#SHAREorg

SHARE
Technology · Connections · Results

What is OpenStack ?

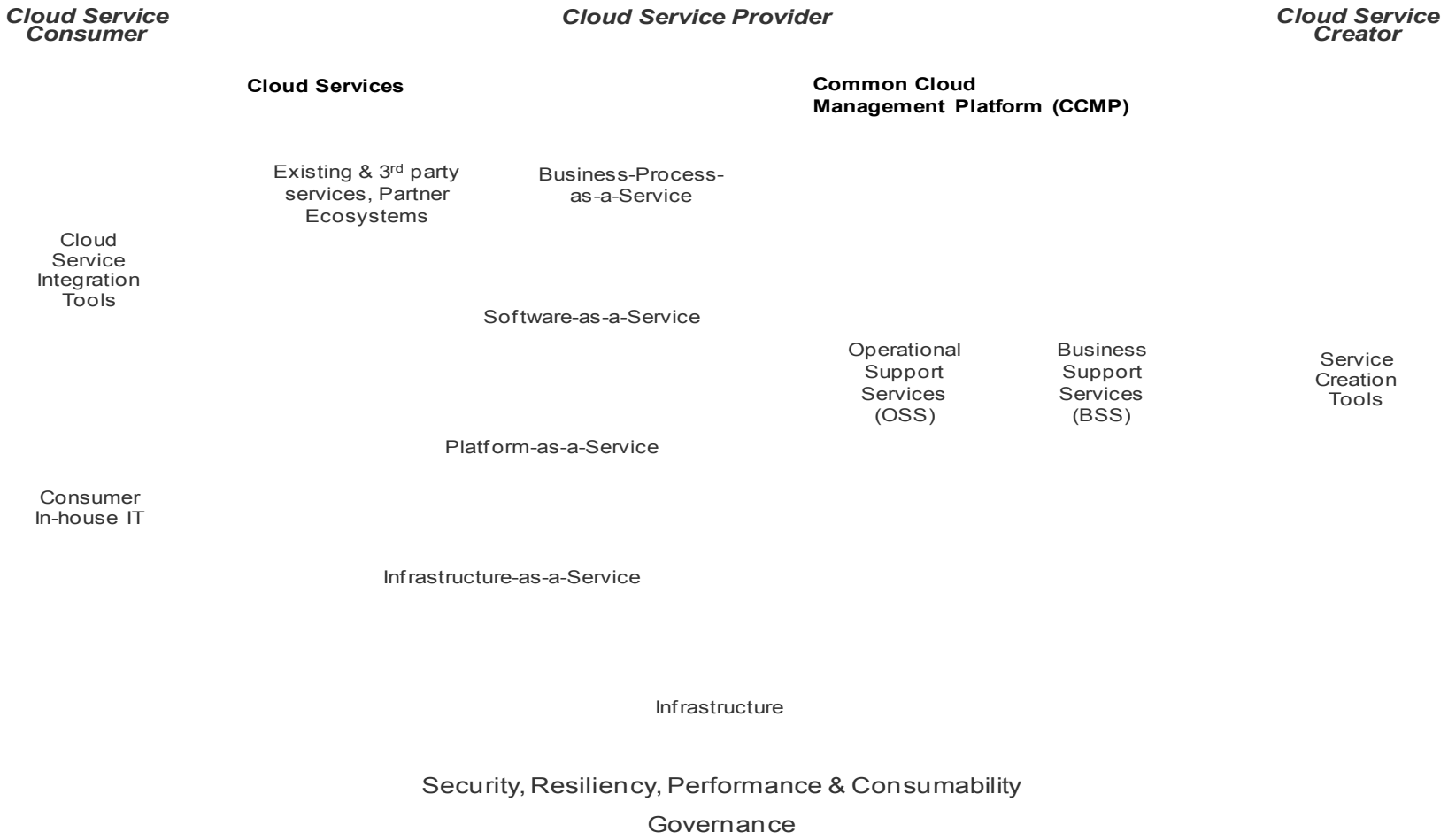
Mike Buzzetti
buzzetti@us.ibm.com
IBM

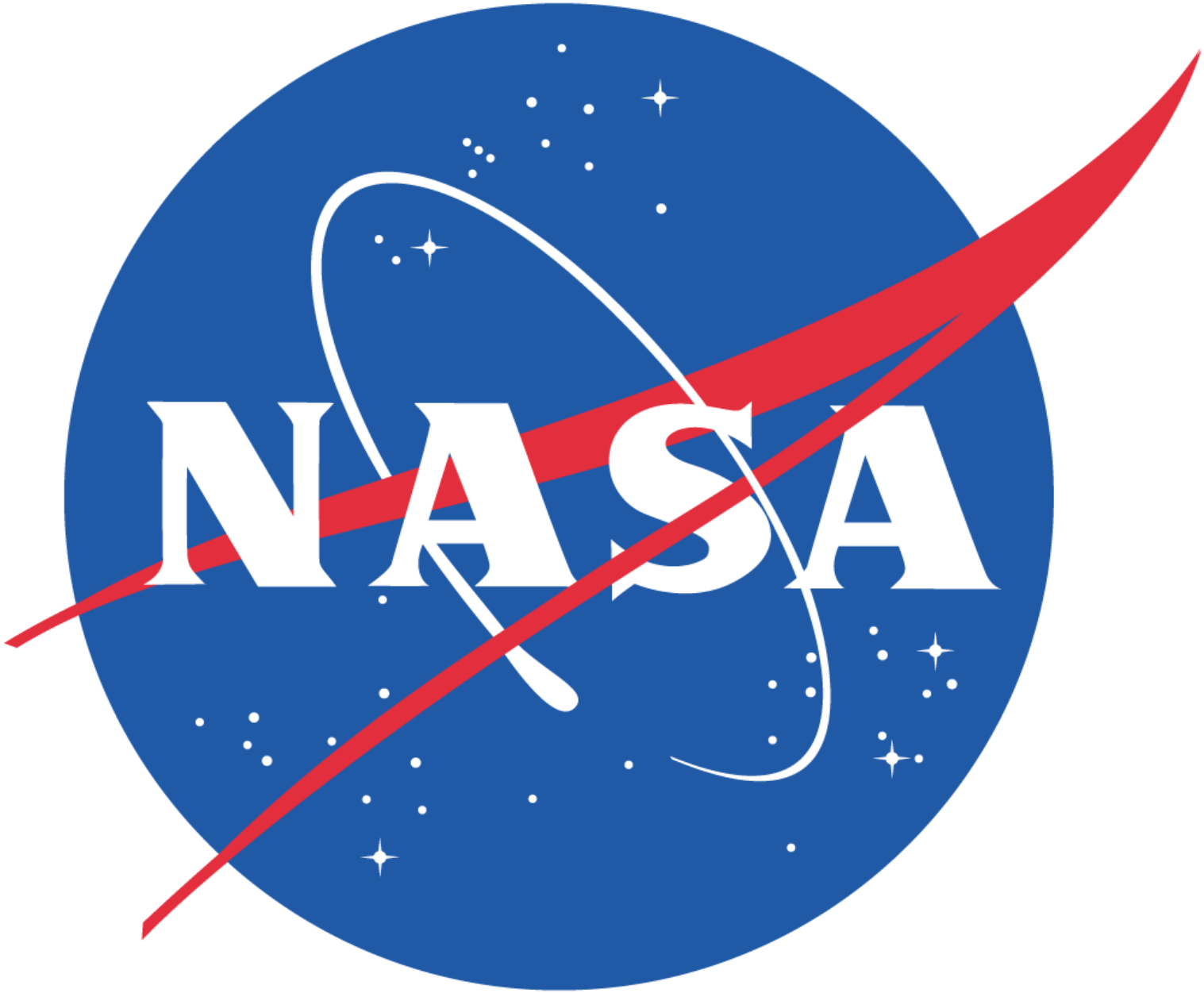
14221

SHARE in Boston

I am here to help **buzzetti@us.ibm.com**

IBM's Reference Architecture for Cloud




NASA

RackSpace

Community

More than 6000 people and 100 companies

Active online community through mailing lists, IRC, wiki

Bi-yearly design summits

Companies need to donate money AND people that ACTIVELY contribute

Platinum Members

AT&T

Canonical

HP

IBM

Nebula

Rackspace

Red Hat, Inc.

SUSE

and many more …
http://www.OpenStack.org/foundation/com

Relase Names

These codenames are chosen by popular vote using the basic Launchpad poll feature over the ~openstack group. Codenames are cities or counties near where the corresponding OpenStack design summit took place. An exception (called the Waldon exception) is granted to elements of the state flag that sound especially cool.

Austin: The first design summit took place in Austin, TX
Bexar: The second design summit took place in San Antonio, TX
Cactus: Cactus is a city in Texas
Diablo: Diablo is a city in the bay area near Santa Clara, CA
Essex: Essex is a city near Boston, MA
Folsom: Folsom is a city near San Francisco, CA
Grizzly: Grizzly is an element of the state flag of California
design summit takes place in San Diego, CA
Havana: Havana is an unincorporated community in Oregon
design summit takes place in Oregon
Ichang?: Design Summet to take place in Hong-Kong

Commitment

Top 10 Contributors to OpenStack by Release

Essex April 2012			Folsom Sept 2012			Grizzly April 2013		
	Commits	Authors		Commits	Authors		Commits	Authors
Rackspace	2,275	52	Rackspace	973	49	Red Hat	1,854	38
Red Hat	507	12	Red Hat	861	15	Rackspace	944	58
Nebula	431	8	Nebula	464	9	IBM	895	41
HP	261	21	HP	179	25	HP	401	49
Canonical	118	7	Isi	144	3	Nebula	286	10
Nicira	105	3	Cloudscaling	140	4	Intel	243	13
Citrix	87	9	IBM	139	16	eNovance	238	7
Enovance	54	2	Sina	135	18	VMware	186	4
Cloudscaling	52	1	Canonical	108	5	Cloudscaling	175	3
Isi	40	2	Inktank	90	2	Dreamhost	171	7

Source: bitergia.com

OpenStack design tenets focus on delivering Cloud Computing Platform on an available, scalable, and elastic control plane

Basic Design Tenets

1) Scalability and elasticity are our main goals

2) Any feature that limits our main goals must be optional

3) Everything should be asynchronous
- If you can't do something asynchronously, see #2

1) All required components must be horizontally scalable

2) Always use shared nothing architecture (SN) or sharding
- If you can't Share nothing/shard, see #2

1) Distribute everything
- Especially logic. Move logic to where state naturally exists.

1) Accept eventual consistency and use it where it is appropriate.

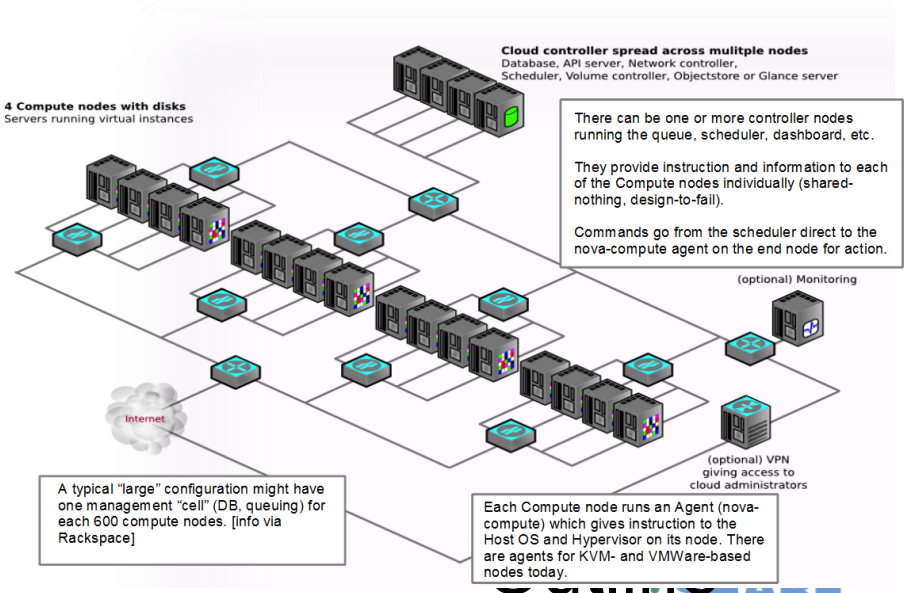
2) Test everything.
- We require tests with submitted code. (We will help you if you need it)

Sources:
http://www.openstack.org/downloads/openstack-compute-datasheet.pdf
http://wiki.openstack.org/BasicDesignTenets

OpenStack Open Source Cloud 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

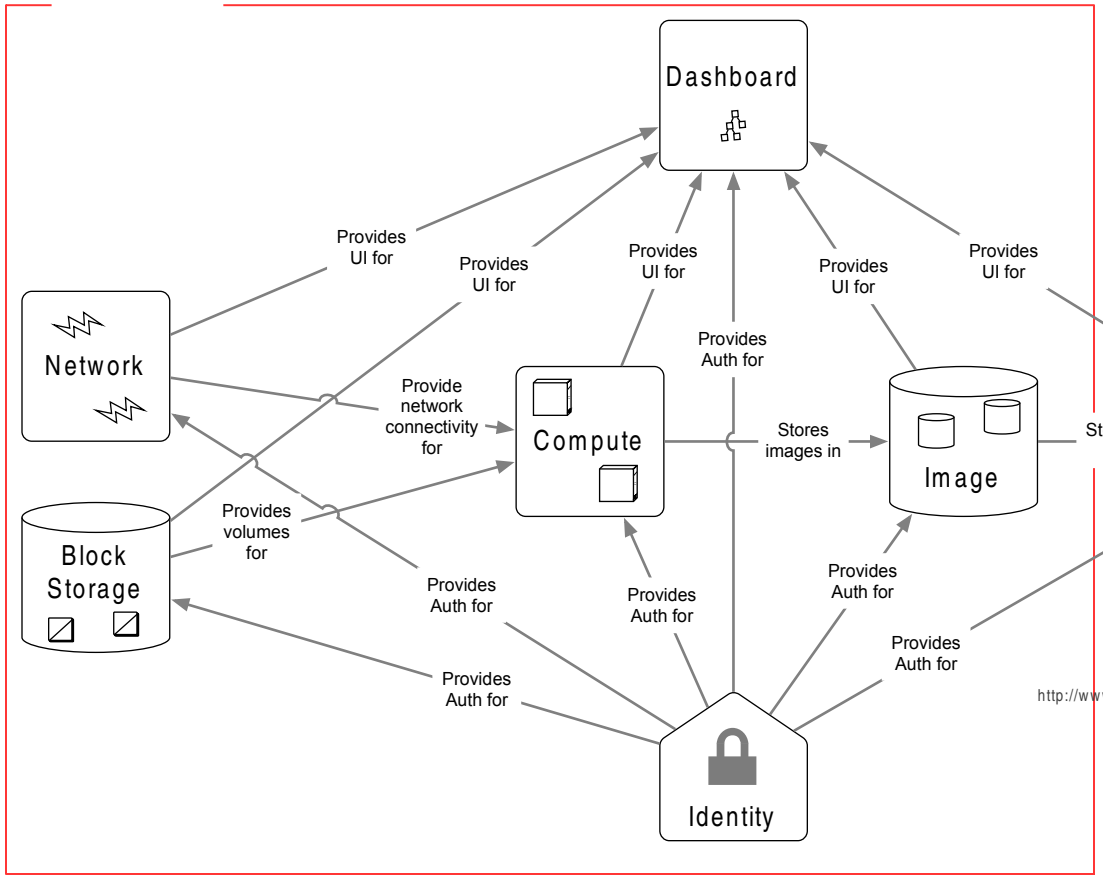
Level



Level

OpenStack is comprised of six core projects delivering an IaaS solution + a project delivering an Object Storage solution

Focus



IaaS

Compute (Nova)

Block Storage (Cinder)

Network (Quantum)
Provision and manage virtual resources

Dashboard (Horizon)
Self-service portal

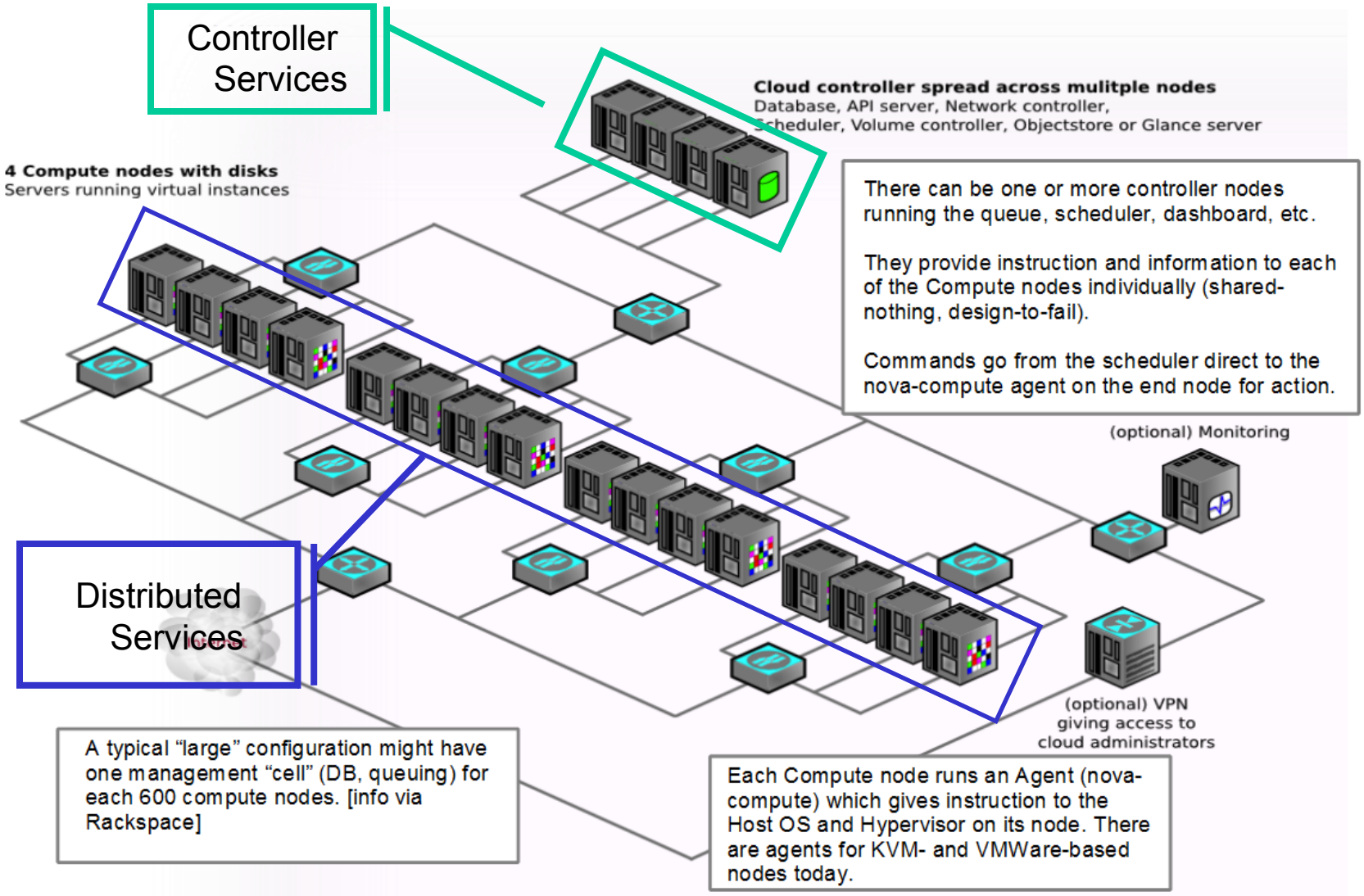
Image (Glance)
Catalog and manage server images

