OpenStack & CEPH

Author: Furkat Gofurov **e-mail:**(furkatgafurov1994@gmail.com)

Course author: Andrey Shevel **e-mail:**(Shevel.Andrey@gmail.com)

Course: Cluster, Grid, Cloud computing systems

Place: ITMO University (Saint Petersburg National Research University of Information Technologies, Mechanics and Optics)

Date: 6.06.2017

PERCCOM asmus undus

Erasmus+

ITMO UNIVERSITY

OpenStack

OpenStack is a cloud operating system to control and manage large pools of storage, compute and networking resources throughout a datacenter, all managed with the use of dashboard which enables administrator to control while empowering their users to provision resources through a web interface.

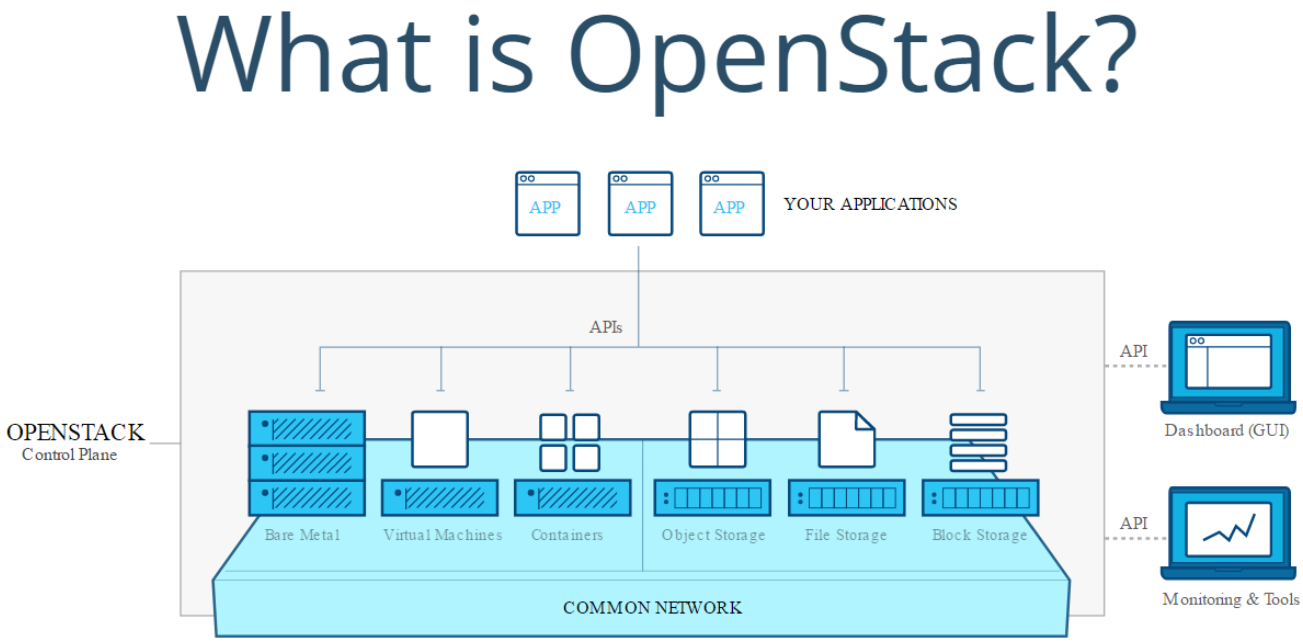


Fig 1. OpenStack Architecture

The OpenStack Mission: to produce a ubiquitous Open Source Cloud Computing platform that is easy to use, simple to implement, interoperable between deployments, works well at all scales, and meets the needs of users and operators of both public and private clouds.

What can OpenStack Automate?

