**DEPLOY POSTGRES CONTAINER TO KUBERNETES**

Prerequisites:

Docker Desktop installed and running or minikube

Access to kubectl

1. Create a yaml file for the postgres secret password
2. Create a configuration map to store POSTGRES\_USER & POSTGRES\_DB
3. Create a Stateful set yaml file to deploy the postgres database
4. Create a Service yaml file that services the database.
5. Deploy all 4 yaml files with kubectl apply –f <filename>. Start with the secret first, then the config map, then the db StatefulSet deployment file, then the Service file.
6. Issue the kubectl get services command to view which port (Node Port) kubernetes exposes for postresdb
7. User the port to connect to the database instance
8. Open a psql cli window and enter localhost, port number, database name, postgres username, and password.

PS C:\data\minikubeTests\MessageBoardApp\.k8s\_3**> kubectl apply -f .\env-configmap.yaml**

configmap/env created

PS C:\data\minikubeTests\MessageBoardApp\.k8s\_3> **kubectl apply -f .\db-deployment.yaml**

**configmap/postgres-configuration created**

**secret/postgres-credentials** created

**persistentvolume/postgres-pv** created

**persistentvolumeclaim/postgres-pvc** created

statefulset.apps/**postgres-statefulset** created

PS C:\data\minikubeTests\MessageBoardApp\.k8s\_3> **kubectl apply -f .\db-service.yaml**

service/postgres-service created

PS C:\data\minikubeTests\MessageBoardApp\.k8s\_3> **kubectl get services**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 13h

postgres-service NodePort 10.106.196.41 <none> 5432:**30740**/TCP 3m2s

**ACCESS THE POSTGRES DATABASE FROM OUTSIDE THE KUBERNETES CLUSTER:**

Bring up a PSQL CLI window prompt: (pwd is super-secret). You can view the pwd unencrypted in the kubernetes dashboard. Look under Secret and click on the “eye” icon to view the clear text.

Server [**localhost**]:

Database [postgres]: **django-db**

Port [5432]: 30740

Username [postgres]: **postgres-user**

Password for user postgres-user:

psql (12.1, server 12.0)

WARNING: Console code page (437) differs from Windows code page (1252)

8-bit characters might not work correctly. See psql reference

page "Notes for Windows users" for details.

Type "help" for help.

django-db=# **\l**

List of databases

Name | Owner | Encoding | Collate | Ctype | Access privileges

-----------+---------------+----------+------------+------------+-------------------------------------

**django-db | postgres-user | UTF8 | en\_US.utf8 | en\_US.utf8 |**

postgres | postgres-user | UTF8 | en\_US.utf8 | en\_US.utf8 |

template0 | postgres-user | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/"postgres-user" +

| | | | | "postgres-user"=CTc/"postgres-user"

template1 | postgres-user | UTF8 | en\_US.utf8 | en\_US.utf8 | =c/"postgres-user" +

| | | | | "postgres-user"=CTc/"postgres-user"

(4 rows)

django-db=# \df

List of functions

Schema | Name | Result data type | Argument data types | Type

--------+------+------------------+---------------------+------

(0 rows)

django-db=# select count(\*) from django-db

django-db-# **\q**

Press any key to continue . . .

**db-deployment.yaml**

---

apiVersion: v1

kind: **ConfigMap**

metadata:

name: postgres-configuration

labels:

app: db

data:

POSTGRES\_DB: "django-db"

POSTGRES\_USER: "postgres-user"

---

apiVersion: v1

kind: **Secret**

metadata:

name: postgres-credentials

type: Opaque

data:

# TODO This should not be in version control in real deployments

password: c3VwZXItc2VjcmV0

---

apiVersion: v1

kind: **PersistentVolume**

metadata:

name: postgres-pv

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 100Mi

accessModes:

- ReadWriteOnce

hostPath:

path: /mnt1/postgres-data

---

apiVersion: v1

kind: **PersistentVolumeClaim**

metadata:

name: postgres-pvc

labels:

type: local

spec:

storageClassName: manual

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 100Mi

volumeName: postgres-pv

---

apiVersion: apps/v1

kind: **StatefulSet**

metadata:

name: postgres-statefulset

labels:

app: db

spec:

serviceName: "db"

replicas: 1

selector:

matchLabels:

app: db

template:

metadata:

labels:

app: db

spec:

containers:

- name: db

image: postgres:12.0-alpine

envFrom:

- configMapRef:

name: postgres-configuration

env:

- name: POSTGRES\_PASSWORD

valueFrom:

secretKeyRef:

name: postgres-credentials

key: password

ports:

- containerPort: 5432

name: postgresdb

volumeMounts:

- name: postgres-volume-mount

mountPath: /var/lib/postgresql/data

readinessProbe:

exec:

command:

- bash

- "-c"

- "psql -U$POSTGRES\_USER -d$POSTGRES\_DB -c 'SELECT 1'"

initialDelaySeconds: 15

timeoutSeconds: 2

livenessProbe:

exec:

command:

- bash

- "-c"

- "psql -U$POSTGRES\_USER -d$POSTGRES\_DB -c 'SELECT 1'"

initialDelaySeconds: 15

timeoutSeconds: 2

volumes:

- name: postgres-data

emptyDir: {}

- name: postgres-volume-mount

persistentVolumeClaim:

claimName: postgres-pvc

**db-service.yaml**

apiVersion: v1

kind: Service

metadata:

name: postgres-service

labels:

app: db

spec:

ports:

- port: 5432

name: db

type: NodePort

selector:

app: db

**env-configMap.yaml**

apiVersion: v1

data:

DB\_ENGINE: django.db.backends.postgresql

DB\_HOST: postgres-service

DEBUG: "1"

DJANGO\_ALLOWED\_HOSTS: localhost 127.0.0.1 [::1] messageboard.com

SECRET\_KEY: '\*olgb7z'

kind: **ConfigMap**

metadata:

creationTimestamp: null

labels:

io.kompose.service: db-env

name: env