CLOUD COMPUTING

OPENSTACK - Overview

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
IIT KHARAGPUR

What is OpenStack?

- OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed and provisioned through APIs with common authentication mechanisms.

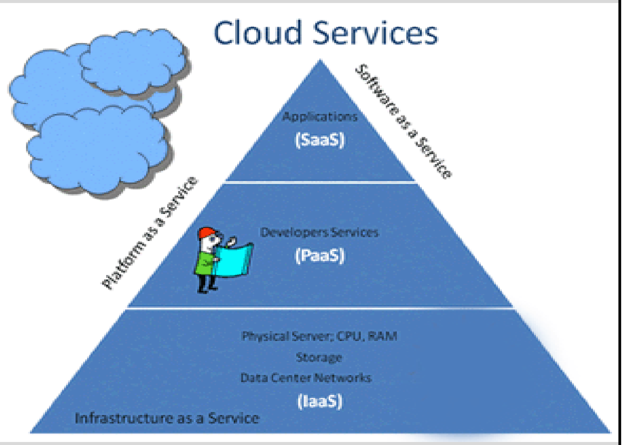
- A dashboard is also available, giving administrators control while empowering the users to provision resources through a web interface.

- Beyond standard infrastructure-as-a-service functionality, additional components provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.

Source: OpenStack, http://www.doc.openstack.org

OpenStack Capability

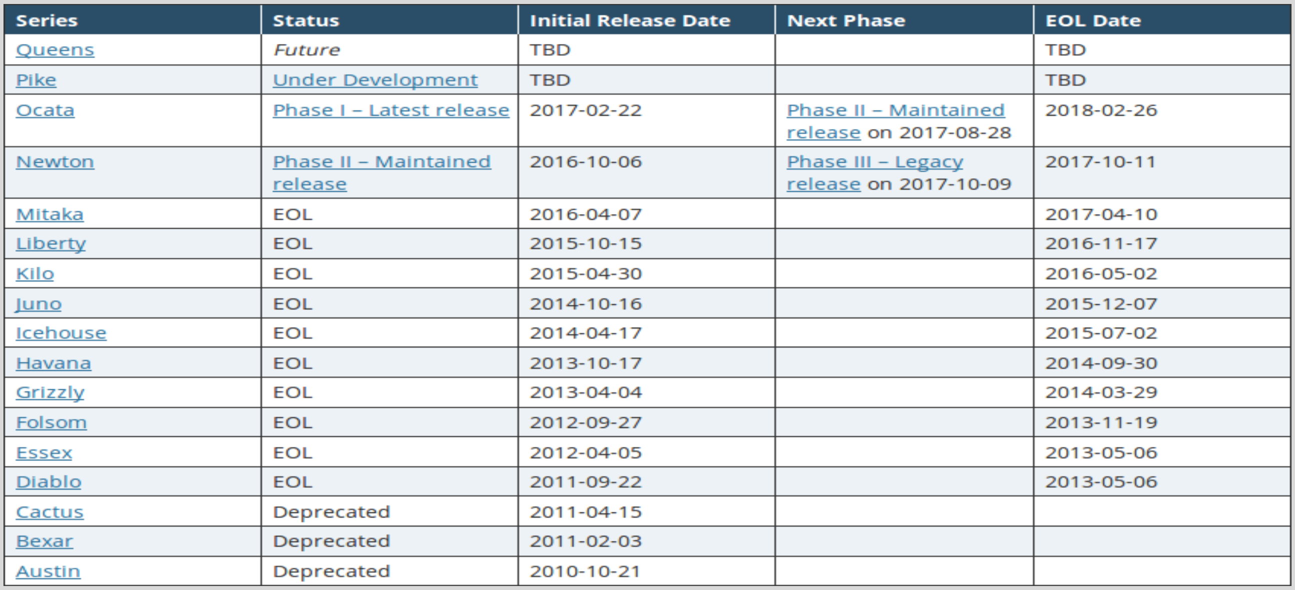
- Software as Service (SaaS)
  - Browser or Thin Client access
- Platform as Service (PaaS)
  - On top of IaaS e.g. Cloud Foundry
- Infrastructure as Service (IaaS)
  - Provision Compute, Network, Storage



OpenStack Capability

- Virtual Machine (VMs) on demand
  - Provisioning
  - Snapshotting
- Network
- Storage for VMs and arbitrary files
- Multi-tenancy
  - Quotas for different project, users
  - User can be associated with multiple projects

