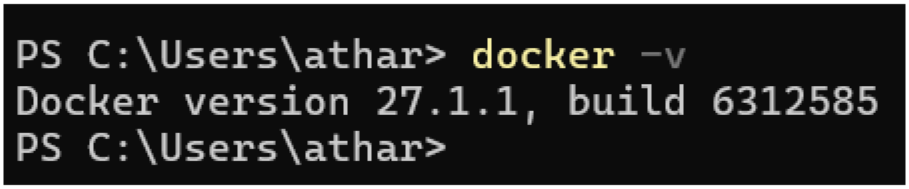
Name:Atharv Nikam **Div:D15C** **Roll no: 36**

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

Prerequisites:

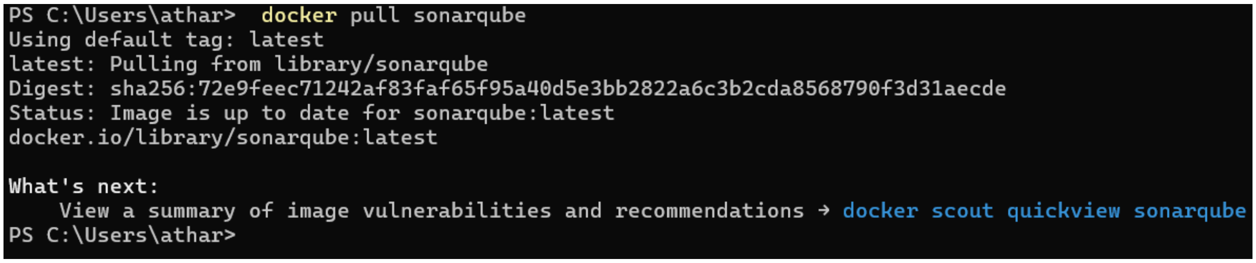
1. **Docker:**
   Confirm Docker is installed and functioning by running the command:



```
PS C:\Users\athar> docker -v
Docker version 27.1.1, build 6312585
PS C:\Users\athar>
```

2 . SonarQube Image Installation:

Use the following command to download the SonarQube image via Docker:



```
PS C:\Users\athar>  docker pull sonarqube
Using default tag: latest
latest: Pulling from library/sonarqube
Digest: sha256:72e9feec71242af83faf65f95a40d5e3bb2822a6c3b2cda8568790f3d31aecde
Status: Image is up to date for sonarqube:latest
docker.io/library/sonarqube:latest

What's next:
    View a summary of image vulnerabilities and recommendations → docker scout quickview sonarqube
PS C:\Users\athar>
```

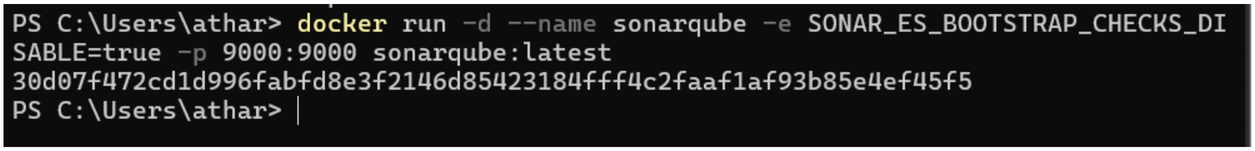
3.Jenkins Installation:

Ensure Jenkins is already installed and properly configured on your system

Name:Atharv Nikam Div:D15C Roll:36

Experiment Steps:

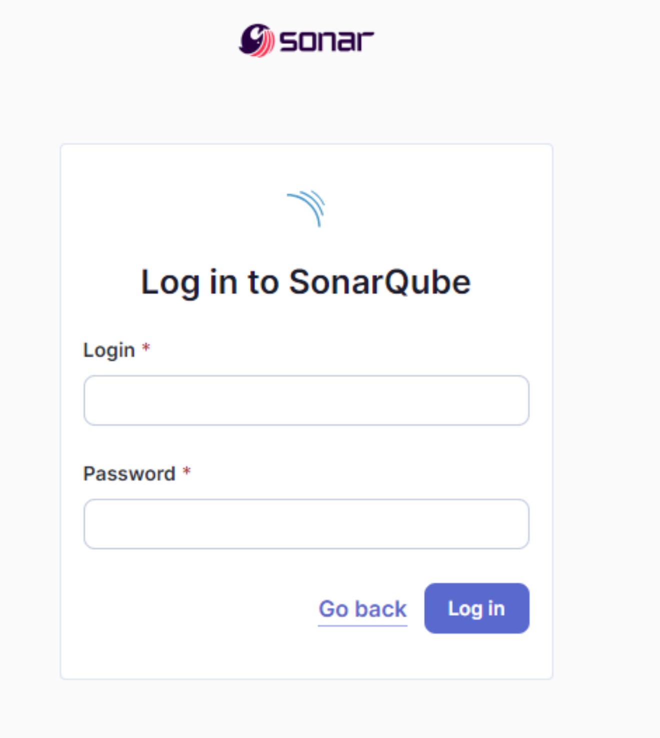
Step 1:
Run the SonarQube Docker container using the command:



```
PS C:\Users\athar> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DI
SABLE=true -p 9000:9000 sonarqube:latest
30d07f472cd1d996fabfd8e3f2146d85423184fff4c2faaf1af93b85e4ef45f5
PS C:\Users\athar> |
```

