OpenSDS Manageability using Swordfish for Cloud-native Frameworks

Anjaneya 'Reddy' Chagam, Sr. Principal Engineer, Intel Corporation
Steven Tan, VP & CTO Cloud Storage, Huawei

OpenSDS

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration.   **No product can be absolutely secure.**

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. For more complete information about performance and benchmark results, visit http://www.intel.com/benchmarks .

Software and workloads used in performance tests may have been optimized for performance only on Intel microprocessors. Performance tests, such as SYSmark and MobileMark, are measured using specific computer systems, components, software, operations and functions. Any change to any of those factors may cause the results to vary. You should consult other information and performance tests to assist you in fully evaluating your contemplated purchases, including the performance of that product when combined with other products.   For more complete information visit http://www.intel.com/benchmarks .

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.

The benchmark results may need to be revised as additional testing is conducted. The results depend on the specific platform configurations and workloads utilized in the testing, and may not be applicable to any particular user's components, computer system or workloads. The results are not necessarily representative of other benchmarks and other benchmark results may show greater or lesser impact from mitigations.

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Agenda
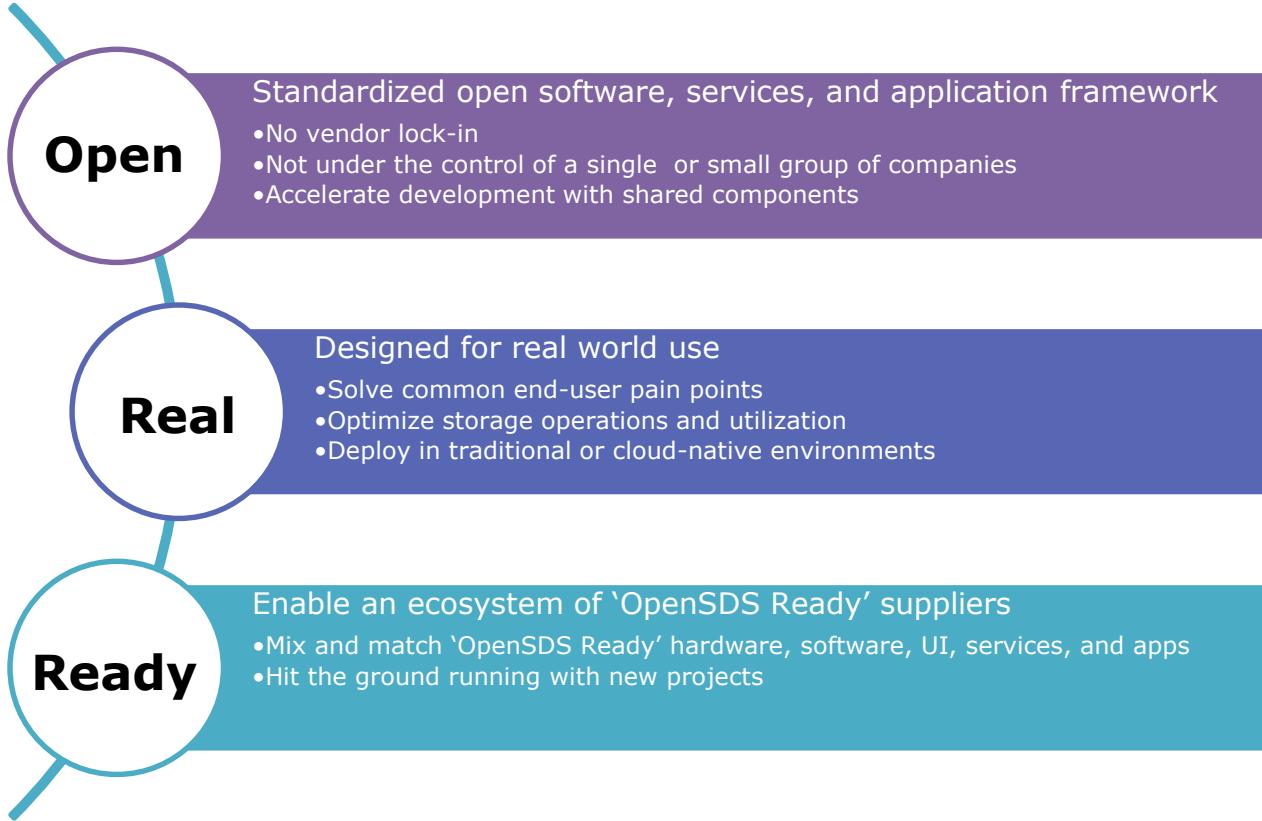
- OpenSDS Overview & Architecture
- OpenSDS Kubernetes Integration Architecture
- OpenSDS Swordfish Manageability
- OpenSDS Flash Management
- Summary
- Q&A

Overview

An open source community working to address **data storage integration** challenges, particularly in **scale-out cloud native environments** with **heterogeneous storage platforms**.

Goals

Open
Standardized open software, services, and application framework
- No vendor lock-in
- Not under the control of a single or small group of companies
- Accelerate development with shared components

Real
Designed for real world use
- Solve common end-user pain points
- Optimize storage operations and utilization
- Deploy in traditional or cloud-native environments

Ready
Enable an ecosystem of 'OpenSDS Ready' suppliers
- Mix and match 'OpenSDS Ready' hardware, software, UI, services, and apps
- Hit the ground running with new projects

Overview

Open SDS Platform	Control/ Management Plane	Standard REST API, Single Pane Management, Policy-Based, Storage and Data Services, Orchestration and Automation
	Data Plane	Data Reliability, High Availability, Data Protection, Data Mobility, Data Reduction, I/O Performance
	Storage/ Service Layer	DAS (HDD/Flash), SAN, NAS, AFA, Commodity, Private Cloud, Public Cloud

Decoupled control plane from data plane and storage/service layer

Manageability Focus

Orchestration & Automation

Data lifecycle, protection, replication, migration, security, governance, optimization

Multi-Cloud Data Management

Placement, access, security, and search across private and public clouds - AWS, Azure, GCP, and more

