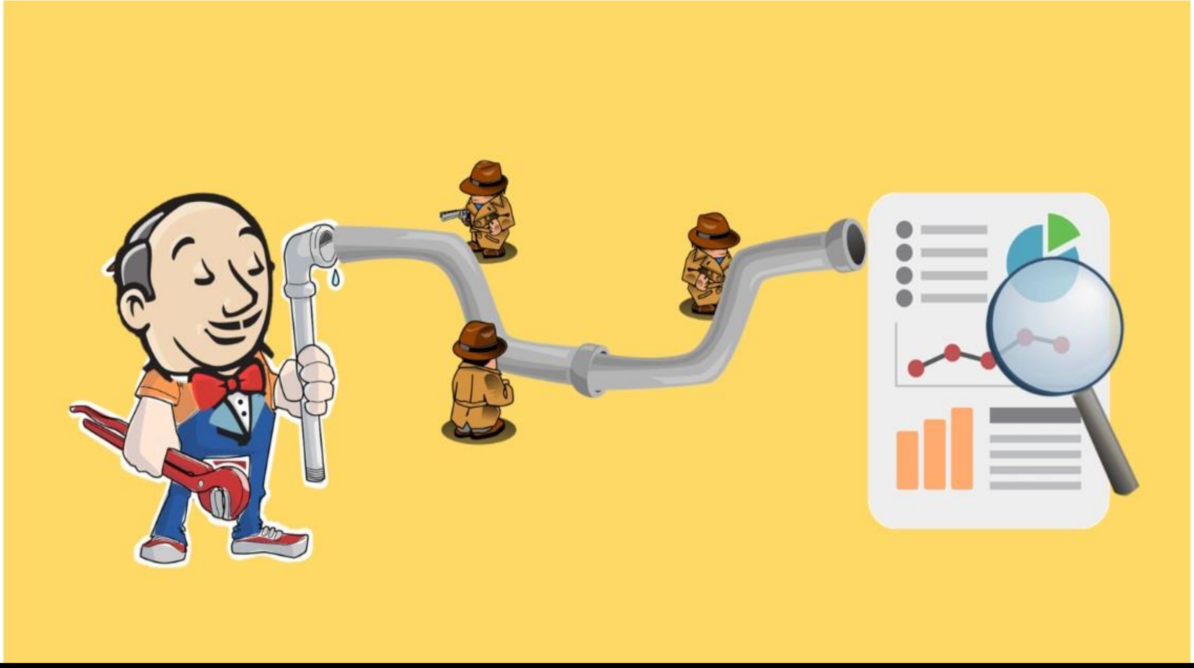


Project on SonarQube Integration with Jenkins server

Project Documentary

Divya Vasant Satpute

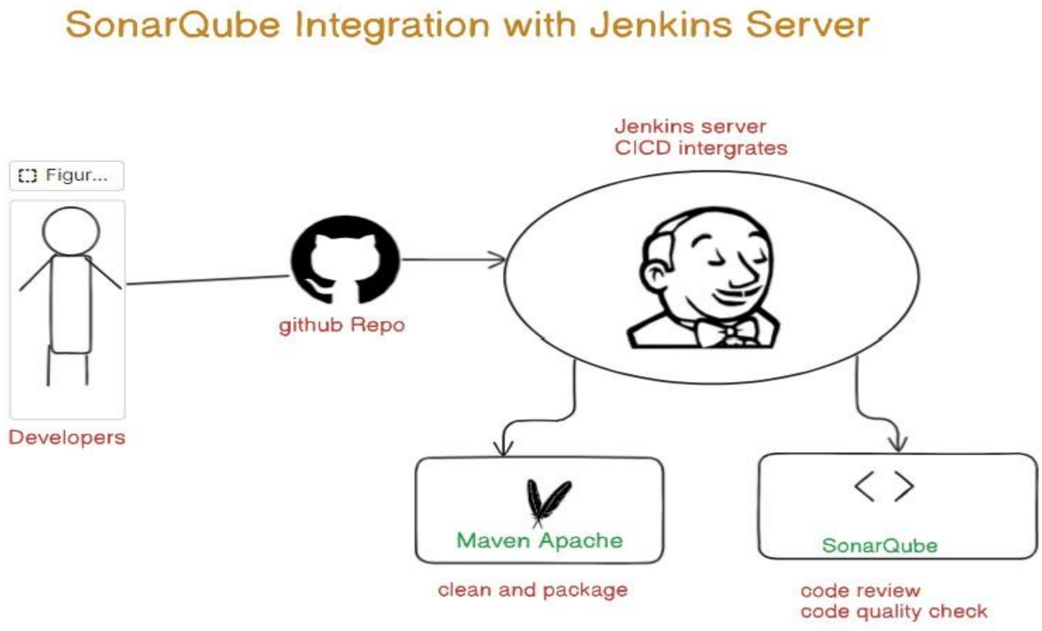
DREAMS SOURCE learnwithdivya.hashnode.dev

