



Containercon  
CHINA 中国

CLOUDOPEN

THINK OPEN

开放性思维

# Disaster Recovery and Data Protection for Kubernetes Persistent Volumes

Xing Yang, Principal Architect, Huawei

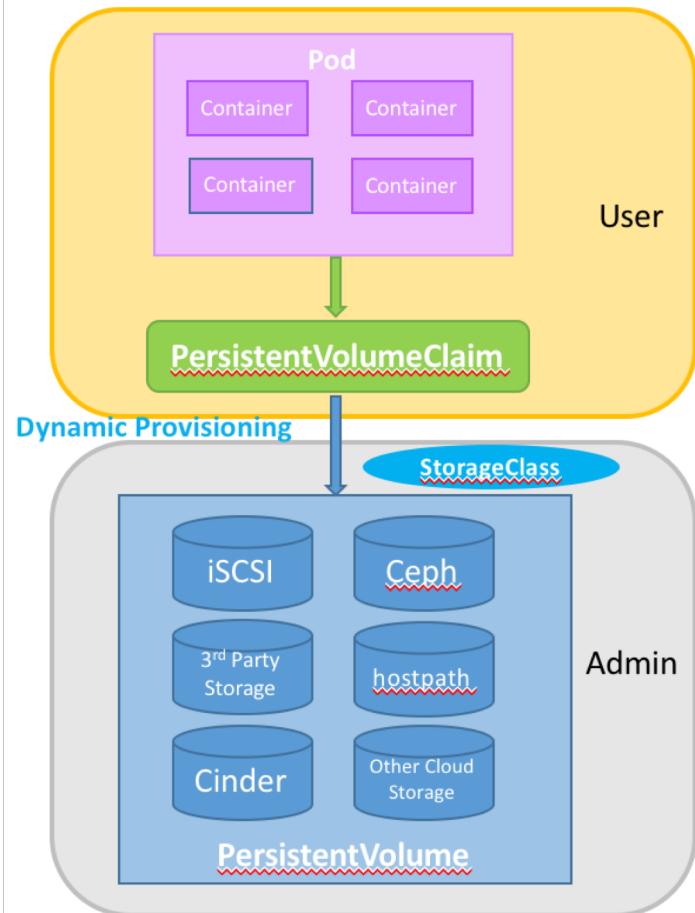


# Agenda

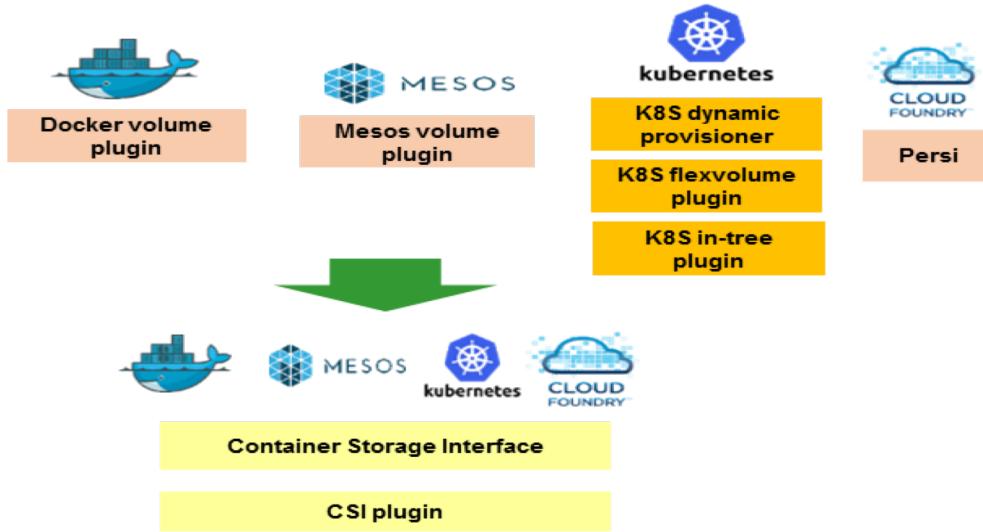
- Kubernetes Persistent Volumes and CSI
- Why OpenSDS for Kubernetes and CSI
- OpenSDS Overview
- Provision and Manage Persistent Volumes
- Disaster Recovery for Persistent Volumes
- Data Protection for Persistent Volumes
- OpenSDS Roadmap for Aruba and Bali Release
- OpenSDS Community
- Demo

# Kubernetes Persistent Volumes

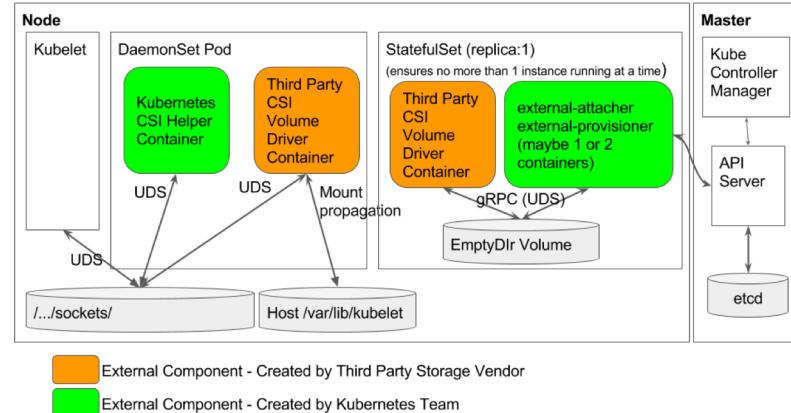
- A PersistentVolume (PV) is a piece of storage in the cluster that has been provisioned by an administrator.
- A PV can be provisioned statically or dynamically.
- A PersistentVolumeClaim (PVC) is a request for storage by a user through a StorageClass.
- A StorageClass provides a way for administrators to describe the “classes” of storage they offer. Different classes might map to different quality-of-service levels (or “profiles”) in other storage systems.
- A StorageClass needs to specify a provisioner for dynamic provisioning.