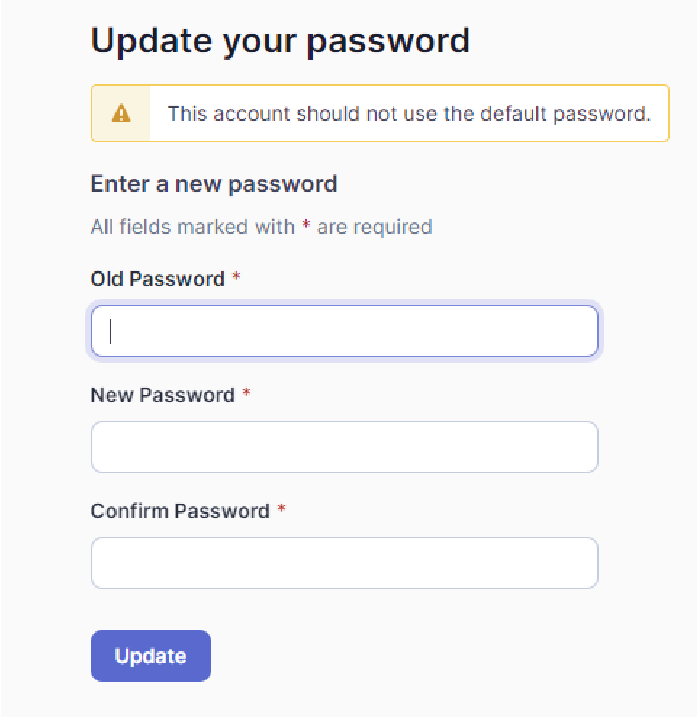
Step 2:

After SonarQube starts, navigate to `http://localhost:9000` in your browser.



Step 3:

Log in using the default credentials (`admin`/`admin`). After logging in, you will be prompted to change the password. Ensure to note the new password.



Update your password

This account should not use the default password.

Enter a new password

All fields marked with * are required

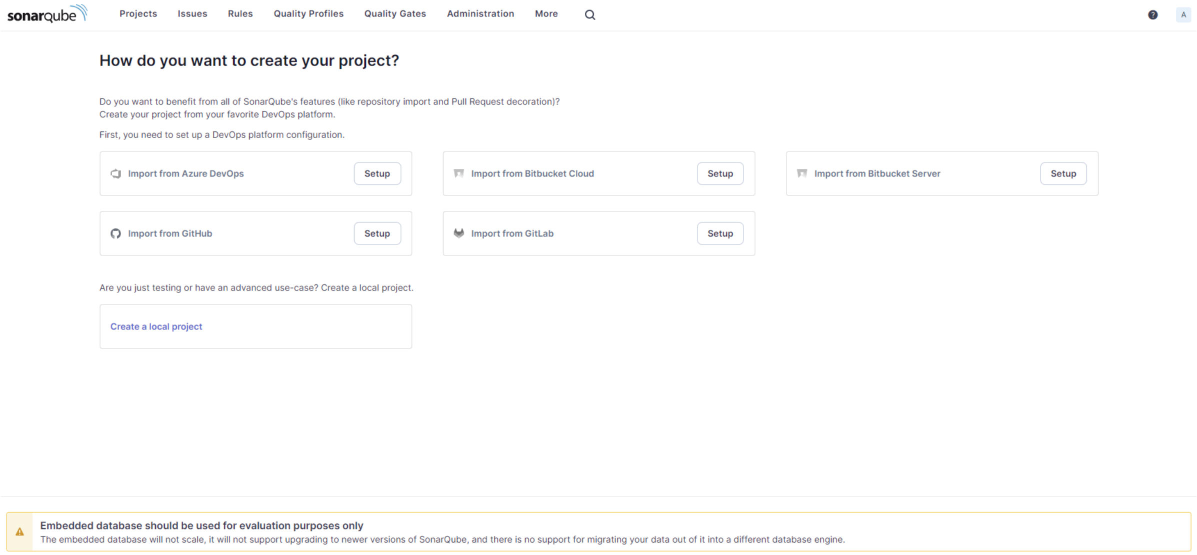
Old Password *

New Password *

Confirm Password *

Step 4:

Once logged in, create a new project by clicking on "Create a Local Project." Provide a project name and a project key, then proceed.



How do you want to create your project?

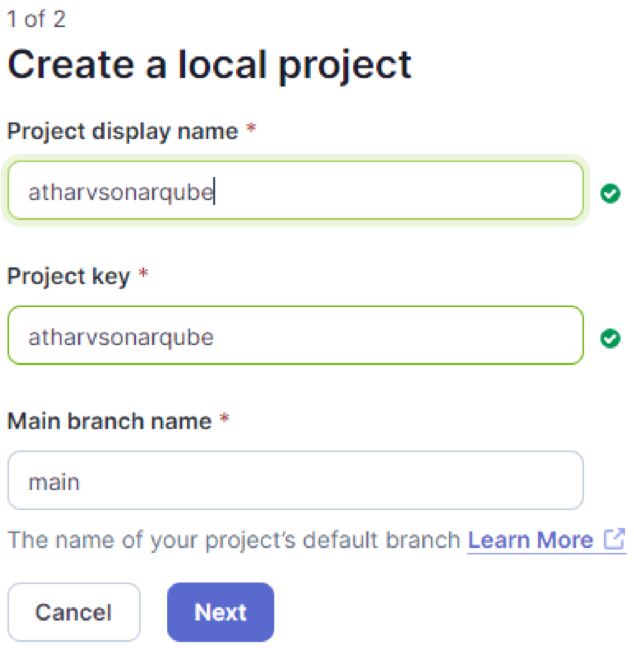
Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)?
Create your project from your favorite DevOps platform.

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

Are you just testing or have an advanced use-case? 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.

