.

Lesson 08 Demo 06

Deploying an Application with Load Balancer on AKS

Objective: To deploy an application with a load balancer on Azure Kubernetes Service (AKS)

Tools required: Azure management tools

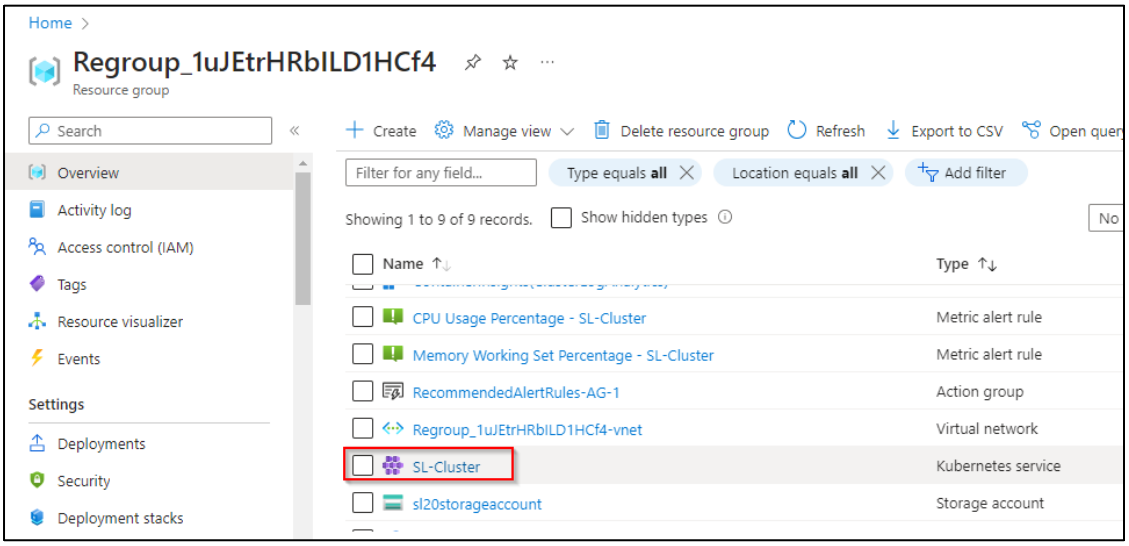
Prerequisites: Lesson 08: Demo 01, 02, and 03

Steps to be followed:

1. Deploy the app on AKS with a load balancer

Step 1: Deploy the app on AKS with a load balancer

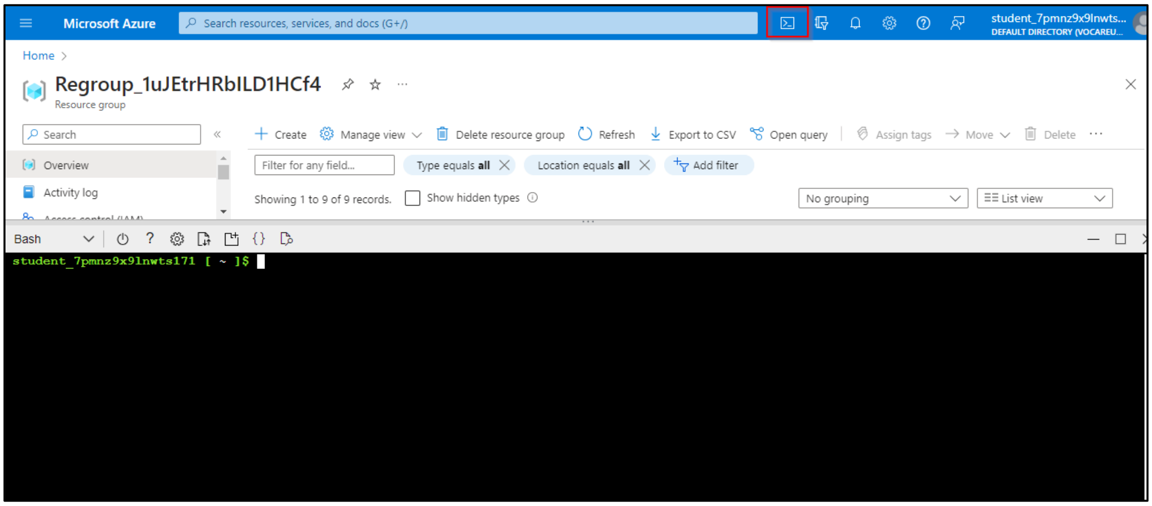
1.1 Navigate to the **SL-Cluster**



Name	Type
CPU Usage Percentage - SL-Cluster	Metric alert rule
Memory Working Set Percentage - SL-Cluster	Metric alert rule
RecommendedAlertRules-AG-1	Action group
Regroup_1uJEtrHRbILD1HCf4-vnet	Virtual network
SL-Cluster	Kubernetes service
sl20storageaccount	Storage account

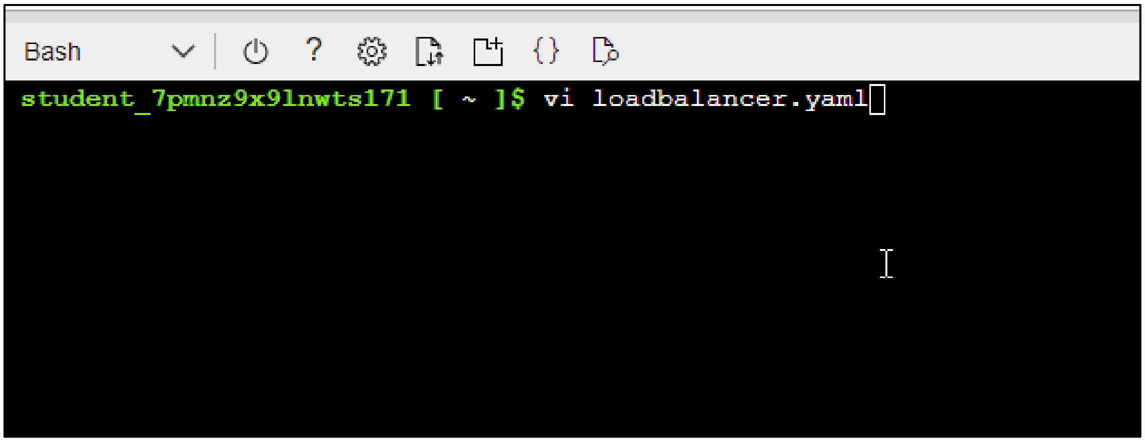
Note: For detailed steps on creating a Resource group, refer to Lesson 08 Demo 01.

