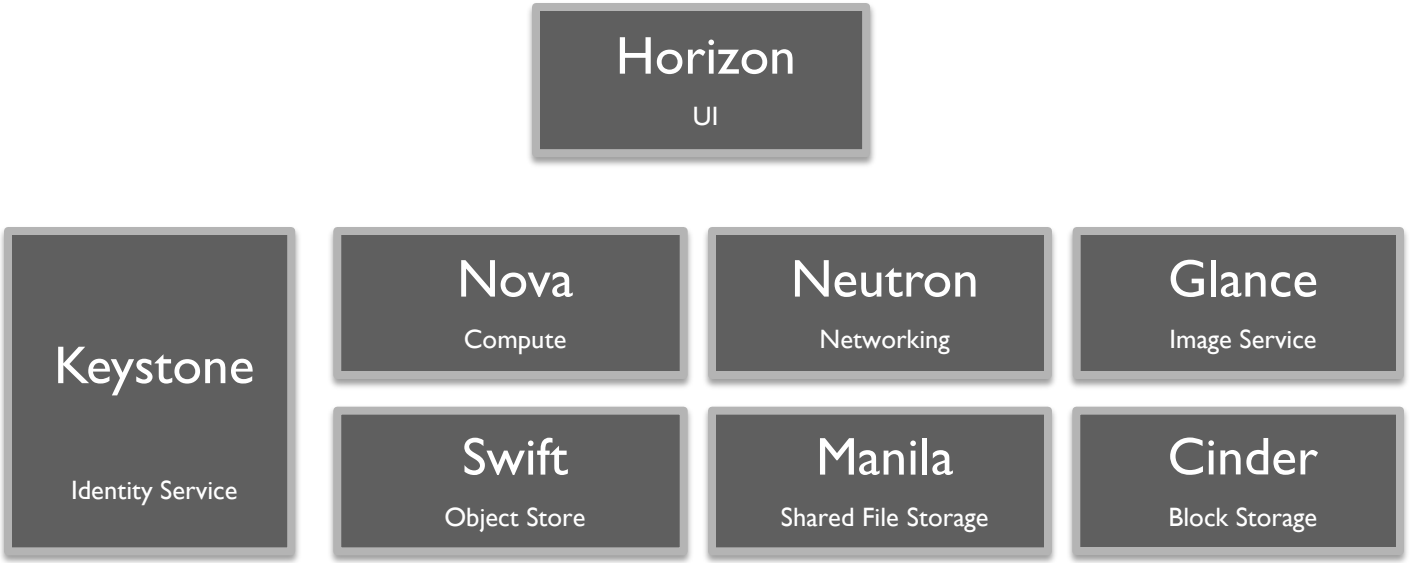
SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2016

Introduction to OpenStack Cinder

Sean McGinnis
Dell

OpenStack Components



Cinder Mission Statement

To implement services and libraries to provide on demand, self-service access to Block Storage resources. Provide Software Defined Block Storage via abstraction and automation on top of various traditional backend block storage devices.

