OpenStack Cinder Deep Dive
Grizzly Release
Cinder Overview

- **Cinder manages persistent storage**
  - Data volumes that are attached to VM instances
  - Boot from volume

- **Project exists since Folsom release, spun off from Nova-volume**

- **Volumes have a lifecycle independent of VM instances**

- **For example:**
  - Cinder: create volume
  - Nova: boot VM instance
  - Nova: attach volume to instance (will call Cinder)
  - (More details later)
Architectural Overview
Architectural Overview
cinder-api

- **Volume create/delete/list/show**
  - Create from volume, image, snapshot
- **Snapshot create/delete/list/show**
- **Volume attach/detach (called by Nova)**
- **Others:**
  - Volume types (more later)
  - Quotas
  - Backups
Architectural Overview
cinder-volume

Main components:

- API: for cinder-api to communicate with cinder-volume
- Manager: Generic code to implement API
- Drivers: Called by Manager, contains back-end-specific code to communicate with various storage types (e.g., Linux LVM, storage controllers from various vendors, distributed file systems, etc.)

- Admin can run multiple cinder-volume instances, each with its own configuration file describing settings and the storage back-end
- As of Grizzly, one cinder-volume instance can manage multiple back-ends
- Each back-end driver is generally configured to interact with one storage pool
- Multi-threading
Example: High-Level Data and Control Flow 1

Legend
- Teal: Persistent volume control
- Purple: Persistent volume data
Example: High-Level Data and Control Flow 2

Note that iSCSI is just an example – several additional protocols are supported (e.g., FC, NFS)

Legend
- Persistent volume control
- Persistent volume data
Example: Flow for attach a volume to instance

1. **Nova calls Cinder via its API, passing connection information**
   - e.g., host name, iSCSI initiator name, FC WWPNs
2. **cinder-api passes message to cinder-volume**
3. **Manager does initial error checking and calls volume driver**
4. **Volume driver does any necessary preparation to allow the connection**
   - e.g., give the nova host permissions to access the volume
5. **Volume driver returns connection information, which is passed to Nova**
   - e.g., iSCSI iqn and portal, FC WWPN
6. **Nova creates the connection to the storage using the returned information**
7. **Nova passes the volume device/file to the hypervisor**
Architectural Overview

cinder client

REST

AMPQ

AMPQ

AMPQ

SQL DB

cinder-schedule

cinder-api

cinder-volume

cinder-backup
cinder-scheduler

- **Chooses which back-end to place a new volume on**
- **Configurable plugins for schedulers**
- **Filter scheduler has plugins for filters and weights**

**Filter scheduler:**
1. **Starts with list of all back-ends**
2. **Filters according to capabilities**
   - Drivers report capabilities and state (e.g., free space)
   - Admins create volume_types which specify requirements
   - Users optionally specify a volume_type when creating a volume
3. **Sorts according to weights**
   - e.g., available free space
4. **Returns best candidate**
Scheduler / volume_types example

Create Volume

Volume Name
web-volume

Description
Additional information here...

Type

tier1
tier1b
**tier2**
tier2c

Description:
Volumes are block devices that can be attached to instances.

Volume Quotas

Total Gigabytes (350 GB) 650 GB Available
Number of Volumes (3) 7 Available

Cancel  Create Volume
Scheduler / volume_types example

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>web-volume</td>
<td></td>
<td>300GB</td>
<td>Available</td>
<td>tier2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>log-volume</td>
<td></td>
<td>100GB</td>
<td>Available</td>
<td>tier2c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>data-volume</td>
<td></td>
<td>200GB</td>
<td>Available</td>
<td>tier1b</td>
<td></td>
<td></td>
</tr>
<tr>
<td>meta-volume</td>
<td></td>
<td>50GB</td>
<td>Available</td>
<td>tier1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Scheduler / volume_types example

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>web-volume</td>
<td></td>
<td>300GB</td>
<td>Available</td>
<td>tier2</td>
<td>HDDs</td>
<td>Edit Attachments</td>
</tr>
<tr>
<td>log-volume</td>
<td></td>
<td>100GB</td>
<td>Available</td>
<td>tier2c</td>
<td>Compressed</td>
<td>Edit Attachments</td>
</tr>
<tr>
<td>data-volume</td>
<td></td>
<td>200GB</td>
<td>Available</td>
<td>tier1b</td>
<td>Flash/HDD</td>
<td>Edit Attachments</td>
</tr>
<tr>
<td>meta-volume</td>
<td></td>
<td>50GB</td>
<td>Available</td>
<td>tier1</td>
<td>Flash</td>
<td>Edit Attachments</td>
</tr>
</tbody>
</table>
Scheduler / volume_types example

The image shows the OpenStack Dashboard with a list of volumes. The volumes and their details are as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Size</th>
<th>Status</th>
<th>Type</th>
<th>Attached To</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>web-volume</td>
<td></td>
<td>300GB</td>
<td>In-Use</td>
<td>tier2</td>
<td>Attached to web-vm on /dev/vdb</td>
<td>Edit Attachments</td>
</tr>
<tr>
<td>log-volume</td>
<td></td>
<td>100GB</td>
<td>In-Use</td>
<td>tier2</td>
<td>Attached to db-vm on /dev/vdd</td>
<td>Edit Attachments</td>
</tr>
<tr>
<td>data-volume</td>
<td></td>
<td>200GB</td>
<td>In-Use</td>
<td>tier1b</td>
<td>Attached to db-vm on /dev/vdc</td>
<td>Edit Attachments</td>
</tr>
<tr>
<td>meta-volume</td>
<td></td>
<td>50GB</td>
<td>In-Use</td>
<td>tier1</td>
<td>Attached to db-vm on /dev/vdb</td>
<td>Edit Attachments</td>
</tr>
</tbody>
</table>
Looking Forward to Havana: Proposed Features

- Code cleanup and reorganization
- Attach volume to multiple hosts
- Read-only volumes
- ACLs
- Disk encryption
- FC SAN Zone / Access Control management
- Transfer volume ownership
- Volume Migration
- Work towards locality between instances and volumes
- Scheduler hints
- Volume rate limiting
Looking Forward to Havana: Proposed Drivers

- IBM GPFS
- IBM zVM
- EMC Isilon (iSCSI)
- Local disk partitions
- Hitachi HUS (DF850) (iSCSI)
- Dell Equallogic
- Violin Memory v6000 (iSCSI)

Current drivers:
- Coraid (AoE)
- EMC VMAX/VNX (iSCSI)
- GlusterFS (GlusterFS)
- HP 3PAR (iSCSI/FC)
- HP LeftHand (iSCSI)
- Huawei T-series/Dorado (iSCSI)
- IBM Storwize family/SVC (iSCSI/FC)
- IBM XIV (iSCSI), LVM (iSCSI)
- NetApp (iSCSI/NFS)
- Nexenta (iSCSI)
- NFS (NFS)
- RBD (Ceph)
- Scality SOFS (scality)
- Sheepdog (sheepdog)
- Solaris (iSCSI)
- SolidFire (iSCSI)
- Windows Server 2012 (iSCSI)
- Zadara (iSCSI)
Thank you!

Questions?