

ADVANCE DEVOPS EXP 7

Name : **Devansh Wadhwani**

Class : **D15A**

Roll No. : **64**

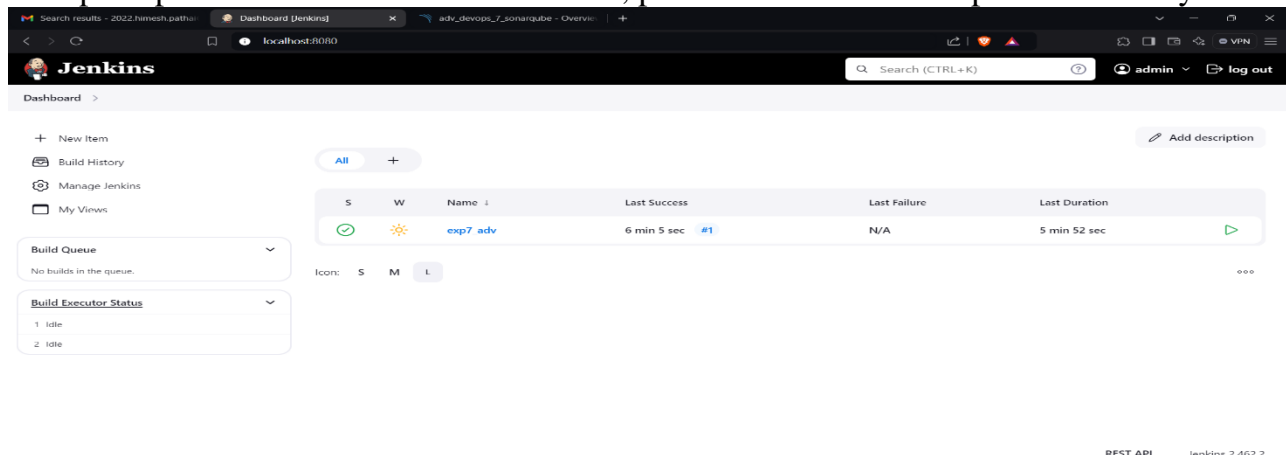
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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

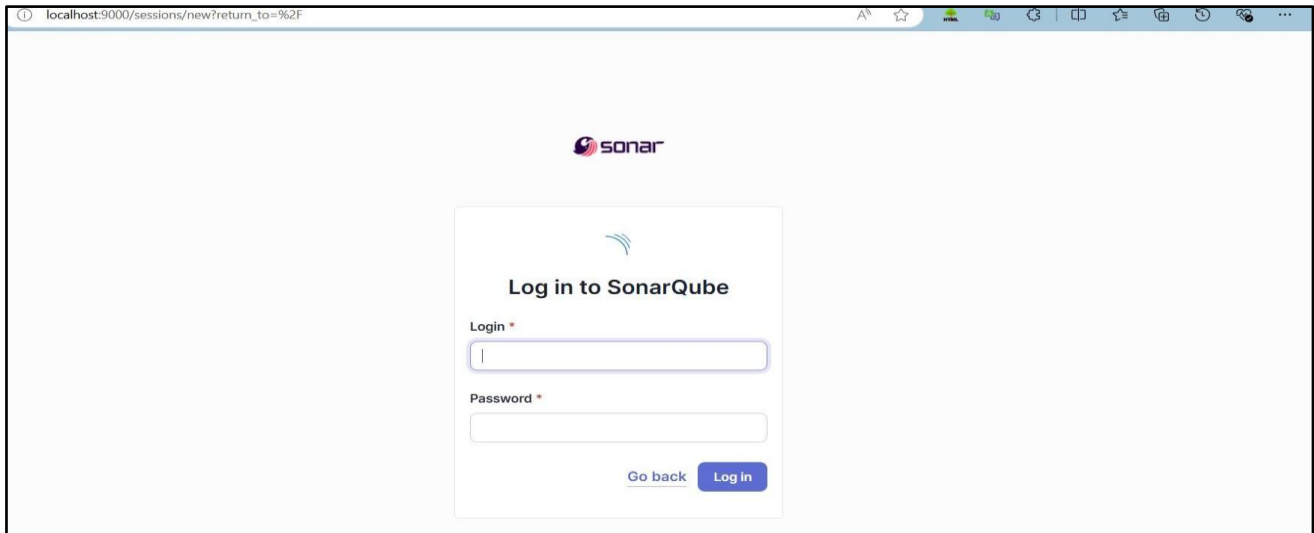
sonarqube:latest

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

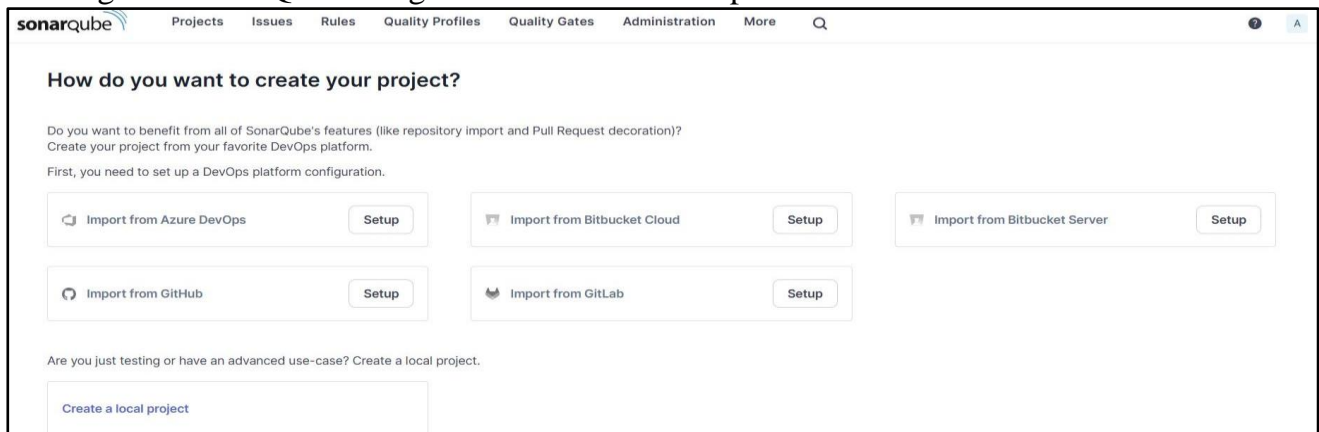
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:
latest
docker: error during connect: Head "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/_ping": open //./pipe/dockerDesktop
LinuxEngine: The system cannot find the file specified.
See 'docker run --help'.