# Container Storage Interface (CSI)

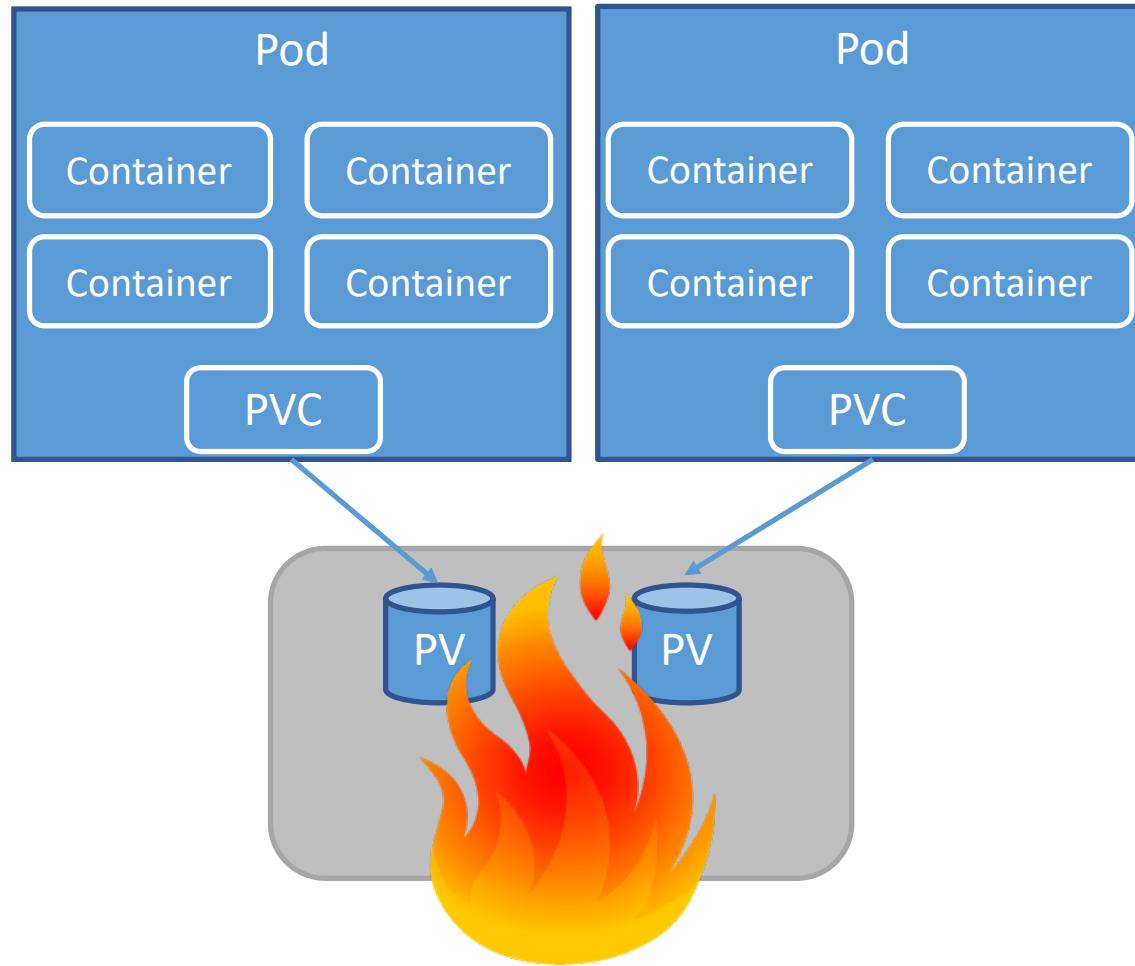


CSI is an industry standard defined to enable storage vendors to develop a plugin once and have it work across a number of container orchestration systems.



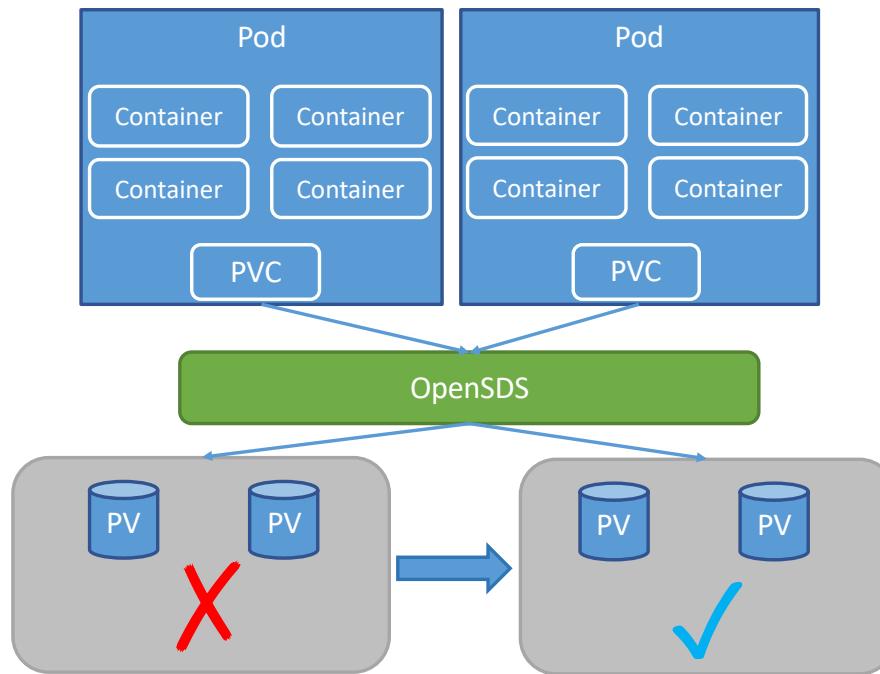
Source: <https://github.com/kubernetes/community/blob/master/contributors/design-proposals/storage/container-storage-interface.md>

# What Happens When Disaster Strikes



# Why OpenSDS for Kubernetes and CSI

- Storage functionalities in Kubernetes and CSI are still evolving.
- OpenSDS can provide additional storage functionalities such as data protection and disaster recovery.
- Provide unified control for traditional cloud and cloud native environment.



# OpenSDS Overview - Core Projects

## SUSHI

The Northbound Plug-ins Project

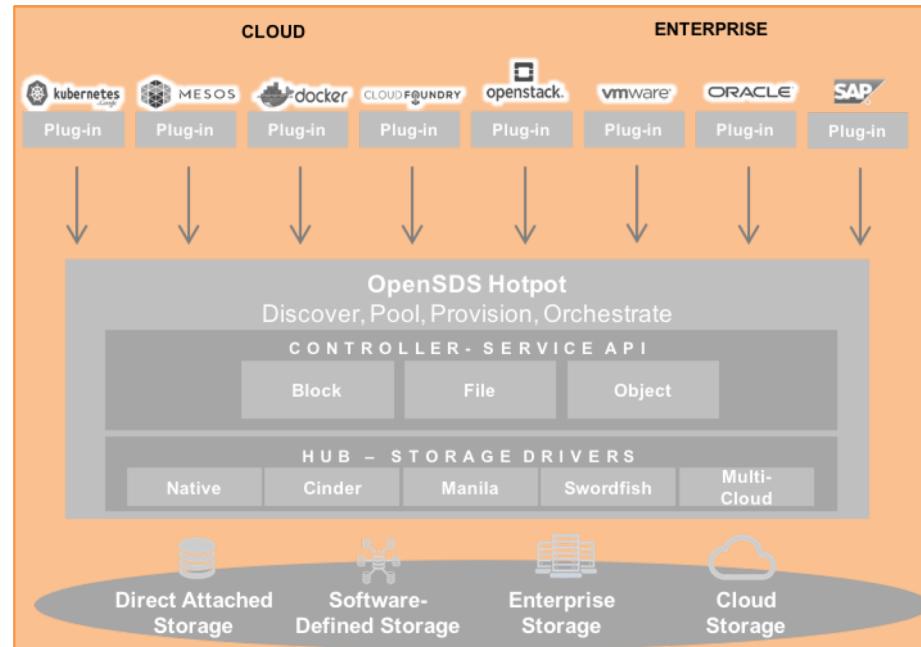
Common plug-ins to enable OpenSDS storage services for cloud and application frameworks



