OpenStack Cloud Storage

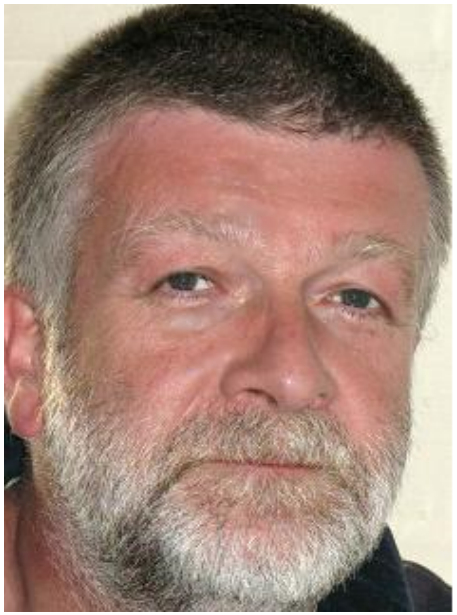
January 14, 2015

Today's Presenters

SNIA™
Cloud Storage Initiative

Sam Fineberg, Ph.D.
Distinguish Technologist
HP Storage

Alex McDonald, SNIA – CSI
Cloud Storage Initiative Chair
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.

Agenda

SNIA™ Cloud Storage Initiative

- Quick OpenStack overview
- Cloud Storage
- OpenStack and…
  - Cinder
  - Swift
  - Manila
- Summary, Q&A

What is OpenStack®

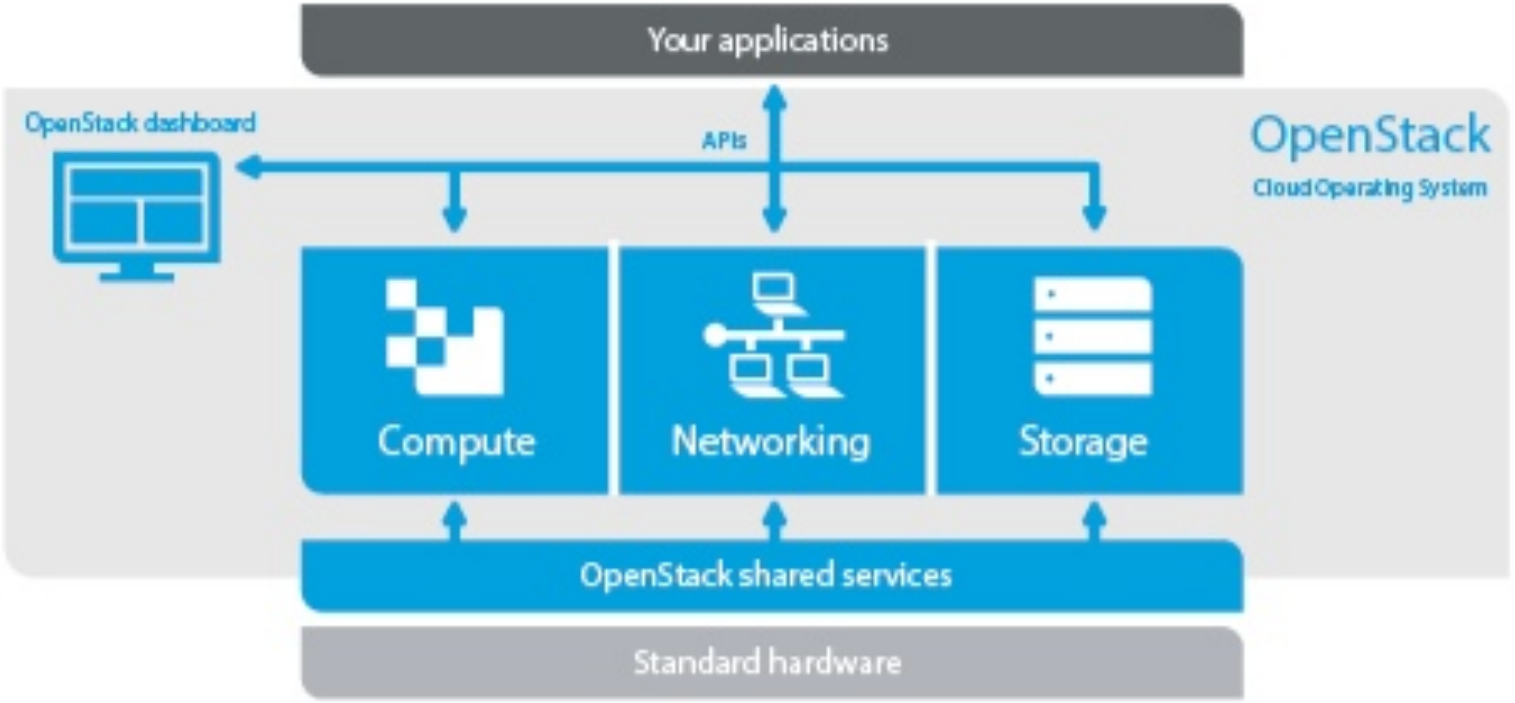
Free open source (Apache license) software governed by a non-profit foundation (corporation) with a mission **to produce the ubiquitous Open Source Cloud Computing platform that will meet the needs of public and private clouds regardless of size, by being simple to implement and massively scalable.**

