

OpenStack Cinder Deep Dive

Michał Dulko

August 4th, 2016

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

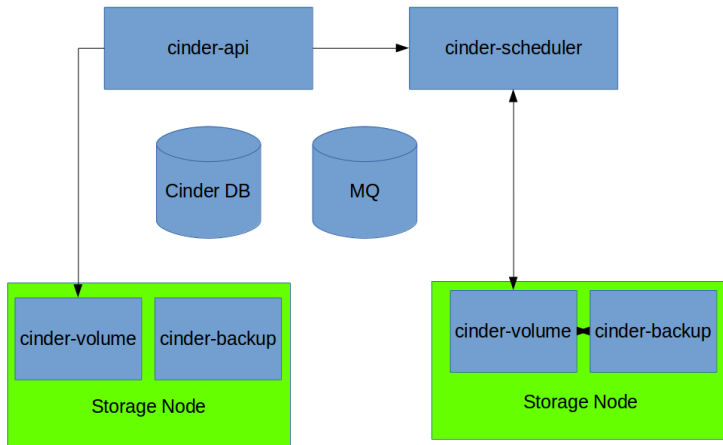
Other/optional features

- ▶ 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

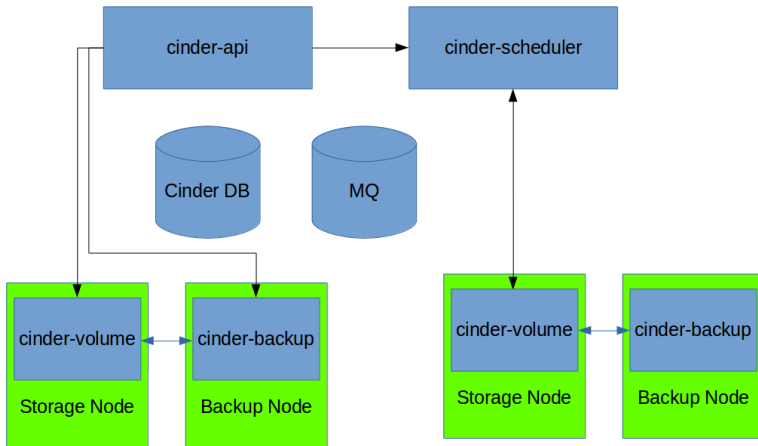
Other/optional features

- ▶ 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

40	643f9169	1/10/13	Huang	11
41	c53d8e34	5/3/12	Jenkins	1
42	c53d8e34	5/3/12	Jenkins	1
43	643f9169	1/10/13	Huang	11
44	d17cc23c	2/14/13	Huang	12
45	d17cc23c	2/14/13	Huang	12
46	643f9169	1/10/13	Huang	11
47	c53d8e34	5/3/12	Jenkins	1
48	a771e45a	6/3/13	Vilgelm	16
49	a771e45a	6/3/13	Vilgelm	16
50	a771e45a	6/3/13	Vilgelm	16
51	3fd7857a	1/6/14	Traeger	28
52	3fd7857a	1/6/14	Traeger	28
53	a771e45a	6/3/13	Vilgelm	16
54	c53d8e34	5/3/12	Jenkins	1
55	c53d8e34	5/3/12	Jenkins	1
56	c53d8e34	5/3/12	Jenkins	1
57	51418bdd	11/27/12	Griffith	8
58	c53d8e34	5/3/12	Jenkins	1
59	12e4d923	12/3/15	Pham	57
60	863b6afe	7/19/12	Bryant	3
61	bcd9f363	3/10/14	Percoco	32
62	bcd9f363	3/10/14	Percoco	32
63	6c708d12	2/18/13	Basnight	13
64	6c708d12	2/18/13	Basnight	13
65	c53d8e34	5/3/12	Jenkins	1
66	a771e45a	6/3/13	Vilgelm	16

```
from cinder.volume import rpcapi

scheduler_driver_opt = cfg.StrOpt(
    'scheduler_driver_opt',
    default=None,
    help='The driver to use for scheduling volumes.'
)

CONF.register_opt(scheduler_driver_opt)

QUOTAS = quota.QUOTAS

LOG = logging.getLogger(__name__)

class SchedulerManager(manager.Base):
    """Chooses a host to create volumes on"""

    RPC_API_VERSION = '1.11'

    target = messaging.Target(version=RPC_API_VERSION)

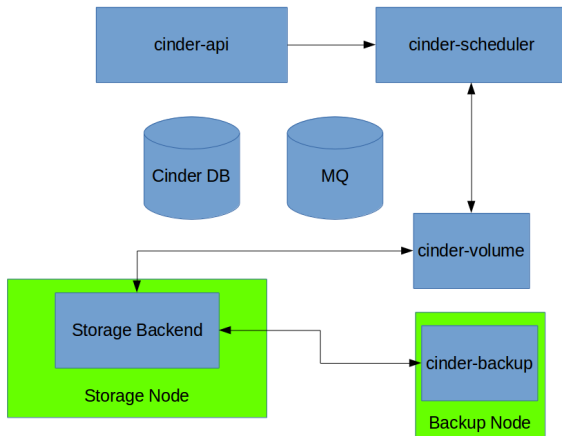
    def __init__(self, scheduler_driver=None, service_name=None,
                 *args, **kwargs):
        if not scheduler_driver:
            scheduler_driver = CONF.scheduler_driver
```

Commit Message

Initial fork out of Nova.

Close

Architecture (non-LVM-backends)

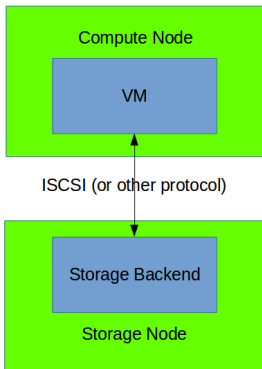


Attach to VM or detach from VM

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
- ▶ Ironix support

Future

- ▶ Replication v2.1
 - ▶ replication of groups of volumes
- ▶ Ironi 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!