

Project on SonarQube Integration with Jenkins server

Project Documentary

Divya Vasant Satpute

DREAMS SOURCE learnwithdivya.hashnode.dev

Project on SonarQube server Integration with Jenkins Server

Table content



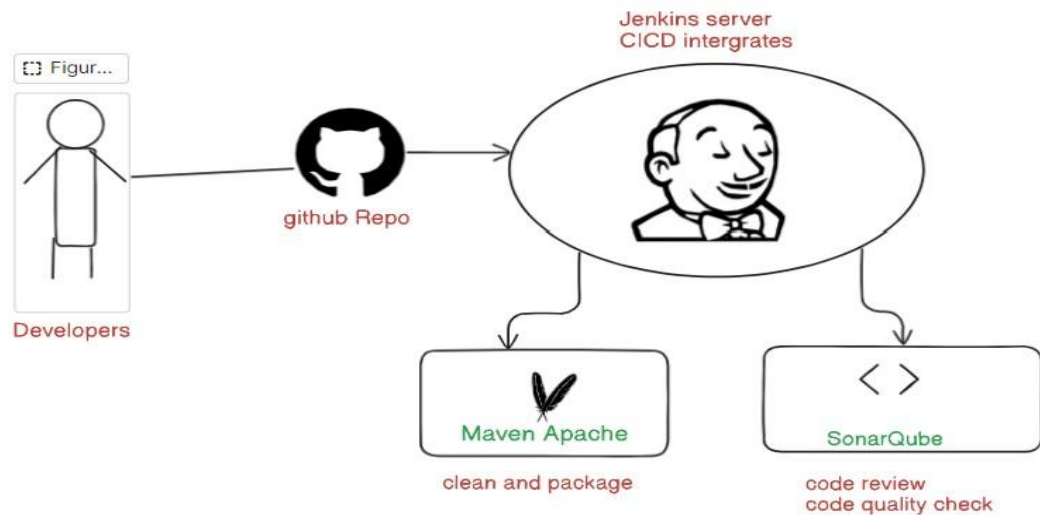
SonarQube Integration with Jenkins



1. Introduction
 - Overview
 - Objectives
2. Components
 - Jenkins Server
 - SonarQube
 - SonarQube Scanner Plugin for Jenkins
3. Setup and Configuration
 - Prerequisites
 - Jenkins Installation
 - SonarQube Installation
 - Integration Steps
 - Install SonarQube Scanner Plugin
 - Configure SonarQube Server in Jenkins
 - Configure Jenkins Build Job
 - Run Jenkins Job
4. Benefits
 - Improved Code Quality
 - Efficient Development
 - Visibility
5. Conclusion

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

SonarQube Integration with Jenkins Server



Setting Up Jenkins and Sonar Server

On AWS EC2 Machine Deploy Jenkins Server

Prerequisite (Step 1)

- Jenkins installation on AWS EC2
- Create an EC2 instance with ubuntu Linux AMI
- Connect to your EC2 instance
- Update all packages by following command:

```
$sudo apt update -y
```

- Install java by following command:
\$ sudo apt install openjdk-11-jdk -y

Step 2

- Jenkins installation on AWS EC2 Using apt
- Add Jenkins to your apt Repository using following command

```
$sudo wget -q -O /usr/share/keyrings/jenkins-keyring.asc https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key  
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/"  
| sudo tee /etc/apt/sources.list.d/jenkins.list > /dev/null  
sudo apt-get update  
sudo apt-get install Jenkins
```

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

- Start and Enable Jenkins Service

```
$sudo systemctl start Jenkins
```

```
$sudo systemctl enable jenkins
```

```
$sudo systemctl status Jenkins
```

