Adv. Devops Experiment no. 7

Name: Rohan Lalchandani
Class: D15A Roll no.: 25

Aim: To understand the Kubernetes Cluster Architecture, install and Spin Up a Kubernetes Cluster on Linux Machines/Cloud Platforms.

Theory:

Static application security testing (SAST), or static analysis, is a testing methodology that analyzes source code to find security vulnerabilities that make your organization's applications susceptible to attack. SAST scans an application before the code is compiled.
It's also known as white box testing.

Why SAST is Important in DevOps ?

Early Detection of Vulnerabilities: SAST helps identify vulnerabilities early in the development phase, allowing developers to fix issues before the code is deployed to production.

Shifts Security Left: Incorporating SAST in DevOps practices supports the "Shift Left" approach in security, where testing starts earlier in the pipeline rather than waiting until the final stages.

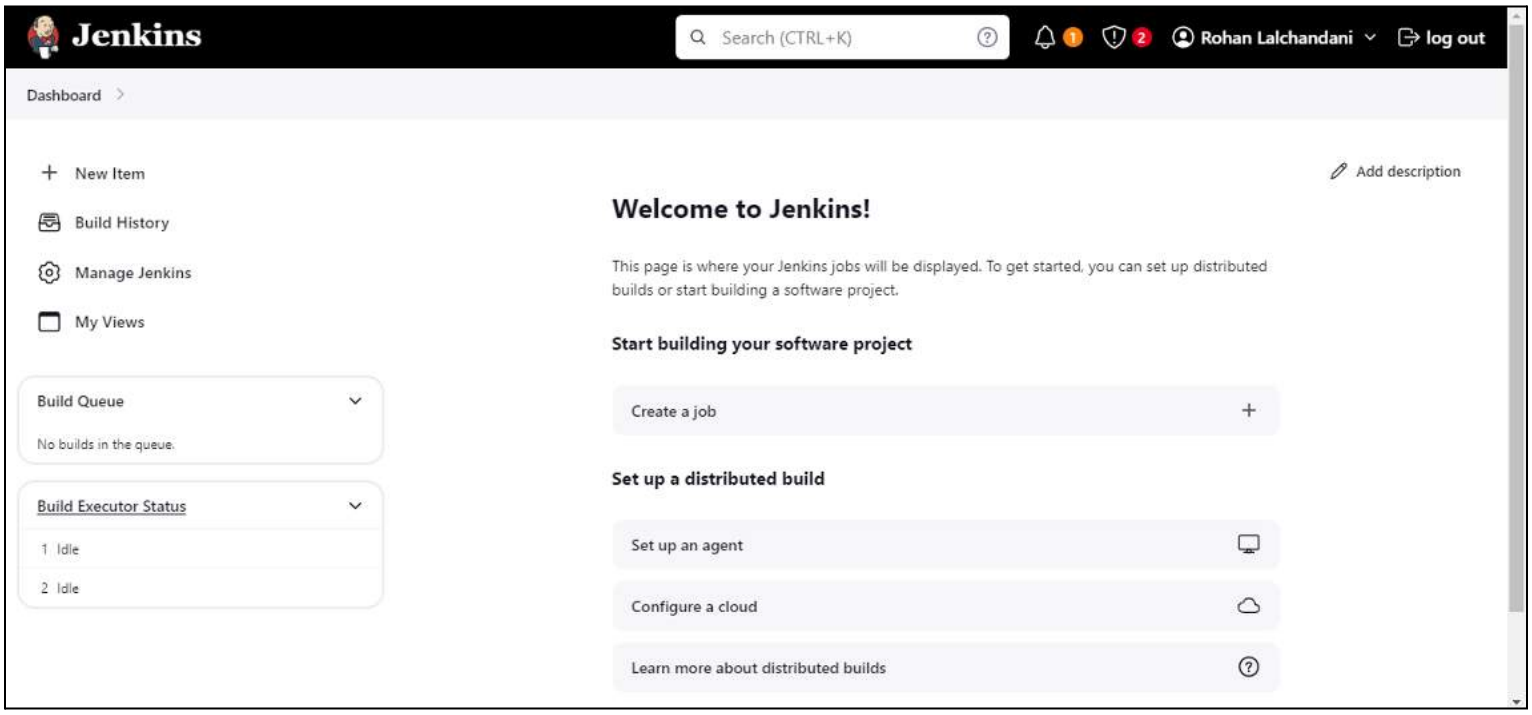
Automation-Friendly: SAST tools can be integrated with CI/CD pipelines, ensuring automated security checks on every code commit or pull request.

SAST in DevOps Workflow Example

1. **Code Commit**: Developers push code changes to a version control system like Git.
2. **Automated Build**: The CI system (e.g., Jenkins) automatically triggers a build and kicks off the SAST scan as part of the pipeline.
3. **SAST Scan**: The SAST tool scans the code and reports vulnerabilities if present. Some popular SAST tools for DevOps include:
4. **Build Failures or Warnings**: If critical vulnerabilities are found, the build fails or issues warnings, depending on the security policies in place.
5. **Developer Feedback**: Developers receive feedback, either through IDE plugins, the CI/CD dashboard, or via notifications, so they can address the issues quickly.
6. **Security Approval**: Once all critical vulnerabilities are resolved, the build proceeds, and the application can move to the next stage of deployment.

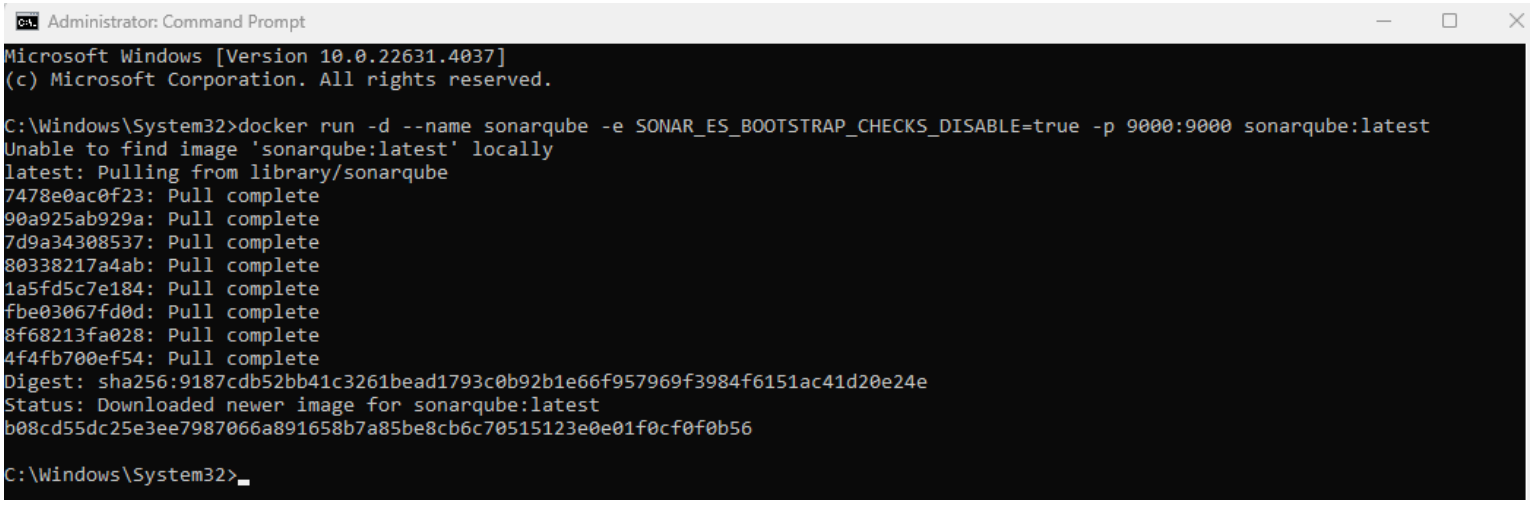
Implementation:

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

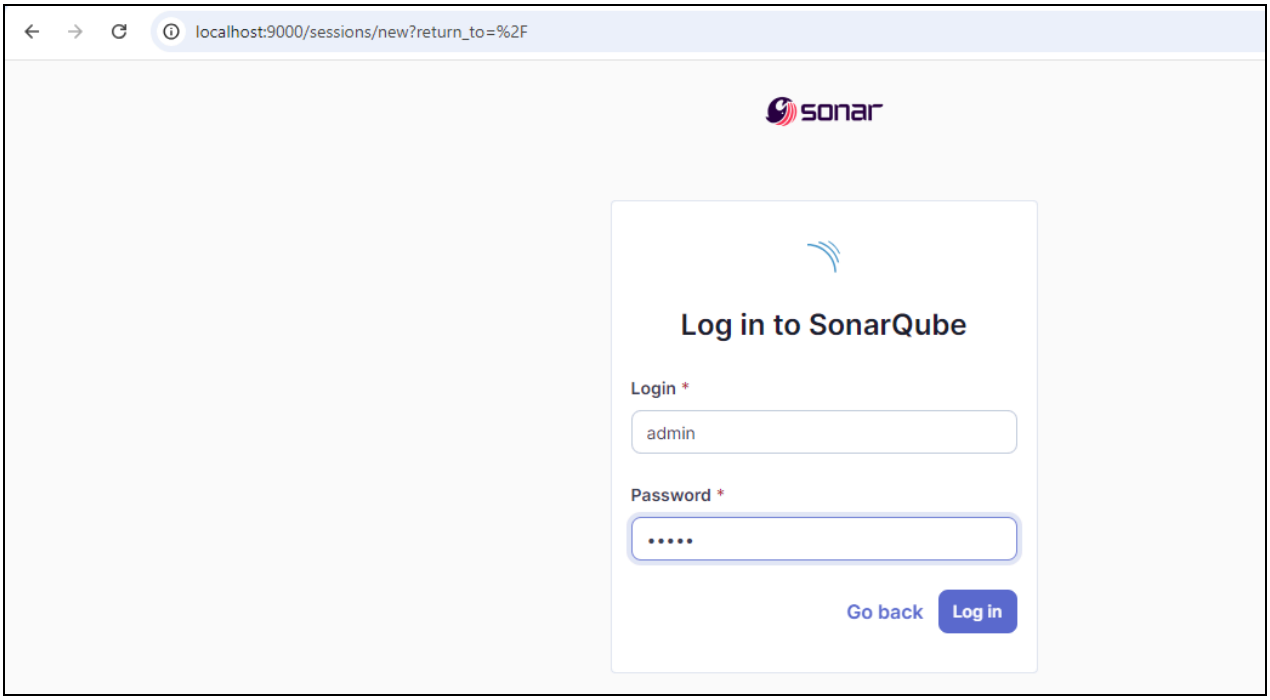


Step 2: Run Sonarqube’s image using the following command.

