

# Mini Project(Zomato)

**Author:** Krishna Prasad Janney

**Email Id:** Janneyk92@gmail.com

**Clean:** IT WILL CLEAN THE WORKSPACE IN JENKINS JOB

**CODE:** WE WILL GET THE CODE FROM GITHUB TO SERVER

**SONAR:** IT WILL SCAN THE ENTIER CODE AND DISPLAY THE BUGS AND CODE SMELLS

**DEPENDENCIES INSTALL:** AS WE ARE NEED NODE JS, THIS STAGE WILL DOWNLOAD ALL THE DEPENDENCIES FROM PACKAGE.JSON FILES

**OWASP:** OPEN WEB APPLICATION SECURE PROJECT- IT WILL CHECK THE DEPENDENCIES

**TRIVY SCAN:** IN THIS STAGE ALL THE FILES WILL GET COPED INTO TRIVYFS.TXT FILE

**BUILD:** WE WILL BUILD THE DOCKER IMAGE

**PUSH:** PUSH THE DOCKER IMAGE INTO DOCKER HUB

**TRIVY** IT WILL SCAN THE IMAGE

**CONTAINER:** CONTAINER WILL GET CREATED AND APPLICATION WILL GET DEPLOYE

**Step 1:** Launch EC2 INSTANCE with t2.Large

**STEP 2:** INSTALL JENKINS, GIT, DOCKER & TRIVY

**Trivy:**

```
wget https://github.com/aquasecurity/trivy/releases/download/v0.18.3/trivy_0.18.3_Linux-64bit.tar.gz
```

```
tar zxvf trivy_0.18.3_Linux-64bit.tar.gz
```

```
mv trivy /usr/local/bin/
```

```
Vim .bashrc
```

```
export PATH=$PATH:/usr/local/bin/
```

```
source .bashrc
```

trivy -version

**JENKINS:** <https://pkg.jenkins.io/redhat-stable/>

amazon-linux-extras install java-openjdk11 -y

sudo wget -O /etc/yum.repos.d/jenkins.repo https://pkg.jenkins.io/redhat-stable/jenkins.repo

sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key

yum install jenkins -y

systemctl start jenkins

systemctl status jenkins

Connect jenkins with instance PUBLIC-IP:8080

Click on Installed suggested plugins (user name-krishna, pass- K123456789)

### **Git & Docker:**

yum install git docker -y

systemctl start docker

chmod 777 ///var/run/docker.sock

### **STEP 3: INSTALL THE FOLLOWING JENKINS PLUGINS**

SonarQube Scanner

NODEJS

OWASP DEPENDENCY CHECK

DOCKER PIPELINE

Eclipse Temurin installer

Dashboard>Manage Jenkins>Plugins>Available plugins>

The screenshot shows the Jenkins web interface. At the top is a navigation bar with the Jenkins logo, user profile 'krishna', and a 'log out' button. Below the navigation bar is a breadcrumb trail: Dashboard > Manage Jenkins > Plugins.

## Plugins

A search bar contains the text 'Search available plugins'. To its right are buttons for 'Install' and a refresh icon.

- Updates:** A sidebar menu on the left lists 'Available plugins' (selected), 'Installed plugins', 'Advanced settings', and 'Download progress'.
- Plugin List:** A table displays installed plugins with columns for 'Install', 'Name', and 'Released'.
 

