**Steps-**

# **Integrating Jenkins with SonarQube:**

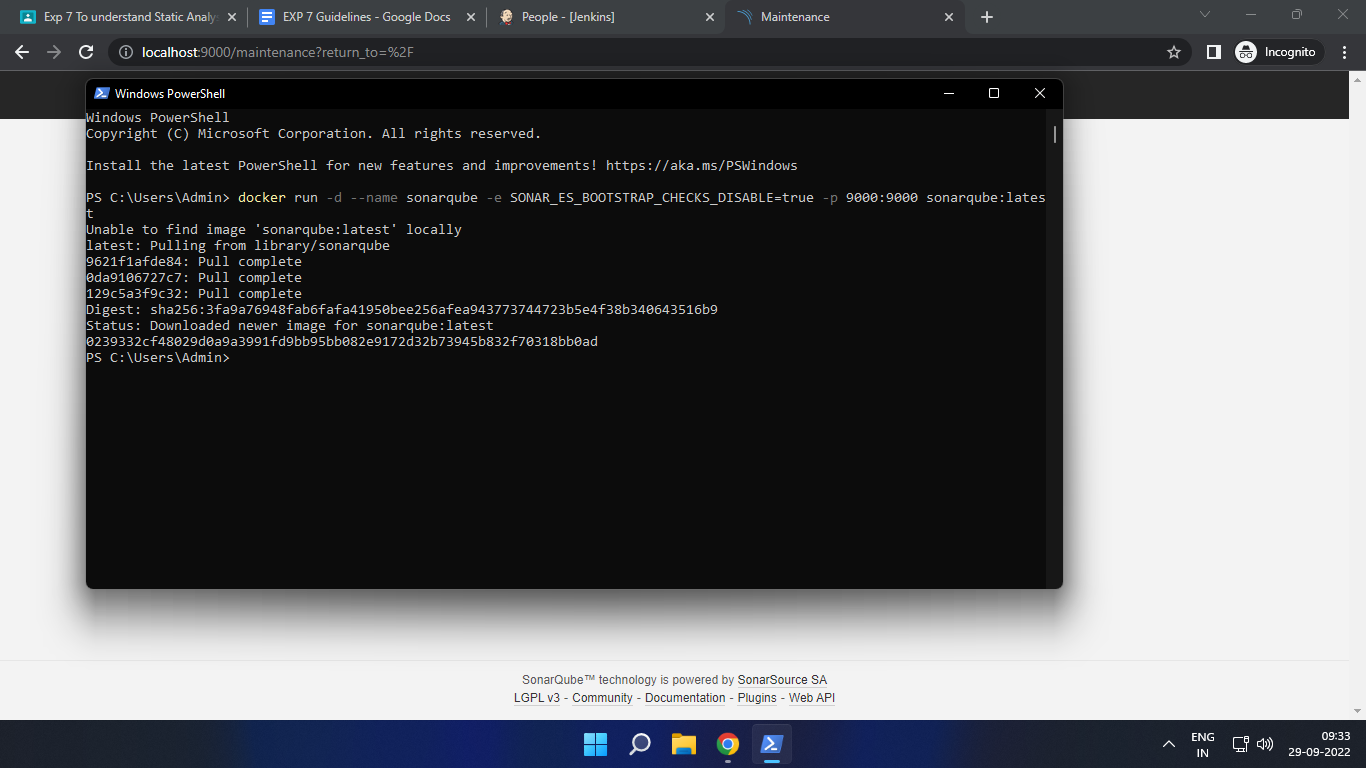
| Windows installation  Step 1 Install JDK 1.8  Step 2 download and install jenkins  <https://www.blazemeter.com/blog/how-to-install-jenkins-on-windows>  **Ubuntu installation**  [**https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-20-04#installing-the-default-jre-jdk**](https://www.digitalocean.com/community/tutorials/how-to-install-java-with-apt-on-ubuntu-20-04#installing-the-default-jre-jdk)  Step 1 Install JDK 1.8  sudo apt-get install openjdk-8-jre  sudo apt install default-jre  <https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04>  [Open SSH](https://www.cyberciti.biz/faq/ubuntu-linux-install-openssh-server/) |
| --- |

## **Prerequisites:**

* [Jenkins installed](https://www.digitalocean.com/community/tutorials/how-to-install-jenkins-on-ubuntu-20-04)
* [Docker Installed](https://docs.docker.com/engine/install/ubuntu/) (for SonarQube)
* SonarQube Docker Image

