The Abdus Salam International Centre for Theoretical Physics

2494-28

Workshop on High Performance Computing (HPC) Architecture and Applications in the ICTP

14 - 25 October 2013

OpenStack overview

Antonio Messina
University of Zurich
Switzerland

University of Zurich[UZH]

GC3: Grid Computing Competence Center

OpenStack overview

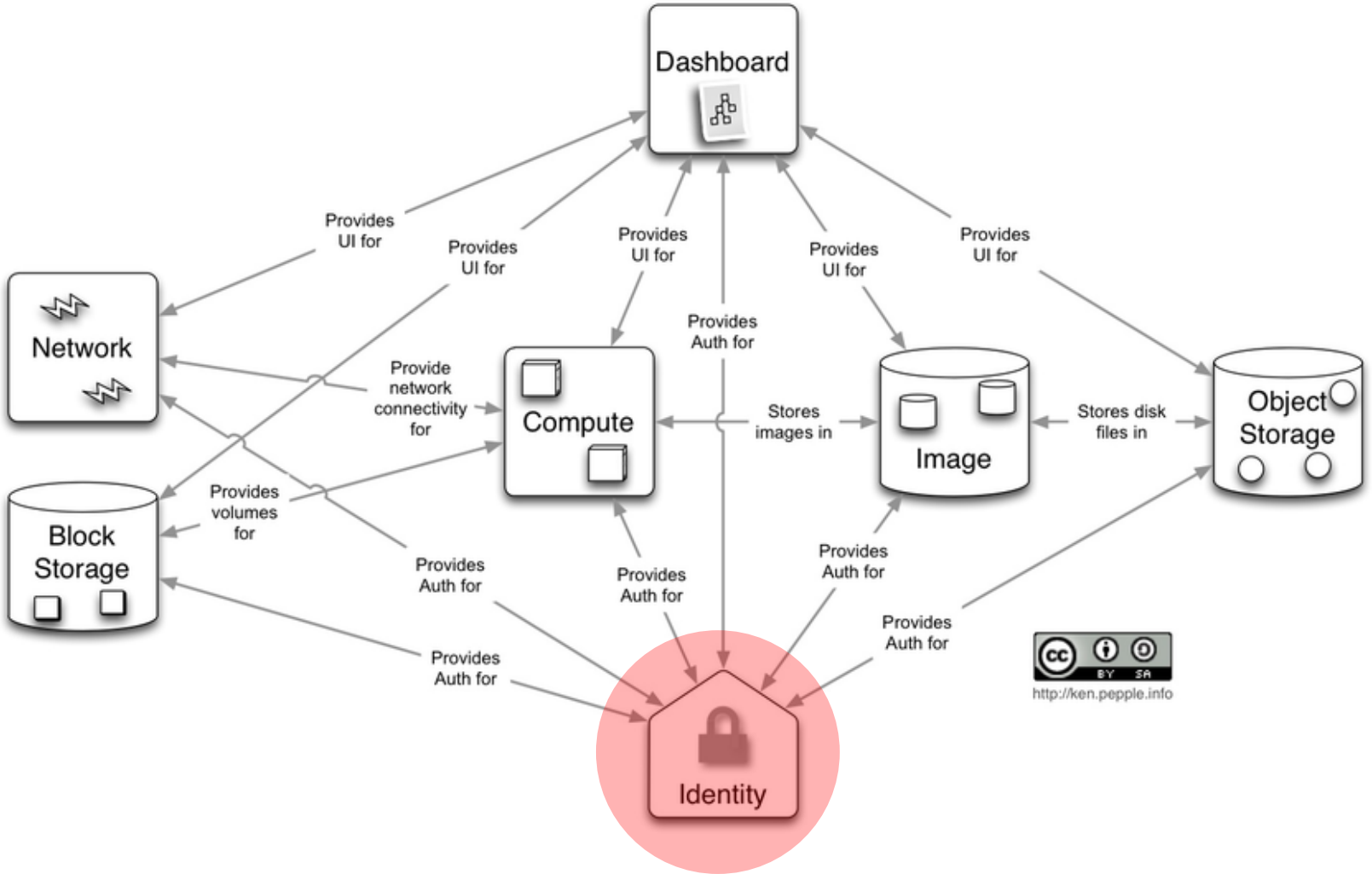
Antonio Messina <antonio.messina@uzh.ch>

