

ADVANCE DEVOPS EXP 7

Niraj S. Kothawade
D15A - 24

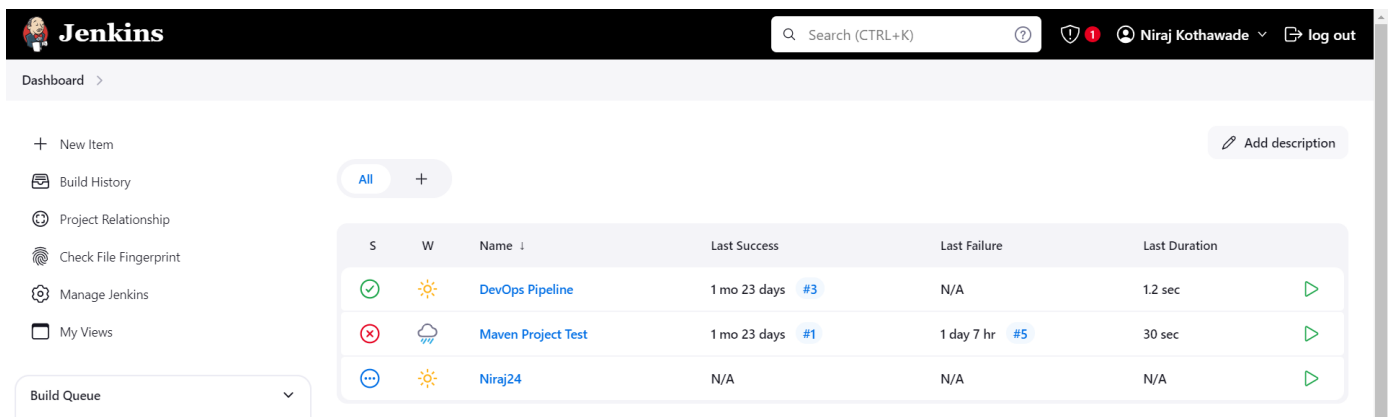
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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. The top navigation bar includes the Jenkins logo, a search bar, and user information (Niraj Kothawade). The left sidebar contains links to 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main content area displays a table of build history. The table has columns for 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. Three builds are listed: 'DevOps Pipeline' (Success, 1.2 sec), 'Maven Project Test' (Failure, 30 sec), and 'Niraj24' (Success, N/A).

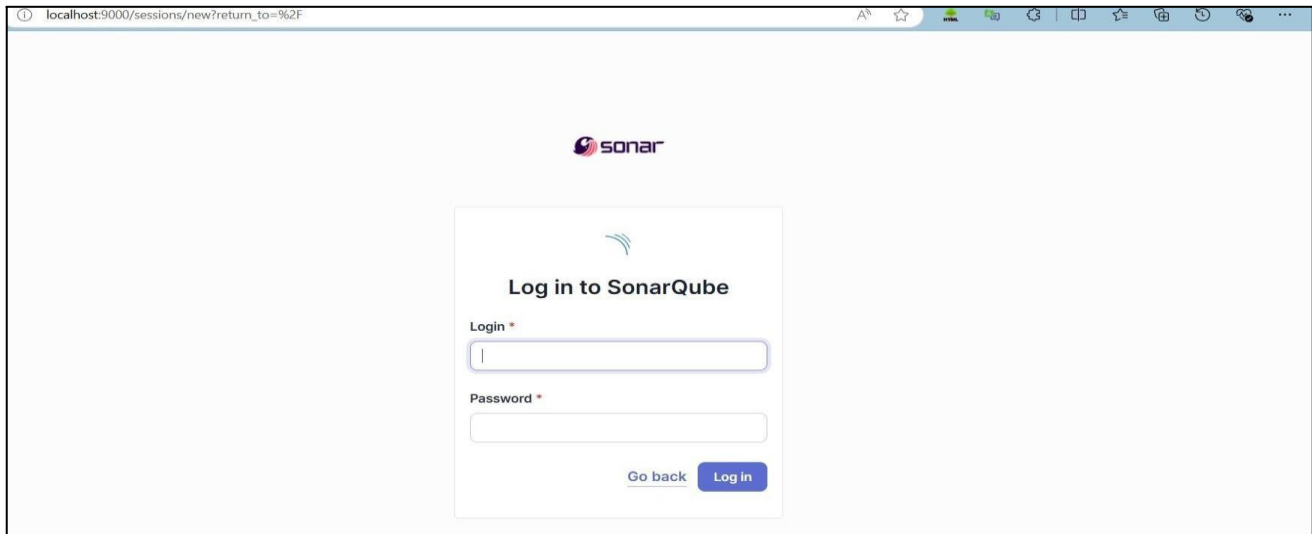
S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	DevOps Pipeline	1 mo 23 days #3	N/A	1.2 sec
✗	☁	Maven Project Test	1 mo 23 days #1	1 day 7 hr #5	30 sec
...	☀	Niraj24	N/A	N/A	N/A

