**ADVANCE DEVOPS EXP 7**

Prajjwal Pandey **-** D15A - 33

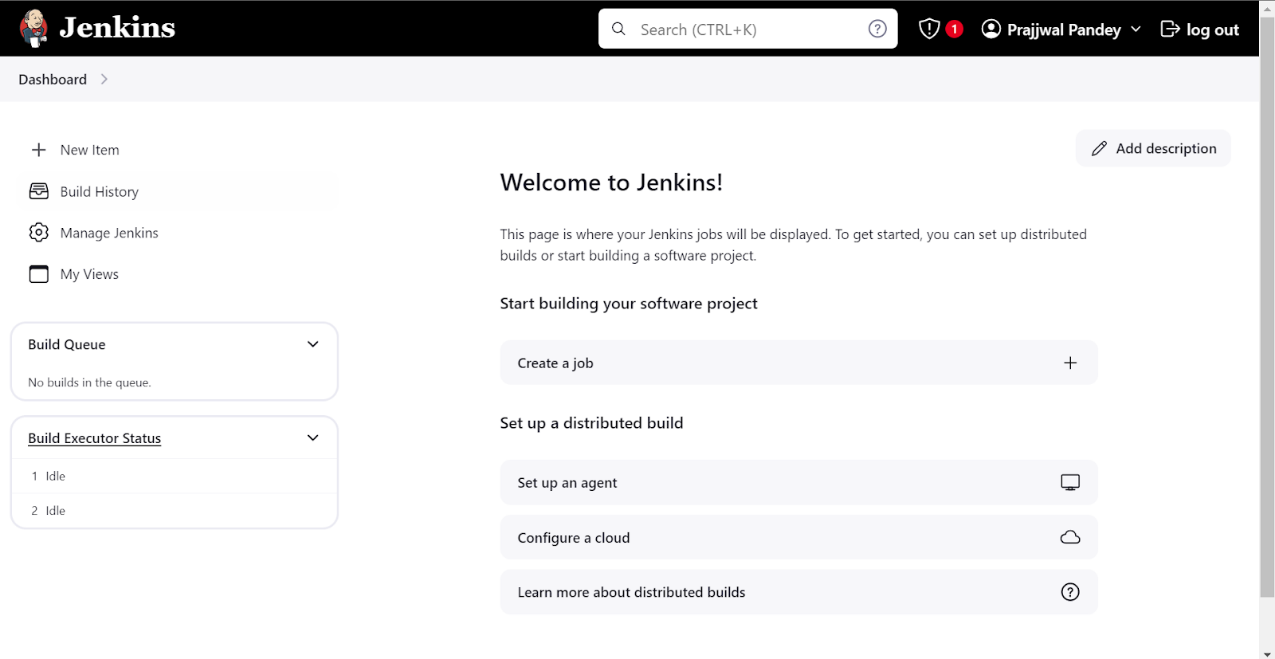
**Aim:** To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

**Integrating Jenkins with SonarQube:**

* Jenkins installed
* Docker Installed (for SonarQube)

