

Adv DevOps Practical 7

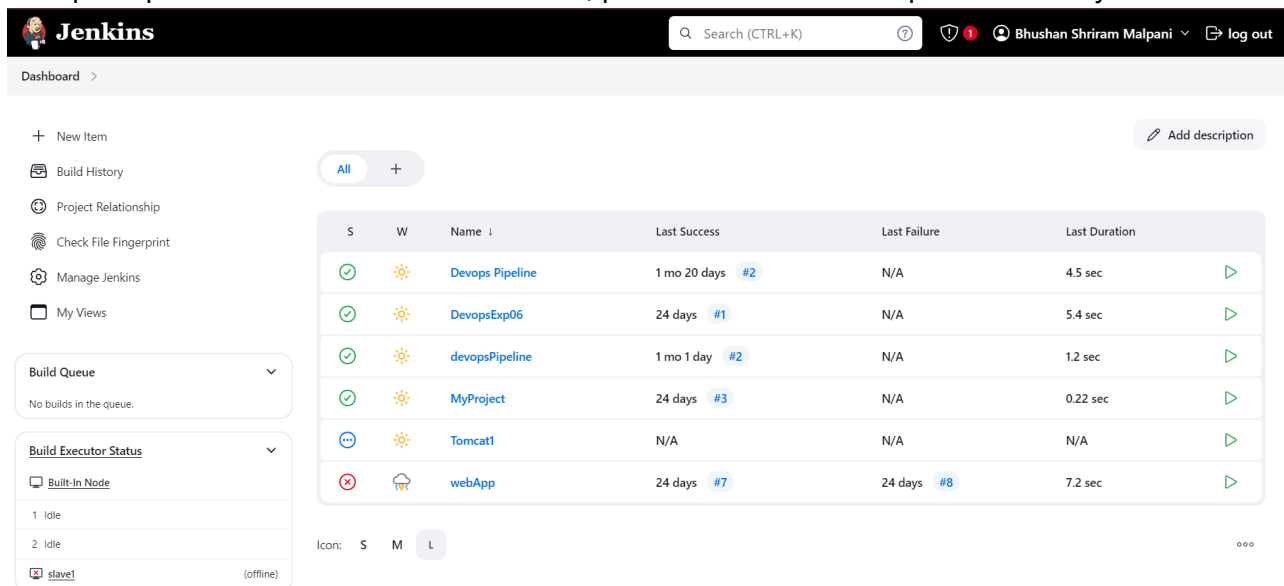
Aim: To understand Static Analysis SAST process and learn to integrate Jenkins SAST to SonarQube/GitLab.

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 interface. At the top, there's a search bar and a user profile for 'Bhushan Shriram Malpani'. The main content area displays a table of builds with columns for status, name, last success, last failure, and last duration. The builds listed are 'Devops Pipeline', 'DevopsExp06', 'devopsPipeline', 'MyProject', 'Tomcat1', and 'webApp'. The 'webApp' build is currently failing, indicated by a red icon and a duration of 7.2 sec.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Devops Pipeline	1 mo 20 days #2	N/A	4.5 sec
✓	☀	DevopsExp06	24 days #1	N/A	5.4 sec
✓	☀	devopsPipeline	1 mo 1 day #2	N/A	1.2 sec
✓	☀	MyProject	24 days #3	N/A	0.22 sec
...	☀	Tomcat1	N/A	N/A	N/A
✗	☁	webApp	24 days #7	24 days #8	7.2 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

