Name:Sahil Motiramani Div:D15C Roll No.35

Adv DevOps Practical 7

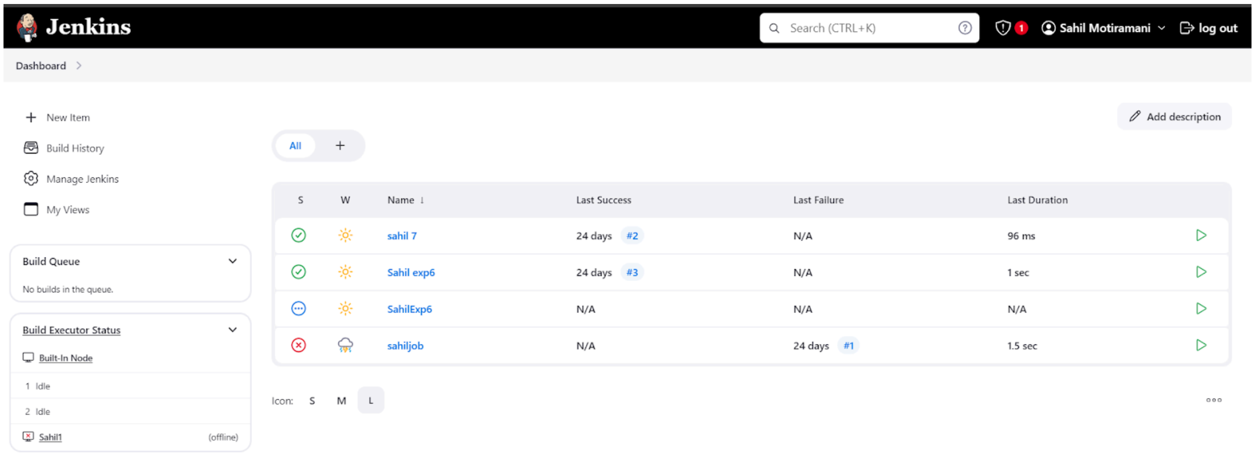
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

Integrating Jenkins with SonarQube:

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

Steps to integrate Jenkins with SonarQube

1. Open up Jenkins Dashboard on localhost, port 8090 or whichever port it is at for you.

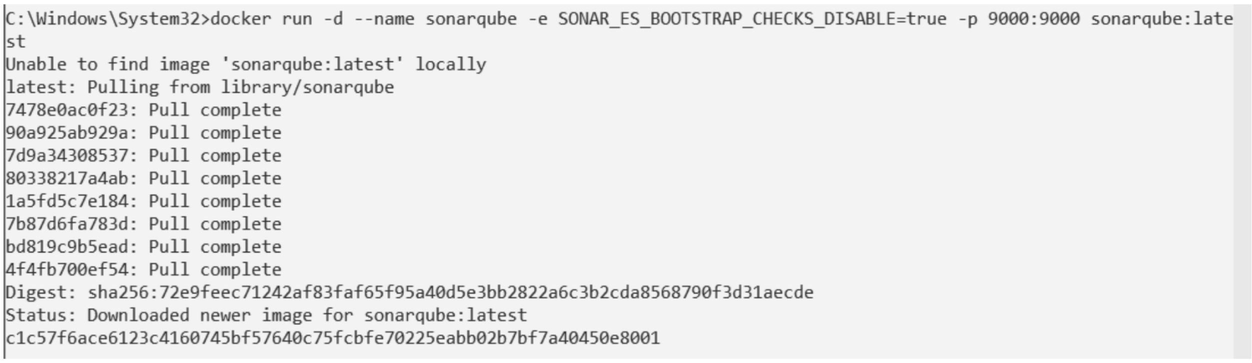


S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	sahil 7	24 days #2	N/A	96 ms	▷
✓	☀	Sahil exp6	24 days #3	N/A	1 sec	▷
⋯	☀	SahilExp6	N/A	N/A	N/A	▷
✗	🌧	sahiljob	N/A	24 days #1	1.5 sec	▷

2. 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

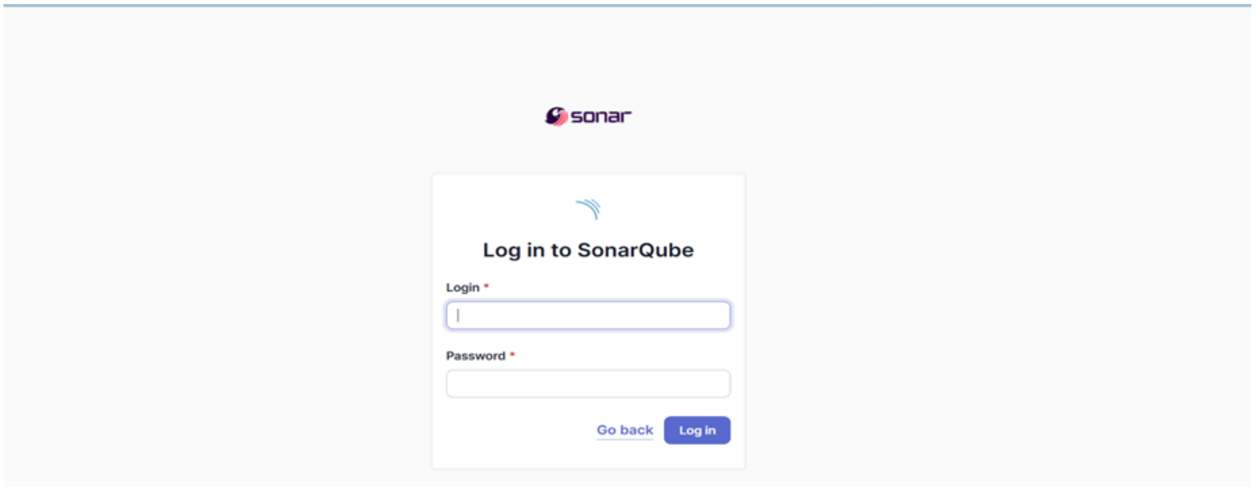
------------------Warning: run below command only once



```
C:\Windows\System32>docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:late
st
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
c1c57f6ace6123c4160745bf57640c75fcbfe70225eabb02b7bf7a40450e8001
```