Project on SonarQube server Integration with Jenkins Server

Table content

🛠️ SonarQube Integration with Jenkins 🚀

PROJECT ON 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
```

- 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

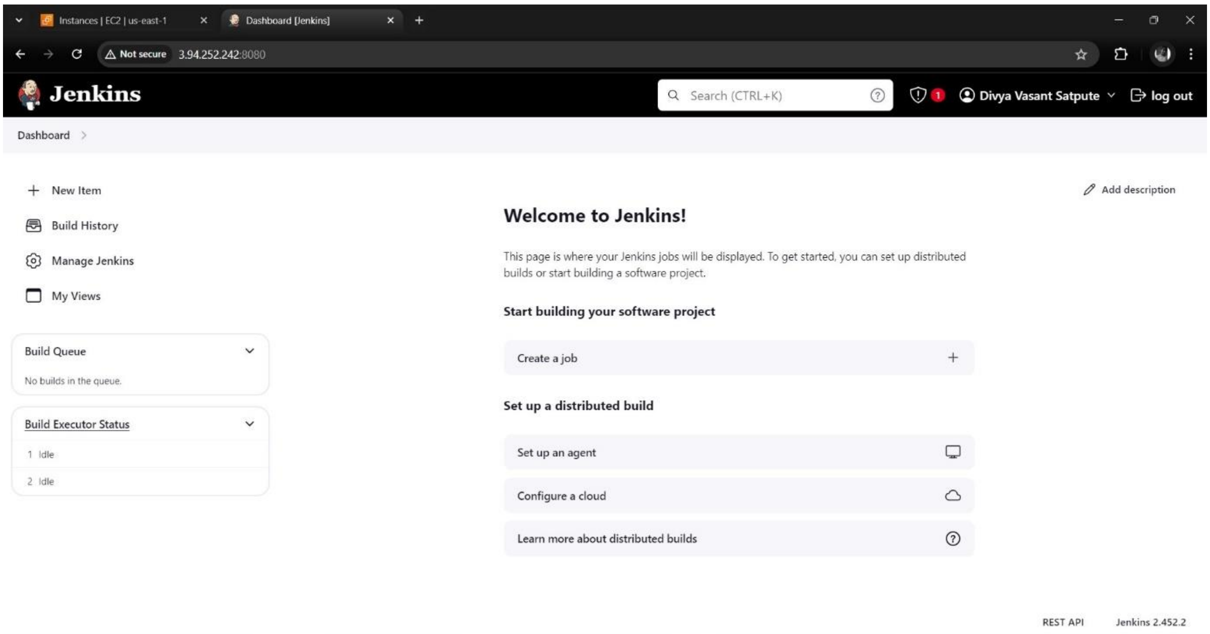
Open your EC2 instance public IP (https://<public_IP>:8080/) along with port 8080 in your favorite browser. And provide the administration password obtained during the installation.

