

ADVANCE DEVOPS EXP 7

Name: Soham Satpute

Class:D15A

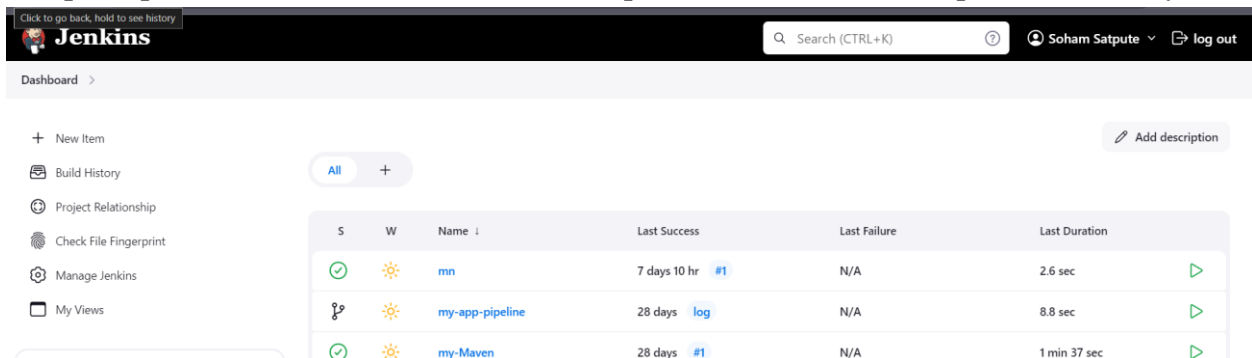
Roll No:52

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

Jenkins integration with SonarQube:

Steps:

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

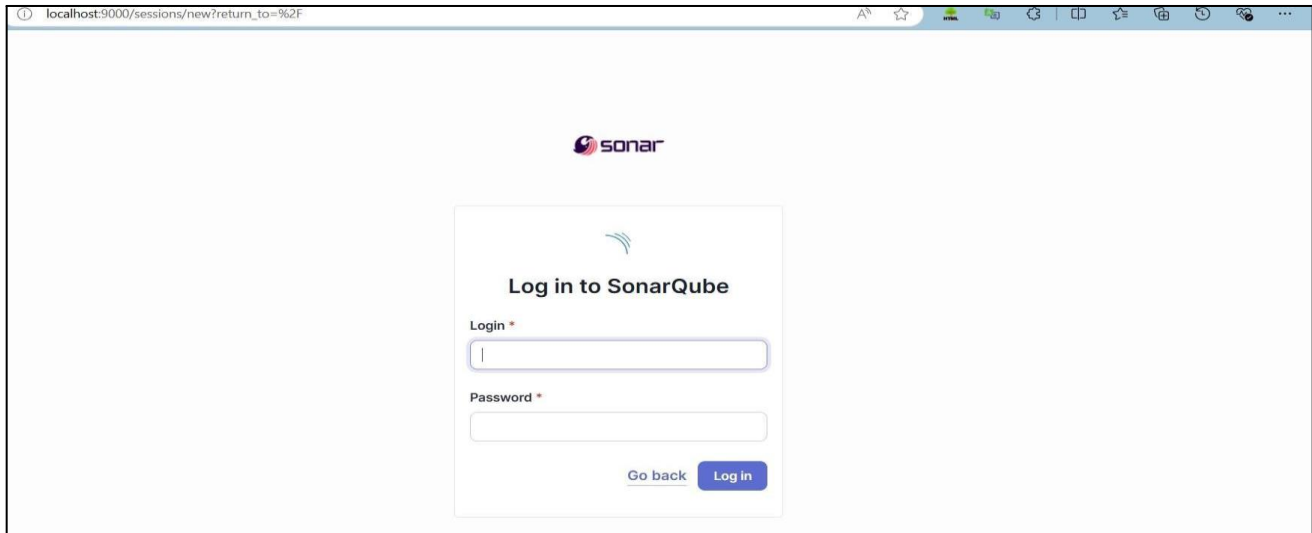


2.Run SonarQube in a Docker container using this command:

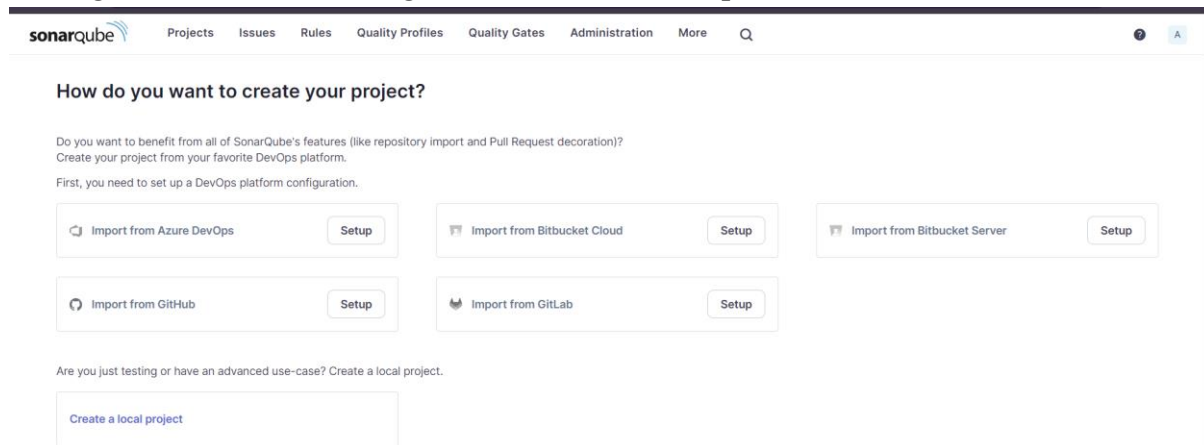
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

```
PS C:\Windows\system32> 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
5ab3928e5e27607e3661d129731e4e600a9019574c7dc2767aa9b3bfdaa941be
```

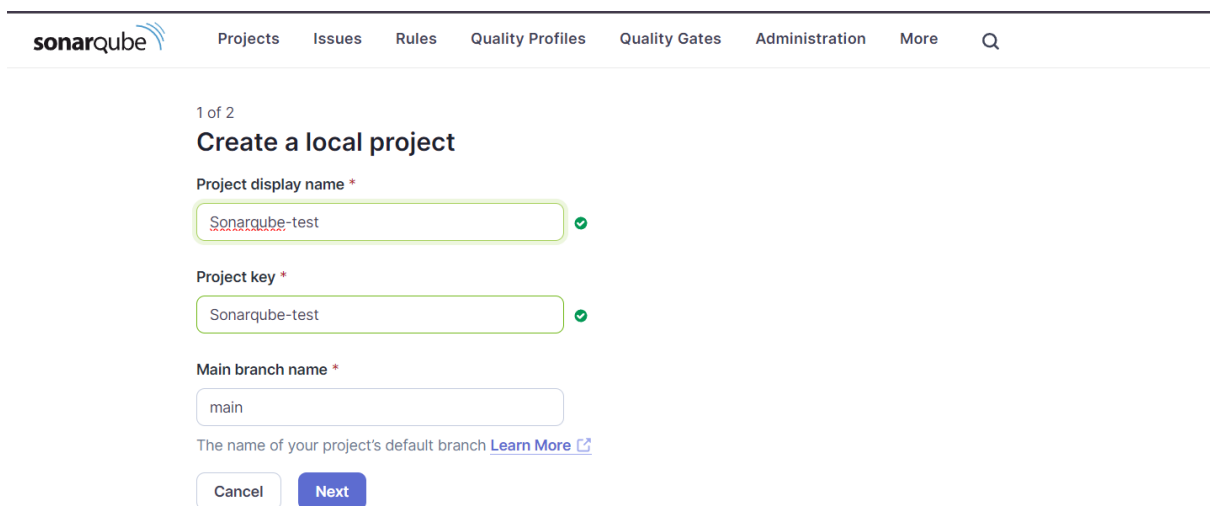
3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.

