**OpenStack’s Programs**

**Notes taken from LinuxAcademy.com**

Here is the list of the different components that make up OpenStack:

Integrated programs:

* Compute (Nova)
* Networking (Neutron)
* Object Storage (Swift)
* Block Storage (Cinder) STORAGE SERVICES
* Image Service (Glance)
* Identity Service (Keystone)
* Dashboard (Horizon)
* Telemetry (Ceilometer)
* Ochestration (Heat) SHARED SERVICES
* Database (Trove)
* Data Processing (Sahara)

Under development (Incubator projects):

* Bare Metal (Ironic)
* Queue Service (Zaqar)
* Shared file system (Manila)
* DNS Service (Designate)
* Key Management (Barbican)

**OpenStack Nova**

* OpenStack Compute (Nova) is a cloud computing fabric controller
* Architecture is designed to scale horizontally on standard hardware with no proprietary hardware or software requirements
* 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 configurations
* KVM and XenServer are available choices for hypervisor technology, together with Hyper-V and Linux container technology such as LXC

**OpenStack Swift**

* OpenStack Object Storage (Swift) is a scalable and redundant storage system
* Objects and files are written to multiple disk drives spread throughout servers in a data center
* Storage clusters scale horizontally simply by adding new servers
* If a server in the cluster or a hard drive was to fail, OpenStack replicates its contents from other active nodes to new locations in the cluster
* Inexpensive commodity hard drives and servers can be used
* Not a traditional file system, it’sa distributed storage system for static data such as virtual machine images, photo storage, email storage, backups and archives
* 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

**OpenStack Cinder**

* OpenStack Block Storage (Cinder) provides persistent block level storage devices for use with OpenStack compute instances
* The block storage system manages the creation, attaching, and detaching of the block devices to servers
* Block storage volumes are fully integrated into OpenStack Compute and the Dashboard allowing for cloud users to manage their own storage needs
* In addition to using simple Linux server storage, it has unified storage support for numerous storage playforms including Ceph, NetApp, Nexenta, SolidFire, and Zadara
  + Yahoo uses NetApp for their mail and Flickr filers. These servers have literally hundreds of 1-2TB hard drives
  + How NetApp works – The long version -<https://www.youtube.com/watch?v=fkQPnXVRMKU>
* Block storage is appropriate for performance sensitive scenarios such as database storage, expandable file systems, or providing a server with access to raw block level storage
* Snapshot management provides powerful functionality for backing up data stored on block storage volumes
* Snapshots can be restored or used to create a new block storage volume

**OpenStack Neutron**

* OpenStack Networking (Neutron) is a pluggable, scalable, and API-driven system for managing networks and IP addresses
* Formerly known as Quantum
* OpenStack provides flexible networking models to suit the needs of different applications or user groups
* Standard models include flat networks or VLANs for separation of servers and traffic
* OpenStack Networking manages IP addresses, allowing for dedicated static Ips or DHCP
* Floating Ips allow traffic to be dynamically re-routed to any of your compute resources, which allows you to redirect traffic during maintenance or in the case of failure
* Users can create their own networks, control traffic, and connect servers and devices to one or more networks

**OpenStack Horizon**

* OpenStack Dashboard (Horizon) provides administrators and users a graphical interface to access, provision, and automate cloud-based resources
* Allows for third party products and services, such as billing, monitoring, and additional management tools
* The dashboard is just one way to interact with OpenStack resources
* Developers can automate access or build tools to manage their resources using the native OpenStack API or the EC2 compatibility API

**OpenStack Keystone**

* OpenStack Identity (Keystone) provides a central directory of users mapped to the OpenStack services they can access
* The OpenStack Identity service enables administrators to:
  + Configure centralized policies across users and systems
  + Create users and tenants and define permissions for compute, storage, and networking resources by using role-based access control (RBAC) features
  + Integrate with an existing directory, like LDAP, to provide a single source of authentication across the enterprise
* The OpenStack Identity Service enables users to:
  + List the services to which they have access
  + Make API requests
  + Log into the web dashboard to create resources owned by their account

**OpenStack Glance**

* OpenStack Image Service (Glance) provides discovery, registration, and delivery services for disk and server images
* The Image Service API provides a standard REST interface for querying information about disk images and lets clients stream the images to new servers
  + REST - Unlike SOAP-based web services, there is no "official" standard for **RESTful** web**APIs**. This is because **REST** is an architectural style, while SOAP is a protocol. Even though **REST** is not a standard per se, most **RESTful** implementations make use of standards such as HTTP, URL, JSON, and XML.
* Capability of the Image Service:
  + Admins can create base templates from which their users can start new compute instances
  + Users can choose from available images or create their own from existing servers
  + Snapshots can also be stored in the Image Service so that virtual mchines can be backed up quickly
* A multi-format image registry, the service allows uploads of private and public images in a variety of formats such as:
  + Raw
  + Machine (kernel/ramdisk outside of image, also known s AMI)
  + VHD (Hyper-V)
  + VDI (VirtualBox)
  + Qcow2 (Qemu/KVM)
  + OVF (Vmware, others)

**OpenStack Incubator Projects**

* As the OpenStack community grows, new projects will be added and some existing projects will want to become official OpenStack core projects

Purpose of Incubation

* The incubation process serves several purposes with the overall goal of ensuring that core OpenStack projects meet a high standard of usefulness and equality
  + Sustainable development process
  + Growing the core development team
  + Establishing an initial user base
  + Maturing the software to an acceptable level of stability
  + Integration with OpenStack processes around testing, releases, and community management

Each of the OpenStack programs:

* Is also a “Core” and top level component
* Has an elected “Project Technical Lead”
* Has separate developers and design teams
* Has a well defined public API
  + Except Horizon, which is the Web based GUI, all other projects have a RESTful (JSON/HTTP) API
* Has a seaparate database and isolated persistent layer