**HELM Project By DevOps Shack**

**# Install Helm on Ubuntu**

sudo apt update && sudo apt upgrade -y

curl https://raw.githubusercontent.com/helm/helm/main/scripts/get-helm-3 | bash

# Manifest YAML

---

# Secrets for MySQL

apiVersion: v1

kind: Secret

metadata:

name: mysql-secret

type: Opaque

data:

MYSQL\_ROOT\_PASSWORD: VGVzdEAxMjM=

MYSQL\_DATABASE: YmFua2FwcGRi

---

# ConfigMap for MySQL

apiVersion: v1

kind: ConfigMap

metadata:

name: mysql-config

data:

MYSQL\_DATABASE: "bankappdb"

---

# Storage Class for EBS

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: ebs-sc

provisioner: ebs.csi.aws.com

parameters:

type: gp3

fsType: ext4

reclaimPolicy: Retain

volumeBindingMode: WaitForFirstConsumer

---

# Persistent Volume Claim

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: mysql-pvc

spec:

accessModes:

- ReadWriteOnce

storageClassName: ebs-sc

resources:

requests:

storage: 5Gi

---

# MySQL Deployment

apiVersion: apps/v1

kind: Deployment

metadata:

name: mysql

spec:

selector:

matchLabels:

app: mysql

strategy:

type: Recreate

template:

metadata:

labels:

app: mysql

spec:

containers:

- image: mysql:8

name: mysql

resources:

requests:

memory: "512Mi"

cpu: "500m"

limits:

memory: "1Gi"

cpu: "1"

env:

- name: MYSQL\_ROOT\_PASSWORD

valueFrom:

secretKeyRef:

name: mysql-secret

key: MYSQL\_ROOT\_PASSWORD

- name: MYSQL\_DATABASE

valueFrom:

configMapKeyRef:

name: mysql-config

key: MYSQL\_DATABASE

ports:

- containerPort: 3306

name: mysql

volumeMounts:

- mountPath: /var/lib/mysql

name: mysql-data

volumes:

- name: mysql-data

persistentVolumeClaim:

claimName: mysql-pvc

---

# MySQL Service

apiVersion: v1

kind: Service

metadata:

name: mysql-service

spec:

ports:

- port: 3306

targetPort: 3306

selector:

app: mysql

---

# BankApp Deployment

apiVersion: apps/v1

kind: Deployment

metadata:

name: bankapp

spec:

replicas: 1

selector:

matchLabels:

app: bankapp

template:

metadata:

labels:

app: bankapp

spec:

containers:

- name: bankapp

image: adijaiswal/bankapp:latest

resources:

requests:

memory: "256Mi"

cpu: "250m"

limits:

memory: "512Mi"

cpu: "500m"

ports:

- containerPort: 8080

env:

- name: SPRING\_DATASOURCE\_URL

value: jdbc:mysql://mysql-service:3306/bankappdb?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true

- name: SPRING\_DATASOURCE\_USERNAME

value: root

- name: SPRING\_DATASOURCE\_PASSWORD

valueFrom:

secretKeyRef:

name: mysql-secret

key: MYSQL\_ROOT\_PASSWORD

---

# BankApp Service

apiVersion: v1

kind: Service

metadata:

name: bankapp-service

spec:

type: LoadBalancer

ports:

- port: 80

targetPort: 8080

selector:

app: bankapp

**Step 1: Create a Helm Chart Directory**

Run the following command to create a new Helm chart:

helm create bankapp-chart

This creates a directory structure for the Helm chart, including templates, values.yaml, and other necessary files.

**Step 2: Organize Resources in the Templates Directory**

Place each resource in the templates/ directory of the Helm chart, and parameterize values where needed.

**Example Directory Structure:**

bankapp-chart/

├── Chart.yaml

├── values.yaml

├── templates/

│ ├── secrets.yaml

│ ├── configmap.yaml

│ ├── storageclass.yaml

│ ├── pvc.yaml

│ ├── mysql-deployment.yaml

│ ├── mysql-service.yaml

│ ├── bankapp-deployment.yaml

│ ├── bankapp-service.yaml

**Step 3: Convert YAML Files to Helm Templates**

**1. secrets.yaml**

Template the secret values:

apiVersion: v1

kind: Secret

metadata:

name: {{ .Values.mysql.secretName }}

type: Opaque

data:

MYSQL\_ROOT\_PASSWORD: {{ .Values.mysql.rootPassword | b64enc }}

**2. configmap.yaml**

Parameterize the ConfigMap:

apiVersion: v1

kind: ConfigMap

metadata:

name: {{ .Values.mysql.configName }}

data:

MYSQL\_DATABASE: {{ .Values.mysql.database }}

**3. storageclass.yaml**

Template the storage class:

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

name: {{ .Values.storageClass.name }}

provisioner: ebs.csi.aws.com

parameters:

type: {{ .Values.storageClass.type }}

fsType: {{ .Values.storageClass.fsType }}

reclaimPolicy: {{ .Values.storageClass.reclaimPolicy }}

volumeBindingMode: {{ .Values.storageClass.volumeBindingMode }}

**4. pvc.yaml**

Template the PersistentVolumeClaim:

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: {{ .Values.mysql.pvcName }}

spec:

accessModes:

- ReadWriteOnce

storageClassName: {{ .Values.storageClass.name }}

resources:

requests:

storage: {{ .Values.mysql.storageSize }}

**5. mysql-deployment.yaml**

Template the MySQL deployment:

apiVersion: apps/v1

kind: Deployment

metadata:

name: {{ .Values.mysql.name }}

spec:

selector:

matchLabels:

app: {{ .Values.mysql.name }}

strategy:

type: Recreate

template:

metadata:

labels:

app: {{ .Values.mysql.name }}

spec:

containers:

- image: {{ .Values.mysql.image }}

name: {{ .Values.mysql.name }}

resources:

requests:

memory: {{ .Values.mysql.resources.requests.memory }}

cpu: {{ .Values.mysql.resources.requests.cpu }}

limits:

memory: {{ .Values.mysql.resources.limits.memory }}

cpu: {{ .Values.mysql.resources.limits.cpu }}

env:

- name: MYSQL\_ROOT\_PASSWORD

valueFrom:

secretKeyRef:

name: {{ .Values.mysql.secretName }}

key: MYSQL\_ROOT\_PASSWORD

- name: MYSQL\_DATABASE

valueFrom:

configMapKeyRef:

name: {{ .Values.mysql.configName }}

key: MYSQL\_DATABASE

ports:

- containerPort: 3306

name: mysql

volumeMounts:

- mountPath: /var/lib/mysql

name: mysql-data

volumes:

- name: mysql-data

persistentVolumeClaim:

claimName: {{ .Values.mysql.pvcName }}

**6. mysql-service.yaml**

Template the MySQL service:

apiVersion: v1

kind: Service

metadata:

name: {{ .Values.mysql.serviceName }}

spec:

ports:

- port: 3306

targetPort: 3306

selector:

app: {{ .Values.mysql.name }}

**7. bankapp-deployment.yaml**

Template the BankApp deployment:

apiVersion: apps/v1

kind: Deployment

metadata:

name: {{ .Values.bankapp.name }}

spec:

replicas: {{ .Values.bankapp.replicas }}

selector:

matchLabels:

app: {{ .Values.bankapp.name }}

template:

metadata:

labels:

app: {{ .Values.bankapp.name }}

spec:

containers:

- name: {{ .Values.bankapp.name }}

image: {{ .Values.bankapp.image }}

resources:

requests:

memory: {{ .Values.bankapp.resources.requests.memory }}

cpu: {{ .Values.bankapp.resources.requests.cpu }}

limits:

memory: {{ .Values.bankapp.resources.limits.memory }}

cpu: {{ .Values.bankapp.resources.limits.cpu }}

ports:

- containerPort: 8080

env:

- name: SPRING\_DATASOURCE\_URL

value: jdbc:mysql://{{ .Values.mysql.serviceName }}:3306/{{ .Values.mysql.database }}?useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true

- name: SPRING\_DATASOURCE\_USERNAME

value: root

- name: SPRING\_DATASOURCE\_PASSWORD

valueFrom:

secretKeyRef:

name: {{ .Values.mysql.secretName }}

key: MYSQL\_ROOT\_PASSWORD

**8. bankapp-service.yaml**

Template the BankApp service:

apiVersion: v1

kind: Service

metadata:

name: {{ .Values.bankapp.serviceName }}

spec:

type: LoadBalancer

ports:

- port: 80

targetPort: 8080

selector:

app: {{ .Values.bankapp.name }}

**Step 4: Update values.yaml**

Define values to be used in the templates:

mysql:

name: mysql

secretName: mysql-secret

configName: mysql-config

pvcName: mysql-pvc

serviceName: mysql-service

image: mysql:8

database: bankappdb

rootPassword: "Test@123"

storageSize: 5Gi

resources:

requests:

memory: "512Mi"

cpu: "500m"

limits:

memory: "1Gi"

cpu: "1"

bankapp:

name: bankapp

serviceName: bankapp-service

image: adijaiswal/bankapp:latest

replicas: 1

resources:

requests:

memory: "256Mi"

cpu: "250m"

limits:

memory: "512Mi"

cpu: "500m"

storageClass:

name: ebs-sc

type: gp3

fsType: ext4

reclaimPolicy: Retain

volumeBindingMode: WaitForFirstConsumer

**Step 5: Deploy the Helm Chart**

Package and deploy the Helm chart:

1. Package the chart:

helm package bankapp-chart

1. Deploy the chart:

helm install bankapp ./bankapp-chart

1. Verify the deployment:

kubectl get all

Bottom of Form