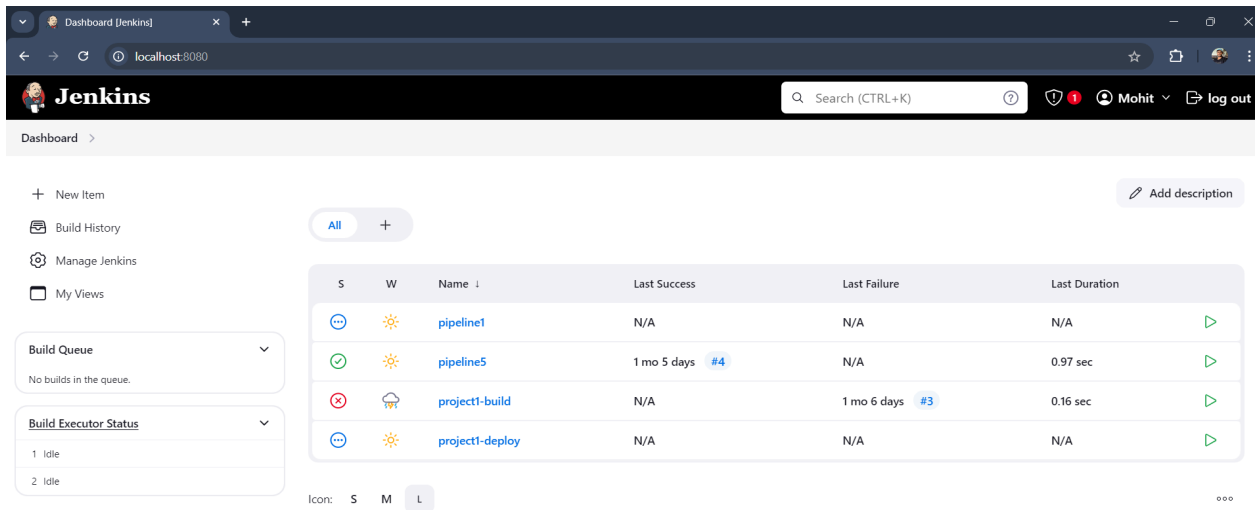


EXPERIMENT 7

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

- Open up Jenkins Dashboard on localhost, port 8080 or whichever port it is at for you.



The screenshot shows the Jenkins Dashboard interface. On the left, there is a sidebar with navigation links: 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these, there are two sections: 'Build Queue' (showing 'No builds in the queue.') and 'Build Executor Status' (showing '1 Idle' and '2 Idle'). The main area displays a table of builds. The table has columns for 'S' (Status), 'W' (Icon), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The builds listed are 'pipeline1', 'pipeline5', 'project1-build', and 'project1-deploy'. Below the table, there are icons for 'S', 'M', and 'L'.

S	W	Name	Last Success	Last Failure	Last Duration
...	...	pipeline1	N/A	N/A	N/A
...	...	pipeline5	1 mo 5 days #4	N/A	0.97 sec
...	...	project1-build	N/A	1 mo 6 days #3	0.16 sec
...	...	project1-deploy	N/A	N/A	N/A

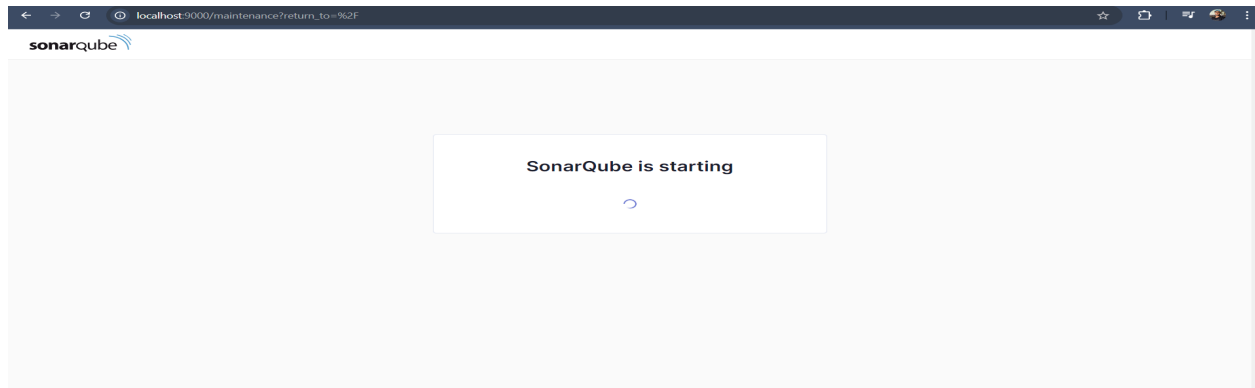
- Run SonarQube in a Docker container using this command

