



OpenSDS Aruba Release

Integration with Openstack and Containers

Rakesh Jain

Researcher & Architect, IBM Research

Vice-chair, OpenSDS TSC

@rakeshjn



About Me

- With IBM for about sixteen years
- About ten years in storage management and optimization
- Currently in Research
- Worked on IBM Spectrum Control (TPC) – the storage management product
- Worked on IBM Cloud (CMS) storage provisioning
- Working on DR solution for cloud native Watson Health offering
- Developed big data solution providing storage intelligence
- I also do Internet of Things work – application management
- Involved with OpenSDS for about a year, TSC vice-chair



Agenda

- General Introduction
- OpenSDS Aruba Release
 - Storage Profiles
 - Integration with Openstack
 - Cloud Native support
 - Dashboard



Mission

The OpenSDS project promotes the use of **simplified storage interfaces** using a **scalable storage controller architecture** with **open standard APIs**, with the objective of providing **application- oriented storage services**



Why OpenSDS?

- Truly Software Defined!!!
- Storage Profiles for defining policies
- Orchestration for automated tasks
- Array based and Host based replication
- Multi region storage management
- Cloud native storage provisioning for a large set of vendors
- Cloud connectivity
- Variety of replication options
- Block, File, Object support
- Installable as Docker containers



OpenSDS Design

Common Interface

Universal OpenSDS plug-ins for different frameworks. Open API for integration

Policy-Based Controls

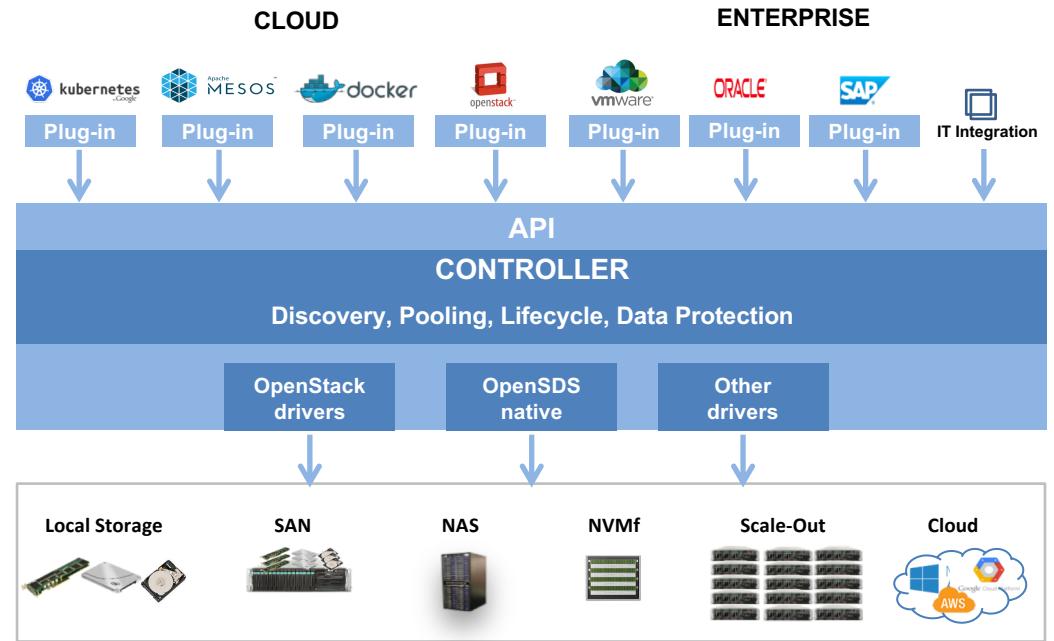
Pooling of storage resources with policy-based controls for provisioning, data protection, lifecycle management and more

