



COPENHAGEN

What is the Container Storage Interface (CSI)?

May 1, 2018

Saad Ali

Senior Software Engineer, Google

Co-Author of CSI

 github.com/saad-ali

 twitter.com/the_saad_ali



Agenda

- Background
- Introduction to CSI
- CSI with Kubernetes
- Current Status
- Q/A



What's the problem?

Kubernetes “In-tree” Volume Plugins are awesome =)

- Powerful abstraction for file and block storage.
- Automate provisioning, attaching, mounting, and more!
- Storage portability via PV/PVC/StorageClass objects



What's the problem?

Kubernetes “In-tree” Volume Plugins are painful =(

- Painful for Kubernetes Developers
 - Testing and maintaining external code
 - Bugs in volume plugins affect critical Kubernetes components
 - Volume plugins get full privileges of kubernetes components (kubelet and kube-controller-manager)
- Painful for Storage Vendors
 - Dependent on Kubernetes releases
 - Source code forced to be open source



We're not alone!

Most Container orchestration (CO) systems want to be able to offer users as many storage systems as possible.



Most Storage providers (SP) want to be able to make their system available to as many users as possible with as little work as possible.



Solution: CSI!

CSI makes Kubernetes volume layer truly extensible.



What is CSI?

CSI is:

- Interface that enables 3 main use cases:
 - Create/delete volume
 - Attach/detach volume to a node
 - Mount/unmount volume to a node
- Evolving
 - Example Create/delete snapshots functionality coming soon
- Open Source
 - Contributions welcome!

CSI is not:

- Packaging, deployment, monitoring, etc. of a plugin.
- Defining "grades of storage" or a "storage catalogue"



CSI Overview

- Interface defined using gRPC
- Segmented into three gRPC “services”:
 - Controller
 - Maybe run anywhere in the cluster.
 - Node
 - Must run only on the node where workload referencing volume is scheduled.
 - Identity
 - Plugin information



CSI Overview

Let's talk about naming:

- Create/delete volume
- Attach/detach volume
- Mount/unmount volume



CSI Overview

Let's talk about naming:

- Create/delete volume
- ~~Attach/detach volume~~
- ~~Mount/unmount volume~~

ControllerPublishVolume/ControllerUnpublishVolume
NodePublishVolume/NodeUnpublishVolume



CSI Overview

Controller service

- ControllerGetCapabilities
- CreateVolume
- DeleteVolume
- ControllerPublishVolume
- ControllerUnpublishVolume
- ListVolumes
- GetCapacity
- ValidateVolumeCapabilities

Node service

- NodeGetCapabilities
- NodePublishVolume
- NodeUnpublishVolume
- NodeStageVolume
- NodeUnstageVolume
- NodeGetId

Identity service

- GetPluginCapabilities
- GetPluginInfo
- Probe





CSI with Kubernetes

How to use a CSI Volume?

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: fast-storage
provisioner: com.example.csi-driver
parameters:
  type: pd-ssd
  csiProvisionerSecretName: mysecret
  csiProvisionerSecretNamespace: mynamespace
```

```
-----
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: my-request-for-storage
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 5Gi
  storageClassName: fast-storage
```

