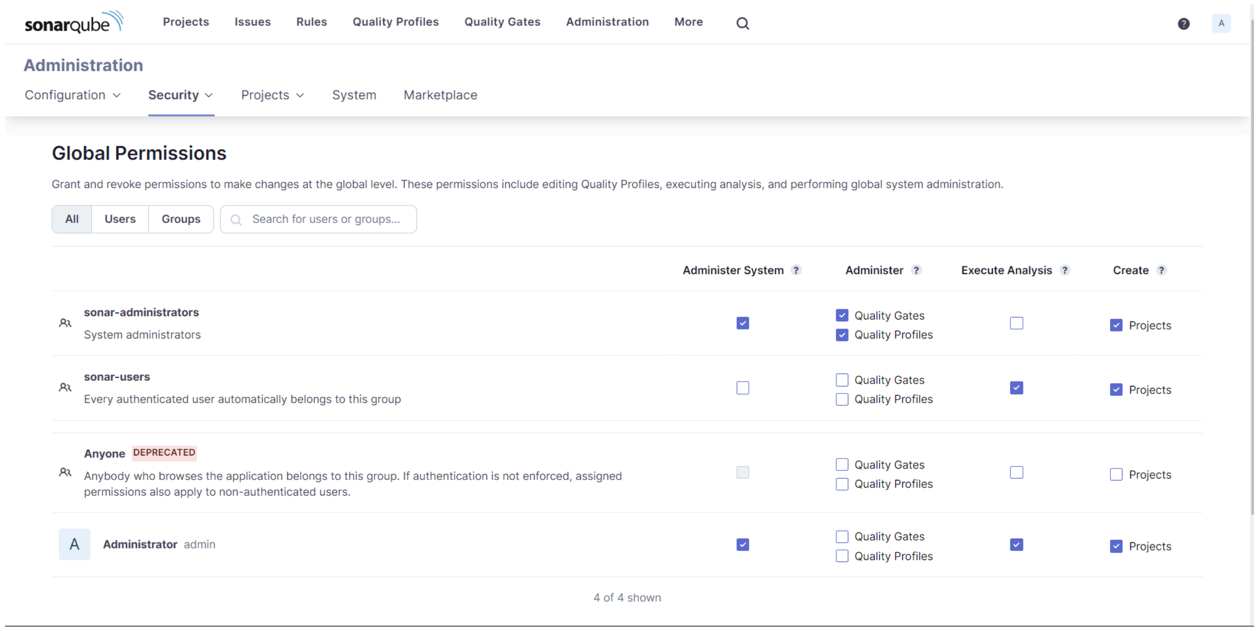
Path to project properties

Analysis properties

```
sonar.projectKey=sonarqube-test
sonar.login=admin
sonar.password=sanno4444
sonar.sources=.
sonar.host.url=http://localhost:9000
```

