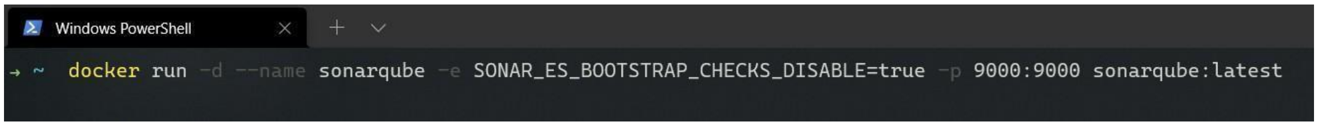
Name: Prathamesh Shettty
Div/Roll No: D15B/57
Exp 7

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

Steps to integrate Jenkins with SonarQube

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

Run SonarQube in a Docker container using this command -

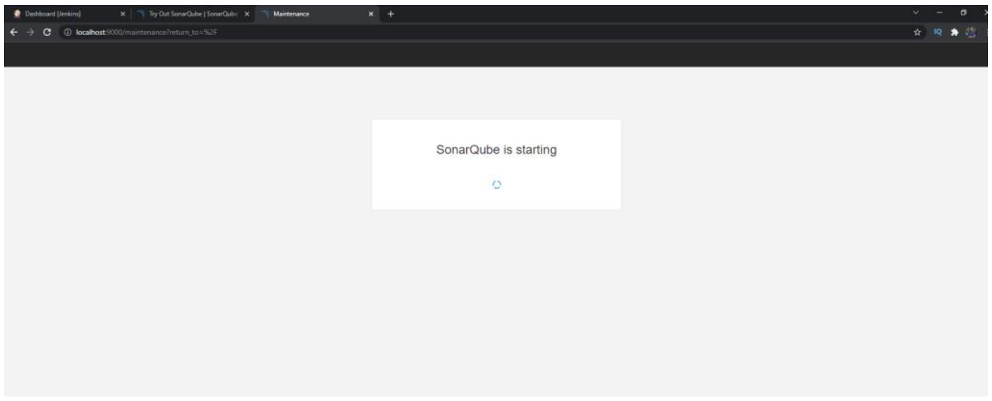


```
→ ~  docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

Warning: run below command only once

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

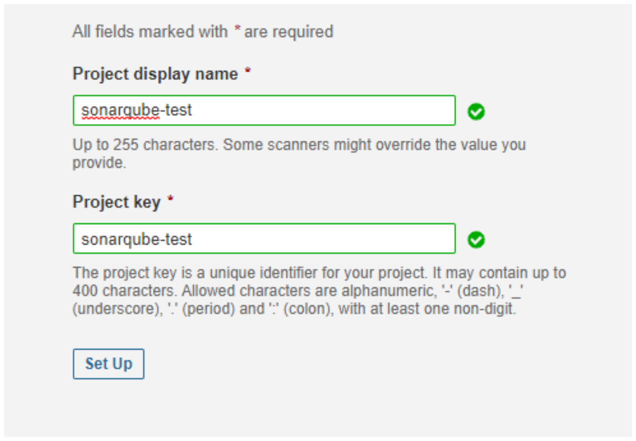
Once the container is up and running, you can check the status of SonarQube at localhost port 9000.


Dashboard [Jenkins]
Try Out SonarQube | SonarQube
Maintenance
localhost:9000/maintenance?return_to=%2F

SonarQube is starting

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

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



All fields marked with * are required

