Openstack Workshop

14/01/2015

Deployments: Packstack RDO - Icehouse

Session 1: Basic lab Introduction – Hands On

Hardware/Software requirements for this session:

HOST (your portable)
RAM : 8GB, Min(4G)
HD : 500GB, Min(80G Free)
OS : Linux / Win / OSX

Hypervisor : Virtualbox / KVM
Provision/Orchestration: Vagrant / Puppet / SH Scripts
Vagrant Boxes: Centos64 / Centos65

Tested:
6.4: https://github.com/2creatives/vagrant-centos/releases/download/v6.4.2/centos64-x86_64-20140116.box
Provider: Virtualbox 4.3.16

6.5: https://github.com/2creatives/vagrant-centos/releases/download/v6.5.3/centos65-x86_64-20140116.box
Provider: Virtualbox 4.3.20

Vagrant files and puppet manifests repository
Github repo: https://github.com/vmucuge/openstack-workshop

Adapt your vagrant configuration as your environment needs

- Create Virtual Networks (192.168.56.0/24 192.168.33.0/24)
- IDE/SATA Adapters (Controller Node)
- Adapt Memory size of VM to fit your hardware limitations

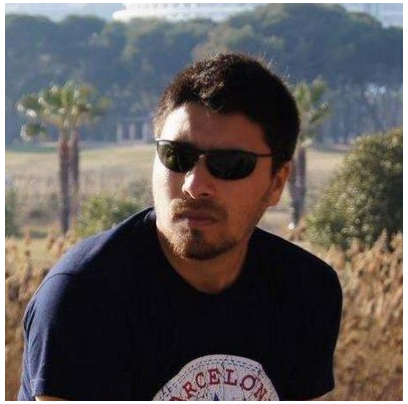
Who we are

Vinicius Galvao

- Systems Engineer
- In Amaris for +-2 years
- Europe +12 years
- Like football, summer, technology and nice stuff
- Investigating Openstack +- 3 years
- 1st Telefonica I+D Private Cloud with EPG group
- Some commercial advertisements: Adidas, Pepsi
- Current job: Groupalia

Jorge Medina

- Coder
- Senior Unix/Linux Engineer
- Monje, Fakir y Guerrero
- Current job: Groupalia
- https://twitter.com/mnothic



Outline for this workshop

Session 1:
Understand and review basic components of an Openstack Platform

Horizon: UI Dashboard
Keystone: Authentication Mechanism
Nova: Compute resources
Cinder: Block Storage
Glance: Images catalog
Neutron: Networking

This session will not make you the Openstack Expert

Cloud Computing

https://en.wikipedia.org/wiki/Cloud_computing

Cloud computing is a recently evolved computing terminology or metaphor based on utility and consumption of computing resources.
Cloud computing involves deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid.

But, WTF is the cloud?

Nobody understands, nobody knows what it is!
It is a fucking mystery!!!

