



SOF 4790U Distributed Systems  
Lab 01 Introduction to Google Kubernetes Engine (GKE)  
18 September 2022  
Group 1- Michelle Cheng 100696572

## Introduction

The purpose of this lab is to introduce the use of Google Kubernetes Engine (GKE) and to gain an understanding with docker, kubernetes and the google cloud platform (GCP).

Main Objectives:

1. Get familiar with Docker images and containers.
2. Learn various Kubernetes tools.
3. Learn how to use Google Cloud Platform (GCP).
4. Compose YAML files to deploy cloud applications.

## Discussion

### *Docker*

Docker is a tool used to deploy applications in lightweight containers in order to improve efficiency for them in different system environments. These containers consist of dependencies and suitable frameworks required for an application to run and are isolated for each application.

Docker image is a set of instructions in which docker containers are created from and used to execute code in a container. A container is a runtime instance of the docker image and allows dependencies to be packaged with the application. Docker consist of a Docker Hub where all docker images can be found, managed and shared amongst various container images.

### *Kubernetes*

Kubernetes is a platform that uses containers and clusters in order to manage workload and services on a containerized application. It works by replicating the service and deploying more containers when needed. Load balancing services are also used in Kubernetes by using a load balancer to direct the IP address to the replicated service. This supports scalability, availability and recovery of containers when a failure is detected.

Kubernetes worker nodes run containerized applications and a master node controls them. This set of worker and master nodes is known as a cluster. Pods contain the containerized application, deployment manages the replicas of these pods and service manages connectivity of the pods.

### *Docker images vs virtual machines*

Docker occupies less space due to the lack of a guest OS and uses less memory compared to a virtual machine. This factor speeds up the boot-up time needed and allows for high efficiency. Docker is also held on a single engine which allows for better performance and scalability. It is portable across different platforms and the data can be shared between various containers which is not possible with virtual machines.

## Design

MongoDB is a document-oriented database which is used for high volume data storage. It utilizes collections which contain sets of documents. In order to deploy MongoDB using GKE a mongodb.yaml file was configured.

The service parameters with name mongodb-service were configured as seen below. 27017 is the default port used for MongoDB.

```

1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: mongodb-service
5  spec:
6    type: LoadBalancer
7    ports:
8      - port: 27017
9    selector:
10     app: mongodb
11

```

Deployment parameters with name mongodb-deployment were configured as seen below. The deployment creates 1 replicated pod and the containerPort was also directed to 27017.

```

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: MONGO_ROOT_PASSWORD
31                value: password
32              - name: MONGO_USER
33                value: user
34              - name: MONGO_PASSWORD
35                value: sofe4790u
36              - name: MONGO_DATABASE
37                value: MongoDB
38            ports:
39              - containerPort: 27017
40              name: mongodb
41

```

In order to deploy the file to GKE, the command below was used to create the mongodb-service and mongodb-deployment.

```
michelle0109cheng@cloudshell:~ (agile-skyline-362514)$ kubectl apply -f mongodb.yaml
service/mongodb-service created
deployment.apps/mongodb-deployment created
```

As shown below, one pod with the name mongo-db-deployment-58bd5586cb-h4jppj was created and is running as expected. The mongodb-service is running on an external IP 35.152.54.40 with the mongodb-deployment.

```
michelle0109cheng@cloudshell:~ (agile-skyline-362514)$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
mongodb-deployment-58bd5586cb-h4jppj 1/1     Running   0           18s
mysql-deployment-5496fdc956-22rm8     1/1     Running   0           9m57s

michelle0109cheng@cloudshell:~ (agile-skyline-362514)$ kubectl get service
NAME            TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes      ClusterIP     10.80.0.1     <none>         443/TCP          8h
mongodb-service  LoadBalancer 10.80.8.47    34.152.54.40   3306:32059/TCP   6m28s
mysql-service    LoadBalancer 10.80.12.131  35.234.253.50  3306:31597/TCP   16m
mysqlservice     LoadBalancer 10.80.8.150   34.152.27.191  3306:32048/TCP   7h53m

michelle0109cheng@cloudshell:~ (agile-skyline-362514)$ kubectl get deployment
NAME                READY   UP-TO-DATE   AVAILABLE   AGE
mongodb-deployment  1/1     1             1           9m17s
mysql-deployment    1/1     1             1           18m
```

## Deliverable

Audible video link:

[https://drive.google.com/file/d/1wrWFg7gaeqZaN\\_q6c9XEdeXtQSI-9YU/view?usp=sharing](https://drive.google.com/file/d/1wrWFg7gaeqZaN_q6c9XEdeXtQSI-9YU/view?usp=sharing)

## Resources

- <https://github.com/GeorgeDaoud3/SOFE4790U-lab1>
- <https://www.youtube.com/watch?v=rOTqprHv1YE>
- <https://www.youtube.com/watch?v=cC46cg5FFAM>