Lesson 08 Demo 07

Creating a Persistent Volume

Objective: To create a persistent volume using Azure disks in Azure Kubernetes Service (AKS)

Tools required: Azure management tool

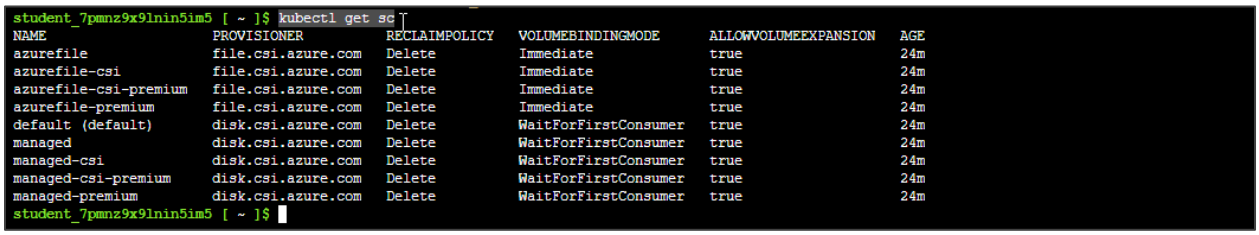
Prerequisites: Configure an AKS cluster and a storage account (Refer to Lesson 08, Demo 01 and Demo 02)

Steps to be followed:

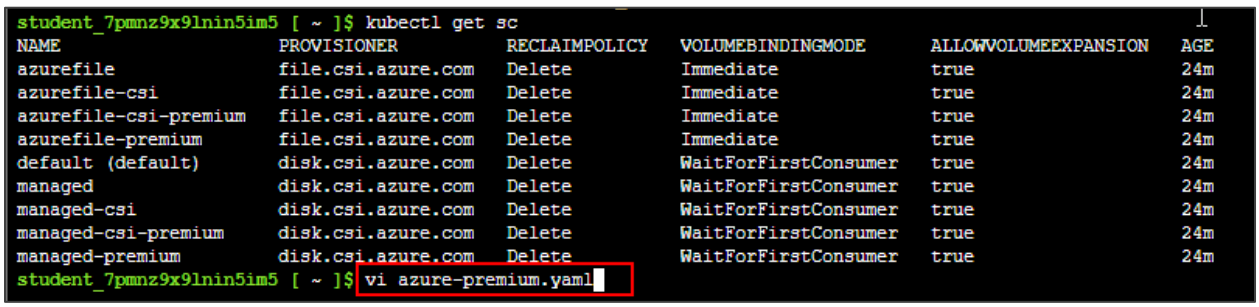
1. Create a persistent volume claim using AKS bash

Step 1: Create a persistent volume claim using AKS bash

1.1 Execute the following command in AKS bash to check the pre-existing storage classes:
kubectl get sc

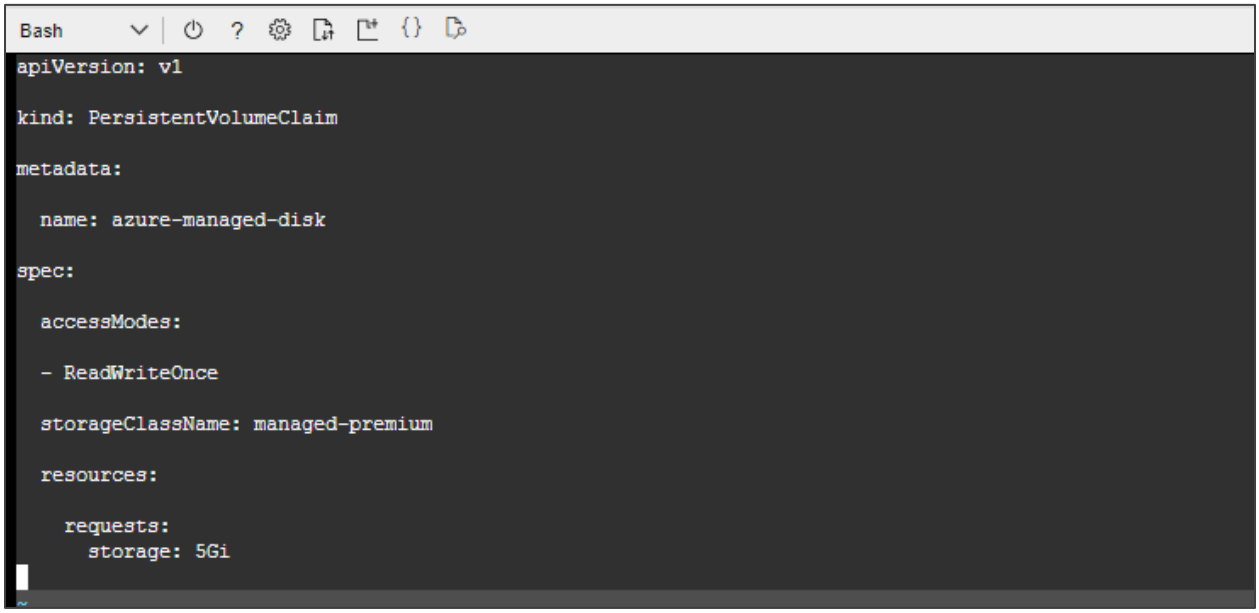


1.2 Use the following command to create a file named **azure-premium.yaml**:
vi azure-premium.yaml



1.3 Enter the following code within the **azure-premium.yaml** file:

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: azure-managed-disk
spec:
 accessModes:
 - ReadWriteOnce
 storageClassName: managed-premium
 resources:
  requests:
   storage: 5Gi
```



1.4 View the contents of the **azure-premium.yaml** file using the following command:
cat azure-premium.yaml

