

Despliegue de una Infraestructura Cloud Privada de Código Abierto

Maria A. Murazzo

Departamento e Instituto de Informática
FCEfy N – UNSJ
Complejo Islas Malvinas (CUIM), San Juan, Argentina

Fernando G. Tinetti

III-LIDI, Fac. de Informática, UNLP
Comisión de Inv. Científicas Prov. de Bs. As.
50 y 120, 2do. Piso, 1900, La Plata, Argentina

Nelson R. Rodriguez

Departamento e Instituto de Informática
FCEfy N – UNSJ
Complejo Islas Malvinas (CUIM), San Juan, Argentina

Resumen

Cloud computing es un tecnología que esta en constante evolución, prueba de esto es la apuesta que grandes empresas, tales como IBM, Google, Amazon, Microsoft, Red Hat, VMWare, entre otros están haciendo para ofrecer soluciones que saquen provecho de sus beneficios. Sin embargo, y a pesar de las ventajas que ofrecen las soluciones públicas, las organizaciones están demandando mayor grado de seguridad y privacidad en sus datos, por ello es que están tomando gran impulso las cloud privadas. De los productos que ofrece el mercado para implementar una infraestructura cloud privada, las de código abierto son las que mayor aceptación están teniendo debido a la robustez, el apoyo de empresas líderes en el área (como Canonical), la disponibilidad del código en repositorios públicos (como Git o GitHub) y la comunidad de usuarios que constantemente se encarga de mantener actualizaciones.

Este trabajo tiene como objetivo analizar las ventajas de instalar una infraestructura de cloud de código abierto. Actualmente existen varias alternativas y la selección dependerá de las necesidades y los objetivos que se persiguen en la instalación del cloud. En este caso se ha seleccionado como producto a OpenStack, el cual es una solución de infraestructura privada con capacidad de extenderse al ámbito híbrido con un gran crecimiento en lo que se refiere a empresas participantes en el proyecto, usuarios involucrados en el desarrollo y frecuencia de actualizaciones.

Palabras Claves: cloud computing, IaaS, código abierto, cloud privada, OpenStack

1. Introducción

Cloud Computing es un nuevo modelo de computación que permite la entrega de recursos de IT (Information Technology, Tecnología de la Información) bajo

demanda. En este ambiente, los recursos son aprovisionados dinámicamente y se ofrecen como servicio en Internet [1]. Cloud Computing es un paradigma que permite el cómputo a gran escala mediante la abstracción de la complejidad de la plataforma de desarrollo y la infraestructura sobre la que se monta, a través de la encapsulación del hardware, la compartición de recursos y la virtualización. Cloud computing se puede ver como la combinación de tecnologías ya existentes, muy conocidas y disponibles desde principios de la década del 90, tales como grid computing, utility computing y virtualización.

Atendiendo a la definición dada por el NIST (National Institute of Standards and Technology) [2], cloud computing es un modelo tecnológico que permite el acceso ubicuo, adaptado y bajo demanda en red a un conjunto compartido de recursos de computación configurables (por ejemplo: redes, servidores, equipos de almacenamiento, aplicaciones y servicios), que pueden ser rápidamente aprovisionados y liberados con un esfuerzo de gestión reducido o interacción mínima con el proveedor del servicio.

La característica básica de este modelo es que los recursos y servicios informáticos tales como infraestructura, plataforma y aplicaciones, son ofrecidos y consumidos como servicios a través de Internet sin que los usuarios tengan que tener ningún conocimiento de lo que sucede detrás. Cloud Computing es un esquema que a veces se expresa como EaaS (Everything as a Service, Cualquier cosa como Servicio) [3].

El NIST resalta cinco aspectos claves del cloud: autoservicio bajo demanda, acceso a la red ubicua, un conjunto de recursos independiente de la ubicación, rápida elasticidad y servicio a medida. Además, clasifica los modelos de cloud computing en dos grandes categorías: Modelos de servicio y Modelos de despliegue.

