Storage in Openstack:
Block Storage, Ephemeral Storage, Object Storage

Marica Antonacci - INFN Bari

Scuola di Cloud Storage
Bari, 3-6 Ottobre 2016

Outline

- Openstack intro
- Types of Storage Services
- Backend storage for Openstack components
- Ceph: de-facto storage backend for Openstack

Openstack Architecture

Openstack

- In July **2010** Rackspace Hosting and NASA jointly launched an open-source cloud-software initiative known as OpenStack

- More than **500 companies** joined the project (AMD, Cisco, EMC, Dell, IBM, Intank, Intel, Rackspace Hosting, Red Hat, SUSE Linux, VMware, Yahoo!, ..)

- **Six-month**, time-based release cycle with frequent development milestones

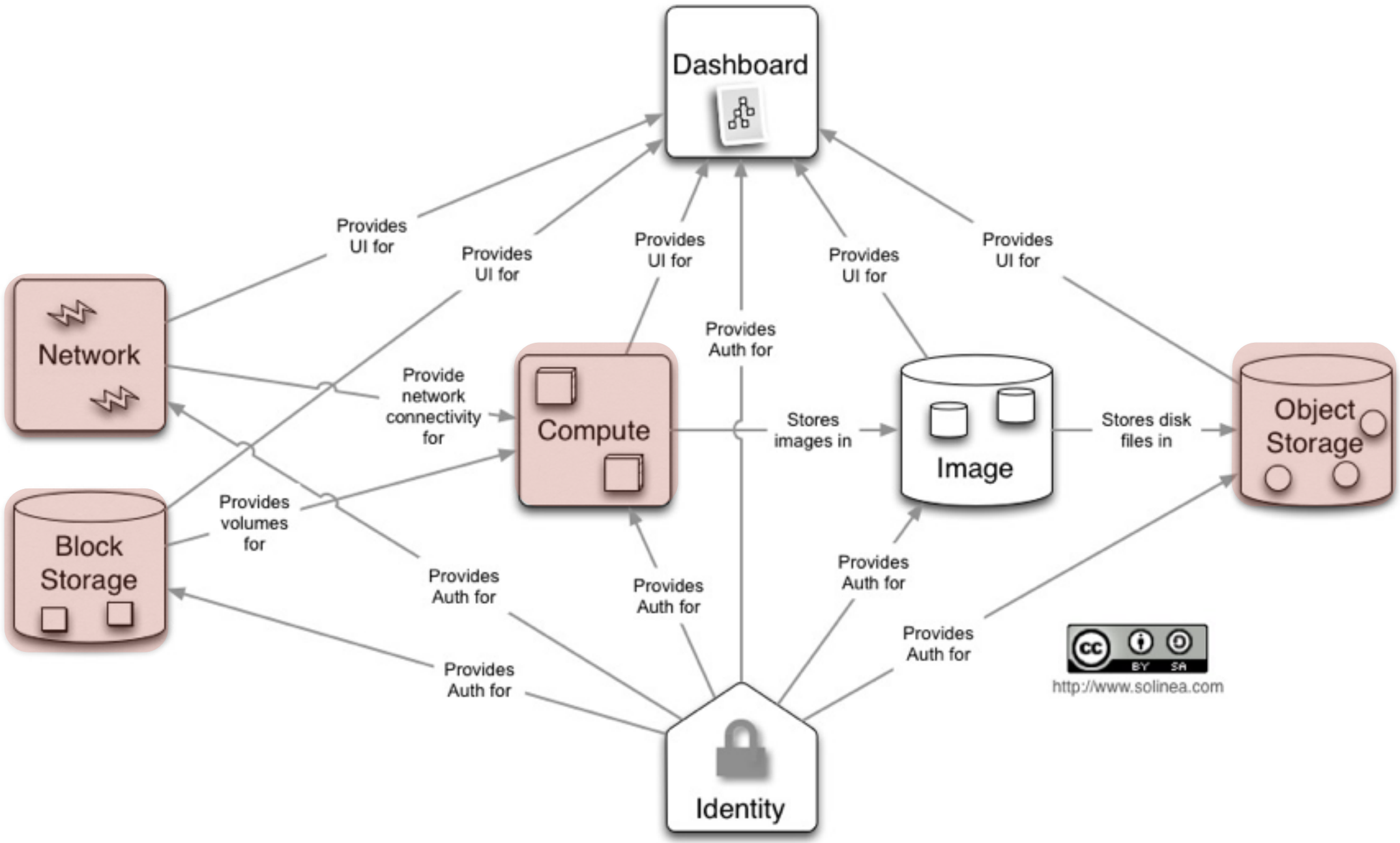
  - Current (13th) release: ***Mitaka*** - April 2016