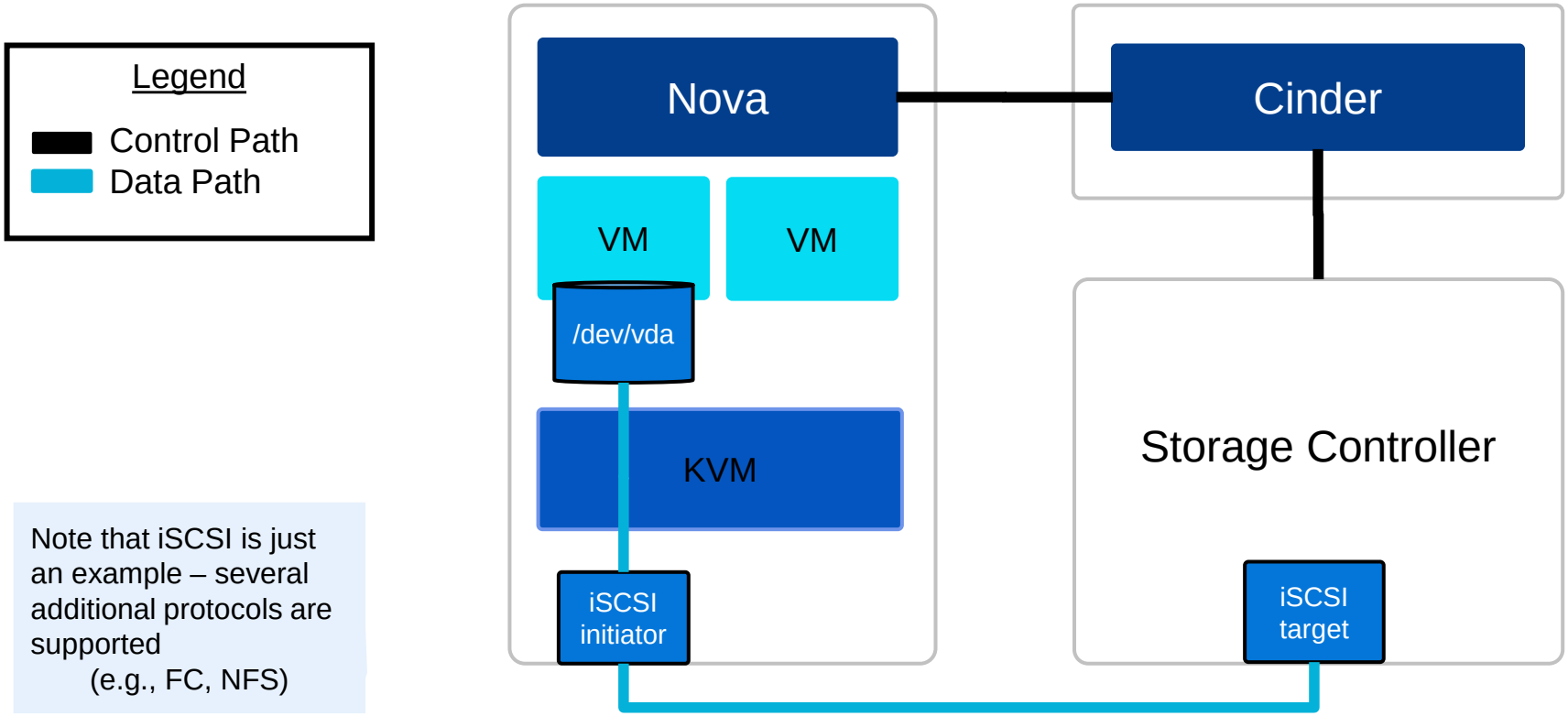
What is Cinder

- Created in the OpenStack Folsom Release (2012)
  - Spun off from Nova volume
- Cinder manages block storage
  - Not shared file – Manila
  - Not object storage – Swift
  - Provides management abstraction over a variety of backends
  - Provides:
    - Create/delete
    - Attach/detach
    - Snapshots
    - Backup
- Volumes have lifecycles independent of VMs

Where Does Cinder Fit?

