



A Journey in the OpenStack shining world

Stefano Cozzini

eXact lab srl

stefano.cozzini@exact-lab.it

Adapted for MHPC from the original one
presented at INAF ICT meeting

- Introducing Cloud and Open**Stack**
 - What is ?
 - Why and when we need it ?
- Overview of the Open**Stack** services
- ~~Examining Open**Stack** fundamental services~~
 - Basic bricks of an open**Stack** infrastructure
- Open**Stack** for the users
 - A few easy steps to start your infrastructure

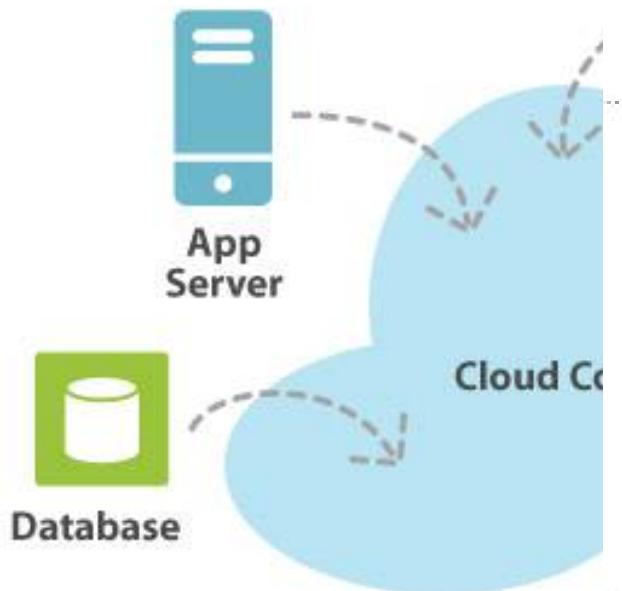
- Where/when to use Open**Stack** and why ?
 - Standard info
 - Some comparison
- ~~Open**Stack** from the sys.adm. P.O.V.~~
 - ~~A more detailed view of the basic components~~
 - ~~Tips to install / manage it~~
- eXact-lab and CNR/IOM cloud infrastructures
 - Our ideas and our efforts

Slides and materials to arrange them are mainly coming from openstack site, presentation/documents around the net and from contribution of a few people

- Antonio Messina (UZH, Zurich)
- Arturo Sandrigo/Francesco De Giorgi (eXact-lab)
- Giuseppe Brandino (eXact-lab)



Cloud Computing 101



Cloud Computing
everything and the kitchen :

TECNOLOGIA



**Ora l' Europa punta
sul “clound computing”**

Ecco l'agenda digitale Ue:
con standard comuni, obblighi
chiari e compatibilità provider
sono previsti 2,5 milioni
di posti di lavoro entro il 2020

- A model of computation and data storage based on “pay as you go” access to “unlimited” remote data center capabilities
- A cloud infrastructure provides a framework to manage scalable, reliable, on-demand access to applications
- Cloud services provide the “invisible” backend to many of our mobile applications
- High level of elasticity in consumption
- Historical roots in today’s Internet apps
 - Search, email, social networks
 - File storage (Live Mesh, Mobile Me, Flickr, ...)

When I want pizza..



At home
with my wife

At home without
my wife
And without cash

At home without
my wife
With cash

PIZZERIA !

You Manage

Dining Table
Soda
Electric/Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese

You Manage

Dining Table
Soda
Electric/Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese

You Manage

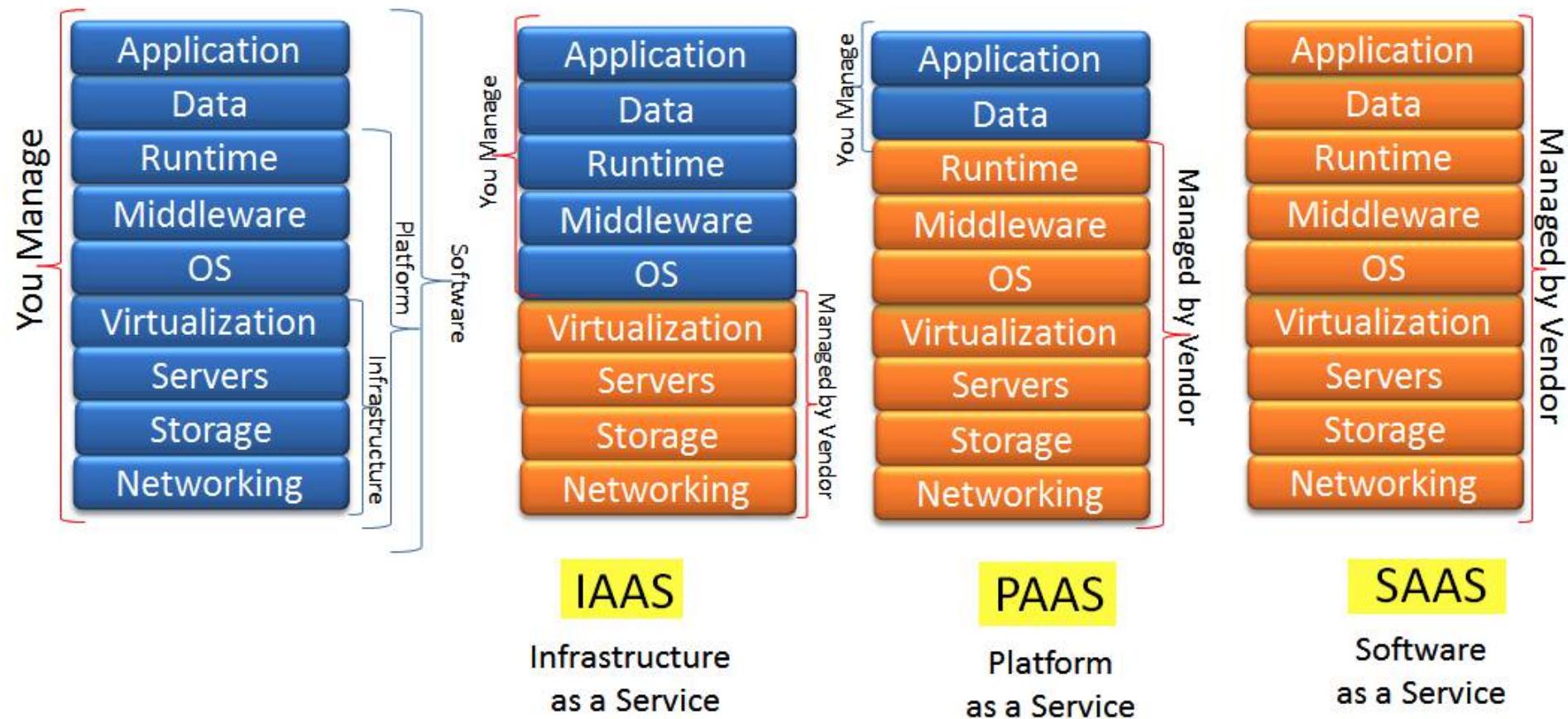
Dining Table
Soda
Electric/Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese

Managed by Vendor

Dining Table
Soda
Electric/Gas
Oven
Fire
Pizza Dough
Tomato Sauce
Toppings
Cheese

Managed by Vendor

When I manage IT stuff...



Taken and Slightly changed from
<https://cloudcelebrity.wordpress.com/2011/11/22/introduction-to-cloud-services-iaas-paas-saas/>

- A cloud infrastructure is:

An infrastructure to provide users with the most flexible way to allocate computational power and storage space (and any other IT services)

quickly provide a customized IT infrastructure
integrate infrastructure provisioning in your data-analysis.

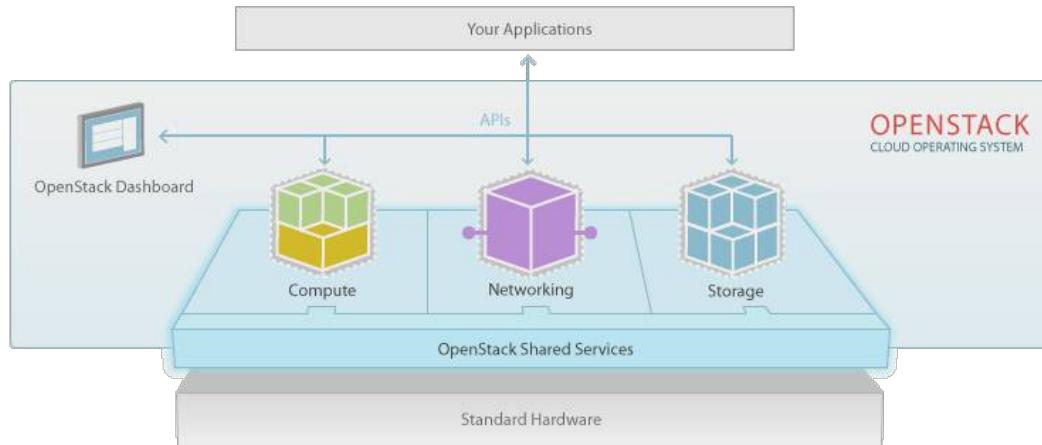
“You are providing cloud services when your users do not bother your sys. Adm. to setup his/her virtual machine..”

“The difference is that a true cloud provides self-service capability, elasticity, automated management, scalability and pay-as you go service that is not inherent in virtualization.”



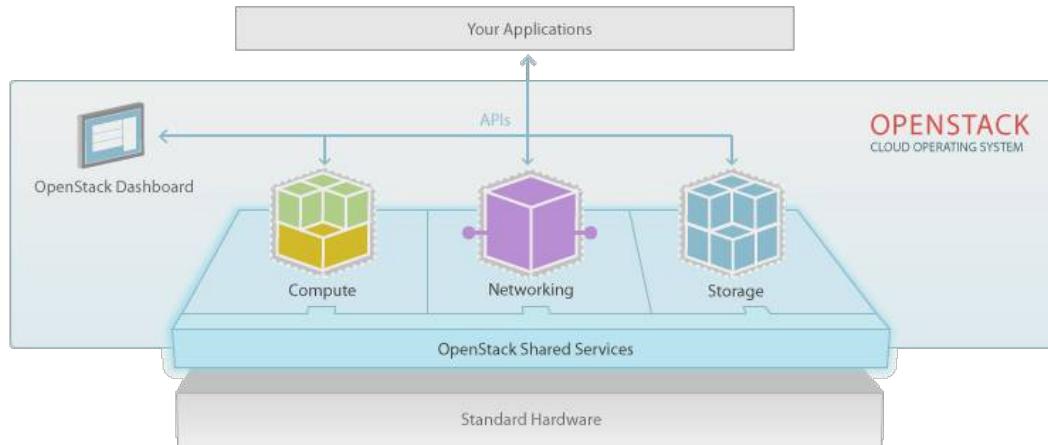
Introducing Open**Stack**

What is OpenStack?



Open**Stack** is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

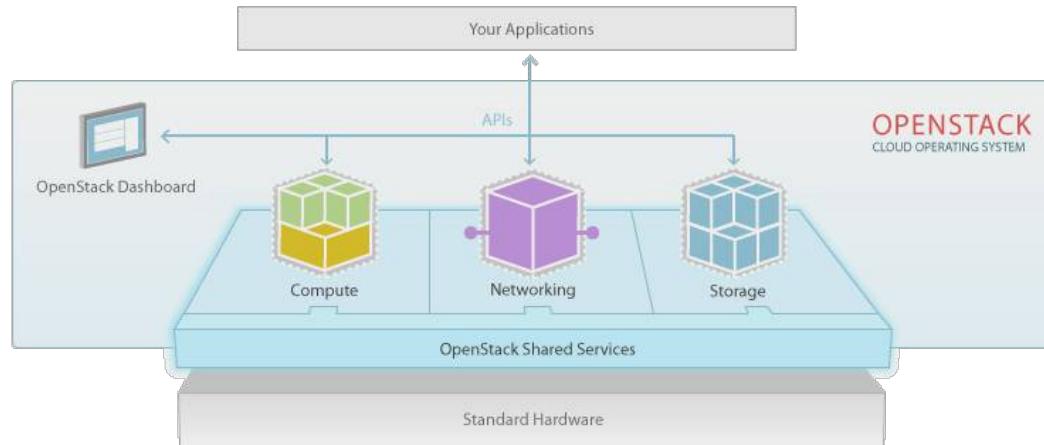
What is OpenStack?



Open**Stack** is a **cloud operating system** that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

It is a complex piece of software plus a buzzword

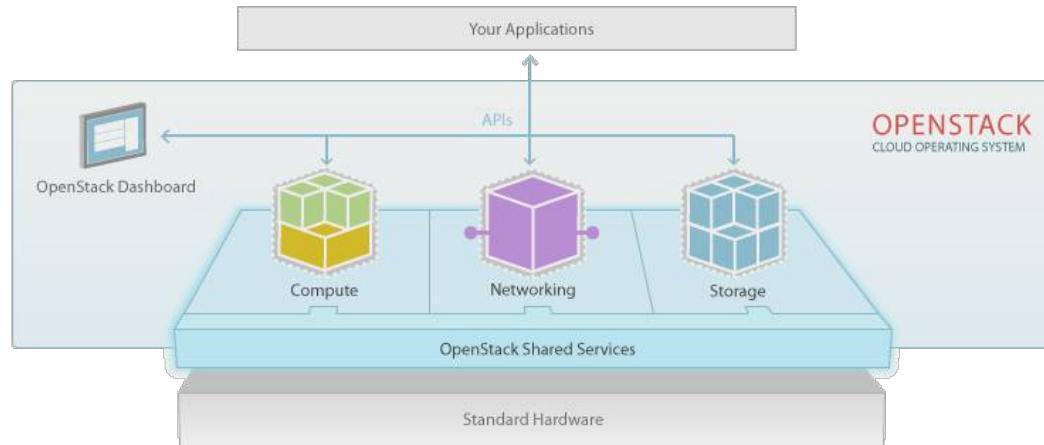
What is OpenStack?



Open**Stack** is a cloud operating system that controls **large pools** of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

It is scalable (thousands of nodes)

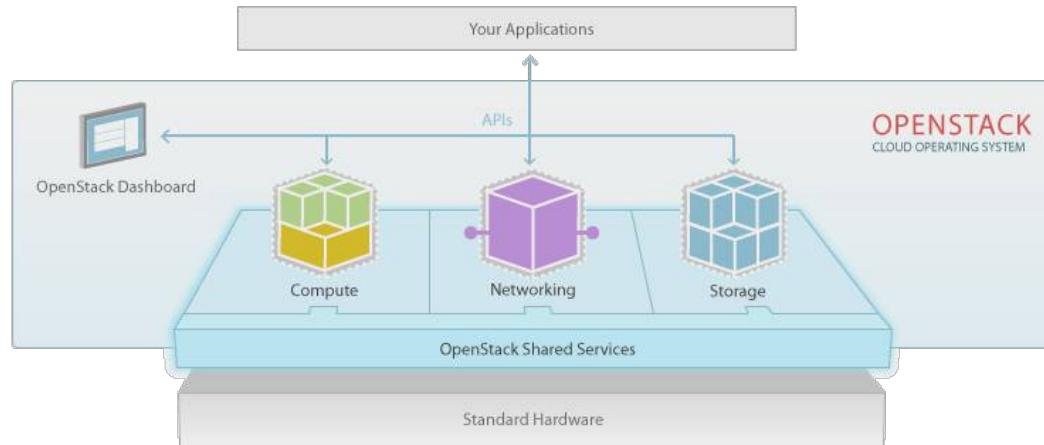
What is OpenStack?



Open**Stack** is a cloud operating system that controls large pools of **compute**, **storage**, and **networking resources** throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web Interface.

