Grid y Computación de Altas Prestaciones

GRyCAP

Tema 3.3: Introducción a Cloud Computing – Tecnologías para Clouds “on premises”

Master de Computación Paralela, Distribuida y Grid

contenido

Grid y Computación de Altas Prestaciones GRyCAP

1. Introducción
   1. ¿Qué es un Gestor Cloud?
   2. Clouds Privados vs Cloud Públicos
2. Integración de infraestructuras
3. Estándares
   1. OCCI
   2. CDMI
4. Herramientas para el Despliegue de Infraestructuras Clouds
   1. OpenStack
5. Conclusiones

Introducción

Grid y Computación de Altas Prestaciones
GRyCAP

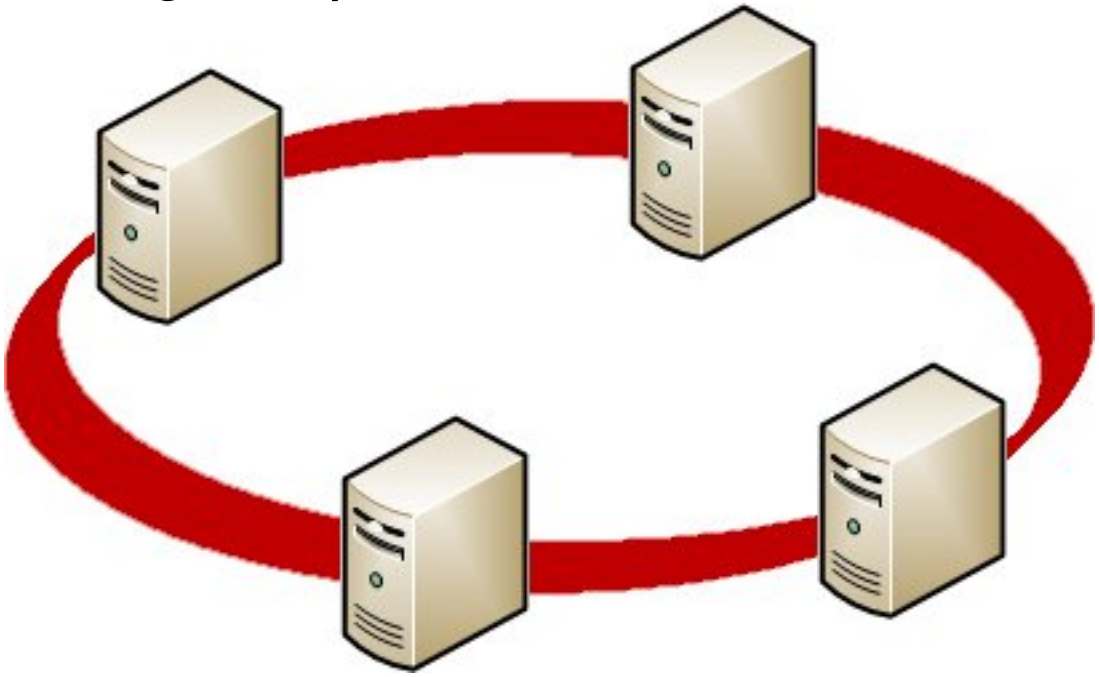
- Ya sabemos muchas cosas del Cloud Computing
  - Sabemos qué es el SaaS
    - Básicamente son aplicaciones que podemos usar
  - Tenemos idea de qué es el PaaS
    - Una plataforma de servicios y utilidades para crear SaaS
  - Sabemos qué es el IaaS
    - Máquinas virtuales que lanzamos en algún sitio, acceso a red, disco, etc.

Introducción

- Pero... ¿cómo se "construye un Cloud”?
  - El usuario quiere poder lanzar a ejecución su máquina virtual o su aplicación y no preocuparse de nada más

¿Qué es un Gestor Cloud?

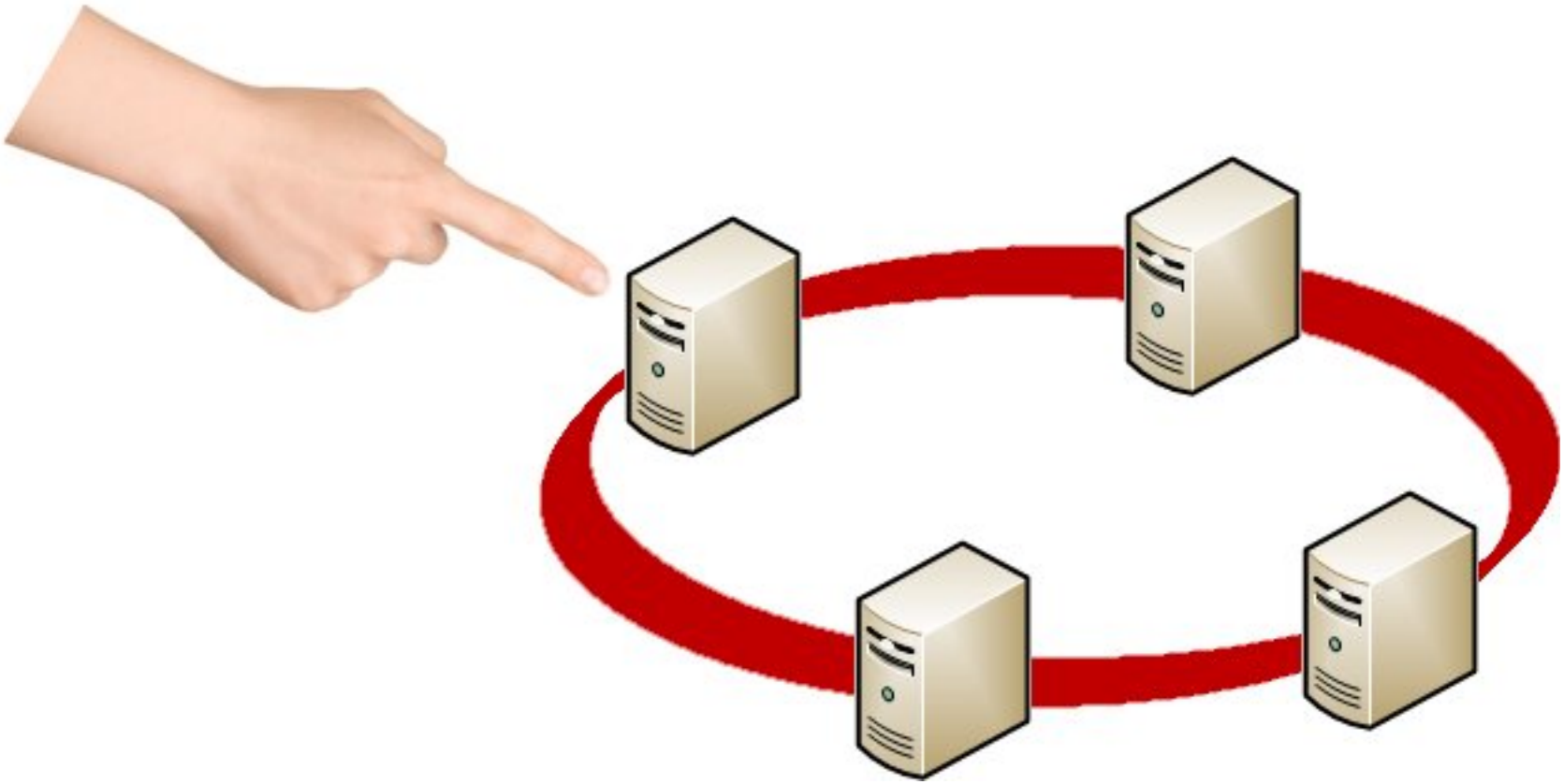
- Tenemos una serie de servidores de virtualización y queremos lanzar una máquina virtual
  - En ellos está instalado algún hipervisor como VMWare, KVM, Xen, etc.



¿Qué es un Gestor Cloud?

Grid y Computación de Altas Prestaciones GRyCAP

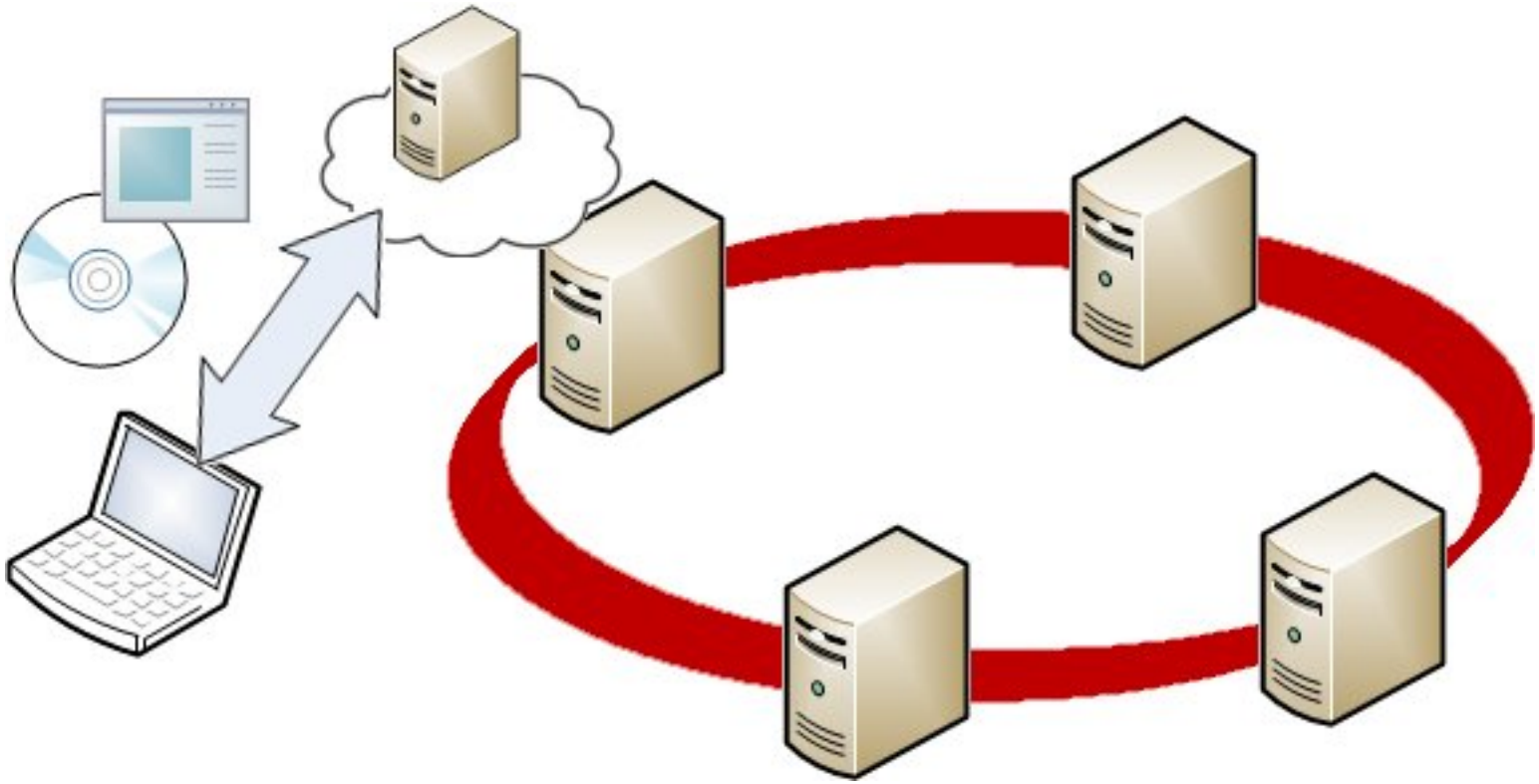
- Decidimos en qué servidor vamos a alojar la máquina virtual



¿Qué es un Gestor Cloud?