  - Next release: ***Newton*** - Scheduled 6 October 2016

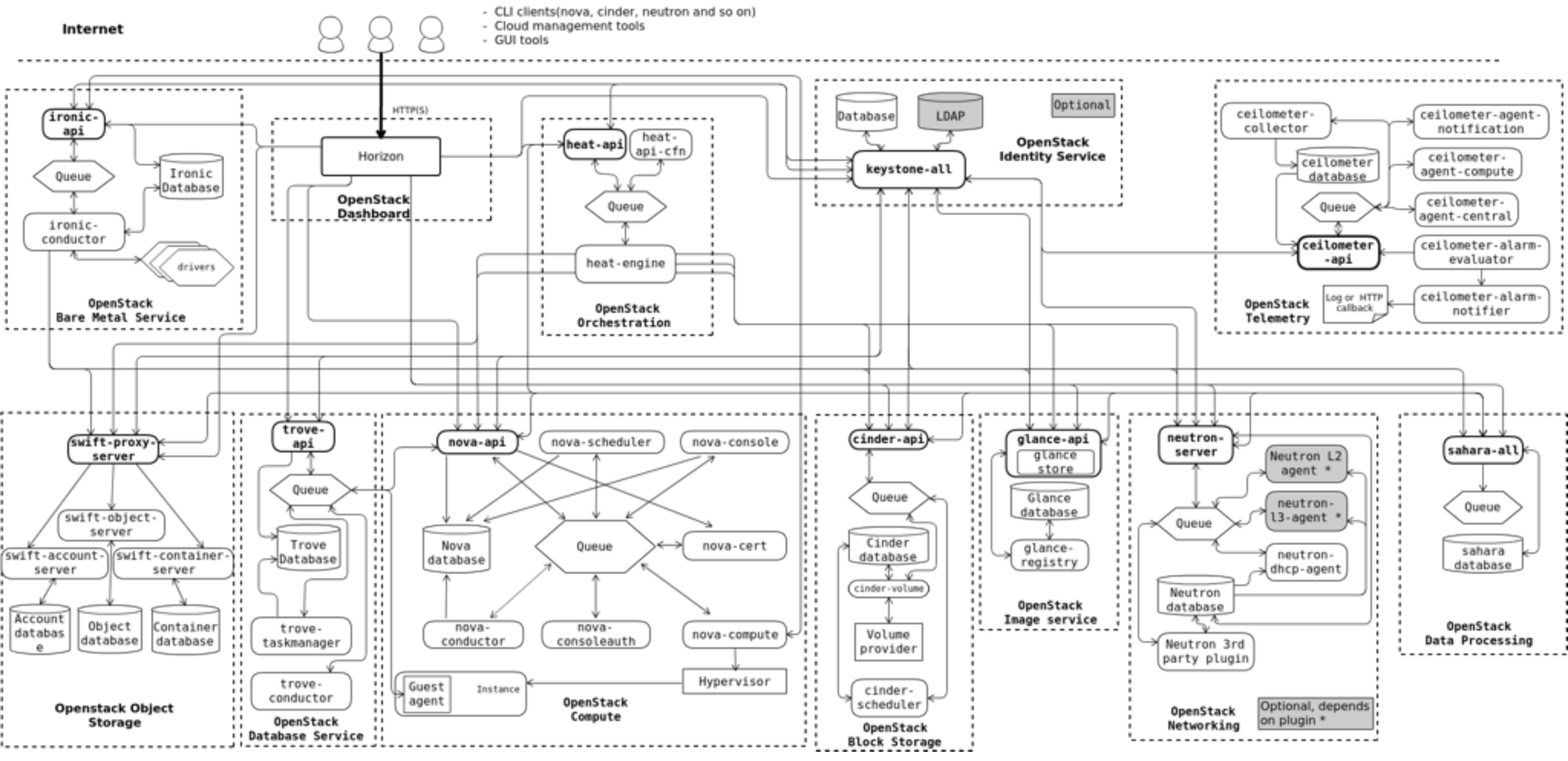
Openstack capabilities

- Virtual Machines (***VMs***) **on demand**
  - provisioning
  - snapshotting
- **Networks**
- **Storage** for VMs and arbitrary files
- **Multi-tenancy**
  - quotas for different projects, users
  - users can be associated with multiple projects

Conceptual architecture



Logical Architecture



Main components

- Keystone - Identity Service
- Nova - Compute Service
- Glance - Image Service
- Cinder - Block Storage Service
- Swift - Object Storage Service
- Neutron - Networking Service

Openstack Storage Services

Openstack Storage

1. **Ephemeral** storage with Nova
2. **Persistent Block** Storage with Cinder
3. **Object** Storage with Swift
4. **File Share** Service with Manila [optional add-on]

Ephemeral Storage

- Ephemeral storage is allocated for an instance and is deleted when the instance is deleted.
  - used to run the operating system and scratch space
- By default, Compute stores ephemeral drives as files on **local disks** on the Compute node
  - /var/lib/nova/instances
  - only VM migration moves the disk image to another compute node (Nova copies it via SSH)

- **What if the user needs to persist his data?**

Block Storage

- Add additional **persistent** storage to a virtual machine
- It is accessed through a **block device** that can be partitioned, formatted, and mounted
- Can be resized
- Persists until the user deletes it
- Can be encrypted
- **Use case**: provide persistent storage for long-running services that require strong consistency and low-latency connectivity (e.g. databases)

Object Storage

- Stores **unstructured** data, including VM images
- ***Eventually*** **consistent**
- **Highly available**. Can be replicated across different data centers
- Provides **REST** APIs (native and standard, e.g. S3, CDMI) and offer simple web services interfaces for access
- **Use-cases**: Storage for backup files database dumps, and log files; Large data sets (e.g. multimedia files); backend storage of the Image Service

File Share Storage

- A distributed file system solution that allows to expose and consume your data through a **file interface**.

- Multiple instances can access the same shared service.