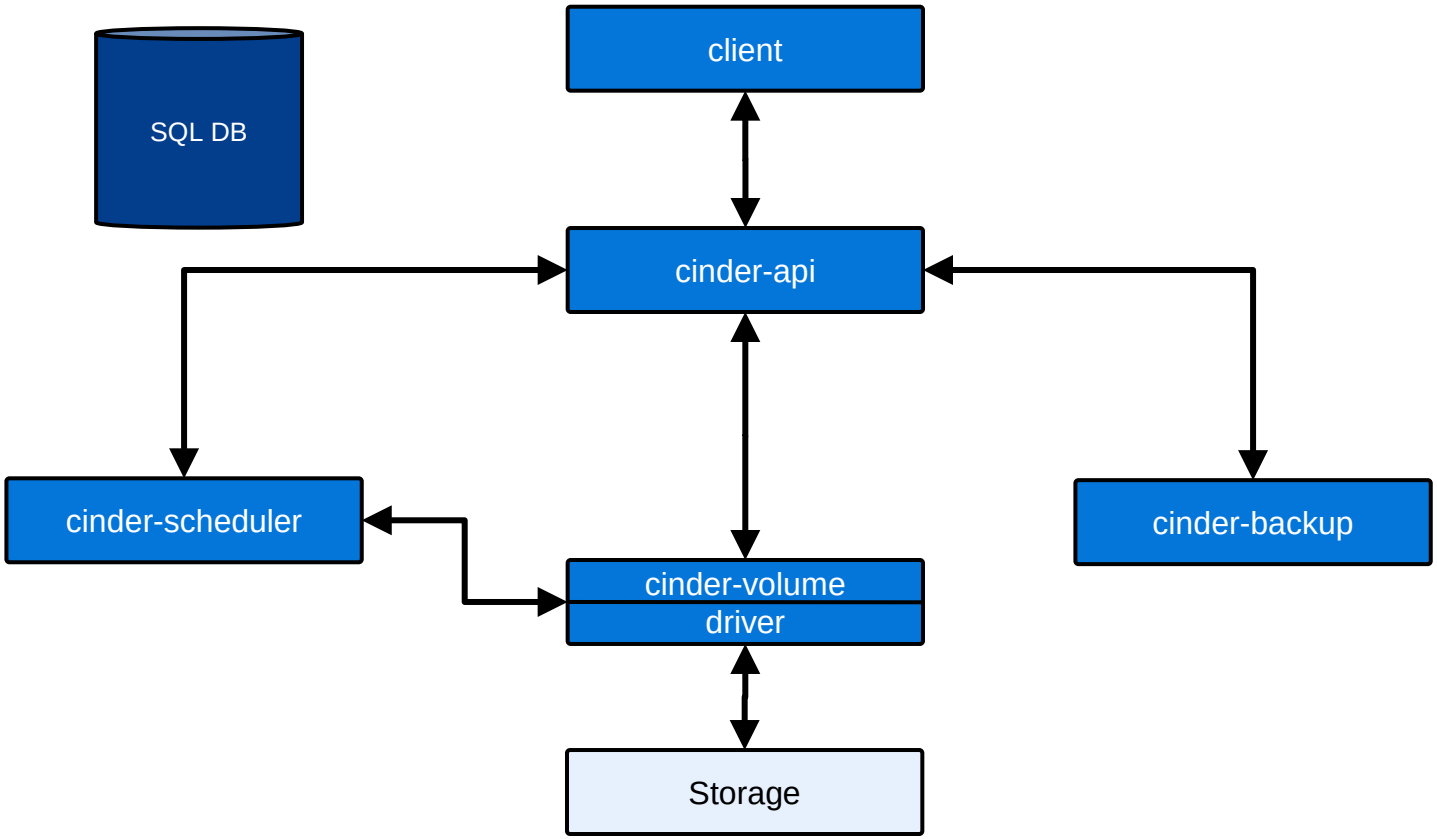
- Cinder provides API's to interact with vendors' storage backends
- Exposes vendor's storage hardware to the cloud
- Provides persistent storage to VMs, containers, bare metal…
- Enables end users to manage their storage without needing to know anything about the type of backing storage device

Where Does Cinder Fit?



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

Cinder Architecture



Cinder Services

- API
  - REST interface to Cinder
  - Generally runs on control node
- Scheduler
  - Takes requests from the API service
  - Works with the volume services to satisfy requests
  - Generally runs on the control node

Cinder Services

- Volume
  - Interacts with storage backends
    - Create/delete/attach/detach/etc…
  - Can run on control node or different host
  - Does NOT run on every Nova compute host
- Backup
  - Interface to backup volumes to storage like object store or NFS share

Service Configuration – cinder.conf

- Used by all of Cinder's services
- Usually located at /etc/cinder/cinder.conf
- Provides settings such as database connection string, message queue settings, service options, etc.
- Sections for defining backend configurations
  - Driver to load
  - Driver specific settings

Service Configuration – cinder.conf

- Configure logging levels, location, format
  - Set debug=True to get verbose details
  - Default logging goes to /var/log/cinder
- Any changes to cinder.conf require service restarts to pick up new settings
- Sample file with defaults and descriptions can be found here: http://docs.openstack.org/developer/cinder/sample_config.html

Service Configuration – cinder.conf

```
[database]
connection = mysql+pymysql://root:cinder@127.0.0.1/cinder?charset=utf8

[oslo_concurrency]
lock_path = /opt/stack/data/cinder

[oslo_messaging_rabbit]
rabbit_userid = stackrabbit
rabbit_password = cinder
rabbit_hosts = 127.0.0.1

[sn448]
volume_backend_name = sn448
volume_driver = cinder.volume.drivers.dell.dell_storagecenter_iscsi.DellStorageCenterISCSIDriver
san_ip = 172.23.57.17
san_login = Admin
san_password = password
dell_sc_ssn = 448
dell_sc_server_folder = STM
dell_sc_volume_folder = STM
```

