



**Faculty of Engineering and Applied Science**

**SOFE 4790U Distributed Systems**

**CRN 44425**

**Lab # 1**

**Mamun Hossain - 100553073**

**Introduction:**

The objective of this lab was to familiarize ourselves with the Google Cloud Platform, and the usage of containers and how to access them using Kubernetes.

**Discussion:**

For this lab, we began to familiarize ourselves with different softwares, such as the Google cloud platform, Docker while using Kubernetes. Docker is an open platform that allows for development, shipment, and running applications. It will allow you to be able to separate your applications in one infrastructure and transfer that over to another infrastructure without any compatibility errors (eg. Windows to MacOS). Kubernetes, also known as K8s is an open-source system for automating deployment, scaling, and management of containerized applications. There are several built-in commands using "kubectl" that will allow you to deploy applications, rolling out changes, scaling the application to fit your needs, monitoring your applications and so much more, which in turn allows a way to make managing the applications much easier.

In this lab, we ran this software to deploy a MySQL image provided to us. We were able to manage the MySQL image by searching for its IP address, deploying it, and scaling it to the needs required for us in this lab. We were also able to deploy multiple deployments once we were able to grasp an understanding, allowing us to understand how Kubernetes can manage several containers at the same time.

The main thing we want to look at is why use Docker? In our labs, why can't we just run a VM to make things how we perceive as easier? There are a lot of benefits to a VM, such as:

- All OS resources are available to the applications
- It is cheaper than having multiple machines
- Has well known security tools and controls

That being said, a VM isn't perfect. As we have deployed different containers in the lab, we can see that if we were to run multiple containers on our VM at once, it would put a lot of load on top of the system. The best way to alleviate this issue is to have a server run multiple workloads on an OS, and that's where Docker comes in. Docker lets you make containers within a program more portable and flexible to use, and Kubernetes orchestrates all that information and helps manage it as well. Some benefits with a container vs a VM would be the following:

- A container would be very lightweight, where a VM is heavyweight
- A VM starts up in several minutes, whereas a container starts up almost immediately
- VMs require too much memory space, whereas a container requires less
- A VM is very isolated, so it has much stronger security whereas a container is not as secure

With this information, we can see why it would be much easier to deploy multiple containers, such as MySQL and the extra file we had made later on the lab, because if we were to do this in the VM, it would require much more time and it would pose more difficulty as we can just simply push all the commands with the container and Kubernetes to reach the goal we want, without putting too much stress on our resources.

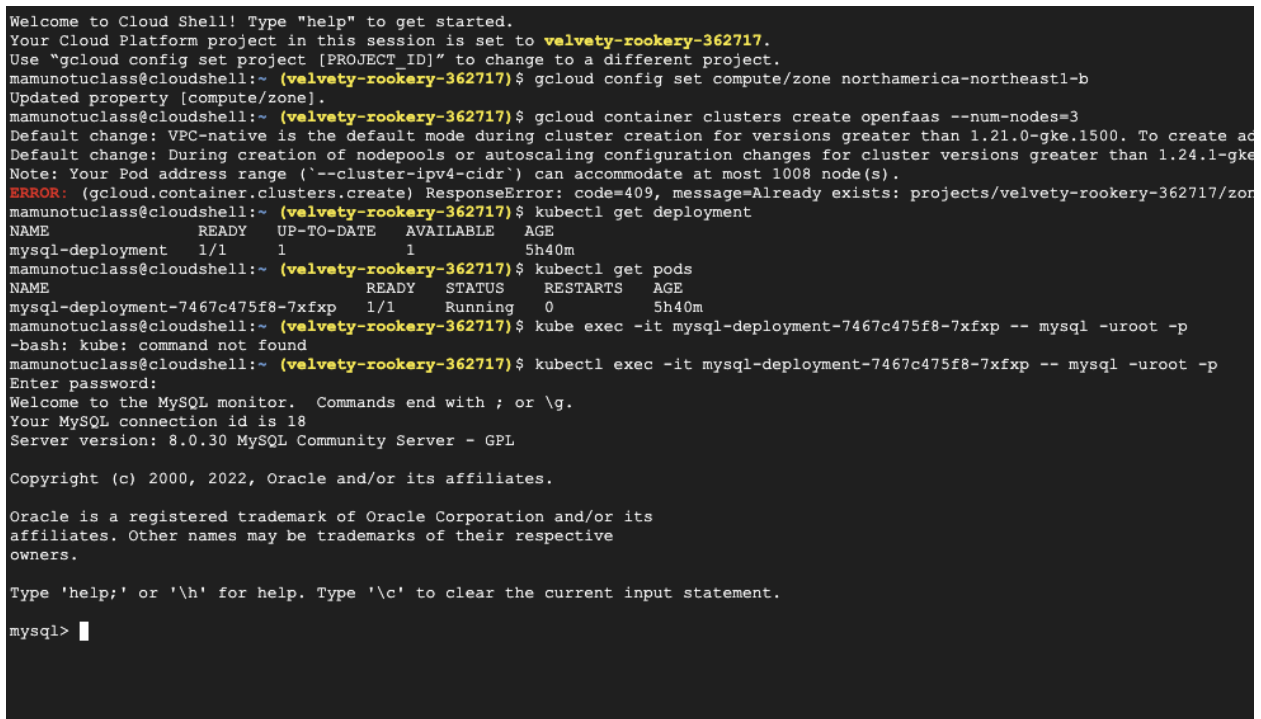
## Deployment of MySQL Database:



```
CLOUD SHELL
Terminal (velvety-rookery-362717) x + v

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to velvety-rookery-362717.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ gcloud config set compute/zone northamerica-northeast1-b
Updated property [compute/zone].
mamunotucass@cloudshell:~ (velvety-rookery-362717) $
```