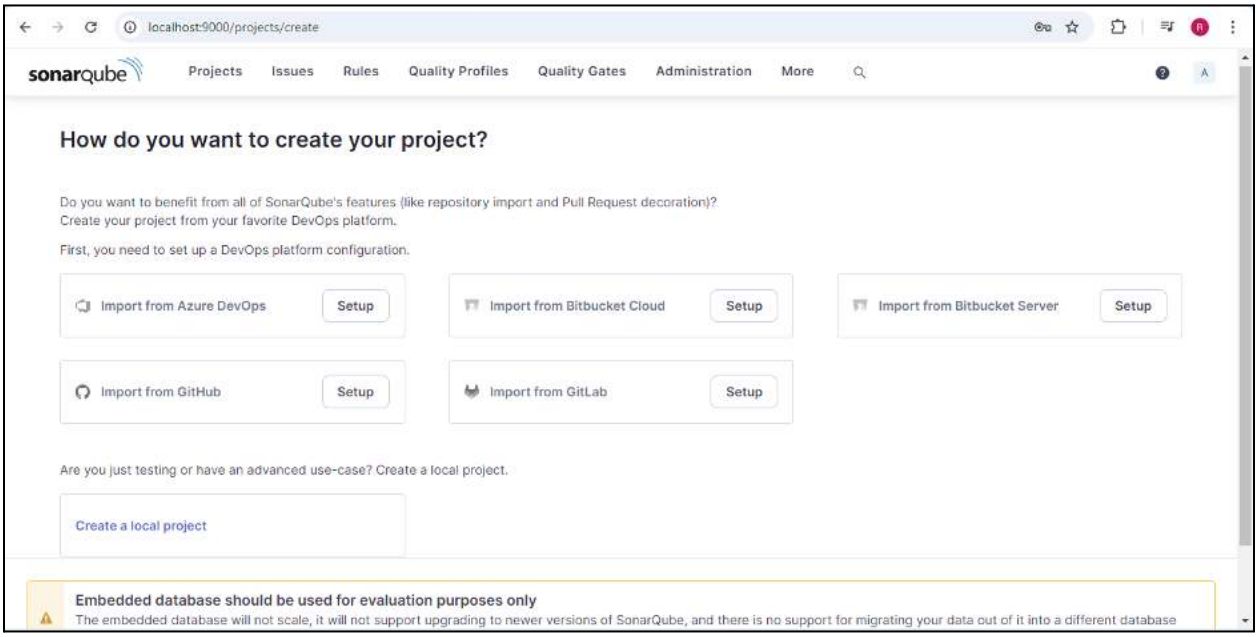
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest



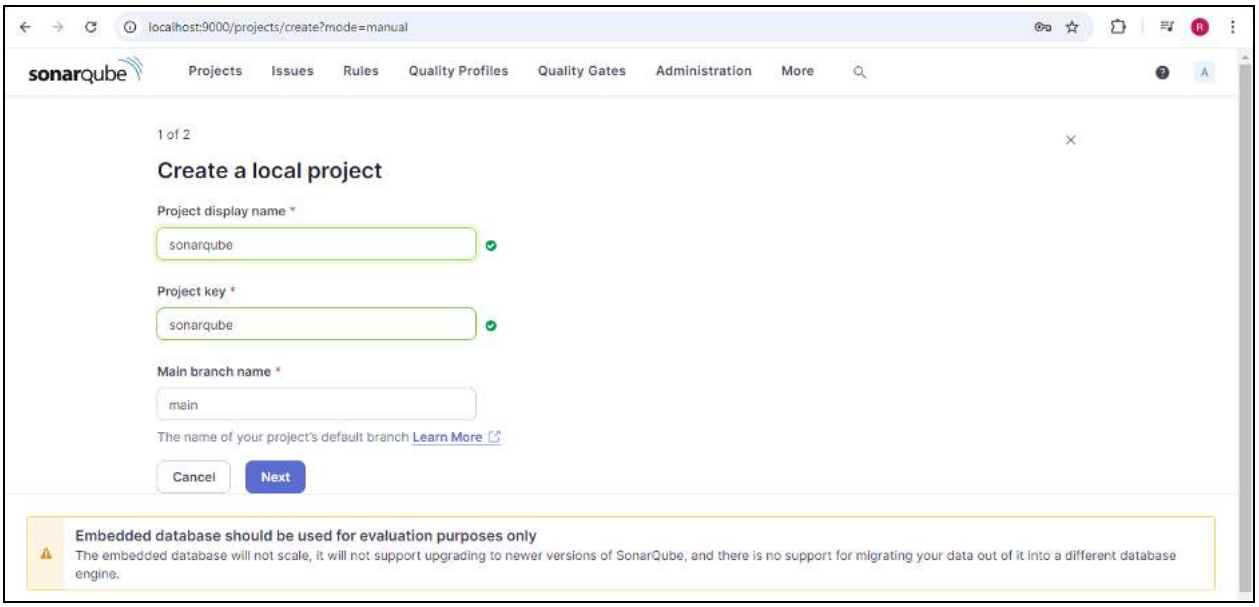
Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



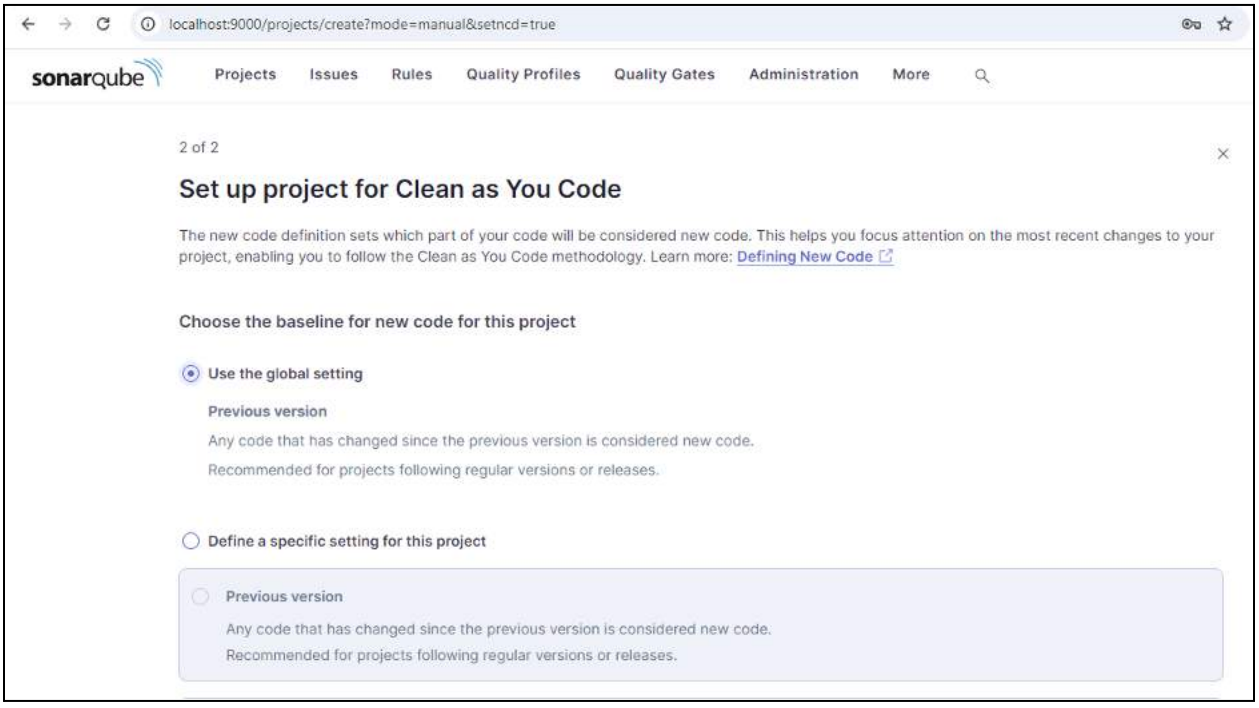
4.Login to SonarQube using username admin and password admin.



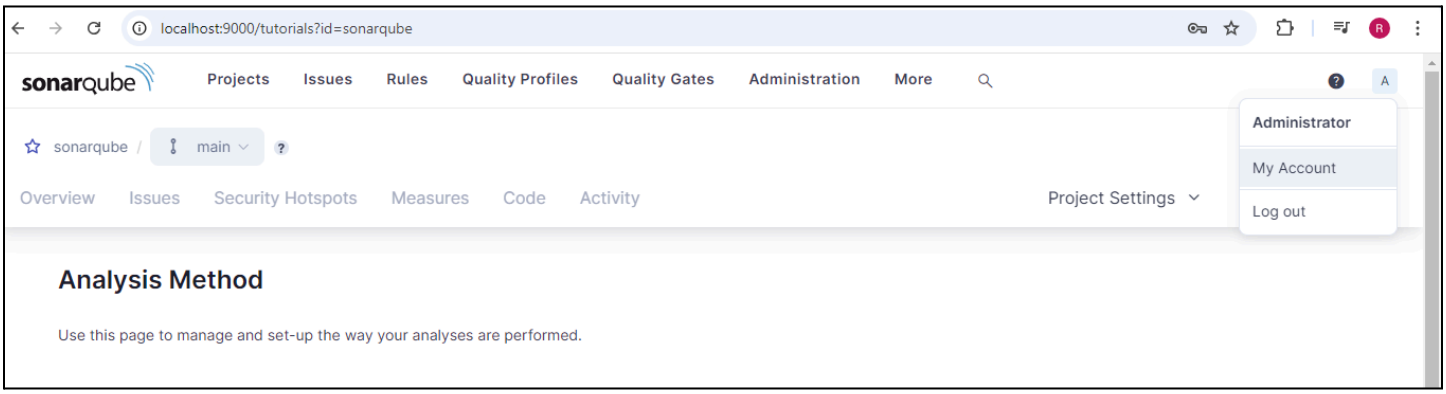
5.Create a manual project in SonarQube with the name sonarqube



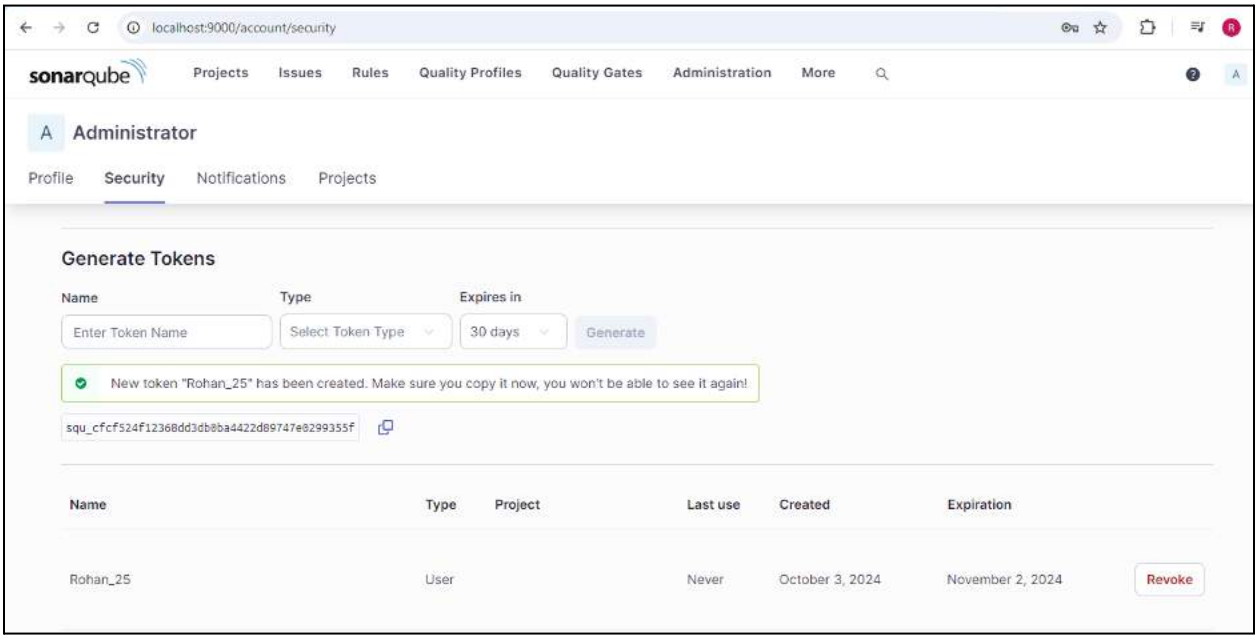
Setup the project by selecting the following options.



