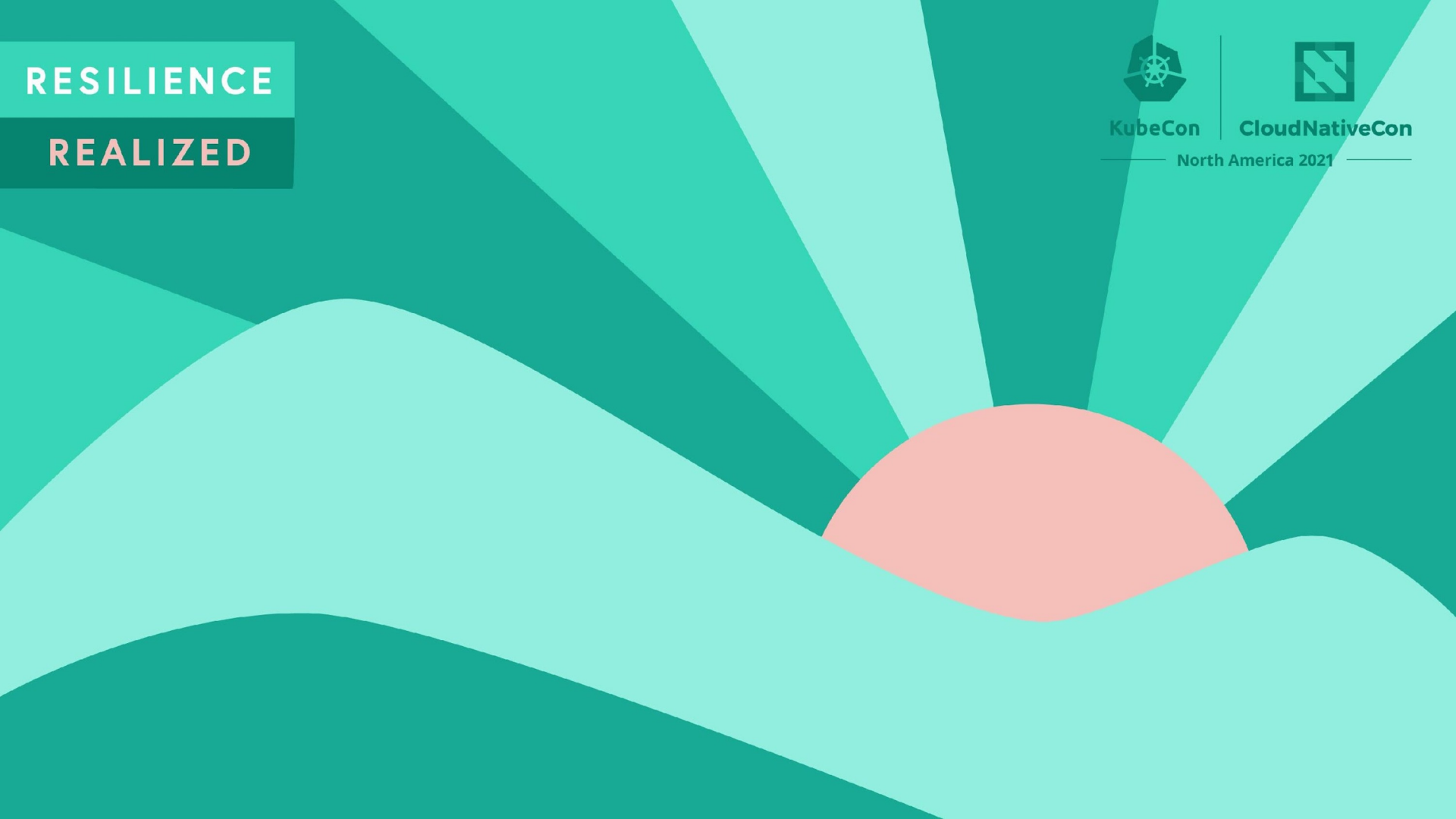

RESILIENCE
REALIZED
KubeCon
CloudNativeCon
North America 2021


KubeCon | CloudNativeCon

North America 2021

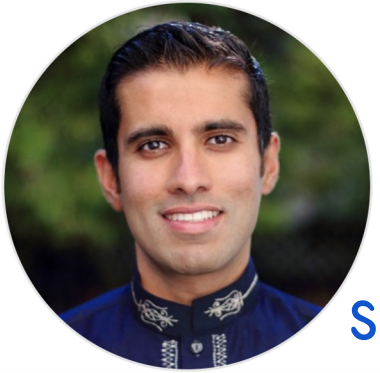
RESILIENCE

REALIZED

Kubernetes SIG-Storage Introduction & Update

Xing Yang, VMware & Michelle Au, Google

SIG-Storage Leads



Saad Ali

SIG-Storage
Co-Chair

Xing Yang

Michelle Au

SIG-Storage
Tech Lead

Jan Šafránek

Agenda

- SIG-Storage Introduction
- SIG-Storage Update


KubeCon | CloudNativeCon

North America 2021

RESILIENCE

REALIZED

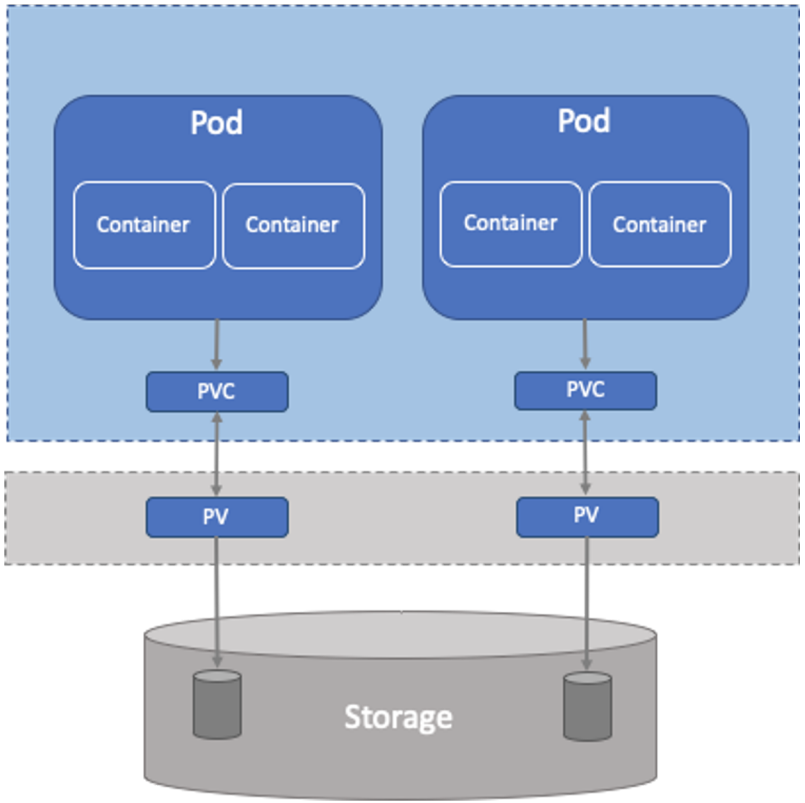
SIG-Storage Introduction

Agenda for Introduction

- Persistent Storage
- Ephemeral Storage
- Volume Plugins
  - Container Storage Interface (CSI)
- How to get involved

Persistent Storage

- Persistent Storage has a lifecycle independent of the pod
- Pod
  - Mounts PVC into container(s)
- PersistentVolumeClaim (PVC)
  - Requested by user for storage
  - Used by Pod(s)
- PersistentVolume (PV)
  - Points to physical volume on storage system
  - Created by admin in “static provisioning”
  - Created by Kubernetes in “dynamic provisioning”
- PVC and PV has one-to-one bi-directional binding
- StorageClass
  - A way for admin to describe the “classes” of storage
  - Parameters for dynamic provisioning