OpenStack History



Series	Status	Initial Release Date	Next Phase	EOL Date
Queens	*Future*	TBD		TBD
Pike	Under Development	TBD		TBD
Ocata	Phase I – Latest release	2017-02-22	Phase II – Maintained release on 2017-08-28	2018-02-26
Newton	Phase II – Maintained release	2016-10-06	Phase III – Legacy release on 2017-10-09	2017-10-11
Mitaka	EOL	2016-04-07		2017-04-10
Liberty	EOL	2015-10-15		2016-11-17
Kilo	EOL	2015-04-30		2016-05-02
Juno	EOL	2014-10-16		2015-12-07
Icehouse	EOL	2014-04-17		2015-07-02
Havana	EOL	2013-10-17		2014-09-30
Grizzly	EOL	2013-04-04		2014-03-29
Folsom	EOL	2012-09-27		2013-11-19
Essex	EOL	2012-04-05		2013-05-06
Diablo	EOL	2011-09-22		2013-05-06
Cactus	Deprecated	2011-04-15		
Bexar	Deprecated	2011-02-03		
Austin	Deprecated	2010-10-21		

**Started as a collaboration between NASA and Rackspace*

OpenStack Major Components

- Service - Compute
- Project - Nova

Manages the lifecycle of compute instances in an OpenStack environment. Responsibilities include spawning, scheduling and decommissioning of virtual machines on demand.

OpenStack Major Components

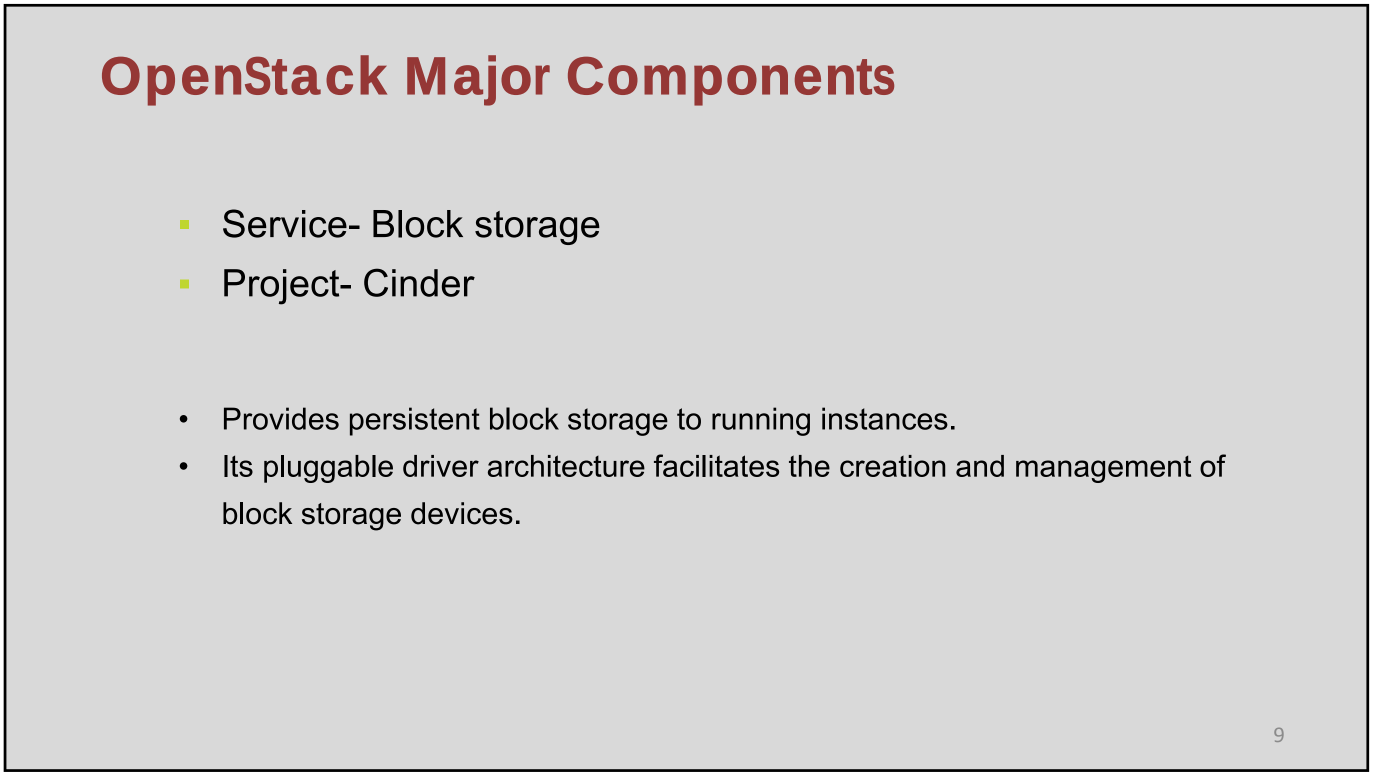
- Service - Networking
- Project - Neutron

- Enables *Network-Connectivity-as-a-Service* for other OpenStack services, such as OpenStack Compute.
- Provides an API for users to define networks and the attachments into them.
- Has a pluggable architecture that supports many popular networking vendors and technologies.