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

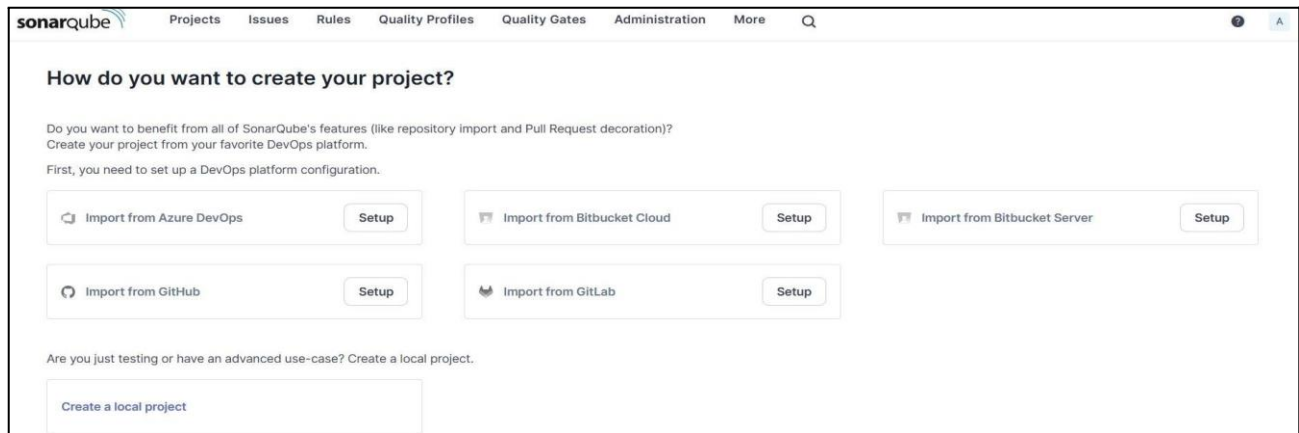
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest

```
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
```

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



- Login to SonarQube using username admin and password admin.



- 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. 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