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
ea1a6dd7f7949aff3c5e0545da8b7f0dc95dc0090110c7361e9279101cf4e81c
PS C:\WINDOWS\system32>
```

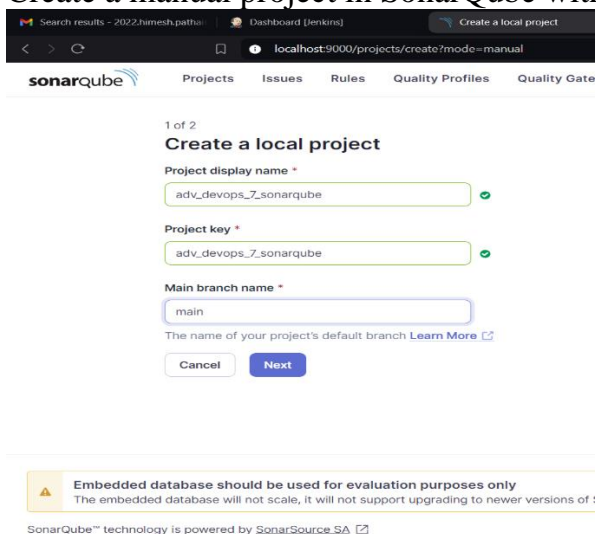
3. Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



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

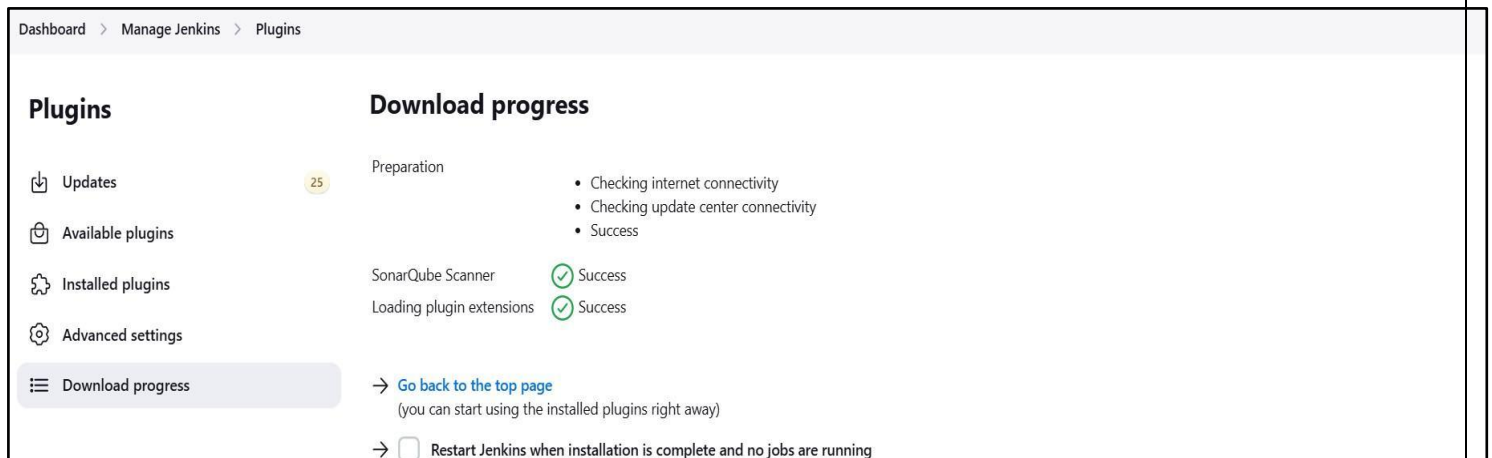
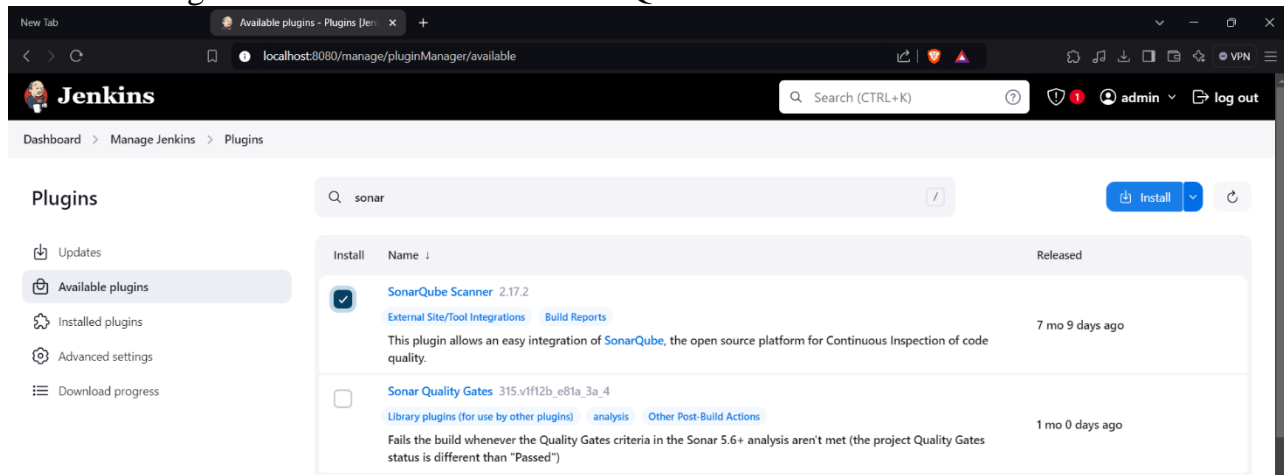


5. Create a manual project in SonarQube with the name sonarqube



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: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

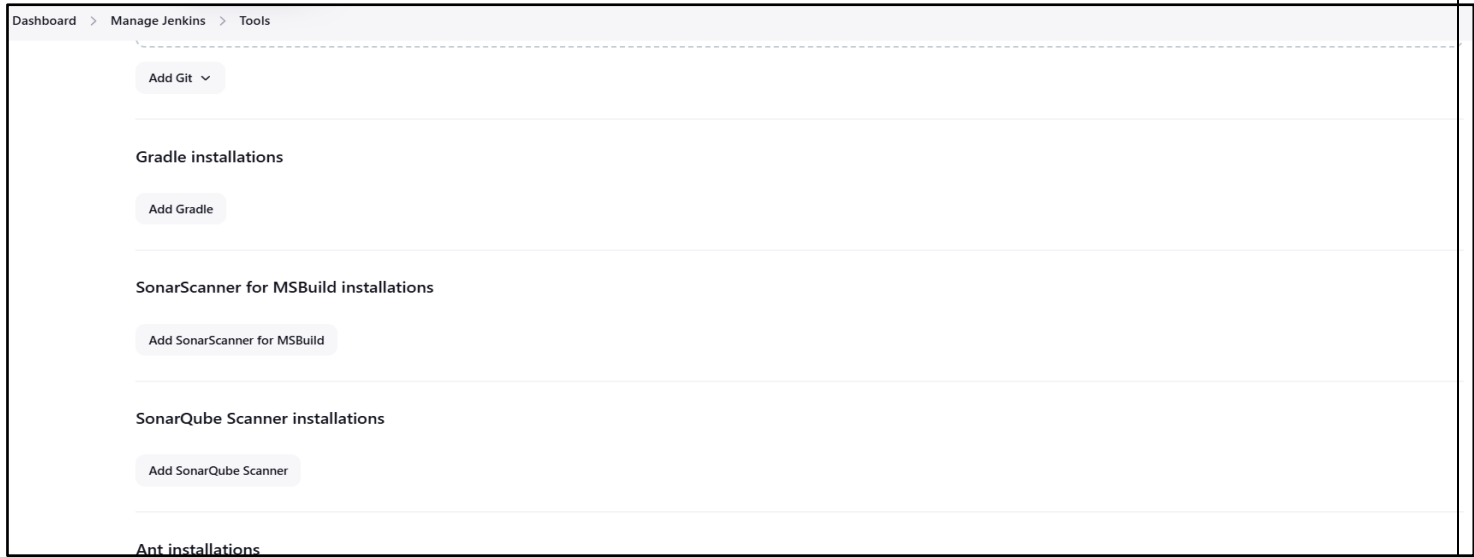
SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add

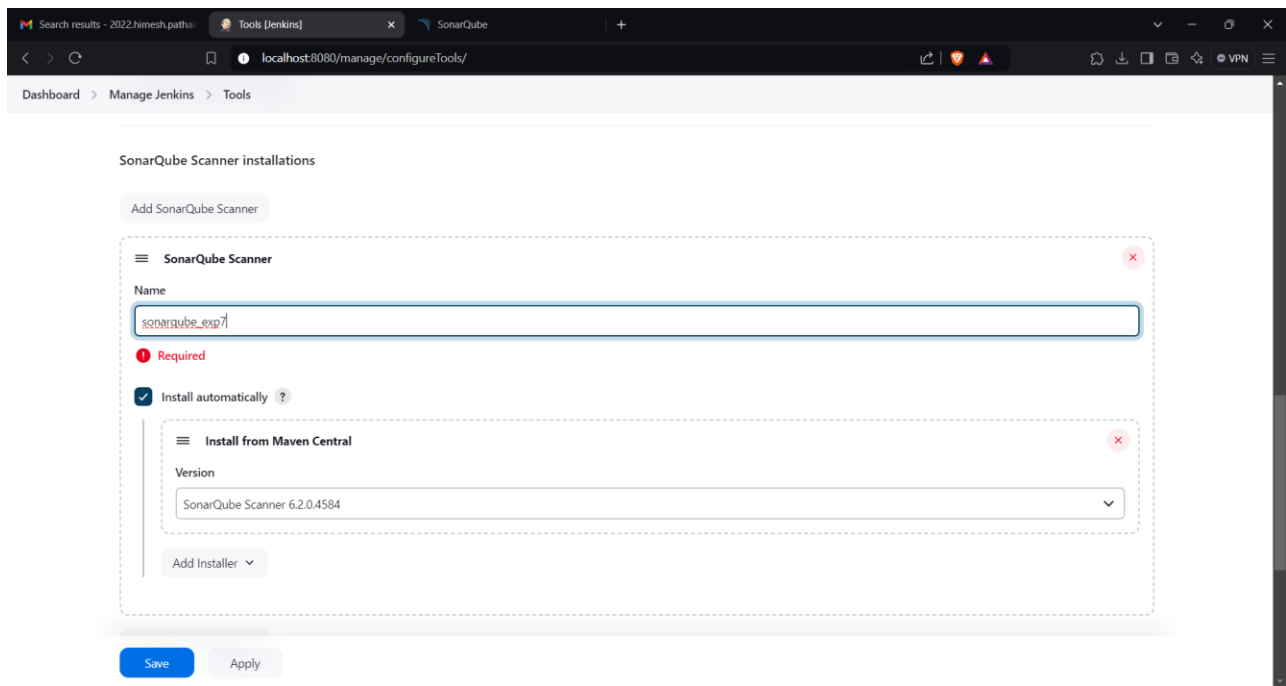
Advanced

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



Dashboard > Manage Jenkins > Tools

