SDC 19

September 23-26, 2019
Santa Clara, CA

OpenSDS and SNIA Partnership

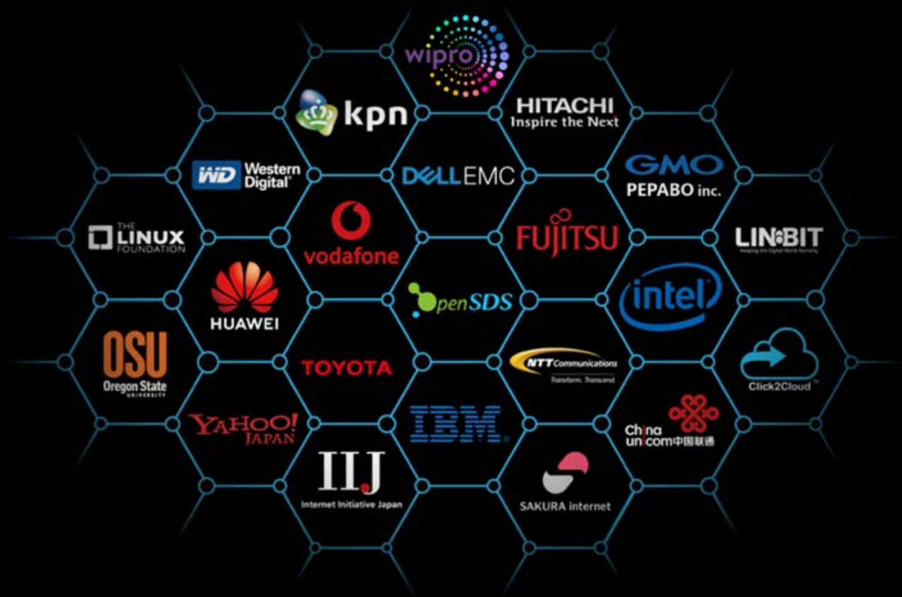
Rakesh Jain
IBM Research

OpenSDS TSC Vice Chair
@rakeshjn

Steven Tan
Futurewei

OpenSDS TSC Chair
@stevenphtan

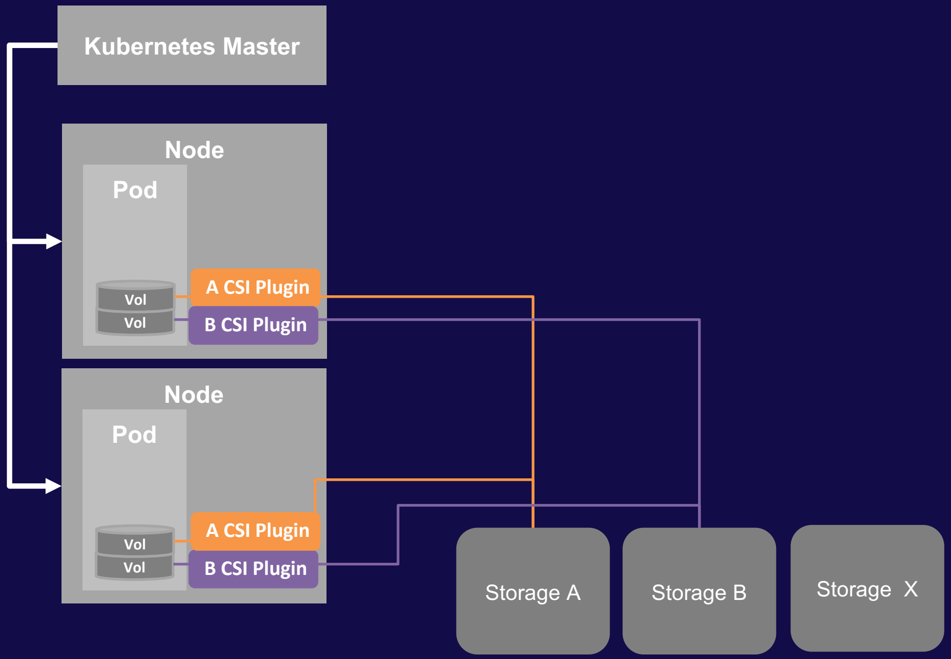
The OpenSDS Project



MISSION

An open source community working to address data and storage management integration challenges for digital transformation

