

Adv DevOps Practical 7

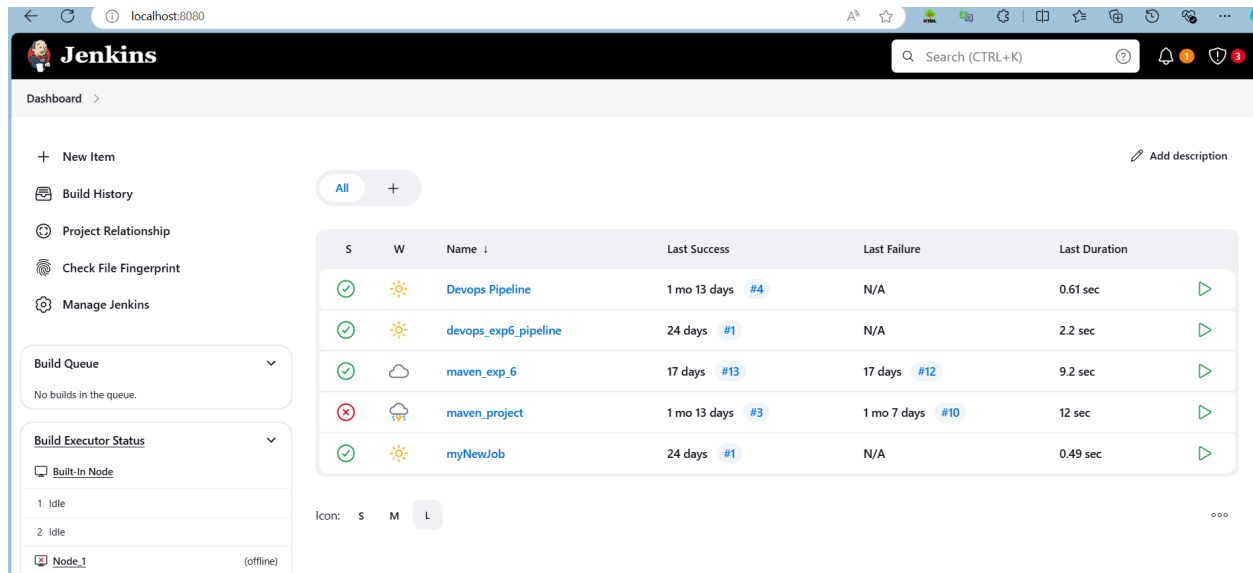
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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.



The screenshot shows the Jenkins Dashboard at localhost:8080. The left sidebar contains navigation links: New Item, Build History, Project Relationship, Check File Fingerprint, and Manage Jenkins. The main area displays a table of builds with columns: S, W, Name, Last Success, Last Failure, and Last Duration. The table lists five builds: Devops Pipeline, devops_exp6_pipeline, maven_exp_6, maven_project, and myNewJob. The 'maven_project' build is marked as failed.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Devops Pipeline	1 mo 13 days #4	N/A	0.61 sec
✓	☀	devops_exp6_pipeline	24 days #1	N/A	2.2 sec
✓	☁	maven_exp_6	17 days #13	17 days #12	9.2 sec
✗	☁	maven_project	1 mo 13 days #3	1 mo 7 days #10	12 sec
✓	☀	myNewJob	24 days #1	N/A	0.49 sec

