

Adv.DevOps Exp 07

Name- Ishika Devare RollNo- 14 Batch- A

Adv. DevOps

Experiment No. 7

Aim - To understand static Analysis SAST process and learn to integrate jenkins, SAST to sonarqube / GitLab.

Theory -

What is SAST?

Static Application security Testing (SAST), or static analysis is a testing methodology that analyzes source code to find security vulnerabilities that make organizations applications susceptible to attack. SAST scans an application before code is compiled. It is also known as 'White box Testing'.

What problem does SAST solve?

A key strength of SAST tool is the ability to analyze 100% of the codebase. SAST takes place very early in software development life cycle (SDLC) as does not require a working application and can take place without being executed. It helps developers identify vulnerabilities in initial stages of development and quickly resolve issues. It's important to note that SAST tools must be run on application on a regular basis.

Implementation:

Firstly install docker and jenkins

Note- Got error while installing sonarqube as docker engine was not starting, so if this is the case try reinstalling docker

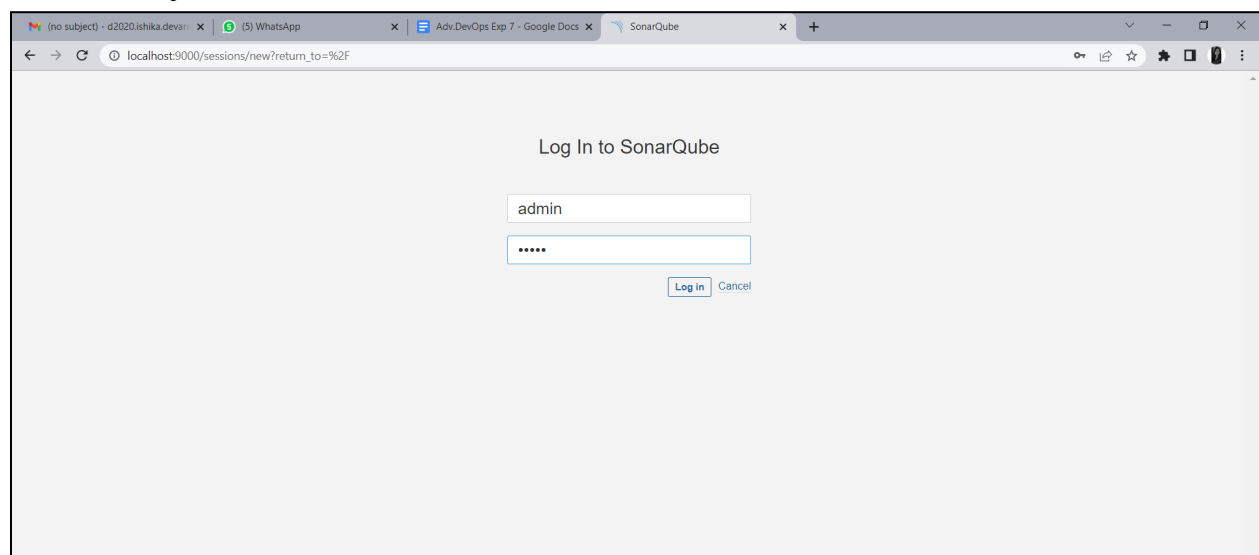
```
PS C:\Users\Ishika Devare> docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
docker: error during connect: This error may indicate that the docker daemon is not running.: Post "http://%2F%2F.%2Fpipe%2Fdocker_engine/v1.24/containers/create?name=sonarqube": open //./pipe/docker_engine: The system cannot find the file specified.
See 'docker run --help'.
PS C:\Users\Ishika Devare>
```