**Steps to integrate Jenkins with SonarQube**

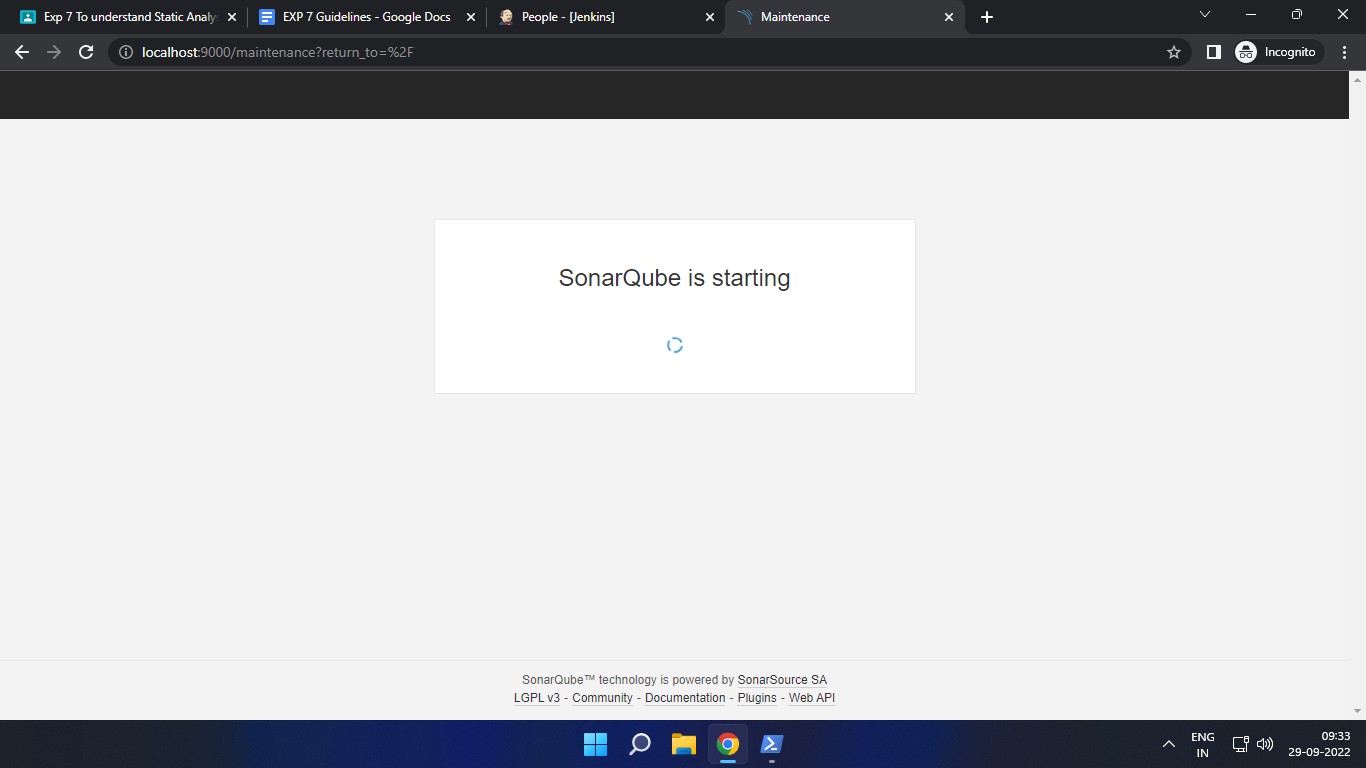
1. Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.
2. Run SonarQube in a Docker container using this command -



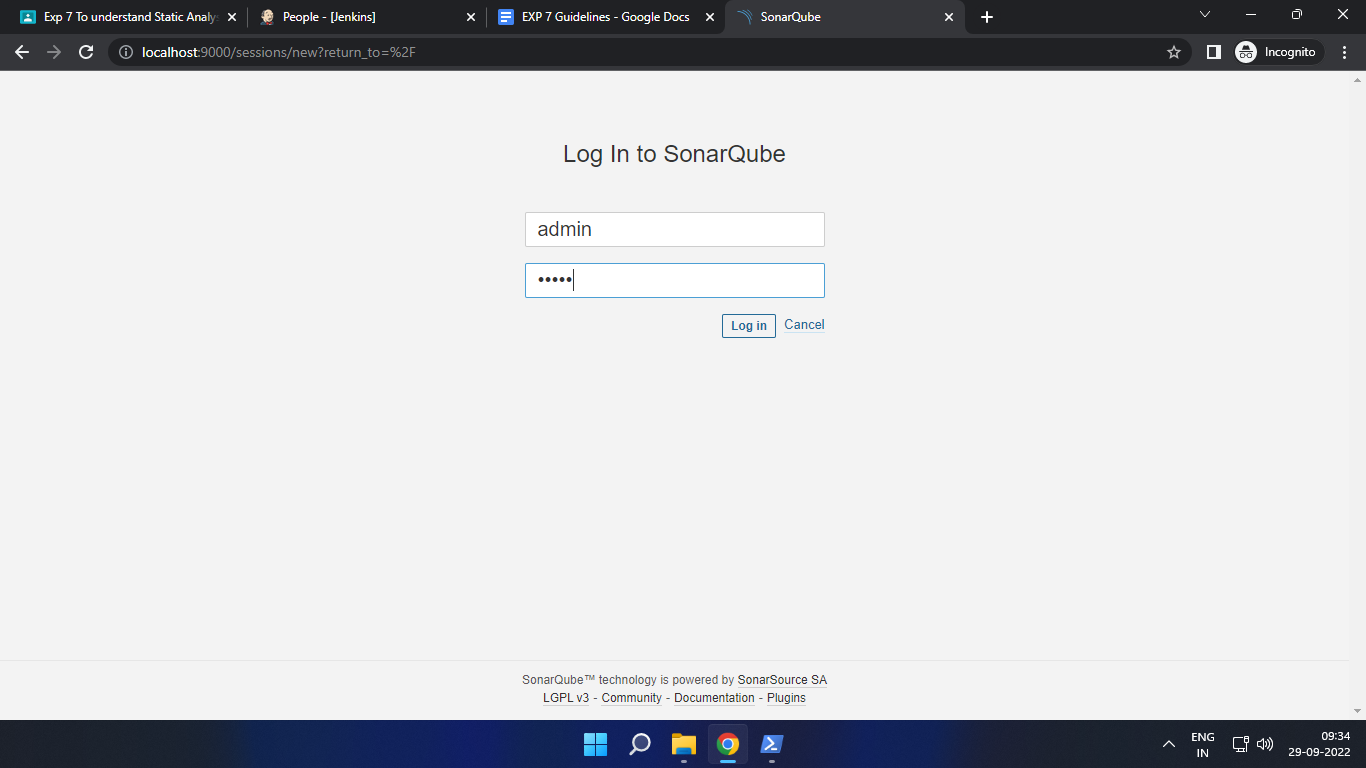
Warning: run below command only once

docker run -d --name sonarqube -e SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest

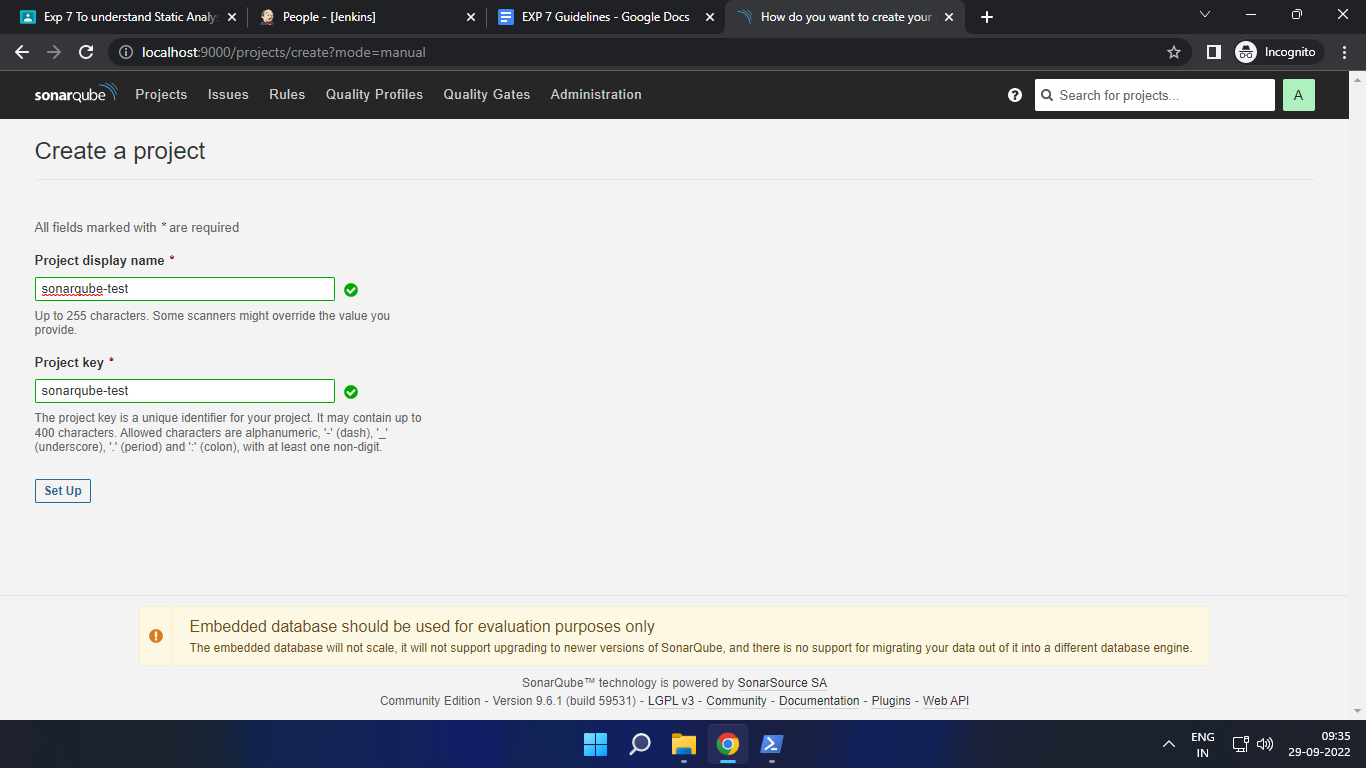
1. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