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

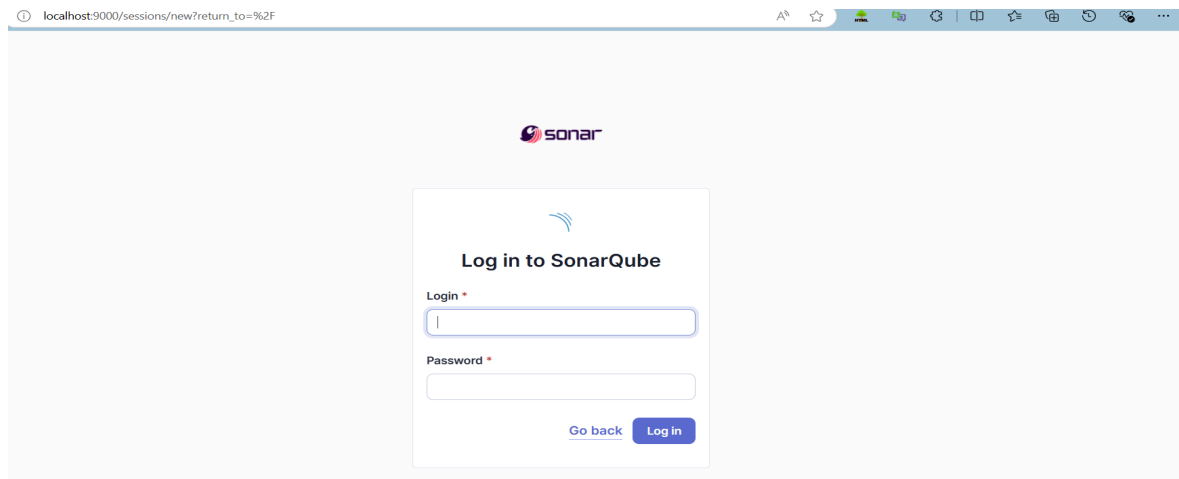
`docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest`

-----Warning: run below command only once

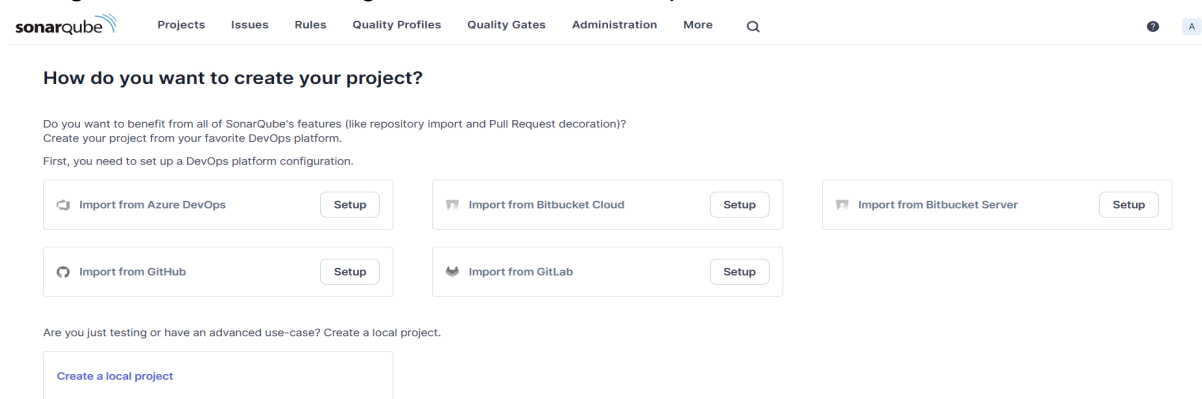
```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Ayush Maurya> 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
762bedf4b1b7: Pull complete
95f9bd9906fa: Pull complete
a32d681e6b99: Pull complete
aabdd0a18314: Pull complete
5161e45ecd8d: Pull complete
aeb0020dfa06: Pull complete
01548d361aea: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:bb444c58c1e04d8a147a3bb12af941c57e0100a5b21d10e599384d59bed36c86
Status: Downloaded newer image for sonarqube:latest
4af48468290f95b22362652ee37b96c935b0bed754945c62cf3b0d5d51a2ac0c
PS C:\Users\Ayush Maurya>
```

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

1 of 2

Create a local project

Project display name *

Project key *

Main branch name *

The name of your project's default branch [Learn More](#)

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 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.

