

## Sonarqube with Jenkins

### Sonarqube:

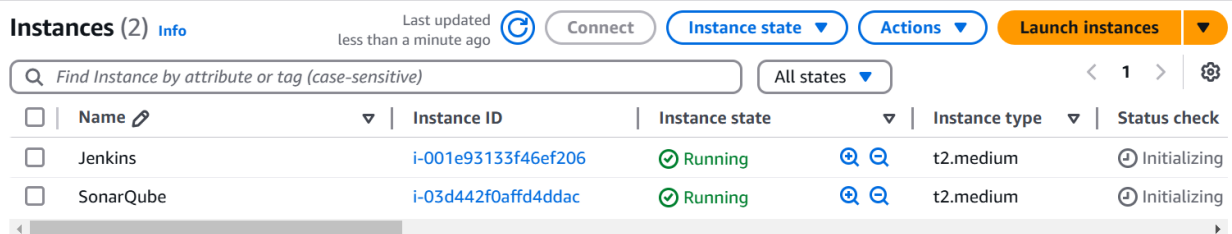
**SonarQube** is an open-source platform designed for continuous code quality and security inspection. It performs **static code analysis** to detect bugs, vulnerabilities, and code smells across various programming languages. SonarQube integrates seamlessly with development pipelines, providing actionable insights and maintaining code health through measures like code coverage, duplications, and maintainability ratings.

### Summary:

I have prepared a comprehensive document on SonarQube that guides you through the process of integrating SonarQube with Jenkins for seamless code analysis and quality management. To provide a practical demonstration, I have included an example application from GitHub, showcasing how to assess its code quality using SonarQube's quality gates within a Jenkins pipeline. This hands-on approach ensures a clear understanding of the integration process and the effective use of SonarQube for maintaining high code standards.

### Steps to follow:

1. Launch 2 Ubuntu Instance with atleast t2.medium with minimum 20GiB of storage.



The screenshot shows the AWS Management Console 'Instances' page. It displays two instances: 'Jenkins' and 'SonarQube'. Both are in a 'Running' state. The 'Jenkins' instance has ID 'i-001e93133f46ef206' and the 'SonarQube' instance has ID 'i-03d442f0affd4ddac'. Both are using 't2.medium' instance types. The 'Status check' for both is 'Initializing'.

Name	Instance ID	Instance state	Instance type	Status check
Jenkins	i-001e93133f46ef206	Running	t2.medium	Initializing
SonarQube	i-03d442f0affd4ddac	Running	t2.medium	Initializing

2. Update **the apt in both the servers** using “sudo apt update”

3. Setting up Sonarqube using Docker

3.1 Install docker in your sonarqube server

- sudo apt install docker.io -y

```
ubuntu@ip-172-31-38-56:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-38-56:~$
```