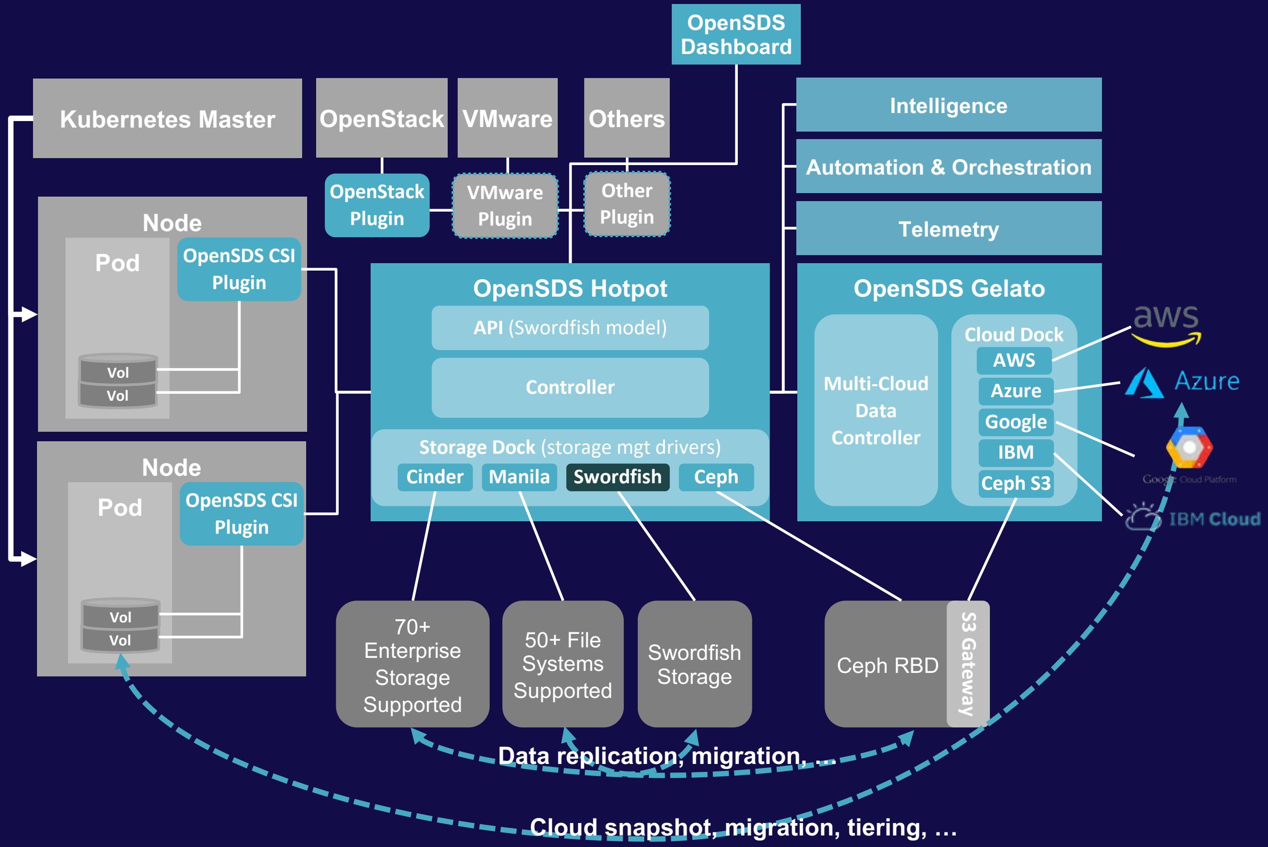
Kubernetes Today



PROBLEMS

- Multiple plugins management & upgrades
- Most storage do not have plugins
- Data isolated in each vendor storage
- Data protection, security, etc. work differently
- Monitoring and analysis difficult