## HOTPOT

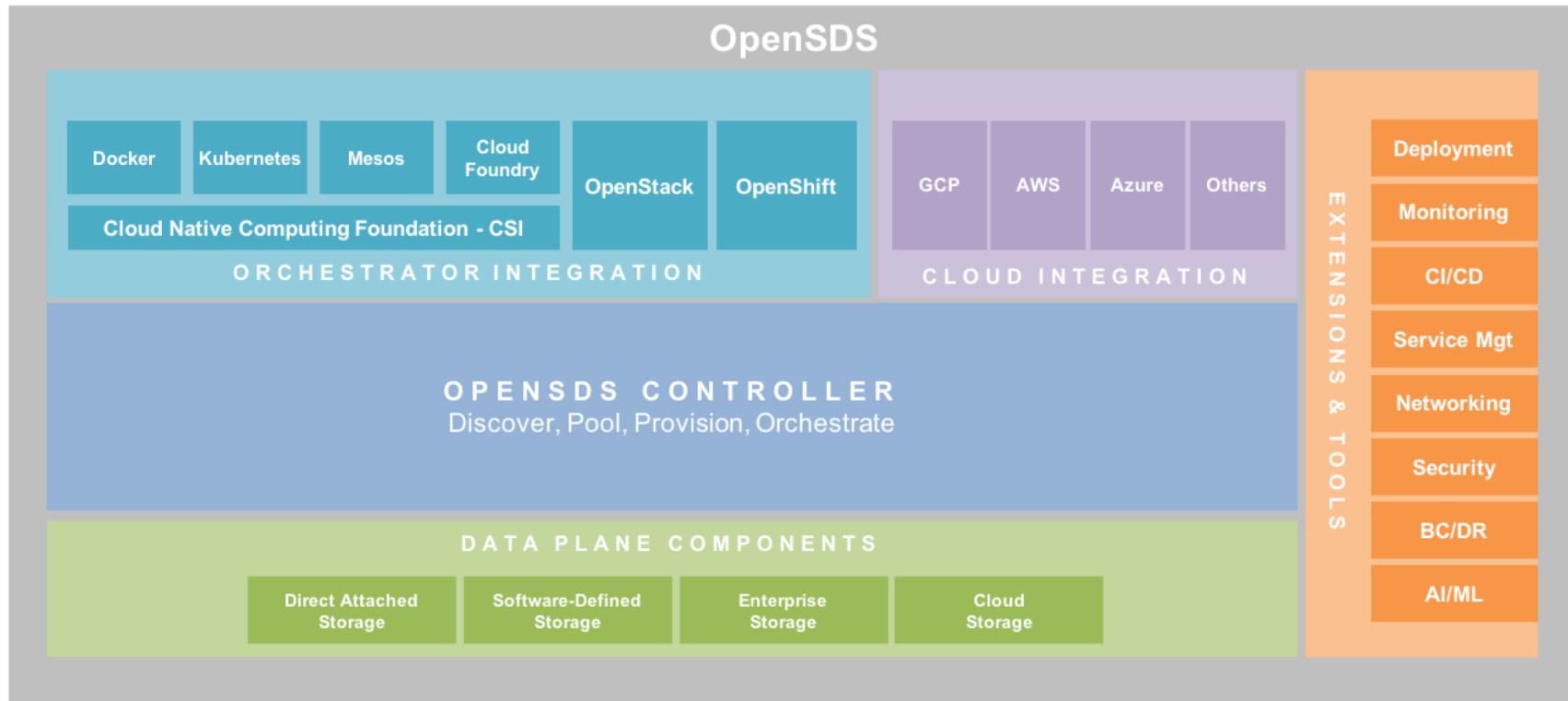
The Storage Controller Project

Single control for block, file, and object services across storage on premise and in clouds