## Configured time zone



```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to velvety-rookery-362717.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ gcloud config set compute/zone northamerica-northeast1-b
Updated property [compute/zone].
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ gcloud container clusters create openfaas --num-nodes=3
Default change: VPC-native is the default mode during cluster creation for versions greater than 1.21.0-gke.1500. To create a
Default change: During creation of nodepools or autoscaling configuration changes for cluster versions greater than 1.24.1-gke
Note: Your Pod address range ('--cluster-ipv4-cidr') can accommodate at most 1008 node(s).
ERROR: (gcloud.container.clusters.create) ResponseError: code=409, message=Already exists: projects/velvety-rookery-362717/zon
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
mysql-deployment  1/1     1            1           5h40m
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mysql-deployment-7467c475f8-7xfxp  1/1     Running   0           5h40m
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ kube exec -it mysql-deployment-7467c475f8-7xfxp -- mysql -uroot -p
-bash: kube: command not found
mamunotucass@cloudshell:~ (velvety-rookery-362717) $ kubectl exec -it mysql-deployment-7467c475f8-7xfxp -- mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
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>
```

Created container clusters, deployed image, received pod information and accessed MySQL server from that information

```
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> show databases
-> show databases ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'show databases' at line 2
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql>
```

Now viewing project 'My First Project' in organization 'No organization' X

## Viewed database with Kubernetes commands

```
4 rows in set (0.00 sec)

mysql> exit
Bye
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl exec -it mysql-deployment-7467c475f8-7xfxp -- mysql -uroot -p '12345'
Enter password:
ERROR 1049 (42000): Unknown database '12345'
command terminated with exit code 1
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl exec -it mysql-deployment-7467c475f8-7xfxp -- mysql -uroot -p'12345'
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 20
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> CREATE USER 'user'@'%' IDENTIFIED BY 'sofe4790u';
ERROR 1396 (HY000): Operation CREATE USER failed for 'user'@'%'
mysql> GRANT ALL PRIVILEGES ON *.* TO 'user'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql>
```

## Altered password and then exited MySQL server

```
mysql> CREATE USER 'user'@'%' IDENTIFIED BY 'sofe4790u';
ERROR 1396 (HY000): Operation CREATE USER failed for 'user'@'%'
mysql> GRANT ALL PRIVILEGES ON *.* TO 'user'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-
-> kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-
-> service
-> kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql--service
-> kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-service
-> exit
-> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-
kub' at line 1
mysql> exit
Bye
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-services
service/mysql-services exposed
mamunotuc@cloudshell:~ (velvety-rookery-362717)$
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get service
NAME                TYPE                CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes           ClusterIP           10.80.0.1        <none>            443/TCP          5h52m
mysql-service        LoadBalancer       10.80.6.241      34.85.27.245     3306:30346/TCP   4h16m
mysql-services       LoadBalancer       10.80.1.119      <pending>        3306:31549/TCP   13s
mamunotuc@cloudshell:~ (velvety-rookery-362717)$
```

## Retrieved external IP

```

mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl expose deployment mysql-deployment --type=LoadBalancer --name=mysql-services
service/mysql-services exposed
mamunotuc@cloudshell:~ (velvety-rookery-362717)$
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get service
NAME      TYPE      CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
kubernetes  ClusterIP  10.80.0.1        <none>           443/TCP          5h52m
mysql-service  LoadBalancer  10.80.6.241      34.95.27.245     3306:30346/TCP   4h16m
mysql-services  LoadBalancer  10.80.1.119      <pending>        3306:31545/TCP   13s
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ mysql -uuser -psofe4790u -h34.95.27.245
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 21
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>