Additional arguments

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



Administration

Global Permissions

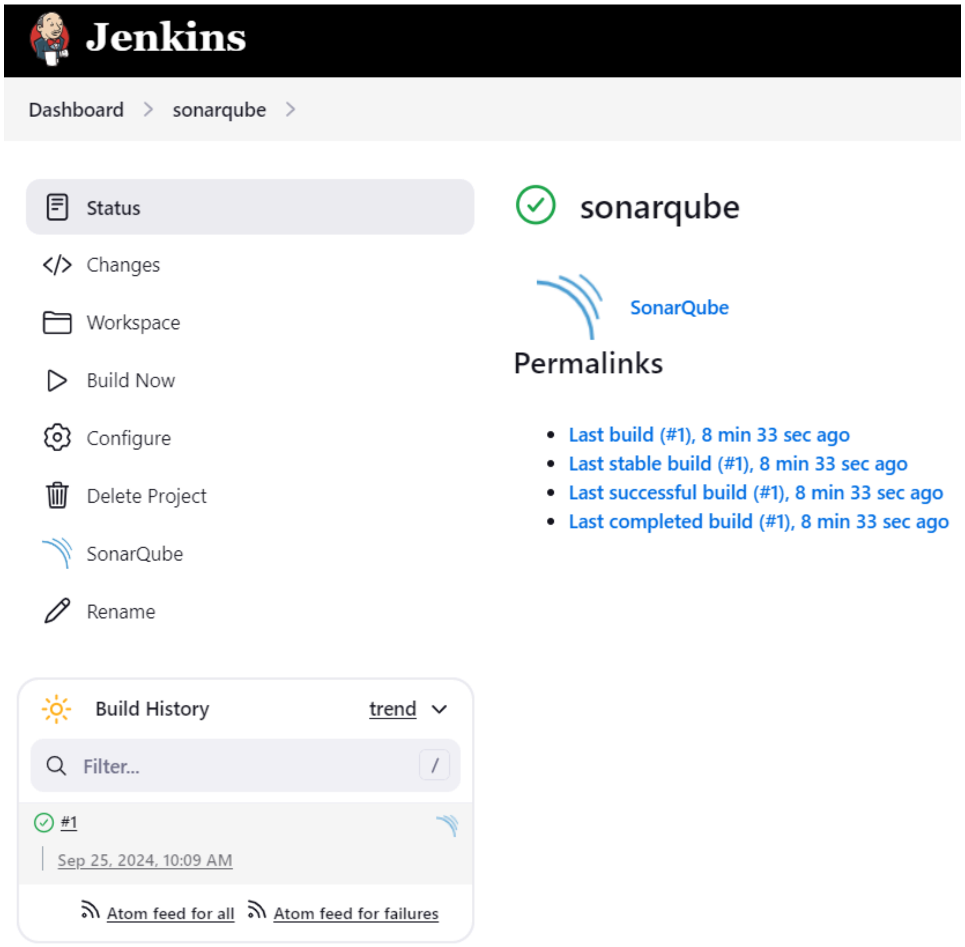
Grant and revoke permissions to make changes at the global level. These permissions include editing Quality Profiles, executing analysis, and performing global system administration.

	Administer System	Administer	Execute Analysis	Create
sonar-administrators System administrators	✓	✓ Quality Gates ✓ Quality Profiles	☐	✓ Projects
sonar-users Every authenticated user automatically belongs to this group	☐	☐ Quality Gates ☐ Quality Profiles	✓	✓ 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.	☐	☐ Quality Gates ☐ Quality Profiles	☐	☐ Projects
Administrator admin	✓	☐ Quality Gates ☐ Quality Profiles	✓	✓ Projects