Wide Storage Support

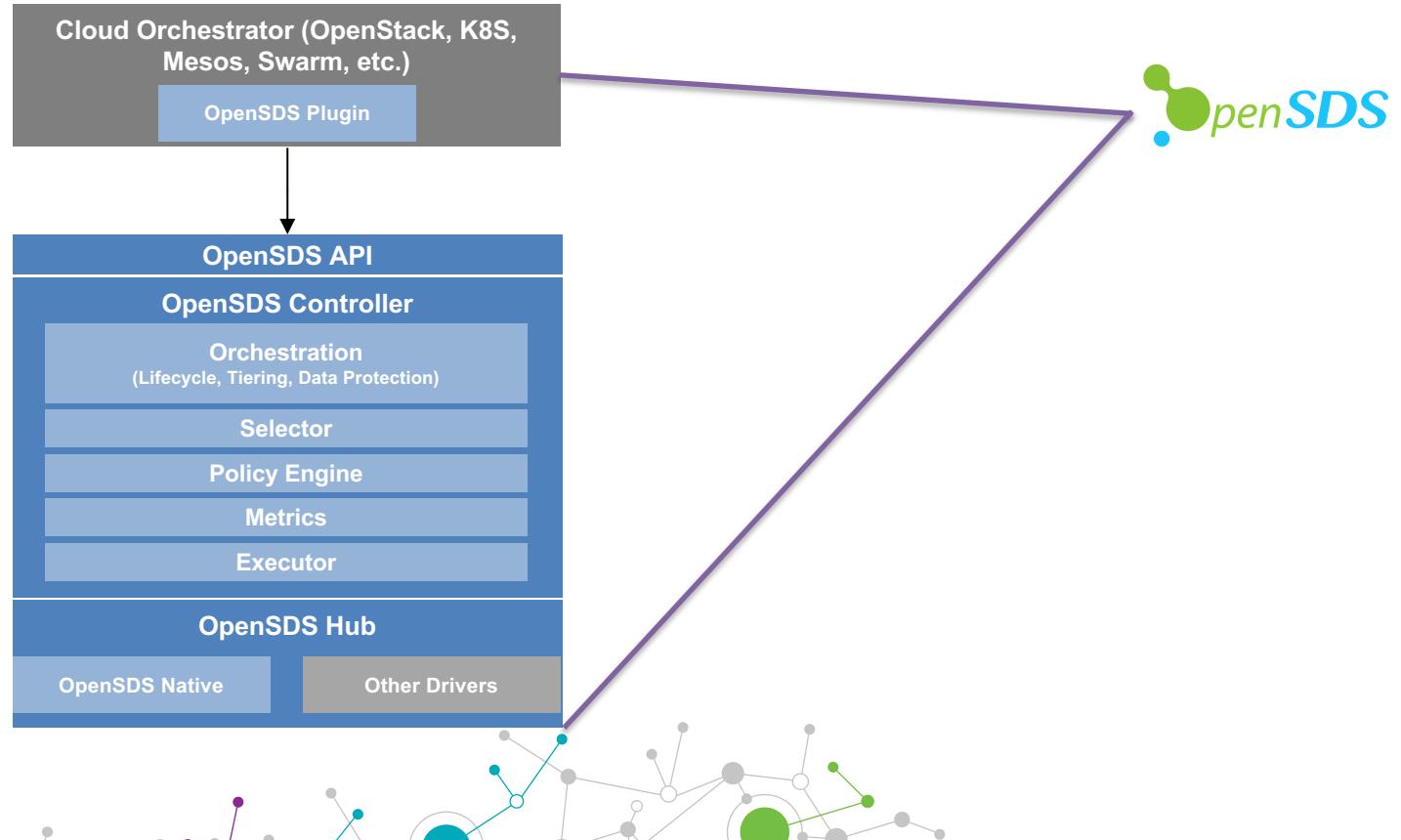
Comes with support for range of Cinder and Manila storage drivers. Extensible for new storage