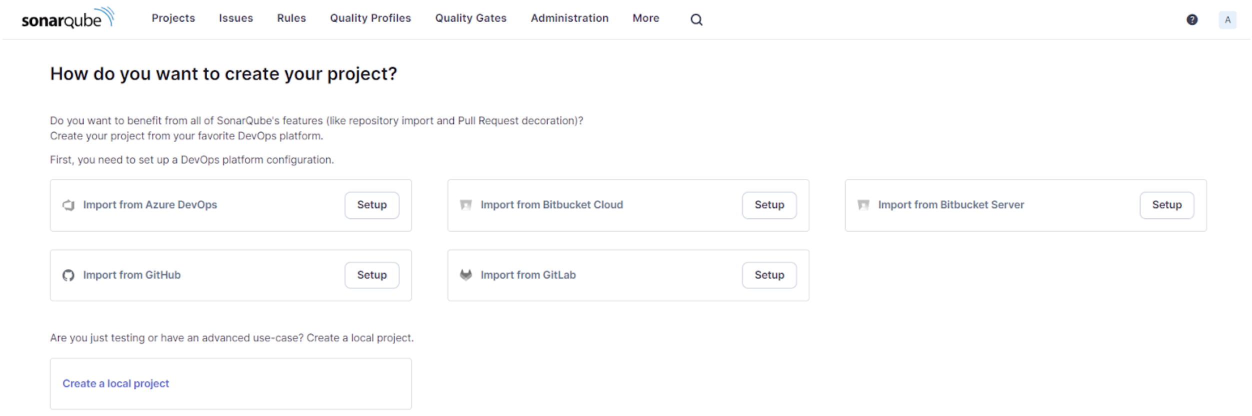
3. Once the container is up and running, you can check the status of SonarQube at

localhost port 9000.



Name:Sahil Motiramani Div:D15C Roll No.35

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



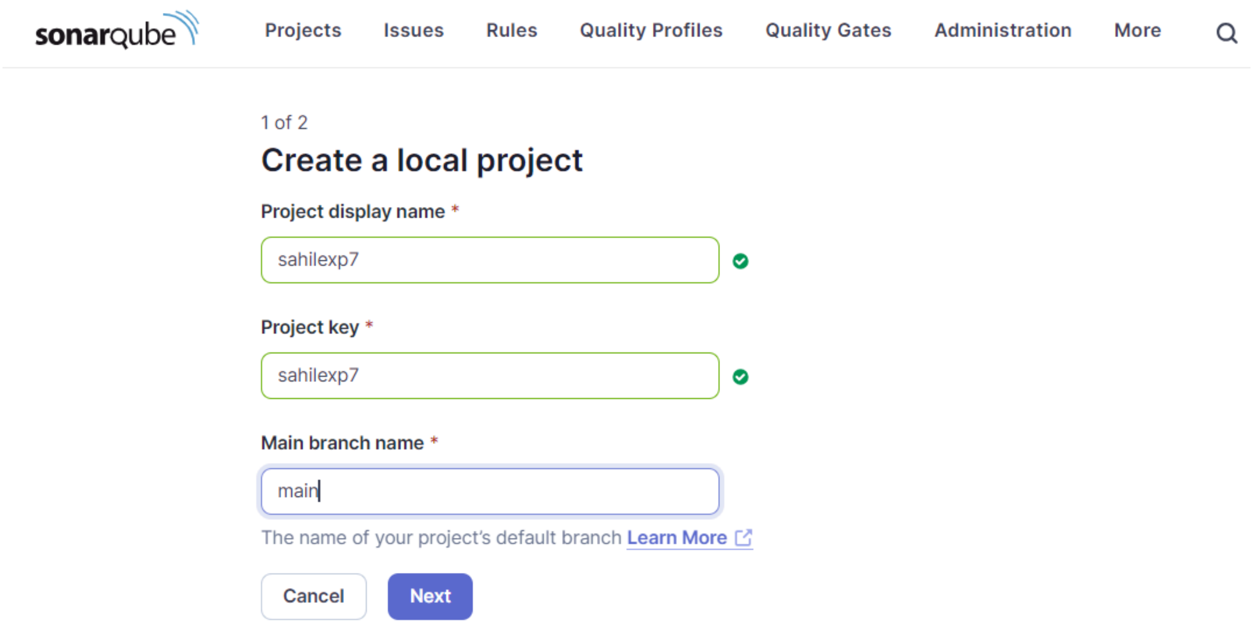
How do you want to create your project?

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.

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



1 of 2

Create a local project

Project display name *

sahilexp7

Project key *

sahilexp7

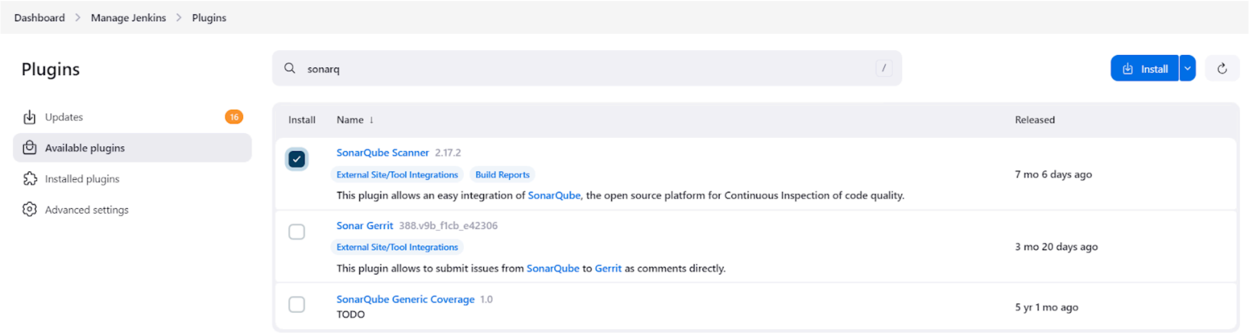
Main branch name *

main

The name of your project's default branch Learn More

Setup the project and come back to Jenkins Dashboard.

