Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación

Centro de Investigación en Comunicación y Redes (CICORE)
Laboratorio de Redes Móviles e Inalámbricas (ICARO)

Diseño e implementación de una infraestructura de virtualización basada en un entorno de nube para el Laboratorio ICARO

Trabajo Especial de Grado
presentado ante la ilustre
Universidad Central de Venezuela
por el bachiller:
Francisco Lugo Estrella
para optar al título de Licenciado en Computación.

Tutor: David Pérez Abreu

Caracas, 2014

Universidad Central de Venezuela
Facultad de Ciencias
Escuela de Computación
Centro de Investigación en Comunicación y Redes (CICORE)
Laboratorio de Redes Móviles e Inalámbricas (ICARO)

Acta del veredicto

Quienes suscriben, Miembros del Jurado designado por el Consejo de la Escuela de Computación para examinar el Trabajo Especial de Grado, presentado por el Bachiller Francisco Lugo Estrella C.I.: 19954122, con el título "Diseño e implementación de una infraestructura de virtualización basada en un entorno de nube para el Laboratorio ICARO", a los fines de cumplir con el requisito legal para optar al título de Licenciado en Computación, dejan constancia de lo siguiente:

Leído el trabajo por cada uno de los Miembros del Jurado, se fijó el día 19 de Mayo, a las 14:30, para que su autor lo defendiera en forma pública, en la Sala 01 de la Escuela de Computación, lo cual este realizó mediante una exposición oral de su contenido, y luego respondió satisfactoriamente a las preguntas que les fueron formuladas por el Jurado, todo ello conforme a lo dispuesto en la Ley de Universidades y demás normativas vigentes de la Universidad Central de Venezuela. Finalizada la defensa pública del Trabajo Especial de Grado, el jurado decidió aprobarlo.

En fe de lo cual se levanta la presente acta, en Caracas el 19 de Mayo de 2014, dejándose también constancia de que actuó como Coordinador del Jurado el Profesor Tutor David Pérez Abreu.

Prof. David Pérez Abreu (Tutor)

Profa. María Elena Villapol
(Jurado Principal)

Prof. Jaime Parada
(Jurado Principal)

RESUMEN

Título:
Diseño e implementación de una infraestructura de virtualización basada en un entorno de nube para el Laboratorio ICARO

Autor:
Francisco Lugo Estrella

Tutor:
David Pérez Abreu

Las tendencias actuales en el campo de las Tecnologías de la Información se ajustan a los crecientes volúmenes y flujos de datos en una red global, adicionalmente, pareciera que la demanda en almacenamiento y procesamiento de datos supera siempre a la oferta, haciendo necesario superar constantemente los límites. Es con base en estos hechos que el paradigma de procesamiento y almacenamiento en la nube se ha abierto camino para ocupar el nicho de las Tecnologías de la Información. En el presente trabajo de investigación, se explora el tema de la computación en la nube, una tecnología en auge entre usuarios comunes, desarrolladores y organizaciones con requerimientos de tecnologías de la información. La nube brinda acceso a recursos computacionales elásticos a gran escala y permite desplegar aplicaciones accesibles a través de Internet. La investigación explora la opción del despliegue de una nube en un centro de datos local a una organización. En particular se toma como caso de estudio el sistema de nube OpenStack.

Palabras claves:
Computación en la nube, Nube, OpenStack, Tecnologías de la Información

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

En la actualidad, el término de entorno en la nube, o simplemente nube, es muy popular en el ámbito de las tecnologías de la información; sin embargo, lo que se referencia con este término puede variar. Una analogía interesante y conocida sobre la computación en la nube es la de las plantas eléctricas y sus usuarios. Antes, en cualquier organización que hacía uso de energía eléctrica para funcionar, se utilizaban generadores internos que proveían de energía de forma independiente. Cuando las condiciones fueron las ideales, es decir, cuando la energía eléctrica podía ser provista como un servicio externo completo y con garantías, se comenzó a utilizar este servicio para satisfacer las necesidades energéticas de las organizaciones. Esto traía ventajas en comparación con el esquema anterior, como minimizar los gastos en mantenimiento, porque la organización se encargaba de pagar por el servicio y poco más. Al conectar un aparato eléctrico, no es necesario conocer cómo fue generada la energía ni cómo llega a la toma, tampoco parece haber un límite en el tiempo de uso del aparato, hay un nivel de abstracción que permite un amplio dinamismo. Este proceso de cambio se puede llevar a las tecnologías de la información, con organizaciones que puedan prestar un servicio masivo y confiable, posiblemente retirando un módulo en las organizaciones que requieran este servicio.