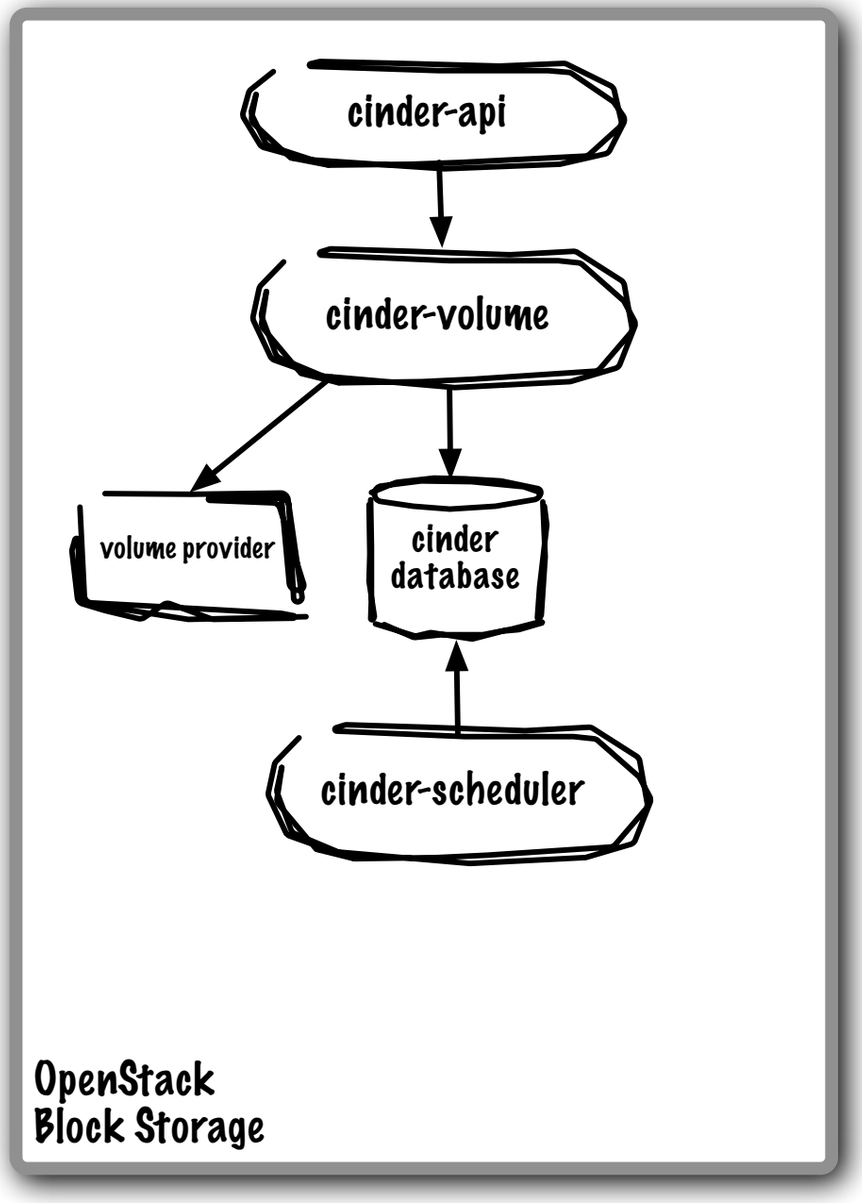
- It supports multiple backends in the form of drivers.

- Like the cinder block storage, the file system storage is also persistent and used to add persistent storage to a virtual machine and detach storage from one instance to another without data loss.

Backend Storage for Openstack components

Openstack Block Storage Service: Cinder

- Block data for volumes
- Stored in one or more backend storage devices
  - 45+ supported volume drivers (*)
- Basic functions: create, attach/detach, delete
- Advanced functions:
  - extend volumes, take snapshots and backup, clone volumes
  - QoS support
  - encryption



(*) Block Storage Drivers Support Matrix: https://wiki.openstack.org/wiki/CinderSupportMatrix

Cinder backup

- **Backups**: archived copies of a volume. Useful to prevent loss of data (Use-case: disaster-recovery)

- Managed by a separate service: **cinder-backup** (not installed in the default configuration)

- Configurable drivers:

  ➡ Swift

  ➡ Ceph

  ➡ GlusterFS

  ➡ NFS (since Kilo)

  ➡ IBM Tivoli Storage Manager

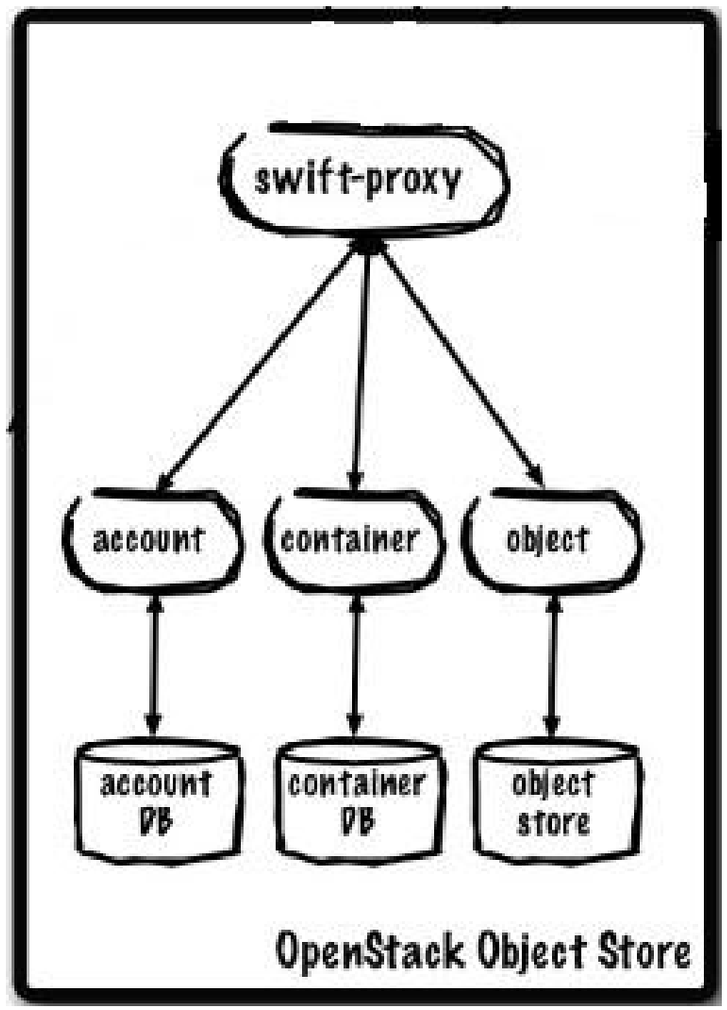
  ➡ Google Cloud Storage (since Mitaka)

New features in Mitaka

- Volume **Replication**
  - "primitive base level" enabling the automatic replication of data between storage arrays of the same type in different data centers
  - purely for disaster recovery
  - Drivers: SolidFire, IBM, Dell, EMC, HP, Huawei, Storewize, Pure Storage
- **Consistency** groups
- Ability to backup snapshots

Openstack Object Storage Service: Swift

- Swift is a highly available, distributed, eventually consistent object/blob store
- By default, Swift places three copies of every object in as **unique-as-possible** locations -- first by region, then by zone, server and drive
- Provides **RESTfull** APIs
- **Multi-site** deployment
- ***Container-to-container*** synchronization