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



Dashboard > Manage Jenkins > Plugins

Plugins

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 6 days ago
□	Sonar Gerrit 388.v9b_f1cb_e42306 External Site/Tool Integrations This plugin allows to submit issues from SonarQube to Gerrit as comments directly.	3 mo 20 days ago
□	SonarQube Generic Coverage 1.0 TODO	5 yr 1 mo ago

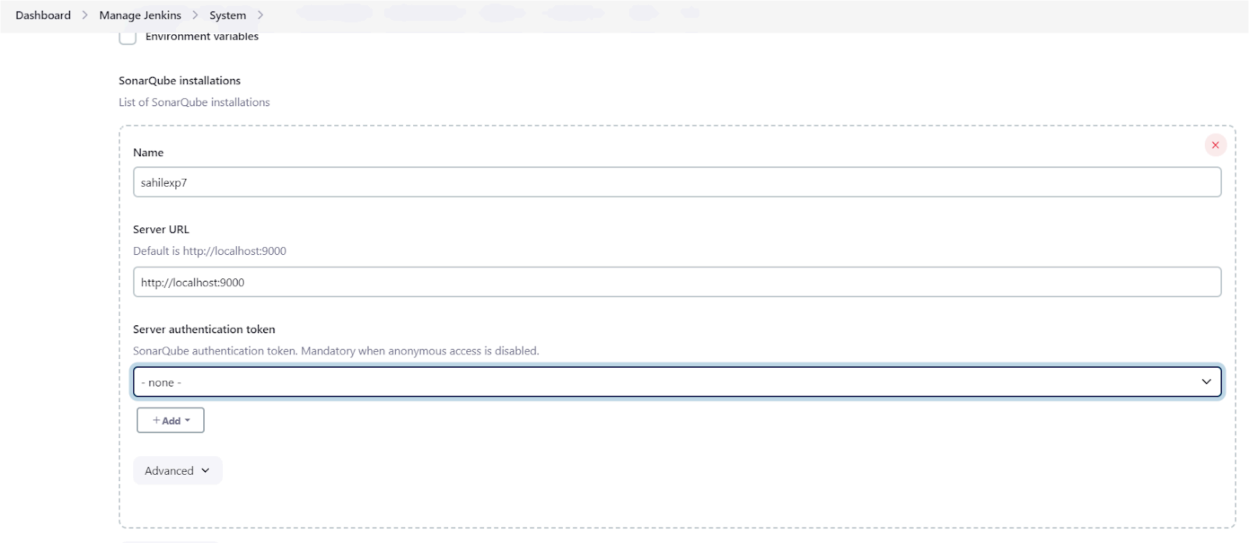
6. Under Jenkins ‘Manage Jenkins’ then go to ‘system’, scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me