Name:Atharv Nikam Div:D15C Roll:36



1 of 2

Create a local project

Project display name *

atharvsonarqube ✓

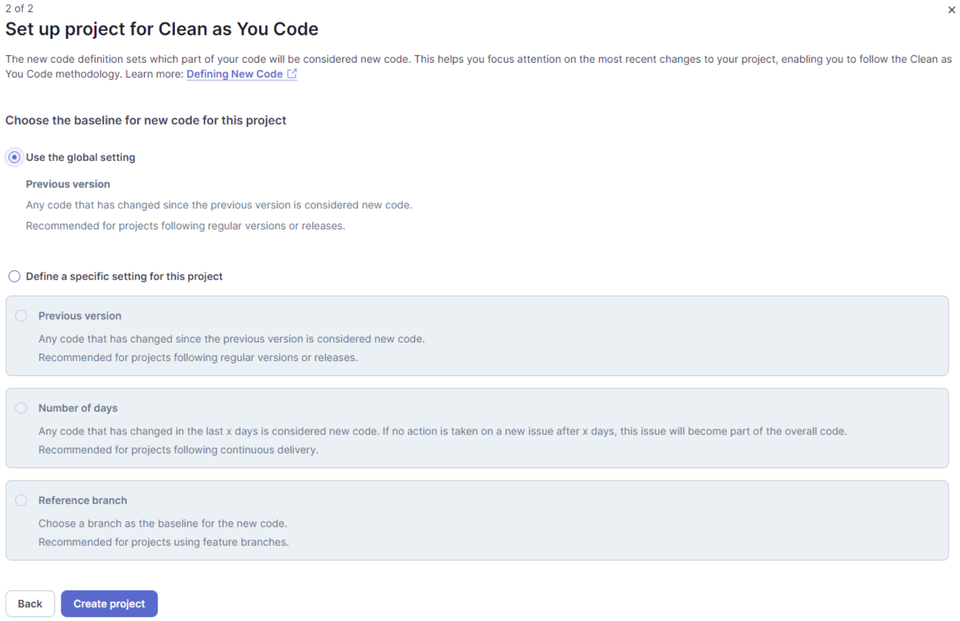
Project key *

atharvsonarqube ✓

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 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

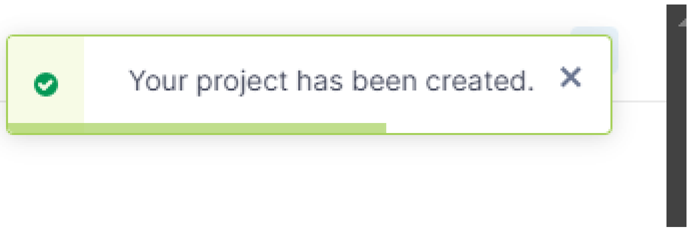
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.

Recommended for projects following continuous delivery.

○ Reference branch

Choose a branch as the baseline for the new code.

Recommended for projects using feature branches.

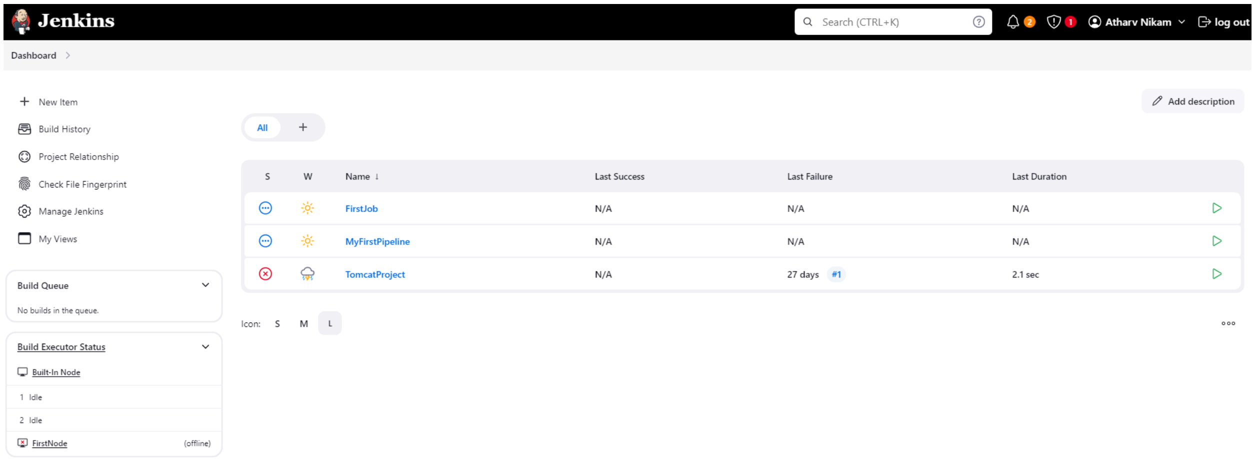


✓ Your project has been created. ✕

Name:Atharv Nikam Div:D15C Roll:36

Step 5:

Open Jenkins on its assigned port (`http://localhost:<port_number>`).



