OpenStack Cinder Deep Dive

Michał Dulko

August 4th, 2016

Cinder's Mission

Cinder's Mission

To implement services and libraries to provide on-demand, self-service access to Block Storage resources via abstraction and automation on top of other block storage devices.

Cinder drivers

Cinder is an abstraction layer for around 80 storage backends:

- Open: LVM, GlusterFS, Ceph, NFS. . .
- Proprietary: NetApp, SolidFire, Dell, EMC, HPE, Fujitsu, Hitachi, IBM, Lenovo, VMWare, Violin, Quobyte, Scality, Tegile. . .
- Protocols: iSCSI, NFS, RBD, Fiber Channel, proprietary. . .
- Backup: Swift, RBD, GlusterFS, NFS, IBM TSM

Required features

- Volume Create/Delete
- Volume Attach/Detach
- Snapshot Create/Delete
- Create Volume from Snapshot
- Get Volume Stats
- Copy Image to Volume
- Copy Volume to Image
- Clone Volume
- Extend Volume

Other/optional features

Other/optional features

- Backups
  - CPU bound!
  - Depends on cinder-backup service

Other/optional features

- Backups
  - CPU bound!
  - Depends on cinder-backup service
- Encryption
  - Many restrictions

Other/optional features

- Backups
  - CPU bound!
  - Depends on cinder-backup service
- Encryption
  - Many restrictions
- *Replication*
  - Low number of supporting drivers
  - Replication v1 - single volume replication
  - Replication v2 - backend-level replication

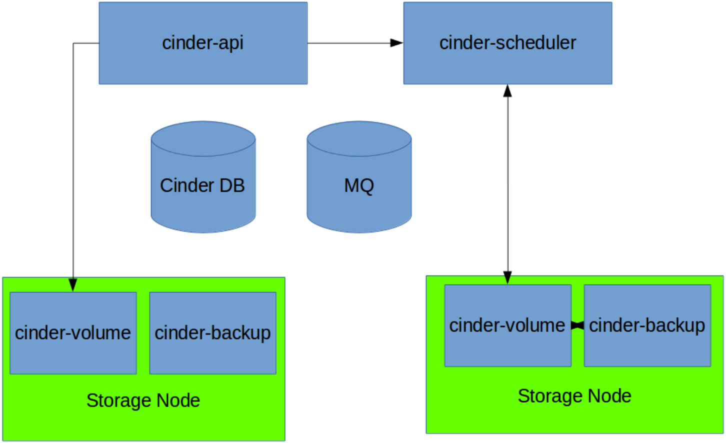
Other/optional features

- Backups
  - CPU bound!
  - Depends on cinder-backup service
- Encryption
  - Many restrictions
- *Replication*
  - Low number of supporting drivers
  - Replication v1 - single volume replication
  - Replication v2 - backend-level replication
- *Consistency groups and snapshots*
  - Low number of supporting drivers
  - Quite reliable

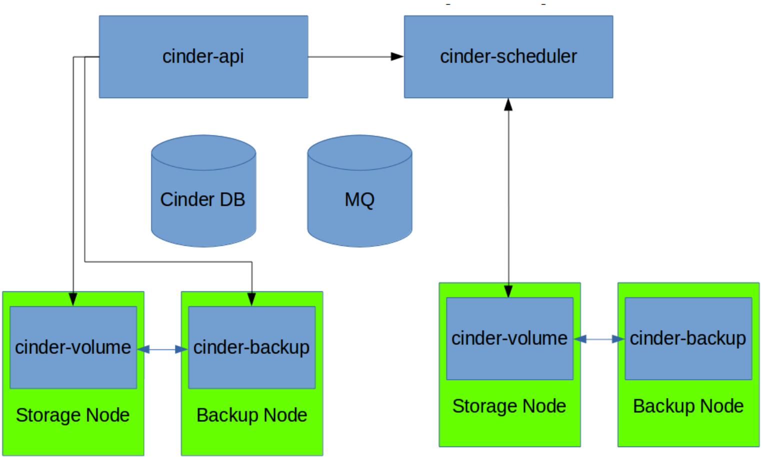
Other/optional features

- Backups
  - CPU bound!
  - Depends on cinder-backup service
- Encryption
  - Many restrictions
- *Replication*
  - Low number of supporting drivers
  - Replication v1 - single volume replication
  - Replication v2 - backend-level replication
- *Consistency groups and snapshots*
  - Low number of supporting drivers
  - Quite reliable
- *QoS support*
  - Moderate number of supporting drivers

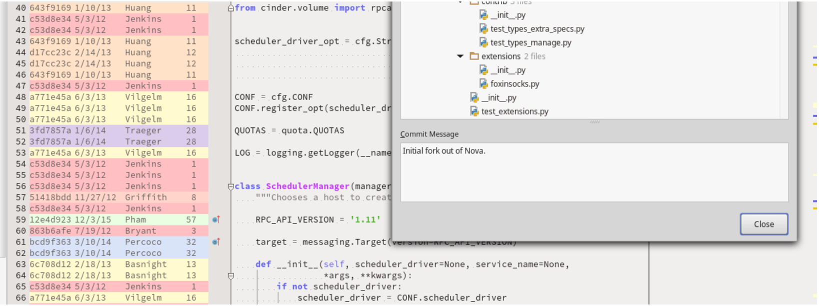
Architecture (pre-Mitaka)