Data & Storage Intelligence

Data collection, monitor, reports, analytics, forecast, ML/AI

The Core Projects

**SUSHI**

The Northbound Plug-ins Project
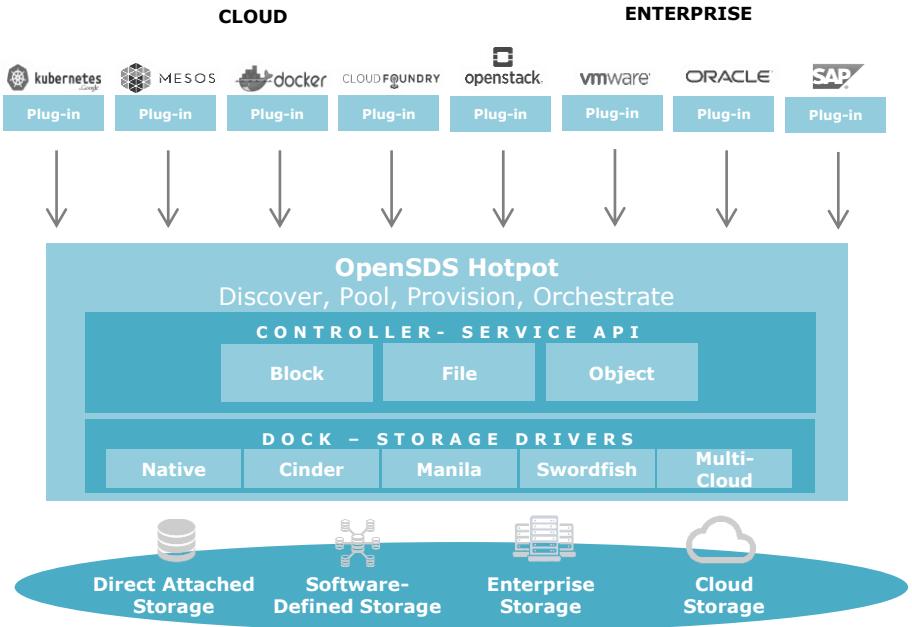
Common plug-ins to enable OpenSDS storage services for cloud and application frameworks