sahilexp7

In **Server URL** Default is http://localhost:9000



Dashboard > Manage Jenkins > System >

□ Environment variables

SonarQube installations

List of SonarQube installations

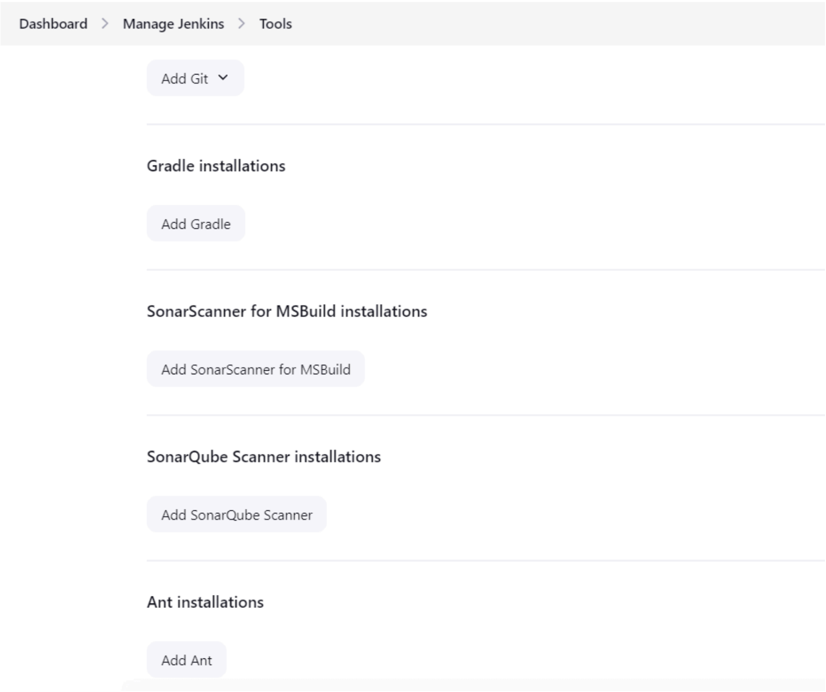
Name: sahilexp7

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

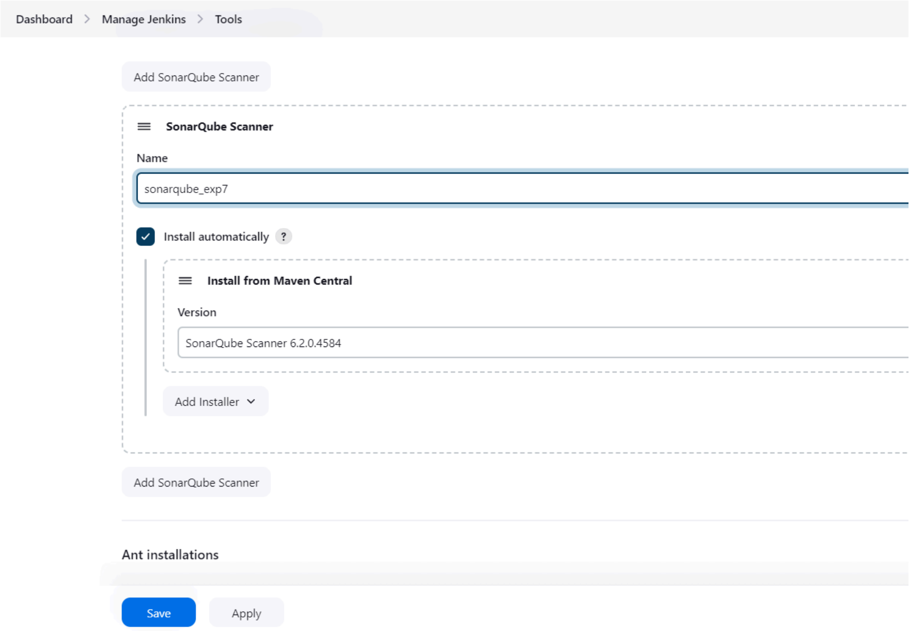
Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
- none -

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

Dashboard > Manage Jenkins > Tools


Dashboard > Manage Jenkins > Tools
Add Git
Gradle installations
Add Gradle
SonarScanner for MSBuild installations
Add SonarScanner for MSBuild
SonarQube Scanner installations
Add SonarQube Scanner
Ant installations
Add Ant

Check the “Install automatically” option. → Under name any name as identifier → Check the “Install automatically” option.

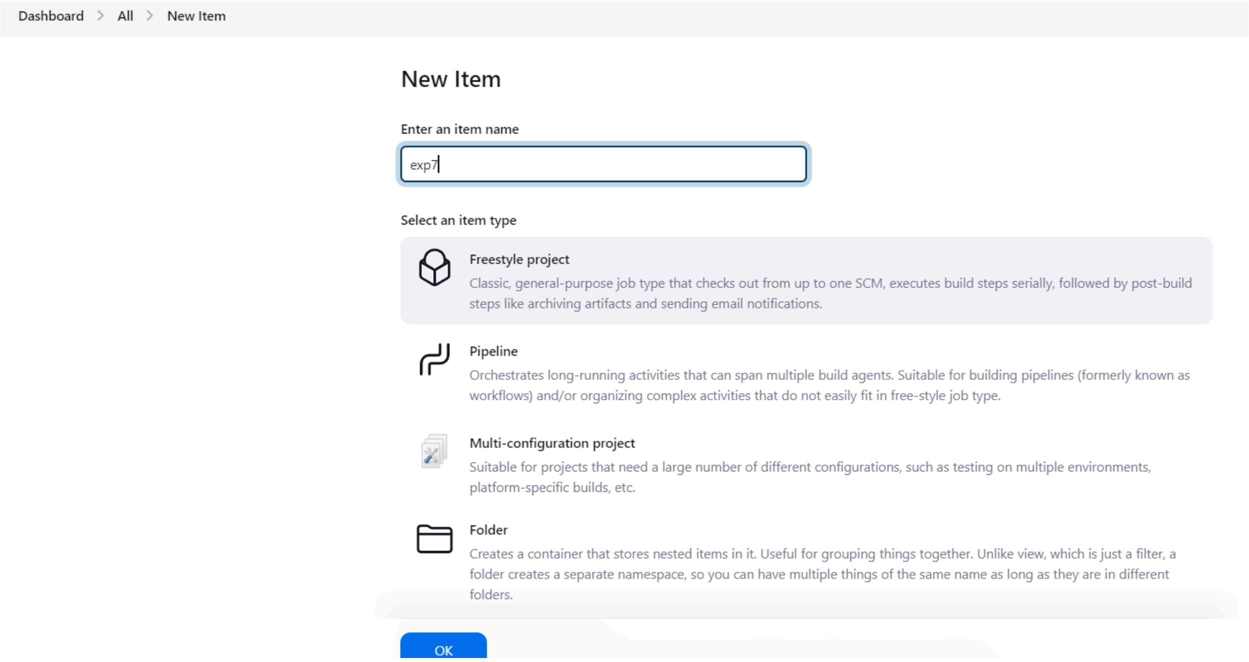


Name: sonarqube_exp7

☑ Install automatically ?

Version: SonarQube Scanner 6.2.0.4584

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



Enter an item name

exp7

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.