Grid y Computación de Altas Prestaciones GRyCAP

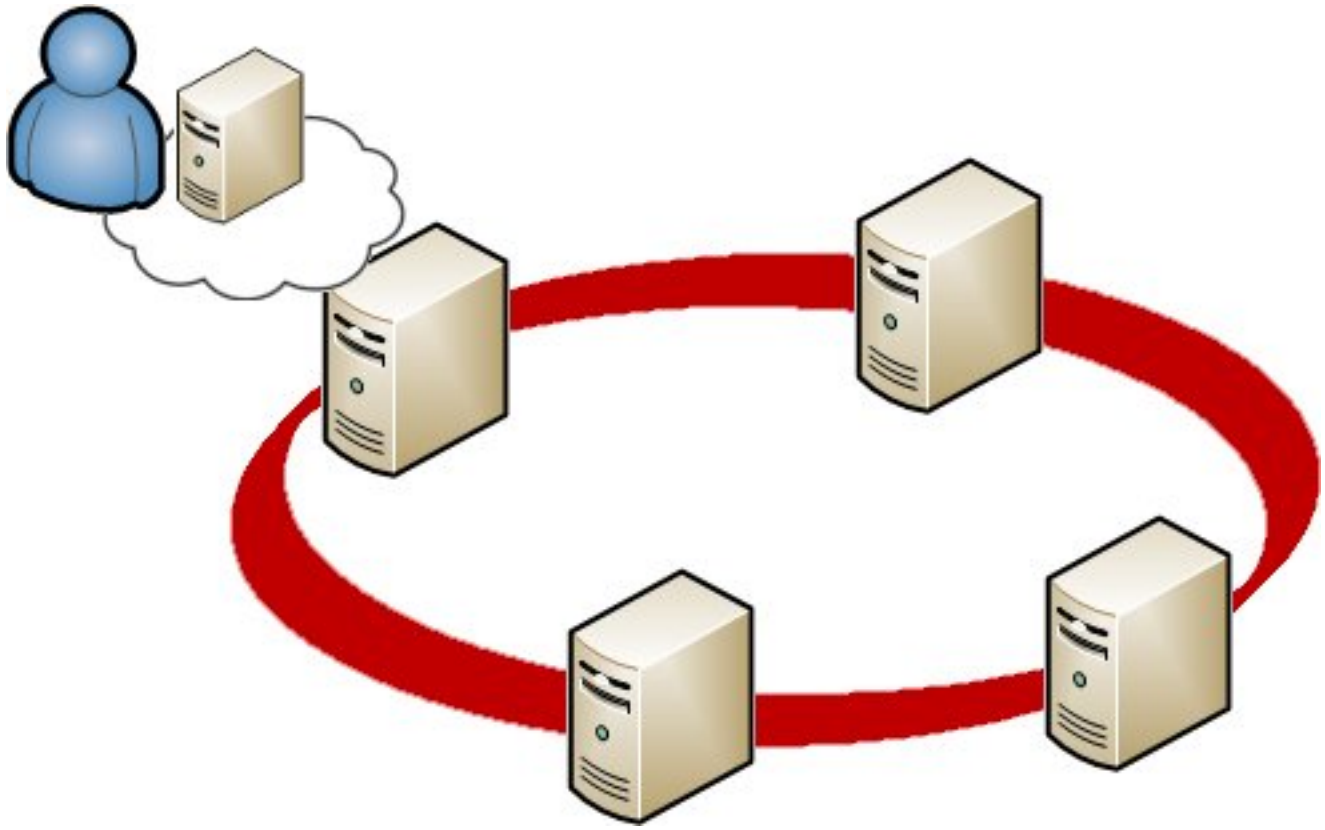
- Se despliega la máquina virtual
  - Con el sistema operativo, las aplicaciones, red, etc.



¿Qué es un Gestor Cloud?

Grid y Computación de Altas Prestaciones GRyCAP

- Se proporciona la IP de la máquina al usuario para que pueda trabajar con ella (IaaS).



¿Qué es un Gestor Cloud?

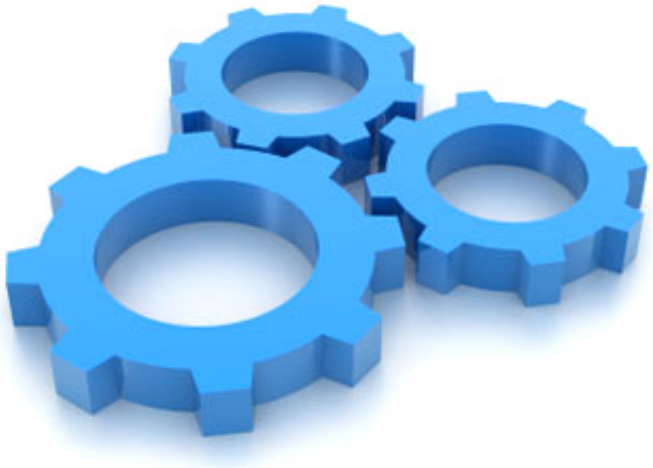
Grid y Computación de Altas Prestaciones GRyCAP

- Y así ya tendríamos al usuario en el Cloud

  (... y además, ha usado un IaaS)

¿Qué es un Gestor Cloud?

- Esto parece muy sencillo de automatizar
  1. Nos piden una máquina virtual (con unas características).
  2. Averiguamos donde es mejor ponerla en funcionamiento.
  3. La ponemos en funcionamiento (y le proporcionamos las características solicitadas).
  4. Le entregamos al usuario su máquina virtual.



¿Qué es un Gestor Cloud?

- Una vez podemos hacer esto se plantean distintas posibilidades
  - Establecer las políticas de selección de los servidores que alojan cada máquina virtual.
  - Gestionar el ciclo de vida de una máquina
    - apagado, pausa, parado, salvar el estado, etc.
  - Conectar distintas máquinas entre ellas
  - Establecer la conectividad de red y las posibilidades de acceso
    - IPs públicas/privadas, firewalls, aislamiento, etc.
  - Migrar las máquinas virtuales de un host a otro
    - para poner en mantenimiento un equipo, ahorrar energía, etc.

¿Qué es un Gestor Cloud?

Grid y Computación de Altas Prestaciones GRyCAP

- Una vez resueltas todas estas tareas, tendremos un gestor de nuestra plataforma Cloud que nos permitirá proporcionar IaaS.
- El uso posterior que hagamos, los mecanismos de acceso que proporcionemos, etc. determinará si es un Cloud público o privado.

Clouds Privados y Públicos

- Unas de las diferencias principales entre los Clouds públicos y los Clouds privados
  - A qué usuarios se permite el acceso
    - En un Cloud privado se permitirá el acceso a usuarios “cercanos” a quien ofrece el servicio (empleados, colaboradores, etc.).
    - En un Cloud público se permitirá el acceso, generalmente, a quien pague por los servicios ofertados.
  - Los tipos de servicios que se ofrecen hacia estos usuarios (es proporcional a la tipología de usuarios).
    - En un Cloud público los usuarios no estarán dispuestos a aceptar caídas de servicio, falta de prestaciones, pérdidas de ficheros, interferencias entre máquinas de distintos usuarios, etc.
    - En un Cloud privado se puede informar a los usuarios para que se planifiquen porque va a ocurrir una interrupción de servicio, las máquinas pueden estar en la misma subred, etc.

Clouds Privados

- Aquí nos vamos a centrar en la creación de Clouds con Recursos Propios con el objetivo de crear Clouds Privados
  - Qué dicen que es un “cloud privado”
    - “Above the Clouds: A Berkeley View of Cloud Computing”, Michael Armbrust, et al.
      - “We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public.”
    - Definición del NIST
      - “The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units).”
- A menudo se utiliza el término On-Premise Private Clouds
  - On-Premise se refiere dentro de las instalaciones (premises) de la propia organización

Gestores CLOUD de tipo iaas

- OpenNebula
  - “OpenNebula is the open-source industry standard for data center virtualization, offering the most feature-rich, flexible solution for the comprehensive, complete management of virtualized data centers to enable on-premise IaaS clouds in existing infrastructures.”
- OpenStack
  - “OpenStack is a global collaboration of developers and cloud computing technologists producing the ubiquitous open source cloud computing platform for public and private clouds”
- Eucalyptus
  - “Eucalyptus enables the creation of on-premise Infrastructure as a Service clouds, with no requirements for retooling the organization's existing IT infrastructure or for introducing any specialized hardware”
- CloudStack
  - “CloudStack is open source software written in java that is designed to deploy and manage large networks of virtual machines, as a highly available, scalable cloud computing platform.”
- Nimbus, Entropy, AbiQuo, Enomaly, nimbula, AeolusProject, Okeanos, etc.

disclaimer

- OpenNebula, OpenStack, etc., son herramientas para el despliegue de Clouds sobre una infraestructura física existente.
  - No se utilizan exclusivamente para el despliegue de Clouds privados.
- Una empresa puede utilizar OpenNebula u OpenStack (entre otros) para desplegar un Cloud en sus instalaciones (on-premise) y luego ofrecer acceso a su Cloud al público mediante un modelo de pago por uso.
  - Sería un Cloud público desplegado con ayuda de una herramienta que típicamente se utilizar para desplegar Clouds privados.

Gestores Cloud

- OpenNebula
  - Es uno de los sistemas más conocidos por la comunidad científica.
  - Es un sistema “de referencia” para la construcción de centros de datos virtuales.
  - Ha sido financiado por la comisión Europea en distintos proyectos y convocatorias.
  - Desarrollado en la Univ. Complutense de Madrid, hay una empresa detrás (C12G Labs) que da soporte y contribuye de forma activa al proyecto.
  - Se estudia con detalle en la asignatura de IAC.
- OpenStack
  - Es otro de los sistemas más conocidos por la comunidad.
  - Hay un gran número de empresas muy relevantes apoyándolo (RackSpace, HP, Dell, Microsoft, Intel, Cisco, Citrix, etc.)
    - Se ha creado una fundación que de soporte a largo plazo.
  - Proviene de un desarrollo propio de la NASA y de RackSpace y ha ido ganando protagonismo en muy poco tiempo.

Cloud vs Clouds

- Cloud Computing
  - Un estilo de computación donde los recursos TIC se proporcionan de forma masivamente escalable a clientes externos a través de Internet.
- Premisa:
  - Ninguna infraestructura cloud puede crear una aparente disponibilidad infinita de recursos capaz de servir un número masivo de usuarios en cualquier momento y desde cualquier parte.
- En la actualidad se está investigando en tecnologías cloud avanzadas que puedan ser capaces de permitir construir federaciones de clouds.

Grid y Computación de Altas Prestaciones
GRyCAP

ONE, Open Stack, Ninbus, … Ya estamos otra vez…

¿Y yo qué gestor elijo?