- **Massively scalable** cloud operating system that controls large pools of **compute, storage**, and **networking** resources
- **Community open source** with contributions from **1000+ developers** and **180+** participating **organizations**
- **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®

SNIA™
Cloud Storage Initiative

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.



OpenStack® programs (13 Integrated; 2 supporting)

SNIA™ Cloud Storage Initiative

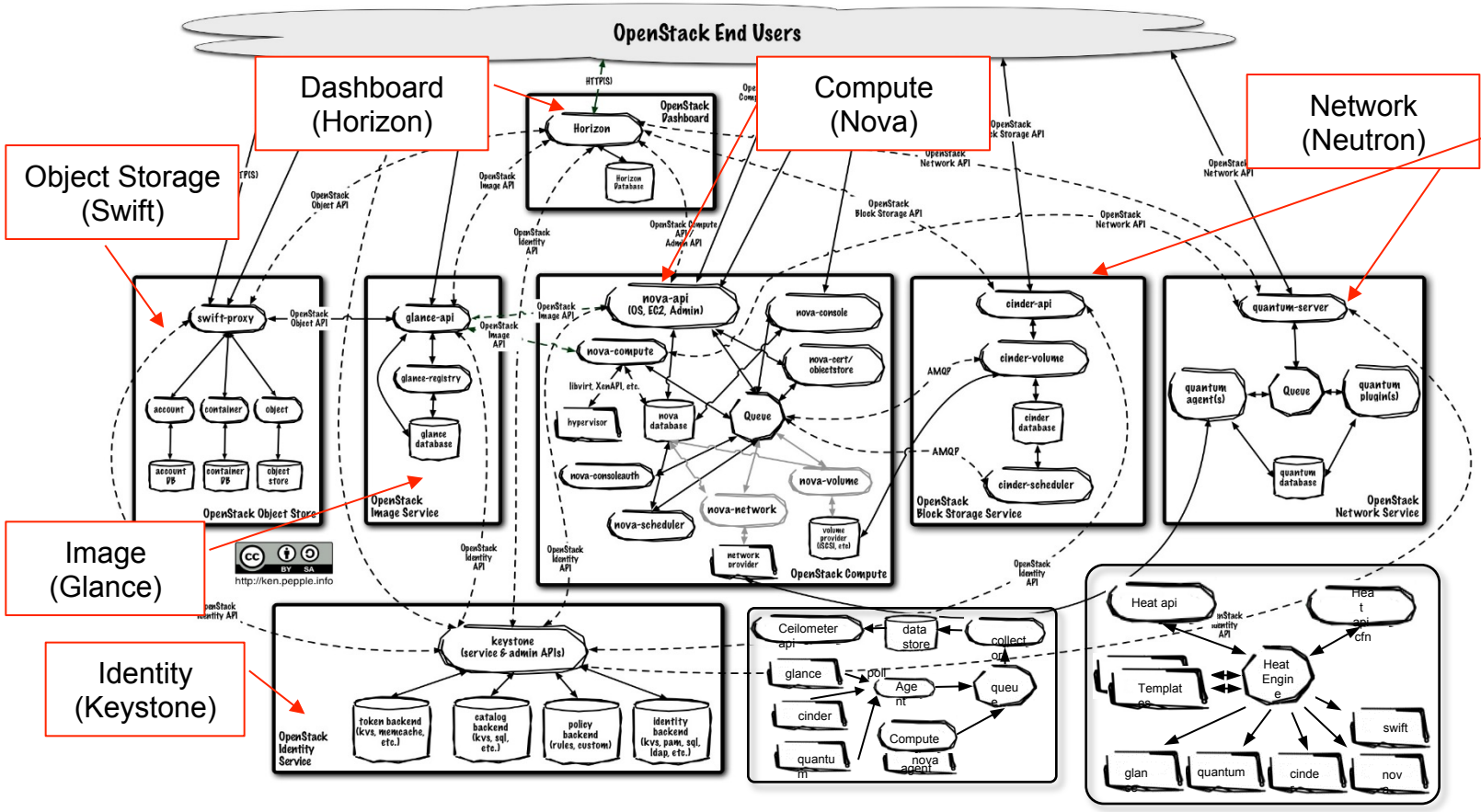
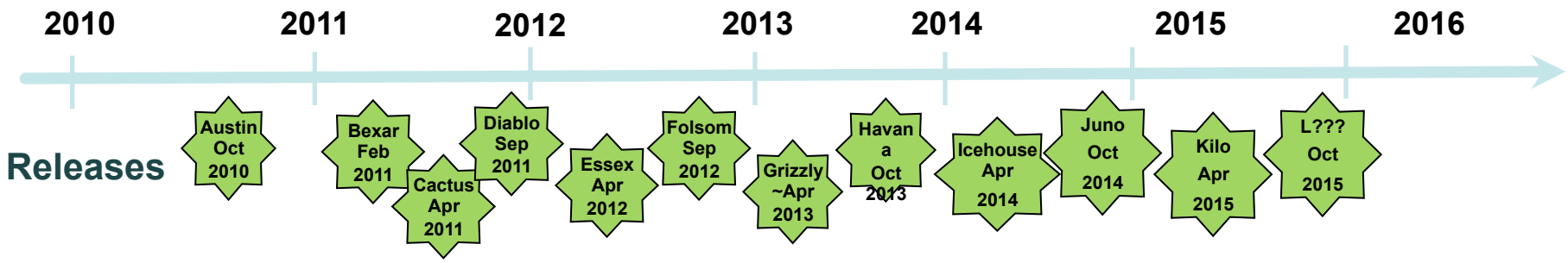