Step 1: Installing SonarQube from the Docker Image

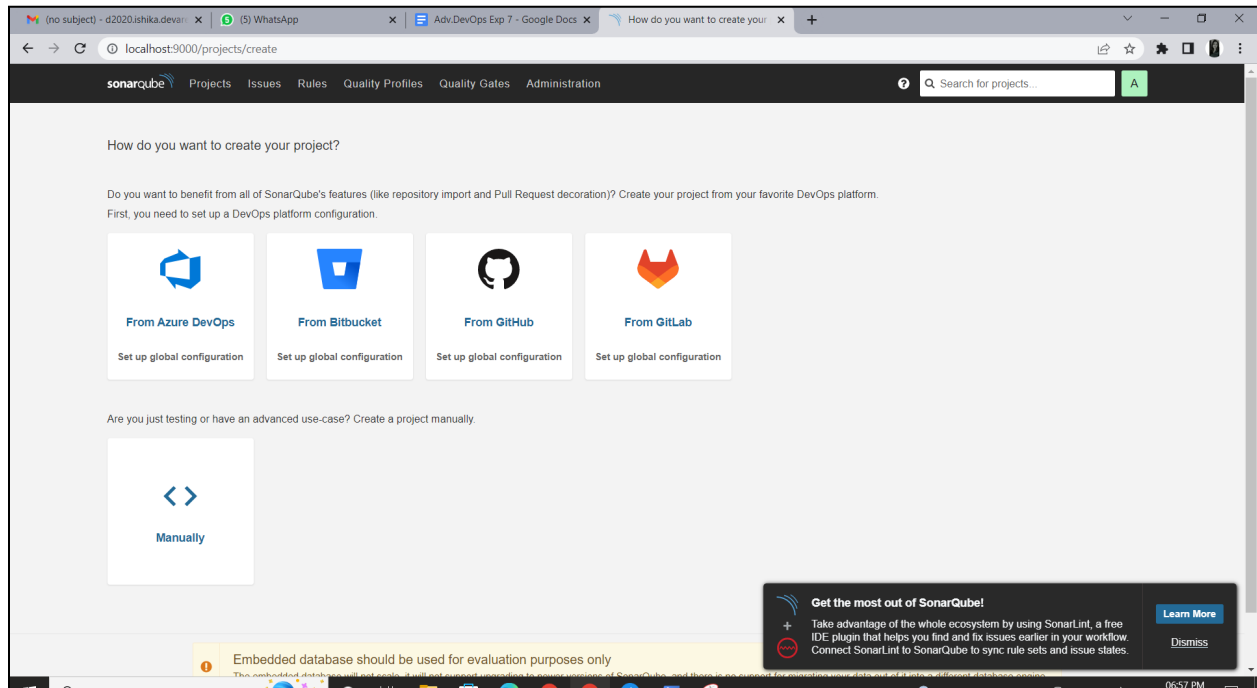
```
$ docker run -d --name sonarqube -e SONAR_ES_BOOTSTRAP_CHECKS_DISABLE=true -p 9000:9000 sonarqube:latest
```

```
PS C:\Users\Ishika Devare> 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
9621f1afde84: Pull complete
0da9106727c7: Pull complete
129c5a3f9c32: Pull complete
Digest: sha256:3fa9a76948fab6fafa41950bee256afea943773744723b5e4f38b340643516b9
Status: Downloaded newer image for sonarqube:latest
88155e951738bf8bfc100c6546c780186a059d4415872fdedfec971cdc8aef5e
PS C:\Users\Ishika Devare>
```

Step 2: After installation of SonarQube, go to the SonarQube page by typing: <http://localhost:9000/> on your browser. If you see such a page then you have successfully installed it.



Step 3: Login using the username as “admin” and password as “admin”. And then you will see the home page of SonarQube.

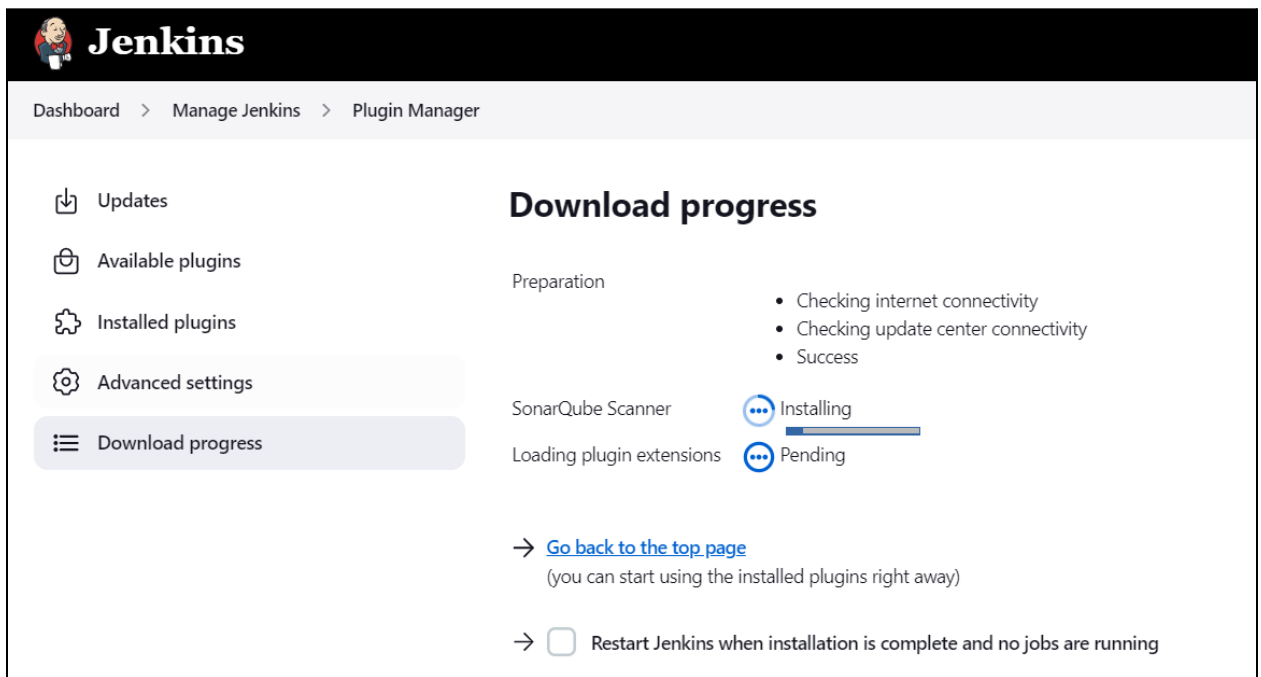
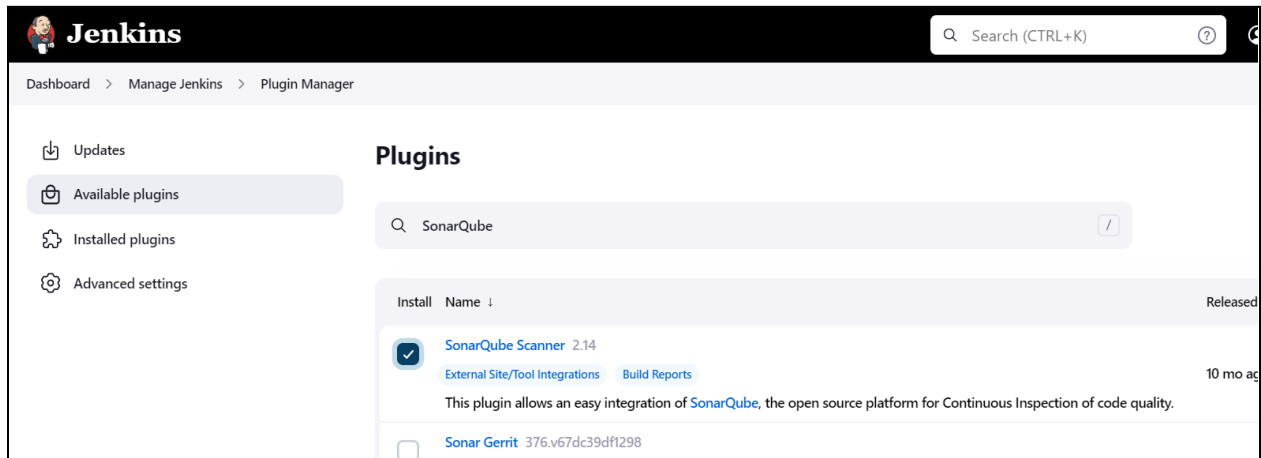