OpenStack Major Components

- Service - Object storage
- Project - Swift

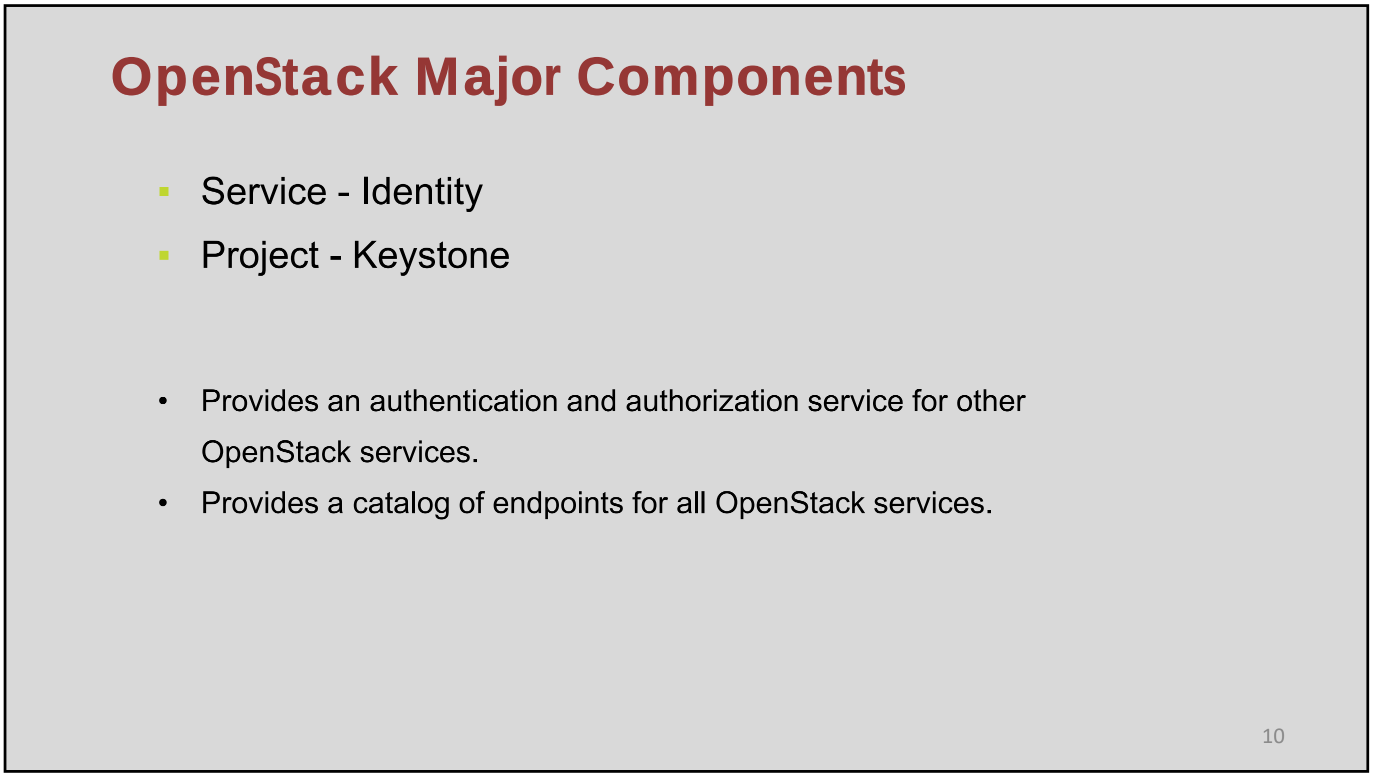
- Stores and retrieves arbitrary unstructured data objects via a RESTFul, HTTP based API.
- It is highly fault tolerant with its data replication and scale-out architecture. Its implementation is not like a file server with mountable directories.
- In this case, it writes objects and files to multiple drives, ensuring the data is replicated across a server cluster.



