openstack.

Agosto 2022

Your ReadWriteMany (RWX) Storage in k8s with Manila CSI

Victoria Martinez de la Cruz
OpenInfra Days México

Agenda

- Presentar k8s storage & CSI
- Cinder CSI, Manila CSI
- OpenStack Manila
- Cómo hacer un deployment con Manila CSI
- Cómo usar (demo)
- Trabajo futuro

Storage en k8s

- Incluso las aplicaciones *stateless* necesitan *storage* persistente
- Los *Physical Volumes* (PV) fueron introducidos (GA) en k8s 1.14
- *Drivers In-tree*
    - requiere que los *vendors* contribuyan a k8s
    - requiere que los *vendors* se alineen a la cadencia de *releases*
    - y otros *contributors* en *upstream* participen de *reviews*, mantenimiento y demás
- Se necesita una forma más flexible para agregar storage drivers a los *container orchestrators*

Container Storage Interface (CSI)

- Desarrollado como un estándar para exponer sistemas de *block storage* y *file storage* a aplicaciones *containerizadas*
- Permite a los *vendors* de *storage* a desarrollar un plugin una única vez y utilizarlo en diferentes *container orchestrators* (CO)
- Más información en Container Storage Interface (CSI) for Kubernetes GA
- La migración de entornos usando plugins in-tree a sus versiones CSI es un esfuerzo que está en desarrollo. Hay una herramienta de migración que puede ser utilizada.

openstack.

Cloud Provider OpenStack CSI drivers

Cinder CSI

Block Storage

Modos RWO y RWX

Soporte para múltiples *backends*

Hard multitenancy (via Keystone)

Manila CSI

File Storage

Modos RWO y RWX

Soporte para múltiples *backends*

Hard multitenancy (via Keystone)

Modos de acceso

- **ReadWriteOnce (RWO)**
  - El volumen puede ser montado como *read-write* en un único nodo
  - Ejemplo: aplicaciones con estado
- **ReadOnlyMany (ROX)**
  - El volumen puede ser montado como *read-only* en múltiples nodos
  - Ejemplo: *backups*
- **ReadWriteMany (RWX)**
  - El volumen puede ser montado como read-write en múltiples nodos
  - Ejemplo: shared file systems
- **ReadWriteOncePod (RWOP)** (new in k8s 1.22)
  - El volumen puede ser montado como read-write por un único pod
  - Ejemplo: manipulación de información sensibles
  - Más información aquí

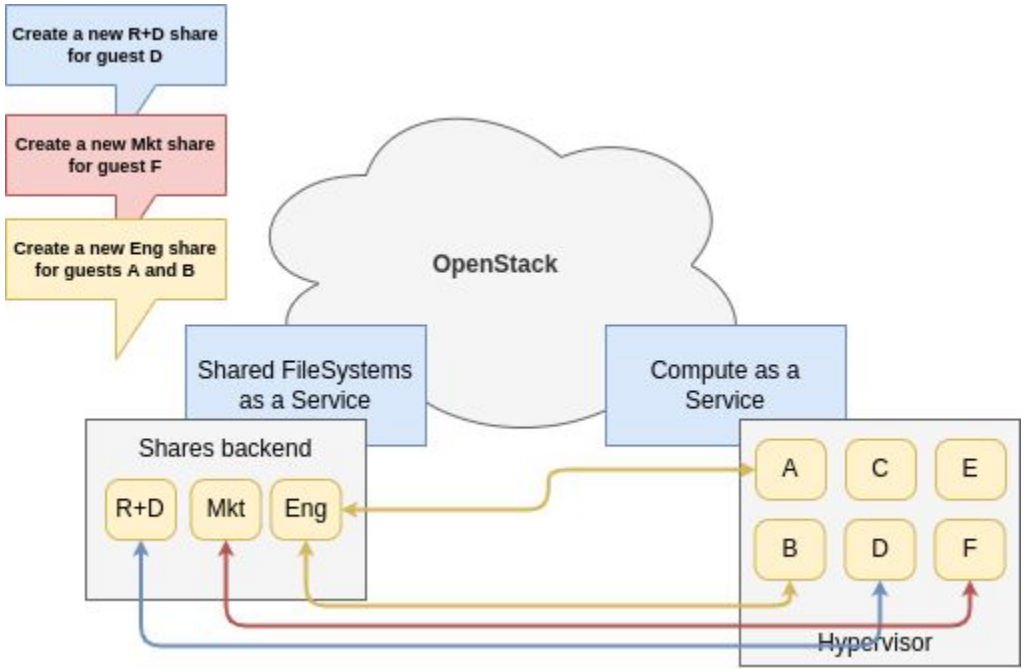
openstack.

