

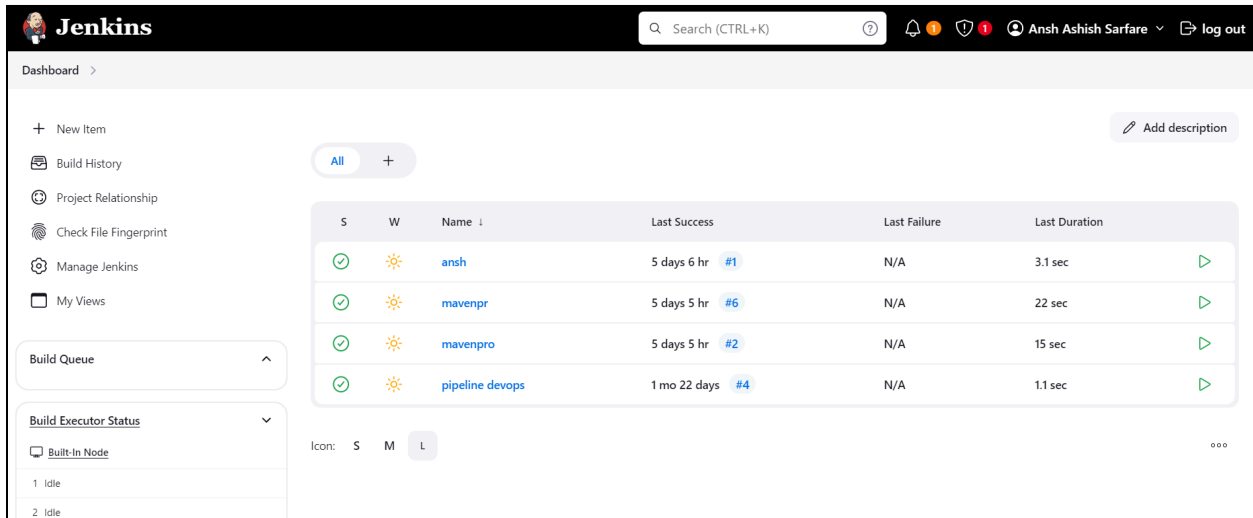
ADVANCE DEVOPS EXP-7

ANSH SARFARE

D15A/50

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

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



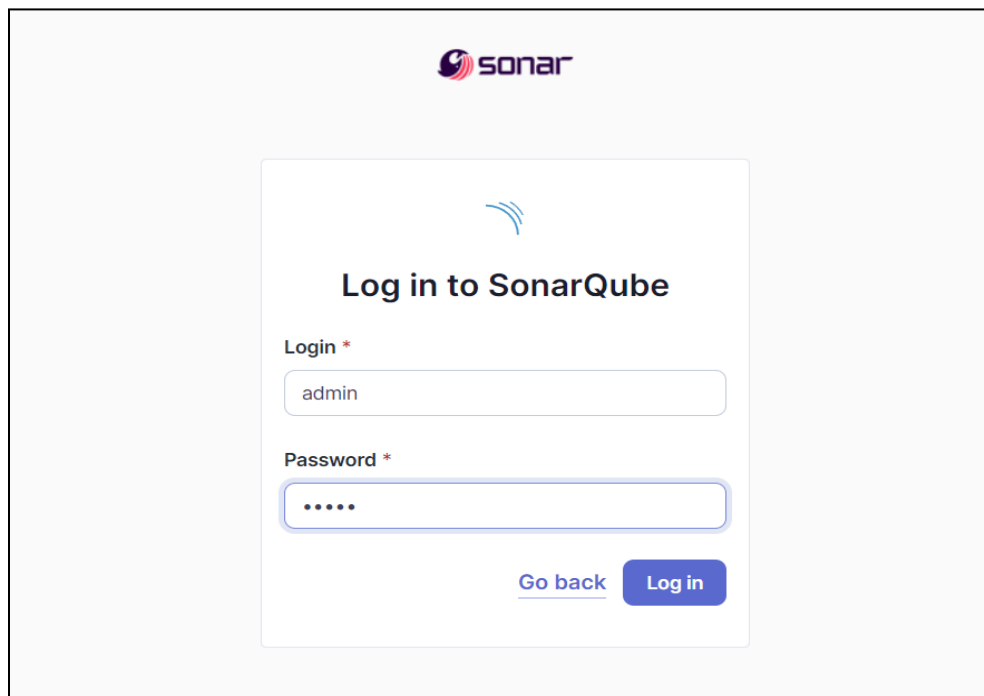
The screenshot shows the Jenkins Dashboard with a sidebar on the left containing links like 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of build history for the job 'ansh'. The table has columns for status (S), warnings (W), name, last success, last failure, and last duration. Below the table, there are sections for 'Build Queue' and 'Build Executor Status'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	ansh	5 days 6 hr #1	N/A	3.1 sec
✓	☀	mavenpr	5 days 5 hr #6	N/A	22 sec
✓	☀	mavenpro	5 days 5 hr #2	N/A	15 sec
✓	☀	pipeline devops	1 mo 22 days #4	N/A	1.1 sec

