How to Survive an OpenStack Cloud Meltdown with Ceph

Federico Lucifredi, Sean Cohen, Sébastien Han

OpenStack Summit Vancouver
May 22, 2018

What's the Problem?

DatacenterDynamics
https://bit.ly/2IEC3t4

TWS
https://bit.ly/2IAQXVv

Additional coverage

DataCenter Knowledge
https://bit.ly/2x32Iyn

@hohocho
https://bit.ly/2IVor0b

S3 Service Disruption
https://amzn.to/2meIOup

Redefining Disaster Recovery

- The problem space
  - Disaster recovery (DR)
  - High availability (HA)
  - Fault tolerance (FT)
  - And everyone's favorite… Backup.
- Key metrics
  - Recovery point objective (RPO)
  - Recovery time objective (RTO)

Meme via SJVN
https://lnkd.in/d8QXRZJ

Cloud Meltdown Prevention

State of the art

Different failure domains

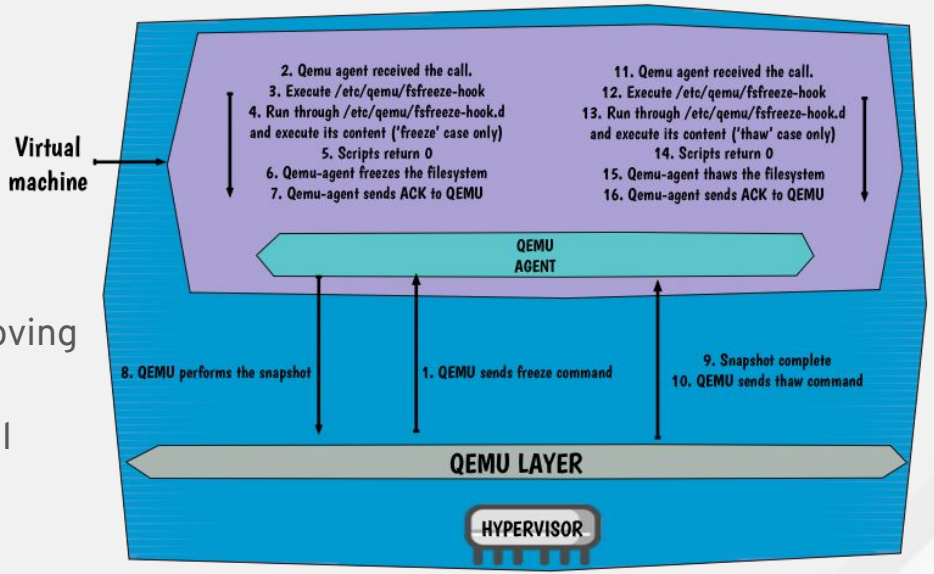
Ways to segregate your data:

- Ceph and failure domains:
  - Always build redundancy from racks (1 replica per rack)
  - Go as granular as you want (room, pod, pdu, row, rack...)
  - Between pools
- Different Ceph clusters in the same datacenter, but different facility
- Different Ceph clusters geographically spread

Nova and Cinder snapshots

Operational Recovery (Human error, Database corruptions, Malicious code etc…)

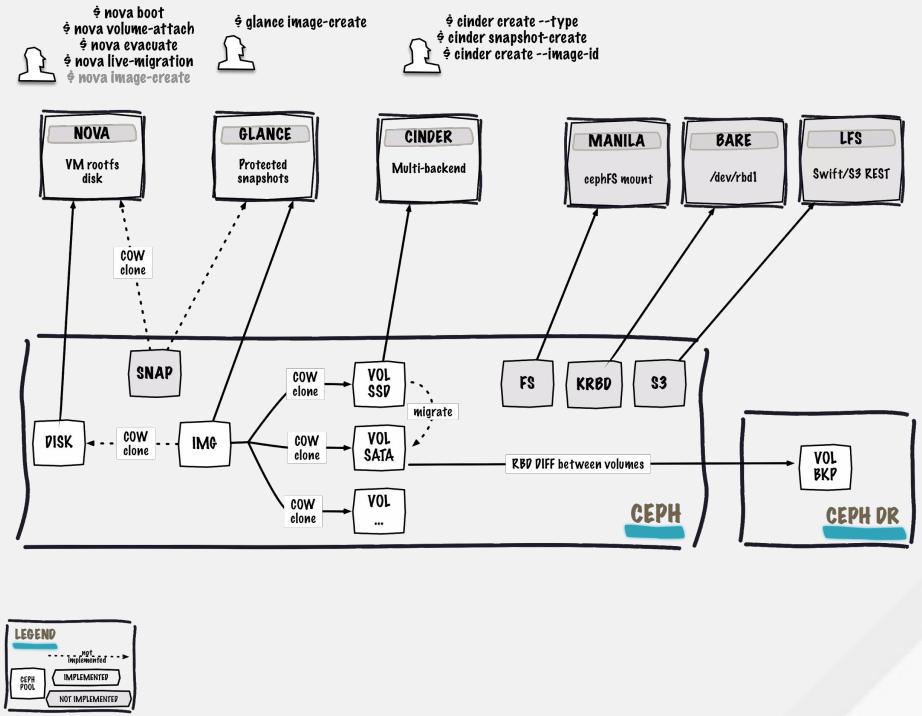
- **NOT a backup**, just for convenience
- Difference:
  - Nova: new image in Glance
  - Cinder: internal snapshot with layering
- Snapshot are fast and transparent, no data moving the network
- Always use a qemu-guest-agent in your virtual machine