Image source: http://docs.openstack.org

OpenStack® releases



6-Month Release Cycles

- Spring and fall releases for predictable availability
- Release names use alphabetic naming

Planning

- Developers plan the next release at the Design Summit
- Sessions are selected by projects leads and are generally driven by blueprint topics
- Project Technical Leads (PTLs) accept a number of blueprints into the release plan

Development &Testing

- Milestone iterations (commonly 5 weeks) are defined and followed
- Development/documentation occurs per the blueprints and plan
- End of cycle testing with a defined Release Criteria process

Release

- Release Candidate is made available
- Next release is open for development
- After hardening, release occurs

Cloud Storage 101

SNIA™
Cloud Storage Initiative

Block Storage

- Generally SCSI protocol based, organized by volumes
- Boot volumes for VMs
- Ephemeral vs. Persistent
- Not directly consumed by applications, usually used to hold a filesystem
- Low level storage abstraction upon which file and object storage is built

File Storage

- Files organized in directory hierarchy and accessed by pathname
- File-based NAS protocols like NFS and CIFS
- Rich and complex application support: random access, multiple readers, in-place file updates, locking, etc.

Object Storage

- Efficient flat namespace: objects organized by accounts, containers, object keys, and metadata
- HTTP / REST / URL based – easily scriptable, many language choices
- Relatively simple interface compared to file storage
- Scalable to very high object counts
- More easily scaled across multiple geographies
- Ideal for relatively static data

Cloud Storage 101

SNIA™ Cloud Storage Initiative

Cinder

Block Storage

- Generally SCSI protocol based, organized by volumes
- Boot volumes for VMs
- Ephemeral vs. Persistent
- Not directly consumed by applications, usually used to hold a filesystem
- Low level storage abstraction upon which file and object storage is built

Manila

File Storage

- Files organized in directory hierarchy and accessed by pathname
- File-based NAS protocols like NFS and CIFS
- Rich and complex application support: random access, multiple readers, in-place file updates, locking, etc.

Swift

Object Storage

- Efficient flat namespace: objects organized by accounts, containers, object keys, and metadata
- HTTP / REST / URL based – easily scriptable, many language choices
- Relatively simple interface compared to file storage
- Scalable to very high object counts
- More easily scaled across multiple geographies
- Ideal for relatively static data

OpenStack Block Storage - Cinder

SNIA Cloud Storage Initiative

- Nova Volume
  - Originally OpenStack Compute (Nova) included support for ephemeral volumes
    - Used for boot/runtime storage of VMs
    - Volumes were typically backed by VM server files
  - Nova Volume had limited support persistent volumes on iSCSI
- Beginning with the Folsom release, a separate persistent block storage service, Cinder, was created
  - Cinder is a core part OpenStack project
  - Consists of a plug-in interface for supporting various block storage devices

Cinder Core Functionality

- Volumes
  - Create, Show, Update, Extend, Delete Volume
  - List Volume Summaries/Details
- Snapshots
  - Create, Show, Update, Delete Snapshot
  - List Snapshot Summaries/Details
- Volume types
  - List/Show volume types
  - Volume types defined in cinder.conf file, “extra specs” for advanced features
- Other features
  - Backup to swift
  - Volume migration
  - QoS – extended specs