SOLUCIÓN: ESTÁNDARES

Minimizando Riesgos: Interoperabilidad

Grid y Computación de Altas Prestaciones
GRyCAP

Data Portability	Standards	Ease of Migration & Deployment	Developer Choice
How can I keep control over my data?	**What technology standards are important for Cloud Platforms?**	**Will your Cloud Platforms help me migrate my existing technology investments to the cloud?**	**How can I leverage my developers' and IT professionals' skills in the cloud?**
Customers own their data and Cloud Platforms should make it easy and efficient to securely move customers data in and out.	Cloud Platforms should reuse existing and commonly used standards when appropriate and may lead to the creation of new standards where existing standards are not sufficient.	Cloud Platforms should provide, for existing technologies that are appropriate to the cloud, a migration path that preserves, in a secure way, existing investments in applications and IT resources.	Cloud Platforms should enable developer choice in tools, languages and runtimes.

http://www.microsoft.com/cloud/interop/

Minimizando Riesgos: Estándares

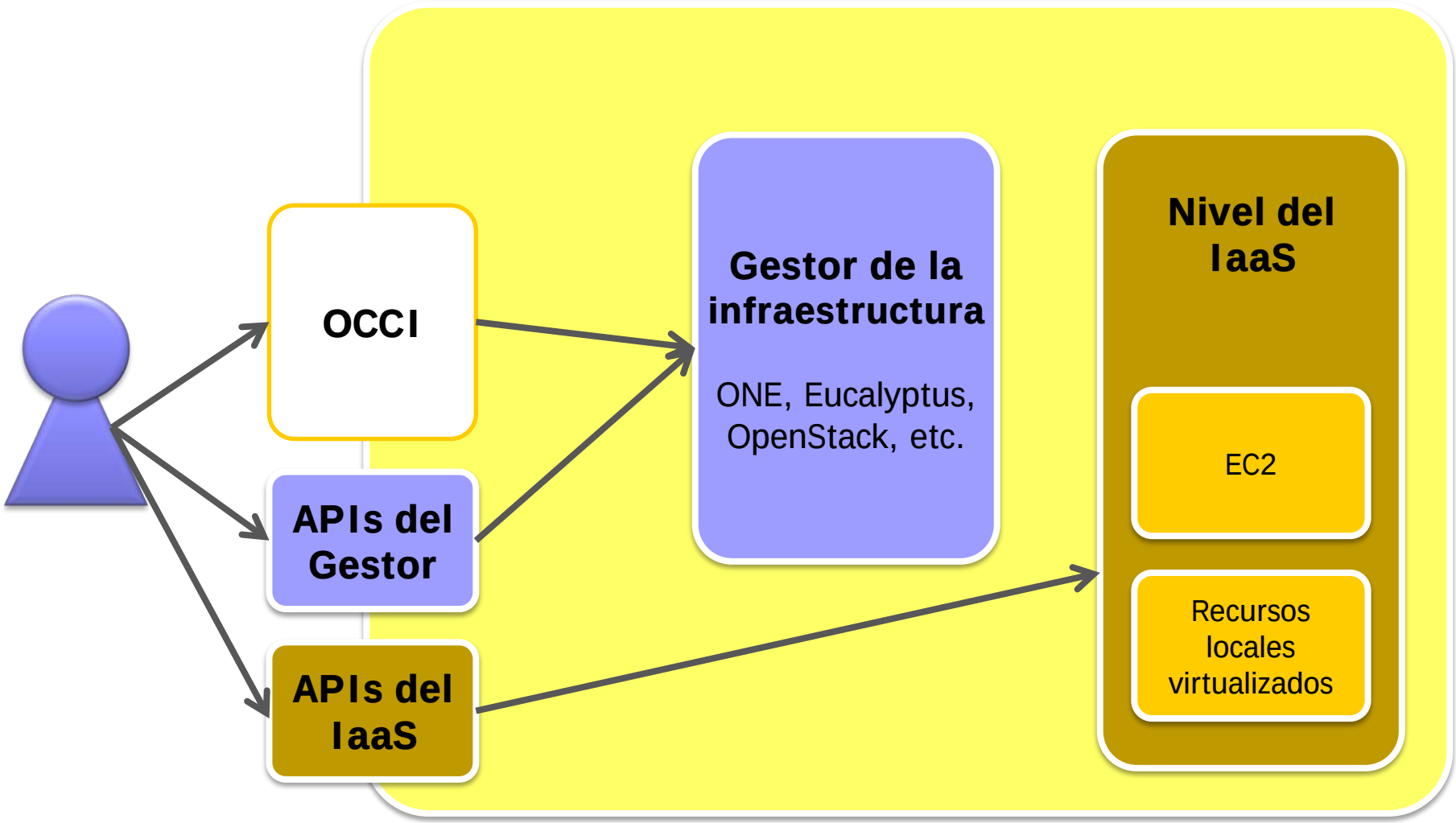
- Uso de estándardes en la medida de lo posible
  - OCCI, CDMI, OGF (BES, JSDL), OASIS (WS-*, SAML, RUS, UR).
- Mínimo esfuerzo para la migración de una plataforma a otra
  - P.e. los datos deberían ser transportables de forma sencilla y programática como un bloque.
- Soporte de diferentes lenguajes y entornos de trabajo.

Estándares en Cloud Computing

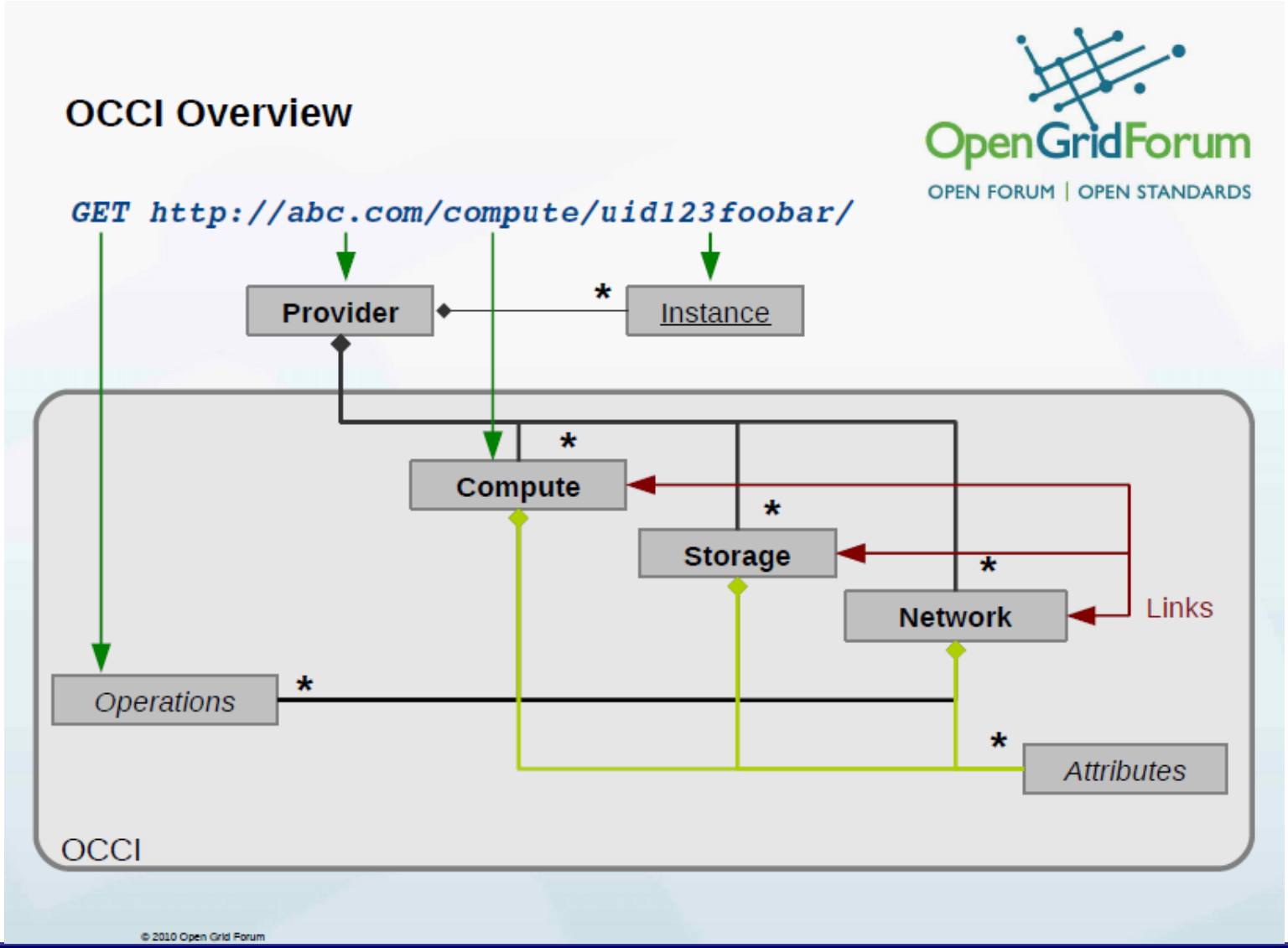
- OCCI (Open Cloud Computing Interface)
- OCCI proporciona un interfaz, normalmente sobre HTTP para interaccionar con gestores de infraestructuras virtualizadas
  - Como Open Nebula, Euclayptus, OpenStack, etc.
  - Se combina con OVF para permitir la interoperabilidad entre diferentes gestores de máquinas virtuales.
- OCCI se creó originalmente como un API para gestionar remotamente IaaS
  - Permitiendo un modelo interoperable para el despliegue elasticidad autónoma y monitorización.
  - Interoperabilidad y extensibilidad.

Estándares en Cloud Computing: OCCI

Grid y Computación de Altas Prestaciones
GRyCAP



El Estándar en global

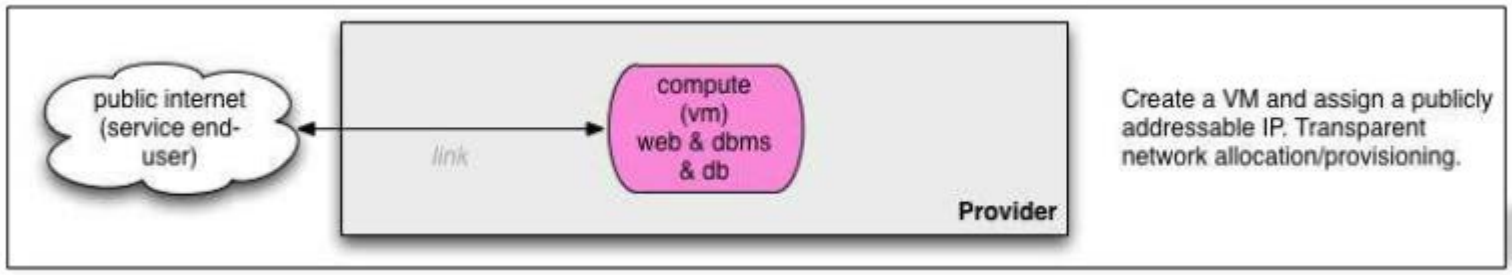


Ejemplo Simple

Grid y Computación de Altas Prestaciones
GRyCAP

```
> POST /compute/ HTTP/1.1#
> User-Agent: curl/7.21.1 (i386-pc-solaris2.11) libcurl/7.21.1
OpenSSL/0.9.8o zlib/1.2.3 libidn/1.9
> Host: localhost:8888
> Accept: */*
> Content-type: text/occi
> Category: compute;
scheme="http://schemas.ogf.org/occi/infrastructure#"
> Category: ubuntu; scheme="http://example.com/templates/os#"
> Category: small; scheme="http://example.com/templates/compute#"
>

< HTTP/1.1 201 OK
< Content-Length: 2
< Content-Type: text/html; charset=UTF-8
< Location: http://localhost:8888/compute/ec7e854d-5b1c-cb24-
cb57-875b0a404fd1
< Server: pyocci OCCI/1.1
<
OK
```