Cinder Ceph backup

- Efficient Ceph to Ceph backups
- Always incremental - block differential
- Works between:
  - Ceph Pools (not recommended)
  - Ceph clusters

- Deeper Workload backup with change block tracking based style on RBD Diff, available via 3rd party vendors

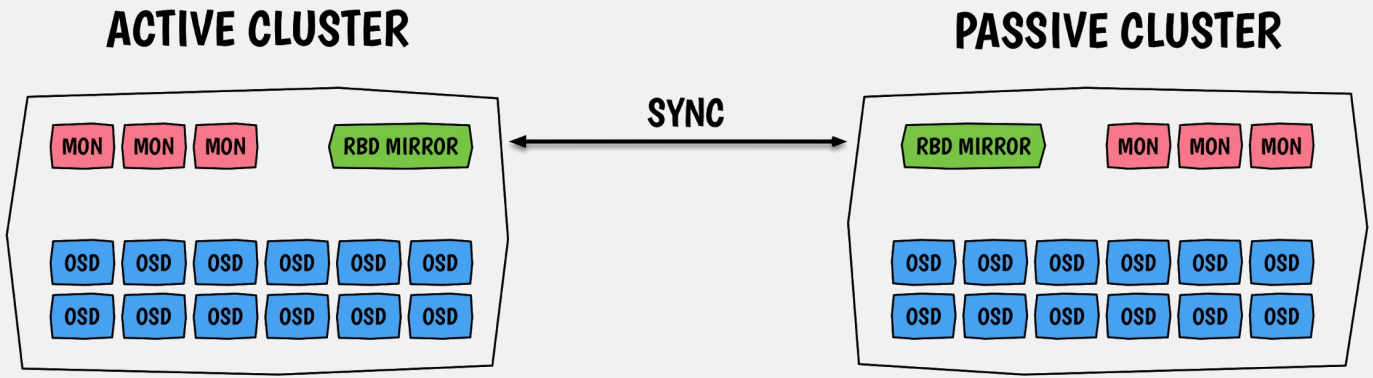


Disaster Recovery with RBD Mirroring

RBD Mirroring - Ceph Jewel

Ceph Jewel release (Apr 2016):

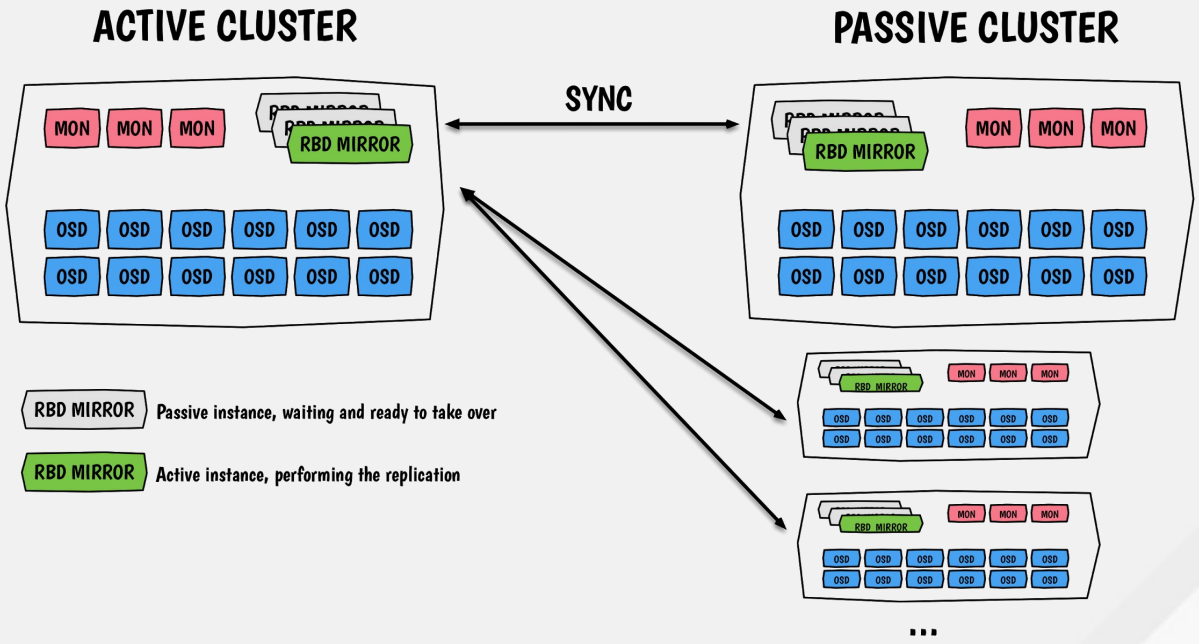
- Added ‘rbd-mirror’ daemon to synchronise Ceph images from one cluster to another
- Relies on two new RBD image features:
  - journaling: enables journaling for every transaction on the image
  - mirroring: tells the rbd-mirror daemon to replicate images
- Images have states: primary and non-primary (promote and demote calls)
- 1:1 relationship between daemons but cross-replication possible



RBD Mirroring - Ceph Luminous

Ceph Luminous release
(Aug 2017, current):