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.

Search results - 2022.himesh.pathi | New Item [Jenkins] | adv_devops_7_sonargube - Overview | +

localhost:8080/view/all/newJob

Jenkins

Search (CTRL+K) | admin | log out

Dashboard > All > New Item

New Item

Enter an item name

exp7_adv

» A job already exists with the name 'exp7_adv'

Select an item type

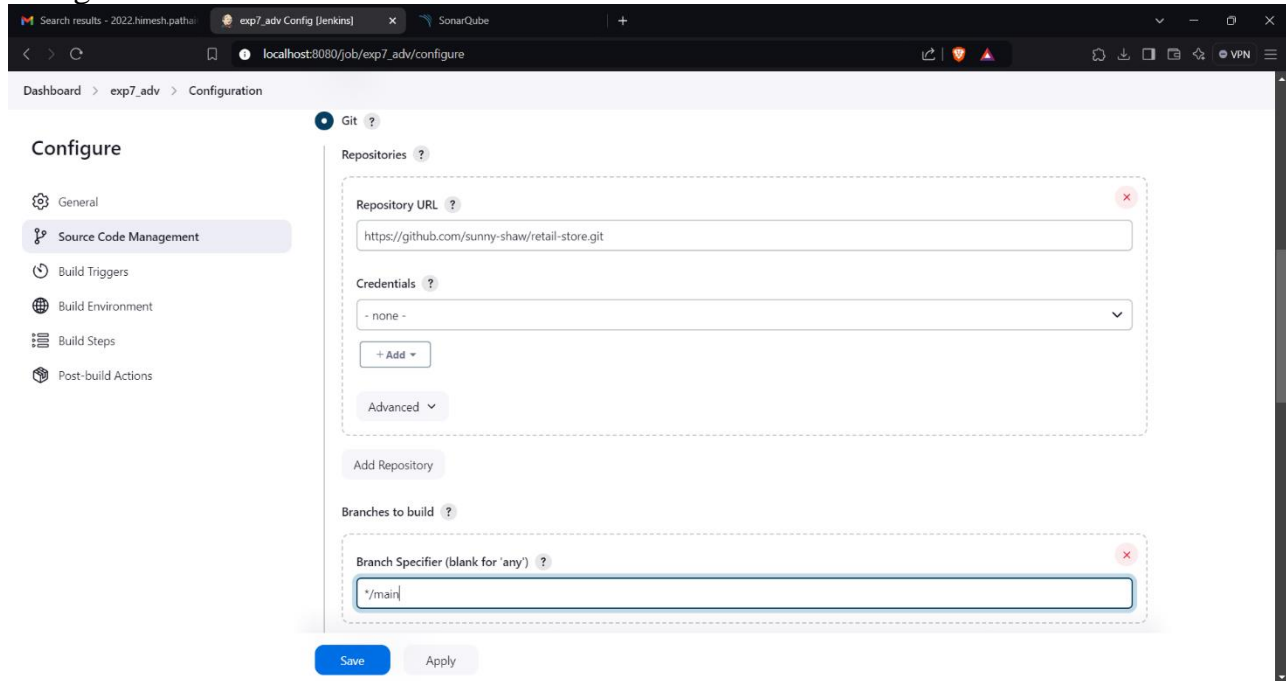
- Freestyle project**
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
- Pipeline**
Orchestrates long-running activities that can span multiple build agents. Suitable 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

OK

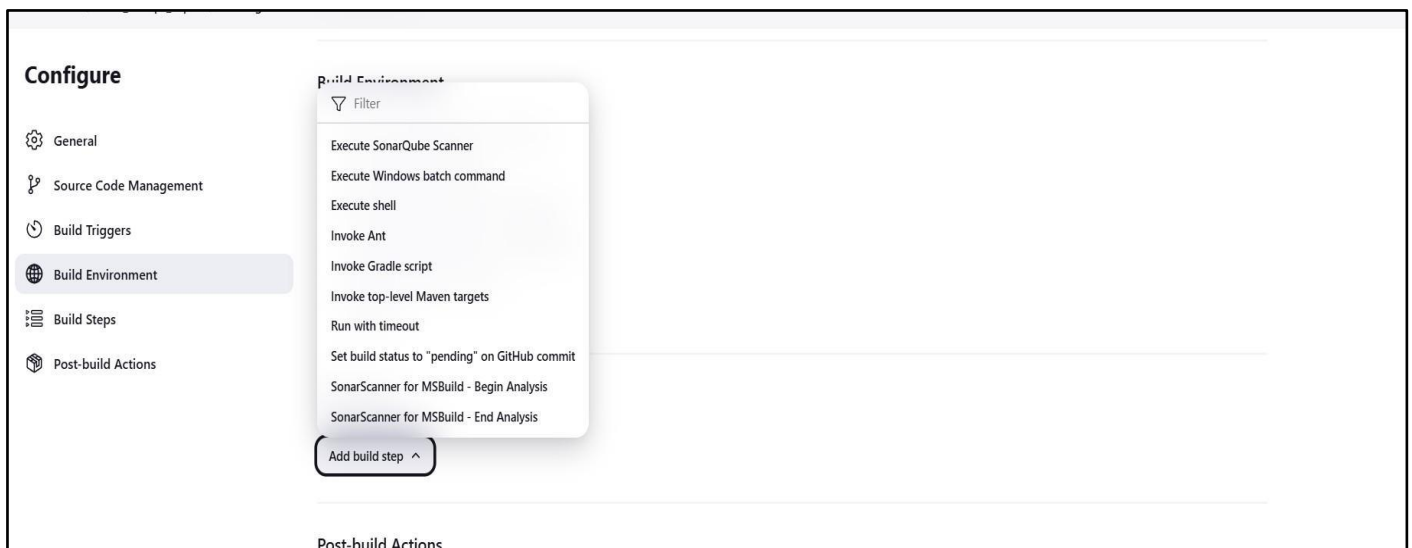
9. Choose this GitHub repository in Source Code Management.

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

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