Pod, PVC, StorageClass


```yaml
kind: Pod
apiVersion: v1
metadata:

    name: mysql
spec:
    volumes:
    - name: data
       persistentVolumeClaim:
            claimName: mysql-pvc
     containers:
     - image: mysql:5.6
        name: mysql
       env:
       - name: MYSQL_ROOT_PASSWORD
         value: password
       volumeMounts:
       - name: data

         mountPath: /var/lib/mysql
```

```yaml
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
    name: mySC
    annotations:
        storageclass.kubernetes.io/is-default-class: "true"
provisioner: kubernetes.io/my-driver
reclaimPolicy: Retain
allowVolumeExpansion: true
mountOptions:
  - debug
volumeBindingMode: Immediate
parameters:
```

```yaml
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
    name: mysql-pvc
spec:
    resources:
        requests:
            storage: 1Gi
    accessModes:
        - ReadWriteOnce
    storageClassName: mySC
```

Ephemeral Storage

- Ephemeral Storage has a lifecycle bound to the pod
- Local ephemeral storage
  - Emptydir
  - Inject data into pod
    - Secrets
    - Configmaps
    - Downward APIs
- CSI ephemeral volumes: can be provided by CSI drivers
- Generic ephemeral volumes: any plugin supports PVC

Local Ephemeral Storage - Emptydir

- Empty at pod startup
- with storage coming locally from the kubelet base directory (usually the root disk) or RAM
- Used as scratch space by containers in the Pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-pd
spec:
  containers:
  - image: k8s.gcr.io/test-webserver
    name: test-container
    volumeMounts:
    - mountPath: /cache
      name: cache-volume
  volumes:
  - name: cache-volume
    emptyDir: {}
```

Local Ephemeral Storage - Secret

- Used to pass sensitive information, such as passwords, to pods
- Mounted as files for use by pods

```yaml
apiVersion: v1
kind: Secret
metadata:
  name: mysecret
type: Opaque
data:
  username: YWRtaW4=
  password: MWYyZDFlMmU2N2Rm
```

```yaml
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mypod
    image: redis
    volumeMounts:
    - name: foo
      mountPath: "/etc/foo"
      readOnly: true
  volumes:
  - name: foo
    secret:
      secretName: mysecret
```

Local Ephemeral Storage - ConfigMap

- ConfigMap provides a way to inject configuration data into pods
- Used to store non-confidential data in key-value pairs

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: configmap-pod
spec:
  containers:
    - name: test
      image: busybox
      volumeMounts:
        - name: config-vol
          mountPath: /etc/config
  volumes:
    - name: config-vol
      configMap:
        name: log-config
        items:
          - key: log_level
            path: log_level
```

Local Ephemeral Storage - DownwardAPI

- Makes downward API data available to applications
- Mounts a directory and writes the requested data

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: kubernetes-downwardapi-volume-example
spec:
  containers:
    - name: client-container
      image: k8s.gcr.io/busybox
      command: ["sh", "-c"]
      Args:......
      volumeMounts:
        - name: podinfo
          mountPath: /etc/podinfo
  volumes:
    - name: podinfo
      downwardAPI:
        items:
          - path: "labels"
            fieldRef:
              fieldPath: metadata.labels
          - path: "annotations"
            fieldRef:
              fieldPath: metadata.annotations
```

CSI Inline Ephemeral Volume

- Specialized volumes, “Secrets”
- Pod knows what it wants
- Usually read-only volumes
- Requires special CSI drivers such as Secret Store CSI Driver

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: some-pod
spec:
  containers:
    ...
  volumes:
      - name: vol
        csi:
          driver: inline.storage.kubernetes.io
          volumeAttributes:
              foo: bar
```

