1. #az login #az cli
2. az group create --name BA\_Kuber --location eastus
3. az ad sp create-for-rbac –skip-assignment --name BA\_kubernetes-cluster\_service-principal

#Created a Kubernetes cluster, resource group, and service principal

1. az aks create --name BA\_cluster --node-count 2 --enable-addons monitoring --resource-group BA\_Kuber --vm-set-type VirtualMachineScaleSets --load-balancer-sku standard --enable-cluster-autoscaler --min-count 1 --max-count 2 --generate-ssh-keys --service-principal a1ed48d2-c2cb-46b5-a11d-aa24ff376bed --client-secret BPV8Q~~NfS29L\_-5AuKZb0wS~RfWIabo~OziAbgF
2. az aks get-credentials --resource-group BA\_Kuber --name BA\_cluster

# gives Kubernetes information

1. kubectl get nodes #gets current available nodes
2. kubectl get pods

#Deploying a Hello World API in AKS

1. kubectl create deployment hello-world-rest-api --image=BA\_cluster/hello-world-rest-api:0.0.1.RELEASE # deployment.apps/hello-world-rest-api created

kubectl expose deployment hello-world-rest-api --type=LoadBalancer --port=8080

# service/hello-world-rest-api exposed   
# NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

hello-world-rest-api LoadBalancer 10.0.182.139 52.226.44.0 8080:30718/TCP 51s

kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 28m

#Deploying a Web App to AKS

1. kubectl create deployment todowebapp-h2 --image=BA\_cluster/todo-web-application-h2:0.0.1-SNAPSHOT
2. kubectl expose deployment todowebapp-h2 --type=LoadBalancer --port=8080
3. kubectl get service

# NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

hello-world-rest-api LoadBalancer 10.0.182.139 52.226.44.0 8080:30718/TCP 23m

kubernetes ClusterIP 10.0.0.1 <none> 443/TCP 51m

todowebapp-h2 LoadBalancer 10.0.150.137 4.157.112.151 8080:30874/TCP 48s

#Deploying a Web App using MySQL to AKS

1. Git clone <https://github.com/in28minutes/kubernetes-crash-course.git> #clones repo
2. cd kubernetes-crash-course
3. cd 03-todo-web-application-mysql/backup/02-final-backup-at-end-of-course
4. ls

#config-map.yaml mysql-database-data-volume-persistentvolumeclaim.yaml mysql-service.yaml todo-web-application-deployment.yaml

mysql-database-data-volume-persistentvolumeclaim-aws.yaml mysql-deployment.yaml secret.yaml todo-web-application-service.yaml

PS /home/babatunde/kubernetes-crash-course/03-todo-web-application-mysql/backup/02-final-backup-at-end-of-course>

1. kubectl apply -f mysql-database-data-volume-persistentvolumeclaim.yaml,mysql-deployment.yaml,mysql-service.yaml

#Takeaways

quota for clusters is 4 not 8

authentication after creating the cluster (had to edit the code individually)

had to change clusters to 2 instead of 4 to get it to work.

Double checking spaces

Login issues with external IP’s (pages not loading)

Yaml file version not the same as Kubernetes version

Because the versions were different and didn’t have the original file I could not update it.