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.



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

Search results - 2022.himesh.pathi

exp7_adv [jenkins]

SonarQube

localhost:8080/job/exp7_adv/

Jenkins

Search (CTRL+K)

admin

log out

Dashboard > exp7_adv >

Status

Changes

Workspace

Build Now

Configure

Delete Project

SonarQube

Rename

exp7_adv

Add description

SonarQube

Permalinks

Build History

trend

Filter...

#1

Sep 26, 2024, 11:07 AM

Atom feed for all

Atom feed for failures

localhost:8080/job/exp7_adv/#

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

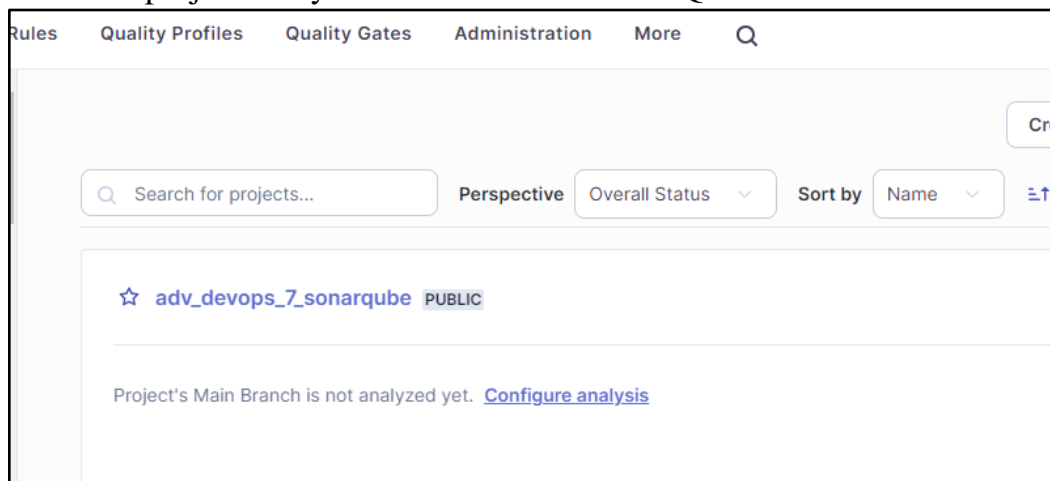
Step 1: Generate a New Authentication Token in SonarQube

1. Login to SonarQube:

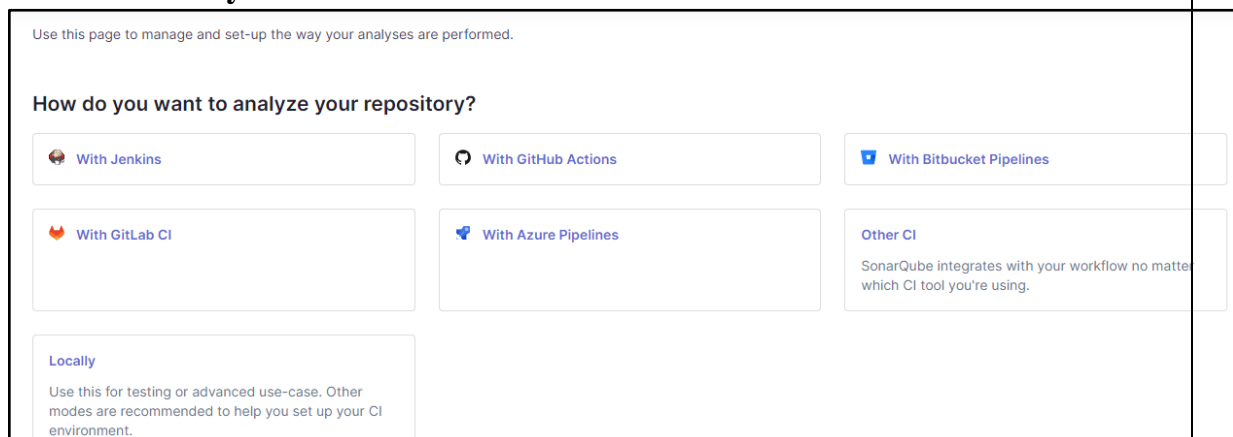
- Open your browser and go to **http://localhost:9000**.
