Faculty of Engineering & Applied Science

SOFE4790U – Distributed Systems

Lab 1 – CRN 44425

Due Date: 09/25/2022

First Name	Last Name	Student ID
Abdul	Bhutta	100785884

Discussion

A docker is a **containerization** which is used to develop and deploy applications that can run on any hardware or operating system and allow for faster changes without any interruption within the production of the software. An **image** can be created which can be deployed and run on a system which separates the main dependencies from other applications on the computer and allow the application to be carried over to any other environment. An example of a docker image used in the lab was the prebuilt MySQL and MongoDB version 4.08.

Kubernetes or K8s are used to organize, automate deployment, and scale many containers to run on a **cluster** of nodes or also known as **container-orchestration**. A **pod** can be used to deploy one or multiple containers which all share the same resources. Kubernetes allows the user with **auto-scaling** which is extremely useful for users as the network bandwidth increases, it will increase the containers to balance the heavy workload until the heavy traffic becomes less, which then will decrease the resources that were allocated. In the table below, the advantages and disadvantages are shown of using docker versus a virtual machine.

Advantages	Disadvantages
Pre-built images are available for the applications, where a pre-built VM for an application may be hard to find.	Security is weaker than a virtual machine.
Runtime(boot) in seconds where a VM may take minutes to boot up.	It is a bit challenging to learn docker concepts than a GUI platform such as a virtual machine.
Container can be created faster than a virtual machine.	Data is not persistent and maybe lost.
Docker images allow the user for scalability.	

1. MySQL Deployment

i. Deploying MySQL application on the

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl create deployment mysql-deployment --image mysql/mysql-server --port=3306
deployment.apps/mysql-deployment created
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get deployment
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
mongodb-standalone   1/1     1            1           47h
mysql-deployment     1/1     1            1           21s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-standalone-5c6cb86f45-m69nl  1/1     Running   0          47h
mysql-deployment-7467c475f8-9kf6k    1/1     Running   0          28s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$
```

ii. Accessing MySQL using the temporary password through the pod

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-standalone-5c6cb86f45-m69nl  1/1     Running   0          2d
mysql-deployment-7467c475f8-9kf6k    1/1     Running   0          8m7s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl logs mysql-deployment-7467c475f8-9kf6k 2>&1 |grep GENERATED
[Entrypoint] GENERATED ROOT PASSWORD: z&_e0vw,NMI9633gVI^HC61IA_/f%?q9
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl exec -it mysql-deployment-7467c475f8-9kf6k -- mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 12
Server version: 8.0.30

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

iii. Changing the root password and creating a new user (sofe4790u) while allowing the user full privileges

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'password' ;
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE USER 'user'@'%' IDENTIFIED BY 'sofe4790u';
Query OK, 0 rows affected (0.02 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'user'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)
```

iv. Creating a load balancer service to access the pods through an external IP address

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$  kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-service
service/mysql-service exposed
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$
```

v. Accessing MySQL through an external IP Address

```
bhutta_abdul@cloudshell:~$ kubectl get services
NAME              TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)          AGE
kubernetes        ClusterIP      10.84.0.1     <none>           443/TCP          9d
mongodb-service   LoadBalancer   10.84.8.48    35.226.185.35    3306:30647/TCP   20h
mysql-service     LoadBalancer   10.84.0.54    34.171.222.236   3306:30136/TCP   7d
bhutta_abdul@cloudshell:~$ mysql -uuser -psofe4790u -h34.171.222.236
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1616
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

vi. Deploying MySQL using YAML

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl apply -f mysql.yaml
service/mysql-service created
deployment.apps/mysql-deployment created
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get services
NAME                TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
kubernetes          ClusterIP      10.84.0.1      <none>           443/TCP          2d22h
mongodb-standalone  LoadBalancer   10.84.13.125   35.226.185.35    3306:31458/TCP   2d
mysql-service       LoadBalancer   10.84.0.54     <pending>        3306:30136/TCP   9s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get deployments
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
mongodb-standalone  1/1     1            1           2d
mysql-deployment    1/1     1            1           24s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-standalone-5c6cb86f45-m69nl  1/1     Running   0          2d
mysql-deployment-5496fdc956-8d4mc    1/1     Running   0          55s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get services
NAME                TYPE           CLUSTER-IP     EXTERNAL-IP      PORT(S)          AGE
kubernetes          ClusterIP      10.84.0.1      <none>           443/TCP          2d22h
mongodb-standalone  LoadBalancer   10.84.13.125   35.226.185.35    3306:31458/TCP   2d
mysql-service       LoadBalancer   10.84.0.54     34.171.222.236   3306:30136/TCP   59s
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$
```

vii. Logging into MySQL using new external IP

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ mysql -uuser -psofe4790u -h34.171.222.236
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.30 MySQL Community Server - GPL

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

viii. Running SQL Statements