1. Login to SonarQube using username *admin* and password *admin*.

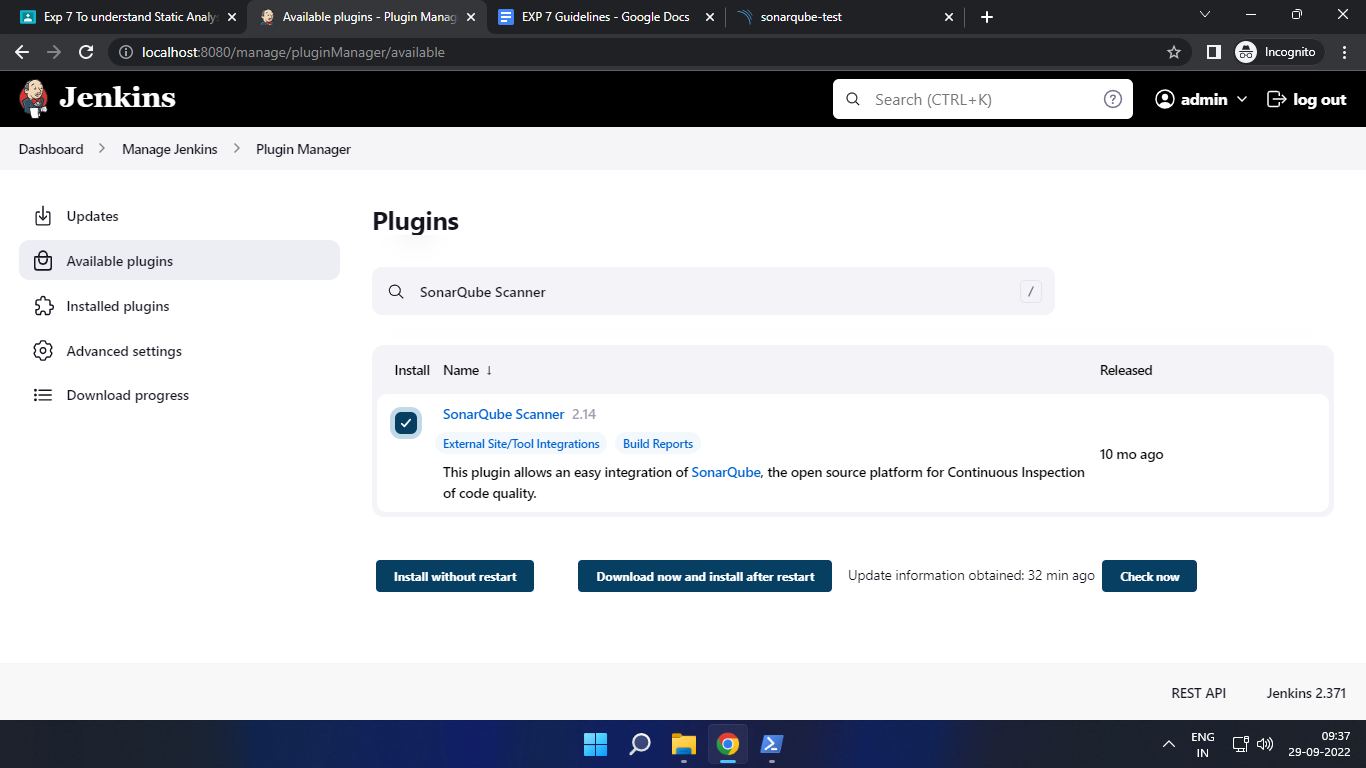


1. Create a manual project in SonarQube with the name **sonarqube**

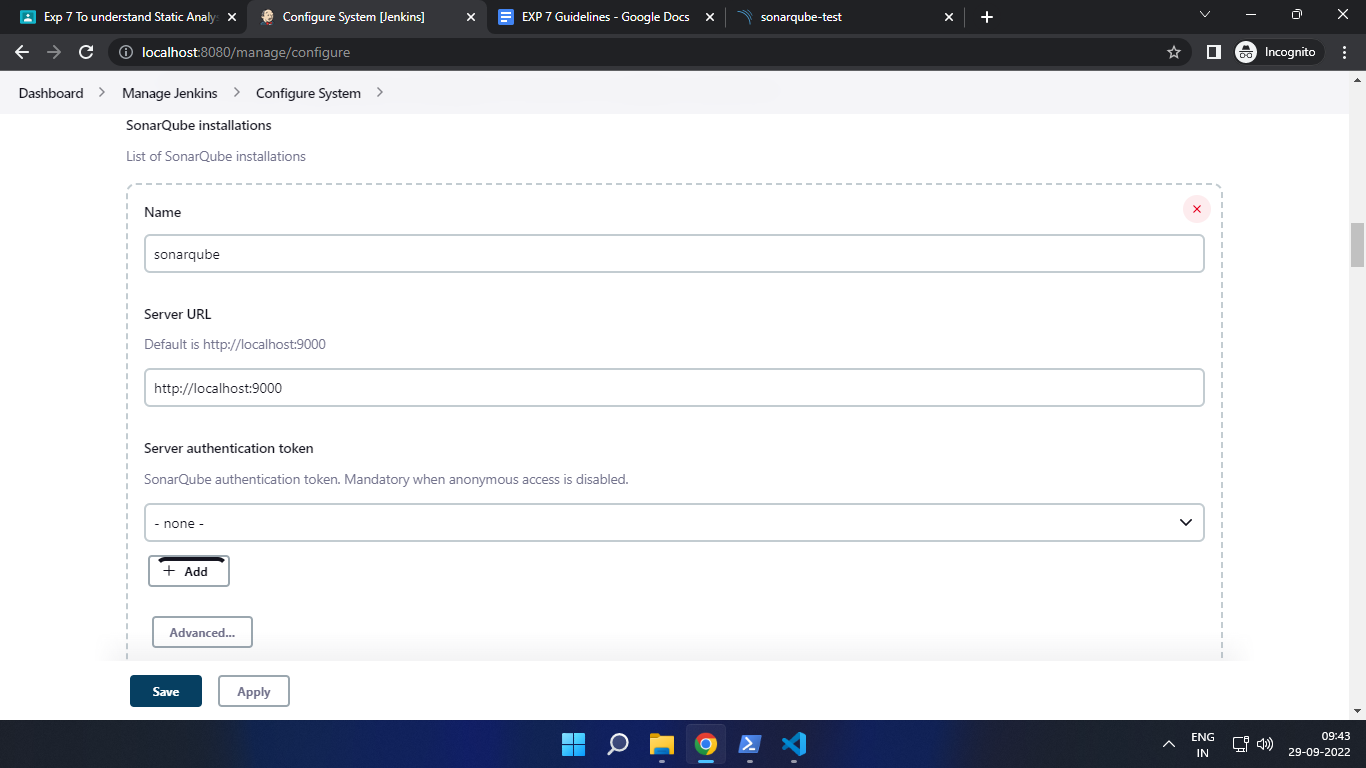
****

Setup the project and come back to Jenkins Dashboard.