- Log in with your admin credentials (default username is **admin**, and the password is either **admin** or your custom password if it was changed).

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 tokenUse existing token

Token name ?

Expires in

"adv_devops_7_sonarqube"

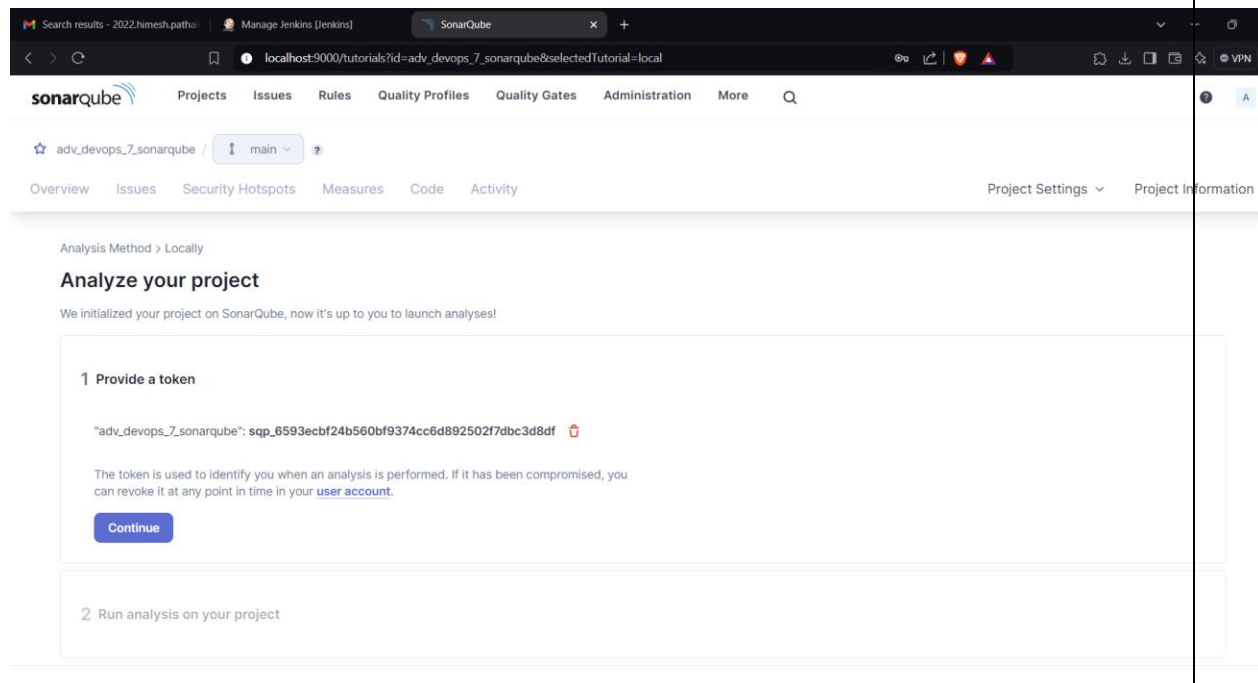
1 year

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.