S	W	Name ↓	Last Success	Last Failure	Last Duration
		FirstJob	N/A	N/A	N/A
		MyFirstPipeline	N/A	N/A	N/A
		TomcatProject	N/A	27 days #1	2.1 sec

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

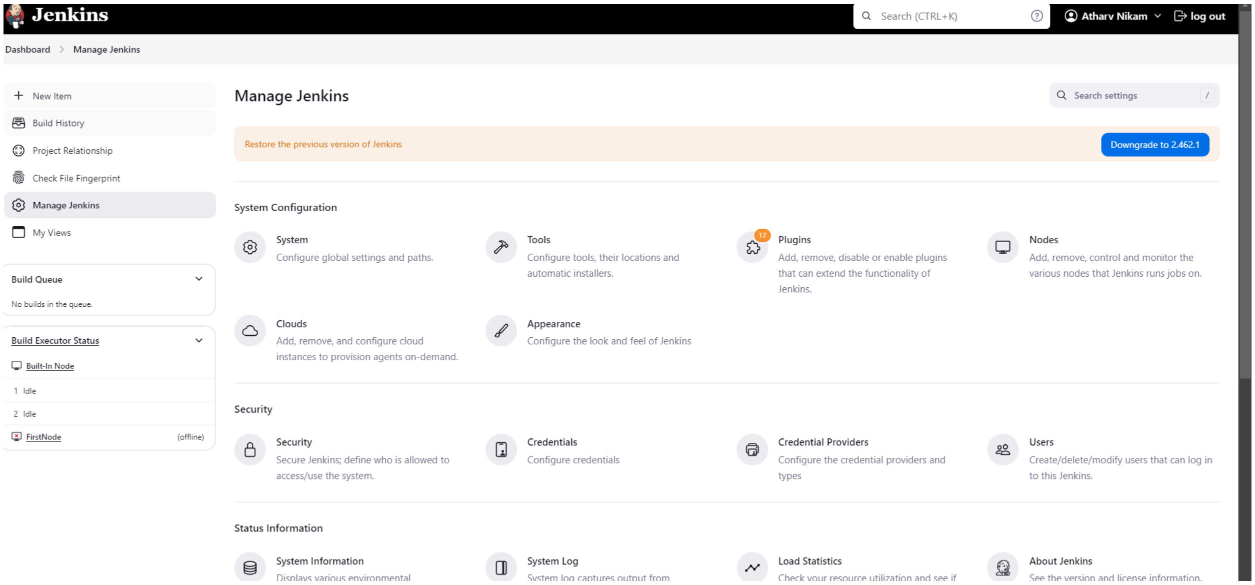
1 Idle

2 Idle

FirstNode (offline)

Step 6:

In Jenkins, go to **Manage Jenkins**, search for **SonarQube Scanner for Jenkins**, and install the plugin.



Manage Jenkins

Restore the previous version of Jenkins

System Configuration

System

Configure global settings and paths.

Tools

Configure tools, their locations and automatic installers.

Plugins

Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes

Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Clouds

Add, remove, and configure cloud instances to provision agents on-demand.

Appearance

Configure the look and feel of Jenkins

Security

Security

Secure Jenkins; define who is allowed to access/use the system.

Credentials

Configure credentials

Credential Providers

Configure the credential providers and types

Users

Create/delete/modify users that can log in to this Jenkins.

Status Information

System Information

Displays various environmental

System Log

System log captures output from

Load Statistics

Check your resource utilization and see if

About Jenkins

See the version and license information.

Build Queue

No builds in the queue.

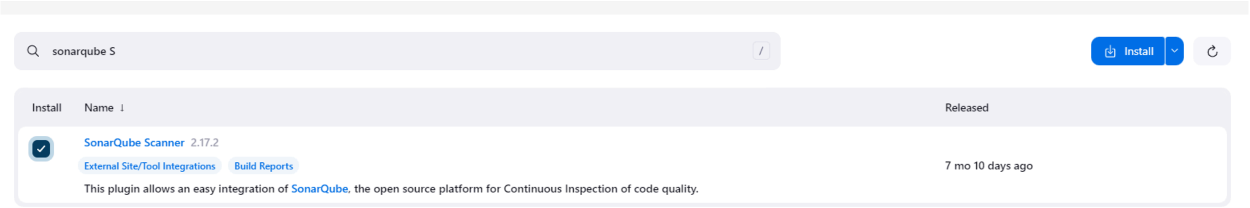
Build Executor Status

Built-In Node

1 Idle

2 Idle

FirstNode (offline)

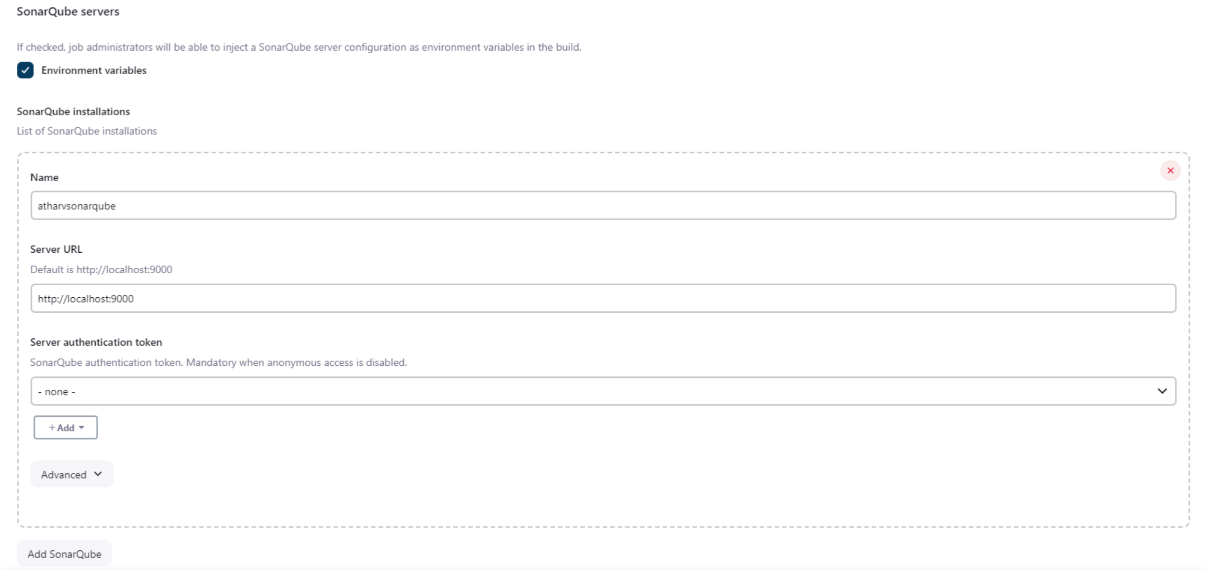


sonarqube S

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 10 days ago

Step 7:

After installation, navigate to **Manage Jenkins → System**. Under **SonarQube servers**, add your SonarQube server and configure it with the necessary authentication token.