En el Modelo de Servicio, el IaaS (Infrastructure-as-a-Service, Infraestructura como Servicio) ofrece una plataforma de virtualización donde se le proporciona a los clientes recursos computacionales para que puedan desplegar y ejecutar software (por ejemplo: Amazon EC2, Rackspace, GoGrid). El PaaS (Platform-as-a-Service, Plataforma como Servicio), brinda a los usuarios la capacidad de desplegar sus aplicaciones en la infraestructura del cloud, utilizando diferentes lenguajes y herramientas de programación que el proveedor del servicio soporte (por ejemplo: Google Apps Engine, Windows Azure). Con respecto al SaaS (Software-as-a-Service, Software o Aplicaciones como Servicio), también llamado CloudApp, proporciona a los clientes el uso de aplicaciones del proveedor, que se ejecutan en su infraestructura. El usuario no se preocupa por detalles como dónde está instalado el software, qué tipo de sistema operativo utiliza o el lenguaje en el que cada aplicación está escrita (por ejemplo: GoogleApps, Salesforce, and Facebook).

Por otro lado, se encuentra el Modelo de Despliegue, que se refiere a la posición (localización) y administración (gestión) de la infraestructura de cloud y pueden ser públicos (manejadas por terceras partes), comunitarios (operada por varias organizaciones para uso compartido) o privados (propiedad de una organización). Como una cuarta opción aparece el modelo híbrido como la combinación de cualquiera de los tres modelos anteriores.

De las tres capas del modelo de servicio, en este trabajo se pondrá énfasis en el IaaS que básicamente maneja los recursos computacionales (cómputo, almacenamiento, conectividad, etc.). Pero para contar con una infraestructura escalable, es importante una cierta independencia del hardware físico, y es por ello que una parte importante del IaaS es la capa de virtualización que se construye con el objeto de abstraerse del hardware y crear una plataforma de recursos virtuales (tales como: procesador, memoria y otros dispositivos) proveyendo aislamiento y seguridad.

Gracias a la virtualización, la infraestructura cloud se convierte en un modelo de aprovisionamiento in-house, donde los hipervisores son los responsables de la administración eficiente de la infraestructura virtual, mediante políticas de administración de los recursos físicos. El uso de esta capa de virtualización provee aislamiento, encapsulación, independencia del hardware y soporte para pilas de protocolos heterogéneos. Esto en conjunto permite provisión de recursos bajo demanda, consolidación de servidores, balanceo de carga y redistribución dinámica de la infraestructura física.

En función de todo lo enunciado, el primer paso para instalar una arquitectura de cloud computing es montar la infraestructura sobre la cual se apoyará los demás de los componentes. El resto del trabajo se centra en describir las ventajas de seleccionar los productos de infraestructura de código abierto, entre los cuales el

que mayor aceptación ha tenido en los últimos años en la comunidad académica es OpenStack. Por último, se hace hincapié en la arquitectura OpenStack y en su última versión estable, poniendo especial énfasis en el proceso de instalación mediante DevStack.

2. Infraestructura de Cloud Computing

Construir una infraestructura de cloud involucra varias actividades, de las cuales la primera y más importante es la selección del producto que se utilizará. Esta tarea es complicada y no trivial pues existen muchas soluciones disponibles, cada una de las cuales poseen sus características propias y están orientadas a resolver problemas específicos.

Desde el punto de vista del modelo de servicio, el IaaS es la capa más baja del modelo y la encargada de brindar el soporte para que el usuario acceda a un autoservicio de recursos; ejemplos de estas plataformas son Amazon EC2, Google Compute Engine, IBM Smart Cloud, entre los más conocidos. Estas soluciones son propietarias, lo cual implica que hay que pagar por los recursos que se necesitan y se usan; además, esto es adecuado para las organizaciones que no tienen interés en los pormenores de la infraestructura, esto es, no les interesa tener el control de los recursos, sino solo necesitan tercerizar su centro de datos, mediante el uso de un cloud público.

En contraposición, se encuentran las soluciones de código abierto, las cuales están diseñadas para permitir a las organizaciones montar una infraestructura cloud privada y cuando sea necesario extenderla a un modelo híbrido. En los últimos años las soluciones de código abierto dedicadas a la administración y gestión de entornos de Cloud Computing han crecido exponencialmente, grandes compañías como Citrix, Red Hat, IBM, RackSpace, HP, Cisco, entre otras, están apostando fuertemente por este tipo de soluciones.