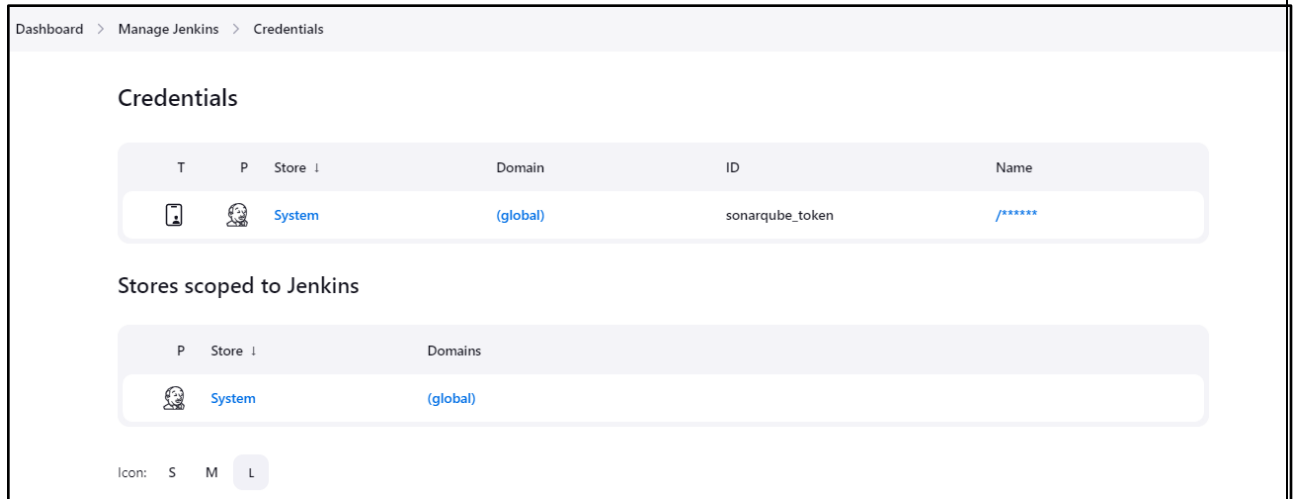


Step 2: Update the Token in Jenkins

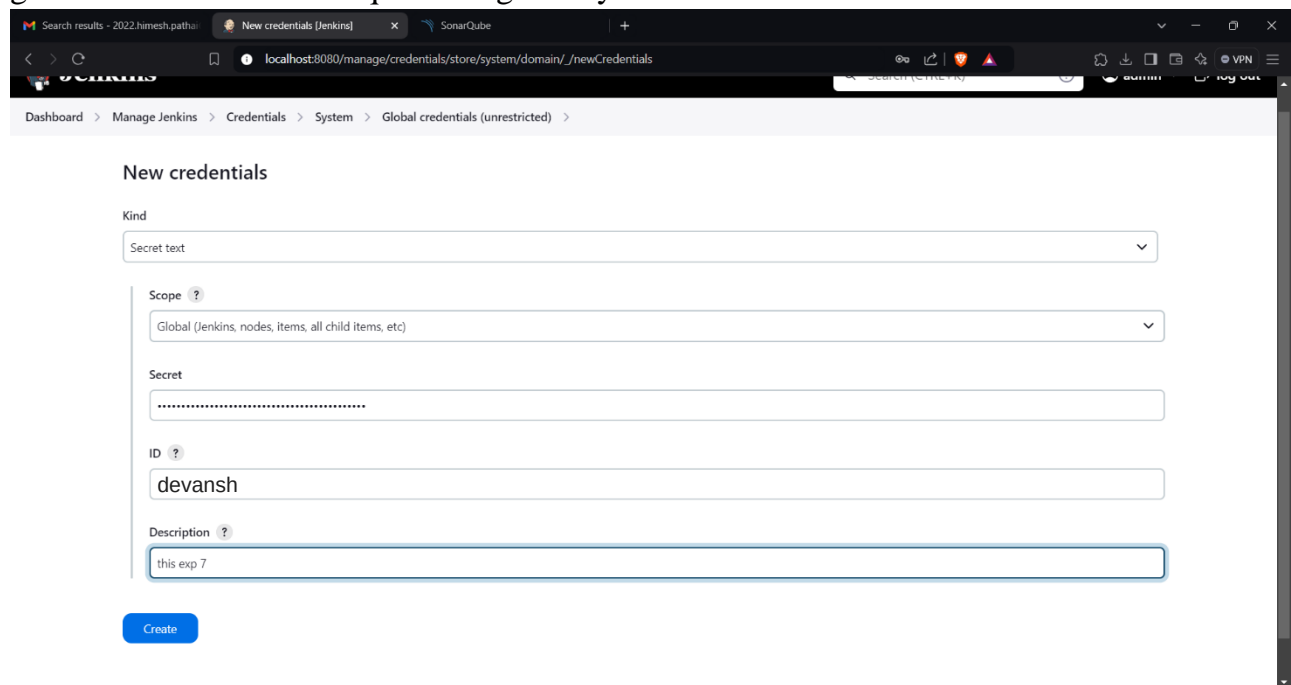
1. Go to Jenkins Dashboard:

- Open Jenkins and log in with your credentials.

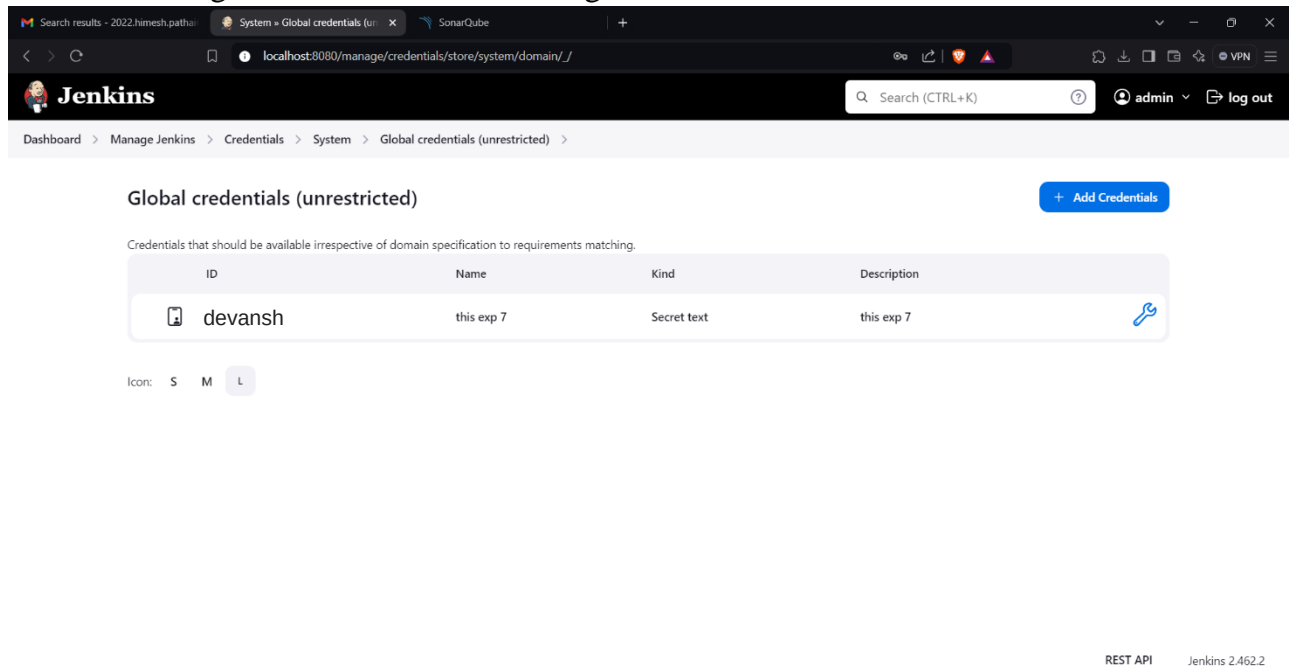
2. Go to Dashboard—>Manage Jenkins—>Credentials



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.