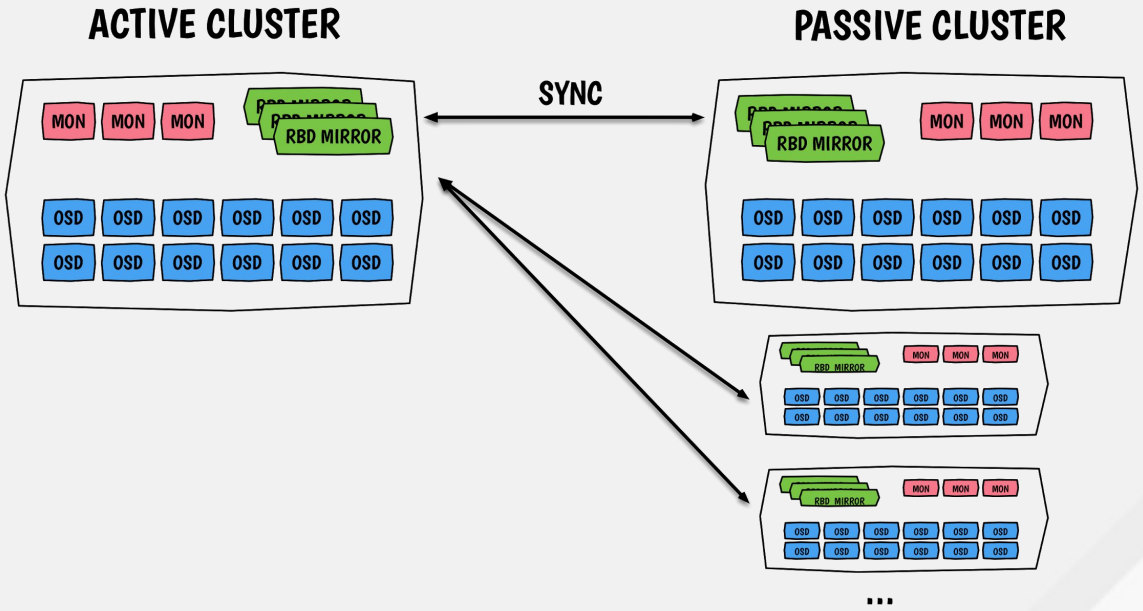
- RBD-Mirror, multiple instances with failover
  - **Only** a single instance is active
- Multiple secondary sites (unidirectional)
- Delayed replication (for unwanted change)



RBD Mirroring - Ceph Mimic

Ceph Mimic release (rc1 May 2018):

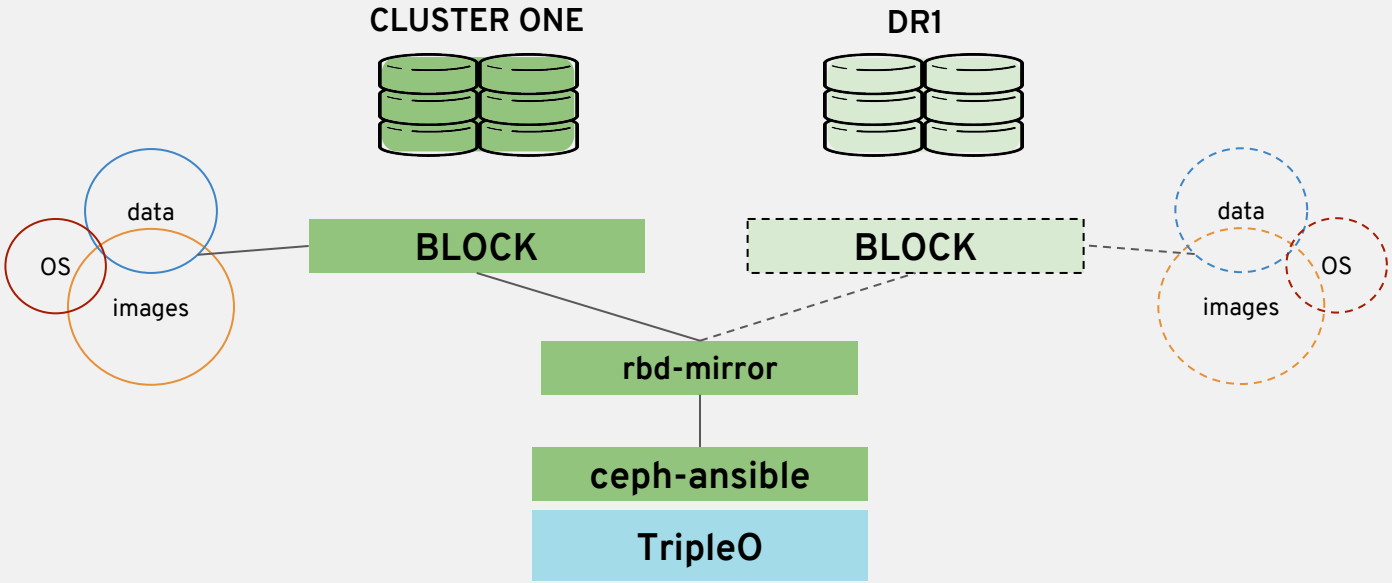
- RBD-Mirror, multiple active instances
  - ‘Simple’ algorithm for chunking
  - Naive chunking policy, divides m daemons and n images in a way
- Dashboard V2 mirroring integration
- Delayed deletion
- Clone non-primary images (for Glance)