Cinder Supported Devices

SNIA Cloud Storage Initiative

- Available drivers include
  - Ceph RADOS Block Device
  - Coraid AoE driver configuration
  - Dell EqualLogic volume driver
  - EMC SMI-S iSCSI driver
  - GlusterFS driver
  - HDS iSCSI volume driver
  - HP 3Par StoreServ Fibre Channel and iSCSI drivers
  - HP StoreVirtual (LeftHand)
  - Huawei storage driver
  - IBM XIV/DS8K volume driver
  - IBM GPFS volume driver
  - IBM Storwize family and SVC volume driver
  - NetApp unified driver
  - Nexenta drivers
  - NFS driver
  - SolidFire
  - VMware VMDK driver
  - Windows
  - XenAPINFS
  - XenAPI Storage Manager volume driver
  - Zadara

Cinder volume attachment

- NFS/shared file system
  - Volume file created on file share, attached through libvirt or other VM specific mechanism
  - All VM hosts must already be attached to file share
- iSCSI
  - Attach to iSCSI volume over TCP/IP network
  - Cinder provisions volumes on target, coordinates initiator and target connection
- FC
  - Pre-zoned/flat – Grizzly-Havana
  - Zoning – Icehouse-Juno
- Other options
  - Ceph RBD, ATAoE, …

Key new Cinder features - Juno

- Key new Cinder Features
  - Volume Replication
  - Consistency Groups and Snapshots of Consistency Groups
  - Support for Volume pools
  - Completion of i18n-enablement
  - Honor Glance protected properties in Image Upload
  - Enable ability to restrict bandwidth usage on volume-copy operations
  - Add Volume Num Weigher Scheduling
- Horizon (UI) improvements for Cinder
  - Enabling resetting the state of a snapshot/volume
  - Supporting upload-to-image
  - Volume retype
  - QoS (quality of service) support

15

Cinder Futures

- Automated discovery
- Asynchronous error reporting
- State machine
  - better error recovery
  - Less locks
- Over subscription ratios
  - for thin provisioning, dedupuplication, compression
- Horizon (UI) improvements

OpenStack Object Storage - Swift

- Swift was one of the original components of OpenStack
  - Originally developed by RackSpace
  - Distributed scale-out shared-nothing object storage
- Web APIs for data access
  - HTTP Put, Get, Post, Delete, …
- Swift is open source code, not an API standard
  - There is no compatibility suite and the API is subject to change
  - Enhanced direct implementations
    - SwiftStack, HP (StoreAll), etc.
  - Some vendors implement Swift compatible APIs
    - Ceph, Cleversafe, etc.
  - Some vendors have swift plugins – diskfile layer
    - Gluster, Seagate Kinetic, etc.

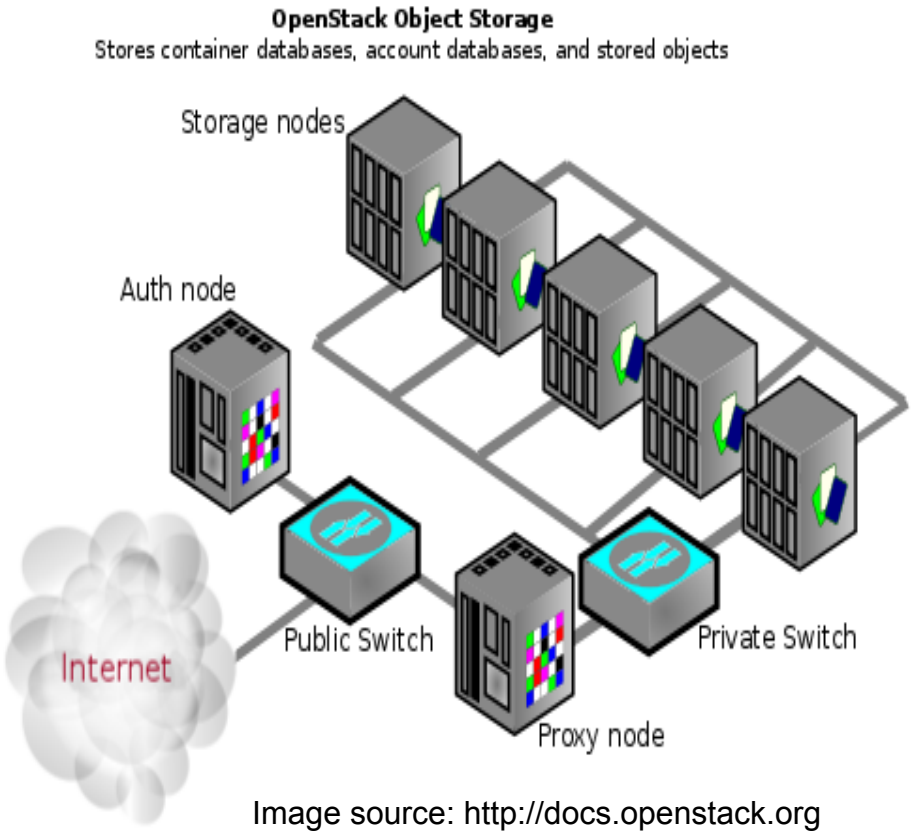


Image source: http://docs.openstack.org

Swift Storage Model

- Users authenticate with Keystone
- Each account owns a set of containers
- Containers hold a set of objects
  - Containers have metadata
  - Access permissions at container
- Objects contain data
  - Size is limited, composite objects supported for larger objects
  - Objects also have metadata
  - User specifies object name
  - Name can contain pseudo-paths