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

atharvsonarqube

Server URL

Default is http://localhost:9000

http://localhost:9000

Server authentication token

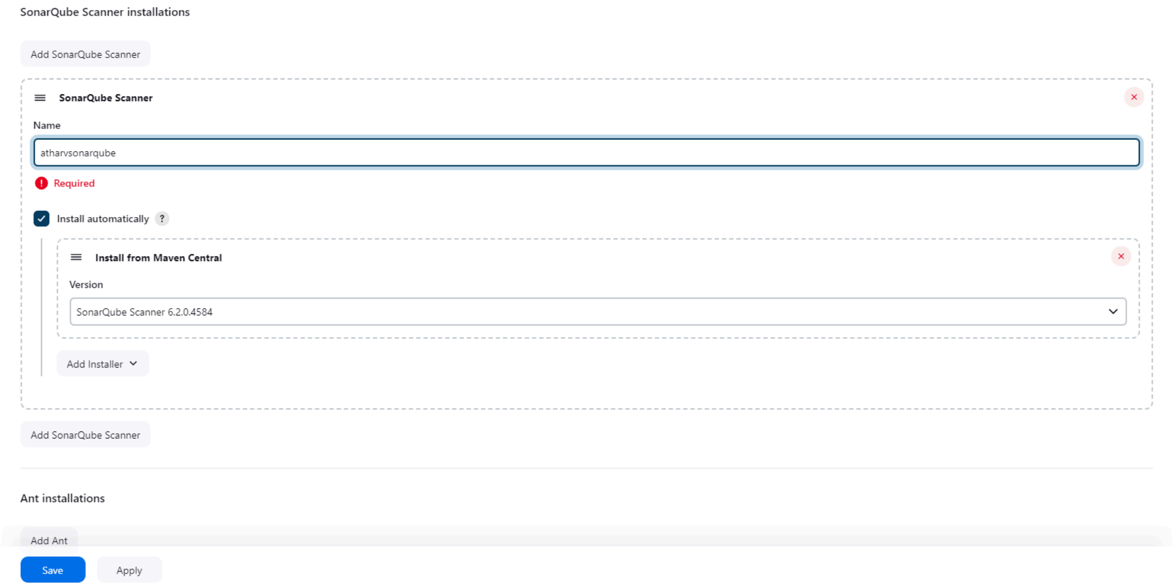
SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

Name:Atharv Nikam Div:D15C Roll:36

Step 8:

Next, go to **Manage Jenkins → Tools**. Under **SonarQube Scanner**, choose the latest configuration and enable automatic installation.



SonarQube Scanner installations

SonarQube Scanner

Name: atharvsonarqube

Required

☑ Install automatically

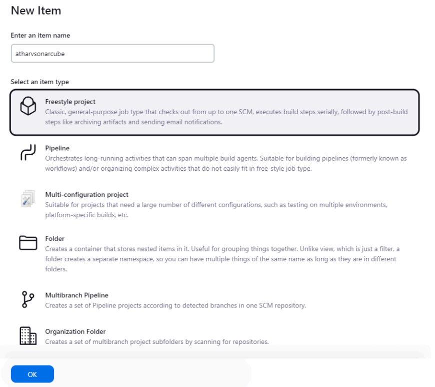
Install from Maven Central

Version: SonarQube Scanner 6.2.0.4584

Ant installations

Step 9:

Create a new item in Jenkins, selecting a **Freestyle Project**.



New Item

Enter an item name: atharvsonarcube

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.

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.

Step 10:
In the Source Code Management section, use this GitHub repository:

```
https://github.com/shazforiot/MSBuild_firstproject
```

This repository contains a sample project to test.

Step 11:

In the Build section, add the SonarQube Scanner. Enter the required SonarQube project details such as project key, login credentials, source path, and the server URL.

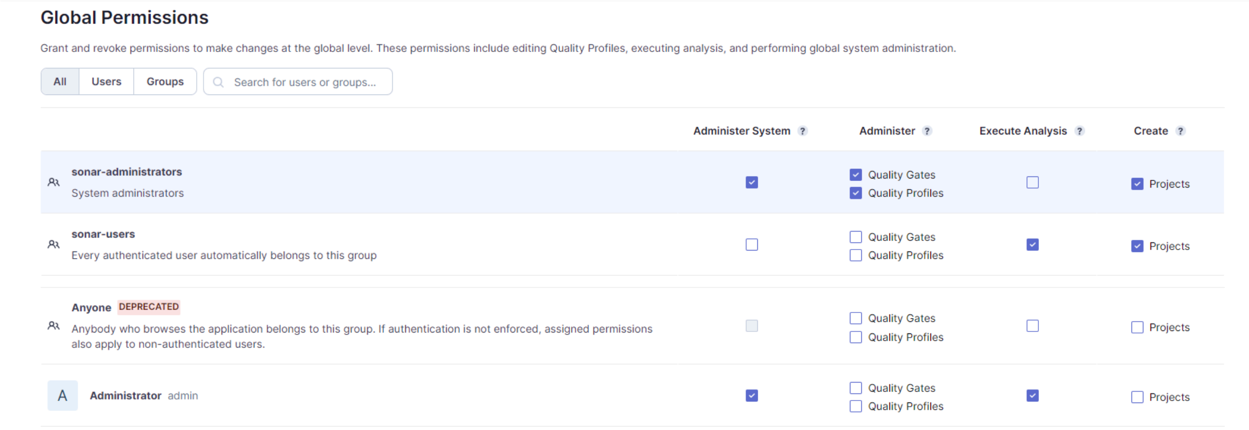
```
sonar.projectKey=atharvsonarqube

sonar.login=admin

sonar.password=atharv@123

sonar.sources=.

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

Step 12:
Grant the local user (e.g., admin) permission to execute analysis on SonarQube by navigating to:

```
http://localhost:<port_number>/admin/permissions
```

Check the "Execute Analysis" box.

Name:Atharv Nikam Div:D15C Roll:36



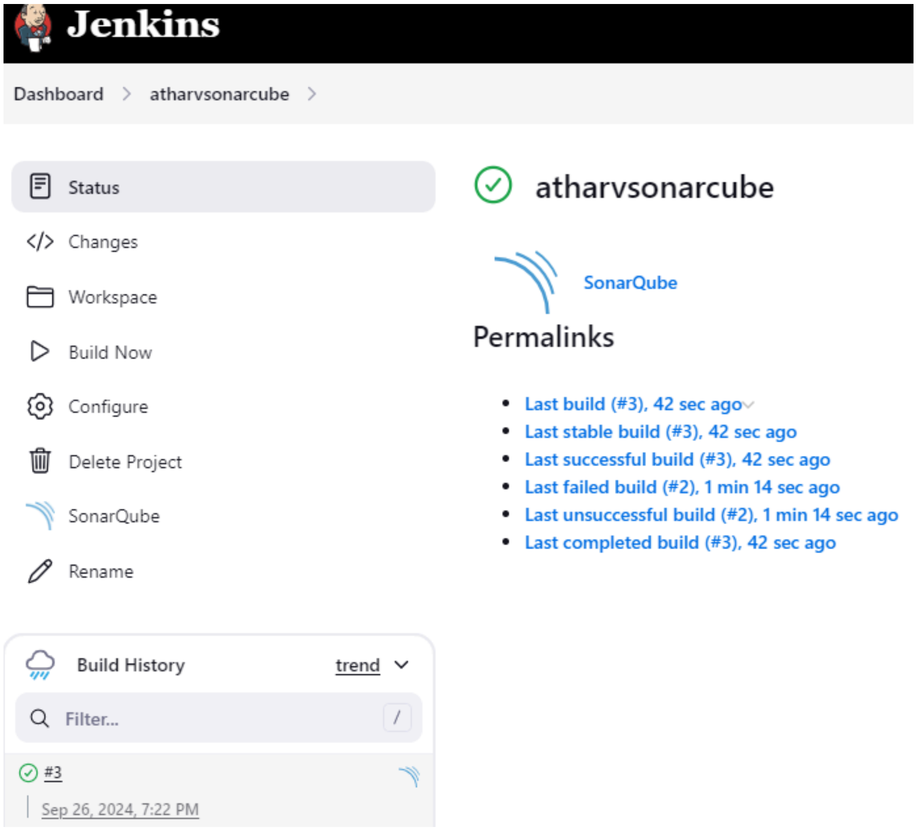
Global Permissions

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
A **Administrator** admin	☑	□ Quality Gates □ Quality Profiles	☑	□ Projects

Step 13:

Go back to Jenkins and trigger a build by selecting **Build Now** for the project you created.



atharvsonarcube

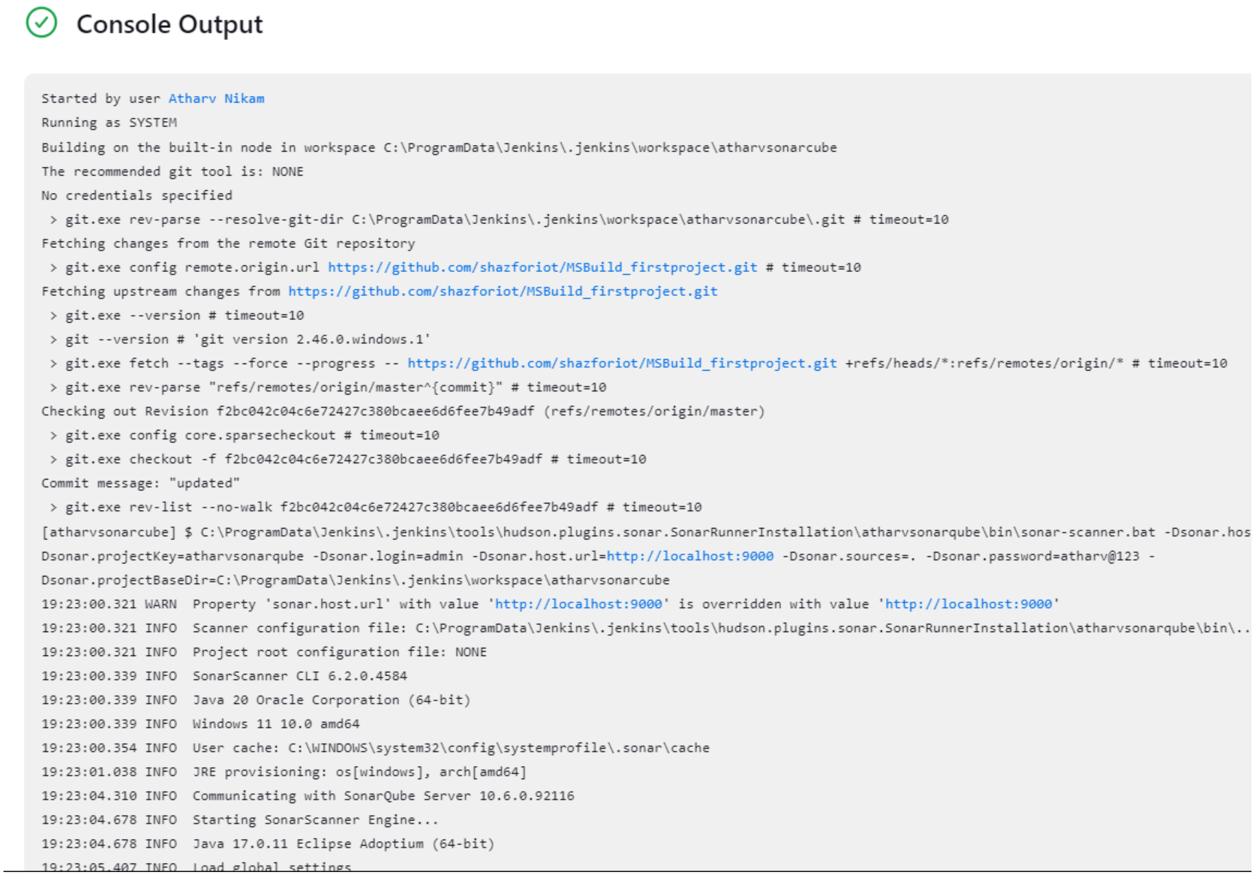
SonarQube

Permalinks

- Last build (#3), 42 sec ago
- Last stable build (#3), 42 sec ago
- Last successful build (#3), 42 sec ago
- Last failed build (#2), 1 min 14 sec ago
- Last unsuccessful build (#2), 1 min 14 sec ago
- Last completed build (#3), 42 sec ago

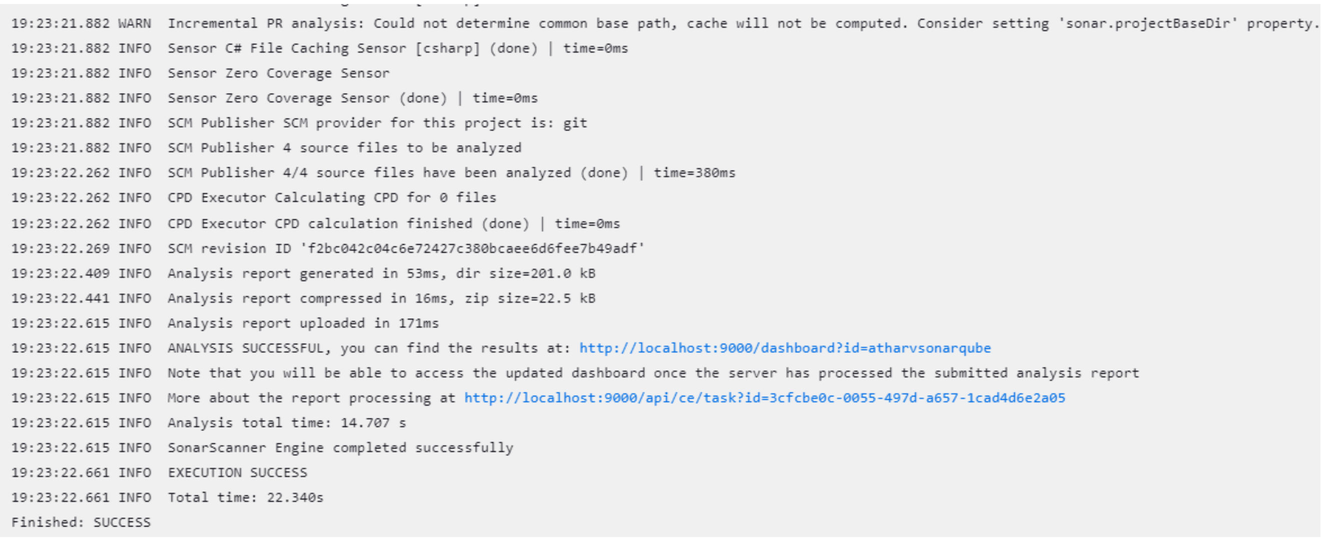
#3
Sep 26, 2024, 7:22 PM

Name:Atharv Nikam Div:D15C Roll:36



Console Output

