DEVOPS DAY 5 – Deploy Java app to minikube automated with jenkins

**Deploy Java App to Minikube Automated with Jenkins**

**1. Overview**

Automating the deployment of a Java application to Minikube using Jenkins involves building the application, creating a Docker image, pushing it to a container registry, and deploying it to Minikube using Kubernetes manifests.

**2. Key Concepts**

**A. Jenkins Pipeline**

Jenkins automates the CI/CD process using a declarative pipeline. The pipeline consists of multiple stages such as:

* **SCM Checkout:** Fetches code from a repository (GitHub/GitLab).
* **Build & Test:** Uses Maven (mvn package) to compile and test the Java application.
* **Docker Build & Push:** Builds a Docker image of the application and pushes it to Docker Hub.
* **Deploy to Minikube:** Uses kubectl to apply Kubernetes deployment and service files.

**B. Minikube**

Minikube is a lightweight Kubernetes cluster for local development and testing. It allows developers to run Kubernetes locally and deploy applications without needing a cloud-based cluster.

**Commands to Start Minikube:**

sh

CopyEdit

minikube start

kubectl cluster-info

kubectl get nodes

**C. Docker**

Docker is used to package the Java application into a container image, making it portable and easy to deploy across environments.

**Dockerfile Example:**

dockerfile

CopyEdit

FROM openjdk:11

COPY target/webapp.jar /app/webapp.jar

WORKDIR /app

CMD ["java", "-jar", "webapp.jar"]

**D. Kubernetes Deployment**

Kubernetes YAML files define how the application should be deployed inside the Minikube cluster.

**Deployment YAML Example:**

yaml

CopyEdit

apiVersion: apps/v1

kind: Deployment

metadata:

name: webapp

spec:

replicas: 1

selector:

matchLabels:

app: webapp

template:

metadata:

labels:

app: webapp

spec:

containers:

- name: webapp

image: saranavinashb/webapp1

ports:

- containerPort: 8080

**Apply Deployment:**

sh

CopyEdit

kubectl apply -f deployment.yml

kubectl get pods

pipeline {

agent any

stages {

stage('scm') {

steps {

git branch: ''

}

}

stage('builb-clean') {

steps {

sh "mvn clean"

}

}

stage('build-validate') {

steps {

sh "mvn validate"

}

}

stage('build-com') {

steps {

sh "mvn compile"

}

}

stage('build-test') {

steps {

sh "mvn test"

}

}

stage('build-install') {

steps {

sh "mvn package"

}

}

stage('build to images') {

steps {

script{

sh 'docker build -t .'

}

}

}

stage('push to hub') {

steps {

script{

withDockerRegistry(credentialsId: 'Docker\_cred', url: 'https://index.docker.io/v1/') {

sh 'docker push '

}

}

}

}

stage('Deploy App') {

steps {

withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'mukubeconfig\_011', namespace: '', restrictKubeConfigAccess: false, serverUrl: 'https://192.168.49.2:8443') {

sh 'kubectl apply -f deployment.yml --validate=false'

}

}

}

stage('Test') {

steps {

withKubeConfig(caCertificate: '', clusterName: 'minikube', contextName: 'minikube', credentialsId: 'mukubeconfig\_011', namespace: '', restrictKubeConfigAccess: false, serverUrl: 'https://192.168.49.2:8443') {

sh 'minikube service my-service --url | xargs curl'

}

}

}

}

}