What is Openstack

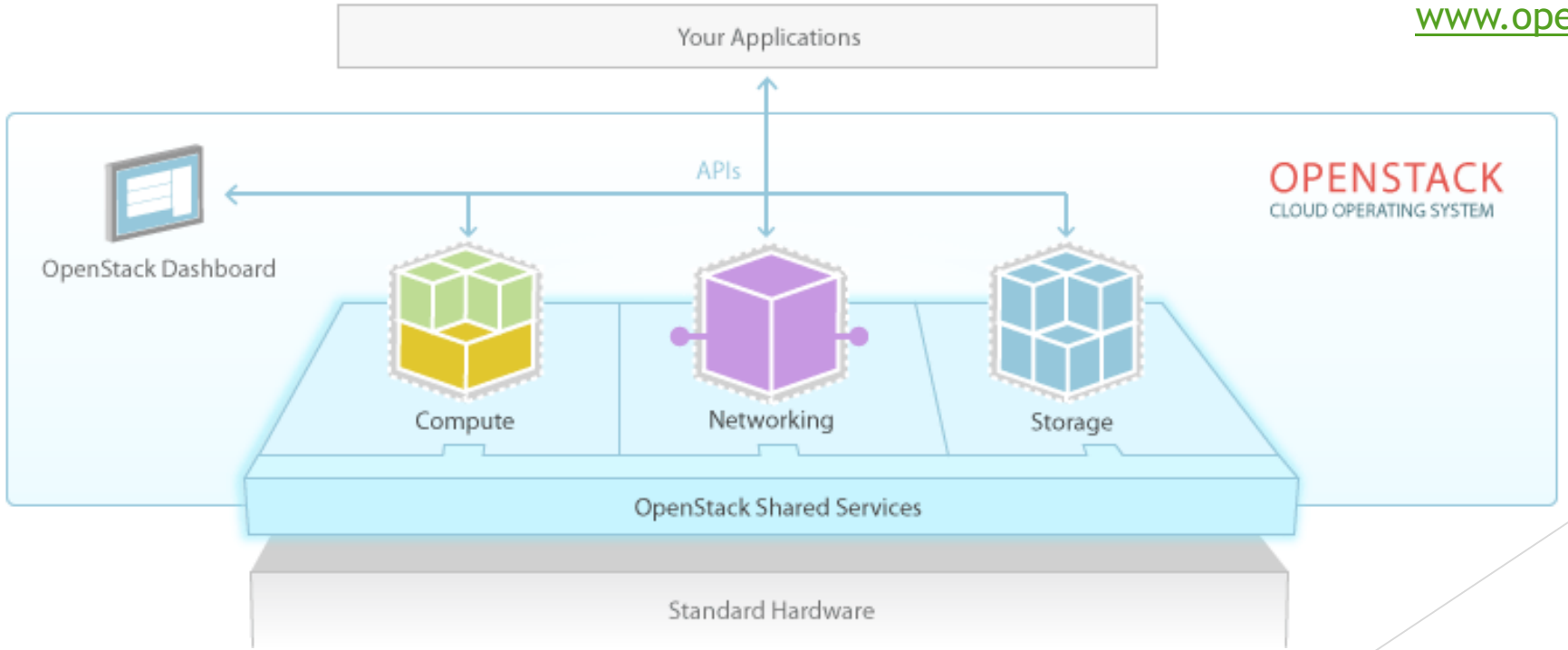
OpenStack is a set of software tools for building and managing cloud computing platforms for public and private clouds.

