Step 1 - Creating a container with an emptyDir volume

Create a YAML File

vi emptyDir.yaml

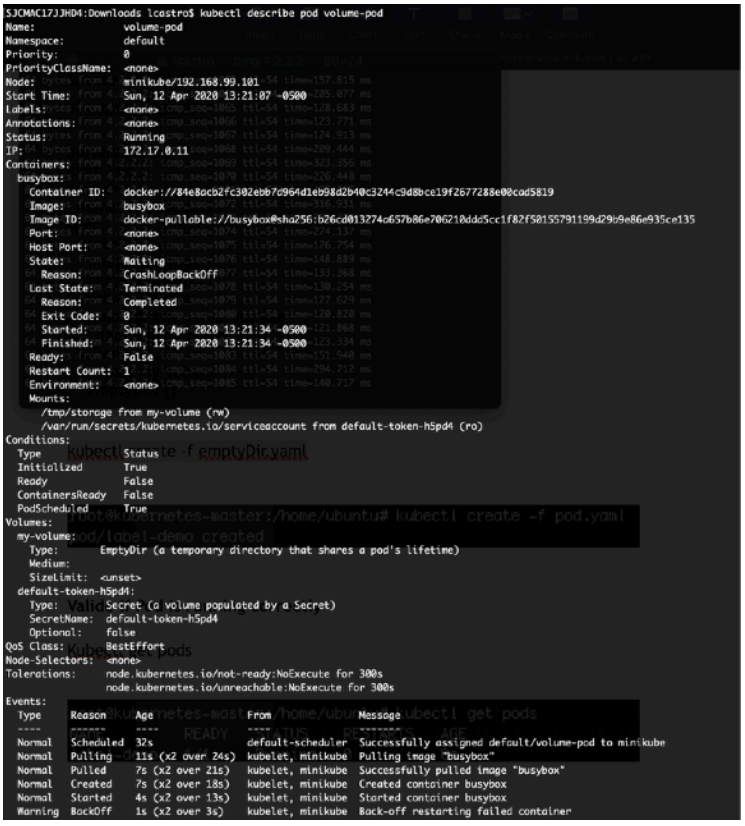
```
apiVersion: v1
kind: Pod
metadata:
  name: volume-pod
spec:
  containers:
  - image: busybox
    name: busybox
    volumeMounts:
    - mountPath: /tmp/storage
      name: my-volume
  volumes:
  - name: my-volume
    emptyDir: {}
```

kubectl create -f emptyDir.yaml

```
SJCMAC17JJHD4:Downloads lcastro$ kubectl create -f emptyDir.yaml
pod/volume-pod created
```

Valide Pods emptyDir

Kubectl describe pod volume-pod



Step 2 - Creating a Persistence Volume

vi pv.yaml

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv-log
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 100Mi
  hostPath:
    path: /pv/log
```

kubectl create -f pv.yaml

```
SJCMAC17JJHD4:Downloads lcastro$ kubectl create -f pv.yaml
persistentvolume/pv-log created
```

Validate PV has been created

Kubectl get pv

```
SJCMAC17JJHD4:Downloads lcastro$ kubectl get pv
NAME      CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM   STORAGECLASS   REASON   AGE
pv-log    100Mi      RWX            Retain           Pending                                     8s
```

Kubectl describe pv pv-log

```
SJCMAC17JJHD4:Downloads lcastro$ kubectl describe pv pv-log
Name:            pv-log
Labels:          <none>
Annotations:     <none>
Finalizers:      []
StorageClass:
Status:          Pending
Claim:
Reclaim Policy:  Retain
Access Modes:    RWX
VolumeMode:      Filesystem
Capacity:        100Mi
Node Affinity:   <none>
Message:
Source:
    Type:          HostPath (bare host directory volume)
    Path:          /pv/log
    HostPathType:
Events:            <none>
```

Step 3 - Create a Persistence Volume Claim

vi pvc.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
 name: claim-log-1
spec:
 accessModes:
 - ReadWriteMany
 resources:
   requests:
     storage: 50Mi
```

kubectl create -f pvc.yaml

```
SJCMAC17JJHD4:Downloads lcastro$ vi pvc.yaml
SJCMAC17JJHD4:Downloads lcastro$ kubectl create -f pvc.yaml
persistentvolumeclaim/claim-log-1 created
```