Disaster Recovery with RBD Mirroring

EASIER WAYS SAFEGUARD DATA IN OPENSTACK DEPLOYMENTS

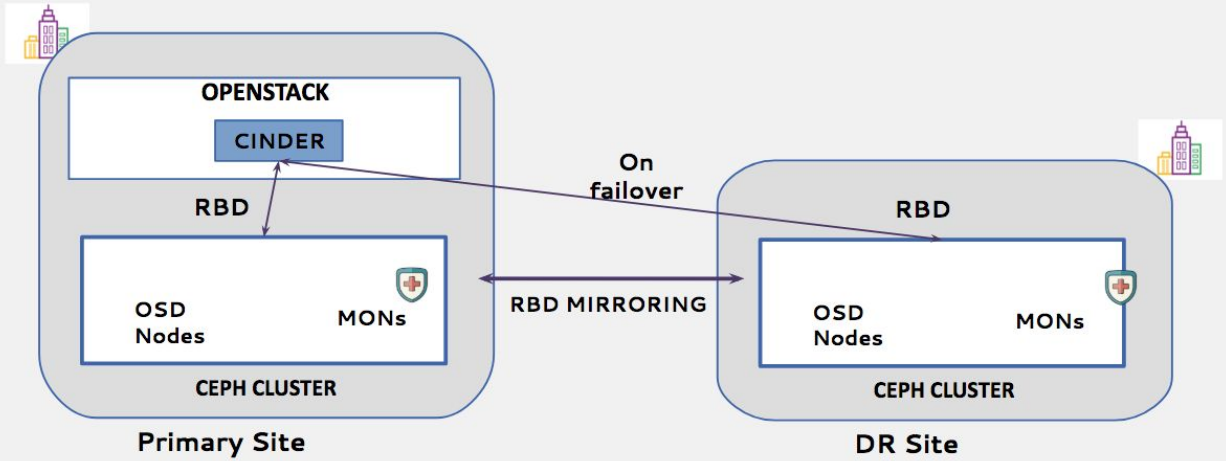
RBD MIRRORING DEPLOYED WITH TRIPLEO



Cinder Replication w/RBD Mirroring

Cinder RBD Replication

- New in Queens release: Tripleo deployment of RBD-mirror daemon using ceph-ansible
- Replication is enabled per-volume
- Cinder assumes Ceph cluster mirroring is configured and functioning
- Underlying configuration can mirror to multiple sites, cinder can be configured to reflect this configuration and allow multiple failover targets
- Failover changes cinder runtime configuration and promotes each volume to the secondary.
- Upcoming feature to promote a secondary to primary
- A/A replication in RBD merged this cycle



Cinder Replication Failover / Promote

What's available today?

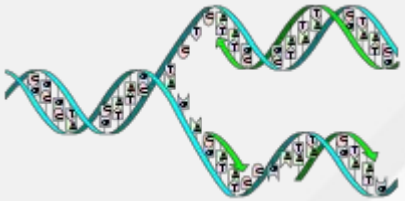
- Ocata release - RBD Cinder Volume Replication driver
    - Volume based added rather than just pool level (based on Cinder Volume Types)
    - Both clusters are expected to be configured with rbd-mirror with keys in place and image mirroring enabled on the pool.
    - The RBD driver will enable replication per-volume if the volume type requests it.
    - Supports Multiple mirroring clusters

- After failing backend A over to backend B, there is was no mechanism in Cinder to promote backend B to the master backend in order to then replicate to a backend C. We also lacked some management commands to help rebuild after a disaster if states get out of sync.

Cinder Replication Promotion Failback

Rocky Release focus

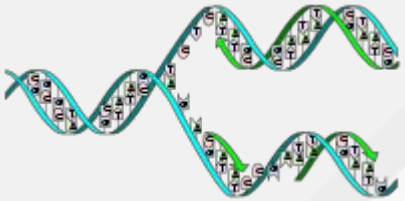
- Add support for promoting a failed over backend
  - Cinder's failover support allows us to test a disaster recovery configuration by initiating a failover operation and verifying that the driver correctly passes control of the replicated volumes to a secondary site.
  - With the recent merge of the RBD replication in the Cinder driver, this failover is now supported for ceph-backed cinder deployments.
  - Cinder also provides a failback operation to allow an administrator to undo a previous failover.



Cinder Replication Promotion Failback

Rocky Release focus

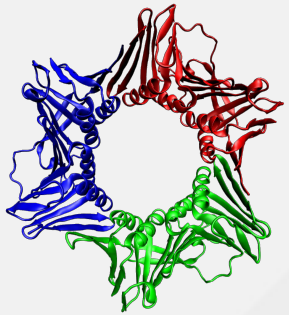
- Add support for Active/Active replication in Cinder RBD driver
  - This allows you to configure multiple volume backends that are all a member of the same cluster participating in replication.
  - The new feature breaks failover_host into the two requisite parts (Failover and Failover_completed) in order to support both single-backend and multiple backends configured as active/active.



Cinder RBD Replication

Gaps

- Nova attached volumes are no longer connected
- Cannot replicate a multi-attach volume or multi-attach a replicated volume
- No “gate testing” yet if the promotion works



Use your RBD Backups for DR

Cinder RBD Backup with RBD Mirroring

Ceph Cinder Backup recovery procedure

When something goes wrong:

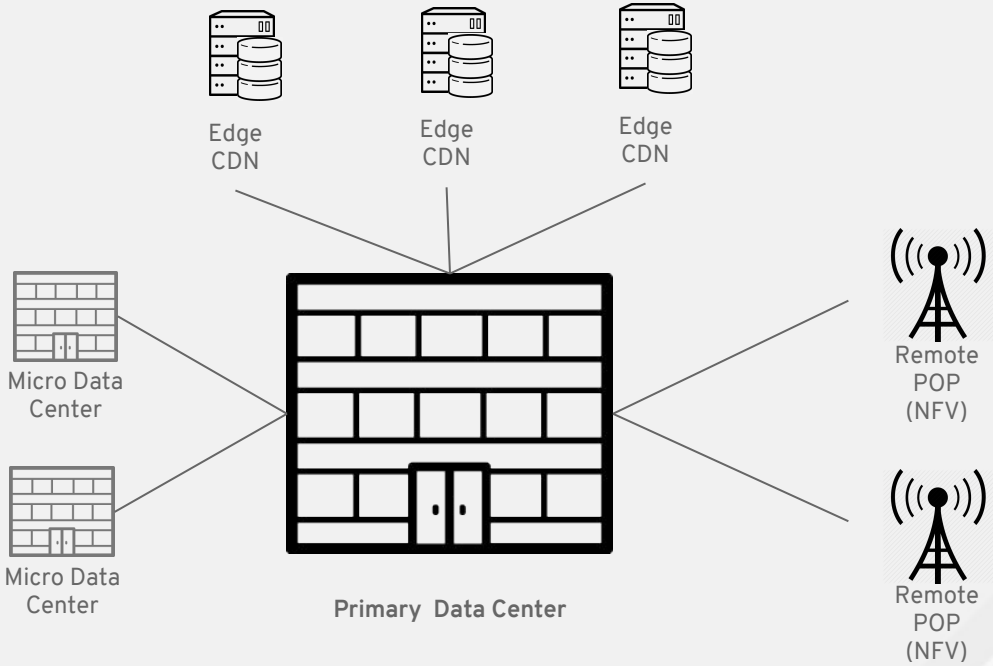
- Re-configure Cinder Backup to point to the mirrored cluster
- Force detach the volume
- Restore the backup to a new volume
- Attach the volume again to the virtual machine
  - If boot from volume:
    - Reset-state (should be in failed state)
    - Start the virtual machine again
- Best Practice - use Export and import backup metadata
  - Exporting and storing this encoded string metadata allows you to completely restore the backup, even in the event of a catastrophic database failure.
    - $ cinder backup-export BACKUP_ID
    - $ cinder backup-import METADATA

Protecting the Edge

Protecting the Edge

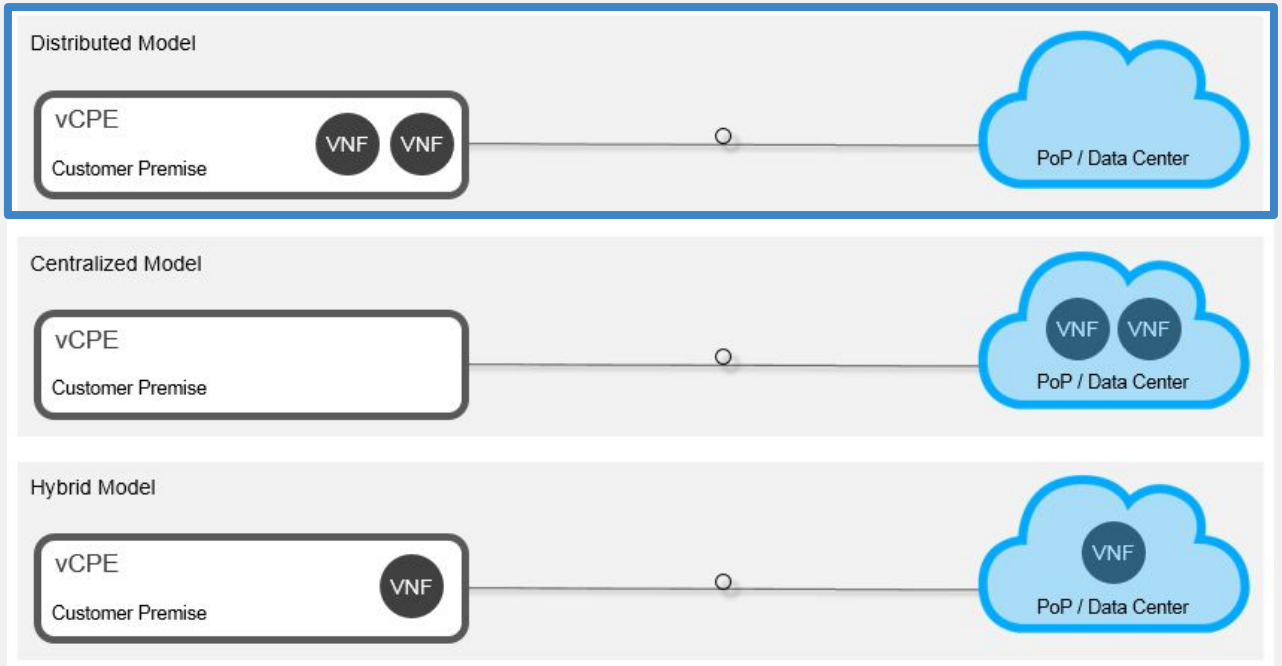
Compute and Storage supporting key Apps at the edge

- Edge sites hosting content for low-latency delivery
- Remote POPs running virtual network functions
- Micro data centers capturing IOT telemetry fo real-time processing