Step 4: Create a manual project in SonarQube with the name “AdvDevOps-EXP7” and set up the project.

A screenshot of the SonarQube "Create a project" form. The form is titled "Create a project" and has a subtitle "All fields marked with * are required". It contains two main sections: "Project display name *" and "Project key *". Both sections have a text input field with the value "AdvDevOps-Exp7" and a green checkmark icon to the right. Below the "Project display name" field, there is a note: "Up to 255 characters. Some scanners might override the value you provide." Below the "Project key" field, there is a note: "The project key is a unique identifier for your project. It may contain up to 400 characters. Allowed characters are alphanumeric, '-' (dash), '_' (underscore), '.' (period) and ':' (colon), with at least one non-digit." At the bottom of the form, there is a "Set Up" button. The browser's address bar shows "localhost:9000/projects/create?mode=manual".

Now open the Jenkins Dashboard in the new tab of the browser by typing localhost:8080

Step 5: Go to Manage Jenkins and search for SonarQube Scanner for Jenkins and install it.



Step 6: Under Jenkins , Dashboard > Manage Jenkins >Configure System , look for SonarQube Servers and enter the details. Enter the Server Authentication Token if needed.

Dashboard > Manage Jenkins > Configure System >

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

☒ **Environment variables** Enable injection of SonarQube server configuration as build environment variables

SonarQube installations

List of SonarQube installations

Name

AdvDevops-Exp7

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

- none -

+ Add

Advanced...

Save Apply

Step 7: Search SonarQube Scanner under Dashboard > Manage Jenkins > Global Tool Configuration. Choose the latest configuration and choose Install Automatically.

Dashboard > Manage Jenkins > Global Tool Configuration

SonarQube Scanner

SonarQube Scanner installations

List of SonarQube Scanner installations on this system

Add SonarQube Scanner

SonarQube Scanner

Name

AdvDevops-Exp7

☒ **Install automatically** ?