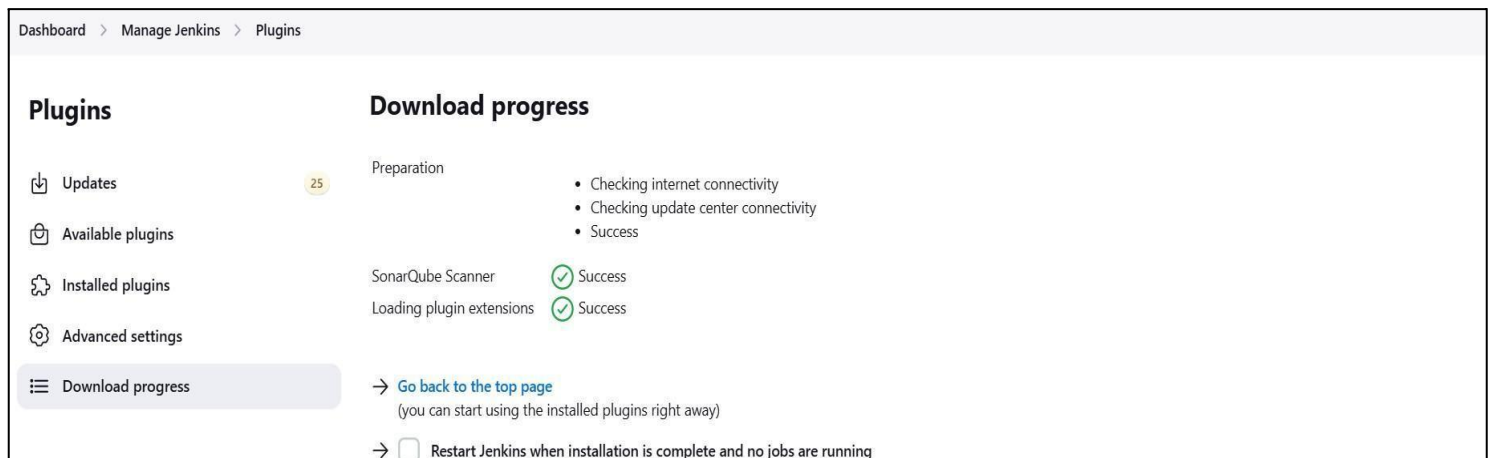
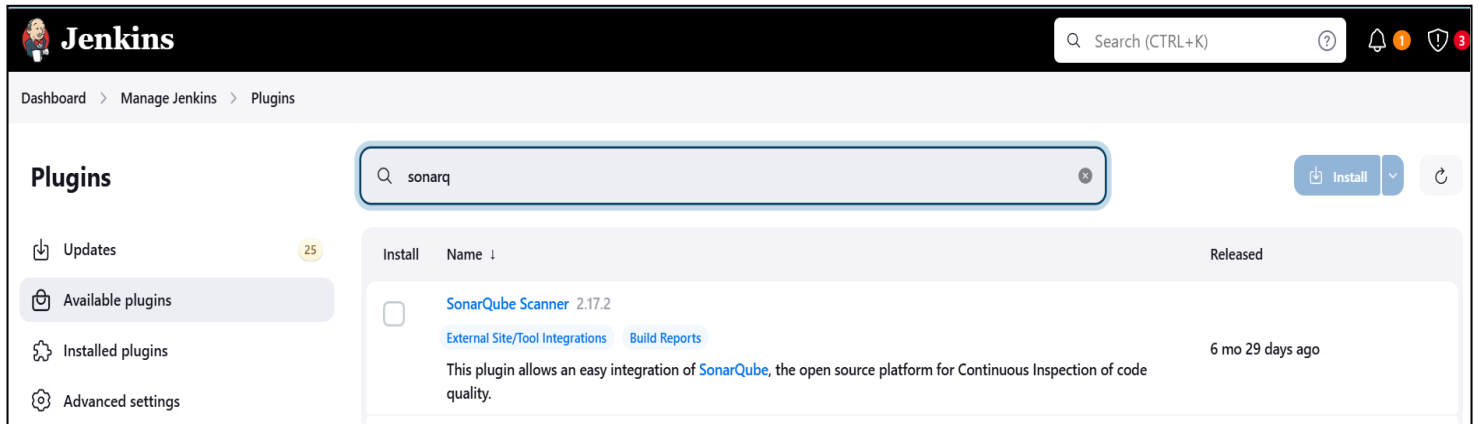
Any code that has changed since the previous version is considered new code. Recommended for projects following regular versions or releases.

☐ 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.



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>

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

☐ Environment variables

SonarQube installations

List of SonarQube installations

Name

adv_devops_7_sonarqube

Server URL

Default is `http://localhost:9000`

`https://localhost:9000`

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add

Advanced

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

Add Git ▾

Gradle installations

Add Gradle

SonarScanner for MSBuild installations

Add SonarScanner for MSBuild

SonarQube Scanner installations

Add SonarQube Scanner

Ant installations

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

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner ✕

Name

sonarqube_exp7

☒ Install automatically ?

☰ Install from Maven Central ✕

Version


SonarQube Scanner 6.1.0.4477 ▾


Add Installer ▾


Add SonarQube Scanner


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


» Required field

**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.

**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK


branch Pipeline

Customize list of Pipeline projects according to detected branches in one SCM repository.