Step-2: Run SonarQube in a Docker container using this command :- a] docker -v
b] docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

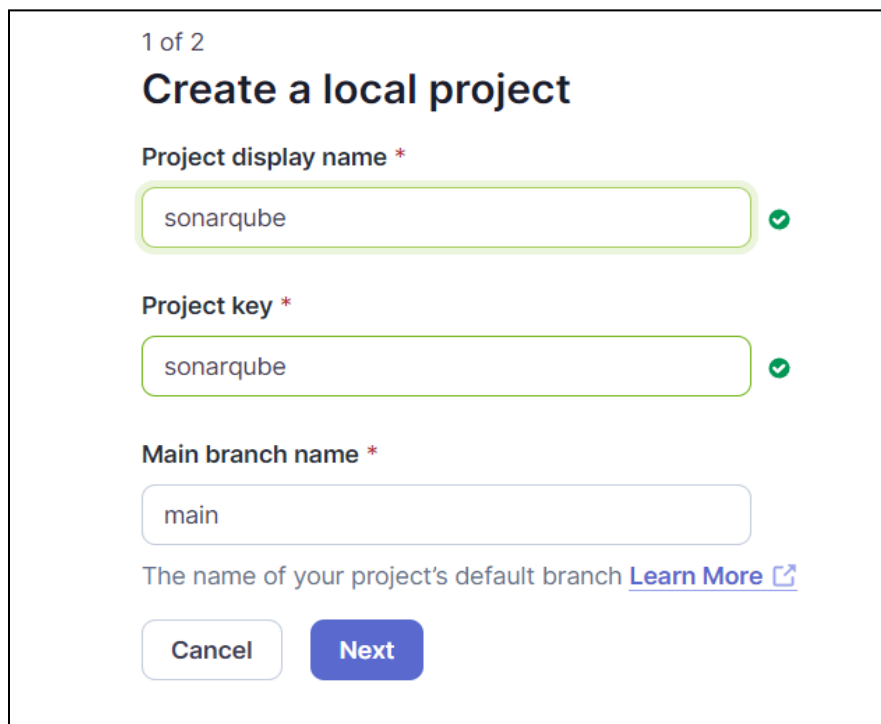
```
PS C:\Users\Ansh> docker -v
Docker version 27.0.3, build 7d4bcd8
PS C:\Users\Ansh> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
latest: Pulling from library/sonarqube
7478e0ac0f23: Pull complete
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
e4a886abe0db4f8a8c19e8f125ce97244d50eea97f9e98f2d829b724bc95c973
PS C:\Users\Ansh> |
```

Step-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 also “admin”.



The image shows the SonarQube login interface. At the top is the Sonar logo. Below it is a white box with the title "Log in to SonarQube". Inside this box, there are two input fields: "Login *" with the value "admin" and "Password *" with masked characters ".....". Below the password field are two buttons: "Go back" (a link) and "Log in" (a blue button).