Install	Name	Released
<input checked="" type="checkbox"/>	<b>SonarQube Scanner</b> 2.16.1 <a href="#">External Site/Tool Integrations</a> <a href="#">Build Reports</a> This plugin allows an easy integration of <a href="#">SonarQube</a> , the open source platform for Continuous Inspection of code quality.	1 mo 26 days ago
<input checked="" type="checkbox"/>	<b>OWASP Dependency-Check</b> 5.4.3 <a href="#">Security</a> <a href="#">DevOps</a> <a href="#">Build Tools</a> <a href="#">Build Reports</a> This plug-in can independently execute a <a href="#">Dependency-Check</a> analysis and visualize results. Dependency-Check is a utility that identifies project dependencies and checks if there are any known, publicly disclosed, vulnerabilities.	2 mo 27 days ago
<input checked="" type="checkbox"/>	<b>NodeJS</b> 1.6.1 <a href="#">npm</a> NodeJS Plugin executes <a href="#">NodeJS</a> script as a build step.	3 mo 21 days ago
<input checked="" type="checkbox"/>	<b>Docker Pipeline</b> 572.v950f58993843 <a href="#">pipeline</a> <a href="#">DevOps</a> <a href="#">Deployment</a> <a href="#">docker</a> Build and use Docker containers from pipelines.	3 mo 26 days ago
<input checked="" type="checkbox"/>	<b>Eclipse Temurin installer</b> 1.5 Provides an installer for the JDK tool that downloads the JDK from <a href="https://adoptium.net">https://adoptium.net</a>	1 yr 1 mo ago

## SETUP SONAR USING DOCKER(container):

Image-name:lts-community

```
docker run -d --name sonar -p 9000:9000 sonarqube:lts-community
```

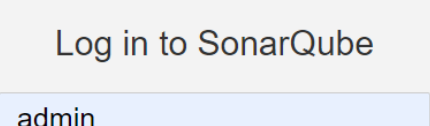
After creating the sonar container, access the sonarqube with Public-Ip:9000 port number.

Login to the sonar dashboard with the following and credentials After entering the credentials

we have to set a new password.

username-admin

password-admin



Log in to SonarQube

admin

.....

Log in Cancel

## Update the password

## Update your password

This account should not use the default password.

Enter a new password

All fields marked with \* are required

Old Password \*

\*\*\*\*\*

New Password \*

\*\*\*\*\*

Confirm Password \*

\*\*\*\*\*

Update

Now configure NodeJs, Java & DP-Check:

### JDK installations

Add JDK

≡

JDK

×

Name

jdk17

☒ Install automatically ?

≡

Install from adoptium.net

?

Version ?

jdk-17.0.8.1+1

Add Installer ▾

Add JDK

≡

NodeJS

×

Name

node16

☒ Install automatically ?

≡

Install from nodejs.org

×

Version

NodeJS 16.2.0

For the underlying architecture, if available, force the installation of the 32bit package. Otherwise the build will fail

☐ Force 32bit architecture

Global npm packages to install

Specify list of packages to install globally -- see npm install -g. Note that you can fix the packages version by using the syntax `packageName@version`

Global npm packages refresh hours

Duration, in hours, before 2 npm cache update. Note that 0 will always update npm cache

72

SonarQube Scanner installations

Add SonarQube Scanner

☰ SonarQube Scanner

✕

Name

mysonar

❗ Required

☒ Install automatically ?

☰ Install from Maven Central

✕

Version

SonarQube Scanner 5.0.1.3006

Add Installer ▾

Dependency-Check installations

Add Dependency-Check

☰ Dependency-Check

✕

Name

DP-Checker

☒ Install automatically ?

☰ Install from github.com

✕

Version

dependency-check 6.5.1

Add Installer ▾

CONFIGURE ALL THE PLUGINS INTO JENKINS:

Goto your Sonarqube Server. Click on Administration ----> Security ----> Users → Click on Tokens and Update Token ----> Give it a name ----> and click on Generate Token.

Create User

All fields marked with \* are required

Login \*

krishna

Minimum 3 characters

Name \*

krishna

Email

k9@gmail.com

Password \*

.....

SCM Accounts

Add

Login and email are automatically considered as SCM accounts

Create

Cancel

sonarqube

ProjectsIssuesRulesQuality ProfilesQuality GatesAdministration

Search for projects...

A

Administration

ConfigurationSecurityProjectsSystemMarketplace

Users

Create and administer individual users.