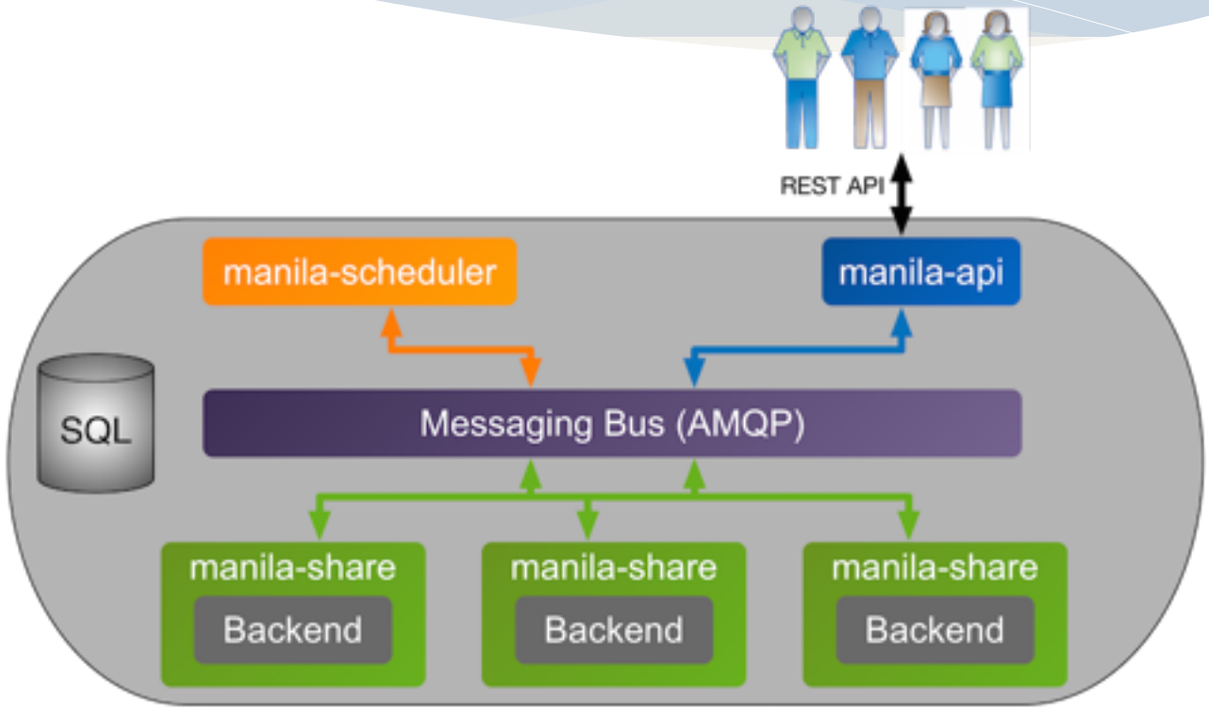


Swift new features

- **Data-at-rest encryption** (swift 2.9.0+)
  - implemented in the proxy server
  - each object is encrypted with it's own unique, randomly-chosen key
- **ProxyFS** file system access
  - to allow completely ***bimodal*** access so that you can read and write via a file system and can read and write via the Swift API

Openstack File Share Service: Manila

- Manila drivers include
  ZFS on Linux
  LVM,
  CephFS native,
  GlusterFS (NFS or native),
  HDFS,
  GPFS,
  NetAPP Clustered Data ONTAP,
  Tegile IntelliFlash

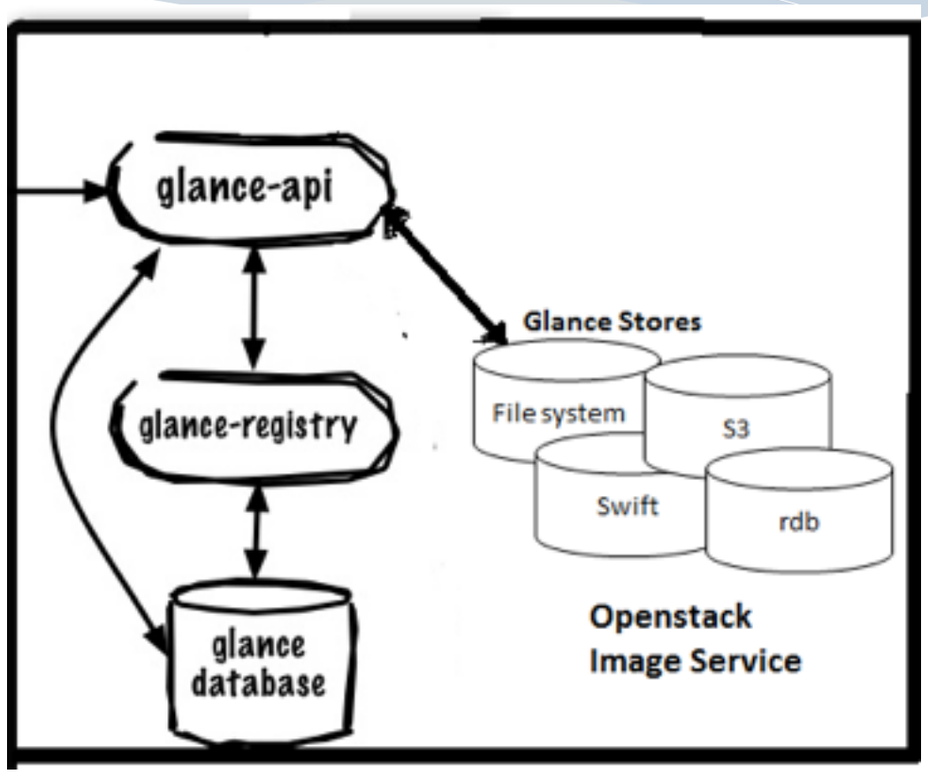


- Share replication allows tenants to configure and manage replication relationships between OpenStack availability zones to ensure data availability even if an availability zone fails.

Features support mapping:
http://docs.openstack.org/developer/manila/devref/share_back_ends_feature_support_mapping.html

The Image Service: Glance

- The primary objective of Glance is to publish a catalog of virtual machine images.