Step-4: Create a local project in SonarQube with the name sonarqube



The image shows the "Create a local project" page in SonarQube. It is labeled "1 of 2" at the top. The title is "Create a local project". There are three input fields: "Project display name *" with the value "sonarqube" and a green checkmark, "Project key *" with the value "sonarqube" and a green checkmark, and "Main branch name *" with the value "main". Below the last field is a link "Learn More" with an external link icon. At the bottom are two buttons: "Cancel" and "Next".

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.

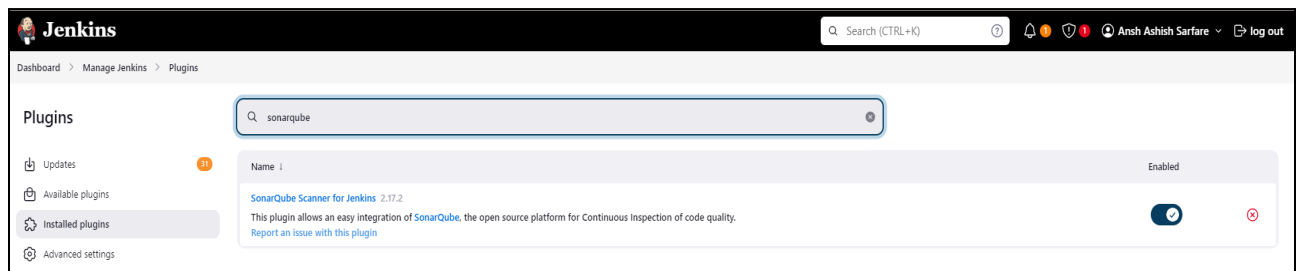
☐ Reference branch

Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

BackCreate project

Step-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.



Step-6: Under 'Manage Jenkins → System', look for SonarQube Servers and enter these details. Name : sonarqube, Server URL : http://localhost:9000

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

Default is http://localhost:9000

http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add

Advanced

Step-7: Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically. Manage Jeknins → Tools → SonarQube Scanner Installation.

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

Name

sonarqube

☒ Install automatically ?

☰ Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

Add Installer


Add SonarQube Scanner


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


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.

**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

Step-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.

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 ?

Save

Apply

Step-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=ansh

sonar.sources=.

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

Configure

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

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
sonar.login=admin
sonar.password=ansh
sonar.sources=.
sonar.host.url=http://localhost:9000

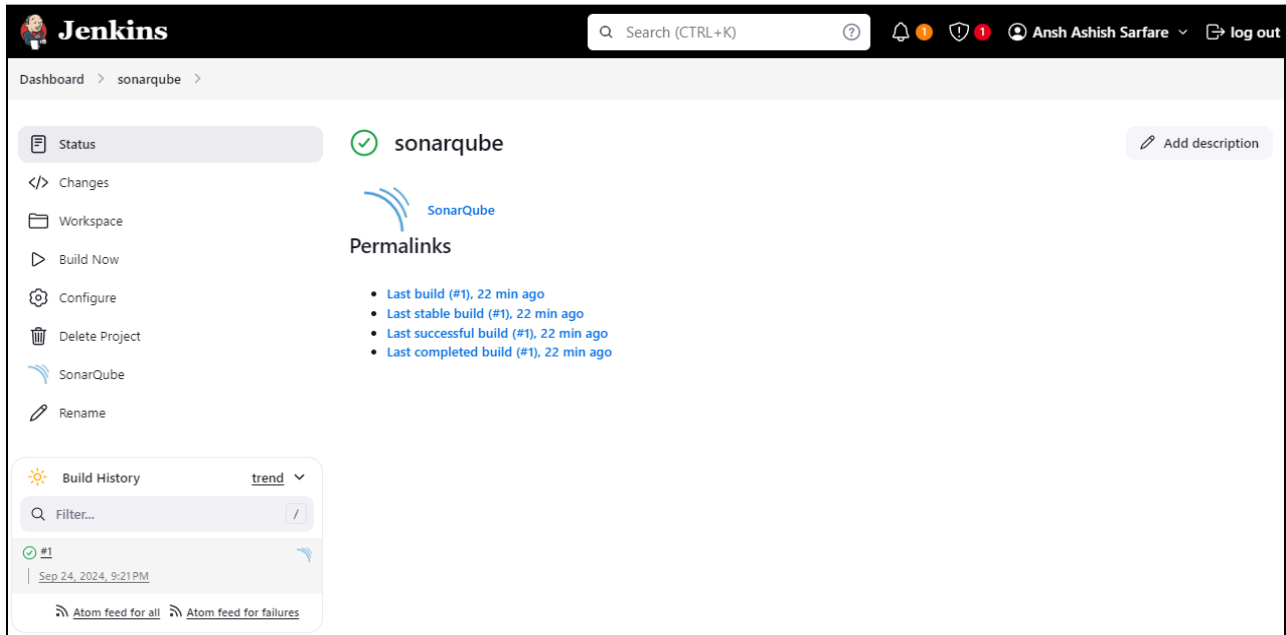
Additional arguments ?