Identity (Keystone)
Unified authentication and authorization

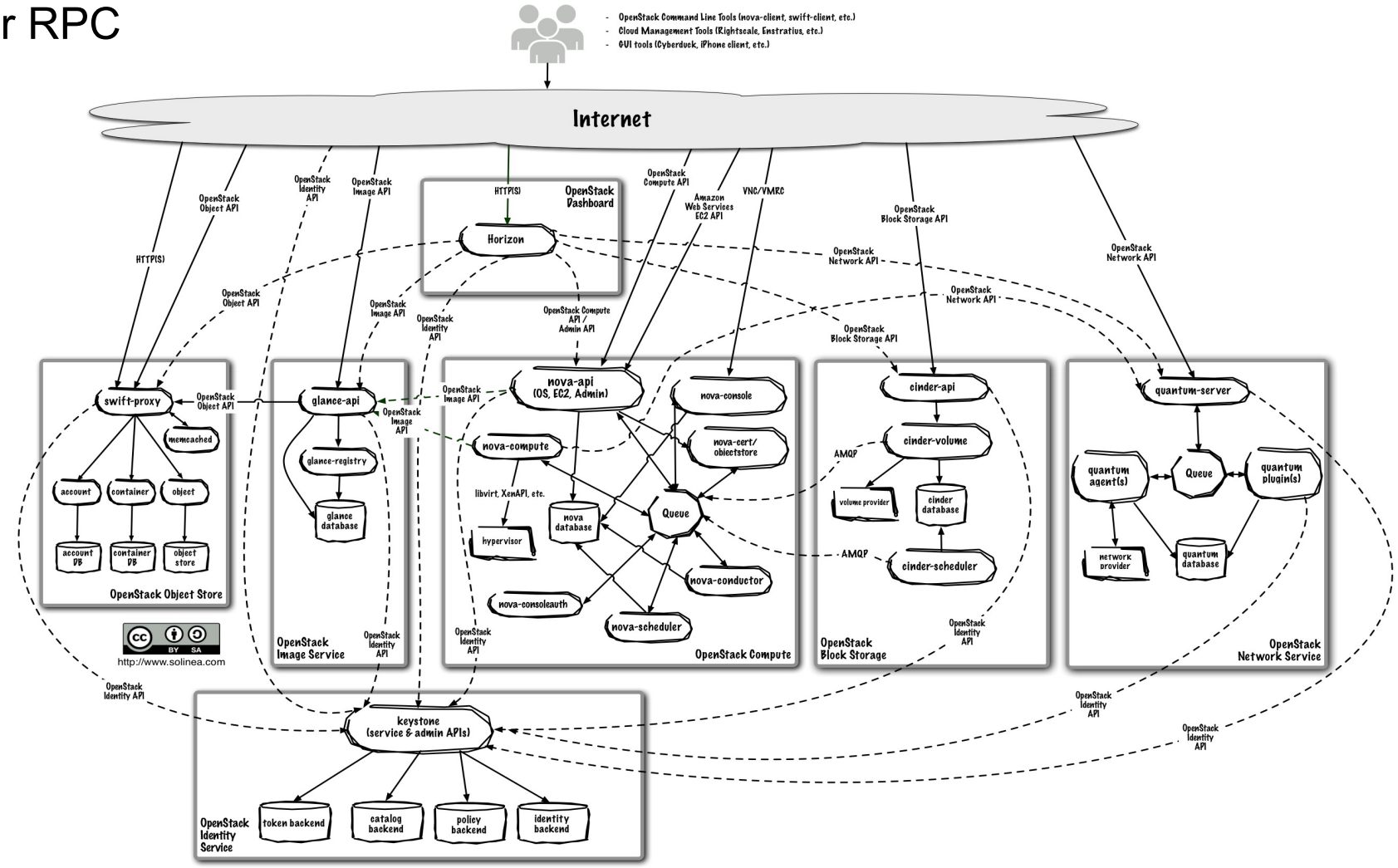
Object Storage (Swift)
petabytes of secure, reliable object storage

mage Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

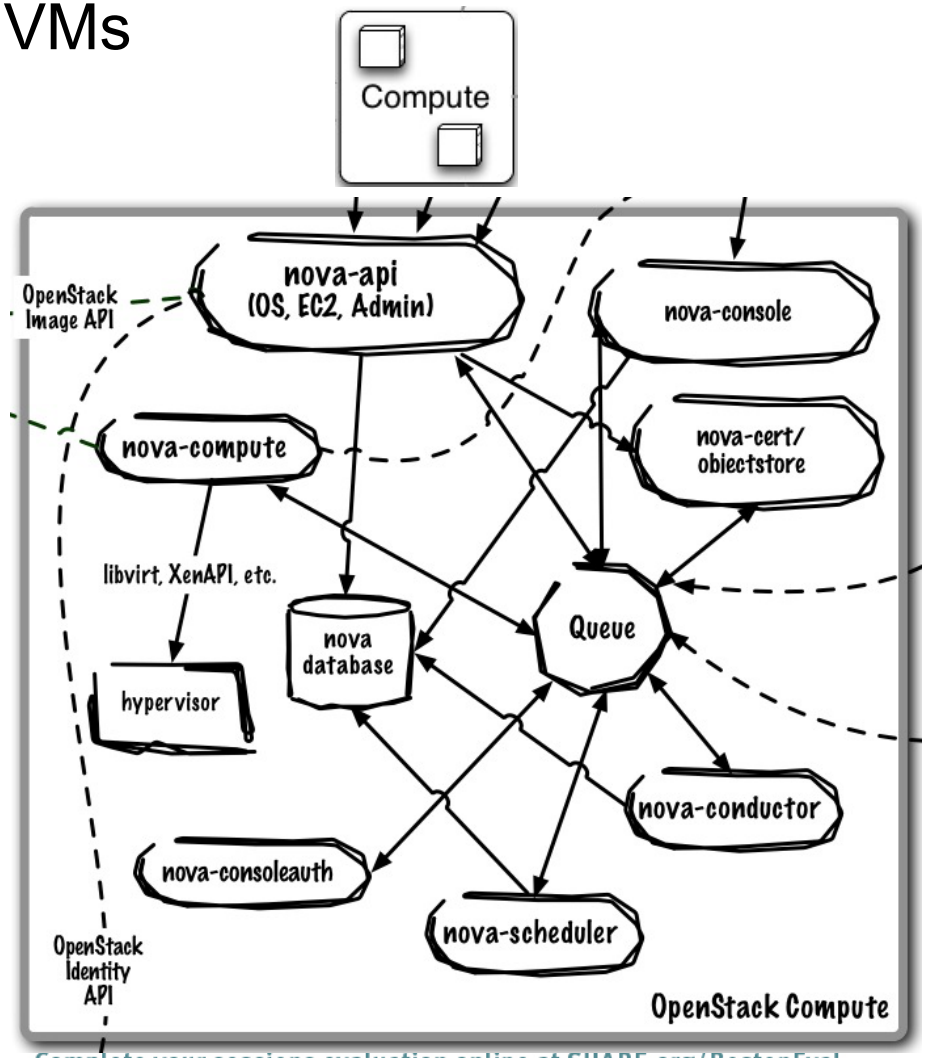
OpenStack services can be categorized into two groups – controller services and distributed services (but all can be scaled-out!)



Deployments consist of projects interfacing over public APIs, with each project composed of multiple services interfacing via private APIs over RPC



Compute (Nova) is a horizontally scalable offering on-demand compute resources by provisioning and managing VMs



Core Use Case:

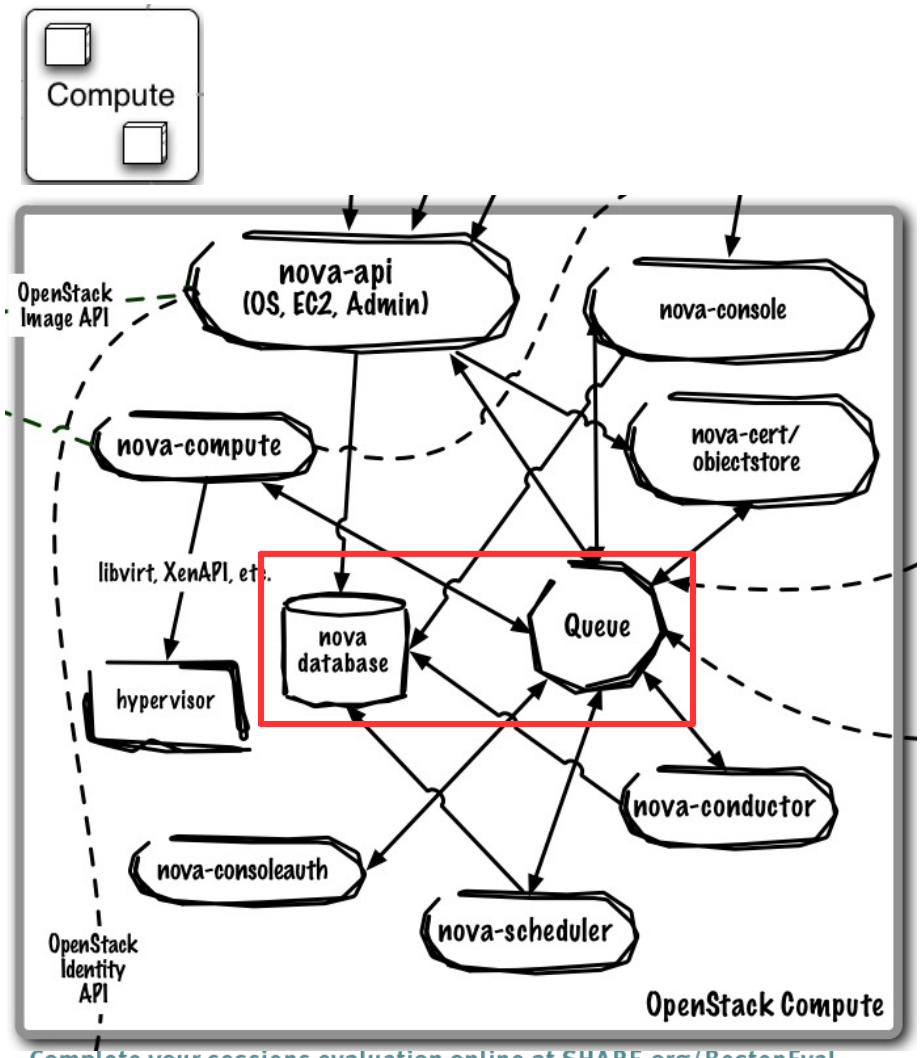
- Provision and manage virtualized compute resources (CPU, memory, disk, network)

Key Capabilities:

- REST-based APIs with rate limiting and authentication
- Manage Local Area Networks (LAN)
- Live migration of guests
- VM management (Instance)
  - Run, reboot, suspend, resize, terminate instances
- Floating IP addresses
- Security Groups
- RBAC with Projects & Quotas
- Manage to KVM, Xen (XenServer, Xen Cloud Platform), LXC, VMware vSphere 4.1+, Hyper-V, Bare Metal, PowerVM (limited)

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

Database and Queue are central to the Nova control plane



Core Use Case:

- Queue provides RPC messaging between services
- Database provides data persistence

Runs As: Controller Service

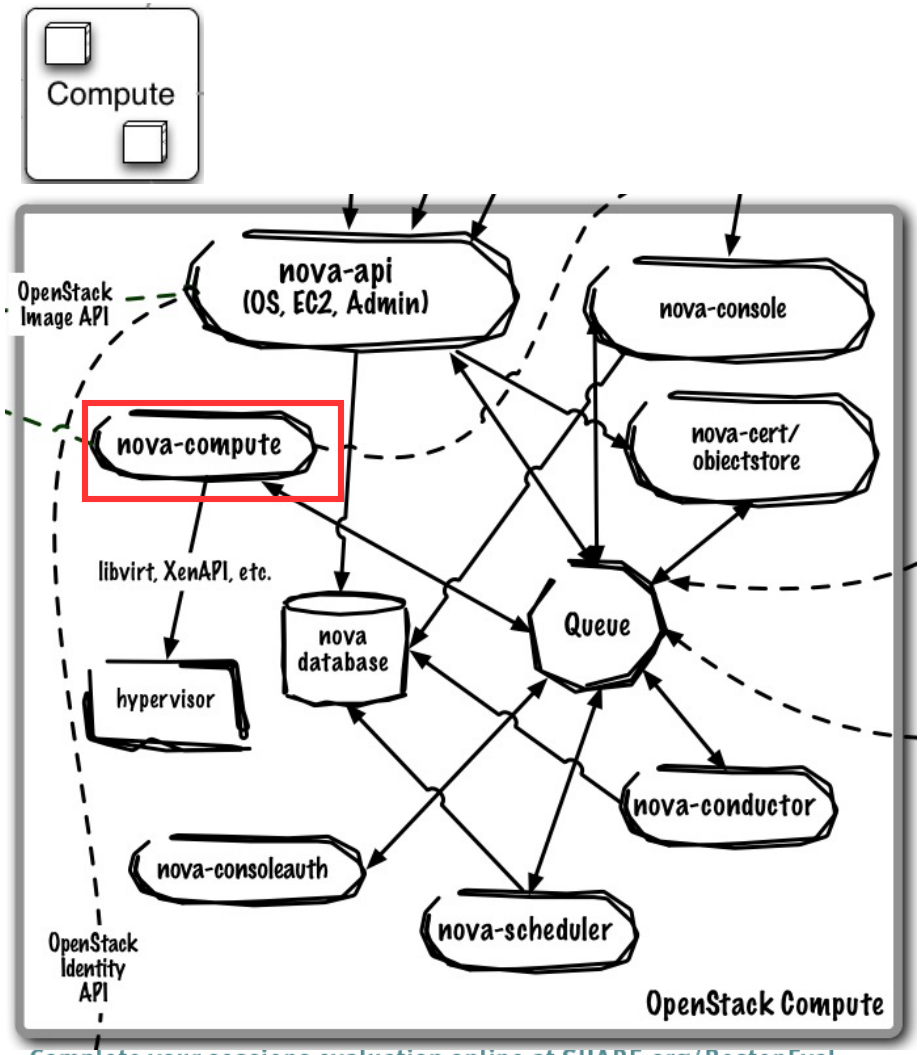
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- ZeroMQ implementation available to decentralize queue

Key Capabilities:

- Community uses RabbitMQ as default queue, MySQL DB (IBM uses Apache Qpid and DB2)
- Single “cell” (1 Queue, 1 Database) typically scales from 500 – 1000 physical machines
  - Cells can be rolled up to support larger deployments
- Communications route through queue
  - API requests are validated and placed on queue
  - Workers listen to queues based on role or role + hostname
  - Responses are dispatched back through queue

nova-compute manages individual hypervisors and compute nodes



Core Use Case:

- Manage all interactions with single hypervisor control point

Runs As: Distributed Service

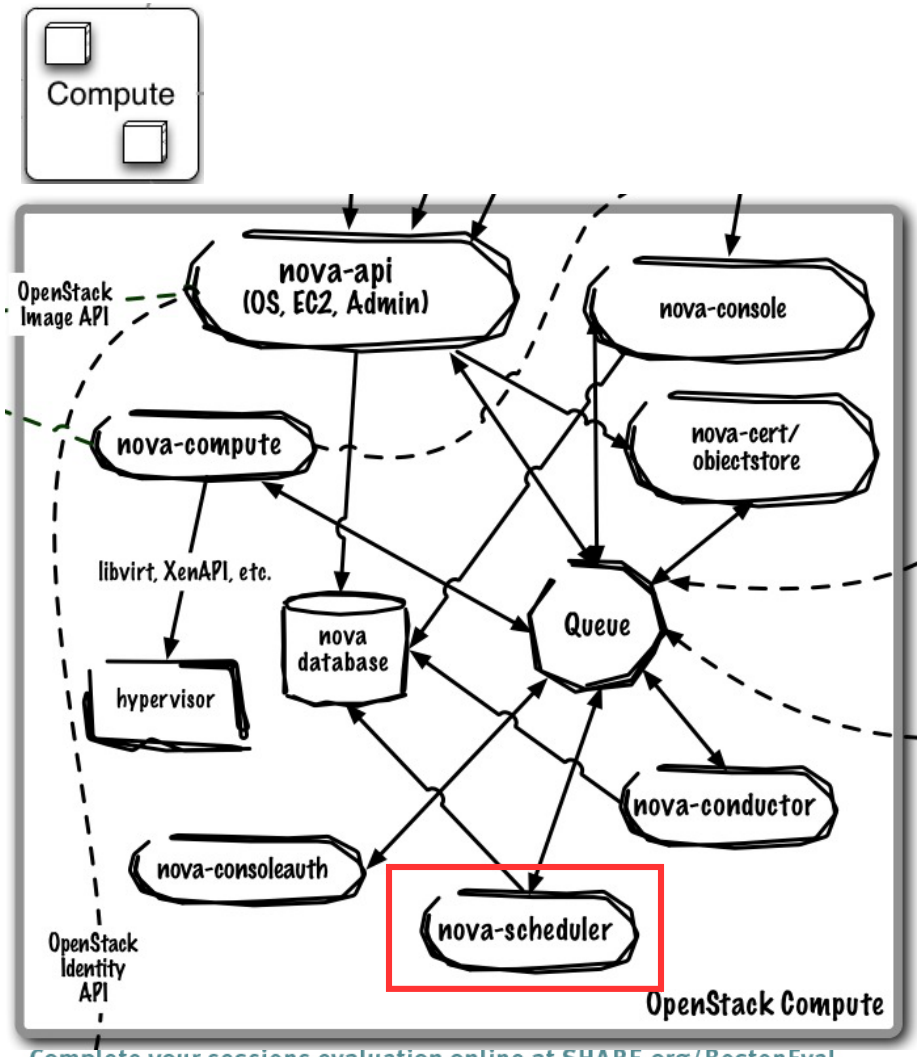
Deployment Considerations:

- Many nova-compute instances exist in the environment to ensure compute provisioning is always available
- Single nova-compute is not HA, manage single hypervisor to minimize failure domain
- No direct database acces is required

Key Capabilities:

- Create and manage virtual machines on hypervisor
- Attach networks and volumes to physical host (iSCSI, FC), expose to guest virtual machines
- Implementation point for security groups defining firewall rules for guest network traffic
- Uses plug-in model to manage to different hypervisors

nova-scheduler allocates virtual resources to compute nodes



Core Use Case:

- Selects compute node to run virtual machine on

Runs As: Controller Service

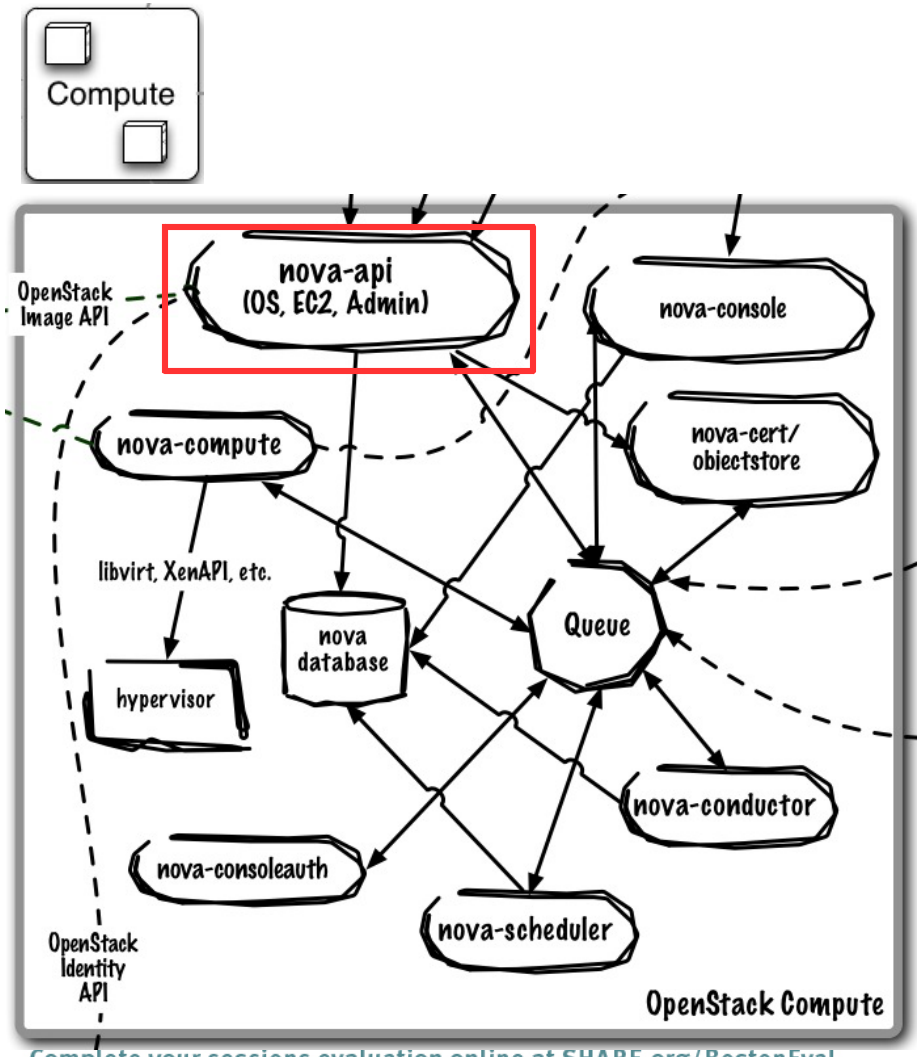
Deployment Considerations:

- Default scheduler is horizontally scalable
- For other schedulers (e.g. Platform EGO), follow their specific best practice

Key Capabilities:

- Default scheduled is allocation-based using a series of filters to reduce set of applicable hosts and uses costing functions to provide weight
- Platform EGO adds utilization-based scheduling to default allocation based

nova-api supports multiple API implementations and is the entry point into the cloud



Core Use Case:

- Accept, validate, authenticate, and distribute incoming REST API requests

Runs As: Controller Service

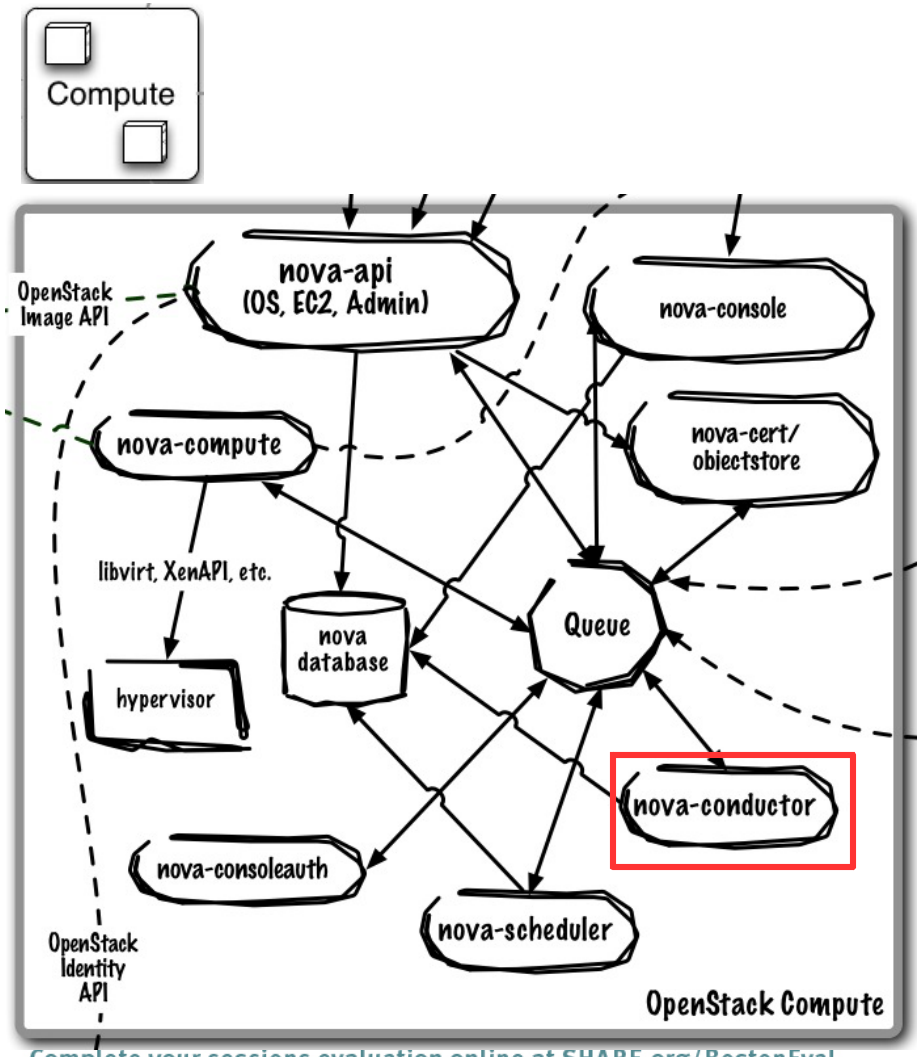
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Compute API
  - EC2 API (subset)
- Robust extensions mechanism to add new capabilities

nova-conductor manages database interactions on behalf of compute nodes



Core Use Case:

- Handles all database requests for nova-compute service

Runs As: Controller Service

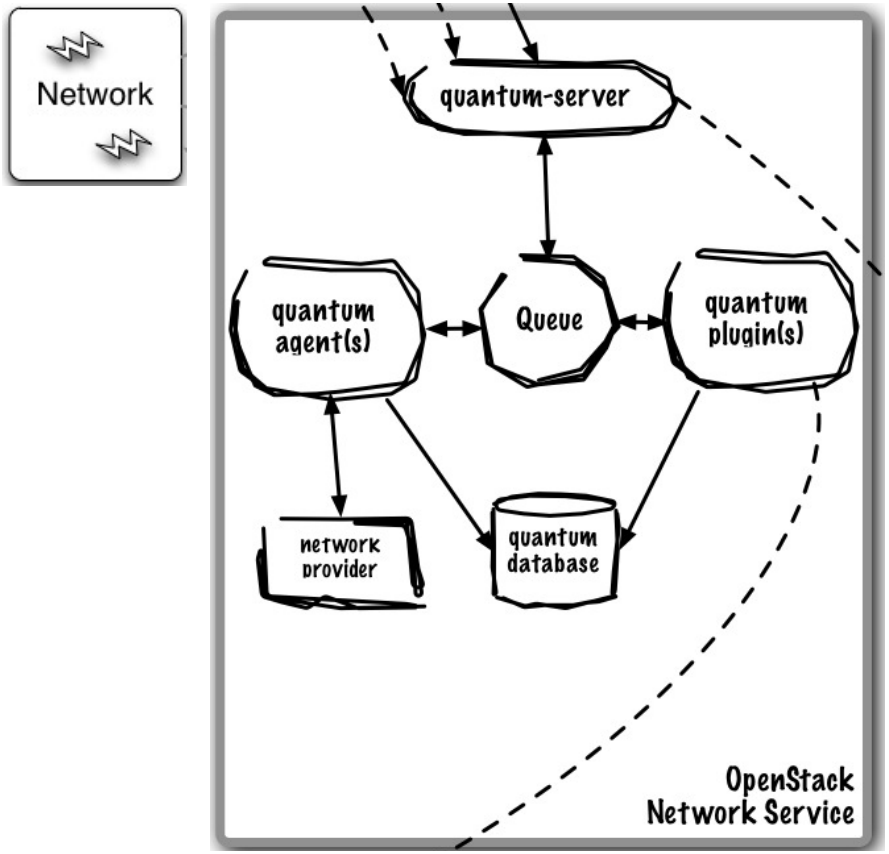
Deployment Considerations:

- Horizontally scalable, start many instances

Key Capabilities:

- Talks directly to database on behalf of compute nodes

Network (Quantum) is a pluggable, scalable and API-driven system for managing networks and IP addresses



Core Use Cases:

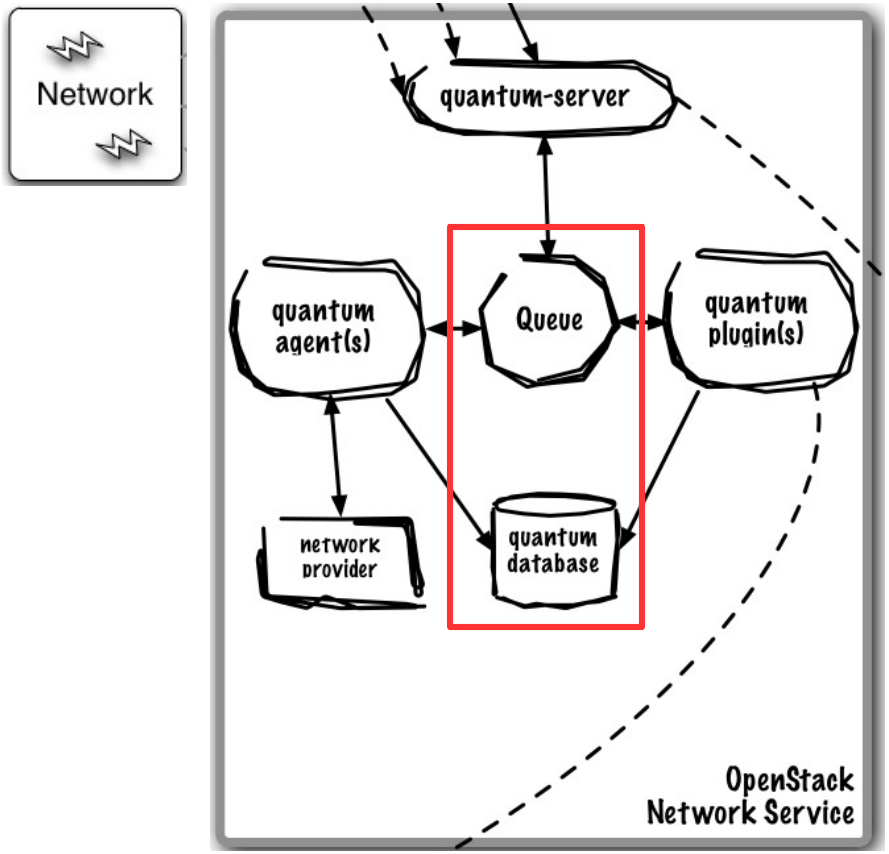
- Provision and manage virtualized network resources (networks, ports, attachments)

Key Capabilities:

- Flexible networking models to suit the needs of different applications or user groups
- Create/delete tenant-specific L2 networks
- Attach / Detach host to network
- L3 support (dedicated static and DHCP, Floating IPs, DHCP, Routing)
- L4-7 Support (Load Balancers)
- Extension framework enabling deploy and management of additional network services: intrusion detection systems (IDS), load balancing, firewalls and virtual private networks (VPN)
- Support for
  - OpenFlow (Big Switch, Floodlight, NEC controllers)
  - Numerous SDN and network virtualization providers (e.g Niciria, Midokura, Plum Grid, Brocade, Mellanox)
  - OpenVswitch
  - Cisco Nexus

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

Database and Queue are central to the Quantum control plane



Core Use Case:

- Queue provides RPC messaging between services
- Database provides data persistence

Runs As: Controller Service

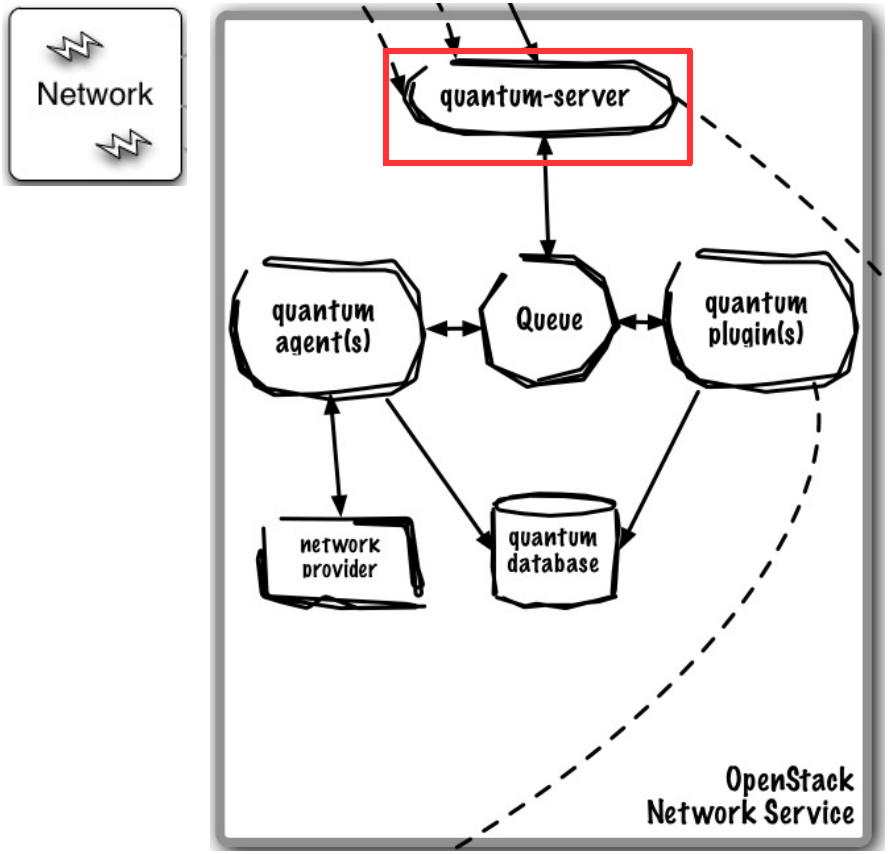
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- ZeroMQ implementation available to decentralize queue
- Can use same Queue as Nova

Key Capabilities:

- Community uses RabbitMQ as default queue, MySQL DB (IBM uses Apache Qpid and DB2)

quantum-server implements the OpenStack Network API



Core Use Case:

- Accept, validate, authenticate, and distribute incoming REST API requests

Runs As: Controller Service

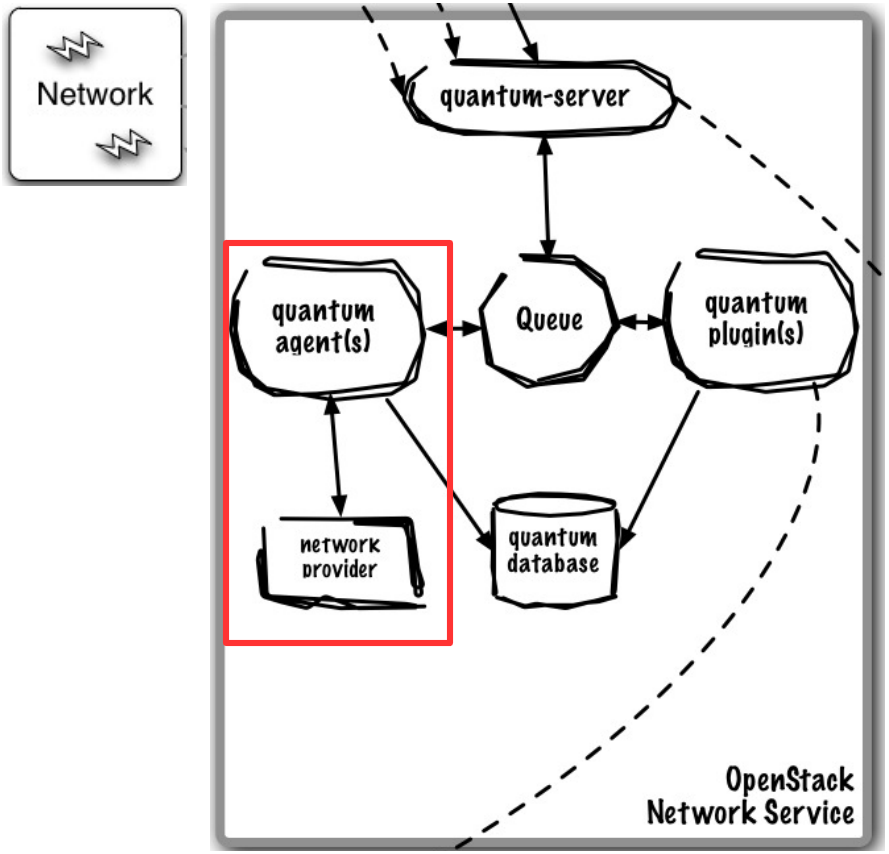
Deployment Considerations:

- Use active/passive or active/active for HA using Linux HA methods (e.g. corosync)

Key Capabilities:

- Requires access to a database for persistent storage
- Passes user requests to the configured OpenStack Networking plug-in for additional processing
- Relies on the OpenStack Identity Project (Keystone) for authentication and authorization of all API request.