Project display name *

sonarqube-test

Up to 255 characters. Some scanners might override the value you provide.

Project key *

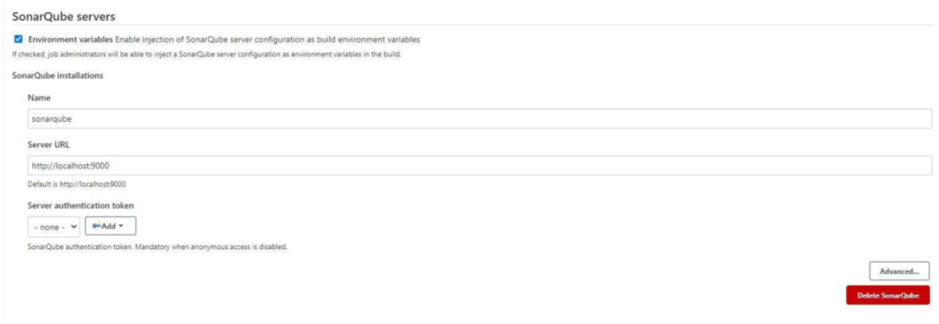
sonarqube-test

The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit.

Set Up

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

Enter the Server Authentication token if needed.



SonarQube servers

☑ Environment variables Enable injection of SonarQube server configuration as build environment variables

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

SonarQube installations

Name

sonarqube

Server URL

http://localhost:9000

Default is http://localhost:9000

Server authentication token

- none - Add

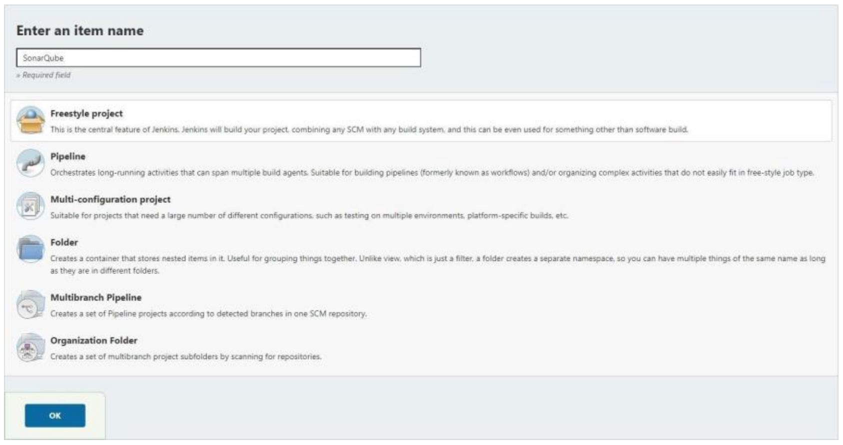
SonarQube authentication token. Mandatory when anonymous access is disabled.

Advanced...

Delete SonarQube

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