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



Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

No builds in the queue.

On AWS EC2 Machine Deploy SonarQube Server

Environment Setup🖥️

Pre-requites:

Java is installed on your Machine

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

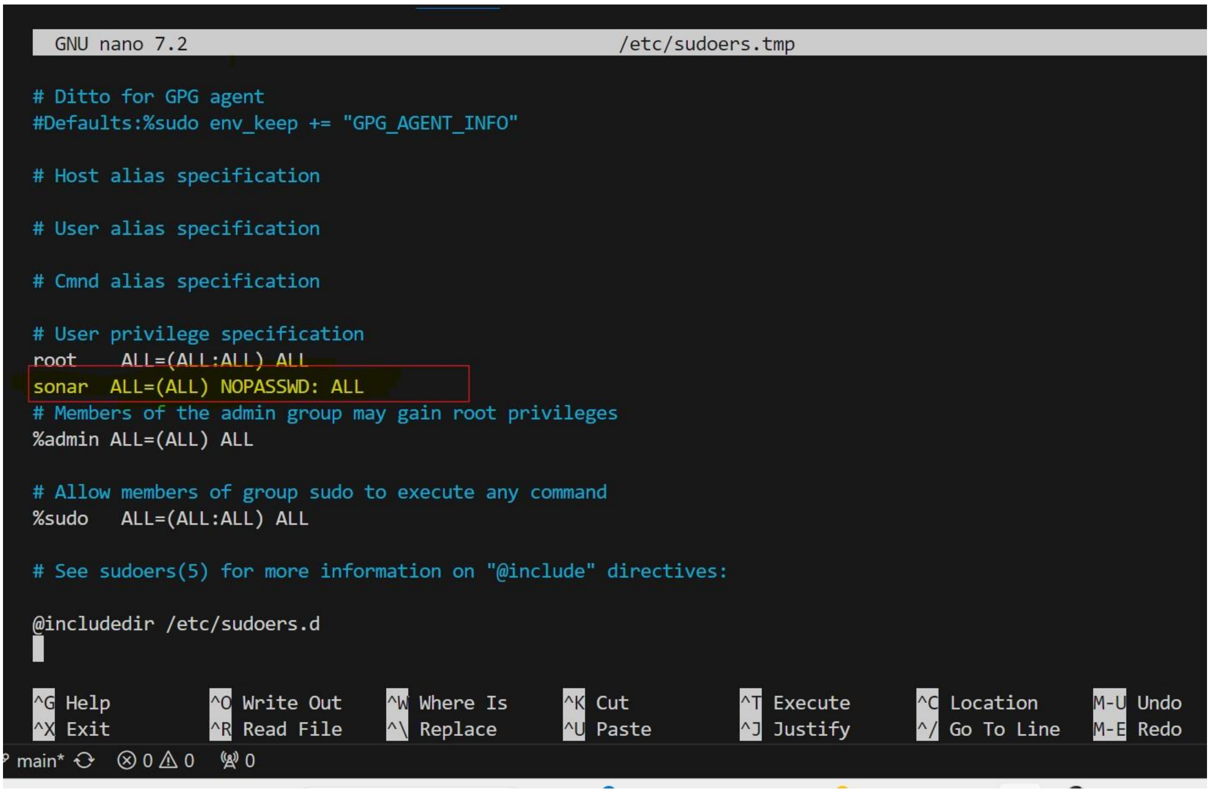
create new user

$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:

@includedir /etc/sudoers.d


^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo
```