Quantum uses an agent model to add additional functionality to a deployment



Core Use Case:

- plugin-agent: runs alongside nova-compute to manage physical host network connectivity
- dhcp-agent: provides DHCP to tenant networks
- l3-agent: provides L3/NAT forwarding for external network access

Runs As: Distributed Service (plugin-agent) or Controller Service (dhcp-agent, l3-agent)

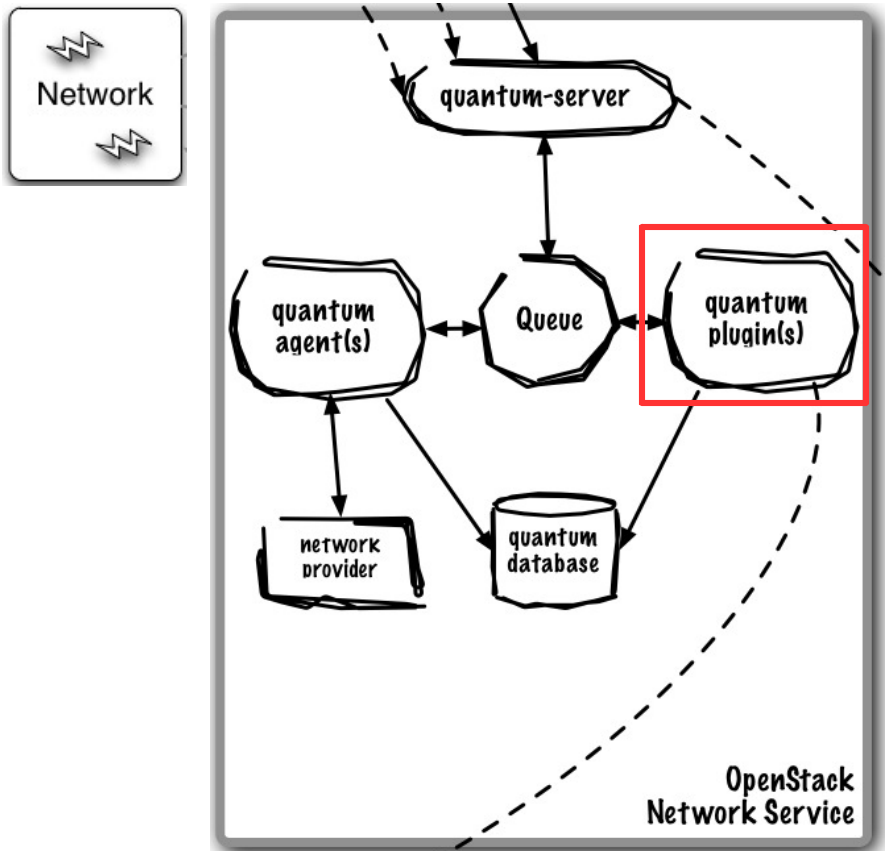
HA:

- plugin-agent: same as nova-compute, single instance is not HA, minimize failure domain
- dhcp-agent, l3-agent: running many ensure ensures availability to provision new, can use active/passive or active/active for HA of provisoined node.

Key Capabilities:

- plugin-agent: runs alongside nova-compute to manage physical host network connectivity
- dhcp-agent: provides DHCP to tenant networks
- l3-agent: provides L3/NAT forwarding for external network access

Quantum plugins are vendor or technology-specific plugins that map virtual network topology onto infrastructure



Core Use Case:

- Map virtual network topology onto infrastructure

Runs As: Controller Service

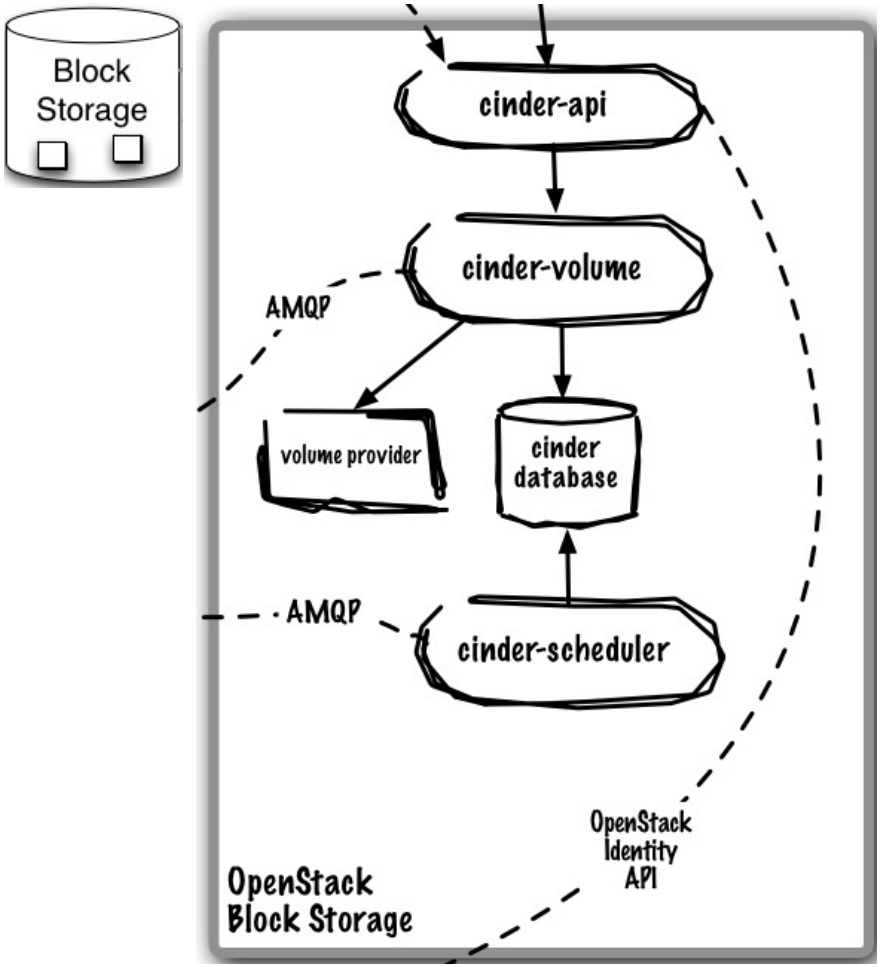
HA:

- Dependent on implementation

Key Capabilities:

- Uses plug-in model to support vendor-specific or technology-specific implementation that translates virtual networks to physical network

Storage (Cinder) exposes block devices to be connected to compute instances for expanded storage, better performance and enterprise storage platform integration



Core Use Cases:

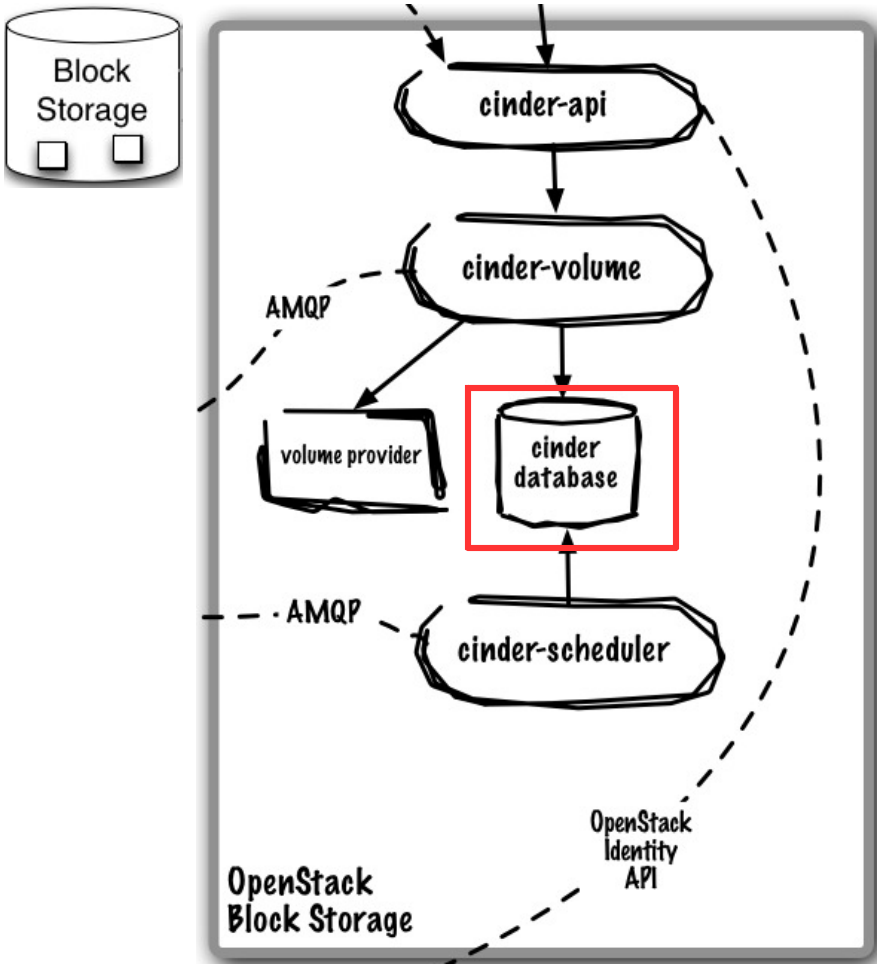
- Provision and manage lifecycle of volumes and their exposure for attachment

Key Capabilities:

- Persistent block level storage devices for use with OpenStack compute instance
- Manage the creation, attaching and detaching of the block devices to servers
- Support for booting virtual machines from Cinder-backed storage
- Snapshot and restore functionality
- Supports following
  - LVM-backed volumes (iSCSI)
  - XIV (iSCSI)
  - SVC (iSCSI and Fiber Channel)
  - NetApp (iSCSI and NFS)
  - EMC (iSCSI)
  - HP/Lefthand (iSCSI)
  - RADOS block devices (e.g. Ceph distributed file system)
    (full list at Cinder Support Matrix)

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

Database and the Queue are the core of Cinder's control plane



Core Use Case:

- Queue provides RPC messaging between services
- Database provides data persistence

Runs As: Controller Service

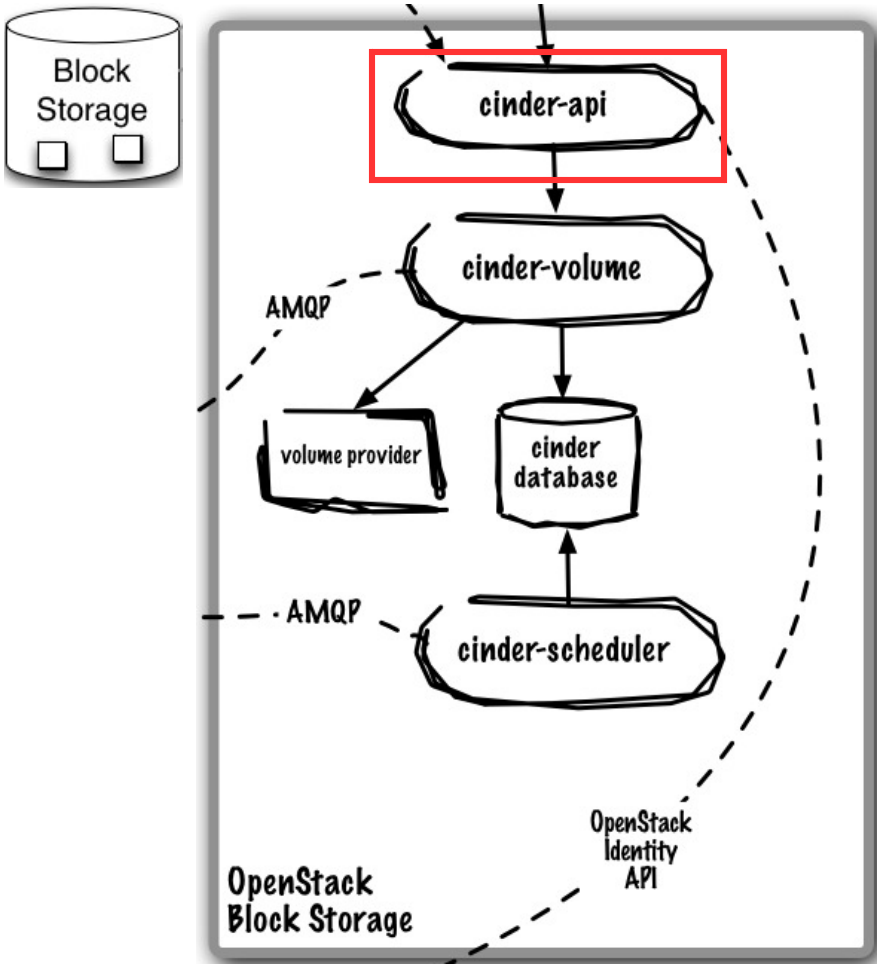
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- ZeroMQ implementation available to decentralize queue
- Can use same queue/database as Nova

Key Capabilities:

- Community uses RabbitMQ as default queue, MySQL DB (IBM uses Apache Qpid and DB2)

cinder-api is the entry point to OpenStack Volume Service



Core Use Case:

- Accept, validate, authenticate, and distribute incoming REST API requests

Runs As: Controller Service

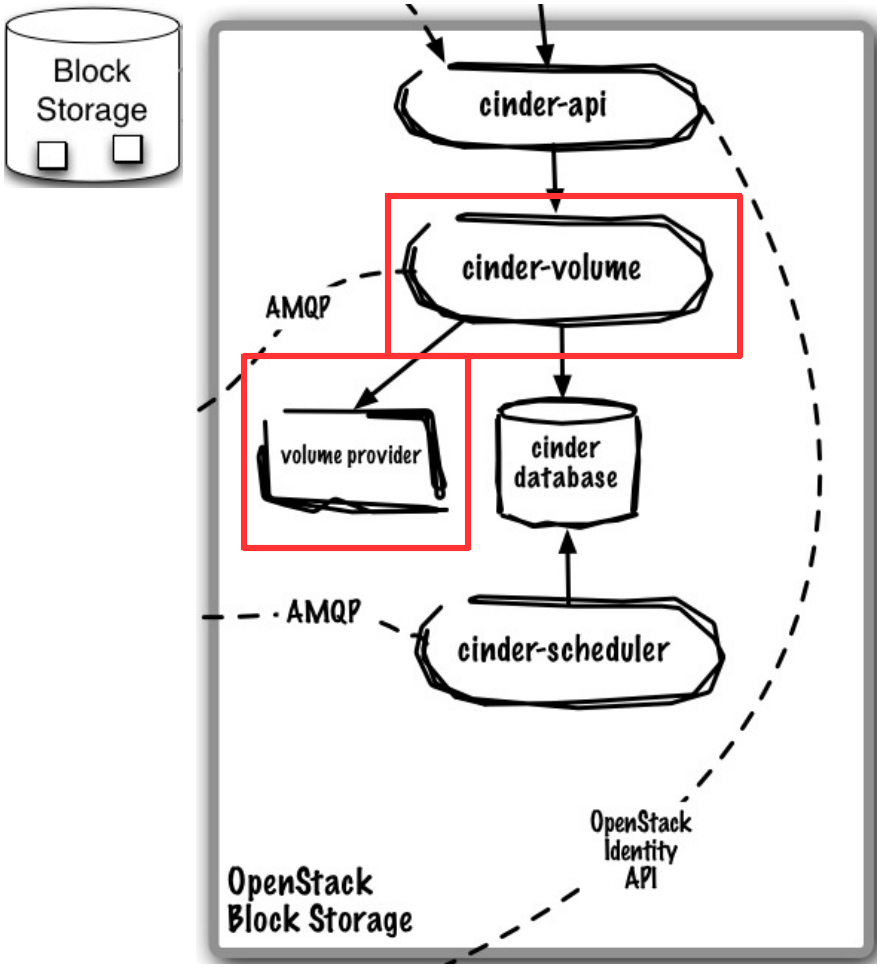
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Volume API
- Robust extensions mechanism to add new capabilities

cinder-volume manages individual block-based volume providers



Core Use Case:

- Manages interactions with single block volume provider

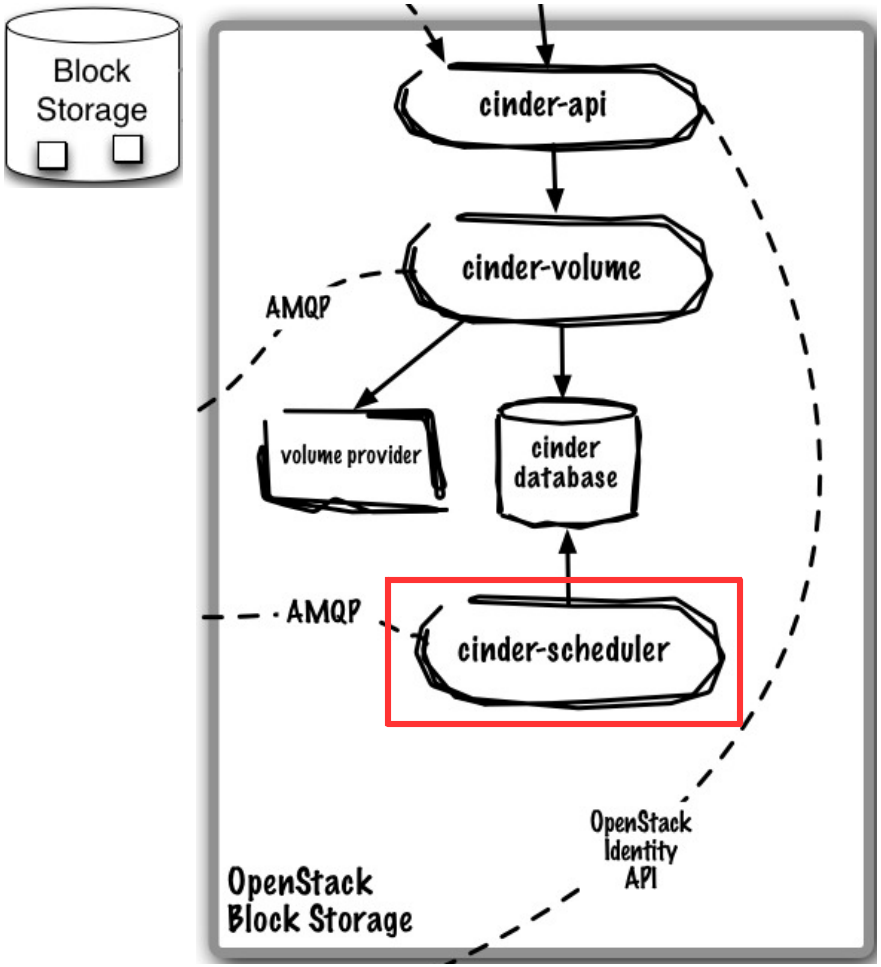
Runs As: Distributed Service

Deployment Considerations:

- Many cinder-volume instances exist in the environment to ensure volume provisioning is always available
- Single cinder-volume is not HA, manage single provider to minimize failure domain

Key Capabilities:

- Create and manage volumes on storage backend
- Expose volumes to physical host (e.g. iSCSI, FC)
- Uses plug-in model to support differing storage systems



Core Use Case:

- Selects cinder-volume service to place volume on

Runs As: Controller Service

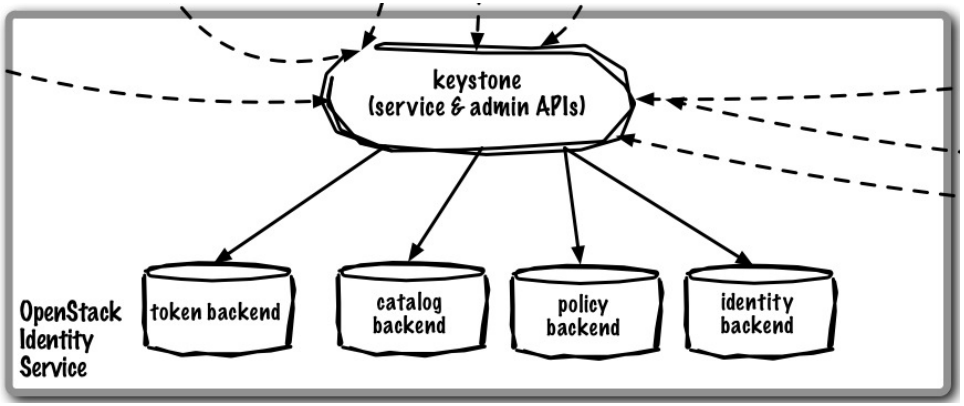
Deployment Considerations:

- Default scheduler is horizontally scalable

Key Capabilities:

- Default scheduled is allocation-based using a series of filters to reduce set of applicable hosts and uses costing functions to provide weight

Identity Service (Keystone) offers project-wide identity, token, service catalog, and policy services designed for integration with existing systems



Core Use Cases:

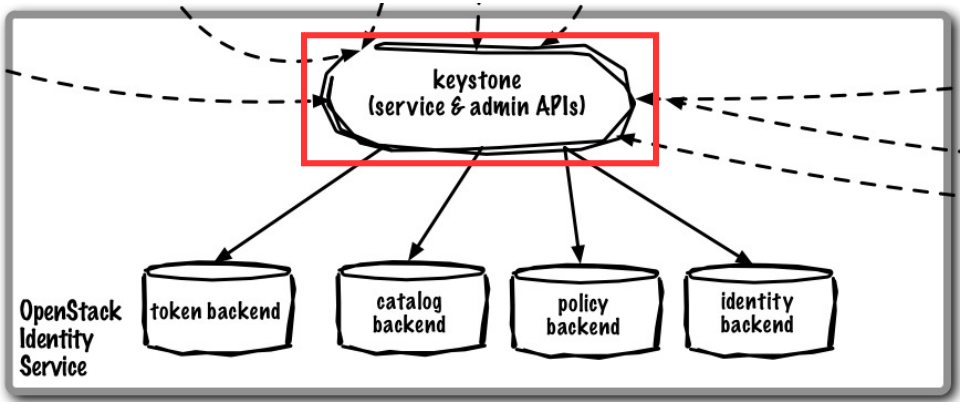
- Installation-wide authentication and authorization to OpenStack services

Key Capabilities:

- Authenticate user / password requests against multiple backends (SQL, LDAP, etc) (Identity Service)
- Validate / manage tokens used after initial username/password verification (Token Service)
- Endpoint registry of available services (Service Catalog)
- Authorize API requests (Policy Service)
- Domain / Project / User model with RBAC for access to compute, storage, networking
- Policy service provides a rule-based authorization engine and the associated rule management interface.

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

keystone service is the entry point for all AuthN and AuthZ in OpenStack



Core Use Case:

- Handle and service all Identity REST API requests

Runs As: Controller Service

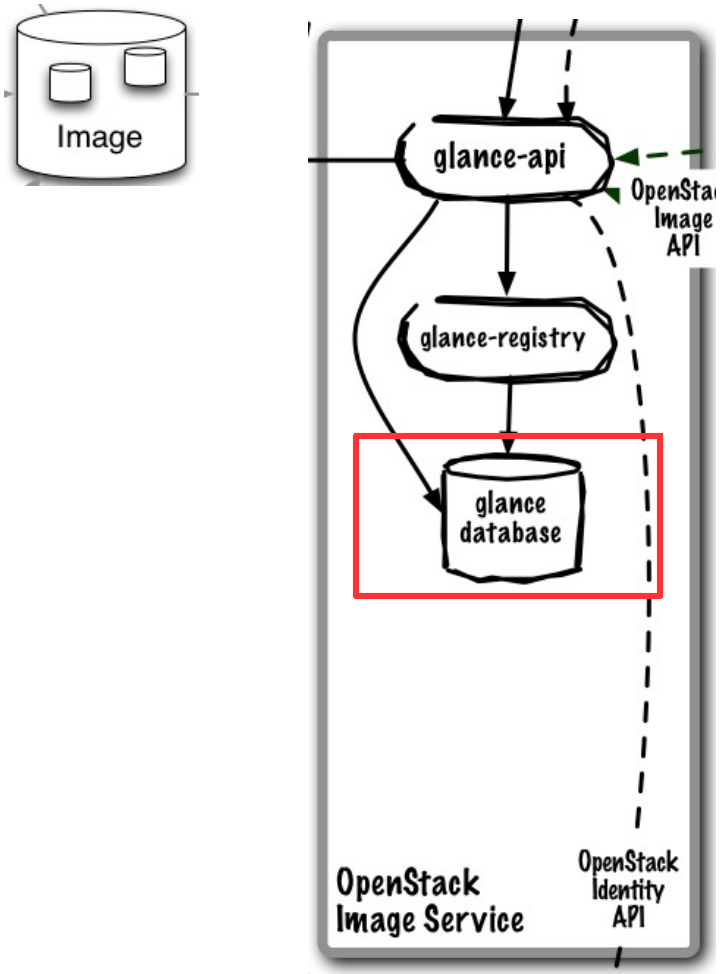
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Identity API
- Pluggable backends for each function: identity, token, catalog, and policy

Glance database persists all image related metadata



Core Use Case:

- Persist image-related metadata

Runs As: Controller Service

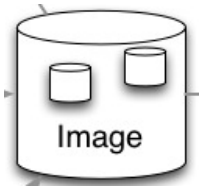
Deployment Considerations:

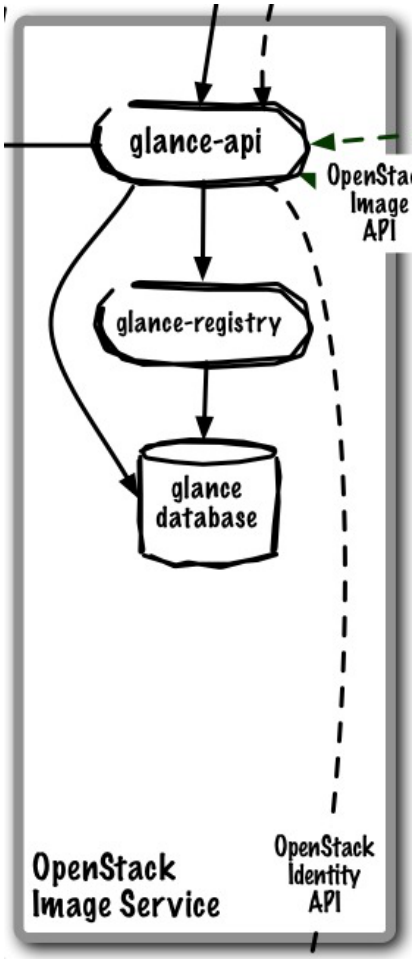
- Use DB and Queue clustering/HA methods
- Can use same queue/database as Nova

Key Capabilities:

- Persists image-related metadata

Image Service (Glance) provides registration, discovery, and delivery services for virtual disk and server images





Core Use Cases:

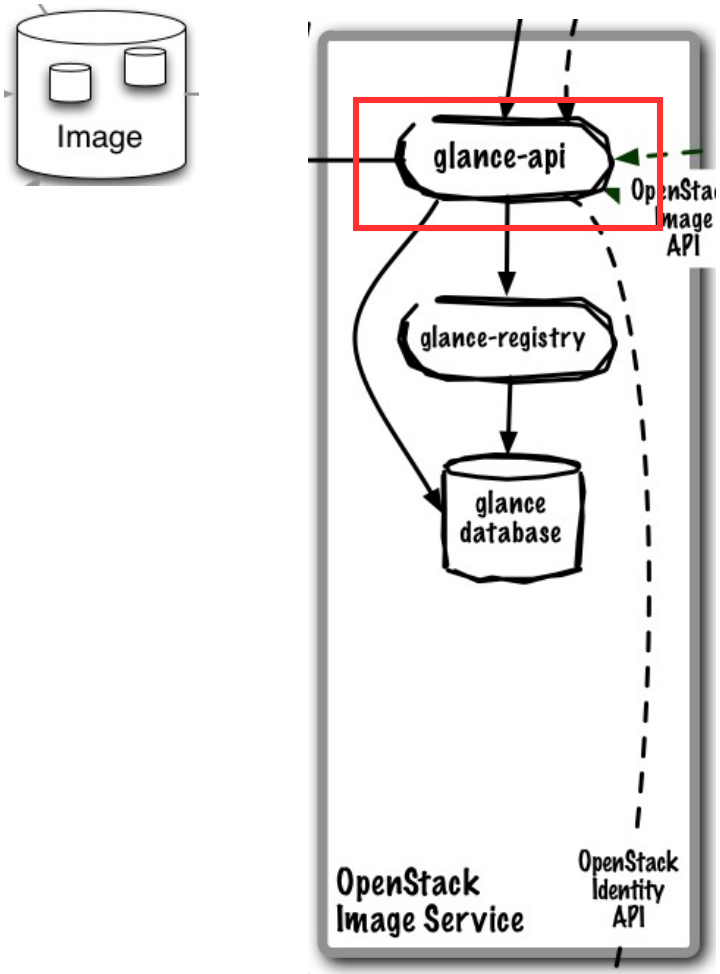
- Administrator registers available guest images
- End-user discovers available guest images
- Deliver image to compute node on provisioning

Key Capabilities:

- Image Registry (storage optional and is delegated to a configurable store)
- Administrators can create base templates from which 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 machines can be backed up quickly
- Supported formats: Raw, Machine (a.k.a. Amazon AMI), VHD (Hyper-V), VDI (VirtualBox), qcow2 (Qemu/KVM), VMDK (VMWare), OVF (VMWare, others)

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

glance-api routes incoming REST API Requests



Core Use Case:

- Routes REST API requests to the appropriate handler

Runs As: Controller Service

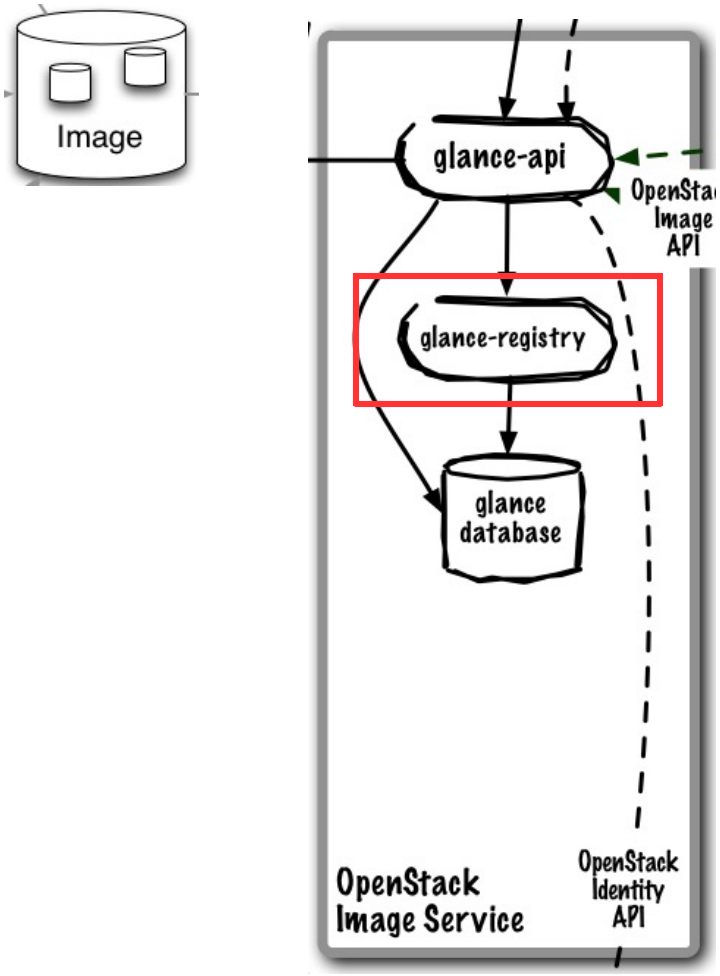
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Image API
- Routes requests from clients to registries of image metadata and to its backend stores
- Pluggable image store backends

glance-registry services Image Service API requests



Core Use Case:

- Services Identity REST API requests

Runs As: Controller Service

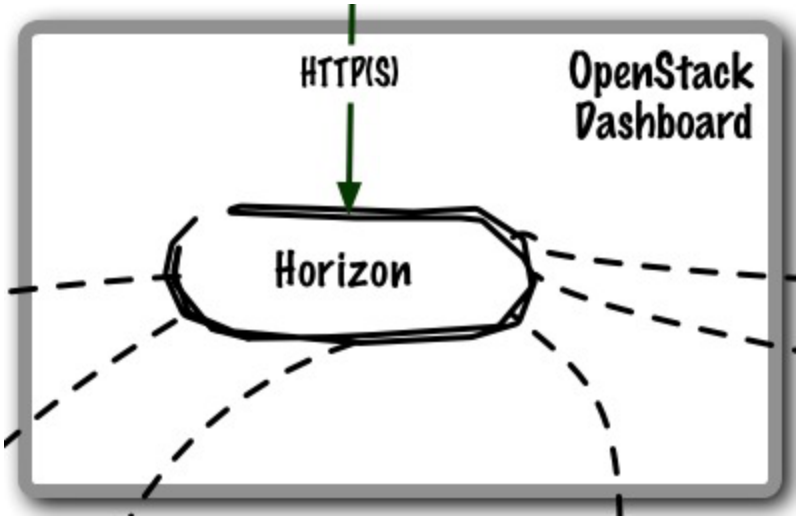
Deployment Considerations:

- (to be determined)

Key Capabilities:

- APIs supported
  - OpenStack Image API

Horizon (Dashboard) enables administrators and users to access, provision, and manage resources through a self-service portal GUI



NOT SHIPPED BY IBM

Core Use Cases:

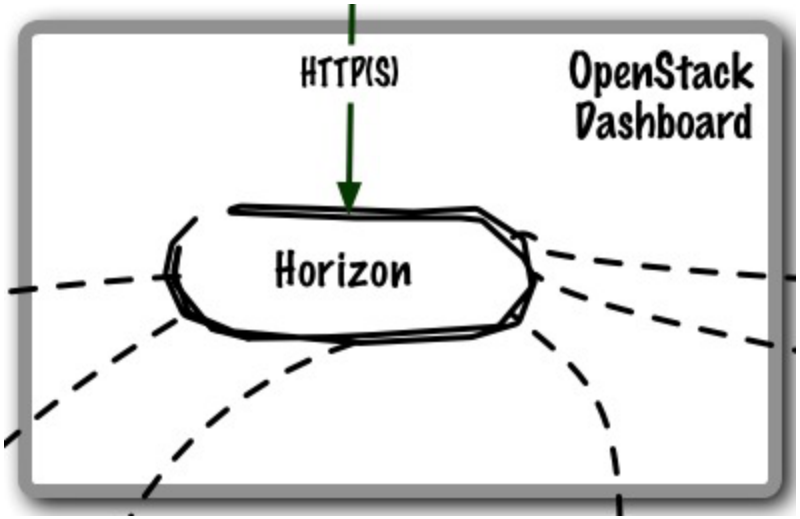
- Self-service portal for compute and object storage
- Cloud administration (users/projects, quotas, etc.)

Key Capabilities:

- Thin wrapper over APIs, no local state
- Registration pattern for applications to hook into
- Out-of-the-box support for all core OpenStack projects.
- Anyone can add a new component as a “first-class citizen”.
- Visual and interaction paradigms are maintained throughout.