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



Enter an item name

SonarQube

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.

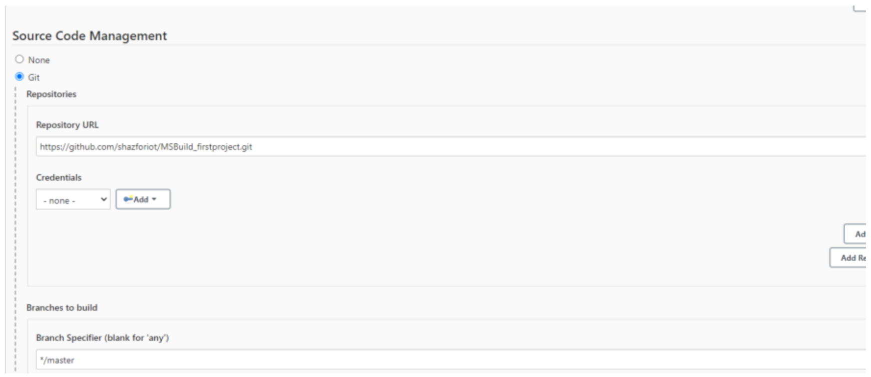
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Choose this GitHub repository in Source CodeManagement.



Source Code Management

( ) None
(●) Git

Repositories

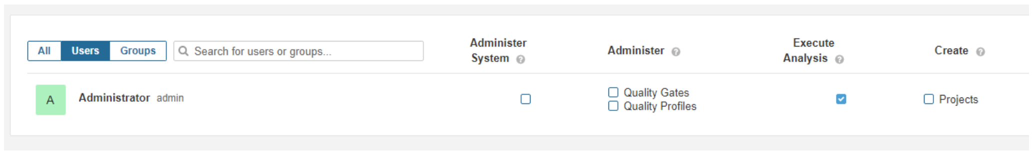
Repository URL: https://github.com/shazforiot/MSBuild_firstproject.git

Credentials: - none -

Branches to build

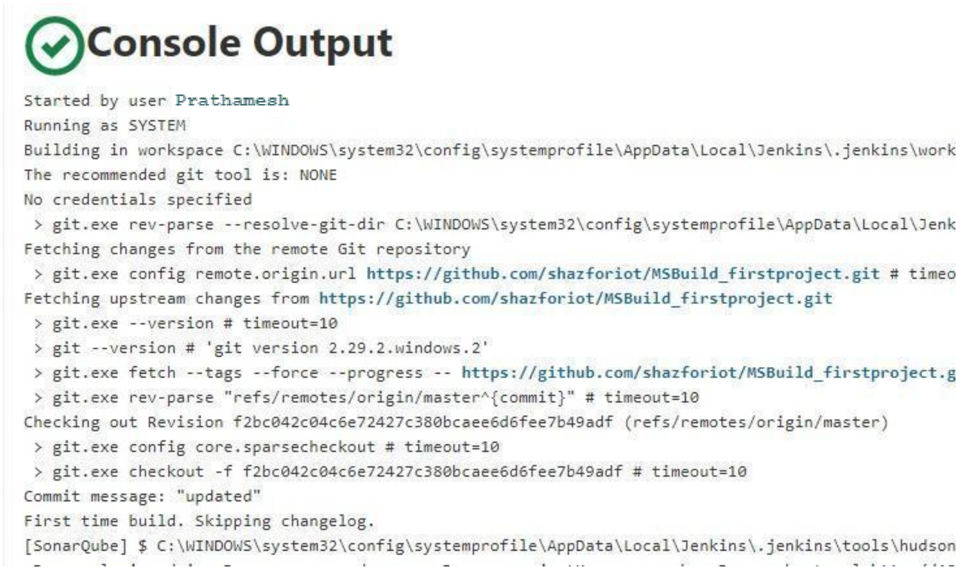
Branch Specifier (blank for 'any'): */master

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



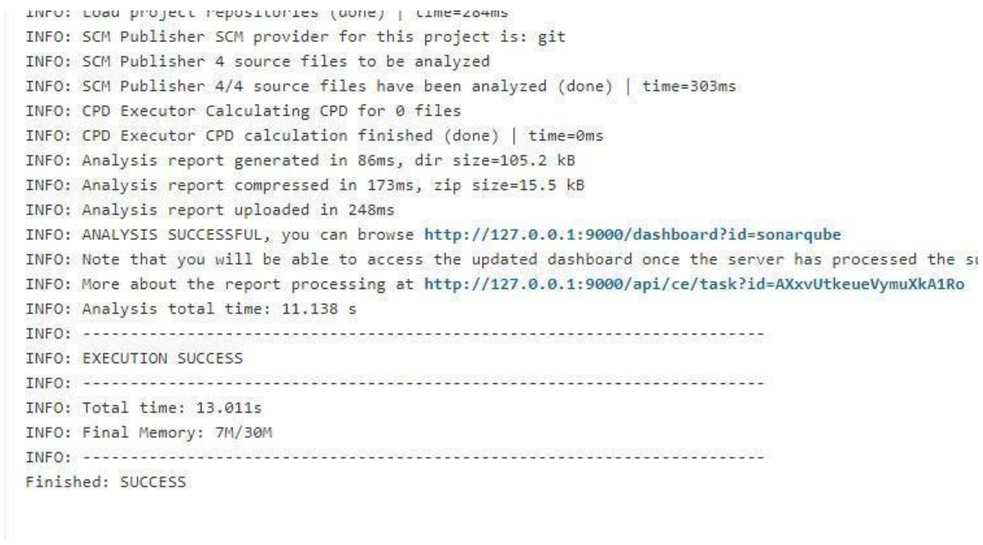
All / Users / Groups	Administer System	Administer	Execute Analysis	Create
A Administrator admin	☐	☐ Quality Gates ☐ Quality Profiles	✓	☐ Projects

Check the console output.



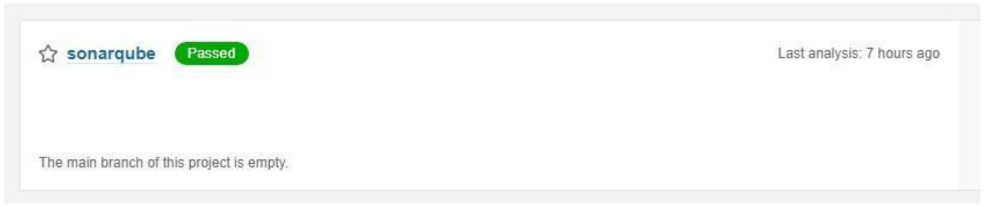
Console Output