OpenSDS for Kubernetes Overview



BENEFITS

- One plugin for each framework
- Add new storage with same plugin
- 100+ block and file storage supported
- Data ops across storage systems
- Common data protection, security, etc.
- Monitoring and analytics is aggregated
- Extend across data centers and clouds
- Common platform for K8S and others
- Automate workflows

OpenSDS & Swordfish

OpenSDS and SNIA Swordfish

- OpenSDS involvement with Swordfish is in the following ways
  - Uses Swordfish data model and entities, but has its own APIs different than Swordfish
  - Provides feedback to SNIA
  - Plans to implement Swordfish APIs in parallel to its own APIs

OpenSDS Profiles

- OpenSDS uses Profiles as abstraction of storage requirements
- Profile is like Gold, Silver or Platinum level etc defined by the admin
- Profile includes storage details, data protection etc.

OpenSDS uses Swordfish Entities

- The OpenSDS Profiles design adopts the LineOfService concept defined in Swordfish specification. The capabilities defined by Swordfish specification covers the following domains:
  - Data Storage
  - IO Connectivity
  - Data Protection
  - Data Security
  - IO Performance

Data Storage Capabilities

- Describes capabilities of the system to support various data storage service options-
  - ProvisioningPolicy: Represents Thick or Thin
  - IsSpaceEfficient: Represents Compression/Deduplication
  - RecoveryTimeObjectives: Nearline, Offline, Online Active, OnlinePassive

Sample of adding data storage capabilities into profile

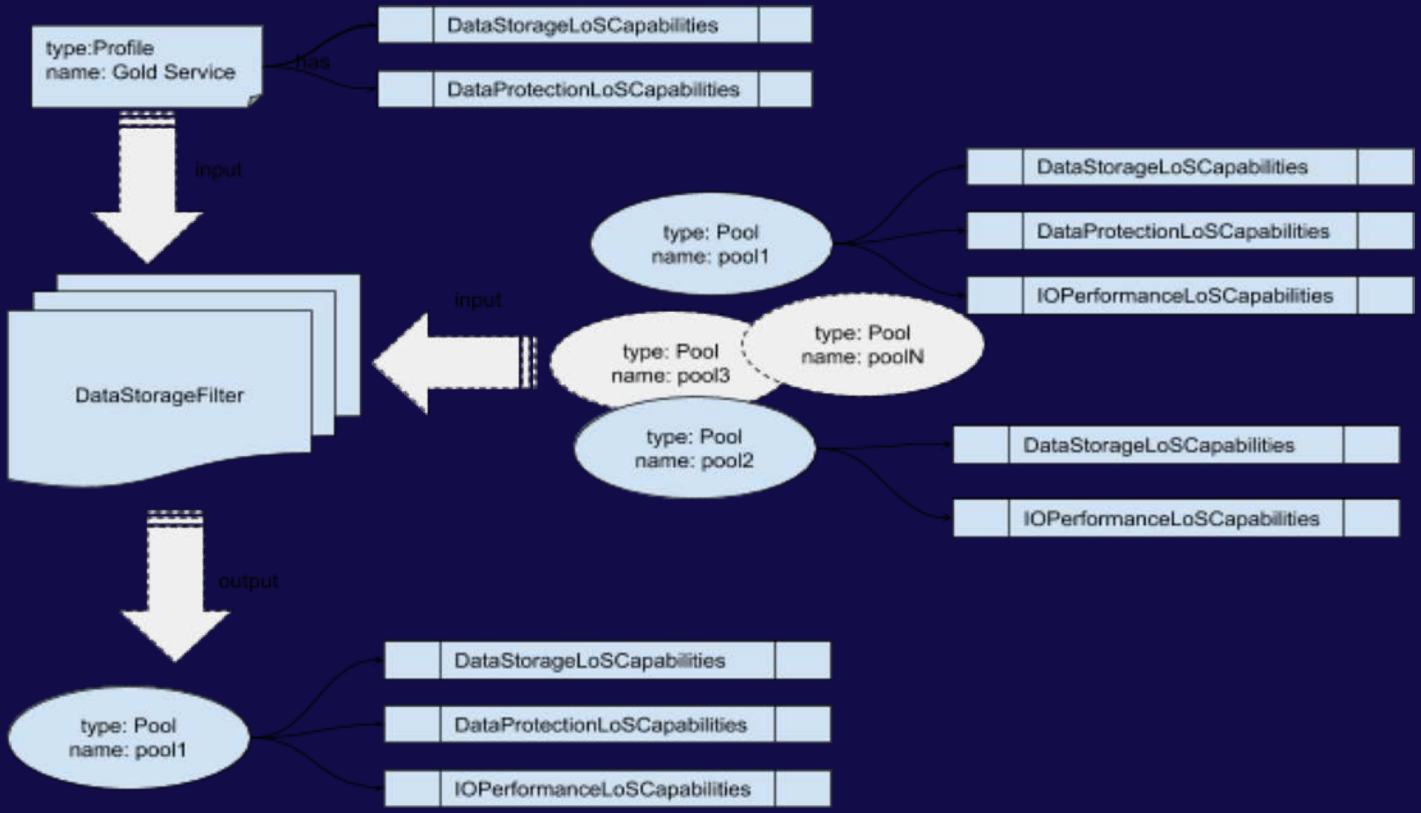
- URL: /v1/{projectId}/profiles/{profileId}/extras