OpenStack Major Components

- Service- Block storage
- Project- Cinder

- Provides persistent block storage to running instances.
- Its pluggable driver architecture facilitates the creation and management of block storage devices.



OpenStack Major Components

- Service - Identity
- Project - Keystone

- Provides an authentication and authorization service for other OpenStack services.
- Provides a catalog of endpoints for all OpenStack services.

OpenStack Major Components

- Service - Image service
- Project - Glance

- Stores and retrieves virtual machine disk images.
- OpenStack Compute makes use of this during instance provisioning.

OpenStack Major Components

- Service - Telemetry
- Project - Ceilometer

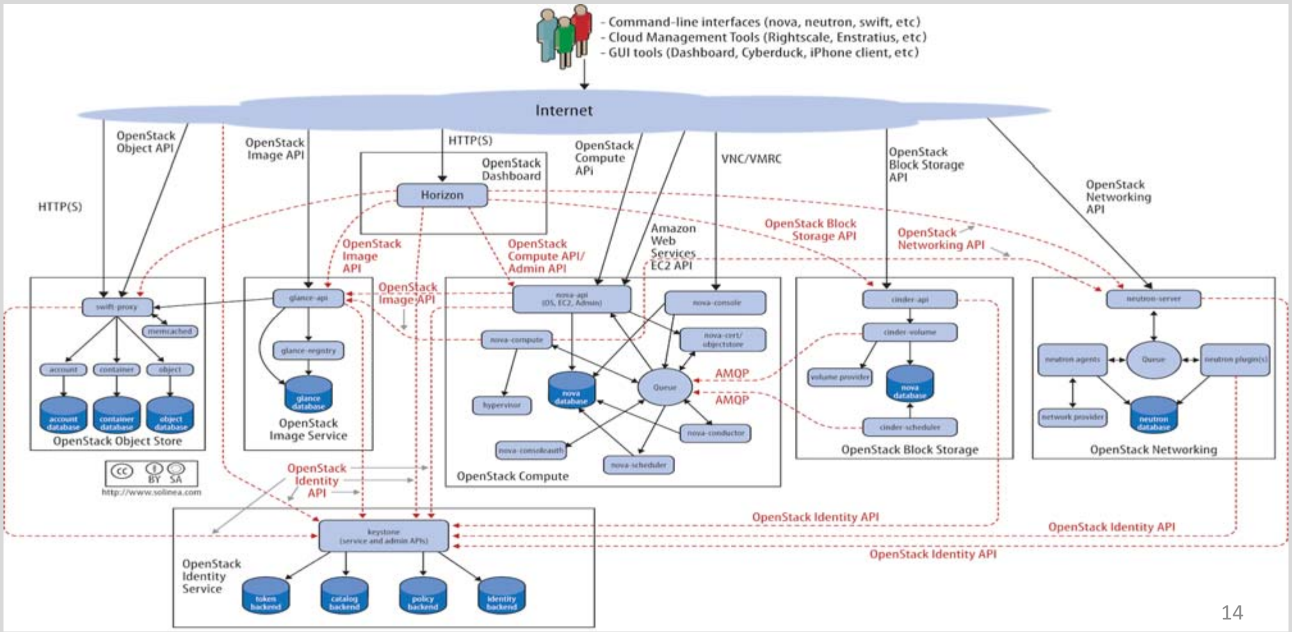
- Monitors and meters the OpenStack cloud for billing, benchmarking, scalability, and statistical purposes.

OpenStack Major Components

- Service - Dashboard
- Project - Horizon

- Provides a web-based self-service portal to interact with underlying OpenStack services, such as launching an instance, assigning IP addresses and configuring access controls.