Trieste, 24.10.2013

Introduction

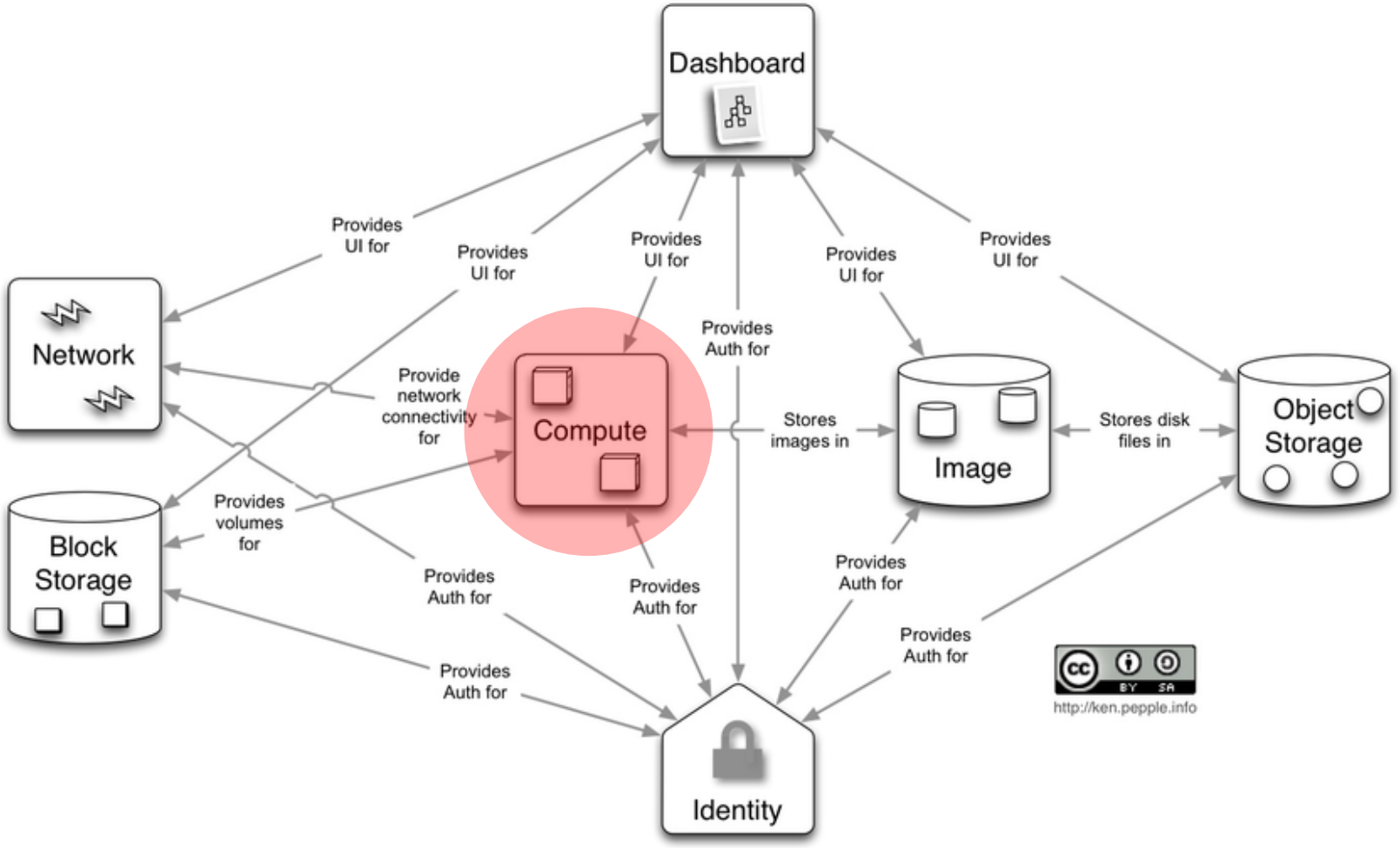
- Open source project (Apache 2.0).
- Very actively developed.
- Many contributors, including commercial companies.
- Biggest contributor is **Rackspace**.
- Releases every 6 months. (current: **Havana**).
- Currently the only real alternative to proprietary clouds.