Validate PVC has been created

kubectl describe pvc claim-log-1

```
root@kubernetes-master:/home/ubuntu# kubectl describe  pvc claim-log-1
Name:          claim-log-1
Namespace:     default
StorageClass:
Status:        Bound
Volume:        pv-log
Labels:        <none>
Annotations:   pv.kubernetes.io/bind-completed: yes
               pv.kubernetes.io/bound-by-controller: yes
Finalizers:    [kubernetes.io/pvc-protection]
Capacity:      100Mi
Access Modes:  RWX
VolumeMode:    Filesystem
Mounted By:    webapp
Events:        <none>
```

Validate PV has a Claim from PVC created

kubectl get pv

```
SJCMAC17JJHD4:Downloads lcastro$ kubectl get pv
NAME                                       CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS      CLAIM                              STORAGECLASS   REASON   AGE
pv-log                                     100Mi      RWX            Retain           Pending                                                                 5m50s
pvc-1164b547-4ace-11ea-9af0-08002716856b   100Gi      RWO            Delete           Bound       twistlock/twistlock-console        standard               63d
pvc-d98486ec-7cec-11ea-9b76-08002716856b   50Mi       RWO            Delete           Pending     default/claim-log-1                standard               109s
```

Step 4 - Use a PersistentVolumeClaim in a Pod

vi pod-pvc.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: webapp
spec:
  containers:
  - name: event-simulator
    image: kodekloud/event-simulator
    env:
    - name: LOG_HANDLERS
      value: file
    volumeMounts:
    - mountPath: /log
      name: log-volume
  volumes:
  - name: log-volume
    persistentVolumeClaim:
      claimName: claim-log-1
```

kubectl create -f pod-pvc.yaml

```
root@kubernetes-master:/home/ubuntu# kubectl create -f pod-pvc.yaml
pod/webapp created
```

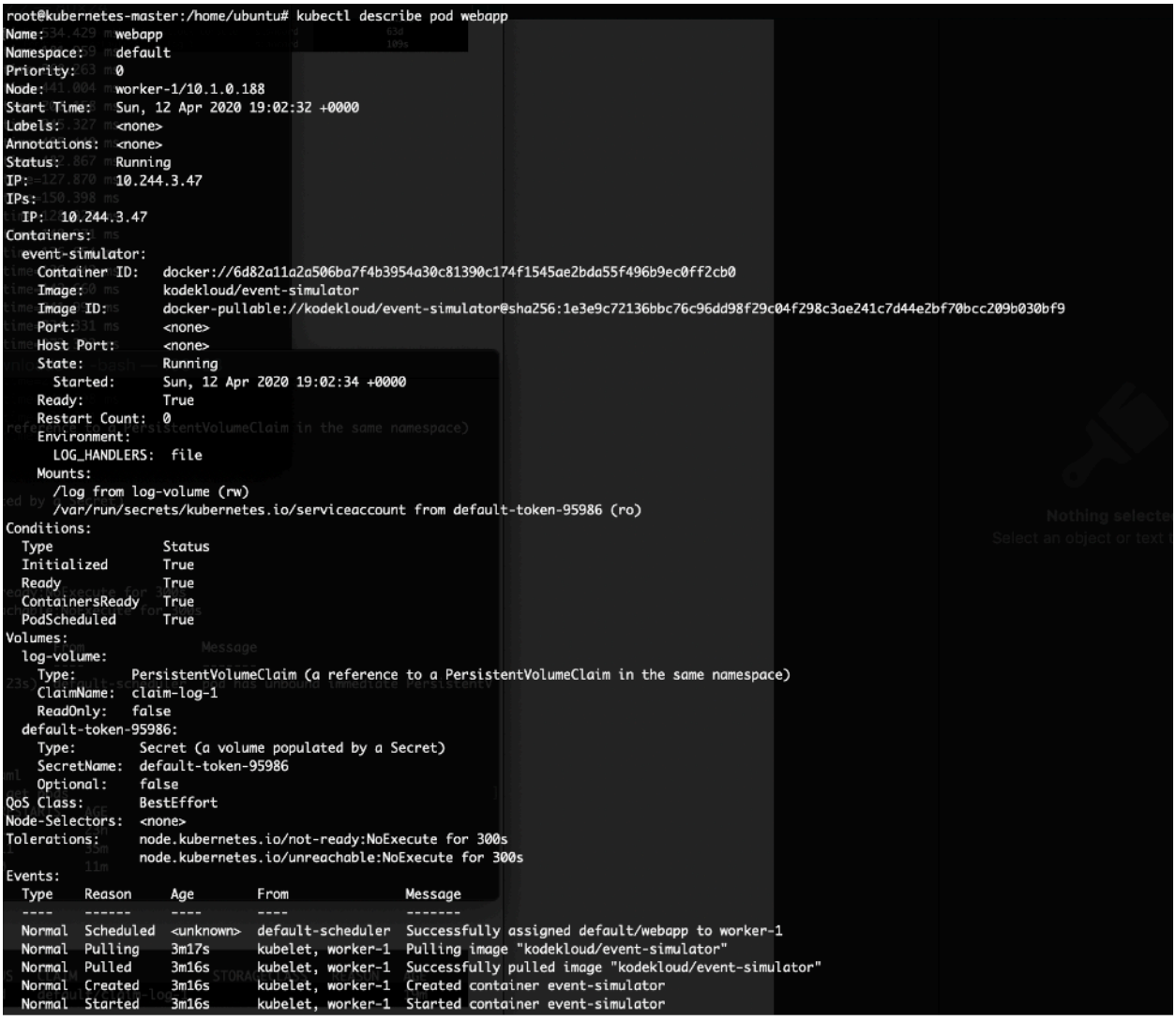
Validate that PV has been Claimed from Pod

kubectl get pv