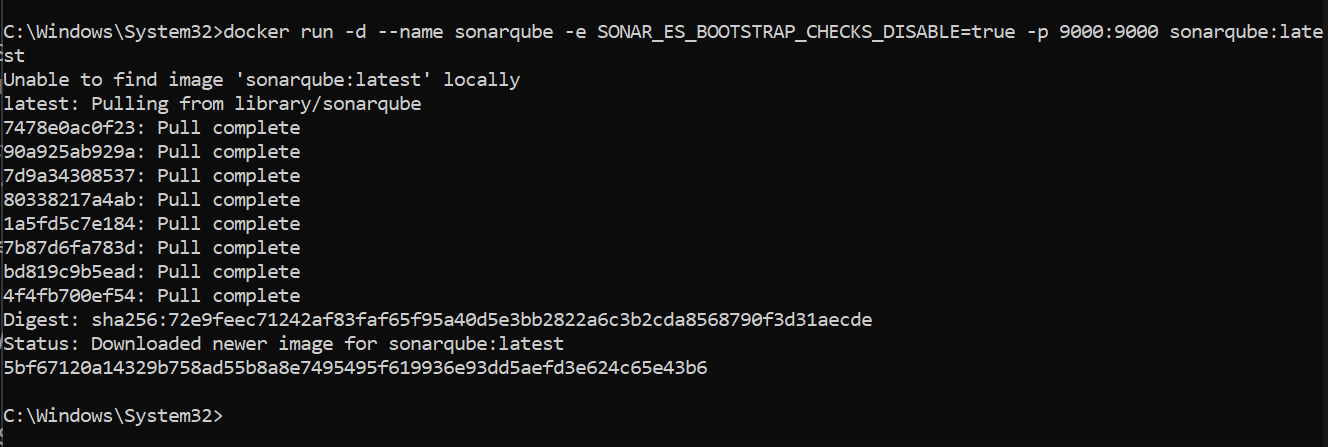
● SonarQube Docker Image

**Steps to integrate Jenkins with SonarQube**

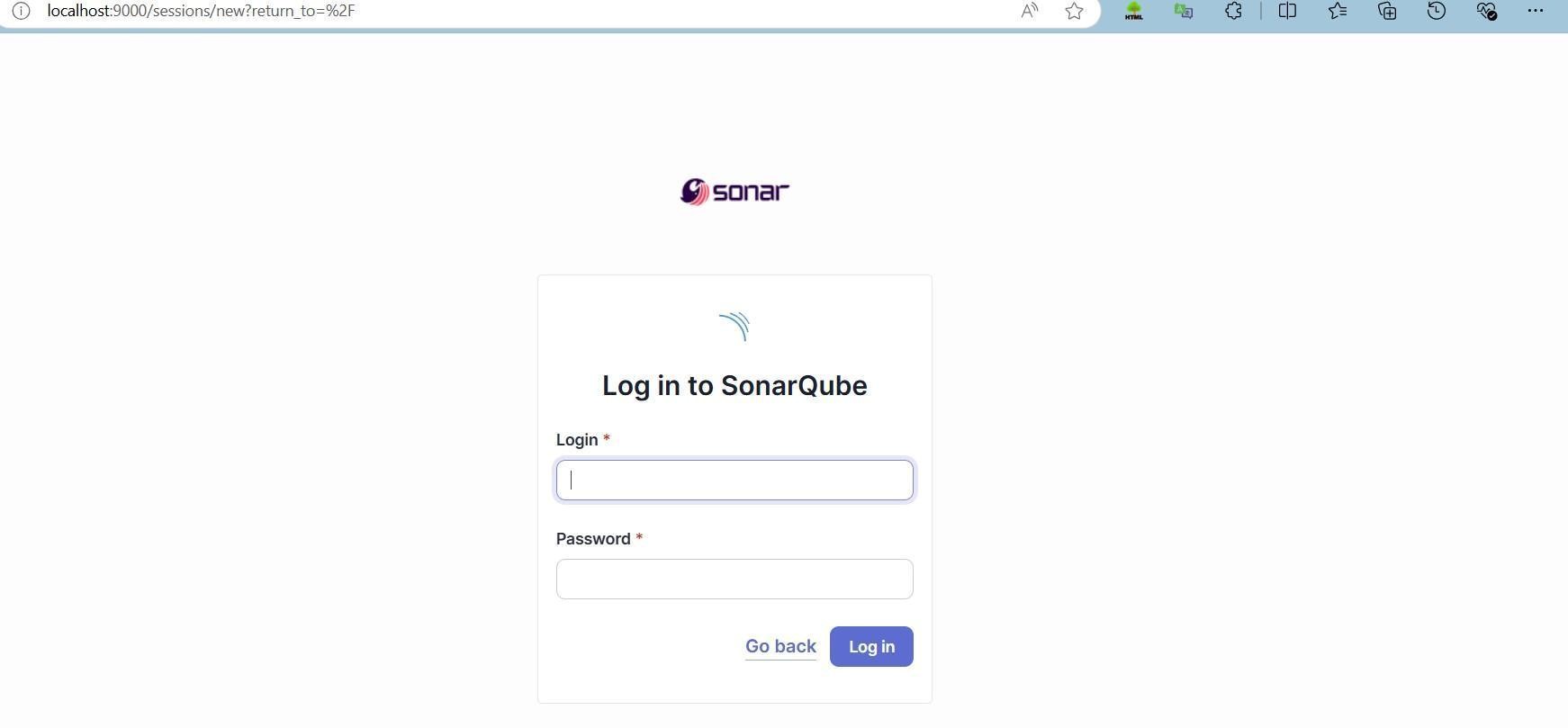
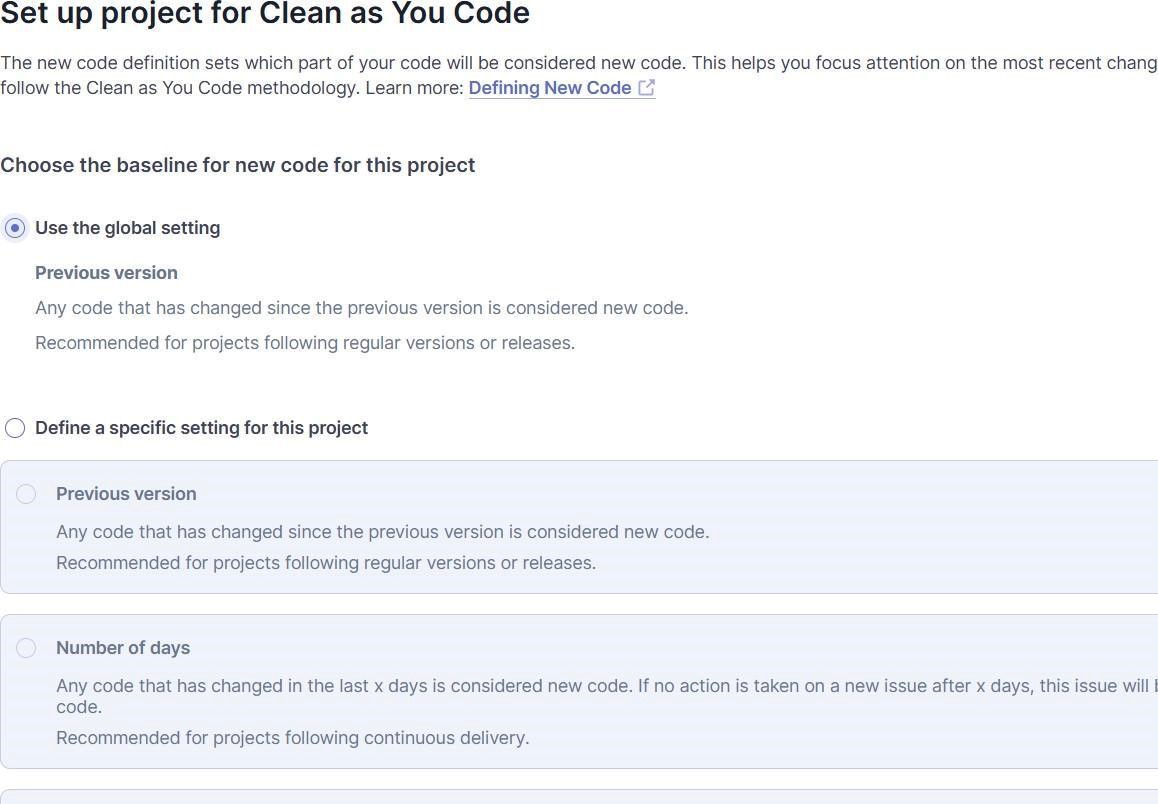
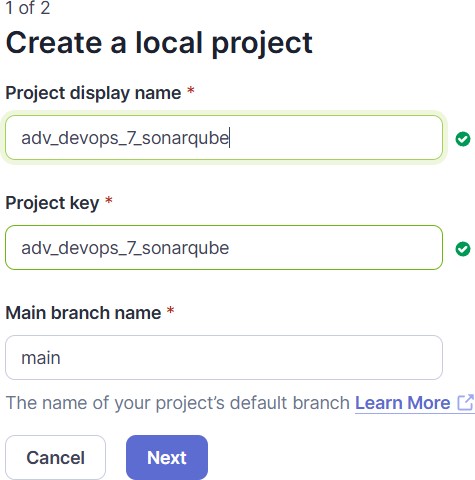
1.Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.