```
student_7pmnz9x9lnin5im5 [ ~ ]$ cat azure-premium.yaml
apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: azure-managed-disk

spec:

  accessModes:

  - ReadWriteOnce

  storageClassName: managed-premium

  resources:

    requests:
      storage: 5Gi

student_7pmnz9x9lnin5im5 [ ~ ]$
```

1.5 Create the **persistentvolumeclaim** using the following command:
kubectl apply -f azure-premium.yaml

```
student_7pmnz9x9lnin5im5 [ ~ ]$ kubectl apply -f azure-premium.yaml
persistentvolumeclaim/azure-managed-disk created
student_7pmnz9x9lnin5im5 [ ~ ]$
```

1.6 Create a file named **azure-pvc-disk.yaml** using the following command:
vi azure-pvc-disk.yaml

```
student_7pmnz9x9lnin5im5 [ ~ ]$ vi azure-pvc-disk.yaml
```

This file will specify the pod that will be used to access the Azure disk.

1.7 Enter the following code within the **azure-pvc-disk.yaml** file:

```
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    resources:

      requests:
        cpu: 100m
        memory: 128Mi
      limits:
        cpu: 250m
        memory: 256Mi
    volumeMounts:
    - mountPath: "/mnt/azure"
      name: volume
  volumes:
    - name: volume
      persistentVolumeClaim:
        claimName: azure-managed-disk
```

```
Bash
kind: Pod
apiVersion: v1
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
      limits:
        cpu: 250m
        memory: 256Mi
    volumeMounts:
    - mountPath: "/mnt/azure"
      name: volume
  volumes:
    - name: volume
      persistentVolumeClaim:
        claimName: azure-managed-disk
~
```

1.8 Create the pod using the following command:

kubectl apply -f azure-pvc-disk.yaml

```
student_7pmnz9x9lnin5im5 [ ~ ]$ kubectl apply -f azure-pvc-disk.yaml
pod/mypod created
student_7pmnz9x9lnin5im5 [ ~ ]$
```

1.9 Verify the pod state using the following commands:

kubectl get pvc

kubectl get pods

```
student_7pmnz9x9lnin5im5 [ ~ ]$ kubectl get pvc
NAME                 STATUS   VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS       AGE
azure-managed-disk   Bound    pvc-bf49ce23-1b7c-4665-bfcf-67e777f3df58   5Gi        RWO            managed-premium    5m45s
student_7pmnz9x9lnin5im5 [ ~ ]$ kubectl get pods
NAME    READY   STATUS    RESTARTS   AGE
mypod   1/1     Running   0          86s
student_7pmnz9x9lnin5im5 [ ~ ]$
```

1.10 Execute the following command to describe the pod attributes:

kubectl describe pod mypod

```
student_7pmnz9x9lnin5im5 [ ~ ]$ kubectl describe pod mypod
Name:             mypod
Namespace:        default
Priority:         0
Service Account:  default
Node:             aks-agentpool-17423166-vmss000000/10.224.0.5
Start Time:       Thu, 19 Oct 2023 21:31:54 +0000
Labels:           <none>
Annotations:      cni.projectcalico.org/containerID: 613ad92fbe2fb27027aaa1642ff3f574e5ad8bb3cc9d2b78281f2e801427129e
                  cni.projectcalico.org/podIP: 10.244.0.6/32
                  cni.projectcalico.org/podIPs: 10.244.0.6/32
Status:           Running
IP:               10.244.0.6
IPs:
  IP:  10.244.0.6
Containers:
  mypod:
    Container ID:   containerd://885c6ce05eadfdd564998f9ad862c0c40bf50997cc9fe45b647e47c60230d4f4
    Image:          mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine
    Image ID:       mcr.microsoft.com/oss/nginx/nginx@sha256:f84780a5ad654515bcd9ba2f35e20935e1246799f198683dd2c4f74d19ae9e5e
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Thu, 19 Oct 2023 21:32:09 +0000
    Ready:          True
    Requests:
      cpu:        100m
      memory:     128Mi
    Environment:  <none>
    Mounts:
      /mnt/azure from volume (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-8mtql (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  volume:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  azure-managed-disk
    ReadOnly:   false
  kube-api-access-8mtql:
    Type:                    Projected (a volume that contains injected data from multiple sources)
    TokenExpirationSeconds:  3607
    ConfigMapName:           kube-root-ca.crt
    ConfigMapOptional:       <nil>
    DownwardAPI:             true
QoS Class:                   Burstable
Node-Selectors:              <none>
Tolerations:                 node.kubernetes.io/memory-pressure:NoSchedule op=Exists
                             node.kubernetes.io/not-ready:NoExecute op=Exists for 300s
                             node.kubernetes.io/unreachable:NoExecute op=Exists for 300s
Events:
  Type    Reason                  Age    From                  Message
  ----    ------                  ----   ----                  -------
  Normal  Scheduled               4m12s  default-scheduler     Successfully assigned default/mypod to aks-agentpool-17423166-vmss000000
  Normal  SuccessfulAttachVolume  4m1s   attachdetach-controller  AttachVolume.Attach succeeded for volume "pvc-bf49ce23-1b7c-4665-bfcf-67e777f3df58"
  Normal  Pulling                 3m59s  kubelet               Pulling image "mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine"
```

By following these steps, you have successfully created a persistent volume with Azure disks attached to the Kubernetes pod in Azure Kubernetes Service (AKS).