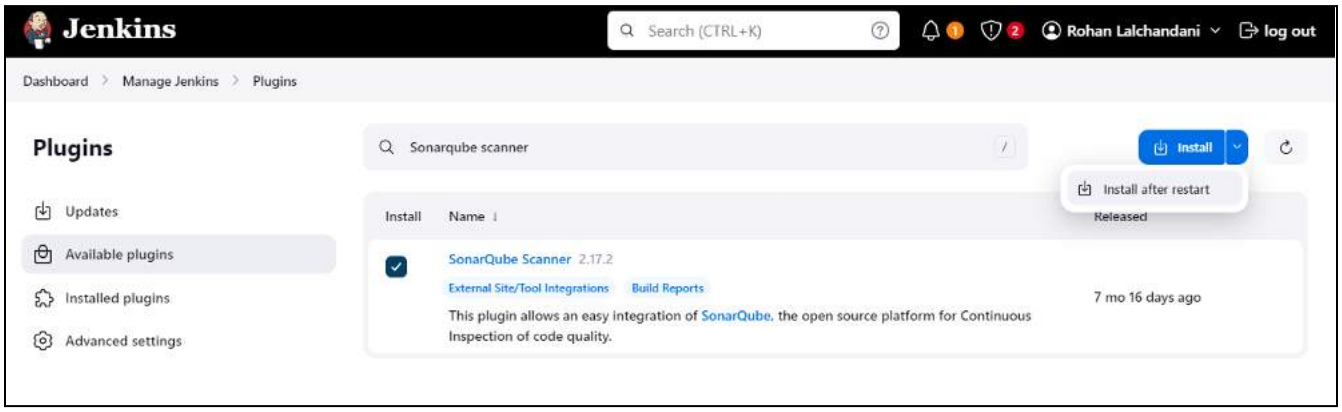
Now, for setting up sonarqube in Jenkins we need to have the login and password, for that we need to generate token.
Go to My Account > Security

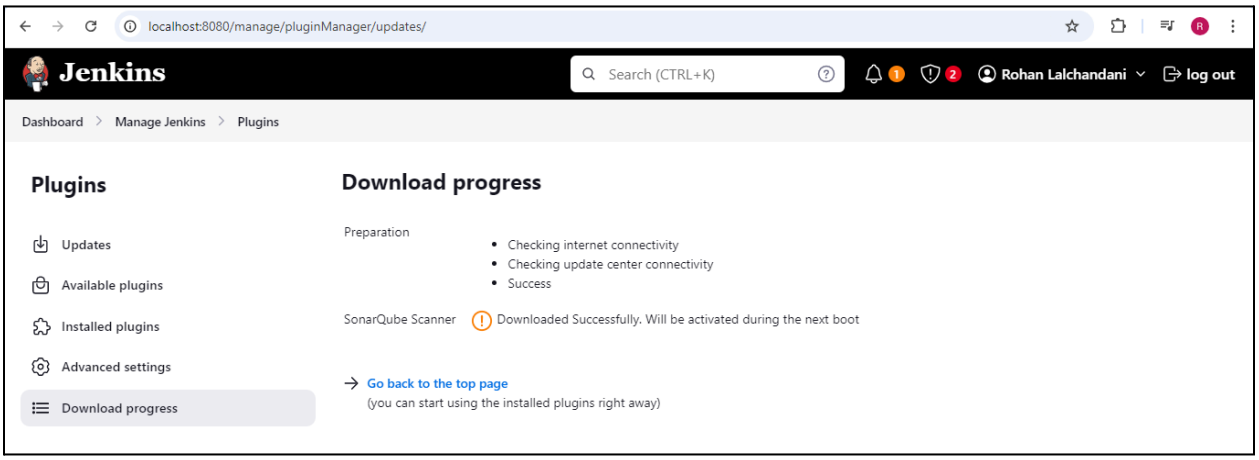


Type the details and click on generate token, copy the token and save it in a notepad.

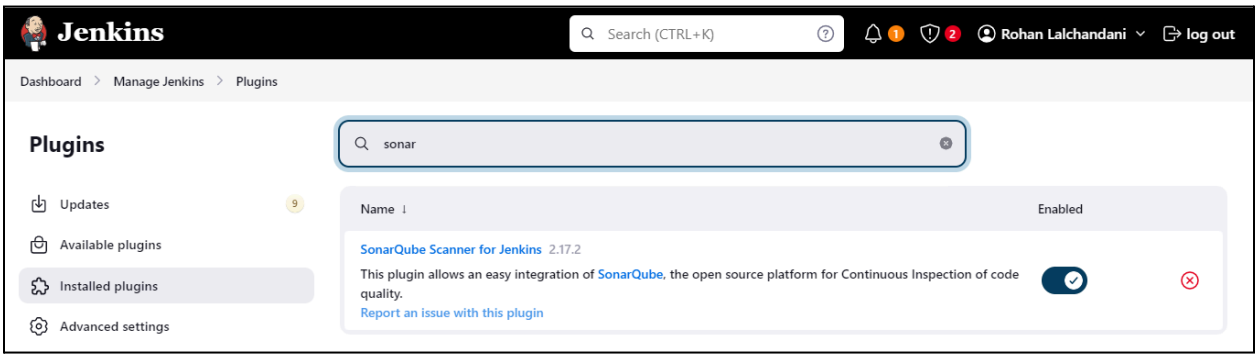


Setup the project and come back to Jenkins Dashboard. Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

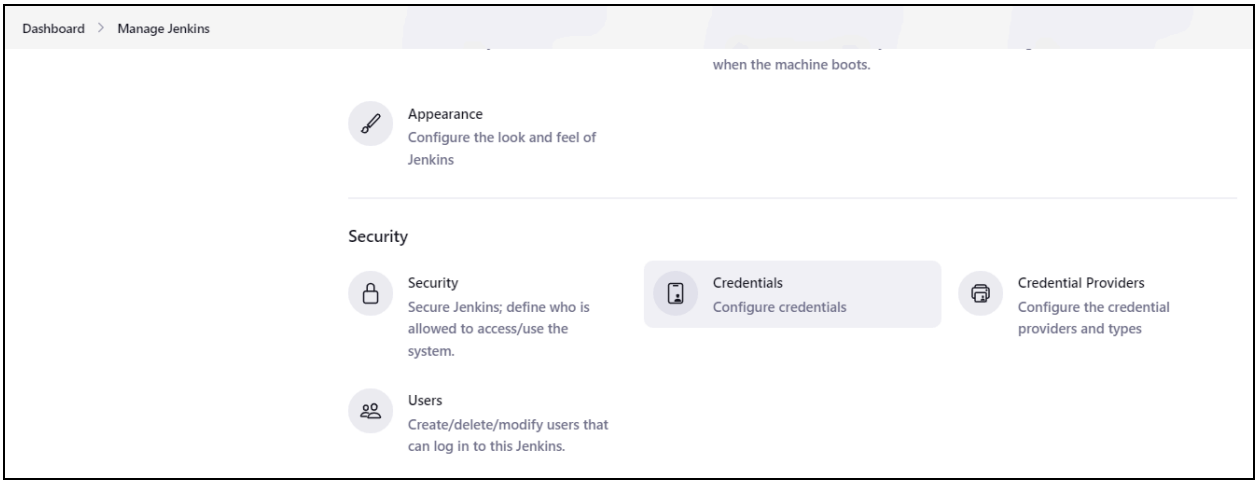




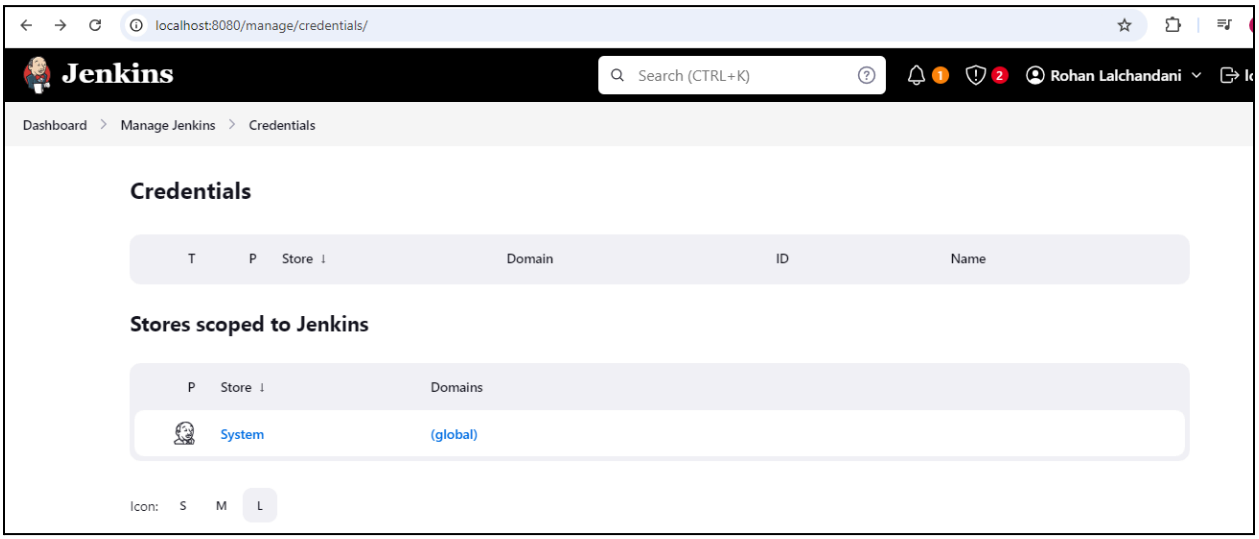
Go to Installed plugins to see if sonarqube scanner is successfully installed