Cinder Drivers

- To be included in the Cinder source tree, drivers must:
  - Have running CI testing against all patches
  - Implement all care functionality
  - Meet all coding standards of the community
- Acceptable to provide drivers directly to customers – not an absolute requirement to be in tree
- In tree drivers can be found here: http://docs.openstack.org/developer/cinder/drivers.html

Cinder Drivers

- Block Device Driver (local)
- Blockbridge (iSCSI)
- CloudByte (iSCSI)
- Coho (NFS)
- CoprHD (FC, iSCSI, scaleio)
- Datera (iSCSI)
- Dell Equallogic (iSCSI)
- Dell Storage Center (iSCSI/FC)
- Disco (disco)
- DotHill (iSCSI/FC)
- DRBD (DRBD/iSCSI)
- EMC VMAX (iSCSI/FC)
- EMC VNX (iSCSI/FC)
- EMC ScaleIO (scaleio)
- EMC XtremIO (iSCSI/FC)
- FalconStor (iSCSI/FC)
- Fujitsu ETERNUS (iSCSI/FC)
- GlusterFS (GlusterFS)
- HGST (NFS)
- HPE 3PAR (iSCSI/FC)
- HPE LeftHand (iSCSI)
- HPE MSA (iSCSI/FC)
- HPE XP (FC)
- Hitachi HBSD (iSCSI/FC)
- Hitachi HNAS (iSCSI/NFS)
- Huawei (iSCSI/FC)
- IBM Flashsystem (iSCSI/FC)
- IBM GPFS (GPFS/NFS)
- IBM Storage (iSCSI/FC)
- IBM Storwize SVC (iSCSI/FC)
- Infortrend (iSCSI/FC)
- Kaminario (iSCSI)
- Lenovo (iSCSI/FC)
- **LVM (iSCSI) – *Reference****
- NetApp ONTAP (iSCSI/NFS/FC)
- NetApp E Series (iSCSI/FC)
- Nexenta (iSCSI/NFS)
- Nexenta Edge (iSCSI/NBD)
- **NFS – *Reference***
- Nimble Storage (iSCSI)
- Oracle Zfssa (iSCSI/NFS)
- Pure Storage (iSCSI/FC)
- ProphetStor (iSCSI/FC)
- Quobyte (quobyte)
- **RBD (Ceph) - *Reference***
- Scality SOFS (scality)
- Sheepdog (sheepdog)
- SMBFS (SMB)
- SolidFire (iSCSI)
- Synology (iSCSI)
- Tegile (iSCSI/FC)
- Tintri (NFS)
- Violin (FC)
- VMware (VMDK)
- Virtuozzo Storage (NFS)
- Windows (SMB)
- X-IO (iSCSI/FC)
- Zadara (iSCSI)
- ZTE (iSCSI)

Minimum Driver Features

- Drivers must implement support for the core features:
  - Volume create/delete
  - Volume attach/detach
  - Snapshot create/delete
  - Create volume from snapshot
  - Clone volume
  - Extend volume