Hay que tomar en cuenta que las tecnologías de la información están cada vez más involucradas en lo cotidiano, y que los flujos y volúmenes de datos crecen en Internet. La nube propone una forma de solventar algunos de los inconvenientes que se encuentran al abordar esta situación.

1.1. Plantemiento del problema

El uso de la computación en la nube ha crecido desde sus primeras implementaciones. Las organizaciones que hacen uso de las tecnologías de la información están siempre atentas, pero no hay manera de que pudiesen haber anticipado el crecimiento y penetración de la nube en el campo de las tecnologías de información. Grandes empresas han decidido adoptar este modelo como una nueva forma de manejar la información en sus *centro de datos*, aprovechando sus ventajas en múltiples áreas. Esto les ha permitido hacer disponibles servicios especiales, tanto a consumidores finales como a proveedores de otros servicios. Ser proveedor de proveedores implica grandes volúmenes, y en estos casos el modelo se explota al máximo. Pero las ventajas de la computación en la nube no se aprecian exclusivamente en data centers masivos. Actualmente, es posible comprar casi cualquier tipo de servicio en la nube, desde cualquier parte del mundo, para satisfacer necesidades internas de una organización. Por ejemplo, delegar algunas operaciones a una entidad externa, permite a las organizaciones ahorrar en gastos de mantenimiento de equipos y consumo de energía, aunque, por distintos motivos, esta no siempre resulta la mejor opción. Además, las implementaciones del modelo siguen incorporando nuevas características y funcionalidades, haciendo posible la creación de nubes privadas, internas en la organización. El software libre se involucra en diferentes etapas de la construcción de una nube, desde la virtualización de la infraestructura hasta la implementación de sistemas de nube. Esto resulta ideal para organizaciones pequeñas o medianas con requerimientos puntuales. Tomando en cuenta lo anterior, se plantea la siguiente interrogante: ¿Es posible diseñar e implementar una infraestructura de virtualización basada en un entorno de nube para el Laboratorio ICARO?

1.2. Objetivo general

El objetivo general del presente trabajo es:

Diseñar e implementar una infraestructura de nube basada en virtualización para Laboratorio ICARO.

1.3. Objetivos específicos

Los objetivos específicos del presente trabajo de investigación son:

- Diseñar el modelo de servicio a ser desplagado tomando en cuenta una infraestructura de virtualización.
- Seleccionar las herramientas de software y hardware acordes con los diseños de solución planteados.
- Instalar, configurar y adecuar los componentes de hardware y software necesarios.
- Diseñar e implementar escenarios de pruebas de estrés y corrección.
- Configurar y adecuar el entorno de virtualización en la nube con base en los resultado obtenidos en las pruebas realizadas.
- Lanzar a producción la solución configurada y construida.

1.4. Justificación

La proliferación de sistemas de nube es consecuencia directa del aumento en su uso y del interés que muestran las organizaciones en esta tecnología; cada vez más organizaciones adoptan la nube como su forma de manejar la información internamente. Es la misma idea del data center convencional pero aprovechando las ventajas de la nueva tecnología disponible. Estas ventajas, que se comentan a lo largo del documento, pueden llegar a superar las del *centro de datos* convencional en algunos casos.

En el Laboratorio ICARO de la Escuela de Computación de la Universidad Central de Venezuela, los estudiantes tienen acceso directo a máquinas físicas para llevar a cabo prácticas de diferentes materias. Para completar la mayoría de estas prácticas, es necesario instalar software y aplicar diferentes configuraciones, haciendo necesario un tiempo de preparación del laboratorio antes de la práctica y más tiempo, después de la práctica, para restaurar el sistema a su estado original. Esta situación indica que una posible solución está en la virtualización, pero una solución de virtualización local para cada máquina física no lograría ningún cambio, se repetirían las mismas operaciones en

máquinas virtuales en lugar de máquinas físicas y tomaría la misma cantidad de tiempo, antes y después de la práctica. Sería necesaria una solución en la que las máquinas físicas tengan acceso a una infraestructura de virtualización, que puedan solicitar recursos, utilizarlos y descartarlos sin requerir ningún cambio en la configuración de la máquina física, es por esto que un ambiente de virtualización en la nube resulta prometedor para dicha organización. Además, en el Laboratorio ICARO hay dispositivos de almacenamiento en red que no están siendo utilizados y, adicionalmente, el laboratorio cuenta con un conjunto de servidores que se encuentran subutilizados. Se propone darles un uso, convirtiéndolos en elementos de un sistema que resuelva la problemática mencionada. Incursionar en la computación en la nube se ha convertido en un deber para organizaciones que mantienen contacto directo con la tecnología; y una institución en la que se intenta generar conocimientos, no debe descartar la oportunidad de indagar e investigar sobre una nueva tecnología.