- Get the initial administration password

```
$sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Step 3

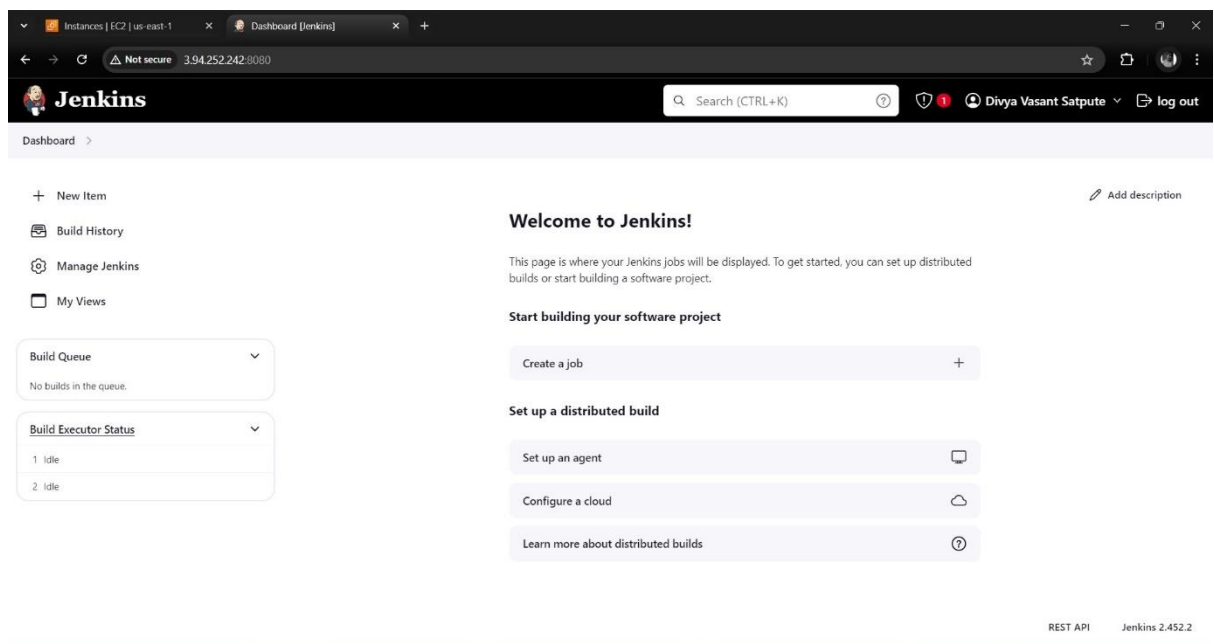
Open your EC2 instance public IP (https://<public_IP>:8080/) along with port 8080 in your favorite browser. And provide the administration password obtained during the installation.

Note: Make sure you enable 8080 Port in Security Group Inbound Rules.

Provided password which we have copied to unlock Jenkins.

Select "Install Suggested Plugins" Card (It will install those plugins)

Here is your Jenkins server all set



On AWS EC2 Machine Deploy SonarQube Server

Environment Setup

Pre-requisites:

Java is installed on your Machine

if SonarQube 7.6 ----> Java 1.8 version is installed

@Divya_vasant_satpute
Learnwithdivya.hashnode.dev

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

if SonarQube 7.6 -----> java 11 version is Installed

Note: We can Check this Compatability in official SonarQube Website

Hardware Requirements

Minimum RAM: 2 GB



Step for deploy SonarQube

Create EC2 Instance with 4 GB Ram (t2.medium)

Connect with EC2 instance using Git-Bash

Switch to Root User

```
$sudo -i
```

Update packages

```
$sudo apt update -y
```

Install wget package using following command

```
$sudo apt install wget -y
```

change directory

```
$cd /opt
```

Java installation

```
$sudo apt install openjdk-11-jdk -y
```

```
$java -version Install java openjdk
```

unzip package Install

```
$sudo apt install unzip -y
```

install SonarQube zip file change version according to your version

```
$wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-8.4.0.35506.zip
```

unzip file