  Method: POST

  JSON schema definition:

```
{
  "DataStorageLineOfService":{
    "RecoveryTimeObjective": ”nearline”,
    "ProvisioningPolicy": "Thin",
    "IsSpaceEfficient": true
  }
}
```

Use of Swordfish Lines of Service in OpenSDS


type:Profile
name: Gold Service
has
DataStorageLoSCapabilities
DataProtectionLoSCapabilities
input
DataStorageFilter
input
type: Pool
name: pool1
DataStorageLoSCapabilities
DataProtectionLoSCapabilities
IOPerformanceLoSCapabilities
type: Pool
name: pool3
type: Pool
name: poolN
type: Pool
name: pool2
DataStorageLoSCapabilities
IOPerformanceLoSCapabilities
output
type: Pool
name: pool1
DataStorageLoSCapabilities
DataProtectionLoSCapabilities
IOPerformanceLoSCapabilities

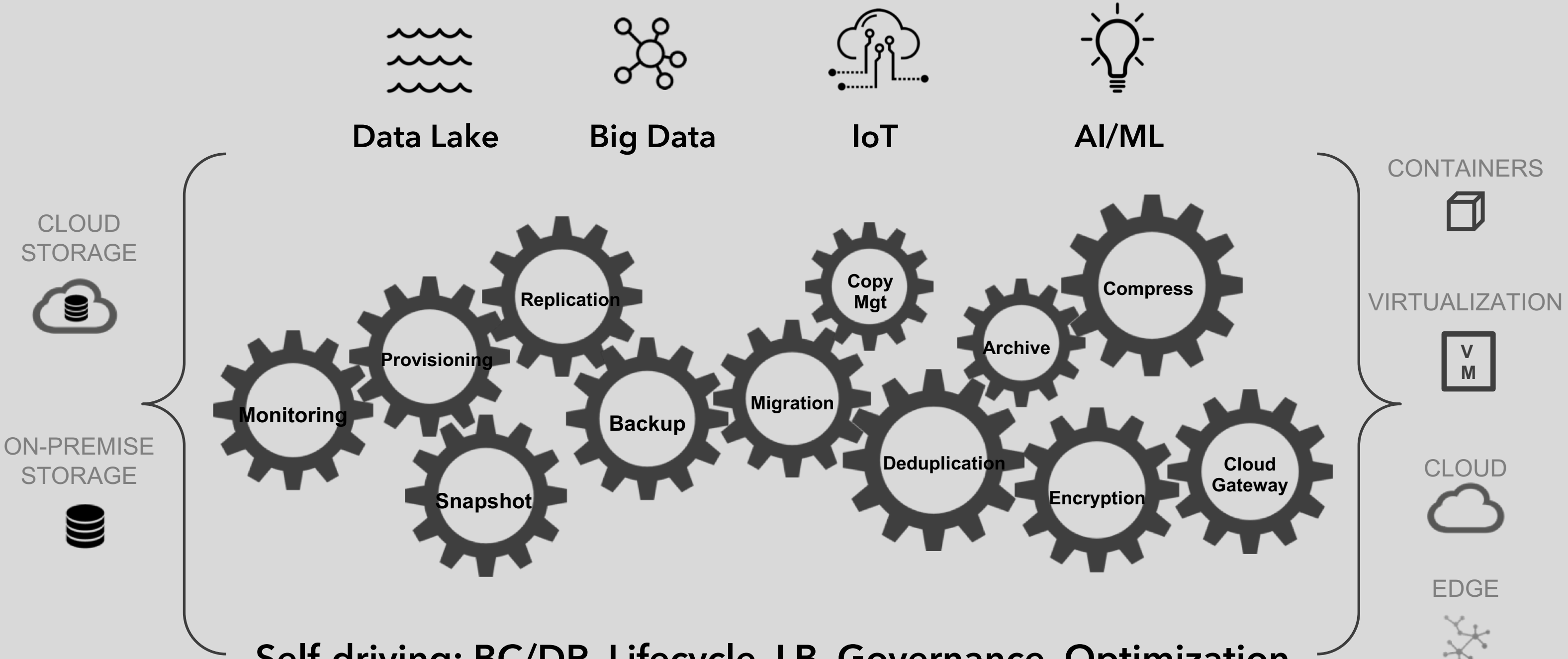
Next Steps

- OpenSDS is revisiting its APIs to align more or completely with Swordfish
- OpenSDS manages storage systems which may or may not have Swordfish implementation. By using such storage systems behind OpenSDS, they automatically get Swordfish compatible APIs through OpenSDS
- As part of the implementation, OpenSDS is working closely with SNIA Swordfish WG to provide feedback on current specs.