Install from Maven Central

Version

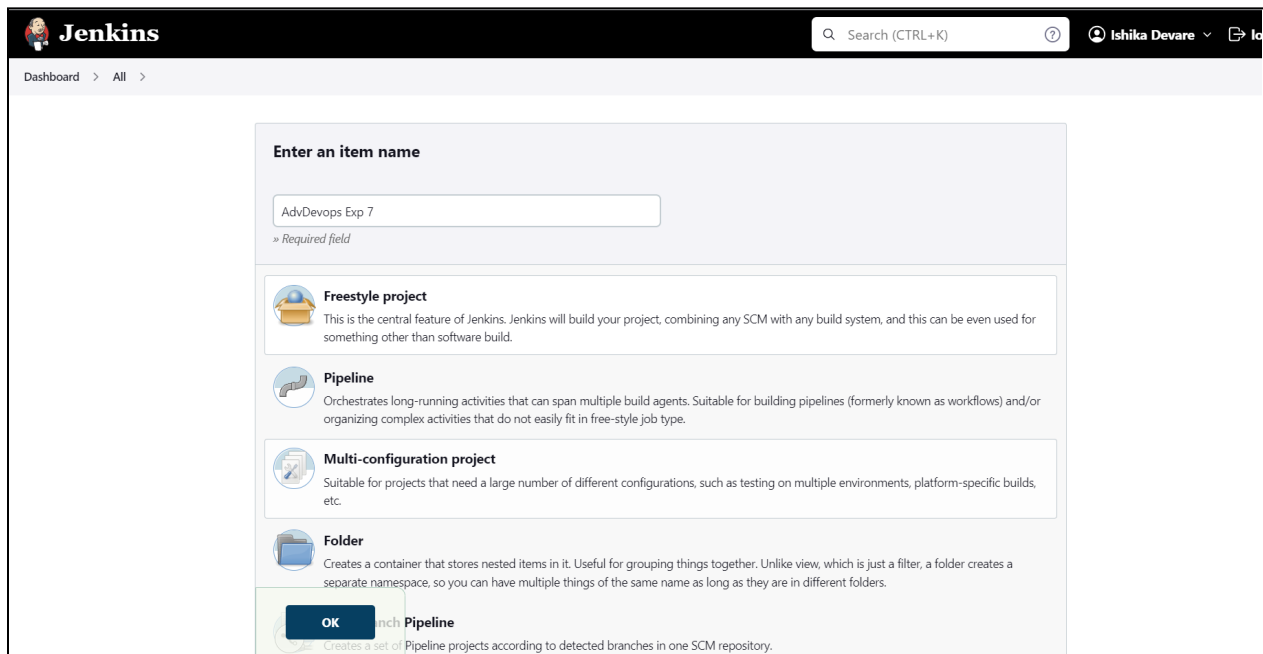
SonarQube Scanner 4.7.0.2747

Add Installer

Save Apply

Step 8: After the configuration, create a New Item in Jenkins, choose a freestyle

Project.

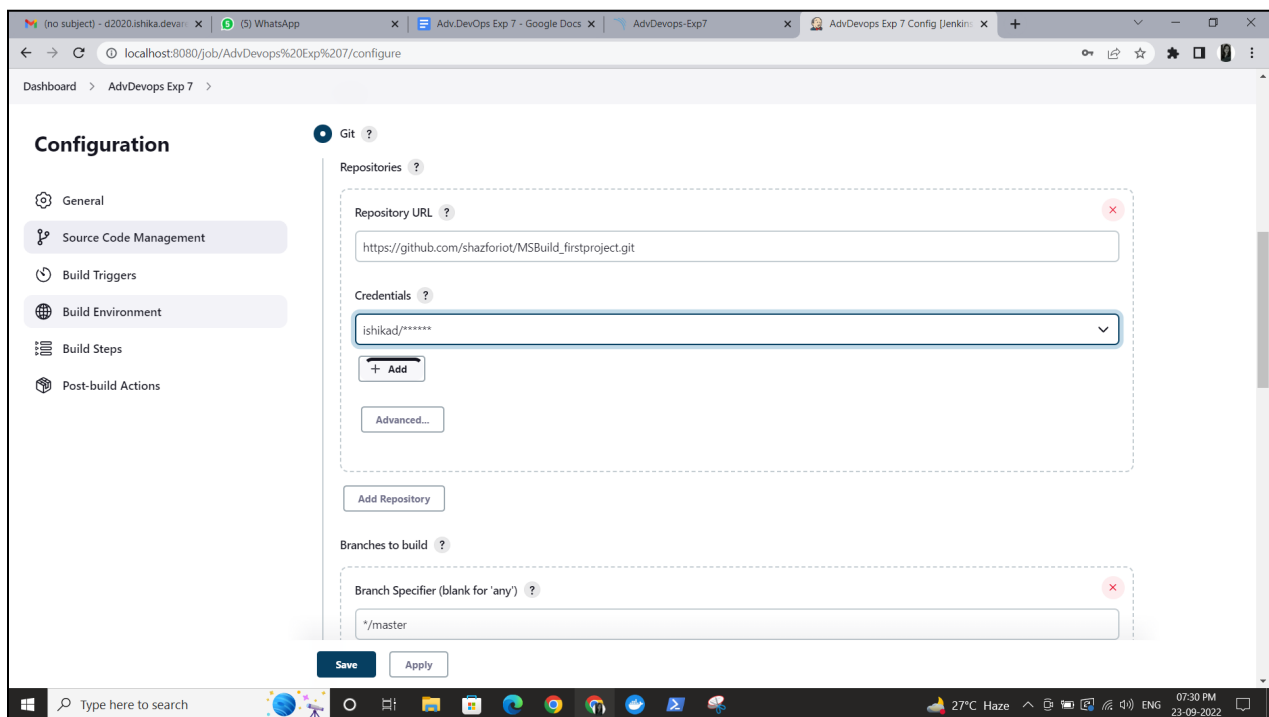


The image shows the Jenkins 'Enter an item name' dialog box. At the top, there's a search bar with 'Search (CTRL+K)' and a user profile 'Ishika Devare'. Below the search bar, the breadcrumb 'Dashboard > All >' is visible. The main section is titled 'Enter an item name' and contains a text input field with the value 'AdvDevOps Exp 7'. Below the input field, it says '» Required field'. There are four options listed: 'Freestyle project', 'Pipeline', 'Multi-configuration project', and 'Folder'. Each option has an icon and a description. At the bottom, there are 'OK' and 'Cancel' buttons. The 'Pipeline' option is highlighted, and a tooltip is visible: 'Creates a set of Pipeline projects according to detected branches in one SCM repository.'