OpenStack logical view



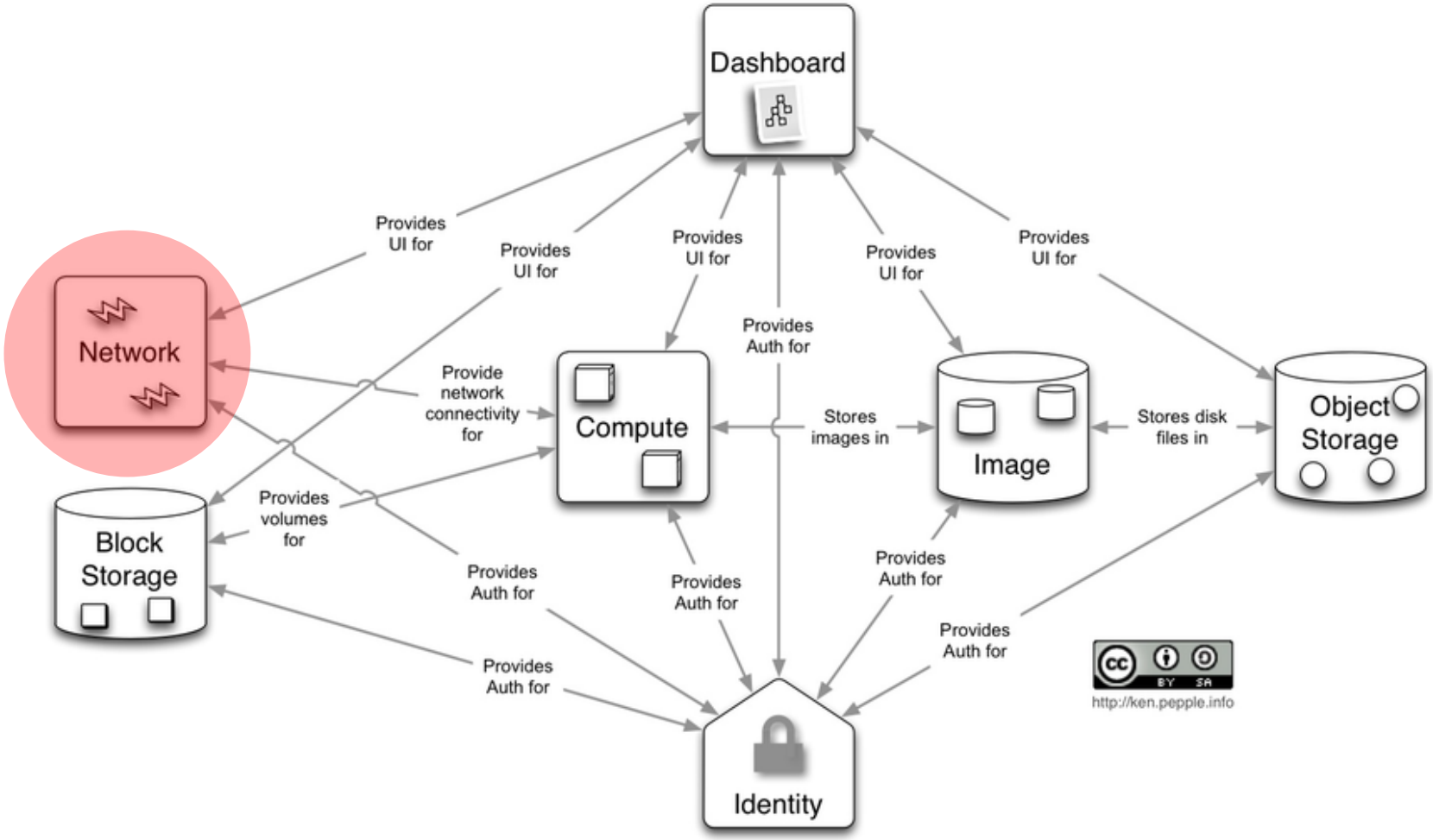
Keystone provides the authentication service