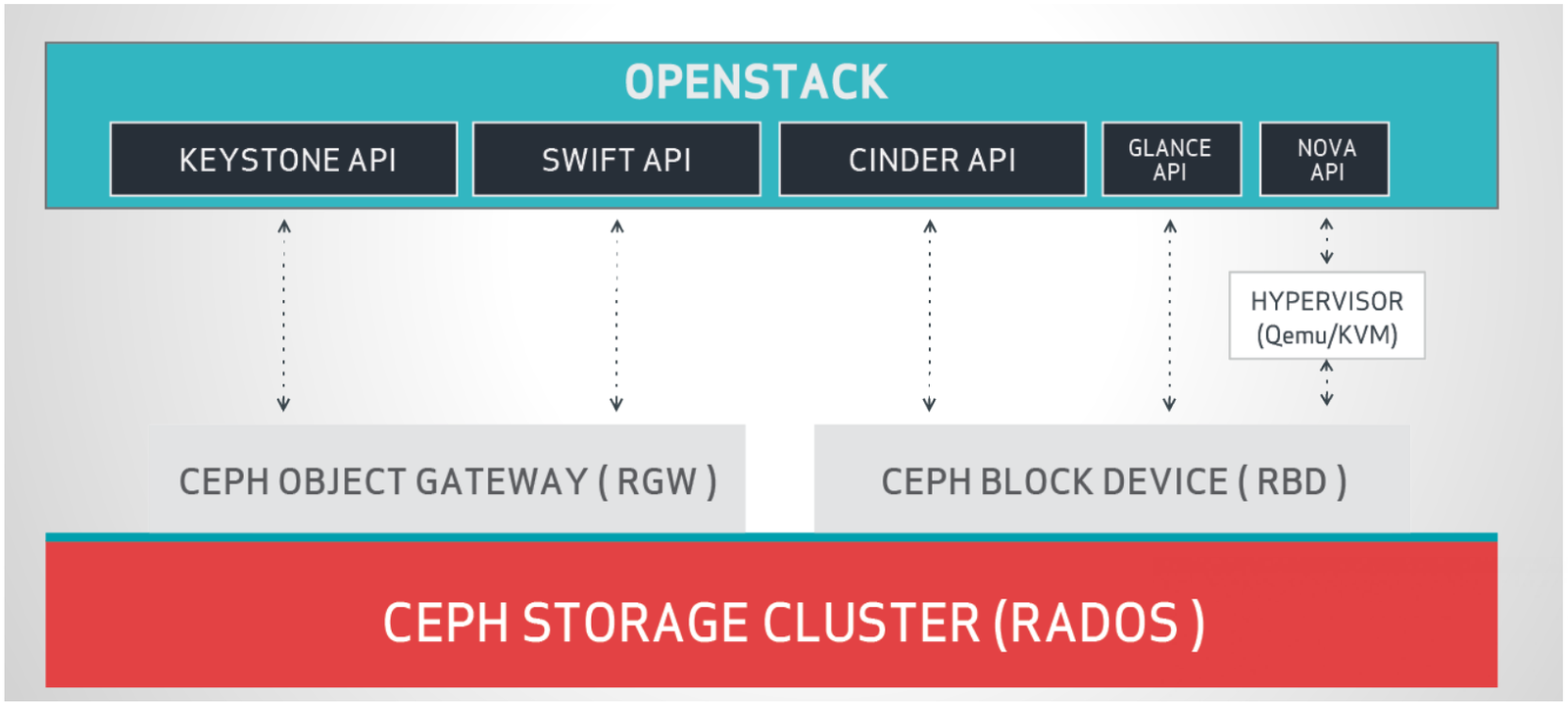


- Main components:
  - **glance-api**: accepts Image API calls for image discovery, retrieval and storage
  - **glance-registry**: stores, processes, and retrieves metadata for images
  - **storage backend** (filesystem, rbd, swift, s3, cinder, etc.)

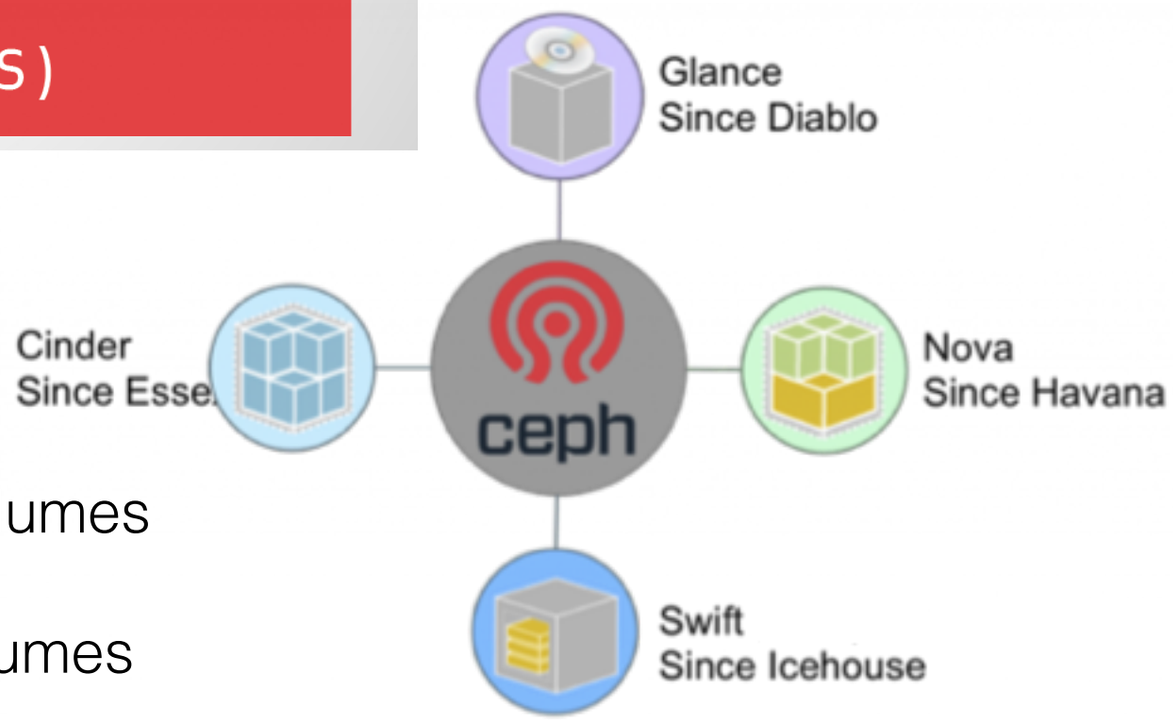
Glance images

- Image block data
- Read-only
- Can be massive file sizes (100+ GB for some Windows images)
- Huge array of backend store drivers
  - ➡ Worst option: filesystem (unless it's a shared filesystem)
  - ➡ Better options: rbd, sheepdog, swift and s3
- These are distributed storage systems with built-in redundancy
- Choose one based on degree of familiarity, size of deployment