JVM Options ?

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

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.					
<div> <div>All</div> <div>Users</div> <div>Groups</div> </div> <div> <div>Q</div> <div>Search for users or groups...</div> </div>					
		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
	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
4 of 4 shown					

Step-12: Run The Build and check the console output.



The Jenkins Dashboard for the 'sonarqube' job. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'Ansh Ashish Sarfare'. The left sidebar contains a list of actions: Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main content area shows the job status as 'Success' with a green checkmark. Below the status, there are 'Permalinks' for the last build, last stable build, last successful build, and last completed build, all dated '22 min ago'. A 'Build History' section on the left shows a single build '#1' from 'Sep 24, 2024, 9:21 PM'.

Jenkins Search (CTRL+K) ? Ansh Ashish Sarfare log out

Dashboard > sonarqube >

Status

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

Build History trend

Filter...

#1

Sep 24, 2024, 9:21 PM

Atom feed for all Atom feed for failures

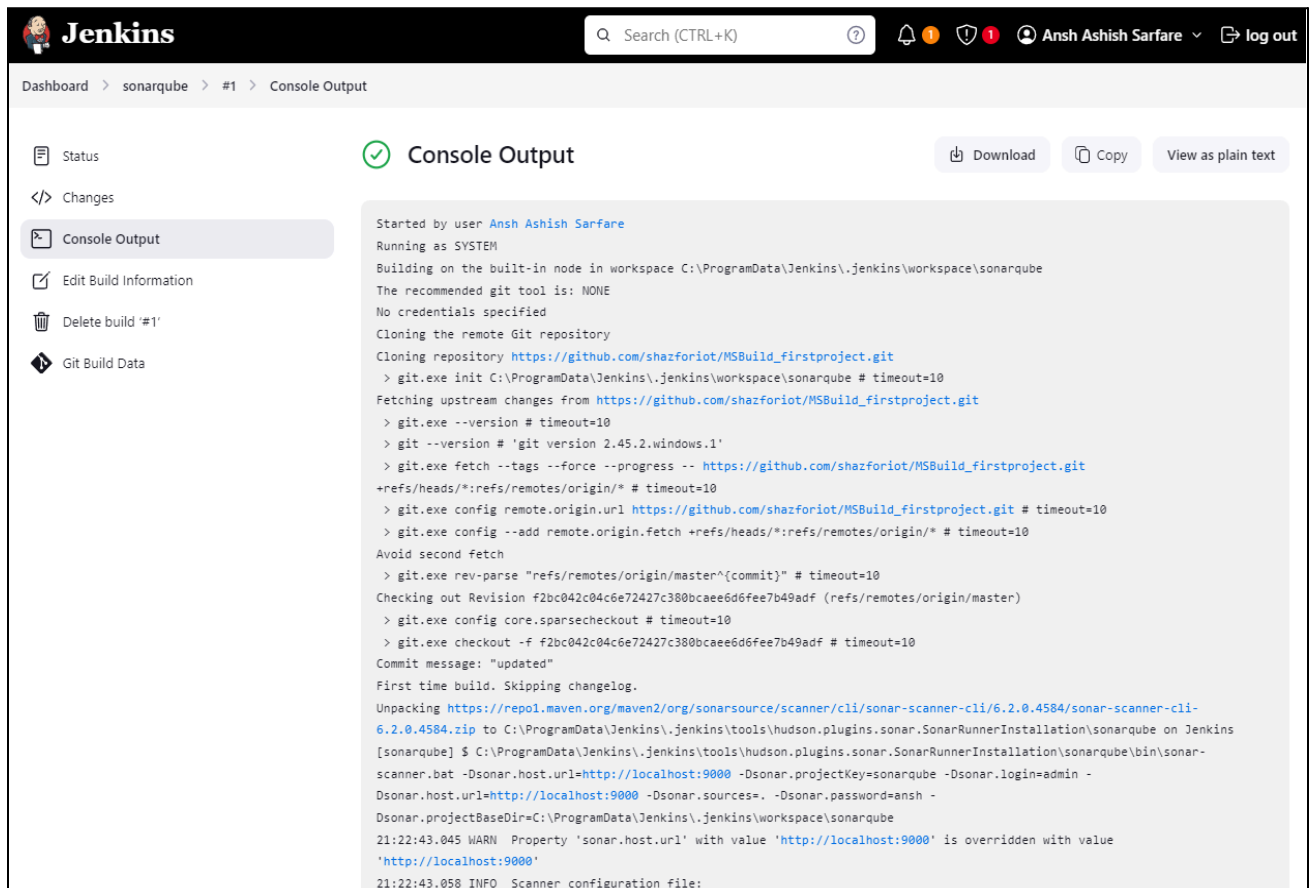
sonarqube

SonarQube

Permalinks

- Last build (#1), 22 min ago
- Last stable build (#1), 22 min ago
- Last successful build (#1), 22 min ago