Enterprise Class

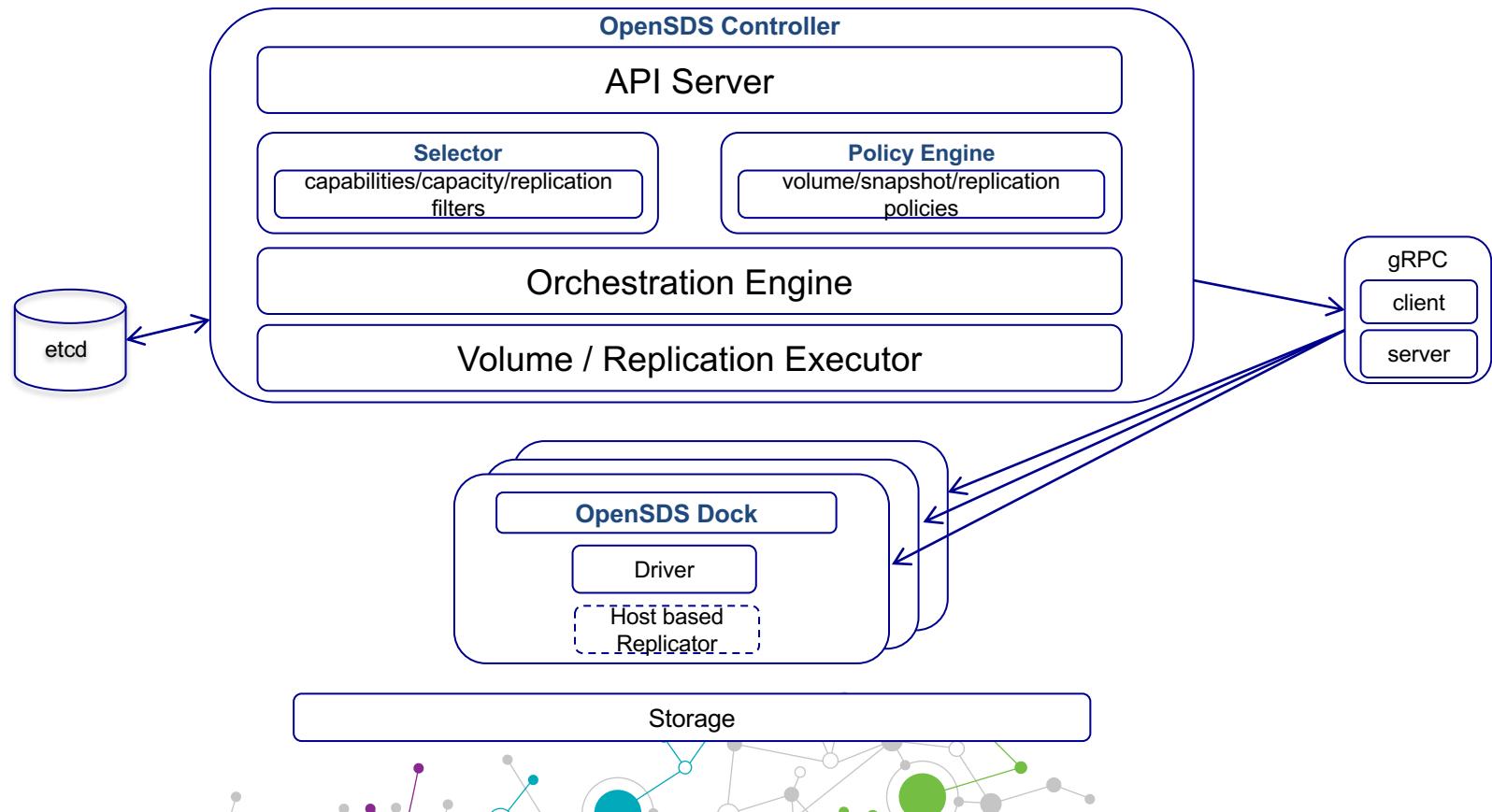
OpenSDS API's and policies support x86 HW and enable the use of enterprise-class storage features