main* ⟳ ⊗ 0 ⚠ 0 0

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

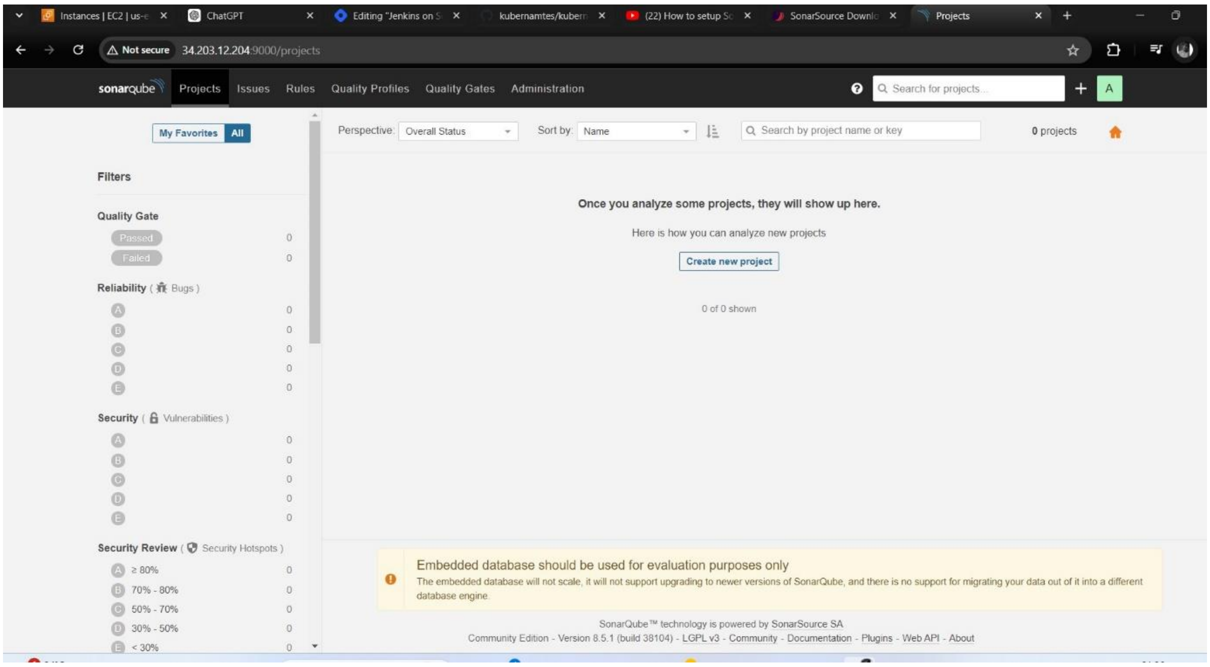
$./sonar.sh start

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



Once you analyze some projects, they will show up here.

Here is how you can analyze new projects

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA

Community Edition - Version 8.5.1 (build 38104) - LGPL v3 - Community - Documentation - Plugins - Web API - About

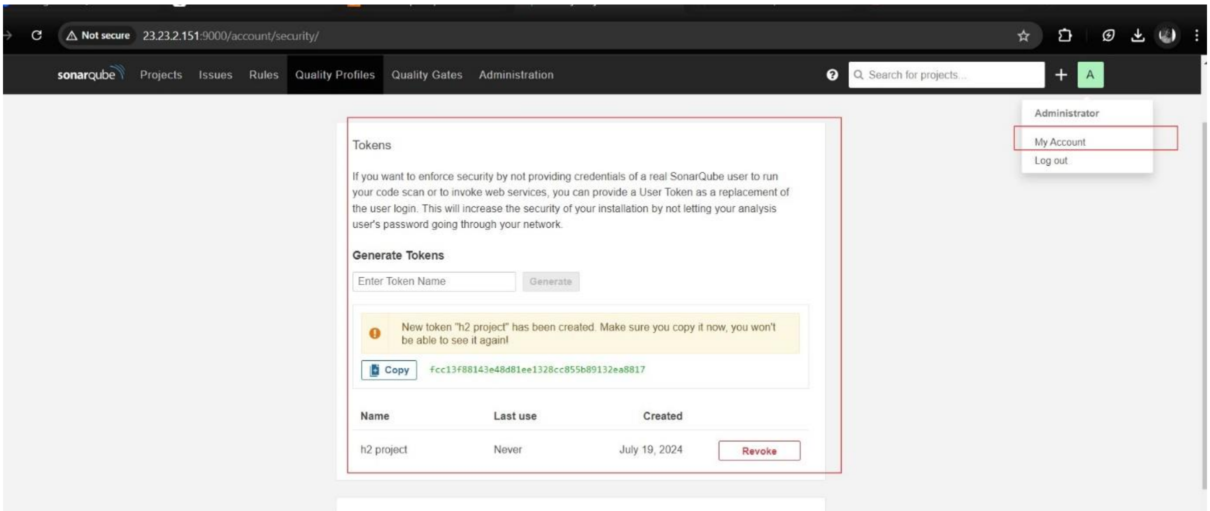
Now Sonar Server with Jenkins Integration

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



Tokens

If you want to enforce security by not providing credentials of a real SonarQube user to run your code scan or to invoke web services, you can provide a User Token as a replacement of the user login. This will increase the security of your installation by not letting your analysis user's password going through your network.

Generate Tokens

Enter Token Name | Generate

New token "h2 project" has been created. Make sure you copy it now, you won't be able to see it again!