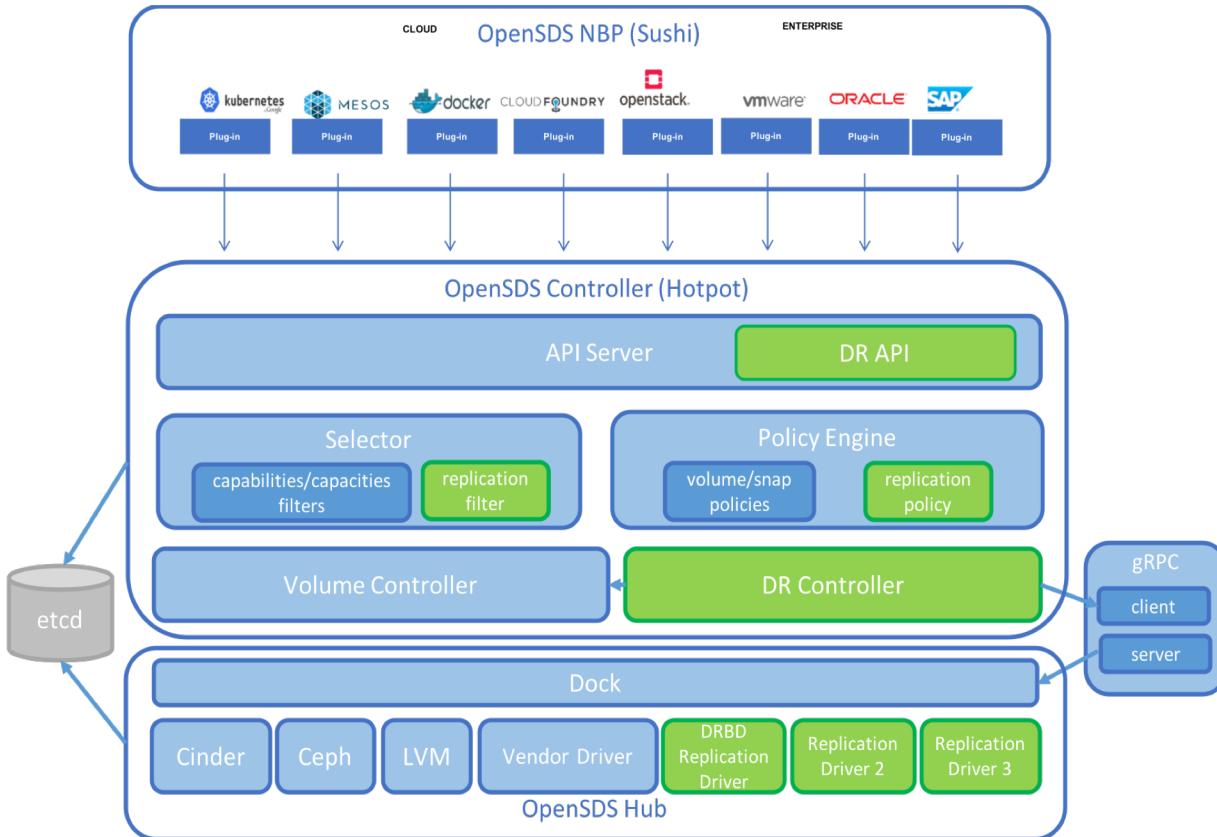


***OpenSDS is an Open Source Project  
under the Linux Foundation***

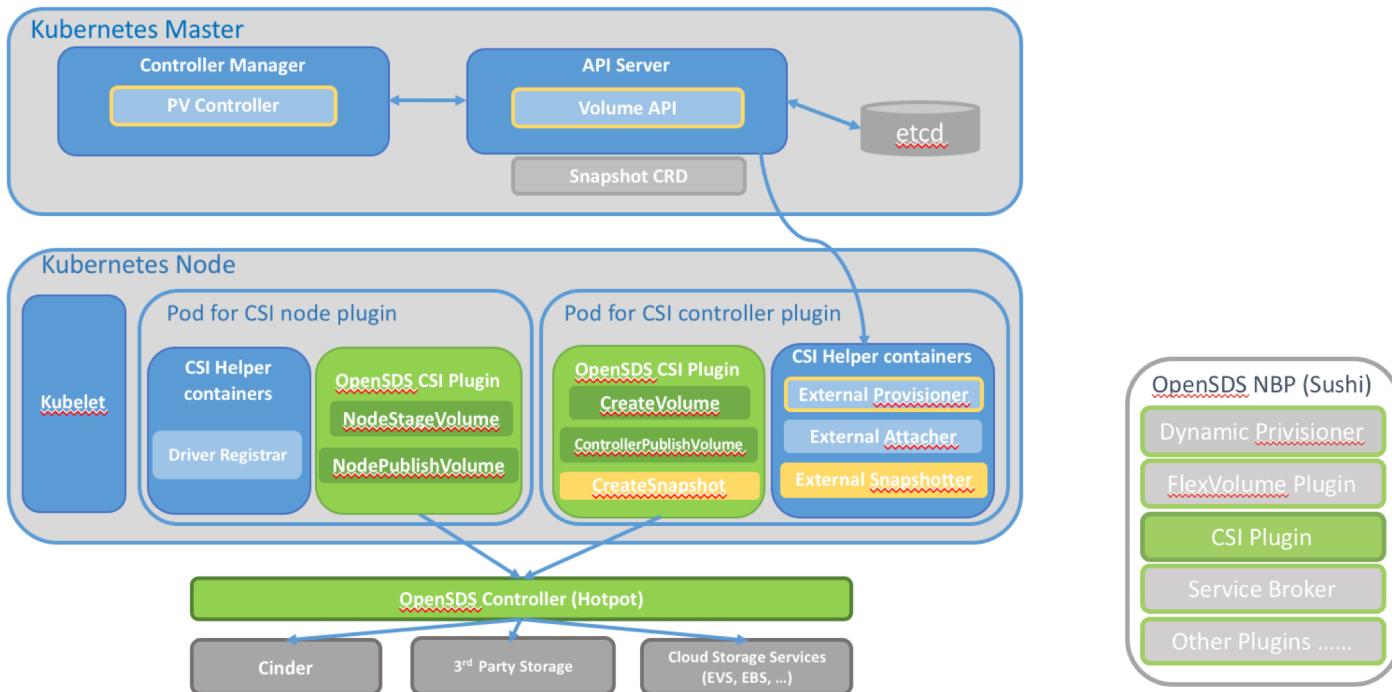
# OpenSDS Overview - Project Framework



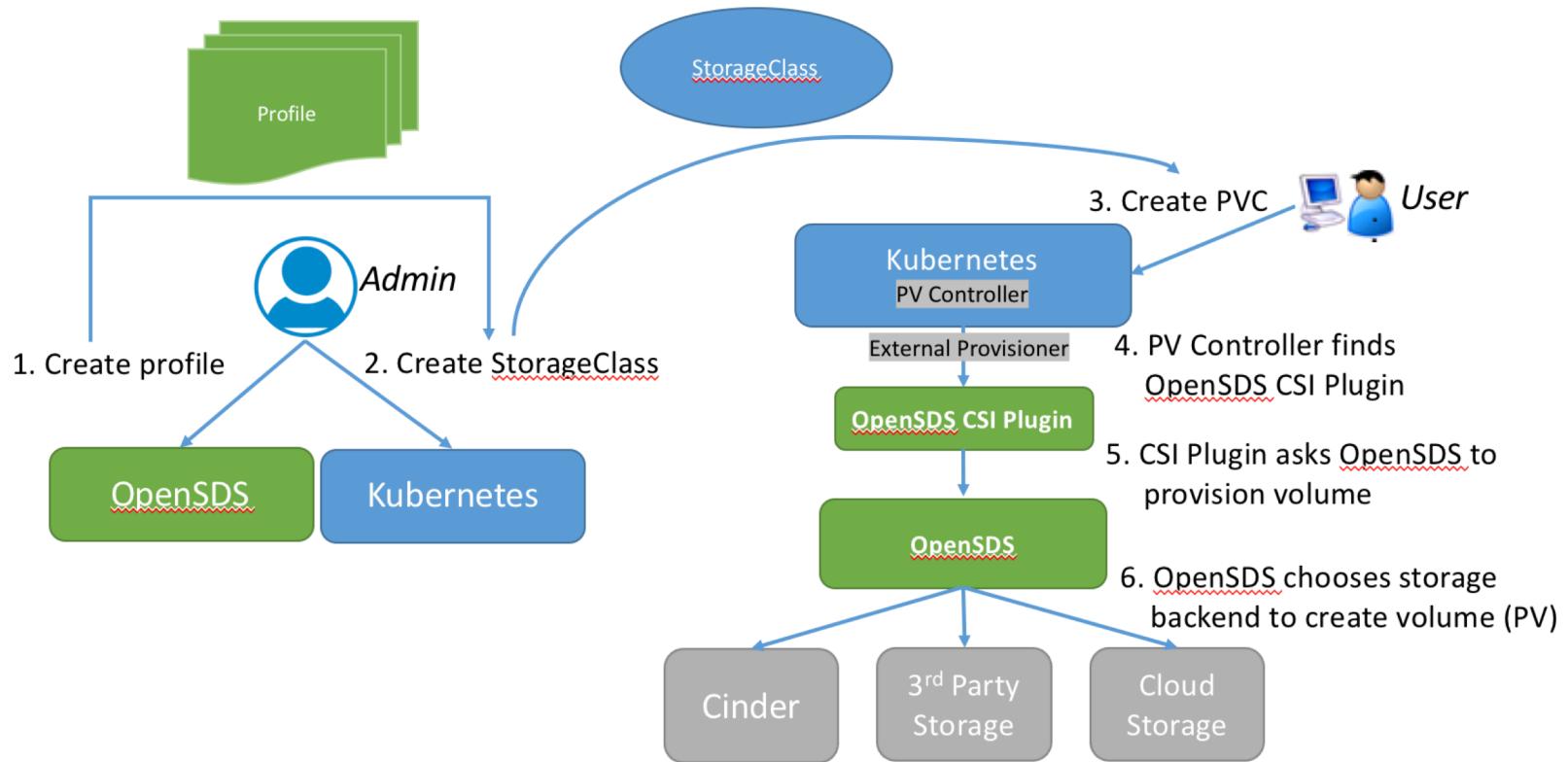
# OpenSDS Overview - Architecture



# Provision and Manage Persistent Volumes using OpenSDS



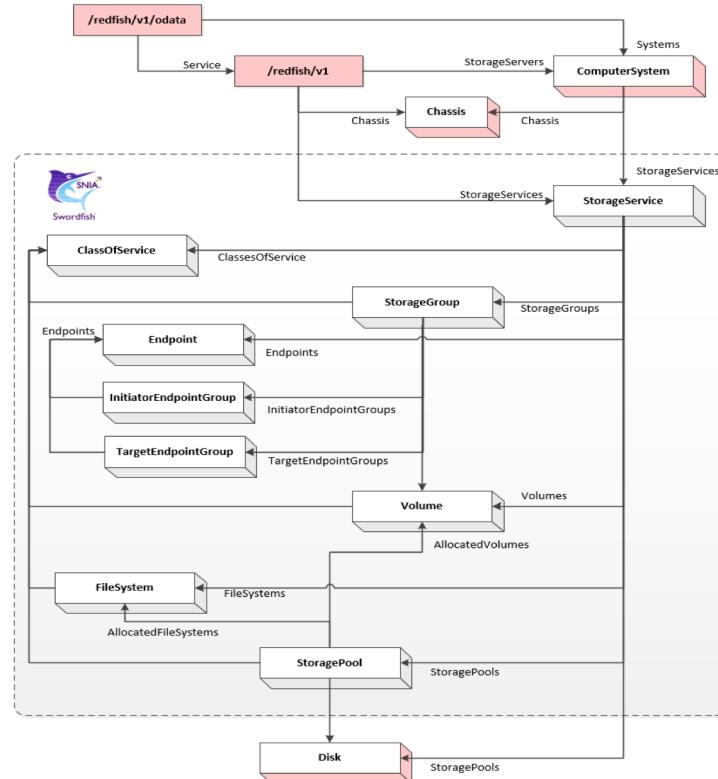
# Profiles: Mapping OpenSDS Profile to K8S StorageClass