```
kind: Pod
apiVersion: v1
metadata:
  name: my-pod
spec:
  containers:
    - name: my-frontend
      image: nginx
      volumeMounts:
        - mountPath: "/var/www/html"
          name: my-csi-volume
  volumes:
    - name: my-csi-volume
      persistentVolumeClaim:
        claimName: my-request-for-storage
```

What happens to the “in-tree” volumes?

- Design in-progress to silently proxy to CSI drivers
- Volumes that must move to CSI
 - Cloud provider dependent volumes
 - Remote persistent storage volumes



Deploying CSI driver on Kubernetes Cluster

Up to Storage Vendor. Typically:

- Containerized driver deployed via a StatefulSet and DaemonSet.
- Easy as `kubectl create -f storageprovider.yaml`

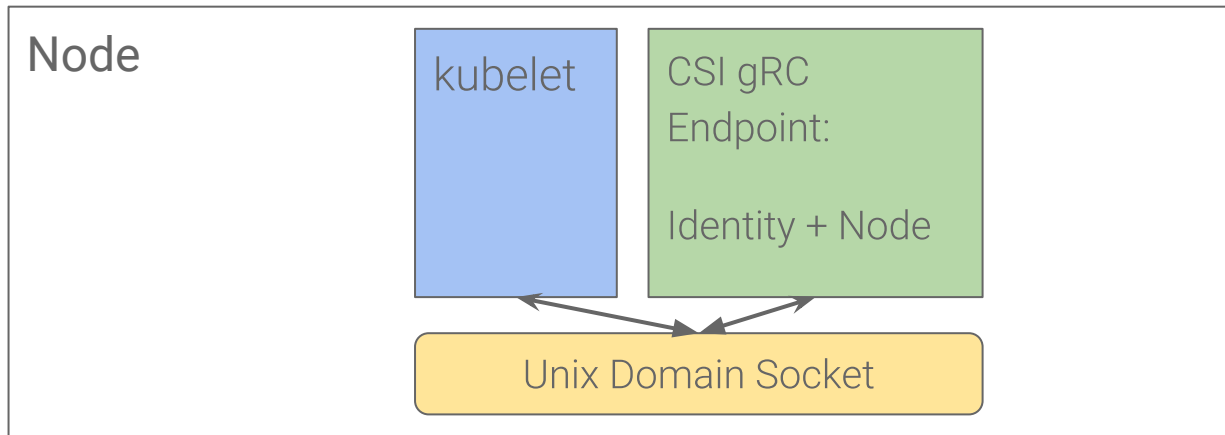


What does a storage vendor have to do?

- Suggested:
 - Write a containerized CSI driver
 - Write YAML to deploy driver along with sidecar containers (provided by Kubernetes dev team).



Node to CSI Communication



- UDS at known location. Future: plugin registration.
- Mount (Setup) calls from kubelet maps to NodePublishVolume