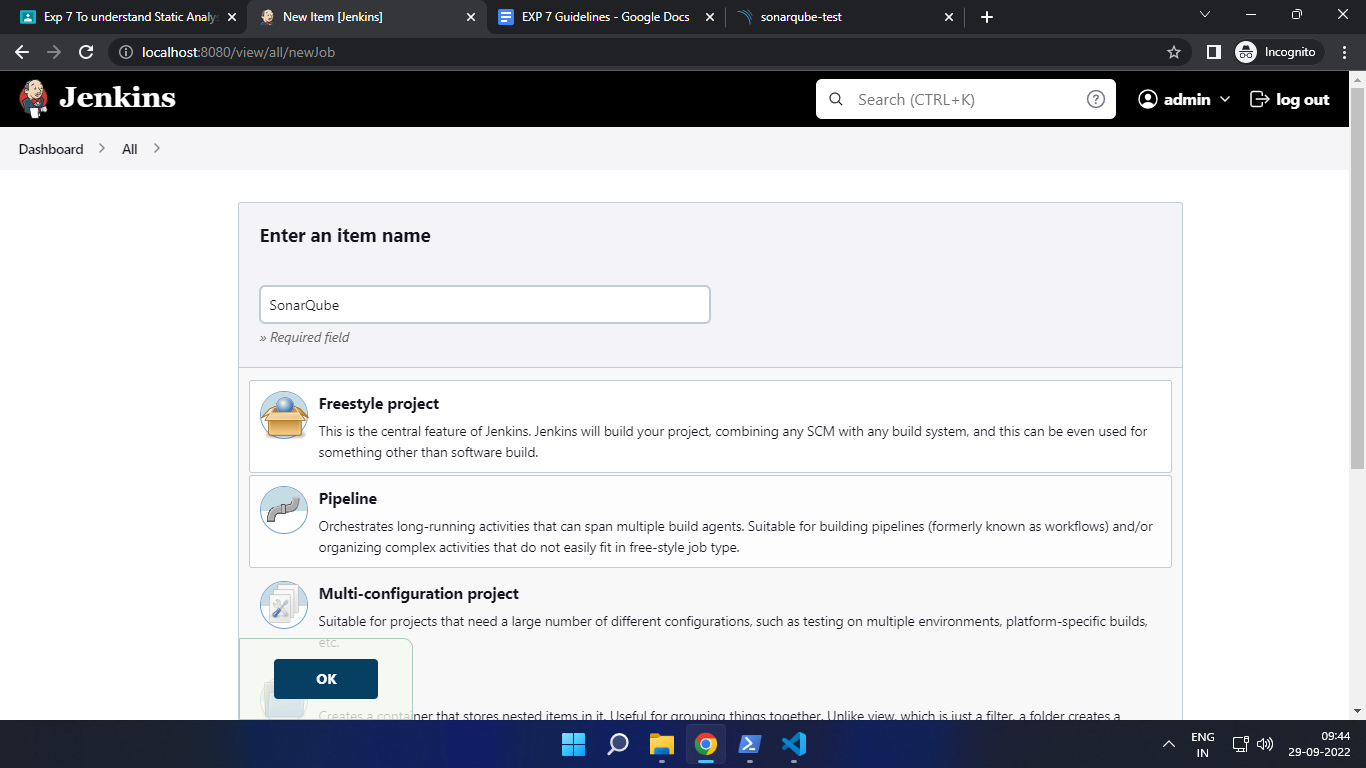
Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.

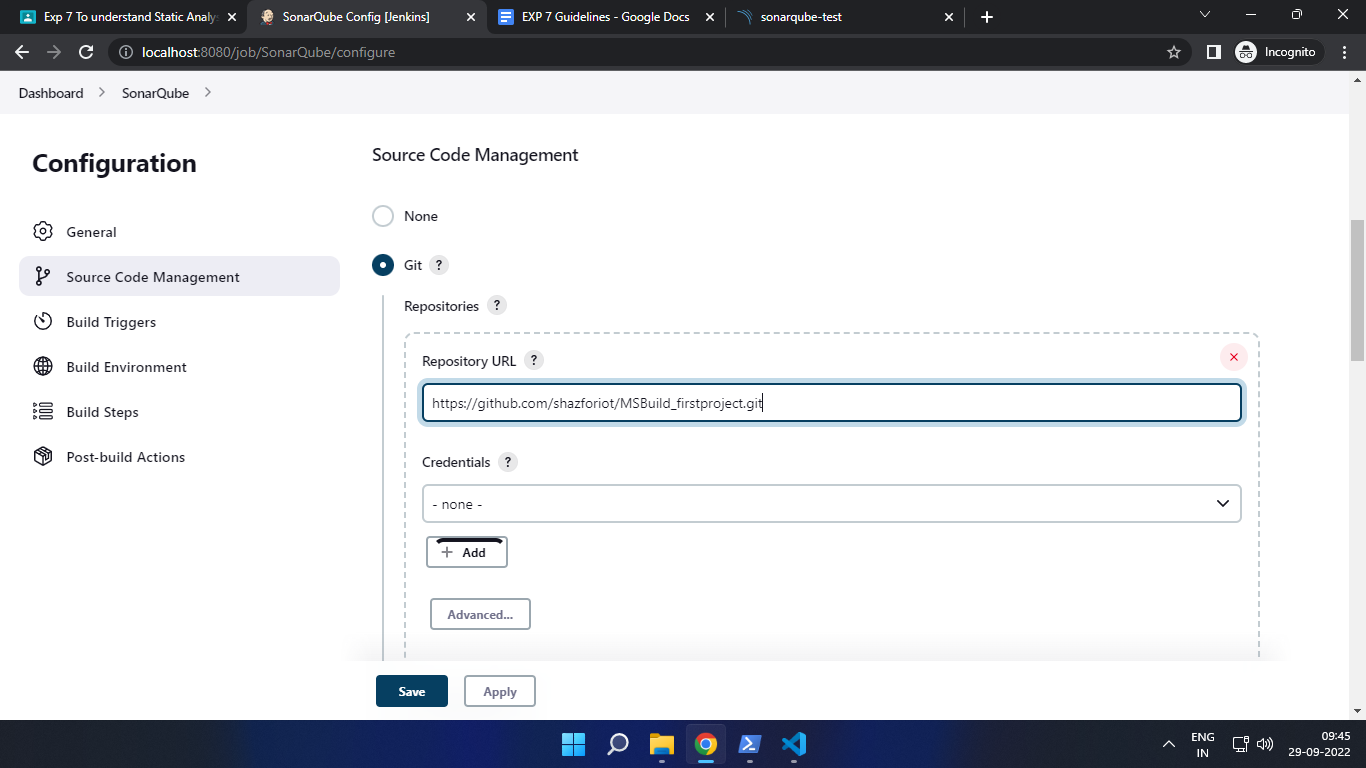


1. Under Jenkins ‘Configure System’, look for SonarQube Servers and enter the details.

Enter the Server Authentication token if needed.

1. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.
2. After the configuration, create a New Item in Jenkins, choose a freestyle project.