Not only CPUs but also storage and network

What is OpenStack?



Open**Stack** is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a **dashboard** that gives administrators control while **empowering their users** to provision resources through a web interface.

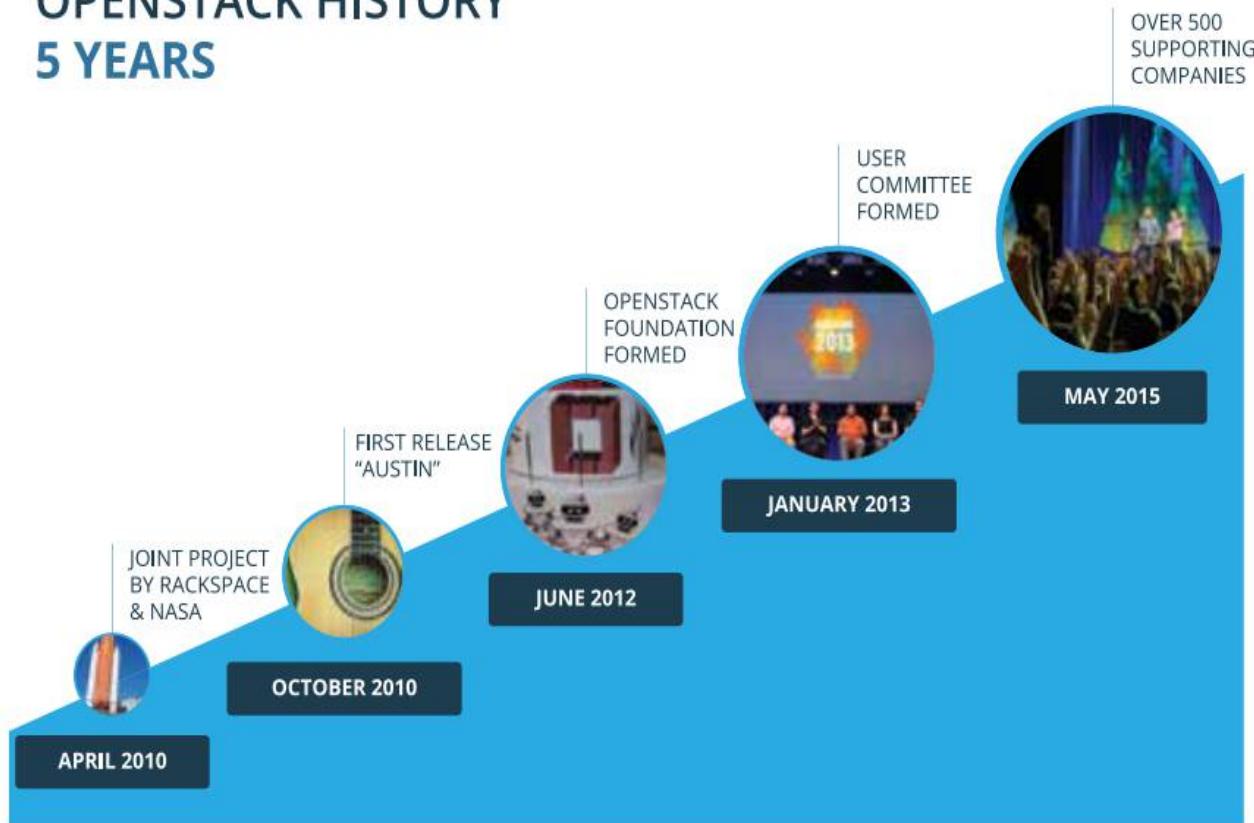
User-centric

- The OpenStack project is an open source cloud computing platform that supports all types of cloud environments.
- The project aims for simple implementation, massive scalability, and a rich set of features.
- Cloud computing experts from around the world contribute to the project.
- OpenStack provides an **Infrastructure-as-a-Service (IaaS)** solution through a variety of complementary services.
- Each service offers an application programming interface (API) that facilitates this integration

- Virtual machines (VMs) on demand
 - provisioning
 - snapshotting
- Networks
- Volumes
- Storage for VMs and arbitrary files
- Multi-tenancy
 - quotas for different projects, users
 - user can be associated with multiple projects

OpenStack history

OPENSTACK HISTORY 5 YEARS



Release Series

OpenStack is developed and released around 6-month cycles. After the initial release, additional stable point releases will be released in each release series. You can find the detail of the various release series here on their series page. Subscribe to the [combined release calendar](#) for continual updates.

Series	Status	Initial Release Date	EOL Date
<u>Queens</u>	<i>Future</i>	TBD	TBD
<u>Pike</u>	<i>Future</i>	TBD	TBD
<u>Ocata</u>	<u>Under Development</u>	2017-02-22 (planned)	TBD
<u>Newton</u>	Current stable release, security-supported	2016-10-06	TBD
<u>Mitaka</u>	Security-supported	2016-04-07	2017-04-10
<u>Liberty</u>	EOL	2015-10-15	2016-11-17
<u>Kilo</u>	EOL	2015-04-30	2016-05-02
<u>Juno</u>	EOL	2014-10-16	2015-12-07

- written in Python (plus auxiliary shell scripts)
- Built around **independent components**
- Highly distributed architecture
 - designed for very big installations
- intrinsic HA of most OpenStack services
 - (MySQL and RabbitMQ have to be properly configured)
- *SQL database used to store persistent data
- RabbitMQ used for RPC and notification
- **RESTful APIs** for all the services

- creation is done via Web GUI, CLI or network APIs
 - => it can be scripted
- Actual provisioning of the VMs can be delegated to the user.
 - => no need to fill up a form or open an issue to create a VM.
- Time to provision a VM can be very short (depending on the type of setup!!)
 - 10 seconds to create
 - 20- 50 seconds to login (depending on the image)

OpenStack believes in open source, open design, open development, all in an open community so anyone can participate.

The long-term vision for OpenStack is to produce a ubiquitous open source cloud computing platform that meets the needs of public and private cloud providers regardless of size.

OpenStack services control large pools of compute, storage, and networking resources throughout a data center

- **Open development model:** Code reviews and roadmaps are public. The code for OpenStack is freely available under the Apache 2.0 license.
- **Open design process:** Every six months the development community holds a design summit to gather requirements and write specifications for the upcoming release. The summit is open to the public and attendees include users, operators, developers, and upstream project personnel.
- **Open community:** OpenStack is dedicated to producing a healthy, vibrant, and active developer and user community. Most decisions are made by consensus. All processes are documented, open and transparent.

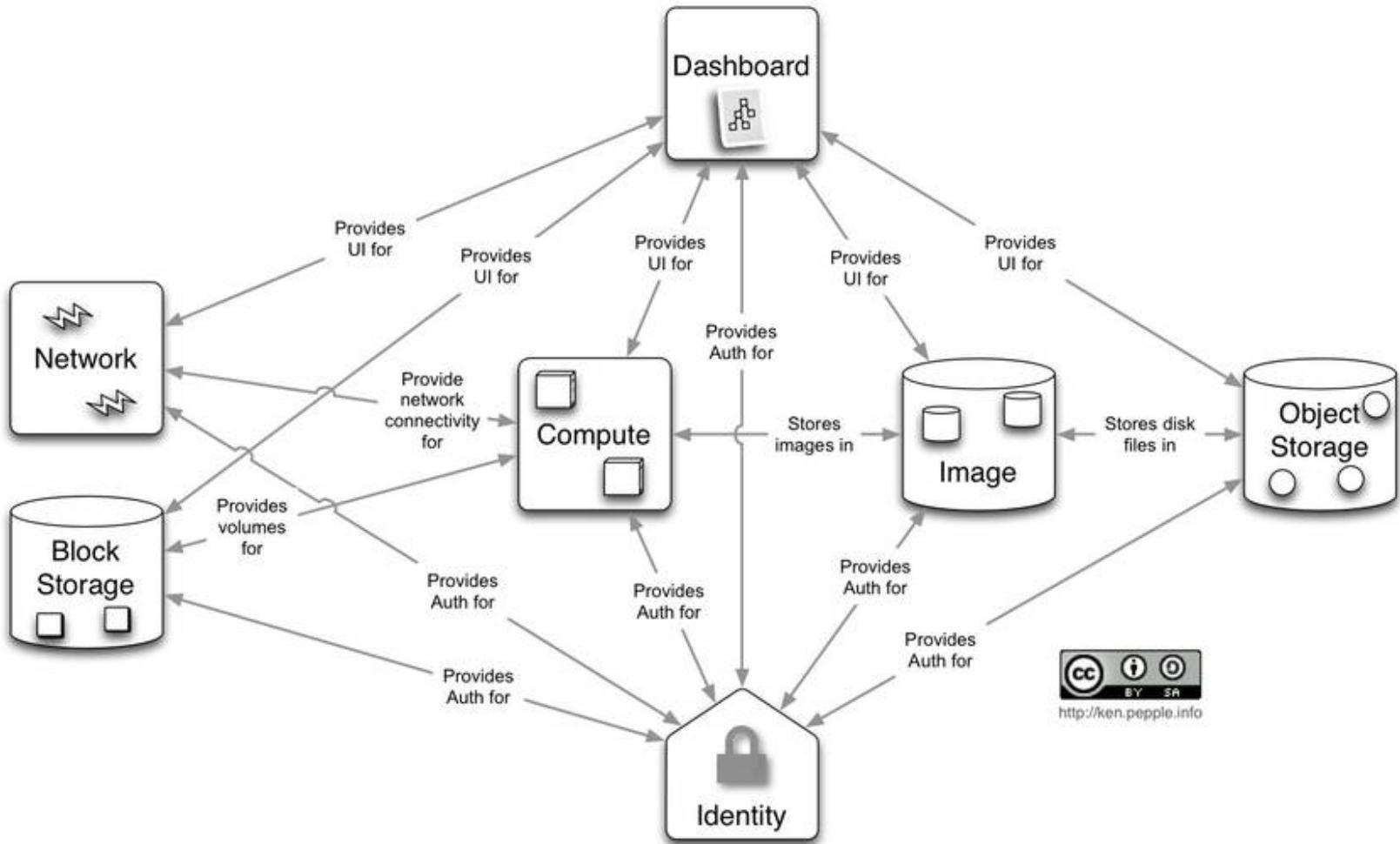
- Created by a large growing community
 - 24,000 people
 - 495 supporting companies
- over 20 million lines of code
- One recent ("Kilo") release stats:
 - 1,500 developers
 - 170 different companies
 - 20,000 patches

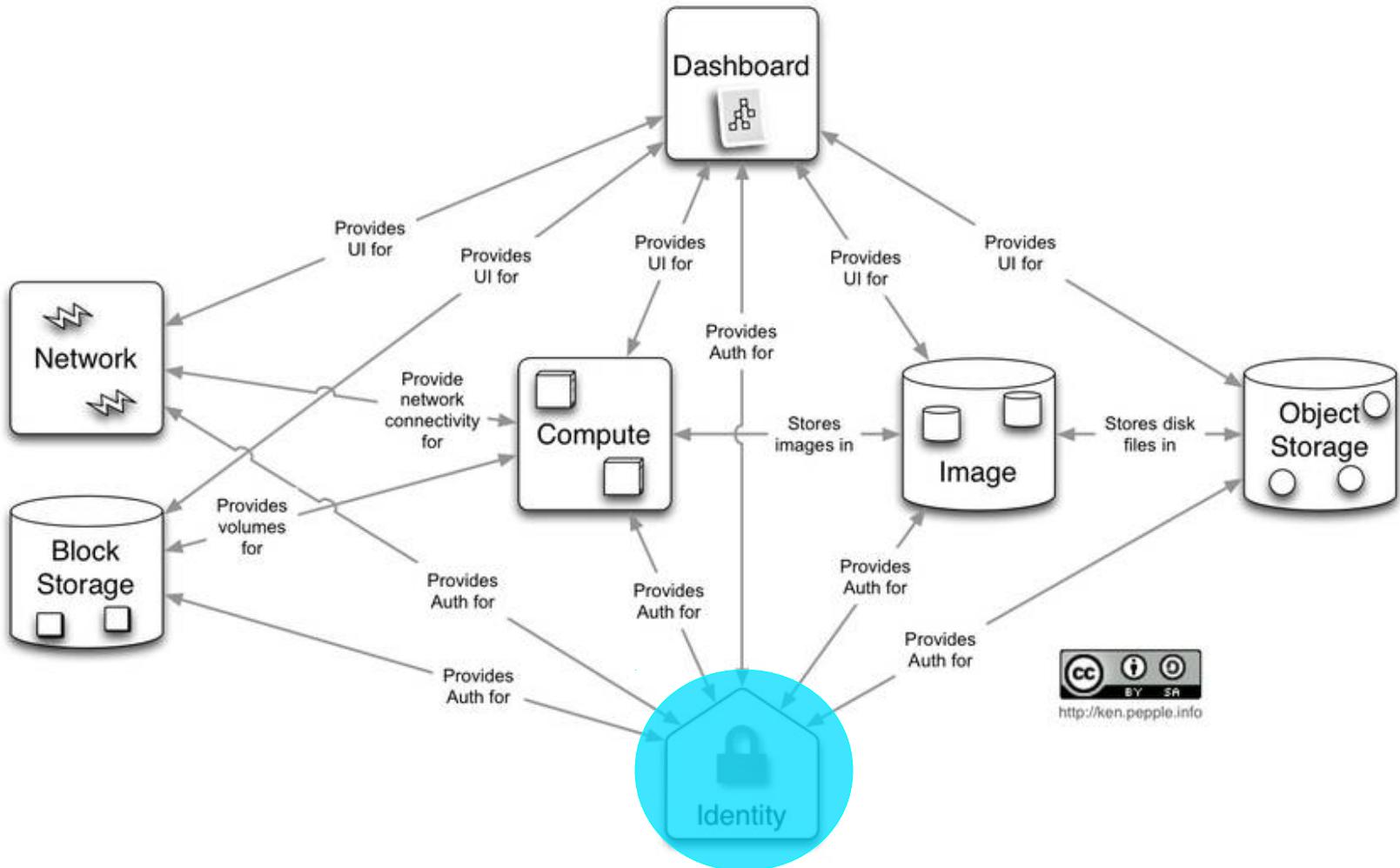
- open source software for building clouds
- release cycle is every 6 months
 - Last one (Newton) out on 15/10/16
- an umbrella over multiple independent programs (components)
- All components talk via RESTful API
- Most components have dedicated DB (SQL) and Message Queuing system
- some talk to 3rd party components using their native APIs