Un licenciamiento gratuito permite reducir los costos de la infraestructura. Pero este hecho debe estar balanceado con los costos de soporte y de desarrollo para personalizar el código de la solución. El licenciamiento en código abierto normalmente es menos complicado que en soluciones propietarias. Adicionalmente, estas soluciones permiten tener un control mayor sobre las fases de pruebas y evaluación de tecnologías Cloud.

En el mercado existen numerosos productos de este tipo, tales como: Eucalyptus, Nimbus, OpenStack, OpenNebula, CloudStack, entre los que poseen mayor aceptación. De todas ellas se han seleccionado aquellas que tienen mayor aceptación en la comunidad académica y mayor soporte de la comunidad de usuarios: OpenStack, CloudStack y OpenNebula.

- *OpenStack*: Es una herramienta de colaboración global, libre y de código abierto distribuido bajo los términos de la licencia Apache. Es una de las iniciativas de cloud que cuenta con más apoyos,

entre los que están Dell, Cisco y HP, así como RackSpace.

- *CloudStack*: software código abierto para crear, controlar y desplegar infraestructuras cloud. Originalmente fue desarrollado por cloud.com y posteriormente por Citrix, actualmente es un proyecto de la Apache Software Foundation (ASF). Implementa Amazon EC2, S3 APIs, vCloud API y su propia API CloudStack, escrita en Java. Soporta Xen, KVM, VMware y acceso a Amazon EC2s.
- *OpenNebula*: conjunto de herramientas código abierto que permiten manejar la infraestructura de virtualización de un centro de datos para construir cloud privadas, públicas e híbridas. Esta escrita en C++, Ruby y Shell. Soporta Xen Server, KVM, vSphere, y acceso a Amazon EC2s.

3. Infraestructura Seleccionada

En este trabajo se ha decidido trabajar con *OpenStack* (www.openstack.com), que es la base de *OpenCloud* (www.opencloud.com), una iniciativa que busca la interacción estandarizada de las plataformas cloud de diferentes empresas. Es un proyecto global para crear una plataforma de cómputo código abierto en el cloud, que cumpla con las necesidades de los proveedores de servicios en clouds públicos, privados e híbridos, independientemente de su tamaño, que sea fácil de implementar y escalable.

El proyecto fue fundado en octubre de 2010 por la empresa Rackspace Cloud y por la agencia espacial estadounidense, NASA. Actualmente es gestionado por la Fundación OpenStack y cuenta con el apoyo de más de 178 empresas, entre las que destacan Rackspace, Oracle, AMD, Cisco, Canonical, IBM, HP, Dell Red Hat, Suse Linux, VMware, Yahoo y KIO Networks en América Latina. Las características más importantes de OpenStack, se pueden resumir en:

- *Es escalable*: OpenStack es usado por compañías multinacionales cuyo volúmenes de datos se mide en petabytes, con arquitecturas distribuidas en 1 millón de nodos, con 60 millones de máquinas virtuales y billones de objetos almacenados.
- *Es compatible y flexible*: debido al soporte de múltiples hipervisores: ESX, Hyper-V, KVM, LXC, QEMU, UML, Xen y XenServer.
- *Es abierto*: todo su código se puede modificar y adaptarse a cualquier necesidad. Además, el proyecto tiene un proceso de validación para el desarrollo y adopción de nuevos estándares.

El principal objetivo del Proyecto OpenStack es proveer una infraestructura para construir una arquitectura escalable y de código abierto sin las restricciones

impuestas por los productos propietarios, mediante el uso de estándares abiertos capaces de lograr su extensión a un entorno híbrido.

OpenStack está escrito totalmente en Python y actualmente implementa dos APIs de control: la API EC2 y Rackspace. Es el tercer proyecto código abierto en importancia a nivel mundial; y se lo describe como un "sistema operativo de cloud", ya que se desempeña como el coordinador de los recursos que sustentan a los servicios [4].

Un cloud en OpenStack se compone de cuatro nodos con diferentes funciones dependiendo de los componentes de OpenStack que tenga instalados, como muestra la figura 1.