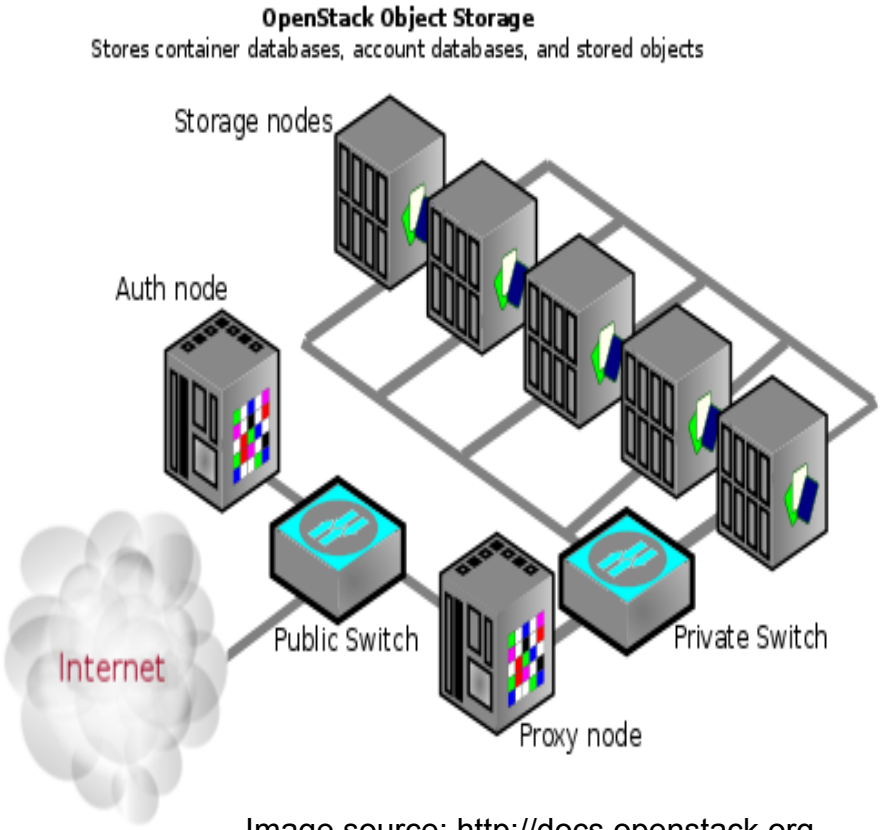


Image source: http://docs.openstack.org

Swift Operation

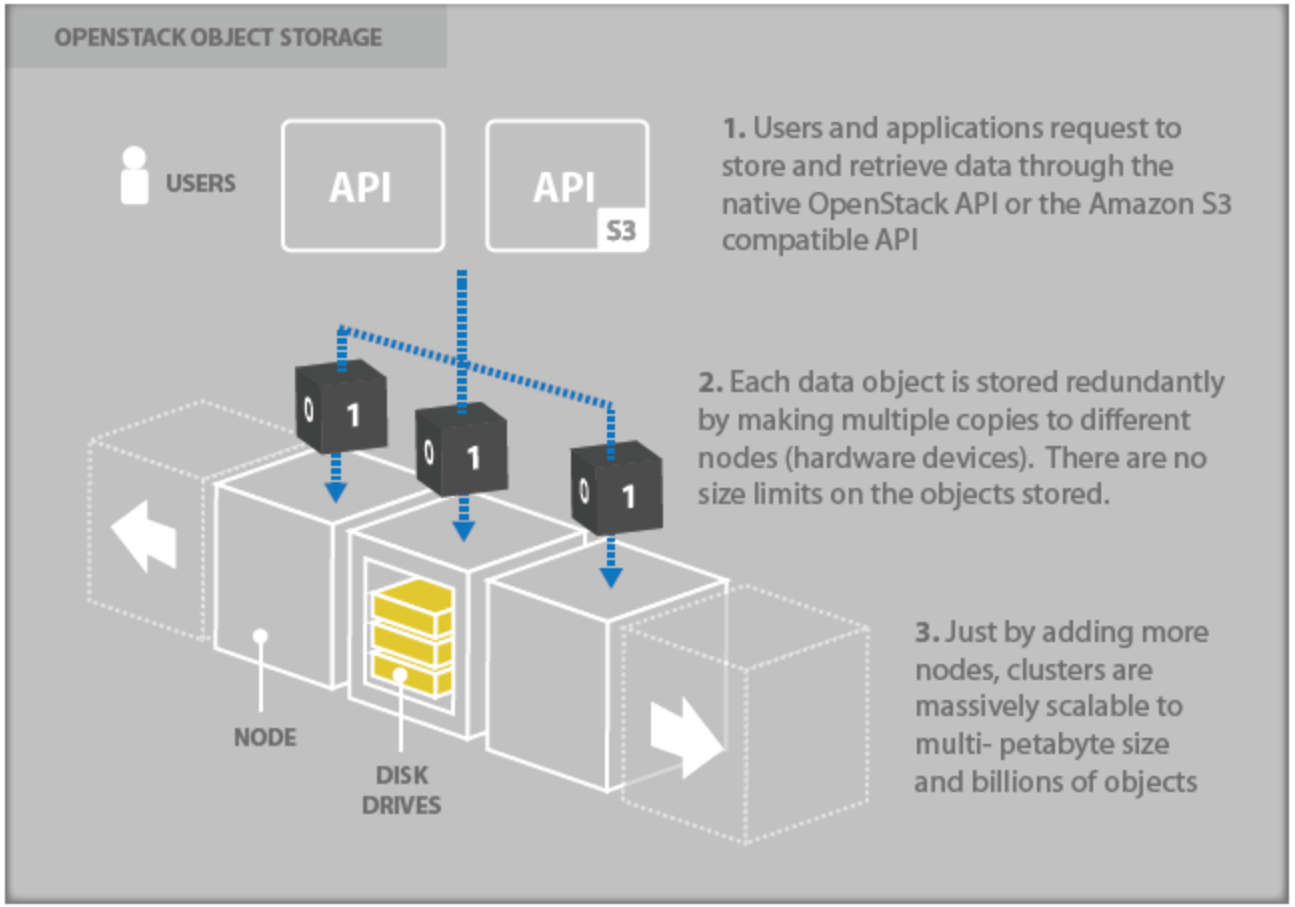


Image source: http://rogerluethy.wordpress.com

Swift Details

SNIA Cloud Storage Initiative

- The Ring
  - hash map between names and physical location
  - rings for accounts, containers, and objects, “partitions” are replicated by default 3 times
- Proxy Server
  - responsible for tying together the Object Storage architecture.
  - exposes public developer API, extensible using “middleware”
- Container Server
  - primary job is to handle listings of objects, stored as SQLite files
- Account Server
  - similar to the Container Server, responsible for list of containers
- Object Server
  - stores, retrieves and deletes objects
  - objects are stored as files, extensible using diskFile API

Swift Details (cont.)

SNIA™ Cloud Storage Initiative

- Replication
  - keeps the system in a consistent state through temporary errors
  - compares local data with remote copies to maintain the latest version.
  - replication updates are push based, updating is rsync or ssync
  - pushes records over HTTP or rsync/ssync database files.
