Protecting the Galaxy

Multi-Region Disaster Recovery with OpenStack and Ceph

Sean Cohen
Sébastien Han
Federico Lucifredi

Your savers

Sean Cohen - Principal Product Manager Red Hat OpenStack Platform

Sébastien Han - Senior Domain Architect - http://www.sebastien-han.fr/

Federico Lucifredi - Product Manager Director, Red Hat Ceph Storage

DISCLAIMER

THIS PRESENTATION **ONLY FOCUSES ON DATA** DISASTER RECOVERY

Our Mission

IT organizations require a disaster recovery strategy addressing outages with loss of storage, or extended loss of availability at the primary site.

The general idea is to seamlessly and transparently backup OpenStack images and block devices from one site to another. So in an event of a failure resources in site A can be manually brought online in site B.

Assumptions

While designing your cloud environment, you must make sure that:

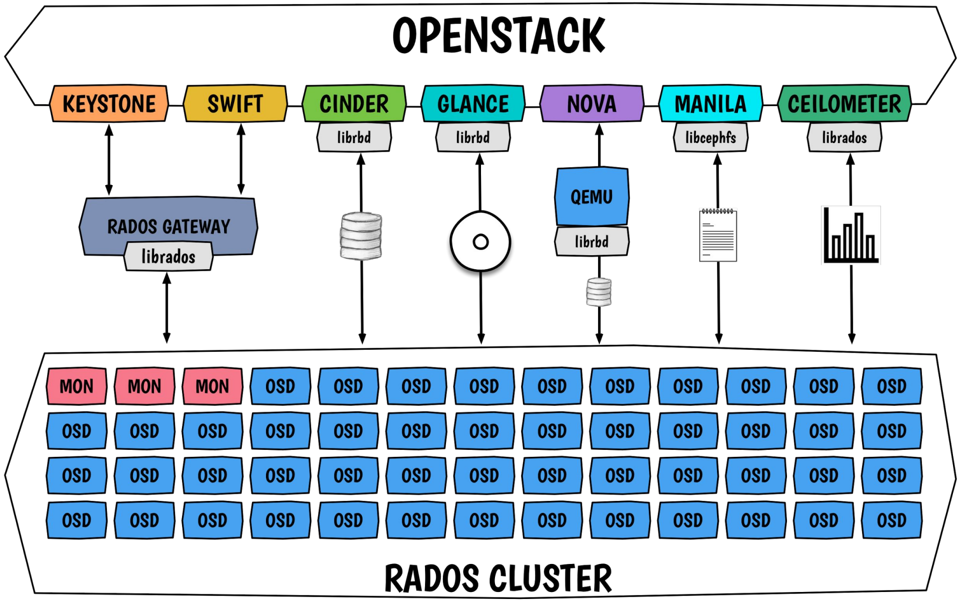
1. Images are template of your applications
2. Applications data is **always** hosted on Cinder block devices
3. **Only** ephemeral data should be stored on the virtual machine root disk
4. Your application stack is managed by **Heat** (or another automation tool)

In a failure scenario, the user ‘simply’ re-bootstraps the application stack using Heat, configures it using its configuration management system then starts the application.

