KubeCon | CloudNativeCon

North America 2017

K8s Storage Developments for Stateful Workloads

Erin Boyd, Software Engineer, *Red Hat*
Michelle Au, Software Engineer, *Google*

Agenda

- Background
- Problem
- Solution
- Demo
- Future

Expected Knowledge

Kubernetes

- Pods
- Labels
- Nodes
- PersistentVolumeClaims
- PersistentVolumes
- StorageClasses
- StatefulSets


© Brandon Hill

Background

Local Persistent Volumes & Raw Block Volumes

Stateful, Distributed Workloads

- Cassandra, MongoDB, GlusterFS, etc.
  - Replicate sharded data for high availability, fault tolerance
  - Critical infrastructure / applications
- Data locality for performance
- Data gravity
  - Execute on where the data is today
- High performance tuning

Background

Kubernetes Features that benefit stateful workloads

- StatefulSets for stable identity and volumes
- Pod Disruption Budget for controlled disruption
- Pod Affinity, Anti-Affinity for co-location, spreading (1.6 beta)
- Pod Priority and Preemption (1.8 alpha)

Problem

Difficult to access high performance local storage

Hostpath volumes have a lot of problems

- Not portable
- Security risk!

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  nodeName: node-1
  volumes:
    - name: data
      hostPath:
        path: /mnt/some-disk
  containers:
  ...
```

Problem

Today's workarounds

- Manually maintain a Pod spec for each node
- Custom scheduler and/or operator
- Custom local disk reservation and lifecycle manager

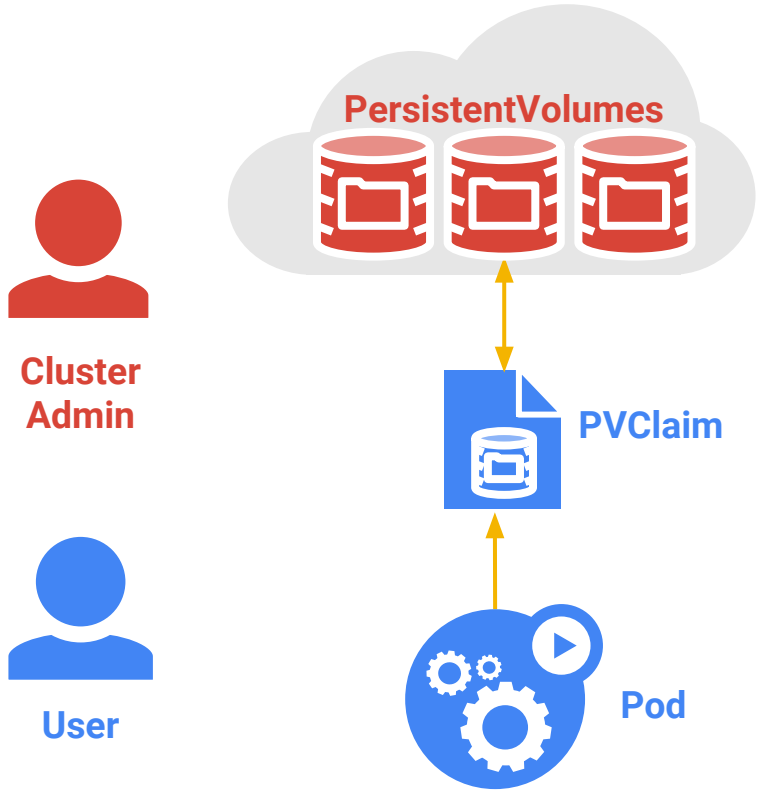
Consequences

- Can't leverage existing Kubernetes features
  - StatefulSets, scaling, rolling updates, etc.
- High barrier to entry for adopting Kubernetes

Local Persistent Volumes

Extend existing PersistentVolumeClaim, PersistentVolume model

- PVC: User's storage requests
  - "I need 100GB of fast storage"
- PV: Cluster's specific volume implementation
  - "I have a 100GB local volume available on node-1 at this /mnt/disks/ssd0"



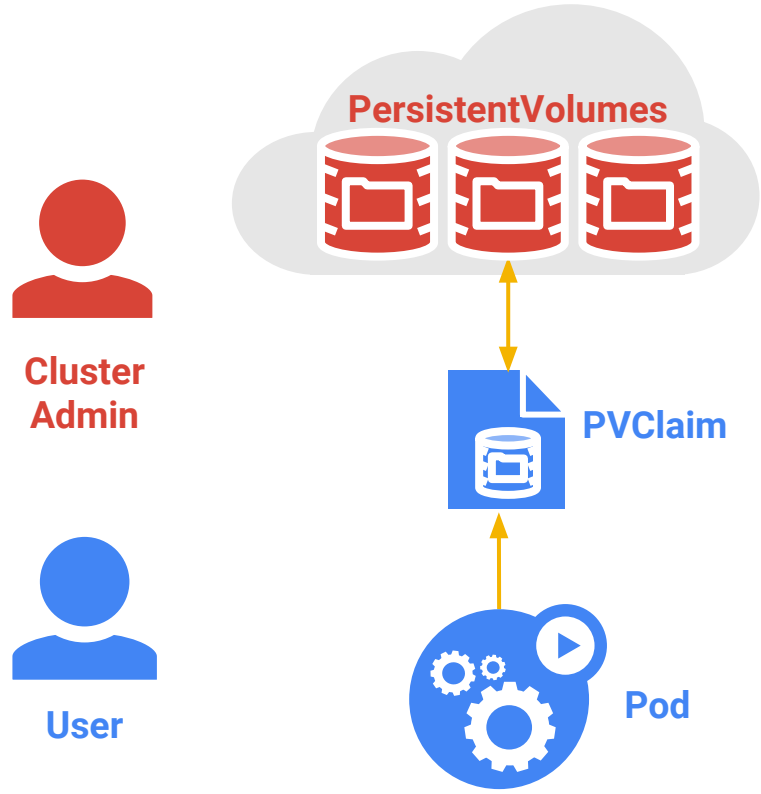
Example: User's Pod and Claim