- Driver handling Node calls must run on every node.

Master to CSI Communication

Problem:

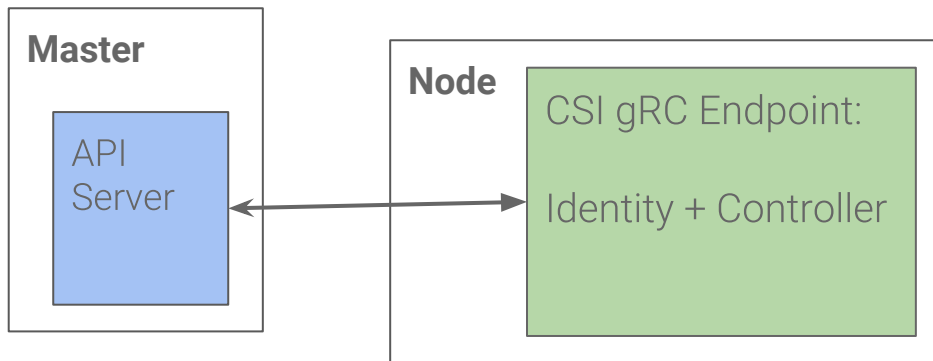
- Master may not be able to run 3rd party code on master (e.g. GKE)
 - Assume CSI service can't run locally.
- CSI endpoint discovery?
 - k8s Service?
- Secure Communication between k8s and CSI endpoint
 - Complicated authn system?



Master to CSI Communication

Solution:

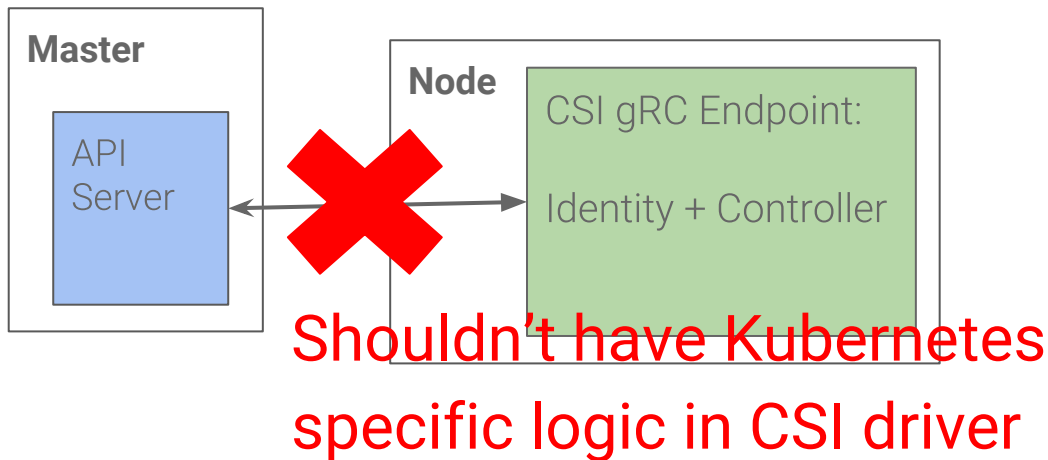
- CSI driver monitors Kubernetes API and triggers CSI operations as needed



Master to CSI Communication

Solution:

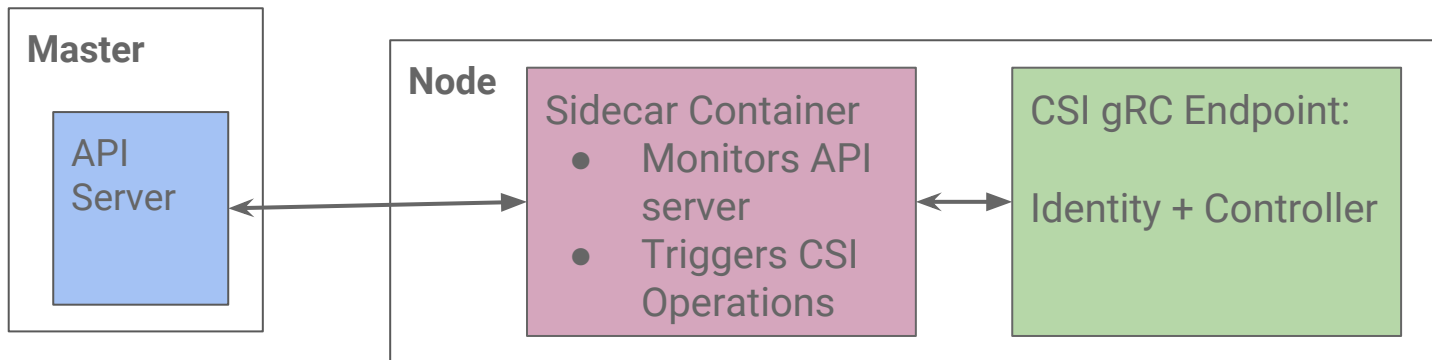
- CSI driver monitors Kubernetes API and triggers CSI operations as needed



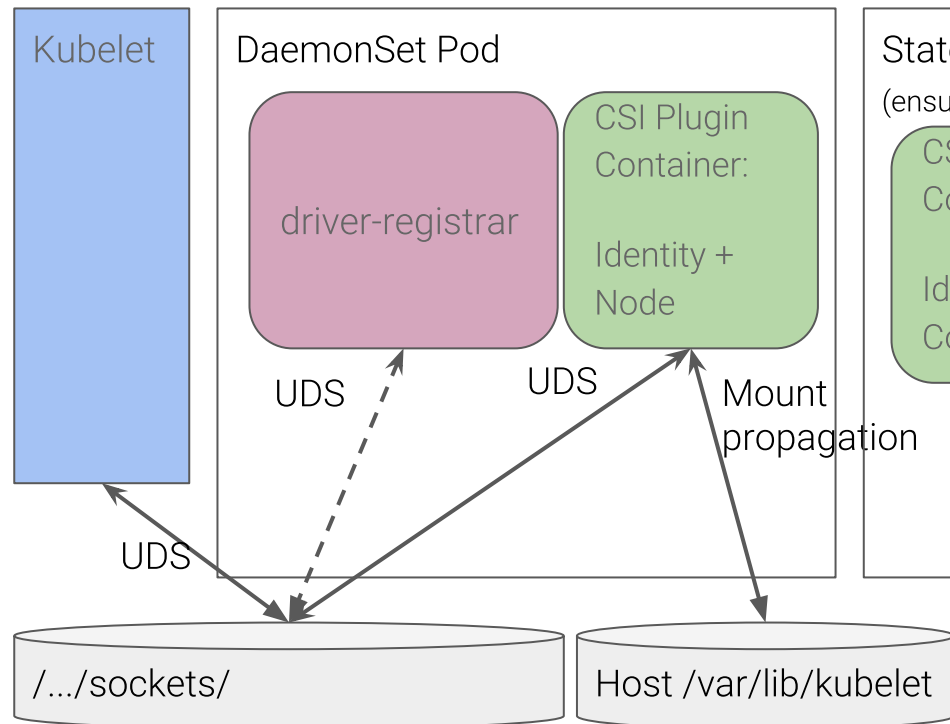
Master to CSI Communication

Solution:

- CSI driver monitors Kubernetes API and triggers CSI operations as needed

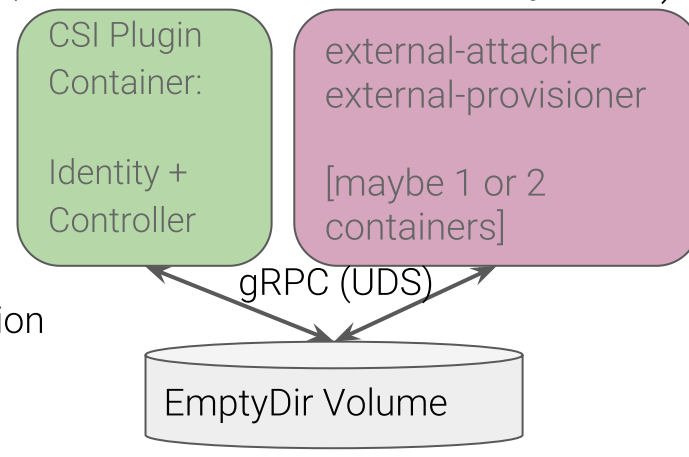


Node

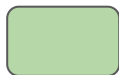
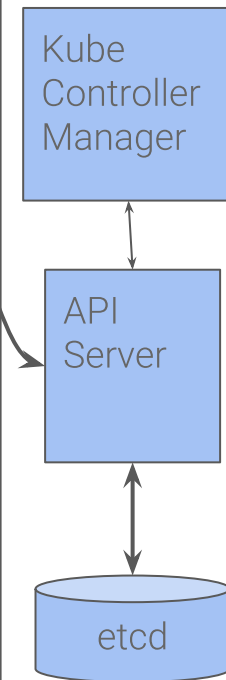


StatefulSet (replica:1)

(ensures no more than 1 instance running at a time)



Master



Third Party Storage Vendor Container



Sidecar containers by Kubernetes Team

UDS - Unix Domain Socket

Status of CSI

- Implementations Started
 - Kubernetes introduced alpha support in v1.9, beta in v1.10
 - Mesos 1.5 supports CSI (experimental)
 - Cloud Foundry has experimental support for CSI in recent versions
 - Docker is planning support



CONTAINER
STORAGE
INTERFACE

<https://github.com/container-storage-interface>

- May 2017
 - Spec published in GitHub repo
- December 2017
 - v0.1 release
- February 2017
 - v0.2 release



CONTAINER
STORAGE
INTERFACE

github.com/container-storage-interface

Questions?

Get Involved!

- Container Storage Interface Community
 - github.com/container-storage-interface/community
 - Meeting every week, Wednesdays at 9 AM (PT)
 - Mailing list:
 - container-storage-interface-community@googlegroups.com
- Kubernetes Storage Special-Interest-Group (SIG)
 - github.com/kubernetes/community/tree/master/sig-storage
 - Meeting every 2 weeks, Thursdays at 9 AM (PST)

Contact me:

- Saad Ali, Google
- github.com/saad-ali
- twitter.com/the_saad_ali