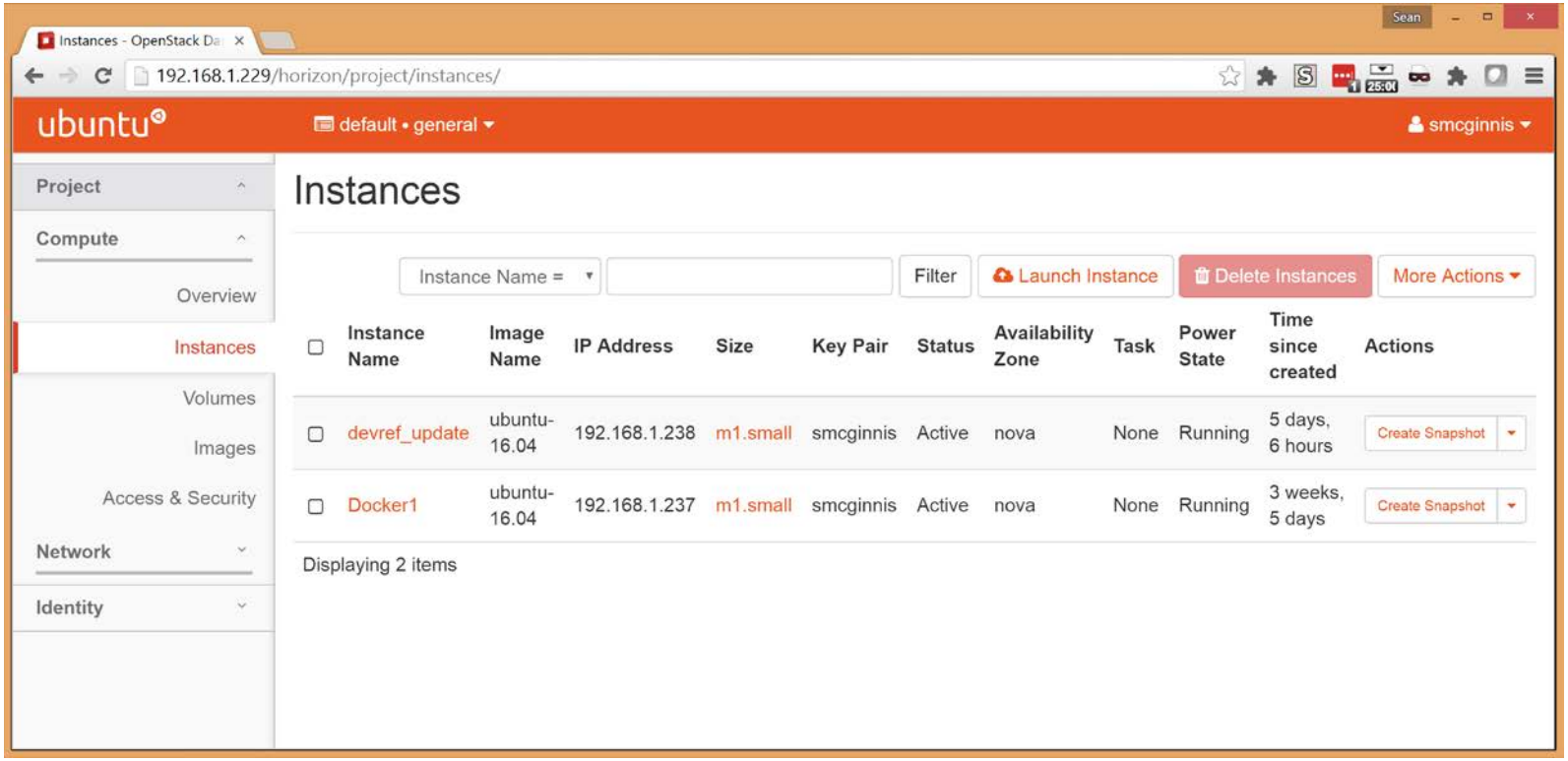
Fibre Channel Support

- Fibre Channel Zone Manager
- Dynamically create and delete switch zoning
- Driver to support fabric management:
  - Brocade
  - Cisco

Clients

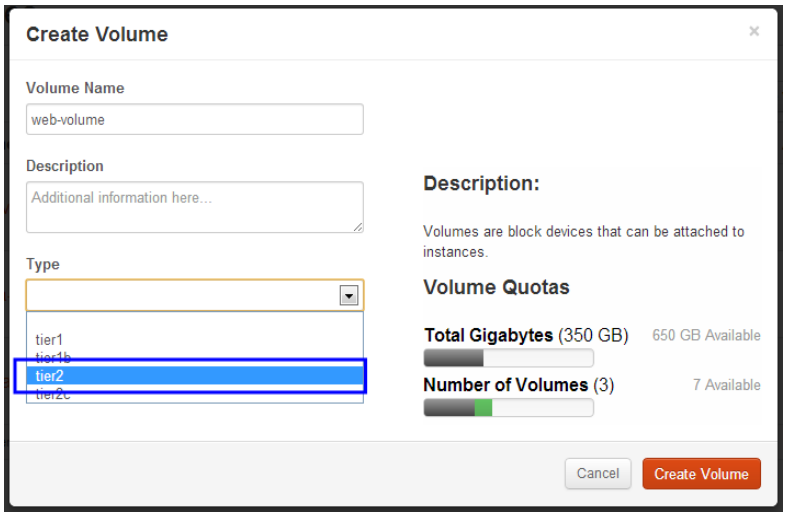
- Cinder Client
  - python-cinderclient is the command line interface to Cinder
    - `cinder create 1 --name Test`
  - Also client library for Python code
  - Uses REST to communicate with the cinder-api service
- OpenStack Client
  - All projects moving to OpenStack Client
    - `openstack volume create --size 1 Test`