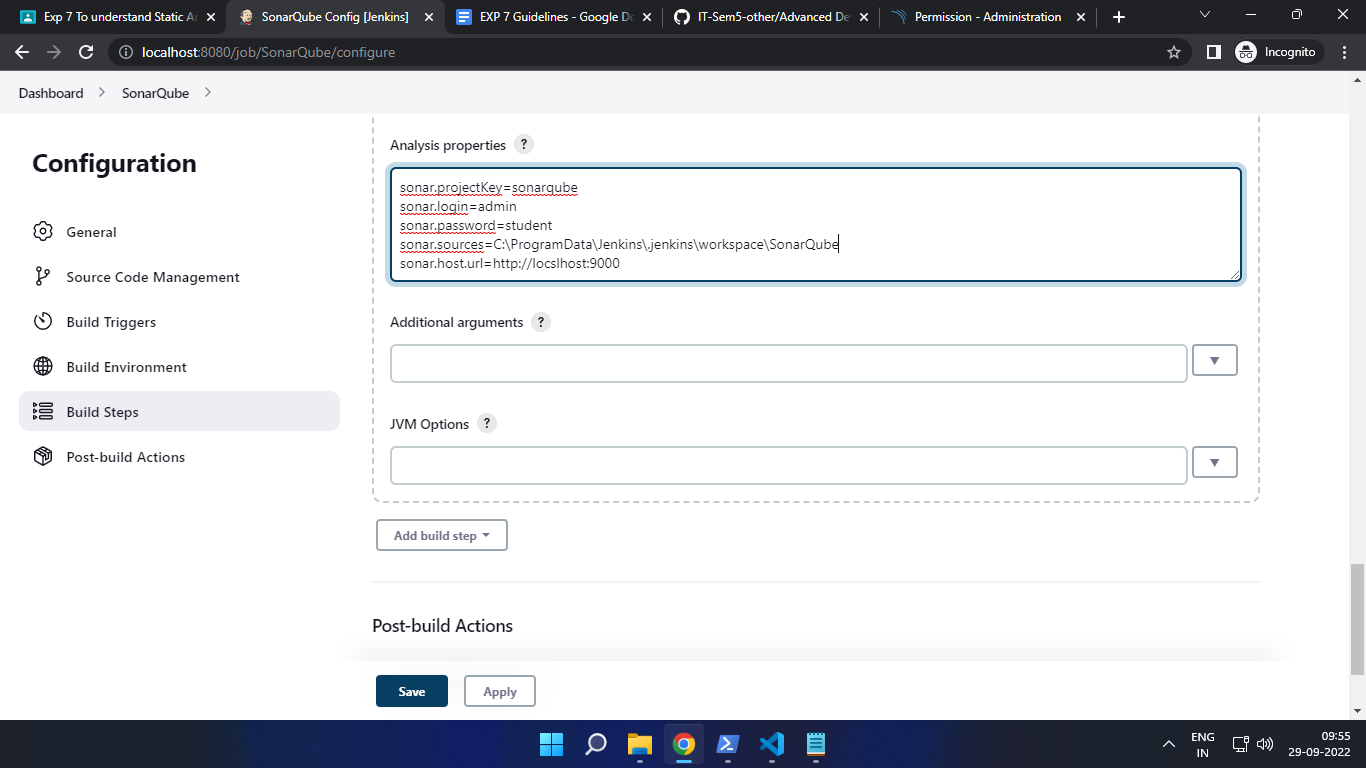


1. Choose this GitHub repository in Source Code Management. 

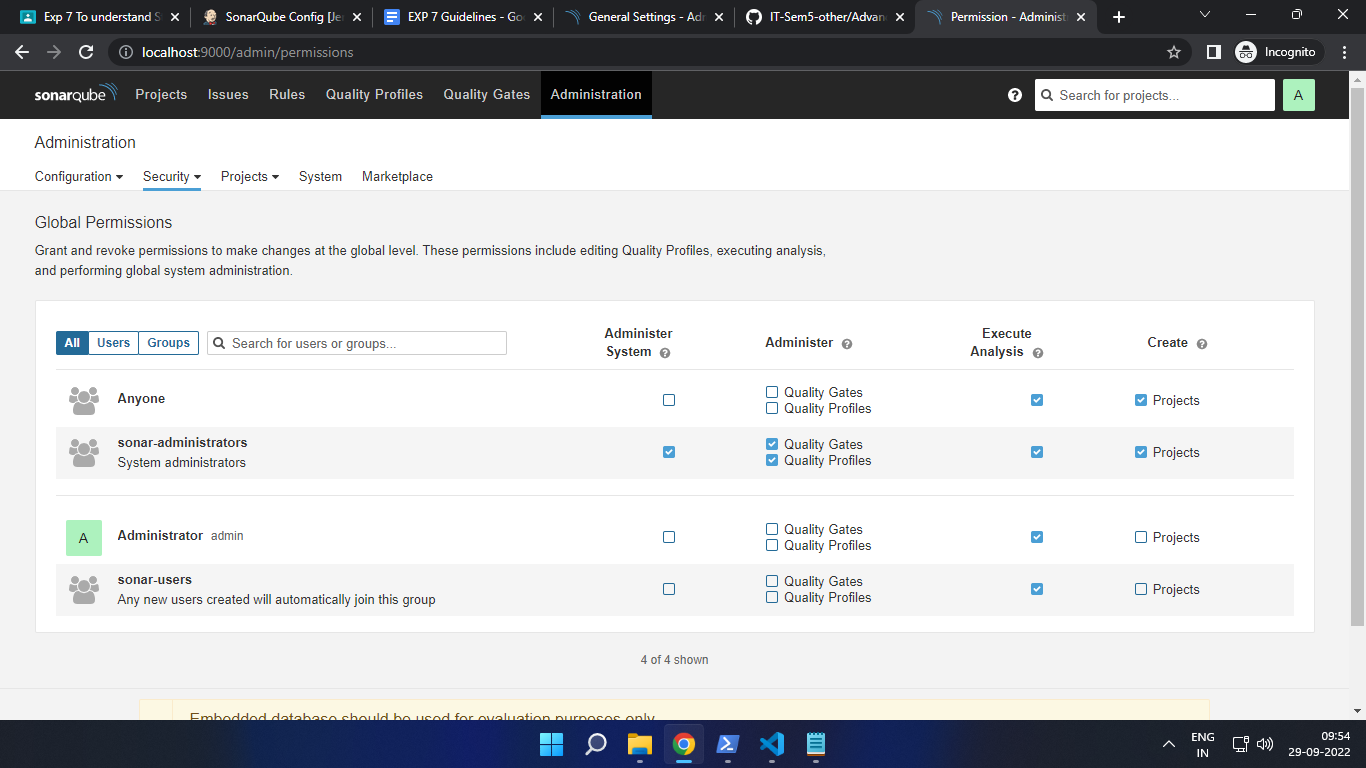
<https://github.com/shazforiot/MSBuild_firstproject.git>