1.2 Launch the Cloud Shell by clicking on its icon



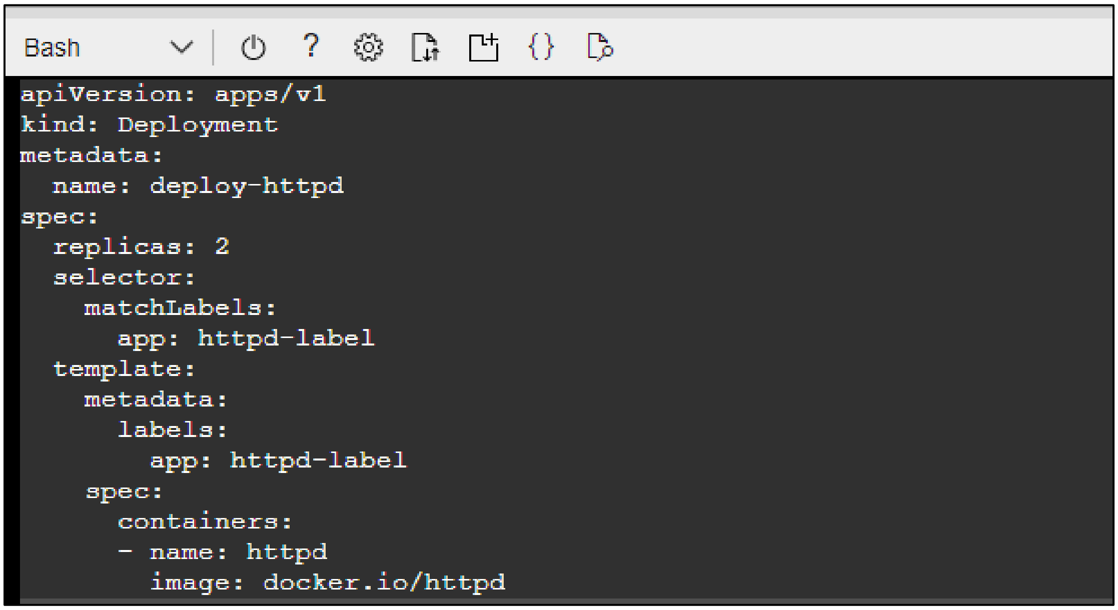
```
student_7pmnz9x9lnwts171 [ ~ ]$
```

1.3 Create the **loadbalancer.yaml** file to define the pod configuration. Enter the following command and then input the code:
vi loadbalancer.yaml



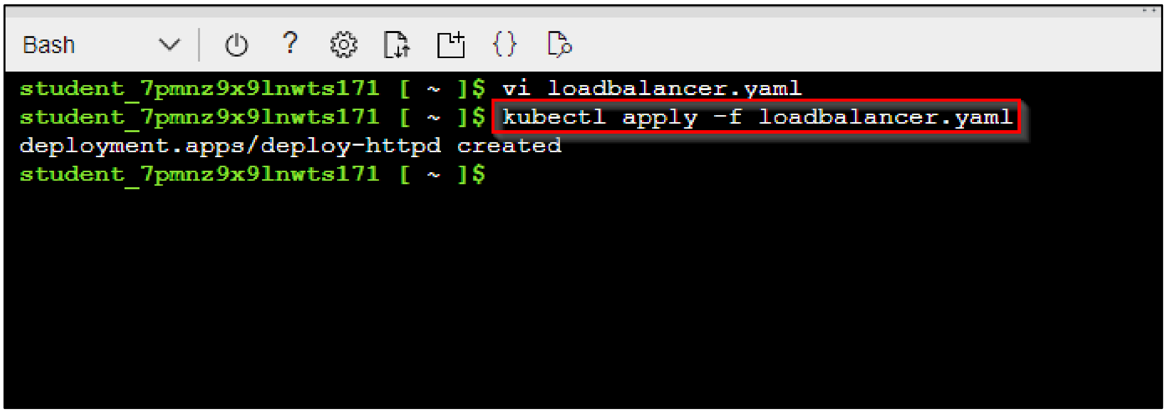
```
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer.yaml
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-httpd
spec:
  replicas: 2
  selector:
    matchLabels:
      app: httpd-label
  template:
    metadata:
      labels:
        app: httpd-label
    spec:
      containers:
      - name: httpd
        image: docker.io/httpd
```



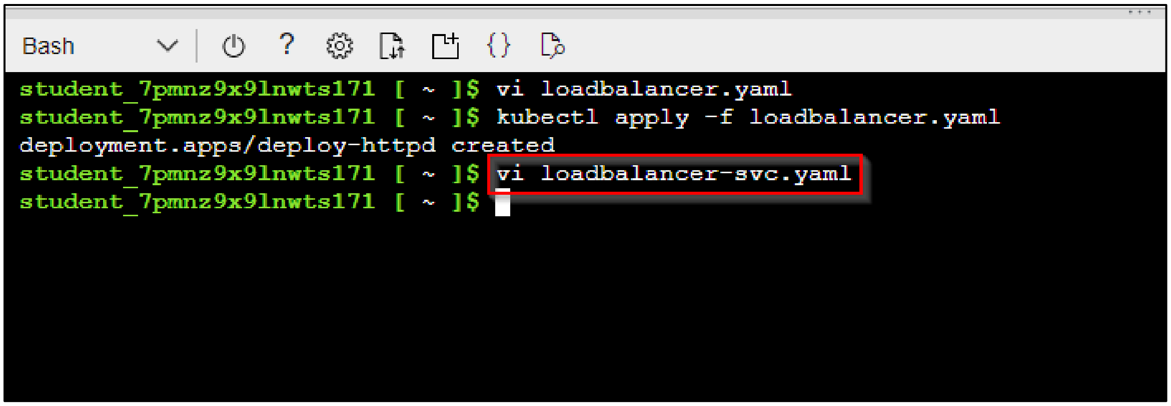
```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: deploy-httpd
spec:
  replicas: 2
  selector:
    matchLabels:
      app: httpd-label
  template:
    metadata:
      labels:
        app: httpd-label
    spec:
      containers:
      - name: httpd
        image: docker.io/httpd
```