1.5. Distribución del documento

El Capítulo 1 muestra una breve descripción del contenido del documento, haciendo énfasis en el planteamiento del problema y los objetvos de la investigación. En el Capítulo 2 se presenta la definición y algunos conceptos inherentes al tema de la computación en la nube, esto con la intención de comenzar a entender su importancia y el impacto que puede tener en otros sistemas. También incluye una visión general de los problemas de seguridad a los que se enfrenta la computación en la nube, y una vista de las implementaciones actuales de sistemas de nube. El Capítulo 4 se presentan aspectos de diseño de una nube en general, y la descripción del diseño de la solución implantada, en particular el sistema de nube OpenStack. También muestra la implementación de la solución, y se detallan las especificaciones del sistema instalado. En el Capítulo 3 se especifica el método y las herramientas utilizadas en la investigación. En el Capítulo 5 se explican las pruebas llevadas a cabo sobre el sistema instalado y se muestran los resultados obtenidos. En el Capítulo 6 se presentan las conclusiones de la investigación. Finalmente el Capítulo 7 provee una guía para la instalación del caso de estudio de la investigación.

Capítulo 2

Marco teórico

En este Capítulo se presenta la definición de computación en la nube que se sigue en esta investigación, así como algunas ideas relacionadas con el tema.

2.1. Computación en la nube

Computación en la nube (*cloud computing*) es un término que se refiere a los sistemas de hardware y software especializados, que permiten prestar servicios con características particulares de elasticidad en la presentación de los recursos de cómputo. También se utiliza para ciertas aplicaciones disponibles como un servicio a través de Internet.

Para propósitos generales, es común que este término sea tomado para designar a un modelo de interacción, en el que los distintos componentes de varios sistemas de cómputo deben permitir el acceso conveniente y ubicuo a conjuntos configurables de recursos compartidos (red, cómputo, almacenamiento y servicios administrativos). Para cumplir el modelo, estos recursos deben poder ser obtenidos y liberados rápidamente, con un mínimo esfuerzo de administración o servicio de interacción extra[1]. Otros entes involucrados en el modelo son el proveedor de la nube, que administra y controla la infraestructura en su totalidad, incluyendo el nivel físico, el consumidor de la nube que utiliza los recursos disponibles (comúnmente siendo proveedor de algún servicio en la nube) y el usuario, que tiene acceso a las aplicaciones elásticas en la nube (usuario de servicio), tres conjuntos no excluyentes.

La computación en la nube sigue evolucionando, las organizaciones involucradas en el desarrollo de sistemas de nube muestran un gran interés, haciendo un esfuerzo constante para encontrar la mejor forma de implementar el modelo. A continuación veremos las características esenciales que completan la definición.

2.1.1. Definición NIST

Según el NIST (National Institute of Standards and Technology), el modelo de cómputo en la nube está compuesto por cinco características esenciales, tres modelos de servicio y cuatro modelos de despliegue, con los que se pretende categorizar los sistemas de cómputo en la nube que permitan un uso seguro y efectivo de la misma [1].

Las **características esenciales** propuestas por el NIST permiten identificar las partes necesarias de un sistema genérico de cómputo en la nube:

- Autoservicio por demanda. Un consumidor puede aprovisionarse de capacidades de cómputo unilateralmente cuando las necesite, sin requerir interacción humana con el proveedor de cada servicio.

- Acceso a través de la red. Los recursos están disponibles en la red y son accedidos mediante mecanismos estándares que permiten el uso de plataformas heterogéneas como tabletas, teléfonos inteligentes, computadores portátiles y estaciones de trabajo.

- Conjuntos (pools) de recursos. El proveedor de recursos de cómputo tendrá conjuntos de recursos para servir a múltiples consumidores utilizando un modelo que lo permita, con diferentes recursos físicos o virtuales dinámicamente asignados y reasignados de acuerdo con las demandas del consumidor.

- Elasticidad. Los recursos pueden ser elásticamente provistos y liberados, en algunos casos automáticamente para escalar rápidamente y estar a la medida de las demandas. Para el consumidor, los recursos usualmente parecen ser ilimitados y pueden ser requeridos en cualquier momento y en medidas variables.

- Servicio medido. El uso de los recursos puede ser monitoreado, controlado y reportado transparentemente tanto para el proveedor como para el consumidor.

