**Microsoft OpenHack: Containers (Challenge Based Training)**

Azure Credentials provided by the Openhack Team:

**Username**: hacker21gx@msftopenhack6846ops.onmicrosoft.com

**Password**: kZ7\*EfLg

**Day 1:**

**Challenge 1:**

Create a VM in Azure Virtual Machine , download the RDP (Remote Desktop Protocol) file and run the Virtual Machine, perform the the following activities in the VM.

Build the docker file and then upload the docker image to Azure Container Registry(ACR) using Azure CLI and kubectl

**Docker Commands:**

Docker run mcr.microsoft.com/mssql/server:2017-latest

docker network create NetworkPOI *#Creating docker network*

docker run --name sql1 --network <networkname> -h sql1 -e 'ACCEPT\_EULA=Y' -e 'MSSQL\_SA\_PASSWORD=Passw0rd' -p 1433:1433 -d mcr.microsoft.com/mssql/server:2017-latest

**Building the Dockerfile for all the 5 projects**

(Go to that specific folder, open cmd and run this command):

docker build -t "registryeqf2771.azurecr.io/tripinsights/poi:1.0" .  
docker build -t "registryeqf2771.azurecr.io/tripinsights/trips:1.0" .  
docker build -t "registryeqf2771.azurecr.io/tripinsights/tripviewer:1.0" .  
docker build -t "registryeqf2771.azurecr.io/tripinsights/user-java:1.0" .  
docker build -t "registryeqf2771.azurecr.io/tripinsights/userprofile:1.0" .

**Now go to Azuze CLI to access Microsoft Azure from command line:**

az login

**Login to Azure Container Registry:**

az acr login --name registryeqf2771

**Push Docker Image to ACR (Azure Container Registry):**  
docker push "registryeqf2771.azurecr.io/tripinsights/poi:1.0”  
docker push "registryeqf2771.azurecr.io/tripinsights/trips:1.0”  
docker push "registryeqf2771.azurecr.io/tripinsights/tripviewer:1.0"  
docker push "registryeqf2771.azurecr.io/tripinsights/user-java:1.0”

docker push "registryeqf2771.azurecr.io/tripinsights/userprofile:1.0"

**Day 2:**

**Challenge 2:**

1.Create an Azure Kubernetes Cluster(AKS) in Azure portal

2.Connect to the AKS Cluster and resource group from azure CLI, then attached the AKS cluster to ACR, then create a yaml file of deployment type to deploy container to AKS cluster. Create different pods and deploy the containers to those pods.

3.Create the yaml files of type service to connect the container tripviewer to services trips and userprofile.

az login *#login to azure cli*

az aks get-credentials --resource-group Team2\_Cluster\_group --name Team2\_Cluster *#connect to cluster Team2\_Cluster and resourse group Team2\_Cluster\_group*

az aks update -n Team2\_Cluster -g Team2\_Cluster\_group --attach-acr registryeqf2771 *#connecting AKS cluster to ACR*

kubectl apply -f poi.yaml *#Deploy the container to AKS using yaml file*

kubectl apply -f service-tripviewer.yaml *#Create a service using yaml file*

tripviewer.yaml file

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: tripviewer-skl

labels:

app: web

spec:

replicas: 1

selector:

matchLabels:

app: web

template:

metadata:

labels:

app: web

spec:

containers:

- name: tripviewer

image: registryeqf2771.azurecr.io/tripinsights/tripviewer:1.0

ports:

- containerPort: 82

env:

- name: userprofile

value: "http://userprofile"

- name: trips

value: "http://trips"

- name: SQL\_USER

value: "sqladmineQf2771"

- name: SQL\_DBNAME

value: "mydrivingDB"

- name: SQL\_PASSWORD

value: "Demo@pass123"

- name: SQL\_SERVER

value: "sqlservereqf2771"

service-tripviewer.yaml

apiVersion: v1

kind: Service

metadata:

name: service-tripviewer

spec:

selector:

app: tripviewer

ports:

- port: 82

targetPort: 80

type: LoadBalancer

**Day 3:**

**Challenge 3:**

1.Create a subnet in the vnet which is already created in ‘teamresource’ resourse group and deploy the containers in the subnet. Connect the newly created cluster to the subnet.

2.Create namespaces ‘web’ and ‘api’.Add tripviewer to web and trips, user-java, poi and userprofile to api, create connectivity between web and api and check the endpoint url and see if the application is correctly deployed.

3.Connect newly created cluster to ACR.

4.Create users in Azure Active Directory (AAD) and assign roles to it using RBAC (Role Based Access Control)

az network vnet subnet list -g teamresource --vnet-name vnet *#list all the subnets under vnet*

az aks create -g ResourceGrpDay3-Team2 -n clusterDay3 --vnet-subnet-id "/subscriptions/28fe2197-362d-463e-bc5b-b731bdae4ba4/resourceGroups/teamResources/providers/Microsoft.Network/virtualNetworks/vnet/subnets/aksnteam2" --generate-ssh-key

*# attach the clusterDay3 to subnet*

kubectl create namespace api

kubectl create namespace web *#create namespaces*

az aks update -n clusterDay3 -g ResourceGrpDay3-Team2 --attach-acr registryeqf2771 *#pushing the container to ACR*

**Challenge 4:**

1.Use an external key vault to store and access secrets inside your cluster, and ensured that access does not require a secret stored in the cluster

2.Ensured that all links on the Trip Viewer site are reachable

az aks enable-addons --addons NewKeyVault --name clusterDay3 --resource-group ResourceGrpDay3-Team2

az keyvault create -n NewKeyVault -g ResourceGrpDay3-Team2 -l eastasia