An overview of Open**Stack** services

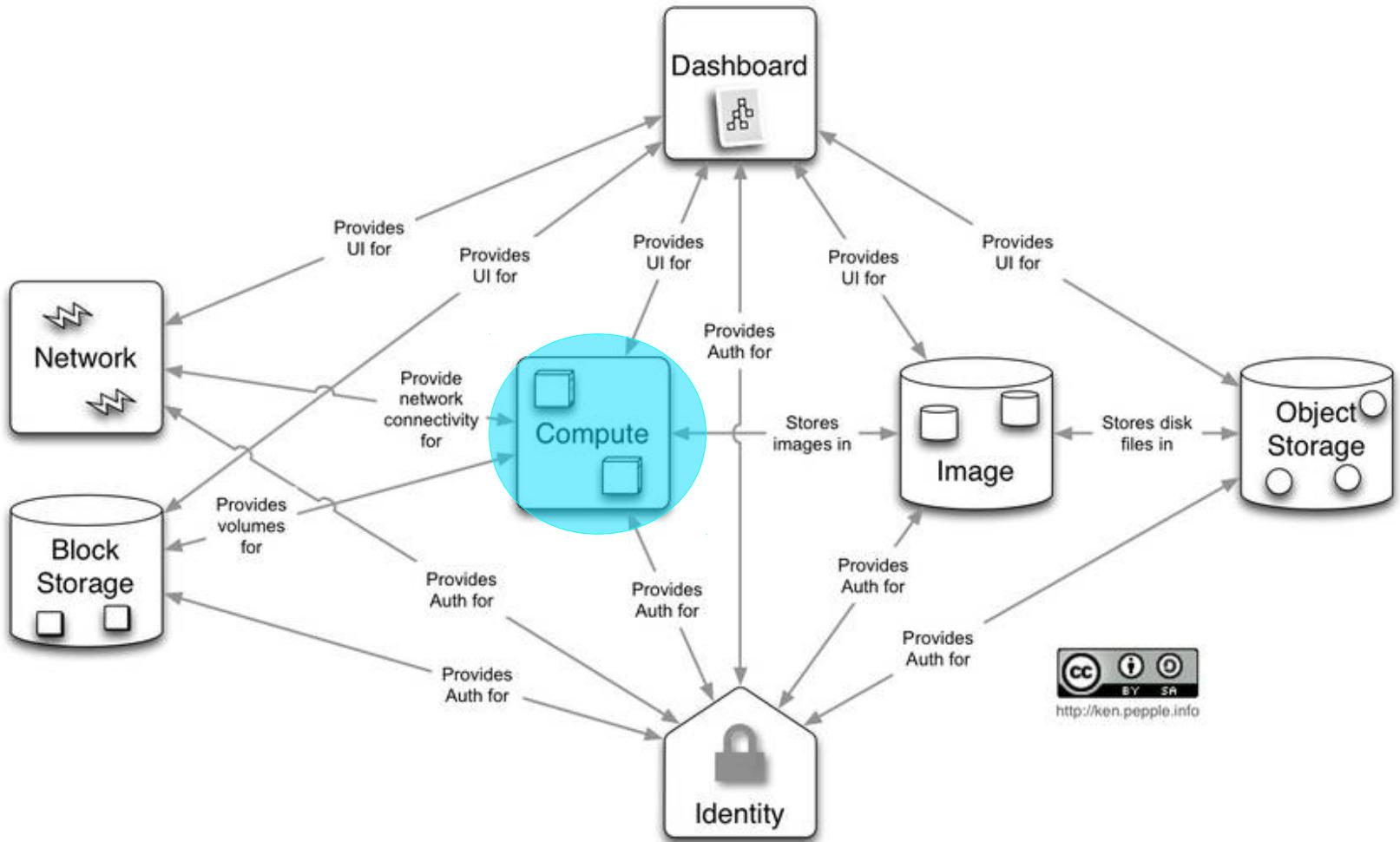
Logical view



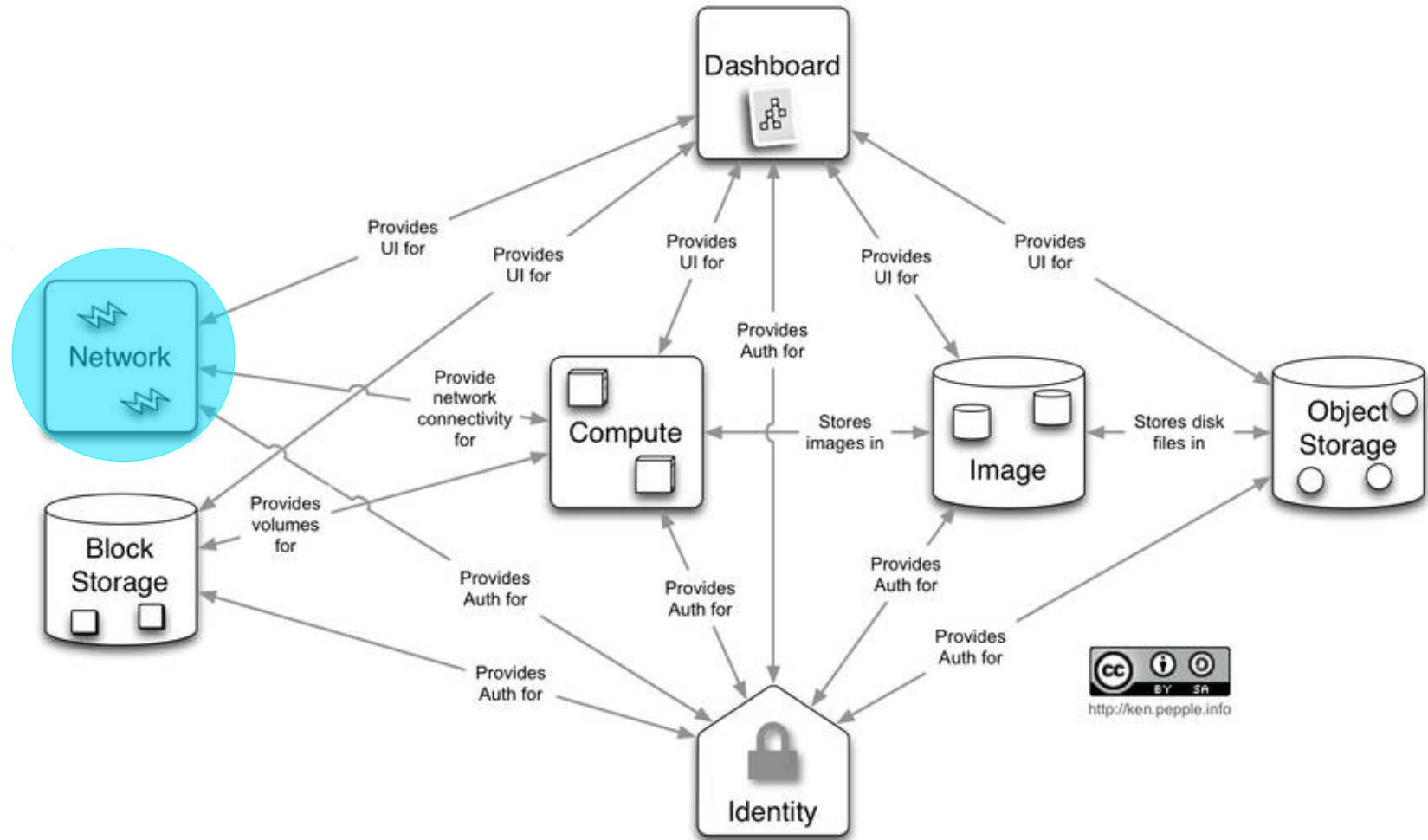


<http://ken.pepple.info>

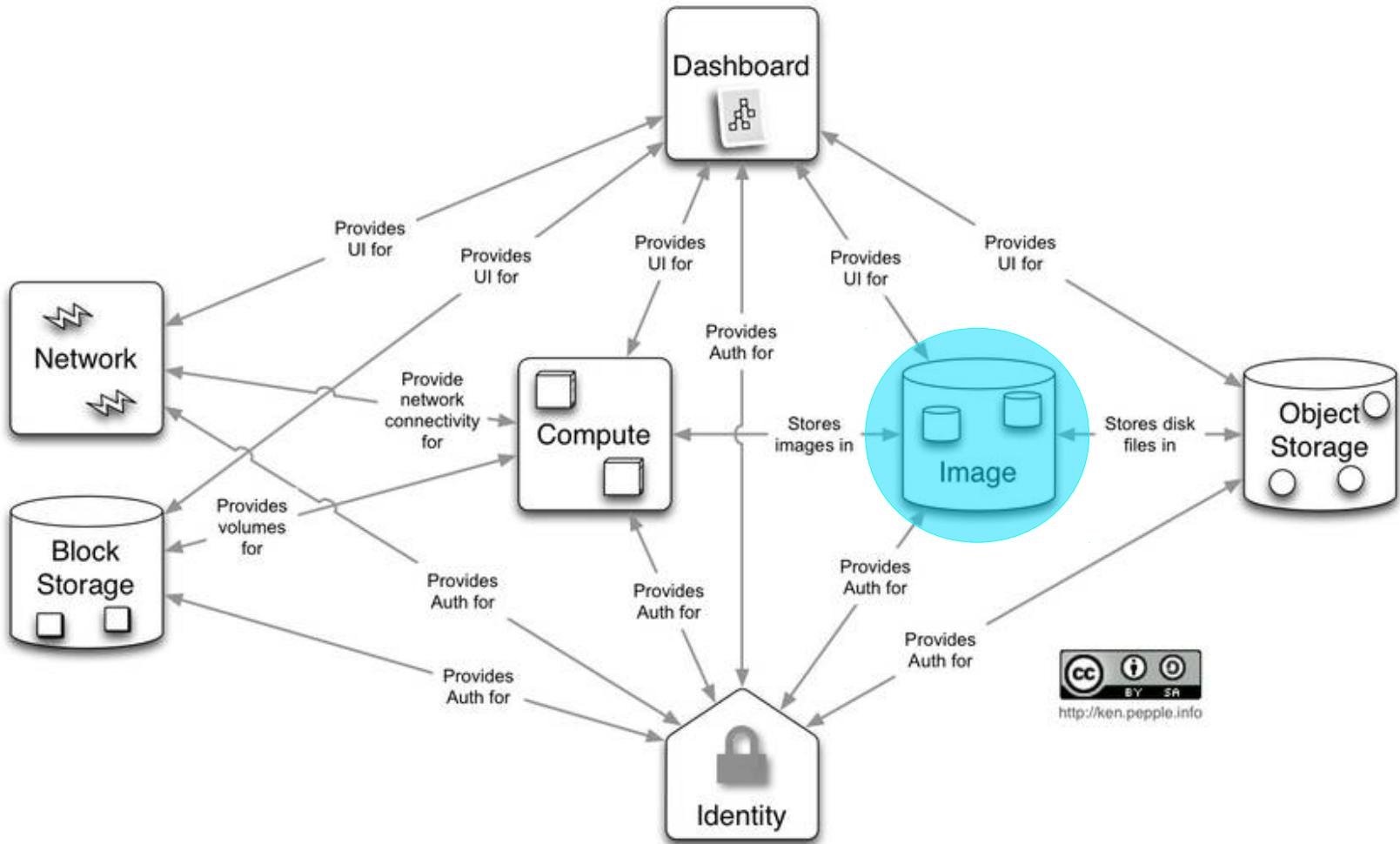
KEYSTONE: authentication service



NOVA: computational services

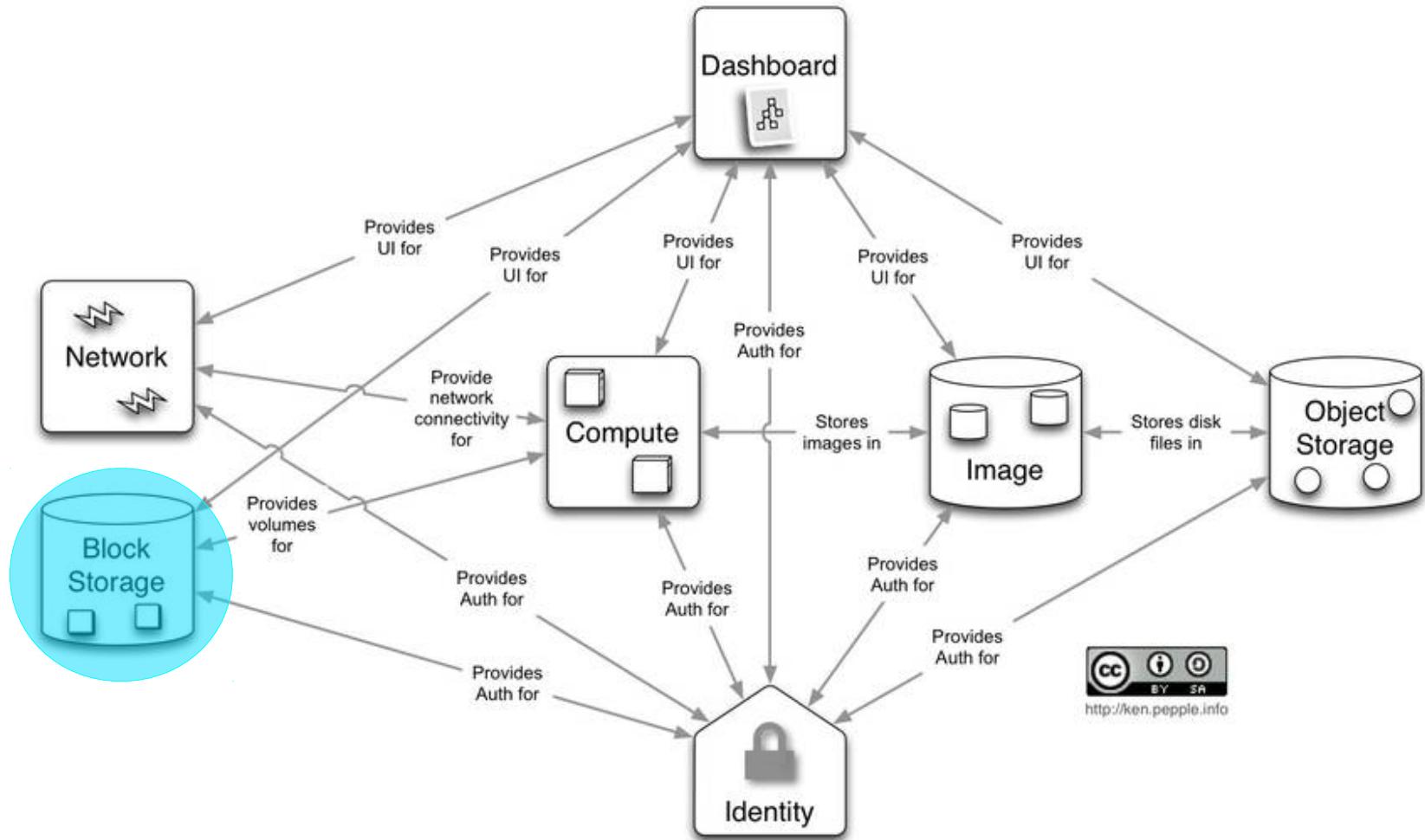


NEUTRON: network services

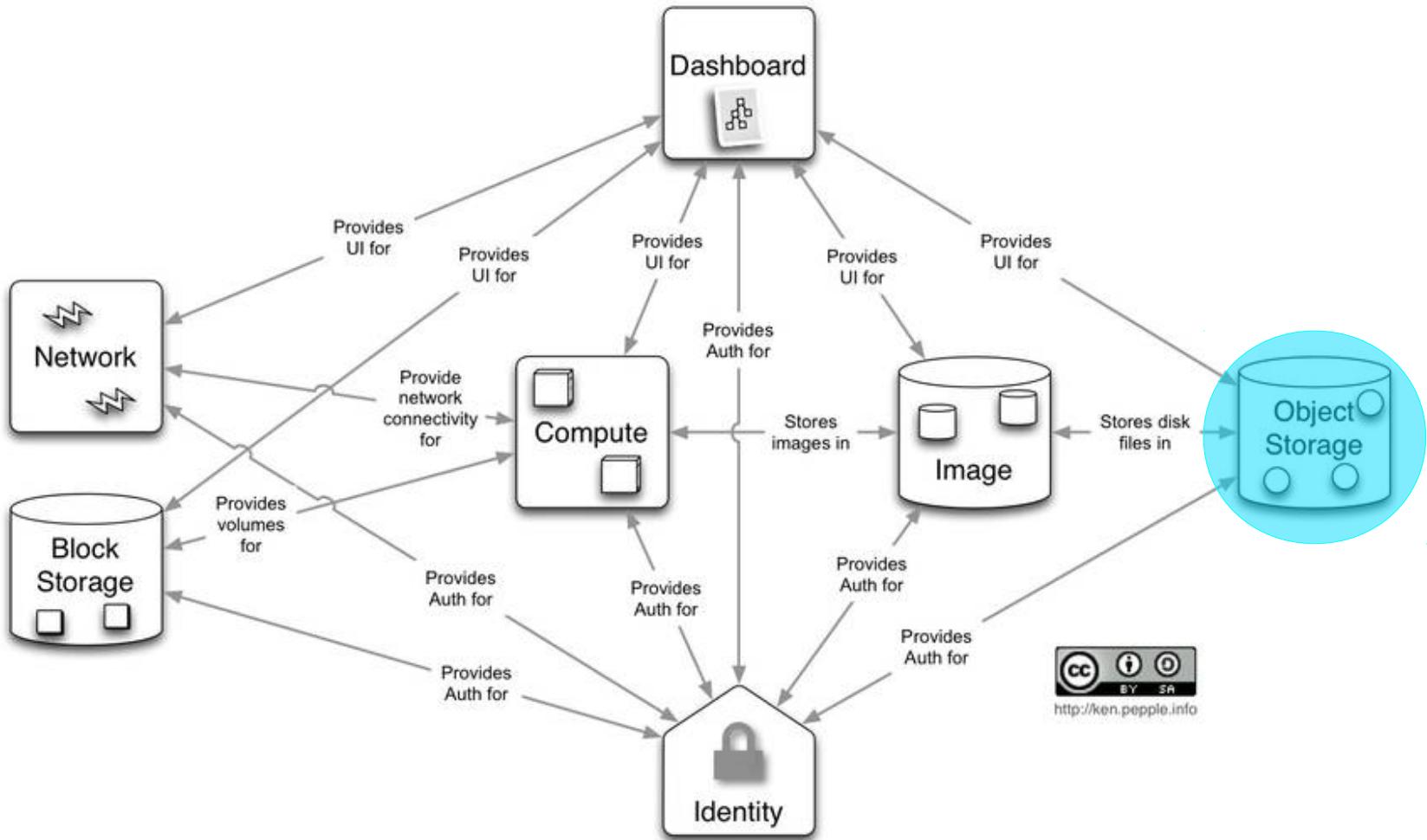


<http://ken.pepple.info>

GLANCE: image services

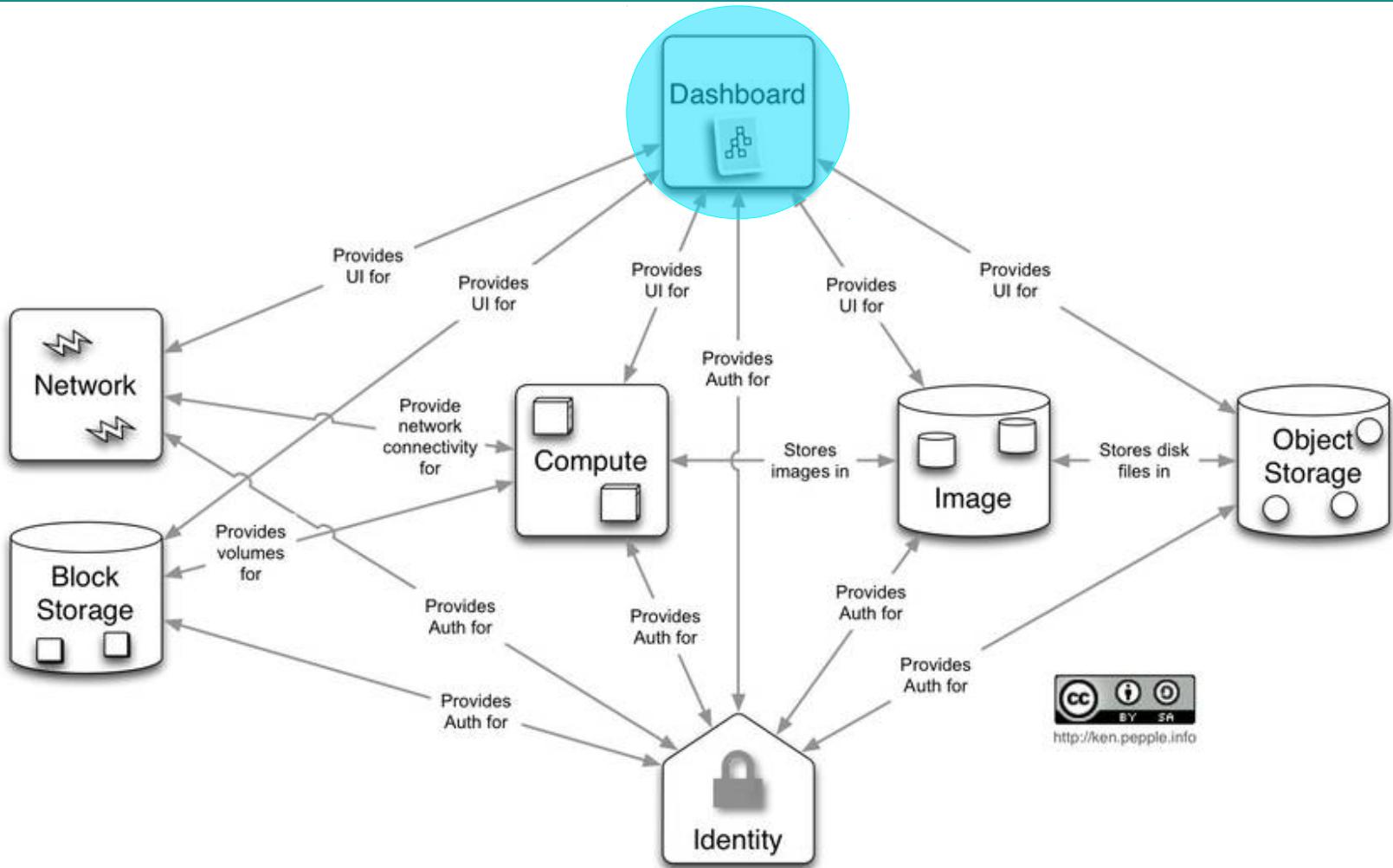


CINDER: block storage services



SWIFT: object storage services

Logical view



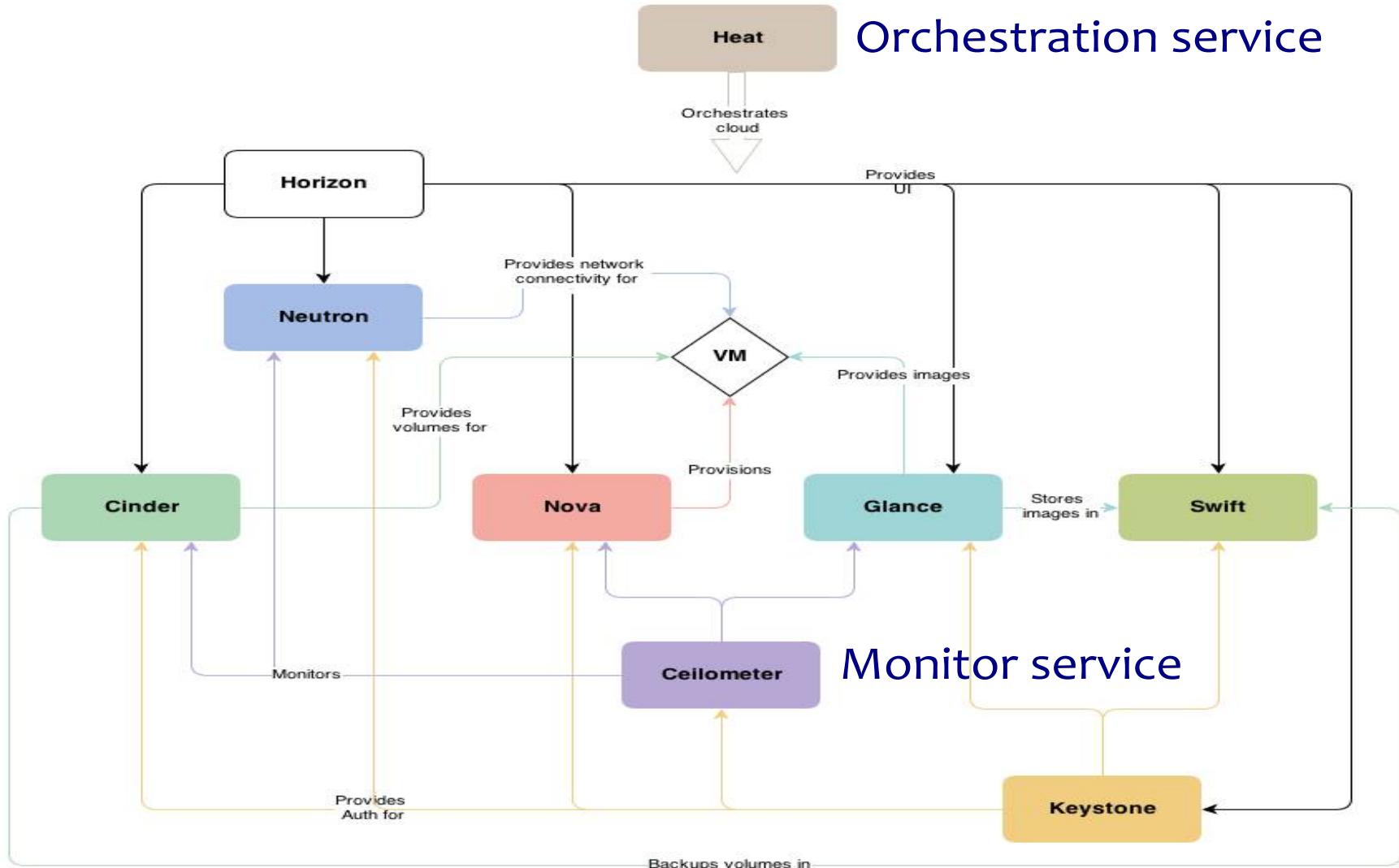
HORIZON: web user interface

List of services so far.



- Compute -> Nova
- Networking -> Neutron
- Object Storage -> Swift
- Block Storage -> Cinder
- Image Service -> Glance
- Identity -> Keystone
- Dashboard -> Horizon

There are more (Kilo release)



- Telemetry – Ceilometer
- Orchestration - Heat
- Database - Trove
- Data Processing - Sahara
- Bare-Metal Provisioning – Ironic

From the installation guide



Table 1.1. OpenStack services

Service	Project name	Description
Dashboard	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.
Compute	Nova	Manages the lifecycle of compute instances in an OpenStack environment. Responsibilities include spawning, scheduling and decommissioning of virtual machines on demand.
Networking	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.
Storage		
Object Storage	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.
Block Storage	Cinder	Provides persistent block storage to running instances. Its pluggable driver architecture facilitates the creation and management of block storage devices.
Shared services		
Identity service	Keystone	Provides an authentication and authorization service for other OpenStack services. Provides a catalog of endpoints for all OpenStack services.
Image service	Glance	Stores and retrieves virtual machine disk images. OpenStack Compute makes use of this during instance provisioning.