```
Started by user Atharv Nikam
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube\.git # timeout=10
Fetching changes from the remote Git repository
 > 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
 > git.exe --version # timeout=10
 > git --version # 'git version 2.46.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > 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"
 > git.exe rev-list --no-walk f2bc042c04c6e72427c380bcaee6d6fee7b49adf # timeout=10
[atharvsonarcube] $ C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\atharvsonarqube\bin\sonar-scanner.bat -Dsonar.hos
Dsonar.projectKey=atharvsonarqube -Dsonar.login=admin -Dsonar.host.url=http://localhost:9000 -Dsonar.sources=. -Dsonar.password=atharv@123 -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\.jenkins\workspace\atharvsonarcube
19:23:00.321 WARN  Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
19:23:00.321 INFO  Scanner configuration file: C:\ProgramData\Jenkins\.jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\atharvsonarqube\bin\..
19:23:00.321 INFO  Project root configuration file: NONE
19:23:00.339 INFO  SonarScanner CLI 6.2.0.4584
19:23:00.339 INFO  Java 20 Oracle Corporation (64-bit)
19:23:00.339 INFO  Windows 11 10.0 amd64
19:23:00.354 INFO  User cache: C:\WINDOWS\system32\config\systemprofile\.sonar\cache
19:23:01.038 INFO  JRE provisioning: os[windows], arch[amd64]
19:23:04.310 INFO  Communicating with SonarQube Server 10.6.0.92116
19:23:04.678 INFO  Starting SonarScanner Engine...
19:23:04.678 INFO  Java 17.0.11 Eclipse Adoptium (64-bit)
19:23:05.407 INFO  Load global settings
```

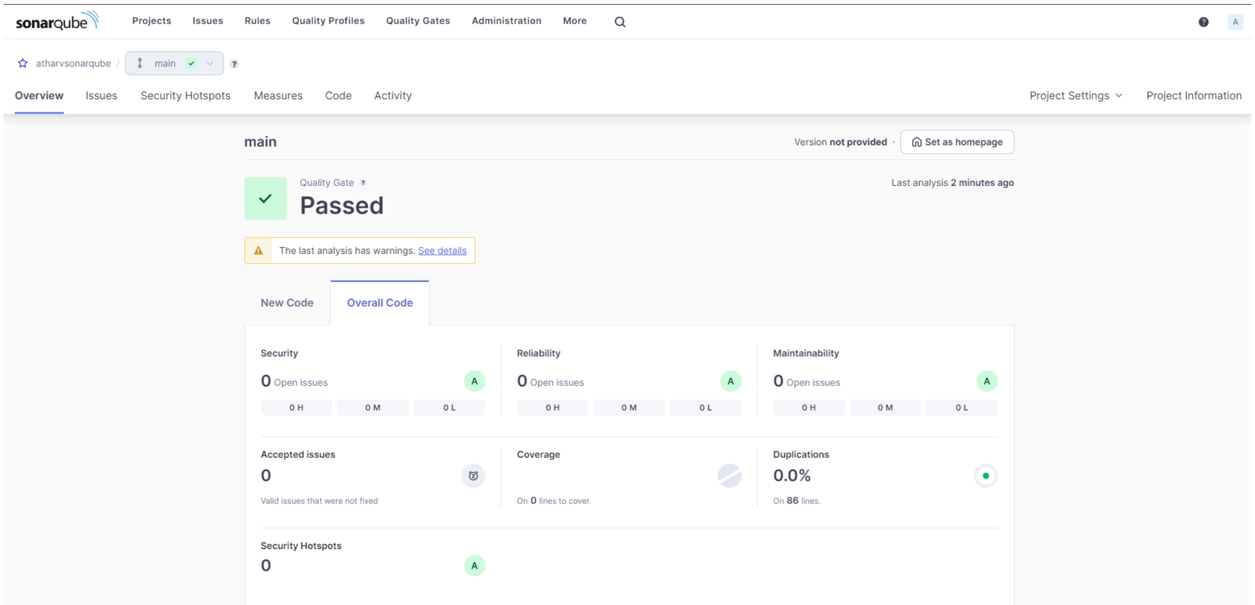