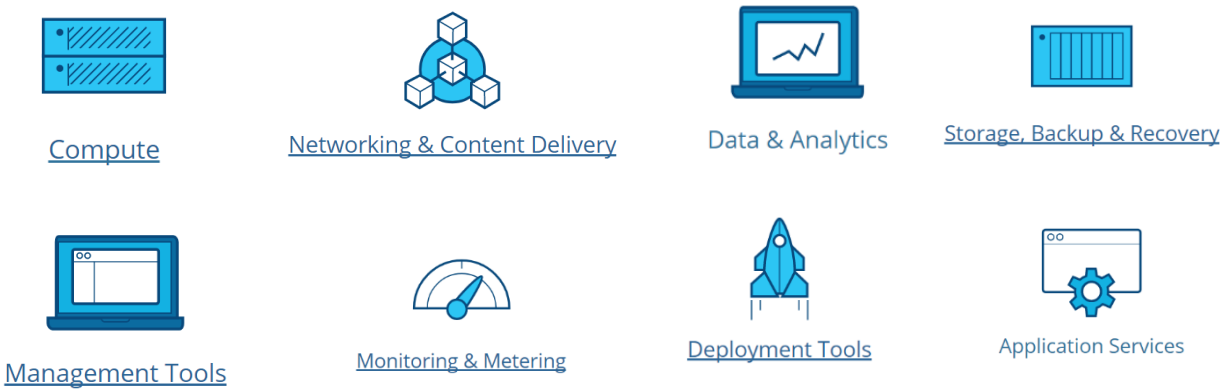


Fig 2. OpenStack components

Popular Project Set

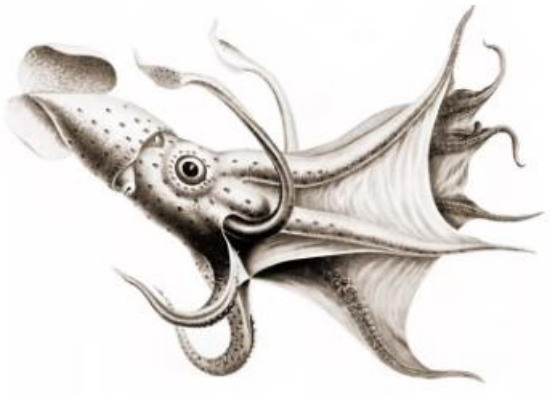
1. OpenStack Compute (**Nova**) is designed to manage and automate pools of computer resources and can work with widely available virtualization technologies, as well as bare metal and high-performance computing (HPC) configurations.
2. OpenStack Networking (**Neutron**) is a system for managing networks and IP addresses.
3. OpenStack Object Storage (**Swift**) is a scalable redundant storage system. Objects and files are written to multiple disk drives spread throughout servers in the data center, with the OpenStack software responsible for ensuring data replication and integrity across the cluster.
4. OpenStack Image (**Glance**) provides discovery, registration, and delivery services for disk and server images.
5. OpenStack Identity (**Keystone**) provides a central directory of users mapped to the OpenStack services they can access. It acts as a common authentication system across the cloud operating system and can integrate with existing backend directory services like LDAP.
6. OpenStack Block Storage (**Cinder**) provides persistent block-level storage devices for use with OpenStack compute instances.

Advantages

• Option of having private or public clouds
• Available anytime at any computer or location through a web browser
• Low costs per megabyte of storage and customers pay for what they use
• Provides an infrastructure as a server (Iaas) for managing large groups of public or private clouds
• Has “Dashboard” for letting users organize and access data/resources
• Unlimited storage
• Protects drive failures by preventing and controlling data corruption
• Users can set an expiration time (e.g. expiry for sales offers in amazon)

Disadvantages

• Servers are not always reliable and issues could dissatisfy customers
• Technical support is offered ONLY through email and chat
• Uploads are time consuming
• Software still being produced
• Software is constantly changing and the user must keep up with up to date with changes
• Is not compatible with multi-languages or multi-currency

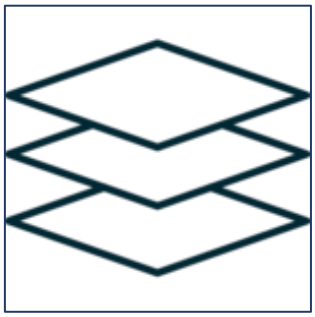


The Ceph Difference

Ceph's CRUSH algorithm liberates storage clusters from the scalability and performance limitations imposed by centralized data table mapping. It replicates and re-balance data within the cluster dynamically-eliminating this tedious task for administrators, while delivering high-performance and infinite scalability.

Ceph Storage

The power of Ceph can transform your organization's IT infrastructure and your ability to manage vast amounts of data. Ceph's foundation is the Reliable Autonomic Distributed Object Store (RADOS), which provides your applications with **object**, **block**, and **file system** storage in a single unified storage cluster-making. Ceph flexible, highly reliable and easy for you to manage.