2.Run SonarQube in a Docker container using this command **docker run -d --name sonarqube -e**

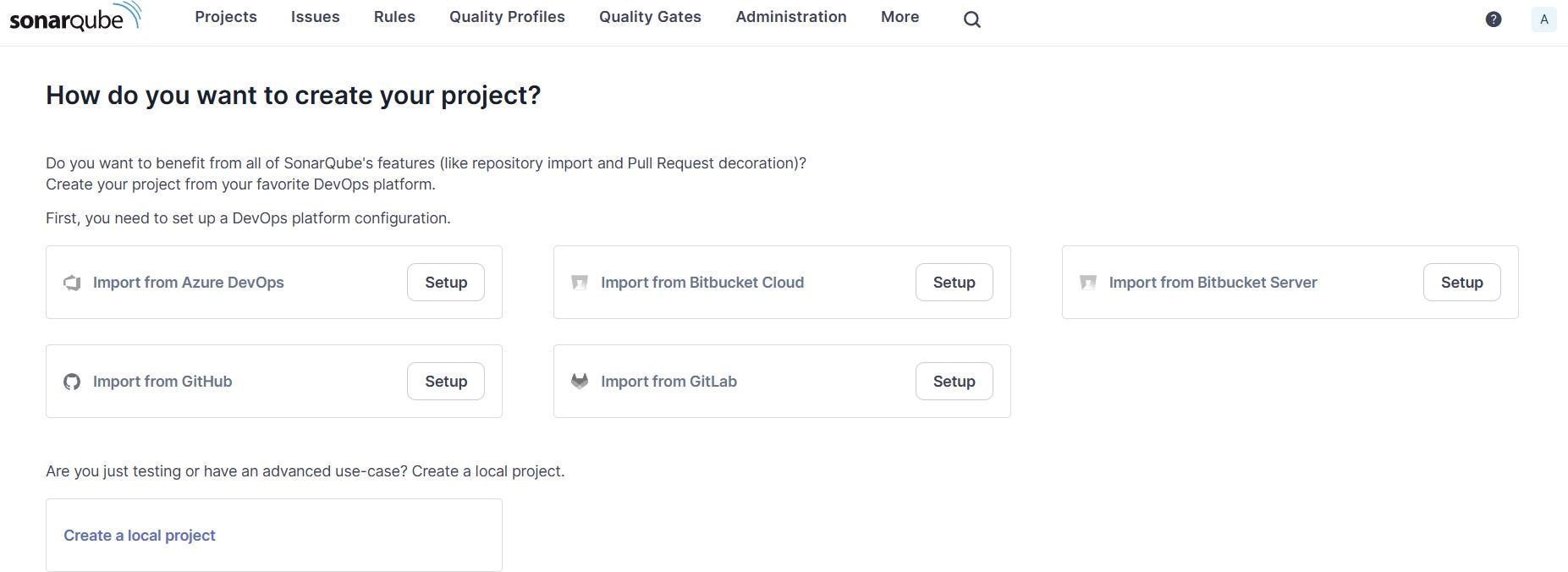
**SONAR\_ES\_BOOTSTRAP\_CHECKS\_DISABLE=true -p 9000:9000 sonarqube:latest**