▷
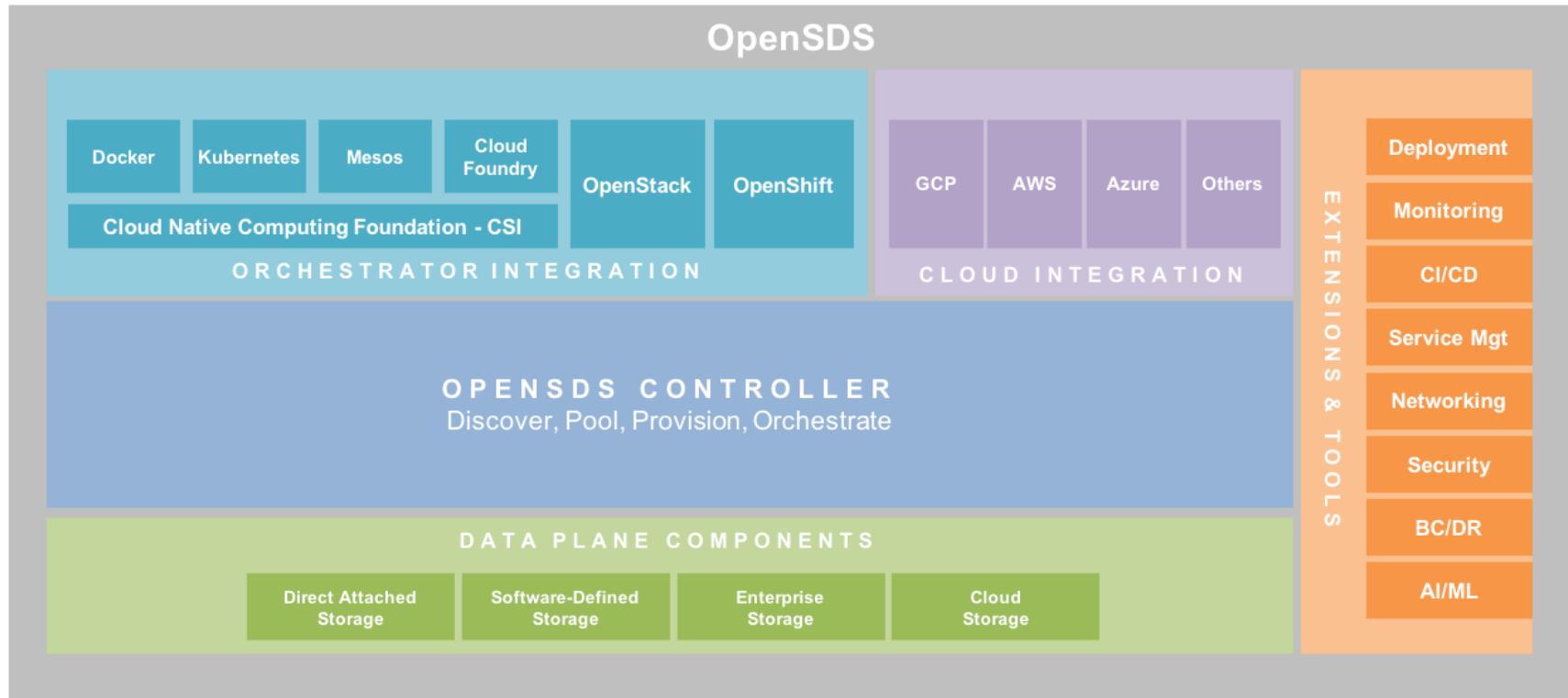
HOTPOT

The Storage Controller Project

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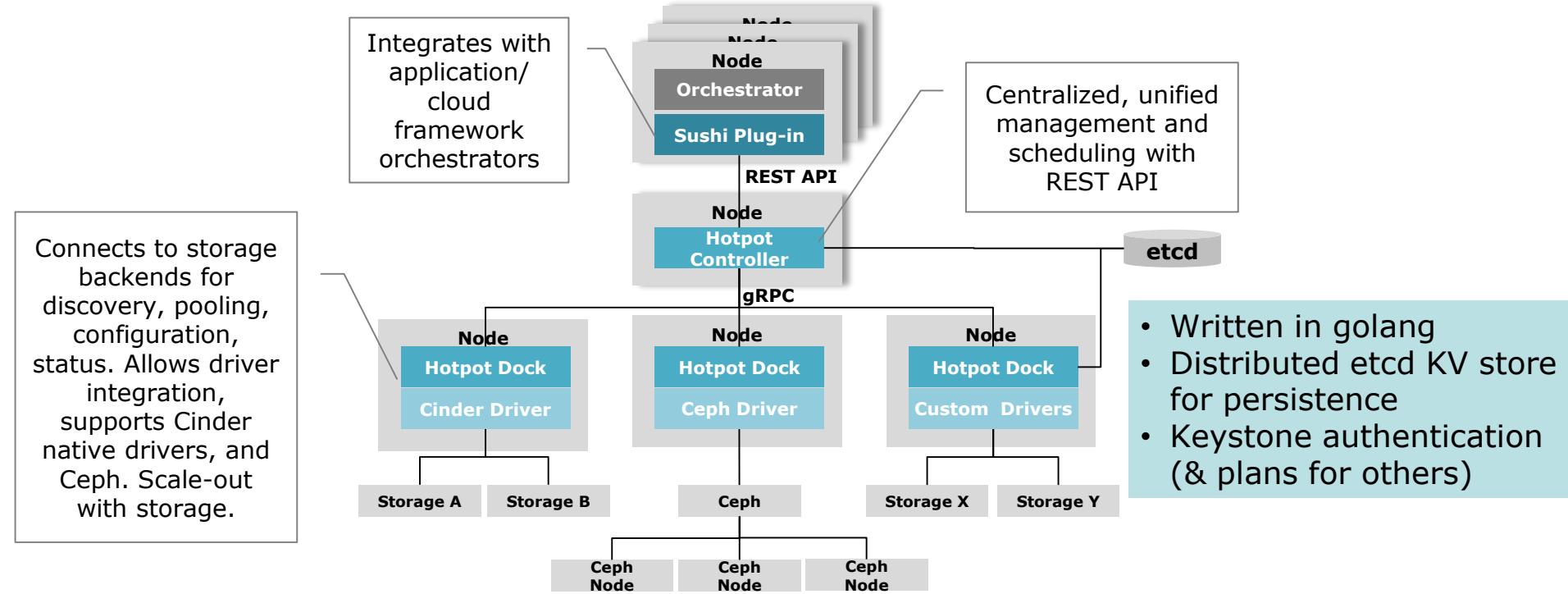
Single control for block, file, and object services across storage on premise and in clouds

CLOUD   ENTERPRISE

kubernetes	MESOS	docker	CLOUD FOUNDRY	openstack	vmware	ORACLE	SAP
Plug-in	Plug-in	Plug-in	Plug-in	Plug-in	Plug-in	Plug-in	Plug-in

OpenSDS Hotpot
Discover, Pool, Provision, Orchestrate

CONTROLLER- SERVICE API

Block	File	Object

DOCK – STORAGE DRIVERS

Native	Cinder	Manila	Swordfish	Multi-Cloud

Direct Attached Storage   Software-Defined Storage   Enterprise Storage   Cloud Storage

OpenSDS Framework

OpenSDS Architecture

OpenSDS Architecture diagram

Integrates with application/ cloud framework orchestrators

Node
Orchestrator
Sushi Plug-in

REST API

Centralized, unified management and scheduling with REST API

Node
Hotpot Controller

etcd

gRPC

Connects to storage backends for discovery, pooling, configuration, status. Allows driver integration, supports Cinder native drivers, and Ceph. Scale-out with storage.

Node
Hotpot Dock
Cinder Driver

Node
Hotpot Dock
Ceph Driver

Node
Hotpot Dock
Custom Drivers

Storage A **Storage B**

Ceph

Storage X **Storage Y**

Ceph Node **Ceph Node** **Ceph Node**

- Written in golang
- Distributed etcd KV store for persistence
- Keystone authentication (& plans for others)

The Dashboard
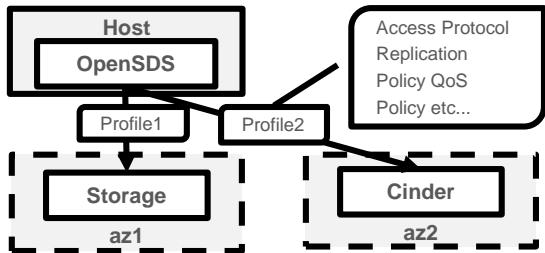
Sample End User Scenarios
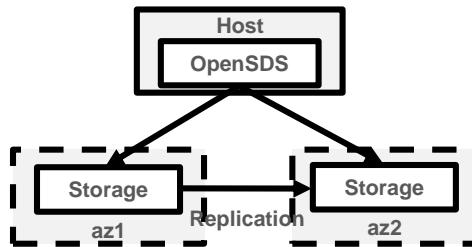
1. **Consolidation** of storage control and management across OpenStack and K8S clusters

2. **Snapshot backups** to cloud and restore to any storage platform

3. **Host-based replication** across different storage backends

4. **Data lifecycle management** to move data across storage tiers and archival for long term retention

5. **Multi-cloud data control** to allow the use of any cloud object store

6. **Orchestrate and automate** provisioning and backups across different storage platforms and backup software