Step 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 image shows the Jenkins Configuration page for 'AdvDevOps Exp 7'. The breadcrumb is 'Dashboard > AdvDevOps Exp 7 >'. The left sidebar has a 'Configuration' section with sub-items: 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. The 'Source Code Management' tab is selected. The main area is titled 'Git' and contains a 'Repositories' section. The 'Repository URL' field is filled with 'https://github.com/shazforiot/MSBuild_firstproject.git'. The 'Credentials' dropdown is set to 'ishikad/*****'. There are 'Add' and 'Advanced...' buttons. Below the repository section is an 'Add Repository' button. The 'Branches to build' section has a 'Branch Specifier (blank for \'any\')' field filled with '*/master'. At the bottom, there are 'Save' and 'Apply' buttons. The Windows taskbar at the bottom shows the time as 07:30 PM on 23-09-2022.

Step 10: Under Build > Execute SonarQube Scanner, enter these Analysis properties. Mention the SonarQube Project Key, Login, Password, and Host URL.

sonar.projectKey=AdvDevops-EXP7

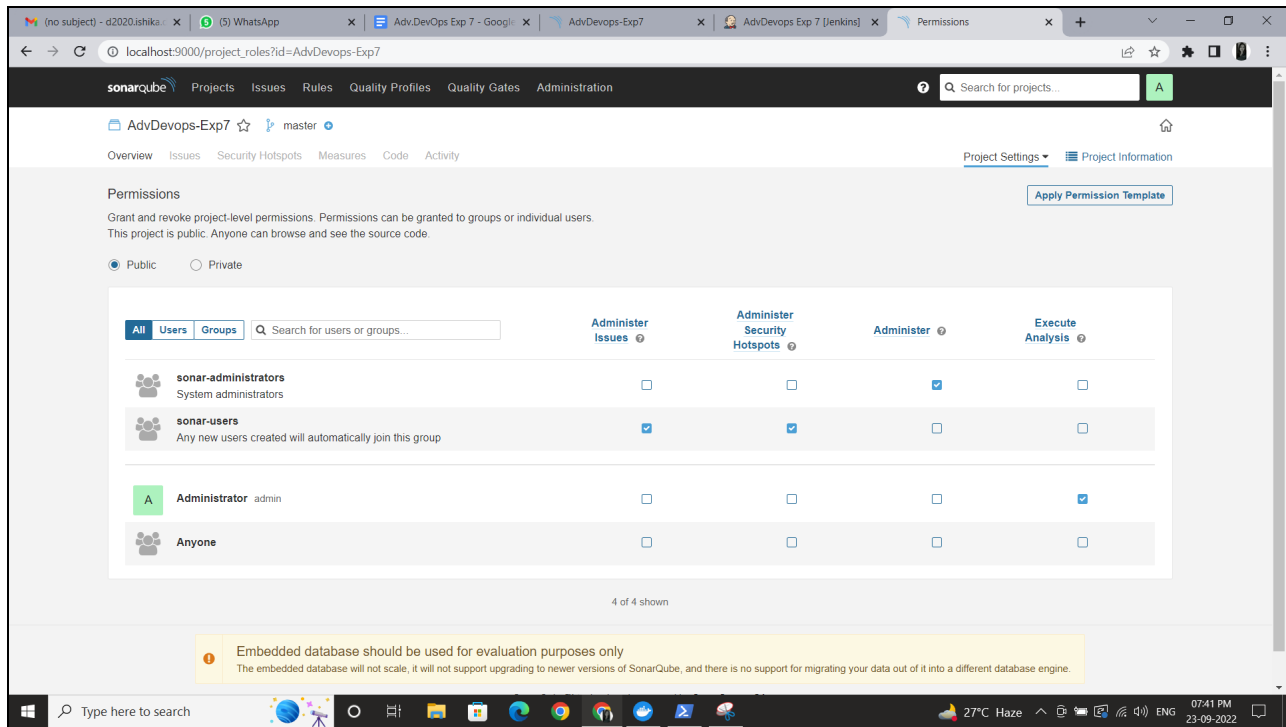