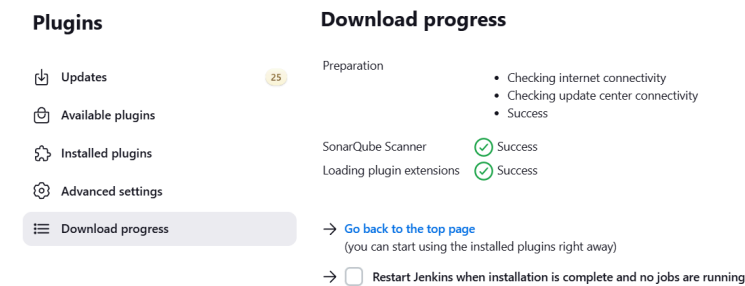
☐ Number of days
Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will be closed.
Recommended for projects following continuous delivery.

Setup the project and come back to Jenkins Dashboard.

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



The screenshot shows the Jenkins 'Manage Jenkins' > 'Plugins' page. A search bar at the top contains 'sonarq'. Below the search bar, a table lists the 'SonarQube Scanner' plugin (version 2.17.2). The table has columns for 'Install', 'Name', and 'Released'. The 'Install' column shows an 'Install' button. The 'Name' column contains the plugin name and links for 'External Site/Tool Integrations' and 'Build Reports'. The 'Released' column shows the release date '6 mo 29 days ago'. On the left sidebar, the 'Available plugins' section is selected, showing 25 updates. The bottom navigation bar includes 'Dashboard', 'Manage Jenkins', and 'Plugins'.



The screenshot shows the Jenkins 'Download progress' page. The left sidebar has 'Download progress' selected. The main content area shows the installation progress for the 'SonarQube Scanner' plugin. The progress is shown as a series of steps: 'Preparation' (checking internet connectivity, checking update center connectivity, success), 'SonarQube Scanner' (success), and 'Loading plugin extensions' (success). Below the progress bar, there are two links: 'Go back to the top page' (you can start using the installed plugins right away) and 'Restart Jenkins when installation is complete and no jobs are running'.

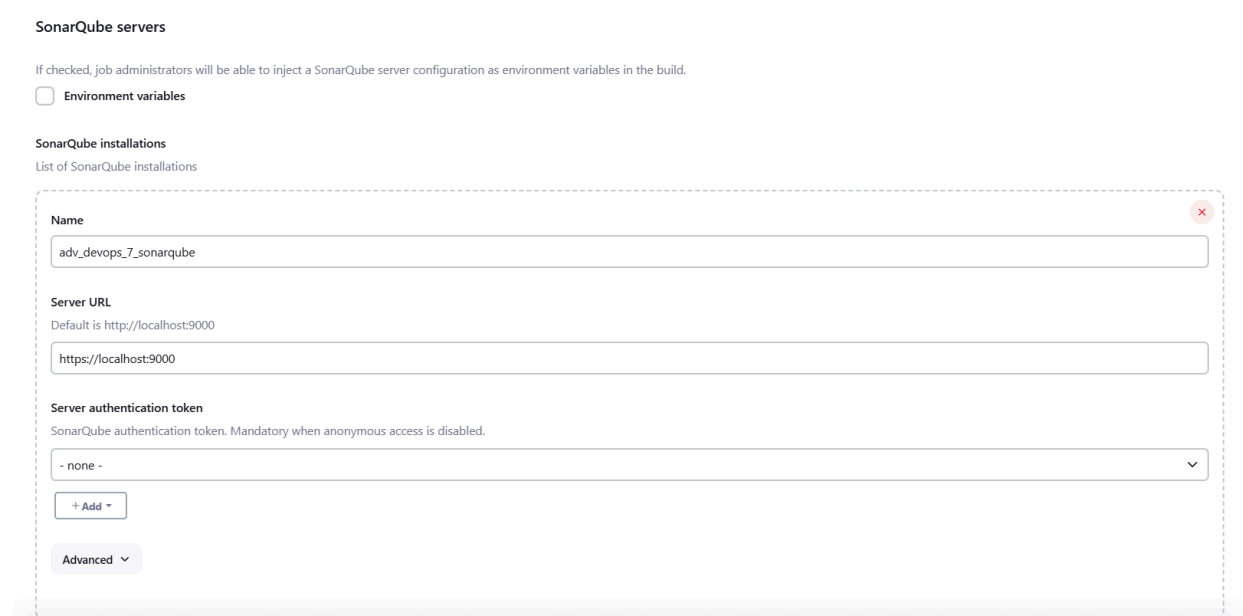
6. Under Jenkins 'Manage Jenkins' then go to 'system', scroll and look for **SonarQube Servers** and enter the details.

