Technical instructions to deploy Uber solutions on Azure Kubernetes Service

The aim of this document is to describe the steps to follow to successfully build and deploy the different Uber solutions on Azure Kubernetes Service.

Steps:

1-Create an azure kubernetes cluster by following the steps in this link (<https://k21academy.com/microsoft-azure/solution-architect/aks-cluster/>)

2-Open the terminal inside the cluster and clone the following git repo (branch main):

<https://192.168.26.40/cci/intern/pfe2/uss-microservices.git>

3- Build and deploy a docker image for each microservice(customer, gateway,operation,payment,search,view)

If you face any issues in building the image from the azure cluster u can do it locally.

Example:

docker build -t yourDockerhubYousername/imageName -t yourDockerhubYousername/imageName:0.0.1 .

Once the image was built successfully we need to push it to the docker hub:

docker login

docker push yourDockerhubYousername/imageName

4-Update the image source in the deployment yaml file for each microservice with the same image pushed in your docker hub (Step 3 : yourDockerhubYousername/imageName)

5- Run the file deploy-bd.sh under the script folder in odre tp deploy the database resources(Run chmod +x deploy-bd.sh first)

6- Run the file deploy-elk.sh under the script folder in order to deploy elasticsearch and kibana(Run chmod +x deploy-elk.sh first)

**Install Nginx ingress controller**

In order to expose the deployed services externally we must use ingress controller which will intercept the trafic from the load balancer and dispatch it to the sepcific requested services :

**Steps to install nginx ingress**

Run the following commands on your azure cli :

1. NAMESPACE=ingress-basic
2. helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
3. helm repo update

4- helm install ingress-nginx ingress-nginx/ingress-nginx \

--create-namespace \

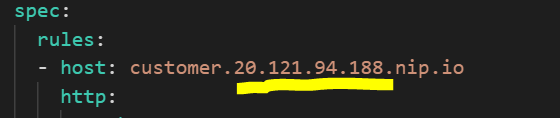
--namespace $NAMESPACE \

--set controller.service.externalTrafficPolicy=Local \

--set controller.service.annotations."service\.beta\.kubernetes\.io/azure-load-balancer-health-probe-request-path"=/healthz

After that run (kubectl get svc – namespace ingress-basic) in order to get the external ip address

Copy that address and replace the existing ones in every ingress yaml file (under ingress folder) like this:



After updating all ingress yaml files with the new external ip address, apply these file by running(kubectl apply –f PATH\_TO\_INGRESS\_FILE )

Now we can test that our services are exposed by copying the host name in your browser

7-Once we have pgadmin deployed and exposed with ingress we can use it to create the databases for all the microservices (except operation which was already created in the deployment of the database image)

8- once we created the databases we must update the datasource url in each config map of these microservices with the same name of the db created in step 7

9- Run the file set-env.sh under the script folder in order to deploy the necessary resources on the clustrer(Run chmod +x set-env.sh first)

Useful links:

<https://techdozo.dev/deploying-a-restful-spring-boot-microservice-on-kubernetes/>

<https://learn.microsoft.com/en-us/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices>

<https://reflectoring.io/spring-boot-elasticsearch/>