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

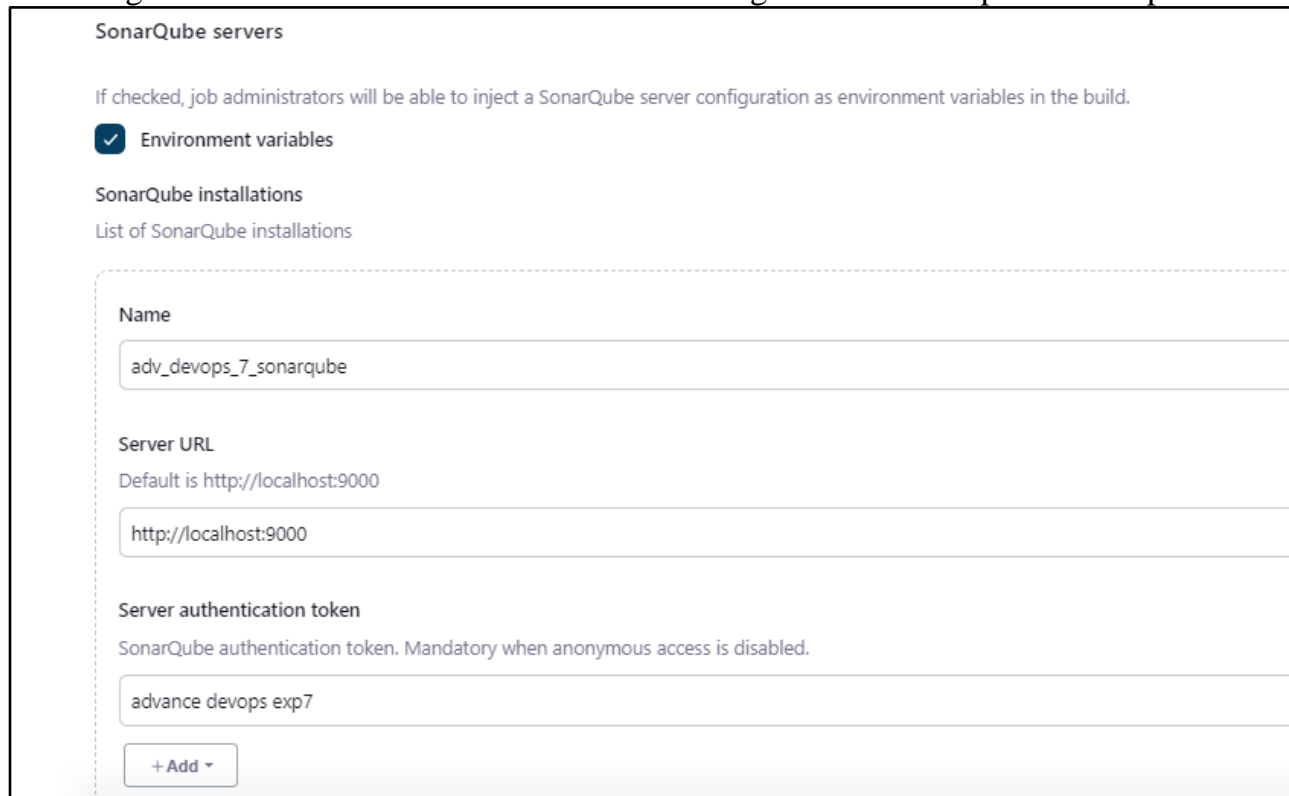


The screenshot shows the Jenkins web interface. The breadcrumb navigation is: Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted). The page title is "Global credentials (unrestricted)". Below the title is a blue button labeled "+ Add Credentials". A sub-header reads: "Credentials that should be available irrespective of domain specification to requirements matching." Below this is a table with the following data:

ID	Name	Kind	Description
devansh	this exp 7	Secret text	this exp 7

At the bottom of the table is a blue key icon. Below the table are icons for "Icon: S M L". In the bottom right corner, it says "REST API" and "Jenkins 2.462.2".

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



The screenshot shows the "SonarQube servers" configuration page. It includes a checkbox for "Environment variables" which is checked. Below this is the "SonarQube installations" section with the sub-header "List of SonarQube installations". There is a dashed box containing the following fields:

- Name:** A text input field containing "adv_devops_7_sonarqube".
- Server URL:** A text input field containing "http://localhost:9000". Above the field, it says "Default is http://localhost:9000".
- Server authentication token:** A text input field containing "advance devops exp7". Above the field, it says "SonarQube authentication token. Mandatory when anonymous access is disabled."

At the bottom of the dashed box is a button labeled "+ 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 ?

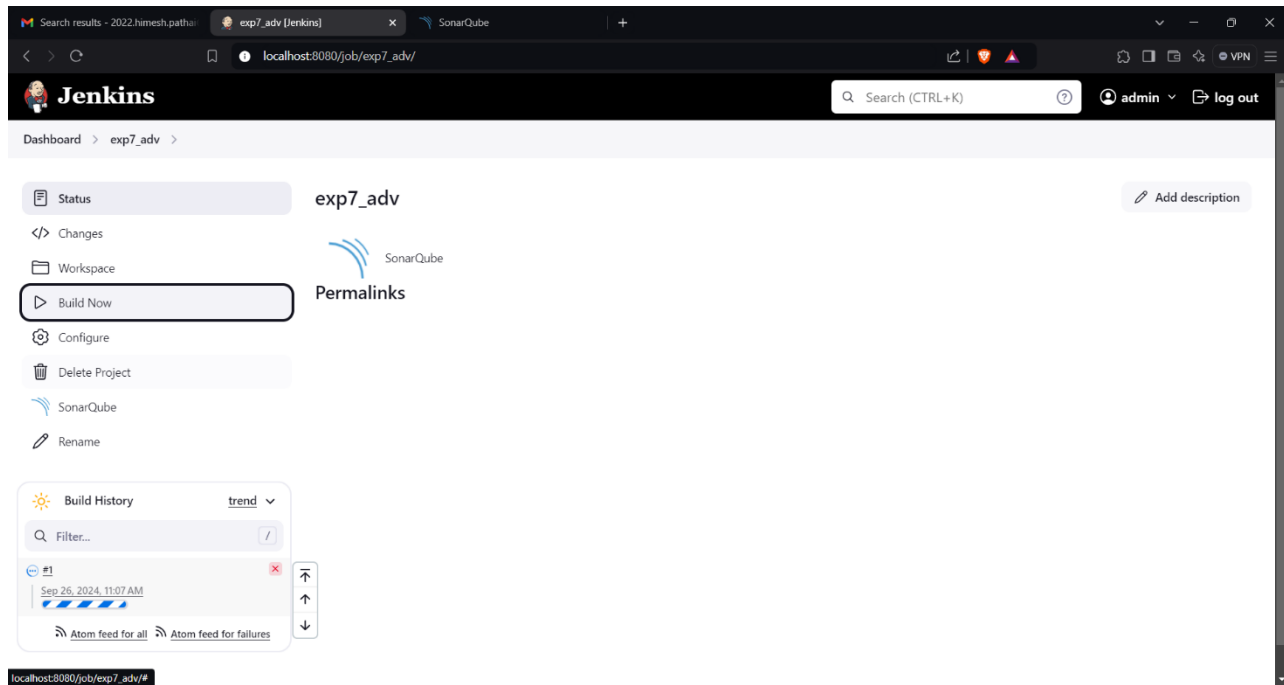
Analysis properties ?

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

Additional arguments ?