OpenSDS Building Blocks



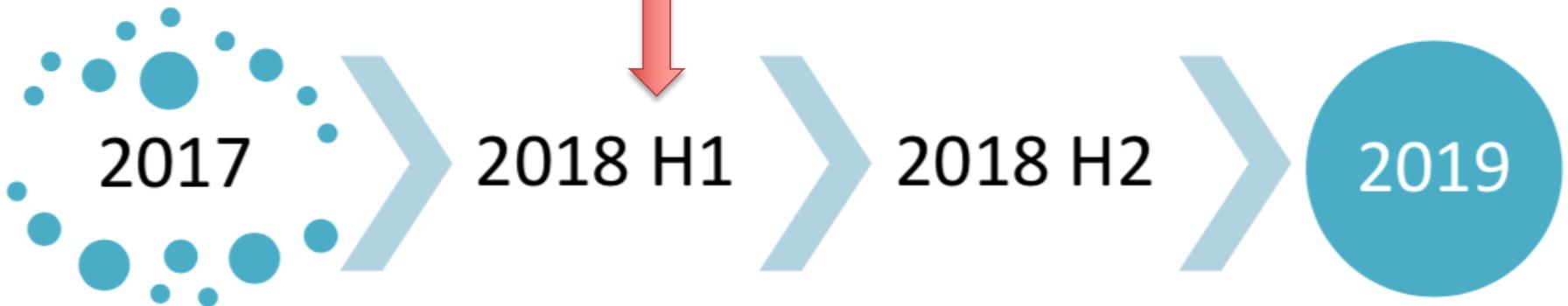
OpenSDS Building Blocks



Aruba Release

- First production level release planned for June 2018
- What's in the release
 - Volume CRUD operations
 - Storage Profiles
 - Groups
 - Snapshot
 - Openstack integration
 - Cloud Native CSI support
 - Kubernetes Plugin
 - South bound drivers
 - Ceph, Cinder, Dorado, SVC





BETA

- K8S Plugin
- CLI
- Vol CRUD
- Vol Snapshot
- SB Drivers
 - Cinder
 - Ceph

RELEASE A

- OpenStack
- Dashboard
- Virtual Pools
- Group Snapshot
- **Storage Profiles**
- **CSI Support**
- Enumeration Listing
- Block Storage IBM, Huawei

RELEASE B

- Multi-OpenStack
- Enumeration Filtering
- Availability HA Mirroring
- **Monitoring**
- S3
- Block Storage HDS, HPE, IBM SVC
- **SB Swordfish**
Dell-EMC, NetApp
- **Replication**

- File Support
- Performance
- Availability Optimization
- Protection
- Tiering
- Security
- Sharing
- Analytics
- Networking
- Multi-tenancy
- NVMeF
- SCM

OpenSDS Roadmap

Storage Profiles

- OpenSDS storage profiles represent service levels
 - Platinum, Gold, Silver etc.
- Aims to define the standard capabilities rules to abstract out storage pools from different storage vendors
- Influenced by SNIA **Swordfish** specifications. Covers following domains:
 - Data Storage
 - IO Connectivity
 - Data Protection
 - Data Security
 - IO Performance



Data Storage Capabilities

- ProvisioningPolicy
 - Thick or Thin volume
- IsSpaceEfficient
 - Compressed/Deduplication volume or not
- RecoveryTimeObjective
 - Immediate or offline



IO Connectivity Capabilities

- Access Protocol
 - iSCSI, Fiberchannel, FICON, FCoE etc.
- MaxIOPS
 - Maximum IOs per second that the connection shall allow for the selected access protocol.