Enter the Server Authentication token if needed.

In SonarQube installations: Under **Name** add <project name of sonarqube> for me

adv_devops_7_sonarqube

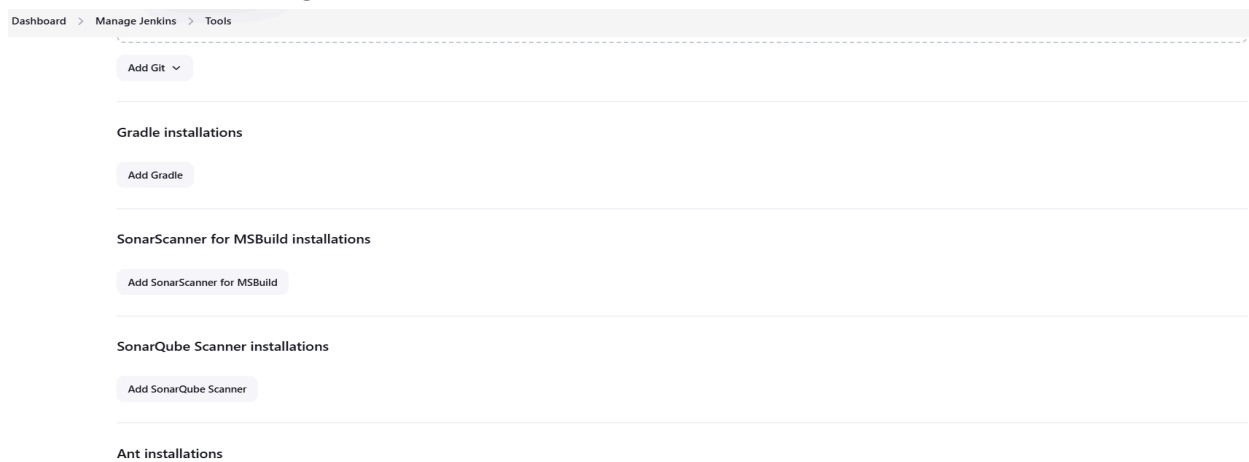
In **Server URL** Default is **http://localhost:9000**



The screenshot shows the Jenkins 'SonarQube servers' configuration page. The page has a title 'SonarQube servers' and a description: 'If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.' There is a checkbox labeled 'Environment variables'. Below this, there is a section 'SonarQube installations' with a description: 'List of SonarQube installations'. The main form area contains a table with columns 'Name', 'Server URL', and 'Server authentication token'. The 'Name' column has a text input field with the value 'adv_devops_7_sonarqube'. The 'Server URL' column has a text input field with the value 'https://localhost:9000'. The 'Server authentication token' column has a dropdown menu with the value 'none'. There is an 'Add' button and an 'Advanced' dropdown menu.