Estándares en Cloud Computing

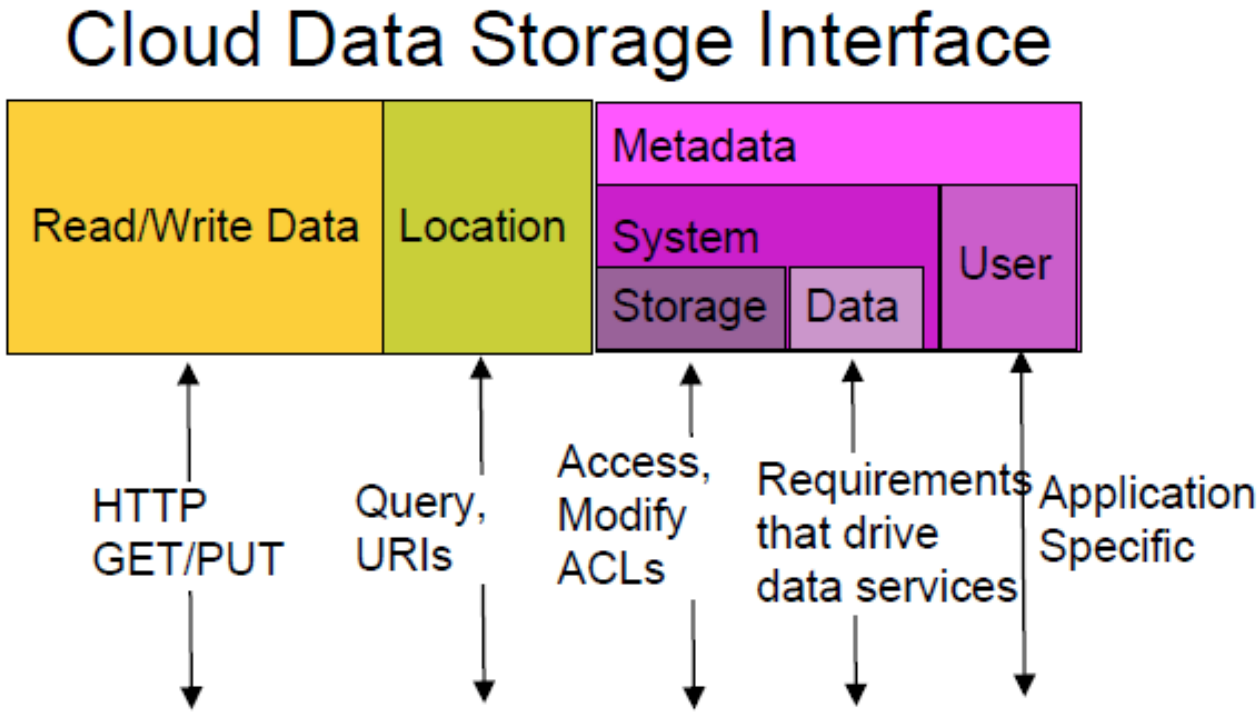
Grid y Computación de Altas Prestaciones GRyCAP

- CDMI (Cloud Data Management Interface)
  - Estándar promocionado por SNIA.
  - Define un modelo de interfaz para diferentes tipos de objetos de almacenamiento.
  - Permite la definición de metadatos asociados.
  - Propone un interfaz basado en REST.
  - Los objetos soportados son
    - Objects: Equivalente a Ficheros binarios.
    - Queues: Colas de mensajes para comunicar instancias en el cloud.
    - Documents: Equivalente a Tablas, en proceso de definición.

Estándares en Cloud Computing

Grid y Computación de Altas Prestaciones GRyCAP

- Modelo de Dominio de Recursos de CDMI



Grid y Computación de Altas Prestaciones
GRyCAP

OpenStack

introducción

- De acuerdo a la página web: http://www.openstack.org
  - OpenStack consiste en software de código abierto para la construcción de Clouds públicos y privados.
  - OpenStack ofrece un Sistema Operativo Masivamente Escalable y de Código Abierto para la Nube.
- Permite gestionar recursos de cómputo, almacenamiento y de red de un centro de datos, gestionado a través tanto de un panel de control como de una CLI.
- Ofrece como característica adicional a otras herramientas la gestión de almacenamiento de objetos.

Origen de OpenStack

- Inicialmente fundada por Rackspace y la NASA.
- 2012: OpenStack Foundation: Fundación independiente
  - Proteger, reforzar y promover OpenStack.
  - Abierta: Más de 7000 miembros individuales y más de 850 organizaciones.
  - Actualmente está la versión Icehouse, aunque la versión Juno se liberará el mes que viene.
- Código abierto y disponible bajo licencia Apache License
  - Más de 31029 contribuyentes, 177 paises, 543 empresas y 20M de líneas de código.
  - Código revisado por pares, discutido y testeado (test unitarios y funcionales) antes de integrarse con el resto.
  - Cualquiera puede contribuir.

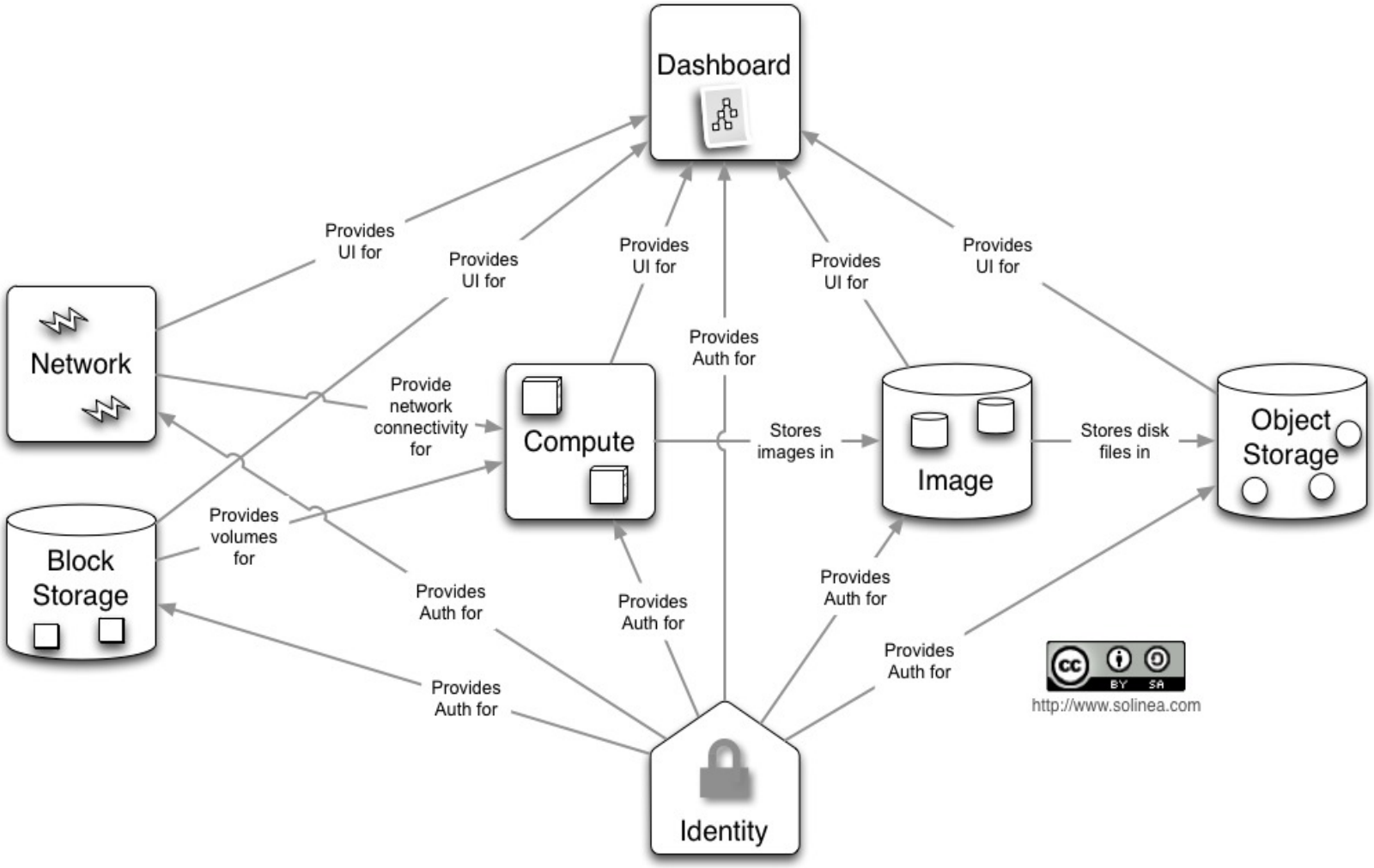
Componentes de la release Kilo

Grid y Computación de Altas Prestaciones GRyCAP

- Nova: proporciona servidores virtuales bajo demanda.
- Glance: indexa las imágenes de máquinas virtuales y su metadata.
- Swift: almacenamiento de datos como objetos binarios.
- Horizon: un “dashboard” a través de web para los servicios de OpenStack.
- Keystone: sistema de autenticación y publicación de los servicios.
- Neutron: proporciona redes virtuales.
- Cinder: proporciona volúmenes de almacenamiento para VMs.
- Ceilometer: monitoriza el estado de los recursos inactivos.
- Heat: permite la orquestación de diferentes servicios a partir de un lenguaje de especificación.