Data Protection Capabilities

- IsIsolated
 - Indicates replica is in separate fault domain from source
- MinLifeTime
 - The lifetime of replica
- RecoveryGeographicObjective
 - Geographic scope of failure domain: Datacenter, Rack, Rackgroup, Region, Row, Server
- RecoveryPointObjective
 - Source data that can be lost, by duration
- RecoveryTimeObjective
 - Maximum time allowed to access a replica: Immediate, Offline etc.
- Replica Type
 - Mirror, Snapshot, Clone etc
- Schedule
 - For periodic replica



For future release

- Data Security Capabilities
 - AntivirusEngineProvider
 - AntivirusScanPolicies
 - ChannelEncryptionStrength
 - DataSanitizationPolicy
 - HostAuthenticationType
 - MediaEncryptionStrength
 - SecureChannelProtocol
 - UserAuthenticationType
- IO Performance Capabilities
 - AverageIOOperationLatencyMicroseconds
 - IOOperationsPerSecondIsLimited
 - MaxIOOperationsPerSecondPerTerabyte
 - SamplePeriod



OpenSDS Provisioning Profile

Provisioning profile will include DataStorageLineOfService and IOConnectivityLineOfService definitions, and optionally the DataProtectionLineOfService definition

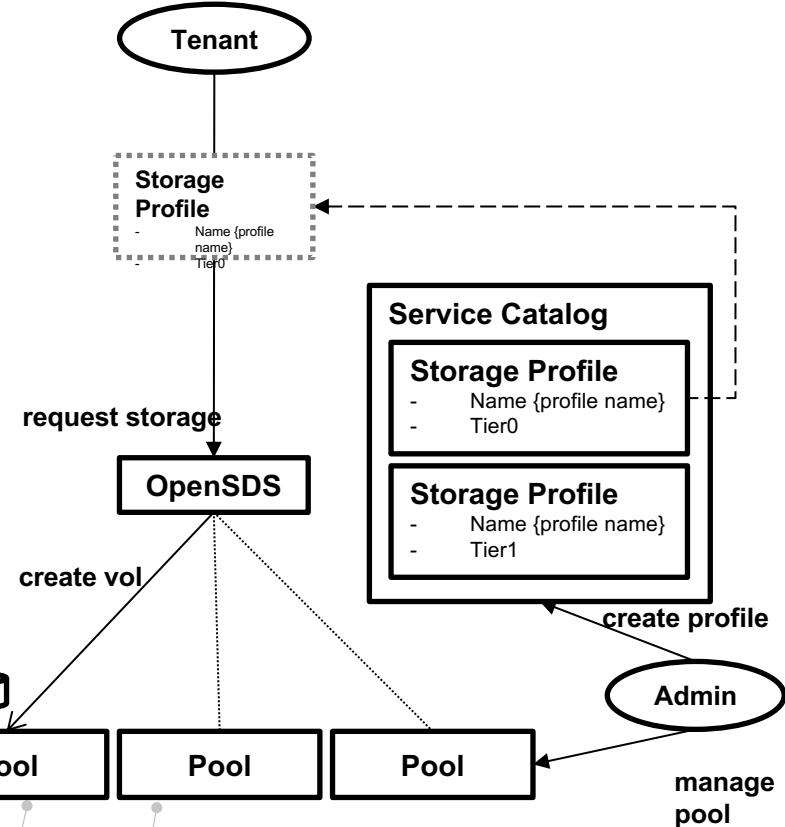
```
{
  "DataStorageLineOfService": {
    "ProvisioningPolicy": "Thin",
    "IsSpaceEfficient": true,
    "RecoveryTimeObjective": 0,
  },
  "IOConnectivityLineOfService": {
    "AccessProtocol": "iSCSI",
    "MaxIOPs": null
  },
  "DataProtectionLineOfService": {.....},
  ....
}
```



```
{
  "DataProtectionLineOfService": {
    "IsIsolated": true,
    "MinLifetime": null,
    "RecoveryGeographicObjective": "Datacenter",
    "RecoveryPointObjective": "Offline",
    "RecoveryTimeObjective": null,
    "ReplicaType": "Mirror",
    "Schedule": null,
  },
  "ConsistencyType": SequentiallyConsistent,
  "Retention": null
}
```

How it works

- **Storage Profile:** A storage service profile created by the Admin – maps to one or more Storage Pools
- **Service Catalog:** A list of Storage Profiles available for storage provisioning requests
- **Storage Request:** A storage provisioning request by a Tenant based on a Storage Profile
- **Admin:** Person that manages Storage Pools, creates Storage Profiles
- **Tenant:** Storage consumer that requests for storage through Storage Profile requests eg. End-user, application, K8S pod etc.