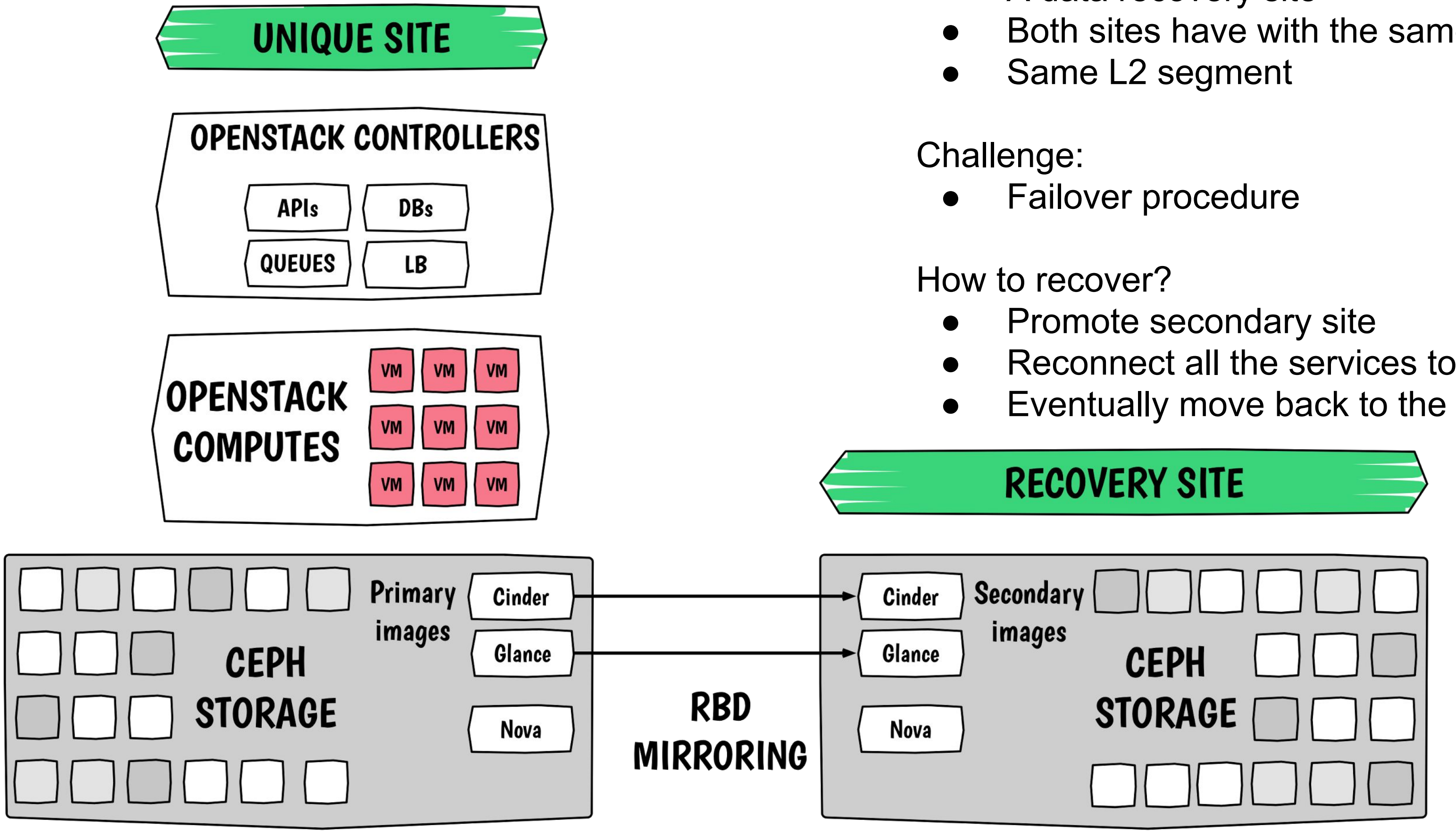
Longstanding effort building our Galaxy

- We have been busy building a strong single site model, now it's time to extend it to multi-site.

- You can start with a single site and add another one later. In the end, you don't need to re-architect your cloud while adding another location.



Use case architectures



Properties:

- Single OpenStack site
- A data recovery site
- Both sites have with the same cluster FSID
- Same L2 segment

Challenge:

- Failover procedure

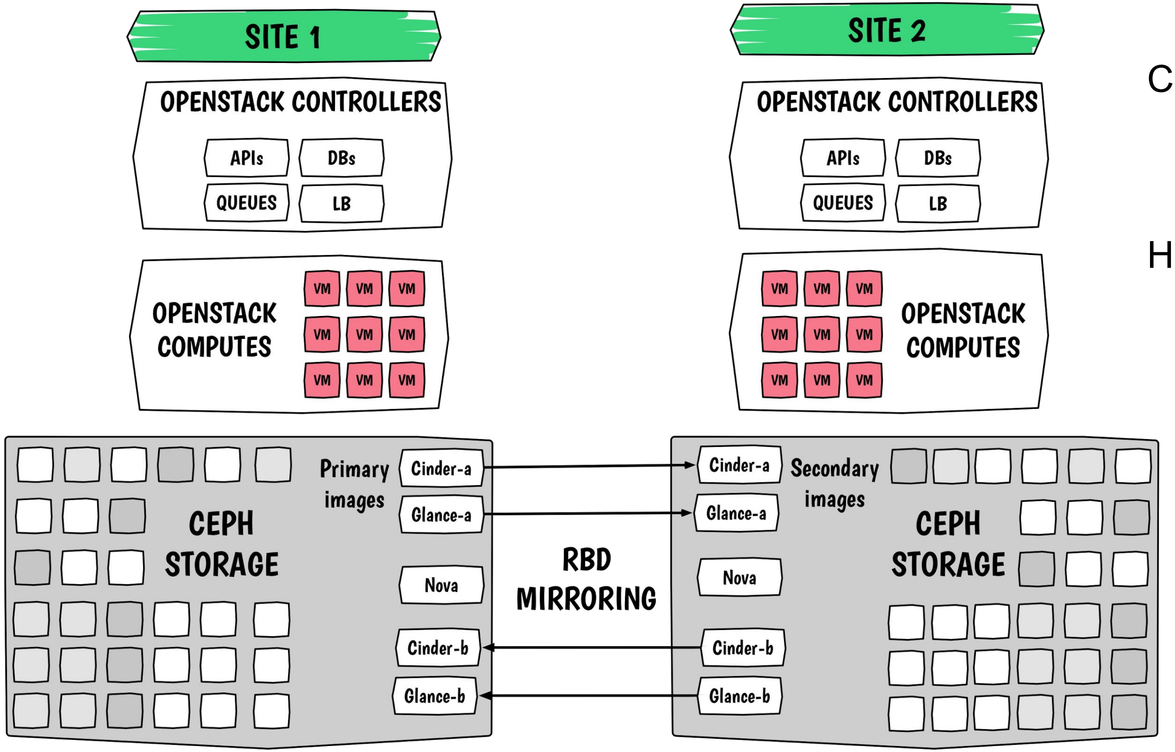
How to recover?

- Promote secondary site
- Reconnect all the services to the recovery cluster
- Eventually move back to the primary site

Expected capabilities

- Multiple **isolated** OpenStack environments
- Each site has **in-live/in-sync backup** of:
  - Glance images
  - Cinder block devices
- In an event of a failure, **any site can recover its data** from another site
- Storage architecture **based on Ceph**

REGIONS
NO SHARED KEYSTONE



Properties:

- Keystone on the controllers (as usual)
- Individual login on each region/site
- Both sites have each other's data
- Both sites have the same cluster FSID

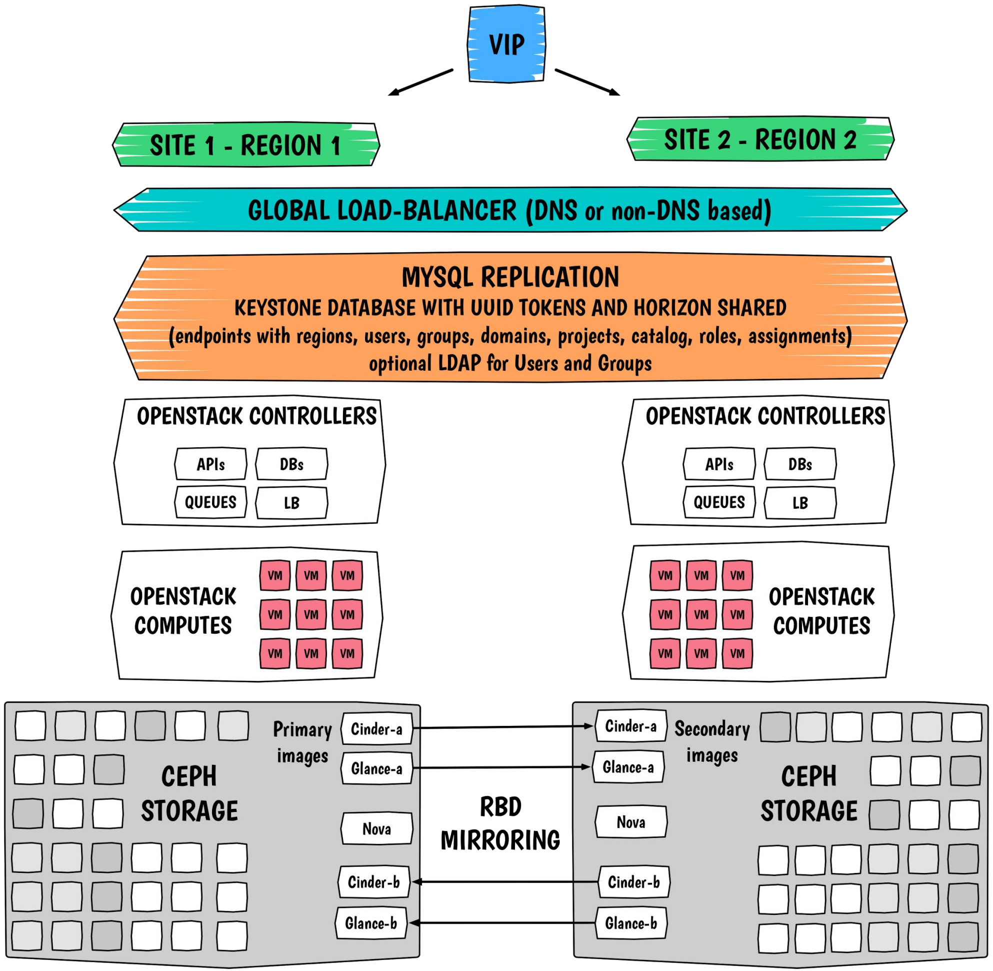
Challenge:

- Replicate metadata for images and volumes

How to recover?

- Promote the secondary site
- Import DB records in the survival site

SHARED KEYSTONE WITH REGIONS



Properties:

- Shared Keystone
- Keystone centralized and replicated DB
- Both sites have each other's data
- Works with N sites
- Both sites have with the same cluster FSID

Challenges:

- Replicate UUID tokens
- MySQL cross-replication over WAN
- Requires low latency and high bandwidth
- Fernet tokens are not ready yet

How to recover?

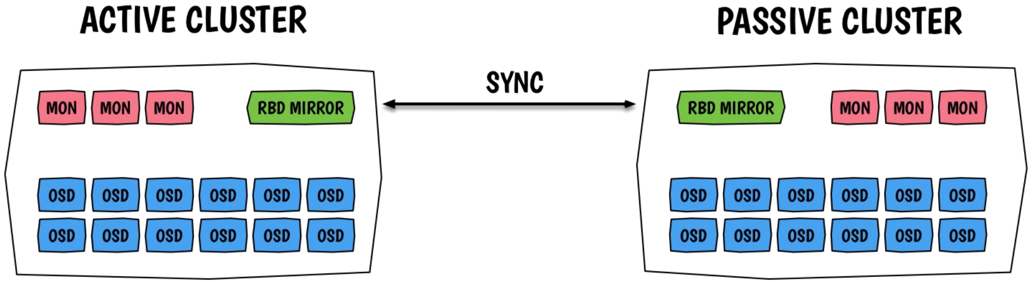
- Promote the secondary site
- Import DB records in the survival site

The road ahead with Ceph RBD Mirroring

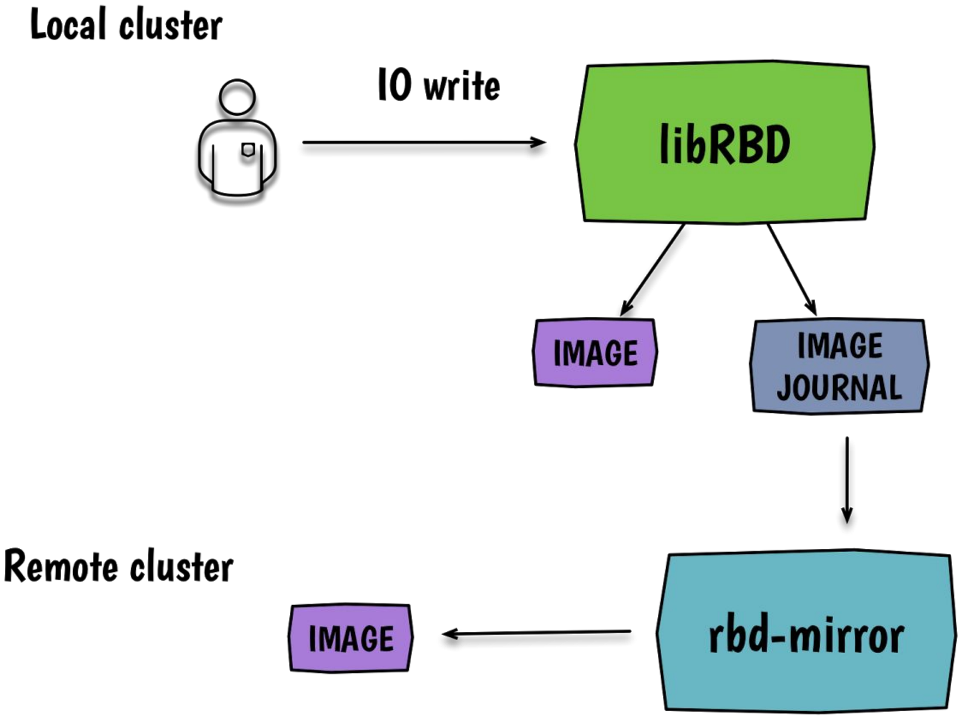
RBD mirroring

Now available with Ceph Jewel and this summer with the upcoming RHCS 2.0 release.