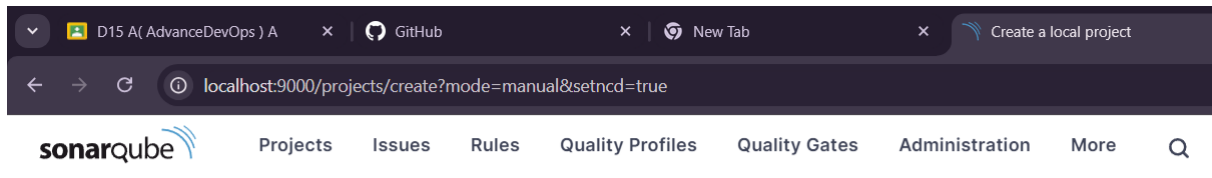


4. Login to SonarQube using username admin and password admin.



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





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 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.

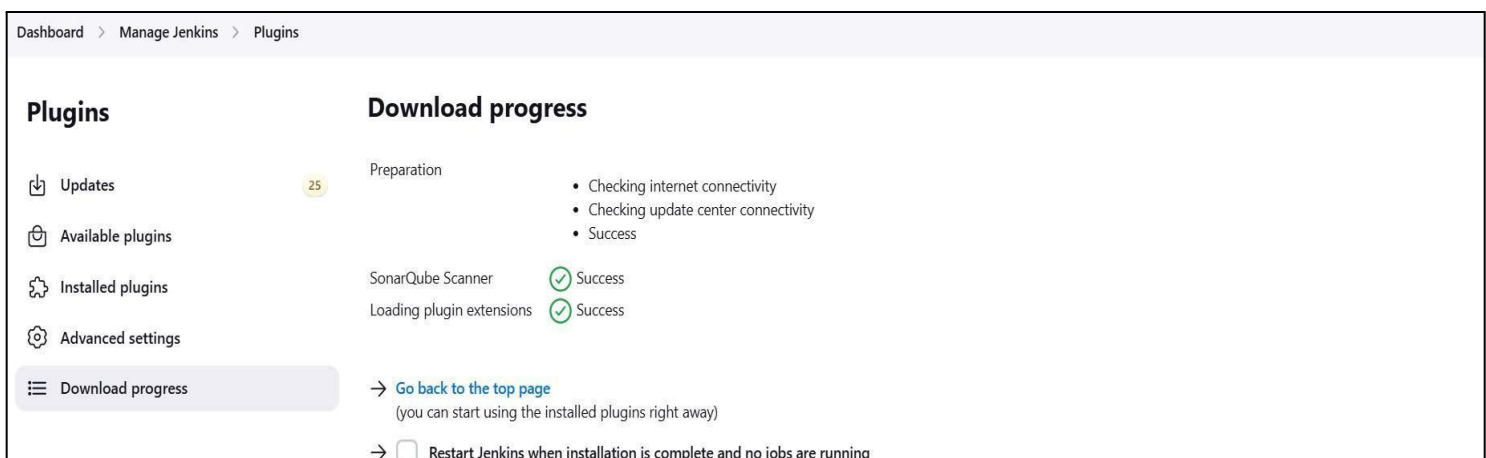
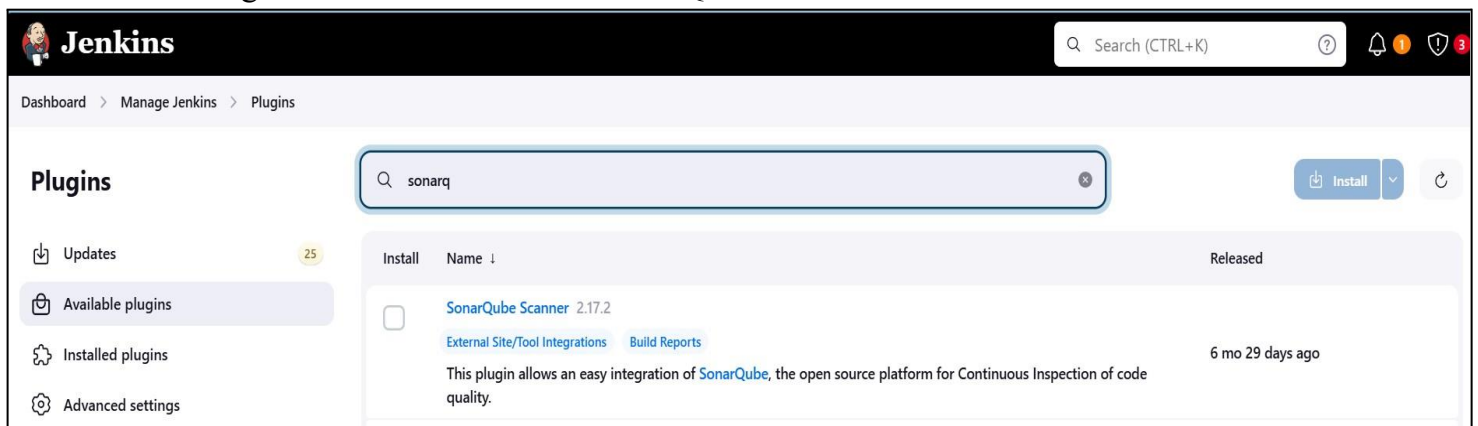
Recommended for projects following regular versions or releases.