OpenStack is managed by the OpenStack Foundation.

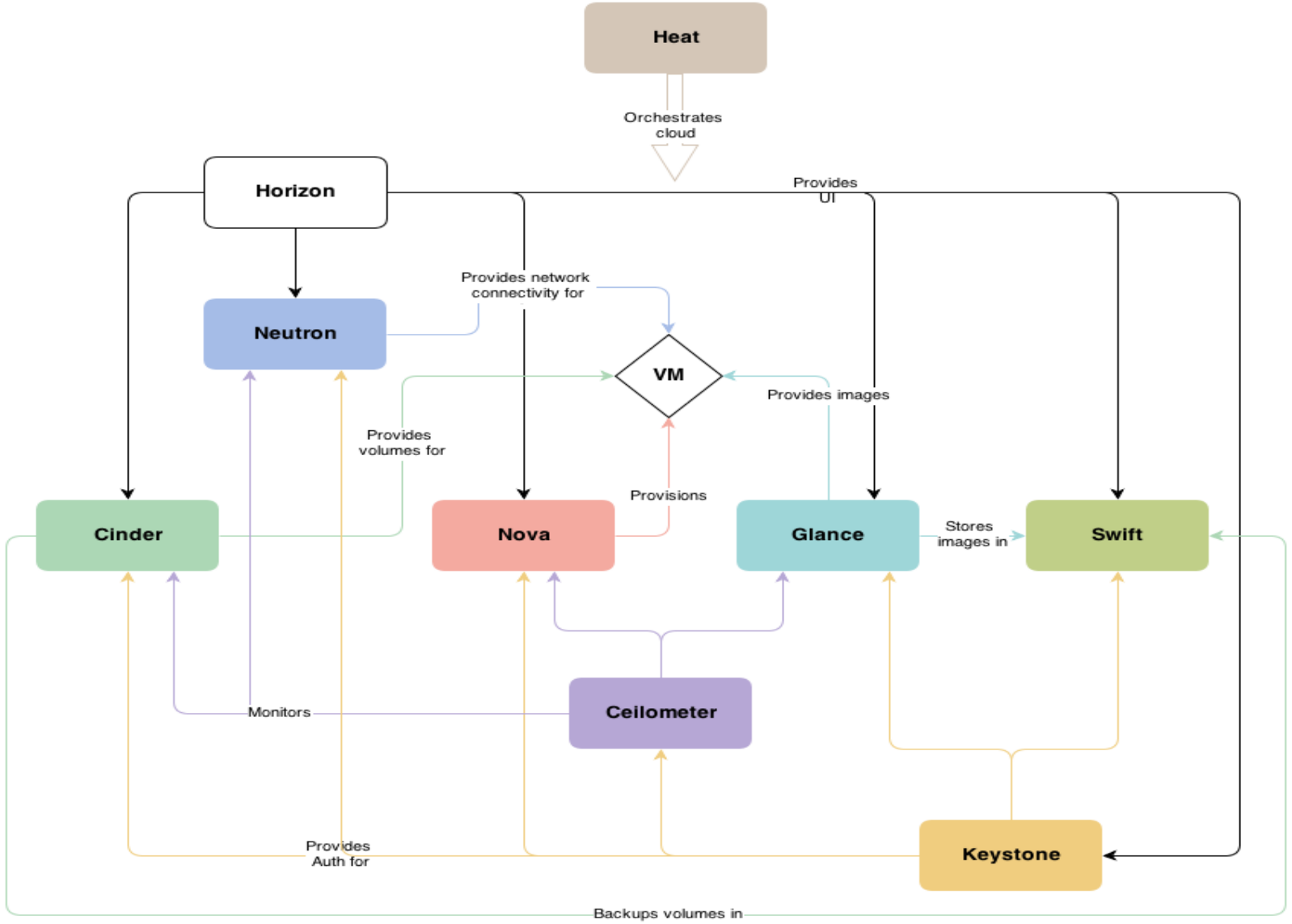
Openstack is NOT only a software, server or service on its own.