Los **modelos de servicio** muestran la interacción entre dos partes importantes en este tipo de sistemas. El proveedor, que maneja y administra la nube con todos sus componentes estáticos, y el consumidor que utiliza la nube.

- Software as a Service (SaaS). Se puede proveer al consumidor de la posibilidad de utilizar las aplicaciones del proveedor siendo ejecutadas en una infraestructura de cómputo en la nube.
- Platform as a Service (PaaS). Esto permite desplegar, en la infraestructura de la nube, una aplicación creada por el consumidor utilizando herramientas soportadas por el proveedor, aportando ventajas para ambos.
- Infrastructure as a Service (IaaS). Es la capacidad de permitir al consumidor utilizar distintos recursos de cómputo de bajo nivel en los que puede desplegar y ejecutar software, es equivalente al alquiler de un centro de datos virtual.

Los **modelos de despliegue** permiten caracterizar el ambiente físico en el que la nube está desplegada y cuáles pueden ser sus usuarios.

- Nube Privada. La infraestructura de la nube está provista para el uso exclusivo de una organización, pudiendo agrupar múltiples consumidores.
- Nube Comunitaria. La infraestructura será utilizada exclusivamente por una comunidad cuyos miembros compartan un fin u objetivo particular.
- Nube Pública. La infraestructura de la nube es utilizada por el público general.
- Nube Híbrida. La infraestructura es el resultado de la composición de dos o más infraestructuras de nubes que permanecen siendo entidades únicas pero permiten la portabilidad de datos y aplicaciones.

Un punto a resaltar es que los modelos de servicio propuestos por el NIST pueden ser utilizados como características de productos que pueden ser ofrecidos por una nube o no. Además, existen otros términos que están aislados de la computación en la nube, como *grid computing* que sugiere protocolos para ofrecer computación compartida y almacenamiento remoto, con la diferencia de que estos protocolos no llevan a un ambiente de software y hardware de ningún tipo específico, es decir, estos sistemas no operan como una nube; es más acertado decir que algunos protocolos y paradigmas comunes en *grid computing* y en *clusters* son utilizados como parte del funcionamiento interno de una nube. Los data centers internos a una organización, pequeños o medianos y de uso específico, no logran beneficiarse de las ventajas de la computación en la nube y no son considerados como ejemplos de este modelo. Sin embargo, hay nubes privadas que, aunque no están disponibles al público en general, se les considera una nube por la forma en la que administran y utilizan sus recursos.

2.1.2. Elasticidad y escalabilidad

La elasticidad es la habilidad de un sistema de software para escalar dinámicamente la cantidad de recursos provistos a clientes mientras su carga de trabajo incrementa o decrementa. Representa una de las ventajas principales de la computación en la nube, en la que los recursos son dinámicamente añadidos y liberados. Sin embargo, no hay una forma de definir la elasticidad con precisión [3] y, por lo tanto, tampoco de medirla o cuantificarla, no están incluidos parámetros como cuán rápido se debe asignar un recurso requerido, qué tan seguido son requeridos o liberados, ni otros parámetros que puedan indicar medidas de eficiencia. Esto hace que sea una característica que está presente o no en un sistema de software, y en la computación en la nube debe estar presente para cumplir con las características y además porque permite optimizar la productividad y utilización de los recursos del sistema, ahorrar energía y costos, cuestiones fundamentales para sistemas de alta escalabilidad.

Para la escalabilidad, se puede hablar de dos tipos o formas, porque se puede relacionar con la aplicación y con la plataforma de ejecución. La escalabilidad de aplicación es una propiedad que hace que las aplicaciones mantengan sus objetivos de rendimiento mientras aumenta la carga de trabajo hasta cierto punto, lo que quiere decir que hay un rango de escalabilidad, es finita.

La escalabilidad de aplicación debe estar soportada por la plataforma de ejecución y está limitada por su diseño. Por otro lado, la escalabilidad de la plataforma es la habilidad que tiene la misma para crecer en la cantidad de recursos y mantener el control y la eficiencia, la plataforma de ejecución comprende capas de hardware y software que la aplicación debe utilizar para ser ejecutada. La aplicación puede escalar *verticalmente* para que los nodos puedan soportar un aumento en la carga de trabajo (se añaden más recursos a un nodo) u *horizontalmente* para que el sistema pueda manejar una mayor carga de trabajo (cuando se añaden más nodos físicos o virtuales).

Para garantizar la presencia de estas características en los sistemas de nube se necesita un diseño que las tome en cuenta integralmente. El funcionamiento interno de estos sistemas asigna una alta prioridad al mantenimiento estas características, porque es una de las ventajas principales que brindan al usuario final.