```
$unzip sonarqube-8.4.0.35506.zip
```

Note: SonarQube Server Will Not Run with Root User

we have to create **new user** called **Sonar**

create new user

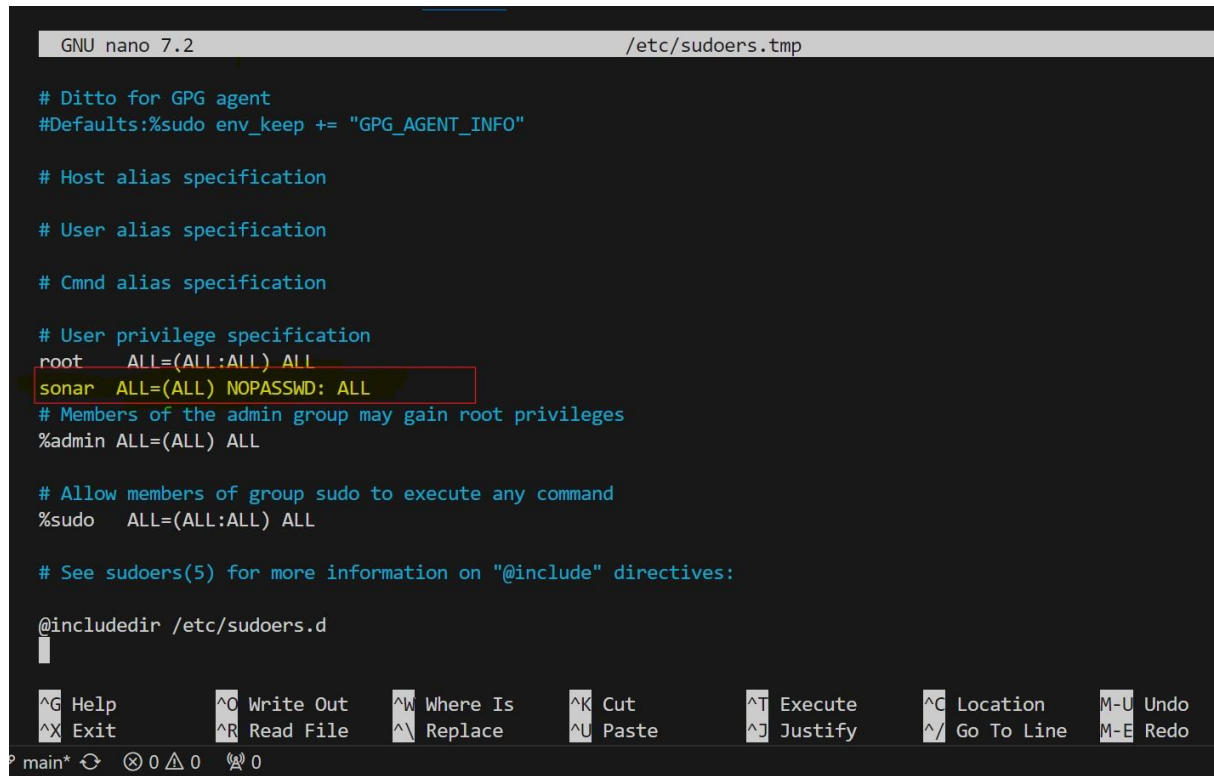
PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

\$useradd sonar

\$visudo

File be look like below image we have to configure user n passwd

sonar ALL=(ALL) NOPASSWD: ALL



```
GNU nano 7.2 /etc/sudoers.tmp

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL
sonar    ALL=(ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