Is a open source standard that allow the Infrastructure team to install, govern, automate, deploy and orchestrate virtual resources in a hybrid infrastructures providing the user a self service platform…

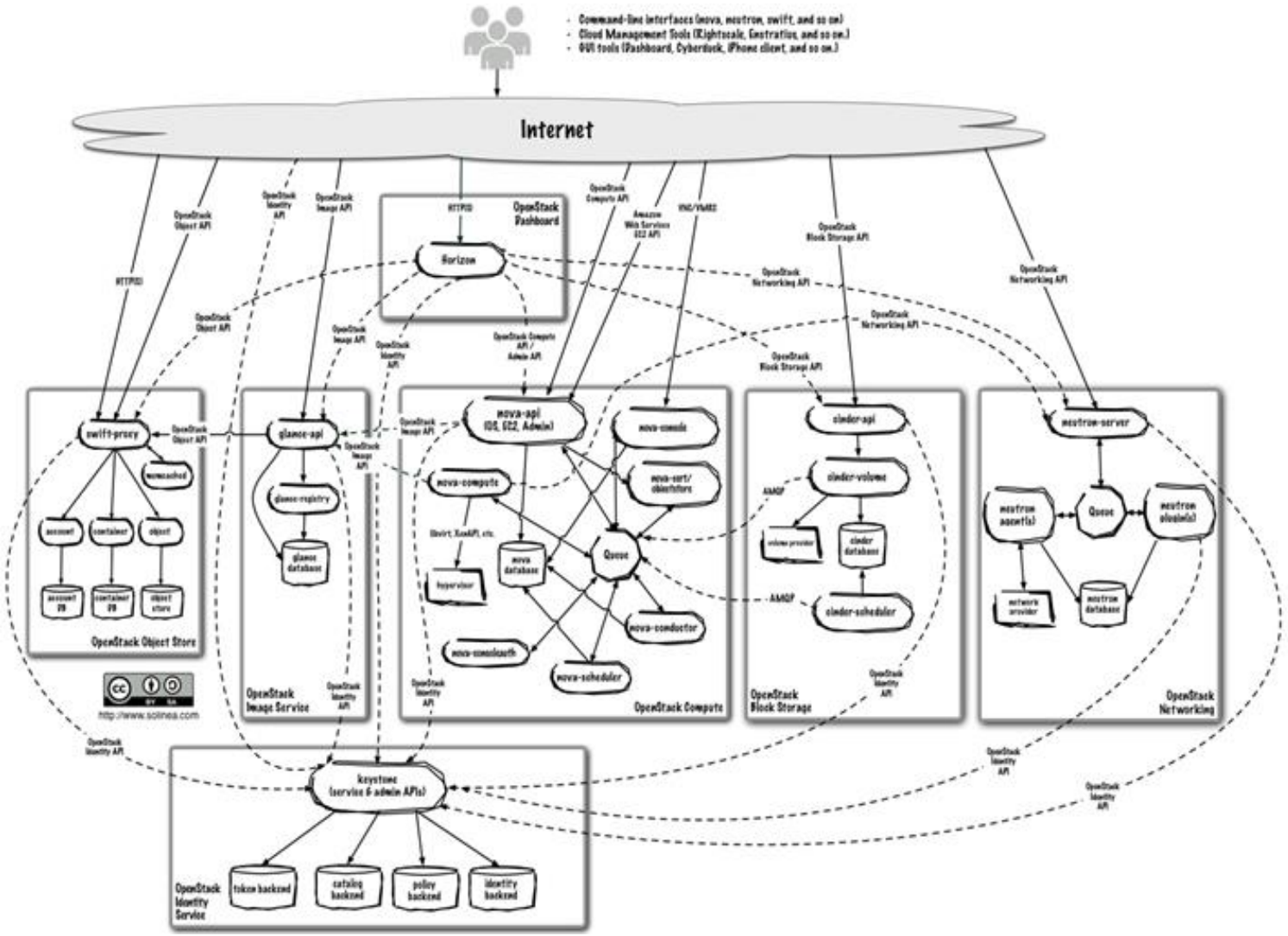
www.openstack.org



Entrypoints Communication flow



Entrypoints Communication flow…



Basic Components of an Openstack Lab

HORIZON
Horizon is the canonical implementation of OpenStack's Dashboard, which provides a web based user interface to OpenStack services.
Including Nova, Swift, Keystone, etc.

NOVA
Compute and controller services
Install Nova compute and controller services.
Deploy an instance from the command line.

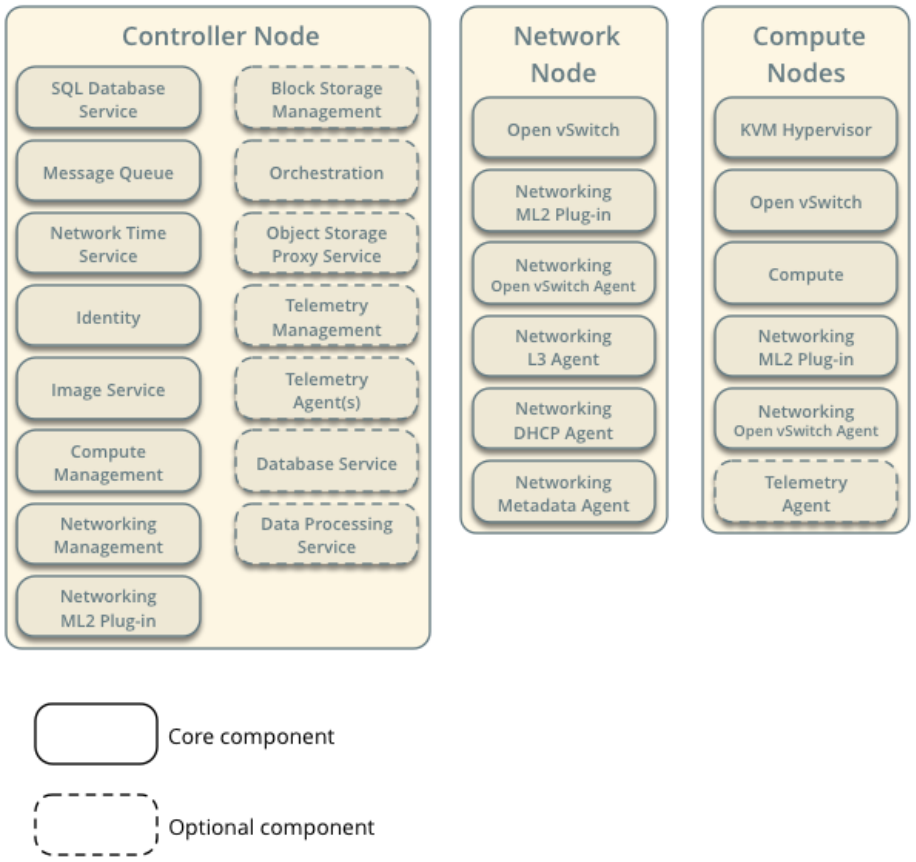
KEYSTONE
Keystone identity service
Install, configure, and use the Keystone authentication services.

