Group 2

1. What are docker images, container, and registry?

Docker images are like classes that are run by containers or docker. A container is a component that, like virtual machines, allows you to run real-time images or programs without a guest OS, a registry is like a storage or a large collection/composition of repositories for those images used by docker

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

- Docker version - this simply outputs the version and status of docker used in the terminal
- Docker build -t hello-world:1.0 . - this command builds the docker file specifying the tag 1 and hello world image name
- Docker images - lists the current docker images
- Docker run -r hello-world:1.0 - the docker run command with specification of the image name and tag will run this program
- Docker ps - lists containers currently running
- Docker ps -a - lists all containers running and stopped
- Docker build -t hello-world:2.0 . - this command builds the docker file specifying the tag 2 and hello world image name
- Docker run -d hello-world:2.0 - the docker run command with specification of the hello world image name and tag 2 will run this program
- Docker logs <container id> - see what is happening in the container

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

- Docker kill <container id> - will stop the containers from running or being active
- Docker -rm <container id> will remove or delete the container

4. What's a multi-container Docker application?

A multi-container Docker application is a system where each container has at least one other container that it can communicate with. These containers all serve a specific purpose/job within the docker system.

5. How are these containers communicated together?

Since each works to perform a specific task for a system. Containers run in isolation however “networking” can be used to connect containers to each other and communicate (send/receive) data or information with each other.

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

- sudo docker create --name firstkill bitnami/wordpress - this will kill or stop the docker container using its name.
- docker image -rmi [image id]

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

- Docker pull mysql: pulls the mysql image for source
- Docker run --name app-db -d -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=myDB mysql: creates a container called “app-db” which will set a root password for mysql database and gives the database name “myDB” with mysql as the image name
- Docker run --name app -d my-web-app:1.0: runs the app in detached mode using the name option for the container.
- Docker rm -f app: stops the container using “-f” and removes it with “rm”
- Docker run --name app -d -p 8080:8080 my-web-app:1.0: Docker allows port 8080 to be available outside of docker and the host machine's port 8080 will be bound to it creating a new application container.
- Docker run --name app -d -p 8081:8080 my-web-app:1.0: Docker instead sets the host machine port to 8081.
- Docker network create app-network: creates a network between containers set for communication.
- Docker network ls: lists all the networks in docker for the containers.
- Docker network connect app-network app-db: connects app-db container to the app network.
- Docker run --name app -d -p 8080:8080 network=app-network my-web-app:1.0: re-creating application container with network in same command.
- Docker compose-up: brings application up by starting and bringing in both containers.

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

- Docker run -p 8080:80 nginx:latest: simply begins running docker container in port 8080 as host.

- Docker cp index.html<container-id>:/usr/share/nginx/html/: specifies the html file that will be used to make the side into container id.
- Docker commit<container-id> cad/web:version1: commits changes made to container based on container id.
- Docker tag cad/web:version1 us.gcr.io/youtube-demo-255723/cad-site:version1: specifies tag and name of container
- Docker push us.gcr.io/youtube-demo-255723/cad-site:version1: pushes the following “cad-site”
- Gcloud config set project youtube-demo-255723: configure the project name to default.
- Gcloud config set compute/zone us-central1-a: set time zone for the gke cluster before creation.
- Gcloud container clusters create gk-cluster --num-nodes=1: actually creates the gk cluster for the containers.
- Gcloud container clusters get-credentials gk-cluster: gives credentials to the gk cluster.
- Kubectl create deployment web-server --image=us.gcr.io/youtube-demo-255723/cad-site:version1: deploys the web server using the image.
- Kubectl expose deployment hello-server --type LoadBalancer --port 80 -target-port 8080: exposes the deployed web server with port specifications.
- Kubectl get pods: shows you the running pods.
- Kubectl get service hello-server: shows you the server specified after the “get” option.

9. What is Kubernetes’ pod, service, node, and deployment?

- **Pod**: pods are the basic deployable and smallest objects in kubernetes and can hold multiple containers. They are essentially running cluster processes that act as single instances.
- **Service**: A service is an abtractual representation for a cluster or group of pods in docker.
- **Deployment**: An instruction that tells kubernetes how the creation and modification of instances for pods should be done.

10. What’s meant by replicas?

When multiple pod instances are run in a process to keep the number of pods constant, this is called a replica. This allows the users to still have access to the application in case a pod fails or is no longer accessible.

11. What are the types of Kubernetes' services? What is the purpose of each?

- **ClusterIP**: Services accessible only within clusters are able to be shown.
- **NodePort**: Each node's IP will expose the service from the port (that is static).
- **LoadBalancer**: The loadbalancer provided by the cloud provider will expose the service.
- **ExternalName**: A predefined external name will map a service by having the CNAME return a value.

https://drive.google.com/drive/u/0/folders/1ivVG-BVKgWnIi_p459wYn1sHu3zk78pS

Cloud Computing - Google Drive → **for videos**