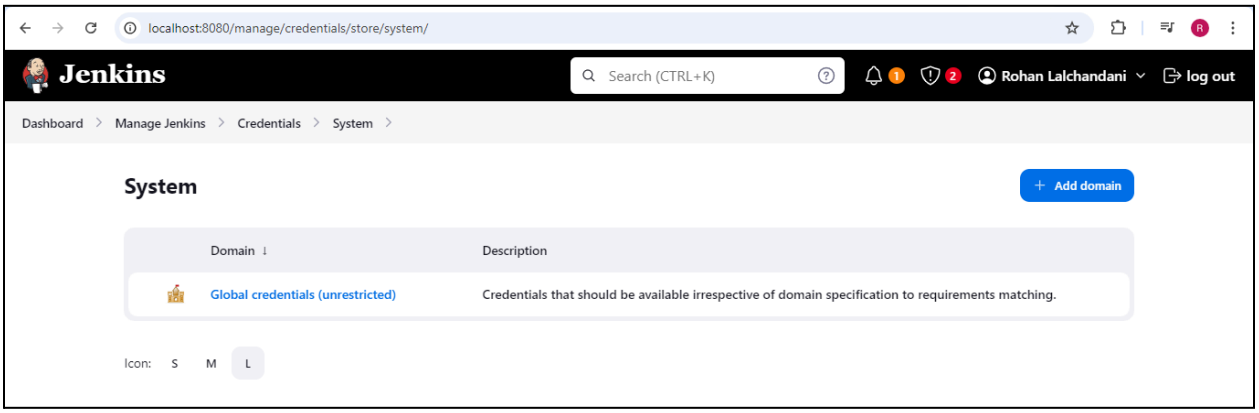
Go to Manage Jenkins > Credentials



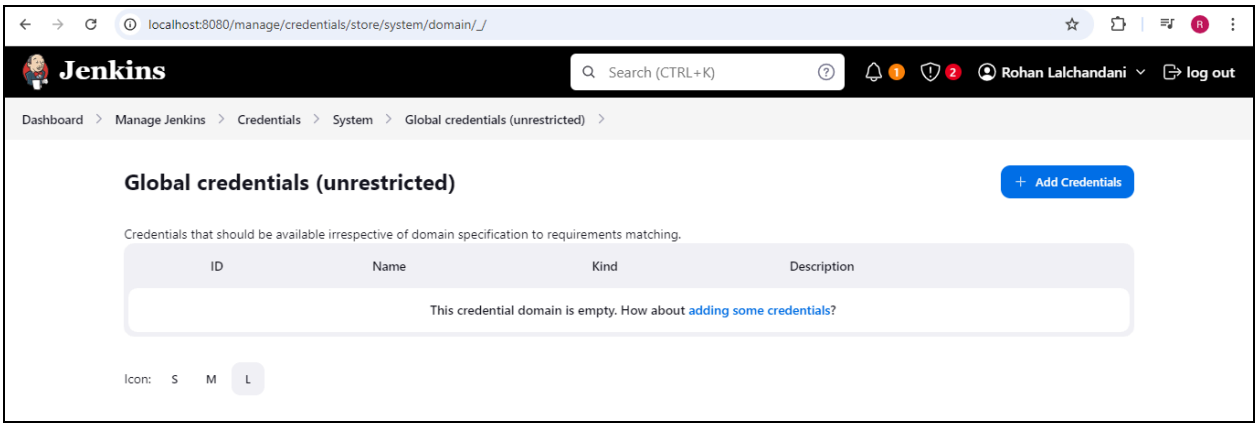
Click on System



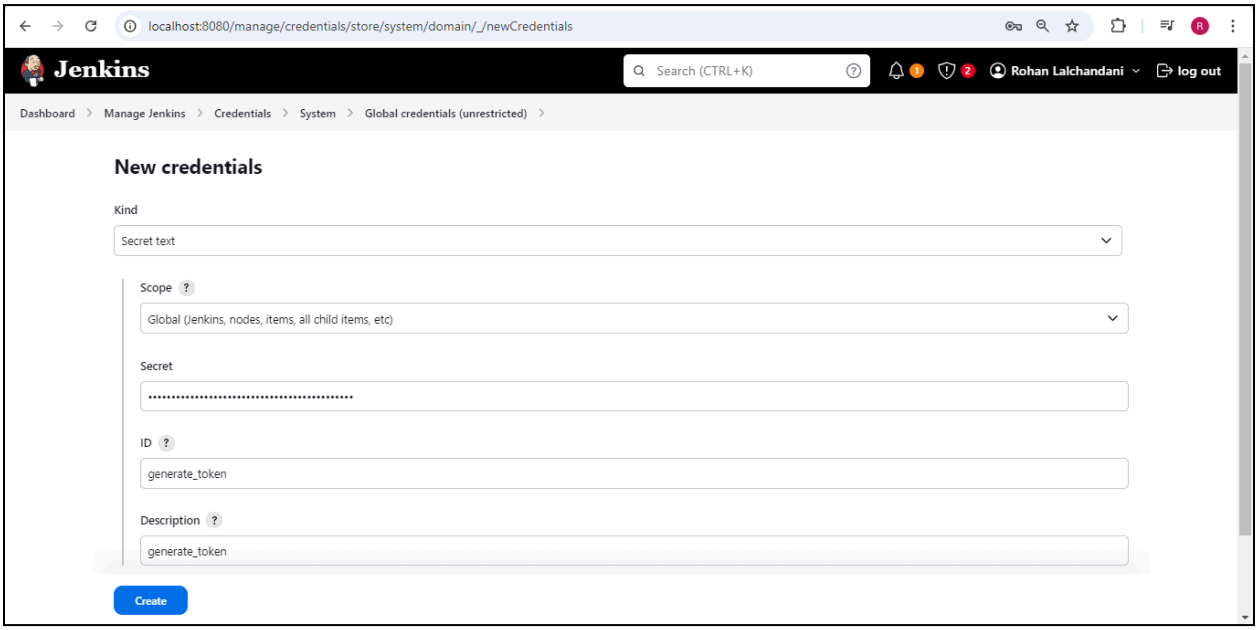
Click on Global Credentials



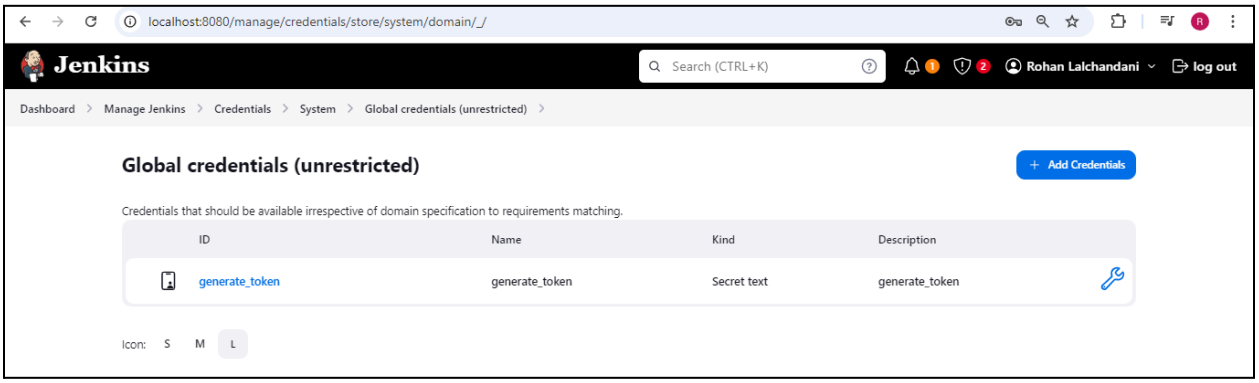
Click on Add Credentials



Now paste the generated token from sonarqube in the secret box and fill in other details as shown below.

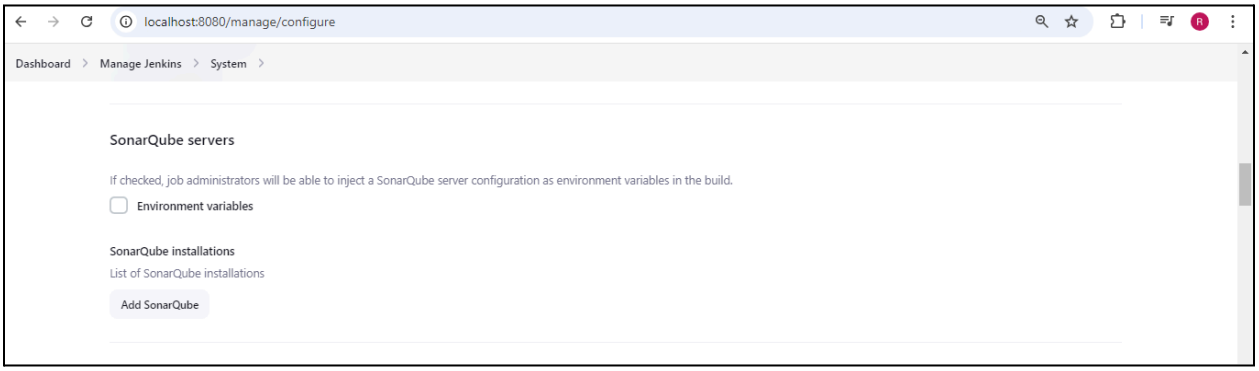


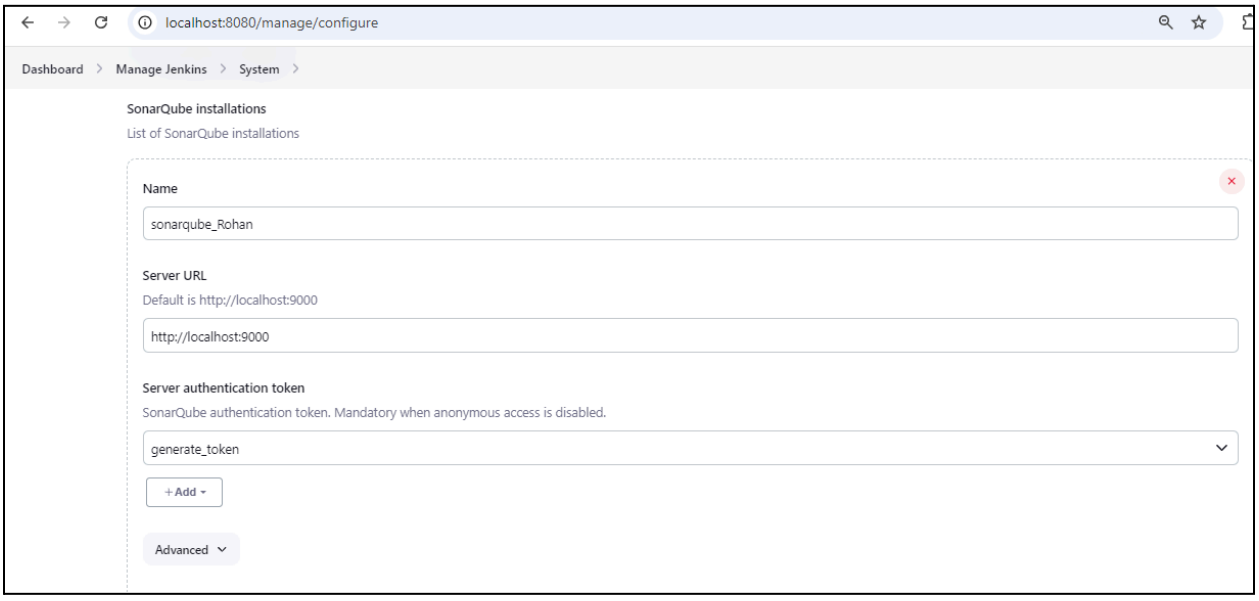
Token has been added successfully to the Jenkins