It is a sample hello-world project with no vulnerabilities and issues, just to test the integration.

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

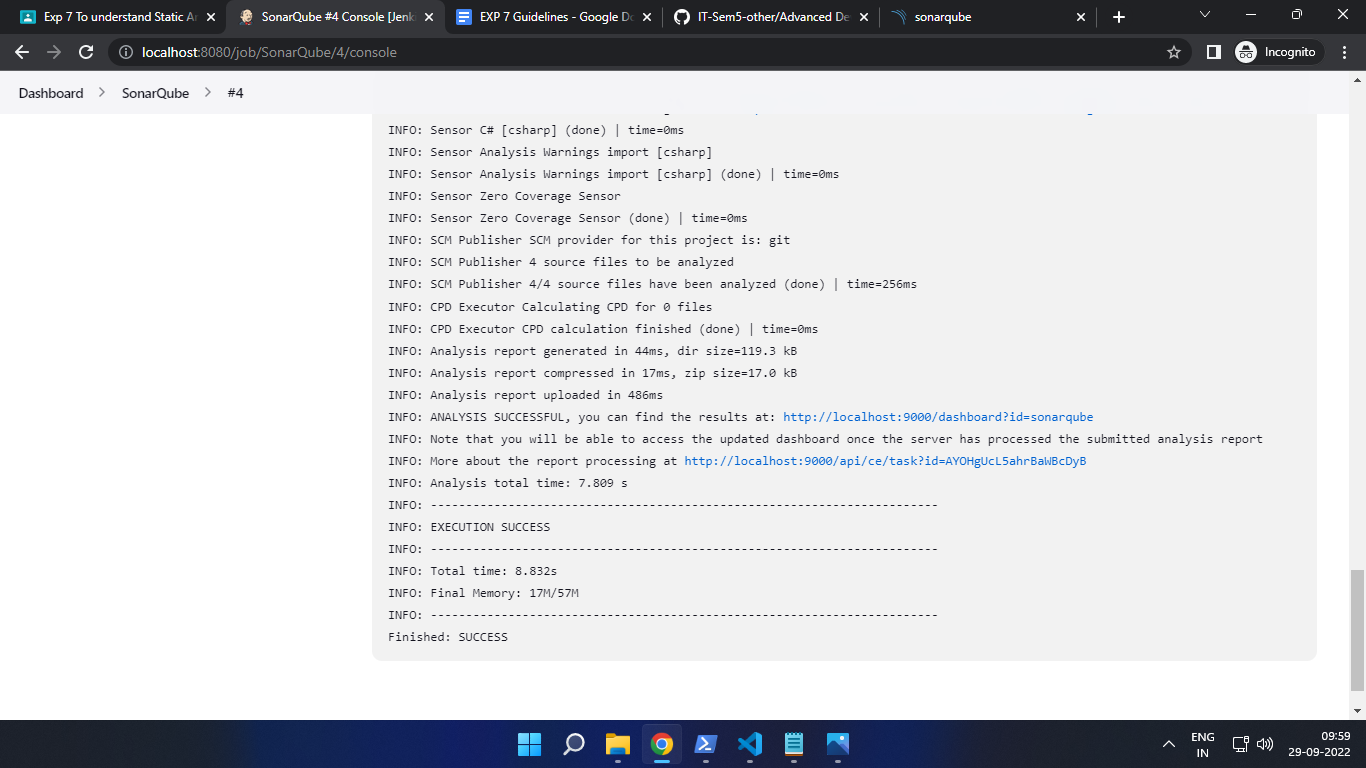
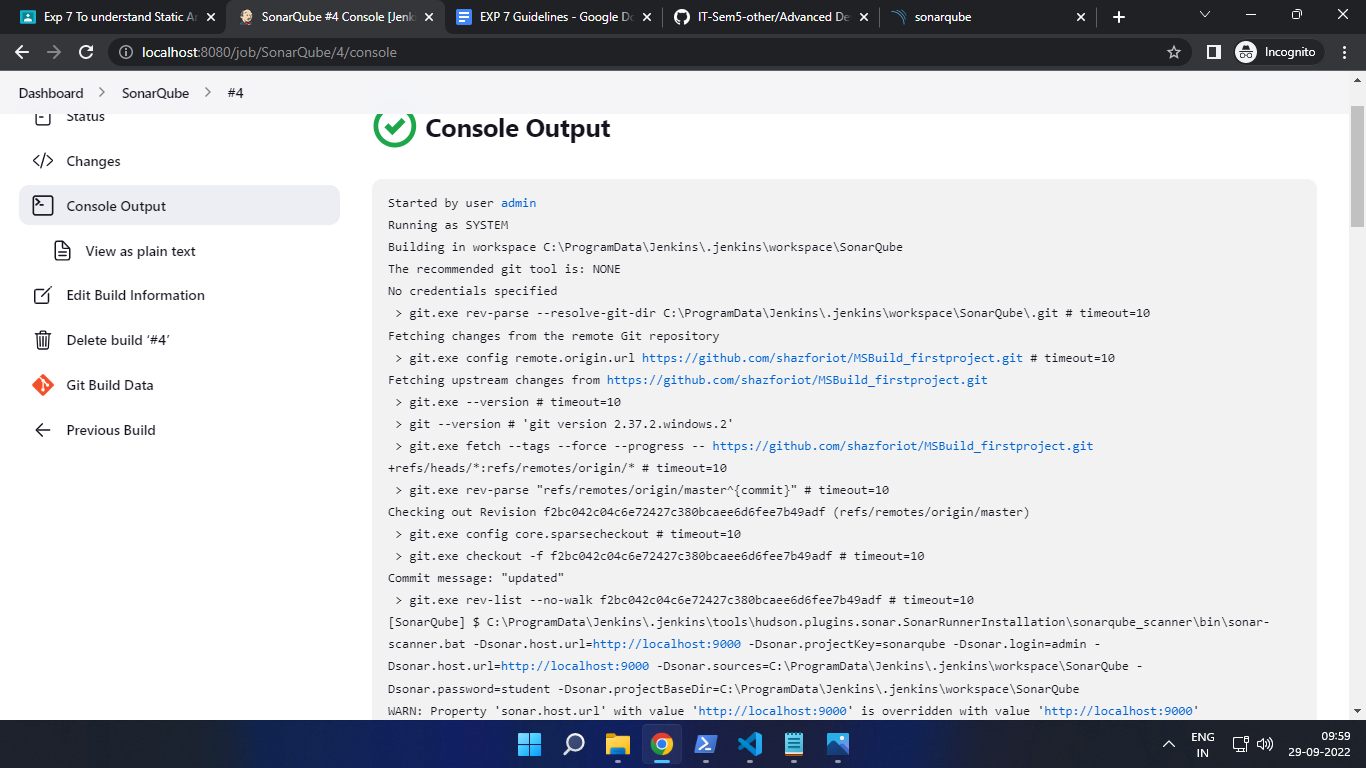