OpenStack Manila and Manila CSI

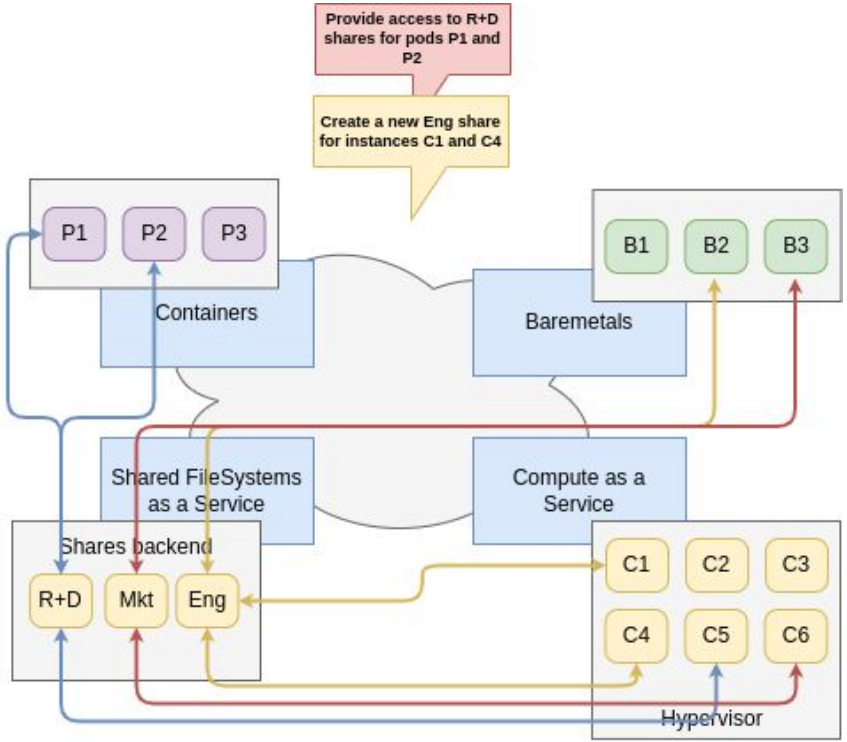
OpenStack Manila y Manila CSI

- *Shared file systems as a service* para OpenStack,
- Soporte para múltiples backends
- En OpenStack, soporta más de 35 *storage backends* (propietarios y *open source*), incluyendo NetApp, Ceph, Dell EMC, Gluster y más
- Soporte para múltiples protocolos
    - En OpenStack soporta NFS, CIFS, GlusterFS, HDFS, CephFS o MAPRFS
    - En k8s, soporta NFS and CephFS
- Manila CSI puede crear, expandir, tomar *snapshots*, restaurar *snapshots* y montar *shares* de Manila

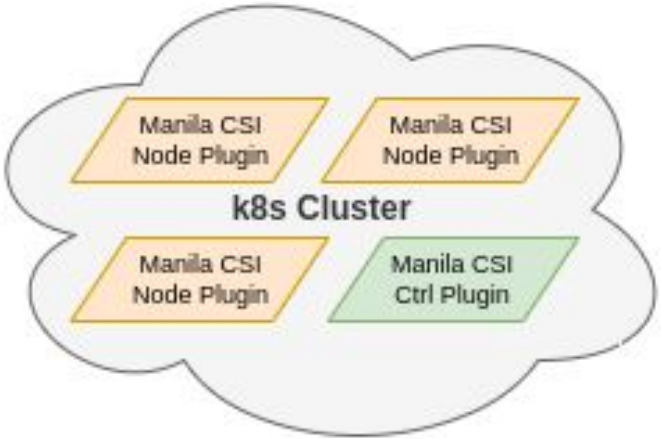
Caso de uso


Create a new R+D share for guest D
Create a new Mkt share for guest F
Create a new Eng share for guests A and B
OpenStack
Shared FileSystems as a Service
Compute as a Service
Shares backend
R+D
Mkt
Eng
A
C
E
B
D
F
Hypervisor