9. Choose this GitHub repository in Source Code Management.

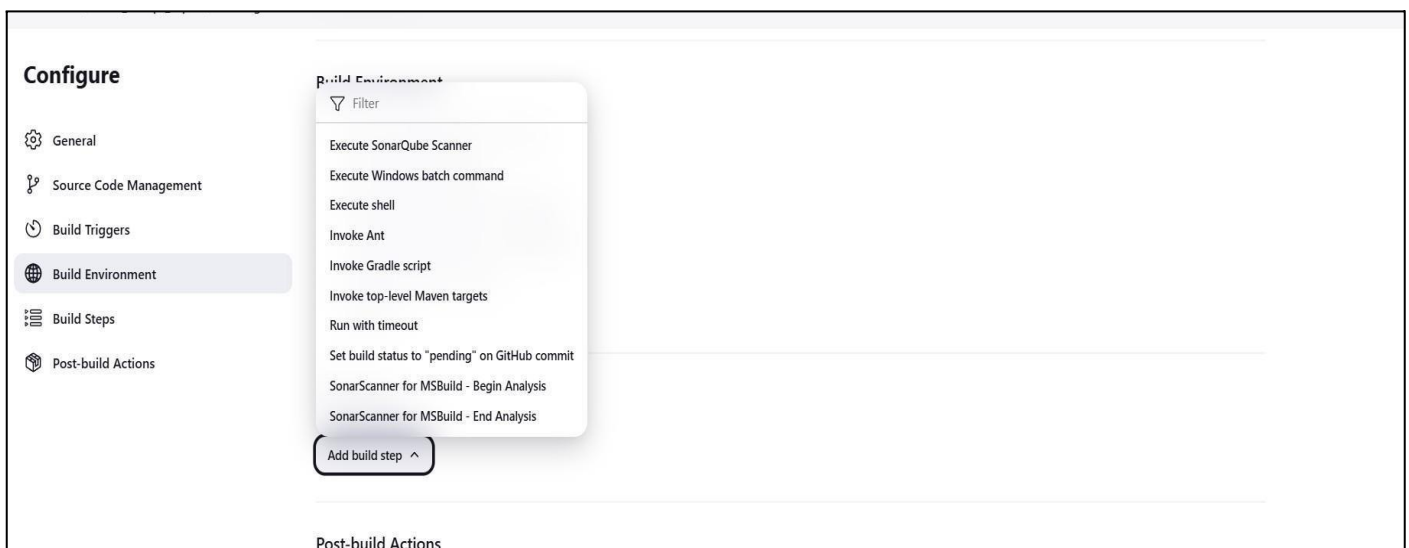
https://github.com/shazforiot/MSBuild_firstproject.git

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 section for 'Repositories'. Inside this section, there is a 'Repository URL' field containing the URL 'https://github.com/shazforiot/MSBuild_firstproject.git'. Below the URL field is a 'Credentials' dropdown menu showing '- none -'. There is a '+ Add' button and an 'Advanced' dropdown menu. At the bottom of the configuration area 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' interface with a sidebar on the left containing a list of configuration categories: General, Source Code Management, Build Triggers, Build Environment (which is highlighted), Build Steps, and Post-build Actions. A 'Build Environment' dropdown menu is open, showing 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 dropdown is an 'Add build step' button.

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

adv_devops_exp7

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

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

Configuration ▾ Security ▾ Projects ▾ System Marketplace				
	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

IF CONSOLE OUTPUT FAILED:

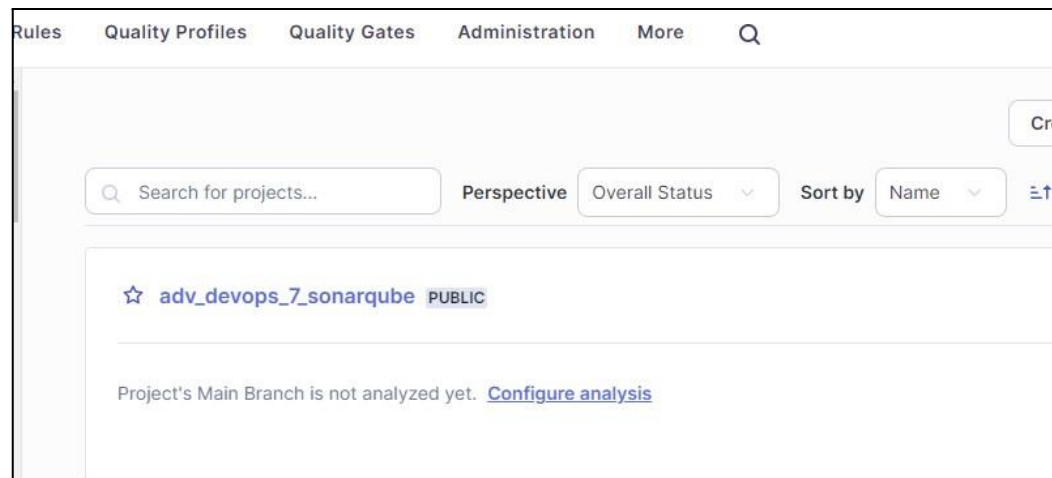
Step 1: Generate a New Authentication Token in SonarQube