Distributed NFVi

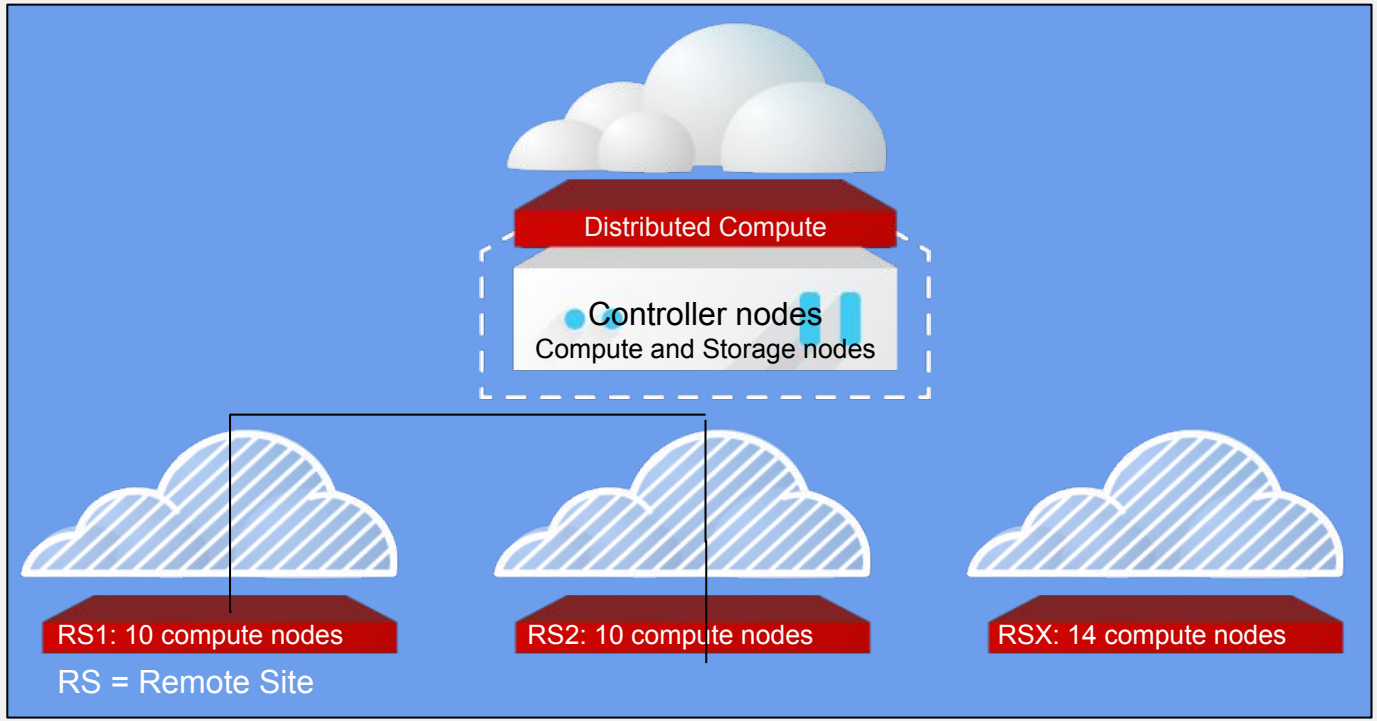
Protecting Virtual Network Functions at the Edge



Source: Brian Lavallée @ Ciena

Distributed Compute Nodes architecture

Protecting Virtual Network Functions at the Edge



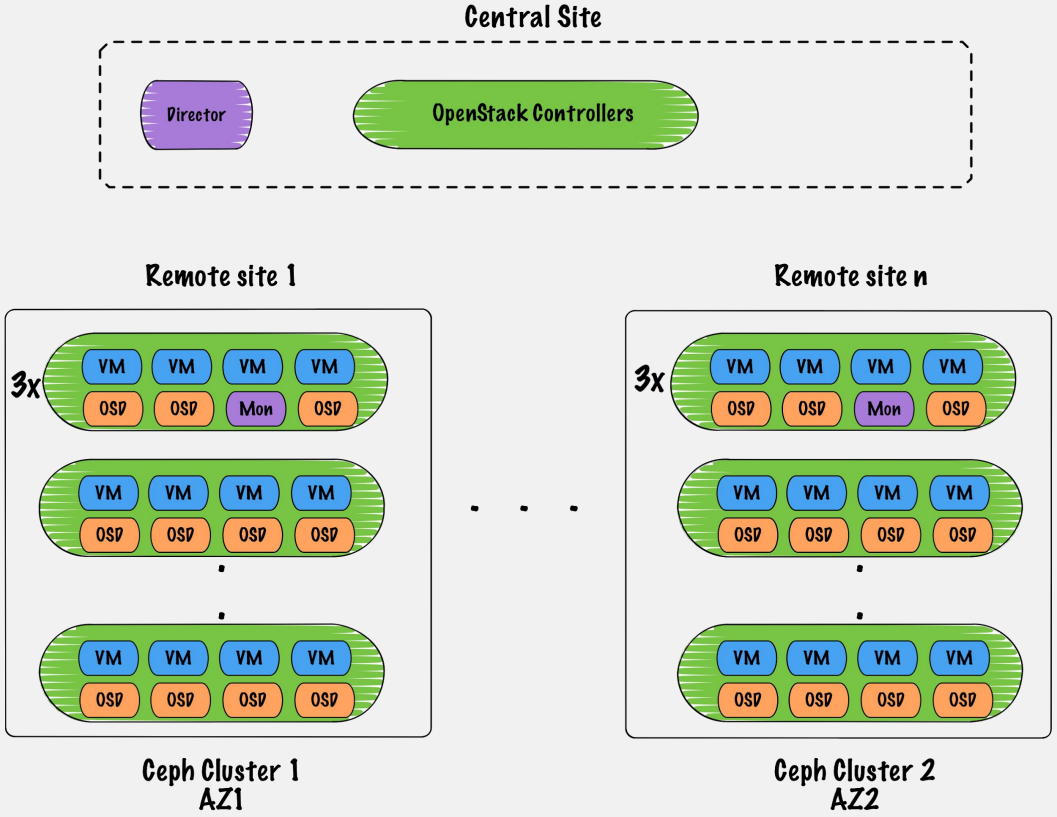
Distributed Compute Nodes with Ceph

Protecting Virtual Network Functions at the Edge

- To ensure performance, Ceph storage nodes need to communicate with Computes on a low latency and high bandwidth network.
- What means Ceph storage nodes are collocated with the computes on the same Local Area Network (LAN) in an Hyper-Converged manner.