Integration with Openstack

- Openstack provisions storage through OpenSDS
- OpenSDS uses Cinder drivers for provisioning



Objective

- Overall long term Openstack integration goal
 - Multi-Openstack
 - Being able to replicate data across Openstack clusters
 - Manage storage across clusters



Openstack Architecture

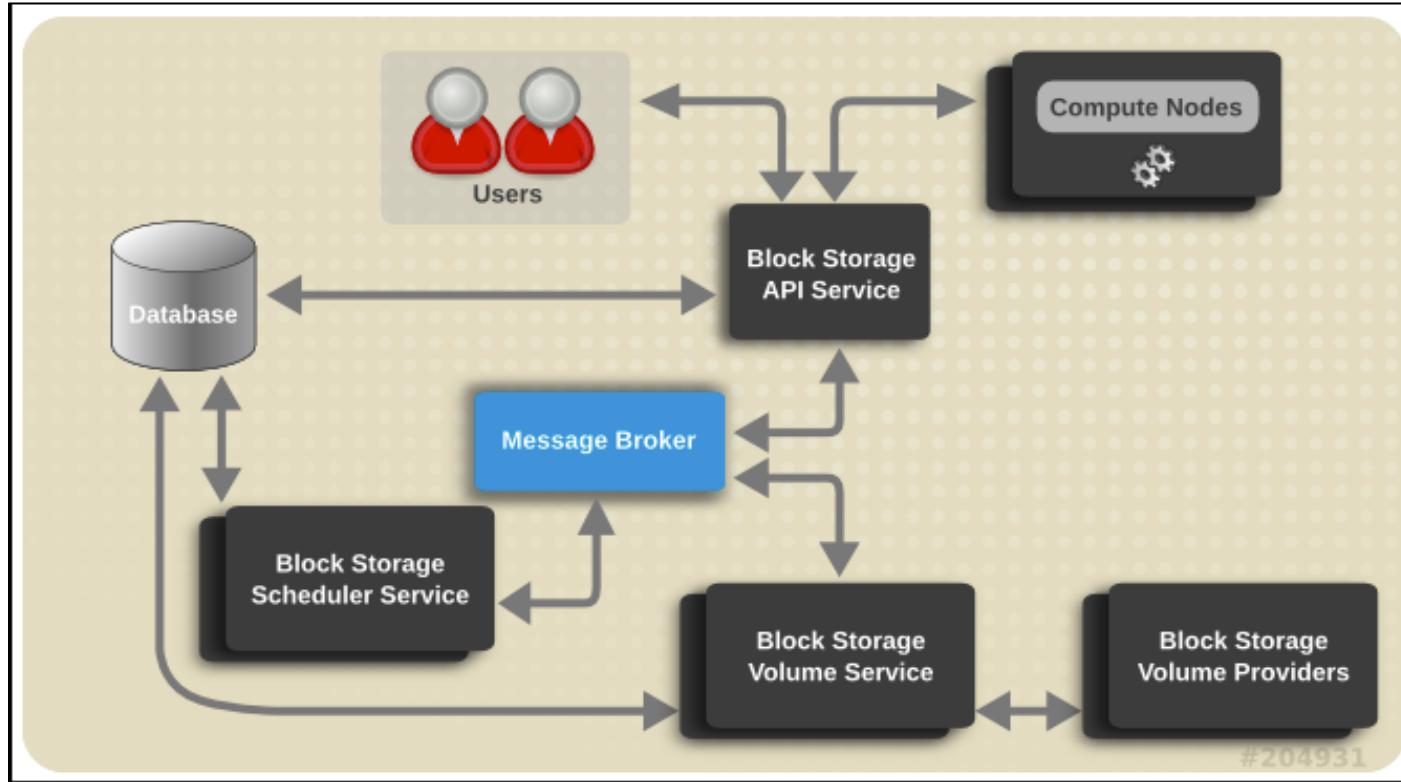
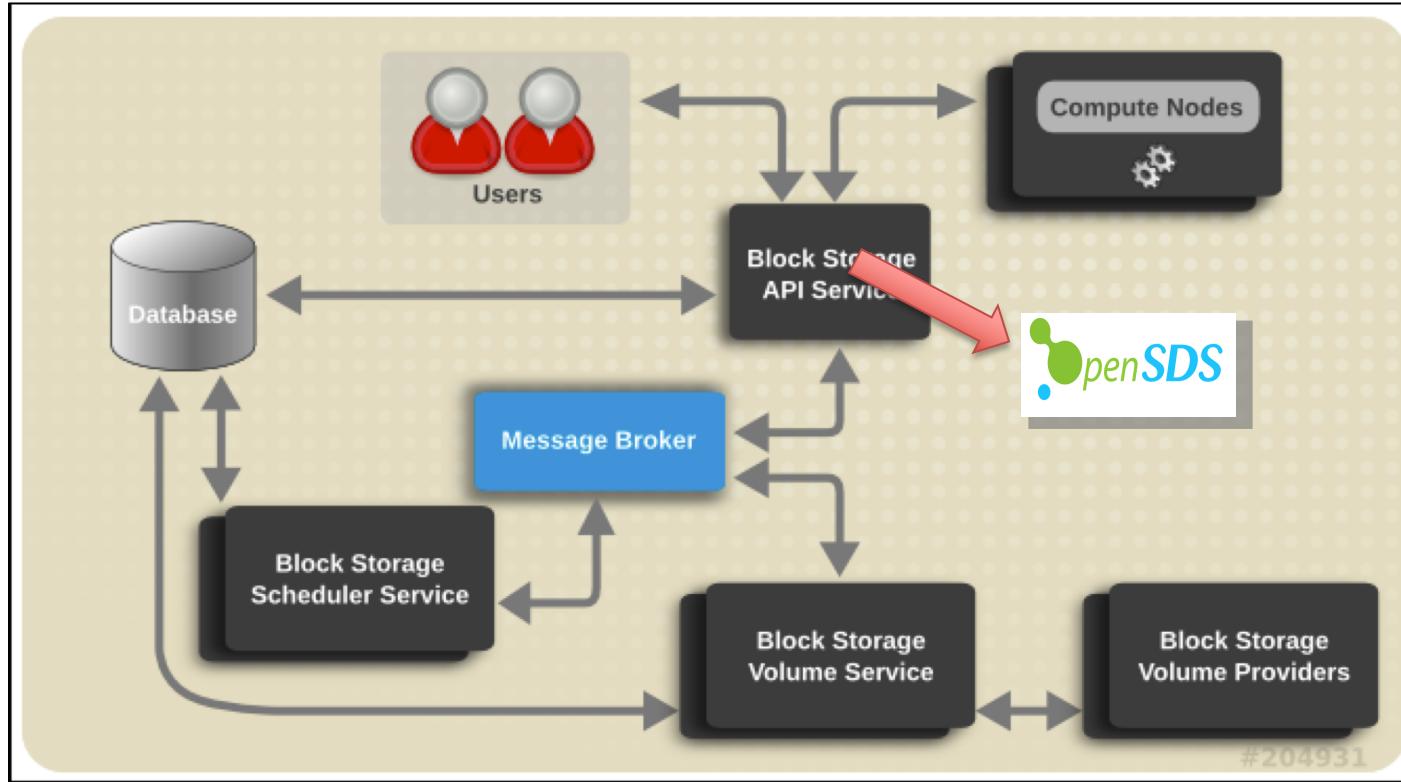