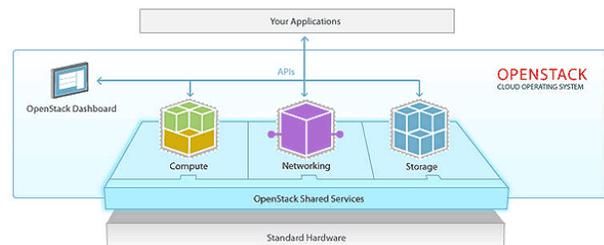


Figura 1: Arquitectura de OpenStack

Los componentes principales de OpenStack son tres y se detallan a continuación:

- *OpenStack Compute (NOVA)*: es el controlador de la estructura básica del Cloud. Es el encargado de iniciar las instancias (máquinas virtuales) de los usuarios y grupos. También es el servicio encargado de la gestión de la red virtual para cada instancia o para las múltiples instancias que formen parte de un proyecto (tenant).
- *OpenStack Object Storage (SWIFT)*: es el servicio encargado del almacenamiento masivo de objetos a través de un sistema escalable, redundante y tolerante a fallos. Las posibles aplicaciones de Object Storage son numerosas, como por ejemplo: almacenamiento simple de ficheros, copias de seguridad, almacenamiento de streamings de audio/vídeo, almacenamiento secundario/terciario, desarrollo de nuevas aplicaciones con almacenamiento integrado, etc.
- *OpenStack Networking (NEUTRON)*: es el componente encargado de la administración de red. Permite hacer uso de su API a través de plugins y agentes capaces de conectar/desconectar puertos, aprovisionar redes y subredes, etc, todo ello vinculado al proveedor que se vayan a usar en el cloud, como por ejemplo switches físicos o virtuales de Cisco, productos de NEC OpenFlow, Open vSwitch, Linux bridging, Ryu Network Operating System o VMware NSX.

Además de estos tres componentes, existen otros, de los cuales los más importantes son:

- *OpenStack Identity Service (KEYSTONE)*: es un servicio usado para la autenticación entre el resto de componentes. Este servicio utiliza un sistema de autenticación basado en tokens y se incorporó en la versión 2012.1 de OpenStack.
- *OpenStack Image Service (CINDER)*: es un servicio para la búsqueda y recuperación de imágenes de máquinas virtuales. Este servicio puede almacenar las imágenes directamente o utilizar mecanismos más avanzados como: usar Object Storage como servicio de almacenamiento, usar Amazon's Simple Storage Solution (S3) directamente, ó usar Object Storage como almacenamiento intermedio de S3.
- *OpenStack Dashboard (HORIZON)*: es un panel web para el manejo de instancias y volúmenes. Este servicio es realmente una aplicación web desarrollada en django que permite comunicarse con las diferentes APIs de OpenStack de una forma sencilla. OpenStack Dashboard es fundamental para usuarios noveles y en general para realizar acciones sencillas sobre las instancias.

Cada uno de estos nodos está formado por diferentes componentes que trabajan en conjunto. Esta integración es lograda a través de APIs que cada servicio ofrece y consume. Gracias a estas APIs, los servicios pueden comunicarse entre ellos y además se posibilita que un servicio sea reemplazado por otro de similares características siempre que se respete la forma de comunicación. Es decir, OpenStack es extensible y se ajusta a las necesidades de quien desee implementarlo [5].

La distribución de OpenStack que se usó fue la versión **Juno** [6], la cual es la penúltima versión estable de la plataforma liberada en octubre de 2014, la última versión estable es **Kilo** y se decidió no implementarla debido a que los componentes nuevos no son significativos para este trabajo.

Openstack está formado por distintos componentes que interactúan para proporcionar los servicios de computación, red y almacenamiento, ofreciendo una interfaz sencilla para que el usuario tenga acceso a sus recursos. En el caso de la versión Juno los componentes son 10, tal como se muestra en la figura 2.