****

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



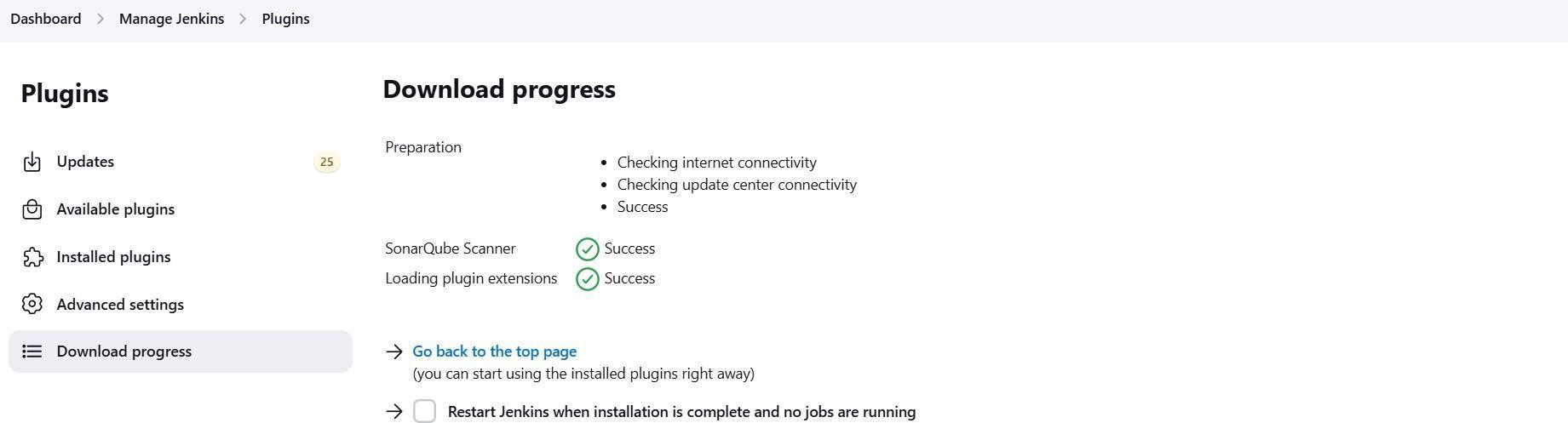
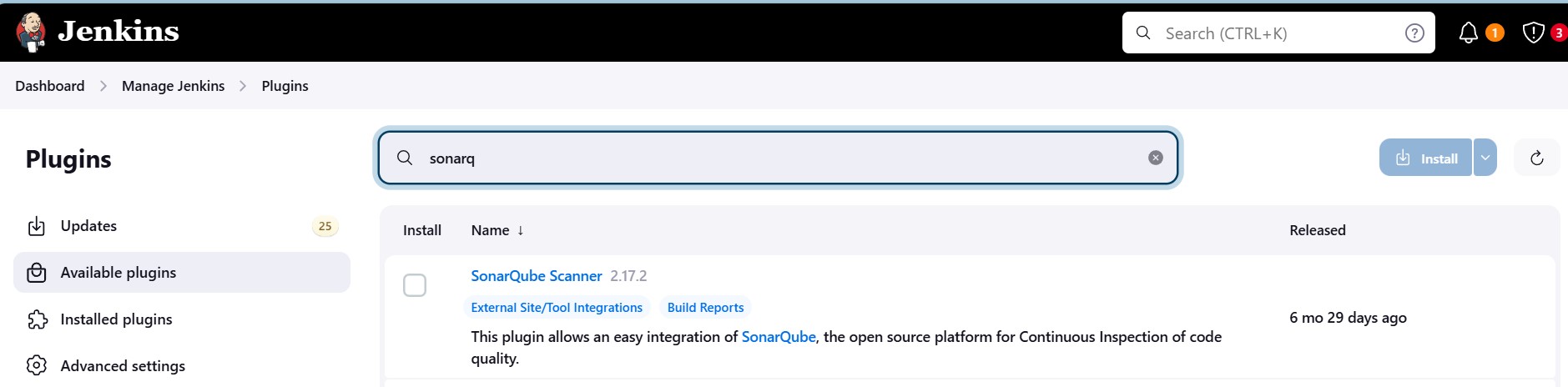
4.Login to SonarQube using username admin and password admin.



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

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:900**](http://localhost:9000/)

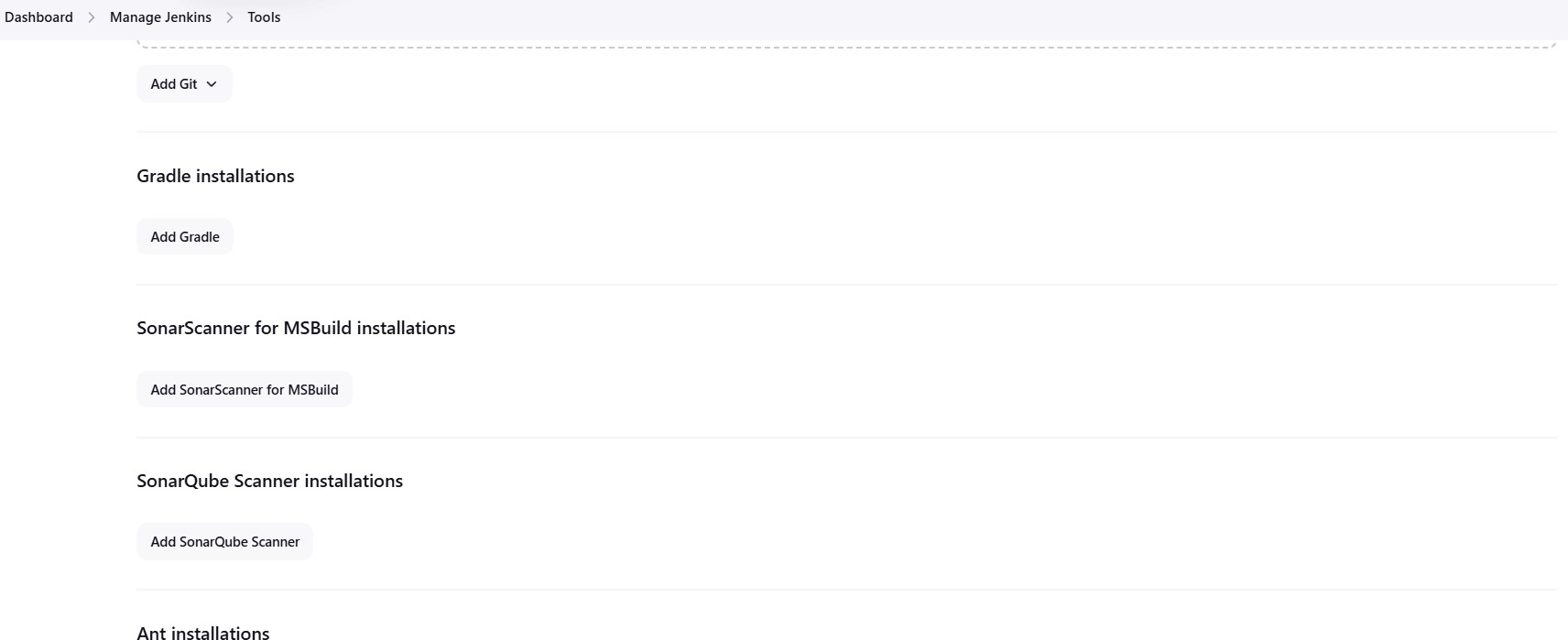
[**0**](http://localhost:9000/)



