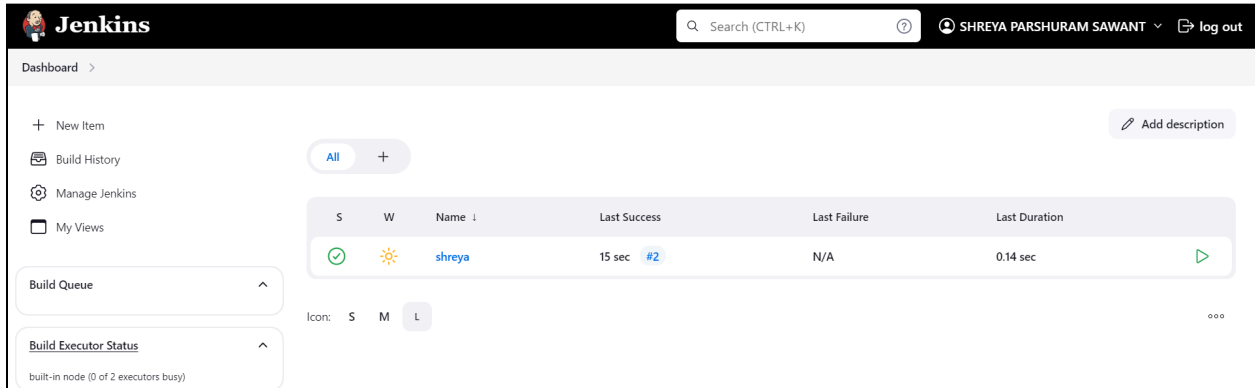


Experiment.7

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.



Step.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
```

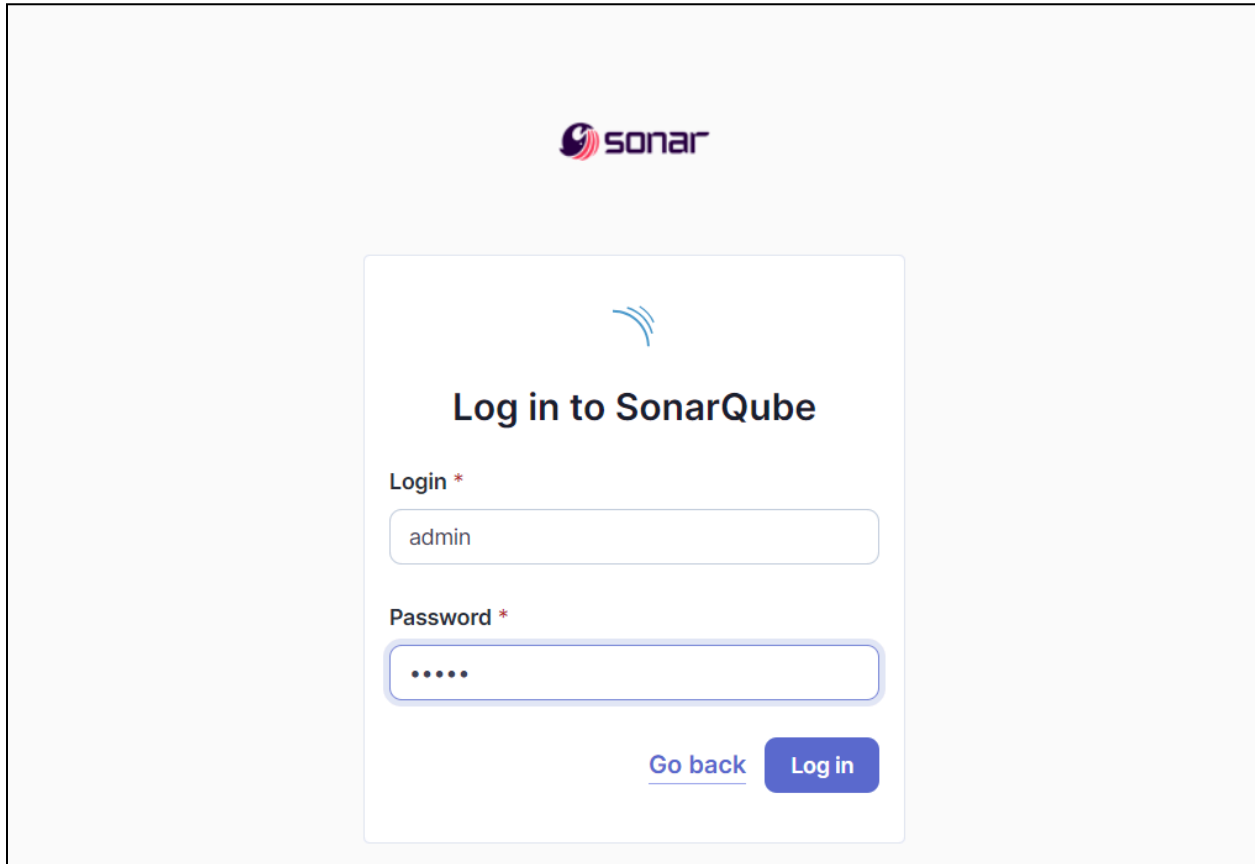
```
PS C:\Users\sawan> docker -v
Docker version 27.1.1, build 6312585
```

```
PS C:\Users\sawan> 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
23c4905f5f439b89ce26341b6f016db921c6d4546f70a2450366e258833428e7
PS C:\Users\sawan> |
```

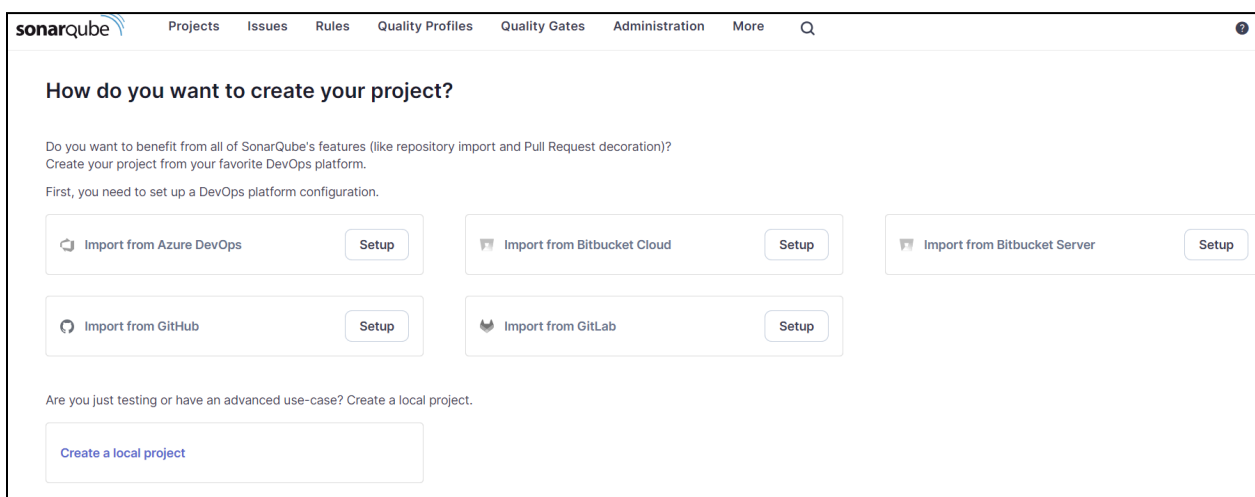
Shreya Sawant
D15A - 54

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

Login to SonarQube using username admin and password admin.



The image shows the SonarQube login page. At the top center is the Sonar logo. Below it is a large white box with a blue Sonar icon and the text "Log in to SonarQube". Underneath, there are two input fields: "Login *" with the text "admin" and "Password *" with masked characters ".....". At the bottom right of the box are two buttons: "Go back" (a link) and "Log in" (a blue button).