From OpenSDS To SODA…

Stuffs for Open Data Autonomy

SODA Open Autonomous Data Management and Storage



Data Lake

Big Data

IoT

AI/ML

CLOUD STORAGE

ON-PREMISE STORAGE

Monitoring

Provisioning

Replication

Snapshot

Backup

Migration

Copy Mgt

Deduplication

Archive

Compress

Encryption

Cloud Gateway

CONTAINERS

VIRTUALIZATION

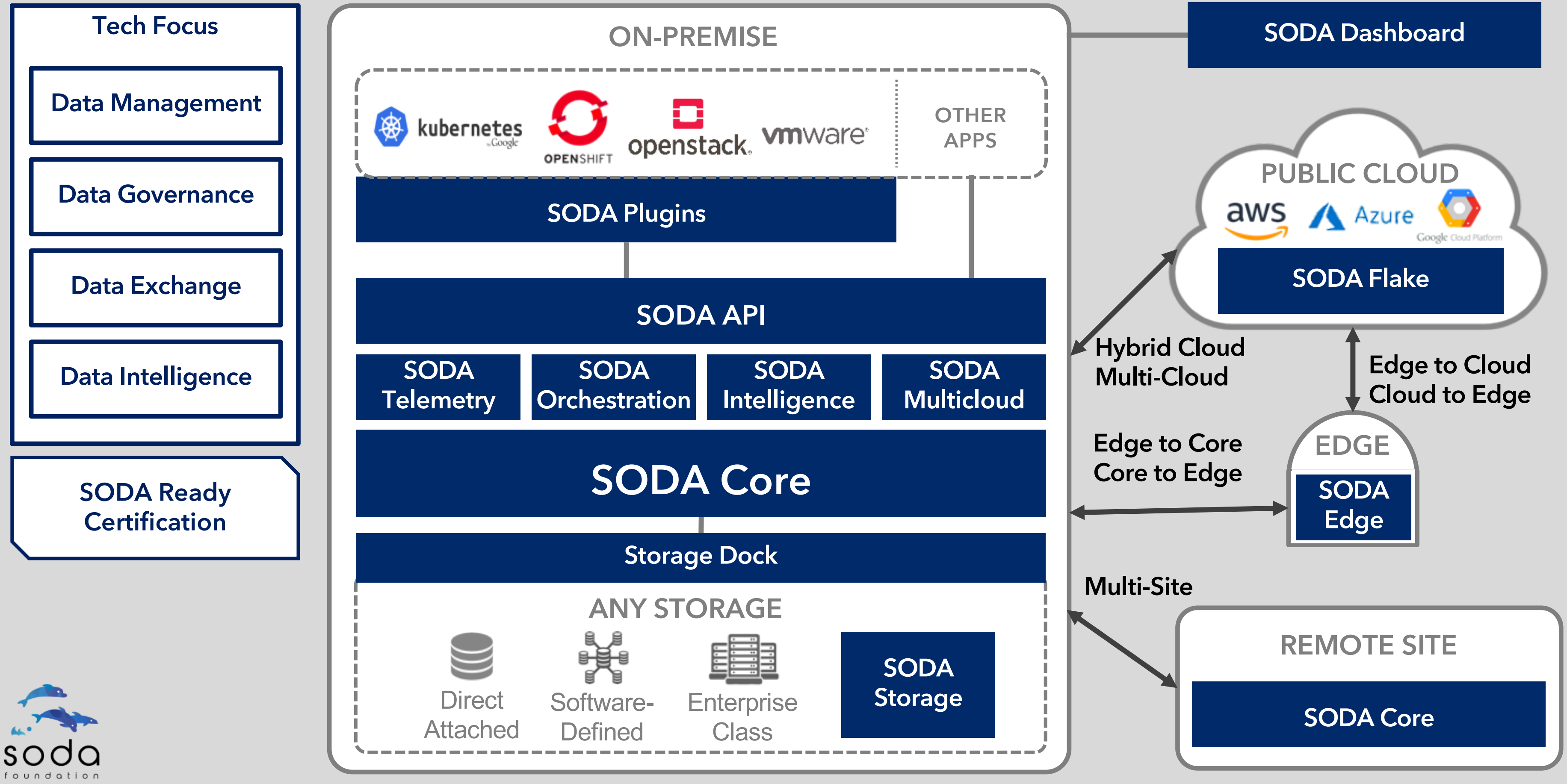
V M

CLOUD

EDGE

Self-driving: BC/DR, Lifecycle, LB, Governance, Optimization …

SODA Overview



Join SODA Foundation

Launching Early 2020