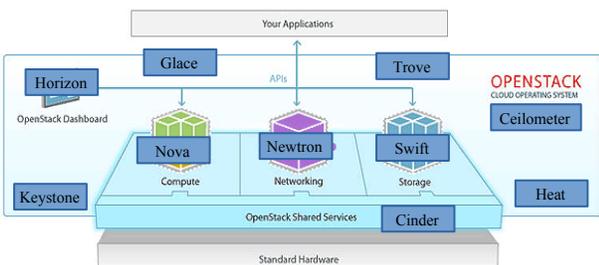


Figura 2: Arquitectura de OpenStack Juno

Además de los componentes mencionados anteriormente OpenStack Juno, incorpora de forma nativa a CEILOMETER (permite realizar análisis estadísticos y de escalabilidad), TROVE (brinda DBaaS, Data Base as a Service, para bases de datos SQL y noSQL), HEAT (ofrece servicios de orquestación de servicios) y SAHARA (este componente permite montar clusters Hadoop en OpenStack).

Desde una perspectiva global, OpenStack está diseñado para "entregar un sistema operativo para el despliegue de clouds masivamente escalables". Para poder lograrlo, cada uno de los servicios que conforman OpenStack están diseñados para trabajar conjuntamente y poder proporcionar una infraestructura como servicio completa. Esta integración se consigue a través de APIs que cada servicio ofrece, y que cada servicio puede consumir. Mientras que estas APIs permiten a cada uno de los servicios utilizar el resto, también permiten al desarrollador poder reemplazar cualquier servicio con otra implementación, siempre y cuando se respeten estas APIs. Dichas APIs también se encuentran disponibles para el usuario final del cloud. Esta arquitectura conceptual se puede ver en la figura 3.

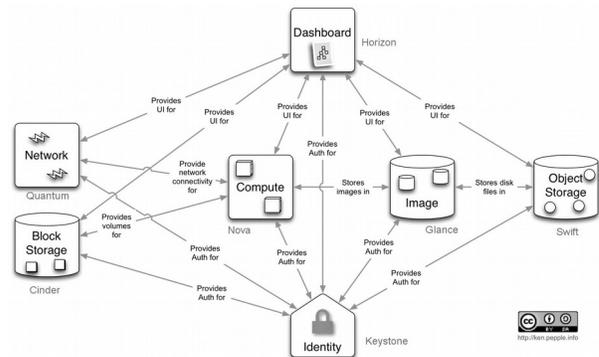


Figura 3: Esquema conceptual de OpenStack Juno

Openstack basa su filosofía de trabajo en una arquitectura que permite interconectar los diferentes componentes mediante APIs de bajo nivel. Esta arquitectura se puede montar de dos maneras diferentes:

- *Monolítica*: donde todos los componentes de la infraestructura están instalados en solo nodo, esto permite reducir la complejidad en las comunicaciones entre nodos.
- *Distribuida*: en este caso cada componente principal (NOVA, NEUTRON Y SWIFT) se encuentran instalados en nodos diferentes, lo cual obliga a tener una buena conexión entre ellos para no degradar la performance de la infraestructura.

En este trabajo se decidió implementar una arquitectura monolítica, debido a que el principal objetivo es la evaluación de la instalación y no la producción. Para ello se debe tener una instalación previa de Ubuntu 14.04 de 64 bits. El repositorio que se usó para acceder a

OpenStack Juno, se encuentra en GitHub, por lo tanto se debe tener instalado el paquete git. Una herramienta que ofrece OpenStack es DevStack [7] la cual ofrece un conjunto de script bash que permiten instalar OpenStack de forma automática. Para realizar la instalación se debe clonar el repositorio git y elegir la rama estable de la versión Juno, como se muestra a continuación.

```
git clone https://github.com/openstack-dev/devstack.git
cd devstack
git checkout stable/juno
```

Antes de ejecutar el script de instalación, se deben personalizar las distintas opciones de configuración, en el sitio de DevStack se provee una guía para realizar esta tarea. Los detalles más importantes a definir son:

- a) una lista de los componentes a instalar,
- b) las contraseñas y
- c) las direcciones en las cuales se podrá acceder a la infraestructura mediante Dashboard.

Una vez que se ha terminado con esta configuración se debe ejecutar el script de instalación. Una vez terminada la instalación, se puede acceder a Horizon mediante Dashboard, accediendo a la url, http://ip_fija. En este punto se despliega la pantalla de autenticación, como muestra la figura 4.