Horizon Dashboard



Volume Types

- Used to request properties of volumes during creation
- Can also control users' access to different storage
- Only admins can create volume types
- Users specify the volume type when they create a volume

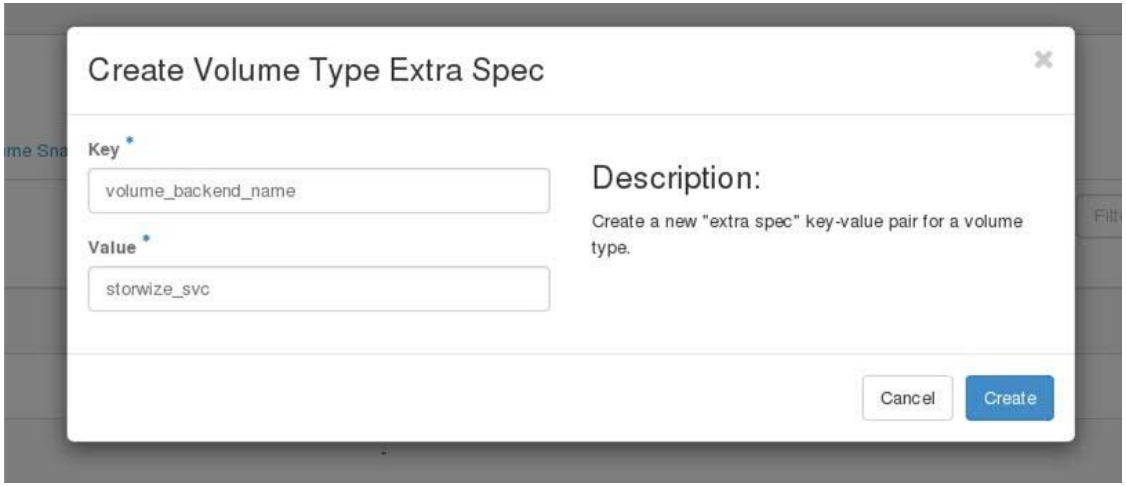


Volume Type Extra Specs

- Extra specs are used to set type properties
- Some standard, some vendor specific
  - `volume_backend_name = lvm1`
  - `sio:provisioning_type = thin`
  - `hp3par:persona = 3`
- Extra specs are only visible to the admin

Volume Type Extra Specs

- Extra specs can be modified via UI, CLI, or API




```
# cinder type-create GoldVolume
# cinder type-key GoldVolume set storagetype:storageprofile=highpriority
# cinder type-create BronzeVolume
# cinder type-key BronzeVolume set storagetype:storageprofile=lowpriorty
```

Retype and Migration

- Retype is used to change the settings of a volume
  - Some retypes can happen without moving data
  - Some require moving the volume to a different backend
- Migration is used to move a volume between two different backends
  - e.g. Move from LVM to Ceph

Retype

Changing volume types:

name:	dellsc1-nightly
extra_specs:	{volume_backend_name: sn12345, storagetype:replayprofile: nightly}

name:	dellsc1-hourly
extra_specs:	{volume_backend_name: sn12345, storagetype:replayprofile: hourly}

```
# cinder create 1 --name vol1 --volume-type dellsc1-nightly
# cinder retype vol1 dellsc1-hourly
```

Retype with Migration

Changing volume types:

name:	dellsc1-nightly
extra_specs:	{volume_backend_name: sn12345, storagetype:replayprofile: nightly}

name:	dellsc1-hourly
extra_specs:	{volume_backend_name: sn54321, storagetype:replayprofile: hourly}

```
# cinder create 1 --name vol1 --volume-type dellsc1-nightly
# cinder retype vol1 dellsc2-hourly   FAILS!
```