- New daemon ‘rbd-mirror’ synchronises 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)



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

RBD Mirroring Setup

- Use different cluster names; routable connectivity
- Deploy the rbd-mirror daemon on each cluster
- Same pool configuration at both sites
- Add peering pool
- Add RBD image settings
  - Enable journaling on image
  - Mirror pool or specific images

Challenges:

- No HA support for RBD-mirror yet
- Two sites only
- LibRBD-only, no current kRBD support

Where are we in Mitaka?

New API’s

- Cinder Replication v2.1 (“*Cheesecake*”) was implemented to support a disaster recovery scenario when an entire host can be failed over to a secondary site.
    - Allowing the preservation of user data access for ‘replication-enabled’ volumes type to allow cloud admins to rebuild/recover their cloud.
    - The new model is **backend/pool-based** rather than volume-based, so in a case of failover, you'll be failing over an entire backend.
    - This is a building block for Ceph Cinder replication support

Gap analysis

- Keystone: no real production readiness for Fernet Tokens yet
- Glance: no way to replicate images metadata to another site
- Nova: no way to replicate quotas, flavors, ssh keys etc…
- Cinder: pending support for Cinder replication API and the RBD driver with RBD mirroring

Some of these issues (metadata replication) are addressed by the Kingbird project: Centralized service for multi-region OpenStack deployments

Putting it all together

The road ahead in Newton

- RBD driver Cinder Replication support
  - Make use of the new replication API to support RBD Mirroring (promote/demote location)
- Necessary changes in the Cinder RBD driver to support RBD mirroring
  - Cinder type to point to a replicated Ceph pool
- Cinder Replication with More Granularity (“*Tiramisu*”)
  - Tiramisu API will be tenant facing. It gives tenant more control on what should be replicated together, i.e., a volume or a group of volumes. (using Replication Groups)
- Kingbird is really young but is a real enabler and the way toward multi-site

THANKS

May the force be with you!

“Backup” slides (mouahahah)

Quid of Cinder Backup?

Cinder backup is **for users only**:

- Should be enabled for Operators who don’t provide backups as part of their SLAs
- Should probably not be enabled if rbd mirroring is configured
- It prevents “accidental” deletion where the mirroring will delete the image on both locations
    - Will happen in RBD mirroring soon

Recovery procedure, **outage with loss of storage**

Applications do not move (and remain offline) but data can be recovered from site two.

1. Virtual machines impacted should be shutdown
2. Demote the current primary images (cinder volumes and glance images) on the local cluster
3. Promote the new primary images on the remote cluster
4. Shutdown services connected to Ceph (Glance, Nova, Cinder)
5. Rebuild the storage entity on site 1 using:
    a. same cluster FSID as the one that died
    b. same pools, users and keys
6. Add site 1 as a peer of Site 2, wait for the backfill of all images and volumes
    a. Site 1 is secondary
    b. Site 2 has been promoted primary earlier
7. Reconnect OpenStack services to the Ceph cluster on Site 1
8. Start the virtual machines on Site 1 and re-attach Cinder block devices

RBD Mirroring Setup (Gory Details Edition)

Use different cluster names; insure there is a routable connection from the remote site *to* the local cluster.

ENABLE MIRRORING: `rbd mirror pool enable {pool-name} {mode} #mode either` **`pool`** `or` **`image`**

ADD CLUSTER PEER: `rbd mirror pool peer add {pool-name} {client-name}@{cluster-name}`

ENABLE IMAGE JOURNALING SUPPORT: `rbd feature enable {pool-name}/{image-name} journaling`

ENABLE IMAGE MIRRORING: `rbd mirror image enable {pool-name}/{image-name} #only in image mode`

IMAGE PROMOTION AND DEMOTION: `rbd mirror image demote {pool-name}/{image-name}`