Modelo de interacción de OpenStack

Grid y Computación de Altas Prestaciones
GRyCAP

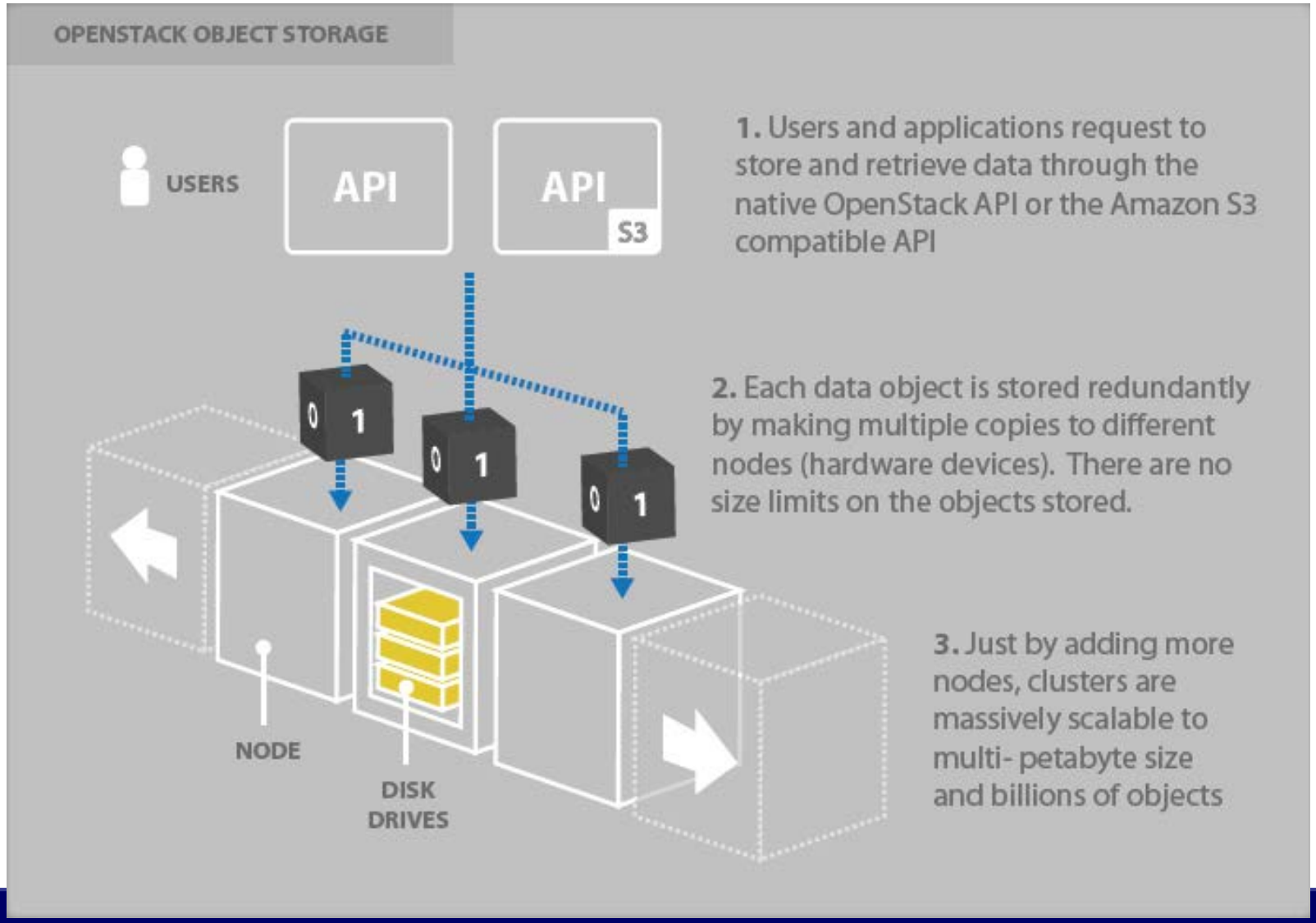


SWIFT

Grid y Computación de Altas Prestaciones
GRyCAP

- Proporciona un almacenamiento de objetos binarios (a la BLOB).
- Varios Componentes
  - swift-proxy-server: acepta peticiones a través del API de objetos OpenStack.
  - Servidores que mapean el almacenamiento físico de los contenedores con objetos del servicio.
  - Servidores de objetos que gestionan los objetos en los nodos de almacenamiento.
  - Procesos periódicos como replicación, auditoria, actualización...

Openstack object storage (II)



Glance (VMIs)

Grid y Computación de Altas Prestaciones GRyCAP

- Catálogo para el registro de imágenes de máquinas virtuales.
- Almacena las imágenes virtuales en elementos consumibles por los servicios de la infraestructura (disco, swift, Ceph, etc.)
- Subcomponentes:
  - glance-api: image discovery, retrieval, creation and storage.
  - glance-registry: storage and retrieval of metadata.
  - glance-cache, glance-reaper, glance-replication.

Glance (VMIs)

Grid y Computación de Altas Prestaciones
GRyCAP

- Coma
  - ima
  - ima
  - ima
  - ima
  - ima
  - ima

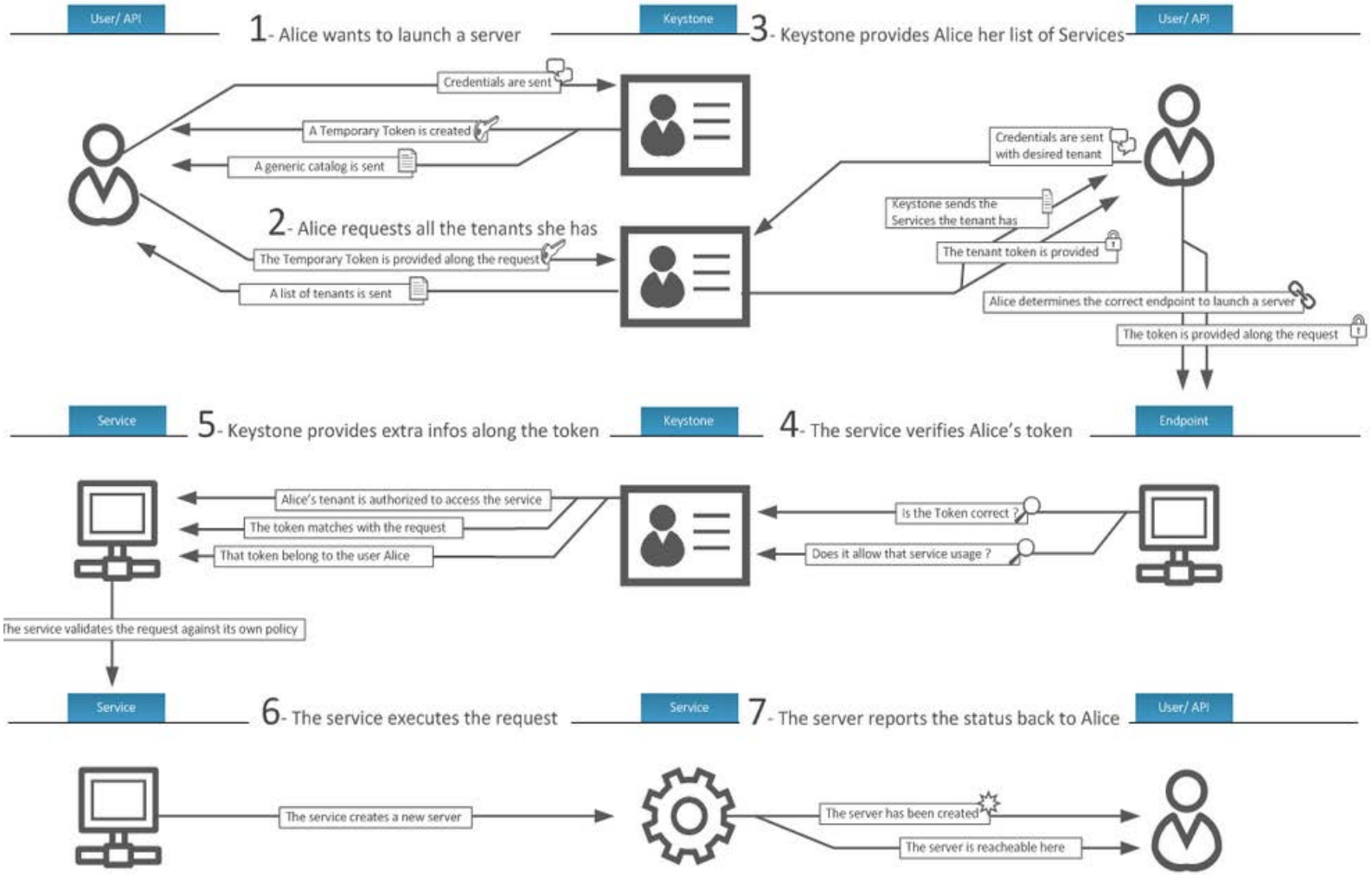
```
dfubuntu@controller:~$ glance image-list
+--------------------------------------+---------------------+-------------+---------------
| ID                                   | Name                | Disk Format | Container Form
+--------------------------------------+---------------------+-------------+---------------
| a1521352-82ba-4ed2-aa70-86ed0459cc05 | cirros-0.3.2-x86_64 | qcow2       | bare
+--------------------------------------+---------------------+-------------+---------------
ubuntu@controller:~$ glance image-show cirros-0.3.2-x86_64
+------------------+--------------------------------------+
| Property         | Value                                |
+------------------+--------------------------------------+
| checksum         | 64d7c1cd2b6f60c92c14662941cb7913     |
| container_format | bare                                 |
| created_at       | 2014-09-27T08:38:39                  |
| deleted          | False                                |
| disk_format      | qcow2                                |
| id               | a1521352-82ba-4ed2-aa70-86ed0459cc05 |
| is_public        | True                                 |
| min_disk         | 0                                    |
| min_ram          | 0                                    |
| name             | cirros-0.3.2-x86_64                  |
| owner            | a413b25d649e4464a09d287cdaad1853     |
| protected        | False                                |
| size             | 13167616                             |
| status           | active                               |
| updated_at       | 2014-09-27T08:38:39                  |
+------------------+--------------------------------------+
ubuntu@controller:~$
```

Autenticación / Autorización

- La autenticación / autorización está basada en los siguientes conceptos:
  - “User”, la representación de un usuario de OpenStack.
  - “Tenant”, grupo o Proyecto que aisla usuarios y recursos.
  - “Domain”, dominio administrativo.
  - “Role”, un conjunto de derechos y privilegios que se aplican a un usuario.
  - “Service”, un servicio OpenStack (nova, glance, etc.)
  - “Endpoint”, una URL desde la que un usuario puede accede a un servicio.
  - “Token”, un fragmento de texto (arbitrario o no) usado para acceder a los recursos. El token contiene los roles de los usuarios.

Autenticación / Autorización

- Un usuario es miembro de 1 ó más “tenants”.
- Un “tenant” es parte de 1 ó más dominios.
- Un usuario puede tener roles específicos en un “tenant” o globalmente en un dominio keystone.
- Un token puede ser asociado con un tenant o no:
  - Los tokens sin contexto no están asociados a ningún “tenant” y se utilizan para descubrimiento (endpoints y tenants disponibles) y sólo los acepta keystone.
  - Los tokens con context se asocial con un “tenant” y son los que se utilizan para interactuar con cualquier otro componente.
- Un token puede estar firmado (PKI) o no (UUID).


User/ API
1- Alice wants to launch a server
Keystone
3- Keystone provides Alice her list of Services
User/ API
Credentials are sent
A Temporary Token is created
A generic catalog is sent
2- Alice requests all the tenants she has
The Temporary Token is provided along the request
A list of tenants is sent
Credentials are sent with desired tenant
Keystone sends the Services the tenant has
The tenant token is provided
Alice determines the correct endpoint to launch a server
The token is provided along the request
Service
5- Keystone provides extra infos along the token
Keystone
4- The service verifies Alice's token
Endpoint
Alice's tenant is authorized to access the service
The token matches with the request
That token belong to the user Alice
Is the Token correct ?
Does it allow that service usage ?
The service validates the request against its own policy
Service
6- The service executes the request
Service
7- The server reports the status back to Alice
User/ API
The service creates a new server
The server has been created
The server is reacheable here

Keystone (gestión de

Grid y Computación de Altas Prestaciones

```
ubuntu@controller:~$ keystone endpoint-list
+----------------------------------+-----------+-----------------------------------------+----------
--------------+-----------------------------------------+----------------------------------+
|                id                |   region  |                publicurl                |
|                 adminurl                 |           service_id           |
+----------------------------------+-----------+-----------------------------------------+----------
--------------+-----------------------------------------+----------------------------------+
| 4c85b0f0c75f48f2b252714251e09329 | regionOne | http://controller:8774/v2/%(tenant_id)s |
http://controller:8774/v2/%(tenant_id)s | http://controller:8774/v2/%(tenant_id)s | f12262b9241b4ed39
|
| 88117b17e5e443c3bc9b8f55d26c592a | regionOne |       http://controller:5000/v2.0       |
http://controller:5000/v2.0       |       http://controller:35357/v2.0      | dde87cee8fa4441584b897
| f326def7842e475b9001e3724aedd2f5 | regionOne |         http://controller:9292          |
http://controller:9292          |          http://controller:9292          | 5936cab450eb45e9b6edb8fd6
+----------------------------------+-----------+-----------------------------------------+----------
--------------+-----------------------------------------+----------------------------------+
ubuntu@controller:~$ keystone service-list
+----------------------------------+----------+----------+--------------------------+
|                id                |   name   |   type   |       description        |
+----------------------------------+----------+----------+--------------------------+
| 5936cab450eb45e9b6edb8fd6d929000 |  glance  |  image   | OpenStack Image Service |
| dde87cee8fa4441584b8972f9c41e2f1 | keystone | identity |    OpenStack Identity    |
| f12262b9241b4ed391876bcdcdeda733 |   nova   | compute  |    OpenStack Compute     |
+----------------------------------+----------+----------+--------------------------+
```