☐ Define a specific setting for this project

☐ Previous version

Setup the project and come back to Jenkins Dashboard.

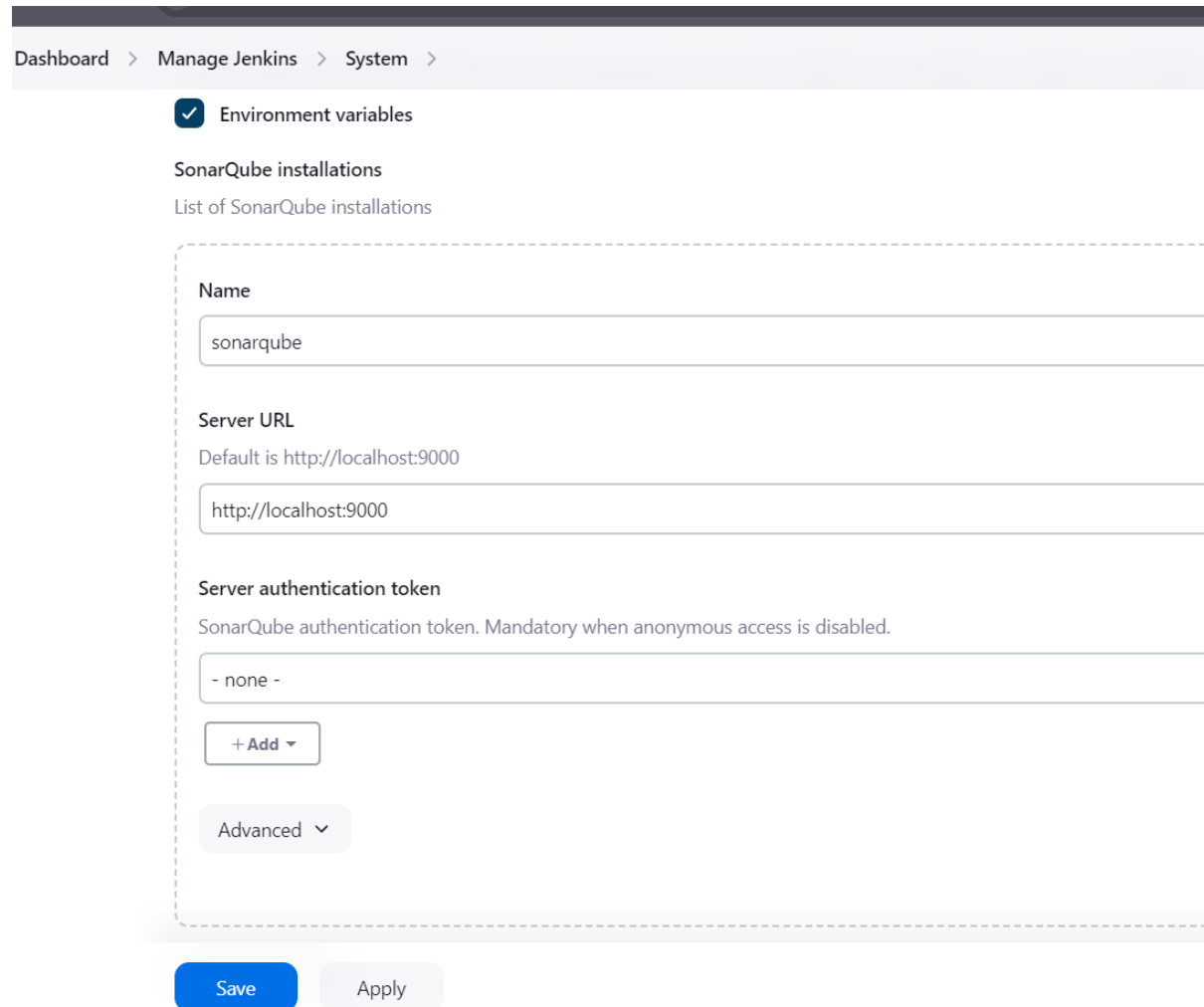
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