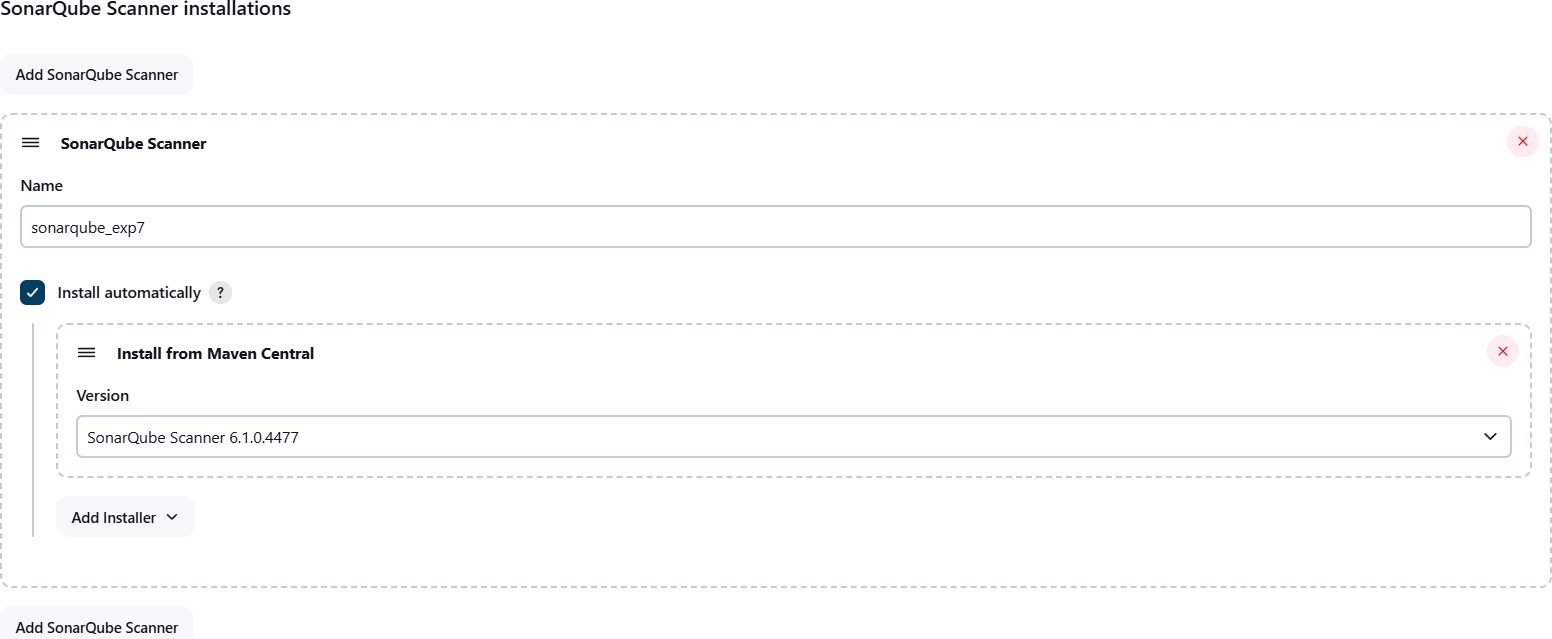
7. Search for SonarQube Scanner under Global Tool Configuration.

Choose the latest configuration and choose Install automatically.

