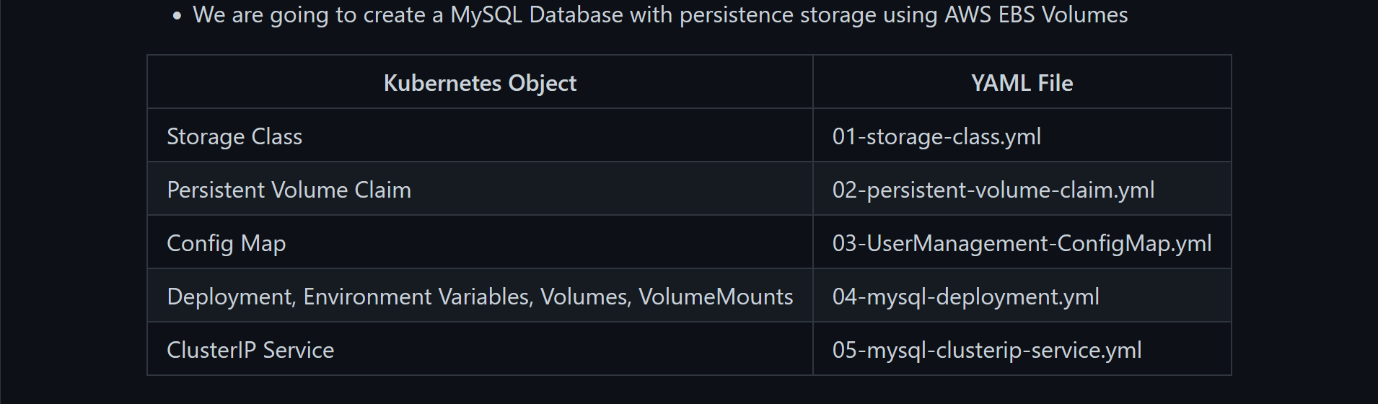
**3. Create Kubernetes Manifests for Storage Class, PVC and ConfigMap**

--- **Reference** - <https://github.com/stacksimplify/aws-eks-kubernetes-masterclass/tree/master/04-EKS-Storage-with-EBS-ElasticBlockStore>

--- **note** - in this lecture we are going to create mysql related kubernetes manifests. Which are required for us and we are going to build them using live template. We will deploy the template and see them how mysql db come up with persistent storage (EBS).

**Introduction**

--- note – we are going to create 5 manifests here.



**Create following Kubernetes manifests**

**Create Storage Class manifest**

--- Reference - <https://kubernetes.io/docs/concepts/storage/storage-classes/#volume-binding-mode>

--- **Important Note**: WaitForFirstConsumer mode will delay the volume binding and provisioning of a PersistentVolume until a Pod using the PersistentVolumeClaim is created.

--- **01-storage-class.yml**

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

  name: ebs-sc

provisioner: ebs.csi.aws.com # if it is azure or gcp then the provisioner is different

volumeBindingMode: WaitForFirstConsumer # WaitForFirstConsumer mode will delay the volume binding and provisioning of a PersistentVolume until a Pod using the PersistentVolumeClaim is created

--- **02-persistent-volume-claim.yml**

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

  name: ebs-mysql-pv-claim

spec:

  accessModes:

    - ReadWriteOnce

  storageClassName: ebs-sc # the storage class name we mentioned above in the metadata. Please these name should be matched.

  resources:

    requests:

      storage: 4Gi

--- **note** – we are going to create a config map manifest, which will be mounted on mysql deployment. Whenever the mysql pod got created, it is also needs to create one small schema named user mgmt schema so this we will use for user management microservice deployment. Whenever the user management microservice got created, it expects user mgmt schema from msql db. For this purpose, we introduced config map concept here.

--- **note** - whatever the configuration you want to sent to your pods then you defined it as a config map and mount it in your container.

**Create ConfigMap manifest**

--- **note** - We are going to create a usermgmt database schema during the mysql pod creation time which we will leverage when we deploy User Management Microservice.

--- **03-UserManagement-ConfigMap.yml**

apiVersion: v1

kind: ConfigMap

metadata:

  name: usermanagement-dbcreation-script

data:

  mysql\_usermgmt.sql: |-

    DROP DATABASE IF EXISTS usermgmt;

    CREATE DATABASE usermgmt;

**Create Persistent Volume Claims manifest**

**# Create Storage Class & PVC**

--- **kubectl apply -f kube-manifests/** - creating all manifests at once. Here we created 3 things.

**# List Storage Classes**

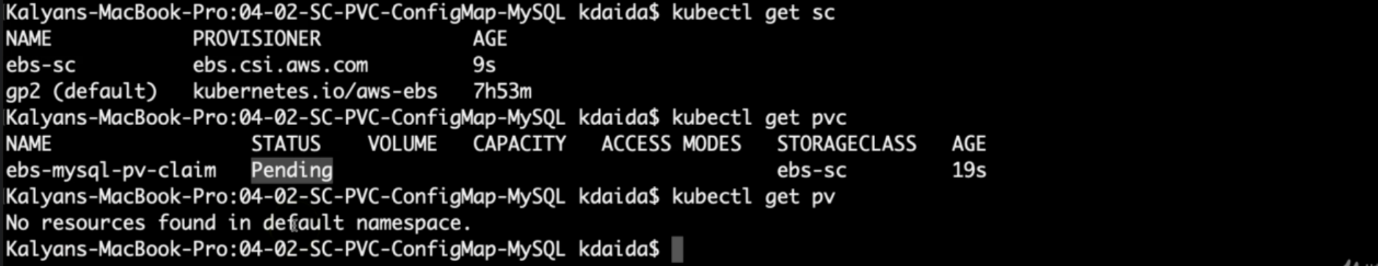
--- **kubectl get sc**

**# List PVC**

--- **kubectl get pvc**

**# List PV**

--- **kubectl get pv**



--- **note** - the ebs-mysql-pv-claim is pending. It is waiting some one to claim. Mysql deployment and cluster ip service for mysql is not created.