Exercise - Volumes

Name: Haffiz Hissham

Date: 21 September 2024

Q1

Create a new PersistentVolumeClaim:

⇨ Name: pv-volume

⇨ Class: csi-hostpath-sc

⇨ Capacity: 10Mi

Create a new Pod which mounts the PersistentVolumeClaim as a volume:

⇨ Name: web-server

⇨ Image: nginx

⇨ Mount path: /usr/share/nginx/html

Configure the new Pod to have ReadWriteOnce access on the volume.

Finally, using kubectl edit or kubectl patch expand the PersistentVolumeClaim to a capacity of 70Mi and record that change.

```yaml
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv
spec:
  storageClassName: csi-hostpath-sc
  capacity:
    storage: 10Mi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"

---

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-volume
spec:
  storageClassName: csi-hostpath-sc
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Mi
```

```
controlplane $ nano pv-volume.yaml
controlplane $
controlplane $ k apply -f pv-volume.yaml
persistentvolume/pv created
persistentvolumeclaim/pv-volume created
controlplane $
controlplane $ k get pvc
NAME        STATUS   VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS      VOLUMEATTRIBUTESCLASS   AGE
pv-volume   Bound    pv       10Mi       RWO            csi-hostpath-sc   <unset>                 4s
controlplane $
```

```
Exam Desktop   Editor   Tab 1   +
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: web-server
spec:
  volumes:
    - name: task-pv-storage
      persistentVolumeClaim:
        claimName: pv-volume
  containers:
    - name: web-server-pod
      image: nginx
#      ports:
#        - containerPort: 80
#          name: "http-server"
      volumeMounts:
        - mountPath: "/usr/share/nginx/html"
          name: task-pv-storage
```

```
controlplane $
controlplane $ nano web-server.yaml
controlplane $ k apply -f web-server.yaml
pod/web-server created
controlplane $
controlplane $ k describe pods web-server
Name:             web-server
Namespace:        default
Priority:         0
Service Account:  default
Node:             node01/172.30.2.2
Start Time:       Sat, 21 Sep 2024 04:56:51 +0000
Labels:           <none>
Annotations:      cni.projectcalico.org/containerID: 805b743058a261e89b8ff7234a087042cb8b72adfe3b17b481db3678f9238ffe
                  cni.projectcalico.org/podIP: 192.168.1.5/32
                  cni.projectcalico.org/podIPs: 192.168.1.5/32
Status:           Running
IP:               192.168.1.5
IPs:
  IP:  192.168.1.5
Containers:
  web-server-pod:
    Container ID:   containerd://ce35596971bd6f4bc255f79bb46f35955bbe4f2e6bbdcad84c2f260d58cf554d
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:04ba374043ccd2fc5c593885c0eacddebabd5ca375f9323666f28dfd5a9710e3
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sat, 21 Sep 2024 04:56:53 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /usr/share/nginx/html from task-pv-storage (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-87nqn (ro)
Conditions:
  Type                        Status
  PodReadyToStartContainers   True
  Initialized                 True
  Ready                       True
  ContainersReady             True
  PodScheduled                True
Volumes:
  task-pv-storage:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  pv-volume
    ReadOnly:   false
  kube-api-access-87nqn:
```

```
  GNU nano 4.8
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv
spec:
  storageClassName: csi-hostpath-sc
  capacity:
    storage: 70Mi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"

---

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: pv-volume
spec:
  storageClassName: csi-hostpath-sc
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 70Mi
```

```
controlplane $
controlplane $ k edit pvc pv-volume
Edit cancelled, no changes made.
controlplane $ nano pv-volume.yaml
controlplane $ k apply -f pv-volume.yaml
persistentvolume/pv configured
Error from server (Forbidden): error when applying patch:
{"metadata":{"annotations":{"kubectl.kubernetes.io/last-applied-configuration":"{\"apiVersion\":\"v1\",\"kind\":\"PersistentVolum
eClaim\",\"metadata\":{\"annotations\":{},\"name\":\"pv-volume\",\"namespace\":\"default\"},\"spec\":{\"accessModes\":[\"ReadWrit
eOnce\"],\"resources\":{\"requests\":{\"storage\":\"70Mi\"}},\"storageClassName\":\"csi-hostpath-sc\"}}\n"}},"spec":{"resources":
{"requests":{"storage":"70Mi"}}}}
to:
Resource: "/v1, Resource=persistentvolumeclaims", GroupVersionKind: "/v1, Kind=PersistentVolumeClaim"
Name: "pv-volume", Namespace: "default"
for: "pv-volume.yaml": error when patching "pv-volume.yaml": persistentvolumeclaims "pv-volume" is forbidden: only dynamically pr
ovisioned pvc can be resized and the storageclass that provisions the pvc must support resize
```

Q2

schedule a pod as follows:

- name: nginx-kusc00401
- image: nginx
- node selector: disk=ssd

```
controlplane $
controlplane $ kubectl label nodes node01 disk=ssd
node/node01 labeled
controlplane $
controlplane $ k get nodes --show-labels
NAME           STATUS   ROLES           AGE   VERSION   LABELS
controlplane   Ready    control-plane   11d   v1.30.0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/a
rch=amd64,kubernetes.io/hostname=controlplane,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/ex
clude-from-external-load-balancers=
node01         Ready    <none>          11d   v1.30.0   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,disk=ssd,kubern
etes.io/arch=amd64,kubernetes.io/hostname=node01,kubernetes.io/os=linux
controlplane $
```

```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: nginx-kusc00401
spec:
  containers:
  - name: nginx-kusc00401
    image: nginx
  nodeSelector:
    disk: ssd
```

```
controlplane $ nano nginx-kusc00401.yaml
controlplane $
controlplane $ k apply -f nginx-kusc00401.yaml
pod/nginx-kusc00401 created
controlplane $
controlplane $ k get pods -o wide
NAME              READY   STATUS              RESTARTS   AGE   IP       NODE     NOMINATED NODE   READINESS GATES
nginx-kusc00401   0/1     ContainerCreating   0          5s    <none>   node01   <none>           <none>
controlplane $
controlplane $ k get pods -o wide
NAME              READY   STATUS    RESTARTS   AGE   IP            NODE     NOMINATED NODE   READINESS GATES
nginx-kusc00401   1/1     Running   0          11s   192.168.1.4   node01   <none>           <none>
controlplane $
```