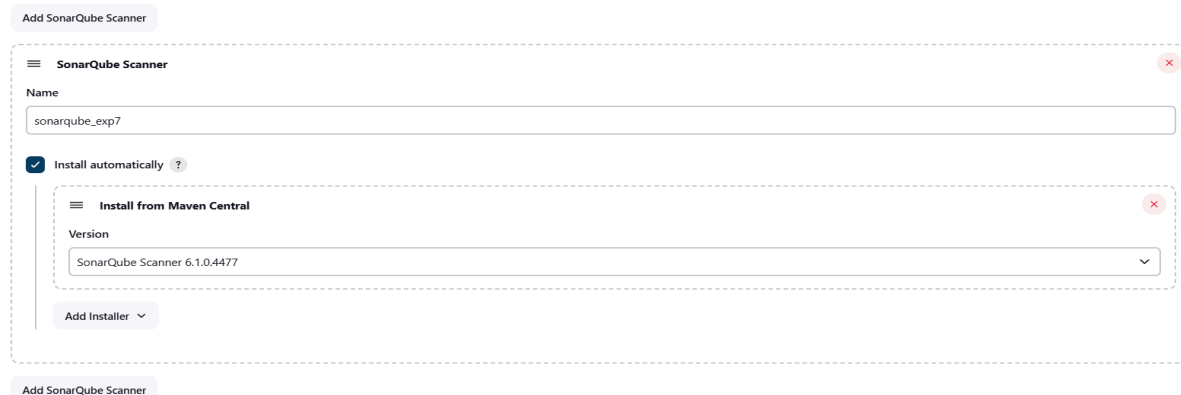
7. Search for SonarQube Scanner under Global Tool Configuration. Choose the latest configuration and choose Install automatically.

Dashboard > Manage Jenkins > Tools

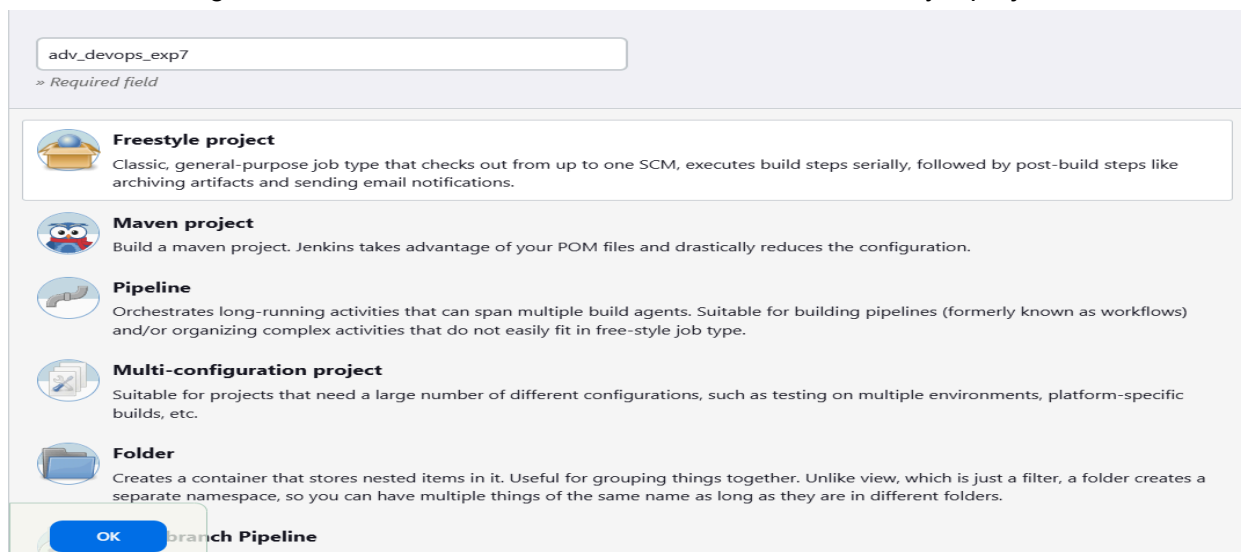


Check the “Install automatically” option. → Under name any name as identifier → Check the “Install automatically” option.

SonarQube Scanner installations



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



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.

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

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

Credentials ?

- none -

+ Add

Advanced

Add Repository

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.

Configure

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

Build Environment

Filter

- 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
- SonarScanner for MSBuild - End Analysis

Add build step

Post-build Actions

Execute SonarQube Scanner

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

Analysis properties ?

```

sonar.projectKey=adv_devops_7_sonarqube
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.sources=.

```

Additional arguments ?