1. Login to SonarQube:

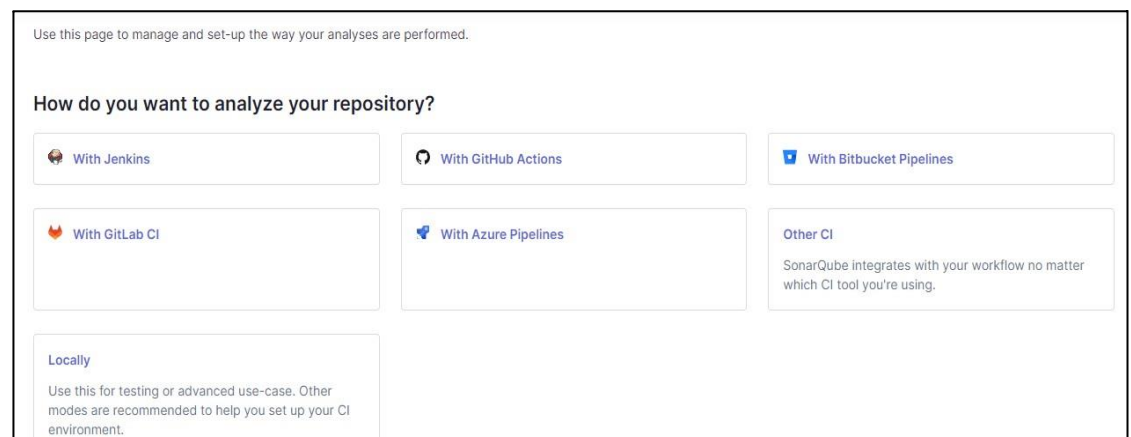
- Open your browser and go to **http://localhost:9000**.
- Log in with your admin credentials (default username is **admin**, and the password is either **admin** or your custom password if it was changed).

2. Generate a New Token:

- Go to the project that you have created on SonarQube.



- Click on **Locally**



- Further, Generate a Project token with the following details and click on generate.

1 Provide a token

Generate a project token

Use existing token

Token name ?

adv_devops_7_sonarqube

Expires in

1 year

Generate

Please note that this token will only allow you to analyze the current project. If you want to use the same token to analyze multiple projects, you need to generate a global token in your [user account](#). See the [documentation](#) for more information.

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

- Copy the token you get here and save it securely as we would need it in Jenkins.

1 Provide a token

adv_devops_7_sonarqube": sqp_bfa5258ea4fd254f00c3d1d4e64205ebefcdd027

The token is used to identify you when an analysis is performed. If it has been compromised, you can revoke it at any point in time in your [user account](#).

Continue

Step 2: Update the Token in Jenkins

1. Go to Jenkins Dashboard:

- Open Jenkins and log in with your credentials.

Jenkins

Search (CTRL+K)