Telemetry	Ceilometer	Monitors and meters the OpenStack cloud for billing, benchmarking, scalability, and statistical purposes.
Higher-level services		
Orchestration	Heat	Orchestrates multiple composite cloud applications by using either the native HOT template format or the AWS CloudFormation template format, through both an OpenStack-native REST API and a CloudFormation-compatible Query API.
Database service	Trove	Provides scalable and reliable Cloud Database-as-a-Service functionality for both relational and non-relational database engines.
Data processing service	Sahara	Provides capabilities to provision and scale Hadoop clusters in OpenStack by specifying parameters like Hadoop version, cluster topology and nodes hardware details.



Examining Open**Stack** fundamental services

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

- User
- Credentials
- Token
 - Associated with a user, an arbitrary bit of text that is used to access resources
- Group of users
- Project
 - Synonym to tenant
- Role
 - Assigned to users or groups for projects
- Domain
 - Higher level of hierarchy – users and projects belong to domains

- Service
 - OpenStack service, such as Compute(Nova), Object storage (swift), Image Service (Glance).
- Endpoint
 - A network accessible address, usually described by URL, from where you can access an OpenStack service
- Role
 - A set of requirements to perform an action over the end point.

Manages the lifecycle of compute instances in an OpenStack environment.

Responsibilities include spawning, scheduling and decommissioning of virtual machines on demand.

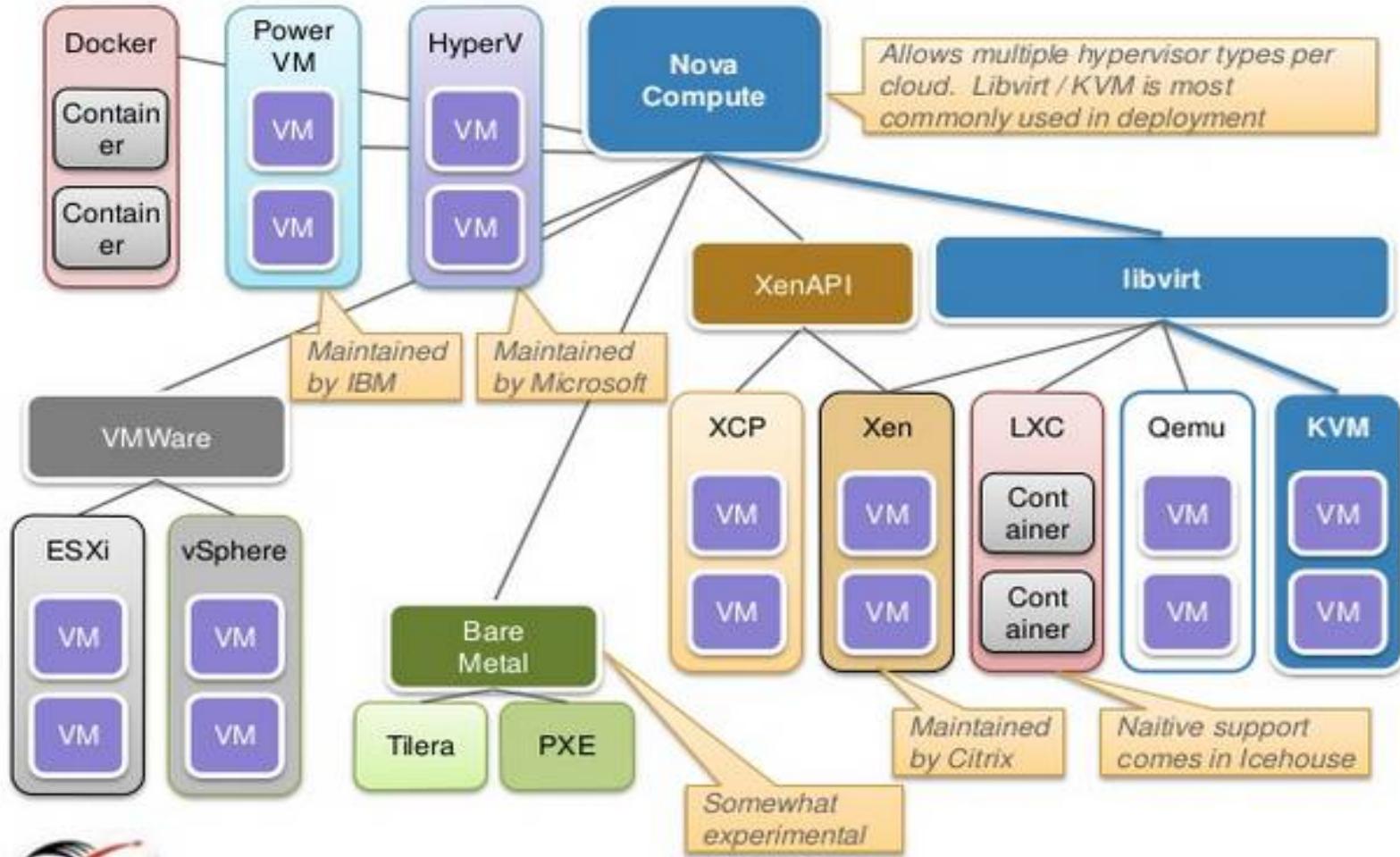


- Nova-api:
Web API frontend, accepts requests, validates them and contact other services if needed.
- Nova-scheduler
 - decides where to start an instance
- Nova-compute
 - running on each compute node, interacts with the hypervisor and actually starts the vm.