Architecture of Openstack



Openstack Work Flow

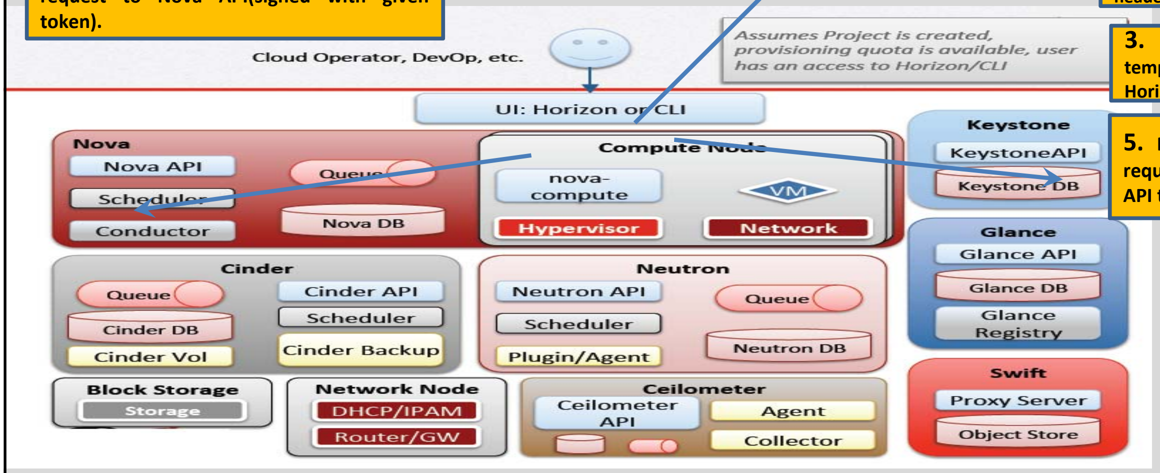
1. User logs in to UI Specifies VM params: name,flavor,keys,etc. and hits "Create" button

2. Horizon sends HTTP request to Keystone. Auth info is specified in HTTP headers.

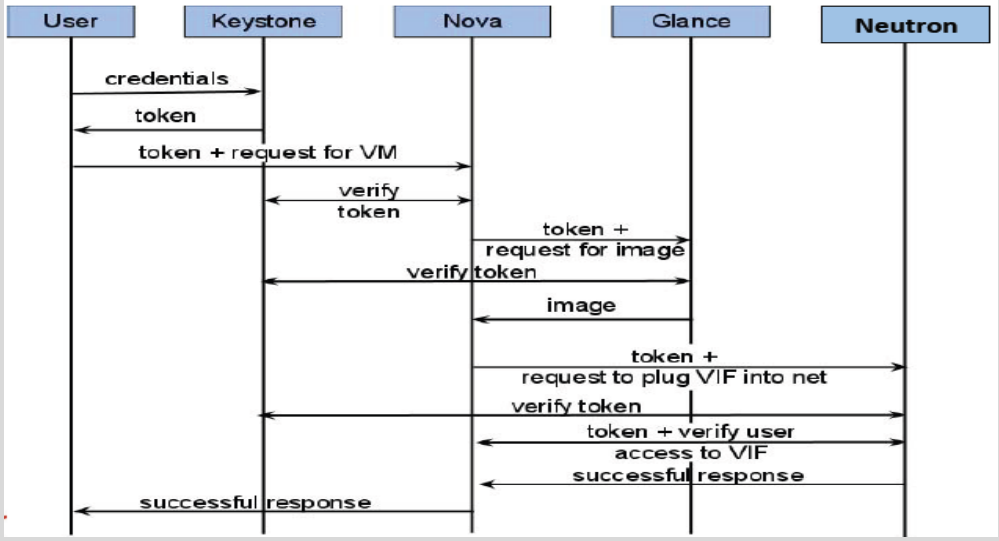
3. Keystone sends temporary token back to Horizon via HTTP.

4. Keystone sends temporary token back to Horizon via HTTP. Horizon sends POST request to Nova API(signed with given token).