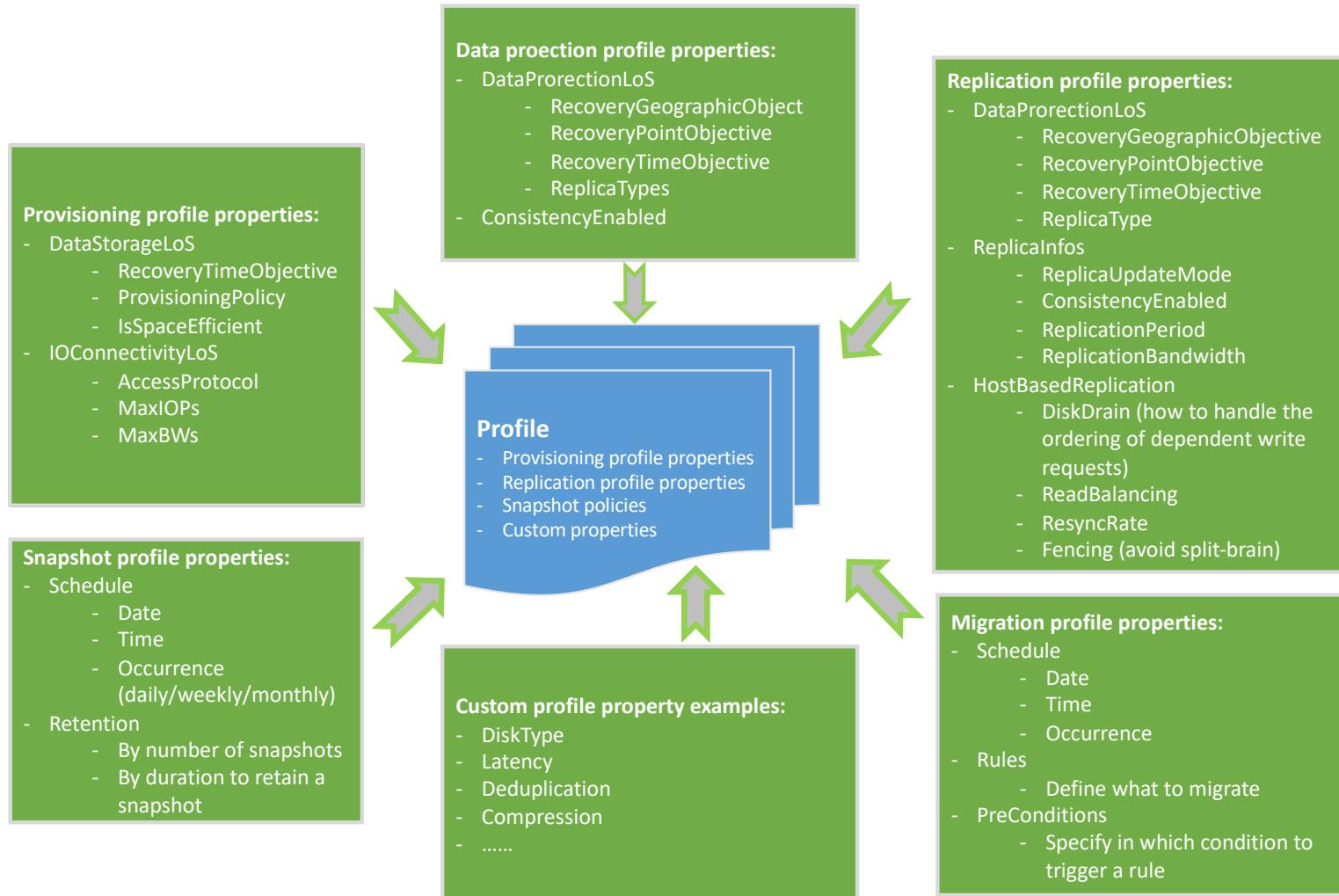
# Profiles: Policy Driven SPDM



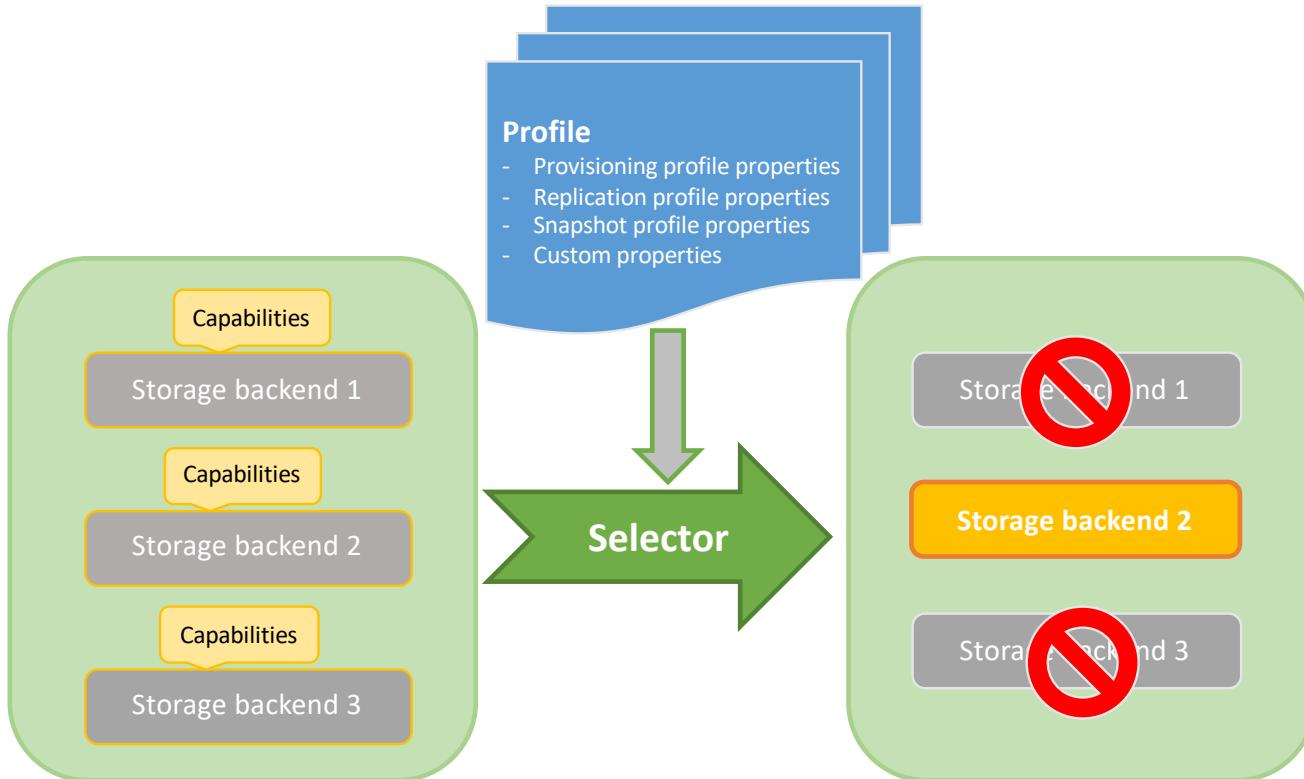
- OpenSDS profile is based on Swordfish specification.
- The SNIA Swordfish™ specification helps to provide a unified approach for the management of storage and servers in hyperscale and cloud infrastructure environments, supported by multiple storage vendors.
- An extension of the DMTF (Distributed Management Task Force) Redfish specification.
  - Redfish is designed by the DMTF's Scalable Platforms Management Forum (SPMF) to create and publish an open industry standard specification and schema for management of scalable platform hardware. It is a RESTful interface over HTTPS in JSON format based on OData v4.



