pipeline {

agent any

environment {

IMAGE\_NAME = "anji432/veerarepo:latest"

MANIFEST\_PATH = "manifest\_file/k8s"

}

stages {

stage('Checkout') {

steps {

git branch: 'main', url: '<https://github.com/anji432/projectcasestudies.git>'

}

}

stage('Build and Test') {

steps {

sh 'ls -ltr'

sh 'mvn clean package'

}

}

stage('Build and Push Docker Image') {

steps {

script {

sh 'docker build -t $IMAGE\_NAME .'

}

}

}

stage('Scan Image with Trivy') {

steps {

script {

// Install Trivy (skip if pre-installed on agent)

sh '''

if ! command -v trivy &> /dev/null; then

echo "Installing Trivy..."

curl -sfL <https://raw.githubusercontent.com/aquasecurity/trivy/main/contrib/install.sh> | sh -s -- -b /usr/local/bin

fi

'''

// Run Trivy scan and fail if HIGH/CRITICAL vulns exist

sh 'trivy image --exit-code 1 --severity HIGH,CRITICAL $IMAGE\_NAME'

}

}

}

stage('Push to DockerHub') {

steps {

withCredentials([usernamePassword(credentialsId: 'DockerHub-Credentials', usernameVariable: 'DOCKER\_USER', passwordVariable: 'DOCKER\_PASS')]) {

sh '''

echo "$DOCKER\_PASS" | docker login -u "anji432" --password-stdin

docker push $IMAGE\_NAME

'''

}

}

}

stage('Static Code Analysis') {

environment {

SONAR\_URL = "[http://13.203.105.236:9000](http://13.203.105.236:9000/)"

}

steps {

withCredentials([string(credentialsId: 'Sonarqube-token-latest', variable: 'SONAR\_AUTH\_TOKEN')]) {

sh '''

mvn sonar:sonar \

-Dsonar.login=$SONAR\_AUTH\_TOKEN \

-Dsonar.host.url=$SONAR\_URL

'''

}

}

}

stage('Deploy to Dev') {

steps {

withKubeCredentials(kubectlCredentials: [[caCertificate: '', clusterName: 'veera-cluster.ap-south-1.eksctl.io', contextName: '', credentialsId: 'k8\_secret\_token', namespace: '', serverUrl: '[https://7E5A221BDABEC23E3E1C11D40BFDF608.gr7.ap-south-1.eks.amazonaws.com](https://7e5a221bdabec23e3e1c11d40bfdf608.gr7.ap-south-1.eks.amazonaws.com/)']]) {

sh 'curl -LO "<https://storage.googleapis.com/kubernetes-release/release/v1.20.5/bin/linux/amd64/kubectl>"'

sh 'chmod u+x ./kubectl'

sh './kubectl apply -f ${MANIFEST\_PATH}/dev/deployment.yaml --namespace=dev'

sh './kubectl rollout status deployment/spring-boot-app --namespace=dev'

}

}

}

stage('Deploy to Test') {

steps {

withKubeCredentials(kubectlCredentials: [[caCertificate: '', clusterName: 'veera-cluster.ap-south-1.eksctl.io', contextName: '', credentialsId: 'k8\_secret\_token', namespace: '', serverUrl: '[https://7E5A221BDABEC23E3E1C11D40BFDF608.gr7.ap-south-1.eks.amazonaws.com](https://7e5a221bdabec23e3e1c11d40bfdf608.gr7.ap-south-1.eks.amazonaws.com/)']]) {

sh 'curl -LO "<https://storage.googleapis.com/kubernetes-release/release/v1.20.5/bin/linux/amd64/kubectl>"'

sh 'chmod u+x ./kubectl'

sh './kubectl apply -f ${MANIFEST\_PATH}/test/deployment.yaml --namespace=test'

sh './kubectl apply -f ${MANIFEST\_PATH}/test/service.yaml --namespace=test'

sh './kubectl rollout status deployment/spring-boot-app --namespace=test'

}

}

}

stage('Approval to Deploy to Prod') {

steps {

script {

input message: "Approve deployment to Prod?", parameters: [

booleanParam(name: 'Proceed', defaultValue: false, description: 'Approve the deployment to Prod')

]

}

}

}

stage('Deploy to Prod') {

steps {

withKubeCredentials(kubectlCredentials: [[caCertificate: '', clusterName: 'veera-cluster.ap-south-1.eksctl.io', contextName: '', credentialsId: 'k8\_secret\_token', namespace: '', serverUrl: '[https://7E5A221BDABEC23E3E1C11D40BFDF608.gr7.ap-south-1.eks.amazonaws.com](https://7e5a221bdabec23e3e1c11d40bfdf608.gr7.ap-south-1.eks.amazonaws.com/)']]) {

sh 'curl -LO "<https://storage.googleapis.com/kubernetes-release/release/v1.20.5/bin/linux/amd64/kubectl>"'

sh 'chmod u+x ./kubectl'

sh './kubectl apply -f ${MANIFEST\_PATH}/prod/deployment.yaml --namespace=prod'

sh './kubectl rollout status deployment/spring-boot-app --namespace=prod'

}

}

}

}

post {

success {

echo 'Deployment successful!'

mail to: 'anji10432@gmail.com',

subject: "Jenkins Pipeline Success: ${env.JOB\_NAME} #${env.BUILD\_NUMBER}",

body: "Good news! Jenkins job '${env.JOB\_NAME}' (build #${env.BUILD\_NUMBER}) completed successfully.\n\nCheck details: ${env.BUILD\_URL}"

}

failure {

echo 'Deployment failed!'

mail to: 'anji10432@gmail.com',

subject: "Jenkins Pipeline Failure: ${env.JOB\_NAME} #${env.BUILD\_NUMBER}",

body: "Oops! Jenkins job '${env.JOB\_NAME}' (build #${env.BUILD\_NUMBER}) failed.\n\nCheck details: ${env.BUILD\_URL}"

}

}

}