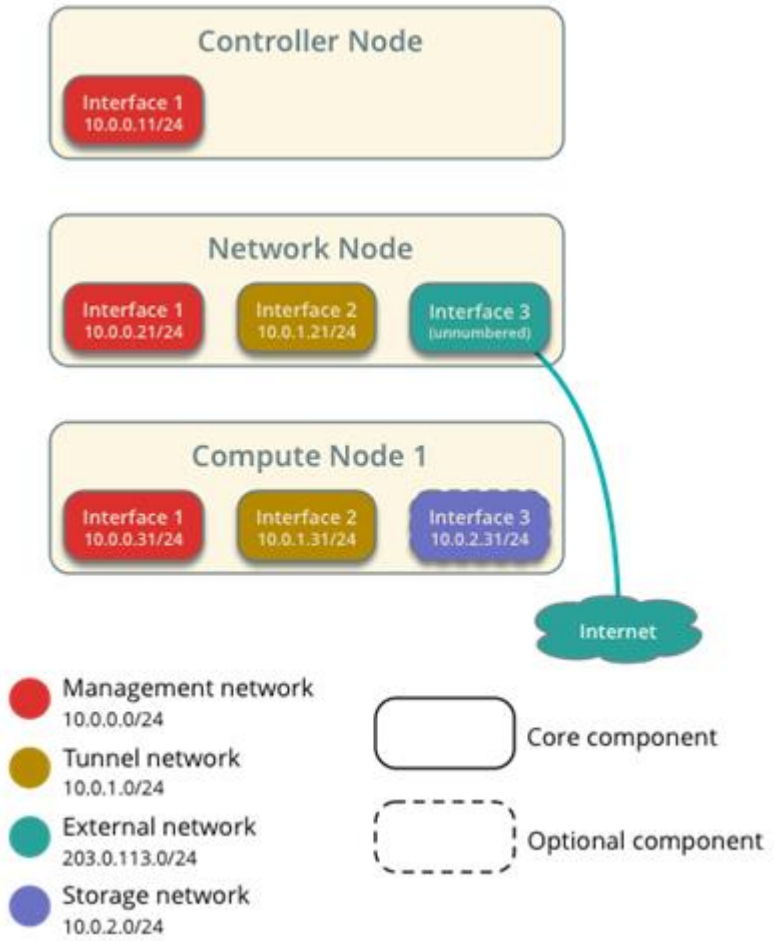
GLANCE
Image service
Install and use the Glance image service.

CINDER
block storage service
Install the Cinder block storage service.
Manage Cinder volumes.

NEUTRON
Networking service
Install, configure, and manage OpenStack Platform networking service.