OpenSDS Use Cases
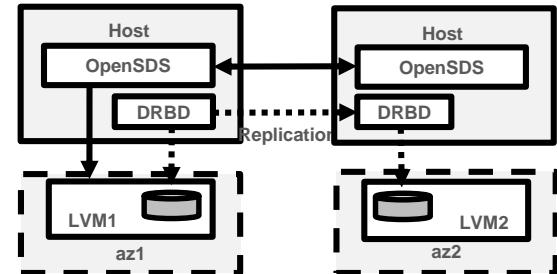
1. Basic Operations(multi storage backend)
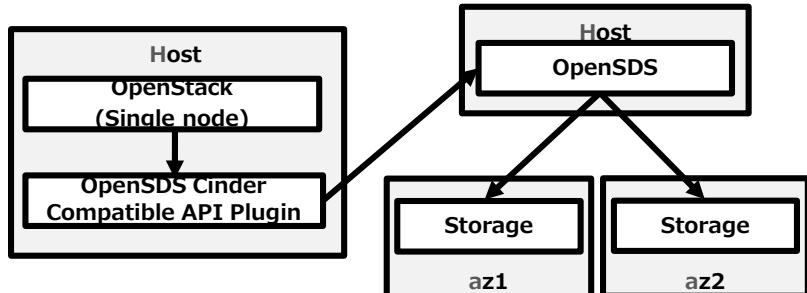
2. Array-based Replication
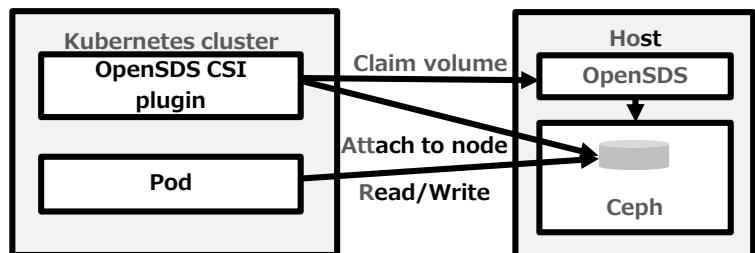
3. Host-based Replication
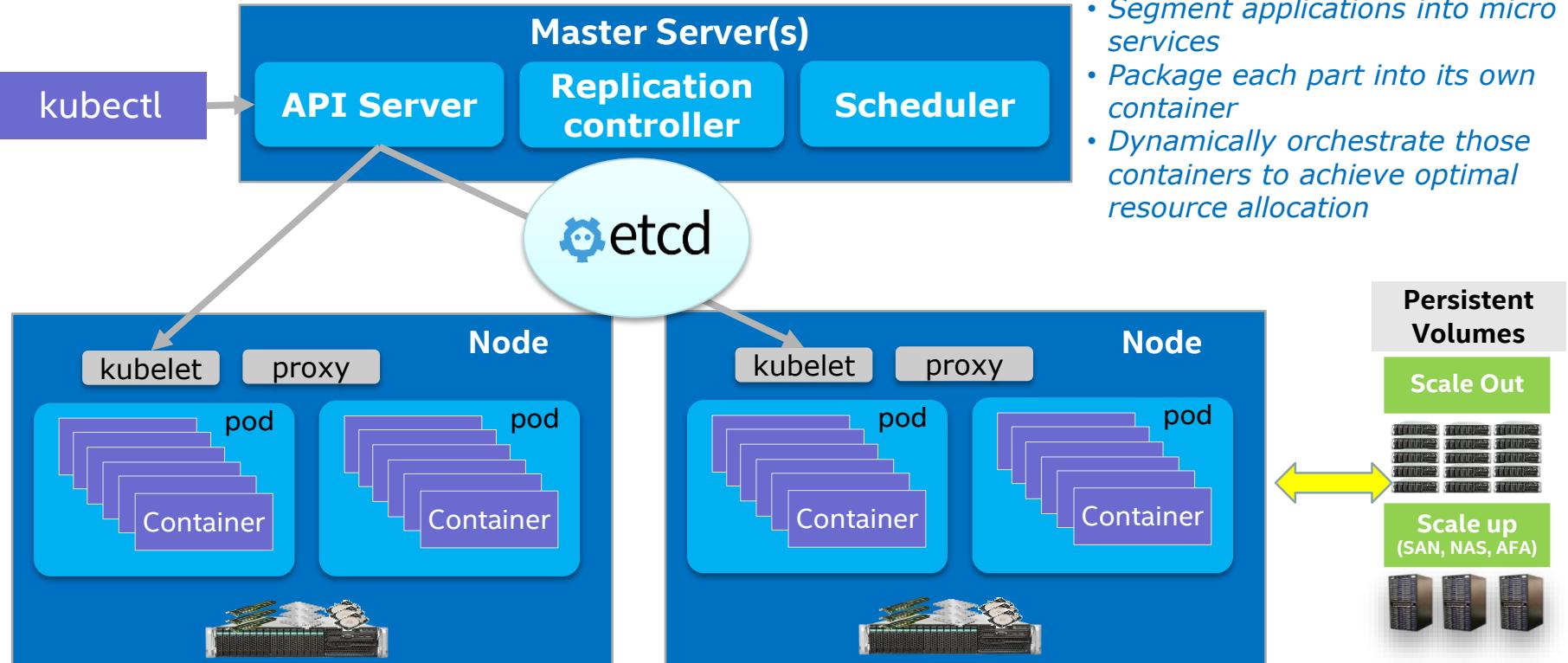
4. OpenStack Integration

5. Kubernetes Integration

Kubernetes Overview

- *Segment applications into micro services*
- *Package each part into its own container*
- *Dynamically orchestrate those containers to achieve optimal resource allocation*

Master Server(s)

kubectl

API Server **Replication controller** **Scheduler**

etcd

Node

kubelet proxy

pod

Container

pod

Container

Node

kubelet proxy

pod

Container

pod

Container

Persistent Volumes

Scale Out

Scale up
(SAN, NAS, AFA)

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)

Docker volume plugin

Mesos volume plugin

kubernetes

K8S dynamic provisioner

K8S flexvolume plugin

K8S in-tree plugin

CLOUD FOUNDRY

Persi

Container Storage Interface

CSI plugin

Node

Kubelet

DaemonSet Pod

Kubernetes CSI Helper Container

Third Party CSI Volume Driver Container

UDS   UDS   Mount propagation

UDS

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

Third Party CSI Volume Driver Container

external-attacher external-provisioner (maybe 1 or 2 containers)

gRPC (UDS)

EmptyDIr Volume

/.../sockets/   Host /var/lib/kubelet

Master

Kube Controller Manager

API Server

etcd

■ External Component - Created by Third Party Storage Vendor

■ External Component - Created by Kubernetes Team

Source: https://github.com/kubernetes/community/blob/master/contributors/
design-proposals/storage/container-storage-interface.md
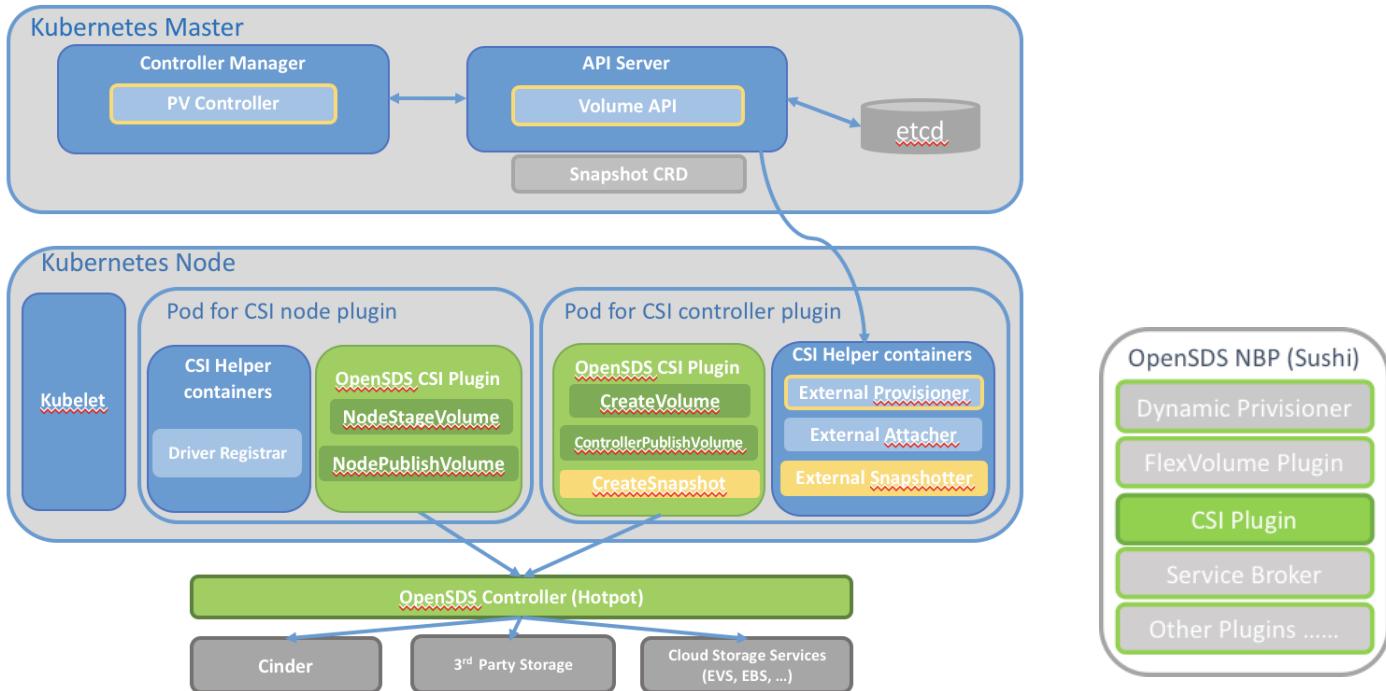
CSI is an industry standard defined to enable storage vendors to develop a plugin once and have it work across a number of container orchestration systems