5. Nova API sends HTTP request to validate API token to Keystone.



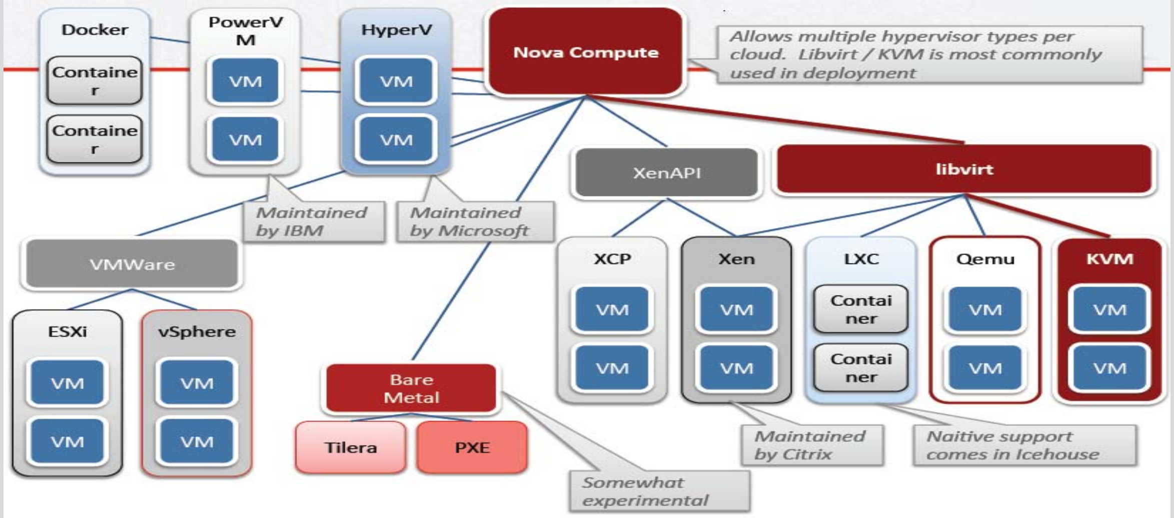
Auth Token Usage



Provisioning Flow

- Nova API makes rpc.cast to Scheduler. It publishes a short message to scheduler queue with VM info.
- Scheduler picks up the message from MQ.
- Scheduler fetches information about the whole cluster from database, filters, selects compute node and updates DB with its ID
- Scheduler publishes message to the compute queue (based on host ID) to trigger VM provisioning
- Nova Compute gets message from MQ
- Nova Compute makes rpc.call to Nova Conductor for information on VM from DB
- Nova Compute makes a call to Neutron API to provision network for the instance
- Neutron configures IP, gateway, DNS name, L2 connectivity etc.
- It is assumed a volume is already created. Nova Compute contacts Cinder to get volume data. Can also attach volumes after VM is built.

Nova Compute Driver



Allows multiple hypervisor types per cloud. Libvirt / KVM is most commonly used in deployment