Nova components could run on different physical server

Nova compute drivers



- Functionality provided by drivers is not 100% similar.
- Exact "run_instance" flow depends on the driver implementation.
- Most features are developed and tested on KVM.
- Our experience is limited to KVM (no need to explore other so far..).
- Our next step: docker (see later)

Service responsible of storing image information 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)

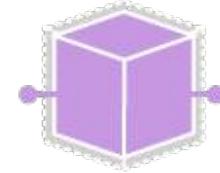
- Creates and export volumes via iSCSI to the compute node.
- Volumes are mounted transparently from the virtual machines.
- Supports multiple storage backends (NFS, LVM, Ceph, GlusterFS but also SAN/NAS devices from IBM, NetApp etc. . .)



- Ephimeral 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 store files, including VM images
 - Managed by Swift

- The graphical interface to access, provision and automate cloud-based resources.
- easy to plug in and expose third party products and services, such as billing, monitoring and additional management tools.
- also brandable for service providers and other commercial vendors who want to make use of it.
- A self-service portal to provision their own resources within the limits set by administrators.
- it provides to cloud administrator an overall view of the size and state of your cloud. You can create users and projects, assign users to projects and set limits on the resources for those projects.

Service responsible of creating and managing networks.



Very feature rich:

VERY COMPLEX TO MANAGE

- L2 and L3 networks.
- Allow creation of multiple networks and subnets.
- Plugin architecture.
- Supports advanced network services (Load Balancer, Firewall, DNS as a service)
- Integrates with network devices (Cisco, Brocade. . .)

OpenStack dashboard login



openstack
DASHBOARD

Log In

User Name

Password

Sign In



Log in

User Name

 3

Password

 eye icon

Connect

OpenStack dashboard login



Instance Overview - OpenStack Dashboard - Mozilla Firefox

Instance Overview - O... nembo.escience-lab.org/dashboard/project/ Most Visited Centos Wiki Documentation Forums Openstack xCAT easybuild

openstack exact Sign Out

Project: Compute

Overview Instances Volumes Images Access & Security Network

Overview

Limit Summary

Resource	Used	Total	Status
Instances	12	20	Used 12 of 20
VCPUs	18	40	Used 18 of 40
RAM	28.0GB	50.0GB	Used 28.0GB of 50.0GB
Floating IPs	8	50	Used 8 of 50
Security Groups	3	10	Used 3 of 10
Volumes	5	10	Used 5 of 10
Volume Storage	720.0GB	1000.0GB	Used 720.0GB of 1000.0GB

Usage Summary

Select a period of time to query its usage:

From: 2015-04-01 To: 2015-04-30 Submit The date should be in YYYY-mm-dd format.

Active Instances: 7 Active RAM: 18GB This Period's VCPU-Hours: 98.70 This Period's GB-Hours: 3666.18

Usage

Instance Name	VCPUs	Disk	RAM	Uptime
Kit DataManager	2	40	4GB	4 months, 2 weeks
paraview-data	1	20	2GB	1 week, 5 days
hadoop-hadoop-task-data002	2	40	2GB	1 week, 5 days
hadoop-hadoop-task-data001	2	40	2GB	1 week, 5 days
hadoop-hadoop-jobtracker001	2	40	2GB	1 week, 5 days
hadoop-hadoop-name001	2	40	2GB	1 week, 5 days
Jenkins	2	40	4GB	4 days, 23 hours

Displaying 7 items

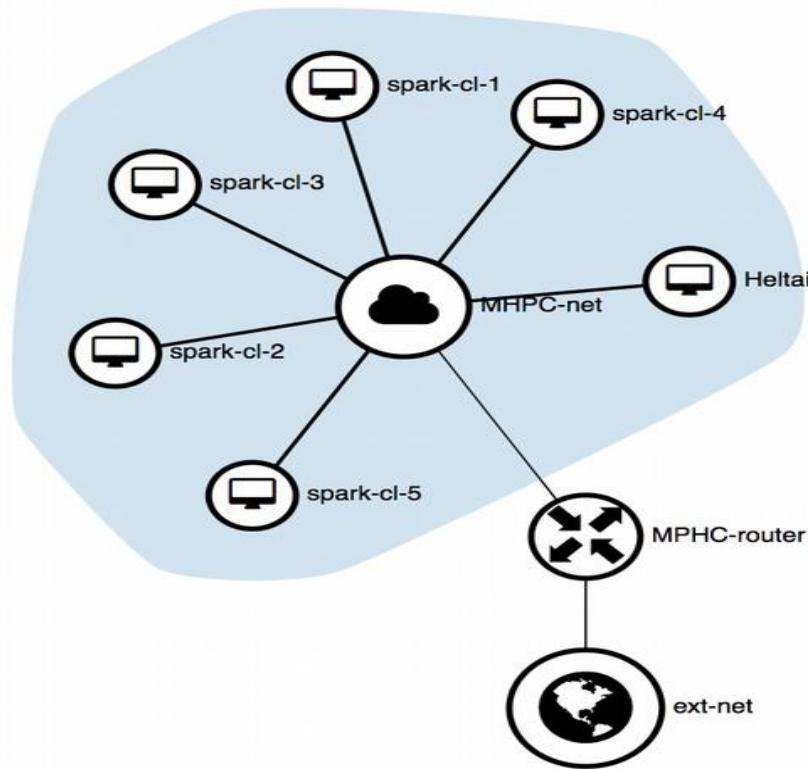
Download CSV Summary



Using
Open**Stack**

- Request Provisioning From UI
- A three steps action:
 - Login to Horizon
 - Specify parameters of VM
 - VM Name
 - Image (OS type)
 - Flavor (specifies CPU, Memory, Disk)
 - Network (required for Neutron)
 - Select/generate SSH keys for remote login
 - Optional (Persistent volumes, comments, etc.)
 - Select/Press "Launch" button

Network topology



Create a private network: step1



Screenshot of the eXact web interface showing the 'Networks' page.

The URL in the browser is nimbo.escience-lab.org/dashboard/project/networks/.

The page displays a table of networks:

Name	Subnets Associated	Shared	External	Status	Admin State	Actions
ext-net		No	Yes	Active	UP	Create Network Delete Networks

Displaying 1 item

Left sidebar navigation:

- Project
- Compute
- Network
 - Network Topology
- Networks
 - Networks (highlighted)
 - Routers
 - Load Balancers
 - Firewalls
 - VPN
- Orchestration
- Object Store
- Data Processing
- Identity

Bottom navigation bar:

nimbo.escience-lab.org/dashboard/project/networks/create

Create a private network: step2



Screenshot of the eXact web interface showing the 'Create Network' wizard.

The 'Networks' page is visible in the background, showing a single item named 'ext-net'. The 'Create Network' dialog is open in the foreground:

- Network Name:** mhpc-1-net
- Admin State:** UP (highlighted with a black arrow)
- Shared:**
- Create Subnet:**

At the bottom of the dialog are buttons: 'Cancel', '< Back', and 'Next >' (highlighted with a black arrow).

Create a private network: step3



Screenshot of the eXact web interface showing the 'Create Network' dialog box.

The dialog box is titled 'Create Network' and has three tabs: Network, Subnet (selected), and Subnet Details. The Subnet tab contains the following fields:

- Subnet Name: mhpc-1-subnet
- Network Address: 192.168.41.0/24
- IP Version: IPv4
- Gateway IP: (empty)
- Disable Gateway:

A large text area provides instructions: "Creates a subnet associated with the network. You need to enter a valid "Network Address" and "Gateway IP". If you did not enter the "Gateway IP", the first value of a network will be assigned by default. If you do not want gateway please check the "Disable Gateway" checkbox. Advanced configuration is available by clicking on the "Subnet Details" tab."

At the bottom of the dialog box are buttons: Cancel, < Back, and Next >. A large arrow points from the 'Next >' button to the 'Actions' column of the main table on the right.

The main interface shows a 'Networks' section with a table:

Status	Admin State	Actions
Active	UP	+ Create Network Delete Networks

The left sidebar includes sections for Project, Compute, Network, Network Topology, Networks, Routers, Load Balancers, Firewalls, VPN, Orchestration, Object Store, Data Processing, and Identity.

Create a private network: step4



The screenshot shows the eXact dashboard interface for creating a new network. The main window displays a 'Networks' list with one item named 'ext-net'. On the left, a sidebar navigation menu includes categories like Project, Compute, Network, Network Topology, Networks, Routers, Load Balancers, Firewalls, VPN, Orchestration, Object Store, Data Processing, and Identity. The 'Network' category is currently selected. A modal dialog box titled 'Create Network' is open, showing the 'Subnet Details' tab. Inside the dialog, there is a checked checkbox for 'Enable DHCP' and a section for 'Allocate IP Pools' which is currently empty. Below it are sections for 'DNS Name Servers' and 'Host Routes', also currently empty. At the bottom of the dialog are three buttons: 'Cancel', '< Back', and a prominent blue 'Create' button. Two arrows point to the 'Create' button: one from the 'Allocate IP Pools' input field and another from the bottom right of the dialog box itself.

Create a private network: step5



Screenshot of the eXact web interface showing the 'Routers' page.

The URL in the browser is nimbo.escience-lab.org/dashboard/project/routers/.

The page title is "Routers".

The sidebar on the left includes sections for Project, Compute, Network, Network Topology, Networks, Routers (selected), Load Balancers, Firewalls, VPN, Orchestration, Object Store, Data Processing, and Identity.

The main content area shows a table with columns: Name, Status, External Network, Admin State, Actions, and a "Create Router" button.

A message at the bottom of the table says "No items to display."

Two arrows point to the "Create Router" button: one from the top right and one from the sidebar's "Routers" section.

Name	Status	External Network	Admin State	Actions
No items to display.				

Create a private network: step6



Screenshot of the eXact web interface showing the 'Create Router' dialog box.

The dialog box is titled 'Create Router'. It contains the following fields:

- Router Name ***: mhpc-1-router
- Description:** Creates a router with specified parameters.
- Admin State**: UP
- External Network**: A dropdown menu showing:
 - Select network
 - Select network
 - ext-netThe option 'ext-net' is highlighted with an orange background.

At the bottom right of the dialog box are two buttons: 'Cancel' and 'Create Router'.

Arrows point from the text labels in the 'External Network' dropdown to the corresponding options in the list.

The background shows the 'Routers' section of the eXact dashboard, with a table listing routers and a sidebar with various project and network management links.

Create a private network: step7



Screenshot of the eXact platform interface showing the Routers dashboard.

The URL in the browser is nimbo.escience-lab.org/dashboard/project/routers/.

The sidebar navigation includes:

- Project
- Compute
- Network
 - Network Topology
 - Networks
- Routers (selected)
- Load Balancers
- Firewalls
- VPN
- Orchestration
- Object Store
- Data Processing
- Identity

The main content area displays the "Routers" table with the following data:

Name	Status	External Network	Admin State	Actions
mhpc-1-router	Active	ext-net	UP	Clear Gateway

A red arrow points to the "mhpc-1-router" row in the table.

Create a private network: step8



mhpc-1-router - CN... +

nimbo.escience-lab.org/dashboard/project/routers/a966597c-3d45-411c-a226-060c902fa619/

Search

Project

Compute

Network

Network Topology

Networks

Routers

Load Balancers

Firewalls

VPN

Orchestration

Object Store

Data Processing

Identity

MHPC-01

Clear Gateway

+ Add Interface

Add Interface

Routers / mhpc-1-router

Overview Interfaces Static Routes

Name Fixed IPs Status Type Admin State Actions

No items to display.

nimbo.escience-lab.org/dashboard/project/routers/a966597c-3d45-411c-a226-060c902fa619/addinterface

Create a private network: step9



Screenshot of the eXact interface showing the 'Add Interface' dialog box.

The dialog box is titled 'Add Interface' and contains the following fields:

- Subnet ***: A dropdown menu showing 'Select Subnet' and 'mhpcl-net: 192.168.41.0/24 (mhpcl-subnet)'. An arrow points to this field.
- Description:** A text area with instructions about connecting a subnet to the router.
- Router Name ***: A text input field containing 'mhpcl-router'.
- Router ID ***: A text input field containing 'a966597c-3d45-411c-a226-060c902fa619'.

At the bottom right of the dialog box are 'Cancel' and 'Submit' buttons. An arrow points to the 'Submit' button.

The background shows the 'Routers / mhpcl-router' interface with tabs for Overview, Interfaces, and Static Routes. The 'Interfaces' tab is selected. The 'Interfaces' section lists one item: 'mhpcl-router'.

Create a private network: step 10



Screenshot of the eXact cloud management interface showing the 'Access & Security' section.

The URL in the browser is nimbo.escience-lab.org/dashboard/project/access_and_security/.

The interface includes a sidebar with categories: Project, Compute, Network, Orchestration, Object Store, Data Processing, and Identity. The 'Network' category is currently selected, indicated by a red border.

The main content area is titled 'Access & Security' and contains tabs for 'Security Groups', 'Key Pairs', 'Floating IPs', and 'API Access'. The 'Floating IPs' tab is active, showing a table with columns: IP Address, Mapped Fixed IP Address, Pool, Status, Actions, and a 'Allocate IP To Project' button.

The table displays the message: 'No items to display.'

Three arrows point upwards from the bottom of the slide towards the screenshot:

- An arrow points from the bottom left towards the 'Network' category in the sidebar.
- An arrow points from the bottom center towards the 'Allocate IP To Project' button.
- An arrow points from the bottom right towards the top right corner of the interface.

Create a private network: step 11



Screenshot of the eXact cloud management interface showing the 'Allocate Floating IP' dialog box.

The dialog box is titled 'Allocate Floating IP'. It contains the following fields and information:

- Pool ***: A dropdown menu set to 'ext-net'.
- Description:** A text area with the placeholder 'Allocate a floating IP from a given floating IP pool.'
- Project Quotas**: A section showing 'Floating IP (0)' and '50 Available'.
- Buttons**: 'Cancel' and 'Allocate IP' (highlighted in blue).

Two arrows point to specific elements: one arrow points to the 'ext-net' selection in the 'Pool' dropdown, and another arrow points to the 'Allocate IP' button.

The background shows the 'Access & Security' dashboard with tabs for 'Security Groups', 'Key Pairs', and 'Floating IPs'.

- Request Provisioning From UI
- A three steps action:
 - Login to Horizon
 - Specify parameters of VM
 - VM Name
 - Image (OS type)
 - Flavor (specifies CPU, Memory, Disk)
 - Network (required for Neutron)
 - Select/generate SSH keys for remote login
 - Optional (Persistent volumes, comments, etc.)
 - Select/Press "Launch" button

Launch panel



Launch Instance

Details * Access & Security * Networking * Post-Creation Advanced Options

Availability Zone
nova

Instance Name *

Flavor *
m1.tiny

Instance Count *
1

Instance Boot Source *
--- Select source ---

Specify the details for launching an instance.
The chart below shows the resources used by this project in relation to the project's quotas.