- Updaters
  - sometimes container or account can not be immediately updated.
  - if an update fails, it is queued locally, the updater will process it
  - eventual consistency
- Auditors
  - Auditors crawl the local server checking integrity
  - If corruption found, file is quarantined, replication will replace
  - other errors are logged

New/future Swift Features

- Key new features in Juno
  - Storage policies
  - Keystone v3 support
  - Server-side account-to-account copy
  - Better partition placement when adding a new server, zone, or region.
  - Zero-copy GET responses using splice()
  - Parallel object auditor
- Future areas
  - Service tokens
    - Better security, for services like Glance
  - Single service node – PACOS servers
  - At rest encryption
  - Improvements to swiftclient and python SDK
  - **Erasure coding**

Swift Futures – Erasure Codes

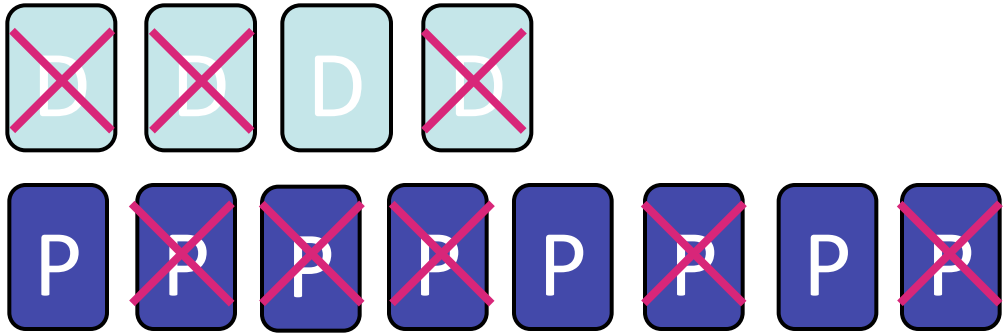
- Erasure coding has been used widely in RAID for years
  - Can yield improved storage efficiency and/or reliability
  - Objects broken into data and parity "shards" by proxy servers,
  - Shards stored like objects, reconstruction at proxy when retrieved