Copy `fcc13f88143e48d81ee1328cc855b89132ea8817`

Name	Last use	Created	
h2 project	Never	July 19, 2024	Revoke

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

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

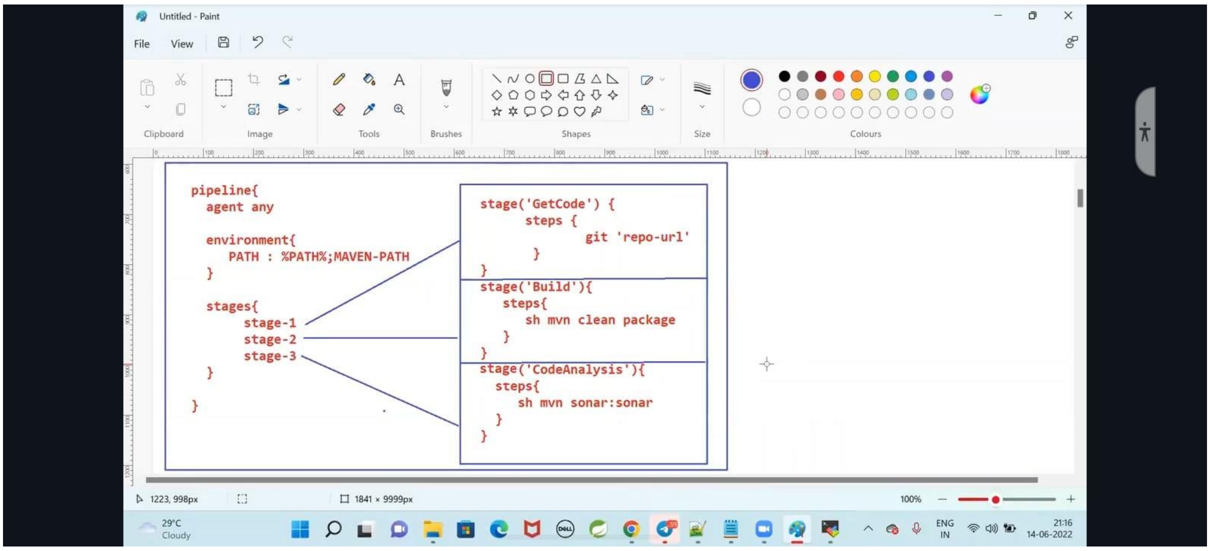
- Click on Manage Jenkins
- Click on Global Tools Configuration
- Click on SonarQube Scanner

Name: Sonar -Scanner-4.7

Select Sonar Version (Sonar -Scanner-4.7)

save it

Create Jenkins Pipeline



```
pipeline{
  agent any

  environment{
     PATH : %PATH%;MAVEN-PATH
  }

  stages{
       stage-1
       stage-2
       stage-3
  }

}
```

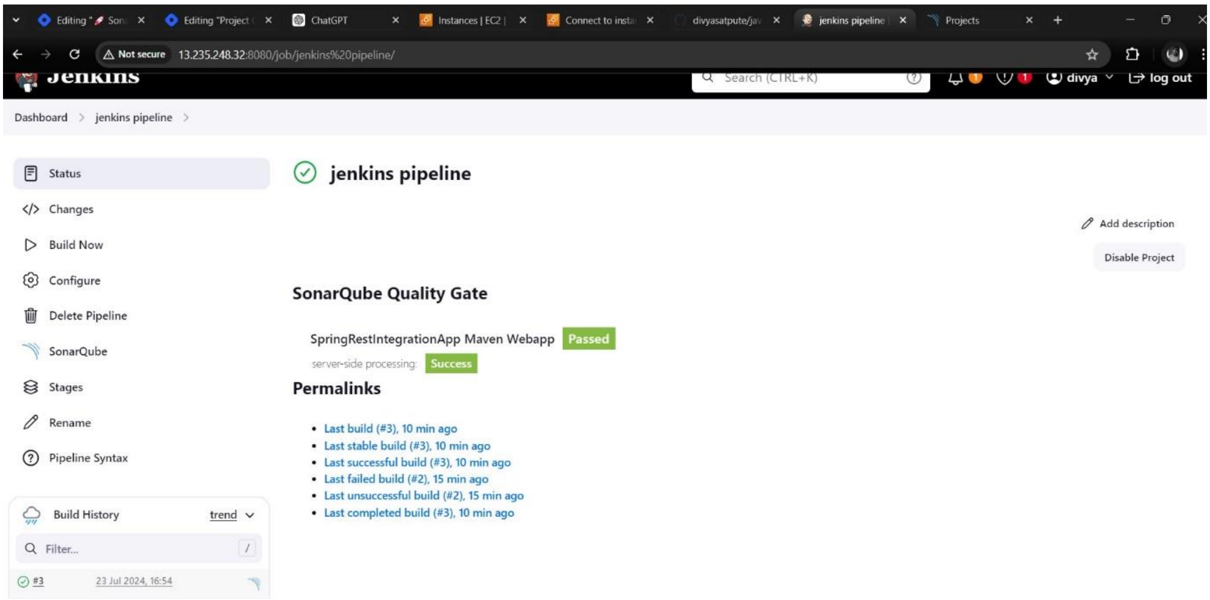
```
stage('GetCode') {
      steps {
              git 'repo-url'
       }
}
stage('Build'){
   steps{
      sh mvn clean package
   }
}
stage('CodeAnalysis'){
  steps{
     sh mvn sonar:sonar
  }
}
```

```
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"
                }
            }
        }
    }
}
```