Maintained by IBM

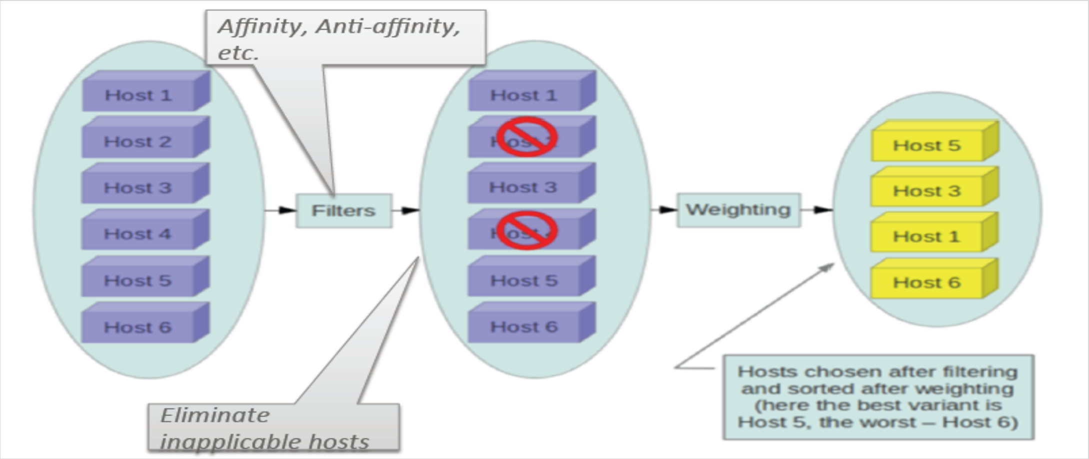
Maintained by Microsoft

Somewhat experimental

Maintained by Citrix

Naitive support comes in Icehouse

Nova scheduler filtering

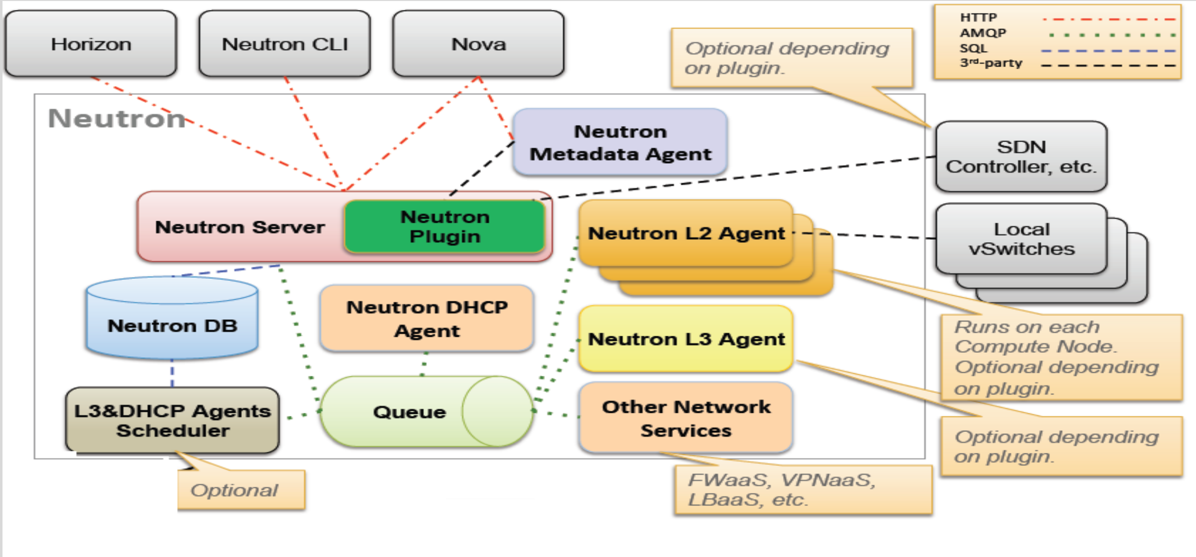


Affinity, Anti-affinity, etc.

Eliminate inapplicable hosts

Hosts chosen after filtering and sorted after weighting (here the best variant is Host 5, the worst – Host 6)

Neutron Architecture



Optional depending on plugin.

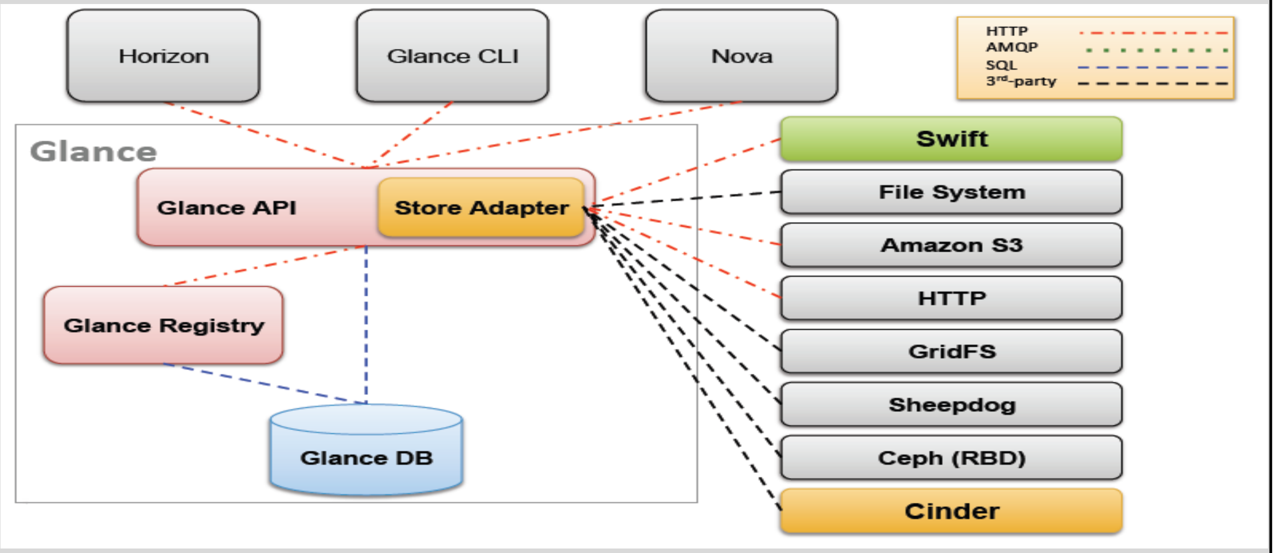
Runs on each Compute Node. Optional depending on plugin.

Optional depending on plugin.

