# Persistent Volume on AKS

## Sources

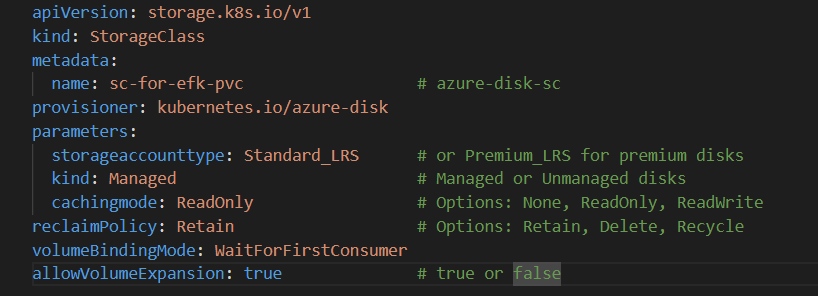
* AWS - <https://github.com/devopsproin/certified-kubernetes-administrator/tree/main/PV-PVC>
* <https://youtu.be/6GkEFqdjdRM?si=RYjAzcfV_KcLA9Sa>
* Azure - <https://github.com/schoudhary22/AKS-AzureDisk>
* <https://www.youtube.com/watch?v=FCBoHZm8fZ4&t=690s>

## Steps

* When using storage class then PV is not required.

1. Create Storage Class. Not required in create in any namespace as it is cluster wide resource.
2. Create PVC in a namespace where deployment is located.
3. Create Deployment with volume details.

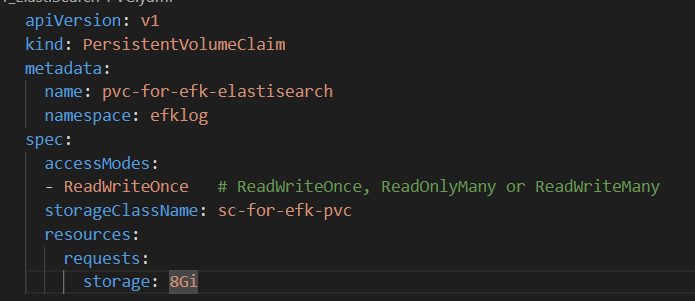
## StorageClass.yaml



|  |
| --- |
| apiVersion: storage.k8s.io/v1  kind: StorageClass  metadata:    name: sc-for-efk-pvc                  # azure-disk-sc  provisioner: kubernetes.io/azure-disk  parameters:    storageaccounttype: Standard\_LRS      # or Premium\_LRS for premium disks    kind: Managed                         # Managed or Unmanaged disks    cachingmode: ReadOnly                 # Options: None, ReadOnly, ReadWrite  reclaimPolicy: Retain                   # Options: Retain, Delete, Recycle  volumeBindingMode: WaitForFirstConsumer  allowVolumeExpansion: true              # true or false |

Kubectl apply -f StorageClass.yaml

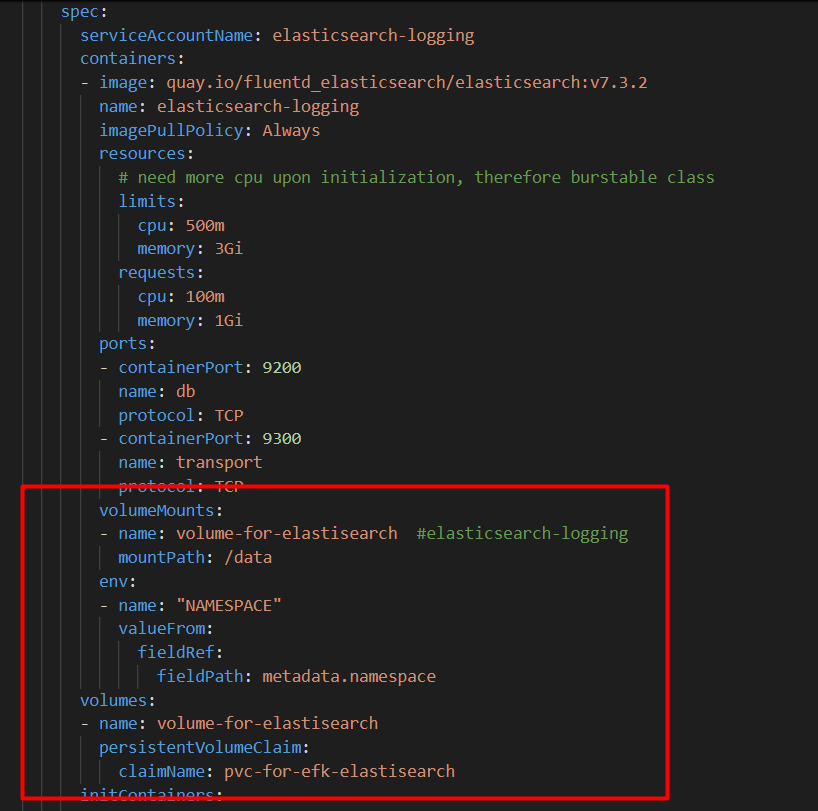
## Pvc.yaml



|  |
| --- |
| apiVersion: v1  kind: PersistentVolumeClaim  metadata:    name: pvc-for-efk-elastisearch    namespace: efklog  spec:    accessModes:    - ReadWriteOnce   # ReadWriteOnce, ReadOnlyMany or ReadWriteMany    storageClassName: sc-for-efk-pvc    resources:      requests:        storage: 8Gi |

Kubectl apply -f pvc.yaml -n namespace

## Deployment / statefulset . yaml



|  |
| --- |
| volumeMounts:          - name: volume-for-elastisearch  #elasticsearch-logging            mountPath: /data          env:          - name: "NAMESPACE"            valueFrom:              fieldRef:                fieldPath: metadata.namespace        volumes:        - name: volume-for-elastisearch          persistentVolumeClaim:            claimName: pvc-for-efk-elastisearch |

Kubectl apply -f deployment.yaml -n namespace