Cloud Computing Project Milestone 1

Owais Quadri 100697281

Questions

What are docker image, container, and registry?

Docker Image: A compiled set of instructions containing the code and dependencies of an app that is used to create a docker container.
Docker Container: An instance of a docker image that is isolated from the host machine, to create an ideal environment for a program or service based on the requirements specified.
Docker Registry: The location where the docker containers can be stored.

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

docker build -t helloworld:1.0 .

- This docker command builds the image named ‘helloworld’ with the ‘1.0’ tag (-t) using a Dockerfile in the current directory specified by ‘.’.

docker ps / docker ps -a

- This command can be used to view the current docker containers running. If the ‘-a’ tag is included, the past docker containers are included as well.

docker images

- This command shows the current docker images in the local system.

docker run helloworld:1.0 / docker run -d helloworld:2.0

- This command allows you to run an image specified by name and tag on a new container. If the tag ‘-d’ is included, the container runs in the background (detached).

docker logs containerID

- where ‘containerID’ is at least the first 4 letters of the container ID or name, this command will print the logs for that running container.

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

The commands that can be used to stop and delete those commands are: “docker stop containerID” and “docker rm containerID” where ‘containerID’ is at least the first 4 letters of the container ID or the name.

Prepare a video showing the container(s) created on your machine, displaying their logs, stopping them, and then deleting them.

Video 1 Link:
https://drive.google.com/file/d/17HzI2hifvCjqKF9qVnOWUGrEiEarDyQQ/view?usp=sharing

What's a multi-container Docker application? And how are these containers communicated together?

A multi container Docker application is an application that connects multiple containers for certain services for an application. Communication between containers is possible through bridge networks. In the video, an Apache tomcat java web application communicates with a mySQL server from a separate container to run the application.

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

The command that can be used to stop the Docker application is the docker-compose stop and docker-compose rm.

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

docker pull mysql

- pulls mysql image from public registry for local use.

docker run –name app-db -d -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=myDB mysql

- runs the mysql image in a detached '-d' container called 'app-db' with environment variables using '-e' tag to specify the root password and target database name.

docker run –name app -d -p 8081:8080 my-web-app:1.0

- runs my-web-app:1.0 on a container named 'app' that is exposing the container port 8080 and connecting it to the system port 8081. Different port names used for clarity.

docker network create app-network

- By default, this command creates a bridge network named 'app-network'. Allows communication between containers.

docker network ls

- shows all networks

docker network connect app-network app-db

- connects 'app-db' container to 'app-network' network

docker compose up -d

- build (if not already built) and start services specified in the docker-compose.yml file

Prepare a video showing the created application, run the webapp, stop the application and delete the application containers.

Video 2 Link:
https://drive.google.com/file/d/1N7ir92IQ9yD-H8I6Hwyh0Se4cbMNEVwI/view?usp=sharing

Prepare a video showing how the container is deployed using Docker and Kubernetes in GCP.

Video 3 Link:
https://drive.google.com/file/d/1UUQPE3BHqoAsUn57tq6L6pkx5YLw3C5r/view?usp=sharing

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

gcloud config set project project-name

- Set the current project using the project ID.

docker run -d -p 8080:80 nginx

- runs nginx image in a detached container, exposing container port 80 (nginx server) and sending to system port 8080.

docker cp index.html containerID:/usr/share/nginx/html/

- copying index.html into the nginx container using its container ID and path to nginx html files.

docker commit containerID imageName:tag

- commit changes and saves to a new image referenced by ‘imageName:tag’

docker tag imageName:tag us.gcr.io/projectName/imageName:tag

- tag ‘imageName:tag’ to a container registry repository where it will be available online.

docker push us.gcr.io/projectName/imageName:tag

- push the image tagged to the repository to the repository on the container registry on Google Cloud.

gcloud config set compute/zone us-central1-a

- make sure the current region is set.

gcloud container clusters create cluster-name --num-nodes=1

- creates a GKE kubernetes cluster with a name ‘cluster-name’ and 1 node.

glcoud container clusters get-credentials cluster-name

- allows changes to be made by the current user in the cluster named ‘cluster-name’.

kubectl create deployment name --image=us.gcr.io/projectName/imageName:tag

- creates a deployment called ‘name’ from the image with an online source.

kubectl expose deployment name --type LoadBalancer --port 80 --target-port 80

- expose the deployment on the container port 80 and internet port 80.

kubectl get deployments

- show all deployments and their progress.

kubectl get pods

- get all the pods and their information.

kubectl get service name

- show the service associated with the name 'name' and their external IP addresses for usage.

kubectl get services

- get all services and show with IP addresses.

Prepare another video describing the YML file and showing how it's deployed on GCP.

Video 4 Link:
https://drive.google.com/file/d/1VB4g99POmPbrVuiy5SCF3oTtqpVyfQoJ/view?usp=sharing

What is Kubernetes' pod, service, node, and deployment?

Kubernetes Pod: A group of one or more containers that are running within the same network.
Kubernetes Service: deploys group of pods (cluster) on a unique external IP address.
Kubernetes Node: a virtual (or physical) machine that is doing work for the cluster.
Kubernetes Deployment: instructions given to tell kubernetes how to deploy the pods.

What's meant by replicas?

Replicas are a number of pods that are running the same application.

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

LoadBalancer: using a cloud provider to host a Kubernetes cluster.
ExternalName: mapping a service to a DNS name specified.
NodePort: Setting up a load balancing solution exposed on a static port that can be defined or automatically assigned.
ClusterIP: Can handle communications between the front-end and back-end of an app.