Jenkins Pipeline with DOcker/ECR + EKS

* Jenkins Pipeline With Docker + EKS

pipeline {

agent any

environment {

REGISTRY = "your-dockerhub-username"

IMAGE\_NAME = "your-app"

KUBE\_CONTEXT = "arn:aws:eks:ap-south-1:123456789012:cluster/your-eks-cluster"

}

stages {

stage('Checkout') {

steps {

git branch: 'main', url: 'https://github.com/your-org/your-repo.git'

}

}

stage('Build with Maven') {

steps {

sh 'mvn clean package -DskipTests'

}

}

stage('Docker Build & Push') {

steps {

script {

docker.withRegistry("https://index.docker.io/v1/", 'dockerhub-credentials-id') {

def app = docker.build("${REGISTRY}/${IMAGE\_NAME}:${BUILD\_NUMBER}")

app.push()

app.push("latest") // optional, always points to the latest build

}

}

}

}

stage('Deploy to EKS') {

steps {

withKubeConfig([credentialsId: 'eks-kubeconfig', contextName: "${KUBE\_CONTEXT}"]) {

sh """

kubectl set image deployment/${IMAGE\_NAME}-deployment \

${IMAGE\_NAME}-container=${REGISTRY}/${IMAGE\_NAME}:${BUILD\_NUMBER} --record

kubectl rollout status deployment/${IMAGE\_NAME}-deployment

"""

}

}

}

}

post {

success {

mail to: 'team@example.com',

subject: "SUCCESS: Job ${env.JOB\_NAME} [${env.BUILD\_NUMBER}]",

body: "✅ Deployment successful with image tag ${BUILD\_NUMBER}."

}

failure {

mail to: 'team@example.com',

subject: "FAILED: Job ${env.JOB\_NAME} [${env.BUILD\_NUMBER}]",

body: "❌ Deployment failed. Please check Jenkins logs."

}

}

}

* + Instructions
    - Dockerfile for packaging a Java application.

FROM openjdk:17-jdk-slim

WORKDIR /app

COPY target/\*.jar app.jar

ENTRYPOINT ["java", "-jar", "/app/app.jar"]

* + - Kubernetes Deployment (deployment.yaml)

apiVersion: apps/v1

kind: Deployment

metadata:

name: your-app-deployment

labels:

app: your-app

spec:

replicas: 2

selector:

matchLabels:

app: your-app

template:

metadata:

labels:

app: your-app

spec:

containers:

- name: your-app-container

image: your-dockerhub-username/your-app:latest

ports:

- containerPort: 8080

* + - Generate kubeconfig using below command

Aws eks update-kubeconfig --region ap-south-1 --name your-eks-cluster

* + - * Copy the ~/.kube/config file.
      * Add it to Jenkins as a Secret File Credential with ID: eks-kubeconfig.
    - Install plugins in Jenkins
      * Kubernetes CLI Plugin (for withKubeConfig).
      * Docker Pipeline Plugin.

---------------------------------------------------------------------------------------------------------------------------------------

**Jenkins pipeline with ECR+ EKS**

pipeline {

agent any

environment {

AWS\_ACCOUNT\_ID = "123456789012" // your AWS account ID

AWS\_REGION = "ap-south-1" // region of your EKS/ECR

ECR\_REPO = "your-app" // ECR repository name

IMAGE\_NAME = "${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com/${ECR\_REPO}"

KUBE\_CONTEXT = "arn:aws:eks:ap-south-1:${AWS\_ACCOUNT\_ID}:cluster/your-eks-cluster"

}

stages {

stage('Checkout') {

steps {

git branch: 'main', url: 'https://github.com/your-org/your-repo.git'

}

}

stage('Build with Maven') {

steps {

sh 'mvn clean package -DskipTests'

}

}

stage('Docker Build & Push to ECR') {

steps {

script {

// Login to ECR

sh """

aws ecr get-login-password --region ${AWS\_REGION} \

| docker login --username AWS --password-stdin ${AWS\_ACCOUNT\_ID}.dkr.ecr.${AWS\_REGION}.amazonaws.com

"""

// Build image

def app = docker.build("${IMAGE\_NAME}:${BUILD\_NUMBER}")

// Push versioned tag

app.push()

// Push latest tag

sh "docker tag ${IMAGE\_NAME}:${BUILD\_NUMBER} ${IMAGE\_NAME}:latest"

sh "docker push ${IMAGE\_NAME}:latest"

}

}

}

stage('Deploy to EKS') {

steps {

withKubeConfig([credentialsId: 'eks-kubeconfig', contextName: "${KUBE\_CONTEXT}"]) {

sh """

kubectl set image deployment/${ECR\_REPO}-deployment \

${ECR\_REPO}-container=${IMAGE\_NAME}:${BUILD\_NUMBER} --record

kubectl rollout status deployment/${ECR\_REPO}-deployment

"""

}

}

}

}

post {

success {

echo "✅ Deployment successful to EKS with image ${IMAGE\_NAME}:${BUILD\_NUMBER}"

}

failure {

echo "❌ Deployment failed. Please check Jenkins logs."

}

}

}