@include /etc/sudoers.d

```

change Ownership for Sonar Folder

\$chown -R sonar:sonar /opt/sonarqube-8.4.0.35506

Change file permission

\$chmod -R 775 /opt/sonarqube-8.4.0.35506

Now switch to Sonar user

\$su – sonar

Go to SonarQube folder

\$cd sonarqube-8.4.0.35506

\$cd bin

\$cd linux-x86-64

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

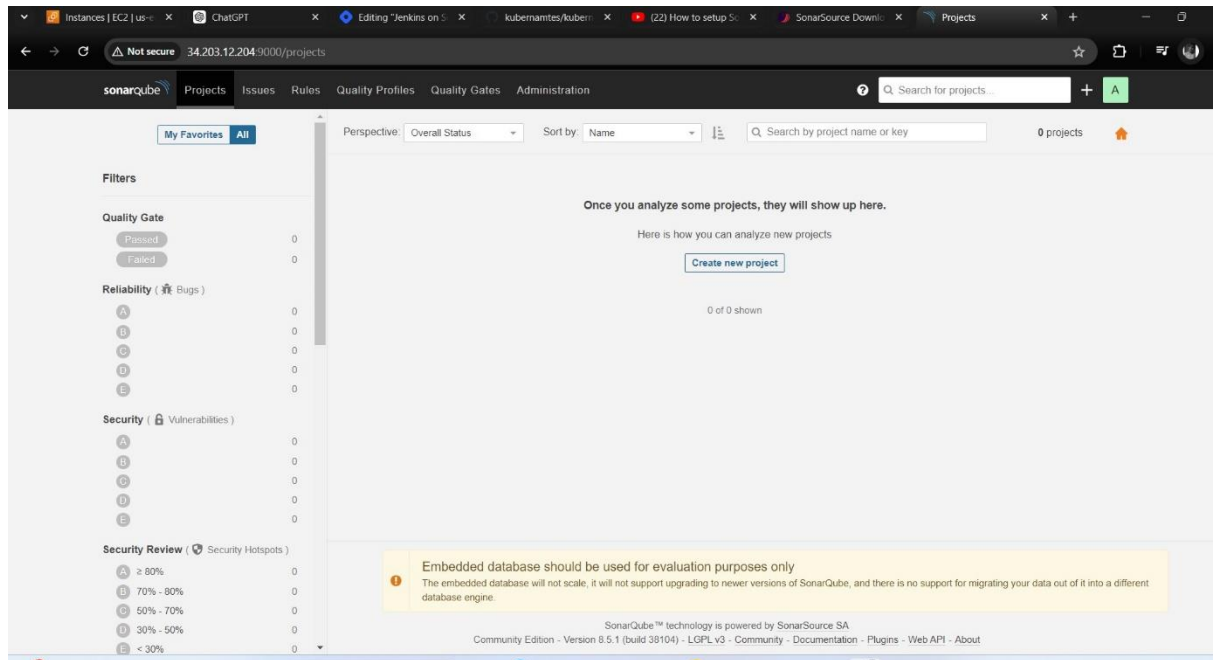
`./sonar.sh start`

Congratulations to yourself!

Access your SonarQube:

URL: `http:// EC2-VM-Public-IP :9000/`

Note: SonarQube runs on 9000 port by Default. Enable this port in security group as custom TCP



Now Sonar Server with Jenkins Integration

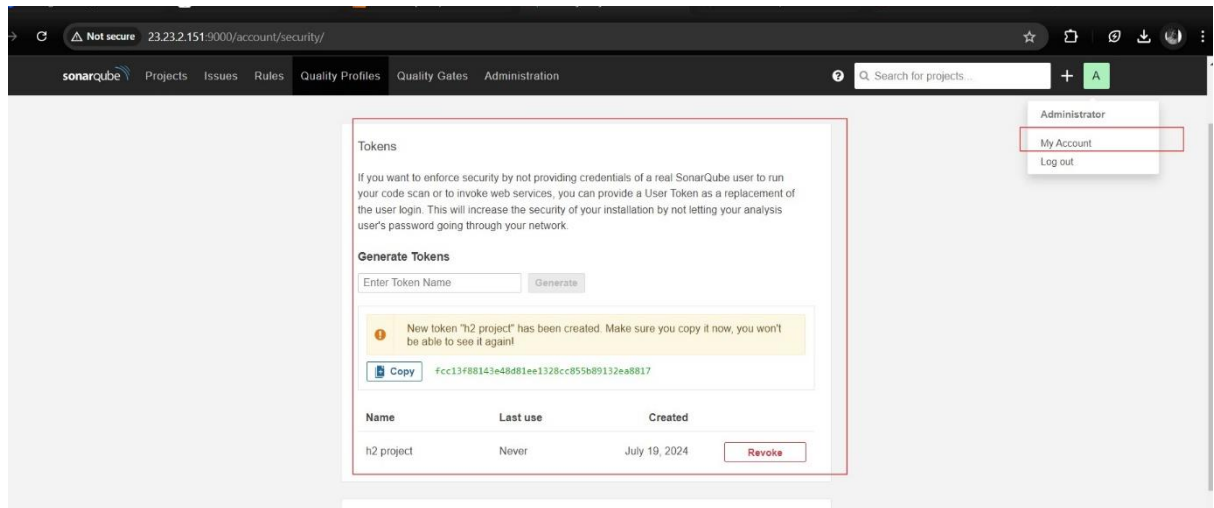
Pre - Requisites

1. SonarQube Server (already we done that using above steps)
2. Jenkins Server (already we done that using above steps)
3. On SonarQube Server generate a Token

Steps configure Token

Go to Sonar --->Login ---> Click On profile ----> My Account ---> Security -----
>Generate Token

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER



4. On Jenkins Server
5. Install Apache maven
6. Install Sonar Plugins