- Last completed build (#1), 22 min ago

Add description



The Jenkins Console Output for the 'sonarqube' job, build '#1'. The top navigation bar is the same as the dashboard. The left sidebar shows the 'Console Output' tab selected. The main content area displays the build log, which starts with 'Started by user Ansh Ashish Sarfare' and 'Running as SYSTEM'. The log shows the process of cloning a repository from GitHub, checking out the code, and running the SonarQube scanner. The scanner configuration is shown at the bottom, including the host URL, project key, login, and password.

Jenkins Search (CTRL+K) ? Ansh Ashish Sarfare log out

Dashboard > sonarqube > #1 > Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#1'

Git Build Data

Console Output

Download Copy View as plain text

Started by user Ansh Ashish Sarfare

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 2.45.2.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 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](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 -Dsonar.login=admin -Dsonar.host.url=<http://localhost:9000> -Dsonar.sources=. -Dsonar.password=ansh -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\sonarqube

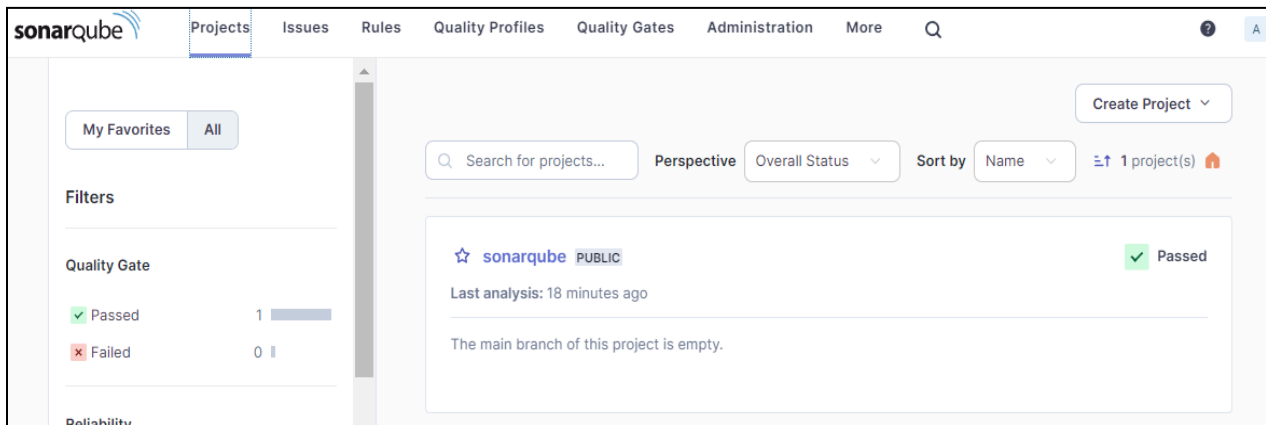
21:22:43.045 WARN Property 'sonar.host.url' with value '<http://localhost:9000>' is overridden with value '<http://localhost:9000>'