```
19:23:21.882 WARN  Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
19:23:21.882 INFO  Sensor C# File Caching Sensor [csharp] (done) | time=0ms
19:23:21.882 INFO  Sensor Zero Coverage Sensor
19:23:21.882 INFO  Sensor Zero Coverage Sensor (done) | time=0ms
19:23:21.882 INFO  SCM Publisher SCM provider for this project is: git
19:23:21.882 INFO  SCM Publisher 4 source files to be analyzed
19:23:22.262 INFO  SCM Publisher 4/4 source files have been analyzed (done) | time=380ms
19:23:22.262 INFO  CPD Executor Calculating CPD for 0 files
19:23:22.262 INFO  CPD Executor CPD calculation finished (done) | time=0ms
19:23:22.269 INFO  SCM revision ID 'f2bc042c04c6e72427c380bcaee6d6fee7b49adf'
19:23:22.409 INFO  Analysis report generated in 53ms, dir size=201.0 kB
19:23:22.441 INFO  Analysis report compressed in 16ms, zip size=22.5 kB
19:23:22.615 INFO  Analysis report uploaded in 171ms
19:23:22.615 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=atharvsonarqube
19:23:22.615 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
19:23:22.615 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=3cfcbe0c-0055-497d-a657-1cad4d6e2a05
19:23:22.615 INFO  Analysis total time: 14.707 s
19:23:22.615 INFO  SonarScanner Engine completed successfully
19:23:22.661 INFO  EXECUTION SUCCESS
19:23:22.661 INFO  Total time: 22.340s
Finished: SUCCESS
```

Name:Atharv Nikam Div:D15C Roll:36

Step 14:

Once the build process is complete, head back to SonarQube and verify the analysis results linked to your project.



main

Version not provided

Quality Gate

Passed

Last analysis 2 minutes ago

The last analysis has warnings. See details

Security
0 Open issues A
0 H 0 M 0 L

Reliability
0 Open issues A
0 H 0 M 0 L

Maintainability
0 Open issues A
0 H 0 M 0 L

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

Conclusion:

Through this experiment, we explored how to set up and perform Static Application Security Testing (SAST) using Jenkins in combination with SonarQube. By leveraging Docker, we utilized a SonarQube container without needing a local installation. Following the setup of Jenkins and SonarQube, we analyzed a sample project from GitHub for potential vulnerabilities. Once the project was built, the analysis confirmed the security status, providing valuable insight into the effectiveness of SAST in the CI/CD pipeline.