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

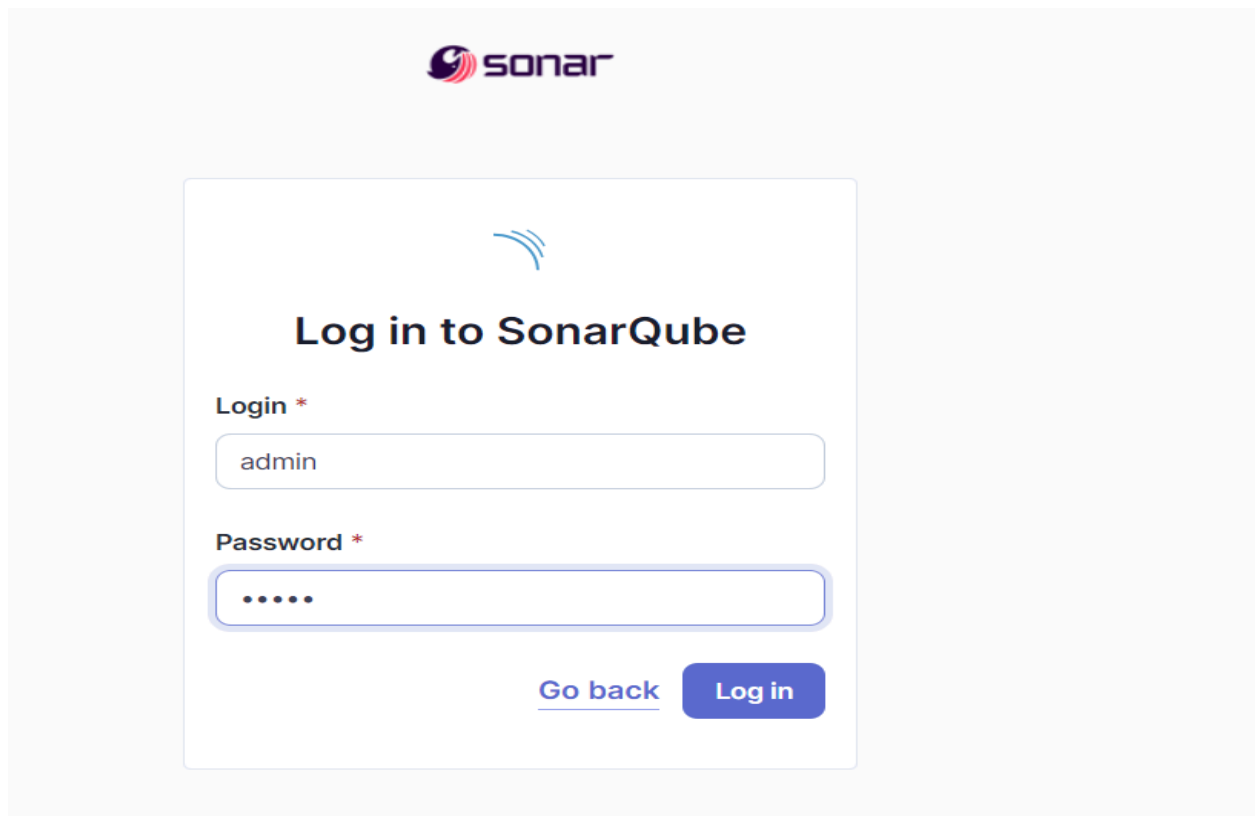
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\mrroh> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
Unable to find image 'sonarqube:latest' locally
```

- 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](#) 

Cancel

Next

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



Name ↓

Enabled

SonarQube Scanner for Jenkins 2.17.2

This plugin allows an easy integration of [SonarQube](#), the open source platform for Continuous Inspection of code quality.

[Report an issue with this plugin](#)



- Under Jenkins 'Configure System', look for SonarQube Servers and enter the details.

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

sonarqube

Server URL

Default is http://localhost:9000

http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add +

Advanced

Add SonarQube

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

SonarQube Scanner installations

Add SonarQube Scanner

SonarQube Scanner

Name

sonarqube

☒ Install automatically ?

Install from Maven Central

Version

SonarQube Scanner 6.2.0.4584

Add Installer

Add SonarQube Scanner

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

Jenkins

Search (CTRL+K)

Mohit

log out

Dashboard

All

New Item

New Item

Enter an item name

sonarqube

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.

- Choose this GitHub repository in Source Code Management.

Git ?

Repositories ?

Repository URL ?

Credentials ?
- none -
+ Add +


Advanced ▾

Add Repository


Branches to build ?


Branch Specifier (blank for 'any') ?


Add Branch


 **Jenkins**


Dashboard > sonarcube >


 Status


 Changes


 Workspace

 Build Now


 Configure

 Delete Project


 SonarQube



 Rename

sonarcube

 SonarQube

Permalinks

 Build History **trend** ^

 Atom feed for all  Atom feed for failures

- Check the console output.

Dashboard > AdDevopsLab7 > #1 > Console Output

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '91'

Git Build Data

✓ Console Output

