

NAME:ATHARVA SHINDE

ROLL.NO:54

CLASS:D15C

Adv DevOps Practical 7

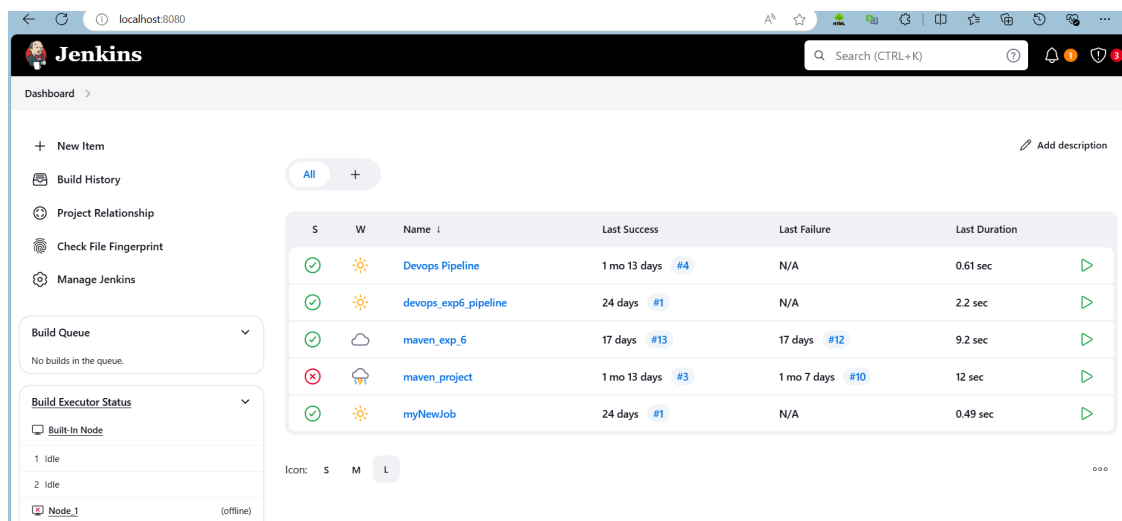
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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.



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

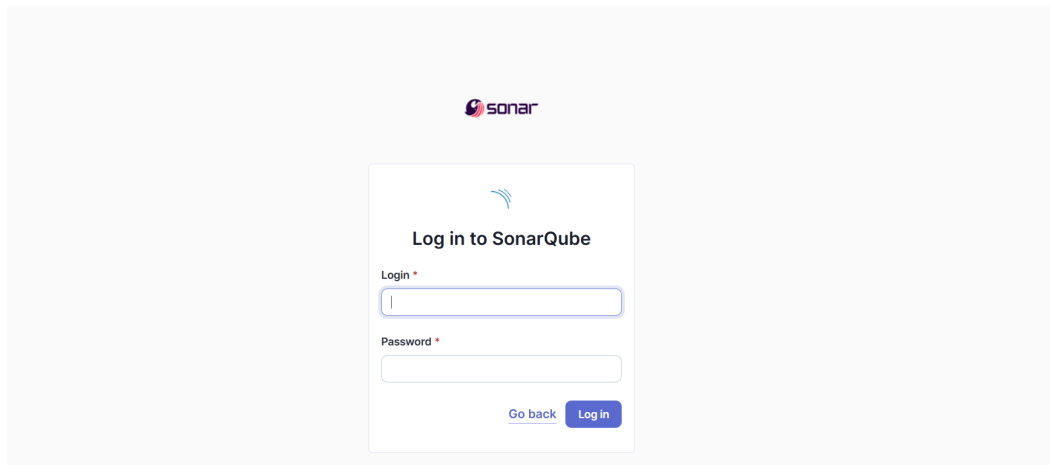
```
docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p
```

```
9000:9000 sonarqube:latest
```

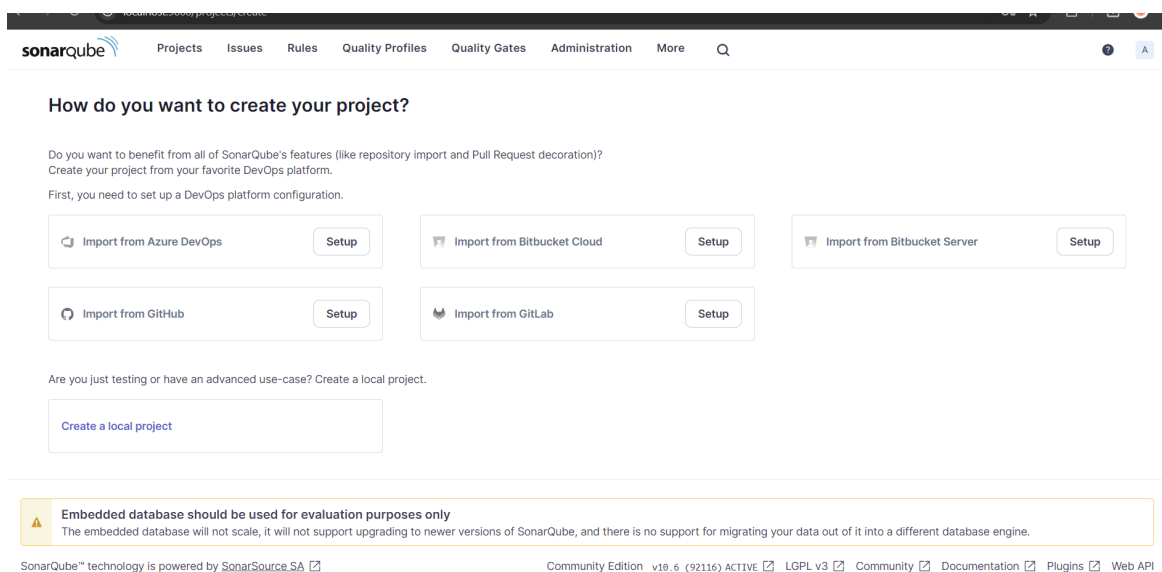
-----Warning: run below command only once

```
C:\Users\athar>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
cc1cc40d5c849124ca7dcbc177cd2d17953733ddad728014f6a580dbf5ff15ab
```

