SpringBoot and MySQL on Kubernetes

# Deploy MySQL on Kubernetes:

1. Reference: <https://github.com/IBM/spring-boot-microservices-on-kubernetes#11-use-mysql-in-container>
2. Create file account-database.yaml
3. Copy the below contents to the file:

---

apiVersion: v1

kind: Service

metadata:

name: account-database

labels:

app: office-space

spec:

ports:

- port: 3306

protocol: TCP

targetPort: 3306

selector:

app: office-space

tier: database

---

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: account-database

labels:

app: office-space

spec:

strategy:

type: Recreate

template:

metadata:

labels:

app: office-space

tier: database

spec:

containers:

- image: mysql:5.6

name: account-database

env:

- name: MYSQL\_USER

value: michaelbolton

- name: MYSQL\_PASSWORD

value: password

- name: MYSQL\_ROOT\_PASSWORD

value: password

- name: MYSQL\_DATABASE

value: dockercon2017

ports:

- containerPort: 3306

name: db

1. Run below command to deploy MySQL

kubectl create -f account-database.yaml

1. Run below command to see whether MySQL service and deployment are successful

kubectl get svc,deploy

1. Run below command to check the pods

Kubectl get pods

## Create secrets.yaml

1. Copy below contents to the file:

---

apiVersion: v1

kind: Secret

metadata:

name: demo-credentials

type: Opaque

data:

username: bWljaGFlbGJvbHRvbg==

password: cGFzc3dvcmQ=

host: YWNjb3VudC1kYXRhYmFzZQ==

port: MzMwNg==

1. To apply the configuration, run below command

kubectl apply -f secrets.yaml

1. Use below commands to encode and decode values:

echo mickel | base64

bWlja2VsCg==

echo bWlja2VsCg== | base64 --decode

mickel

## Create springboot application with Hibernate and MySQL dependencies

1. Put below Dockerfile in the project directory

FROM maven:3.3.9-jdk-8-alpine

COPY . /app

WORKDIR /app

RUN apk update && apk add mysql mysql-client

CMD java -jar target/\*.jar

1. Build the jar file and copy the folder to linux machine
2. Run below command to build the image

sudo docker build -t kokateaniket/compute-interest-api .

1. sudo docker images
2. Tag the image with kokateaniket/compute-interest-api

sudo docker tag ac0ef72835a1 kokateaniket/compute-interest-api

1. Push the image to docker hub

sudo docker push kokateaniket/compute-interest-api

1. Create a file comput-interest-api.yaml
2. Copy the below contents to the file

---

apiVersion: v1

kind: Service

metadata:

name: compute-interest-api

labels:

app: office-space

spec:

ports:

- port: 8080

protocol: TCP

targetPort: 8080

nodePort: 30666

selector:

app: office-space

tier: compute

type: LoadBalancer

---

apiVersion: extensions/v1beta1

kind: Deployment

metadata:

name: compute-interest-api

labels:

app: office-space

spec:

strategy:

type: Recreate

template:

metadata:

labels:

app: office-space

tier: compute

spec:

containers:

- image: kokateaniket/compute-interest-api:latest

imagePullPolicy: Always

name: compute-interest-api

env:

- name: MYSQL\_DB\_USER

valueFrom:

secretKeyRef:

name: demo-credentials

key: username

- name: MYSQL\_DB\_PASSWORD

valueFrom:

secretKeyRef:

name: demo-credentials

key: password

- name: MYSQL\_DB\_HOST

valueFrom:

secretKeyRef:

name: demo-credentials

key: host

- name: MYSQL\_DB\_PORT

valueFrom:

secretKeyRef:

name: demo-credentials

key: port

ports:

- containerPort: 8080

1. Deploy the springboot app in kubernetes

kubectl create -f comput-interest-api.yaml

1. Check the deployment with

Kubectl get svc,deploy

## Verify the deployment

1. Go to kubernetes dashboard. Login with admin credentials

https://34.233.195.223:6443/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/#!/overview?namespace=default

1. Go to services menu
2. Click on External endpoint of the compute-interest-api. It may take couple of minutes to expose the service as load balancer
3. You should see the URL working