- In Distributed NFV, Ceph storage is used as a back-end for:
  - Nova: stateless VNF's requires ephemeral storage on Ceph
  - Cinder: statefull VNF's requires persistent storage use Cinder volumes
  - Glance: use Ceph in the Compute zone as a backend to spawn VNFs quickly (as you do not want to rely on image transfer delay between 2 remote sites)
  - Leverage Ceph "copy on write" to speed-up the spawing process.

Distributed Compute Nodes with Ceph



Distributed Compute Nodes with Ceph

Protecting Images the Edge

- Distributed NFV use case also requires to support multi-backend (multiple Ceph clusters) for Glance , where the image is uploaded once and replicated on all the POPs using Ceph.
- With Ceph storage nodes in each compute zone, without having to use additional physical servers for the control plane, one solution being to support the collocation of Ceph MON and Ceph OSD on the same physical servers .

- Benefits
  - Reduces hardware footprint since we have a centralized control plane
  - Brings the ability to manager multiple Ceph cluster from the centralized control plane
  - Brings resilience with multiple AZs

Distributed Compute Nodes with Ceph

Protecting Images the Edge

- Using a third party proxy
  - Project MixMatch
    - Combines resources across federated OpenStack deployments
    - The Proxy Service will forward REST API requests to a remote service provider which is federated using Keystone-to-Keystone Federation (K2K).
    - The proxy learns the location of resources and is able to forward requests to the correct service provider.
    - This allows OpenStack services to use resources provided by other federated OpenStack deployments, ex. Nova attach a remote volume.

Distributed Compute Nodes with Ceph

Protecting Images the Edge

Introducing Glance Multi-backend - Rocky release work in progress

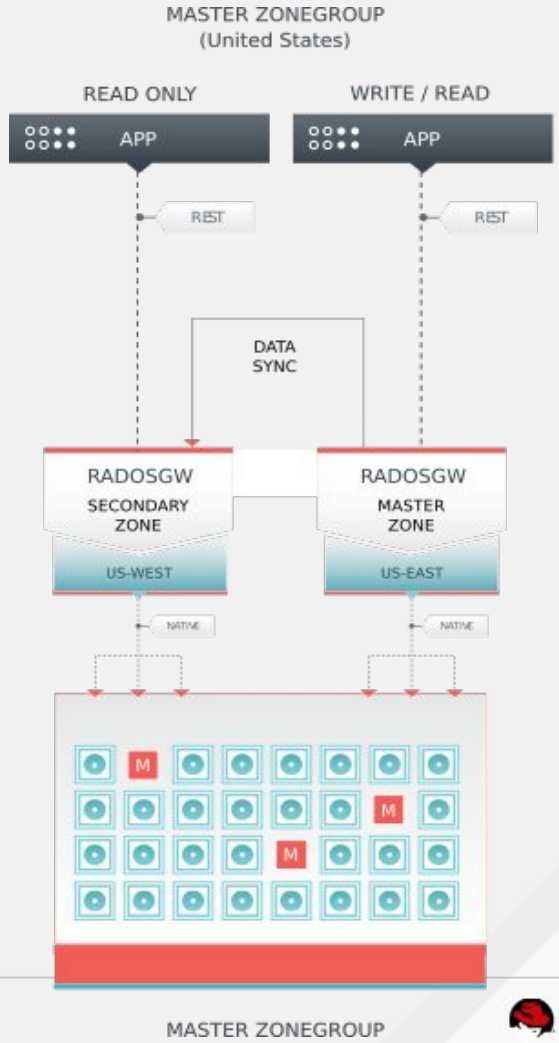
- Ability to store Glance images in different storage backends
- Multi-store support opens the door to a handful of image replication and backend targeting use cases which can provide additional value related to fault tolerance, image locality and policy based image hosting.
- Allowing Glance Metadata to manage the location awareness in a the centralized control plane, where store drivers need to express the different storage backends.
- Support multiple Ceph clusters (one per backend) where one backend will represent a distributed site
- Allow to specify specific Glance backend at the edge - for example CDN specific image.
- Glance does not need to deal with how to sync images between clusters, rather than add the store logic to support it.
- Additional work needed in Stein release for Store drivers to interact with Glance.

Object Replication

Ceph Rados Gateway (RGW)

RGW Multisite replication (Global Clusters)

- Global object storage clusters with a single namespace
- Enables deployment of clusters across multiple locations
- Locally immediately consistent, remote eventually consistent
- Clusters synchronize
  - allowing users to read from or write to the closest one
- Active-active replication
  - Metadata operations reserved for the master



redhat

Where we going?

Roadmap: Ceph Rados Gateway sync to S3

Mimic
May 2018

- "Cloudsync"
- No policy support

Nautilus
H1 2019

- "archive"
- Time-based
- Manual retrieval

Future

- Tiering
- Heat-based data migration
- Migrate back on access

Ability to sync Ceph Rados Gateway private object store to public S3 Cloud providers (e.g: Amazon S3).