Object storage

- partial or complete reads and writes
- snapshots
- atomic transactions with features like append, truncate and clone range
- object level key-value mappings

Block storage

- Thinly provisioned
- Resizable images
- Image import/export
- Ability to mount with Linux or QEMU KVM clients!

File System

- Stronger data safety for mission-critical applications
- Virtually unlimited storage to file systems
- Applications that use file systems can use Ceph FS natively
- Ceph automatically balances the file system to deliver maximum performance.

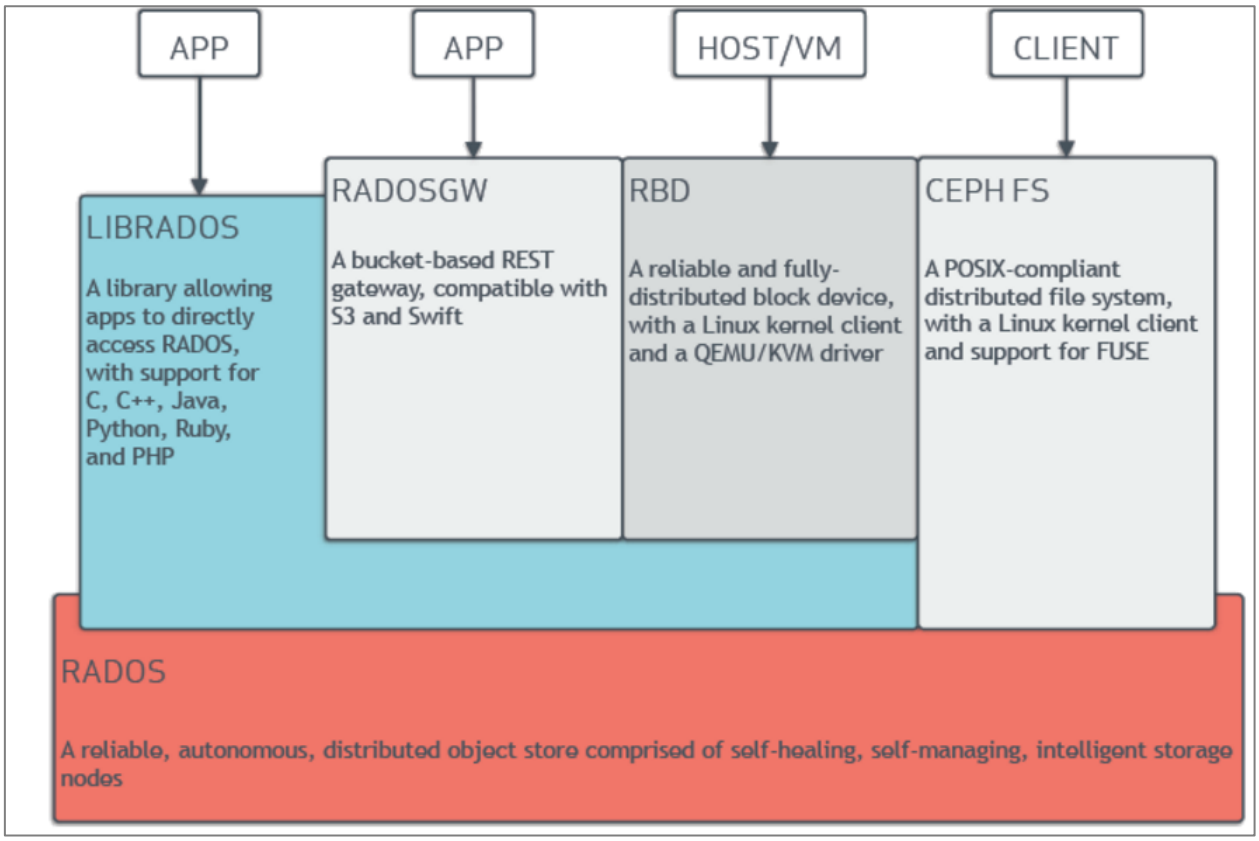


Fig 3. Ceph Architecture

Ceph has its own set of issues, especially in a cloud context. Its multi-region support, while often cited as an advantage, is also a master-slave model. With replication possible only from master to slave, uneven load distribution in an infrastructure that covers more than two regions.

Ceph's two-region design is also impractical as writes are only supported on the master, with no provision to block writes on the slave. In a worst case scenario, such a configuration can corrupt the cluster.