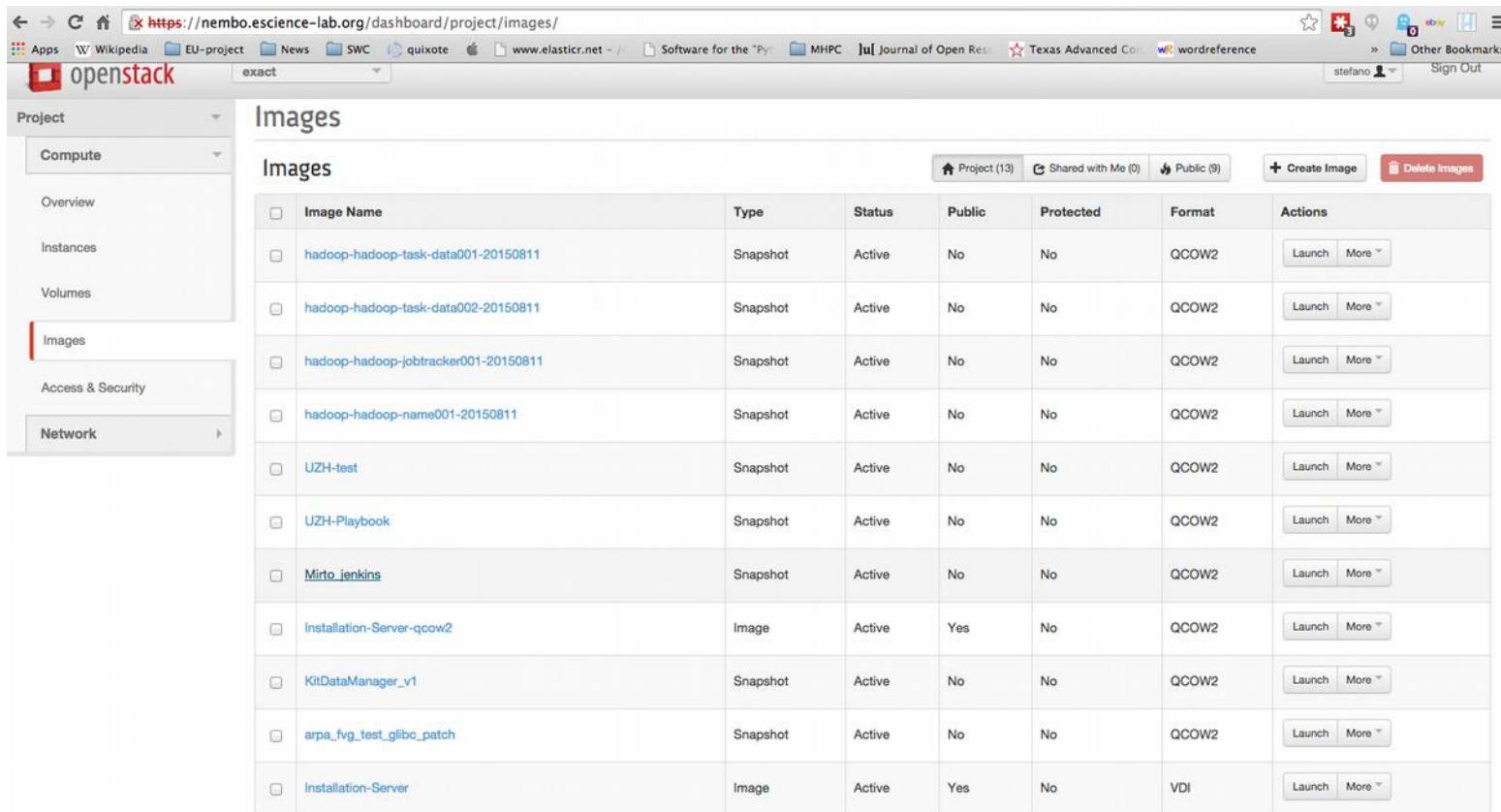
Flavor Details

Name	m1.tiny
VCPUs	1
Root Disk	1 GB
Ephemeral Disk	0 GB
Total Disk	1 GB
RAM	512 MB

Project Limits

Number of Instances	12 of 20 Used
Number of VCPUs	18 of 40 Used
Total RAM	35,328 of 51,200 MB Used

VM parameters: image (OS)



The screenshot shows the OpenStack dashboard interface at <https://nembo.escience-lab.org/dashboard/project/images/>. The left sidebar is titled 'Project' and includes 'Compute' (selected), 'Overview', 'Instances', 'Volumes', 'Images' (selected), 'Access & Security', and 'Network'. The main content area is titled 'Images' and displays a table of VM images. The table columns are: Image Name, Type, Status, Public, Protected, Format, and Actions. The table contains 11 rows of data:

Image Name	Type	Status	Public	Protected	Format	Actions
hadoop-hadoop-task-data001-20150811	Snapshot	Active	No	No	QCOW2	[Launch] [More]
hadoop-hadoop-task-data002-20150811	Snapshot	Active	No	No	QCOW2	[Launch] [More]
hadoop-hadoop-jobtracker001-20150811	Snapshot	Active	No	No	QCOW2	[Launch] [More]
hadoop-hadoop-name001-20150811	Snapshot	Active	No	No	QCOW2	[Launch] [More]
UZH-test	Snapshot	Active	No	No	QCOW2	[Launch] [More]
UZH-Playbook	Snapshot	Active	No	No	QCOW2	[Launch] [More]
Mirto_jenkins	Snapshot	Active	No	No	QCOW2	[Launch] [More]
Installation-Server-qcow2	Image	Active	Yes	No	QCOW2	[Launch] [More]
KitDataManager_v1	Snapshot	Active	No	No	QCOW2	[Launch] [More]
arpa_fvg_test_glibc_patch	Snapshot	Active	No	No	QCOW2	[Launch] [More]
Installation-Server	Image	Active	Yes	No	VDI	[Launch] [More]

- Virtual hardware templates are called "flavors" in OpenStack, defining sizes for RAM, disk, number of cores, and so on.

```
$ nova flavor-list
```

ID	Name	Memory_MB	Disk	Ephemeral	VCPUs	extra_specs
1	m1.tiny	512	1	0	1	{}
2	m1.small	2048	10	20	1	{}
3	m1.medium	4096	10	40	2	{}
4	m1.large	8192	10	80	4	{}
5	m1.xlarge	16384	10	160	8	{}

- A network should be defined..

Launch Instance

Project & User * Details * Access & Security Networking * Post-Creation

Advanced Options

Selected networks

Choose network from Available networks to Selected networks by push button or drag and drop, you may change NIC order by drag and drop as well.

Available networks

♦ br-10-2 (4ac1ccb2-ebcd4d41bb25:20a410e537bf) +

♦ ext-net (5cd87059-1fc0-4d88-9d48-0d8e31873cd5) +

Cancel Launch

VM parameters: volumes

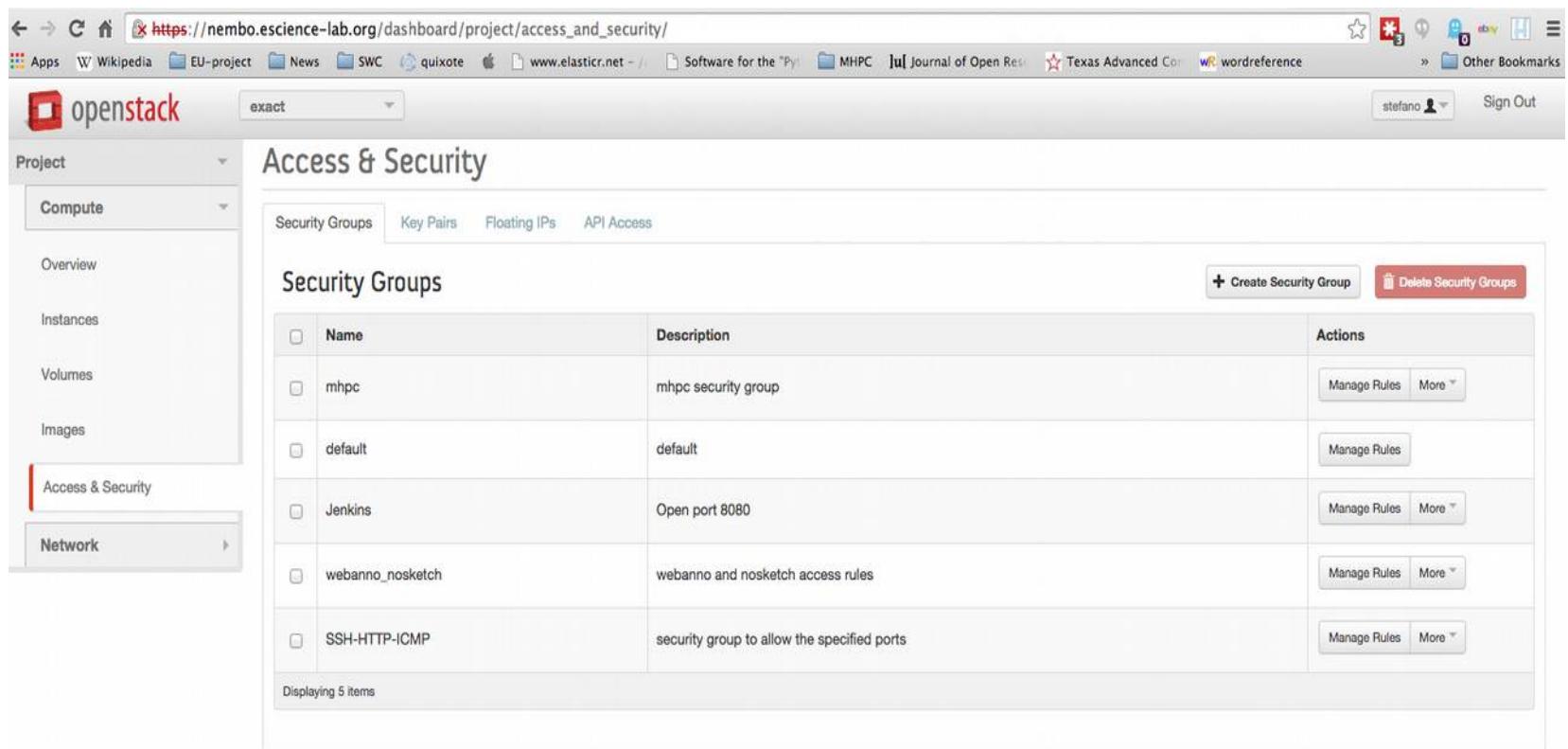


- Optionally Volumes can be attached

The screenshot shows the OpenStack Dashboard interface for managing volumes and snapshots. The top navigation bar includes links for 'Volumes & Snapshots', 'Documentation', 'Forums', 'Openstack', 'xCAT', and 'easybuild'. The main menu on the left is set to 'Compute' under the 'Project' section, with options for 'Overview', 'Instances', 'Volumes' (which is selected and highlighted in red), 'Images', 'Access & Security', and 'Network'. The central content area is titled 'Volumes & Snapshots' and displays the 'Volumes' tab. It features a table with columns for Name, Description, Size, Status, Type, Attached To, Availability Zone, and Actions. Two volumes are listed: 'paraview-data' (150GB, In-Use, Attached to 'paraview-data' on /dev/vdb, nova) and 'xeris_test' (20GB, Available, nova). The 'Actions' column for 'xeris_test' includes buttons for 'Edit Volume', 'More', 'Edit Attachments', and 'Create Snapshot', with 'Edit Attachments' currently highlighted in a dropdown menu.

<input type="checkbox"/>	Name	Description	Size	Status	Type	Attached To	Availability Zone	Actions
<input type="checkbox"/>	paraview-data	Paraview test data container	150GB	In-Use	-	Attached to paraview-data on /dev/vdb	nova	Edit Volume More
<input type="checkbox"/>	xeris_test		20GB	Available	-		nova	Edit Attachments Create Snapshot

- Optionally some security settings can be defined..



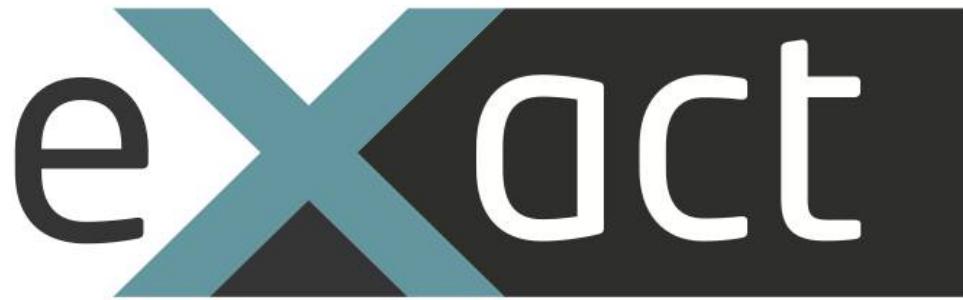
The screenshot shows the OpenStack dashboard at https://nembo.escience-lab.org/dashboard/project/access_and_security/. The left sidebar is collapsed, showing 'Project' and 'Compute' under 'Compute'. The main content area is titled 'Access & Security' and displays the 'Security Groups' tab. It lists five security groups:

Name	Description	Actions
mhpc	mhpc security group	Manage Rules More
default	default	Manage Rules
Jenkins	Open port 8080	Manage Rules More
webanno_nosketch	webanno and nosketch access rules	Manage Rules More
SSH-HTTP-ICMP	security group to allow the specified ports	Manage Rules More

At the bottom, it says 'Displaying 5 items'.

AFTER pressing “LAUNCH” :

1. Authentication is performed
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. (if requested) 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.



Where/when to use Open**Stack**
and why?