# Profiles: Definitions



# Profiles: Mapping Profiles to Capabilities



# Provision: StorageClass with Profile Parameter

## HighPerformanceSC.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: opensds-csi-high-performance-sc
provisioner: csi-opensdsplugin
parameters:
  profile: High-Performance
```

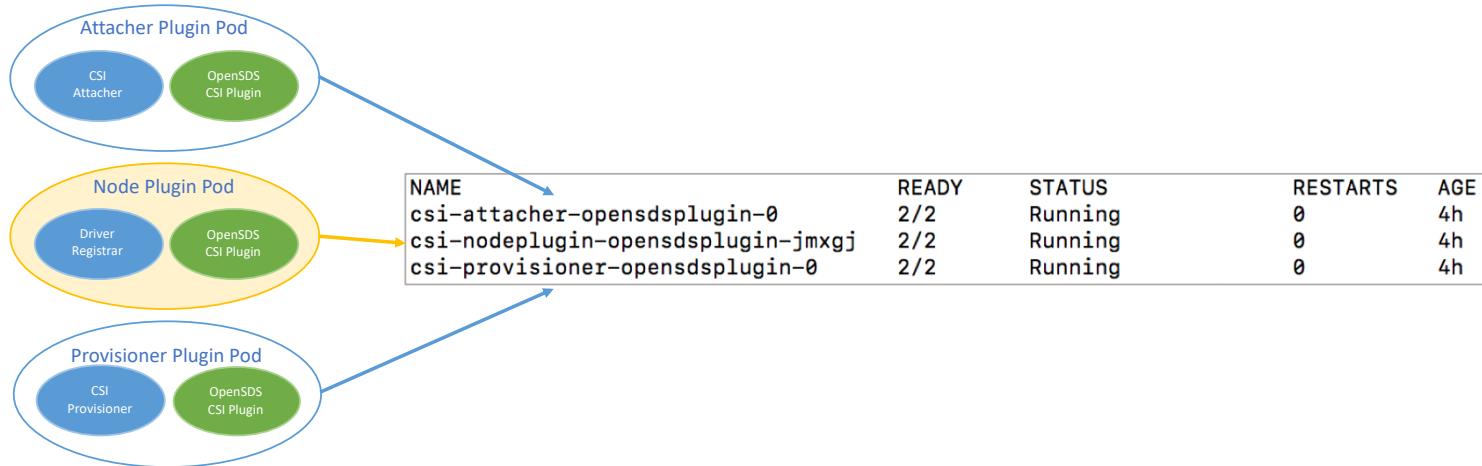
*Note: profile parameter can be profile id or name*

## HighPerformancePVC.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: opensds-csi-high-performance-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: opensds-csi-high-performance-sc
```

# Provision: Running OpenSDS CSI Plugin

- Create OpenSDS CSI plugin pods:  
`kubectl create -f csi/server/deploy/kubernetes`
- Three pods can be found by `kubectl get pod`:



# Provision: Using OpenSDS Volume

- Create nginx application  
`kubectl create -f  
csi/server/examples/kubernetes/nginx.yaml`
- An OpenSDS volume is mounted at  
`/var/lib/www/html.`  
`docker exec -it <nginx container id> /bin/bash`

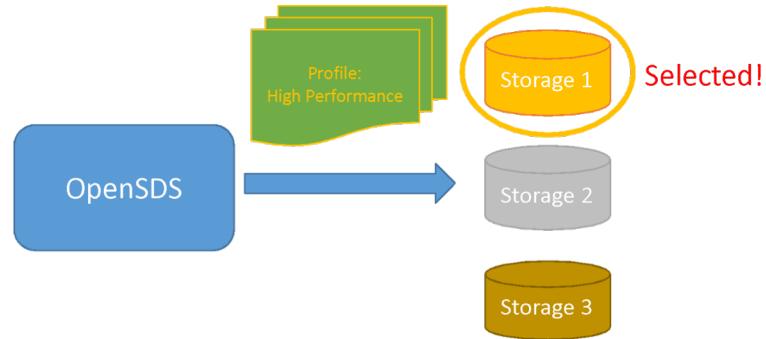
```
root@nginx:/# mount | grep html  
/dev/sda on /var/lib/www/html type ext4 (rw,relatime,data=ordered)
```

nginx.yaml

```
apiVersion: v1  
kind: Pod  
metadata:  
  name: nginx  
spec:  
  containers:  
    - image: nginx  
      imagePullPolicy: IfNotPresent  
      name: nginx  
      ports:  
        - containerPort: 80  
          protocol: TCP  
      volumeMounts:  
        - mountPath: /var/lib/www/html  
          name: csi-data-opensdsplugin  
  volumes:  
    - name: csi-data-opensdsplugin  
      persistentVolumeClaim:  
        claimName: opensds-csi-high-performance-pvc  
        readOnly: false
```

# Disaster Recovery: Replication Profile

- RecoveryTimeObjective
- RecoveryPointObjective
- RecoveryGeographicObjective
- ReplicaType
  - Mirror
- ReplicationUpdateMode
  - Sync, Async, Active, Adaptive
- ConsistencyEnabled
- ReplicationPeriod
- ReplicationBandwidth