```
apiVersion: v1
kind: Pod
metadata:
  name: my-pod
spec:
  nodeName: node-1
  volumes:
    - name: data
      hostPath:
        path: /mnt/some-disk
      persistentVolumeClaim:
        claimName: my-pvc
  containers:
    ...
```

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: my-pvc
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
  storageClassName: my-class
```

Example: Admin

```
apiVersion: v1
kind: PersistentVolume
metadata:
  Name: local-volume-1
spec:
  accessModes:
  - ReadWriteOnce
  capacity:
    storage: 100Gi
  storageClassName: my-class
  local:
    path: /tmp/my-test1
   nodeAffinity:
     requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
            - node-1
```



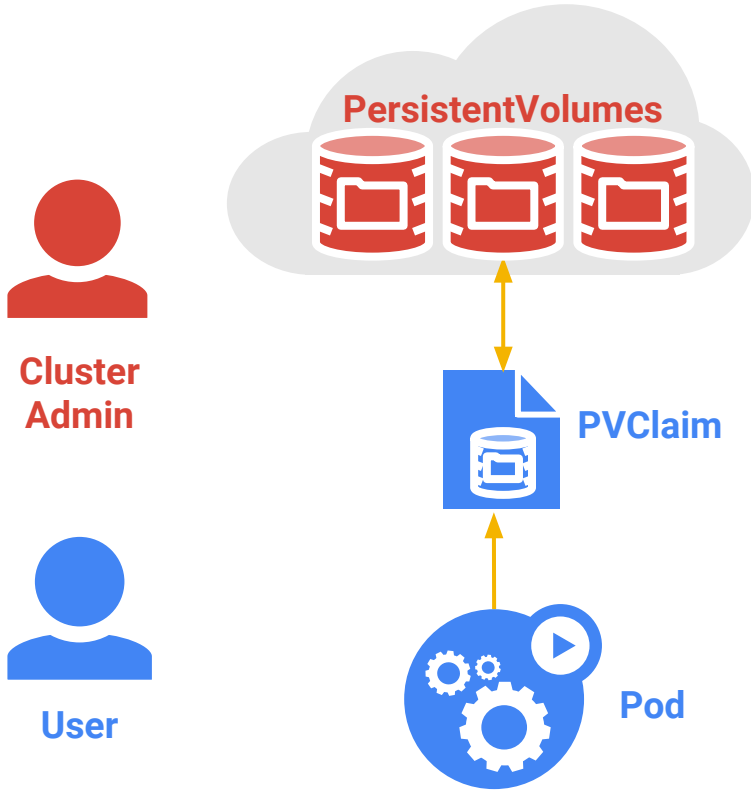
Local Persistent Volumes

1.7 Alpha

- “Local” PersistentVolume type with NodeAffinity
- Scheduler logic for data gravity

1.9 Alpha

- Perform PVC/PV binding during pod scheduling



Local Persistent Volumes

- Portable, consistent user experience
  - Across local and remote storage
  - Across clusters, environments
- General mechanism for volume topology
- Lowers the barrier for distributed, stateful workloads

Raw Block Volumes

1.9 alpha feature goals

- Expose Raw block devices in line with Kube primitives
- Enable durable access to raw block storage
- Provide flexibility for users/vendors to support all storage types
  - Prior to v1.8 all users got a volume with a filesystem
- Break GitHub

This page is taking way too long to load.

Sorry about that. Please try refreshing and contact us if the problem persists.

Contact Support — GitHub Status — @githubstatus

Example: Admin

```
apiVersion: v1
kind: PersistentVolume
metadata:
  Name: local-volume-1
spec:
  volumeMode: Block
  capacity:
    storage: 100Gi
  storageClassName: my-class
  local:
    path: /dev/xdb
   nodeAffinity:
     ...
```

Example: User's Pod and Claim

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: raw-pvc
spec:
  volumeMode: Block
  accessModes:
    - ReadWriteOnce
  storageClassName: my-class
  resources:
    requests:
      storage: 100Gi
```

```
apiVersion: v1
kind: Pod
metadata:
   name: my-db
spec:
   containers:
   - name: mysql
     image: mysql
     volumeDevices:
     - name: my-db-data
       devicePath: /var/lib/mysql
   volumes:
   - name: my-db-data
     persistentVolumeClaim:
        claimName: raw-pvc
```

Demo

See how easy it is to switch between remote and local storage!

Replicated MySQL example using StatefulSets:
https://kubernetes.io/docs/tasks/run-application/run-replicated-stateful-application/

Try it out yourself:

- Follow local volume user guide to bring up a cluster with some local disks
- Take existing StatefulSet examples and Helm charts, and change the StorageClassName in the PersistentVolumeClaim to your local StorageClass

Summary

1.9 Alpha Features

- Local persistent volumes with node affinity and smarter scheduling
- Consumption of statically provisioned raw block persistent volumes for Fibre Channel

Building blocks for stateful, distributed, performant workloads

Future

- Dynamically provision volumes during pod scheduling
- Dynamically provision raw block volumes
- Raw block support for remaining volume plugins:
  - Local volumes
  - GCE PD
  - AWS EBS
  - GlusterFS
  - Ceph
  - Cinder
- CSI interface update for block devices

Links

Local volume user guide
https://github.com/kubernetes-incubator/external-storage/tree/master/local-volume

Raw Block volume user guide
https://kubernetes.io/docs/concepts/storage/persistent-volumes/

Questions?

Get Involved!

- Kubernetes Storage Special-Interest-Group (SIG)
- Bi-monthly meetings Thursdays at 9 AM (PST)
- http://slack.k8s.io/

Erin Boyd <eboyd@redhat.com>
Github: @erinboyd
Twitter: @erinaboyd

Michelle Au <msau@google.com>
Github: @msau42
Twitter: @_msau42_

http://kubernetes.io

Images by Connie Zhou