sonar.login=*your username*

sonar.password=*your password*

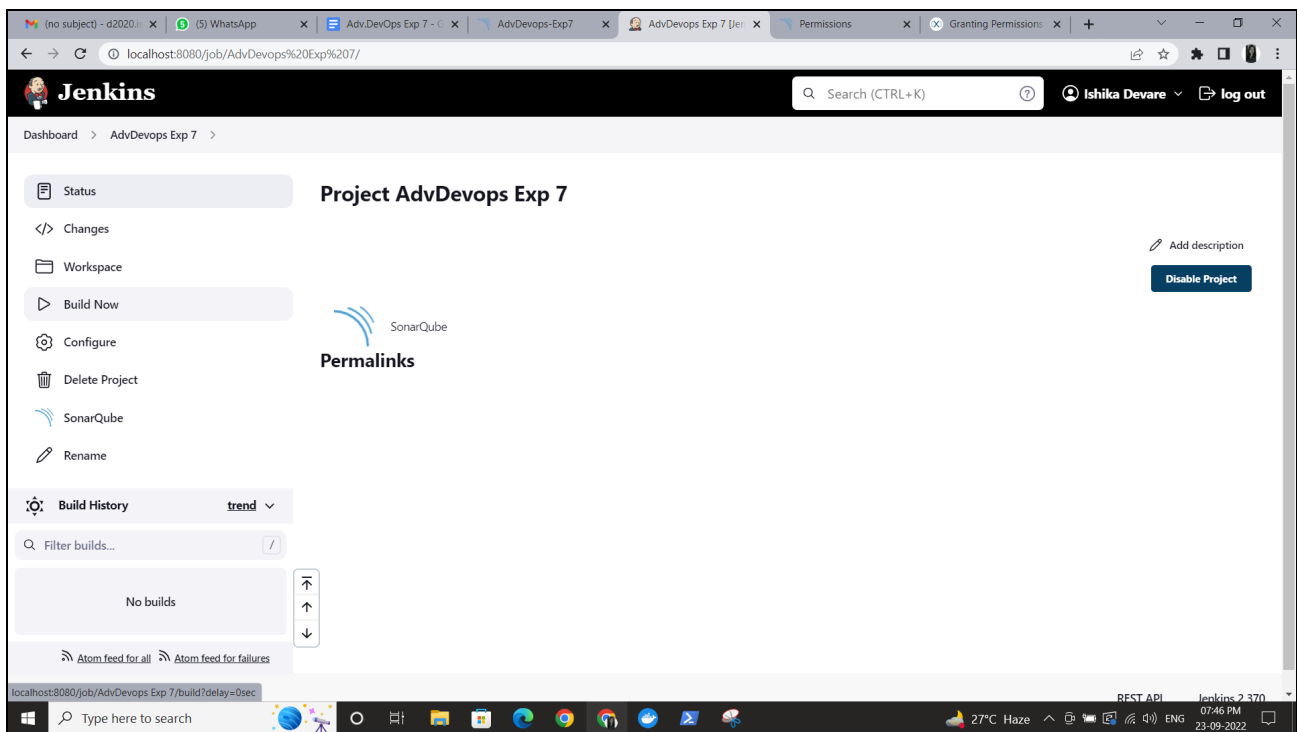
sonar.hosturl=<http://localhost:9000/>

The screenshot shows the Jenkins configuration page for a build step named 'Execute SonarQube Scanner'. The left sidebar contains a 'Configuration' menu with options: General, Source Code Management, Build Triggers, Build Environment, Build Steps (selected), and Post-build Actions. The main area is titled 'Build Steps' and contains the configuration for the selected step. It includes fields for 'Task to run', 'JDK' (set to 'Inherit From Job'), 'Path to project properties', and 'Analysis properties'. The 'Analysis properties' field contains the following text: sonar.projectKey=AdvDevops-Exp7, sonar.login=admin, sonar.password=[redacted], and sonar.hosturl=http://localhost:9000/. At the bottom, there are 'Save' and 'Apply' buttons.

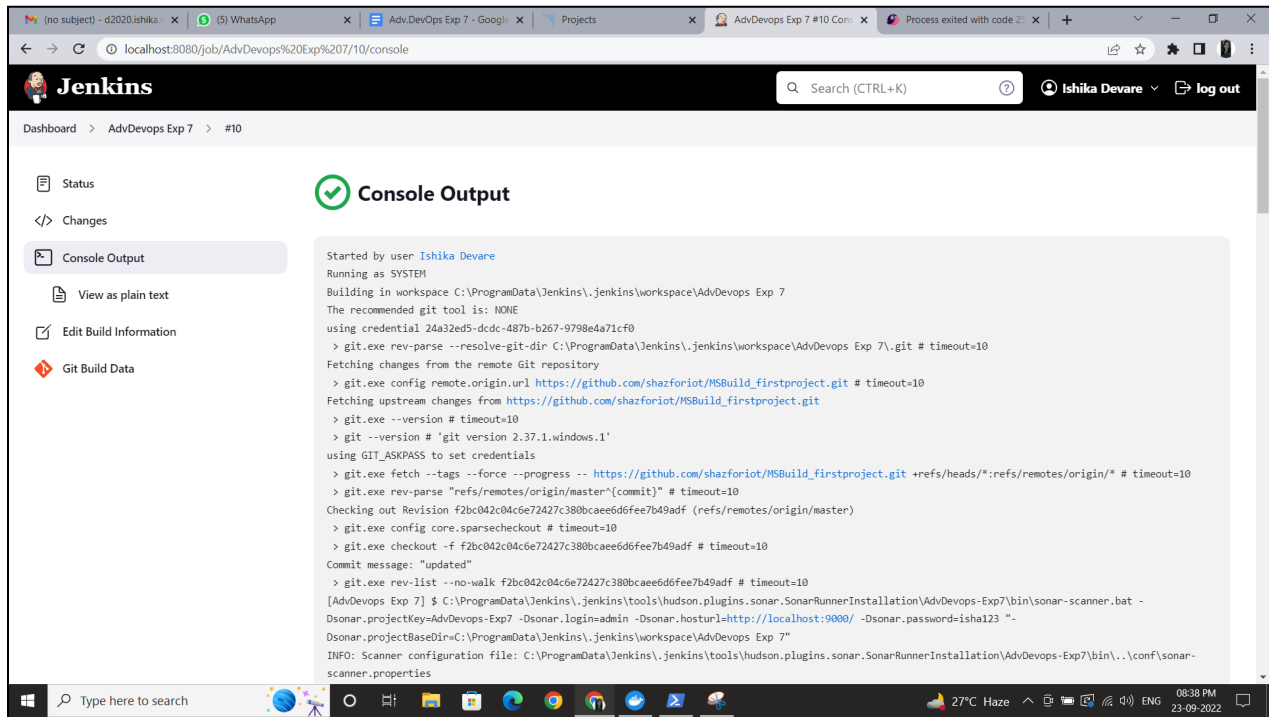
Step 11: Go to <http://localhost:9000/> and enter your previously created username. In the project setting go to Permissions and grant the Admin user Execute Permissions.