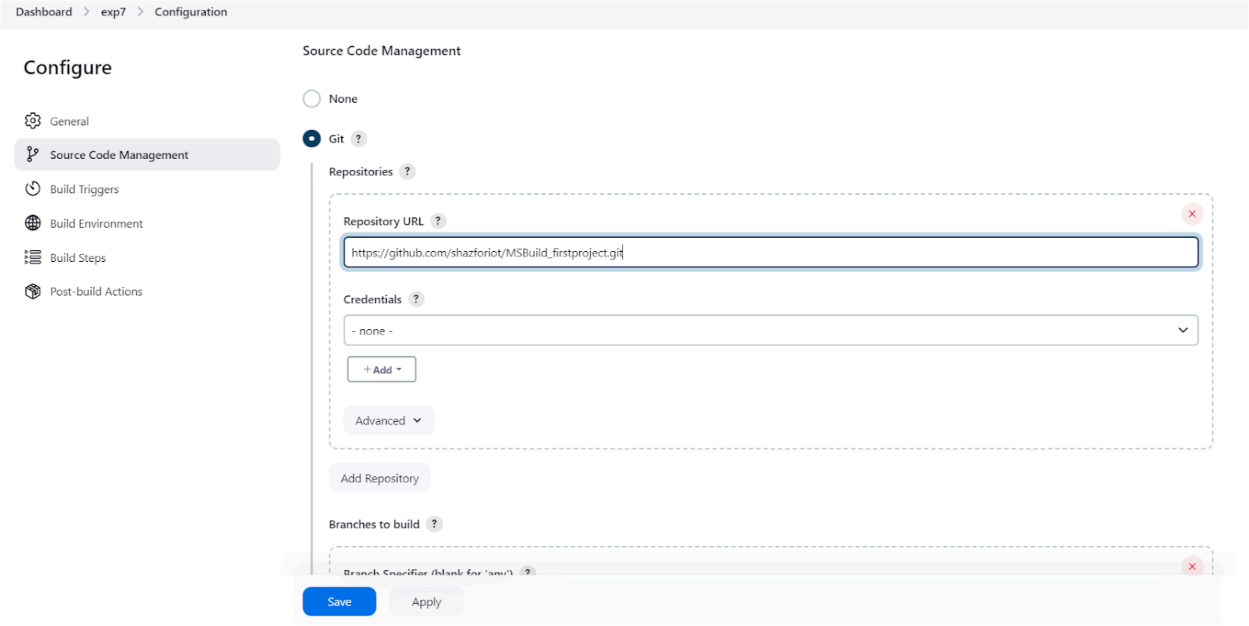
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

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.

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.



Source Code Management

○ None

● Git

Repositories

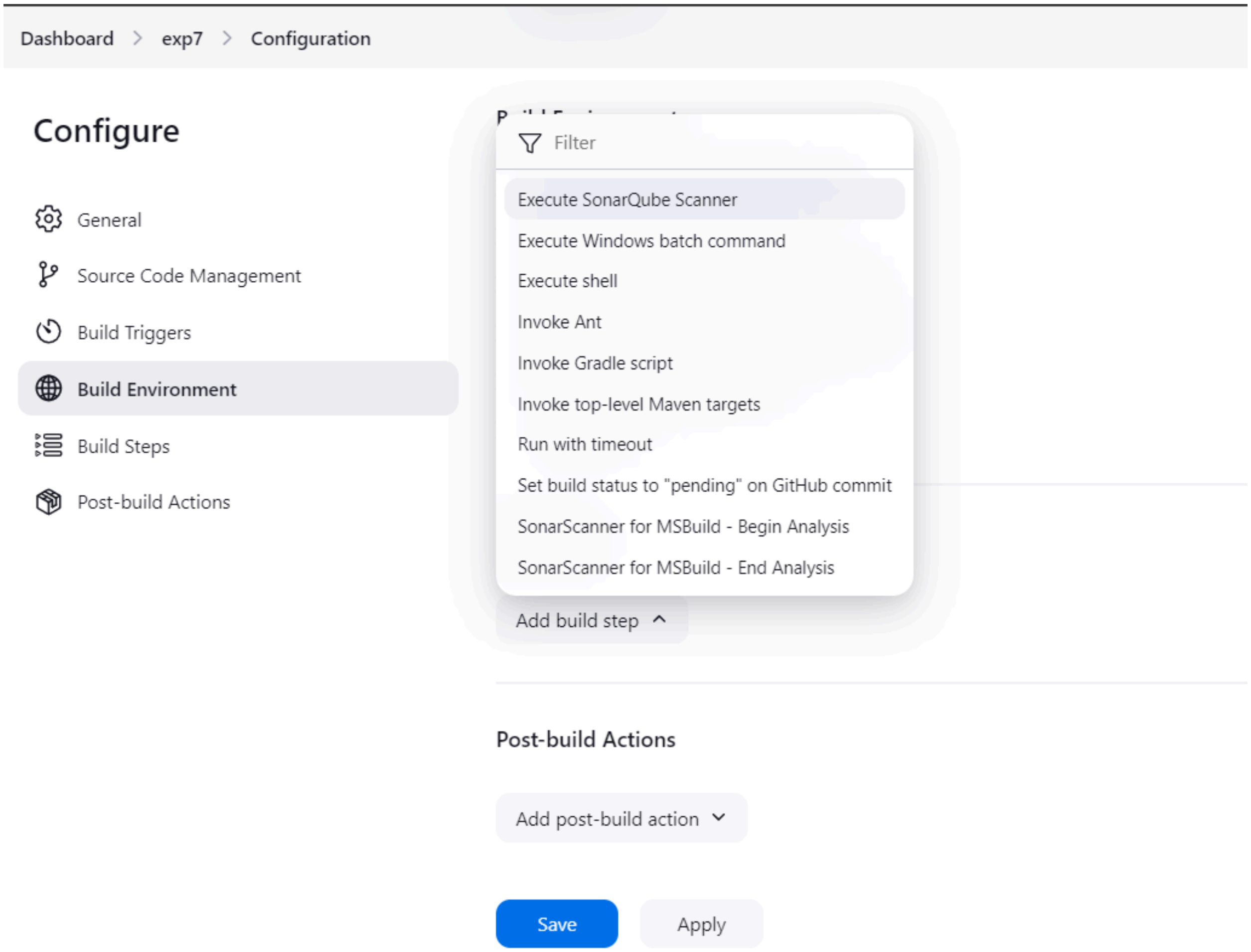
Repository URL: https://github.com/shazforiot/MSBuild_firstproject.git