# Disaster Recovery: Replication Example

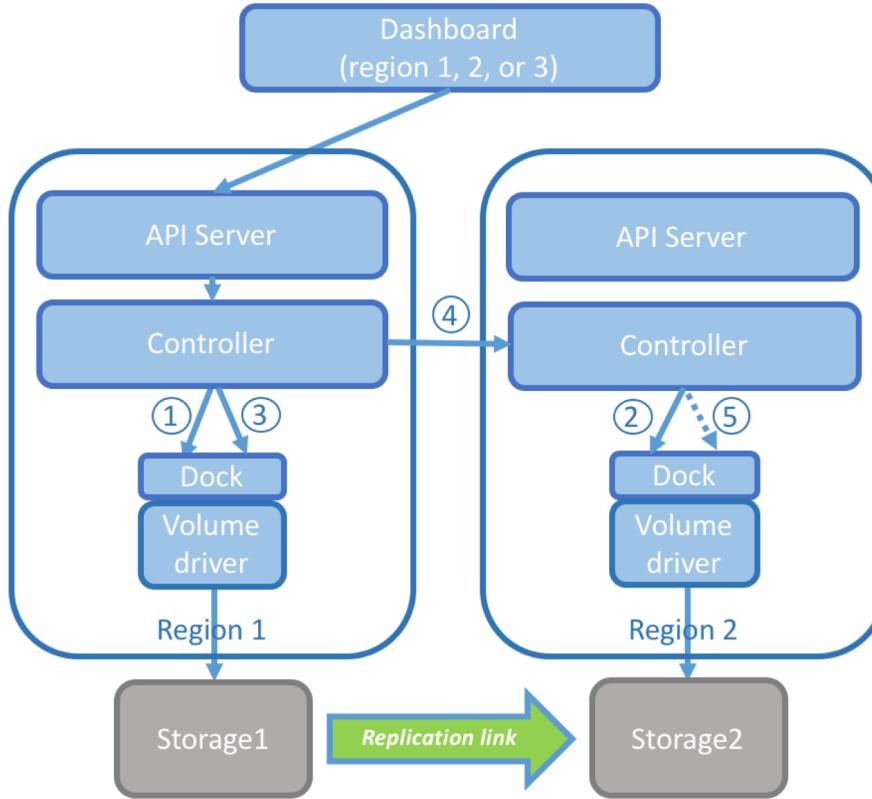
## ReplicationSC.yaml

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: opensds-csi-replication-sc
provisioner: csi-opensdsplugin
parameters:
  profile: replication-profile
enableReplication: "true"
```

## ReplicationPVC.yaml

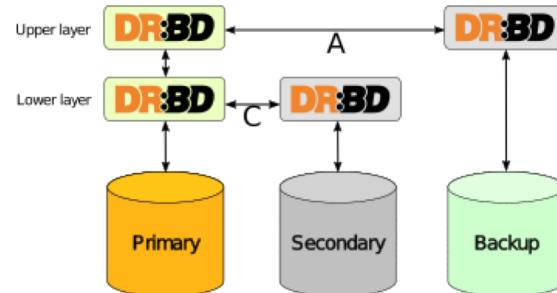
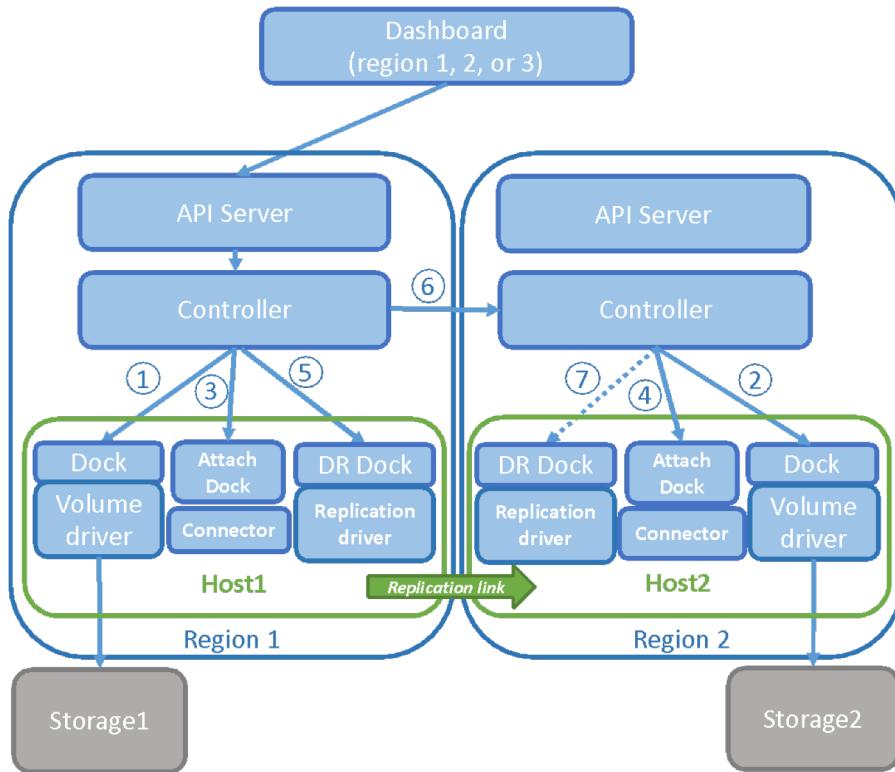
```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: opensds-csi-replication-pvc
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: opensds-csi-replication-sc
```

# Disaster Recovery: Array-based Replication



- Create source and target volume
- Create replication on the arrays

# Disaster Recovery: Host-based Replication



Source: <http://docs.linbit.com/docs/users-guide-9.0/>

- Create source and target volume
- Attach volumes
- Create replication on the hosts

# Disaster Recovery: Replication Functions

Create Replication:  
`osdsctl replication create <primary volume id> <secondary volume id> [flags]`

Flags:

<code>-d, --description string</code>	the description of created replication
<code>-h, --help</code>	help for create
<code>-n, --name string</code>	the name of created replication
<code>-p, --primary_driver_data string</code>	the primary replication driver data of created replication
<code>-m, --replication_mode string</code>	the replication mode of created replication, value can be sync/async
<code>-t, --replication_period int</code>	the replication period of created replication, the value must be greater than 0
<code>-s, --secondary_driver_data string</code>	the secondary replication driver data of created replication

Enable Replication:  
`osdsctl replication enable <replication id>`

Disable Replication:  
`osdsctl replication disable <replication id>`

Failover Replication:  
`osdsctl replication failover <replication id> [flags]`

Flags:

<code>-a, --allow_attached_volume</code>	whether allow attached volume when failing over replication
<code>-h, --help</code>	help for failover
<code>-s, --secondary_backend_id string</code>	the secondary backend id of failover replication

## Failover Replication (failback)

Create Replication

Enable Replication

Delete Replication

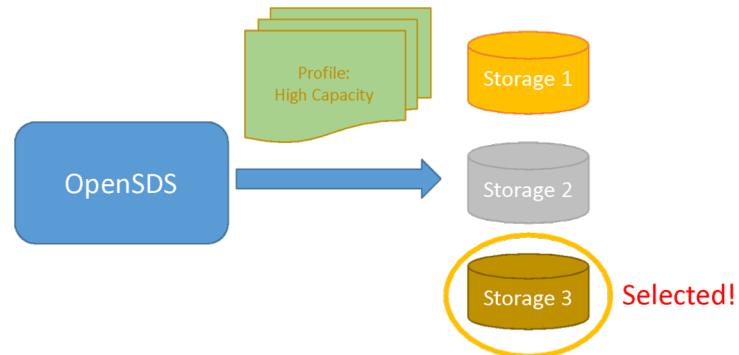
Disable Replication

Show Replication

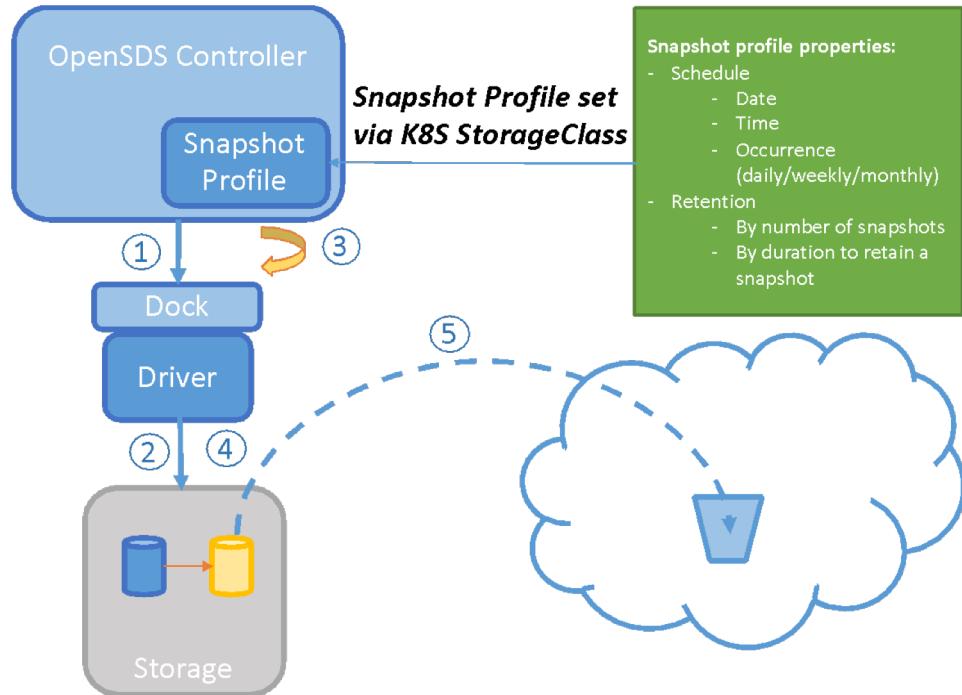
List Replications

# Data Protection: Snapshot Profile

- Schedule
  - Date
  - Time
  - Occurrence (daily/weekly/monthly)
- Retention
  - By number of snapshots
  - By duration to retain a snapshot
- Topology