Openstack neutron

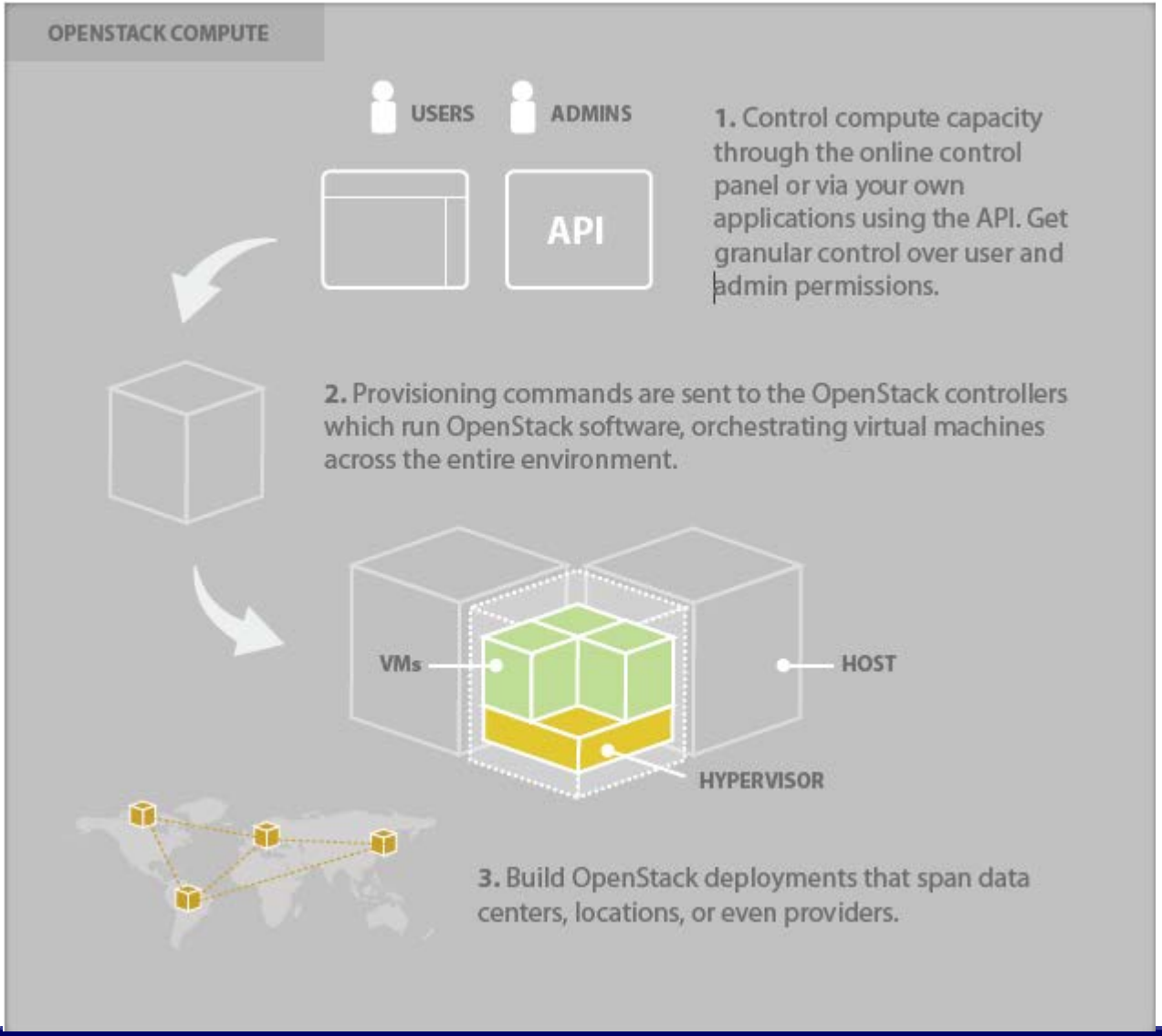
- Sistema para la gestión de redes y direcciones IP.
- Proporciona capacidades de red a las instancias.
  - Soporte para redes simples y VLANs, para la separación de servidores y tráfico entre máquinas virtuales.
- Los usuarios pueden crear sus propias redes, así como conectar los dispositivos de sus VMs a diferentes redes.
- Soporta *Floating IPs*
  - Asignar y reasignar direcciones IP a las VM.
- Permite la creación de redes VPN (Virtual Private Network)

NOVA

Grid y Computación de Altas Prestaciones GRyCAP

- Proporciona servidores virtuales bajo demanda
  - Soporta varios tipos de hipervisores: Xen, KVM, VMWare ESX, QEMU, UML, Microsoft Hyper-V y Citrix XenServer, Linux containers (LXC).
- Se compone de varios sub-componentes
  - nova-api: OpenStack Compute API, EC2, OCCI (no nativo).
  - nova-compute: lanza las instancias.
  - nova-scheduler: planifica las solicitudes.
  - nova-consoleauth: proporciona autorización para peticiones VNC.
  - nova-xvpcproxy: VNC proxy
  - nova-conductor: Realiza consultas a la BBDD mediante RPC.
  - nova-cert: Maneja certificados.
  - database: almacena el estado de la nube (Ips configuradas, instancias en marcha, tipos disponibles, etc.)
  - message queue: hub para pasar mensajes y llamadas RPC calls.

Openstack nova (II)



- Despliegue de Máquinas Virtuales sobre recursos físicos.
- Soporta el API de Amazon EC2.
- Gestión de usuarios y permisos.

Openstack object storage (I)

- Permite la creación de almacenamiento escalable y redundante de objetos usando clusters de servidores (al estilo de Amazon S3)
- Almacenamiento de objetos más bien estáticos que pueden ser almacenados, obtenidos y actualizados:
  - Ejemplos: Imágenes de Máquinas Virtuales, Fotos, archivos de copia de seguridad, etc.
- No es un sistema de archivos ni se orienta a tiempo real.
- Gestión automática de la replicación
  - Los objetos almacenados se escriben en varios dispositivos hardware. Replicación en otros nodos en caso de fallo de nodo.

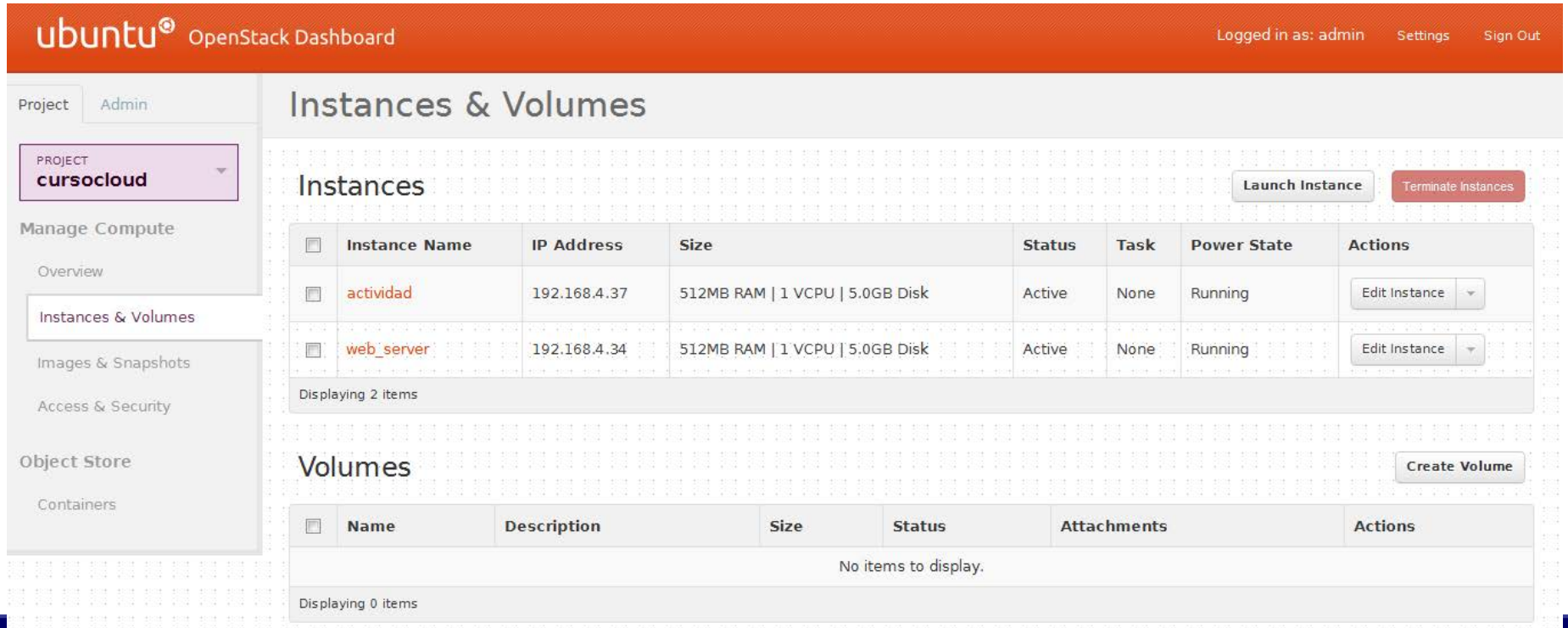
Otros servicios

Grid y Computación de Altas Prestaciones GRyCAP

- Horizon dashboard
  - Provides access and management trough a web interface.
  - Developed in Django.
  - Extensible and modular.
  - Uses the public APIs.
- Cinder
  - Proporciona volúmenes (almacenamiento de bloques) a las instancias.
  - Varios componentes:
    - cinder-api
    - cinder-scheduler planifica las peticiones en un volume.
    - cinder-volume gestiona el dispositivo físico. Varios ackends: iSCSI + LVM, NetApp, etc.

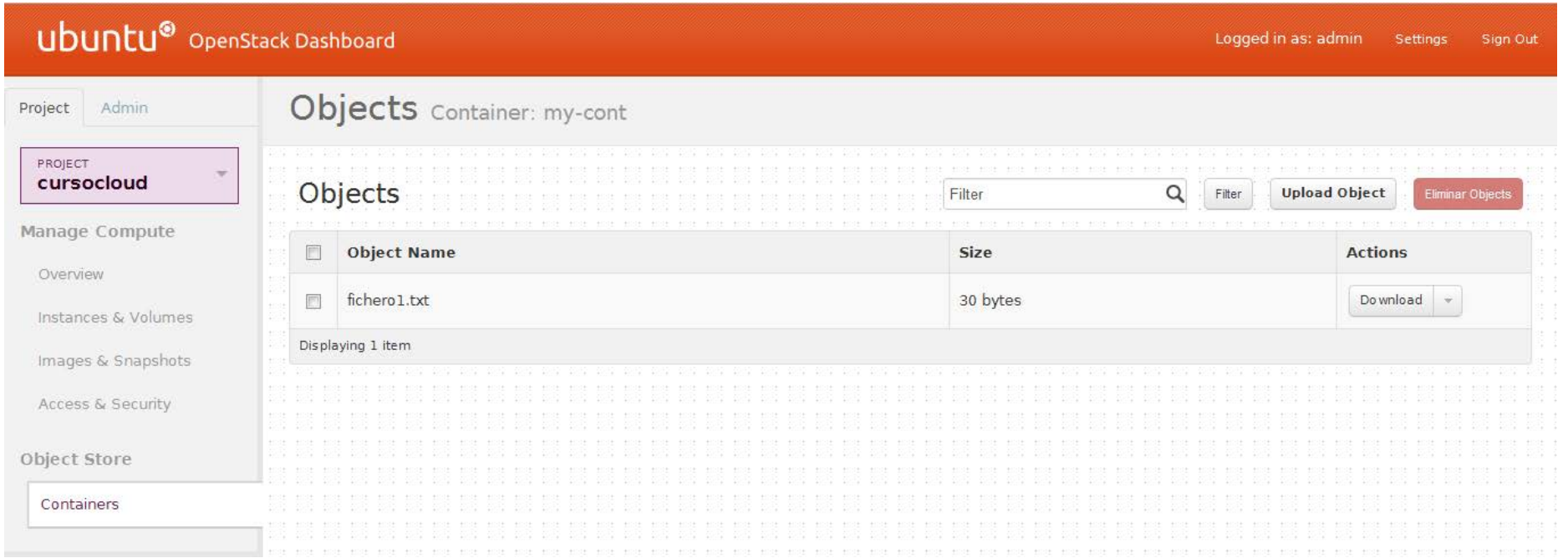
Openstack dashboard (I)

- El panel de control accesible vía Web permite gestionar las instancias (VMs) y volúmenes, las imágenes (VMIs), las configuraciones de red, etc.