```

Started by user Prajakt Upadhye
Running as SYSTEM
Building on the built-in node in workspace C:\ProgramData\Jenkins\workspace\AdDevopsLab7
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/PrajaktaUpadhye/MSBuild_FirstProject.git
> git.exe init C:\ProgramData\Jenkins\workspace\AdDevopsLab7 # timeout=10
Fetching upstream changes from https://github.com/PrajaktaUpadhye/MSBuild_FirstProject.git
> git.exe --version # timeout=10
> git --version # "git version 2.39.2.windows.1"
> git.exe fetch --tags --force --progress -- https://github.com/PrajaktaUpadhye/MSBuild_FirstProject.git # timeout=10
> git.exe config remote.origin.url https://github.com/PrajaktaUpadhye/MSBuild_FirstProject.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/master^{commit}" # timeout=10
Checking out Revision f2bc042c04c672427c380bcae6d8fee794bdf (refs/remotes/origin/master)
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f f2bc042c04c672427c380bcae6d8fee794bdf # timeout=10
Commit message: "updated"
First time build. Skipping changelog.
Uploading https://repol.maven.org/sewn2/org/sonarsource/scanner/cli/sonar-scanner-cli/5.0.1.3006/sonar-scanner-cli-5.0.1.3006.zip to
C:\ProgramData\Jenkins\workspace\AdDevopsLab7\tools\hudson.plugins.sonar.SonarRunnerInstallation\AdDevopsLab7
[AdDevopsLab7] $ C:\ProgramData\Jenkins\workspace\AdDevopsLab7\tools\hudson.plugins.sonar.SonarRunnerInstallation\AdDevopsLab7\sonar-scanner.bat -Dsonar.projectKey=AdDevopsLab7 -Dsonar.login=admin -Dsonar.hosturl=http://localhost:9000/ -Dsonar.password=admin -Dsonar.projectBaseDir=C:\ProgramData\Jenkins\workspace\AdDevopsLab7
INFO: Scanner configuration file: C:\ProgramData\Jenkins\workspace\AdDevopsLab7\tools\hudson.plugins.sonar.SonarRunnerInstallation\AdDevopsLab7\conf\sonar-scanner.properties
INFO: Project root configuration file: NONE
INFO: SonarScanner 5.0.1.3006
INFO: Java 18.0.2.1 Oracle Corporation (64-bit)

INFO: Sensor IaC Docker Sensor [iac] (done) | time=74ms
INFO: ----- Run sensors on project
INFO: Sensor C# [csharp]
WARN: Your project contains C# files which cannot be analyzed with the scanner you are using. To analyze C# or VB.NET, you must use the SonarScanner for .NET 5.x or higher, see
https://redirect.sonarsource.com/doc/install-configure-scanner-msbuild.html
INFO: Sensor C# [csharp] (done) | time=0ms
INFO: Sensor Analysis Warnings Import [csharp]
INFO: Sensor Analysis Warnings Import [csharp] (done) | time=15ms
INFO: Sensor C# File Caching Sensor [csharp]
WARN: Incremental PR analysis: Could not determine common base path, cache will not be computed. Consider setting 'sonar.projectBaseDir' property.
INFO: Sensor C# File Caching Sensor [csharp] (done) | time=0ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=2ms
INFO: SCM Publisher SCM provider for this project is: git
INFO: SCM Publisher 4 source files to be analyzed
INFO: SCM Publisher 4/4 source files have been analyzed (done) | time=792ms
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: Analysis report generated in 83ms, dir size=139.0 kB
INFO: Analysis report compressed in 13ms, zip size=19.2 kB
INFO: Analysis report uploaded in 241ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=AdDevopsLab7
INFO: Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
INFO: More about the report processing at http://localhost:9000/api/ci/task?id=AYqpP1tk06Kzcbt14cp
INFO: Analysis total time: 16.844 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 31.521s
INFO: Final Memory: 20M/74M
INFO: -----
Finished: SUCCESS

```

- Once the build is complete, check the project in SonarQube.

☆ sonarqube

Passed

Last analysis: 7 hours ago

The main branch of this project is empty.

localhost9000/dashboard?id=sonarqube-test&codeScope=overall

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministrationMore

sonarqube-test / main

OverviewIssuesSecurity HotspotsMeasuresCodeActivityProject SettingsProject Information

main

Version not providedSet as homepage

Quality Gate

Passed

Last analysis 14 minutes ago

The last analysis has warnings. See details

New CodeOverall Code

Security

0 Open issues

0 H0 M0 L

Reliability

0 Open issues

0 H0 M0 L

Maintainability

0 Open issues

0 H0 M0 L

Accepted issues

0

Valid issues that were not fixed

Coverage

On 0 lines to cover.

Duplications

0.0%

On 86 lines.

Security Hotspots