OpenSDS Kubernetes Architecture
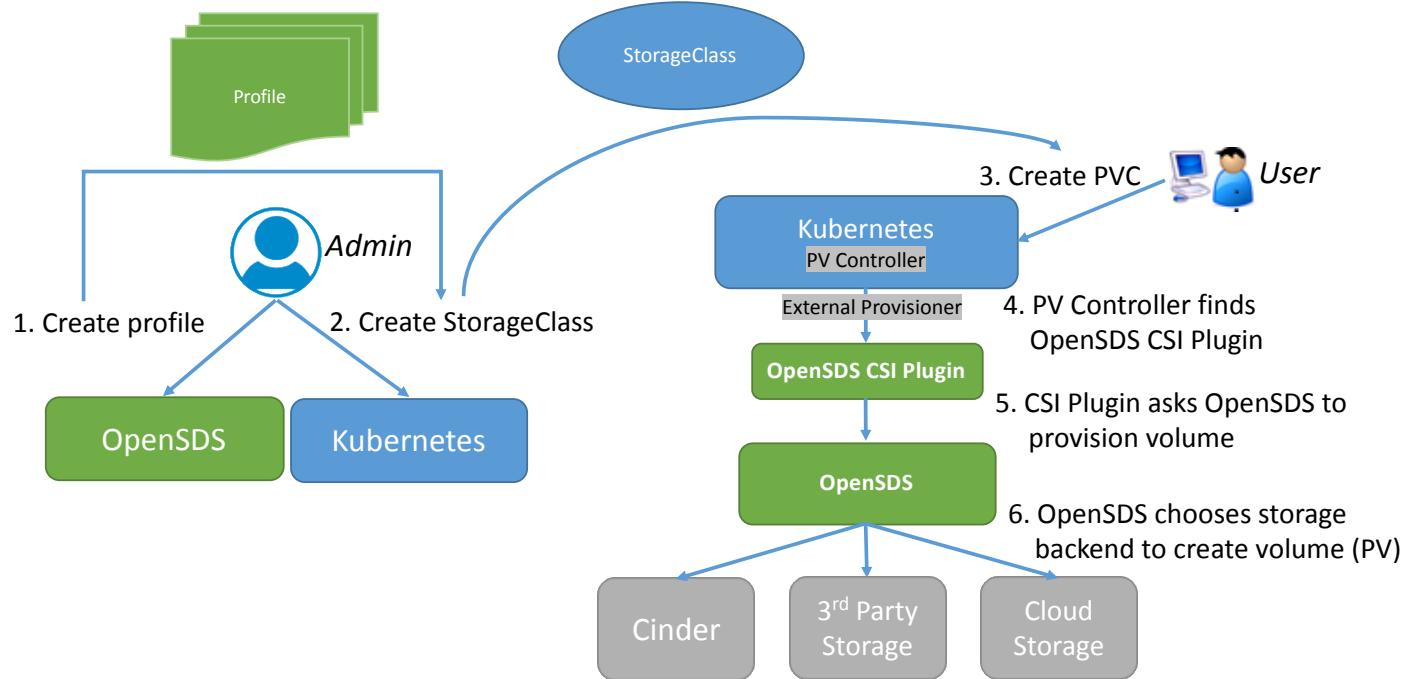
OpenSDS provides driver integration and rich storage services (e.g. data protection, disaster recovery)