- Service providers offering an IaaS compute platform or services higher up the stack (PaaS)
- IT departments acting as cloud service providers for business units and project teams
- Processing big data with tools like Hadoop
- Scaling compute up and down to meet demand for web resources and applications
- High-performance computing (HPC) environments processing diverse and intensive workloads

Cattle vs Pets

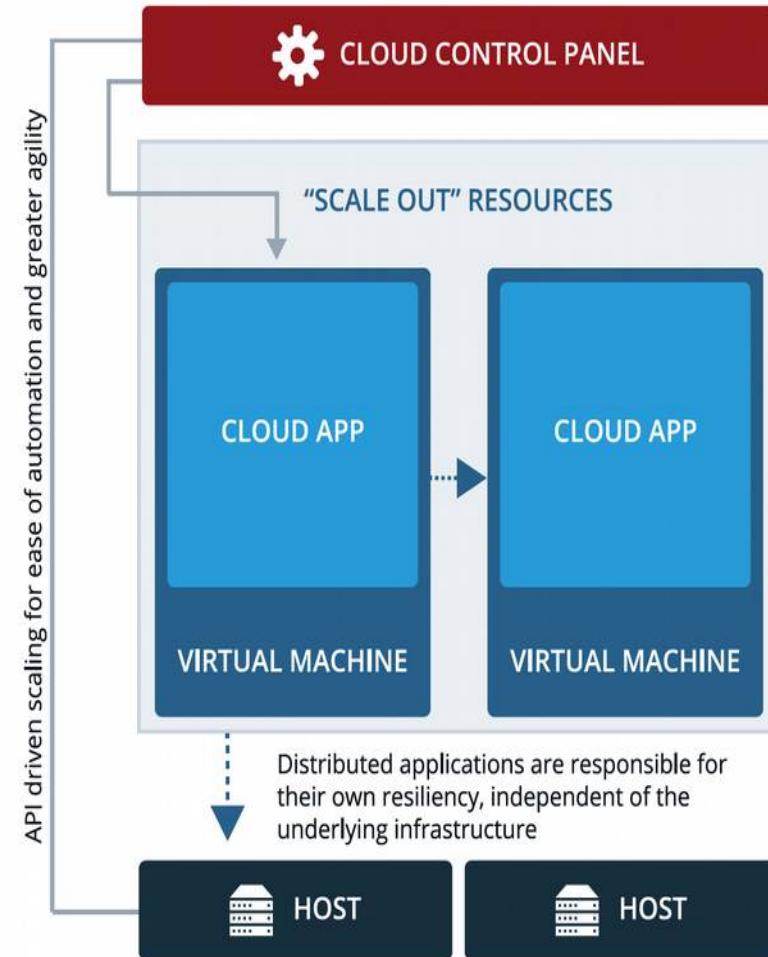
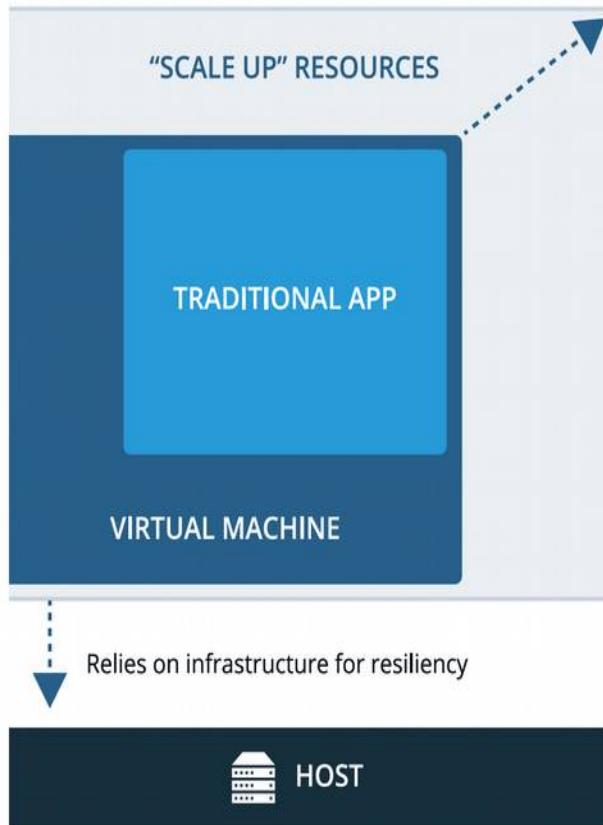
- pets are given names like
 - cicci.example.org
- you care about them
- they are unique, you check on them every day
- when they get ill, you nurse them back to health



- cattle are given names like
 - vm-001.example.org
- they are all the same
- when they get ill, you shoot them and get another one

Borrowed from @randybias at Cloudscaling <http://www.slideshare.net/randybias/the-cloud--revolution--cyber--press--forum--philippines>

Scale up vs scale out



Virtualization vs. Cloud

Which cloud?



- **General purpose:** Uses common components that address 80% of common use cases.
- **Compute focused:** For compute intensive workloads such as high performance computing (HPC).
- **Storage focused:** For storage intensive workloads such as data analytics with parallel file systems.
- **Network focused:** For high performance and reliable networking, such as a content delivery network (CDN).
- **Multi-site:** For applications that require multiple site deployments for geographical, reliability or data locality reasons.
- **Hybrid cloud:** Uses multiple disparate clouds connected either for failover, hybrid cloud bursting, or availability.
- **Massively scalable:** For cloud service providers or other large installations
- **Specialized cases:** Architectures that have not previously been covered in the defined use cases.

From <http://docs.openstack.org/arch-design/arch-design.pdf>

According to a survey by ZDNet, respondents perceived benefits from the ability to deploy:

- applications faster (54 percent);
- reduced effort to deploy applications (40 percent);
- Streamlined development and testing (38 percent);
- reduced application deployment costs (31 percent);
- server consolidation (25 percent).

- quota management vs. fairsharing
- software installation (user vs. admin)
- scalability vs. performance
- shared vs. exclusive resource
- continuous growth vs. fixed size

- Do you need a cluster ?
- Buy it !
 - buy the machines
 - find a room
 - setup air conditioning and ensure you have enough power
 - hire a system administrator
- Run on someone else's cluster
 - it may not have all the software you need
 - need to negotiate policies
 - resource usage conflicts
- Make your own cluster in the Cloud
 - you choose the software and the configuration
 - You create it when you need it

What about performance ?



- Not so big concern
 - hypervisors can reduce around 5-10% the performance.
 - linpack and stream small reduction: ~5%.
 - Networking: bandwidth about 5%, for latency about 10% reduction.
- Numbers are above are perfectly acceptable if you reduce the provisioning time from months to minutes..

- Many around
 - Starcluster
 - Virtualcluster etc..
- Our experience:
 - elasticcluster (UZH)
 - <https://github.com/gc3-uzh-ch/elasticcluster/>
 - Command line tool
 1. creates virtual machines in a cloud
 2. installs and configures the software you want
 3. add and remove nodes if needed

- Different kind of computational clusters:
 - Batch systems:
 - SLURM
 - OpenGridEngine
 - Torque+MAUI
 - Hadoop
 - Matlab Distributed Computing Servers
- Multiple distributed filesystems:
 - OrangeFS/PVFS
 - GlusterFS
 - Ceph
 - HDFS

- IRONIC service could be the solution
 - bare metal (physical server) provisioning in cloud;
- CONTAINERS is the other option

- A few use-cases for bare metal (physical server) provisioning in cloud:
 - High-performance computing clusters
 - Computing tasks that require access to hardware devices which can't be virtualized
 - Database hosting (some databases run poorly in a hypervisor)
 - Single tenant, dedicated hardware for performance, security, dependability and other regulatory requirements
 - Or, rapidly deploying a cloud infrastructure

What are containers ?

Containers are isolated, portable environments where you can run applications along with all the libraries and dependencies they need.

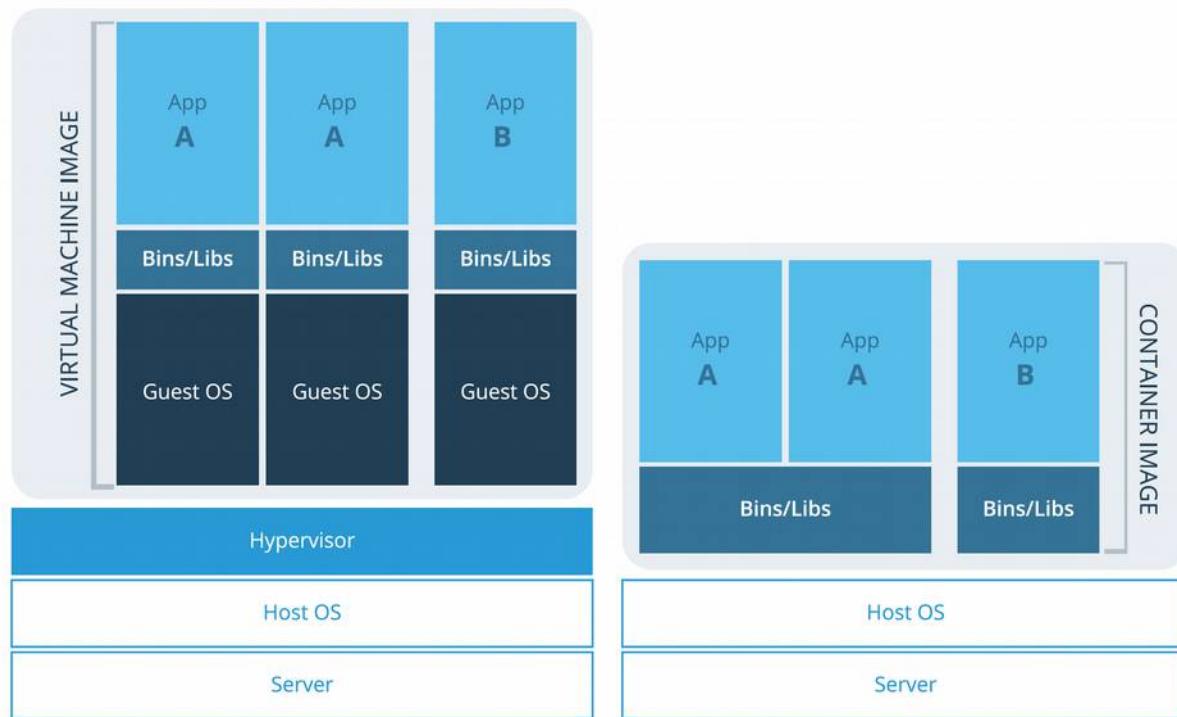


Figure 1: Containers vs. VMs

A Containers team active from May 2014

Integrate container technology in OpenStack,
and providing new services and tools

The objective is to allow users to create and
manage containers with an experience
consistent with what they expect from using the
Nova service to get virtual machines.

The aim is to offer developers a single set of
compatible APIs to manage their workloads,
whether those run on containers, virtual
machines or bare metal.

- **Magnum** is designed to offer container specific APIs for multi-tenant containers-as-a-service with OpenStack
- **Kolla** is designed to offer a dynamic OpenStack control plane where each OpenStack service runs in a Docker container.
- **Murano** is an application catalog solution that allows for packaged applications to be deployed on OpenStack
- With the Liberty release, Magnum and Murano will be production-ready.

What about overhead for HPC



TABLE I. RESULTS FOR PXZ, LINPACK, STREAM, AND RANDOMACCESS. EACH DATA POINT IS THE ARITHMETIC MEAN OF TEN RUNS. DEPARTURE FROM NATIVE EXECUTION IS SHOW WITHIN PARENTHESSES "()". THE STANDARD DEVIATION IS SHOWN WITHIN SQUARE BRACKETS "[]".

Workload	Native	Docker	KVM-untuned	KVM-tuned
PXZ (MB/s)	76.2 [± 0.93]	73.5 (-4%) [± 0.64]	59.2 (-22%) [± 1.88]	62.2 (-18%) [± 1.33]
Linpack (GFLOPS)	290.8 [± 1.13]	290.9 (-0%) [± 0.98]	241.3 (-17%) [± 1.18]	284.2 (-2%) [± 1.45]
RandomAccess (GUPS)	0.0126 [± 0.00029]	0.0124 (-2%) [± 0.00044]	0.0125 (-1%) [± 0.00032]	Tuned run not warranted
Stream (GB/s)	Add	45.8 [± 0.21]	45.6 (-0%) [± 0.55]	
	Copy	41.3 [± 0.06]	41.2 (-0%) [± 0.08]	
	Scale	41.2 [± 0.08]	41.2 (-0%) [± 0.06]	
	Triad	45.6 [± 0.12]	45.6 (-0%) [± 0.49]	

Taken From “An Updated Performance Comparison of Virtual Machines and Linux Containers”

1. Cloud systems are generally about maximizing tenancy/utilization of resources
2. Not so much about individual application performance.

As services provider point 1 should be taken into account

Individual application performance is important but at a second stage

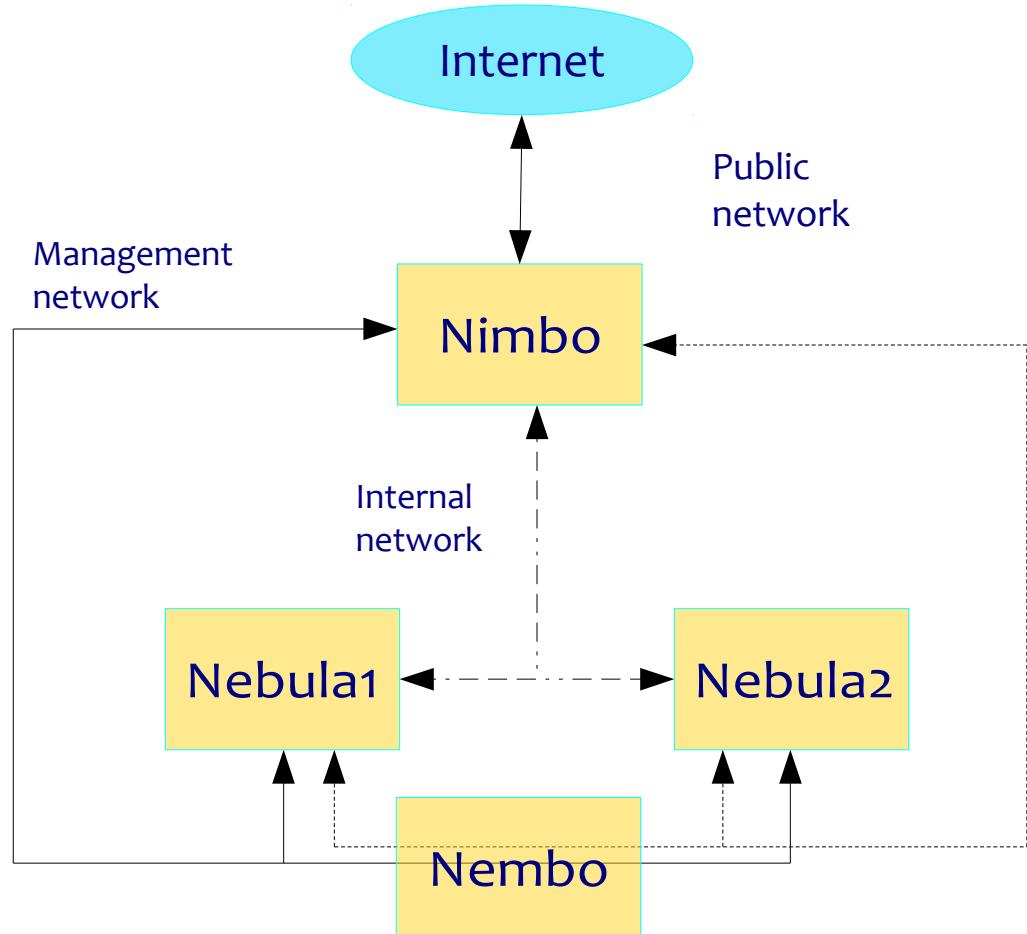


eXact-lab/CNR_IOM Open**Stack**
infrastructures

- eXact lab manages:
 - CNR/IOM cloud infrastructure
 - production
 - eXact lab testbeds
 - Experimental ones
 - COSILT cloud infrastructure
 - Production level