- Tiering to cloud
- Tiering to tape

Roadmap: RBD Mirror

Luminous
August 2017

- active/passive HA
- Triple-O support
- 1:many replication

Mimic
May 2018

- active/active HA
- scale-out
- Naive balancing
- Dashboard integration

Future

- Adaptive balancing

Ability to asynchronously replicate volume images across multiple Ceph clusters

- Streaming changes live to a secondary site (DR)
- Volumes are ready to consume and can be immediately pressed into production

Plan Before the Meltdown

Putting it all together

- **Cloud Meltdown A3** (can happen anywhere to anyone, anytime)
- The good news, **it is preventable** (different failure domains from snaps and backups combined with block and object replication)
- The secret: **design you Apps with HA. Architect OpenStack and Ceph properly.**
- **Distribute your data** using the latest Cinder Replication with Ceph block and object replication
- **Extend your RBD Backups for DR** using Ceph replication
- **Use Ceph to to protect your Edge** footprint by leveraging Ceph/Glance multi-backend support
- **Automate and test it!**
- Next Cloud Meltdown your cloud users may hardly even notice…

THANK YOU

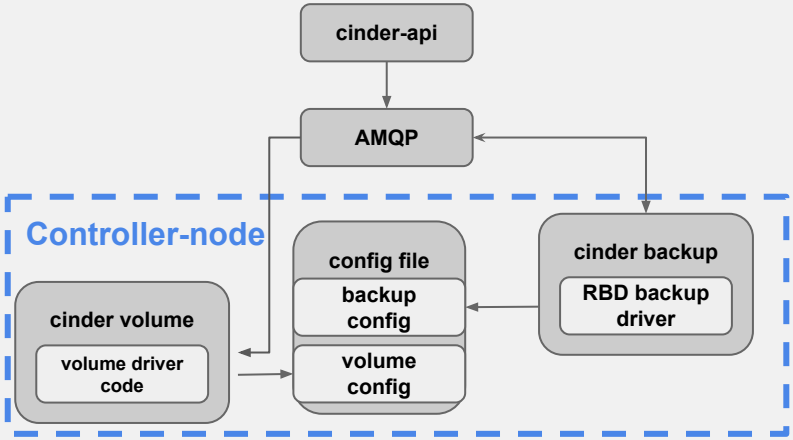
- 0xF2
- leseb_
- SeanCohen_RH

Backup slides (bwahahaha)

Backups - If your cluster is gone

Procedure to resume activity:

- Force detach volumes from instances
- Reconnect all OpenStack services to consume the backup cluster (Glance, Nova, Cinder)
- Restart them
- Re-attach volumes to your instances



Resume activity, rebuild the initial cluster and sync back everything with rbd-mirroring.

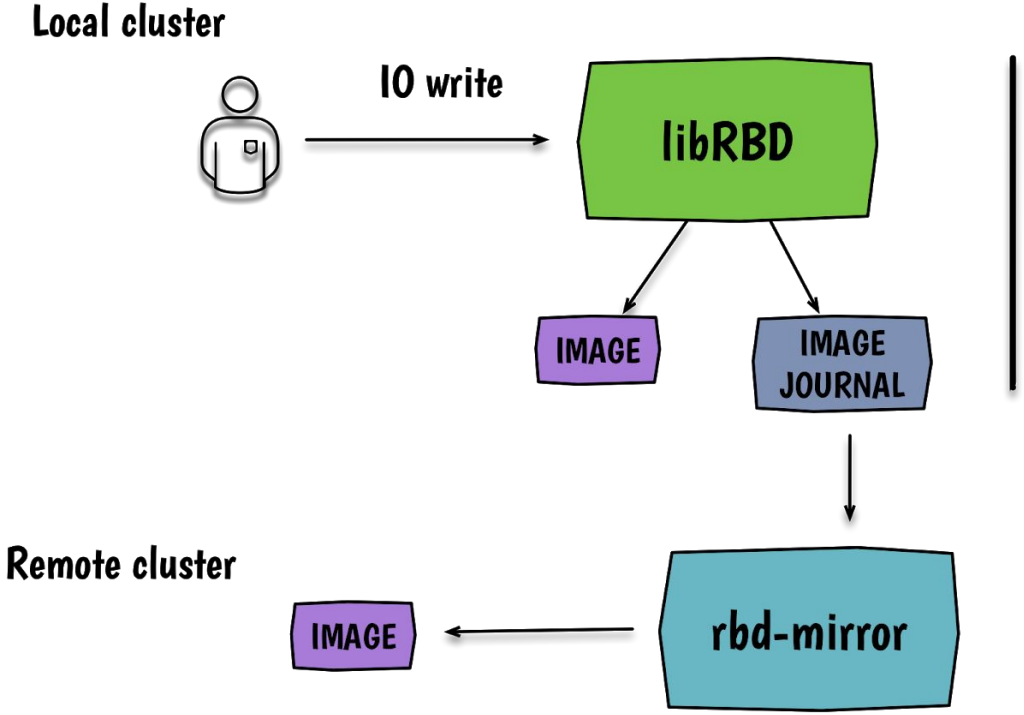
RBD Mirroring Setup

- Use symlinks to point to other cluster configuration
- Routable connectivity between sites
- Deploy the rbd-mirror daemon on each cluster
- Same pools configuration
- Add peering pool
- Add RBD image settings
    - Enable journaling on image
    - Mirror pool or specific images

Considerations:

- LibRBD-only, no current kRBD support
- Bluestore and EC pool for block (multiple pools, mirror only EC pool, leave metadata pool alone)

RBD mirroring write path



1. IO goes into the RBD's image journal
2. Once journaled, acknowledge the client
3. Write to the RBD image occurs
3. RBD mirror daemon replays the journal content at the remote location