Project Milestone-- IaaS: Virtualization and Containerization

Faazil Shaikh - 100707829

Videos:
https://drive.google.com/drive/folders/1DCQaE1tJRTbW55p-nbhjnvHo6pkrm-wD?usp=sharing

Please download the video or select the highest resolution when viewing the video

Questions

5. What are docker image, container, and registry?

Docker Image: instructions for constructing a Docker container.

Docker Container: The application's code and all of its dependencies are packaged together so that it may be shared to another device.

Docker Registry: The registry is a server side app where images are stored and can be distributed

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

docker build (path) - Builds an image from a dockerfile, option: -t to specify name and tag(optional)

docker run (path)- Creates a container from the following image and runs it, option -d to run in detached mode

docker logs (id)- Outputs the logs of the container

docker images - Lists images

docker ps - Lists containers, option -a to list all

docker version - Shows the installed version of docker

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

To stop: docker container stop (id)

To delete: docker container rm (id), option -f to force remove

8. Prepare a video showing the container(s) created on your machine, displaying their logs, stopping them, and then deleting them. (Note: the JDK version must match that installed in your machine and used to

compile the java code. If you have a problem compiled it can download it from the repository from the path: "/v1/out/production/HelloWorldDocker/Main.class" and use OpenJDK:14 in your Dockerfile).

See file **Question8.mkv** in the drive.

Multi-Container Questions

7. **What's a multi-container Docker application?**

Multi-container means that the application has multiple containers that can communicate with each other.

8. **How these containers are communicated together?**

There needs to be a bridge between the containers via a network connection, they can share information in this way

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

docker image rm (name) - Option -f can be used to force remove

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

docker network create (name) - Create a secure connection for the containers

docker network ls - List all networks

docker network connect (network name) (container) - Connect a container to the network specified

docker pull mysql - Gets the latest image of mysql

11. **Prepare a video showing the created application, run the webapp, stop the application and delete the application containers. (Note: if you have a problem generating the war file, you can download it from the repository from the path: "/v2/target/MyWebApp.war").**

See file **Question11.mkv** in the drive.

GCP Questions

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

See file **Kubernetes11.mkv** in the drive.

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

docker run -d -p 8080:80 nginx:latest - Runs docker on the port 8080, option -d to run in detached mode, -p to publish the port to the host

docker cp (file) (container-id):/destination(- copy file(s) to destination inside the container

docker commit (container-id) (repo)(:tag) - commit new version of the container

docker tag (source)(:tag) (target)(:tag) - creates a tag to the provided target image that references the source.

docker push (registry)(:tag) - pushes image to the registry

gcloud config set (project id) - set config for the project

gcloud config set compute/zone (zone) - sets compute zone

gcloud container clusters create (cluster name) --num-nodes=1 – creates new cluster container, option –num-nodes to specify number of nodes

gcloud container clusters get-credentials (cluster name) - get credentials (permissions) for the cluster name

kubectl create deployment (name) --image=(image)(:tag) - creates a new deployment with specified name and specified image to run

kubectl expose deployment (name) --type LoadBalancer --port 80 --target-port 80 - exposes specified deployment to the ports specified