OpenStack logical view



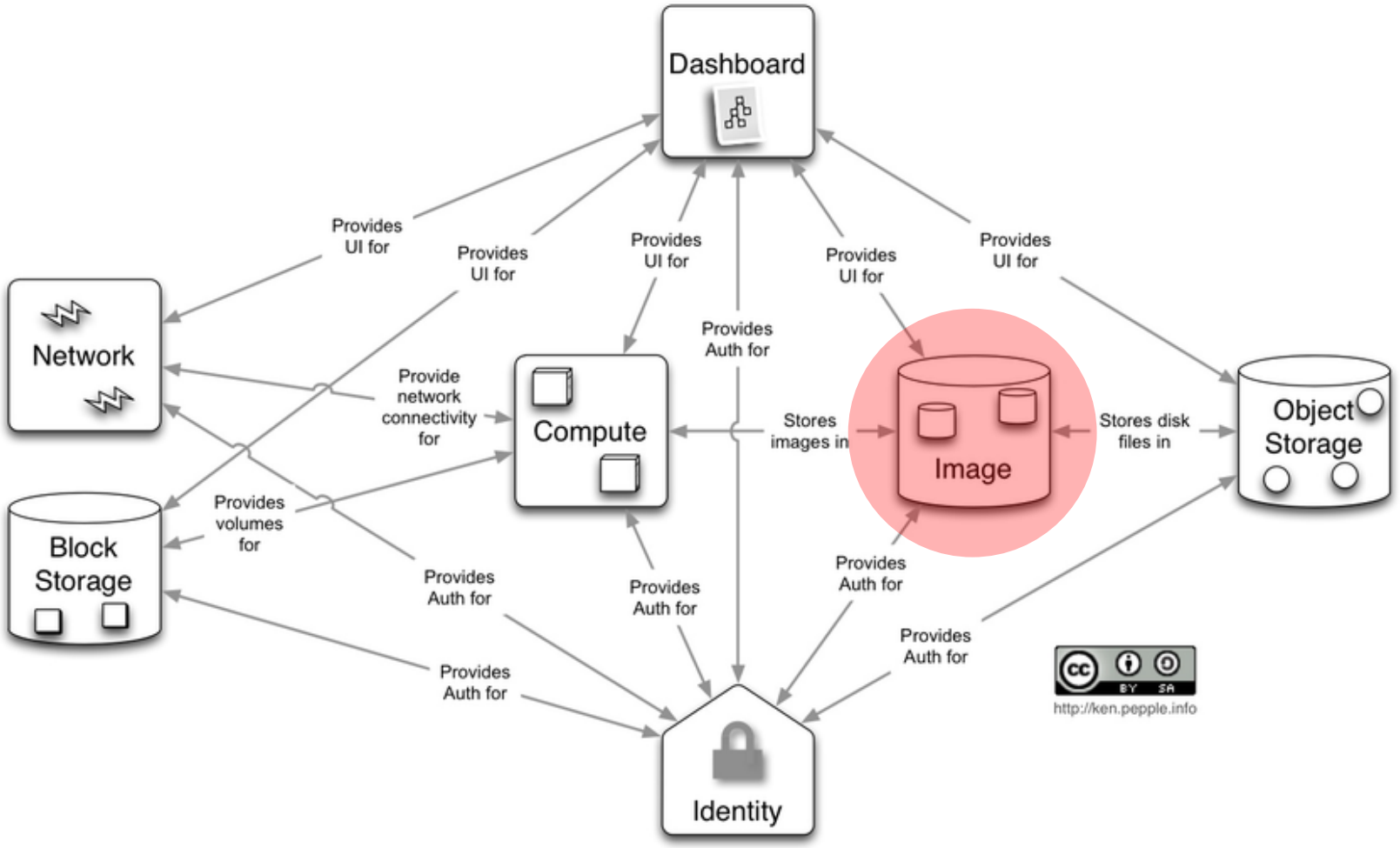
Nova provides computational services

OpenStack logical view



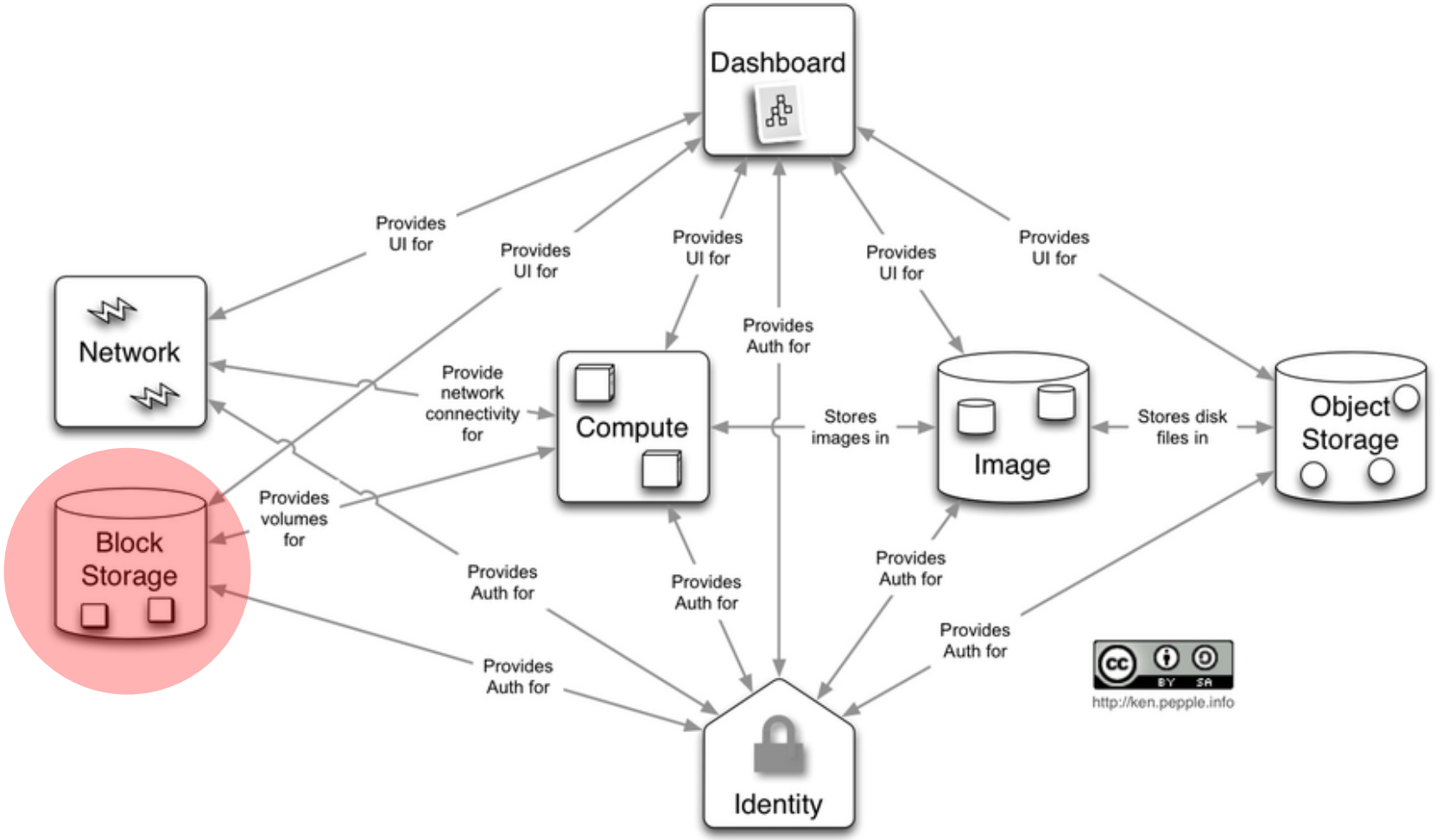
Neutron provides network services

OpenStack logical view



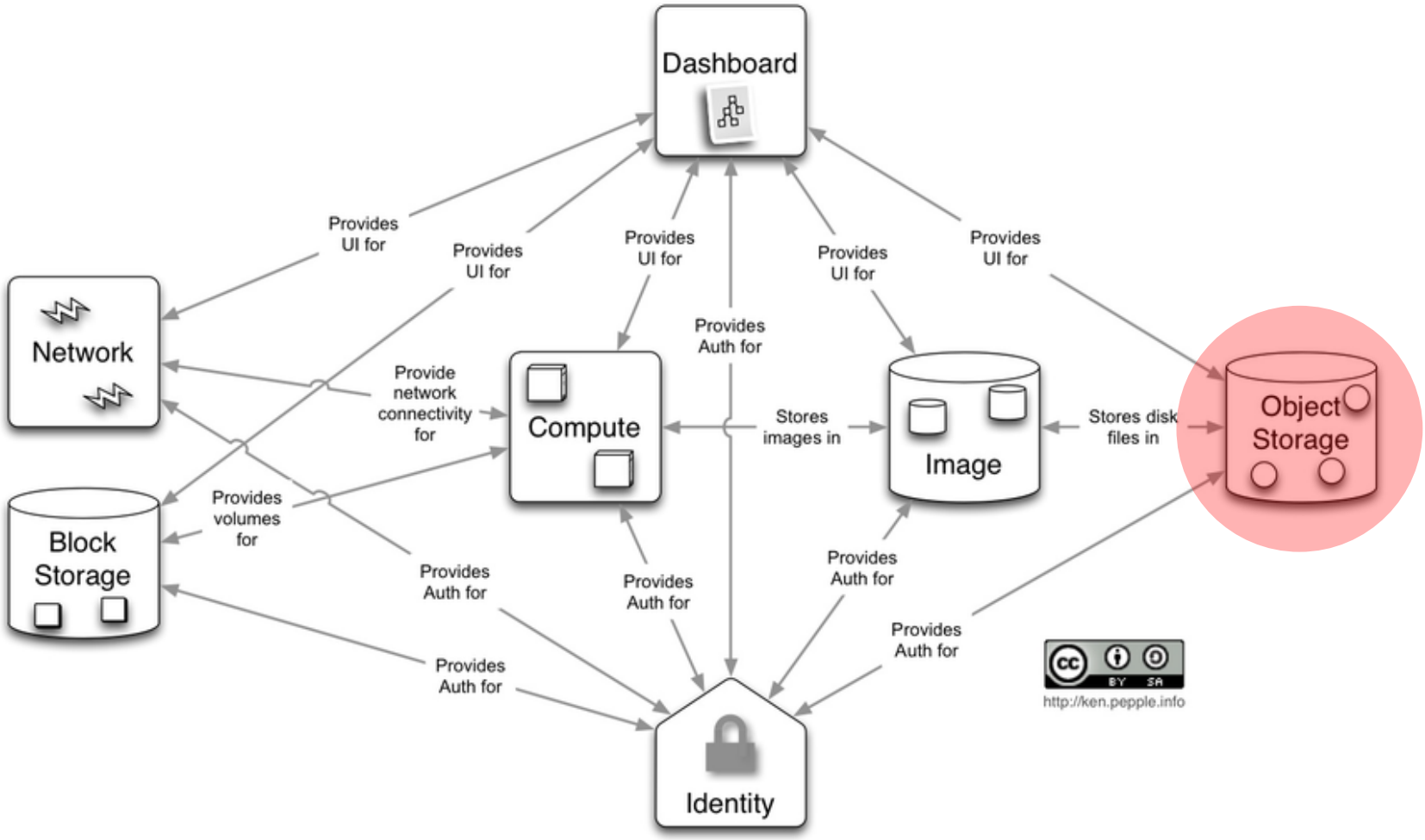
Glance provides image store

OpenStack logical view



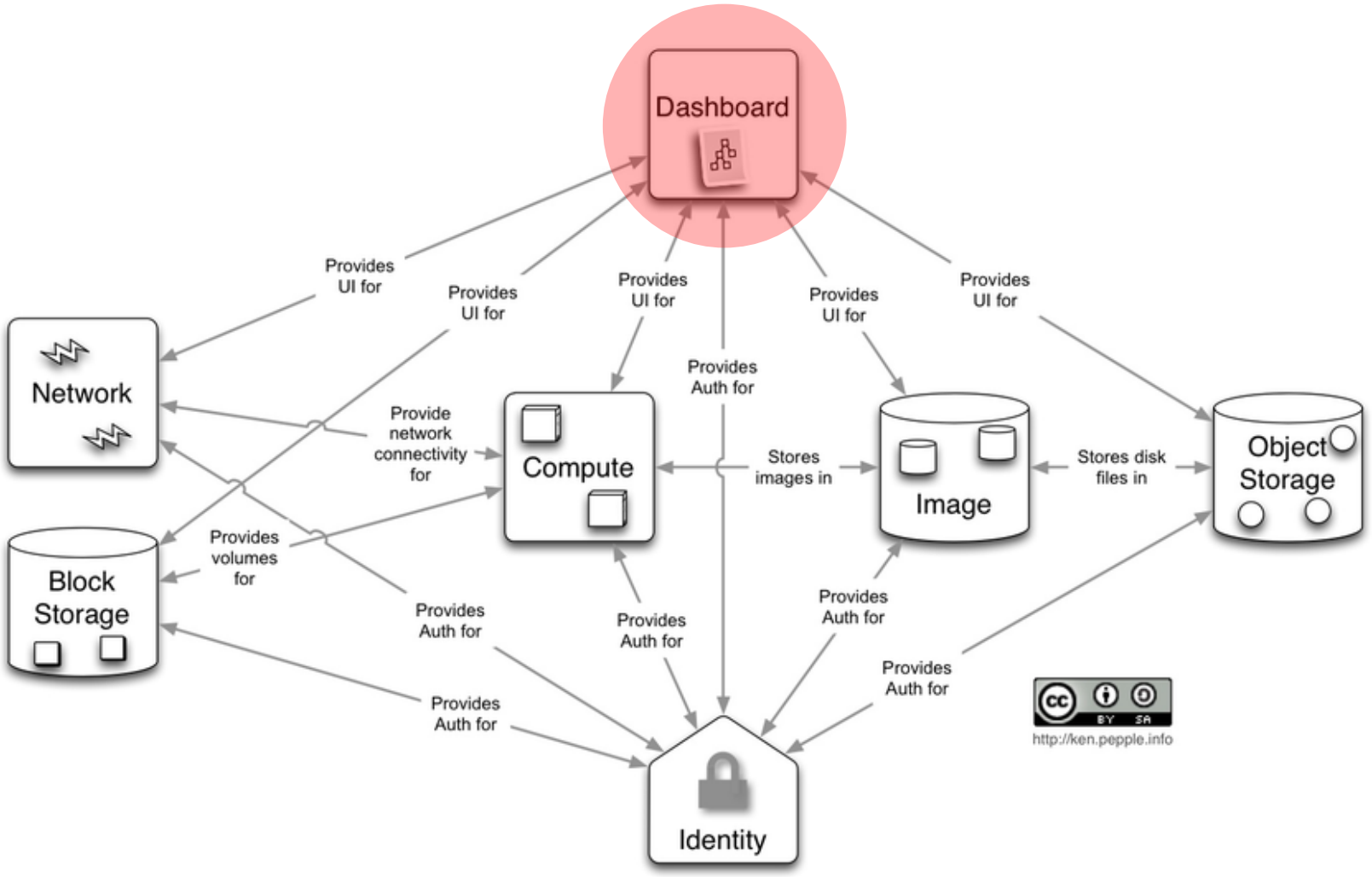