Basic Lab Scenario Virtualization Components

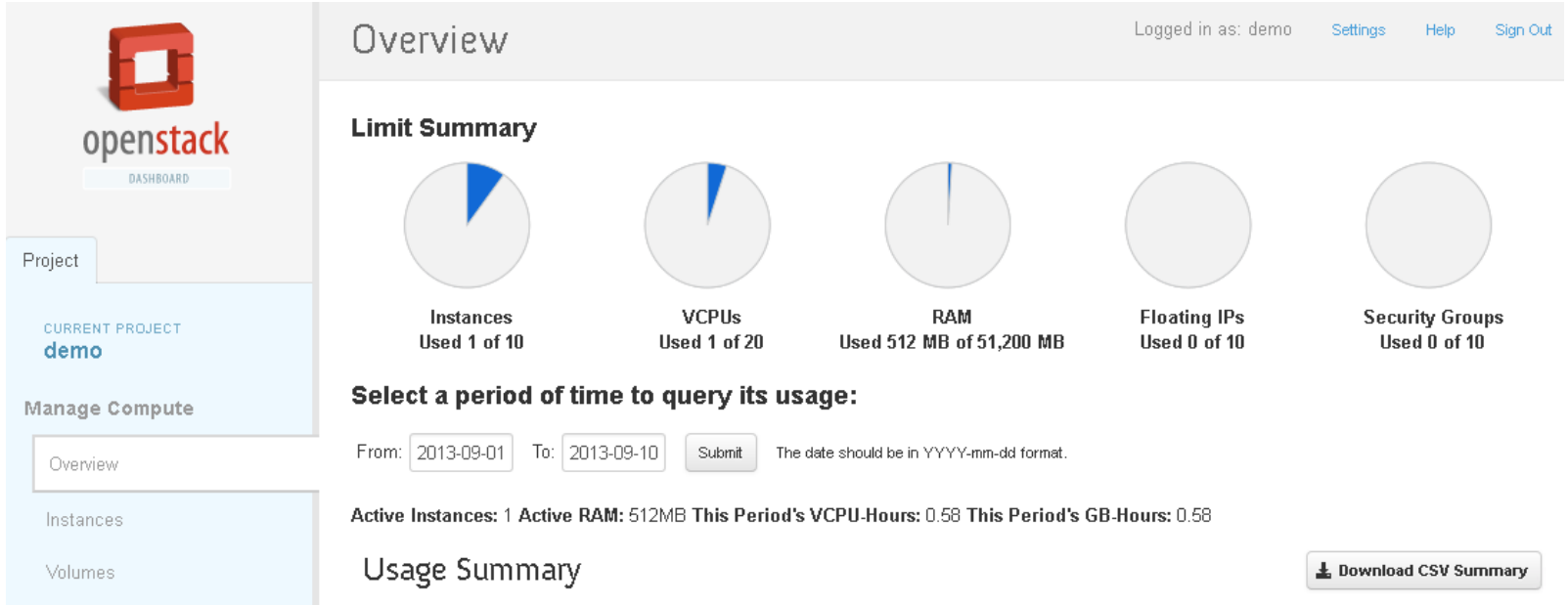




Dashboard (Horizon)

Is a Web interface that enables cloud administrators and users to manage various OpenStack resources and services.

The dashboard enables web-based interactions with the OpenStack Compute cloud controller through the OpenStack APIs.
Horizon enables you to customize the brand of the dashboard.
Horizon provides a set of core classes and reusable templates and tools.
This example deployment uses an Apache web server



Compute (Nova)

Host and manage cloud computing systems. OpenStack Compute is a major part of an Infrastructure-as-a-Service (IaaS) system.

OpenStack Compute interacts with OpenStack Identity for authentication, OpenStack Image Service for disk and server images, and OpenStack dashboard for the user and administrative interface.
OpenStack Compute can scale horizontally on standard hardware, and download images to launch instances.

Identity (Keystone)

Tracking users and their permissions.

Providing a catalog of available services with their API endpoints.