OpenSDS – Provision Volumes

OpenSDS – Profile Mapping to K8S

StorageClass

Profile

Admin

1. Create profile

2. Create StorageClass

OpenSDS

Kubernetes

3. Create PVC

User

Kubernetes
PV Controller

External Provisioner

4. PV Controller finds OpenSDS CSI Plugin

OpenSDS CSI Plugin

5. CSI Plugin asks OpenSDS to provision volume

OpenSDS

6. OpenSDS chooses storage backend to create volume (PV)

Cinder

3rd Party Storage

Cloud Storage

Profile – Policy Driven Management
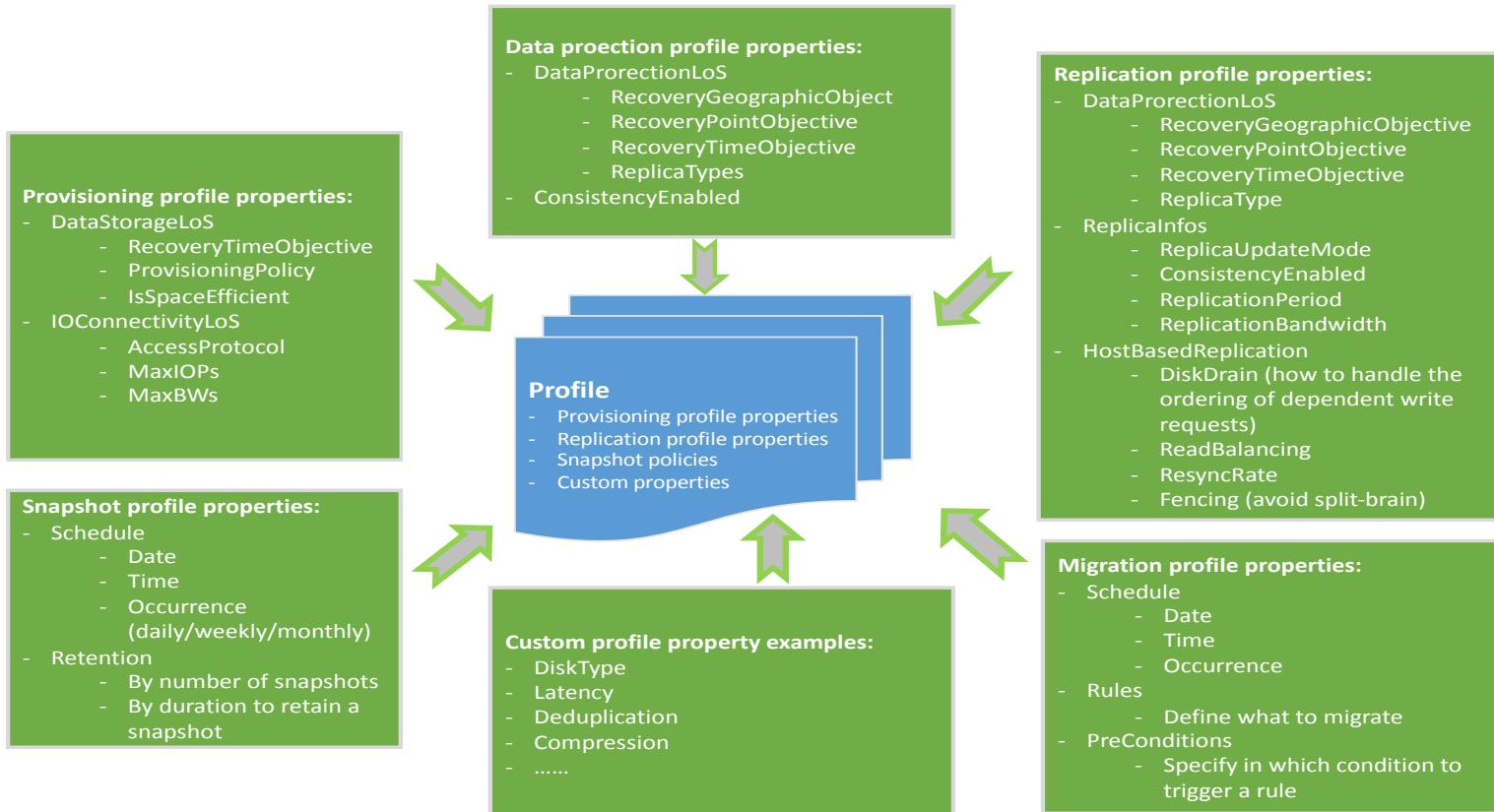
- OpenSDS profile is based on Swordfish specification.

- 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.

Profile **is the key abstraction in "OpenSDS" for storage provisioning to support multi-cloud orchestration**

Source: Swordfish_v1.0.5_Specification
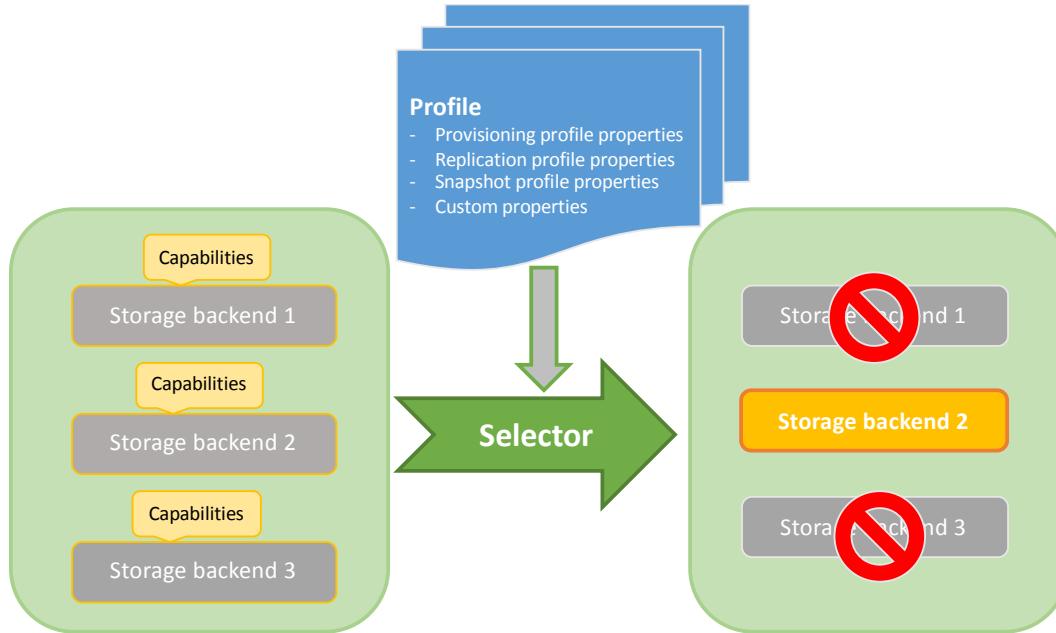
Profile - Swordfish Resource Mapping

Data proection profile properties:
- DataProrectionLoS
    - RecoveryGeographicObject
    - RecoveryPointObjective
    - RecoveryTimeObjective
    - ReplicaTypes
- ConsistencyEnabled

Replication profile properties:
- DataProrectionLoS
    - RecoveryGeographicObjective
    - RecoveryPointObjective
    - RecoveryTimeObjective
    - ReplicaType
- ReplicaInfos
    - ReplicaUpdateMode
    - ConsistencyEnabled
    - ReplicationPeriod
    - ReplicationBandwidth
- HostBasedReplication
    - DiskDrain (how to handle the ordering of dependent write requests)
    - ReadBalancing
    - ResyncRate
    - Fencing (avoid split-brain)

Provisioning profile properties:
- DataStorageLoS
    - RecoveryTimeObjective
    - ProvisioningPolicy
    - IsSpaceEfficient
- IOConnectivityLoS
    - AccessProtocol
    - MaxIOPs
    - MaxBWs

Profile
- Provisioning profile properties
- Replication profile properties
- Snapshot policies
- Custom properties

Snapshot profile properties:
- Schedule
    - Date
    - Time
    - Occurrence (daily/weekly/monthly)
- Retention
    - By number of snapshots
    - By duration to retain a snapshot

Custom profile property examples:
- DiskType
- Latency
- Deduplication
- Compression
- ......

