



Faculty of Engineering and Applied Science

**SOFE 4790U: Distributed Computing**

Fall 2022

Dr. Mohamed El-darieby

TA: Ghadeer Abdelkader

# **Lab 1:**

## **Introduction to Google Kubernetes Engine (GKE)**

Lab Group 20

Adris Azimi (100704571),

Deepak Thangella (100695929),

Harris Athwal (100750935),

Krystian Rusin (100704636)

# Discussion

## Docker

**Software:** The Docker daemon is directly responsible for managing the containers and their respective objects. The daemon actively listens for API requests from the Docker Engine API. The Docker client allows users to interact with Docker, specifically Docker daemons, through a command line interface.

**Objects:** Objects in Docker are what make up an application in Docker. The main objects are images, containers, and networks, in addition to many more.

An image is a read-only binary template file that is used to create a container, it holds all of the metadata required.

A container is an instance of an image that can be modified via the Docker API or CLI.

Networks are what allow communication between docker containers.

**Registries:** Registries are where docker images are stored, the default registry is Docker Hub. Images can be uploaded or downloaded (also known as pulling and pushing) from the corresponding registry.

### Docker compared to VMs:

Advantages:

- Very memory efficient
- Centralized filesystem between containers
- Quick boot time
- Much more lightweight as it does not require its own OS for each container

Disadvantages:

- Can't run on any OS
- Not as isolated as a VM and therefore not as secure
- Lack of GUI applications

## Kubernetes

Kubernetes is a portable and open-source container management system. Kubernetes is mainly deployed with Docker as it was initially designed to solely work with Docker.

The control plane is the main aspect of the Kubernetes cluster. It makes all of the global decisions through various components.

These components are:

- etcd
  - Key value store that is persistent and distributed. Contains configuration and state data of the cluster.
- API server
  - Provides the external and internal interface to K8s using JSON over HTTP.
- Scheduler
  - Listens for freshly created pods with no attached node, and assigns a node for that unscheduled pod.
- Controller
  - Control loop that moves the cluster state to the desired state via communicating with the API server to modify resources accordingly.
- Controller manager
  - Manages a set of core controllers

Nodes are the machines where containers are deployed.

- Kubelet
  - Monitors the state of each node in the cluster ensuring that containers are running in a pod in accordance with the desired state.
- Kube-proxy
  - A combination of a network proxy and a load balancer runs on each node of the cluster. It maintains network activity by routing traffic to certain containers as needed.
- Container
  - Stored inside a pod. It holds applications, libraries and dependencies.

## Design

MongoDB:

We used the marketplace in GCP to deploy an instance of MongoDB which we were able to connect remotely and execute various commands such as `show dbs`, `db.<collection>.insert`, and `db.<collection>.find`.

# Deliverables

## Group Submission Deliverable

<https://drive.google.com/drive/folders/1miB8YNKj704SzhDOIX8-qaYHCfrxxVjd?usp=sharing>

## Individual Work

Adris Azimi:

<https://drive.google.com/drive/folders/1d2KXtSfGB-EBTANKOebIhRNxVbrmdLuu?usp=sharing>

Deepak Thangella:

<https://drive.google.com/drive/folders/1miB8YNKj704SzhDOIX8-qaYHCfrxxVjd?usp=sharing>

Harris Athwal:

<https://drive.google.com/drive/folders/14-VFiUnZaV4HpTFJTaNFM25rgyZEa25?usp=sharing>

Krystian Rusin:

<https://youtu.be/4IKswh6OMnQ>