Configure SonarQube Scanner in Jenkins

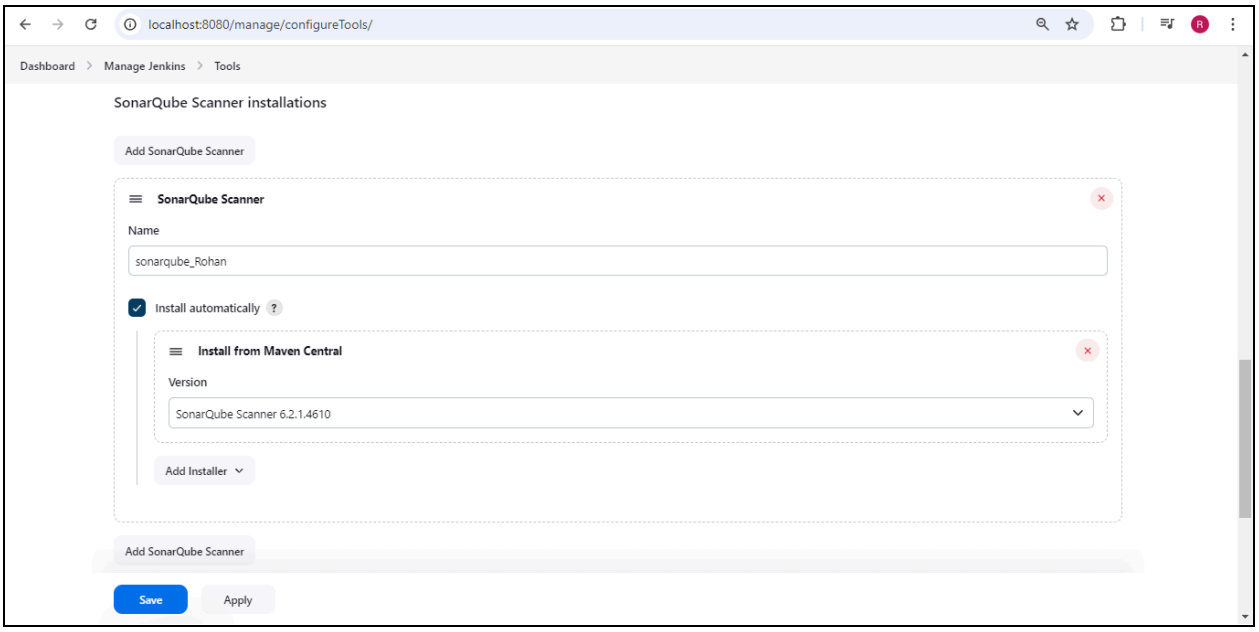
● Go to Manage Jenkins > System. Scroll down to Sonarqube servers, check on environment variables.
● Select he token from drop down menu.
● Click on save.



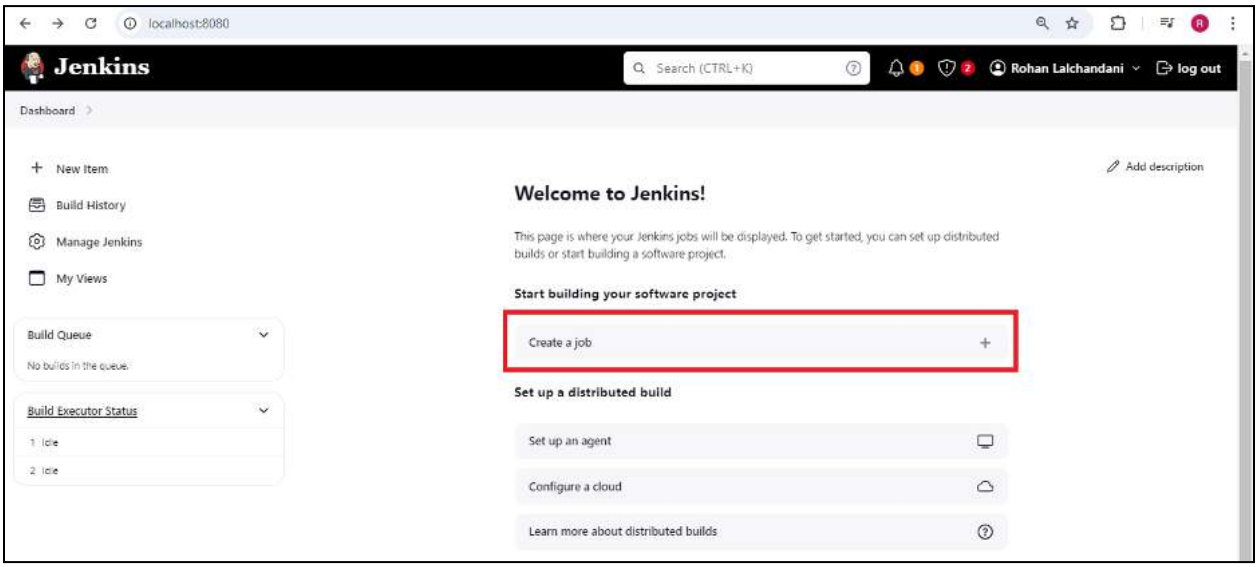


Configure SonarQube Scanner in Jenkins

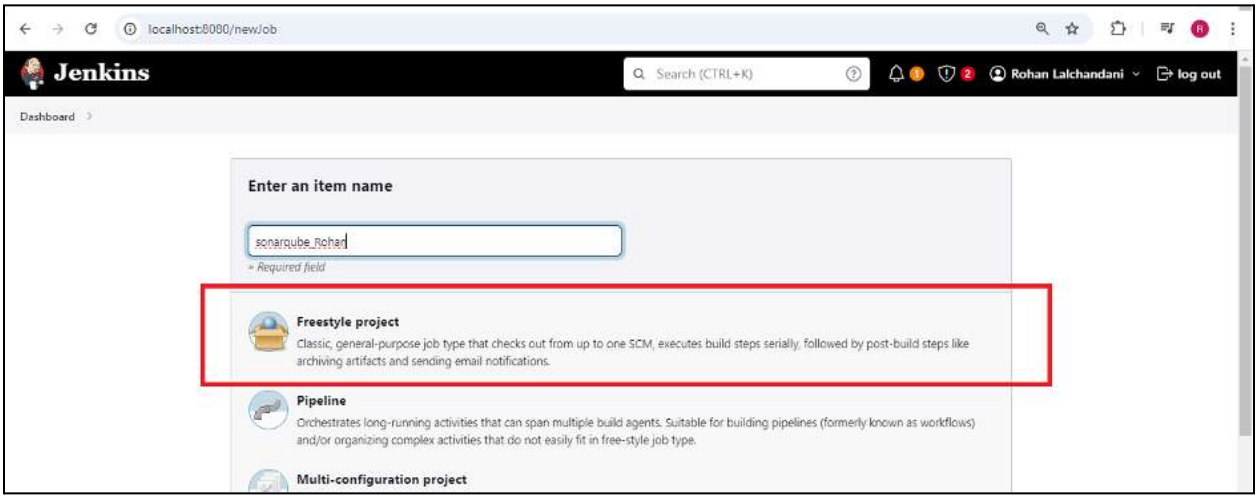
● Go to Manage Jenkins > Global Tool Configuration.
● Scroll down to SonarQube Scanner.
● Choose the latest version and select Install automatically



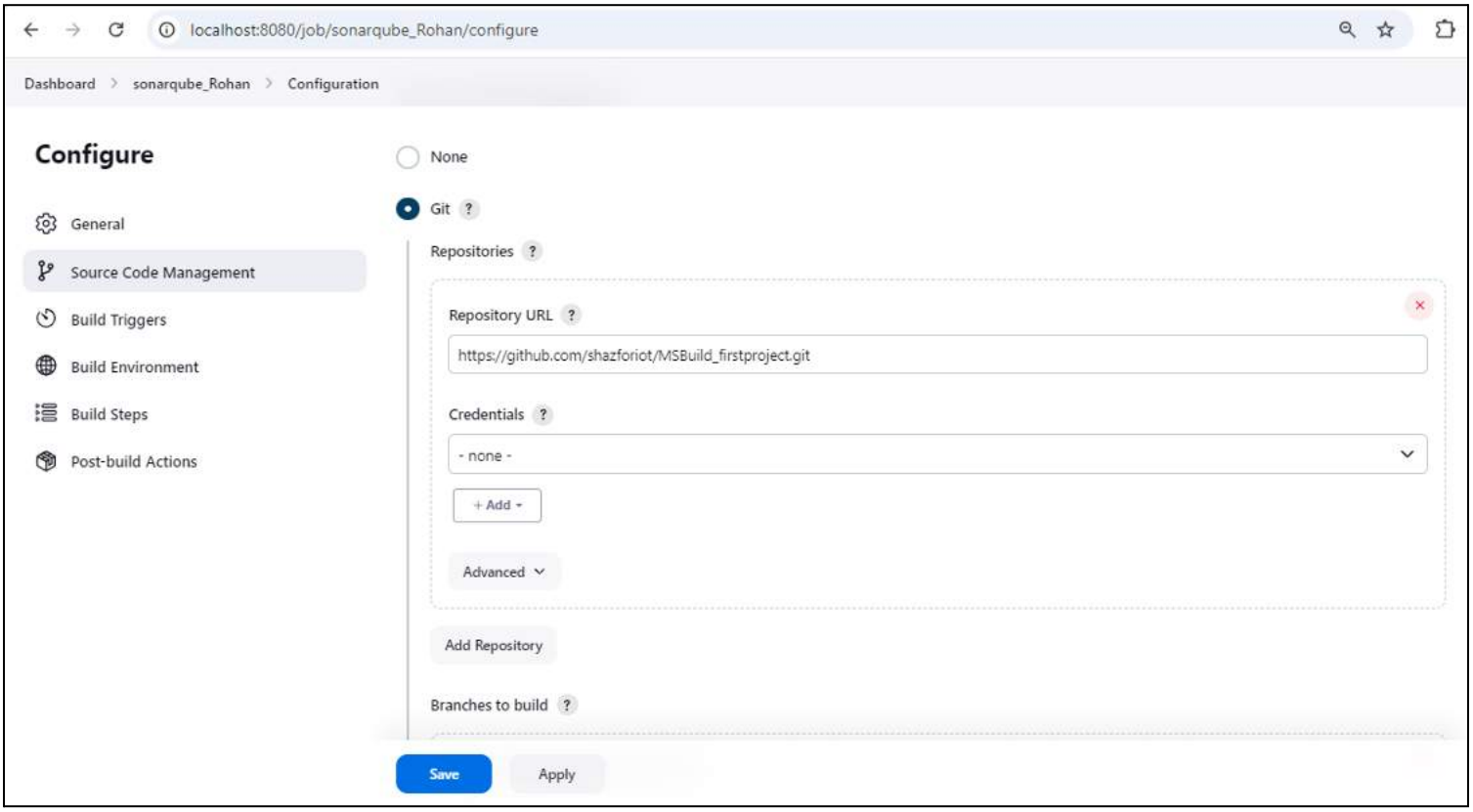
Go to your Jenkins Dashboard and Create a new Job



