Data Protection and OpenStack Mitaka Live Webcast

Moderator: Thomas Rivera, Hitachi Data Systems
Panelists: Sam Fineberg, Hewlett Packard Enterprise
Ben Swartzlander, NetApp

June 22, 2016

Co-Sponsored by:
Today’s Presenters

Moderator

Thomas Rivera
Co-Chair, SNIA Data Protection Committee
Hitachi Data Systems

Panelists

Sam Fineberg, Ph.D.
Distinguished Technologist
Hewlett Packard Enterprise

Ben Swartzlander
OpenStack Architect
NetApp
SNIA Legal Notice

- The material contained in this tutorial is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA Education Committee.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.
OpenStack & Manila

Ben Swartzlander
OpenStack Architect
NetApp
What is OpenStack®

Free open source (Apache license) software governed by a non-profit foundation (corporation) with a mission to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable

- Massively scalable cloud operating system that controls large pools of compute, storage, & networking resources
- Community open source with contributions from 5000+ developers & 300+ organizations (Source: Stackalytics)
- Open web-based API Programmatic Infrastructure as a Service
- Plug-in architecture; allows different hypervisors, block storage systems, network implementations, hardware agnostic, etc.
What is OpenStack® (Cont.)

A series of interrelated projects that control pools of compute, storage, and networking infrastructure exposed as a consistent and open layer (API) for a heterogeneous Infrastructure as a Service (IaaS) environment.
Manila

- Manila is the OpenStack Shared File Systems Service that provides the management of file shares
  - (for example, NFS and CIFS)
  - Intended to be an open standard, highly available, and fault tolerant

- In the first half of this session, we’ll give you a brief overview of Manila, and talk about new features that were delivered for OpenStack Mitaka, focusing on data protection
Customer use case for Manila:

- 62% of disk capacity slated for file storage (Source: IDC)
- Self-service management & provisioning of shared file systems is hard
- Customers invent this themselves via scripting, automation, etc.
Manila: Status

- Juno (Oct 2014) and Kilo (Apr 2015) in production (supported downstream)
- Liberty (Oct 2015) supported upstream
- Mitaka released April 7th -- current stable release
- 18 drivers
  - from 12 storage vendors (RedHat, EMC, Intel, HDS, HPE, Huawei, IBM, NetApp, Quobyte, Oracle, Microsoft, Tegile)
Manila: New in Mitaka

- New first party drivers: LVM, ZFSonLinux
- New third party drivers: CephFS, Tegile
- New share access interface
- Share migration data service
- Export location metadata
- QoS support
- Share Replication…
Share Replication: Use Cases

- Protection from site-level outages
- Recovery from temporary and permanent outages
- End-user accessible (no admin-intervention required)
- Can be tested
- Completely automated
- NOT inter-cloud
- Disruptive (at the protocol layer)
Share Replication: Key Concepts

- **Share (an instance of a shared file system)**
  - User specifies size, access protocol, “share type”
  - Can be accessed concurrently by multiple instances

- **AZ (availability zone)**
  - Arbitrarily-defined failure domain
  - Concept exists across all OpenStack projects

- **Replica (one copy of a replicated share)**
  - Created one at a time, in different AZs
  - Each mirrors the primary
Admin Perspective

- Administrator acquires and configures storage infrastructure
- Replication not supported on all backends & not supported across vendors
- Administrator designs AZ boundaries and replication relationships
- Administrator sets up share types with replication
- Administrator sets quotas and limits
- NOT involved in creation of replicated shares, or failover/failback
End-User Perspective

- Determines what capabilities exist by looking at share types
- Decides which AZ(s) to run application in
- Creates replicated shares and add replicas one at a time
- Decides when to failover/failback storage
- Can TEST that failover/failback works and validate DR plans
- Zero interaction with human administrator
Example: Oracle DB

- Highly available database with VMs in multiple AZ
- NFS storage underneath
- Share replication can be synchronous or asynchronous
- User configures
  - Shares and replicas in each AZ
  - Instances in each AZ with Oracle and appropriate config
- In a disaster
  - User notices loss of primary AZ and invokes Manila set-active-replica, then pokes Oracle software already running in secondary AZ
Example: WordPress

- WordPress running on Apache+MySQL
- NFS storage underneath, asynchronous replication
- User configures
  - Shares and replicas in each AZ
  - Instances in each AZ with Wordpress and LAMP stack
- In a disaster
  - User notices loss of primary AZ and invokes Manila set-active-replica, then starts Wordpress and dependent software (MySQL most importantly) already running in secondary AZ
Future Work: Newton

- Quotas limiting number of replicas
- Support for replication with share servers (secure multi-tenant backends)
- API finalization and non-experimental status
- Admin APIs for snapshot replicas
- Revert to snapshot
Cinder

Sam Fineberg, Ph.D.
Distinguished Technologist
Hewlett Packard Enterprise
What is Cinder?