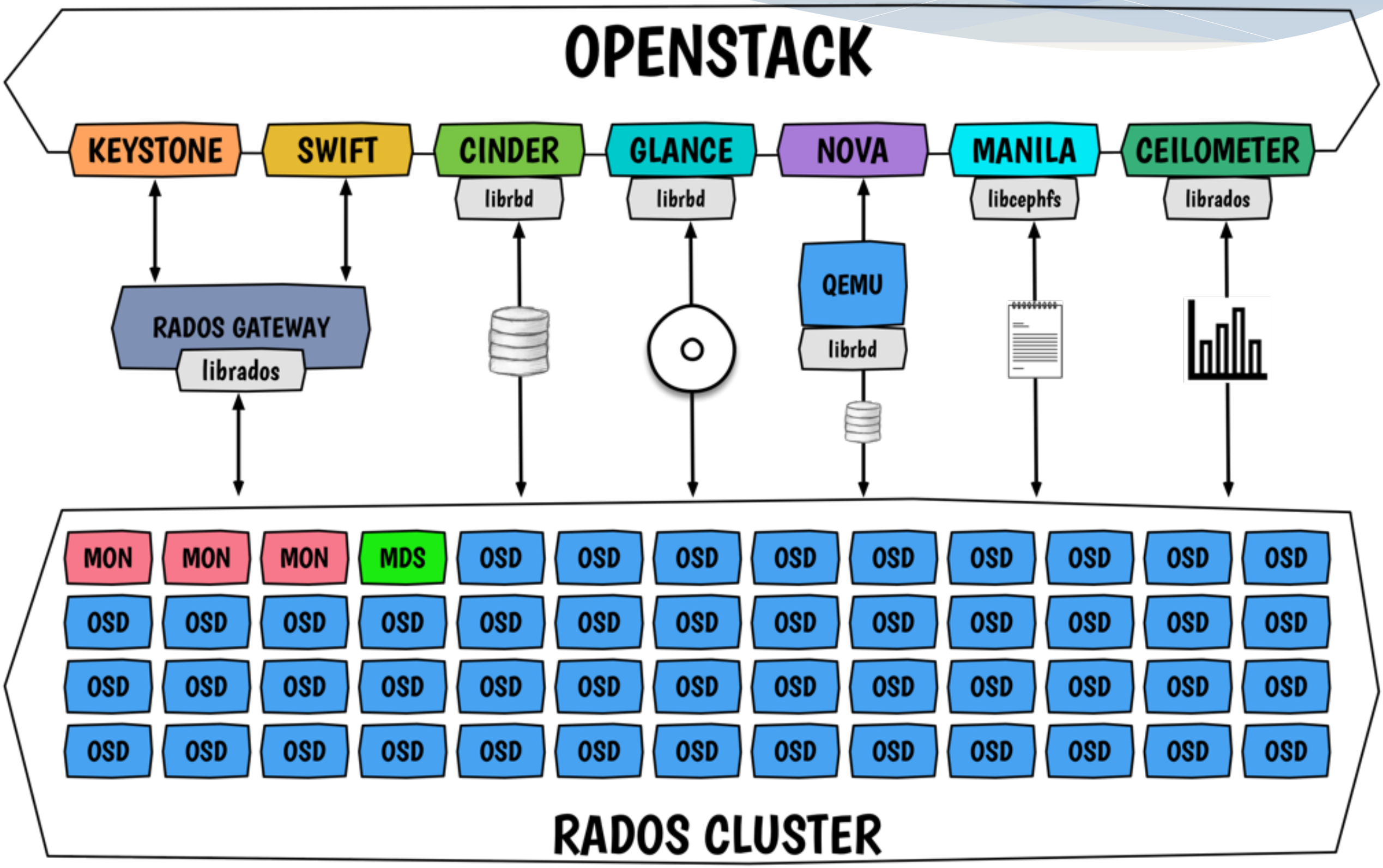
Ceph: de-facto storage backend for Openstack



- Storage consolidation:
  - Glance image storage in RADOS
  - Cinder provisioning of persistent RBD volumes
  - Nova provisioning of ephemeral RBD volumes
  - Swift and Keystone compatible RADOS



