Lesson 8 Lesson-End Project

**Deploy MySQL Using Azure Dynamic Storage Class**

**Project agenda:** To deploy MySQL Using Azure Dynamic Storage Class

**Description:** You have been given a project to deploy MySQL Using Azure Dynamic Storage Class

**Tools required:** kubeadm, kubectl, kubelet, and etcd

**Prerequisites:** A Kubernetes cluster must be set up (follow steps of Lesson 2 Demo 1)

**Expected deliverables:** Add a screenshot showing that MySQL is up and running and that there aren't any issues in Pod logs

**Steps to be followed:**

1. Creating an AKS cluster
2. Create an Azure disk for dynamic persistent volume provisioning
3. Deploy MySQL Pods with Azure disk as PV

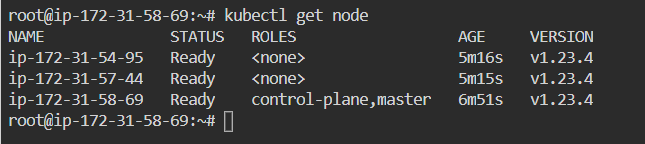
**Step** **1**: **Creating an AKS cluster**

**Note:** Refer to Demos 1 and 3 of Lesson 8

**Step 2: Create an Azure disk for dynamic persistent volume provisioning**

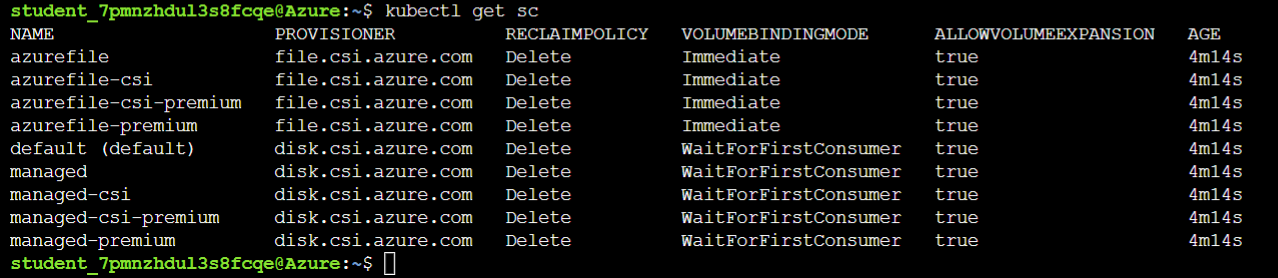
2.1 Validate connectivity between Kubernetes master and worker nodes using the below command:

***kubectl get node***



*2.2*Check for the Storage Classes provided by Azure Kubernetes service using the below command:

**kubectl get sc**



2.2 Create an Azure disk PV manifest file with **azure-pv.yaml** using the content given below:

**apiVersion: v1**

**kind: PersistentVolumeClaim**

**metadata:**

**name: mysql-managed-disk**

**spec:**

**accessModes:**

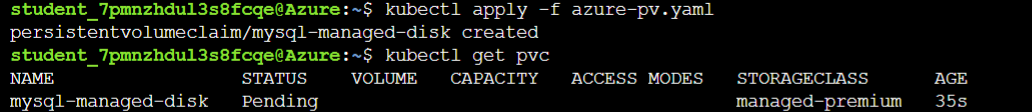
**- ReadWriteOnce**

**storageClassName: managed-premium**

**resources:**

**requests:**

**storage: 5Gi**

****

**Step 3: Deploy MySQL Pods with Azure disk as PV.**

3.1 Create a MySQL deployment manifest file with Azure disk using the content given below:

**apiVersion: apps/v1**

**kind: Deployment**

**metadata:**

**name: mysql**

**labels:**

**app: mysql**

**spec:**

**selector:**

**matchLabels:**

**app: mysql**

**product: mysql**

**strategy:**

**type: Recreate**

**template:**

**metadata:**

**labels:**

**app: mysql**

**product: mysql**

**spec:**

**containers:**

**- image: mysql**

**name: mysql-container**

**env:**

**- name: MYSQL\_DATABASE**

**value: wordpress**

**- name: MYSQL\_RANDOM\_ROOT\_PASSWORD**

**value: "yes"**

**ports:**

**- containerPort: 3306**

**name: mysql**

**volumeMounts:**

**- name: mysql-storage**

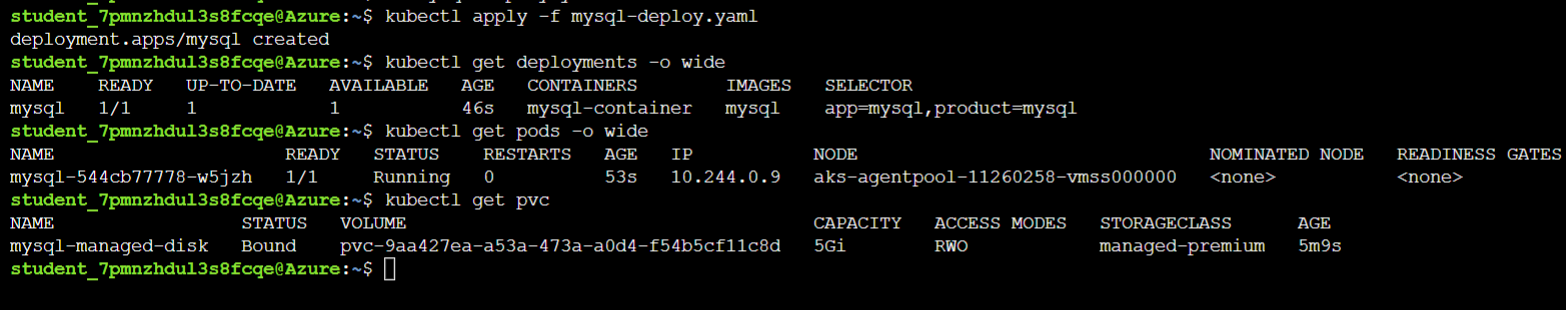
**mountPath: /var/lib/mysql**

**volumes:**

**- name: mysql-storage**

**persistentVolumeClaim:**

**claimName: mysql-managed-disk**

****

**3.2** Once the Deployment is hosted on Kubernetes cluster, validate it to ensure that MySQL is up and running and that there aren't any issues in Pod logs.

**kubectl logs <pod\_name>**