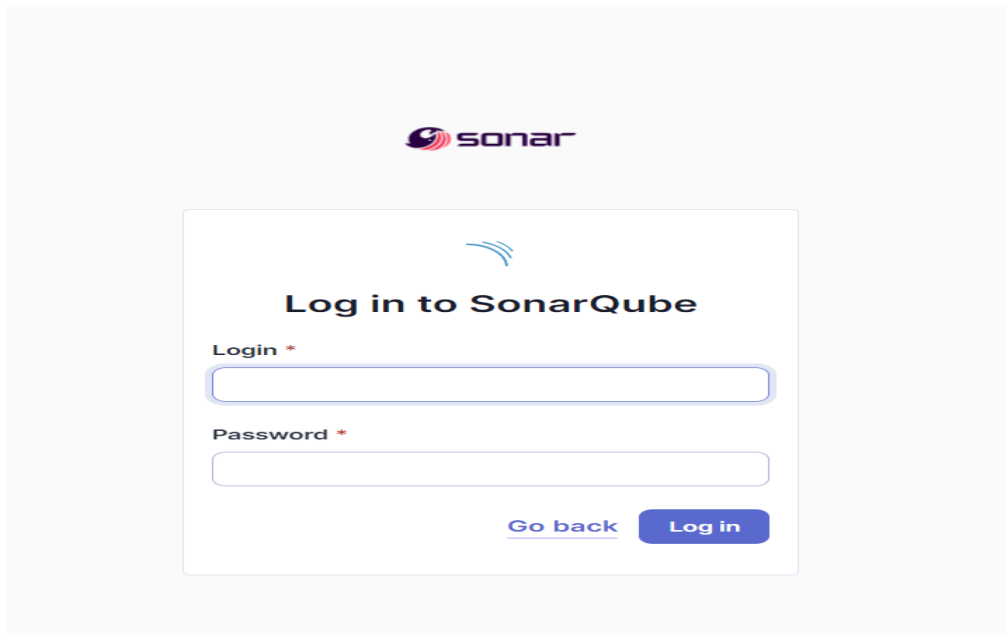
- **Warning: run below command only once**

```

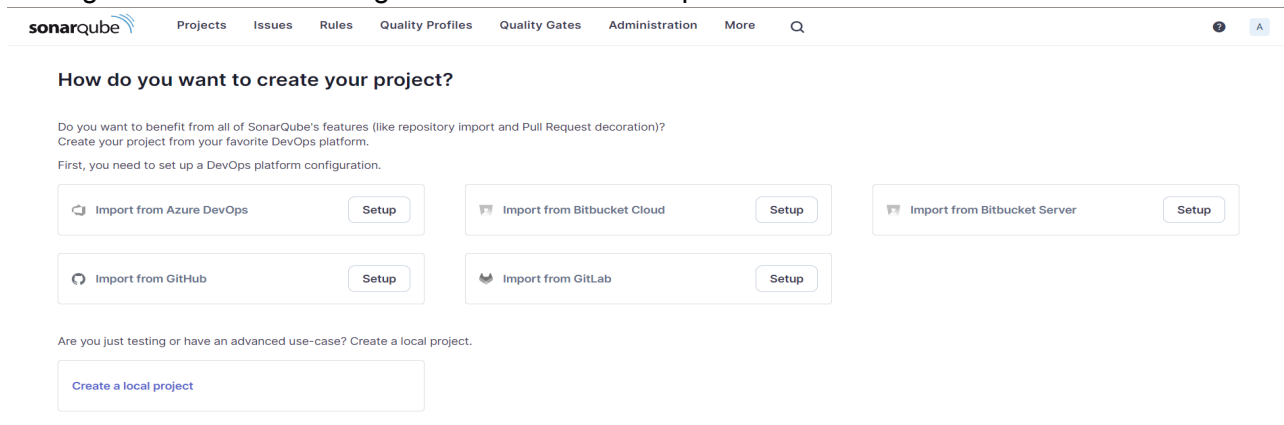
Microsoft Windows [Version 10.0.22631.4169]
(c) Microsoft Corporation. All rights reserved.

C:\Users\HP>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
787074c97d3df5c971edeea44247a0b0b0dbfc9e0e3db74c6d3c61a99180824c
  
```

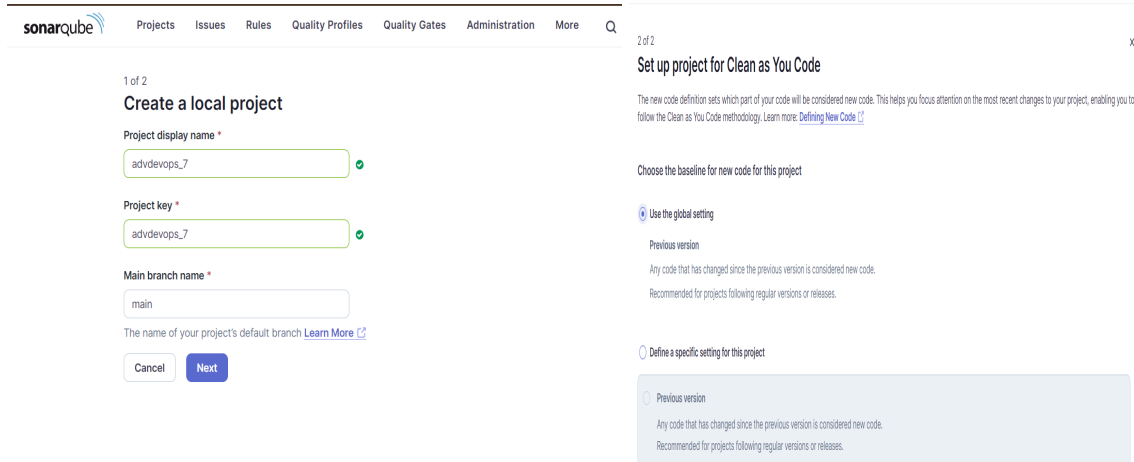
- Once the container is up and running, you can check the status of SonarQube at localhost port 9000.



- Login to SonarQube using username admin and password admin.



- Create a manual project in SonarQube with the name sonarqube



Setup the project and come back to Jenkins Dashboard.

