

Introducción al Cloud Computing



IES Gonzalo Nazareno
CONSEJERÍA DE EDUCACIÓN

Alberto Molina Coballes
Jesús Moreno León
José Domingo Muñoz Rodríguez
IES Gonzalo Nazareno
Dos Hermanas (Sevilla)



Jornadas "Cloud Computing con OpenStack como herramienta didáctica".

Inicios de OpenStack



- Cloud propio desde 2005
 - Cloud servers (IaaS)
 - Cloud files (SaaS)
 - Este software cambia a licencia libre en Abril 2010
-



- Comienza a utilizar Eucalyptus, pero lo descarta por no ser completamente libre (es "open core")
 - Crea el software para IaaS Nebula
 - Nebula cambia a licencia libre en Mayo 2010
-



- Nasa y Rackspace lo inician en Junio de 2010
- Dos componentes principales:
 - OpenStack Compute (nova), deriva de Nebula
 - OpenStack Object Store (swift), deriva de cloud files

Objetivo de OpenStack

“Crear una plataforma en software libre para cloud computing que cumpla con las necesidades de los proveedores de nubes públicas y privadas, independientemente de su tamaño, que sea fácil de implementar y masivamente escalable.”

Principios fundacionales de OpenStack

- Licencia Apache 2.0, no existe versión “enterprise”
- Proceso de diseño abierto
- Repositorios públicos de código fuente
- Todos los procesos de desarrollo deben estar documentados y ser transparentes
- Orientado para adoptar estándares abiertos
- Diseño modular que permite flexibilidad mediante el uso de APIs

OpenStack es libre y abierto

- OpenStack es un proyecto con licencia libre (Apache)
- Diseño abierto:
 - <http://blueprints.launchpad.net/openstack>
 - <http://www.openstack.org/summit/san-diego-2012/>
- Desarrollo abierto:
 - <http://launchpad.net/openstack> y <http://github.com/openstack/>
 - Lenguaje de programación Python
 - <http://bugs.launchpad.net/openstack/>
- Comunidad abierta:
 - <http://www.openstack.org/community/>
 - <http://www.openstack.org/foundation/companies/>
 - <http://lists.openstack.org>
- Comunidad + empresas

Versiones de OpenStack

- Proyecto muy nuevo, pero con un fuerte ritmo de desarrollo
 - Austin 21 Octubre 2010
 - Bexar 3 Febrero 2011
 - Cactus 15 Abril 2011
 - Diablo 22 Septiembre 2011 (Publicación semestral)
 - Essex 5 Abril 2012
 - Folsom 27 Septiembre 2012
 - Grizzly Previsto 4 Abril 2013
- Está previsto que se publiquen dos versiones al año
- Hasta ahora cada versión incluye importantes modificaciones respecto a la anterior
- Essex ha sido la primera versión “completa”
- Desde Cactus, el ritmo de publicación se acopla al de Ubuntu

OpenStack Essex (2012.1)

- ¿Por qué es importante Essex?
 - Primera versión completa de OpenStack para usar en producción
 - Presente en Ubuntu 12.04 LTS. La próxima versión LTS será en 2014
 - Presente en Debian Wheezy (próxima estable). Debian wheezy soportará OpenStack Folsom en backport
- Componentes de OpenStack Essex:
 - OpenStack Compute (nova)
 - OpenStack Object Store (swift)
 - OpenStack Image (glance)
 - OpenStack Identity (keystone) ← Nuevo en Essex
 - OpenStack Dashboard (horizon) ← Nuevo en Essex

<http://wiki.openstack.org/ReleaseNotes/Essex>

OpenStack Folsom (2012.2)

- OpenStack tiene un ritmo de publicación semestral, difícil de incluir en la publicación de distribuciones “estables”. Ubuntu LTS o Debian se publican cada dos años.
- Incluye mejoras en bastantes componentes de OpenStack
- Incluido en Ubuntu 12.10
- Se incluirá en Debian Wheezy mediante backport (repositorio extra menos estable)
- Las principales novedades son la aparición de dos nuevos componentes principales:
 - OpenStack Network Service (Quantum)
 - OpenStack Block Storage (Cinder)

<http://wiki.openstack.org/ReleaseNotes/Folsom>

¿Es OpenStack una buena opción?