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

horizon is the self-service portal implementation



NOT SHIPPED BY IBM

Core Use Case:

- GUI access to OpenStack APIs

Runs As: Controller Service

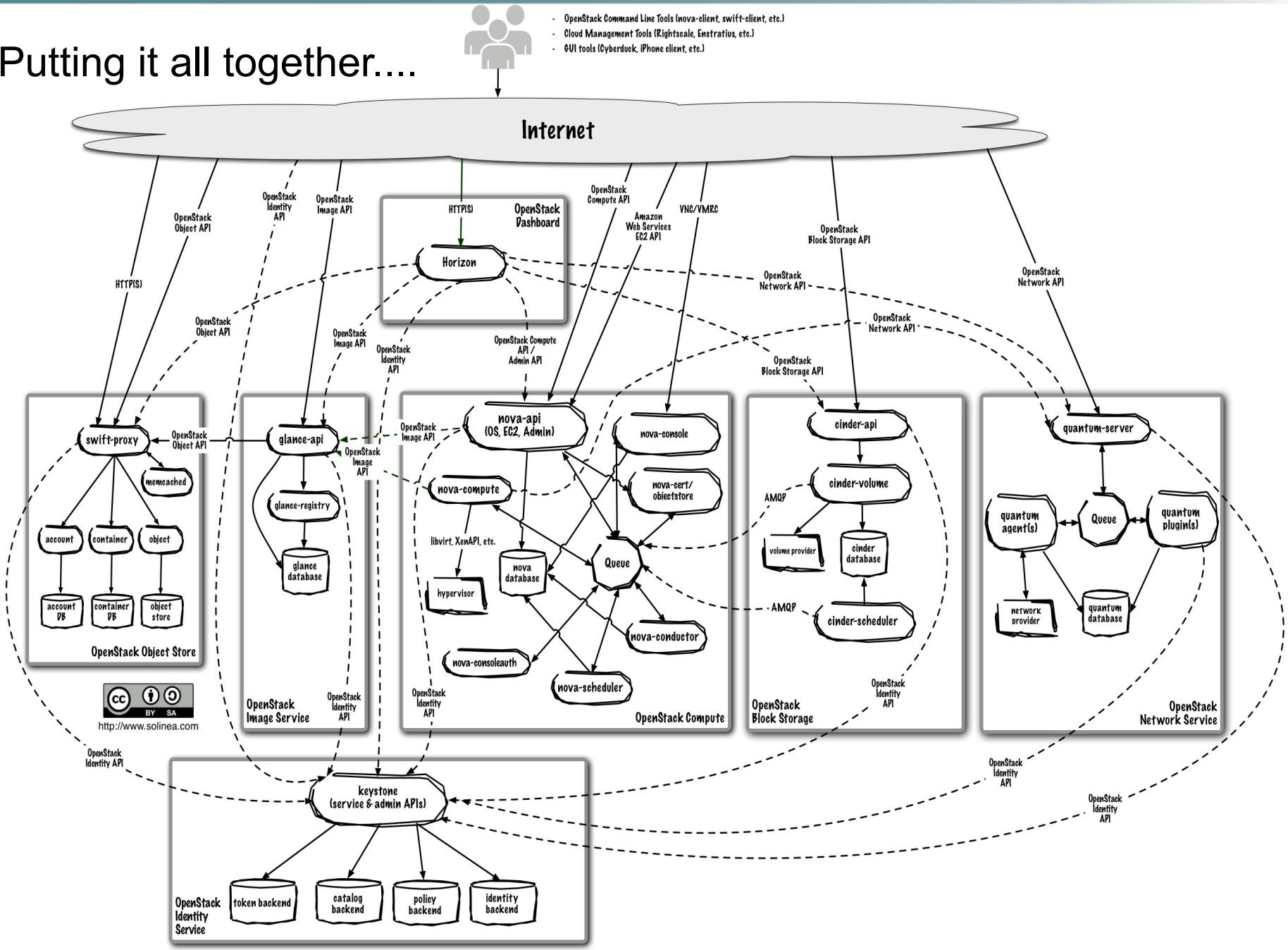
Deployment Considerations:

- (to be determined)

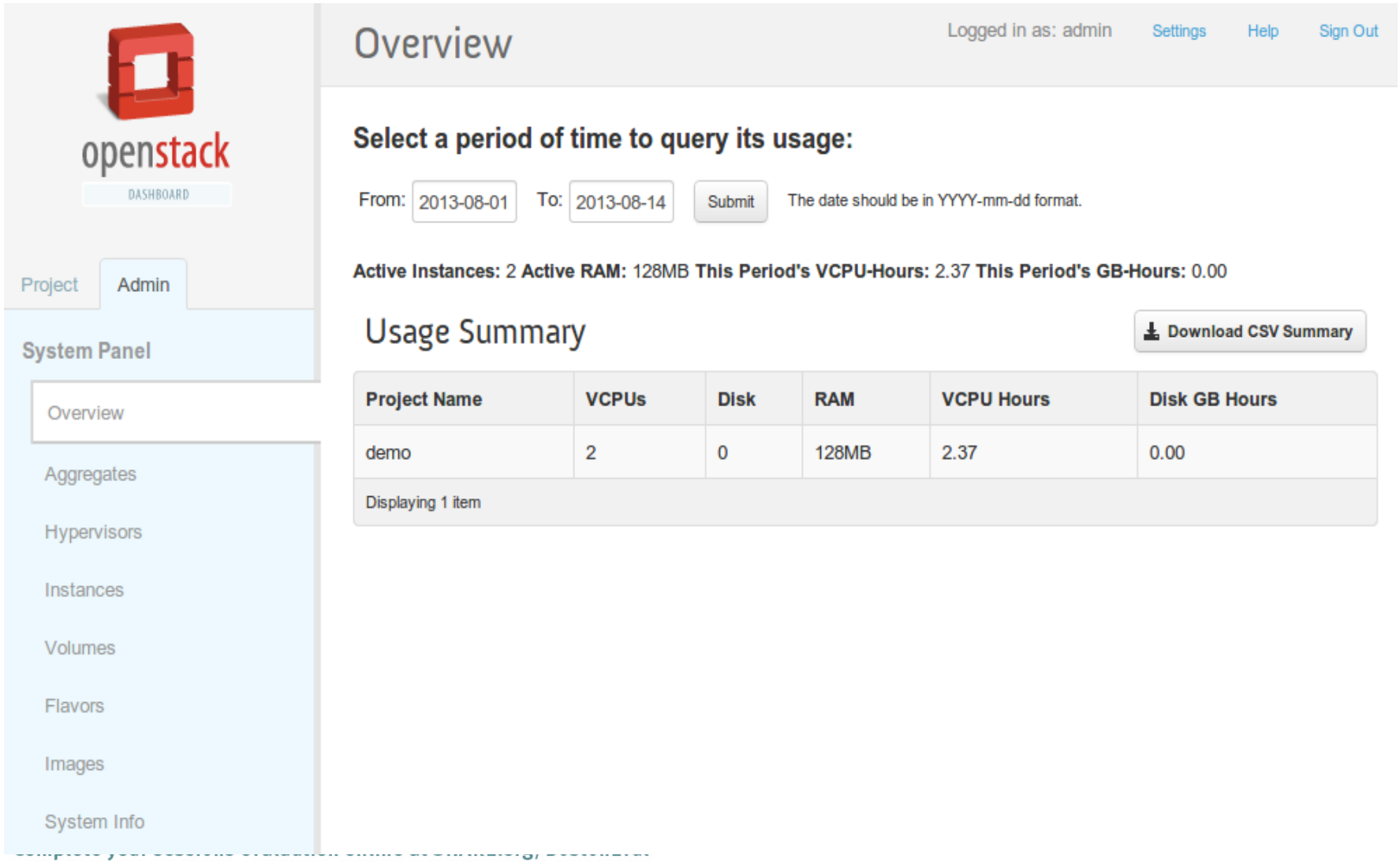
Key Capabilities:

- Provision and manage virtual servers, volumes, and networks
- Create and manage tenants and users

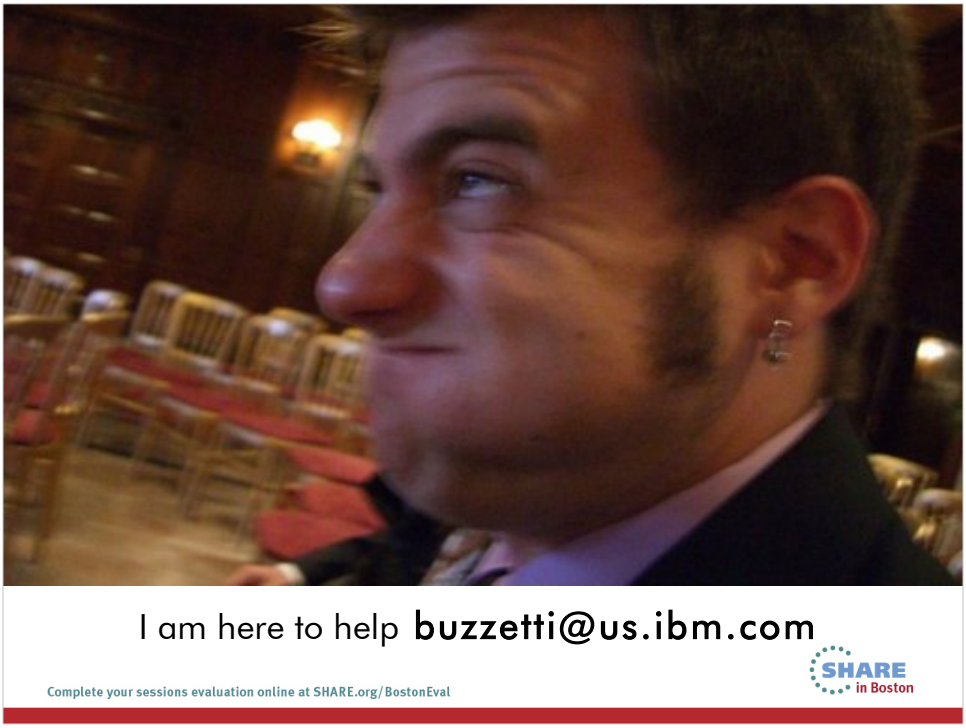
Putting it all together....



Demo

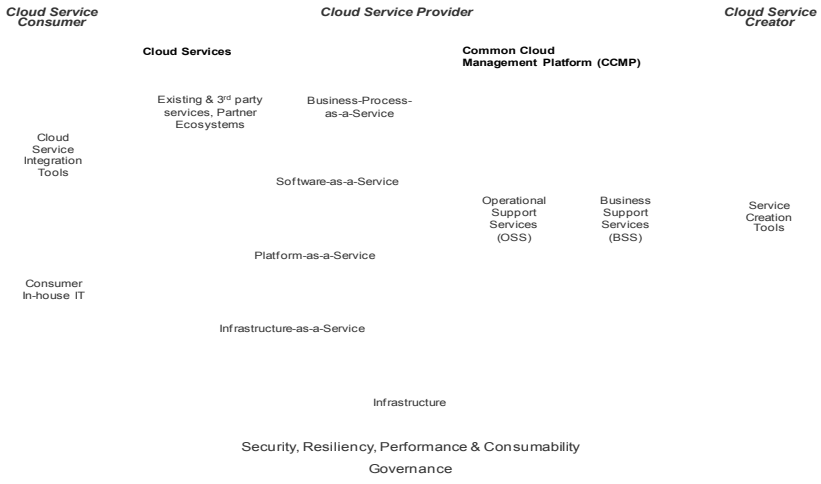

openstack
DASHBOARD
Project
Admin
System Panel
Overview
Aggregates
Hypervisors
Instances
Volumes
Flavors
Images
System Info
Overview
Logged in as: admin
Settings
Help
Sign Out
Select a period of time to query its usage:
From: 2013-08-01
To: 2013-08-14
Submit
The date should be in YYYY-mm-dd format.
Active Instances: 2 Active RAM: 128MB This Period's VCPU-Hours: 2.37 This Period's GB-Hours: 0.00
Usage Summary
Download CSV Summary
Project Name
VCPUs
Disk
RAM
VCPU Hours
Disk GB Hours
demo
2
0
128MB
2.37
0.00
Displaying 1 item

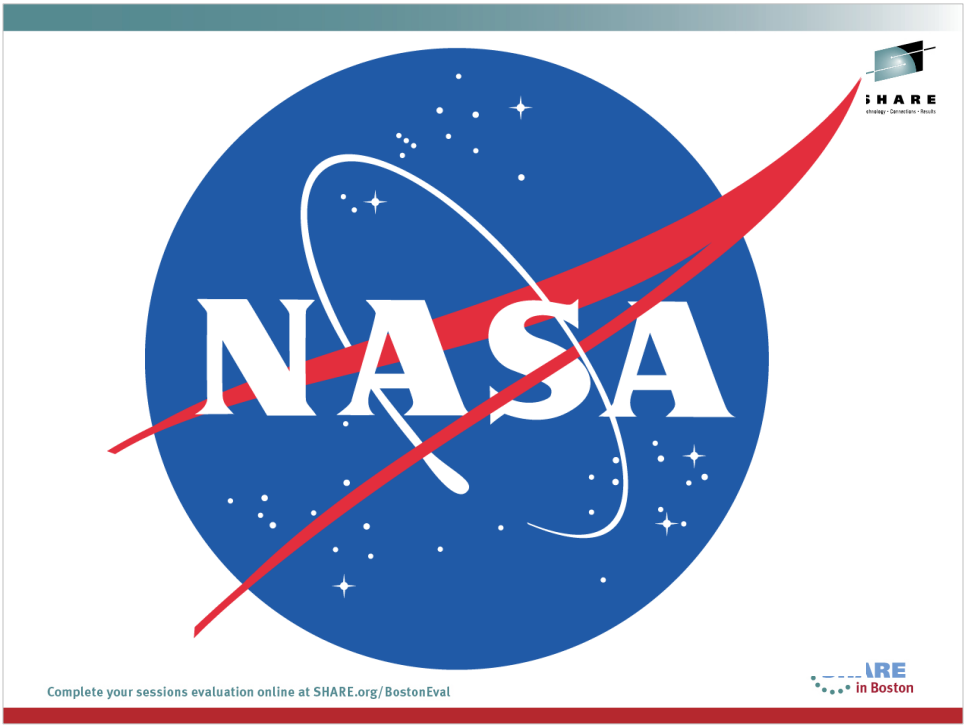
Questions ?



This is me. I am here to help. I include this chart so that people can have my email.
The reason I created this presentation is based on the past few years working with customers. Helping them understand that there is a lot of virtualization out there.
Although I might look young, I have been in the IT field for almost 15 years. Virtualization has been a core technology for me for most of it.

IBM's Reference Architecture for Cloud





http://en.wikipedia.org/wiki/Nebula_(computing_platform)

Nebula, that originated to support NASA research projects, was donated to OpenStack in 2010.

RackSpace

Complete your sessions evaluation online at SHARE.org/BostonEval

http://en.wikipedia.org/wiki/Rackspace

IT hosting company founded in 1998. Contributed its cloud files product in 2010. This became part of the Object Storage portion of OpenStack (Swift).

Community

More than 6000 people and 100 companies

Active online community through mailing lists, IRC, wiki

Bi-yearly design summits

Companies need to donate money AND people that ACTIVELY contribute

Platinum Members

AT&T

Canonical

HP

IBM

Nebula

Rackspace

Red Hat, Inc.

SUSE

and many more ...
http://www.OpenStack.org/foundation/cor

Relase Names

These codenames are chosen by popular vote using the basic Launchpad poll feature over the ~openstack group. Codenames are cities or counties near where the corresponding OpenStack design summit took place. An exception (called the Waldon exception) is granted to elements of the state flag that sound especially cool.

Austin: The first design summit took place in Austin, TX
Bexar: The second design summit took place in San Antonio, TX
Cactus: Cactus is a city in Texas
Diablo: Diablo is a city in the bay area near Santa Clara, CA
Essex: Essex is a city near Boston, MA
Folsom: Folsom is a city near San Francisco, CA
Grizzly: Grizzly is an element of the state flag of California
design summit takes place in San Diego, CA
Havana: Havana is an unincorporated community in Oregon
design summit takes place in Oregon
Ichang?: Design Summet to take place in Hong-Kong

Commitment

Top 10 Contributors to OpenStack by Release

Essex April 2012			Folsom Sept 2012			Grizzly April 2013		
	Commits	Authors		Commits	Authors		Commits	Authors
Rackspace	2,275	52	Rackspace	973	49	Red Hat	1,854	38
Red Hat	507	12	Red Hat	861	15	Rackspace	944	58
Nebula	431	8	Nebula	464	9	IBM	895	41
HP	261	21	HP	179	25	HP	401	49
Canonical	118	7	Isi	144	3	Nebula	286	10
Nicira	105	3	Cloudscaling	140	4	Intel	243	13
Citrix	87	9	IBM	139	16	eNovance	238	7
Enovance	54	2	Sina	135	18	VMware	186	4
Cloudscaling	52	1	Canonical	108	5	Cloudscaling	175	3
Isi	40	2	Inktank	90	2	Dreamhost	171	7

Source: bitergia.com

http://www.domicity.com/2013/06/openstack-accelerating-commitment-by-big-vendors/

OpenStack design tenets focus on delivering Cloud Computing Platform on an available, scalable, and elastic control plane

Basic Design Tenets

1) Scalability and elasticity are our main goals

2) Any feature that limits our main goals must be optional

3) Everything should be asynchronous
- If you can't do something asynchronously, see #2

1) All required components must be horizontally scalable

2) Always use shared nothing architecture (SN) or sharding
- If you can't Share nothing/shard, see #2

1) Distribute everything
- Especially logic. Move logic to where state naturally exists.

1) Accept eventual consistency and use it where it is appropriate.