Cinder provides block persistent store

OpenStack logical view



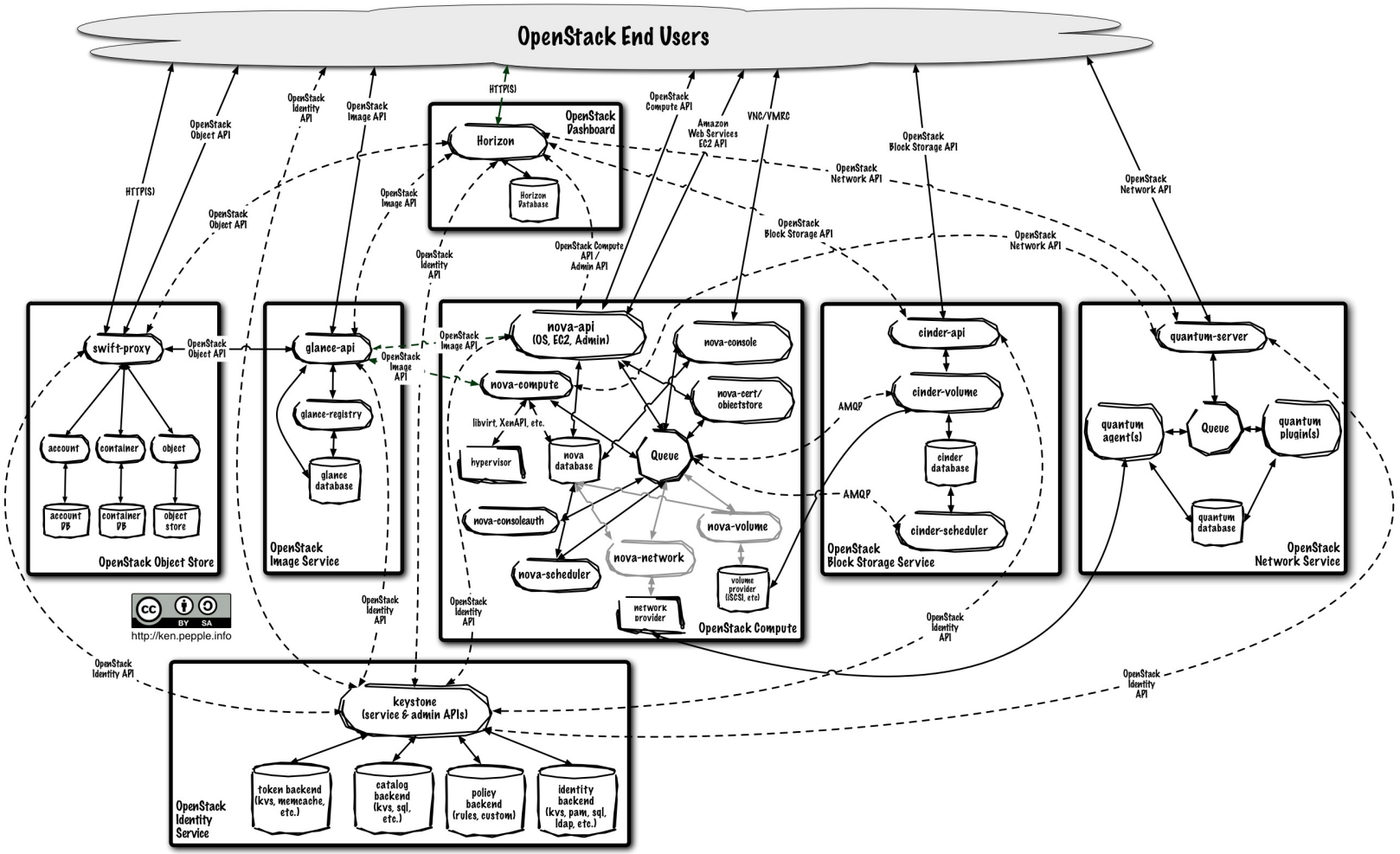
Swift provides object persistent store

OpenStack logical view



Horizon provides web user interface

OpenStack software overview



OpenStack software overview

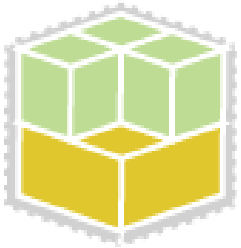
OpenStack Architecture

- Everything is in Python (plus auxiliary shell scripts)
- Build around **independent components**
- **Highly distributed** architecture
- **Intrinsic HA** for OpenStack services (MySQL and RabbitMQ have to be properly configured)
- ***SQL** database used to store persistent data
- **RabbitMQ** used for inter-service communication and notification
- **Web API** services (mostly Django)

keystone - authentication service

- Stores authentication information (*users*, *passwords*, *tokens*, *projects*, *roles*)
- Holds a catalog of available services and their endpoints.
- Can use different backends (SQL database, LDAP)
- It's the entry point for OpenStack API.

nova - compute service



Service responsible of managing virtual instances.