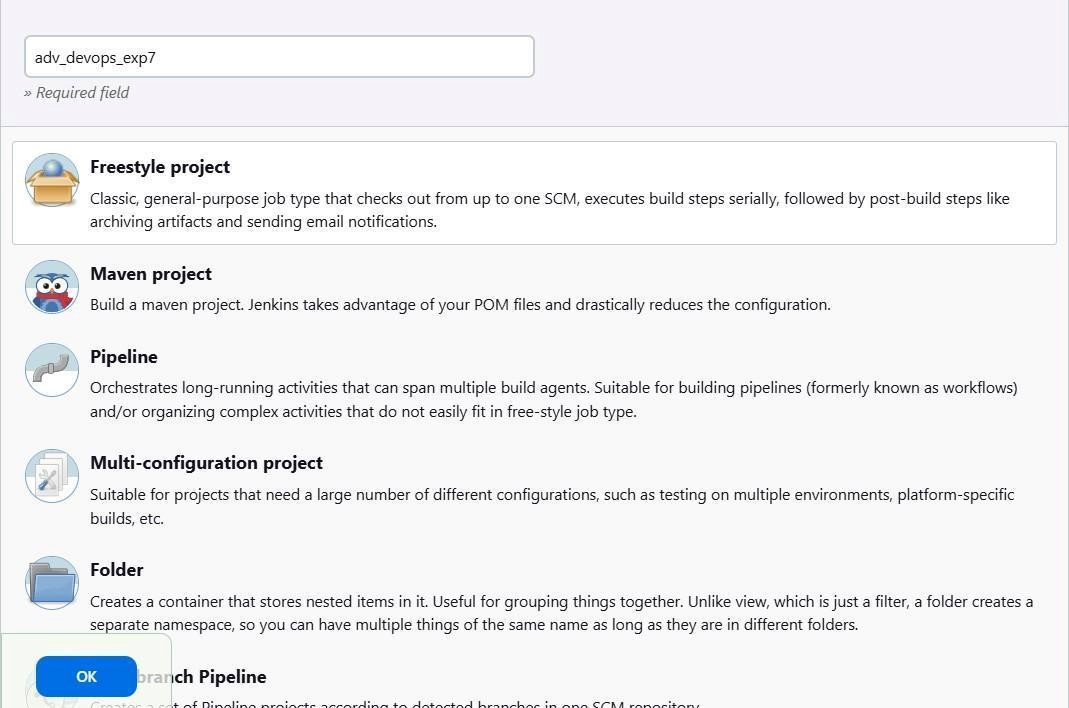
**Dashboard > Manage Jenkins > Tools**



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



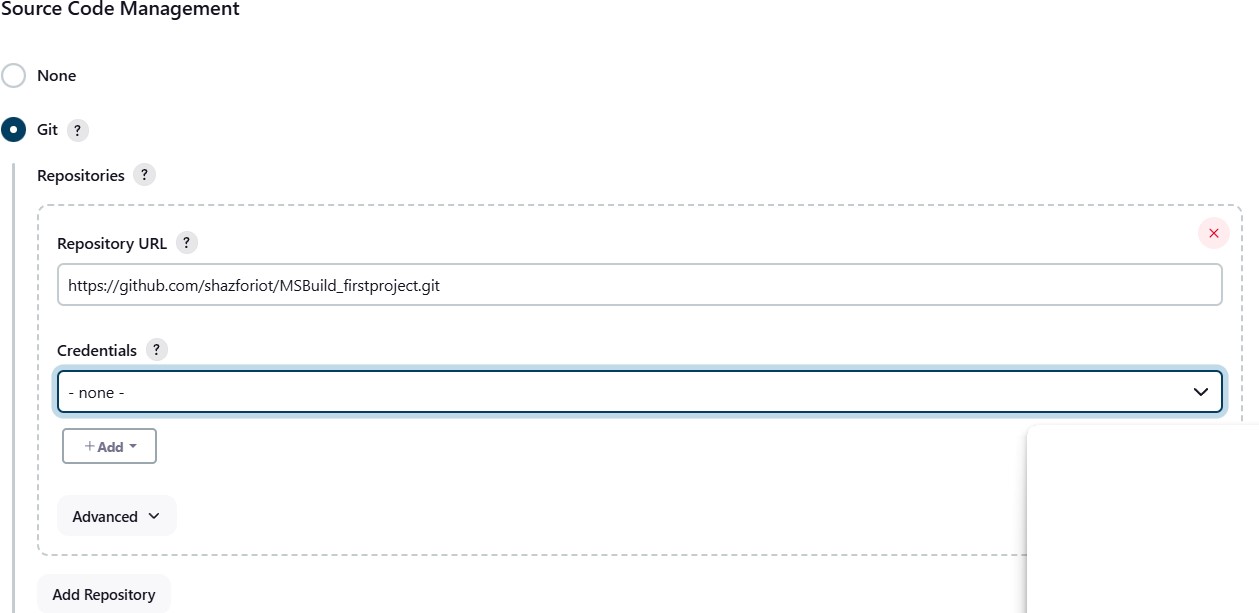
8.After the configuration, create a New Item in Jenkins, choose a freestyle project.