3.2 Running sonarqube as a docker container

- docker run -d -p 9000:9000 sonarqube:lts-community

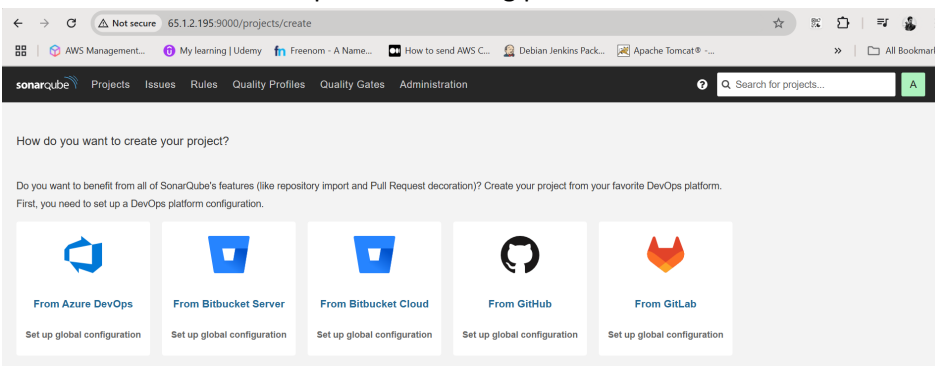
3.3 Verify the sonarqube container is in running status by executing “docker ps”

```

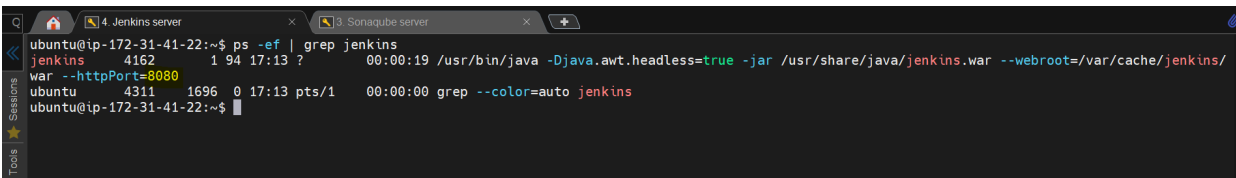
ubuntu@ip-172-31-38-56:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
ubuntu@ip-172-31-38-56:~$ sudo docker run -d -p 9000:9000 sonarqube:lts-community
docker: invalid reference format.
See 'docker run --help'.
ubuntu@ip-172-31-38-56:~$ docker run -d -p 9000:9000 sonarqube:lts-community
docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://%2Fvar%2Frun%2Fdocker.sock/_ping": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
ubuntu@ip-172-31-38-56:~$ sudo chmod 777 /var/run/docker.sock
ubuntu@ip-172-31-38-56:~$ docker run -d -p 9000:9000 sonarqube:lts-community
Unable to find image 'sonarqube:lts-community' locally
lts-community: Pulling from library/sonarqube
6414378b6477: Pull complete
17da8ec43a12: Pull complete
d12988e90d61: Pull complete
f4d133ca2b7f: Pull complete
143733ae87a4: Pull complete
d021fe0fda1e: Pull complete
bc52ff982b3d: Pull complete
4f4fb700ef54: Pull complete
Digest: sha256:aac4b3bdd2ae29c95e1a4c07bc50f0cc81d1914019bc609e92dee25e72ff2e6d
Status: Downloaded newer image for sonarqube:lts-community
a627c20f6287ea45fc1bfc2b05372e7fd6945d499815973ee25e750822dc0fc8
ubuntu@ip-172-31-38-56:~$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
a627c20f6287   sonarqube:lts-community   "/opt/sonarqube/dock..."   2 minutes ago   Up 2 minutes   0.0.0.0:9000->9000/tcp, :::9000->9000/tcp   nifty_ganguly
ubuntu@ip-172-31-38-56:~$

```

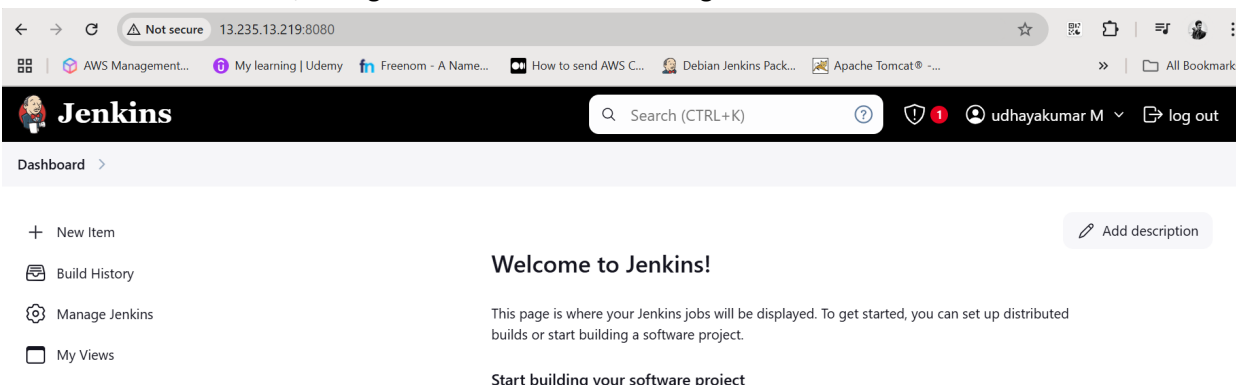
### 3.4 Now access the Sonarqube server using port 9000



### 4. Install Jenkins in our Jenkins server



Once installation is done, configure the Jenkins user and login to the Jenkins console



### 4.2 Install the required plugins mentioned below

Click on Manage Jenkins => Plugins => Available plugins and select the plugin and Click on Install

- SonarQube Scanner
- Eclipse Temurin installerVersion1
- Stage View

## 4.3 Once Plugins are installed successfully, now need to configure it

Manage Jenkins => Tools =>

### 4.3.1 JDK

Dashboard > Manage Jenkins > Tools

Add JDK

≡ JDK

Name

jdk17

☒ Install automatically ?

≡ Install from adoptium.net ?