21:22:43.058 INFO Scanner configuration file:


```
Dashboard > sonarqube > #1 > Console Output

21:23:05.658 INFO 14/14 source files have been analyzed
21:23:05.658 INFO Sensor TextAndSecretsSensor [text] (done) | time=903ms
21:23:05.662 INFO ----- Run sensors on project
21:23:05.748 INFO Sensor C# [csharp]
21:23:05.749 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
21:23:05.749 INFO Sensor C# [csharp] (done) | time=1ms
21:23:05.749 INFO Sensor Analysis Warnings import [csharp]
21:23:05.751 INFO Sensor Analysis Warnings import [csharp] (done) | time=2ms
21:23:05.752 INFO Sensor C# File Caching Sensor [csharp]
21:23:05.752 WARN Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting
'sonar.projectBaseDir' property.
21:23:05.752 INFO Sensor C# File Caching Sensor [csharp] (done) | time=1ms
21:23:05.752 INFO Sensor Zero Coverage Sensor
21:23:05.762 INFO Sensor Zero Coverage Sensor (done) | time=9ms
21:23:05.765 INFO SCM Publisher SCM provider for this project is: git
21:23:05.767 INFO SCM Publisher 4 source files to be analyzed
21:23:06.128 INFO SCM Publisher 4/4 source files have been analyzed (done) | time=360ms
21:23:06.130 INFO CPD Executor Calculating CPD for 0 files
21:23:06.131 INFO CPD Executor CPD calculation finished (done) | time=0ms
21:23:06.134 INFO SCM revision ID 'f2bc042c04c6e72427c380bcabee6d6fee7b49adf'
21:23:06.355 INFO Analysis report generated in 92ms, dir size=201.0 kB
21:23:06.392 INFO Analysis report compressed in 28ms, zip size=22.6 kB
21:23:06.533 INFO Analysis report uploaded in 139ms
21:23:06.534 INFO ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube
21:23:06.534 INFO Note that you will be able to access the updated dashboard once the server has processed the submitted
analysis report
21:23:06.535 INFO More about the report processing at http://localhost:9000/api/ce/task?id=e4638b3c-d13e-449c-991b-639bd23c8ef1
21:23:06.543 INFO Analysis total time: 16.829 s
21:23:06.543 INFO SonarScanner Engine completed successfully
21:23:06.578 INFO EXECUTION SUCCESS
21:23:06.579 INFO Total time: 23.525s
Finished: SUCCESS
```

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



sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

sonarqube / main

main

OverviewIssuesSecurity HotspotsMeasuresCodeActivity

Project SettingsProject Information

main

Version **not provided**

Set as homepage

Quality Gate

Passed

Last analysis 17 minutes ago

The last analysis has warnings. [See details](#)

New CodeOverall Code

Security

0 Open issues

0 H0 M0 L

A

Reliability

0 Open issues

0 H0 M0 L

A

Maintainability

0 Open issues

0 H0 M0 L

A

Accepted issues

0

Valid issues that were not fixed

Coverage

On 0 lines to cover.

Duplications

0.0%

On 86 lines.

Security Hotspots

0

A