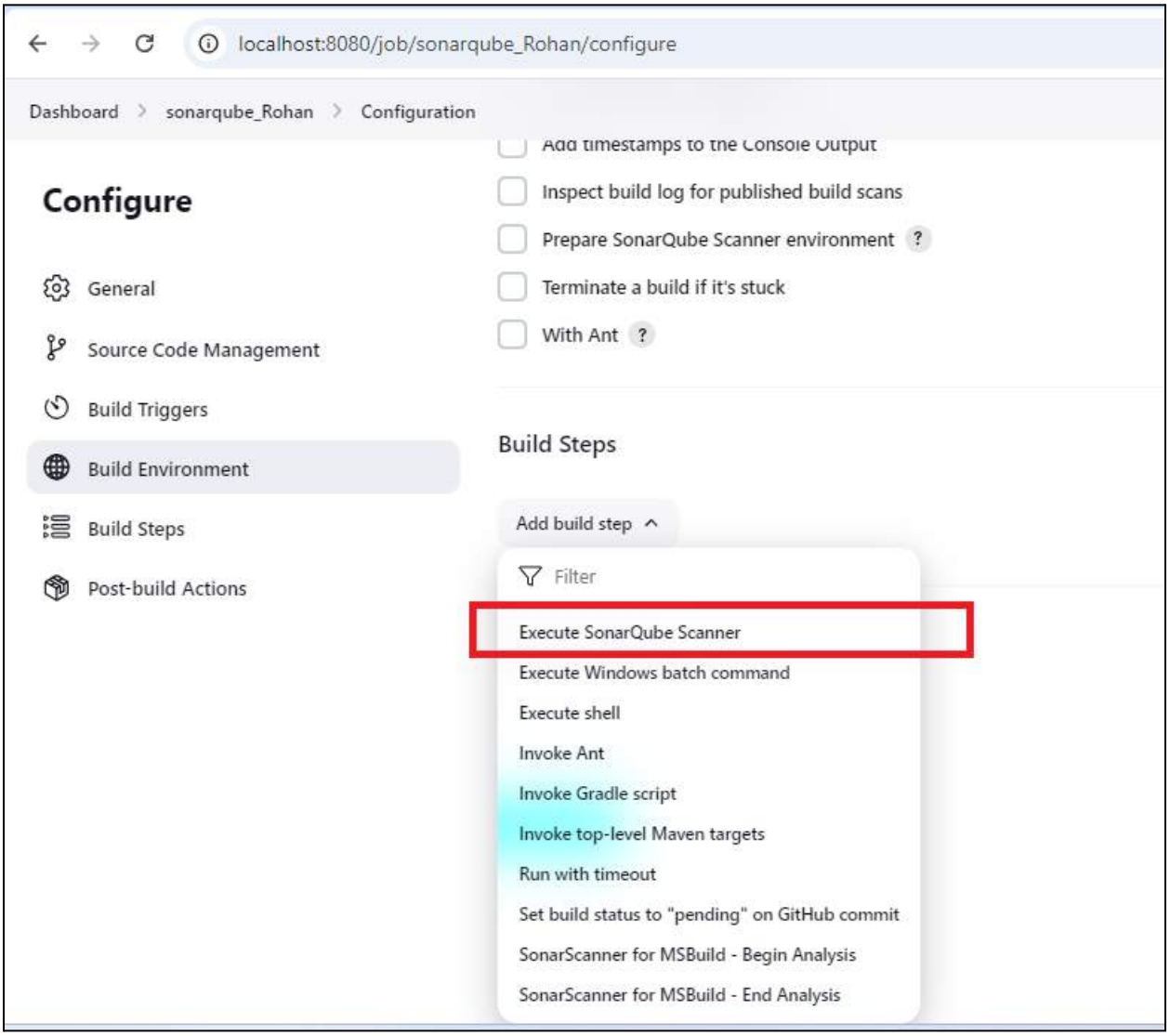
Enter name of the project and select freestyle project



Under Configure put the github link in source code management
https://github.com/shazforiot/MSBuild_firstproject.git

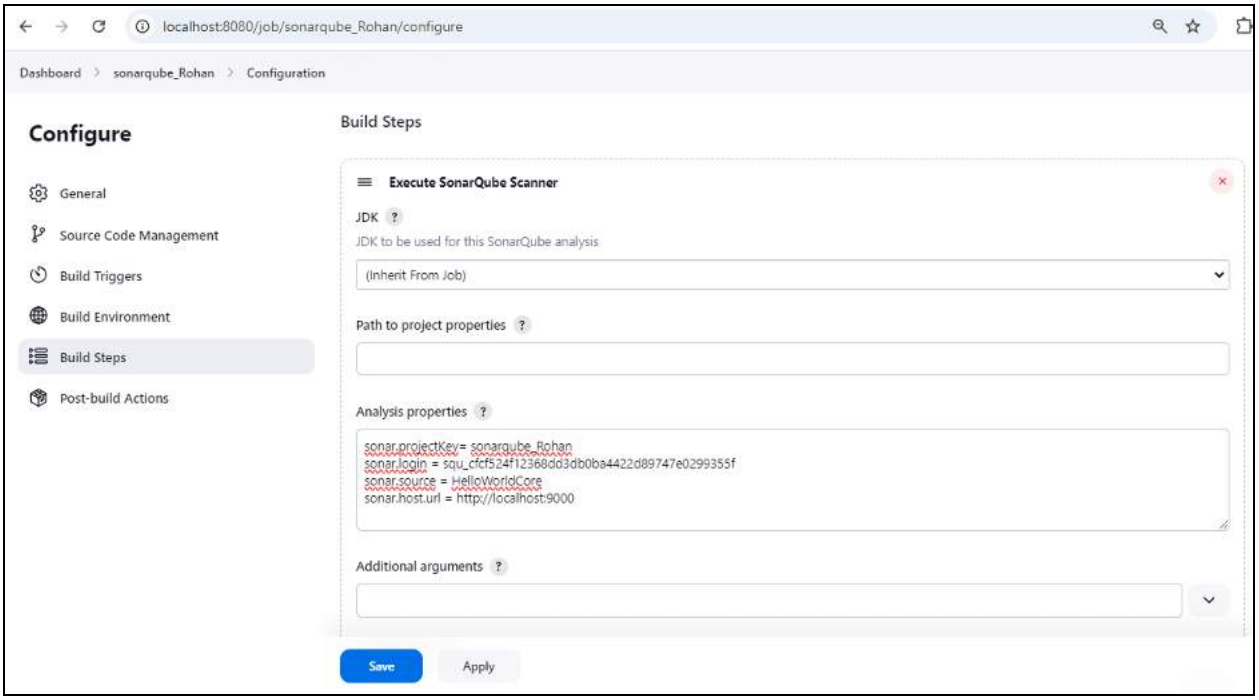


In build steps Select “Execute SonarQube scanner”

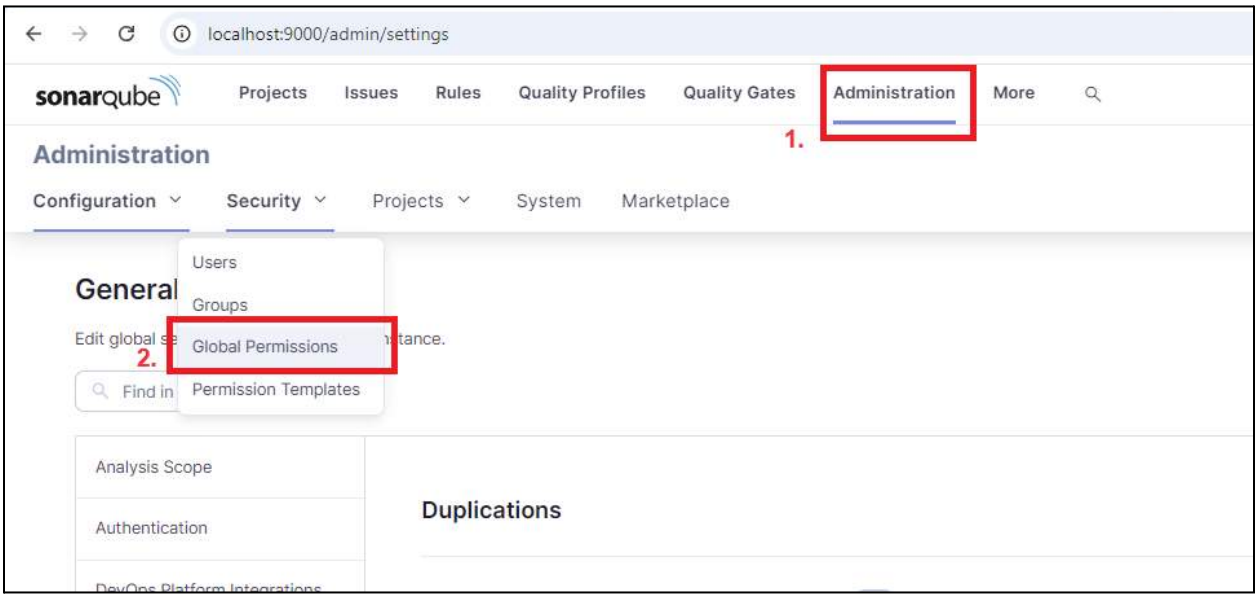


Type the following in analysis properties:

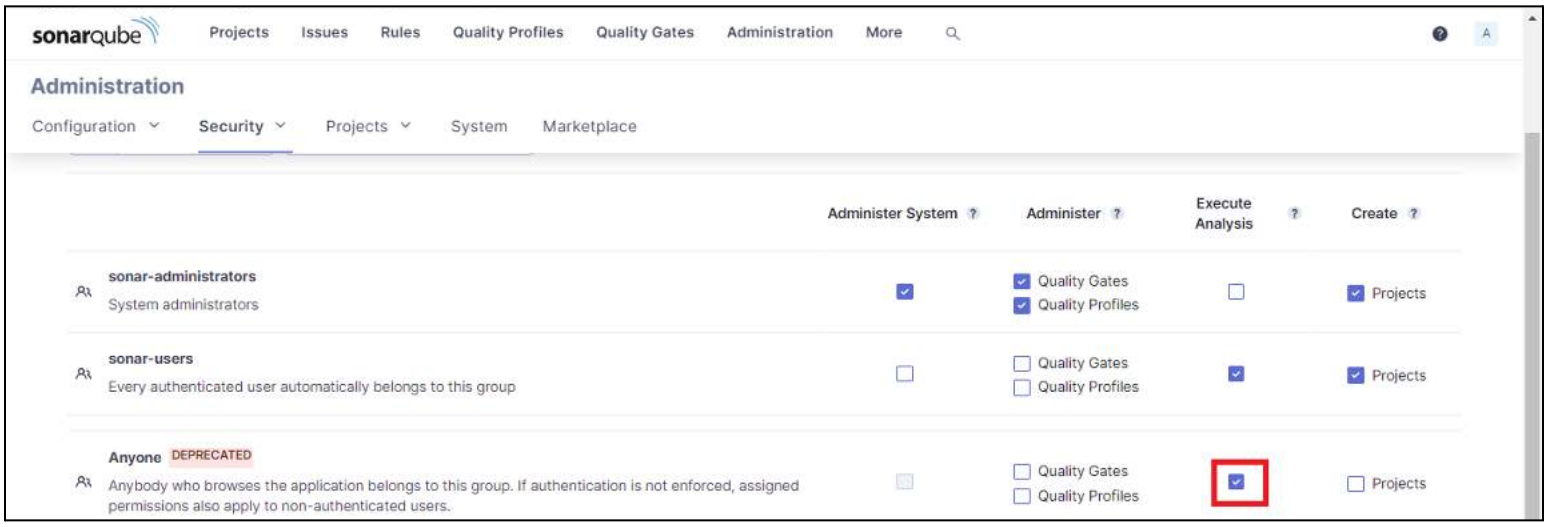
- sonar.projectKey=my_project_name
- sonar.login=your_generated_token
- sonar.sources=HelloWorldCore
- sonar.host.url=http://localhost:9000