- **Block Storage Service for Openstack**
  - Provides persistent storage volumes for compute jobs
  - Plug-in framework w/ more than 70 drivers
  - Multiple storage fabrics
    - iSCSI, FC, NFS, RBD, …
Cinder Volumes

- Cinder storage is allocated as “volumes”
  - A volume is a detachable block storage device, like a disk
  - Volumes can have a “type”, defining device and class of storage
  - Volumes can be attached or detached from Nova jobs
Data Protection Mechanisms in Cinder

- Snapshotting
- Backup
- Replication
- Multi-attach
Snapshottting

- Cinder volumes can be snapshotted
  - Creates a point-in-time copy of the data that a volume contains
  - Snapshots can be used to create volumes, or to restore a volume back to a past state
  - Implementation mechanisms vary, depending on the storage hardware and driver

- By default, a volume being snapshotted must be unattached
  - You can “force” a snapshot to be taken on an attached volume
  - Snapshots are at block level and crash consistent
  - May lead to issues unless if writes cannot be quiesced
    - Not all drivers can support the force option
Consistency Groups

- If you have multiple related volumes you want to snapshot all at once
  - E.g., multiple database tables, logs, etc.
- A consistency group allows you to create snapshots sets at the exact same point in time from multiple volumes
- CG actions
  - When you create or update CGs you include a list of volume IDs
  - There are special CG snapshot commands
  - There is no “force” flag for cgsnapshot-create
Cinder Backup

- Cinder provides a block-level backup
  - Not application or filesystem aware, no consistency guarantees
  - Full and incremental backup support
  - Can backup attached volume using “force” flag
- Volumes can be “restored” from a backup
  - Restores the volume to the backup state
- Backup drivers
  - Default is a “dd” to a swift object
  - Alternative backup implementations are available as vendor specific plugins
  - Plugins may provide alternative targets and/or backup methods
Backup Changes for Mitaka

- **Backup snapshot**
  - You can now backup a cinder snapshot
  - Allows you to “snapshot” a volume and then backup, better consistency and lower downtime

- **New backup drivers**
  - Existing: Swift, Posix, NFS, GlusterFS, Ceph, IBM TSM
  - New driver for Google Cloud Storage

- **Backup driver decoupled from volume nodes**
  - Better scalability, can add backup nodes
  - Less impact of backup on the Cinder service
Replication

- Replication has been a difficult feature to add to Cinder
  - Cinder tries to hide storage devices
  - OpenStack doesn’t really have a concept of a “remote” datacenter
  - Vendor differences

- Multiple attempts
  - Early designs – vendor centric, hidden from cloud and apps
  - V1 – Juno, IBM only
  - V2 – Liberty – No drivers released (intentionally)
  - V 2.1 – multiple drivers in Mitaka
Simplified use case

- Disaster Recovery only
- Admin only
- Fail everything that is replicated
- Non-replicated volumes are “offline”
- Specified as part of volume “type”
Simplified use case

- Disaster Recovery only
- Admin only
- Fail everything that is replicated
- Non-replicated volumes are “offline”
- Specified as part of volume “type”
Replication

Limitations

- Secondary is not visible until after failover
- All volumes failover when disaster is declared
  - Replica storage becomes the only backend
  - No fallback, cleanup after failover is complicated
- Use case will be expanded in later releases

Available drivers in Mitaka

- HPE, Dell, EMC, Huawei, Storwize, IBM, Pure, Solidfire
- More coming in Newton
Multi-attach

- Allow volumes to be attached to multiple hosts or VMs
- Cinder support already in Mitaka
- Some Nova changes merged, still issues
- Not fully functional, expect progress in Newton
Other Enhancements

❖ Live migration
  ❦ Cinder has had volume migration for some time
    › Supports the movement of volumes between Cinder back ends
    › Can also live migrate attached volumes (when using Nova/libvirt)
  ❦ Cinder also supports Nova live migration
    › Migrate VMs to a different Nova host
    › Some limitations

❖ Rolling upgrades
  ❦ Backward compatible RPC APIs
    › Mitaka is backward compatible with Liberty
    › Uses oslo versioned objects, online DB schema upgrade
Q&A / Panel Discussion

Moderator

Thomas Rivera
Co-Chair, SNIA Data Protection Committee
Hitachi Data Systems

Panelists

Sam Fineberg, Ph.D.
Distinguished Technologist
Hewlett Packard Enterprise

Ben Swartzlander
OpenStack Architect
NetApp
Useful SNIA Resources

- This webcast and a copy of the slides will be on the SNIA Cloud Storage and Data Protection Websites and available on-demand
  - http://www.snia.org/forum/csi/knowledge/webcasts
  - http://www.snia.org/forums/dpco/knowledge/watch_learn
- A Q&A from this webcast, including answers to questions we couldn't get to today, will be on the SNIACloud blog
  - http://www.sniacloud.com/
- Please rate the Webcast by selecting the button
Thank You!