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.



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

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

Add Ant

Save Apply

7. 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.


adv_devops_exp7


» 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

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.

Git ?

Repositories ?

Repository URL ?

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

Credentials ?

- none -

+ Add

Advanced

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.

Dashboard > adv_devops_exp7 > Configuration

Configure

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

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

Add post-build action v

Save Apply

Dashboard > adv_devops_exp7 > Configuration

Execute SonarQube Scanner

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job) v

Path to project properties ?

Analysis properties ?

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

Additional arguments ?

JVM Options ?

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

Q

Administration

Configuration

Security

Projects

System

Marketplace

All

Users

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>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.

Build Steps

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_3137473a581901b73c22c23cd54ed302db3cc3c6
sonar.sources=.

Additional arguments ?

12. Run the Jenkins build.

**Jenkins**

Search (CTRL+K) ?

1 Att

Dashboard > adv_devops_exp7 >

Status

</> Changes

Workspace

Build Now


Configure

Delete Project

SonarQube

Rename

adv_devops_exp7

 SonarQube

Permalinks

- Last build (#4), 2 min 8 sec ago
- Last stable build (#4), 2 min 8 sec ago
- Last successful build (#4), 2 min 8 sec ago
- Last failed build (#3), 4 min 33 sec ago
- Last unsuccessful build (#3), 4 min 33 sec ago
- Last completed build (#4), 2 min 8 sec ago

Build History

trend ▼

Filter... /

#4

Sep 24, 2024, 4:43 PM

#3

Sep 24, 2024, 4:40 PM

#2

Dashboard > adv_devops_exp7 > #4 > Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#4'

Timings

Git Build Data

Previous Build

Console Output

Download Copy View as plain text

```
Started by user Atharva Ajit Shinde
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
> git.exe config remote.origin.url https://github.com/shazforiot/MSBuild_firstproject.git # timeout=10
Fetching upstream changes from https://github.com/shazforiot/MSBuild_firstproject.git
> git.exe --version # timeout=10
> git --version # 'git version 2.42.0.windows.2'
> git.exe fetch --tags --force --progress -- https://github.com/shazforiot/MSBuild_firstproject.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe rev-parse "refs/remotes/origin/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c6e72427c380bcae6d6fee7b49adf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
Commit message: "updated"
> git.exe rev-list --no-walk f2bc042c04c6e72427c380bcae6d6fee7b49adf # timeout=10
[adv_devops_exp7] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\sonarqube_exp7\bin\sonar-scanner.bat -
Dsonar.host.url=http://localhost:9000 -Dsonar.projectKey=adv_devops_7_sonarqube -Dsonar.host.url=http://localhost:9000 -
Dsonar.login=sqp_3137473a581901b73c22c23cd54ed302db3cc3c6 -Dsonar.sources=. -
Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\adv_devops_exp7
16:43:14.728 WARN Property 'sonar.host.url' with value 'http://localhost:9000' is overridden with value 'http://localhost:9000'
```

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

sonarqube

Projects Issues Rules Quality Profiles Quality Gates Administration More

adv_devops_7_sonarqube / main

Overview Issues Security Hotspots Measures Code Activity Project Settings Project Information

main Version not provided Set as homepage

Passed Quality Gate Last analysis 3 minutes ago

The last analysis has warnings. See details

New Code Overall Code

Security 0 Open issues 0 H 0 M 0 L	Reliability 0 Open issues 0 H 0 M 0 L	Maintainability 0 Open issues 0 H 0 M 0 L
Accepted issues 0	Coverage 0.0%	Duplications 0.0%

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

Conclusion:

In this project, we integrated Jenkins with SonarQube for automated static application security testing (SAST). We set up SonarQube using Docker, configured Jenkins with the necessary plugins and

authentication, and linked it to a GitHub repository. The SonarQube scanner was added as a build step,

enabling continuous code analysis for vulnerabilities, code smells, and quality issues, ensuring automated

reporting and continuous code quality improvement.