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>, here we have named it as **adv_devops_7_sonarqube** In **Server URL** Default is <http://localhost:9000>



The screenshot shows the Jenkins 'Manage Jenkins' > 'System' > 'Environment variables' page. The 'SonarQube installations' section is active, showing a list of installations. The first installation is named 'sonarqube'. The 'Server URL' is set to 'http://localhost:9000'. The 'Server authentication token' is set to '- none -'. There is a '+ Add' button and an 'Advanced' dropdown menu. At the bottom, there are 'Save' and 'Apply' buttons.

Dashboard > Manage Jenkins > System >

☒ 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 ▾

Save Apply

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

SonarQube Scanner installations ^  Edited

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

Save

Apply

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

Dashboard > All > New Item

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.

OK

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.

The screenshot shows the 'Source Code Management' configuration window. At the top, there are two radio buttons: 'None' and 'Git'. The 'Git' option is selected. Below this, there is a 'Repositories' section. Inside this section, there is a 'Repository URL' field containing the text 'https://github.com/shazforiot/MSBuild_firstproject.git'. Below the URL field is a 'Credentials' dropdown menu currently showing '- none -'. There is a '+ Add' button below the credentials dropdown. At the bottom of the 'Repositories' section, there is an 'Advanced' dropdown menu. At the very bottom of the window, there is an 'Add Repository' button.

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.

The screenshot shows the 'Configure' page with a sidebar on the left containing the following menu items: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment' (which is highlighted), 'Build Steps', and 'Post-build Actions'. A modal window titled 'Build Environment' is open, displaying a list of build steps. The list includes: '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', and 'SonarScanner for MSBuild - End Analysis'. At the bottom of the modal, there is an 'Add build step' button with an upward arrow.