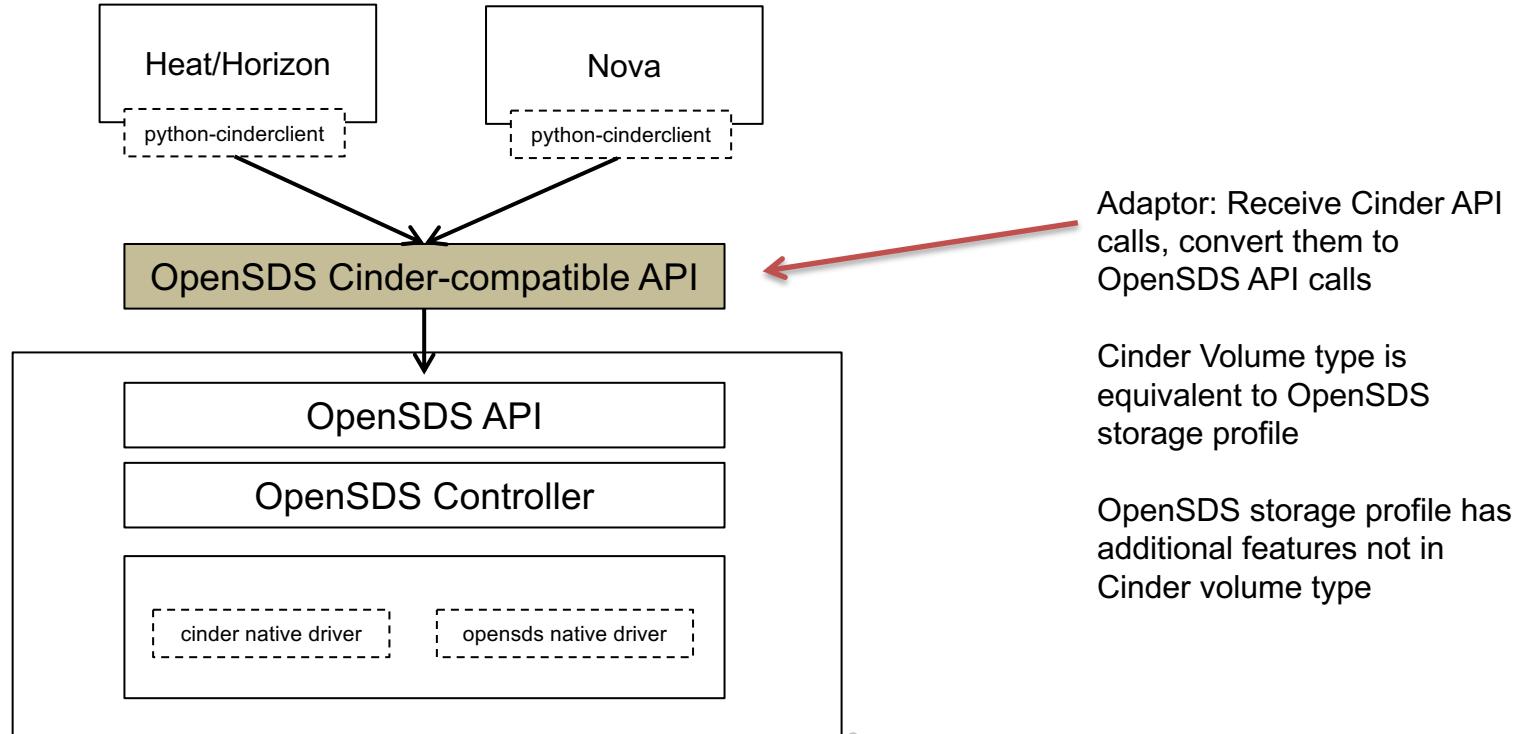
Image: Courtesy Redhat



OpenSDS Integration



OpenSDS Integration

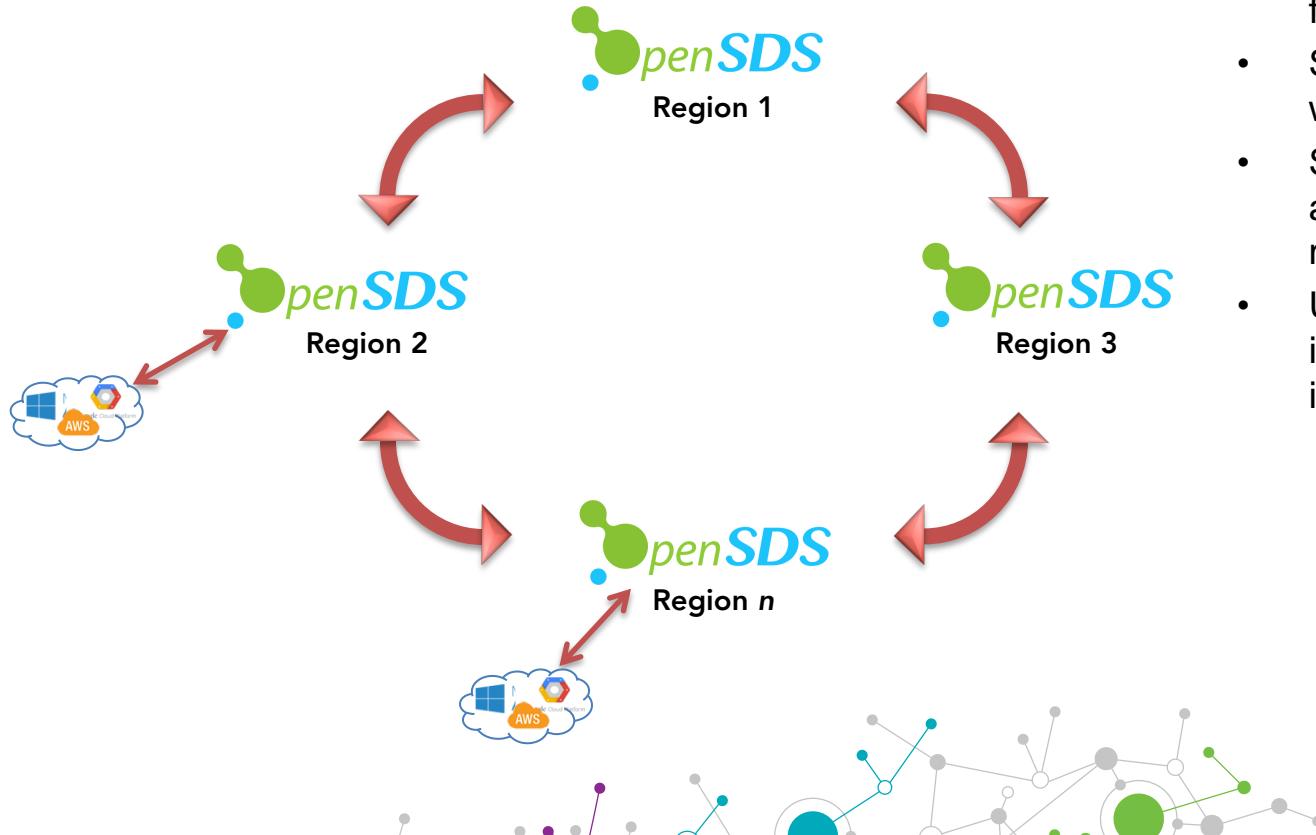


Additional Integration Points

- Integration with Keystone and Glance needed to be able to work in Openstack framework
 - Keystone:
 - To recognize projects/tenants and users
 - Authentication and multi-tenant authorization
 - Glance:
 - Create volume from image
 - Create new image i.e. upload volume as new image