```
mysql> use myDB;
Database changed
mysql> create table person( id int, age int, name varchar(50));
Query OK, 0 rows affected (0.08 sec)

mysql> insert into person values(1,30,'tom');
Query OK, 1 row affected (0.05 sec)

mysql> insert into person values(2,23,'adam');
Query OK, 1 row affected (0.05 sec)

mysql> insert into person values(3,79,'Joe');
Query OK, 1 row affected (0.05 sec)

mysql> select * from person where age>=30;
+------+------+------+
| id   | age  | name |
+------+------+------+
|    1 |   30 | tom  |
|    3 |   79 | Joe  |
+------+------+------+
2 rows in set (0.04 sec)

mysql>
```

2. MongoDB Deployment

MongoDB YAML File

```
apiVersion: v1
kind: Service
metadata:
  name: mongodb-service
spec:
  type: LoadBalancer
  ports:
    - port: 3306
  selector:
    app: database
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mongodb-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: database
  template:
    metadata:
      labels:
        app: database
    spec:
      containers:
        - image: mongo:4.0.8
          name: database
          env:
            - name: MONGO_INITDB_ROOT_USERNAME
              value: abdulbhutta
            - name: MONGO_INITDB_ROOT_PASSWORD
              value: password
```

```
    ports:
      - containerPort: 3306
        name: database
```

Designing the YAML File

The YAML file was designed similarly to the MySQL file while making minor changes. The full YAML file is provided above to view the changes that were made. The service and deployment names were changed to mongodb-service and mongodb deployment. The root username was changed to my full name and the root password was set to password. Once the services and deployments are running, the MongoDB terminal is accessed in administrator mode to create new records. The replica data from MySQL is added to the MongoDB database and a query is executed to display the person's age that is greater or equal to 30.

i. Deploy the YAML File

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl apply -f mongoDB.yaml
service/mongodb-service created
deployment.apps/mongodb-deployment created
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get services
NAME              TYPE           CLUSTER-IP    EXTERNAL-IP      PORT(S)          AGE
kubernetes        ClusterIP      10.84.0.1     <none>           443/TCP          9d
mongodb-service   LoadBalancer   10.84.8.48    35.226.185.35    3306:30647/TCP   81s
mysql-service     LoadBalancer   10.84.0.54    34.171.222.236   3306:30136/TCP   6d4h
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get deployments
NAME                 READY   UP-TO-DATE   AVAILABLE   AGE
mongodb-deployment   1/1     1            1           86s
mysql-deployment     1/1     1            1           6d4h
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$
```

ii. Execute the MongoDB in the pod

```
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl get pods
NAME                                 READY   STATUS    RESTARTS   AGE
mongodb-deployment-5c6cb86f45-zkrsv   1/1     Running   0          2m41s
mysql-deployment-5496fdc956-8d4mc     1/1     Running   0          6d4h
bhutta_abdul@cloudshell:~ (rising-ocean-362417)$ kubectl exec -it mongodb-deployment-5c6cb86f45-zkrsv -- sh
# mongo
MongoDB shell version v4.0.8
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("02327236-a2f5-4117-8c68-a9323b700d1c") }
MongoDB server version: 4.0.8
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
>
```

iii. Switch to admin

```
> use admin
switched to db admin
> show dbs
> db.auth('abdulbhutta', 'password')
1
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

iv. Create a new table and insert the first record

```
> db.person.insertOne ({id: 1, age: 30, name: "Tom"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("632cc72692f8ec0c1e5e8a2f")
}
> db.person.insertOne ({id: 2, age: 23, name: "Adam"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("632cc73992f8ec0c1e5e8a30")
}
> db.person.insertOne ({id: 3, age: 79, name: "Joe"})
{
        "acknowledged" : true,
        "insertedId" : ObjectId("632cc74792f8ec0c1e5e8a31")
}
>
```

v. Run the query to find all the person with age greater than or equal to 30

```
> db.person.find({"age" : {$gte: 30} })
{ "_id" : ObjectId("632cc72692f8ec0c1e5e8a2f"), "id" : 1, "age" : 30, "name" : "Tom" }
{ "_id" : ObjectId("632cc74792f8ec0c1e5e8a31"), "id" : 3, "age" : 79, "name" : "Joe" }
>
```