The OpenStack Ceph Galaxy



2016 OpenStack User Survey:
Which OpenStack Block Storage (Cinder) drivers are in use?

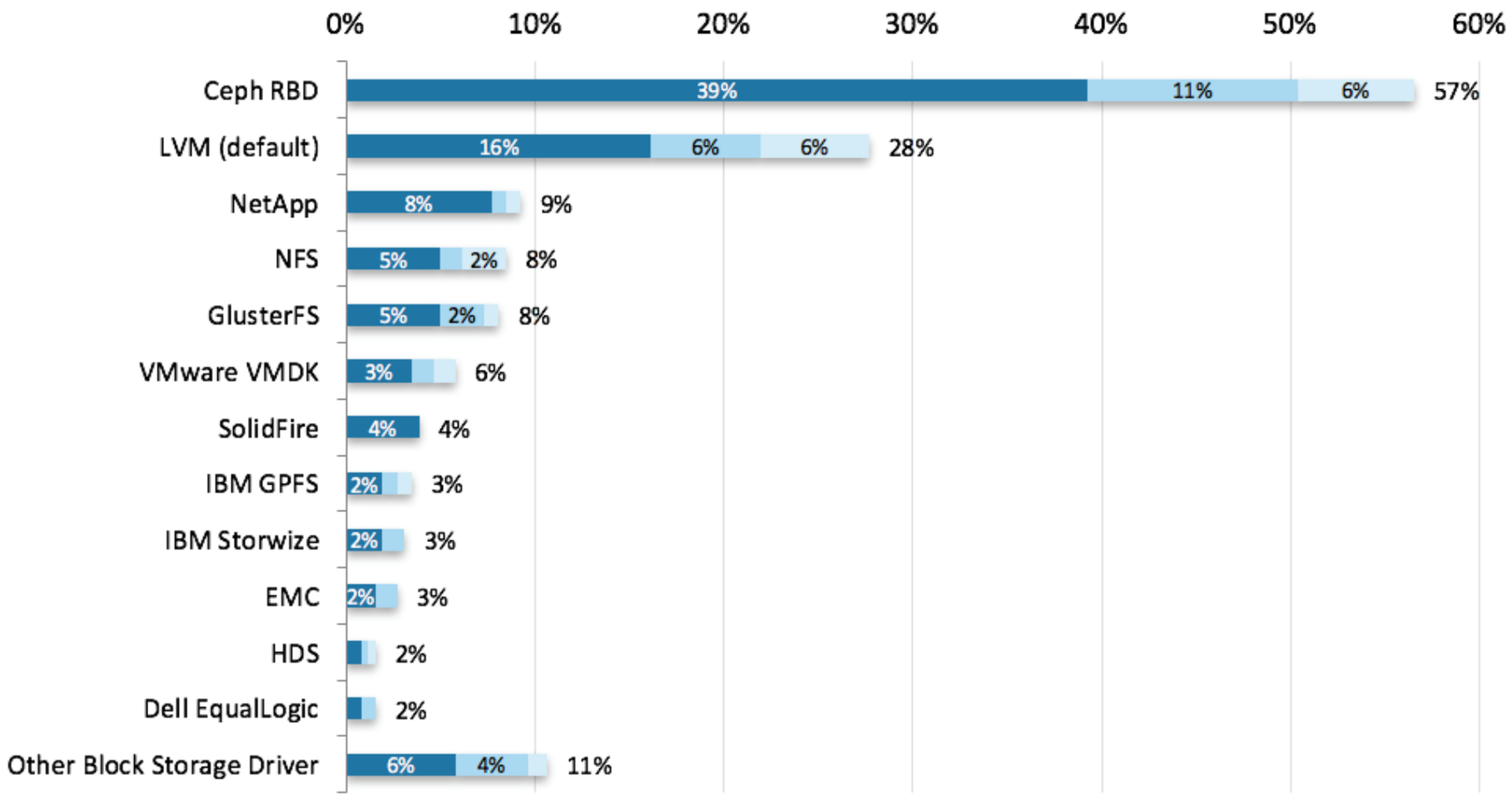


Figure 5.7 n=260

Percentages are rounded to the nearest whole number; bar length shows fractions.

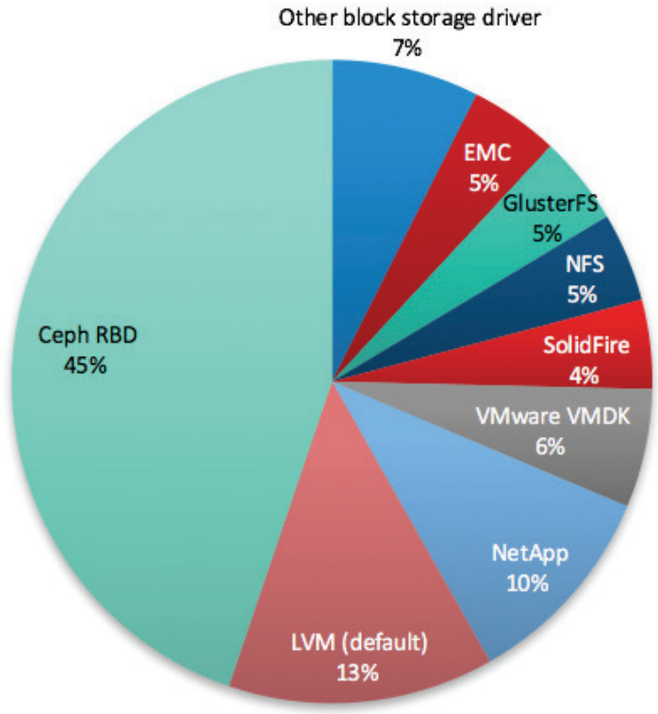


Figure 5.7.1 n=47

Production
Dev/ QA
Proof of Concept

Dedicated pools and users

- Three different pools: *images*, *volumes*, *vms, [backups]*

```
ceph osd pool create volumes 128
ceph osd pool create images 128
ceph osd pool create vms 128
```