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

Dashboard > Manage Jenkins > Plugins

Plugins

Updates 27

Available plugins

Installed plugins

Advanced settings

☒ **SonarQube Scanner** 2.17.2
[External Site/Tool Integrations](#) [Build Reports](#)
This plugin allows an easy integration of **SonarQube**, the open source platform for Continuous Inspection of code quality. 7 mo 6 days ago

☐ **Sonar Quality Gates** 315.vf1f12b_e81a_3a_4
[Library plugins \(for use by other plugins\)](#) [analysis](#) [Other Post-Build Actions](#)
Fails the build whenever the Quality Gates criteria in the Sonar 5.6+ analysis aren't met (the project Quality Gates status is different than "Passed") 27 days ago

☐ **Quality Gates** 2.5
Fails the build whenever the Quality Gates criteria in the Sonar analysis aren't met (the project Quality Gates status is different than "Passed")

Dashboard > Manage Jenkins > Plugins

Plugins

Updates 25

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SonarQube Scanner Success

Loading plugin extensions Success

→ [Go back to the top page](#)
(you can start using the installed plugins right away)

→ ☐ 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**

Dashboard > Manage Jenkins > System >

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

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

+ 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 > System >

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

Server URL
Default is http://localhost:9000

Server authentication token
SonarQube authentication token. Mandatory when anonymous access is disabled.

none -

+

Add

Advanced

▼

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

☒ Install automatically ?

Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

Add Installer

Add SonarQube Scanner


Save Apply


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


New Item


Enter an item name

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.

 **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.

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.

Source Code Management

☐ None

☒ Git ?

Repositories ?

Repository URL ?

Credentials ?

+ Add ▾

Advanced ▾

Save Apply

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 > advdevops07 > 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 ▾

Save Apply

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job)

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=advdevops7
sonar.host.url=http://localhost:9000
sonar.login=admin
sonar.sources=.
```

Additional arguments ?

JVM Options ?

Add build step ▾

Save

Apply

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 Q

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.

JDK ?

JDK to be used for this SonarQube analysis

(Inherit From Job) ▾

Path to project properties ?

Analysis properties ?

```
sonar.projectKey=sonarqube
sonar.projectName=advdevops_7
sonar.projectVersion=1.0
sonar.sources=.
sonar.host.url=http://localhost:9000
sonar.login=sqa_6e1401709960bcb21f8f8a53a65633a2a97ca501
sonar.projectBaseDir=C:/ProgramData/Jenkins/jenkins/workspace/advdevops07
```

Additional arguments ?

 ▾

JVM Options ?

 ▾

12. Run the Jenkins build.

✓ advdevops07



Permalinks

- [Last build \(#4\), 3 min 20 sec ago](#)
- [Last stable build \(#4\), 3 min 20 sec ago](#)
- [Last successful build \(#4\), 3 min 20 sec ago](#)
- [Last failed build \(#3\), 5 min 14 sec ago](#)
- [Last unsuccessful build \(#3\), 5 min 14 sec ago](#)
- [Last completed build \(#4\), 3 min 20 sec ago](#)

Check the console Output

✓ Console Output

[Download](#)[Copy](#)[View as](#)

```
Started by user Bhushan Shriram Malpani
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\.jenkins\workspace\advdevops07
[advdevops07] $ C:\ProgramData\Jenkins\.jenkins\tools\udson.plugins.sonar.SonarRunnerInstallation\exp07\bin\sonar-scanner.bat -
Dsonar.host.url=https://localhost:9000 -Dsonar.projectKey=sonarqube -Dsonar.projectName=advdevops_7 -Dsonar.host.url=http://localhost:9000 -
Dsonar.login=sqa_6e1401709960bcb21f8f8a53a65633a2a97ca501 -Dsonar.projectBaseDir=C:/ProgramData/Jenkins/.jenkins/workspace/advdevops07 -
Dsonar.projectVersion=1.0 -Dsonar.sources=.
14:42:11.487 WARN Property 'sonar.host.url' with value 'https://localhost:9000' is overridden with value 'http://localhost:9000'
14:42:11.495 INFO Scanner configuration file:
C:\ProgramData\Jenkins\.jenkins\tools\udson.plugins.sonar.SonarRunnerInstallation\exp07\bin\..\conf\sonar-scanner.properties
14:42:11.496 INFO Project root configuration file: NONE
14:42:11.508 INFO SonarScanner CLI 6.2.0.4584
14:42:11.510 INFO Java 21.0.4 Eclipse Adoptium (64-bit)
14:42:11.513 INFO Windows 11 10.0 amd64
14:42:11.530 INFO User cache: C:\windows\system32\config\systemprofile\.sonar\cache
14:42:11.955 INFO JRE provisioning: os[windows], arch[amd64]
```

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

☆ advdevops_7 PUBLIC

✓ Passed

Last analysis: 7 minutes ago

The main branch of this project is empty.

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

Conclusion:

In this project, we combined Jenkins with SonarQube to automate static application security testing (SAST). SonarQube was deployed via Docker, while Jenkins was configured with the required plugins and authentication. It was then connected to a GitHub repository. The SonarQube scanner was incorporated as a build step, facilitating ongoing code analysis for vulnerabilities, code quality issues, and other concerns, ensuring continuous improvement through automated reports.