Put: 

Data: 

Parity: 

Fragments, schematically:



Swift Futures – Erasure Codes

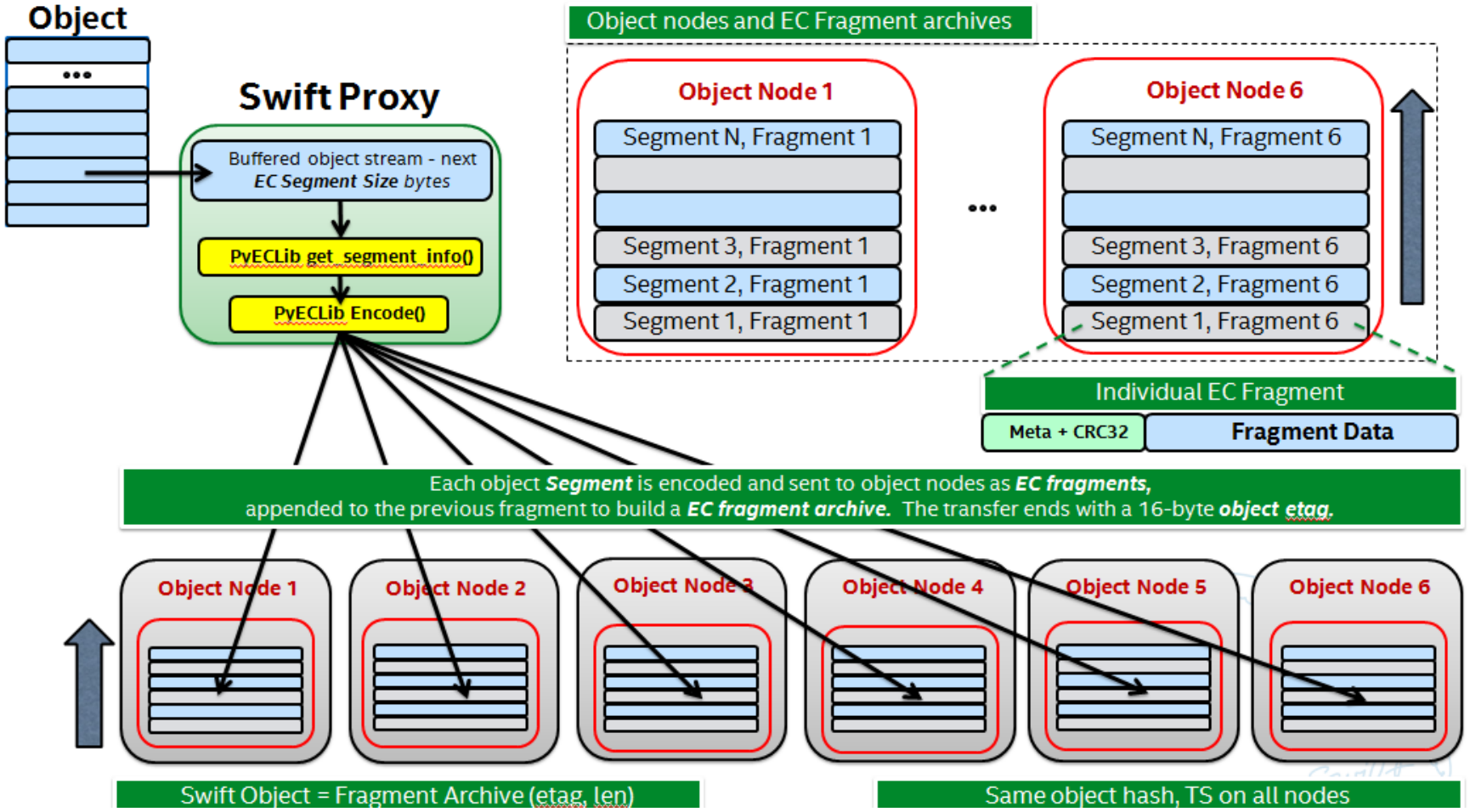


Image source: http://specs.openstack.org/openstack/swift-specs/specs/swift/erasure_coding.html

OpenStack Shared File Storage

- Not originally planned for OpenStack,
  - Clouds like AWS don't have shared file storage
  - Security and networking issues are difficult
- Demand has continued for this type of a service
  - For legacy applications that are not object storage enabled
  - Other options like running NFS on a VM or Gluster across nodes are complex to get right
- Originally proposed as an extension to Cinder
  - Implemented in Grizzly Cinder
  - Split into a new project "Manila" in Havana