Retype with Migration

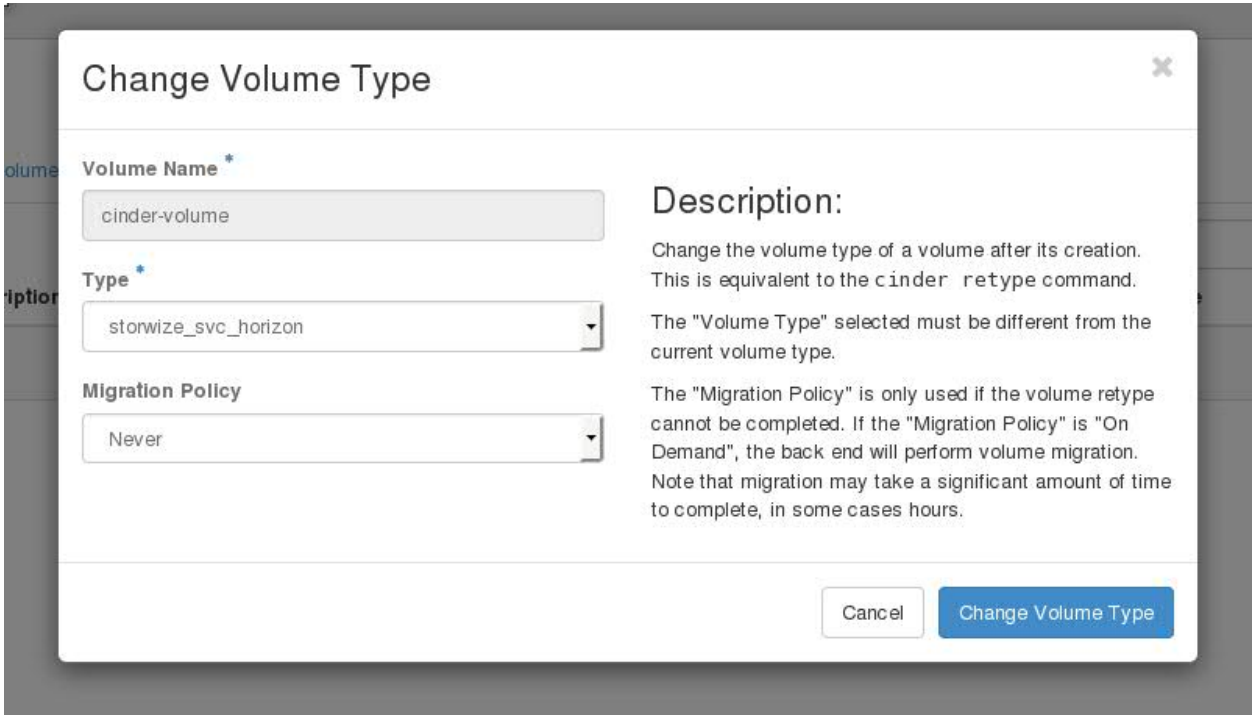
Changing volume types:

name:	dellsc1-nightly
extra_specs:	{volume_backend_name: sn12345, storagetype:replayprofile: nightly}

name:	dellsc1-hourly
extra_specs:	{volume_backend_name: sn54321, storagetype:replayprofile: hourly}

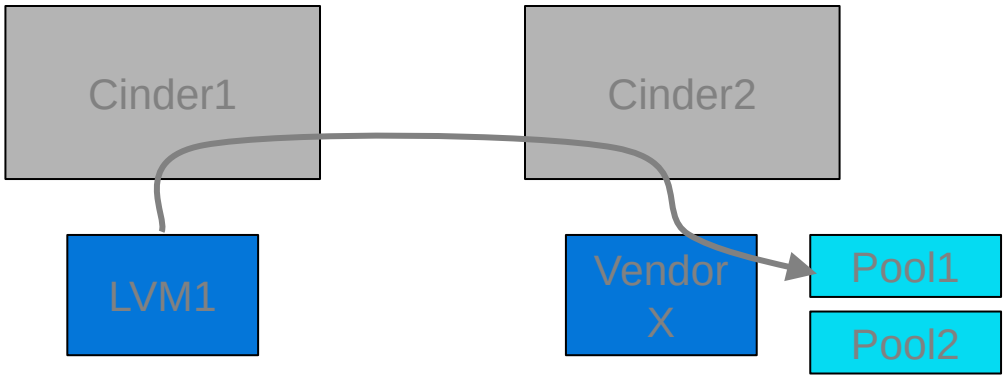
```
# cinder create 1 --name vol1 --volume-type dellsc1-nightly
# cinder retype vol1 dellsc2-hourly –migration-policy on-demand
```