1.4 Deploy the pod by executing the following command:
kubectl apply -f loadbalancer.yaml



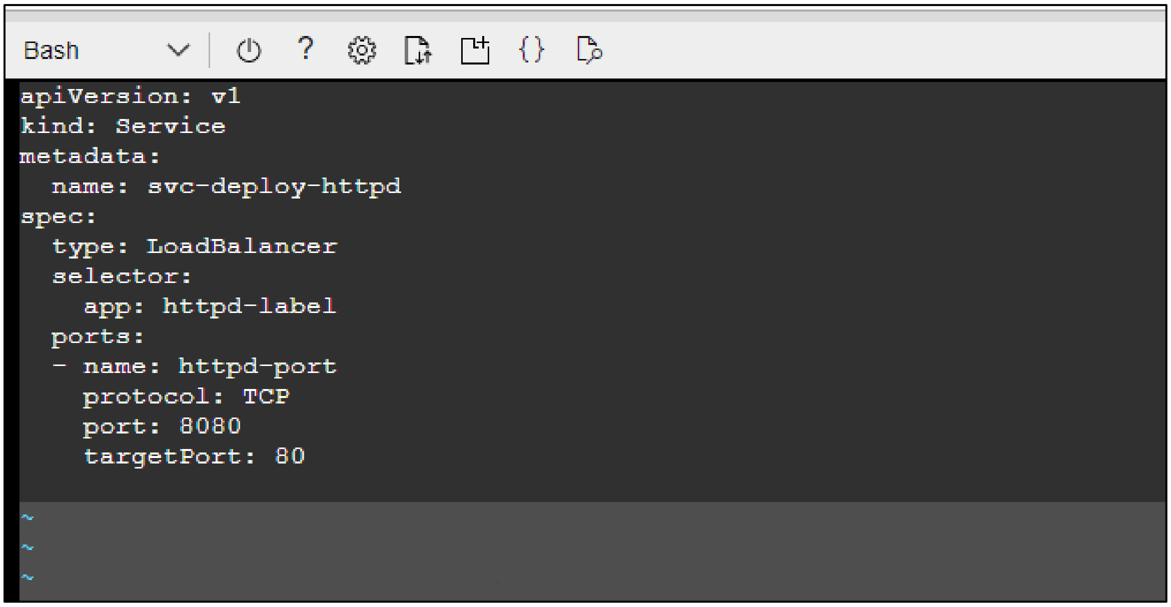
```
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer.yaml
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl apply -f loadbalancer.yaml
deployment.apps/deploy-httpd created
student_7pmnz9x9lnwts171 [ ~ ]$
```