The image shows the SonarQube project creation page. At the top is the SonarQube logo and a navigation bar with links: Projects, Issues, Rules, Quality Profiles, Quality Gates, Administration, More, and a search icon. Below the navigation bar is a heading "How do you want to create your project?". Underneath is a paragraph: "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." Below this are five buttons arranged in two rows: "Import from Azure DevOps", "Import from Bitbucket Cloud", "Import from Bitbucket Server", "Import from GitHub", and "Import from GitLab". Each button has a "Setup" button next to it. At the bottom, there is a paragraph: "Are you just testing or have an advanced use-case? Create a local project." and a button "Create a local project".

Step.4 Create a manual project in SonarQube with the name sonarqube

1 of 2


Create a local project

Project display name *

 CancelNext

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.

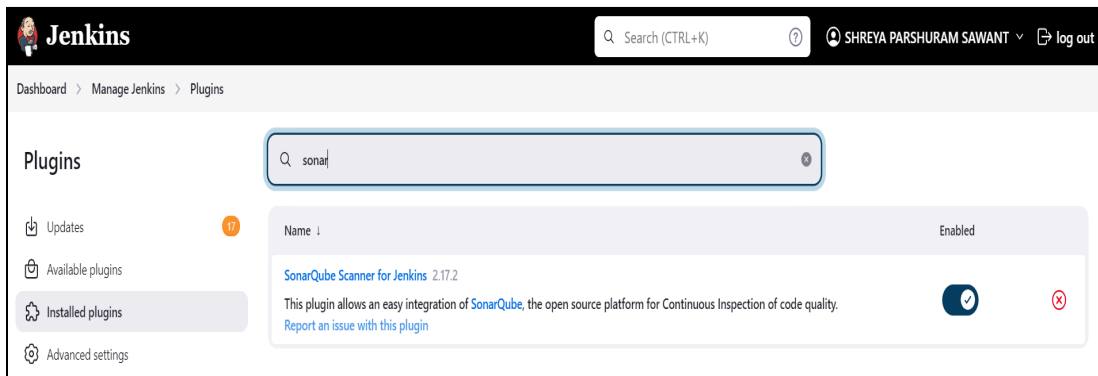
☐ Define a specific setting for this project

☐ Previous version
Any code that has changed since the previous version is considered new code.