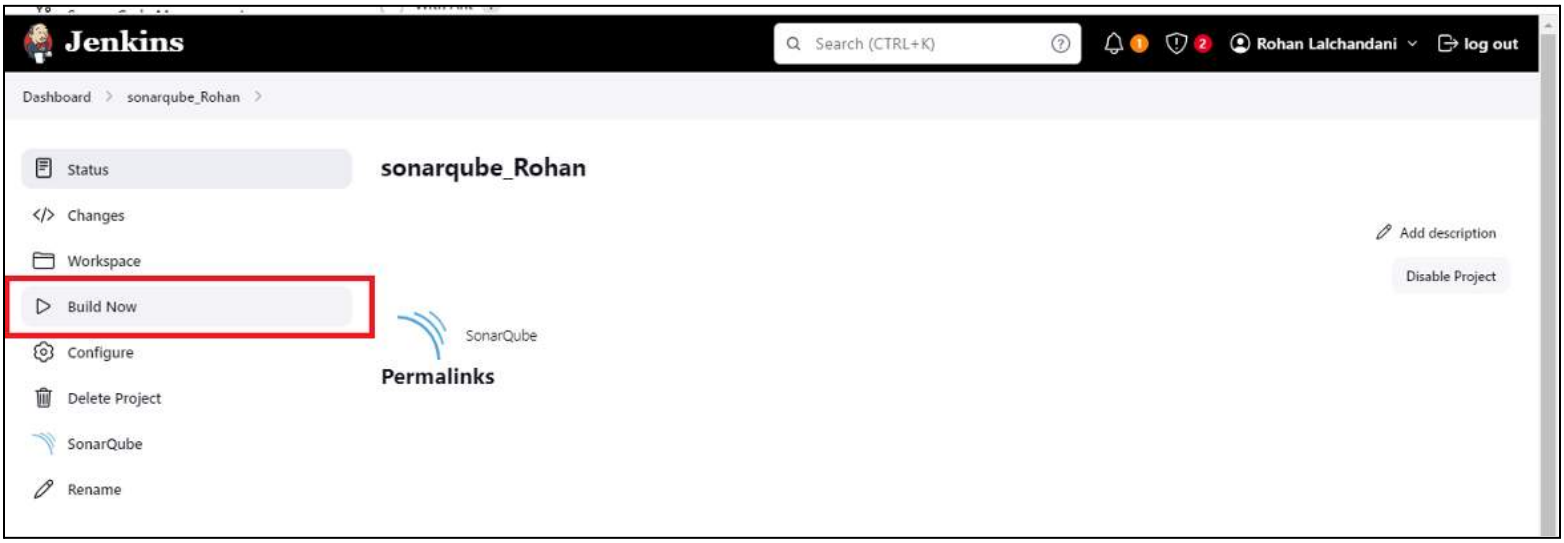
Now we need to enable access to sonarqube.
For that go to the sonarqube website
click on administration and then global permissions.



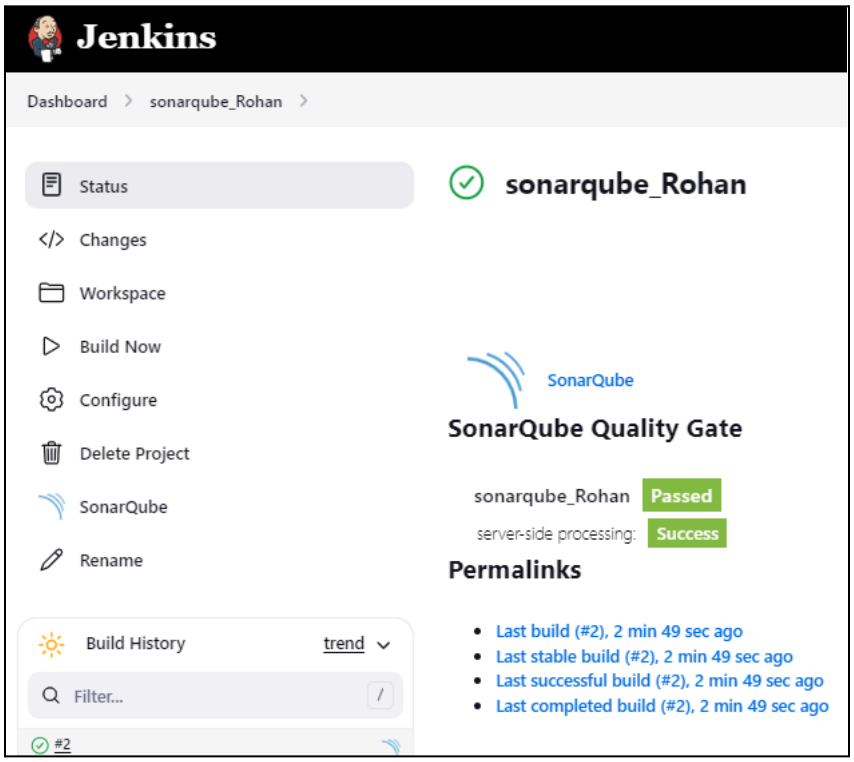
Click on Anyone in the checkbox as shown below.



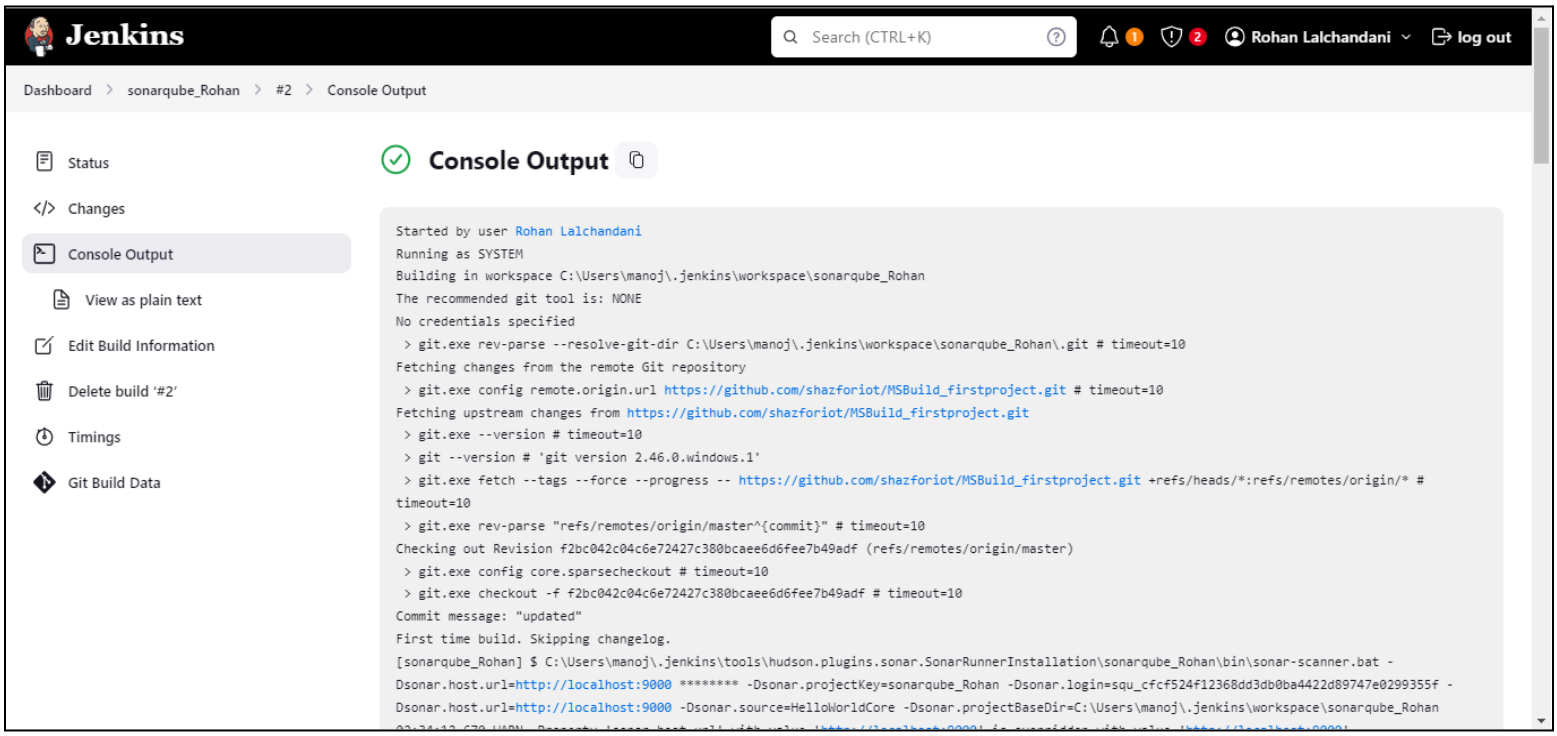
Go to the Jenkins Dashboard and click on “Build Now”



The build has been completed successfully.



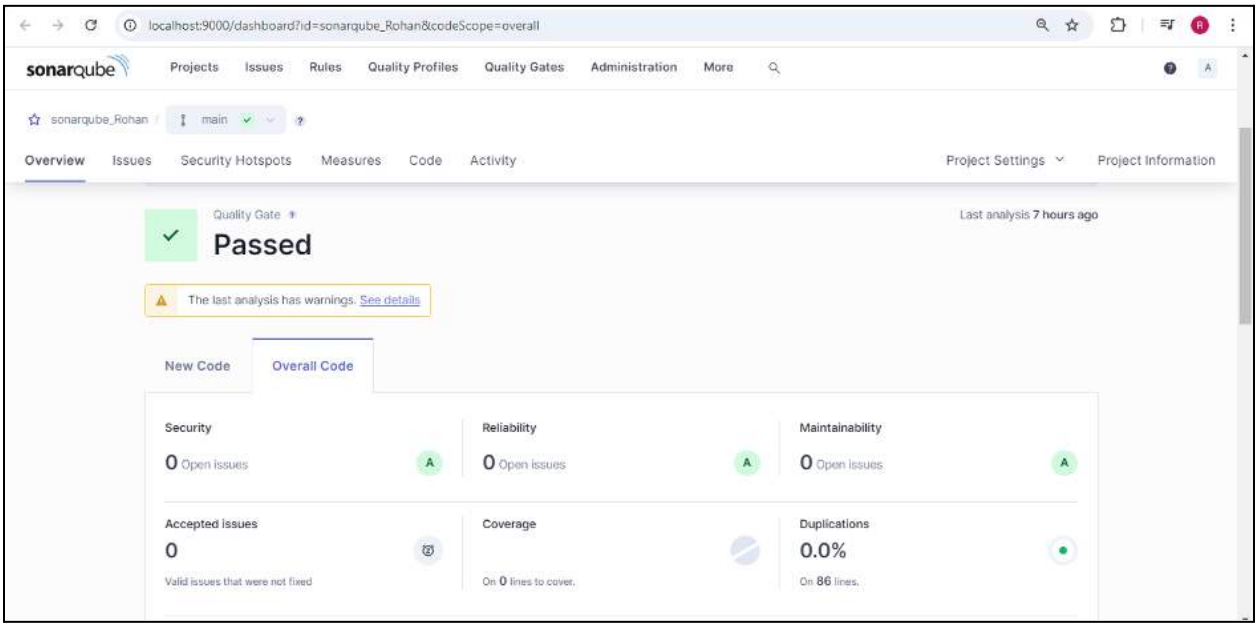
Go to the console output.