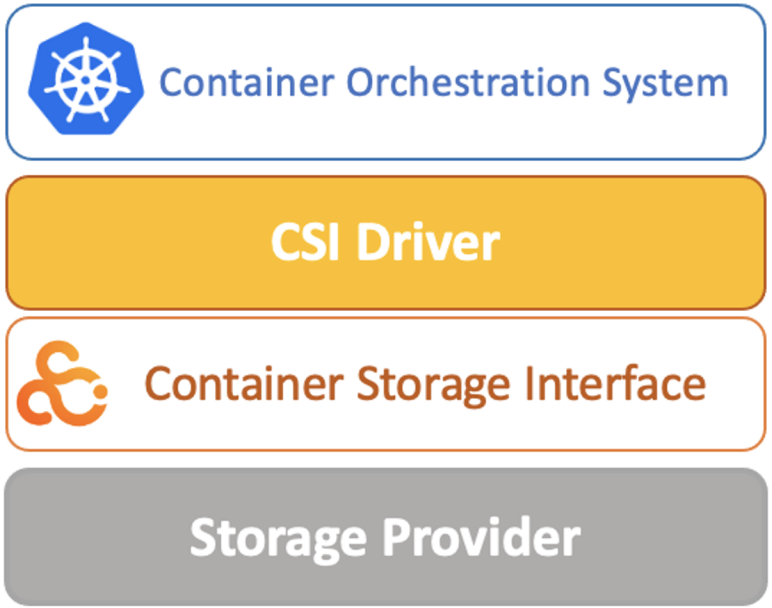
Generic Ephemeral Volume

- “Better EmptyDir”; useful when boot disk is not big enough
- Pod gets “ephemeral” PVC
  - Created on Pod creation
  - Deleted on Pod deletion
  - Just a regular PVC to CSI drivers
- Provisioned PV is usually empty
- All features supported with PVC are supported
  - Cloning, volume snapshot, volume resize, storage capacity tracking, etc.
  - No need to write a special driver
- Targeting GA in 1.23
- KEP, Blog, Docs

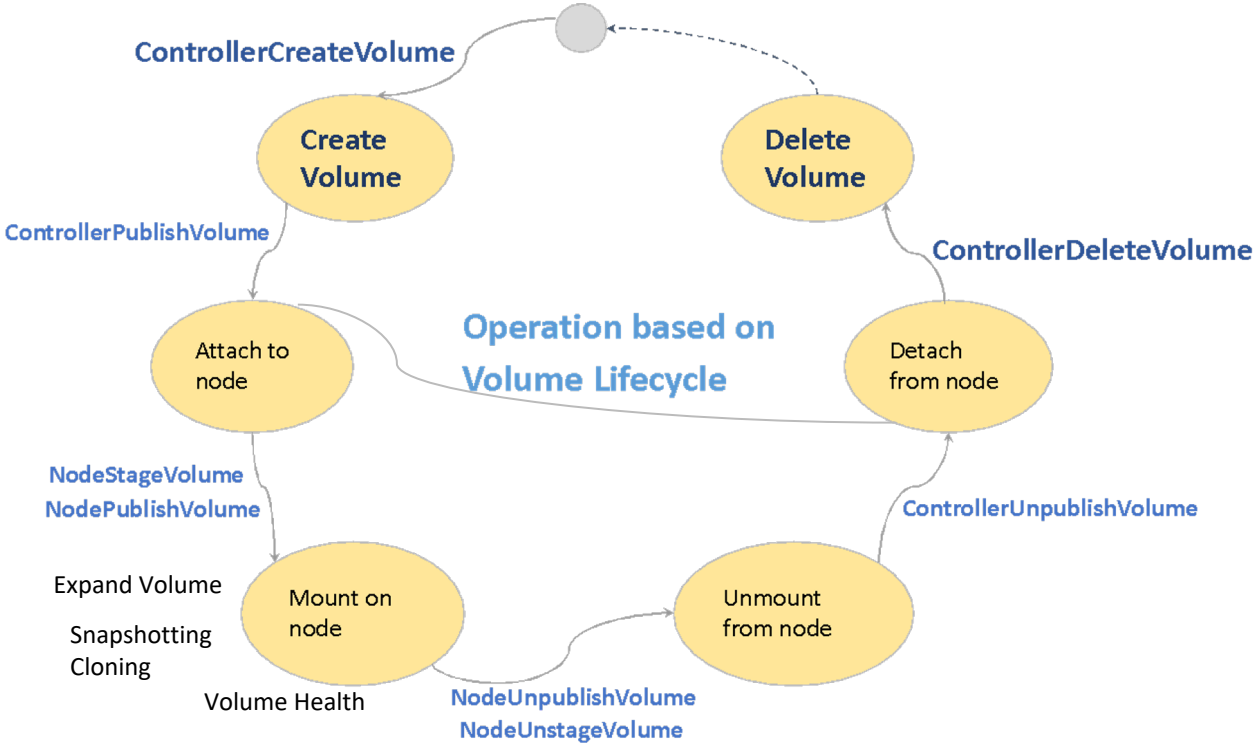
```yaml
kind: Pod
...
spec:
  containers:
    ...
  volumes:
  - name: my-csi-volume
    ephemeral:
      volumeClaimTemplate:
        spec:
          storageClassName: "test-sc"
          accessModes:
          - ReadWriteOnce
          resources:
            requests:
              storage: 4Gi
```