```
root@kubernetes-master:/home/ubuntu# kubectl get pv
NAME     CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                 STORAGECLASS   REASON   AGE
pv-log   100Mi      RWX            Retain           Bound    default/claim-log-1                           19m
```

Validate Pod has the PVC mounted

Kubectl describe pod webapp



```
root@kubernetes-master:/home/ubuntu# kubectl describe pod webapp
Name:         webapp
Namespace:    default
Priority:     0
Node:         worker-1/10.1.0.188
Start Time:   Sun, 12 Apr 2020 19:02:32 +0000
Labels:       <none>
Annotations:  <none>
Status:       Running
IP:           10.244.3.47
IPs:
  IP:  10.244.3.47
Containers:
  event-simulator:
    Container ID:   docker://6d82a11d2a506ba7f4b3954a30c81390c174f1545ae2bda55f496b9ec0ff2cb0
    Image:          kodekloud/event-simulator
    Image ID:       docker-pullable://kodekloud/event-simulator@sha256:1e3e9c72136bbc76c96dd98f29c04f298c3ae241c7d44e2bf70bcc209b030bf9
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Sun, 12 Apr 2020 19:02:34 +0000
    Ready:          True
    Restart Count:  0
    Environment:
      LOG_HANDLERS:  file
    Mounts:
      /log from log-volume (rw)
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-95986 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  log-volume:
    Type:       PersistentVolumeClaim (a reference to a PersistentVolumeClaim in the same namespace)
    ClaimName:  claim-log-1
    ReadOnly:   false
  default-token-95986:
    Type:        Secret (a volume populated by a Secret)
    SecretName:  default-token-95986
    Optional:    false
QoS Class:       BestEffort
Node-Selectors:  <none>
Tolerations:     node.kubernetes.io/not-ready:NoExecute for 300s
                 node.kubernetes.io/unreachable:NoExecute for 300s
Events:
  Type    Reason     Age        From               Message
  ----    ------     ----       ----               -------
  Normal  Scheduled  <unknown>  default-scheduler  Successfully assigned default/webapp to worker-1
  Normal  Pulling    3m17s      kubelet, worker-1  Pulling image "kodekloud/event-simulator"
  Normal  Pulled     3m16s      kubelet, worker-1  Successfully pulled image "kodekloud/event-simulator"
  Normal  Created    3m16s      kubelet, worker-1  Created container event-simulator
  Normal  Started    3m16s      kubelet, worker-1  Started container event-simulator
```

Step 5 - Clean the environment

Kubectl delete pod webapp

Kubectl delete pvc claim-log-1

Kubectl delete pv pv-log

```
root@kubernetes-master:/home/ubuntu# kubectl delete pod webapp
pod "webapp" deleted
root@kubernetes-master:/home/ubuntu# kubectl delete pvc claim-log-1
persistentvolumeclaim "claim-log-1" deleted
root@kubernetes-master:/home/ubuntu# kubectl delete pv pv-log
persistentvolume "pv-log" deleted
```