Credentials: - none -

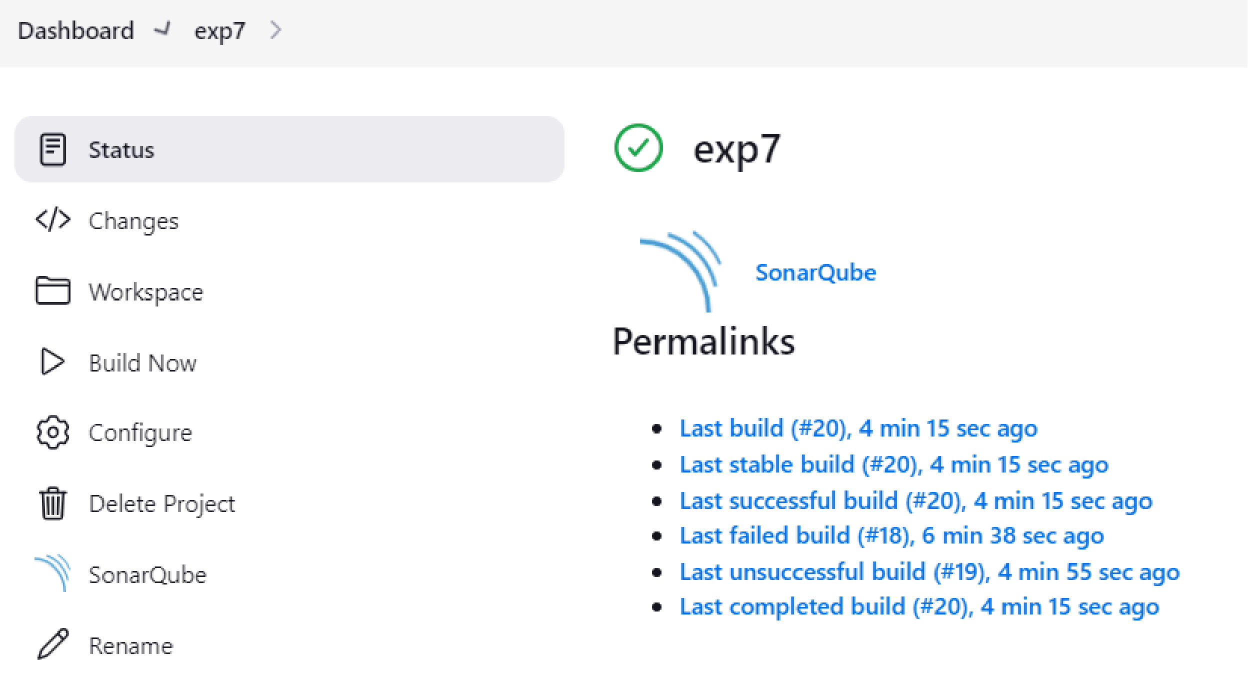
Branches to build

Branch Specifier (blank for 'any')

10. Under **Select project → Configuration → Build steps → Execute SonarQube Scanner**, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.


Dashboard > exp7 > Configuration
Configure
General
Source Code Management
Build Triggers
Build Environment
Build Steps
Post-build Actions
Filter
Execute SonarQube Scanner
Execute Windows batch command
Execute shell
Invoke Ant
Invoke Gradle script
Invoke top-level Maven targets
Run with timeout
Set build status to "pending" on GitHub commit
SonarScanner for MSBuild - Begin Analysis
SonarScanner for MSBuild - End Analysis
Add build step
Post-build Actions
Add post-build action
Save
Apply

Then save

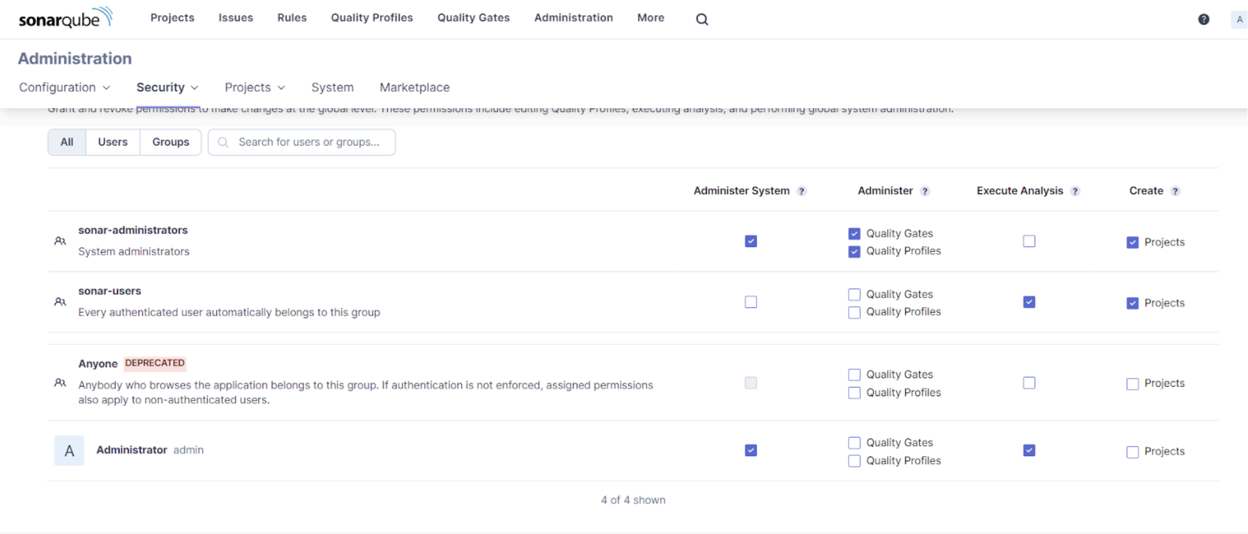


exp7

Permalinks

- Last build (#20), 4 min 15 sec ago
- Last stable build (#20), 4 min 15 sec ago
- Last successful build (#20), 4 min 15 sec ago
- Last failed build (#18), 6 min 38 sec ago
- Last unsuccessful build (#19), 4 min 55 sec ago
- Last completed build (#20), 4 min 15 sec ago