Openstack dashboard (ii)

- También permite gestionar el almacén de objetos y ver los ficheros (objetos) almacenados en los contenedores (buckets), tal y como se hace en Amazon S3.



Openstack vs opennebula

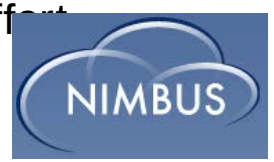
Grid y Computación de Altas Prestaciones GRyCAP

- Ambos permiten el despliegue y gestión de infraestructuras de máquinas virtuales.
- Algunos apuntes:
  - OpenNebula se centra en ofrecer una solución para la virtualización del centro de cálculo, permitiendo a los usuarios construir Clouds privados e híbridos.
  - OpenNebula consta de un software integrado, mientras que OpenStack consta de una serie de servicios que, juntos, permite la creación de Clouds.
    - Aunque las distribuciones de Linux (como Ubuntu) facilitan su instalación.
  - OpenStack se basa en el API de AWS e incluye un grupo de trabajo de estandarización, mientras que OpenNebula se centra en implementar los estándares de facto y las especificaciones de OGF, DMTF y SNIA.
  - OpenStack ofrece almacenamiento de objetos (a la S3), mientras que OpenNebula no.

Otras herramientas para el despliegue de IaaS clouds (I)

Grid y Computación de Altas Prestaciones GRyCAP

- Nimbus
  - **Nimbus Platform** is an integrated set of tools that deliver the power and versatility of infrastructure clouds to scientific users. Nimbus Platform allows you to combine Nimbus, OpenStack, Amazon, and other clouds
  - **Nimbus Infrastructure** is an open source EC2/S3-compatible Infrastructure-as-a-Service implementation specifically targeting features of interest to the scientific community such as support for proxy credentials, batch schedulers, best-effort allocations and others.



- Eucalyptus
  - Eucalyptus enables the creation of on-premise Infrastructure as a Service clouds, with no requirements for retooling the organization's existing IT infrastructure or for introducing any specialized hardware. The Eucalyptus IaaS platform maintains high fidelity with the Amazon Web Services (AWS) API, allowing support for both on-premise and hybrid IaaS clouds.
  - This compatibility allows any Eucalyptus cloud to be turned into a hybrid IaaS deployment, capable of moving workloads between AWS and on-premise data centers. Eucalyptus is compatible with a wealth of tools and applications that also adhere to the de facto AWS API standards.

EUCALYPTUS

Otras herramientas para el despliegue de IaaS clouds (II)

Grid y Computación de Altas Prestaciones GRyCAP

- Abiquo
  - Abiquo is the most complete and advanced solution available on the market today. Abiquo provides class-leading features like virtual to virtual conversion through a platform that is easy to implement and operate, liberating your IT organization from the drudgery of managing thousands of virtual machines, without relinquishing control of the physical infrastructure.
- Enomaly
  - The Elastic Computing Platform (ECP) is the answer for service providers that want to leverage the power, flexibility, and compelling economics of cloud computing.
  - Enomaly ECP Service Provider Edition is a complete "cloud in a box" solution, enabling telcos and hosting providers to deliver revenue-generating Infrastructure-on-demand (IaaS) cloud computing services to their customers, quickly and easily, with a compelling and highly differentiated feature set.

¿Cuál es la mejor herramienta de despliegue de iaas cloud?

Grid y Computación de Altas Prestaciones
GRyCAP

- ¿Qué factores considerar para decidir qué herramienta de despliegue de IaaS Clouds utilizar en nuestra infraestructura?
  - Código Abierto vs Soporte Empresarial
  - Casos de Éxito (Empresariales, Educativos, Científicos)
  - Comunidad Activa de Usuarios y Desarrolladores
    - Foros, Lista de Distribución, etc.
  - Integración con Distribuciones de GNU/Linux para Facilitar el Despliegue
  - ...

Conclusiones

Grid y Computación de Altas Prestaciones GRyCAP

- El despliegue de Clouds para IaaS requiere herramientas que gestionen el ciclo de vida de las máquinas virtuales sobre una infraestructura física en las que media un hipervisor, así como la gestión de red y otros detalles (usuarios, grupos, proyectos, imágenes, etc.).
- Todas las herramientas presentadas ofrecen una funcionalidad similar (en líneas generales).
- Solo el futuro decidirá si hay suficiente cuota de mercado para todas las herramientas.

referencias

Grid y Computación de Altas Prestaciones
GRyCAP

1. OpenStack Project. http://www.openstack.org/
2. OpenNebula Project. http://www.opennebula.org/
3. Eucalyptus (OpenSource). http://open.eucalyptus.com/
4. CloudStack Project. http://www.cloudstack.org/
5. Above the Clouds. A Berkeley View of Cloud Computing. Disponible en http://berkeleyclouds.blogspot.com.es/
6. The NIST Definition of Cloud Computing (SP 800-145). Disponible en http://csrc.nist.gov/publications/PubsSPs.html#800-145
7. OpenNebula and OpenStack Featured in European Report Advances in Clouds. http://blog.opennebula.org/?p=2824
8. http://www.readwriteweb.com/cloud/2012/04/who-wrote-openstack-essex-a-de.php
9. http://www.wired.com/wiredenterprise/2012/04/openstack/3/
10. http://nebula.nasa.gov/blog/2012/05/29/nasa-and-openstack-2012/
11. http://www.crunchbase.com/company/eucalyptus-systems-inc

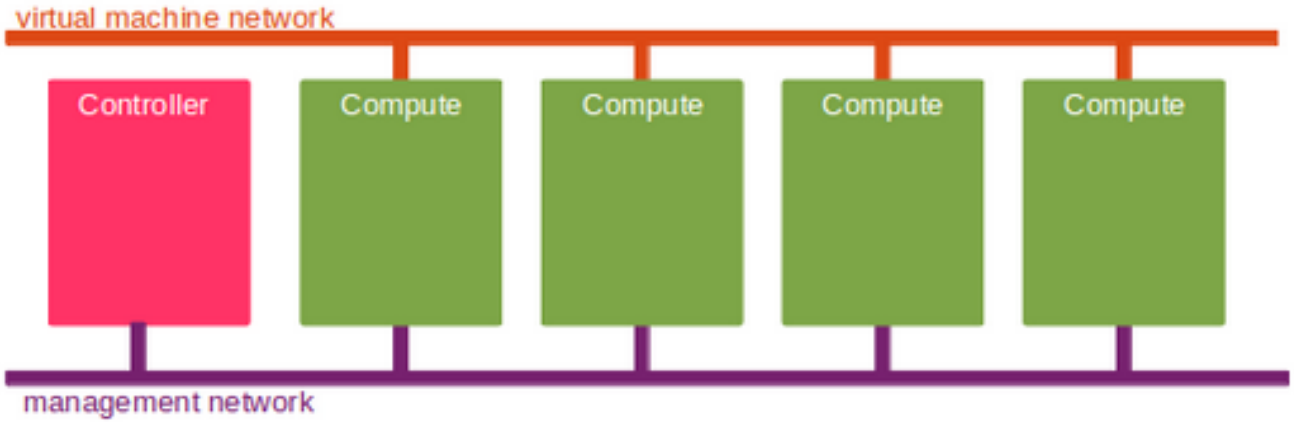
Grid y Computación de Altas Prestaciones
GRyCAP

SESIÓN PRÁCTICA - INSTALACIÓN

Instalación de un servidor OpenStack

Grid y Computación de Altas Prestaciones
GRyCAP

- Vamos a describir la instalación de una infraestructura mínima de dos nodos (controller + compute01).
- El nodo controller tendrá los servicios glance, keystone y de gestión de nova (scheduler, cert, api, metadata, conductor).
- El nodo compute01 tendrá el servicio nova-compute y nova-networks.
- A continuación, los alumnos configurarán un cliente nova para interactuar con la instalación.



Prerrequisitos

Grid y Computación de Altas Prestaciones GRyCAP

- Partimos de una Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-69-virtual x86_64)
- Desde el usuario con privilegios

```
$ sudo su –
```

- Cambiar `localhost` por `controller` en `/etc/hosts`
- Actualización de paquetes

```
$ apt-get install -y python-software-properties
$ add-apt-repository cloud-archive:icehouse
$ apt-get update
$ apt-get dist-upgrade
```

- RabbitMQ

```
$ apt-get install -y rabbitmq-server
$ rabbitmqctl change_password guest RABBIT_PASS
```

Prerrequisitos (2)

Grid y Computación de Altas Prestaciones GRyCAP

- MySQL

```
$ apt-get install -y python-mysqldb mysql-server
$ vi /etc/mysql/my.cnf
(...)
[mysqld]
...
default-storage-engine = innodb
innodb_file_per_table
collation-server = utf8_general_ci
init-connect = 'SET NAMES utf8'
character-set-server = utf8
bind-address=0.0.0.0
(...)

$ /etc/init.d/mysql restart
```

Creación de la base de datos y de los usuarios de los servicios

Grid y Computación de Altas Prestaciones GRyCAP

‘controller’ es el nombre de la máquina, sin la dirección completa (probar si responde al ping)

```
$ mysql -u root -p
mysql> CREATE DATABASE keystone;
mysql> GRANT ALL PRIVILEGES ON keystone.* TO
  'keystone'@'controller' IDENTIFIED BY 'KEYSTONE_DBPASS';
mysql> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%'
  IDENTIFIED BY 'KEYSTONE_DBPASS';

mysql> CREATE DATABASE nova;
mysql> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'controller'
  IDENTIFIED BY 'NOVA_DBPASS';
mysql> GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY
  'NOVA_DBPASS';

mysql> CREATE DATABASE glance;
mysql> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'controller‘
  IDENTIFIED BY 'GLANCE_DBPASS';
mysql> GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%'
  IDENTIFIED BY 'GLANCE_DBPASS';
```

Instalación de keystone (1)

Grid y Computación de Altas Prestaciones GRyCAP

```
$ apt-get install -y keystone
$ vi /etc/keystone/keystone.conf
(...)
[database]
connection =
mysql://keystone:KEYSTONE_DBPASS@controller/keystone
(...)
[DEFAULT]
(...)
admin_token = ADMIN_TOKEN
(...)
$ rm /var/lib/keystone/keystone.db
$ /bin/sh -c "keystone-manage db_sync" keystone
$ service keystone restart
```

Instalación de keystone (2)

- Nos identificamos por un token

```
$ export OS_SERVICE_TOKEN=ADMIN_TOKEN
$ export OS_SERVICE_ENDPOINT=http://controller:35357/v2.0
```

