**DOCKER PROJECT**

**Step 1: Launch an EC2 Instance**

* Instance Type: t2. large
* Storage: Attach an EBS volume of size 30 GB.

**Step 2: Install Required Tools**

**Install 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

sudo mv trivy /usr/local/bin/

vim .bashrc

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

source .bashrc

**Install Jenkins**  
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

**Install Git & Docker**

yum install git docker -y  
systemctl start docker  
chmod 777 /var/run/docker.sock

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step 3: Install Jenkins Plugins Required Plugins:**1. Sonar Scanner  
2. NodeJS  
3. OWASP Dependency Check  
4. Docker Pipeline  
5. Eclipse Temurin Installer Version  
6. Pipeline Stage View

**Step 4: Configure Jenkins**

**SonarQube Setup Using Docker**

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

Access SonarQube: http://<public-ip>:9000

Default credentials:

* Username: admin
* Password: admin (You will be prompted to set a new password)

**Configure SonarQube in Jenkins**

1. **Generate SonarQube Token:**

Navigate to: Administration → Security → Users → Tokens → Create Token

Copy the generated token.

1. **Add Token in Jenkins:**

Go to: Dashboard → Manage Jenkins → Credentials → Add Secret Text

ID: sonar-token

1. **Add SonarQube Server in Jenkins:**

Go to: Dashboard → Manage Jenkins → System → Add Sonar Server

Name: mysonar

URL: http://<sonarqube-ip>:9000

Token: <copied token>

1. **Add SonarQube Scanner:**

Name: mysonar

Install automatically.

Additional Plugin Configurations

Configure NodeJS, Java, and OWASP Dependency Check in Jenkins.

1. **Add webhook in SonarQube:**

Path: Administration → Configuration → Webhooks → Create Webhook

Name: Jenkins

URL: http://<jenkins-ip>:8080/sonarqube-webhook

**Step 5: Now configure NodeJs, Java & DP-Check**

**Install java**

Name – jdk17

Version – jdk-17.0.8.1+1

**Install Nodejs**

Name – node16

Version – nodejs16 16.2.0

**Install DP-Check**

Name – Dp-Check

Version – Dependency-check 6.5.1

**Step 6: Write a Declarative Pipeline**

pipeline {

agent any

tools {

jdk 'jdk17'

nodejs 'node16'

}

environment {

SCANNER\_HOME = tool 'mysonar'

}

stages {

stage("Clean Workspace") {

steps {

cleanWs()

}

}

stage("Clone Code") {

steps {

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

}

}

stage("SonarQube Analysis") {

steps {

withSonarQubeEnv('mysonar') {

sh """

$SCANNER\_HOME/bin/sonar-scanner \

-Dsonar.projectName=zomato \

-Dsonar.projectKey=zomato

"""

}

}

}

stage("Quality Gate") {

steps {

script {

waitForQualityGate abortPipeline: false, credentialsId: 'zomato'

}

}

}

stage("Install Dependencies") {

steps {

sh 'npm install'

}

}

stage("OWASP Dependency Check") {

steps {

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

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

}

}

stage("Trivy Scan") {

steps {

sh 'trivy fs . > trivyfs.txt'

}

}

stage("Build Docker Image") {

steps {

sh 'docker build -t myzomatoimage .'

}

}

stage("Tag & Push Docker Image") {

steps {

script {

withDockerRegistry(credentialsId: 'docker') {

sh 'docker tag myzomatoimage bhagyeshpatil99/mydockerproject:myzomatoimage'

sh 'docker push bhagyeshpatil99/mydockerproject:myzomatoimage'

}

}

}

}

stage("Image Scan with Trivy") {

steps {

sh 'trivy image bhagyeshpatil99/mydockerproject:myzomatoimage'

}

}

stage("Run Container") {

steps {

sh 'docker run -d --name cont1 -p 3000:3000 bhagyeshpatil99/mydockerproject:myzomatoimage'

}

}

}

}

}