1

Niraj Kothawade

log out

Dashboard

+ New Item

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

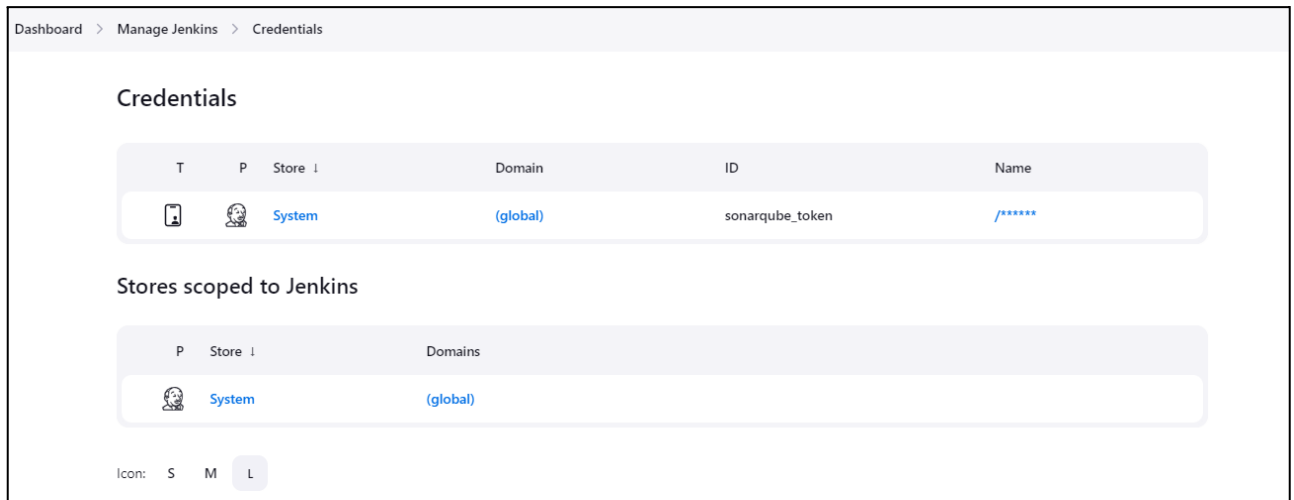
All

+

S	W	Name	Last Success	Last Failure	Last Duration
		DevOps Pipeline	1 mo 23 days #3	N/A	1.2 sec
		Maven Project Test	1 mo 23 days #1	1 day 7 hr #5	30 sec
		Niraj24	N/A	N/A	N/A

Add description

2. Go to Dashboard—>Manage Jenkins—>Credentials



3. Click on **global** under the domains part of Stores scoped to Jenkins section. Further click on add credentials. Proceed with the following details. Make sure to copy the token generated earlier in sonarqube and give any suitable name as the ID.

The screenshot shows the 'Add Credentials' form in Jenkins. The 'Kind' dropdown is set to 'Secret text'. The 'Scope' dropdown is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Secret' field is masked with dots. The 'ID' field contains 'sonarqube-exp7'. The 'Description' field contains 'advance devops exp7'. A blue 'Create' button is at the bottom left.

4. After clicking on create we see that the given token has been added in Jenkins credentials.



5. Now go to **Manage Jenkins**—>**System**—>**SonarQube servers** and proceed with the following details. Reference the authentication token generated in the previous step.

SonarQube servers

If checked, job administrators will be able to inject a SonarQube server configuration as environment variables in the build.

☒ Environment variables

SonarQube installations

List of SonarQube installations

Name
adv_devops_7_sonarqube

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

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.
advance devops exp7

[+ Add](#)

6. Check the SonarQube Scanner Environment and add the server authentication token

Build Environment

☐ Delete workspace before build starts

☐ Use secret text(s) or file(s) ?

☐ Add timestamps to the Console Output

☐ Inspect build log for published build scans

☒ Prepare SonarQube Scanner environment ?

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled. Will default to the one defined in the SonarQube installation.

advance devops exp7

[+ Add](#)

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.

Dashboard > adv_devops_exp7 >

Status

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

Build History

trend

Filter...

