



Faculty of Engineering and Applied Science

SOFE 4790U Distributed Systems

Lab 1: Intro to K8S

**Group 19**

Sunil Tumkur 100620430

<https://github.com/sunilt4/Distributed-Systems>

## Discussion

I have learned that docker is a platform where us developers can build, deploy and manage containerized applications. Images are single files that have all the dependencies and configuration needed in order for the application to run. Containers are an instance of an image that runs the application. Kubernetes, I learned that it is a platform which we can use with docker to automate container operations.

The advantages to using a container are:

- Lightweight and fast
- Efficient
- Portable across different platforms
- Better performance

The disadvantages to using a container are:

- Can only run one OS
- Less secure as they share the same OS

The advantages to using a virtual machine are:

- Security
- Can emulate and run more than 1 OS
- More resources are available to your use

The disadvantages to using a virtual machine are:

- Takes up a lot of memory/space
- Boot up time is long
- Hard to port over the cloud

## Design

### Video 1:

[https://drive.google.com/file/d/1kRnGCw1IZ6HPWrGCIVk\\_2VLLWmWG4r3Q/view?usp=sharing](https://drive.google.com/file/d/1kRnGCw1IZ6HPWrGCIVk_2VLLWmWG4r3Q/view?usp=sharing)

## Video 2:

[https://drive.google.com/file/d/11Q9jG9HBSCuOx2ovwHIRQa39gOgla7I\\_/view?usp=sharing](https://drive.google.com/file/d/11Q9jG9HBSCuOx2ovwHIRQa39gOgla7I_/view?usp=sharing)

```
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ gcloud config set compute/zone northamerica-northeast1-b
Updated property [compute/zone].
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ gcloud container clusters create mongo-db-lab1 --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 advanced routes based clusters, please pass the '--no-enable-ip-alias' flag
Default change: During creation of nodepools or autoscaling configuration changes for cluster versions greater than 1.24.1-gke.800 a default location policy is applied. For Spot and FVM it defaults to ANY, and for all other VM kinds a BAIANC
ID policy is used. To change the default values use the '--location-policy' flag.
Note: Your Pod address range ('--cluster-ip4-cidr') can accommodate at most 1000 node(s).
Creating cluster mongo-db-lab1 in northamerica-northeast1-b... Cluster is being health-checked (master is healthy)...done.
Created [https://console.cloud.google.com/v1/projects/sofe4790-group19-lab1/zones/northamerica-northeast1-b/clusters/mongo-db-lab1].
To inspect the contents of your cluster, go to: https://console.cloud.google.com/kubernetes/workload/_gcloud/northamerica-northeast1-b/mongo-db-lab1/project-sofe4790-group19-lab1
kubectl config entry generated for mongo-db-lab1.
NAME: mongo-db-lab1
LOCATION: northamerica-northeast1-b
MASTER VERSION: 1.22.11-gke.400
MASTER_IP: 35.203.52.47
MACHINE_TYPE: c2-medium
NODE VERSION: 1.22.11-gke.400
NUM_NODES: 3
STATUS: RUNNING
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl get deployment
No resources found in default namespace.
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl get pods
No resources found in default namespace.
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl get clusters
error: the server doesn't have a resource type "clusters"
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl get cluster
error: the server doesn't have a resource type "cluster"
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl get nodes
NAME                                STATUS  ROLES  AGE  VERSION
gke-mongo-db-lab1-default-pool-efbe113-qvrf  Ready  <none>  67s  v1.22.11-gke.400
gke-mongo-db-lab1-default-pool-efbe113-lrkn  Ready  <none>  67s  v1.22.11-gke.400
gke-mongo-db-lab1-default-pool-efbe113-xx26  Ready  <none>  67s  v1.22.11-gke.400
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ C
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl apply -f statefulsets.yaml
statefulset.apps/mongodb-standalone created
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl apply -f service.yaml
service/database created
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl get pod
NAME                                READY  STATUS  RESTARTS  AGE
mongodb-standalone-0                1/1    Running  0          31s
suniltumkur4@cloudshell:~ (sofe4790-group19-lab1) $ kubectl exec -it mongodb-standalone-0 -- sh
```

```
# mongo -u admin -p --authenticationDatabase admin
MongoDB shell version v4.0.8
Enter password:
connecting to: mongodb://127.0.0.1:27017/?authSource=admin&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3a727b80-6816-4770-b349-743783de68c0") }
MongoDB server version: 4.0.8
Server has startup warnings:
2022-09-15T15:55:13.315+0000 I STORAGE [initandlisten]
2022-09-15T15:55:13.315+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-09-15T15:55:13.315+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
---
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.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show dbs
admin      0.000GB
config    0.000GB
local     0.000GB
> db
test
> use testDB
switched to db testDB
> db
testDB
> db.testNames.insert(
... (name: 'Sunil')
... )
WriteResult({ "nInserted" : 1 })
> show collections
testNames
> db.testNames.find()
{ "_id" : ObjectId("63235617eb396bb921d6facd"), "name" : "Sunil" }
> ^C
bye
#
```

MongoDB is another type of database that is used. It does not use the traditional structure as a SQL database where data is stored in schemas and tables. Instead, this database will store data in the format of JSON based documents. MongoDB was used in the lab and it was configured using kubernetes and a YAML file. All of this was required and used as it was stated within the lab instructions. Also, to configure MongoDB with Kubernetes, a YAML file is needed for configuration.

### **Summary of Problem**

- Get familiar with Docker images and containers.
- Learn various Kubernetes tools.
- Learn how to use Google Cloud Platform (GCP).
- Compose mysql and MongoDB YAML files to deploy cloud applications.

### **Summary of Solution**

- Followed the lab instructions which had commands on how to deploy the mysql application.
- Had to create clusters, nodes, pods, and YAML files for configuration

### **A list of things you did not understand and/or your questions:**

- How do we know what properties/keys to add in a YAML file as each YAML file will be configured differently and won't have the same keys. Every application will require different YAML files.

### **Keywords that are fundamental for understanding the text/ video and/ or that were new to you:**

- MongoDB is another type of database that is used. It does not use the traditional structure as a SQL database where data is stored in schemas and tables. Instead, this database will store data in the format of JSON based documents.
- YAML is a type of language in which it is used for creating configuration files. These configuration files can be necessary for applications to run.