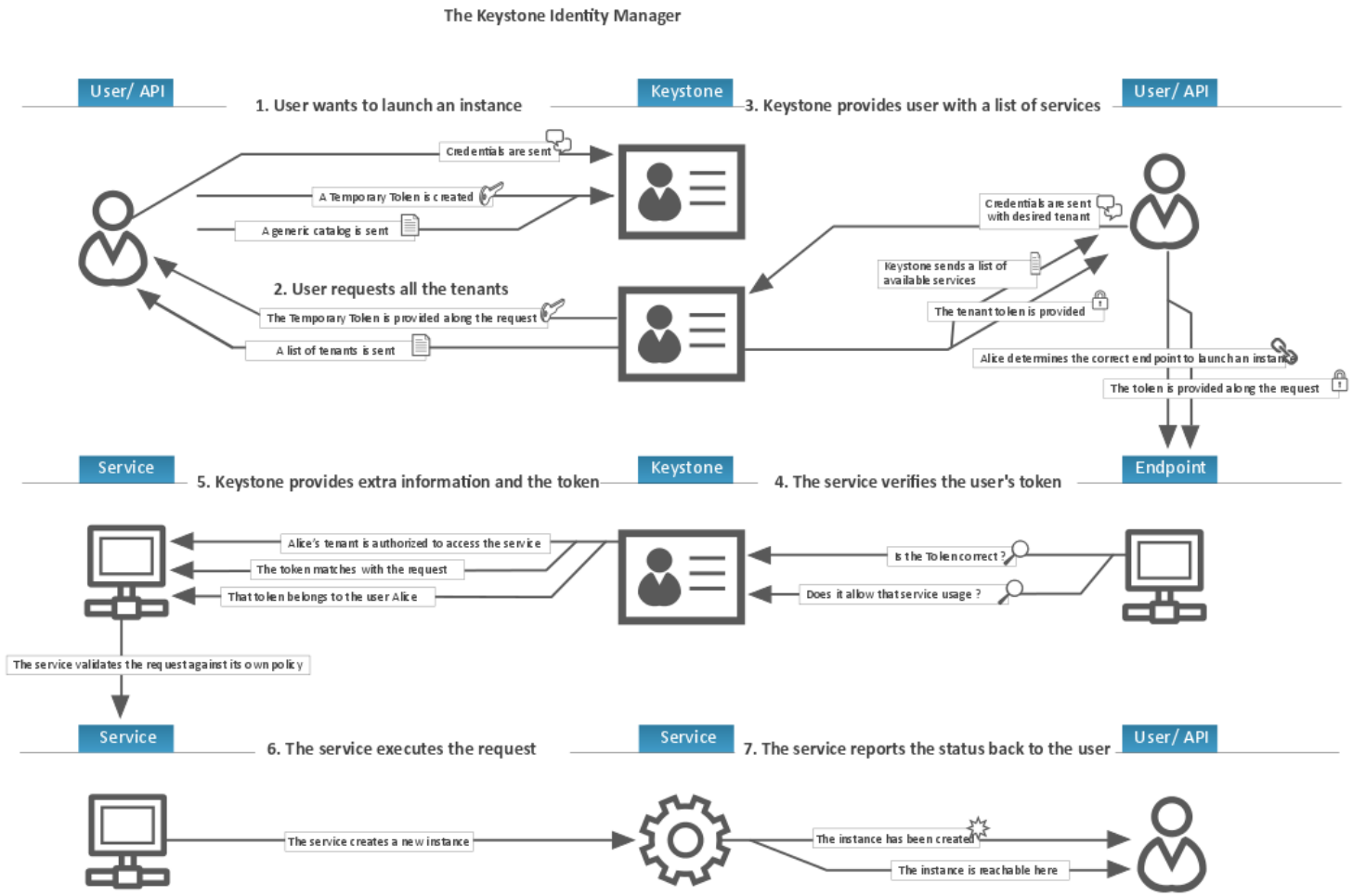


Image (Glance)

Enables users to discover, register, and retrieve virtual machine images. It offers a REST API that enables you to query virtual machine image metadata and retrieve an actual image.

You can store virtual machine images made available through the Image Service in a variety of locations, from simple file systems to object-storage systems like OpenStack Object Storage.

Various repository types are supported including normal file systems, Object Storage, RADOS block devices, HTTP, and Amazon S3. Note that some repositories will only support read-only usage.

Block Storage (Cinder)

Provides block storage devices to instances using various backends. The Block Storage API and scheduler services run on the controller node and the volume service runs on one or more storage nodes. Storage nodes provide volumes to instances using local block storage devices or SAN/NAS backends with the appropriate drivers.

The OpenStack Block Storage service (cinder) adds persistent storage to a virtual machine. Block Storage provides an infrastructure for managing volumes, and interacts with OpenStack Compute to provide volumes for instances. The service also enables management of volume snapshots, and volume types.

Networking (Neutron)

Manages all networking facets for the Virtual Networking Infrastructure (VNI) and the access layer aspects of the Physical Networking Infrastructure (PNI) in your OpenStack environment.

Enables tenants to create advanced virtual network topologies including services such as firewalls, load balancers, and virtual private networks (VPNs).

Service	Default port	Used by
HTTP	80	OpenStack dashboard (`Horizon`) when it is not configured to use secure access.
HTTP alternate	8080	OpenStack Object Storage (`swift`) service.
HTTPS	443	Any OpenStack service that is enabled for SSL, especially secure-access dashboard.
rsync	873	OpenStack Object Storage. Required.
iSCSI target	3260	OpenStack Block Storage. Required.
MySQL database service	3306	Most OpenStack components.
Message Broker (AMQP traffic)	5672	OpenStack Block Storage, Networking, Orchestration, and Compute.

Default listening ports and Firewalls considerations

OpenStack service	Default ports	Port type
Block Storage (cinder)	8776	publicurl and adminurl
Compute (nova) endpoints	8774	publicurl and adminurl
Compute API (nova-api)	8773, 8775	
Compute ports for access to virtual machine consoles	5900-5999	
Compute VNC proxy for browsers (openstack-nova-novncproxy)	6080	
Compute VNC proxy for traditional VNC clients (openstack-nova-xvpvncproxy)	6081	
Proxy port for HTML5 console used by Compute service	6082	
Identity service (keystone) administrative endpoint	35357	adminurl
Identity service public endpoint	5000	publicurl
Image Service (glance) API	9292	publicurl and adminurl
Image Service registry	9191	
Networking (neutron)	9696	publicurl and adminurl
Object Storage (swift)	6000, 6001, 6002	
Orchestration (heat) endpoint	8004	publicurl and adminurl
Orchestration AWS CloudFormation-compatible API (openstack-heat-api-cfn)	8000	
Orchestration AWS CloudWatch-compatible API (openstack-heat-api-cloudwatch)	8003	
Telemetry (ceilometer)	8777	publicurl and adminurl

