Faculty of Engineering & Applied Science

Cloud Computing - 74293

Project Milestone 1:

IaaS: Virtualization and Containerization

Taha Hashmat - 100689792

Google Link for the Recorded Videos:

https://drive.google.com/drive/folders/1aK704cceD-Dd-4y7XUgsP4BjmvQFzX6F?usp=sharing

Q5. What are docker images, container, and registry?

Docker Images: Docker images are considered to be the initial building block of docker. Docker images contain the information needed for a docker container and the code that would be run in that container
Docker Container: Created for the primary purpose of integrating and delivering software in a more precise and fast manner, containers pack up all the code and information needed for an application to run
Docker Registry: The registry allows users to share open source images amongst themselves and for their own use using commands such as push, pull, etc

Q6. List the Docker commands used in the video with a brief description for each command and option.

1. docker build -t hello-world:1.0 . : Builds an image with the name hello-world and tag 1.0.
2. docker images: shows a list of images with name, tag, image id, time created and size
3. docker run hello-world:1.0 : creates a docker container based on the hello-world:1.0 image and terminates it
4. docker ps: shows a list of currently running docker containers
5. docker ps -a: shows a list of all containers, running and terminated
6. docker run -d hello-world:2.0: creates a container in detached mode meaning a container starts and runs in the background
7. docker logs *container id*: check the status of the container

Q7. At the end of the video, there are two running containers, what commands can be used to stop and delete those two containers?

1. Command to stop a container: docker stop *container name*
2. Command to delete a container: docker container rm *container ID*

Q8. What's a multi-container Docker application?

A multi container docker application is an application that requires multiple independent containers that communicate with one another to function, for example a separate container of a MySQL database.

Q9. How are these containers communicated together?

Containers communicate with each other through a network built by docker itself called a bridge. Containers on this network can communicate among themselves.

Q10. What command can be used to stop the Docker application and delete its images?

1. Command to stop a docker app: sudo docker create --name firstkill bitnami/wordpress
2. Command to delete docker images: docker image -rmi *image ID*

Q11. List the new docker commands used in the video with a brief description for each command and option.

1. docker pull mysql - pulls in a mysql image from dockerhub
2. docker run –name app-db -d -e MYSQL_ROOT_PASSWORD=password -e MYSQL_DATABASE=myDB mysql - creating a container for the mysql image
3. docker rm -f - stops containers forcefully
4. docker run –name app -d -p 8080:8080 my-web-app:1.0 - specifying a port on the host machine and the container port that the host port will be bound to
5. docker run –name app -d -p 8081:8080 my-web-app:1.0 - binding container port to an open port i.e. 8081 since 8080 is already busy as with our previous command
6. docker network create app-network - creates a new network alongside 3 other default networks
7. docker network ls - display all present networks
8. docker network connect app-network app-db: connecting db container to the network
9. nvm clean install - building the application file
10. docker run --name app -d -p 8080:8080 network=app-network my-web-app:1.0 - connecting application container to the application network
11. docker compose up -d: creates and starts the containers

Q13. List all used GCP shell commands and their description in your report.

1. docker run -p 8080:80 nginx:latest - exposing port 8080 to the container with nginx image
2. cat index.html - copying all the index.html
3. docker cp index.html 09c7b89bad81:/usr/share/nginx/html/ - path of nginx html in linux
4. docker commit 09c7b89bad81 cad/web:version1 - committed the container and created a new image
5. docker tag cad/web:version1 us.gcr.io/synthetic-nova-340209/cad-site:version1- tagging the docker image with the given repository
6. docker push us.gcr.io/synthetic-nova-340209/cad-site:version1 - pushing the docker image tagged in the last command
7. gcloud config set project synthetic-nova-340209 - setting project id
8. gcloud config set compute/zone us-central1-a - specifies location of a cluster
9. gcloud container clusters create gk-cluster —num-nodes=1 - creates a gk cluster with one node

10. gcloud container clusters get-credentials gk-cluster - deploying the container and gives credential to the gk cluster
11. kubectl create deployment web-server --image=us.gcr.io/alien-drake-340206/cad-site:version1 - deploys web server
12. kubectl expose deployment web-server --type LoadBalancer --port 80 --target-port 8080 - exposing the deployment to the internet
13. kubectl get pods - obtains the running pods
14. kubectl get service web-server - specified server with external IP

Q17. What is Kubernetes' pod, service, node, and deployment?

1. Kubernetes Pod: Pods are the unit of deployment in kubernetes. A pod in kubernetes basically wraps up a bunch of containers that can communicate with each other and deploys it.
2. Kubernetes Service: Pods have their own IP Address associated with them, but since pods are ephemeral, so is their IP Address. In order to address this issue of referencing pods more accurately, they are assigned with their own stable IP Addresses that stay with them regardless of them being dynamically created or destroyed. This method of accessing these kubernetes pods is called a kubernetes service
3. Kubernetes Node: A node can either be a virtual or physical machine which carries the ability of running a certain number of pods in it.
4. Kubernetes Deployment: Replaces pods that failed to run with replicas of pods with no unique identities to keep a streamlined process

Q18. What's meant by replicas?

A specific number of replicas are created of pods in a cluster at any given time to ensure that when a user who is using an application has a pod become suddenly unresponsive or fail, there are replicas present in the cluster to replace them and avoid any complexities

Q19. What are the types of Kubernetes' services? What is the purpose of each?

1. ClusterIP: a default type of service that can only be accessed inside a cluster
2. Nodeport: allows traffic to be sent to the service by opening up ports
3. LoadBalancer: Sends traffic from the internet to the service, onto the pods in the cluster by exposing the service to the internet
4. ExternalName: Returns a CNAME record instead of a cluster IP
5. Headless: A service that returns the specific IP address of different pods in the cluster