Another drawback to Ceph is security. RADOS clients on cloud compute nodes communicate directly with the RADOS servers over the same network Ceph uses for unencrypted replication traffic. If a Ceph client node gets compromised, an attacker could observe traffic on the storage network.

On the other hand, Ceph is strongly consistent across the cluster. This means that when data is read back from Ceph, it is guaranteed to be current. Swift on the other hand will ensure that data written is protected before acknowledging the write, but it may take additional time to update all previous versions of the data across the cluster.

A comparative review of OpenStack (Swift & Cinder) vs Ceph

There are similarities that cause this confusion:

- They are both experiencing significant adoption.
- They are both powered by open source projects with community support.
- They are both software that enables scale-out storage architectures and take advantage of standard hardware.
- They are both productized by commercial companies so all enterprises can utilize them. Ceph via RedHat and Swift via SwiftStack.

Comparison of Ceph vs Swift:

1. Ceph better for databases and other real-time data, where Swift is better for large-scale, multi-region clusters storing unstructured data.
2. Ceph is a block-focused product that has gateways to address it other ways (object, file). On the other hand, Swift is an object-focused product that can use gateways to support file access.
3. Ceph performs better in terms of transfer speed and latency. Ceph clients directly contacts the storage nodes for data retrieval/storage. But in case of swift the traffic to and from the Swift cluster goes through the proxy servers. Thus swift has a bottleneck compared to Ceph.
4. Ceph monitors has monitor nodes which gives cluster maps to the clients and storage nodes. Clients can thus directly contact the storage nodes to access data. This procedure is faster and gives lesser overhead when compared to swift.
5. In order to access a swift storage we use HTTP REST interface. There is no other access point. But ceph can be accessed via a number of methods. Ceph provides a scalable, consistent object store and a bunch of interfaces to access it, including native access, an http REST API, block devices and a filesystem-type interface.
6. In case of read operations ceph performs better over swift. Ceph manages a higher number of read operations than swift when the data size is small. When the object size goes higher, the amount of read operations that the two systems can perform is approximately the same, but each system reaches its highest performance with a different number of threads.
7. For write operations, Ceph performs better when the size of the objects is small. The Ceph I/O Performance scales over Swift because ceph clients connects to OSD's directly. Swifts I/O performance is limited by the proxy server which may increase the bottleneck. So Ceph performs faster and has smaller overhead. The lookup procedure in ceph is faster due to the use of crush algorithm. Cephs response time is excellent for larger objects
8. Ceph performs better for multi user environment as there is less performance degradation as clients increase. Ceph also gives better bandwidth at lower concurrency.

Better…

Whicb one to choose…?	Ceph	Swift
Transfer speed and latency	✓	
Databases and real-time data	✓	
Large-scale, multi-region clusters		✓
Access storage nodes faster and less overhead	✓	
Read operations	✓	
Multi user environment	✓	
Better bandwidth at lower concurrency	✓	
Performance degradation as client size increases	✓	

