Answer the following questions:

**What are docker image, container, and registry?**

A Docker image is the file that is used to run code in a Docker container. It contains all the essential files needed to run the application. A Docker container is a container to package software and its dependencies so that it is easier to run the application in different environments without causing any problems. A Docker registry is a way to store and distribute Docker images with different versions of these images.

**List the Docker commands used in the video with a brief description for each command and option.**

* docker build -t <tag> <path>
  + This command builds the Docker image
  + -t option is to specify the name of the image and the user can optionally specify the version of that image
* docker run [-d] <image>
  + This command runs the Docker container which means the code in the Docker image gets executed
  + -d option is the running the image in detach mode which means the container will start and run in the background without displaying in the console
* docker ps [-a]
  + This command displays all currently running containers
  + -a option shows all Docker containers including containers that already ran and terminated
* docker images
  + This command displays all Docker images created with the image ID, when it was created and the size of the image
* docker logs <Container ID>
  + This command will show what is happening on the container

**At the end of the video, there are two running containers, what commands can be used to stop and delete those two containers?**

* docker stop <container name/ID>
* docker rm <Container name/ID>

Answer the following questions:

**What’s a multi-container Docker application?**

A multi-container Docker application is an application which consists of multiple containers. It packages multiple containers to a single application which makes it easier and require less work to deploy. An example of this would be a web application with a database. In this situation, there will be two containers. One for the web application and another container for the database.

**How these containers are communicated together?**

The containers communicate with each other through a bridge network.

**What command can be used to stop the Docker application and delete its images?**

* Docker-compose down –rmi all

**List the new docker commands used in the video with a brief description for each command and option.**

* Docker pull <image name>
  + Pulls an image or repository from a registry
* Docker run –name <Name>-d -e <var\_name=var\_value> -p <host machine port:container port>
  + This command runs the Docker container
  + -name option specifies a name to the container
  + -e option sets the environmental variables of the container
  + -p option publishes the containers port to the host
* Docker network create <name>
  + This command creates a bridge network for the containers to communicate
* Docker network ls
  + This command lists all networks daemon knows about

**List all used GCP shell commands and their description in your report.**

* gcloud config set project <project name>
  + This command sets the project being used in the cloud console
* docker cp <filename> <container-id>:<container-path>
  + This command copies files from the local filesystem to the container
* docker commit <container id> <img-name>
  + This command creates a new image from the specified container
  + Used for when creating images when changes to the container has been made
* docker tag <source-img-name> <img-name>
  + This command creates another image that refers to a source image
* docker push <img-name>
  + This command pushes the image to the container registry
* gcloud config set compute/zone <region>
  + This command changes the region of the project to the region specified
* gcloud container clusters create <cluster-name> --num-nodes=<num-nodes>
  + This command creates a cluster for containers with the specified name
  + –num-nodes option just lets the user specify the number of nodes to be created in the cluster
* gcloud container clusters get-credentials <cluster-name>
  + This command retrieves the credentials for a running cluster
  + This updates the kubeconfig file with the credentials of the cluster
* kubectl create deployment <deployment-name> --image=<img-name>
  + This command creates a deployment with the specified name and image
* kubectl expose deployment <deployment-name> --type <service> --port <port> --target-port <port>
  + This command exposes the deployment with the specified name
  + --type option allows the user to specify what type of service like for example a LoadBalancer
* kubectl get pods
  + This command lists all pods
* kubectl get service <deployment-name>
  + This command lists the service with the specified name.
  + If used without the deployment name, it will list all services

**Apply the YML file into Kubernetes and run the server (what is the appropriate Cloud shell command?).**

* kubectl apply -f webApp.yml

Answer the following question:

**What is Kubernetes’ pod, service, node, and deployment?**

* A Kubernetes’ pod is like a wrapper for a group of containers with shared storage and resources.
* Expose an application running on a set of pods as a network service
* Kubernetes node is a worker machine where the pod runs on.
* Kubernetes deployment represents a set of multiple, identical Pods with no unique identities. It runs multiple replicas of your application and automatically replaces any instances that fail or become unresponsive.

**What’s meant by replicas?**

* A replica is an identical copy of a Pod

**What are the types of Kubernetes’ services? what is the purpose of each?**

* ClusterIP – Exposes the service on a cluster-internal IP
* NodePort – Exposes a service via a static port on each node’s IP
* LoadBalancer – Exposes the service via the cloud provider’s load balancer
* ExternalName – Similar to other services but can be accessed with name instead of the clusterIP address