Recommended for projects following regular versions or releases.

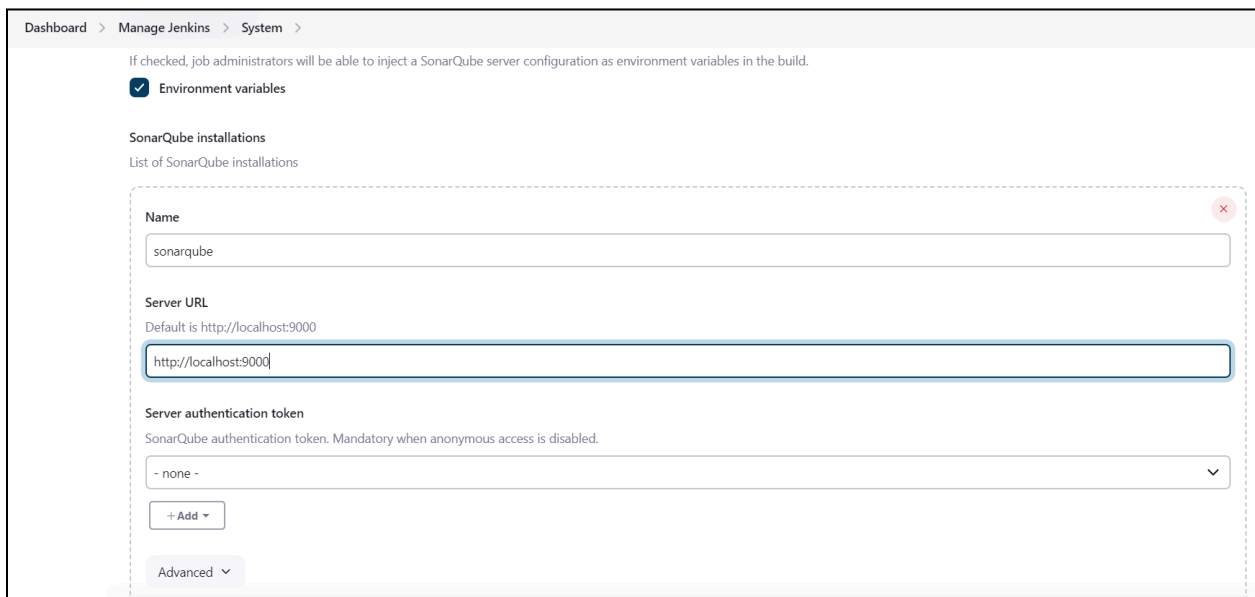
☐ Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.
Recommended for projects following continuous delivery.

Shreya Sawant
D15A - 54

Step.5 Setup the project and come back to Jenkins Dashboard.
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



Step.6 Under Jenkins 'Configure System', look for SonarQube Servers and enter the details.
Enter the Server Authentication token if needed.





Step.7 After the configuration, create a New Item in Jenkins, choose a freestyle project.


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.

**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.

OK

Configure

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

Repository URL ?

Credentials ?

- none -

+ Add

Advanced

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Add Branch

Save Apply

Shreya Sawant

D15A - 54

Step.8 Under Build-> Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.

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=shreya444
sonar.sources=
sonar.url=http://localhost:9000

Additional arguments ?

JVM Options ?

Save

Apply

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.					
All Users Groups		Search for users or groups...			
		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

Shreya Sawant
D15A - 54

Step.9 Run The Build.
Check the console output.

Search (CTRL+K)SHREYA PARSHURAM SAWANTlog out

le Output

✓ Console OutputDownloadCopyView as plain text

Started by user SHREYA PARSHURAM SAWANT
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\jobs\sonarqube\workspace
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\jobs\sonarqube\workspace # 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 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 f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"

The image shows the SonarQube web interface. At the top, there's a navigation bar with 'sonarqube' logo and tabs for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', and 'More'. Below the navigation bar, there's a 'Create Project' button. A search bar 'Search for projects...' is present. The main content area shows a list of projects. The first project is 'sonarqube PUBLIC' with a green checkmark and 'Passed' status. Below it, it says 'Last analysis: 18 minutes ago' and 'The main branch of this project is empty.' On the left sidebar, there's a 'Filters' section with a 'Quality Gate' filter showing 1 'Passed' and 0 'Failed' items.