Create User

Search by login or name...

	SCM Accounts	Last connection	Groups	Tokens
<div>A</div> Administrator admin		< 1 hour ago	sonar-administrators sonar-users	0
<div>K</div> krishna krishna		Never	sonar-users	0

Update Tokens

Tokens of Administrator

Generate Tokens

Name

Enter Token Name

Expires in

30 days

Generate

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

Copy

squ\_fea3c0f3fd2269bb8cc98afcab54a7ba7be1a4a7

Name	Type	Project	Last use	Created	Expiration	
mytoken	User		Never	December 6, 2023	January 5, 2024	Revoke

Done

copy Token

Goto Jenkins Dashboard ----> Manage Jenkins ----> Credentials ----> Add Secret Text with id

New credentials

Kind

Secret text

Scope 

Global (Jenkins, nodes, items, all child items, etc)

Secret

.....

ID 

sonar-token

Description 

Create

In jenkins

Dashboard > Manage Jenkins > System >

☐ Environment variables

SonarQube installations

List of SonarQube installations

Name

mysonar

Server URL

Default is http://localhost:9000

http://34.203.29.231:9000/

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

sonar-token

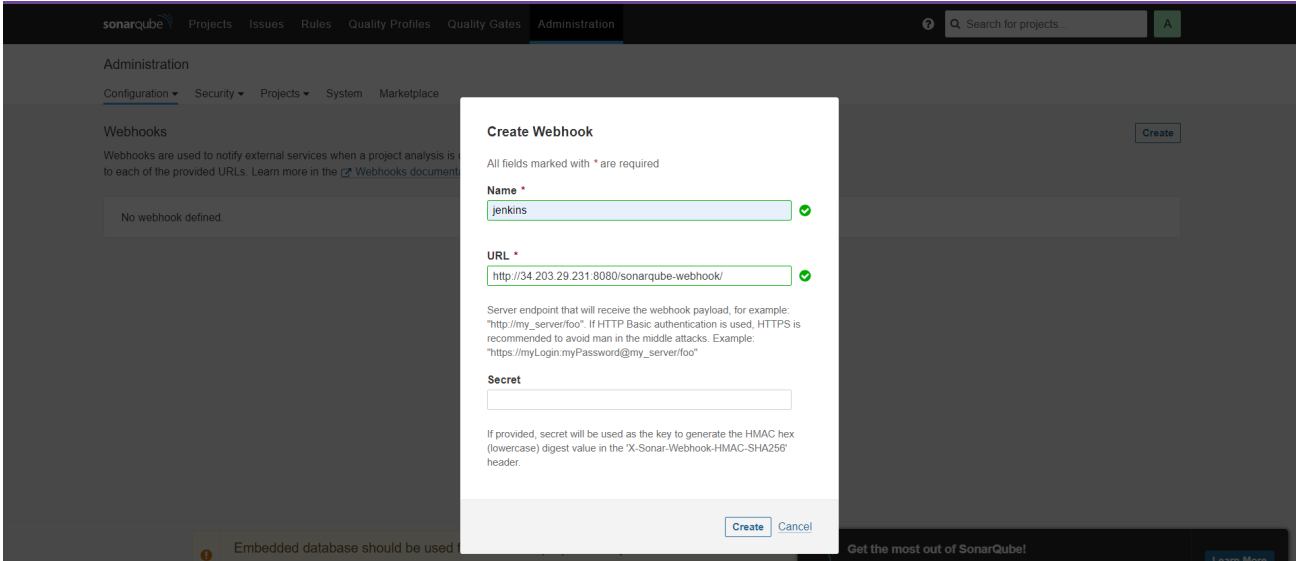
+ Add

Advanced

Save

Apply

Create webhook on Sonarqube



Configure script

pipeline {

agent any

tools {

jdk 'jdk17'

nodejs 'node16'

}

environment {

SCANNER\_HOME=tool 'mysonar'