- A pesar de ser un proyecto muy nuevo, tiene un ritmo de desarrollo muy fuerte
- Cuenta con la mayor comunidad de desarrolladores dentro de los proyectos de software libre para cloud computing (~200 en Essex)
- Más de 100 empresas participan en el desarrollo en diferente medida
- Esto es consecuencia de la orientación libre y abierta del proyecto
- Has oído hablar de OpenStack con motivo, esto no es vaporware

Google Trends:



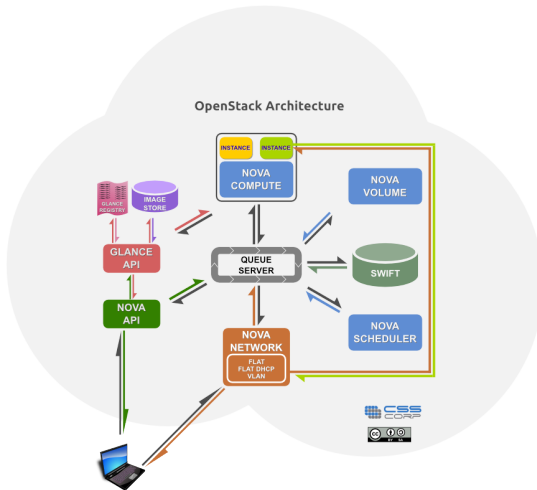
Servicios de OpenStack nova

- Nova es el componente principal de OpenStack y está compuesto por varios servicios independientes:
 - nova-api** Encargado de aceptar las peticiones de los usuarios o del resto de componentes de OpenStack mediante una API RESTful
 - nova-scheduler** Encargado de planificar la ejecución de las instancias en los diferentes nodos del cloud
 - nova-compute** Encargado de ejecutar una instancia sobre un hipervisor
 - nova-network** Encargado de la comunicación de la instancia con el exterior
 - nova-volume** Encargado de gestionar los volúmenes asociados a las instancias
- Los componentes de nova se comunican entre sí mediante AMQP

Funcionamiento típico de OpenStack

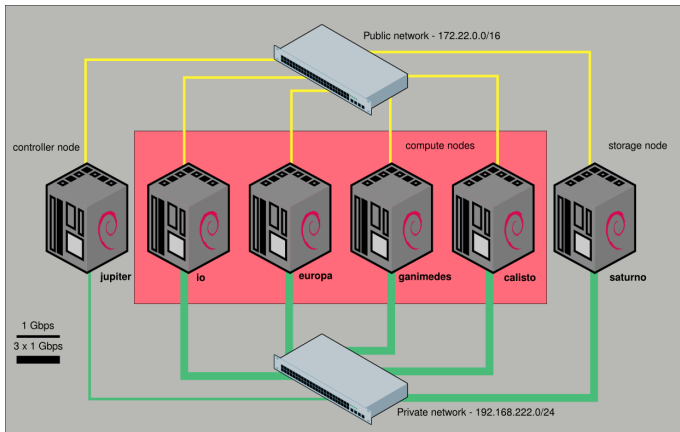
- Un usuario interactúa con la API de nova (bien directamente o indirectamente a través de horizon) para ejecutar una instancia.
- nova-api le pedirá que se autentique previamente con keystone
- Una vez autenticado le mostrará las imágenes disponibles en glance
- Cuando seleccione una imagen y unas características para la instancia, se enviará a nova-scheduler la petición
- Nova-scheduler determinará en que nodo debe ejecutarse la instancia
- Nova-compute del nodo seleccionado se encargará de ejecutar la instancia sobre el hipervisor que disponga
- Nova-network realizará las configuraciones necesarias en la red
- Nova-volume se encargará de gestionar en su caso los volúmenes asociados a la instancia

Funcionamiento de OpenStack



Instalación de componentes de OpenStack

- Dependiendo del número de equipos del cloud y la configuración de red, se instalarán en cada nodo diferentes componentes, p. ej.:



APIs

- Cada componente de OpenStack ofrecen una API RESTful
- Las APIs se pueden utilizar con XML o JSON (por defecto JSON)
- Esto hace OpenStack extensible y adaptable a cada entorno

```
$ nova --debug list
connect: (172.22.222.1, 5000)
send: 'POST /v2.0/tokens HTTP/1.1\r\nHost: 172.22.222.1:5000\r\nContent-Length:124
\r\ncontent-type: application/json\r\naccept-encoding: gzip, deflate\r\naccept: ap
plication/json\r\nuser-agent: python-novaclient\r\n\r\n{"auth": {"tenantName": "te
st", "passwordCredentials": {"username": "user", "password": "testpass"}}}'
reply: 'HTTP/1.1 200 OK\r\n'
connect: (172.22.222.1, 8774)
send: u'GET /v2/aaaaaaaa5894473c8a98f89a895c6b2c/servers/detail HTTP/1.1\r\nHost:
172.22.222.1:8774\r\nnx-auth-project-id: test\r\nnx-auth-token: e9233fef4ce34ee49f7d
b1aaaaaaaa13f\r\naccept-encoding: gzip, deflate\r\naccept: application/json\r\nuser
-agent: python-novaclient\r\n\r\n'
reply: 'HTTP/1.1 200 OK\r\n'
```

ID	Name	Status	Networks
b1724bd0-34f4-4bf1-9444-110eb3531602	demo9	VERIFY_RESIZE	vlan5=10.0.5.6
e82814aa-fb1d-4c29-81ab-c39f99184413	demo10	ACTIVE	vlan5=10.0.5.3