Figura 4: Pantalla de autenticación del Dashboard

Se han creado dos usuarios:

- El usuario administrador se llama admin, con contraseña devstack.
- Un usuario sin privilegios llamado demo, con contraseña devstack.

Una vez validado el usuario aparece la pantalla de administración, como muestra la figura 5.

El componente HORIZON (dashboard) proporciona al administrador y usuarios una interfaz gráfica para acceder, aprovisionar y automatizar los recursos del cloud. Las capacidades del dashboard son:

- Permite controlar a los administradores y usuarios su nivel de uso de recursos de cómputo, almacenamiento y red.
- Como administrador, ofrece una vista global

del tamaño y estado del cloud, permitiendo crear usuarios y proyectos, asignar usuarios a proyectos y configurar limites en los recursos de esos proyectos.

- Como usuarios, provee un ámbito que permite realizar el aprovisionamiento de los recursos propios dentro de los límites establecidos por el administrador.

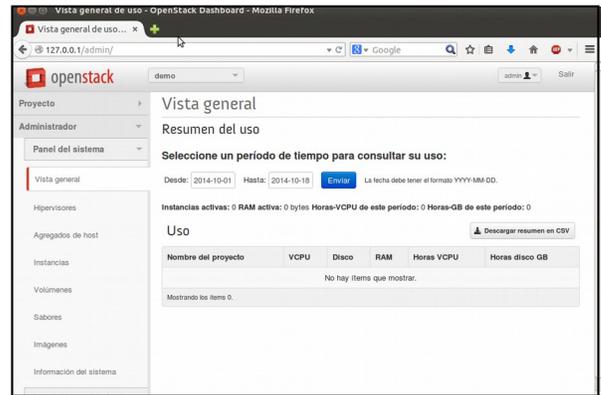


Figura 5: Pantalla del administrador del Dashboard

4. Aspectos importantes de la infraestructura instalada

OpenStack está diseñado para que lo utilicen usuarios muy diversos a los que se les pueden asignar diferentes **roles** que servirán para controlar las acciones que cada uno de estos usuarios puede realizar. En la configuración por defecto, la mayoría de las acciones no llevan asociadas ningún rol, pero en el administrador del cloud recae la responsabilidad de configurar apropiadamente estas reglas con el objeto de mantener integridad, consistencia y eficiencia global de la infraestructura. Una vez definido el rol que va a tener cada usuario, se puede comenzar a trabajar con la configuración de los recursos existentes.

Cada usuario tiene asociado un **proyecto** (tenant) el cual puede ejecutar un conjunto de instancias con características en común y puede utilizar una serie de imágenes de sistemas o tener limitado el uso de recursos del cloud.

Las **imágenes** son plantillas para las máquinas virtuales que se van a crear. Estas imágenes pueden tener diferentes formatos, pueden ser sistemas operativos limpios recién instalados o sistemas con aplicaciones completamente configuradas. El servicio que proporciona las imágenes, Glance, es el responsable de almacenar y gestionar las imágenes en OpenStack.

Las **instancias** son las máquinas virtuales que se ejecutan en los nodos y son gestionadas por Nova. Se pueden lanzar cualquier número de instancias a partir de una determinada imagen. Cada instancia se ejecuta de una copia de una imagen base, por lo que las modificaciones que se realicen en la instancia no alteran la imagen en la que se basa. Mediante el uso de

instantáneas (snapshots) de las instancias, se pueden crear nuevas imágenes que sí guardan todas las modificaciones realizadas hasta ese momento en la instancia.

Cuando se lanza una instancia, se debe seleccionar un conjunto de recursos virtuales, conocido como *sabor* (flavor). Un sabor define para una instancia el número de CPUs virtuales, la RAM, si dispone o no de discos efímeros, etc. OpenStack preinstala una serie de sabores, que el administrador puede modificar.

Para el caso de este trabajo donde se ha instalado OpenStack Juno con una arquitectura monolítica, el principal inconveniente es la imposibilidad de selección de hipervisores, debido a que esta arquitectura obliga a usar Qemu. Este hipervisor es adecuado cuando se quiere montar una infraestructura cloud para desarrollo, pero no se recomienda su implementación en entorno de producción.