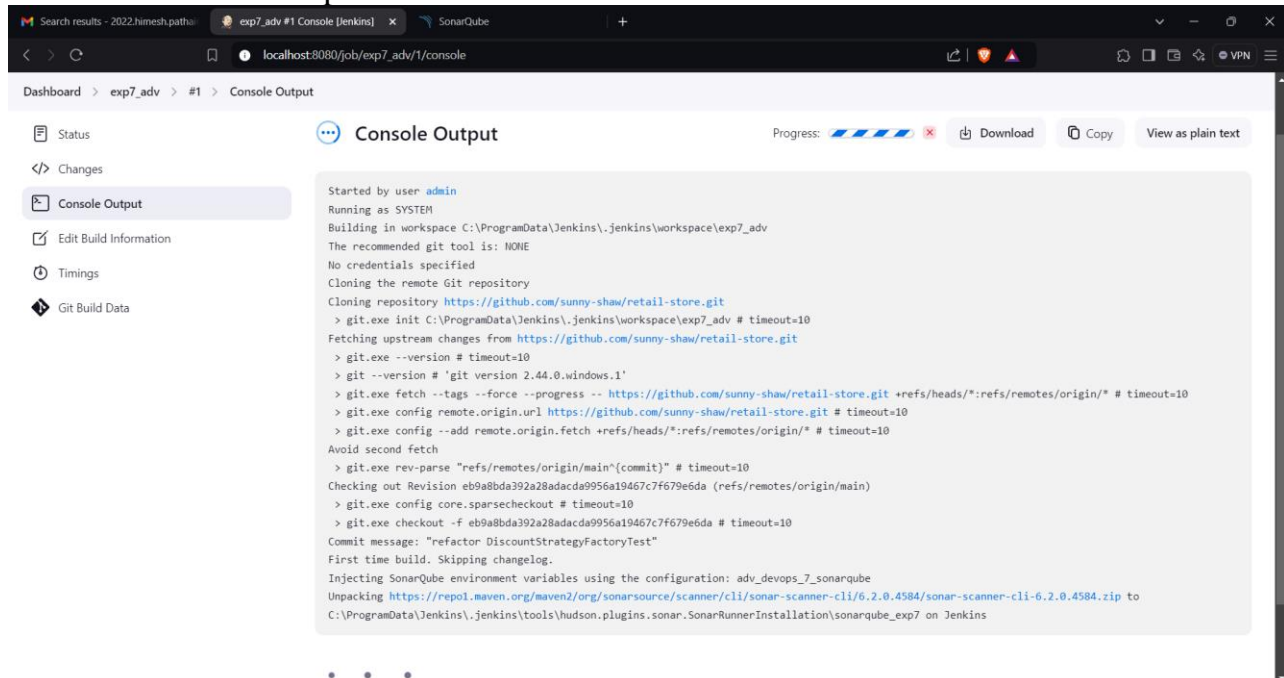
JVM Options ?

12.Run the Jenkins build.



The screenshot shows the Jenkins web interface for a job named 'exp7_adv'. The left sidebar contains a list of actions: Status, Changes, Workspace, Build Now (highlighted), Configure, Delete Project, SonarQube, and Rename. The main content area displays the job name 'exp7_adv' with an 'Add description' link. Below this is a 'Permalinks' section with a SonarQube icon. The 'Build History' section shows a single build from Sep 26, 2024, 11:07 AM, with a progress bar and links for 'Atom feed for all' and 'Atom feed for failures'.

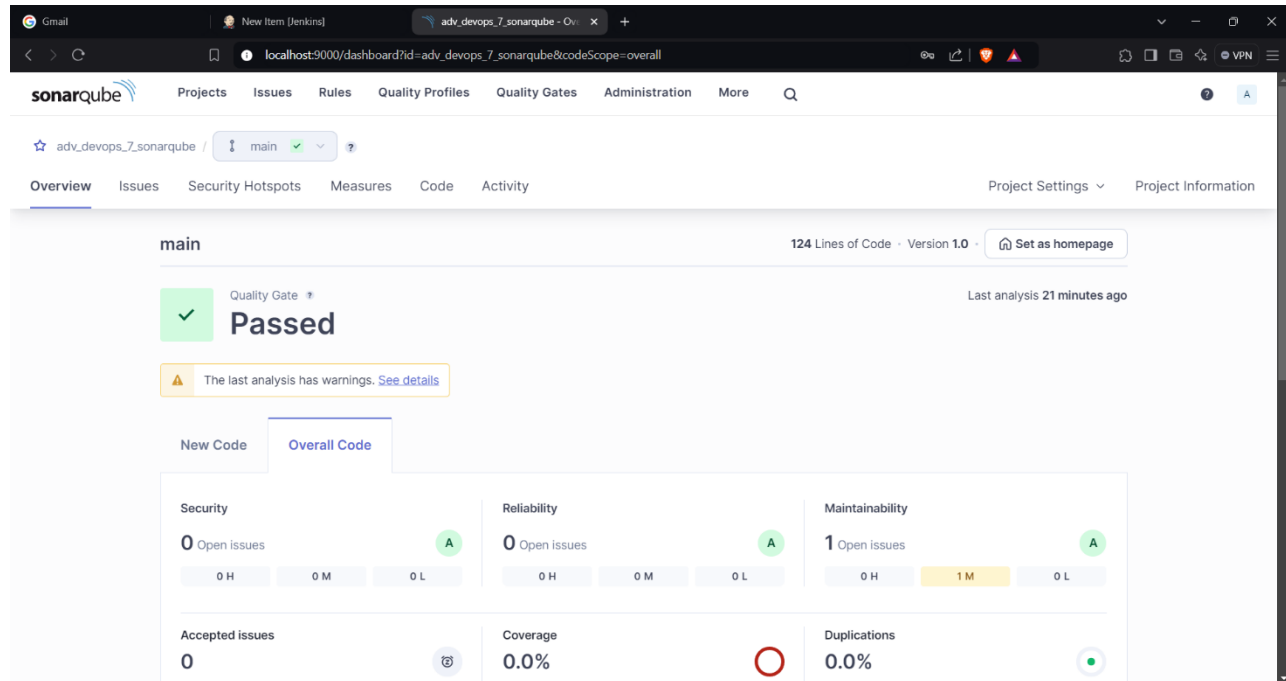
Check the console Output



The screenshot shows the Jenkins web interface for the 'exp7_adv' job, specifically the 'Console Output' view. The left sidebar contains a list of actions: Status, Changes, Console Output (highlighted), Edit Build Information, Timings, and Git Build Data. The main content area displays the console output, which includes the following text:

```
Started by user admin
Running as SYSTEM
Building in workspace C:\ProgramData\jenkins\workspace\exp7_adv
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/sunny-shaw/retail-store.git
> git.exe init C:\ProgramData\jenkins\workspace\exp7_adv # timeout=10
Fetching upstream changes from https://github.com/sunny-shaw/retail-store.git
> git.exe --version # timeout=10
> git --version # 'git version 2.44.0.windows.1'
> git.exe fetch --tags --force --progress -- https://github.com/sunny-shaw/retail-store.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/sunny-shaw/retail-store.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git.exe rev-parse "refs/remotes/origin/main^{commit}" # timeout=10
Checking out Revision eb9a8bda392a28adacda9956a19467c7f679e6da (refs/remotes/origin/main)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f eb9a8bda392a28adacda9956a19467c7f679e6da # timeout=10
Commit message: "refactor DiscountStrategyFactoryTest"
First time build. Skipping changelog.
Injecting SonarQube environment variables using the configuration: adv_devops_7_sonarqube
Unpacking https://repo1.maven.org/maven2/org/sonarsource/scanner/cli/sonar-scanner-cli/6.2.0.4584/sonar-scanner-cli-6.2.0.4584.zip to
C:\ProgramData\jenkins\workspace\exp7_adv\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_exp7 on Jenkins
```

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



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