Q3

Create a new pod called web-pod with image busy box
Allow the pod to be able to set system_time
The container should sleep for 3200 seconds

```
  GNU nano 4.8
apiVersion: v1
kind: Pod
metadata:
  name: web-pod
spec:
  containers:
  - name: web-pod
    image: busybox
    command: [ "sh", "-c", "sleep 3200" ]
    securityContext:
      capabilities:
        add: ["SYS_TIME"]
```

```
controlplane $
controlplane $ nano web-pod.yaml
controlplane $
controlplane $ k apply -f web-pod.yaml
pod/web-pod created
controlplane $
controlplane $ k get pods | grep web-pod
web-pod          1/1     Running   0          10s
controlplane $
controlplane $ k describe pods web-pod
Name:             web-pod
```

```
    Host Port:     <none>
    Command:
      sh
      -c
      sleep 3200
    State:          Running
      Started:      Sat, 21 Sep 2024 05:08:50 +000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
```

Q4

Create a new PersistentVolume named safari-pv. It should have a capacity of 2Gi, accessMode ReadWriteOnce, hostPath /Volumes/Data and no storageClassName defined.

Next create a new PersistentVolumeClaim in Namespace project-tiger named safari-pvc . It should request 2Gi storage, accessMode ReadWriteOnce and should not define a storageClassName. The PVC should bound to the PV correctly.

Finally create a new Deployment safari in Namespace project-tiger which mounts that volume at /tmp/safari-data. The Pods of that Deployment should be of image httpd:2.4.41-alpine.

```
 GNU nano 4.8
apiVersion: v1
kind: PersistentVolume
metadata:
  name: safari-pv
spec:
  storageClassName: ""
  capacity:
    storage: 2Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/Volumes/Data"

---

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: safari-pvc
  namespace: project-tiger
spec:
  storageClassName: ""
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 2Gi
```

```
controlplane $
controlplane $ k create ns project-tiger
namespace/project-tiger created
controlplane $
controlplane $ nano safafi-pv-pvc.yaml
controlplane $
controlplane $ k apply -f safafi-pv-pvc.yaml
persistentvolume/safari-pv configured
persistentvolumeclaim/safari-pvc created
controlplane $
controlplane $ k get pvc
No resources found in default namespace.
controlplane $ k get pvc -n project-tiger
NAME         STATUS   VOLUME      CAPACITY   ACCESS MODES   STORAGECLASS   VOLUMEATTRIBUTESCLASS   AGE
safari-pvc   Bound    safari-pv   2Gi        RWO                           <unset>                 11s
controlplane $
```

```
  GNU nano 4.8
apiVersion: apps/v1
kind: Deployment
metadata:
  creationTimestamp: null
  labels:
    app: safari
  name: safari
  namespace: project-tiger
spec:
  replicas: 1
  selector:
    matchLabels:
      app: safari
  strategy: {}
  template:
    metadata:
      creationTimestamp: null
      labels:
        app: safari
    spec:
      volumes:
        - name: task-pv-storage
          persistentVolumeClaim:
            claimName: safari-pvc
      containers:
      - image: httpd:2.4.41-alpine
        name: httpd
        volumeMounts:
          - mountPath: "/tmp/safari-data"
            name: task-pv-storage
        resources: {}
status: {}
```

```
controlplane $
controlplane $ k create deployment safari --image=httpd:2.4.41-alpine -n project-tiger --dry-run=client -o yaml > safari.yaml
controlplane $
controlplane $ nano safari.yaml
controlplane $
controlplane $ k apply -f safari.yaml
deployment.apps/safari created
controlplane $
controlplane $ k get deployments.apps
No resources found in default namespace.
controlplane $ k get deployments.apps -n project-tiger
NAME     READY   UP-TO-DATE   AVAILABLE   AGE
safari   1/1     1            1           9s
controlplane $
```

```
controlplane $
controlplane $ k describe deployments apps safari -n project-tiger
Name:                   safari
Namespace:              project-tiger
CreationTimestamp:      Sat, 21 Sep 2024 05:21:19 +0000
Labels:                 app=safari
Annotations:            deployment.kubernetes.io/revision: 1
Selector:               app=safari
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  25% max unavailable, 25% max surge
Pod Template:
  Labels:  app=safari
  Containers:
   httpd:
    Image:        httpd:2.4.41-alpine
    Port:         <none>
    Host Port:    <none>
    Environment:  <none>
    Mounts:
      /tmp/safari-data from task-pv-storage (rw)
  Volumes:
   task-pv-storage:
    Type:          PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:     safari-pvc
    ReadOnly:      false
  Node-Selectors:  <none>
  Tolerations:     <none>
Conditions:
  Type           Status  Reason
  ----           ------  ------
  Available      True    MinimumReplicasAvailable
  Progressing    True    NewReplicaSetAvailable
OldReplicaSets:  <none>
NewReplicaSet:   safari-8db94978d (1/1 replicas created)
Events:
  Type    Reason             Age   From                   Message
  ----    ------             ----  ----                   -------
  Normal  ScalingReplicaSet  35s   deployment-controller  Scaled up replica set safari-8db94978d to 1
controlplane $
```