#6

Sep 25, 2024, 10:04 PM

adv_devops_exp7

SonarQube

SonarQube Quality Gate

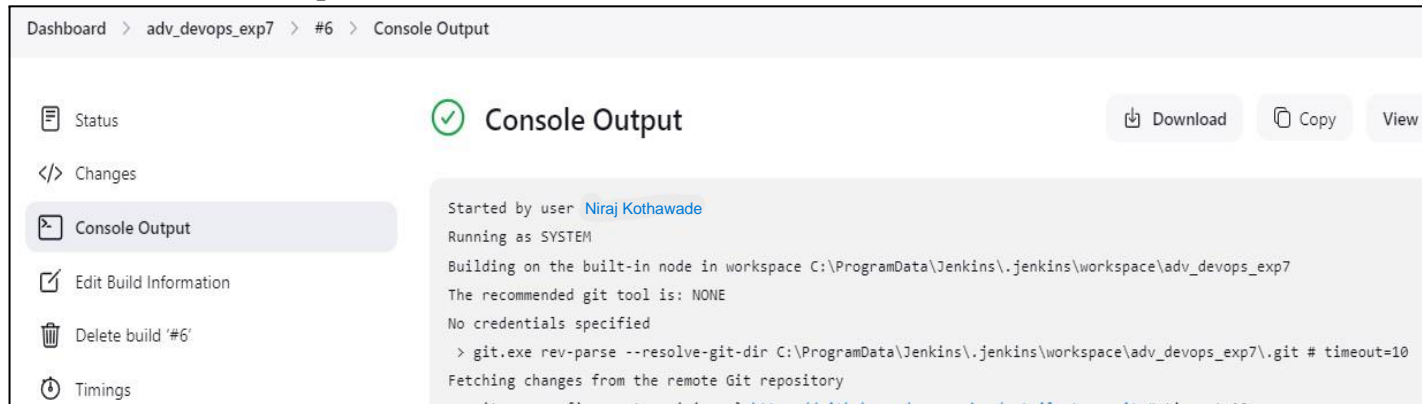
adv_devops_7_sonarqube **Passed**

server-side processing: **Success**

Permalinks

- Last build (#6), 1 min 55 sec ago
- Last stable build (#6), 1 min 55 sec ago
- Last successful build (#6), 1 min 55 sec ago
- Last failed build (#5), 17 min ago
- Last unsuccessful build (#5), 17 min ago
- Last completed build (#6), 1 min 55 sec ago

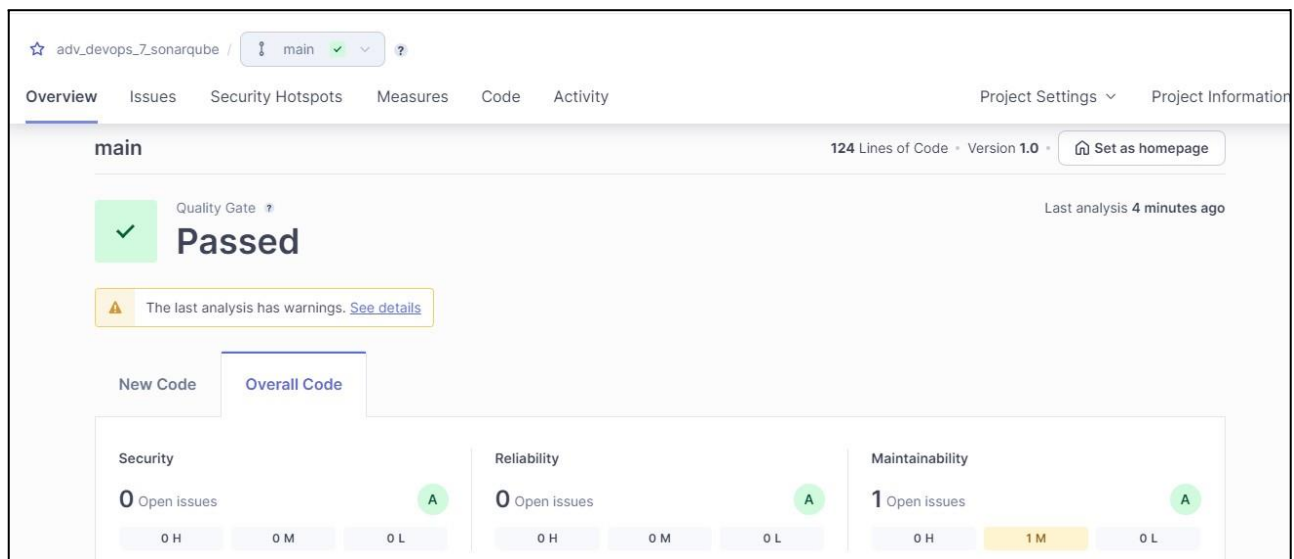
Check the console Output



The screenshot shows the Jenkins web interface for a build named 'adv_devops_exp7' with build number '#6'. The 'Console Output' tab is selected, displaying a green checkmark and the title 'Console Output'. On the left sidebar, there are links for 'Status', 'Changes', 'Console Output' (selected), 'Edit Build Information', 'Delete build '#6'', and 'Timings'. The console output text is as follows:

```
Started by user Niraj Kothawade
Running as SYSTEM
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
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\adv_devops_exp7\.git # timeout=10
Fetching changes from the remote Git repository
```

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



The screenshot shows the SonarQube project overview for 'adv_devops_7_sonarqube' on the 'main' branch. The 'Overview' tab is active. The project has '124 Lines of Code' and 'Version 1.0'. A 'Quality Gate' is shown with a green checkmark and the word 'Passed'. A warning message states: 'The last analysis has warnings. See details'. Below this, there are tabs for 'New Code' and 'Overall Code'. The 'Overall Code' tab is selected, showing three metrics: 'Security' with '0 Open issues' and grade 'A', 'Reliability' with '0 Open issues' and grade 'A', and 'Maintainability' with '1 Open issues' and grade 'A'. Each metric has a horizontal bar chart with '0 H', '0 M', and '0 L' markers. The 'Maintainability' bar has a yellow segment labeled '1 M'.

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