Step 12: Run The Build.



Check the console output.



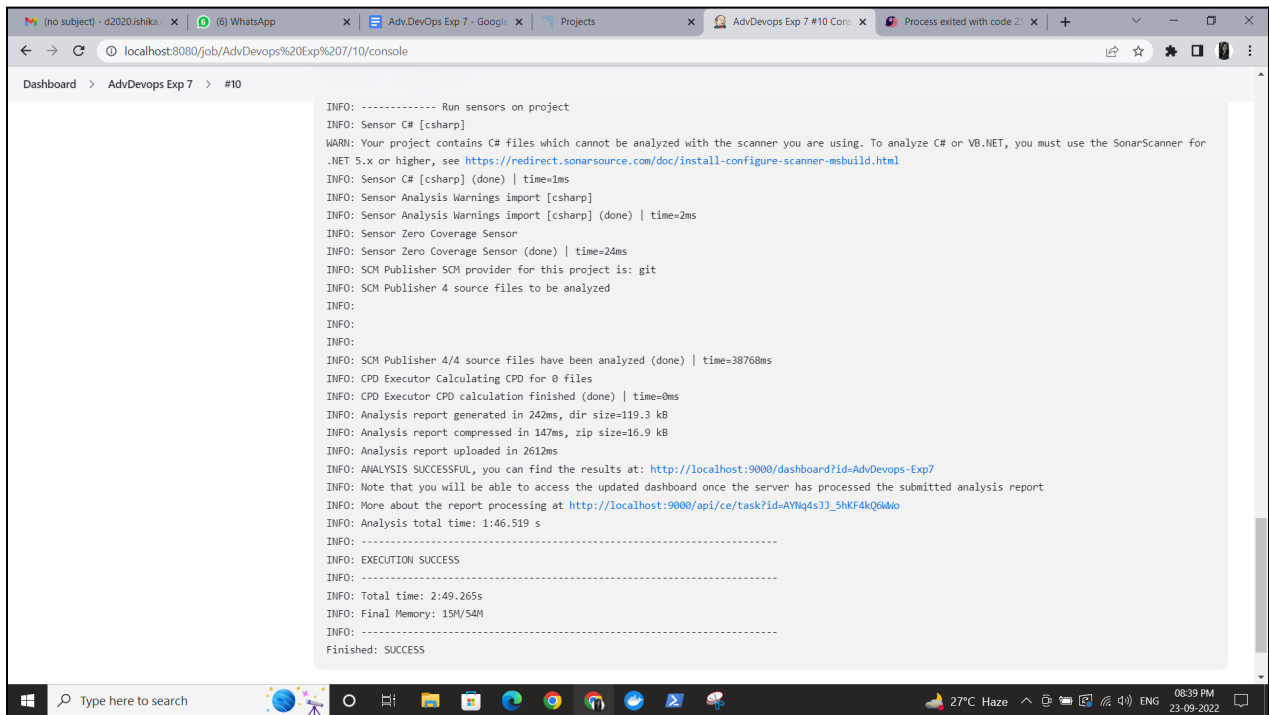
Jenkins Search (CTRL+K) Ishika Devare log out

Dashboard > AdvDevOps Exp 7 > #10

Status
Changes
Console Output
View as plain text
Edit Build Information
Git Build Data