Con respecto a las imágenes, OpenStack trae por defecto las imágenes Cirros. Cirros es una distribución Linux con características mínimas que fue diseñada para su uso como imagen de prueba en Cloud como OpenStack. Si la implementación utiliza QEMU o KVM, se recomienda utilizar las imágenes en formato qcow2 (la imagen más reciente qcow2 de 64 bits es cirros-0.3.2-x86_64-disk.img), sin embargo se decidió agregar una imagen Ubuntu 14.04 en formato qcow2 y de tamaño 254,2 Mbytes, para analizar el comportamiento de la infraestructura. A partir de esta imagen llamada Ubuntu Cloud se generan las instancias, tal como muestra la figura 6.

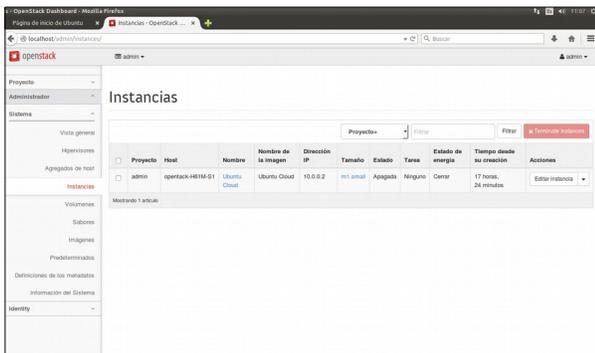


Figura 6: Instancias generadas a partir de la imagen Ubuntu Cloud

En este punto solo queda lanzar las instancias para lo cual una imagen, un sabor y opcionalmente otros atributos. OpenStack copia la imagen base al disco que utilizará la instancia como primer disco (vda), cuanto más pequeña sea la imagen, más rápido será el lanzamiento. Dependiendo del sabor, también se crea un nuevo disco vacío (vdb). El nodo de computación conecta en su caso mediante iSCSI con nova-volume y mapea el volumen escogido como vdc. Se pueden lanzar tantas instancias como capacidad de recursos físicos hayan, como se muestra en la figura 7. Para el caso de

la implementación de este trabajo, debido a que se ha usado un nodo con 4 Gbyte de RAM, solo se pueden levantar dos instancias, antes que se consuma toda la memoria.

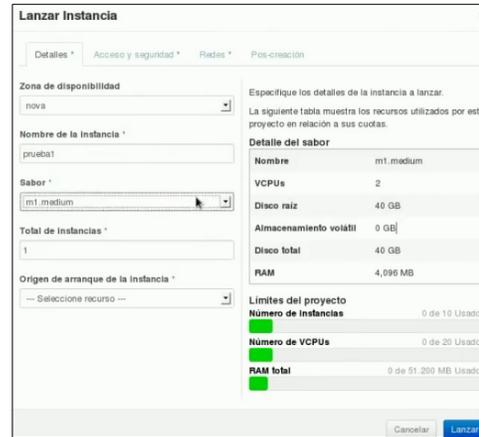


Figura 7: Proceso de lanzamiento de una instancia

Una vez que las instancias están levantadas, el cloud esta listo para las pruebas de producción. Para ello se puede acceder por SSH desde un nova-client.

5. Evaluación de la Infraestructura

Las principales ventajas que ofrece OpenStack están relacionadas con su característica de código abierto, tales como: ahorro de costos, mayor independencia frente a soluciones propietarias, mayor modularidad, mayores opciones de personalización, etc. Un aspecto muy importante es la capacidad de versionado y la estabilidad de cada una, esto se debe a que cada seis meses se libera una versión y se deja disponible en el repositorio Git, cada una de ellas mejora al menos dos componentes de la versión anterior y agrega al menos un componente nuevo.

El mayor inconveniente que se presentó al instalar la infraestructura fue la configuración de los archivos de sistemas y la necesidad de dependencia de ciertos componentes de software necesarios para comenzar el montado (por ejemplo es necesario clonar los repositorios, instalar las librerías necesarias para el soporte de virtualización, conectividad, etc. Una vez solucionado este problema, la herramienta DevStack, facilita el proceso de instalación automatizando las tareas de configuración de cada módulo.