1. Go to http://localhost:9000/<user\_name>/permissions and allow Execute Permissions to the Admin user.

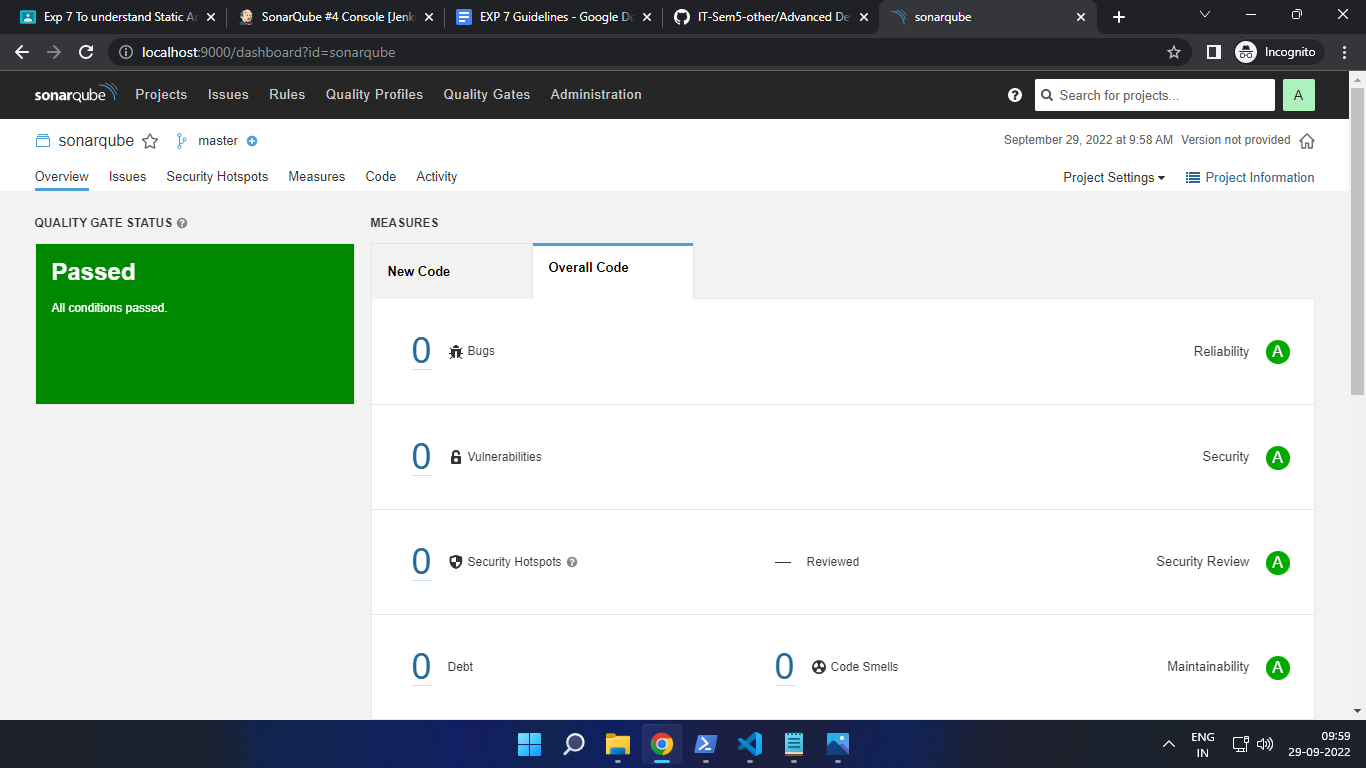


1. Run The Build.

Check the console output.



1. Once the build is complete, check the project in SonarQube.



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

### **Conclusion**

In this experiment, we have understood the importance of SAST and have successfully integrated Jenkins with SonarQube for Static Analysis and Code Testing.