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



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

4 of 4 shown

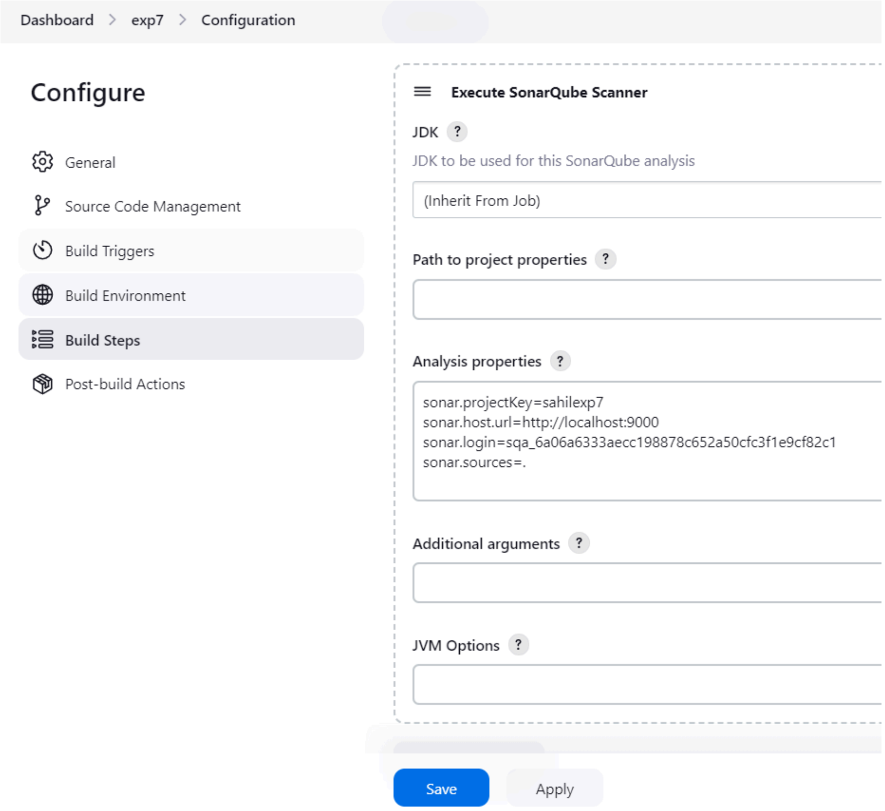
IF CONSOLE OUTPUT FAILED:

Step 1: Generate a New Authentication Token in SonarQube

1. **Login to SonarQube:**
    - Open your browser and go to `http://localhost:9000`.
    - Log in with your admin credentials (default username is `admin`, and the password is either `admin` or your custom password if it was changed).
2. **Generate a New Token:**
    - Click on your **username** in the top-right corner of the SonarQube dashboard.
    - Select **My Account** from the dropdown menu.
    - Go to the **Security** tab.
    - Under **Generate Tokens**, type a name for the token (e.g., "Jenkins-SonarQube").
    - Click **Generate**.
    - Copy the token and save it securely. You will need it in Jenkins.

Step 2: Update the Token in Jenkins

1. **Go to Jenkins Dashboard:**
    - Open Jenkins and log in with your credentials.
2. **Configure the Jenkins Job:**
    - Go to the job that is running the SonarQube scanner (`adv_devops_exp7`).
    - Click **Configure**.
3. **Update the SonarQube Token:**
    - In the SonarQube analysis configuration (either in the pipeline script or under "Build" section, depending on your job type), update the `sonar.login` parameter with the new token.



Execute SonarQube Scanner

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

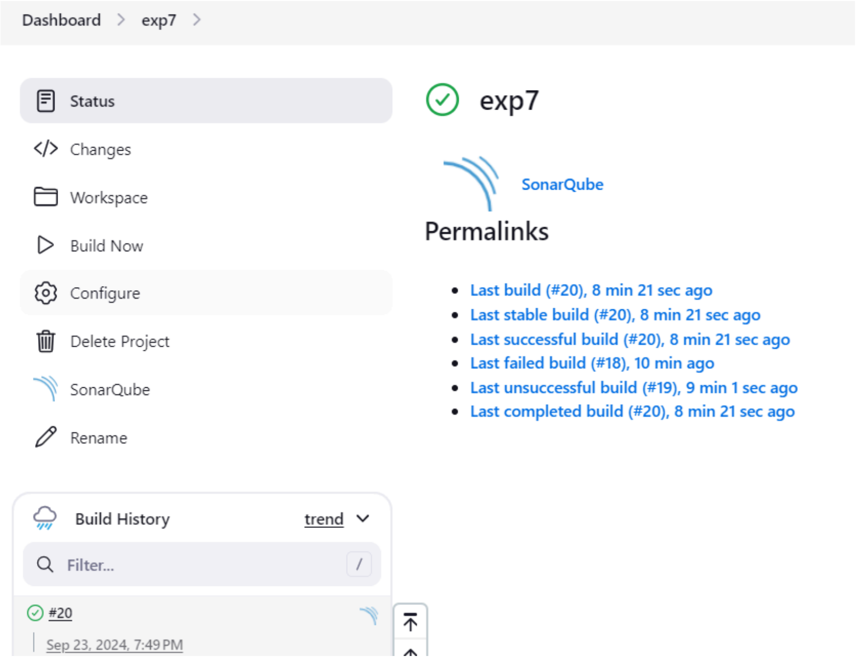
Analysis properties ?

```
sonar.projectKey=sahilexp7
sonar.host.url=http://localhost:9000
sonar.login=sqa_6a06a6333aecc198878c652a50cfc3f1e9cf82c1
sonar.sources=.
```

Additional arguments ?