Caso de uso extendido

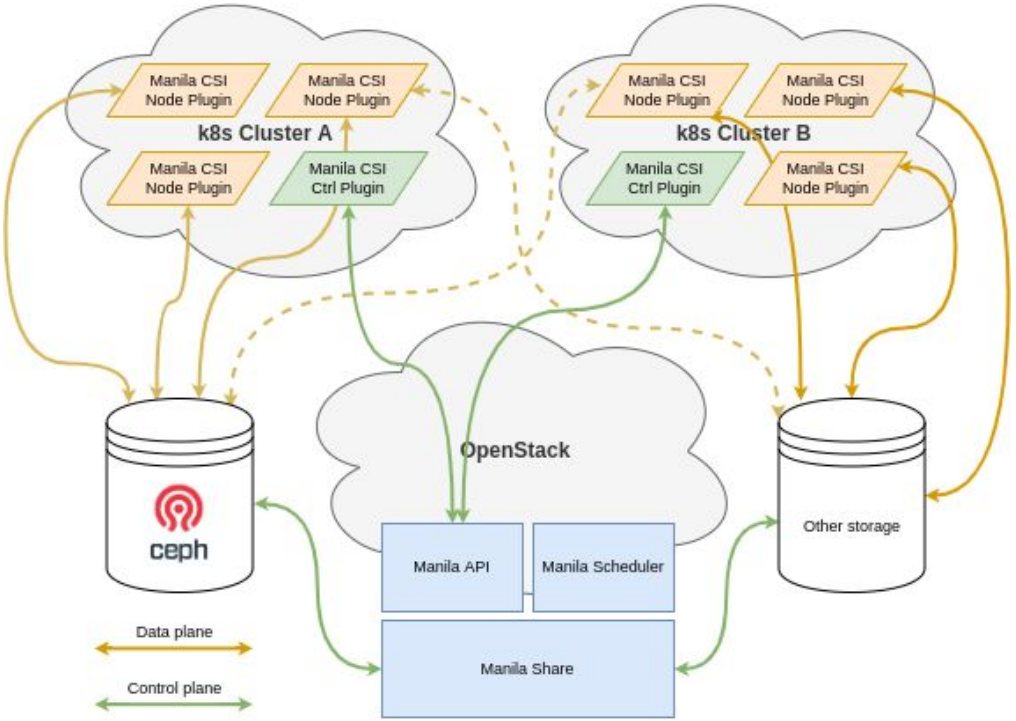


Deployment de referencia de Manila CSI



- Manila CSI Controller Plugin
- Manila CSI Node Plugin funciona como un *proxy*
  - CSI NFS
  - CSI Ceph

Deployment de referencia de Manila CSI



openstack.

Cómo hacer un *deployment* con Manila CSI

Qué necesitaremos

- Manila CSI driver
  - Sidecar containers
  - Roles
  - CRD
  - StateFulSet
  - DaemonSet
- NFS CSI Driver
- StorageClass

Cómo hacer un *deployment* de Manila CSI en k8s

- Deploy Manila CSI driver using helm charts

```
helm repo add cpo https://kubernetes.github.io/cloud-provider-openstack
helm repo update
helm install manila-csi cpo/openstack-manila-csi
```

- Deploy NFS CSI driver

```
helm repo add csi-driver-nfs …/kubernetes-csi/csi-driver-nfs/master/charts
helm install csi-driver-nfs csi-driver-nfs/csi-driver-nfs
```

- Create a StorageClass

  https://github.com/kubernetes/cloud-provider-openstack/tree/master/examples/manila-csi-plugin

Storage Classes

- Definir un StorageClass referenciando a el provisioner manila.csi.openstack.org

```
apiVersion: v1
items:
- apiVersion: storage.k8s.io/v1
  kind: StorageClass
  metadata:
        name: csi-manila-default
  parameters: ...
  provisioner: manila.csi.openstack.org
  reclaimPolicy: Delete
  volumeBindingMode: Immediate
kind: List
metadata: ...
```

openstack.

Cómo hacer un *deployment* de Manila CSI en OpenShift

- Manila CSI driver operator

    … revisa si Manila ha sido instalado. Si Manila está presente en el *deployment*

    1. Instala el driver de Manila CSI
    2. Instala NFS CSI driver
    3. Create un StorageClass para cada *share type*

En OpenShift 4.6+: viene instalado por defecto

openstack.

Cómo usar Manila CSI

PVCs

- Definir un PVC referenciando al StorageClass csi-manila-default que creamos

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: mypvc
spec:
  accessModes:
  - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
  storageClassName: csi-manila-default
```

Demo

Manila CSI RWX demo - https://asciinema.org/a/499321

openstack.

Trabajo futuro y más recursos

Trabajo futuro

- Performance testing
  - I/O tests con FIO
  - Latencia
  - API Scale tests con OpenStack Browbeat
- Backups
  - Usando Velero/Restic y Kanister.io para realizar backups (pueden ver “Your Manila CephFS Share Backups Belong to S3” por Robert Vasek en KubeCon 2022)

Más recursos

- Recursos de la demo
  - https://github.com/vkmc/rxw-storage-k8s-manila-csi
- Storage SIG
  - https://github.com/kubernetes/community/tree/master/sig-storage
  - https://github.com/kubernetes/enhancements/tree/master/keps/sig-storage/625-csi-migration
- "Connecting ecosystems: How Cinder CSI, Ember CSI & Manila CSI leverage OpenStack bits in Kubernetes" by Christian Schwede, Eric Harney, Tom Barron and Gorka Eguileor
- "Dynamic Storage Provisioning of Manila-CephFS Shares on Kubernetes" by Robert Vasek and Ricardo Rocha
- "Your Manila CephFS Share Backups Belong to S3" by Robert Vasek

openstack.

¡Gracias!