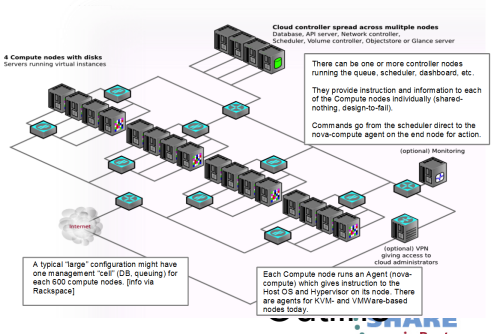
2) Test everything.
- We require tests with submitted code. (We will help you if you need it)

Sources:
http://www.openstack.org/downloads/openstack-compute-datasheet.pdf
http://wiki.openstack.org/BasicDesignTenets

OpenStack Open Source Cloud 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

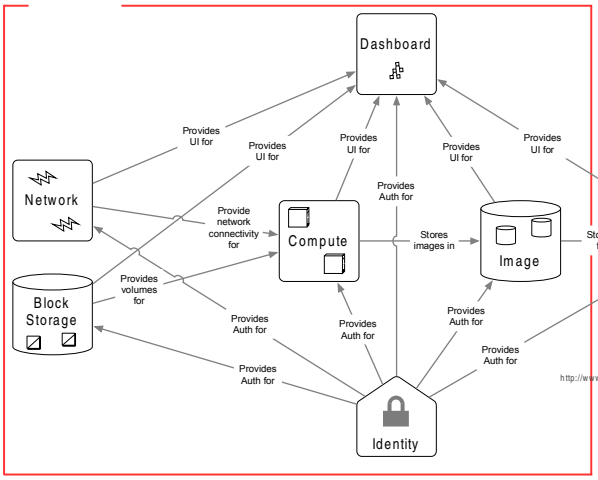
Level



Level

OpenStack is comprised of six core projects delivering an IaaS solution + a project delivering an Object Storage solution

Focus



IaaS

Compute (Nova)

Block Storage (Cinder)

Network (Quantum)
Provision and manage virtual resources

Dashboard (Horizon)
Self-service portal

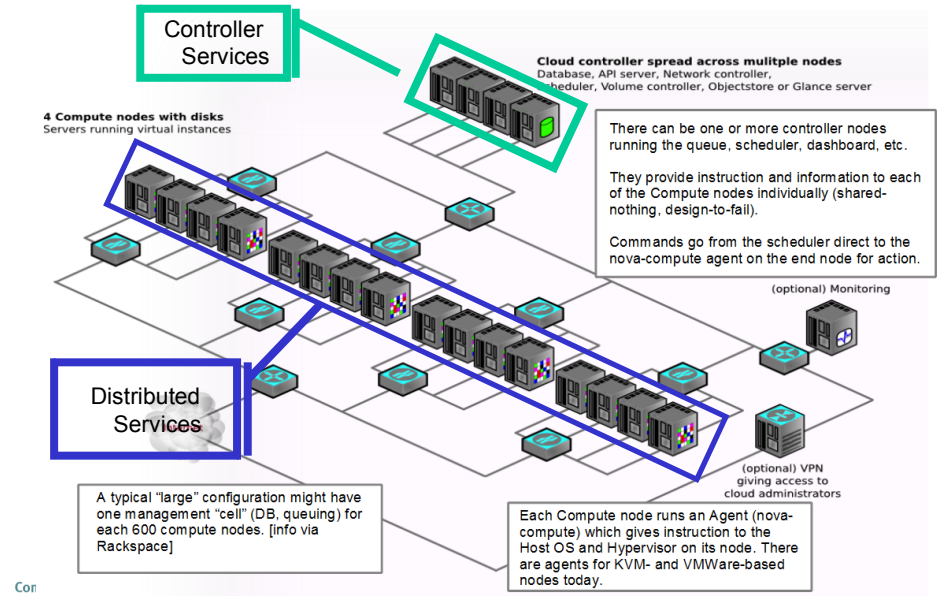
Image (Glance)
Catalog and manage server images

Identity (Keystone)
Unified authentication and authorization

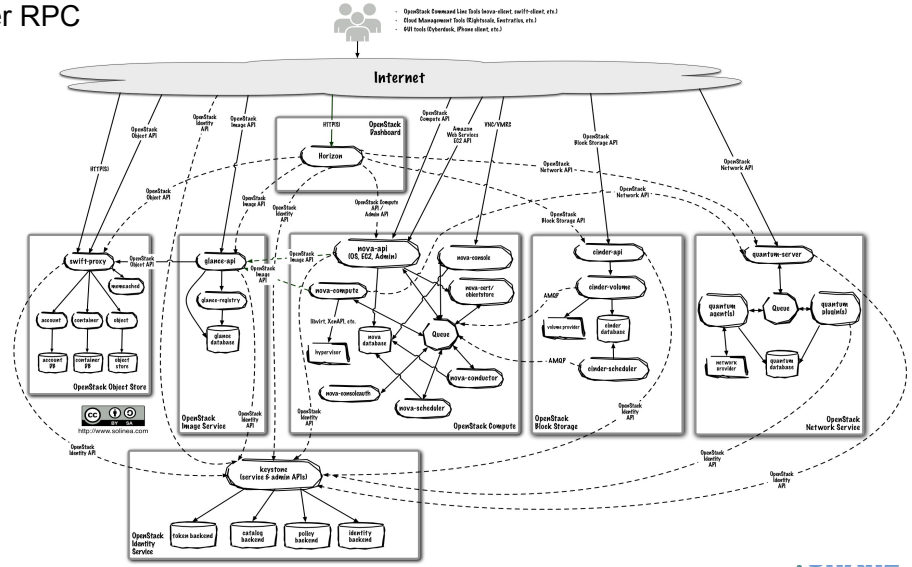
Object Storage (Swift)
petabytes of secure, reliable object storage

mage Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

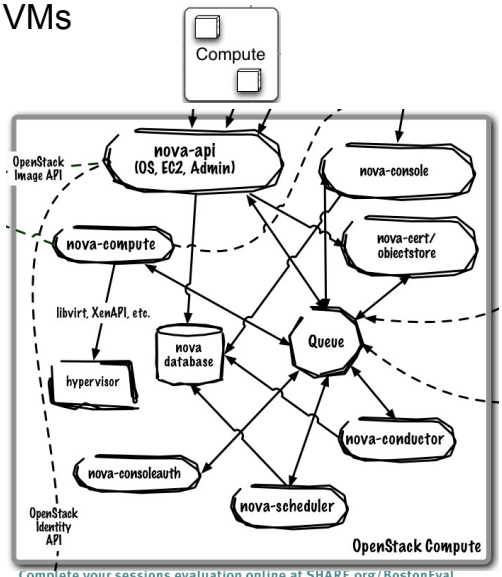
OpenStack services can be categorized into two groups – controller services and distributed services (but all can be scaled-out!)



Deployments consist of projects interfacing over public APIs, with each project composed of multiple services interfacing via private APIs over RPC



Compute (Nova) is a horizontally scalable offering on-demand compute resources by provisioning and managing VMs



Core Use Case:

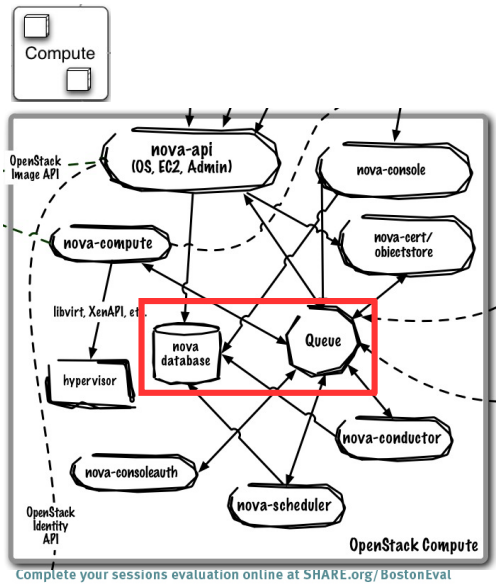
- Provision and manage virtualized compute resources (CPU, memory, disk, network)

Key Capabilities:

- REST-based APIs with rate limiting and authentication
- Manage Local Area Networks (LAN)
- Live migration of guests
- VM management (Instance)
  - Run, reboot, suspend, resize, terminate instances
- Floating IP addresses
- Security Groups
- RBAC with Projects & Quotas
- Manage to KVM, Xen (XenServer, Xen Cloud Platform), LXC, VMware vSphere 4.1+, Hyper-V, Bare Metal, PowerVM (limited)

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

Database and Queue are central to the Nova control plane



Core Use Case:

- Queue provides RPC messaging between services
- Database provides data persistence

Runs As: Controller Service

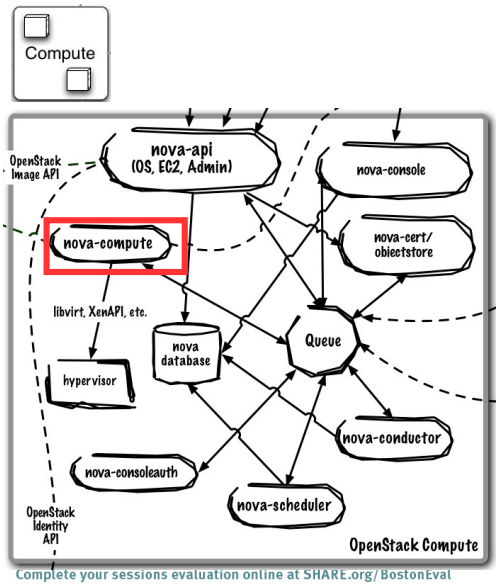
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- ZeroMQ implementation available to decentralize queue

Key Capabilities:

- Community uses RabbitMQ as default queue, MySQL DB (IBM uses Apache Qpid and DB2)
- Single "cell" (1 Queue, 1 Database) typically scales from 500 – 1000 physical machines
  - Cells can be rolled up to support larger deployments
- Communications route through queue
  - API requests are validated and placed on queue
  - Workers listen to queues based on role or role + hostname
  - Responses are dispatched back through queue

nova-compute manages individual hypervisors and compute nodes



Core Use Case:

- Manage all interactions with single hypervisor control point

Runs As: Distributed Service

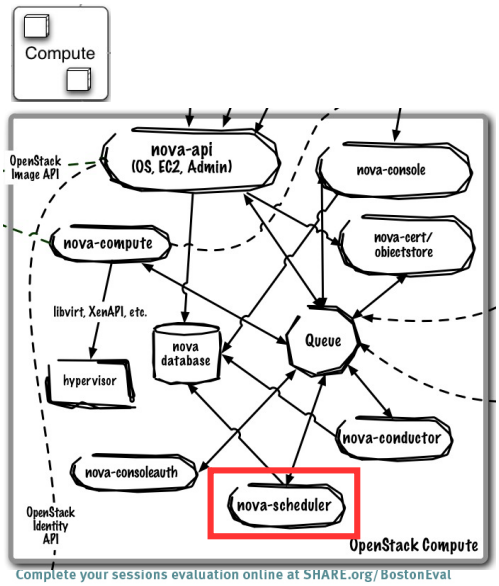
Deployment Considerations:

- Many nova-compute instances exist in the environment to ensure compute provisioning is always available
- Single nova-compute is not HA, manage single hypervisor to minimize failure domain
- No direct database acces is required

Key Capabilities:

- Create and manage virtual machines on hypervisor
- Attach networks and volumes to physical host (iSCSI, FC), expose to guest virtual machines
- Implementation point for security groups defining firewall rules for guest network traffic
- Uses plug-in model to manage to different hypervisors

nova-scheduler allocates virtual resources to compute nodes



Core Use Case:

- Selects compute node to run virtual machine on

Runs As: Controller Service

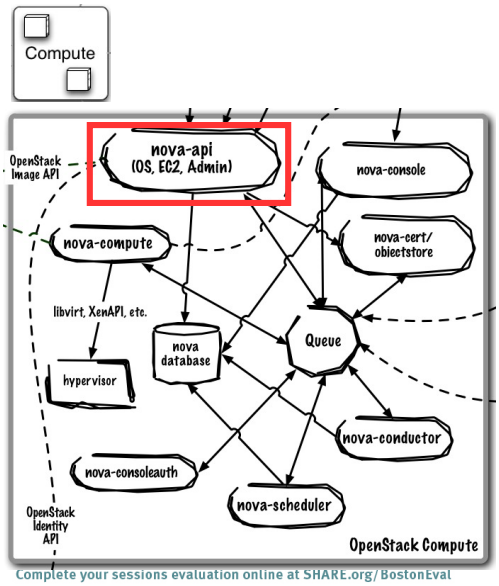
Deployment Considerations:

- Default scheduler is horizontally scalable
- For other schedulers (e.g. Platform EGO), follow their specific best practice

Key Capabilities:

- Default scheduled is allocation-based using a series of filters to reduce set of applicable hosts and uses costing functions to provide weight
- Platform EGO adds utilization-based scheduling to default allocation based

nova-api supports multiple API implementations and is the entry point into the cloud



Core Use Case:

- Accept, validate, authenticate, and distribute incoming REST API requests

Runs As: Controller Service

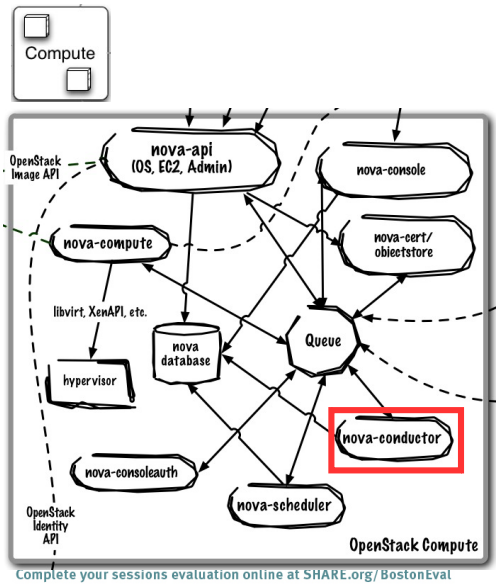
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Compute API
  - EC2 API (subset)
- Robust extensions mechanism to add new capabilities

nova-conductor manages database interactions on behalf of compute nodes



Core Use Case:

- Handles all database requests for nova-compute service

Runs As: Controller Service

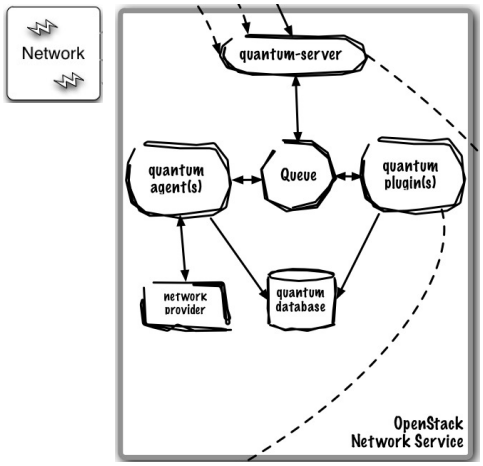
Deployment Considerations:

- Horizontally scalable, start many instances

Key Capabilities:

- Talks directly to database on behalf of compute nodes

Network (Quantum) is a pluggable, scalable and API-driven system for managing networks and IP addresses



Core Use Cases:

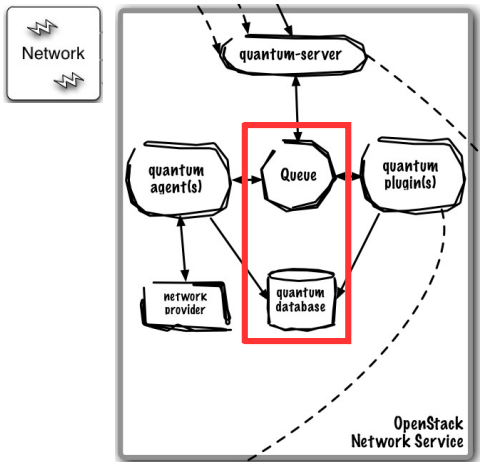
- Provision and manage virtualized network resources (networks, ports, attachments)

Key Capabilities:

- Flexible networking models to suit the needs of different applications or user groups
- Create/delete tenant-specific L2 networks
- Attach / Detach host to network
- L3 support (dedicated static and DHCP, Floating IPs, DHCP, Routing)
- L4-7 Support (Load Balancers)
- Extension framework enabling deploy and management of additional network services: intrusion detection systems (IDS), load balancing, firewalls and virtual private networks (VPN)
- Support for
  - OpenFlow (Big Switch, Floodlight, NEC controllers)
  - Numerous SDN and network virtualization providers (e.g Niciria, Midokura, Plum Grid, Brocade, Mellanox)
  - OpenVswitch
  - Cisco Nexus

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

Database and Queue are central to the Quantum control plane



Core Use Case:

- Queue provides RPC messaging between services
- Database provides data persistence

Runs As: Controller Service

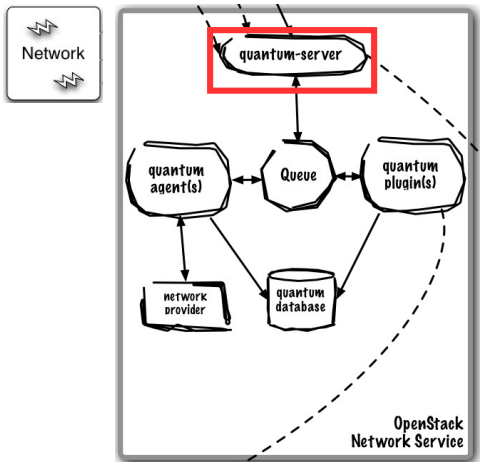
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- ZeroMQ implementation available to decentralize queue
- Can use same Queue as Nova

Key Capabilities:

- Community uses RabbitMQ as default queue, MySQL DB (IBM uses Apache Qpid and DB2)

quantum-server implements the OpenStack Network API



Core Use Case:

- Accept, validate, authenticate, and distribute incoming REST API requests

Runs As: Controller Service

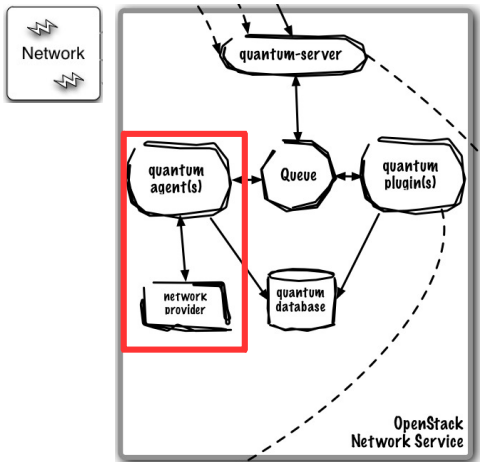
Deployment Considerations:

- Use active/passive or active/active for HA using Linux HA methods (e.g. corosync)

Key Capabilities:

- Requires access to a database for persistent storage
- Passes user requests to the configured OpenStack Networking plug-in for additional processing
- Relies on the OpenStack Identity Project (Keystone) for authentication and authorization of all API request.

Quantum uses an agent model to add additional functionality to a deployment



Core Use Case:

- plugin-agent: runs alongside nova-compute to manage physical host network connectivity
- dhcp-agent: provides DHCP to tenant networks
- l3-agent: provides L3/NAT forwarding for external network access

Runs As: Distributed Service (plugin-agent) or Controller Service (dhcp-agent, l3-agent)

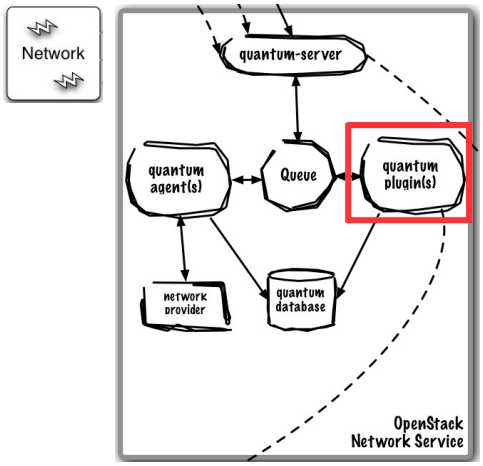
HA:

- plugin-agent: same as nova-compute, single instance is not HA, minimize failure domain
- dhcp-agent, l3-agent: running many ensure ensures availability to provision new, can use active/passive or active/active for HA of provisoined node.