JVM Options ?

Then save

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

adv_devops_exp7

SonarQube

Permalinks

- Last build (#2), 1 day 20 hr ago
- Last stable build (#2), 1 day 20 hr ago
- Last successful build (#2), 1 day 20 hr ago
- Last completed build (#2), 1 day 20 hr ago

Add description

Disable Project

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

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

Administration

Configuration Security Projects System Marketplace

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 ?
<div>sonar-administrators</div> <div>System administrators</div>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Quality Gates <input checked="" type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input checked="" type="checkbox"/> Projects
<div>sonar-users</div> <div>Every authenticated user automatically belongs to this group</div>	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/> Projects
<div>Anyone DEPRECATED</div> <div>Anybody who browses the application belongs to this group. If authentication is not enforced, assigned permissions also apply to non-authenticated users.</div>	<input type="checkbox"/>	<input type="checkbox"/> Quality Gates <input type="checkbox"/> Quality Profiles	<input type="checkbox"/>	<input type="checkbox"/> Projects
<div>A Administrator admin</div>	<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

IF CONSOLE OUTPUT FAILED:

Step 1: Generate a New Authentication Token in SonarQube

1. **Login to SonarQube:**
 - 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).
2. **Generate a New Token:**
 - 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**.
3. **Update the SonarQube Token:**
 - 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.

≡

Execute SonarQube Scanner

✕

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

Analysis properties ?

sonar.projectKey=adv_devops_7_sonarqube
sonar.host.url=http://localhost:9000
-Dsonar.login=sqp_568834b7b5e77a92843e4b3072e044643ce921c1
sonar.sources=.

Additional arguments ?

▼

JVM Options ?

▼

12. Run the Jenkins build.

The screenshot shows the Jenkins project page for 'adv_devops_exp7'. The left sidebar contains a list of actions: Status, Changes, Workspace, Build Now, Configure, Delete Project, SonarQube, and Rename. The main area displays the SonarQube logo and a 'Permalinks' section with a list of build links: Last build (#10), Last stable build (#10), Last successful build (#10), Last failed build (#8), Last unsuccessful build (#8), and Last completed build (#10). Below this is a 'Build History' table showing build #10 as successful on Sep 18, 2024, at 2:36 PM.

Check the console Output

The screenshot shows the Jenkins console output for build #10. The left sidebar includes Status, Changes, Console Output, View as plain text, Edit Build Information, Delete build '#10', and Timings. The main area displays the console output, which starts with 'Started by user unknown or anonymous' and 'Running as SYSTEM'. It then shows the build process: 'Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\adv_devops_exp7', 'The recommended git tool is: NONE', 'No credentials specified', and a series of git commands: 'git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\adv_devops_exp7\.git # timeout=10', 'git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject # timeout=10', 'git.exe --version # timeout=10', and 'git --version # 'git version 2.46.0.windows.1''. The output ends with 'Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject'.

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

The screenshot shows the SonarQube project page for 'adv_devops_7_sonarqube'. The top navigation bar includes 'sonarqube', 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', 'Administration', and 'More'. The project page shows the 'main' branch with a 'Passed' status. A message at the top says 'Don't let issues accumulate. Discover 'Clean as You Code!''. Below this, a 'Quality Gate' section shows a 'Passed' status with a warning icon and the text 'The last analysis has warnings. See details'. The bottom section shows a table with columns for 'New Code', 'Overall Code', 'Security', 'Reliability', and 'Maintainability'.

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

Conclusion:

This integration has successfully built a powerful, automated system for enhancing code security and quality. By continuously scanning for vulnerabilities, code smells, and other potential issues, it ensures proactive maintenance of code standards. The seamless connection to GitHub facilitates easy tracking of changes and instant feedback. Automated reports provide valuable insights, allowing developers to address problems early in the development cycle. This streamlined process enhances both efficiency and security. As a result, the workflow becomes more reliable, with improved overall code integrity. Continuous improvement is ensured through consistent monitoring and analysis.