1. Choose this GitHub repository in Source Code Management.

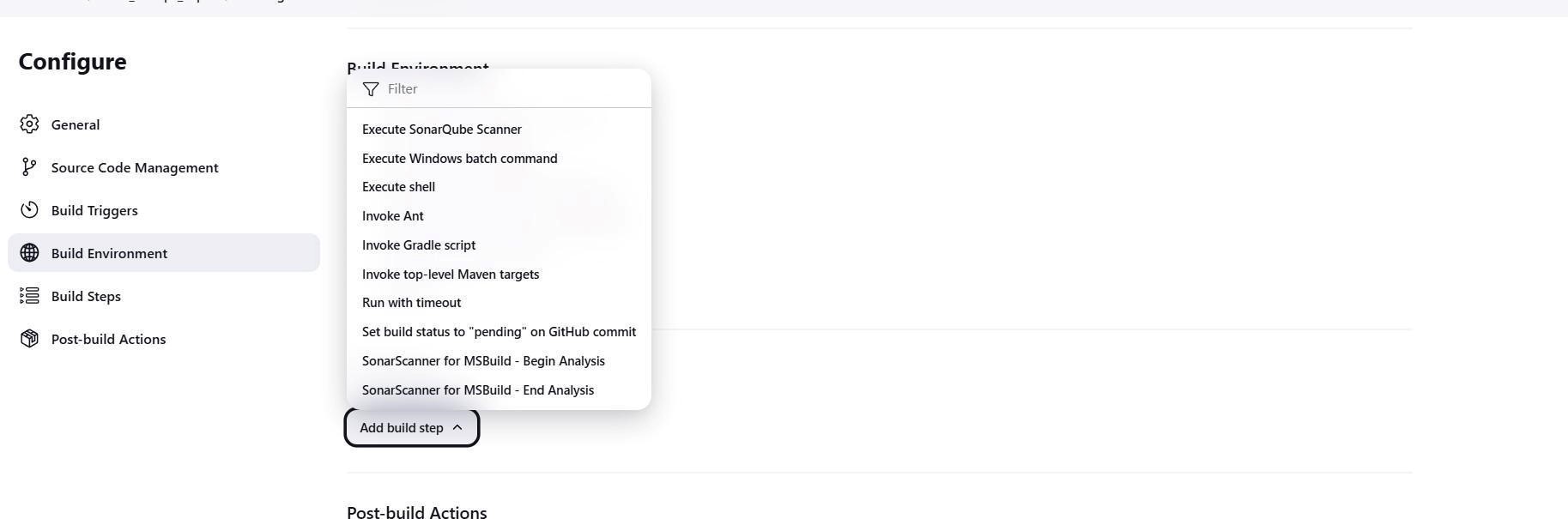
https://github.com/shazforiot/MSBuild\_firstproject.git

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



1. Under **Select project → Configuration → Build steps → Execute**

**SonarQube Scanner**, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, Source path and Host URL.



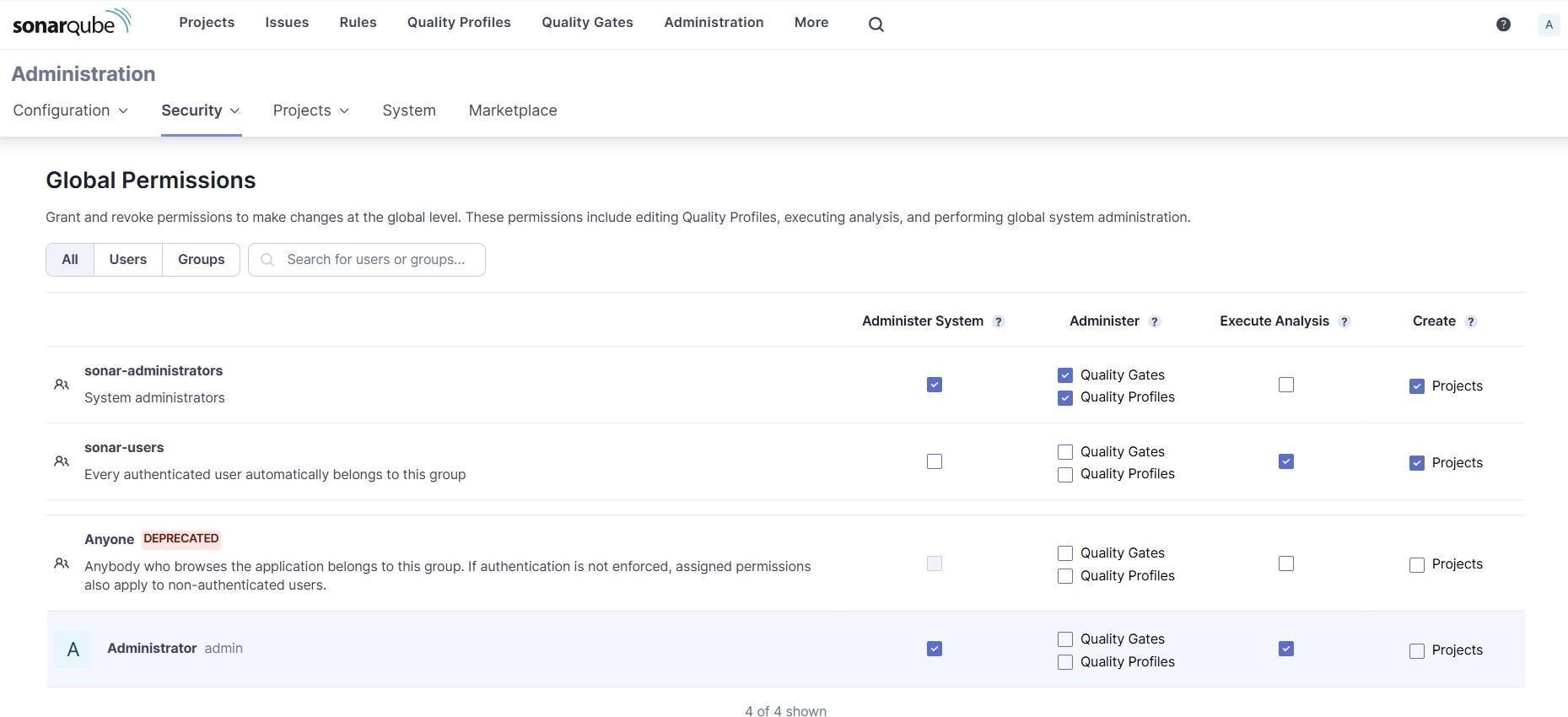
Then

save



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

Permissions to the Admin user



***IF CONSOLE OUTPUT FAILED:***

**Step 1: Generate a New Authentication Token in SonarQube**

* 1. **Login to SonarQube:**
     1. 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).