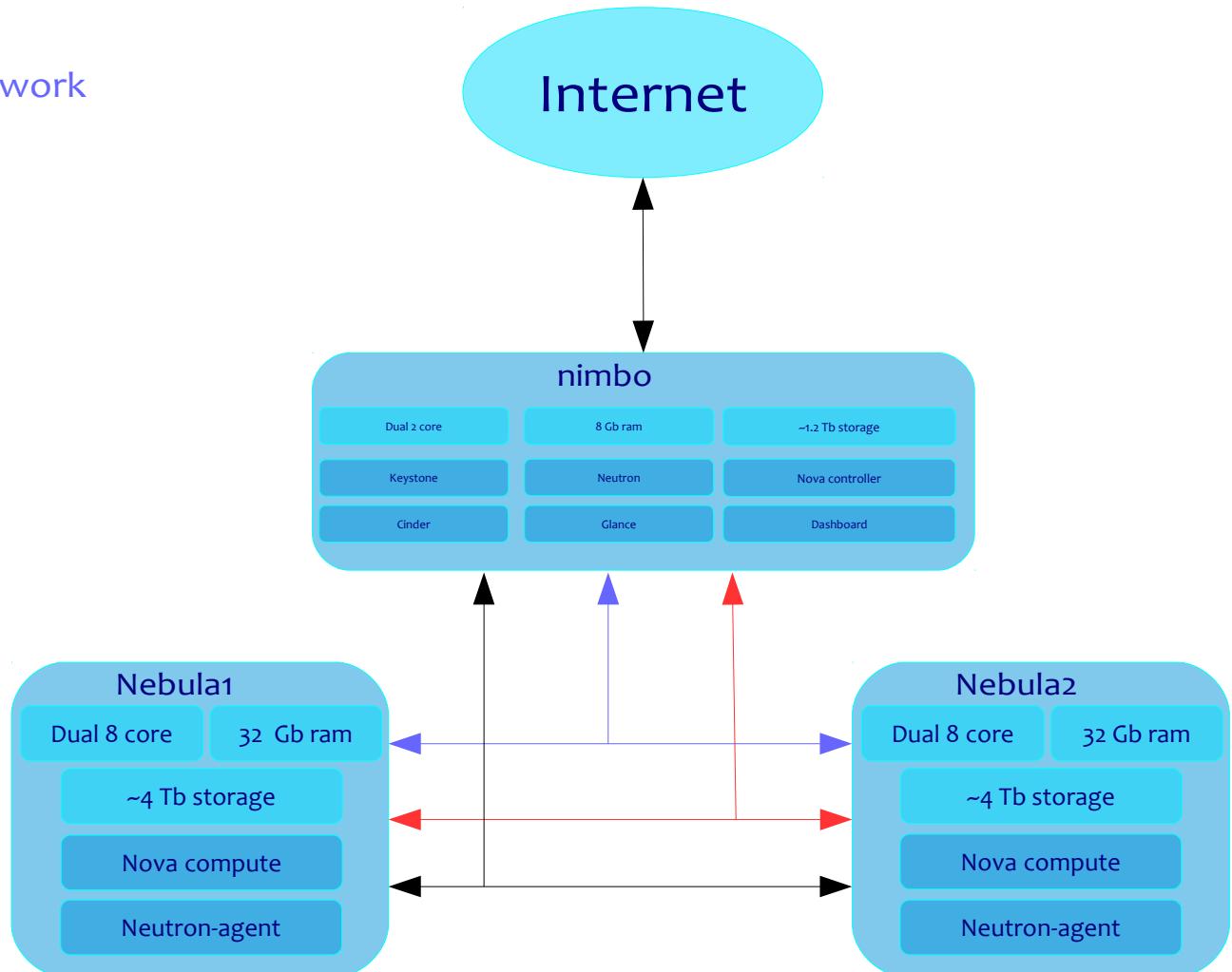
The production environment is composed by

- One central node:
 - provides the central services (Keystone, Nova controller, Neutron, Glance, Cinder)
 - 2 compute nodes (nova-compute)
 - 1 storage node (glance+cinder)



CNR/IOM infrastructure

- █ Management network
- █ Public network
- █ Internal network



3 different networks:

- the management network
 - physical network
- the internal network (high speed network for data transfer)
 - Physical infiniband network + internal virtual network
- OpenStack “public” network:
 - Virtual network managed by neutron

- Server Consolidation
- Simple testing of virtual machine/services for CNR/IOM and eXact users
- Release: Mitaka
- High availability under development
- Integration on CNR/IOM HPC cluster at later stage..(thesis available..)

Testbeds to:

- test the new releases of OpenStack
- plan the migration from one release to another.
- Experimenting/testing new services and/or specific customizations
- On relatively old HW nodes hosted at CNR/IOM premises.

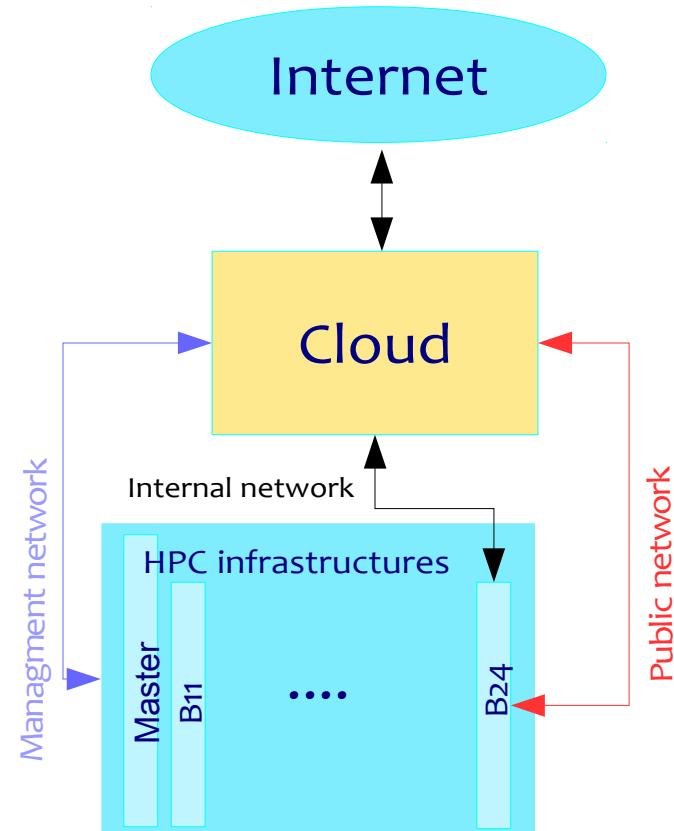
- COSIlt setup an HPC infrastructure to provide HPC services to SMEs..
- Such approach requires that HPC resources should be easily used/provided on demand
- eXact-lab is providing its competence to COSINT to help the consortium to identify
- Openstack is the tool we are using..



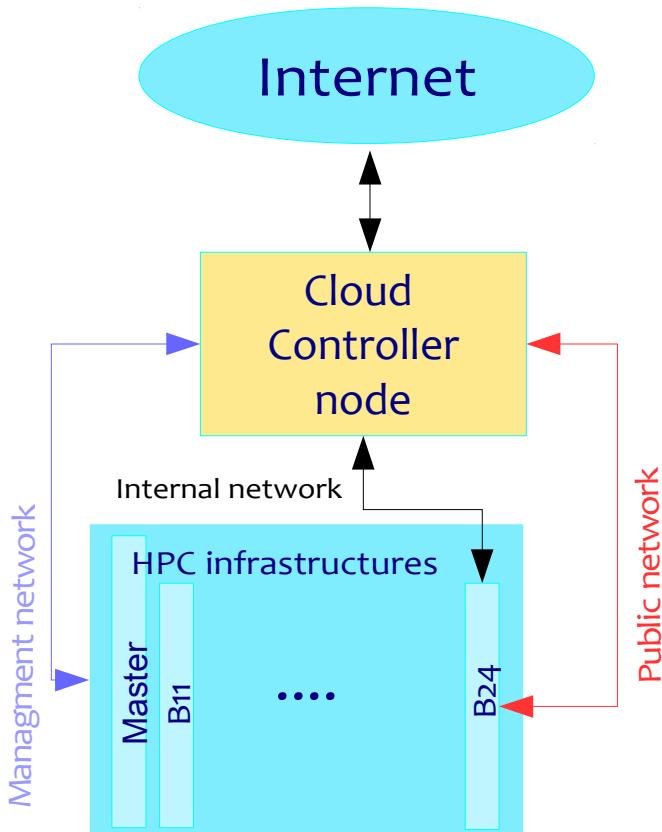
Cosint Cloud Infrastructure



Cloud: Dual Intel
40 core
@2.3GHz
32 Gb ram
2 Tb storage
B24: Dual Intel
Xeon
E5 -2697/v2
24 core
@2.7GHz
64 Gb ram
1 Tb storage



About the network



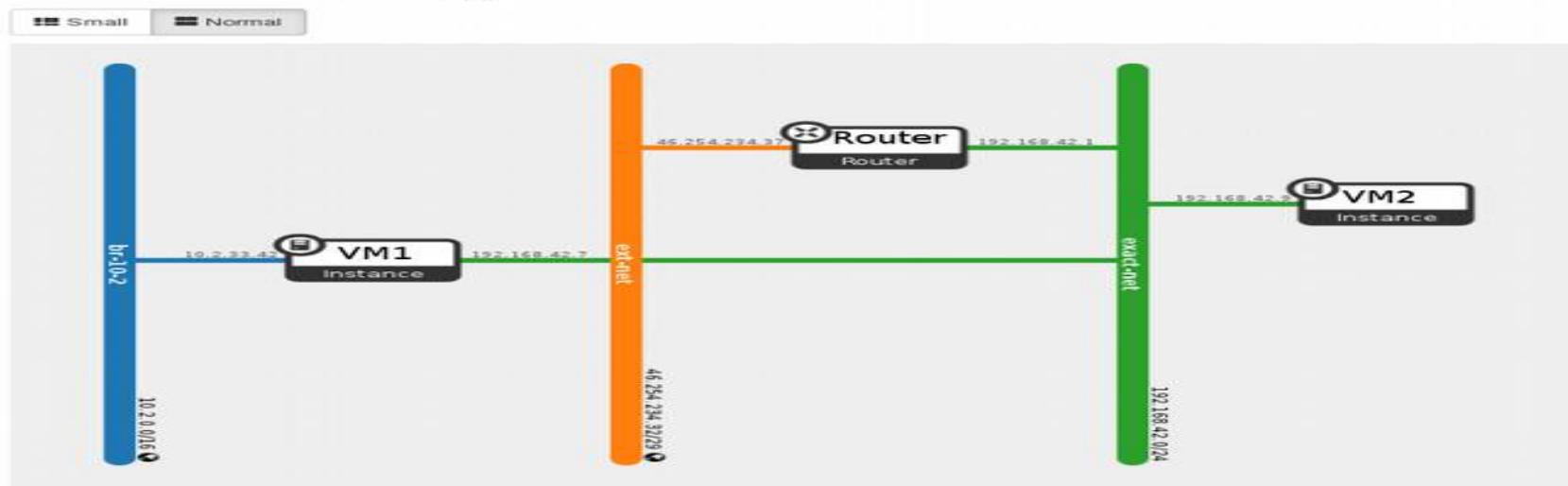
- The internal network is where the virtual network are spawned.
- A virtual network belongs to a tenant and can be shared.
- It is possible to assign a floating IP to a VM spawned on the virtual network
- The security of the VM is managed by the security groups (it is still possible to have a firewall on the VM, but only the connections allowed by the security groups will arrive to the VM)

Network topology example (COSINT):



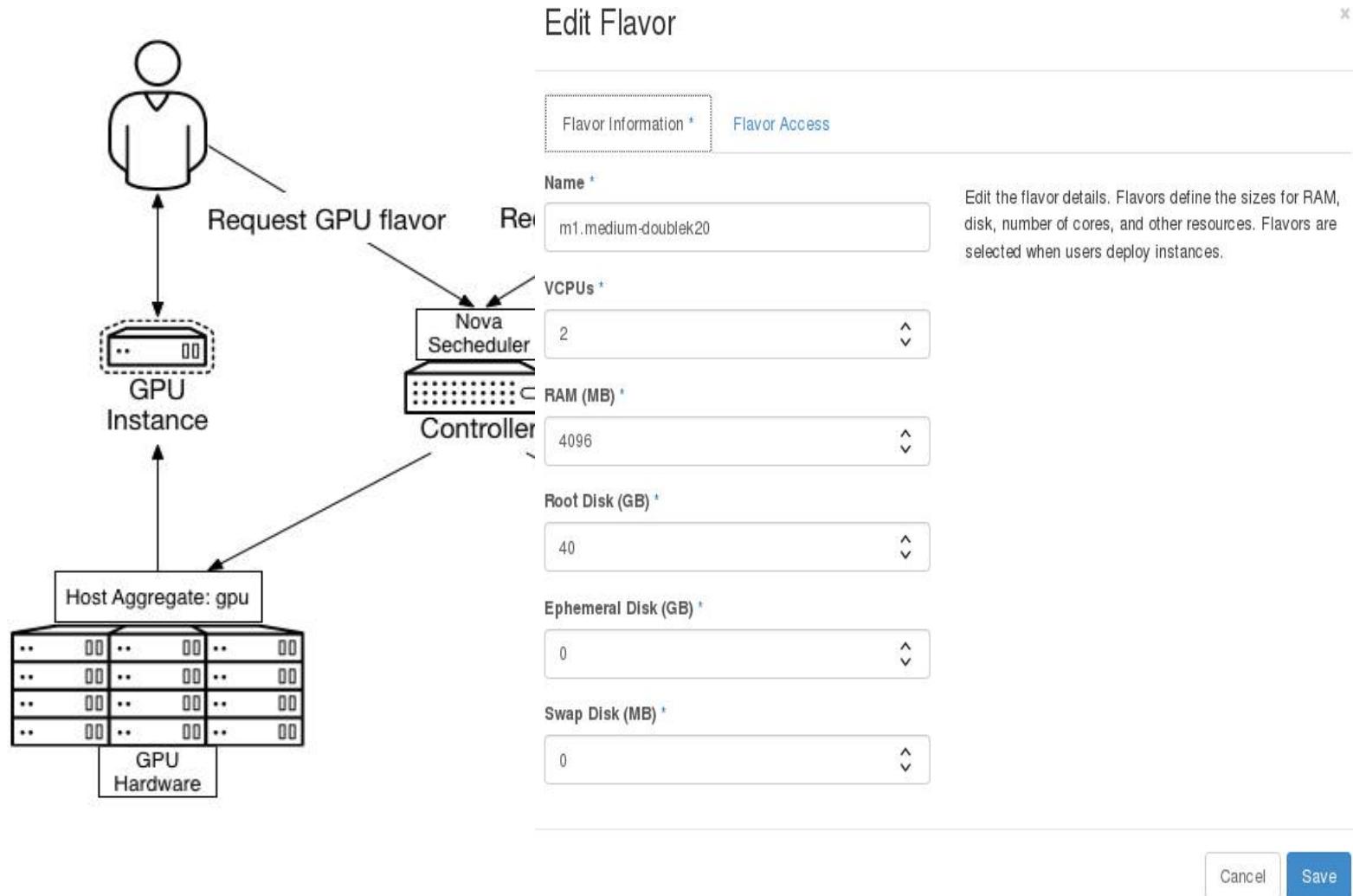
- br-10-2 and ext-net are external network, shared between all the tenants.
- br-10-2 is configured without a router and the VMs connected to this network have only (private) fixed IP
- on ext-net floating IPs are assigned to the Vms (this allow the Vms to be reached from internet).
- The exact-net is the tenant network and is connected to the ext-net via a router (exact-router)

Network Topology



- Provide VM with GPGPU facilities
 - a VM can have direct access to the GPU on the compute node and so provide hardware acceleration
- Done through enabling at nova level “PCI passthrough” mechanism.
- Not yet officially supported by mainstream openstack distribution

Get the GPU flavor



Update Metadata

You can specify resource metadata by moving items from the left column to the right column. In the left columns there are metadata definitions from the Glance Metadata Catalog. Use the "Other" option to add metadata with the key of your choice.

Available Metadata	Filter	Q
Custom	+	
No available metadata		

Existing Metadata	Filter	Q
pci_passthrough:alias	k20:2	-

You can specify resource metadata by moving items from the left column to the right column. In the left columns there are metadata definitions from the Glance Metadata Catalog. Use the "Other" option to add metadata with the key of your choice.

Cancel

Save

- OpenStack complex software not easy to master rapidly
- Complex to use, learning curve is steep
- Stability/Usability is still an issue sometime
- Many interesting projects/components that promise great applications and smart usage for HPC/scientific community