```

Used external IP to access MySQL server in a different method

```

mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl apply -f mysql.yaml
Error from server (Invalid): error when creating "mysql.yaml": Service "mongoDB-service" is invalid: metadata.name: Invalid value: "mongoDB-service": name must be a DNS subdomain (RFC 1035, not RFC 1123) and must not start with a digit, regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?'
Error from server (Invalid): error when creating "mysql.yaml": Deployment.apps "mongoDB" is invalid: [metadata.name: Invalid value: "mongoDB": name must be a DNS subdomain (RFC 1035, not RFC 1123) and must not start with a digit, regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?'], spec.template.spec.containers[0].name: Invalid value: "mongoDB": name must be a DNS subdomain (RFC 1035, not RFC 1123) and must not start with a digit, regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?'], spec.template.spec.containers[0].ports[0].name: Invalid value: "mongoDB": name must be a DNS subdomain (RFC 1035, not RFC 1123) and must not start with a digit, regex used for validation is '[a-z]([-a-z0-9]*[a-z0-9])?']
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl apply -f mysql.yaml
service/mysql-service unchanged
deployment.apps/mysql-deployment configured
mamunotuc@cloudshell:~ (velvety-rookery-362717)$

```

Moving onto the advanced method of the deployment, deploying through the mysql.yaml file

```

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> kubectl delete deployment mysql-deployment
--> show databases;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'kubectl delete deployment mysql-deployment
show databases' at line 1
mysql> exit
Bye
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl delete deployment mysql-deployment
deployment.apps "mysql-deployment" deleted
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl delete service mysql-service
service "mysql-service" deleted

```

Deleting the previous iterations of the deployments to create ease of access

```
(velvety-rookery-362717) x + ▾
Open Editor

mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl delete deployment mysql-deployment
deployment.apps "mysql-deployment" deleted
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl delete service mysql-service
service "mysql-service" deleted
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl apply -f mysql.yaml
service/mysql-service created
deployment.apps/mysql-deployment created
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
mysql-deployment 1/1     1            1           24s
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get deployment
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
mysql-deployment 1/1     1            1           31s
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get pods
NAME          READY   STATUS    RESTARTS   AGE
mysql-deployment-5496fdc956-zw8qg 1/1     Running   0           40s
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get service
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
kubernetes    ClusterIP     10.80.0.1    <none>        443/TCP           22h
mysql-service  LoadBalancer 10.80.10.133 35.234.253.184 3306:31018/TCP    51s
mysql-services LoadBalancer 10.80.1.119  34.152.52.239 3306:31545/TCP    16h
mamunotuc@cloudshell:~ (velvety-rookery-362717)$
```

Retrieved all information again (pod, deployment, service)

```
mysql.yaml > ...
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mysql-service
5  spec:
6    type: LoadBalancer
7    ports:
8      - port: 3306
9    selector:
10     app: mysql
11 ---
12 apiVersion: apps/v1
13 kind: Deployment
14 metadata:
15   name: mysql-deployment
16 spec:
17   replicas: 1
18   selector:
19     matchLabels:
20       app: mysql
21   template:
22     metadata:
23       labels:
24         app: mysql
25     spec:
26       containers:
27       - image: mysql/mysql-server
28         name: mysql
29         env:
30           - name: MYSQL_ROOT_PASSWORD
31             value: password
32           - name: MYSQL_USER
33             value: user
34           - name: MYSQL_PASSWORD
35             value: sofe4790u
36           - name: MYSQL_DATABASE
37             value: myDB
38       ports:
39       - containerPort: 3306
40         name: mysql
```

Code required to run the server with all the necessary information (ie. Port, Load balancer, image, type, name, apps)

```

mysql> exit
Bye
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get service
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes          ClusterIP   10.80.0.1      <none>         443/TCP          23h
mysql-service        LoadBalancer 10.80.10.133   35.234.253.184 3306:31018/TCP    19m
mysql-services       LoadBalancer 10.80.1.119    34.152.52.239 3306:31545/TCP    17h
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ mysql -uuser -psofe4790u -h35.234.253.184
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 24
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> use myDB;
Database changed
mysql> create table person( id int, age int, name varchar(50));
Query OK, 0 rows affected (0.07 sec)

mysql> insert into person values(1,30,'tom');
Query OK, 1 row affected (0.04 sec)

mysql> insert into person values(2,23,'adam');
Query OK, 1 row affected (0.03 sec)

mysql> insert into person values(3,79,'Joe');
Query OK, 1 row affected (0.03 sec)

mysql> select * from person where age>=30;
+-----+-----+-----+
| id | age | name |
+-----+-----+-----+
| 1 | 30 | tom |
| 3 | 79 | Joe |
+-----+-----+-----+
2 rows in set (0.03 sec)

mysql>

