Darren Freeman

Database Assignment #7

In this database assignment we are creating a launching a database where we will be using Kubernetes to deploy Mongo DB as a service via a *yaml* file.

First, we will use the command terminal to make the ***mongo-deployment.yaml*** file

**nano mongo-deployment.yaml**

**Now copy the following into the file**

*apiVersion: apps/v1*

*kind: Deployment*

*metadata:*

*name: mongodb-deployment*

*spec:*

*replicas: 1*

*selector:*

*matchLabels:*

*app: mongodb*

*template:*

*metadata:*

*labels:*

*app: mongodb*

*spec:*

*containers:*

*- name: mongodb*

*image: mongo:4.4*

*ports:*

*- containerPort: 27017*

*env:*

*- name: MONGO\_INITDB\_ROOT\_USERNAME*

*valueFrom:*

*secretKeyRef:*

*name: mongodb-secret*

*key: mongo-root-username*

*- name: MONGO\_INITDB\_ROOT\_PASSWORD*

*valueFrom:*

*secretKeyRef:*

*name: mongodb-secret*

*key: mongo-root-password*

*---*

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: mongo-service*

*spec:*

*selector:*

*app: mongodb*

*ports:*

*- protocol: TCP*

*port: 27017*

*targetPort: 27017*

Task 2:

**run the following command**

*echo -n mongo-root-username | base64*

**You should get *=*** *bW9uZ28tcm9vdC11c2VybmFtZQ==*

*echo -n mongo-root-password | base64*

**You should get** *= bW9uZ28tcm9vdC1wYXNzd29yZGU=*

We then need to create a secret.yaml file and paste the following credentials into the specified values.

*apiVersion: v1*

*kind: Secret*

*metadata:*

*name: mongodb-secret*

*type: Opaque*

*data:*

*mongo-root-username:* ***bW9uZ28tcm9vdC11c2VybmFtZQ==***

*mongo-root-password:* ***bW9uZ28tcm9vdC1wYXNzd29yZA==***

Task 3:

Now we need to deploy the ***secret****.****yaml*** file. First then *the* ***mongo****-****deployment****.****yaml*** file. The ***mongo-deployment yaml*** file uses information (credentials) from the ***secret.yaml*** file.

*kubectl create -f* ***secret.yaml***

*kubectl create -f* ***mongo-deployment.yaml***

Task 4:

**Create the config\_map.yaml and paste the following:**

*apiVersion: v1*

*kind: ConfigMap*

*metadata:*

*name: mongodb-configmap*

*data:*

*database\_url: mongo-service*

After the config\_map is created we need to create the mongo-express deployment that will act as the user interface application. This direct us to the database. The service is listening for port 8081 externally, and redirect it to port 8081 internally. Our **mongo-express** application will connect on port 8081 traffic.

**Create a *mongo-express.yaml* file and paste the following**

*apiVersion: apps/v1*

*kind: Deployment*

*metadata:*

*name: mongoexp-deployment*

*spec:*

*replicas: 1*

*selector:*

*matchLabels:*

*app: mongo-express*

*template:*

*metadata:*

*labels:*

*app: mongo-express*

*spec:*

*containers:*

*- name: mongo-express*

*image: mongo-express*

*ports:*

*- containerPort:* ***8081***

*env:*

*- name: ME\_CONFIG\_MONGODB\_ADMINUSERNAME*

*valueFrom:*

*secretKeyRef:*

*name: mongodb-secret*

*key: mongo-root-username*

*- name: ME\_CONFIG\_MONGODB\_ADMINPASSWORD*

*valueFrom:*

*secretKeyRef:*

*name: mongodb-secret*

*key: mongo-root-password*

*- name: ME\_CONFIG\_MONGODB\_SERVER*

*valueFrom:*

*configMapKeyRef:*

*name: mongodb-configmap*

*key: database\_url*

*---*

*apiVersion: v1*

*kind: Service*

*metadata:*

*name: mongo-exp-service*

*spec:*

*selector:*

*app: mongo-express*

*type:* ***LoadBalancer***

*ports:*

*- protocol: TCP*

*port:* ***8081***

*targetPort:* ***8081***

Now we will need to create the necessary *yaml* files. First create the *config\_map* and then the *mongo-express* file.

*kubectl create -f* ***config\_map.yaml***

*kubectl create -f* ***mongo-express.yaml***

Task 5:

Now for the moment of truth, we can finally log into the database.

Type into your browser ***localhost:8080***

**You should see the following page:**

Graphical user interface, text, application, email

Description automatically generated

On the top right of the application, you will see a box that says “Database Name”

There, type in ***Test*** and hit the +create database button.

Another artifact will appear with a ***view*** button with the words ***Test*** next to it, hit the ***view*** button.

Like before there is now a text field and corresponding button to create a collection. Call the collection we are going to make ***recipe.***

**Copy the following in the document box**

{

    title: 'Chicken Soft Tacos',

    calories\_per\_serving: 205,

    cook\_time: 19,

    desc: 'Mexican soft tacos',

    directions: [

        'Put seasoning on chicken breasts',

        'Grill until done',

        'Chop chicken into peices',

        'Put in totillas'

    ],

    ingredients: [

        {

            name: 'chicken breast',

            quantity: {

                amount: 1,

                unit: 'lbs'

            }

        },

        {

            name: 'taco seasoning',

            quantity: {

                amount: 2,

                unit: 'oz'

            }

        },

        {

            name: 'small flour totillas',

            quantity: {

                amount: 12,

                unit: 'oz'

            }

        }

    ],

    likes: [

        261,

        1,

        415

    ],

    likes\_count: 3,

    prep\_time: 10,

    rating: [

        4,

        4,

        4,

        4,

        2,

        5,

        3

    ],

    rating\_avg: 3.71,

    servings: 5,

    tags: [

        'mexican',

        'quick',

        'easy',

        'chicken'

    ],

    type: 'Dinner'

}

Now click save and you should get the followingGraphical user interface, application, website

Description automatically generated

Once saved you all done!

***Terminé***