on

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=Sonarqube-test  
sonar.sources=.  
sonar.host.url=http://localhost:9000  
sonar.login=admin  
sonar.password=
```

Additional arguments ?

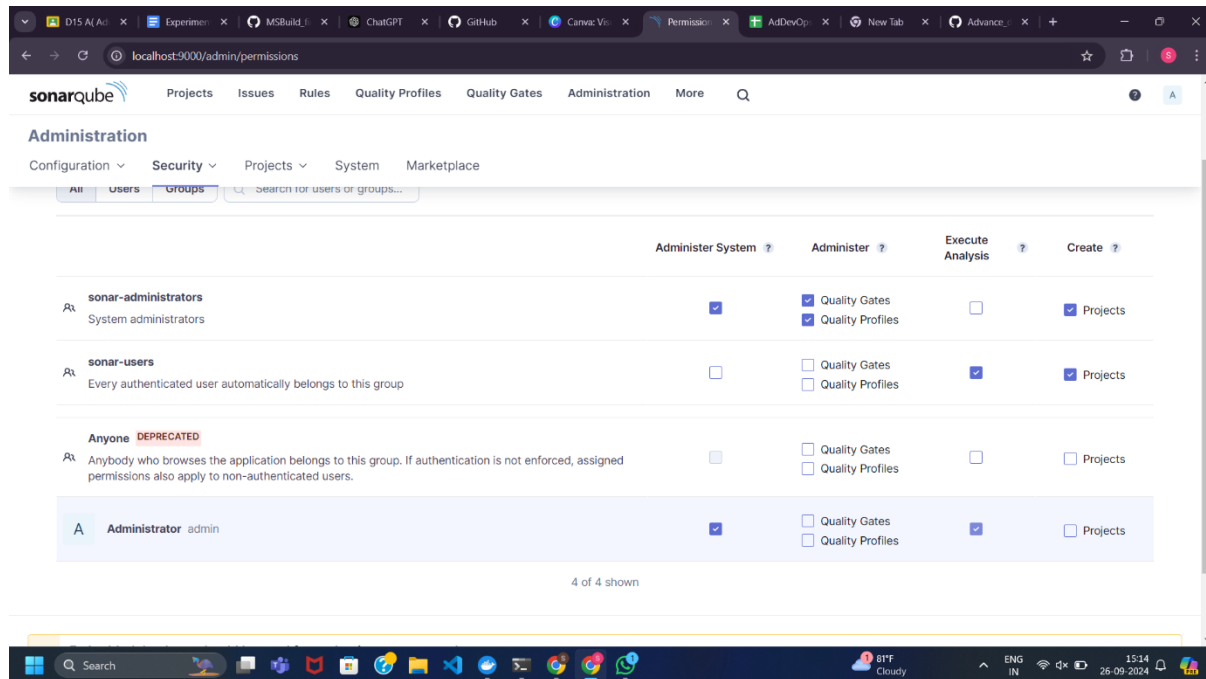
JVM Options ?

Add build step ▾

Save

Apply

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

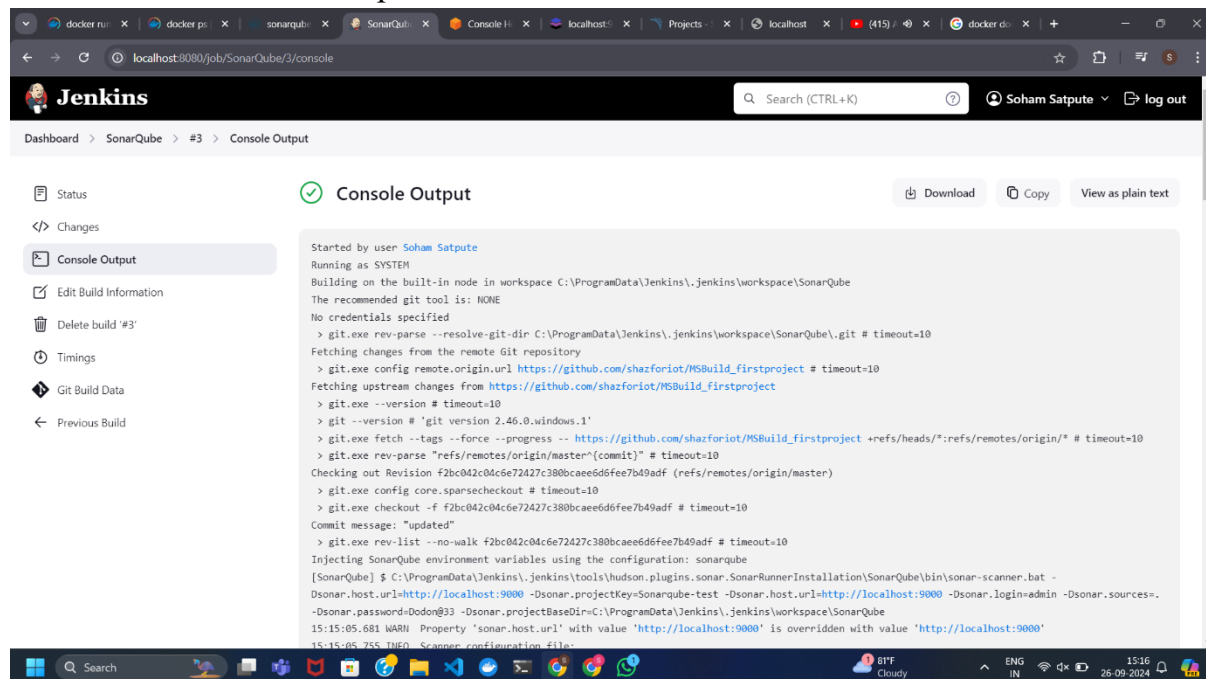


The screenshot shows the SonarQube Administration interface. The 'Groups' tab is selected, displaying a table of user groups and their permissions. The 'Administrator' group is highlighted, and its permissions are being configured.

Group	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 <small>DEPRECATED</small> 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 type="checkbox"/> Projects