JVM Options ?

12. Run the Jenkins build.



exp7

SonarQube

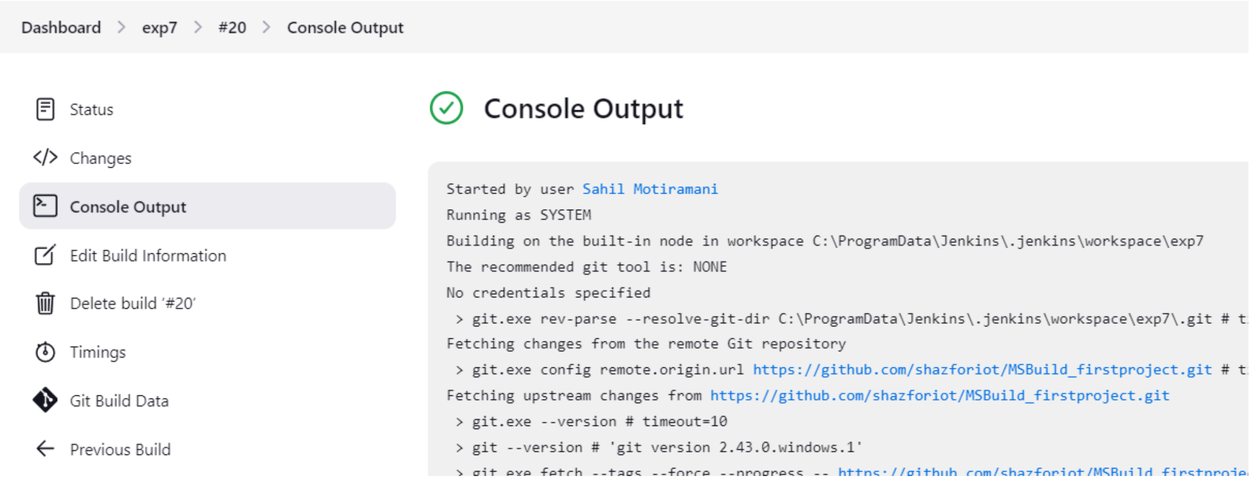
Permalinks

- Last build (#20), 8 min 21 sec ago
- Last stable build (#20), 8 min 21 sec ago
- Last successful build (#20), 8 min 21 sec ago
- Last failed build (#18), 10 min ago
- Last unsuccessful build (#19), 9 min 1 sec ago
- Last completed build (#20), 8 min 21 sec ago

#20 Sep 23, 2024, 7:49 PM

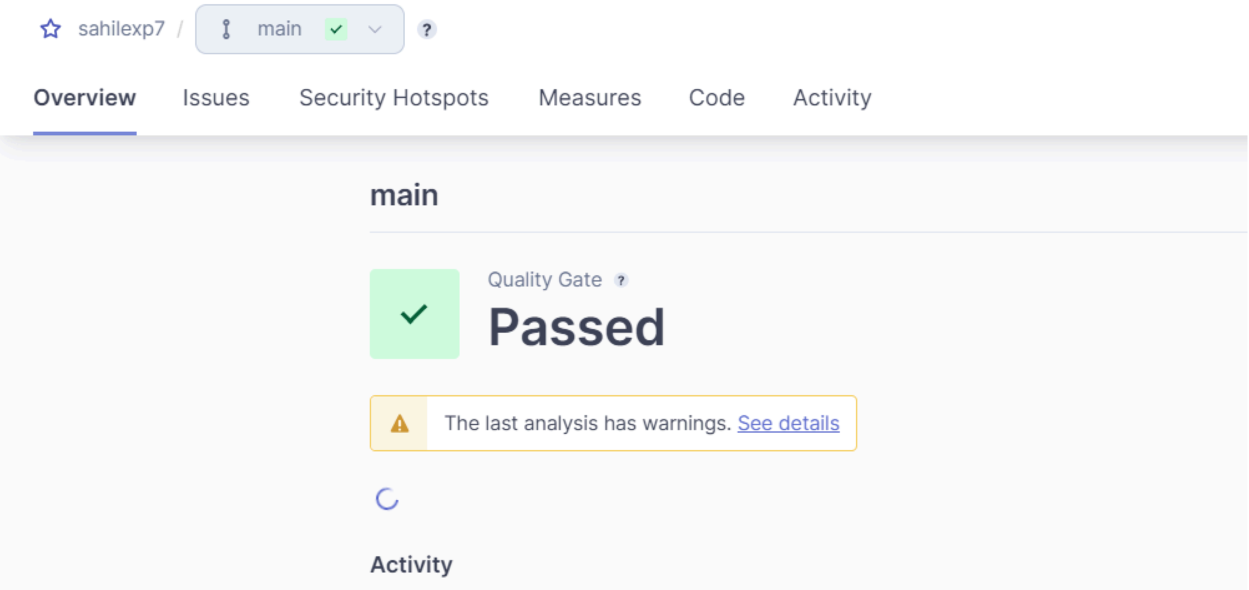
Check the console Output

Console output;



```
Started by user Sahil Motiramani
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\exp7
The recommended git tool is: NONE
No credentials specified
 > git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\.jenkins\workspace\exp7\.git # t
Fetching changes from the remote Git repository
 > git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # t
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
 > git.exe --version # timeout=10
 > git --version # 'git version 2.43.0.windows.1'
 > git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproje
```

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



main

Quality Gate

Passed

The last analysis has warnings. See details

Activity

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

Conclusion:

In this project, we integrated Jenkins with SonarQube for automated static application security testing (SAST). We set up SonarQube using Docker, configured Jenkins with the necessary plugins and authentication, and linked it to a GitHub repository. The SonarQube scanner was added as a build step, enabling continuous code analysis for vulnerabilities, code smells, and quality issues, ensuring automated reporting and continuous code quality improvement.