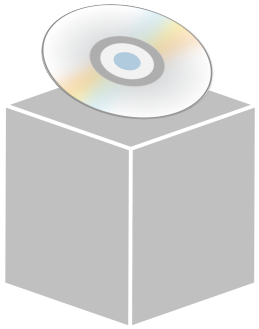
nova-api Web API frontend, accepts requests, validates them and contact other services if needed.

nova-scheduler decides where to start an instance

nova-compute running on each compute node, interacts with the hypervisor and actually starts the vm.

nova-network old, simple, (working) implementation of network service. Does not support Software Defined Networks.

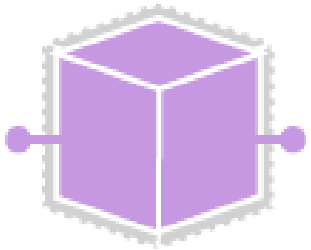
glance - image service



Service responsible of storing image informatins and, optionally, image files.

- Holds information about available images.
- Optionally allow to download and upload images.
- Images can be stored on **different backends** (RDB, S3, Swift, filesystem)

neutron - network service



Service responsible of creating and managing networks. It is supposed to replace **nova-network**.

Still not widely used, but very feature rich.

- L2 and L3 networks.
- Allow creation of multiple networks and subnets.
- Plugin architecture.
- Supports Load Balancer As a Service.
- Integrates with network devices (Cisco, NEC)

cinder - block storage

- Creates and export volumes via iSCSI to the compute node.
- Volumes are mounted **transparently** from the virtual machines.
- Supports **multiple storage backends** (NFS, LVM, Ceph, GlusterFS but also SAN/NAS devices from IBM, NetApp etc. . .)

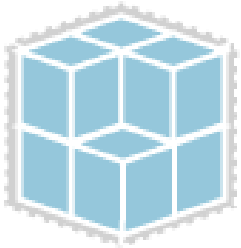
composed of **multiple services**:

cinder-api Web API frontend.

cinder-volume Manages block storage devices. You can have many of these.

cinder-scheduler Decides which cinder-volume has to provide the volume for an instance.

swift - object storage



Object storage distributed service.

- Redundant, scalable object storage on commodity hardware.
- Not a POSIX filesystem.
- Scales horizontally simply by adding new servers.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
   1.1 keystone is contacted and authentication is performed
   1.2 a **token** is saved in the database and returned to the client to be used with later interactions with OpenStack services for this request.
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
   - 2.1 checks via **keystone** the validity of the token
   - 2.2 checks the authorization of the user
   - 2.3 validates parameters and create a new request in the database
   - 2.4 calls the scheduler via queue
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
   - 3.1 reads the request
   - 3.2 find an appropriate host via filtering and weighting
   - 3.3 calls the chosen **nova-compute** host via queue
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance :
   - 4.1 generates a proper configuration for the hypervisor
   - 4.2 get image URI via image id
   - 4.3 download the image
   - 4.4 request to allocate network via queue
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
   - 5.1 gets connection parameters from cinder
   - 5.2 uses iscsi to make the volume available on the local machine
   - 5.3 asks the hypervisor to provision the local volume as virtual volume of the specified virtual machine
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
   - 6.1 allocates a valid private ip
   - 6.2 if requested, it allocates a floating ip
   - 6.3 configures the host as needed (dnsmasq, iptables, Open VSwitch. . .)
   - 6.4 updates the request status
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Life of a virtual machine

1. Authentication is performed either by the web interface **horizon** or **nova** command line tool:
2. **nova-api** is contacted and a new request is created:
3. **nova-scheduler** find an appropriate host
4. **nova-compute** reads the request and start an instance:
5. **nova-compute** contacts **cinder** to provision the volume
6. **neutron/nova-network** configure the network
7. **nova-compute** starts the virtual machine
8. **horizon/nova** poll **nova-api** until the VM is ready.

Notes on installation

- Please, please, please, use a **deployment and configuration manager**. There are many: Puppet, Chef, CFEngine, Ansible, saltstack. . . Just pick the one you like most.
- Do not underestimate the **complexity** of the system.
- Plan in advance, and **plan for failures**.
- Carefully read the Operations Guide.
- Don't tell your security manager, but *security* is probably not the main concern of the developers. . .

Thank you

Questions?