Key Capabilities:

- plugin-agent: runs alongside nova-compute to manage physical host network connectivity
- dhcp-agent: provides DHCP to tenant networks
- l3-agent: provides L3/NAT forwarding for external network access

Quantum plugins are vendor or technology-specific plugins that map virtual network topology onto infrastructure



Core Use Case:

- Map virtual network topology onto infrastructure

Runs As: Controller Service

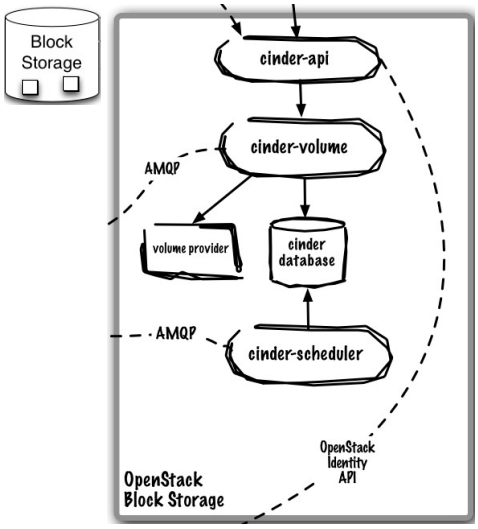
HA:

- Dependent on implementation

Key Capabilities:

- Uses plug-in model to support vendor-specific or technology-specific implementation that translates virtual networks to physical network

Storage (Cinder) exposes block devices to be connected to compute instances for expanded storage, better performance and enterprise storage platform integration



Core Use Cases:

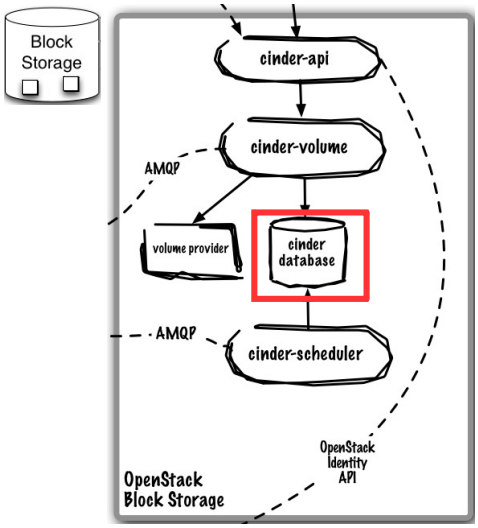
- Provision and manage lifecycle of volumes and their exposure for attachment

Key Capabilities:

- Persistent block level storage devices for use with OpenStack compute instance
- Manage the creation, attaching and detaching of the block devices to servers
- Support for booting virtual machines from Cinder-backed storage
- Snapshot and restore functionality
- Supports following
  - LVM-backed volumes (iSCSI)
  - XIV (iSCSI)
  - SVC (iSCSI and Fiber Channel)
  - NetApp (iSCSI and NFS)
  - EMC (iSCSI)
  - HP/Lefthand (iSCSI)
  - RADOS block devices (e.g. Ceph distributed file system)
    (full list at Cinder Support Matrix)

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

Database and the Queue are the core of Cinder's control plane



Core Use Case:

- Queue provides RPC messaging between services
- Database provides data persistence

Runs As: Controller Service

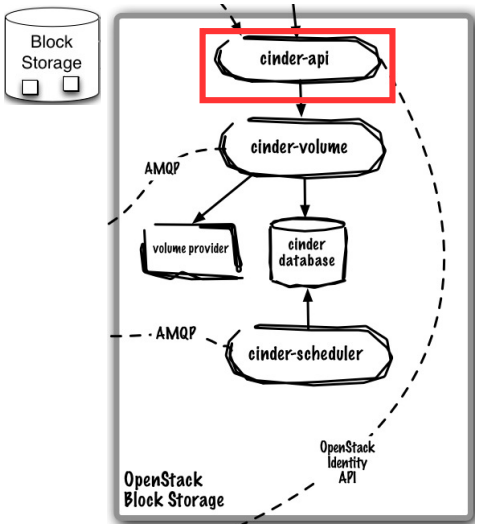
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- ZeroMQ implementation available to decentralize queue
- Can use same queue/database as Nova

Key Capabilities:

- Community uses RabbitMQ as default queue, MySQL DB (IBM uses Apache Qpid and DB2)

cinder-api is the entry point to OpenStack Volume Service



Core Use Case:

- Accept, validate, authenticate, and distribute incoming REST API requests

Runs As: Controller Service

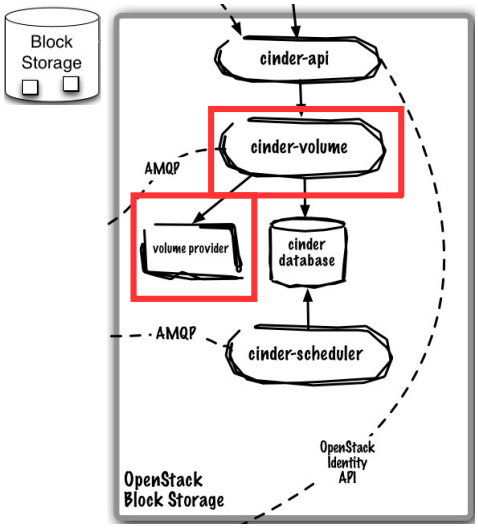
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Volume API
- Robust extensions mechanism to add new capabilities

cinder-volume manages individual block-based volume providers



Core Use Case:

- Manages interactions with single block volume provider

Runs As: Distributed Service

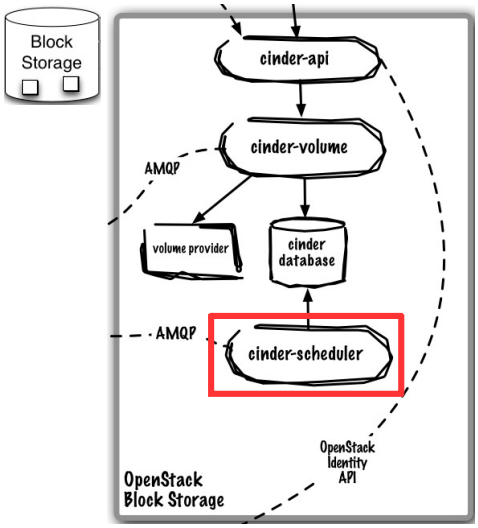
Deployment Considerations:

- Many cinder-volume instances exist in the environment to ensure volume provisioning is always available
- Single cinder-volume is not HA, manage single provider to minimize failure domain

Key Capabilities:

- Create and manage volumes on storage backend
- Expose volumes to physical host (e.g. iSCSI, FC)
- Uses plug-in model to support differing storage systems

cinder-scheduler selects cinder-volume instance to place volume on



Core Use Case:

- Selects cinder-volume service to place volume on

Runs As: Controller Service

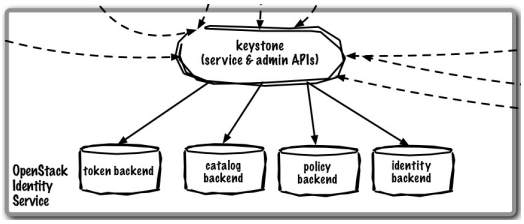
Deployment Considerations:

- Default scheduler is horizontally scalable

Key Capabilities:

- Default scheduled is allocation-based using a series of filters to reduce set of applicable hosts and uses costing functions to provide weight

Identity Service (Keystone) offers project-wide identity, token, service catalog, and policy services designed for integration with existing systems



Core Use Cases:

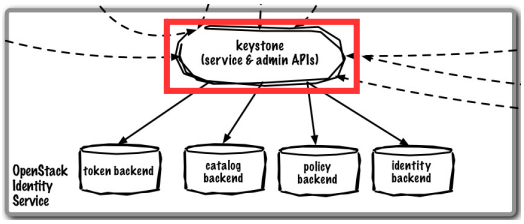
- Installation-wide authentication and authorization to OpenStack services

Key Capabilities:

- Authenticate user / password requests against multiple backends (SQL, LDAP, etc) (Identity Service)
- Validate / manage tokens used after initial username/password verification (Token Service)
- Endpoint registry of available services (Service Catalog)
- Authorize API requests (Policy Service)
- Domain / Project / User model with RBAC for access to compute, storage, networking
- Policy service provides a rule-based authorization engine and the associated rule management interface.

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

keystone service is the entry point for all AuthN and AuthZ in OpenStack



Core Use Case:

- Handle and service all Identity REST API requests

Runs As: Controller Service

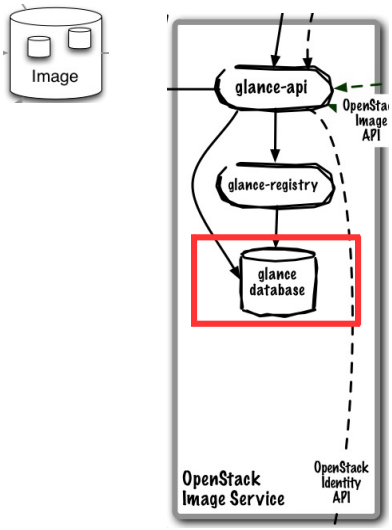
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Identity API
- Pluggable backends for each function: identity, token, catalog, and policy

Glance database persists all image related metadata



Core Use Case:

- Persist image-related metadata

Runs As: Controller Service

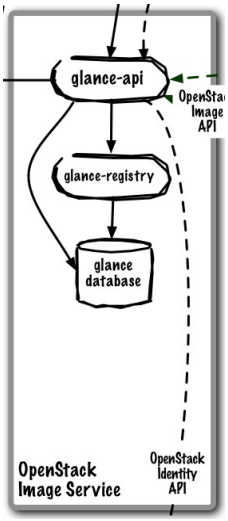
Deployment Considerations:

- Use DB and Queue clustering/HA methods
- Can use same queue/database as Nova

Key Capabilities:

- Persists image-related metadata

Image Service (Glance) provides registration, discovery, and delivery services for virtual disk and server images



Core Use Cases:

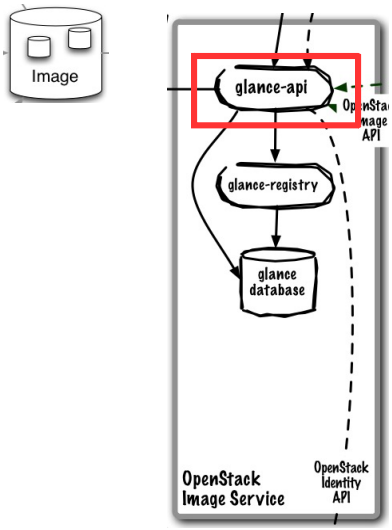
- Administrator registers available guest images
- End-user discovers available guest images
- Deliver image to compute node on provisioning

Key Capabilities:

- Image Registry (storage optional and is delegated to a configurable store)
- Administrators can create base templates from which 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 machines can be backed up quickly
- Supported formats: Raw, Machine (a.k.a. Amazon AMI), VHD (Hyper-V), VDI (VirtualBox), qcow2 (Qemu/KVM), VMDK (VMWare), OVF (VMWare, others)

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

glance-api routes incoming REST API Requests



Core Use Case:

- Routes REST API requests to the appropriate handler

Runs As: Controller Service

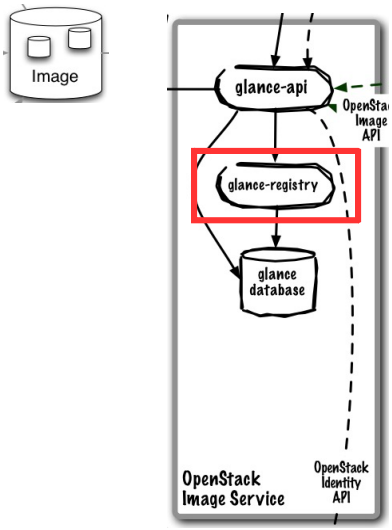
Deployment Considerations:

- Horizontally scalable, start many instances
- Front with load-balancer to present as single endpoint

Key Capabilities:

- APIs supported
  - OpenStack Image API
- Routes requests from clients to registries of image metadata and to its backend stores
- Pluggable image store backends

glance-registry services Image Service API requests



Core Use Case:

- Services Identity REST API requests

Runs As: Controller Service

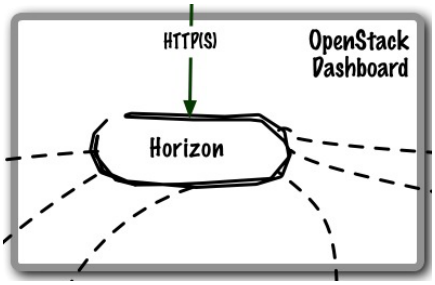
Deployment Considerations:

- (to be determined)

Key Capabilities:

- APIs supported
  - OpenStack Image API

Horizon (Dashboard) enables administrators and users to access, provision, and manage resources through a self-service portal GUI



NOT SHIPPED BY IBM

Core Use Cases:

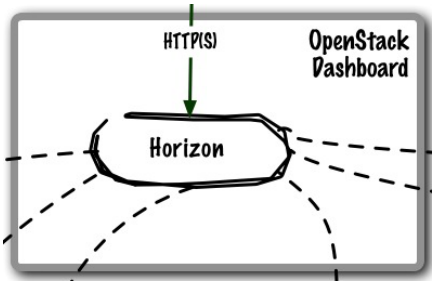
- Self-service portal for compute and object storage
- Cloud administration (users/projects, quotas, etc.)

Key Capabilities:

- Thin wrapper over APIs, no local state
- Registration pattern for applications to hook into
- Out-of-the-box support for all core OpenStack projects.
- Anyone can add a new component as a “first-class citizen”.
- Visual and interaction paradigms are maintained throughout.

Image Source: http://www.solinea.com/2013/04/17/openstack-summit-intro-to-openstack-architecture-grizzly-edition/

horizon is the self-service portal implementation



NOT SHIPPED BY IBM

Core Use Case:

- GUI access to OpenStack APIs

Runs As: Controller Service

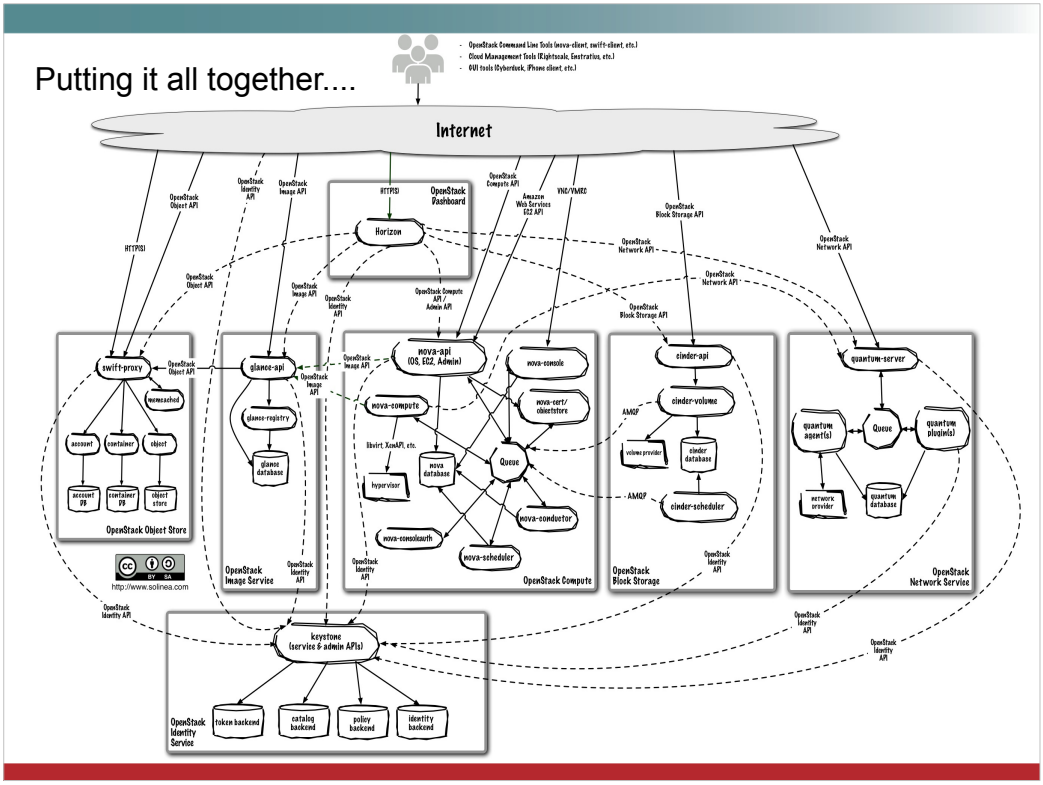
Deployment Considerations:

- (to be determined)

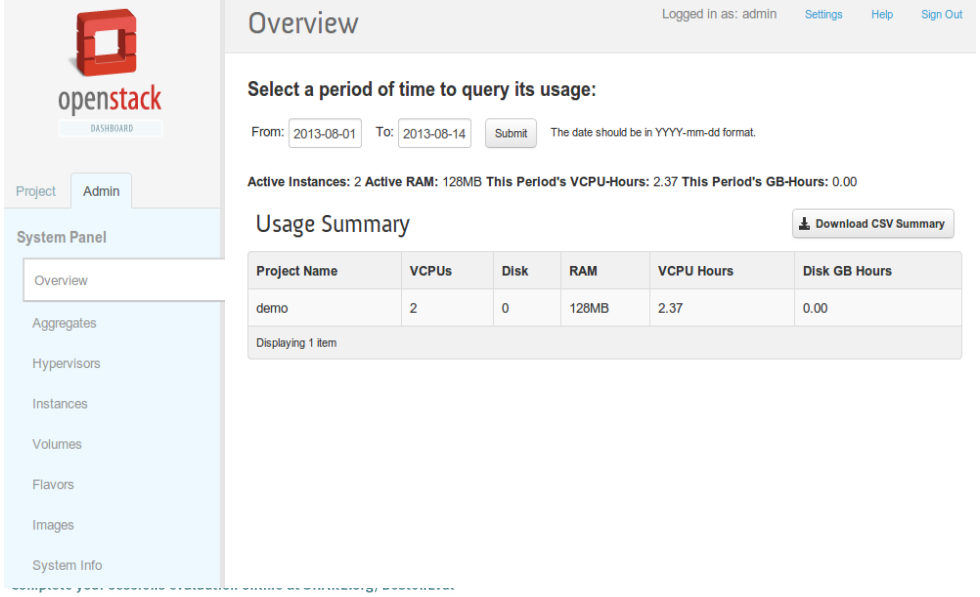
Key Capabilities:

- Provision and manage virtual servers, volumes, and networks
- Create and manage tenants and users

Putting it all together....



Demo


openstack
DASHBOARD
Project
Admin
System Panel
Overview
Aggregates
Hypervisors
Instances
Volumes
Flavors
Images
System Info
Overview
Logged in as: admin
Settings
Help
Sign Out
Select a period of time to query its usage:
From: 2013-08-01
To: 2013-08-14
Submit
The date should be in YYYY-mm-dd format.
Active Instances: 2 Active RAM: 128MB This Period's VCPU-Hours: 2.37 This Period's GB-Hours: 0.00
Usage Summary
Download CSV Summary
Project Name
VCPUs
Disk
RAM
VCPU Hours
Disk GB Hours
demo
2
0
128MB
2.37
0.00
Displaying 1 item