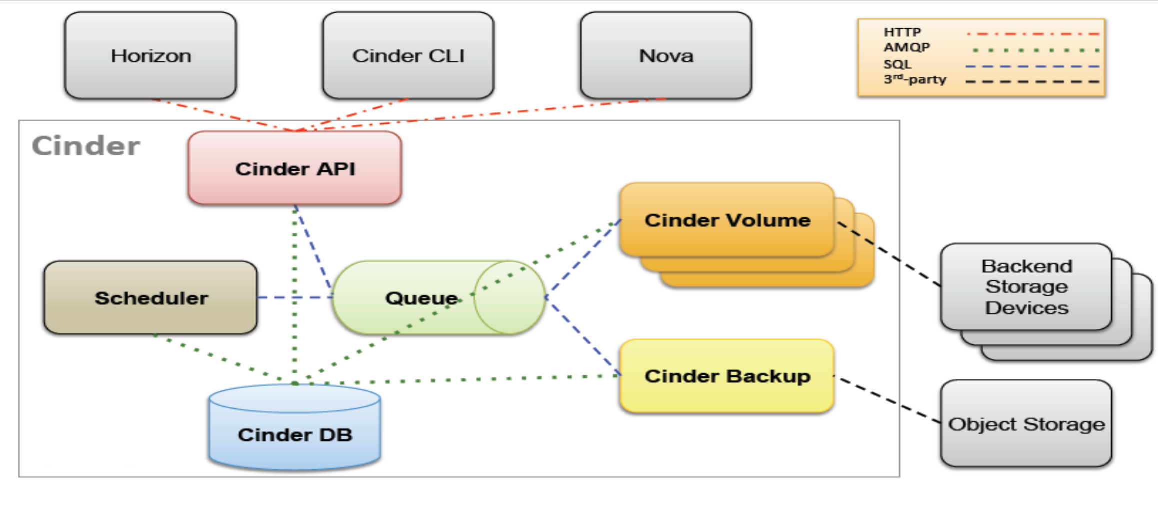
FWaaS, VPNaaS, LBaaS, etc.

Optional

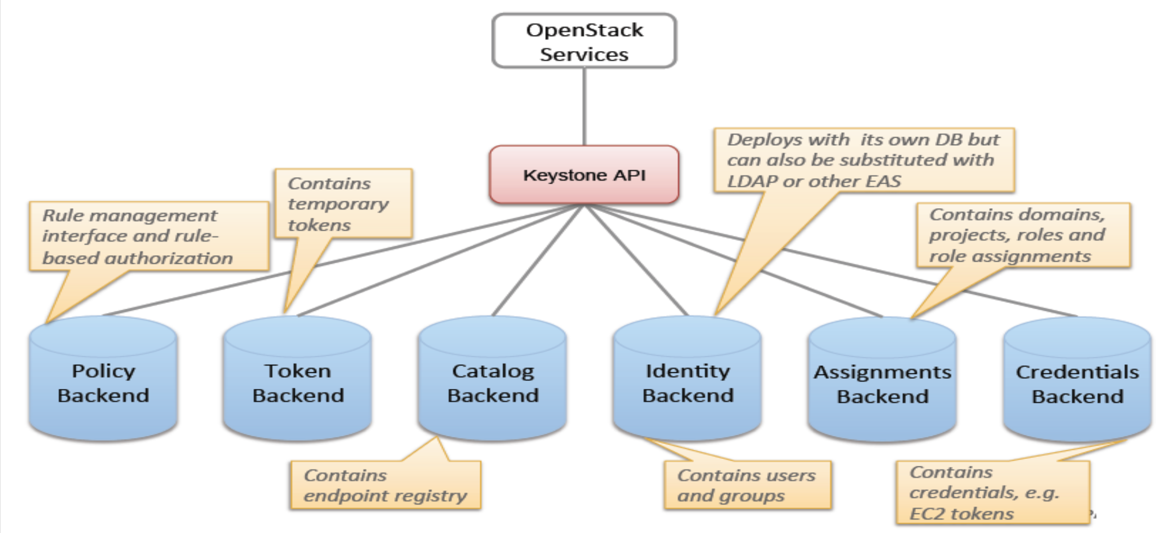
Glance Architecture



Cinder Architecture



Keystone Architecture



Rule management interface and rule-based authorization

Contains temporary tokens

Deploys with its own DB but can also be substituted with LDAP or other EAS

Contains domains, projects, roles and role assignments

Contains endpoint registry

Contains users and groups

Contains credentials, e.g. EC2 tokens

OpenStack Storage Concepts

- **Ephemeral storage:**
    - Persists until VM is terminated
    - Accessible from within VM as local file system
    - Used to run operating system and/or scratch space
    - Managed by Nova
- **Block storage**:
    - Persists until specifically deleted by user
    - Accessible from within VM as a block device (e.g. /dev/vdc)
    - Used to add additional persistent storage to VM and/or run operating system
    - Managed by Cinder
- **Object storage:**
    - Persists until specifically deleted by user
    - Accessible from anywhere
    - Used to add store files, including VM images
    - Managed by Swift

Summary

- Users log into Horizon and initiates VM creation
- Keystone authorizes
- Nova initiates provisioning and saves state to DB
- Nova Scheduler finds appropriate host
- Neutron configures networking
- Cinder provides block device
- Image URI is looked up through Glance
- Image is retrieved via Swift
- VM is rendered by Hypervisor

Thank You!