Version ?

jdk-17.0.11+9

Add Installer ▾

### 4.3.2 sonarqube

Dashboard > Manage Jenkins > Tools

Add SonarQube Scanner

≡ SonarQube Scanner

Name

sonar-scanner

☒ Install automatically ?

≡ Install from Maven Central

Version

SonarQube Scanner 6.1.0.4477

Add Installer ▾

### 4.3.3 Maven

Dashboard > Manage Jenkins > Tools

Add Maven

Maven

Name

maven3

☒ Install automatically ?

Install from Apache

Version

3.9.9

Add Installer

## 5.Generate a sonarqube token to integrate it with Jenkins

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration ? Search for projects... A

Administration

Configuration Security Projects System Marketplace

Users [Create User](#)

Create and administer individual users.

Search by login or name...

	SCM Accounts	Last connection	Groups	Tokens
A Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0

**Tokens of Administrator**

Generate Tokens

Name Expires in

Enter Token Name 30 days [Generate](#)

New token "Token" has been created. Make sure you copy it now, you won't be able to see it again!

[Copy](#) `squ_e1d4bf6894838311bd82e788d9b63`

Name	Type	Project	Last use	Created	Expiration	
Token	User		Never	November 28, 2024	December 28, 2024	<a href="#">Revoke</a>

## 6.Add the generated sonar token with Jenkins credentials as secret text as credential type

Dashboard > Manage Jenkins > Credentials > System > Global credentials (unrestricted) >

Global credentials (unrestricted) [+ Add Credentials](#)

Credentials that should be available irrespective of domain specification to requirements matching.

ID	Name	Kind	Description
sonar-token	sonar-token	Secret text	sonar-token

## 7.Now integrate the sonarqube server with Jenkins server

Manage Jenkins => System => Add sonarqube

Dashboard > Manage Jenkins > System >

#### SonarQube installations

List of SonarQube installations

Name

sonar-server

Server URL

Default is `http://localhost:9000`

`http://65.1.2.195:9000/`

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar-token

+ Add

Once added click on Apply and Save

8. Now we can start with pipeline

8.1 Create one Jenkins job with the type as pipeline.

Dashboard >

+ New Item

Build History

Manage Jenkins

My Views

Add description

All +

S	W	Name	Last Success	Last Failure	Last Duration
🔄	☀️	Sonarqube pipeline	N/A	N/A	N/A

8.2 Git Hub repo <https://github.com/jaiswaladi246/FullStack-Blogging-App/tree/main>

### 8.3 Pipeline script

```
pipeline {
  agent any

  tools {
    maven 'maven3'
    jdk 'jdk17'
  }

  environment{
    SCANNER_HOME = tool 'sonar-scanner'
  }

  stages {
    stage('Git Checkout') {
      steps {
        git branch: 'main', url: 'https://github.com/jaiswaladi246/FullStack-Blogging-App.git'
      }
    }

    stage('Compile Code') {
      steps {
        sh "mvn compile"
      }
    }

    stage('Executing Testcases') {
      steps {
        sh "mvn test"
      }
    }

    stage('Sonarqube Analysis') {
      steps {
        withSonarQubeEnv('sonar-server') {
          sh "' $SCANNER_HOME/bin/sonar-scanner -
Dsonar.projectName=bloggingApp -Dsonar.projectKey=bloggingApp \
```

Configure the pipeline with the above script also make necessary modifications on the script as per your tools and system configurations and run the pipeline

Search (CTRL+K)

1

udhayakumar M

log out

Dashboard > Sonarqube-pipeline >

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Stages

Rename

Sonarqube-pipeline

Add description

Stage View

	Declarative: Tool Install	Git Checkout	Compile Code	Executing Testcases	Sonarqube Analysis
Average stage times: (Average full run time: ~54s)	12s	1s	4s	10s	10s
#5 Nov 28 23:58 No Changes	159ms	1s	9s	21s	21s

## 9. Verify the sonarQube project

sonarqube

Projects

Issues

Rules

Quality Profiles

Quality Gates

Administration

Search for projects...

A

My Favorites

All

Search by project name or key

Create Project

Home

1 project(s)

Perspective: Overall Status

Sort by: Name

bloggingApp

Passed

Last analysis: 5 minutes ago

Bugs

Vulnerabilities

Hotspots Reviewed

Code Smells

Coverage

Duplications

Lines

C

A

E

A

37.1%

38.4%

3.1k

S HTML, Ja...

1 of 1 shown

Now you can able to verify the findings on the code level vulnerabilities, Also configure the quality gate as per your requirements