```

Entered commands from the lab to display database information

## MongoDB deployment:

```

CLOUD SHELL
Terminal (velvety-rookery-362717) x +
Open Editor

Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to velvety-rookery-362717.

mongo-deployment 0/1 1 0 4d9h
mongodb 1/1 1 1 4d9h
mongodb-deployment 1/1 1 1 46s
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get service
NAME                TYPE        CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes          ClusterIP   10.80.0.1      <none>         443/TCP          6d17h
mongo               ClusterIP   10.80.0.103    <none>         27017/TCP        4d9h
mongodb-service      LoadBalancer 10.80.11.26    35.234.253.184 3306:31458/TCP    51s
mysql-services       LoadBalancer 10.80.1.119    34.152.52.239 3306:31545/TCP    6d11h
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl get pods
NAME                READY   STATUS    RESTARTS   AGE
mongo-deployment-6c8cc9d876-k6thn 0/1     ImagePullBackOff 0          47h
mongodb-5d75dc9555-ft9jm 1/1     Running    0          47h
mongodb-deployment-57dd4d4b79-bjfgw 1/1     Running    0          33m
mamunotuc@cloudshell:~ (velvety-rookery-362717)$ kubectl exec -it mongodb-deployment-57dd4d4b79-bjfgw -- sh
# mongosh
Current Mongosh Log ID: 632dba34a6a440ce1e2b23cb
Connecting to:
  mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.5.4
Using MongoDB:
  6.0.1
Using Mongosh:
  1.5.4

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2022-09-23T13:17:53.324+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See https://dochub.mongodb.org/core/prodnotes-filesystem
2022-09-23T13:17:54.094+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2022-09-23T13:17:54.094+00:00: vm.max_map_count is too low
-----

-----
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

```

Following a similar approach to the advanced deployment method for MySQL, we deploy the MongoDB.yaml in similar fashion retrieving the pods, deploying the image, and retrieving the service for the IP. Then accessing the server for mongoDB, we will then be able to access myDb to alter the database for mongoDB as well.

```
test> use myDb
switched to db myDb
myDb> yse test
Uncaught:
SyntaxError: Missing semicolon. (1:3)

> 1 | yse test
    |   ^
    2 |

myDb> use test
switched to db test
test> show dbs
admin    40.00 KiB
config   60.00 KiB
local    40.00 KiB
test>

test> db.test.insert({name: "jack", age: 24})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("632dbb1182404578747715e0") }
}
test> db.test.inset({name: "bob", age: 31})
Uncaught:
SyntaxError: Missing semicolon. (1:13)

> 1 | db.test.inset({name: "bob", age: 31})
    |           ^
    2 |

test> db.test.insert({name: "bob", age: 31})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("632dbb4282404578747715e1") }
}
test> db.test.insert({name:
...
... db.test.insert
... clear
Uncaught:
SyntaxError: Unexpected token, expected ",", (4:0)
```

Seeing the kibibytes for each test, and then manually adding in the information for the database.test to ensure our code works.

```
SyntaxError: Unexpected token, expected ",", (4:0)

2 |
3 | db.test.insert
> 4 | clear
    |   ^
    5 |

test> db.test.insert({name: "george", age: 28})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("632dbb7f82404578747715e2") }
}
test> show collections
test
test> db.test.find()
[
  { _id: ObjectId("632dbb1182404578747715e0"), name: 'jack', age: 24 },
  { _id: ObjectId("632dbb4282404578747715e1"), name: 'bob', age: 31 },
  {
    _id: ObjectId("632dbb7f82404578747715e2"),
    name: 'george',
    age: 28
  }
]
```



Testing our code to make sure we can find the ages that are greater than 23 and ensure that the database is working the way we want.

```
mongodb.yaml > ...
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mongodb-service
5  spec:
6    type: LoadBalancer
7    ports:
8      - port: 3306
9    selector:
10     app: mongo
11 ---
12 apiVersion: apps/v1
13 kind: Deployment
14 metadata:
15   name: mongodb-deployment
16 spec:
17   replicas: 1
18   selector:
19     matchLabels:
20       app: mongodb
21   template:
22     metadata:
23       labels:
24         app: mongodb
25     spec:
26       containers:
27       - image: mongo
28         name: mongodb
29         env:
30           - name: MONGODB_ROOT_PASSWORD
31             value: password
32           - name: MONGODB_USER
33             value: user
34           - name: MONGODB_PASSWORD
35             value: sofe4790u
36           - name: MONGODB_DATABASE
37             value: myDB
38       ports:
39       - containerPort: 3306
40       name: mongodb
```

Code required to access mongoDB server.

## **Conclusion:**

In conclusion, we were able to successfully utilize Kubernetes and enhance our knowledge on Docker to set up different containers and use different applications with .yaml files and increase our knowledge on containerizations.