Drive Innovation in Autonomous Data Management & Storage

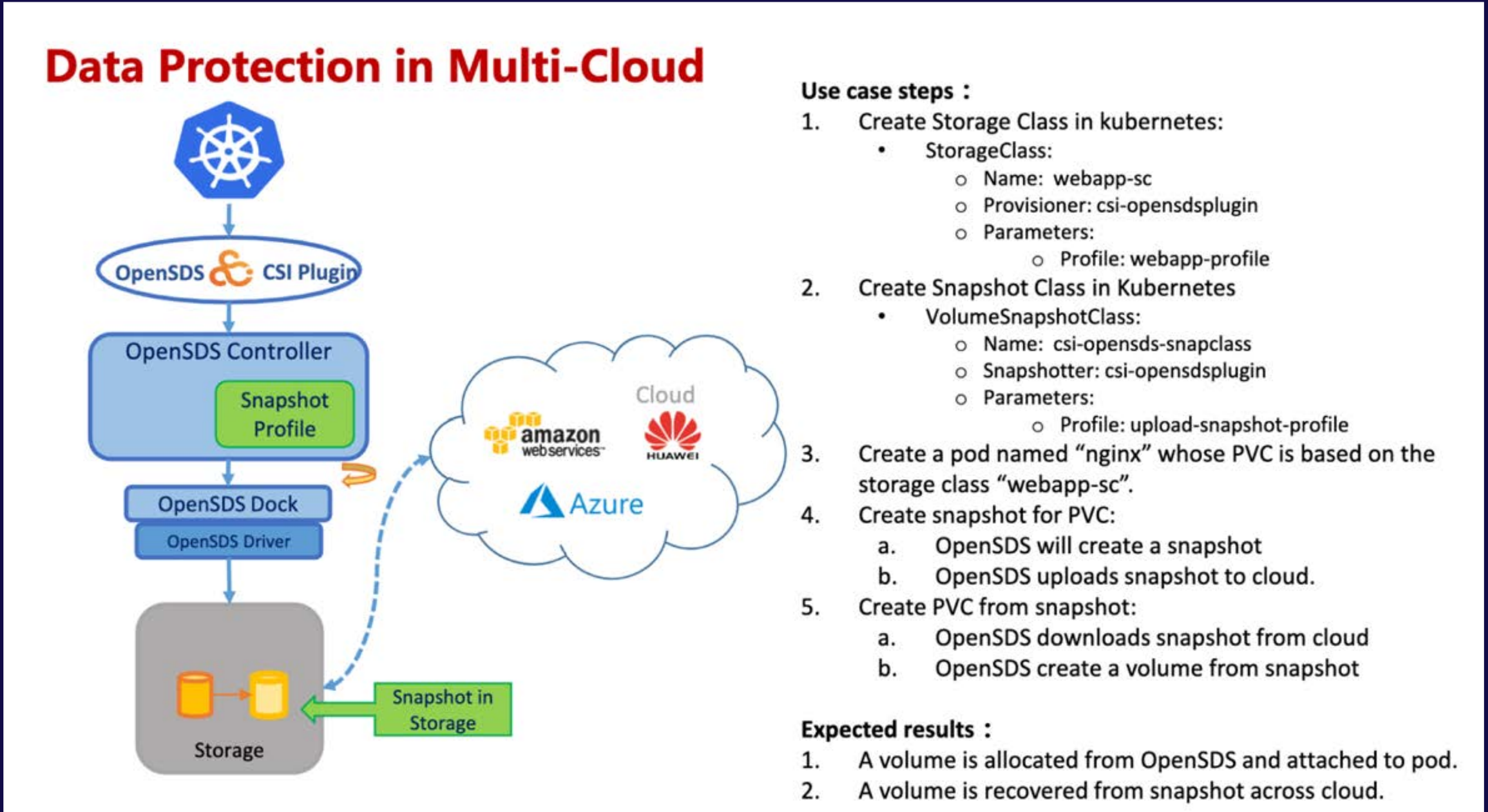
For more info:
www.opensds.io
github.com/opensds

@stevenphtan Steven Tan
@rakeshjn Rakesh Jain

Thank you

Backup

Example Use case



Data Protection in Multi-Cloud

Use case steps：

1. Create Storage Class in kubernetes:
   - StorageClass:
     - Name: webapp-sc
     - Provisioner: csi-opensdsplugin
     - Parameters:
       - Profile: webapp-profile
2. Create Snapshot Class in Kubernetes
   - VolumeSnapshotClass:
     - Name: csi-opensds-snapclass
     - Snapshotter: csi-opensdsplugin
     - Parameters:
       - Profile: upload-snapshot-profile
3. Create a pod named “nginx” whose PVC is based on the storage class “webapp-sc”.
4. Create snapshot for PVC:
   a. OpenSDS will create a snapshot
   b. OpenSDS uploads snapshot to cloud.
5. Create PVC from snapshot:
   a. OpenSDS downloads snapshot from cloud
   b. OpenSDS create a volume from snapshot

Expected results：

1. A volume is allocated from OpenSDS and attached to pod.
2. A volume is recovered from snapshot across cloud.