Scroll down click on the link as shown below to view the SonarQube analysis report.

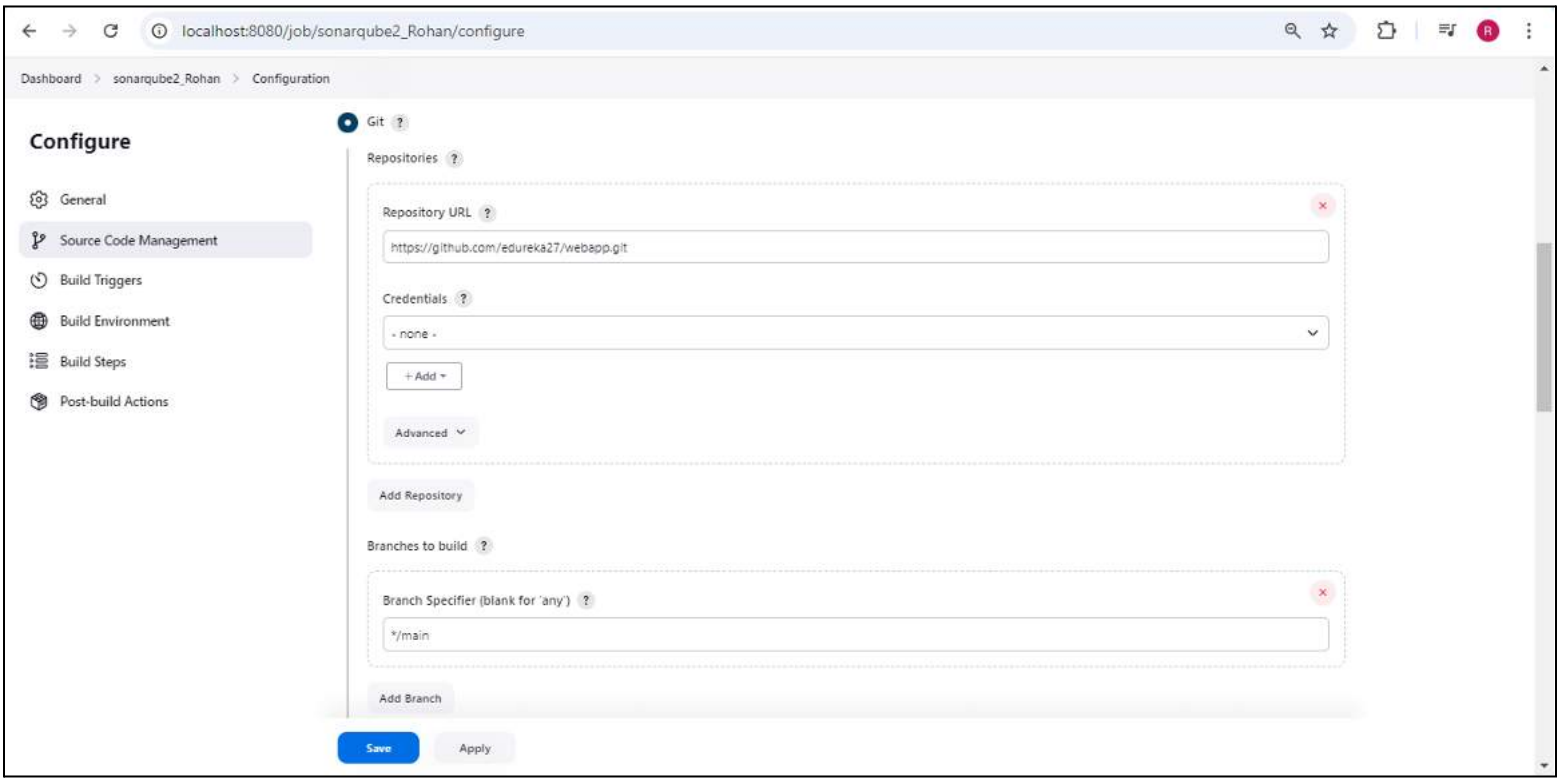
```
02:34:34.415 INFO  Analysis report compressed in 44ms, zip size=23.8 kB
02:34:34.448 INFO  Analysis report uploaded in 31ms
02:34:34.451 INFO  ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=sonarqube_Rohan
02:34:34.451 INFO  Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
02:34:34.451 INFO  More about the report processing at http://localhost:9000/api/ce/task?id=91e88697-16f7-4df8-9bcc-69a576460a6c
02:34:34.468 INFO  Analysis total time: 18.278 s
02:34:34.471 INFO  SonarScanner Engine completed successfully
02:34:34.558 INFO  EXECUTION SUCCESS
02:34:34.559 INFO  Total time: 21.879s
Finished: SUCCESS
```

You can see the code has successfully passed.



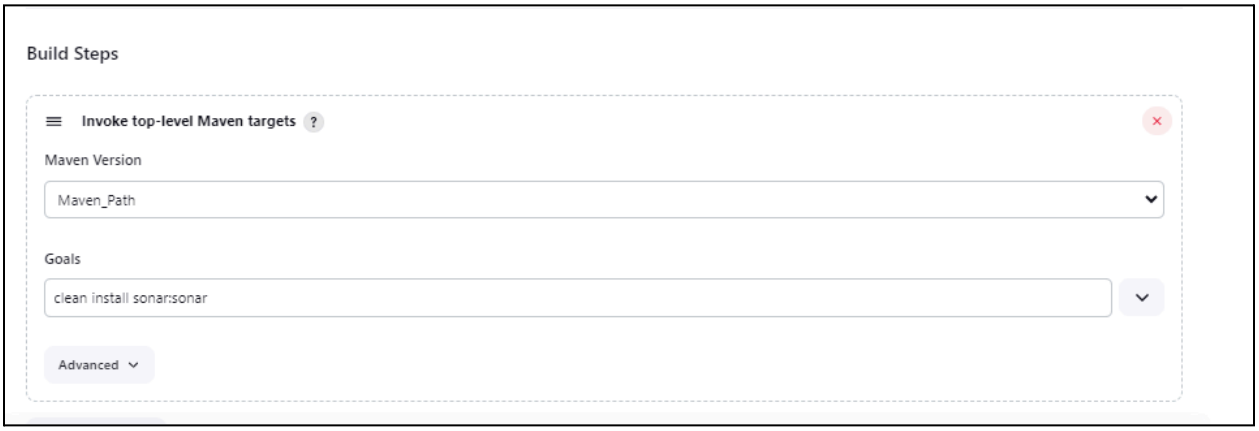
(EXTRA) : Building and Analysis of Maven project.

1. Create a maven project and paste the github link in source code management
https://github.com/edureka27/webapp.git

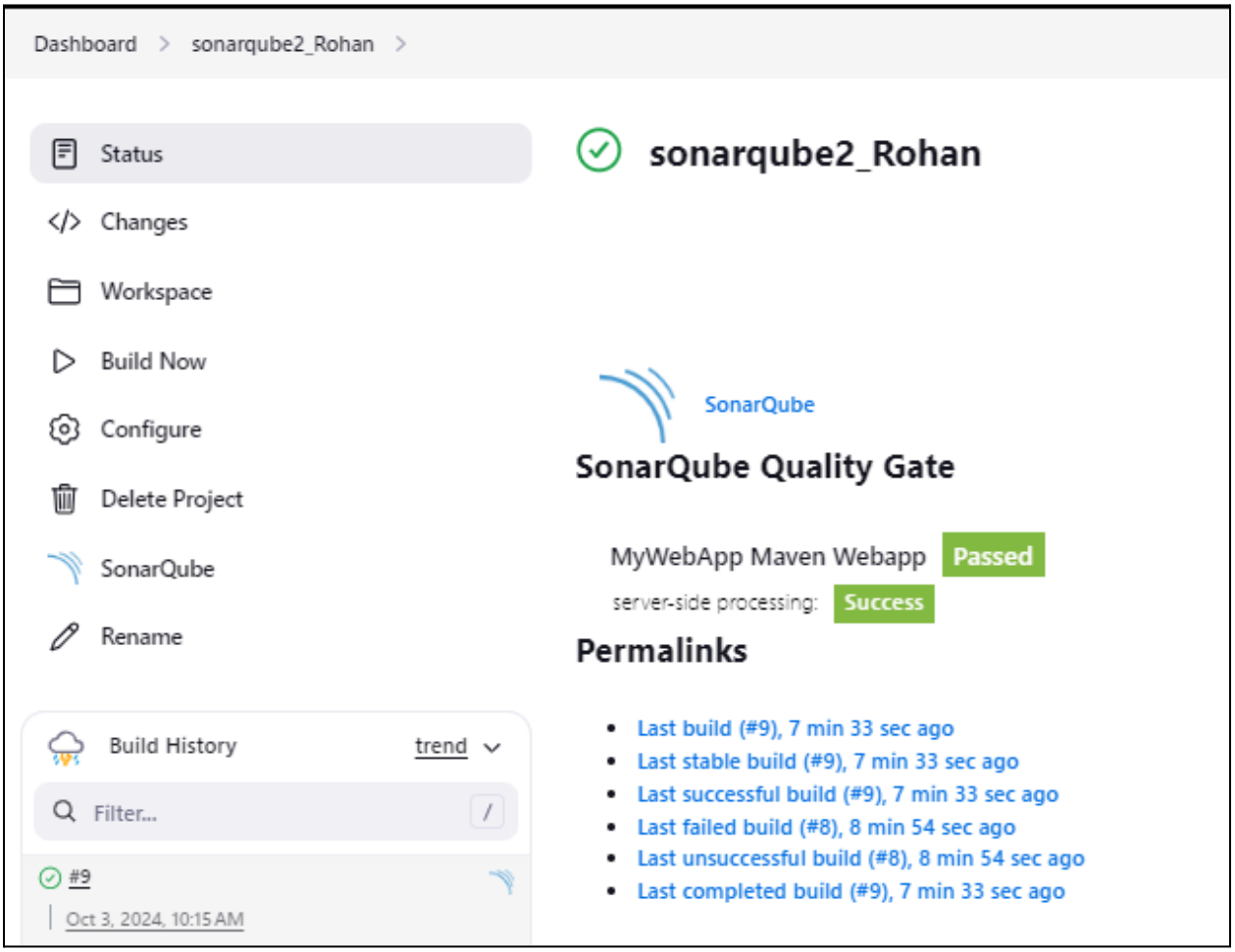


2. Check on prepare sonarqube scanner

3. Give the maven_path and click on save.



Build completed successfully.



Go to the console output and scroll down click on the analysis report link and you will be able to see the sonarqube analysis report.