- Creamos usuarios, grupos de seguridad (tenants) y los roles

```
$ keystone user-create --name=admin --pass=ADMIN_PASS --
  email=ADMIN_EMAIL
$ keystone role-create --name=admin
$ keystone tenant-create --name=admin --description="Admin
  Tenant"
$ keystone user-role-add --user=admin --tenant=admin --
  role=admin
$ keystone user-role-add --user=admin --role=_member_ --
  tenant=admin
$ keystone tenant-create --name=service --description="Service
  Tenant"
$ keystone service-create --name=keystone --type=identity --
  description="OpenStack Identity"
```

Instalación de keystone (2)

- Registramos el propio end-point del keystone

```
$ keystone endpoint-create \
--service-id=$(keystone service-list | awk '/ identity / {print $2}') \
--publicurl=http://controller:5000/v2.0 \
--internalurl=http://controller:5000/v2.0 \
--adminurl=http://controller:35357/v2.0
```

- Comprobamos la instalación

```
$ unset OS_SERVICE_TOKEN OS_SERVICE_ENDPOINT
$ keystone --os-username=admin --os-password=ADMIN_PASS \
--os-auth-url=http://controller:35357/v2.0 token-get
```

Instalación de glance

- Instalamos los paquetes del glance

```
$ apt-get install -y glance python-glanceclient
$ vi /etc/glance/glance-{api,registry}.conf
[keystone_authtoken]
auth_uri = http://controller:5000
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = glance
admin_password = GLANCE_PASS
[paste_deploy]
flavor = keystone
[database]
connection = mysql://glance:GLANCE_DBPASS@localhost/glance
$ /bin/sh -c "glance-manage db_sync" glance
```

Instalación de glance

- Creamos la entrada en el keystone

```
$ keystone user-create --name=glance --pass=GLANCE_PASS --
  email=glance@example.com
$ keystone user-role-add --user=glance --tenant=service --role=admin
$ keystone service-create --name=glance --type=image --
  description="OpenStack Image Service"
$ keystone endpoint-create \
--service-id=$(keystone service-list | awk '/ image / {print $2}') \
--publicurl=http://controller:9292 \
--internalurl=http://controller:9292 \
--adminurl=http://controller:9292

$ service glance-registry restart
$ service glance-api restart
```

Instalación de glance

- Probamos registrando una imagen.

```
$ export OS_USERNAME=admin
$ export OS_TENANT_NAME=admin
$ export OS_PASSWORD=ADMIN_PASS
$ export OS_AUTH_URL="http://controller:5000/v2.0"

$ mkdir /tmp/images
$ cd /tmp/images/
$ wget http://download.cirros-cloud.net/0.3.3/cirros-0.3.3-x86_64-
  disk.img

$ glance image-create --name "cirros-0.3.3-x86_64" \
--disk-format qcow2 --container-format bare --is-public True \
--progress < cirros-0.3.3-x86_64-disk.img

$ glance image-list
$ cd -
```

Instalación de nova

- Instalamos los paquetes del nova

```
$ apt-get install -y nova-api nova-cert nova-conductor nova-
  consoleauth nova-novncproxy nova-scheduler python-
  novaclient
$ vi /etc/nova/nova.conf
[DEFAULT]
(...)
rpc_backend = rabbit
rabbit_host = controller
rabbit_password = RABBIT_PASS
(...)
[database]
connection = mysql://nova:NOVA_DBPASS@controller/nova
(...)
$ rm /var/lib/nova/nova.db
$ /bin/sh -c "nova-manage db sync" nova
```

Instalación de nova

- Creamos usuarios y terminamos de configurar

```
$ export OS_USERNAME=admin; export OS_TENANT_NAME=admin
$ export OS_PASSWORD=ADMIN_PASS
$ export OS_AUTH_URL="http://controller:5000/v2.0"
$ keystone user-create --name=nova --pass=NOVA_PASS --
  email=nova@example.com
$ keystone user-role-add --user=nova --tenant=service --role=admin
$ vi /etc/nova/nova.conf
[DEFAULT]
auth_strategy = keystone
[keystone_authtoken]
auth_uri = http://controller:5000
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = NOVA_PASS
```

Instalación de nova

- Configuración del nova-api

```
$ vi /etc/nova/api-paste.ini
[filter:authtoken]
paste.filter_factory =
keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = NOVA_PASS
auth_version = v2.0
```

Instalación de nova

- Registro de los servicios

```
$ keystone service-create --name=nova --type=compute --
  description="OpenStack Compute"
$ keystone endpoint-create --service-id=$(keystone service-
  list | awk '/ compute / {print $2}') \
  --publicurl=http://controller:8774/v2/\%\(tenant_id\)s \
  --internalurl=http://controller:8774/v2/\%\(tenant_id\)s \
  --adminurl=http://controller:8774/v2/\%\(tenant_id\)s
```

- Restart de los servicios

```
$ service nova-api restart
$ service nova-cert restart
$ service nova-consoleauth restart
$ service nova-scheduler restart
$ service nova-conductor restart
$ service nova-novncproxy restart
```

Instalación de nova

- Comprobación

```
$ nova service-list
$ nova endpoints
$ nova credentials
$ nova list
$ nova image-list
```

Grid y Computación de Altas Prestaciones
GRyCAP

INSTALACIÓN DE UN COMPUTE

Prerrequisitos

Grid y Computación de Altas Prestaciones GRyCAP

- Partimos de una Ubuntu 12.04.5 LTS (GNU/Linux 3.2.0-69-virtual x86_64)
  - Necesitaríamos una máquina con dos interfaces de red
  - **Necesario una máquina con 2 CPUS**
- Desde el usuario con privilegios

```
$ sudo su –
```

- Configuración de Red

```
$ vi /etc/network/interfaces.d/eth1.cfg
# The primary network interface
auto eth0:0
iface eth0:0 inet static
address 192.168.38.253
netmask 255.255.255.0
metric 1
```

- Actualización de paquetes

```
$ add-apt-repository cloud-archive:icehouse
$ apt-get update
$ apt-get -y install ntp python-mysqldb python-software-properties
$ apt-get -y dist-upgrade
$ apt-get -y install nova-compute-qemu python-guestfs
```

Configuración del NOVA

```
$ vi  /etc/nova/nova.conf
rpc_backend =
nova.rpc.impl_kombu
rabbit_host = controller
rabbit_password = RABBIT_PASS


my_ip=$MY_IP
vnc_enabled=True
vncserver_listen=0.0.0.0
vncserver_proxyclient_address
=$MY_IP
novncproxy_base_url=http://
controller:6080/vnc_auto.html

auth_strategy=keystone
glance_host=controller
```

```
[database]
connection =
mysql://nova:NOVA_DBPASS@cont
roller/nova

[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = NOVA_PASS
auth_version = v2.0
```

Configuración del NOVA

```
$ vi /etc/nova/api-paste.ini
[filter:authtoken]
paste.filter_factory =
keystoneclient.middleware.auth_token:filter_factory
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = nova
admin_password = NOVA_PASS
auth_version = v2.0
```

Instalación del gestor de red plano

Grid y Computación de Altas Prestaciones GRyCAP

```
$ apt-get install -y nova-network nova-api-metadata
$ apt-get -y install dnsmasq
$ vi  /etc/nova/nova.conf
network_manager=nova.network.manager.FlatDHCPManager
firewall_driver=nova.virt.libvirt.firewall.IptablesFirewallDriver
network_size=254
allow_same_net_traffic=True
multi_host=True
send_arp_for_ha=True
share_dhcp_address=False
force_dhcp_release=True
flat_network_bridge=br100
flat_interface=eth1  -> flat_interface=eth0:0
public_interface=eth0

$ service nova-network restart
$ service nova-compute restart
$ service nova-api-metadata restart
```

Creación de una red desde el controller

Grid y Computación de Altas Prestaciones
GRyCAP

```
$ nova network-create vmnet --fixed-range-v4=192.168.38.0/24 --bridge-interface=br100 --multi-host=T
$ nova secgroup-add-rule default tcp 22 22 0.0.0.0/0
$ nova secgroup-add-rule default icmp -1 -1 0.0.0.0/0
```

- Comprobamos desde el controller

```
azureuser@OStack-main:/var/log/nova$ nova service-list
+------------------+---------------+----------+---------+-------+------------------------
---+-----------------+
| Binary           | Host          | Zone     | Status  | State | Updated_at
| Disabled Reason |
+------------------+---------------+----------+---------+-------+------------------------
---+-----------------+
| nova-cert        | OStack-main   | internal | enabled | up    | 2014-11-
12T07:56:03.000000 | -               |
| nova-consoleauth | OStack-main   | internal | enabled | up    | 2014-11-
12T07:56:11.000000 | -               |
| nova-scheduler   | OStack-main   | internal | enabled | up    | 2014-11-
12T07:56:09.000000 | -               |
| nova-conductor   | OStack-main   | internal | enabled | up    | 2014-11-
12T07:56:06.000000 | -               |
| nova-compute     | ostack-compute | nova     | enabled | up    | 2014-11-
12T07:55:57.000000 | -               |
| nova-network     | ostack-compute | internal | enabled | up    | -
| -               |
+------------------+---------------+----------+---------+-------+------------------------
---+-----------------+
```

Grid y Computación de Altas Prestaciones
GRyCAP

EJECUCIÓN DE UNA PRUEBA

Puesta en marcha de una máquina

- Desde la máquina controller
- Creación de un par de claves

```
$ nova keypair-add tu_nombre_aquí > privkey.pem
$ nova keypair-list
$ nova image-list
$ nova flavor-list
$ nova boot --flavor 1 --key-name tu_nombre_aquí --image imageid el_nombre_de_tu_instancia
$ nova list
```

- Accedemos con nuestra clave privada

```
$ ssh -i clave_privada cirros@IP_privada
```

Grid y Computación de Altas Prestaciones
GRyCAP

INSTALACIÓN DEL CLIENTE DE NOVA

Instalación del cliente

- La práctica se puede realizar en cualquier máquina Linux que tenga instalado virtualenv
  - De esa forma no es necesario tener permisos de root.
- Preparación

```
$ mkdir /tmp/tutorial
$ virtualenv /tmp/tutorial/VENV
$ source /tmp/tutorial/VENV/bin/activate
```

- Instalación

```
(VENV) $ cd /tmp/tutorial
(VENV) $ git clone https://github.com/openstack/python-novaclient
(VENV) $ pip install python-novaclient
```

- Fichero de configuración

```
(VENV) $ cat > novarc << EOF
#!/bin/bash
export OS_AUTH_URL=http://controller-DNS:5000/v2.0
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=ADMIN_PASS
EOF
```

- Comprobación

```
(VENV) $ source novarc
(VENV) $ nova credentials
(VENV) $ nova endpoints
(VENV) $ nova list
```

Puesta en marcha de una máquina

- Creación de un par de claves

```
(VENV) $ nova keypair-add tu_nombre_aquí > privkey.pem
(VENV) $ nova keypair-list
(VENV) $ nova image-list
(VENV) $ nova flavor-list
(VENV) $ nova boot --flavor m1.small --key-name
tu_nombre_aquí --image 07c98683-8ccd-4001-80fd-
3a8b83596a26 el_nombre_de_tu_instancia
(VENV) $ nova list
```

- Las máquinas se ponen en marcha con una IP privada
  - Es posible asignar una IP pública a una máquina en marcha

```
(VENV) $ nova floating-ip-create
(VENV) $ nova floating-ip-list
(VENV) $ nova add-floating-ip <server> <ip>
```

- Accedemos con la clave privada con nuestro nombre
- Desafortunadamente, por motivos de limitación de puertos e IPS, debemos hacerlo desde el controller.

```
$ ssh -i clave_privada cirros@IP_privada

$ Y si falla: cubswin:)
```