```
Started by user Prathamesh
Running as SYSTEM
Building in workspace C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\work
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenk
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeo
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.29.2.windows.2'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.g
 > 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.
[SonarQube] $ C:\WINDOWS\system32\config\systemprofile\AppData\Local\Jenkins\.jenkins\tools\hudson
```



```
INFO: Load project repositories (done) | time=284ms
INFO: SCM Publisher SCM provider for this project is: git
INFO: SCM Publisher 4 source files to be analyzed
INFO: SCM Publisher 4/4 source files have been analyzed (done) | time=303ms
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: Analysis report generated in 86ms, dir size=105.2 kB
INFO: Analysis report compressed in 173ms, zip size=15.5 kB
INFO: Analysis report uploaded in 248ms
INFO: ANALYSIS SUCCESSFUL, you can browse http://127.0.0.1:9000/dashboard?id=sonarqube
INFO: Note that you will be able to access the updated dashboard once the server has processed the s
INFO: More about the report processing at http://127.0.0.1:9000/api/ce/task?id=AXxvUtkeueVymuXkA1Ro
INFO: Analysis total time: 11.138 s
INFO: ------------------------------------------------------------------------
INFO: EXECUTION SUCCESS
INFO: ------------------------------------------------------------------------
INFO: Total time: 13.011s
INFO: Final Memory: 7M/30M
INFO: ------------------------------------------------------------------------
Finished: SUCCESS
```

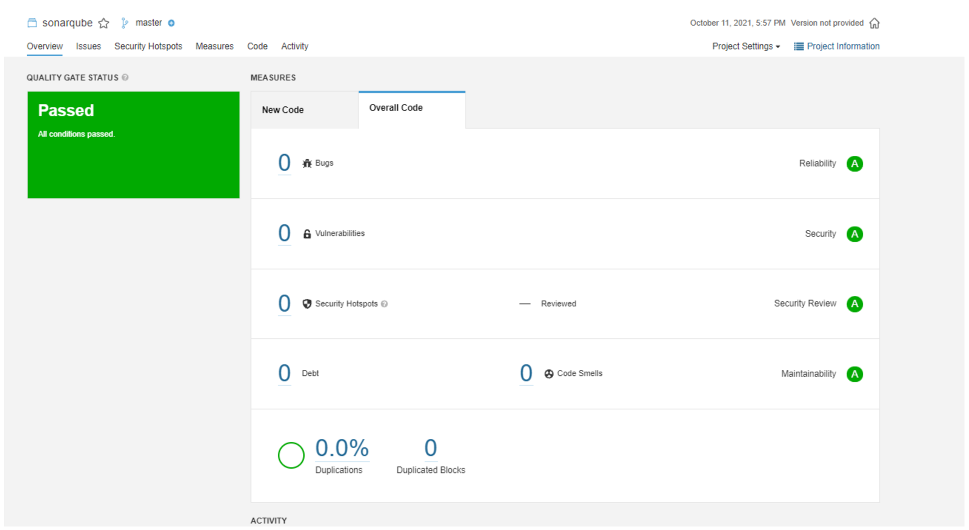
1. Once the build is complete, check the project in SonarQube.



sonarqube Passed

Last analysis: 7 hours ago

The main branch of this project is empty.



sonarqube master

October 11, 2021, 5:57 PM Version not provided

Overview Issues Security Hotspots Measures Code Activity

Project Settings Project Information

QUALITY GATE STATUS

Passed

All conditions passed.

MEASURES

New Code | Overall Code

0 Bugs — Reliability A

0 Vulnerabilities — Security A

0 Security Hotspots — Reviewed — Security Review A

0 Debt — 0 Code Smells — Maintainability A

0.0% Duplications — 0 Duplicated Blocks

ACTIVITY

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.