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



13.235.248.32:8080/job/jenkins%20pipeline/

Dashboard > jenkins pipeline >

jenkins pipeline

SonarQube Quality Gate

SpringRestIntegrationApp Maven Webapp Passed

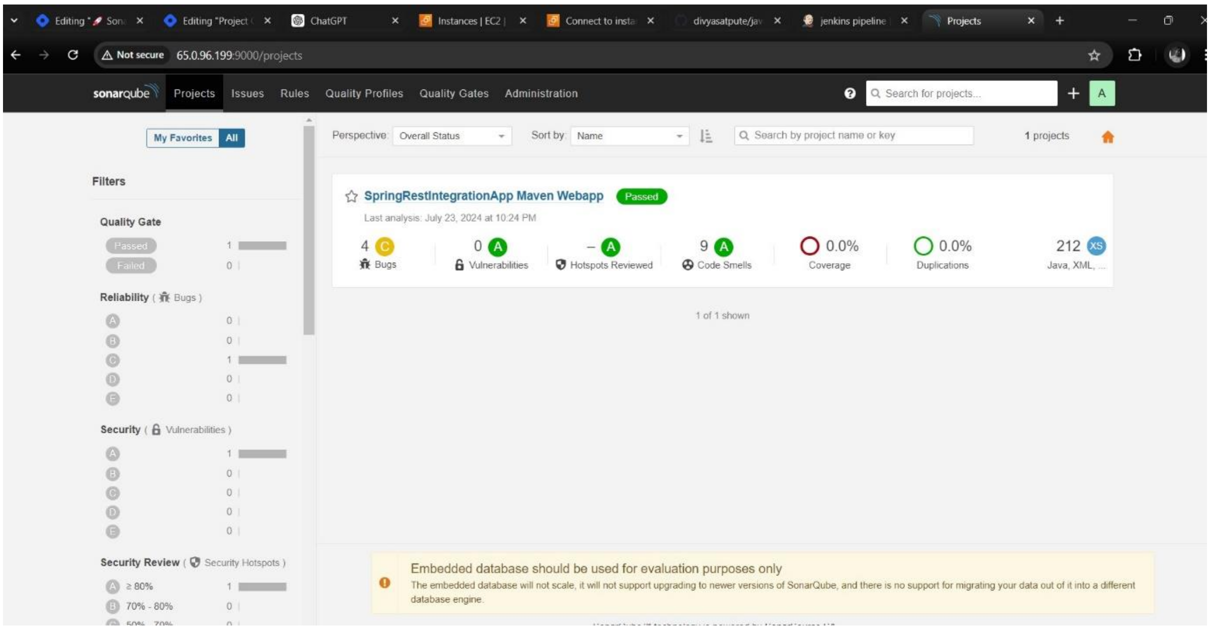
server-side processing: Success

Permalinks

- Last build (#3), 10 min ago
- Last stable build (#3), 10 min ago
- Last successful build (#3), 10 min ago
- Last failed build (#2), 15 min ago
- Last unsuccessful build (#2), 15 min ago
- Last completed build (#3), 10 min ago

Build History

#3 23 Jul 2024, 16:54



65.0.96.199:9000/projects

SpringRestIntegrationApp Maven Webapp Passed

Last analysis: July 23, 2024 at 10:24 PM

4 C	0 A	– A	9 A	0.0%	0.0%	212 XS
Bugs	Vulnerabilities	Hotspots Reviewed	Code Smells	Coverage	Duplications	Java, XML, ...

1 of 1 shown

Embedded database should be used for evaluation purposes only

The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

Filters

Quality Gate

Passed	1
Failed	0

Reliability (Bugs)

A	0
B	0
C	1
D	0
E	0

Security (Vulnerabilities)

A	1
B	0
C	0
D	0
E	0

Security Review (Security Hotspots)

A ≥ 80%	1
B 70% - 80%	0
C 50% - 70%	0

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! 🥂