4 of 4 shown

12. Run The Build.

Name: Shejal Tiwari Div/Roll No:D15C/57



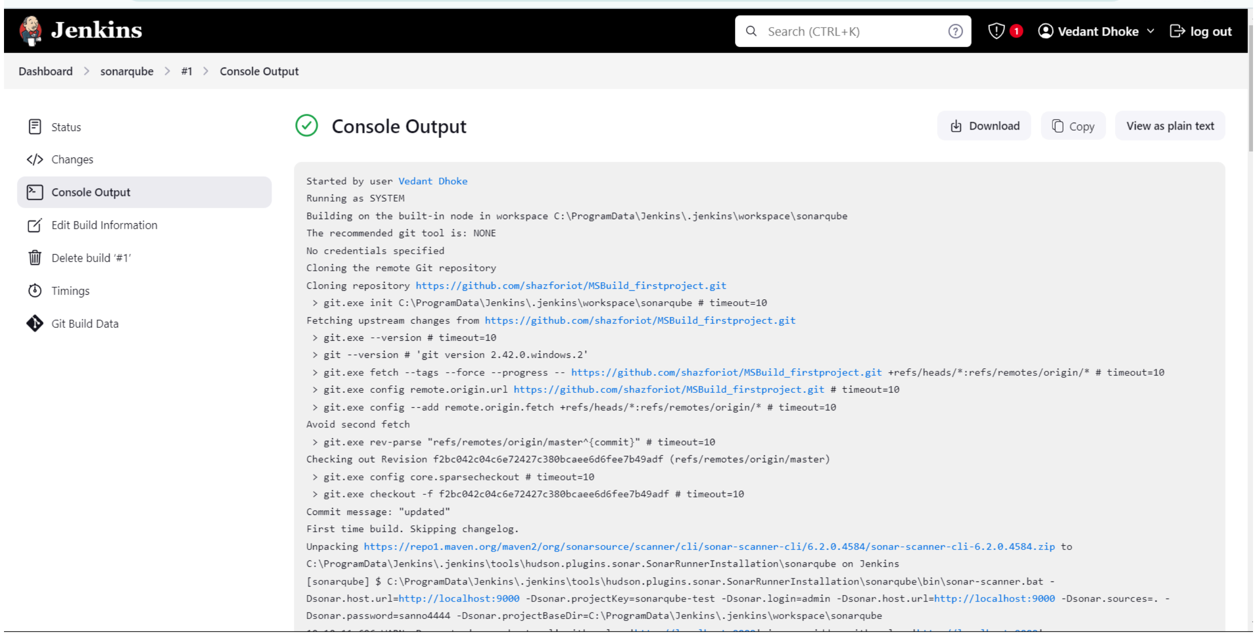
sonarqube

SonarQube

Permalinks

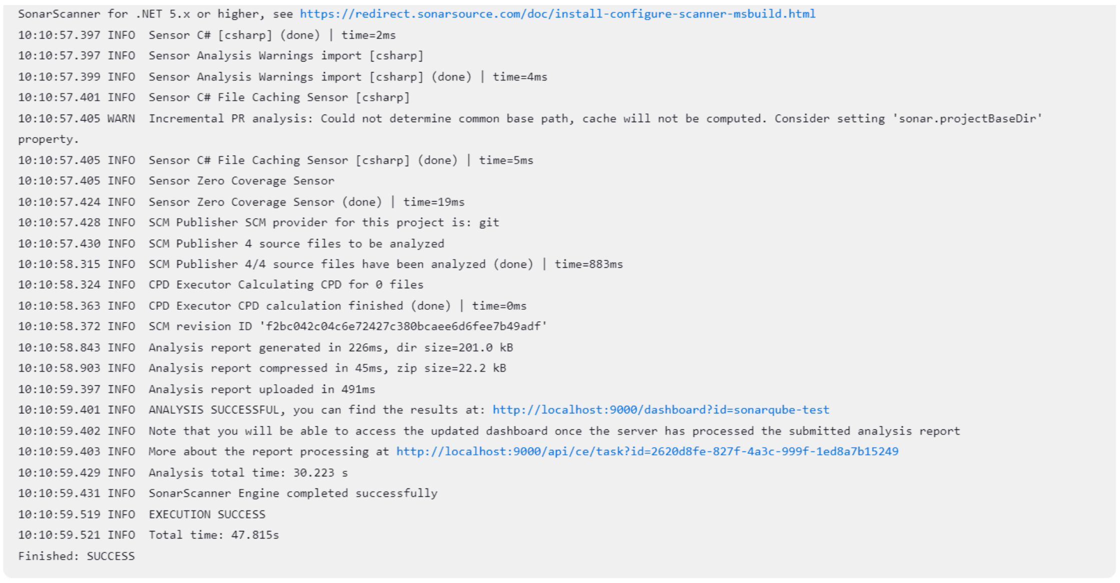
- Last build (#1), 8 min 33 sec ago
- Last stable build (#1), 8 min 33 sec ago
- Last successful build (#1), 8 min 33 sec ago
- Last completed build (#1), 8 min 33 sec ago

Check the console output.