Console Output

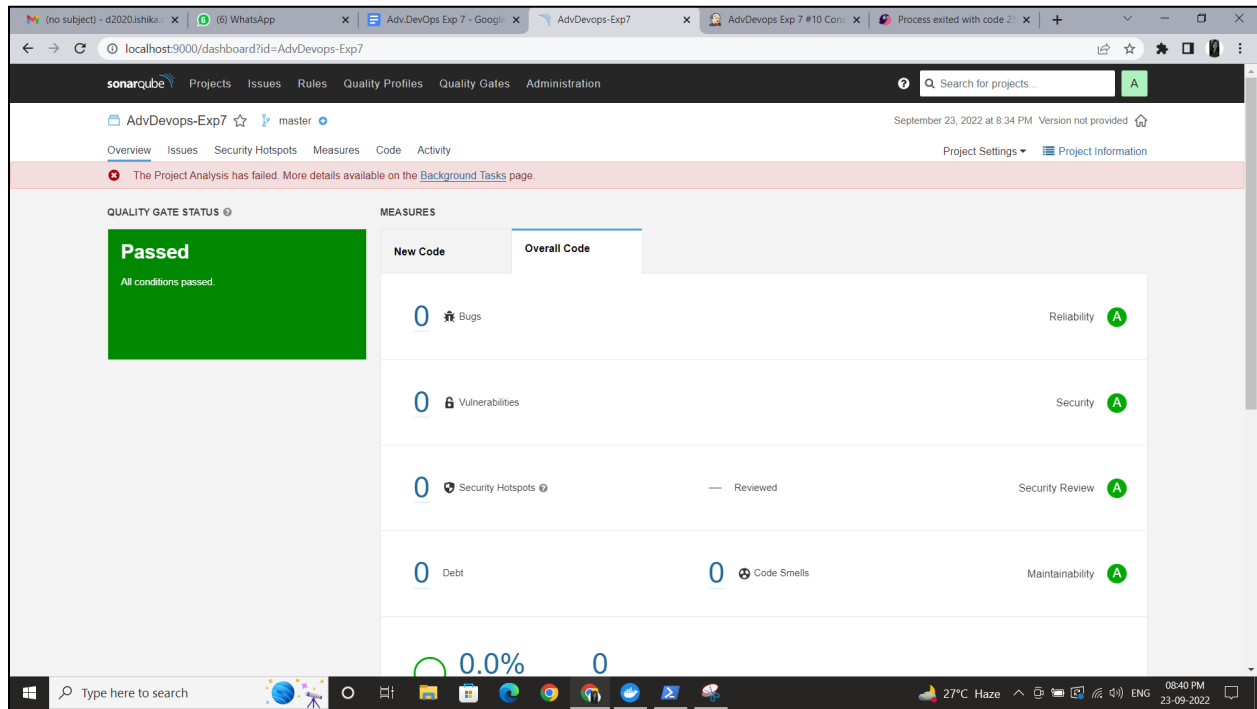
```
Started by user Ishika Devare
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\AdvDevOps Exp 7
The recommended git tool is: NONE
using credential 24a32ed5-dcdc-487b-b267-9798e4a71cf0
> git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\AdvDevOps Exp 7\.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 2.37.1.windows.1'
using GIT_ASKPASS to set credentials
> 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" # 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
[AdvDevOps Exp 7] $ C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\AdvDevOps-Exp7\bin\sonar-scanner.bat -Dsonar.projectKey=AdvDevOps-Exp7 -Dsonar.login=admin -Dsonar.hosturl=http://localhost:9000/ -Dsonar.password=ishal23 *-Dsonar.projectBaseDir=C:\ProgramData\Jenkins\jenkins\workspace\AdvDevOps Exp 7"
INFO: Scanner configuration file: C:\ProgramData\Jenkins\jenkins\tools\hudson.plugins.sonar.SonarRunnerInstallation\AdvDevOps-Exp7\bin\..\conf\sonar-scanner.properties
```



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=1ms
INFO: Sensor Analysis Warnings import [csharp]
INFO: Sensor Analysis Warnings import [csharp] (done) | time=2ms
INFO: Sensor Zero Coverage Sensor
INFO: Sensor Zero Coverage Sensor (done) | time=24ms
INFO: SCM Publisher SCM provider for this project is: git
INFO: SCM Publisher 4 source files to be analyzed
INFO:
INFO:
INFO:
INFO: SCM Publisher 4/4 source files have been analyzed (done) | time=38768ms
INFO: CPD Executor Calculating CPD for 0 files
INFO: CPD Executor CPD calculation finished (done) | time=0ms
INFO: Analysis report generated in 242ms, dir size=119.3 kB
INFO: Analysis report compressed in 147ms, zip size=16.9 kB
INFO: Analysis report uploaded in 2612ms
INFO: ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=AdvDevOps-Exp7
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/ce/task?id=AVHq4sJJ_5HKf4kQ6W6o
INFO: Analysis total time: 1:46.519 s
INFO: -----
INFO: EXECUTION SUCCESS
INFO: -----
INFO: Total time: 2:49.265s
INFO: Final Memory: 15M/54M
INFO: -----
Finished: SUCCESS
```

Step 13: Once the build is complete, check the project in SonarQube.



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

Conclusion - In this experiment, we created manual project in sonarqube and freestyle project in jenkins and integrated jenkins SAST to sonarQube.