# Storage

kubectl explain pod.spec.volumes

kubectl explain pod.spec.volumes.configMap

kubectl explain pod.spec.containers.volumeMounts  
alias k=kubectl

## 

## Non-Persistent/Ephemeral Volume

### emptyDir

apiVersion: v1

kind: Pod

metadata:

name: sharevol

spec:

containers:

- name: c1

image: centos:7

command:

- "bin/bash"

- "-c"

- "sleep 10000"

volumeMounts:

- name: xchange

mountPath: "/tmp/xchange"

- name: c2

image: centos:7

command:

- "bin/bash"

- "-c"

- "sleep 10000"

volumeMounts:

- name: xchange

mountPath: "/tmp/data"

volumes:

- name: xchange

emptyDir: {}

k apply -f pod-vol-emptyDir.yaml

k get pod sharevol

k exec sharevol -c c1 -- ls /tmp

k exec sharevol -c c2 -- ls /tmp

k exec sharevol -c c2 -- touch /tmp/file1

#Delete and recreate the pod to check data

k exec sharevol -c c2 -- ls /tmp

#Data will not be there

### 

### Host Directory

apiVersion: v1

kind: Pod

metadata:

name: webapp

spec:

containers:

- name: event-simulator

image: nginx

volumeMounts:

- name: log-volume

mountPath: /log

volumes:

- name: log-volume

hostPath:

# directory location on host

path: /var/log

# this field is optional

type: Directory

k apply -f pod-vol-dir.yaml

k get pod webapp

k exec webapp -- ls /log

k exec webapp -- touch /log/k8s.txt

k get pods -o wide | grep webapp

#With the worker node identified, Get into that worker node

ls /var/log/

#check for k8s.txt file and add some text

sudo vi /var/log/k8s.txt

k exec webapp -- cat /log/k8s.txt

#file content will be same

## Volume Config

### ConfigMap

config-map.yaml

apiVersion: v1

kind: ConfigMap

metadata:

name: game-demo

data:

*# property-like keys; each key maps to a simple value*

player\_initial\_lives: "3"

ui\_properties\_file\_name: "user-interface.properties"

*# file-like keys*

game.properties: |

*enemy.types=aliens,monsters*

*player.maximum-lives=5*

user-interface.properties: |

*color.good=purple*

*color.bad=yellow*

*allow.textmode=true*

k apply -f config-map.yaml

apiVersion: v1

kind: Pod

metadata:

name: configmap-pod

spec:

containers:

- name: mypod

image: redis

ports:

- containerPort: 80

volumeMounts :

- name: sample

mountPath: /home/labsuser/configfiles/

readOnly: true

volumes :

- name: sample

configMap:

name: game-demo

k apply -f cm-pod

k exec -it configmap-pod -- ls /home/labsuser/configfiles

k exec -it configmap-pod -- cat /home/labsuser/configfiles/game.properties

### 

### Secret

apiVersion: v1

kind: Secret

metadata:

name: mysecret

type: kubernetes.io/basic-auth

stringData:

username: admin

password: t0p-Secret

k apply -f secret

apiVersion: v1

kind: Pod

metadata:

name: secret-pod

spec:

containers:

- name: mypod

image: redis

volumeMounts:

- name: foo

mountPath: "/etc/foo"

readOnly: true

volumes:

- name: foo

secret:

secretName: mysecret

kubectl apply -f sec-pod

k exec -it secret-pod -- cat /etc/foo/username

k exec -it secret-pod -- cat /etc/foo/password

## Persistent Volume

k explain pod.spec.volumes | grep persistentVolumeClaim

k explain pv.spec

pv.yaml

apiVersion: v1

kind: PersistentVolume

metadata:

name: task-pv-volume

labels:

type: local

spec:

storageClassName: manual

capacity:

storage: 10Gi

accessModes:

- ReadWriteOnce

hostPath:

path: "/mnt/data"

k apply -f pv.yaml

k get pv

## Persistent Volume Claim

#PV Status before PVC creation

k get pv

pvc.yaml

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: task-pv-claim

spec:

storageClassName: manual

accessModes:

- ReadWriteOnce

resources:

requests:

storage: 3Gi

k apply -f pvc.yaml

#PVC and PV Status after PVC creation

k get pvc

k get pv

#Pod using PVC

#PVC Status before pod creation

k get pvc

pod-pvc.yaml

apiVersion: v1

kind: Pod

metadata:

name: task-pv-pod

spec:

volumes:

- name: task-pv-storage

persistentVolumeClaim:

claimName: task-pv-claim

containers:

- name: task-pv-container

image: nginx

ports:

- containerPort: 80

name: "http-server"

volumeMounts:

- mountPath: "/usr/share/nginx/html"

name: task-pv-storage

k apply -f pod-pvc.yaml

#PVC Status after pod creation

k get pvc

#Verify Volume by creating a file

k exec -it task-pv-pod -- touch /usr/share/nginx/html/index.html

k exec -it task-pv-pod -- ls /usr/share/nginx/html/

Identify the worker node, where pod is running/scheduled

k get pod task-pv-pod -o wide

In worker nodes, Example: worker02

ls /mnt/data

index.html

k delete -f pod-pvc.yaml

Verify file exists in worker node, Example: worker02

ls /mnt/data

index.html

k apply -f pod-pvc.yaml

k exec -it task-pv-pod -- ls /usr/share/nginx/html/

Index.html

## AKS - Creating PVC with dynamic PV

Creating sc

apiVersion: storage.k8s.io/v1

kind: StorageClass

metadata:

labels:

addonmanager.kubernetes.io/mode: EnsureExists

kubernetes.io/cluster-service: "true"

name: managed-premium-01

resourceVersion: "391"

uid: 08be1a5c-7fcf-4603-91eb-354e2fe27fa3

parameters:

cachingmode: ReadOnly

kind: Managed

storageaccounttype: Premium\_LRS

provisioner: disk.csi.azure.com

reclaimPolicy: Delete

volumeBindingMode: Immediate

### PVC using `Immediate` as volumeBindingMode

Creating PVC using our SC ‘managed-premium-01’

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: azure-managed-disk

spec:

accessModes:

- ReadWriteOnce

storageClassName: managed-premium-01

resources:

requests:

storage: 5Gi

k get pvc

k get pv

As soon as a PVC being created, PV gets created.

## 

### PVC using `WaitforFirstConsumer` as volumeBindingMode

Creating PVC using our SC ‘managed-premium’

apiVersion: v1

kind: PersistentVolumeClaim

metadata:

name: pvc-waitforconsumer

spec:

accessModes:

- ReadWriteOnce

storageClassName: managed-premium

resources:

requests:

storage: 5Gi

k get pvc

k get pv

PVC will be in Pending state, No PV will be created.

apiVersion: v1

kind: Pod

metadata:

name: task-pvc-consumer

spec:

volumes:

- name: task-pv-storage

persistentVolumeClaim:

claimName: pvc-waitforconsumer

containers:

- name: task-pv-container

image: nginx

ports:

- containerPort: 80

name: "http-server"

volumeMounts:

- mountPath: "/usr/share/nginx/html"

name: task-pv-storage

k apply -f pod-pvc.yaml

k get pvc

k get pv

PV and PVC will be created as soon as the Pod being created.

Pod Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal WaitForFirstConsumer 63s (x9 over 3m) persistentvolume-controller waiting for first consumer to be created before binding

Normal ExternalProvisioning 63s persistentvolume-controller Waiting for a volume to be created either by the external provisioner 'disk.csi.azure.com' or manually by the system administrator. If volume creation is delayed, please verify that the provisioner is running and correctly registered.

Normal Provisioning 63s disk.csi.azure.com\_csi-azuredisk-controller-77cf4b5fb6-t4g2p\_602733ad-4d6f-43c9-8a3d-6c5b92071628 External provisioner is provisioning volume for claim "default/pvc-waitforconsumer"

Normal ProvisioningSucceeded 60s disk.csi.azure.com\_csi-azuredisk-controller-77cf4b5fb6-t4g2p\_602733ad-4d6f-43c9-8a3d-6c5b92071628 Successfully provisioned volume pvc-644f1788-bf19-40cf-b844-3772fb83c90f