Architecture (since Mitaka)



Architecture

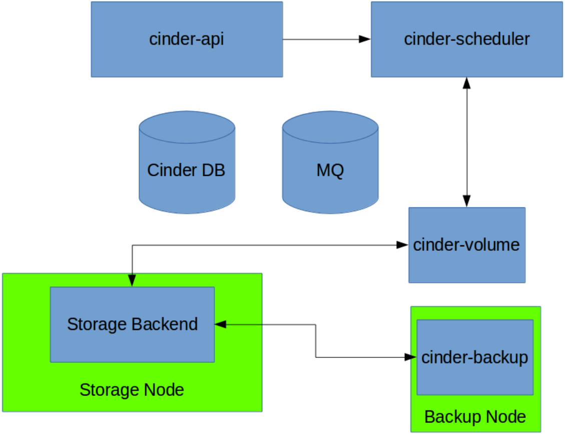


```
643f9169 1/10/13  Huang     11  from cinder.volume import rpca
c53d8e34 5/3/12   Jenkins    1
c53d8e34 5/3/12   Jenkins    1
643f9169 1/10/13  Huang     11  scheduler_driver_opt = cfg.Str
d17cc23c 2/14/13  Huang     12
d17cc23c 2/14/13  Huang     12
643f9169 1/10/13  Huang     11
c53d8e34 5/3/12   Jenkins    1
a771e45a 6/3/13   Vilgelm   16  CONF = cfg.CONF
a771e45a 6/3/13   Vilgelm   16  CONF.register_opt(scheduler_dr
a771e45a 6/3/13   Vilgelm   16
3fd7857a 1/6/14   Traeger   28  QUOTAS = quota.QUOTAS
3fd7857a 1/6/14   Traeger   28
a771e45a 6/3/13   Vilgelm   16  LOG = logging.getLogger(__name
c53d8e34 5/3/12   Jenkins    1
c53d8e34 5/3/12   Jenkins    1
c53d8e34 5/3/12   Jenkins    1  class SchedulerManager(manager
51418bdd 11/27/12 Griffith   8      """Chooses a host to creat
c53d8e34 5/3/12   Jenkins    1
12e4d923 12/3/15  Pham      57      RPC_API_VERSION = '1.11'
863b6afe 7/19/12  Bryant     3
bcd9f363 3/10/14  Percoco   32      target = messaging.Target(version=RPC_API_VERSION)
bcd9f363 3/10/14  Percoco   32
6c708d12 2/18/13  Basnight  13      def __init__(self, scheduler_driver=None, service_name=None,
6c708d12 2/18/13  Basnight  13                   *args, **kwargs):
c53d8e34 5/3/12   Jenkins    1          if not scheduler_driver:
a771e45a 6/3/13   Vilgelm   16              scheduler_driver = CONF.scheduler_driver
```

Commit Message

Initial fork out of Nova.

Architecture (non-LVM-backends)

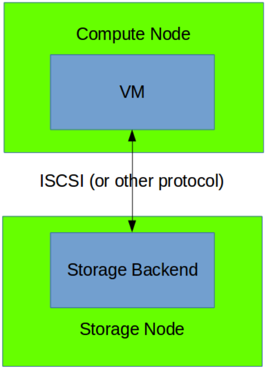


Attach to VM or detach from VM

Complicated chain of internal REST API calls from Nova to Cinder.

Attach to VM or detach from VM

Complicated chain of internal REST API calls from Nova to Cinder.



Pitfalls

Pitfalls

- cinder-scheduler is race-condition prone
    - Nova’s legacy

Pitfalls

- cinder-scheduler is race-condition prone
  - Nova's legacy
- multi-backend support
  - It's like running multiple cinder-volume on one node
  - Deployment without enabled_backends option is deprecated in Newton

Pitfalls

- cinder-scheduler is race-condition prone
  - Nova's legacy
- multi-backend support
  - It's like running multiple cinder-volume on one node
  - Deployment without enabled_backends option is deprecated in Newton
- Cinder usage outside of OpenStack
  - python-brick-cinderclient-ext project
  - You'll still need DB (MySQL), MQ (RabbitMQ) and Keystone

Future

- Replication v2.1
    - replication of groups of volumes

Future

- Replication v2.1
  - replication of groups of volumes
- Ironic support

Future

- Replication v2.1
  - replication of groups of volumes
- Ironic support
- Volume multi-attach support
  - Cinder's side is done. . .

Future

- Replication v2.1
  - replication of groups of volumes
- Ironic support
- Volume multi-attach support
  - Cinder's side is done... since Liberty
  - Still trying to figure out correct Nova-Cinder interactions

Future

- Replication v2.1
  - replication of groups of volumes
- Ironic support
- Volume multi-attach support
  - Cinder's side is done. . . since Liberty
  - Still trying to figure out correct Nova-Cinder interactions
- Live upgrade support
  - Experimental in Mitaka
  - Hopefully Newton will officially support that

Future

- Replication v2.1
  - replication of groups of volumes
- Ironic support
- Volume multi-attach support
  - Cinder's side is done... since Liberty
  - Still trying to figure out correct Nova-Cinder interactions
- Live upgrade support
  - Experimental in Mitaka
  - Hopefully Newton will officially support that
- cinder-volume service clustering AKA *c-vol A/A HA support*
  - Right now it is still risky to run multiple c-vols controlling a single storage backend

Thank you!