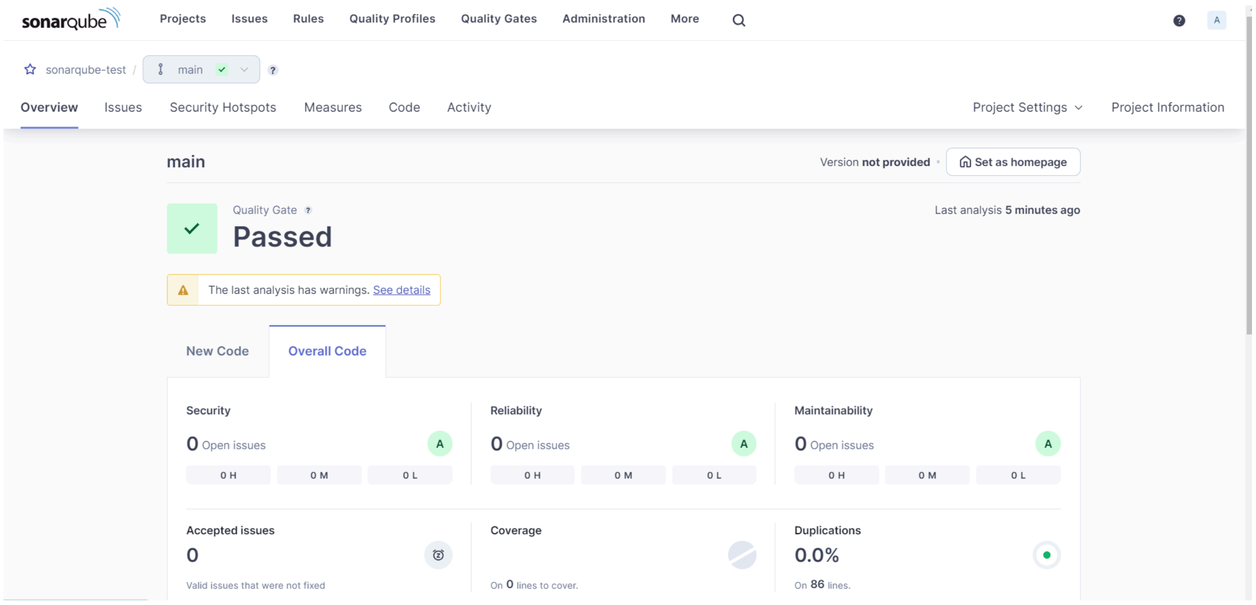
Console Output

```
Started by user Vedant Dhoke
Running as SYSTEM
Building on the built-in node 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 2.42.0.windows.2'
 > 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 f2bc042c04c6e72427c380bcaee6d6fee7b49adf (refs/remotes/origin/master)
 > git.exe config core.sparsecheckout # timeout=10
 > git.exe 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
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 -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=sonarqube-test -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -
Dsonar.password=sanno4444 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\sonarqube
```



```
SonarScanner for .NET 5.x or higher, see https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
10:10:57.397 INFO  Sensor C# [csharp] (done) | time=2ms
10:10:57.397 INFO  Sensor Analysis Warnings import [csharp]
10:10:57.399 INFO  Sensor Analysis Warnings import [csharp] (done) | time=4ms
10:10:57.401 INFO  Sensor C# File Caching Sensor [csharp]
10:10:57.405 WARN  Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir'
property.
10:10:57.405 INFO  Sensor C# File Caching Sensor [csharp] (done) | time=5ms
10:10:57.405 INFO  Sensor Zero Coverage Sensor
10:10:57.424 INFO  Sensor Zero Coverage Sensor (done) | time=19ms
10:10:57.428 INFO  SCM Publisher SCM provider for this project is: git
10:10:57.430 INFO  SCM Publisher 4 source files to be analyzed
10:10:58.315 INFO  SCM Publisher 4/4 source files have been analyzed (done) | time=883ms
10:10:58.324 INFO  CPD Executor Calculating CPD for 0 files
10:10:58.363 INFO  CPD Executor CPD calculation finished (done) | time=0ms
10:10:58.372 INFO  SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
10:10:58.843 INFO  Analysis report generated in 226ms, dir size=201.0 kB
10:10:58.903 INFO  Analysis report compressed in 45ms, zip size=22.2 kB
10:10:59.397 INFO  Analysis report uploaded in 491ms
10:10:59.401 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube-test
10:10:59.402 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
10:10:59.403 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=2620d8fe-827f-4a3c-999f-1ed8a7b15249
10:10:59.429 INFO  Analysis total time: 30.223 s
10:10:59.431 INFO  SonarScanner Engine completed successfully
10:10:59.519 INFO  EXECUTION SUCCESS
10:10:59.521 INFO  Total time: 47.815s
Finished: SUCCESS
```

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



In this way, we have integrated Jenkins with SonarQube for SAST.

Conclusion

1. Docker Container Issues: The SonarQube container might not start because your system doesn't have enough memory or processing power. SonarQube needs around 2GB of RAM to work properly, so if your system is low on resources, the container won't run.

2. Login Problems in SonarQube: You might have trouble logging in with the default username (admin) and password (admin). This could happen if there was a configuration issue with SonarQube or if the default password was changed during previous setups.

3. Jenkins Plugin Installation Errors: While installing the SonarQube Scanner plugin in Jenkins, you might encounter failures due to network issues or proxy settings, preventing the plugin from downloading correctly.

4. Incorrect SonarQube Configuration in Jenkins: While configuring SonarQube in Jenkins, entering the wrong project key, username, or password can cause the scan to fail. Ensuring accurate information is critical for a successful scan.