OpenStack Overview

- OpenStack is an Infrastructure as a service (IaaS) which is known as a Cloud Operating System, that takes resources such as compute, storage, network, virtualization technologies and controls those resources at a data center level.

- OpenStack's basic requirement: "cloud must be simple to implement and massively scalable."

Why OpenStack

Community with Broad Commercial Support

Here’s solution: OpenStack
OpenStack Project

- OpenStack Compute (Nova): Provision, manage and deploy large networks of virtual machines.
- OpenStack Object Store (Swift): Create and store large quantities of data, and access any object within a large scale.
- OpenStack Image Service (Glance): Catalog and manage large inventories of server images.
- OpenStack Quantum Service: Provide network as a service to compute.
- Other components: Dashboard, Authentication (Keystone), ... (CL)

Identity ("Keystone")

Keystone provides a single point of integration for OpenStack's policy, catalog, token, and authentication.
- Keystone handles API requests as well as providing a configurable catalog, policy, token and identity services.
- Standard backends include LDAP or SQL as well as Key Value Stores (KVS).
- Most people will use this as a point of customization for their current authentication services.

Keystone Main Functions

- Provides 4 primary services:
  - Identity: User information authentication
  - Token: After logged in, replace account-password
  - Service catalog: Service units registered
  - Policies: Enforces different user levels
- Can be backed by different databases:
  - LDAP
  - SQL
  - Key Value Stores (KVS)

Keystone -- Process

Object Storage ("Swift")

- Stores and serves objects (files)
- Employs object level replication to safeguard data
- Accepts client requests via Objectstore API or HTTP from clients through swift-proxy
- Maintains distributed account and container databases
- Stores objects according the ring layout on filesystem with extended attributes (XFS, EXT4, etc.)
Glance

- Image storage and indexing.
- Keeps a database of metadata associated with an image, discover, register, and retrieve.
- Built on top of Swift, images store in Swift
- Two servers:
  - Glance-api: public interface for uploading and managing images.
  - Glance-registry: private interface to metadata database
- Support multiple image formats

Image Service ("Glance")

- glance-api accepts image API calls for image discovery, image retrieval and image storage.
- glance-registry stores, processes and retrieves metadata about images (size, type, etc.).
- Database to store the image metadata.
- A storage repository for the actual image files; in many deployments, this is OpenStack Swift

Block Storage ("Cinder")

- cinder-api accepts API requests and routes them to cinder-volume for action.
- cinder-volume acts upon the requests to read or write to the Cinder database to maintain state, interacting with other processes like cinder-scheduler through a message queue and directly upon block storage providing hardware or software.
- Cinder provides a variety of storage options through a diverse architecture. Currently, these include Access Block Storage (ABS), Amazon S3, Swift, NFS, and other storage providers.
- Wills the nova-scheduler, the cinder-scheduler deploys and the proper block storage provider route to create the volume.

Nova

- Major components:
  - API: public facing interface
  - Message Queue: Broker to handle interactions between services, currently based on RabbitMQ
  - Scheduler: coordinates all services, determines placement of new resources requested
  - Compute Worker: hosts VMs, controls hypervisor and VMs when receives cmd on Msg Queue
  - Volume: manages persistent storage

Compute ("Nova")

- nova-api accepts and responds to end-user compute API calls.
- Supports OpenStack Compute API, Amazon's EC2 API, and specific drivers for other cloud computing providers.
- Nova supports the extensible architecture, including the orchestration subsystems.
Nova Compute

- The `nova-compute` process is primarily a worker daemon that creates and terminates virtual machine instances via hypervisor APIs (libvirt for KVM or QEMU, VMware APIs for VMware, etc.).
- The process by which it does so is fairly complex but the topics are simple: accept actions from the queue and then perform a series of system commands (like launching a KVM instance) to carry them out while updating state in the database.

Networking (“Quantum”)

- `quantum-server` accepts API requests and then routes them to the appropriate quantum plugin for action.
- Quantum ships with plugins and agents for:
  - Cisco virtual and physical switches
  - Neteo Switch
  - NEST-OpenFlow products
  - OpenSwitch
  - OpenvSwitch
  - Ryu Network Operating System
  - Midonet
- The common agents are L2 (layer 2), DHCP (dynamic host IP addressing) and the specific plugin for agent.

Dashboard (“Horizon”)

- Django application that users can access in their web browser
- Communicates with each OpenStack service through their API (and sometimes their admin API)
- Server/horizon, login dashboard
- Administrate resources via Graphic Interface
LBaaS feature in Neutron

- Load-balancer-as-a-Service (LBaaS) is a feature added to Quantum during grizzly, and an advanced service in Neutron.
- LBaaS allows the ability to provision on demand load balancers pragmatically
- Allows one to create several instances all running the same application and then distribute the load across them in order to scale out an application and provide high availability.

Example: Loadbalancing with Neutron

- Build a multi-tier application with OpenStack, HTTP port 80 to two web servers, database server over port 3306 accessed by web servers;
- use Open vSwitch Neutron plugin;
- Clients send requests, which to be loadbalanced between the two web servers;

Loadbalancing with OVS Neutron plugin

The end