1.5 Create a load balancer service using the following command and code:
vi loadbalancer-svc.yaml



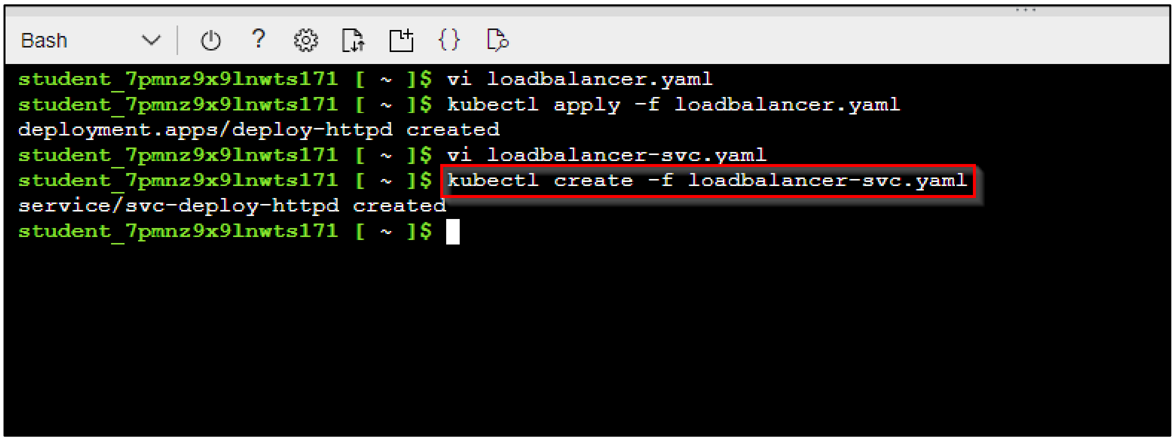
```
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer.yaml
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl apply -f loadbalancer.yaml
deployment.apps/deploy-httpd created
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer-svc.yaml
student_7pmnz9x9lnwts171 [ ~ ]$
```

```
apiVersion: v1
kind: Service
metadata:
  name: svc-deploy-httpd
spec:
  type: LoadBalancer
  selector:
    app: httpd-label
  ports:
  - name: httpd-port
    protocol: TCP
    port: 8080
    targetPort: 80
```



```
apiVersion: v1
kind: Service
metadata:
  name: svc-deploy-httpd
spec:
  type: LoadBalancer
  selector:
    app: httpd-label
  ports:
  - name: httpd-port
    protocol: TCP
    port: 8080
    targetPort: 80
~
~
~
```

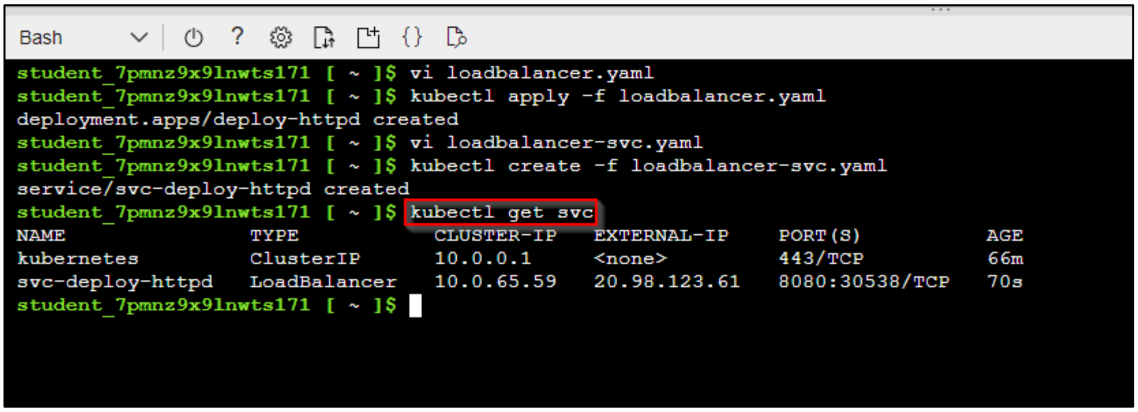
1.6 Create the service by executing the following command::
kubectl create -f loadbalancer-svc.yaml



```
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer.yaml
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl apply -f loadbalancer.yaml
deployment.apps/deploy-httpd created
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer-svc.yaml
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl create -f loadbalancer-svc.yaml
service/svc-deploy-httpd created
student_7pmnz9x9lnwts171 [ ~ ]$
```

1.7 Validate the created service with the following command:

kubectl get svc



```
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer.yaml
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl apply -f loadbalancer.yaml
deployment.apps/deploy-httpd created
student_7pmnz9x9lnwts171 [ ~ ]$ vi loadbalancer-svc.yaml
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl create -f loadbalancer-svc.yaml
service/svc-deploy-httpd created
student_7pmnz9x9lnwts171 [ ~ ]$ kubectl get svc
NAME               TYPE           CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
kubernetes         ClusterIP      10.0.0.1      <none>         443/TCP          66m
svc-deploy-httpd   LoadBalancer   10.0.65.59    20.98.123.61   8080:30538/TCP   70s
student_7pmnz9x9lnwts171 [ ~ ]$ 
```

By following these steps, you have successfully deployed an application and connected it with a load balancer on AKS.