Commands Projet Final MasterClass Cloud/Docker/Kubernetes

1- Build your image :

docker build -t react-web-app .

2- Run a container locally in background from your image through the local port 8989 :

docker run -d -p 8989:3000 react-web-app

3- Delete your container :

docker stop [container id]

docker rm [container id]

4- Send your image in your account on the public registry Docker-Hub :

docker login

docker tag react-web-app [docker id]/react-web-app

docker push [docker id]/react-web-app

https://hub.docker.com/r/registrydockerjeremyagr/react-web-app/

5- Create a cluster in the zone ‘europe-west1-b’ & connect.

Performed on Google Cloud’s online platform.

6- Create a Deployment resource of your react-web-app image previously created :

kubectl apply -f ./deployment.yaml

7- Check if it works: print your pods.

kubectl get pods

8- Create a Service resource which will accept requests on the port 8989 which retargets requests on the Deployment previously created on the correct react-web-app running port :

kubectl apply -f ./service.yaml

9- Check if it works, print your service :

kubectl get service

10- Create an Ingress resource which will accept outdoor requests on the port 80 to redirect it on the Service previously created on the port 8989 :

kubectl apply -f ./ingress.yaml

11- Check if it works: print your Ingress :

kubectl get ingress

12- Get the IP address of your Ingress previously created. Try to target this IP using your favorite browser :

http://34.120.64.32/

13- Create an HPA resource to autoscale your Deployment resource like: min=3, max=50, and autoscale when the CPU reach 65% of its capacity :

kubectl apply -f ./hpa.yaml

14- Check if it works: print your HPA :

kubectl get hpa