Migration profile properties:
- Schedule
    - Date
    - Time
    - Occurrence
- Rules
    - Define what to migrate
- PreConditions
    - Specify in which condition to trigger a rule

Mapping Profile to Storage Systems

OpenSDS K8S CSI Plug-in

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
```
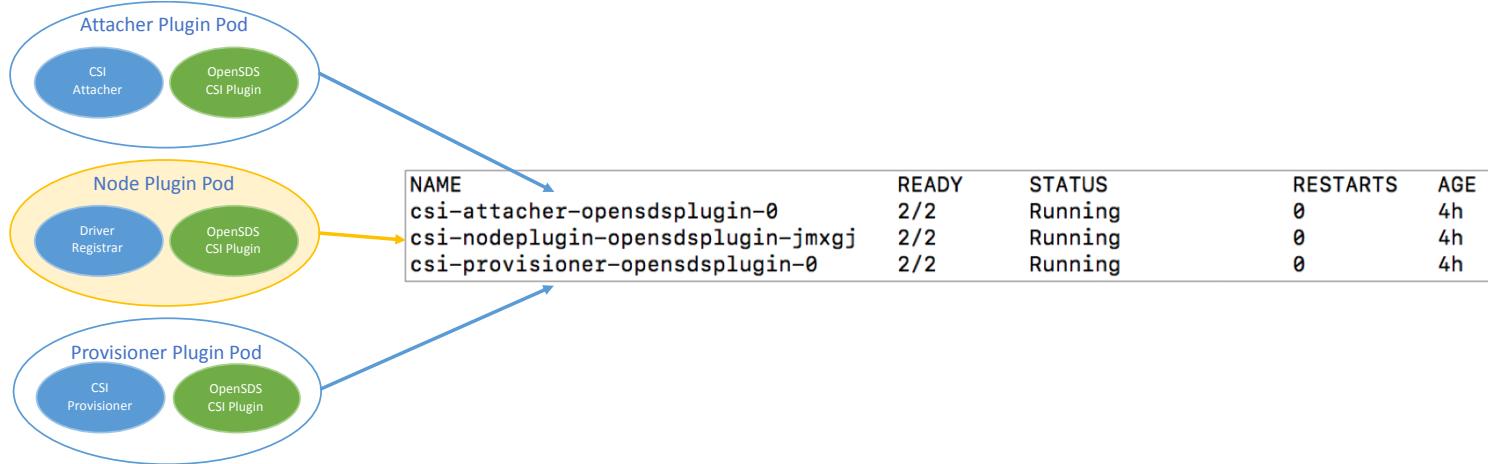
Running OpenSDS CSI Plugin

- Create OpenSDS CSI plugin pods:

  kubectl create -f csi/server/deploy/kubernetes

- Three pods can be found by kubectl get pod :


```
NAME                                READY   STATUS    RESTARTS   AGE
csi-attacher-opensdsplugin-0        2/2     Running   0          4h
csi-nodeplugin-opensdsplugin-jmxgj  2/2     Running   0          4h
csi-provisioner-opensdsplugin-0     2/2     Running   0          4h
```

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)
```
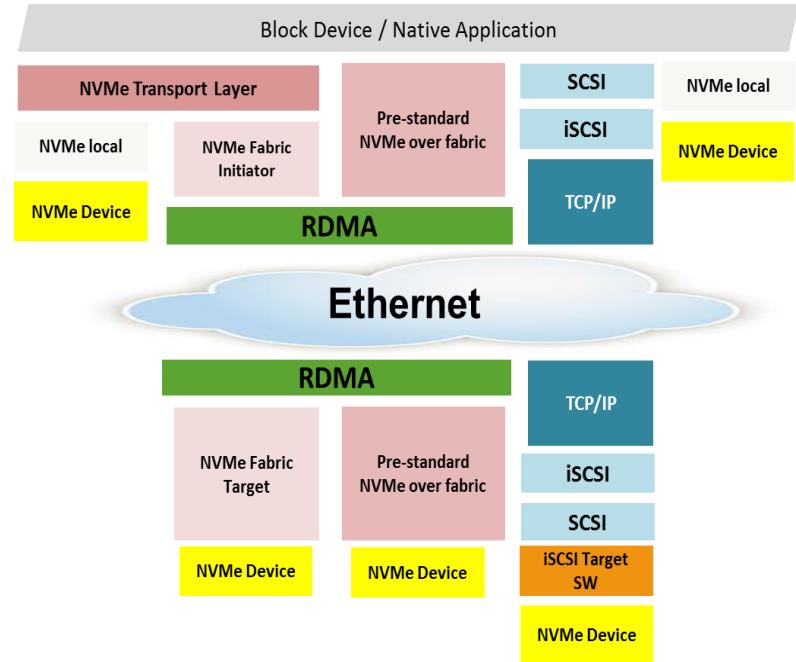
nginx.yaml

```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
```
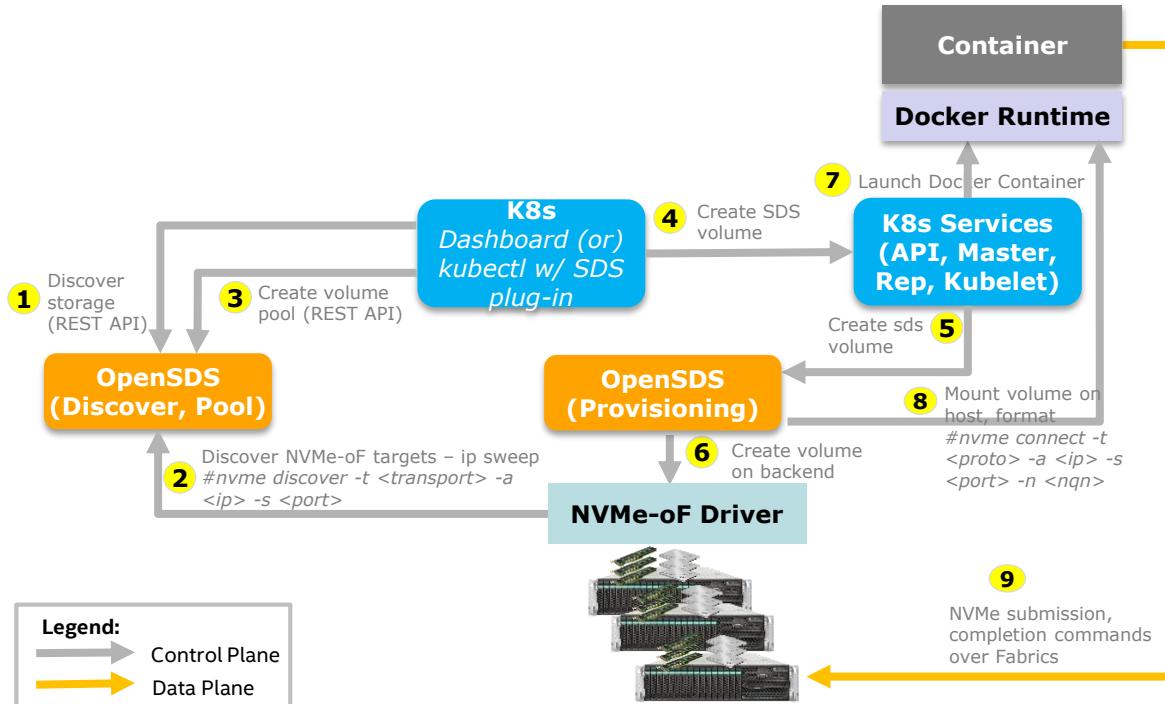
NVMe-oF: Local NVMe Performance