Retype via UI



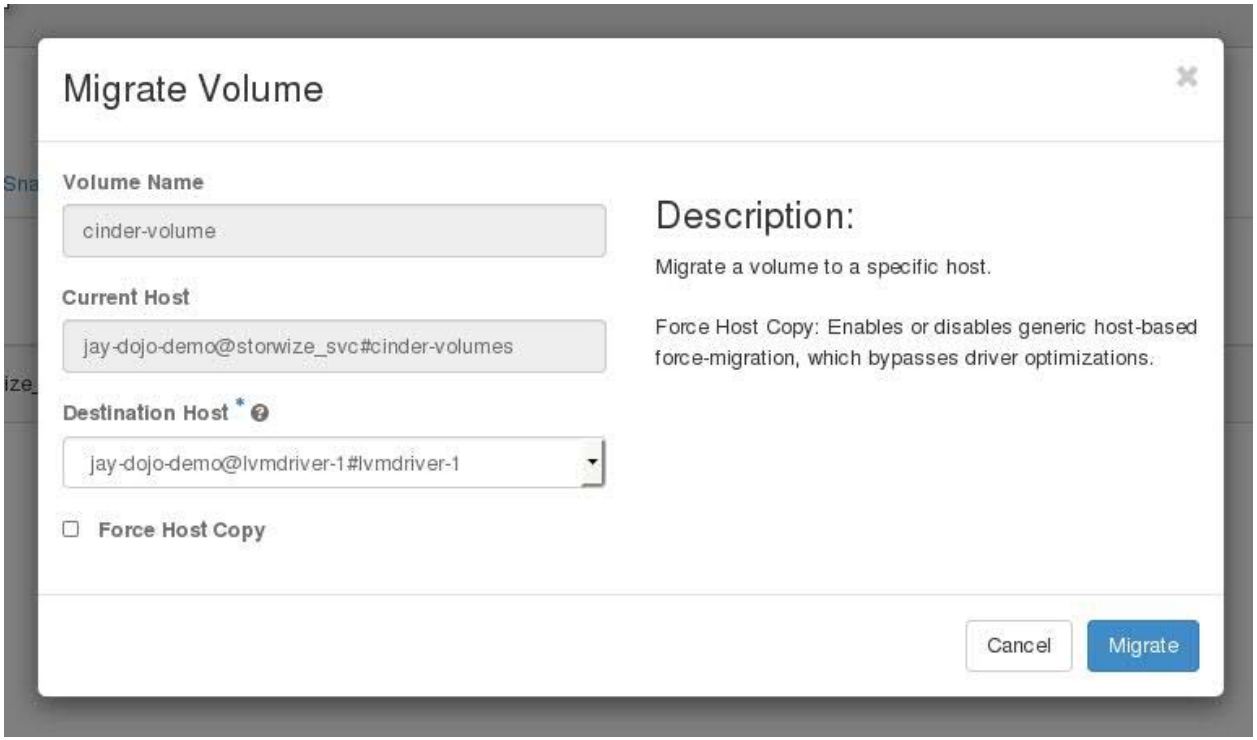
Migration

Migrating volume to new host:




```
# cinder create 1 --name vol1 --volume-type thin_provisioned
# cinder migrate vol1 Cinder2@VendorX#Pool1
```

Migration via UI



Cinder Backup

- Backup and restore volumes
- Must be either in Available state or able to create and mount snapshot
- Several backup drivers supported:
  - Ceph
  - GlusterFS
  - Google Cloud Storage
  - NFS
  - Posix filesystem
  - Swift
  - Tivoli Storage Manager

Cinder Backup

- Backup via CLI, UI, or API
- Needs to be enabled in Horizon
  - /etc/openstack-dashboard/local_settings.py
  - `OPENSTACK_CINDER_FEATURES = {'enable_backup': True}`
- No cron type scheduling in Cinder

```
# cinder backup-create --name MyBackup --description "prepatch" \
        --incremental vol1
# cinder backup-restore a006718b-b583-4d59-9ddb-d1109dc98ebf
```

Cinder Quotas

- Set defaults or per project quotas
- Limit the amount of resources an individual project can consume
- Quota settings:

gigabytes	Total volume and snapshot space consumed
snapshots	Total snapshots allowed
Volumes	Total volumes allowed

Cinder Replication

- Basic support for replication
- Replicate Site A to Site B
- Site A is on fire, failover all volumes to Site B
- New in Mitaka – supported backends and functionality will continue to expanded

Ongoing/Future Work

- Active/active high availability
- Group replication
- Improved error reporting
- Improved Cinder<>Nova interaction