Volume Plugins

- In-tree Plugins
  - CSI migration
- Flexvolume
  - Deprecated
- Container Storage Interface (CSI)
  - Recommended way to write plugins

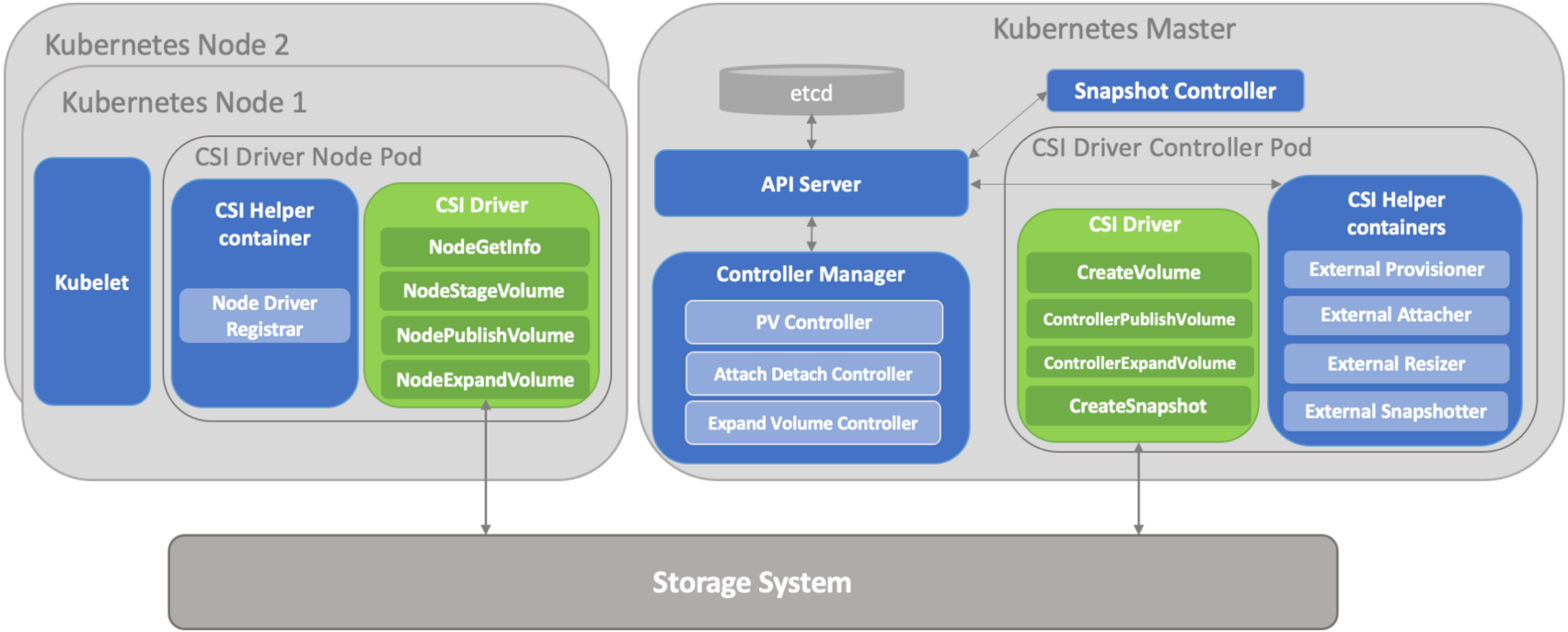


Container Storage Interface (CSI)