Table 1. Ceph vs Swift

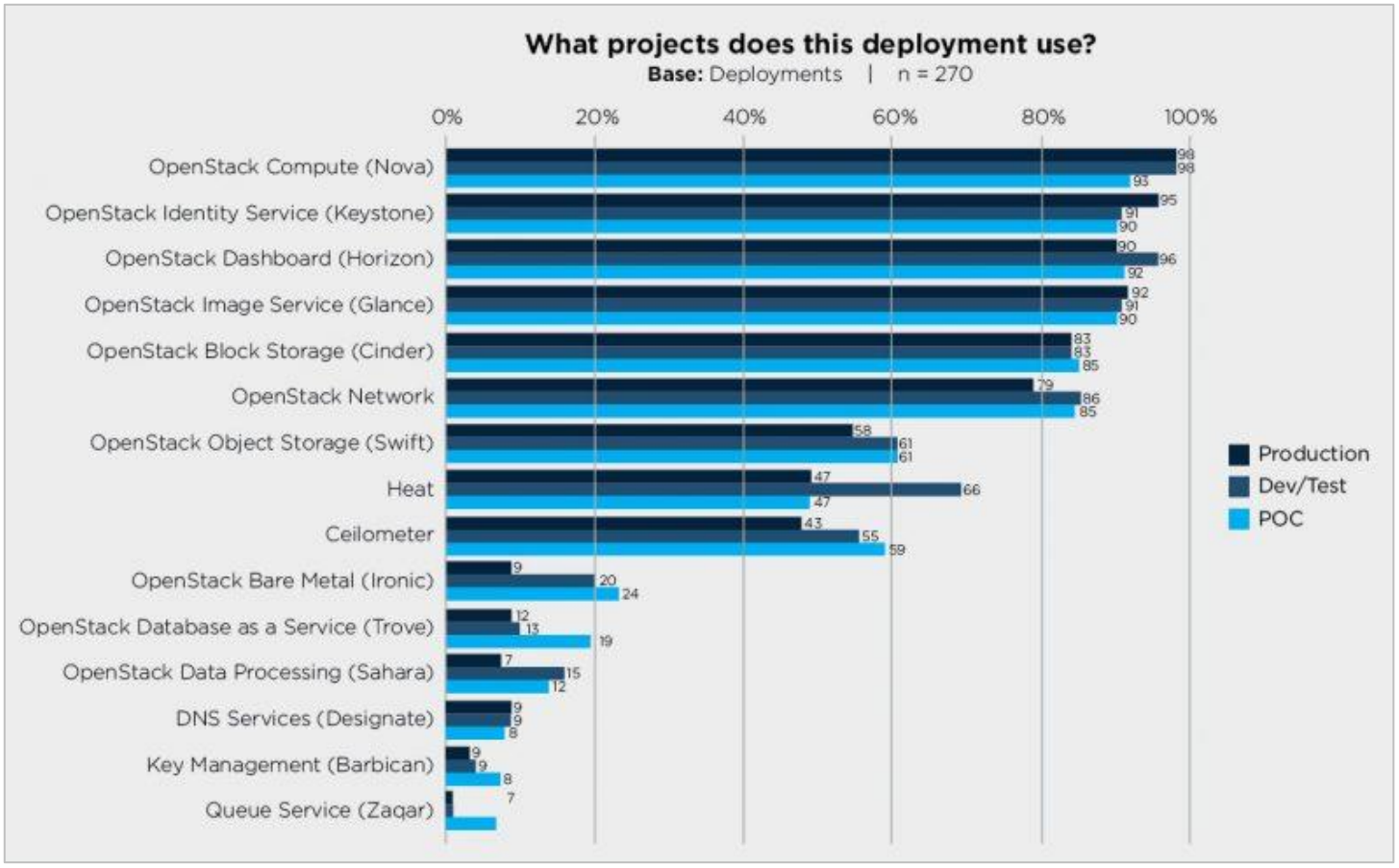


Fig 4. OpenStack Project Deployment

In case of deployment of storages in projects, the above given graph compares Cinder, Swift or general Openstack projects based on 3 categories: Production, dev/test and POC. We can observe that, Cinder (OpenStack Block Storage) and Swift (OpenStack Object Storage) are both popular deployments.

Rather than choosing one over the other, it may make sense to have both alternatives in the same cloud infrastructure. For example, Ceph for local high performance storage while Swift could serve as a multi-region Glance backend where replication is important but speed is not critical.

Comparison of Ceph vs Cinder:

	Ceph	Cinder
Pure Play SDS Controller		✓
Vendor Neutral		✓
Supports many Clouds	✓	
Manage Block Devices	✓	✓
Manage Filesystems	✓	
Manage Object Systems	✓	
Scale-out Design	✓	
Built-in HA / Resiliency	✓	
REST API	✓	✓
GUI	✓	w/Horizon
Extensible via Plugins		✓

Table 2. Ceph vs Cinder

Consider…

If you need…	Ceph	Cinder	Swift
NAS and Scale out NAS	✓		
SAN	✓	✓ Consider with Ceph Plugin or other storage systems	
Shared Filesystems	✓		
Object Storage	✓		✓

Table 3. Ceph vs Cinder vs Swift

References:

1. https://www.openstack.org/software/
2. http://ceph.com/ceph-storage/
3. https://www.youtube.com/watch?v=QBkH1g4DuKE ceph promo video
4. https://www.mirantis.com/blog/ceph-vs-swift-architects-perspective/
5. http://www.sparkmycloud.com/blog/a-performance-review-of-swift-vs-ceph/
6. https://www.swiftstack.com/blog/2016/03/29/ceph-vs-swift...not-a-rivalry/