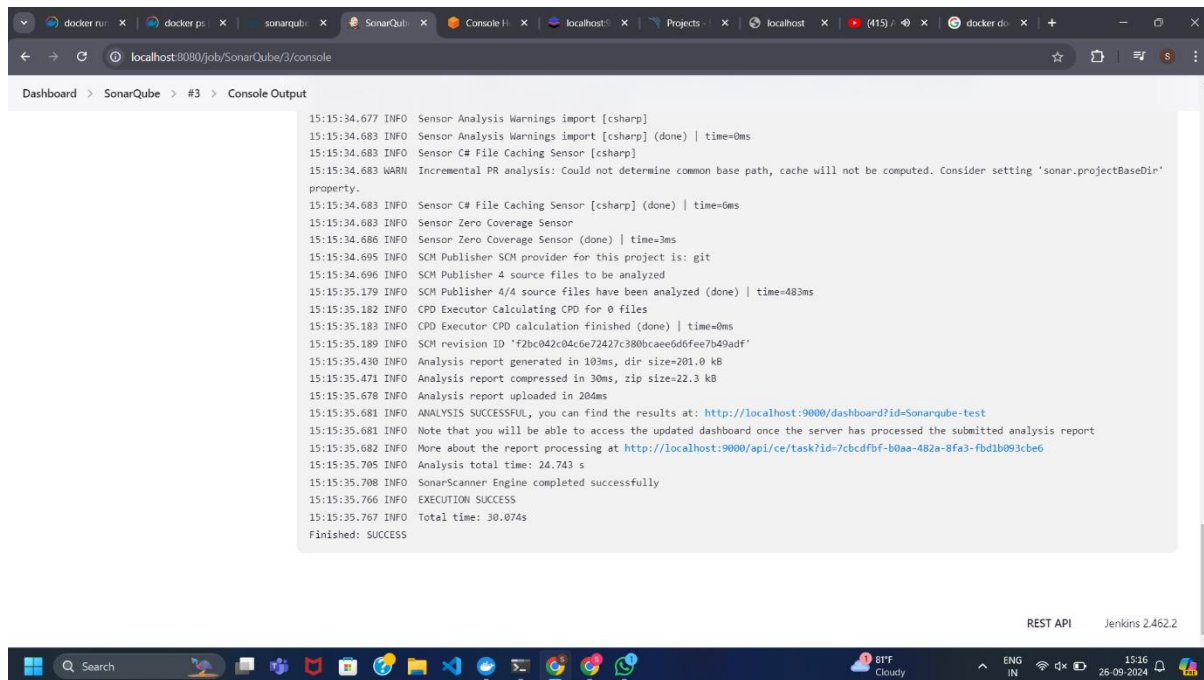
4 of 4 shown

12. Check the console Output

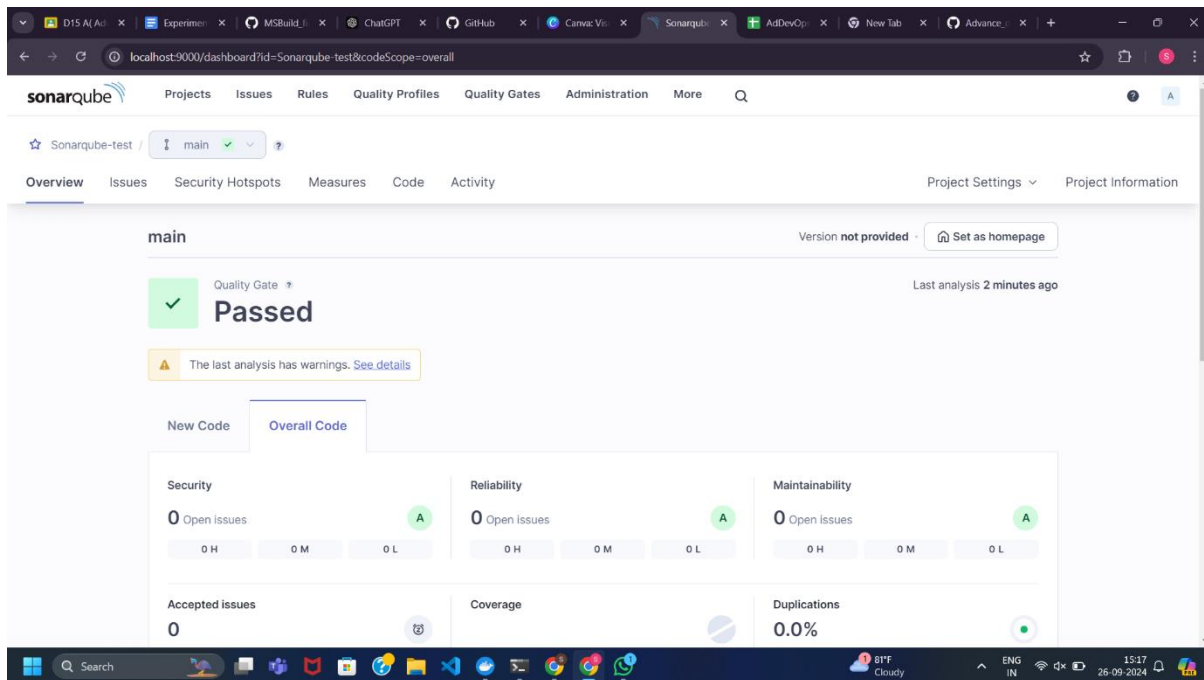


The screenshot shows the Jenkins Console Output page for the 'SonarQube' job. The build log is displayed, showing the execution of the SonarScanner CLI and the configuration of the SonarQube environment.

```
Started by user Soham Satpute
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
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\SonarQube\.git # timeout=10
Fetching changes from the remote Git repository
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject
> 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 +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
Injecting SonarQube environment variables using the configuration: sonarqube
[SonarQube] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\SonarQube\bin\sonar-scanner.bat -
-Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=Sonarqube-test -Dsonar.host.url=http://localhost:9000 -Dsonar.login=admin -Dsonar.sources=.
-Dsonar.password=Dodon@833 -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\SonarQube
15:15:05.681 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
15:15:05.755 INFO Scanner configuration file:
```



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



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