}

stages {

stage ("Clean WS") {

steps {

cleanWs ()

}

}

stage ("Code") {



```
steps {  
    git branch: 'main', url: 'https://github.com/devops0014/Zomato-Project.git'  
}  
}  
  
stage ("Code Quality Analysis") {  
    steps {  
        withSonarQubeEnv('mysonar') {  
            sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=zomato \  
                -Dsonar.projectKey=zomato '''  
        }  
    }  
}  
  
stage ("Quality Gates") {  
    steps {  
        script {  
            waitForQualityGate abortPipeline: false; credentialsId: 'sonar-token'  
        }  
    }  
}  
  
stage ("Dependencies") {  
    steps {  
        sh 'npm install'  
    }  
}  
}
```

Dashboard > docker-pro >

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

Build History

trend

Filter builds...

#6 Dec 6, 2023, 7:47 AM

#5 Dec 6, 2023, 7:44 AM

#4 Dec 6, 2023, 7:42 AM

#3 Dec 6, 2023, 7:38 AM

#2 Dec 6, 2023, 7:36 AM

#1 Dec 6, 2023, 7:34 AM

Atom feed for all

Atom feed for failures

Stage View

Average stage times:  
(Average full run time: ~55s)

	Declarative: Tool Install	Clean WS	Code	Code Quality Analysis	Quality Gates	Dependencies
#6 Dec 06 13:17 No Changes	29s	335ms	3s	12s	442ms (paused for 5s)	794ms
#5 Dec 06 13:14 No Changes						
#4 Dec 06 13:12 No Changes						
#3 Dec 06 13:08 No Changes						
#2 Dec 06 13:06 No Changes						
#1 Dec 06 13:04 No Changes						

Dashboard > Manage Jenkins > System >

Declarative Pipeline (Docker)

Docker Label ?  
tag&push

Docker registry URL ?  
https://hub.docker.com/repositories/krishna689

Registry credentials  
krishna689/\*\*\*\*\*

+ Add

Install all docker plugins:  
Dashboard ----> Manage Jenkins ----> Available Plugins

```
pipeline {
  agent any

  tools {
    jdk 'jdk17'
    nodejs 'node16'
  }

  environment {
    SCANNER_HOME=tool 'mysonar'
```

```
}

stages {

    stage ("Clean WS") {

        steps {

            cleanWs ()

        }

    }

    stage ("Code") {

        steps {

            git 'https://github.com/devops0014/Zomato-Project.git'

        }

    }

    stage ("Code Quality Analysis") {

        steps {

            withSonarQubeEnv('mysonar') {

                sh "' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=zomato \
                -Dsonar.projectKey=zomato '"

            }

        }

    }

    stage ("Quality Gates") {

        steps {

            script {

                waitForQualityGate abortPipeline: false; credentialsId: 'sonar-token'

            }

        }

    }

}
```

```
}

stage ("Dependencies") {

    steps {

        sh 'npm install'

    }

}

stage ("OWASP") {

    steps {

        dependencyCheck additionalArguments: '--scan ./ --disableYarnAudit --
disableNodeAudit',odcInstallation: 'DP-Check'

        dependencyCheckPublisher pattern: '**/dependency-check-report.xml'

    }

}

stage ("trivy") {

    steps {

        sh 'trivy fs . > trivyfs.txt'

    }

}

stage ("Build") {

    steps {

        sh 'docker build -t image1 .'

    }

}

stage ("Tag&Push") {

    steps {

        script {

            withDockerRegistry(credentialsId: 'DockerPass') {
```

```
sh 'docker tag image1 krishna689/mydockerproject:myzomatoimage'
```

```
sh 'docker push krishna689/mydockerproject:myzomatoimage'
```

```
}
```

```
}
```

```
}
```

```
}
```

```
stage ("Scan the image") {
```

```
  steps {
```

```
    sh 'trivy image krishna689/mydockerproject:myzomatoimage'
```