* 1. **Generate a New Token:**
     1. 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**.

* 1. **Update the SonarQube Token:**
     1. 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.

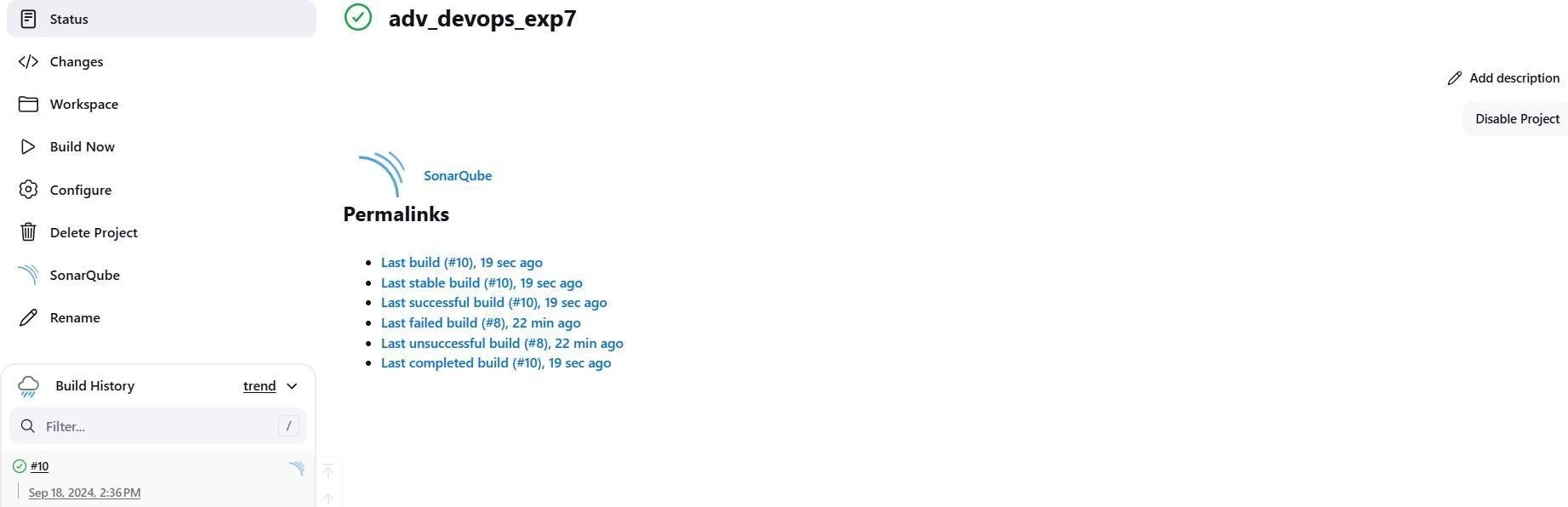
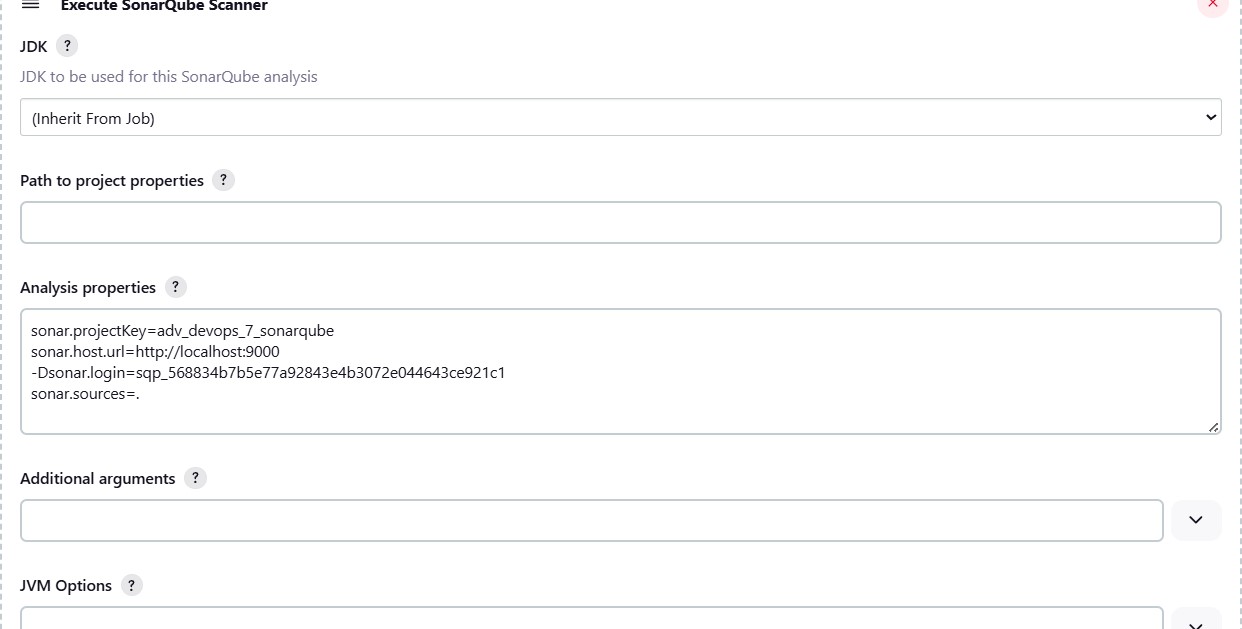
12.

Run

the

Jenkins

build.



Check the console Output 

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