Volume Lifecycle in CSI Spec

CSI Deployment Example



How to Get Involved

- Start at the SIG Storage page:
  - https://github.com/kubernetes/community/tree/master/sig-storage
- Attend the bi-weekly meetings: 9 AM PT every second Thursday.
- Mailing List:
  - kubernetes-sig-storage@googlegroups.com
- Slack channel:
  - #sig-storage
  - #csi
  - #sig-storage-cosi

Resources

- SIG Storage page
- Storage concepts
- CSI driver docs
- CSI spec
- CSI sample driver hostpath deployment example

SIG-Storage Update

Agenda for SIG-Storage Update

- Deep Dive: CSI Migration
- Deep Dive: CSI Windows
- What we did in 1.22
- What we are working on in 1.23
- Features in design/prototyping
- Cross SIG WG/projects

Deep Dive on CSI Migration

- Why? Built-in cloud providers are deprecated and target for removal in 1.24.
- What? CSI Migration allows your existing PVs and StorageClasses using these in-tree volume plugins to continue working even when built-in cloud providers are removed
  - kubernetes.io/aws-ebs
  - kubernetes.io/azure-disk
  - kubernetes.io/azure-file
  - kubernetes.io/cinder
  - kubernetes.io/gce-pd
  - kubernetes.io/vsphere-volume

Deep Dive on CSI Migration

- What do I do?
  - Using a managed Kubernetes distribution? Check your distro's documentation. In many cases, distro will take care of everything.
  - Managing your own Kubernetes?
    - Install the replacement CSI driver for your cloud
    - Enable CSIMigration and CSIMigrationX feature gates (X is the specific driver): detailed ordering
- Caveats
  - CSI-only features do not work (e.g., snapshots, cloning, etc)
    - Manually re-import PV as CSI type
  - Some in-tree functionality has been deprecated and won't work with migration (check Kubernetes release notes)

Deep Dive on CSI Windows

- GA in 1.22
- Windows containers cannot be privileged yet*
- CSI Driver communicates with CSI proxy binary via gRPC to do privileged operations
- Supported protocols
  - NTFS
  - SMB
  - iSCSI (alpha)
- Some available drivers: AWS EBS, Azure Disk, Azure File, GCE PD, SMB

What we did in 1.22

- GA
  - CSI Windows
  - Pass pod service account token to CSI
    - Enables CSI drivers to authenticate as pod
- Alpha
  - Volume populator (re-design)
    - After provisioning, populate pod with data before giving to pod
  - Read Write Once Pod PV Access Mode
    - Enforces at most a single pod can mount a volume at a time
  - Delegate FSGroup to CSI Driver instead of Kubelet
    - More efficient fsgroup handling for certain drivers

What we are working on in 1.23

- Targeting GA
  - Skip volume ownership (FSGroup)
  - CSI FSGroup Policy
  - Generic ephemeral volumes
- Targeting Beta
  - Delegate FSGroup to CSI Driver instead of Kubelet (alpha in 1.22)
  - CSI volume health (metrics)
  - Volume populator
  - On-going effort: CSI migration
  - On-going effort: Volume expansion
- Targeting Alpha
  - Object Storage API (COSI)
  - Recovering from resize failures
  - Prevent PV leaks when deleting out of order
  - Secret Deletion Protection (“Liens”)

Features in Design/Prototyping

- Non-graceful node shutdown
- VolumeSnapshot namespace transfer
- Control volume mode conversion between source and target PVC
- VolumeGroup and VolumeGroupSnapshot

Cross SIG WG/projects

- Data Protection WG
  - Change block tracking (Design)
- SIG-Apps
  - Auto remove PVCs created by statefulset (Targeting Alpha in 1.23)
- SIG-Node
  - ContainerNotifier (Targeting Alpha in 1.23)
- SIG-API-Machinery
  - in-use protection (Liens) (Targeting Alpha in 1.23)

RESILIENCE
REALIZED

KubeCon | CloudNativeCon
North America 2021

Thank You