  - Where to upload snapshot



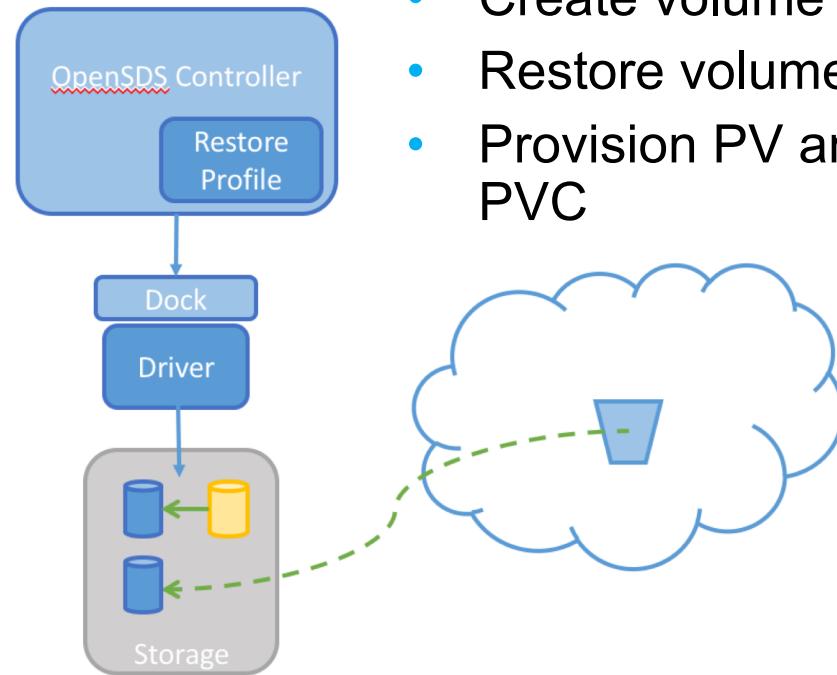
# Data Protection: Periodic Snapshotting



- Take snapshots periodically based on snapshot profile
- Upload snapshots to object store on-premise or in the cloud

# Data Protection: Restore

- Create volume from snapshot
- Restore volume from backup
- Provision PV and bind with PVC



# OpenSDS Roadmap v0.14

**2017H2**

**ZEALAND**

**Storage For  
Kubernetes**

- Kubernetes FlexVolume
- Vol CRUD
- Standalone Cinder Integration
- CSI Support
- Ceph, LVM

**2018H1**

**ARUBA**

**Storage  
Orchestration**

- OpenStack
- Replication Array-Based, Host-Based
- Dashboard
- Virtual Pools
- Storage Profiles
- NVMeoF preview
- Enumeration
- Block Storage
  - Ceph
  - LVM
  - IBM: XIV, Storwize, SVC
  - Huawei: Dorado

**2018H2**

**BALI**

**Storage  
Multi-Cloud**

- Data Migration Offline, Online\*
- Monitoring
- Multi-OpenStack
- S3 Object
- Multi-Cloud Control
- NVMeoF
- Storage Groups Snapshots, Replication
- CSI Mesos\*, Docker\*
- Swordfish Dell-EMC, NetApp

**2019H1**

**CAPRI**

**Storage  
Intelligence**

- Analytics
- Lifecycle
- Data Protection
- File Share

**2019H2++**

- Performance
- Optimization
- Tiering
- Security
- Sharing
- Networking
- SCM

# Governance

## Technical Steering Committee



**Steven Tan, Chairman**  
Huawei, VP & CTO Cloud Solution



**Rakesh Jain, Vice-Chair**  
IBM, Research Engineer and Architect



**Allen Samuels**  
Western Digital, R&D Engineering Fellow



**Anjaneya "Reddy" Chagam**  
Intel, Chief SDS Architect



**Jay Bryant**  
Lenovo, Cloud Storage Lead

## End-User Advisory Committee



**Cosimo Rossetti**  
Vodafone, Lead Storage Architect



**Yusuke Sato**  
Yahoo Japan, Infrastructure Lead



**Kei Kusunoki**  
NTT Communications, Storage Architect



**Yuji Yazawa**  
Toyota ITC, Group Lead

# OpenSDS Community

## Supporting Organizations

An industry-wide open source project for software-defined storage management



OUR NEWEST MEMBER

LINBIT

# Demo

- Array-based replication: Failover storage provisioned by OpenSDS CSI plugin



admin

.....

Login

# Join Us

- Repos: <https://github.com/opensds>
- Slack: <https://opensds.slack.com>
- Mailing list: <https://lists.opensds.io>
- Weekly meetings: <https://github.com/opensds/design-specs/blob/master/README.md#opensds-technical-meetings>



LINUXCON

containercon



CLOUDOPEN

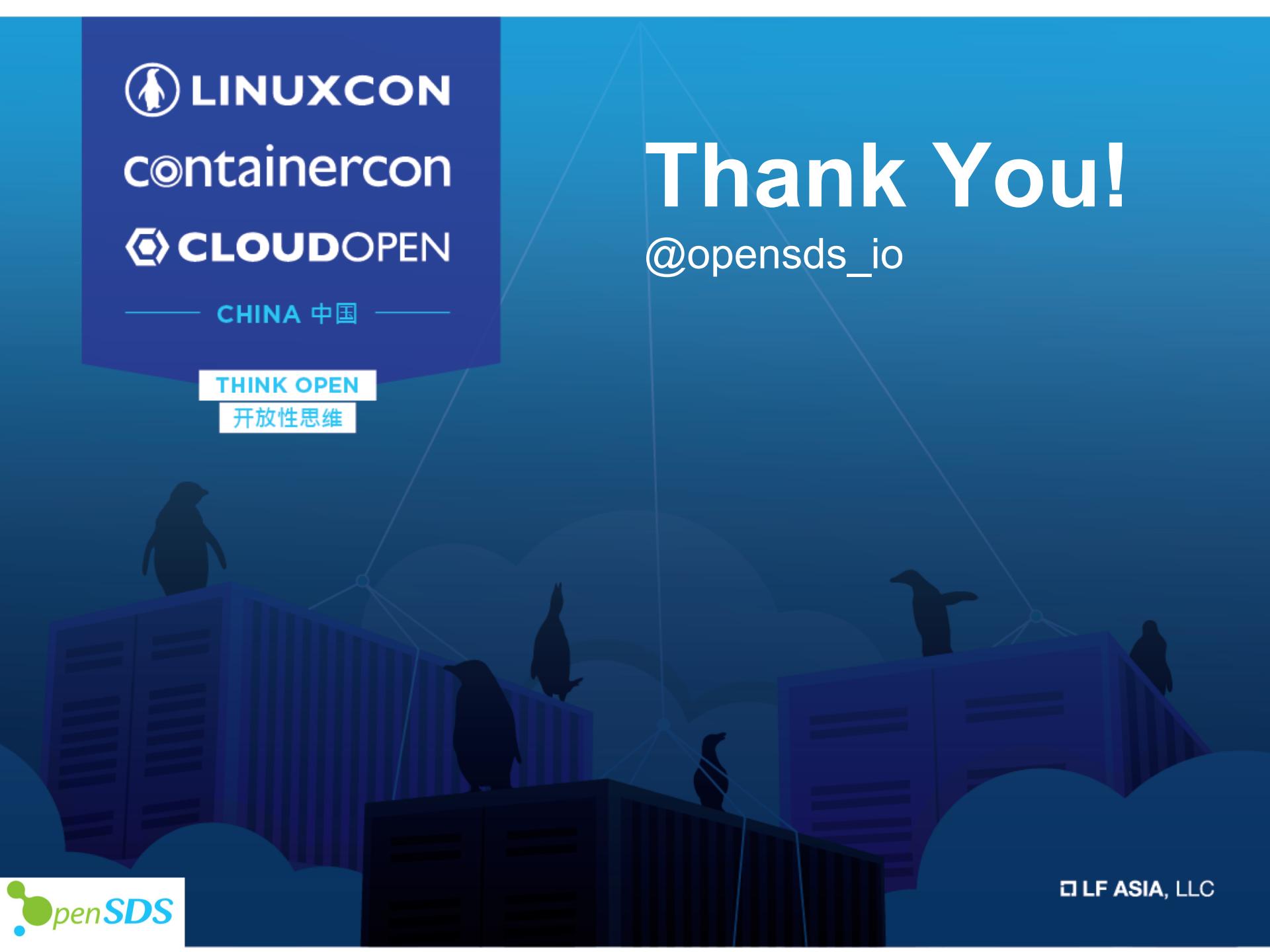
— CHINA 中国 —

THINK OPEN

开放性思维

# Thank You!

@opensds\_io





LINUXCON

Containercon



CLOUDOPEN

— CHINA 中国 —

THINK OPEN

开放性思维