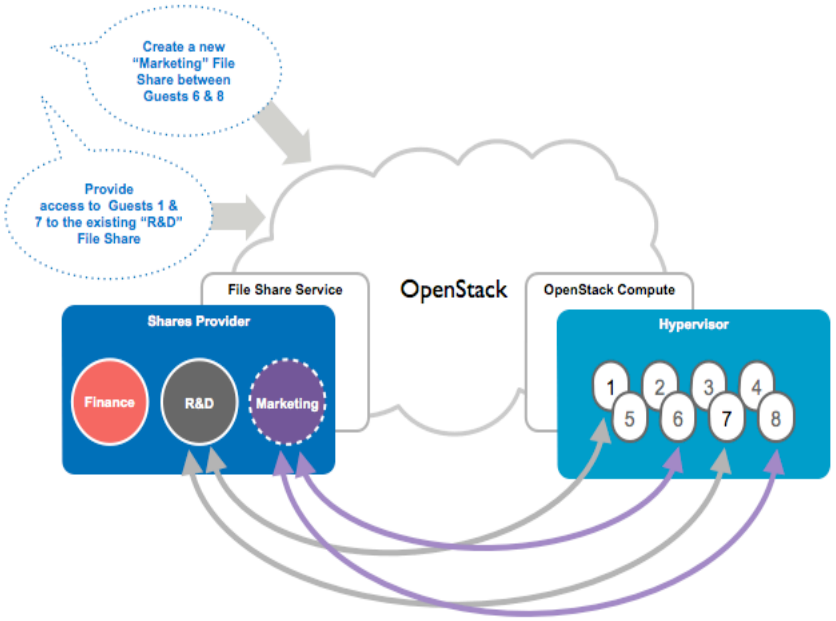


Image source: http://wiki.openstack.org/wiki/Manila

Manila vs. Cinder

- Shared Filesystems
  - Instances are able to communicate directly with the storage backend over the network
  - Uses NFS or CIFS to talk to shares, multiple readers/writers are expected
  - Storage backend is able to serve multiple tenants while maintaining secure separation between them
  - Uses the tenant (neutron) network
- Block storage
  - Usually iSCSI or FC, like a disk
  - Exposed through the hypervisor, actual “mount” is to VM host
  - Typically single attach (or if multiple attach coordination is required to corruption)
  - Utilizes a private storage or admin network

Manila core functionality

- Back-ends
  - Generic – creates a VM running SMB or NFS
  - Vendor drivers
    - For NFS or CIFS filer appliance
    - NetApp, EMC, HP, others
- Functions
  - Create or delete a share
  - Show or list shares
  - Allow/deny access to a share
  - Create/delete share snapshots, list snapshots

Manila Project Status

- Manila emerged as a separate project over Summer 2013
  - Recently promoted to "Incubator" project for Juno
  - Goal is to it part of the Kilo (or later) "integrated release"
  - Development is increasing since the Kilo summit
- Future development areas
  - Multitenancy
    - Compatibility with Neutron
    - Helper VMs
    - Server VMs vs. physical servers
  - Progress towards inclusion in "integrated release"
    - Incubator status is going away
  - Mount Automation
  - Access groups
  - Hardening, broader driver support

Summary

SNIA™ Cloud Storage Initiative

- OpenStack is free open source software for implementing public and private clouds
  - Many companies and individuals are involved in development
  - Products and services based on OpenStack exist today, and more are appearing as it matures
- OpenStack has support for block, object, and file storage
  - Both open source and commercial storage products supported
  - These provide basic and advanced functionality
  - More features, drivers, implementations are in the works
- OpenStack is maturing quickly
  - Give it a try – multiple free and commercial source distributions
  - There is plenty more work to do
  - The development process is open to anyone who wants to help

Questions?

SNIA™
Cloud Storage Initiative

- More Information
  - http://openstack.org
  - http://docs.openstack.org
  - https://launchpad.net/cinder
  - https://launchpad.net/swift
  - https://launchpad.net/manila
- Contact me:
  - fineberg@hp.com

After This Webcast

SNIA Cloud Storage Initiative

- This webcast and a copy of the slides will be posted to the SNIA Cloud Storage Initiative (CSI) website and available on-demand
  - http://www.snia.org/forum/csi/knowledge/webcasts
- A full Q&A from this webcast, including answers to questions we couldn't get to today, will be posted to the SNIA-CSI blog
  - http://www.sniacloud.com/
- Next Webcast: OpenStack Manila – January 29, 2015. Register now at: http://bit.ly/1wPkbOU
- Follow us on Twitter @SNIACloud

Questions?

SNIA™
Cloud Storage Initiative

- More Information
  - http://openstack.org
  - http://docs.openstack.org
  - https://launchpad.net/cinder
  - https://launchpad.net/swift
  - https://launchpad.net/manila
- Contact me:
  - fineberg@hp.com