Cuando se ha finalizado la instalación de OpenStack, el manejo del entorno es bastante amigable desde el Dashboard, sin embargo, se puede manipular los componentes desde línea de comando. Esta tarea si bien es más complicada y requiere de mayor conocimiento de los detalles de configuración, es más potente. Ya sea que se use la interface grafica o la consola, se pueden definir

roles, proyectos, levantar imágenes, lanzar instancias, etc. de manera gráfica.

6. Futuros Trabajos y Conclusiones

Uno de los mayores problemas que se presentaron con la implementación de la arquitectura monolítica fue el soporte de un único hipervisor (QEMU) y la imposibilidad de contar con una infraestructura para la producción. Debido a esto es que se está trabajando en la migración a una arquitectura distribuida, donde cada componente de la infraestructura estará instalado en un nodo diferente, lo que permitirá que se pueda seleccionar qué hipervisor instalar. Además, otro aspecto importante a tener en cuenta es la necesidad de migrar a la última versión estable: Kilo, esta versión tiene la ventaja de contar con el componente IRONIC que permite ofrecer virtualización como servicio, esto permitirá que la infraestructura virtualise en forma nativa sobre el hardware, lo cual disminuye los retardos provocados por el proceso de virtualización.

A partir de esta nueva implementación será posible medir el rendimiento de algoritmos sobre diferentes hipervisores con el objeto de determinar cuál es el de mejores resultados en términos de rendimiento de la virtualización, En este sentido. La idea es tomar algoritmos tradicionales y correrlos sobre arquitecturas tradicionales y sobre OpenStack para determinar cuánto se degrada la performance en una infraestructura de cloud virtualizada. En este sentido, toda la experiencia adquirida en la instalación, configuración y puesta en marcha del cloud será directamente aprovechable.

Un aspecto que es de suma importancia para las infraestructuras de cloud es el manejo de la seguridad, en este trabajo, no se ha estudiado el componente Keystone, que permite el manejo de la autenticación y validación de los usuarios, ni tampoco se ha trabajado con los módulos para consumir en forma segura recursos desde Internet. Este aspecto es muy importante al momento de convertir la infraestructura privada en híbrida.

Entre los aspectos más importantes de la implementación de OpenStack Juno, es la ventaja que provee su modularidad, permitiendo instalar los componentes que se necesiten, el otro aspecto es que esta implementación es no apropiativa, debido a que los

servicios del cloud se levantan cuando se necesitan y puede convivir con otras aplicaciones. Una vez que se ha montado el cloud, se levanta el servicio por consola y se usa mediante el dashboard.

Otro aspecto importante es la posibilidad de instalación sobre hardware sin demasiados requerimientos de memoria y disco, esto ha permitido poder usar nodos con características estándares, esto se debe fundamentalmente a que la instalación se realizó con Ubuntu 14.04 como sistema operativo de base, cuyos requerimientos para instalación son 4GB de RAM y 500 Mbyte de disco. Este es un aspecto distintivo sobre otras infraestructuras de cloud, como por ejemplo OpenNebula cuyo sistema operativo de base recomendable es Debian, el cual requiere mayor capacidad de hardware.

7. Bibliografía

- [1] Schubert, Jeffery, Neidecker-Lutz. "The Future of Cloud Computing: Opportunities for European Cloud Computing Beyond 2010:—expert Group Report. European Commission, Information Society and Media, 2010.
- [2] Mell, Grance. "The NIST definition of cloud computing." NIST Special Publication 800 – 145, 2011.
- [3] Rodríguez, Chávez, Martín, Murazzo, Valenzuela. "Interoperabilidad en cloud computing". WICC 2011. Rosario Argentina.
- [4] OpenStack. "Introduction to OpenStack, Chapter 2. Brief Overview". URL: <http://docs.openstack.org/training-guides/content/module001-ch002-brief-overview.html>.
- [5] OpenStack. "Introduction to OpenStack, Chapter 4. OpenStack Architecture". URL: <http://docs.openstack.org/training-guides/content/module001-ch004-openstack-architecture.html>.
- [6] OpenStack. "Juno". URL: <https://www.openstack.org/software/juno/>
- [7] OpenStack. "DevStack - an OpenStack Community Production". URL: <http://docs.openstack.org/developer/devstack/>