7. Configure SonarQube Credentials
8. Install Sonar Scanner
9. Run Jenkins Pipeline Job
10. Execute Below Commands in Jenkins Server VM CLI

\$sudo su

\$cd /opt

\$wget <https://dlcdn.apache.org/maven/maven-3/3.8.8/binaries/apache-maven-3.8.8-bin.tar.gz>

\$tar -xvf apache-maven-3.8.8-bin.tar.gz

Now go to Jenkins Dashboard

Install Scanner Plugin

- Click on Manage Jenkins
- Click on Plugins then go to Available
- Click On SonarQube Scanner Plugin
- Install It

Configure SonarQube Server

- Click on Manage Jenkins
- Click on configure System
- Go to SonarQube Server
- Add SonarQube Server

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

Name: Sonar -server -8.4

Server URL: (give Your Sonar Server URL)

Add Sonar Server Token (Token We Should add As Secrete Text)

Save it

Configure SonarQube Server Scanner

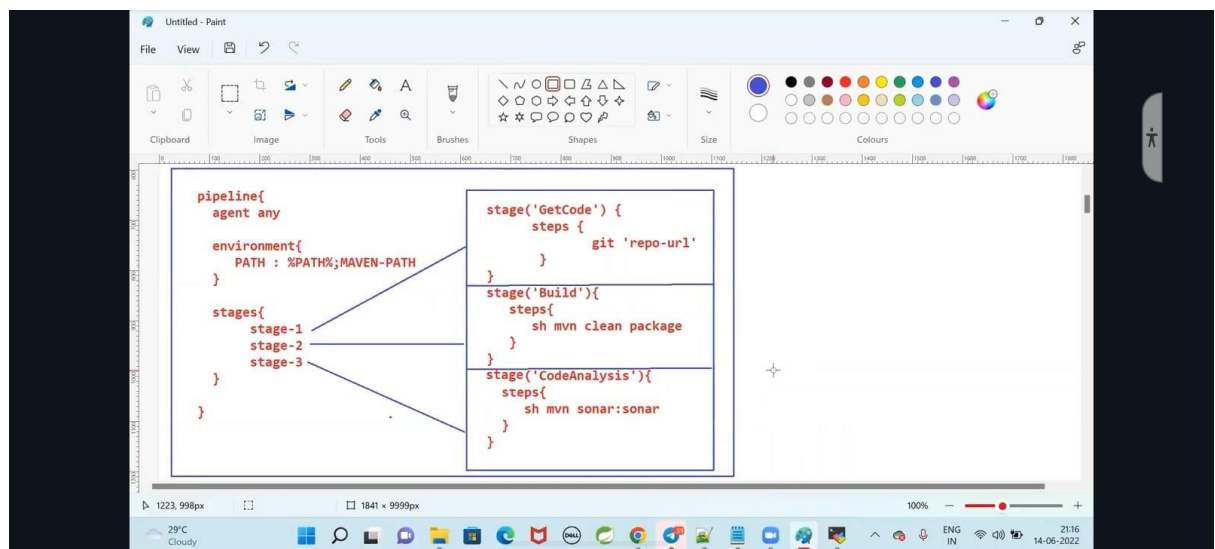
- Click on Manage Jenkins
- Click on Global Tools Configuration
- Click on SonarQube Scanner

Name: Sonar -Scanner-4.7

Select Sonar Version (Sonar -Scanner-4.7)

save it

Create Jenkins Pipeline



PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