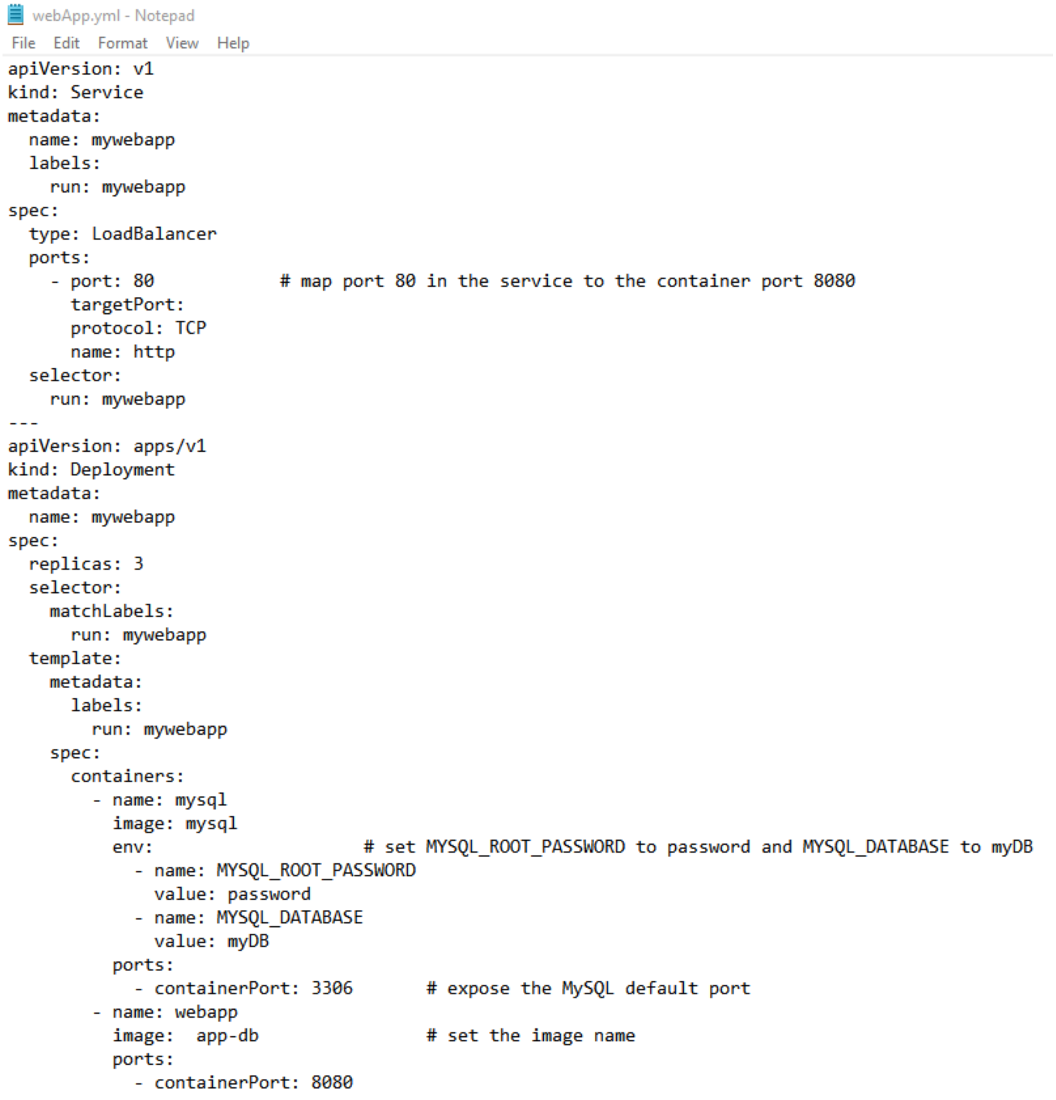
kubectl get pods - list status of the running pods

kubectl get service (name) - list specified service

14.

Prepare a Kubernetes YML (or YAML) file to load the webApp used in steps 6:8 and deploy it using the Kubernetes engine on GCP. The file is a little different than that used by docker-compose.

- **The hostname of all containers is the same and can be accessed by localhost, the address of the MySQL should be changed to localhost and recompiled. (Note: if you have a problem generating the war file, you can download it from the repository from the path "/KGS/target/MyWebApp.war").**
- **Create a new image using the new war file and push it to Google Container Registry.**
- **Follow the comments and fill the missing lines in the "/webApp.yml" file.**
- **Apply the YML file into Kubernetes and run the server (what is the appropriate Cloud shell command?).**



```
apiVersion: v1
kind: Service
metadata:
  name: mywebapp
  labels:
    run: mywebapp
spec:
  type: LoadBalancer
  ports:
    - port: 80            # map port 80 in the service to the container port 8080
      targetPort:
      protocol: TCP
      name: http
  selector:
    run: mywebapp
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mywebapp
spec:
  replicas: 3
  selector:
    matchLabels:
      run: mywebapp
  template:
    metadata:
      labels:
        run: mywebapp
    spec:
      containers:
        - name: mysql
          image: mysql
          env:                    # set MYSQL_ROOT_PASSWORD to password and MYSQL_DATABASE to myDB
            - name: MYSQL_ROOT_PASSWORD
              value: password
            - name: MYSQL_DATABASE
              value: myDB
          ports:
            - containerPort: 3306       # expose the MySQL default port
        - name: webapp
          image:  app-db                # set the image name
          ports:
            - containerPort: 8080
```

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

Pod - A collection of containers that share storage and network resources, as well as all of the specs needed to install them.

Service - A network service that exposes an app operating on a set of pods.

Node - Pods run on nodes, you can have several nodes in a cluster

Deployment - Shows the status of the pods

18. What's meant by replicas?

A replica or a clone is when multiple pod instances are ran to serve as a backup if a pod becomes unavailable

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

loadBalancer - exposes service using the clouds load balancer

nodePort - port for all nodes in the cluster, port must be available

clusterIP - Service is exposed internally

externalName - Map the service to what's in the externalName