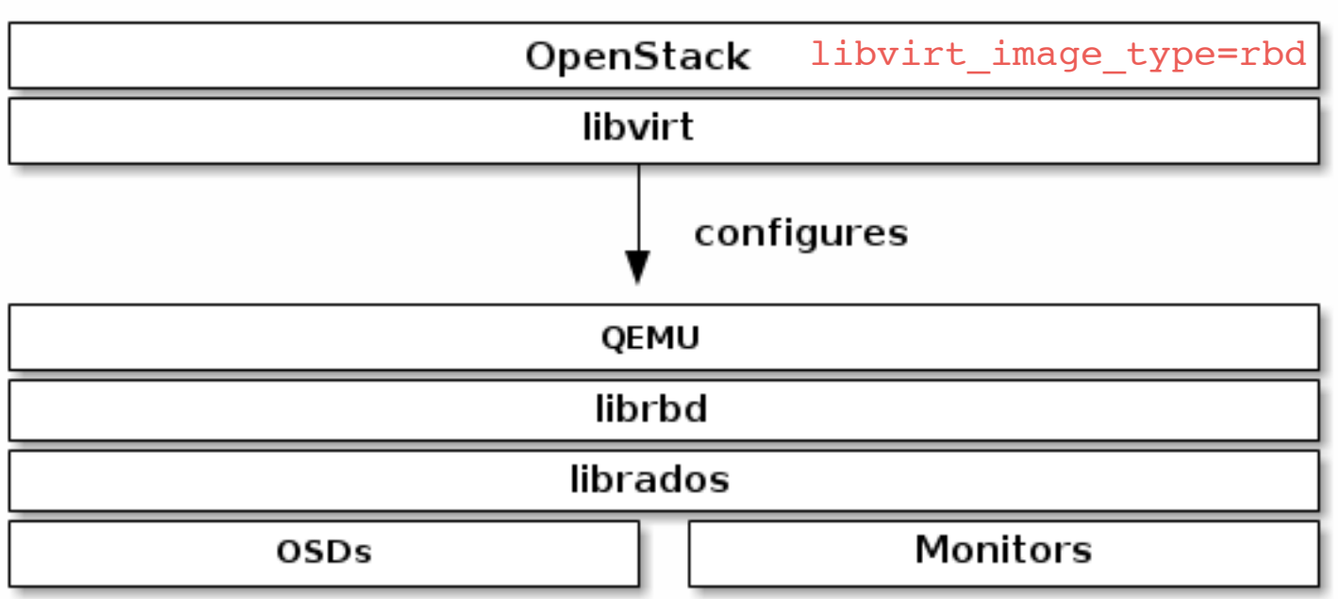
- Dedicated and right-limited user to access the pools
  - prior Icehouse, we had to use client.admin in libvirt to authenticate and interact with the Ceph cluster

```
ceph auth get-or-create client.cinder mon 'allow r' osd 'allow
class-read object_prefix rbd_children, allow rwx pool=volumes,
allow rwx pool=vms, allow rx pool=images’

ceph auth get-or-create client.glance mon 'allow r' osd 'allow
class-read object_prefix rbd_children, allow rwx pool=images'
```

Libvirt + RBD

- ✓ Decouple VM storage from hypervisors
- ✓ Images stored in RADOS
- ✓ Snapshots
- ✓ Live migration
- ✓ Thin provisioning
- ✓ Copy on write cloning
- ✓ Images striped across storage pool



Limitations

- Ceph doesn't support **QCOW2** for hosting virtual machine disk
- nova-compute checks the image format before booting the machine
  - QCOW2 images are converted to RAW
- Instance snapshot
  - **Generic** method (default prior to **Liberty**): snapshot written to the compute local disk then pushed back up to glance **—> Slow and inefficient process!!.**
  - **Direct** method (**Mitaka**): can be enabled if RBD is used to back both nova and glance exploiting copy-on-write clones. An RBD snapshot is taken in Nova and cloned into Glance.

Ceph backup service

- Ceph driver allows backing up volumes of any type to a Ceph object store

- Ceph driver is also capable of detecting if the source volume is stored on the same kind of backend, i.e. Ceph RBD

- In this case, it attempts to perform an incremental backup, falling back to full backup/copy if the former fails.

- It also supports backing up…

  ✓ within the same pool (not recommended)

  ✓ between two different pools

  ✓ between two different Ceph clusters

Ceph backup: under the hood

Workflow executed for the first backup of a volume

1. Create a base backup image used for storing differential exports
2. Snapshot source volume to create a new point-in-time
3. Perform differential transfer:

```
rbd export-diff --id cinder --conf /etc/ceph/ceph.conf --pool volumes volumes/volume-afa33905-0d87-42ff-
ad36-9c75fdcf09ba@backup.4e50e949-3dcd-4ff1-89e0-a6a9c1beb5c1.snap.1418722200.64 -

rbd import-diff --id cinder-backup --conf /etc/ceph/ceph.conf --pool backups - backups/volume-
afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base
```

Results in rbd:

```
# rbd -p volumes ls -l
NAME                                                                                 SIZE PARENT FMT
PROT LOCK
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba                                       10240M          2
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba@backup.4e50e949-3dcd-4ff1-89e0-a6a9c1beb5c1.snap.1418722200.64 10240M          2

# rbd -p backups ls -l
NAME                                                                                                SIZE
PARENT FMT PROT LOCK
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base                                          10240M
2
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base@backup.4e50e949-3dcd-4ff1-89e0-a6a9c1beb5c1.snap.1418722200.64 10240M
2
```

Ceph backup: under the hood (2)

Workflow executed for the next backups

1. Snapshot source volume to create a new point-in-time
2. Perform differential transfer using --from-snap:

```
rbd export-diff --id cinder --conf /etc/ceph/ceph.conf --pool volumes --from-snap backup.
4e50e949-3dcd-4ff1-89e0-a6a9c1beb5c1.snap.1418722200.64 volumes/volume-afa33905-0d87-42ff-
ad36-9c75fdcf09ba@backup.c255e3ca-f01b-4fe6-ad9f-af0524a7b531.snap.1418725945.25 -

rbd import-diff --id cinder-backup --conf /etc/ceph/ceph.conf --pool backups - backups/volume-
afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base
```

Results in rbd:

```
# rbd -p volumes ls -l
NAME                                                                                 SIZE PARENT FMT
PROT LOCK
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba                                       10240M          2
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba@backup.c255e3ca-f01b-4fe6-ad9f-af0524a7b531.snap.1418725945.25 10240M          2

# rbd -p backups ls -l
NAME                                                                                                  SIZE
PARENT FMT PROT LOCK
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base                                            10240M
2
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base@backup.4e50e949-3dcd-4ff1-89e0-a6a9c1beb5c1.snap.1418722200.64 10240M
2
volume-afa33905-0d87-42ff-ad36-9c75fdcf09ba.backup.base@backup.c255e3ca-f01b-4fe6-ad9f-af0524a7b531.snap.1418725945.25 10240M
2
```

Manila with CephFS

- CephFS native
- **Jewel** and **Mitaka**
- CephFSVolumeManager to orchestrate shares
  - CephFS directories
  - with quota
- VM mounts CephFS directory (ceph-fuse, kernel client, …)
- tenant VM talks directory to Ceph cluster; deploy with caution

Manila - Ceph Kraken/Luminous

- Manila hypervisor-mediated FsaaS
  - Terminate CephFS on hypervisor host, expose to guest locally via NFS over VSOCK
  - Guest no longer needs any auth or addr info: connect to 2:// (the hypervisor) and it'll get there
  - new Manila driver
  - new Nova API to attach shares to VMs