```
pipeline {
    agent any

    environment {
        PATH = "$PATH:/opt/apache-maven-3.8.8/bin" // Corrected Maven path
    }

    stages {
        stage("GetCode") {
            steps {
                git "https://github.com/divyasatpute/java-maven-app.git"
            }
        }

        stage("Build") {
            steps {
                sh "mvn clean package"
            }
        }

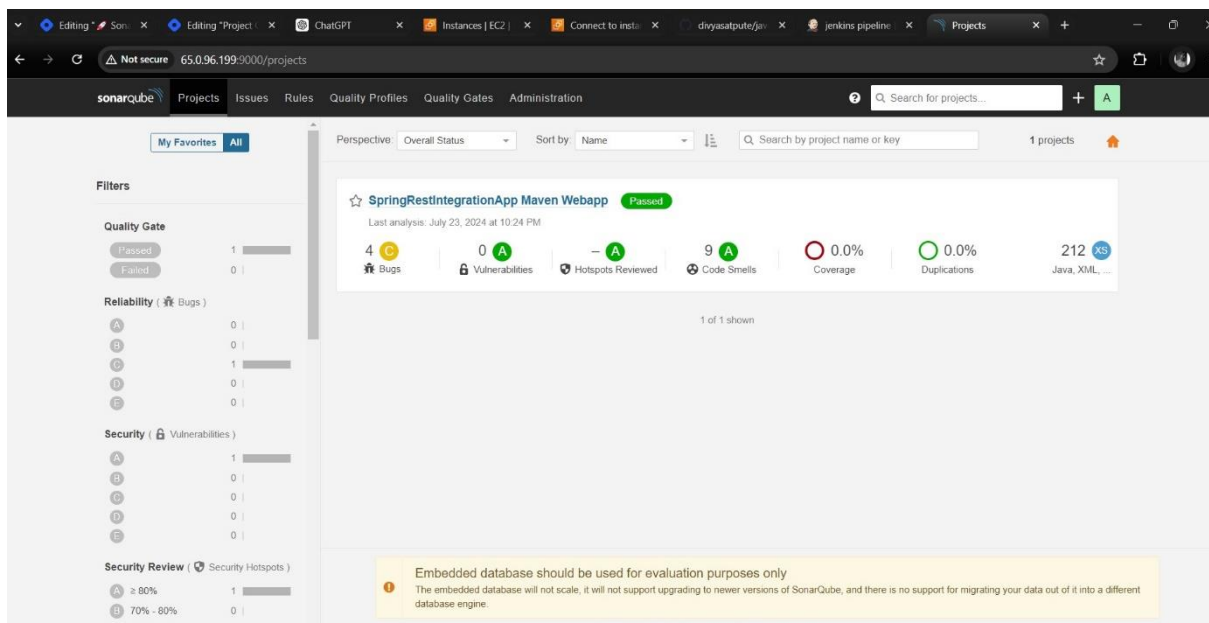
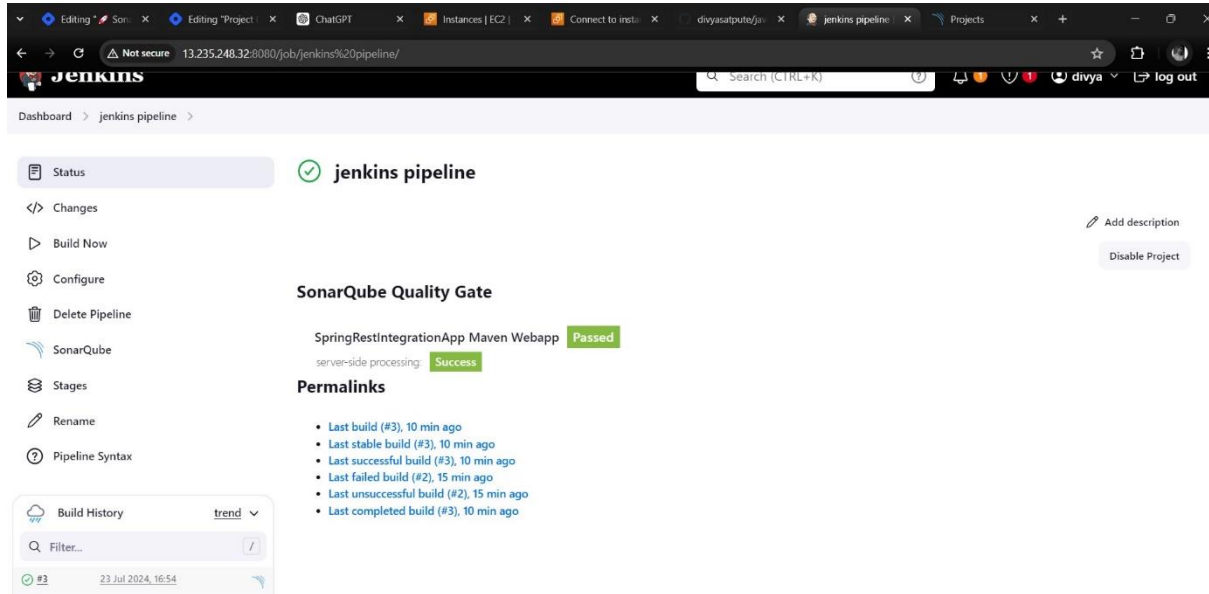
        stage("SonarQube Analysis") {
            steps {
                withSonarQubeEnv('sonarqube-8.4') {
                    sh "mvn sonar:sonar"
                }
            }
        }
    }
}
```

PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

Run Job

- click on New Item
- Give Name---->> Select Pipeline ---->Click on OK
- Paste Pipeline
- Apply and Save

Here are my Some test (POC) Results



PROJECT ON SONARQUBE INTEGRATION WITH JENKINS SERVER

Hurray you completed this project

Hurray! Celebrating Successful Jenkins and SonarQube Integration! 🎉

NOTE: if you face some issue comment below and connect with me on Linkdin
@<https://www.linkedin.com/in/divya-satpute-68666a300/>

Congratulations on setting up Jenkins and SonarQube integration smoothly! 🎉 This achievement highlights your dedication to improving code quality and streamlining development processes. With Jenkins orchestrating builds and SonarQube providing insightful code analysis, your team is empowered to deliver high-quality software efficiently.

Keep up the fantastic work! ✨ Here's to more milestones and continuous improvement in your software development journey. Cheers to innovation and excellence! 🙌