Log Files

Log file	Service name (CentOS/Fedora/openSUSE/Red Hat Enterprise Linux/SUSE Linux Enterprise)
api.log	openstack-nova-api
cert.log [a]	openstack-nova-cert
compute.log	openstack-nova-compute
conductor.log	openstack-nova-conductor
consoleauth.log	openstack-nova-consoleauth
network.log[b]	openstack-nova-network
nova-manage.log	nova-manage
scheduler.log	openstack-nova-scheduler

Log directories:

Horizon: /var/log/dashboard
Nova: /var/log/nova/
Keystone: /var/log/keystone
Glance: /var/log/glance
Cinder: /var/log/cinder
Neutron: /var/log/neutron

Demo Deployment

```
==> controller01: Applying 192.168.33.20_mariadb.pp
192.168.33.20_amqp.pp:                                 [ DONE ]
192.168.33.20_mariadb.pp:                              [ DONE ]
==> controller01: Applying 192.168.33.20_keystone.pp
==> controller01: Applying 192.168.33.20_glance.pp
==> controller01: Applying 192.168.33.20_cinder.pp
192.168.33.20_cinder.pp:                               [ DONE ]
192.168.33.20_glance.pp:                               [ DONE ]
192.168.33.20_keystone.pp:                             [ DONE ]
==> controller01: Applying 192.168.33.20_api_nova.pp
192.168.33.20_api_nova.pp:                             [ DONE ]
==> controller01: Applying 192.168.33.20_nova.pp
==> controller01: Applying 192.168.33.21_nova.pp
192.168.33.20_nova.pp:                                 [ DONE ]
192.168.33.21_nova.pp:                                 [ DONE ]
==> controller01: Applying 192.168.33.20_neutron.pp
==> controller01: Applying 192.168.33.21_neutron.pp
==> controller01: Applying 192.168.33.22_neutron.pp
192.168.33.21_neutron.pp:                              [ DONE ]
192.168.33.22_neutron.pp:                              [ DONE ]
192.168.33.20_neutron.pp:                              [ DONE ]
==> controller01: Applying 192.168.33.22_neutron_fwaas.pp
==> controller01: Applying 192.168.33.20_osclient.pp
==> controller01: Applying 192.168.33.20_horizon.pp
==> controller01: Applying 192.168.33.20_nagios.pp
==> controller01: Applying 192.168.33.20_nagios_nrpe.pp
==> controller01: Applying 192.168.33.21_nagios_nrpe.pp
==> controller01: Applying 192.168.33.22_nagios_nrpe.pp
192.168.33.21_nagios_nrpe.pp:                          [ DONE ]
192.168.33.22_neutron_fwaas.pp:                        [ DONE ]
192.168.33.20_osclient.pp:                             [ DONE ]
192.168.33.22_nagios_nrpe.pp:                          [ DONE ]
192.168.33.20_horizon.pp:                              [ DONE ]
192.168.33.20_nagios.pp:                               [ DONE ]
192.168.33.20_nagios_nrpe.pp:                          [ DONE ]
==> controller01: Applying 192.168.33.20_postscript.pp
==> controller01: Applying 192.168.33.21_postscript.pp
==> controller01: Applying 192.168.33.22_postscript.pp
192.168.33.20_postscript.pp:                           [ DONE ]
192.168.33.22_postscript.pp:                           [ DONE ]
192.168.33.21_postscript.pp:                           [ DONE ]
==> controller01: Applying Puppet manifests                          [ DONE ]
==> controller01: Finalizing                                         [ DONE ]
==> controller01:
==> controller01:   **** Installation completed successfully ******
==> controller01:
==> controller01:
==> controller01: Additional information:
==> controller01:   * File /root/keystonerc_admin has been created on OpenStack client host 192.168.33.20. To use the command lin
==> controller01:   * To access the OpenStack Dashboard browse to http://192.168.33.20/dashboard .
==> controller01: Please, find your login credentials stored in the keystonerc_admin in your home directory.
==> controller01:   * To use Nagios, browse to http://192.168.33.20/nagios username: nagiosadmin, password: nagiosadmin
==> controller01:   * Because of the kernel update the host 192.168.33.20 requires reboot.
==> controller01:   * Because of the kernel update the host 192.168.33.21 requires reboot.
==> controller01:   * Because of the kernel update the host 192.168.33.22 requires reboot.
==> controller01:   * The installation log file is available at: /var/tmp/packstack/20150105-112811-8nCARQ/openstack-setup.log
==> controller01:   * The generated manifests are available at: /var/tmp/packstack/20150105-112811-8nCARQ/manifests
```

Demo
Deployment