Federated OpenSDS



- Connect OpenSDS instances from different regions
- Single pane of glass view of whole storage infrastructure
- Setting up data replication and protection across regions
- Using federated Keystone for identify management across instances

Integration with Openstack

- Openstack provisions storage through OpenSDS
- → OpenSDS uses Cinder drivers for provisioning



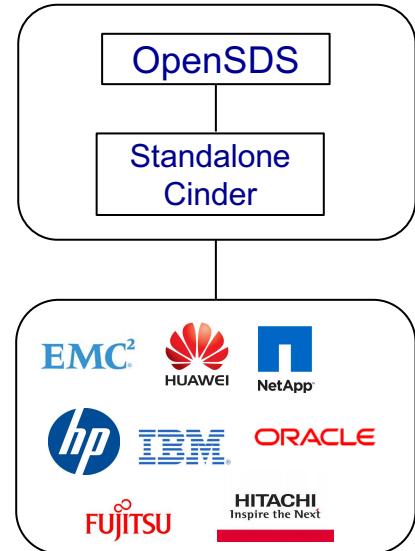
Cinder Drivers

- About eighty storage drivers are available in Cinder developed by various vendors
- No need to reinvent the wheel. These are well tested, community supported drivers



Storage Drivers

- OpenSDS leverages these Cinder drivers instead of asking vendors to create drivers from scratch for OpenSDS
- Leverages containerized standalone Cinder to integrate Cinder drivers into Openstack framework.
- Capability to support native OpenSDS drivers for advanced functionality. Preferred approach.



OpenSDS with Cloud Native



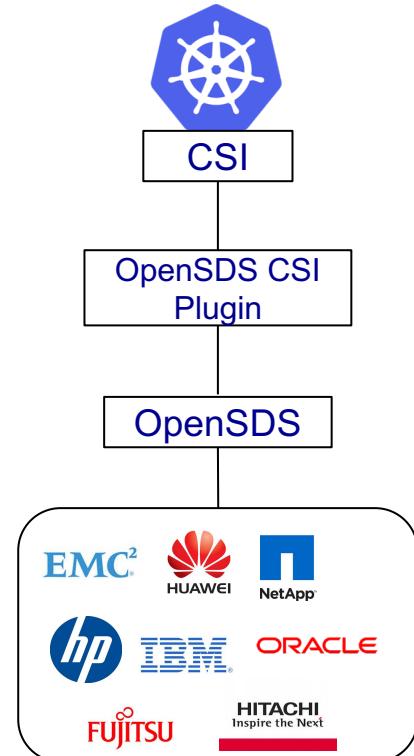
OpenSDS with Cloud Native

- Kubernetes Flexvolume plugin
- Kubernetes Service Broker plugin for OpenSDS
- CSI support



OpenSDS CSI Plugin

- Kubernetes
 - StorageClass = OpenSDS Storage Profile
 - CSI support
 - csi-provisioner-opensdsplugin
 - csi-attacher-opensdsplugin
 - csi-nodeplugin-opensdsplugin



No need to create separate CSI plugins for each vendor. Use OpenSDS CSI plugin which can already provision to a large set of storage vendors.



OpenSDS Dashboard Demo

<https://opensds.github.io/Prototype/>





A large, abstract network graph is positioned at the top right of the slide. It consists of numerous small, semi-transparent circular nodes in various colors (light gray, orange, green, blue, yellow) connected by thin, light-colored lines. A few larger, solid-colored nodes are also present, such as a large orange node on the left and a large green node on the right.

Thank You

@rakeshjn
@opensds_io

Southbound Plug-in Minimum Requirements

The minimum required functions that a southbound OpenSDS plug-in needs to implement include the following.

Zealand(v0.10, Dec 2017)

- Create/Delete Volume
- Get Volume
- Attach/Detach Volume
- Create/Delete Snapshot
- Get Snapshot

Aruba (June 2018):

- Report Capabilities (including capacities)
- Create Volume from Snapshot
- Create Cloned Volume
- Extend Volume (non-attached volume)
- Extend Volume (attached volume)

Bali (Dec 2018):

- Copy Image to Volume / Copy Volume to Image
- Manage/Unmanage Existing Volume
- Get Manageable Volumes
- Manage/Unmanage Existing Snapshot
- Get Manageable Snapshots