- The idea is to extend the efficiency of the local NVMe interface over a network fabric
  - Ethernet or IB
  - NVMe commands and data structures are transferred end to end
- Relies on RDMA for performance
  - Bypassing TCP/IP
- For more Information on NVMe over Fabrics (NVMe-oF)

    http://www.nvmexpress.org/wp-content/uploads/NVMe_Over_Fabrics.pdf

NVMe-oF in OpenSDS

Work In Progress
- Linux Kernel Driver
- *nvme* commands for connect
- Drive assignment
- Profile includes target info

2019 Plans
- Multi-Target Pooling
- Rack aware scheduling
- Virtual Volume Provisioning
- NVMe over TCP/IP

OpenSDS Roadmap v0.18

2017H2
ZEALAND

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

2018H1
ARUBA

- OpenStack
- Replication
  Array-Based, Host-Based
- Dashboard
- Storage Profiles
- Enumeration
- Block Storage
  - Cinder Drivers
  - Ceph
  - LVM
  - Huawei: Dorado

2018H2
BALI

- S3 Object
- Multi-Cloud Data Control
- Monitoring
- Storage Groups
  Snapshots, Replication
- NVMeoF Preview

2019H1
CAPRI*

- Multi-OpenStack
- File Share
- Migration
- Data Protection
- NVMeoF

2019H2++

- Optimization
- Tiering
- Security
- Sharing
- Networking

Governance

Technical Steering Committee

Steven Tan, Chair
Huawei, VP & CTO Cloud Storage Solution

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

Allen Samuels
Western Digital, R&D Engineering Fellow
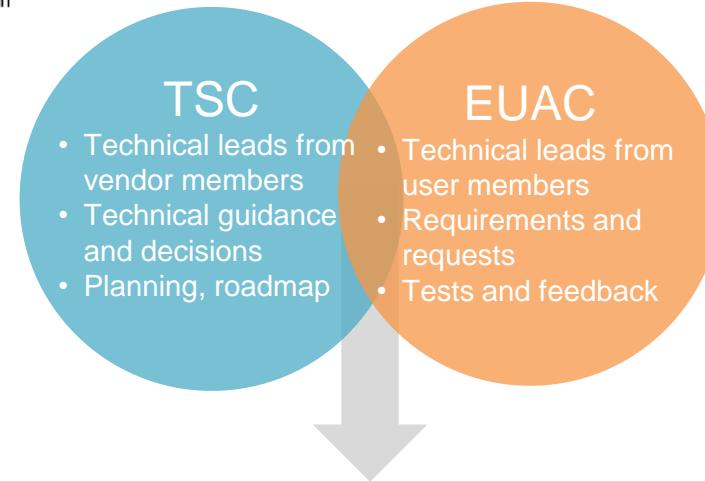
Anjaneya "Reddy" Chagam
Intel, Chief SDS Architect

Jay Bryant
Lenovo, Cloud Storage Lead

Shinya Hamano
Fujitsu, Software Architect

End-User Advisory Committee

Cosimo Rossetti
Vodafone, Lead Storage Architect

Yusuke Sato
Yahoo Japan, Infrastructure Lead
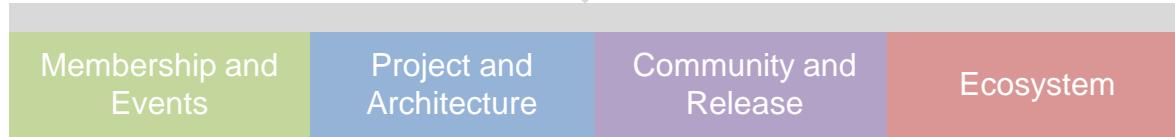
Kei Kusunoki
NTT Communications, Storage Architect

Yuji Yazawa
Toyota ITC, Group Lead

Wim Jacobs
KPN, Senior Architect

TSC
- Technical leads from vendor members
- Technical guidance and decisions
- Planning, roadmap

EUAC
- Technical leads from user members
- Requirements and requests
- Tests and feedback

Membership and Events	Project and Architecture	Community and Release	Ecosystem

Summary

- OpenSDS aim is to address ***data storage integration challenges*** in cloud native environments with heterogeneous storage platforms

- OpenSDS is collaborating with SNIA to deliver Swordfish based DC manageability

- OpenSDS community is fast growing - join and contribute to address emerging storage management challenges

THANK YOU openSDS

🔗 **https://www.opensds.io**

○ **https://github.com/opensds**

✉ **info@opensds.io**

🐦 **@opensds_io**

🍀 **opensds.slack.com**

FIND OUT MORE | **Meet us at the OpenSDS booth!**
Santa Clara, Sep 24-27

BE A MEMBER | **Accepting New Members**
Vendors And End Users Welcome

Backup Slides

Join OpenSDS

End User Benefits

1. Nominate a representative to the End-User Advisory Committee (bi-weekly online EUAC meetings)

2. Leverage OpenSDS technologies to transform storage for cloud native era

3. Seek help and support from OpenSDS TSC, EUAC and developer community

4. Propose projects, specify requirements, and vote on roadmap priorities

5. Participate in OpenSDS sponsored events; speak and exhibit

6. Get vendors to work together

Join OpenSDS

Vendor Benefits

1. Nominate a representative to the Technical Steering Committee (bi-weekly online TSC meetings)

2. Engage end-users and understand their needs first hand

3. Propose projects, specify requirements, and guide the roadmap

4. Participate in OpenSDS sponsored events; speak and exhibit

5. Collaborate and explore potential partnerships with other vendors in an open manner