2.1.3. Virtualización

Los servicios prestados por las grandes nubes son provistos usualmente por data centers compuestos por miles de computadores. Estos son construidos para servir a muchos usuarios y alojar muchas aplicaciones diferentes. Existen distintas formas de hacerlo pero considerando el propósito y los requerimientos, la virtualización del hardware puede ser considerada como la mejor posibilidad para superar la mayoría de los problemas de mantenimiento y administración, permitiendo el dinamismo necesario para el funcionamiento de la nube. Existen distintas formas de virtualización, cada una con sus ventajas y características particulares, permitiendo ajustar el sistema a los requerimientos.

La idea de hacer virtuales los recursos de un computador, incluyendo procesadores, memoria y dispositivos de entrada/salida ha estado bien establecida durante un buen tiempo. La virtualización permite ejecutar múltiples sistemas operativos y pilas de software en una plataforma física. En la Figura 2.1 se muestra la capa de software de usuario, el monitor de la máquina virtual, también llamado hypervisor, que media el acceso al hardware físico presentando una máquina virtual a cada sistema operativo huésped.

El desarrollo de distintas tecnologías ha permitido la adopción creciente de la virtualización. Las ventajas iniciales son una mejor compartición y utiliza-

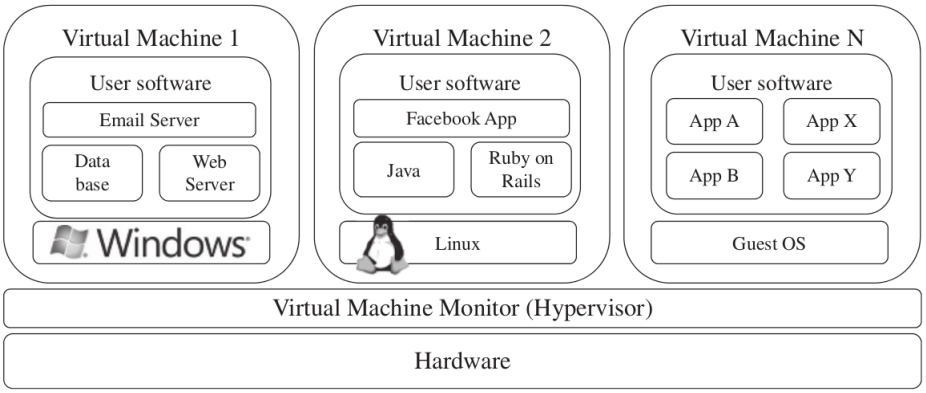

Figura 2.1: Servidor virtualizado con máquinas virtuales ejecutando distintos sistemas operativos y software de usuario

ción de los recursos, mejor administración y mayor confiabilidad. En relación al manejo de la carga de trabajo, la virtualización provee facilidades básicas para la computación en la nube. El aislamiento, la consolidación y la migración, son características deseables para lograr el funcionamiento óptimo de un sistema de computación en la nube.

- El **aislamiento** de la carga de trabajo es conseguido ya que todas las instrucciones de los programas están completamente confinadas en la máquina virtual, lo que ayuda a mejorar la seguridad. Se consigue mayor confiabilidad, porque las fallas de software de una máquina virtual no afectan a las otras. Además, se obtiene un mejor control de rendimiento porque la ejecución de una máquina virtual no perjudica a las otras.
- La **consolidación** de distintas cargas de trabajo individuales y heterogéneas en una plataforma física única lleva a una mejor utilización del sistema y permite resolver incompatibilidades de software y hardware en caso de actualizaciones, permitiendo ejecutar la versión nueva de un sistema operativo y la anterior.
- La **migración** de la carga de trabajo facilita el mantenimiento del hardware, balanceo de carga y recuperación en caso de desastre. Se

> puede hacer encapsulando el estado del sistema operativo huésped en la máquina virtual y permitiéndole ser suspendida, completamente serializada, migrada a otra plataforma y resumida inmediatamente o preservada para ser restaurada luego. El estado de una máquina virtual, incluye la imagen del disco, archivos de configuración y una imagen de la RAM.

Actualmente existen distintas plataformas de hipervisores como VMWare, Xen, y QEMU que son la base de distintos ambientes de computación en la nube. Es posible encontrar muchas posibilidades diferentes entre sí y es común que sean personalizadas para ajustarlas al uso que se les dará. La utilización de la virtualización abre ciertas posibilidades en la implementación del modelo de nube. Algunos sistemas de nube funcionan como un hypervisor de mayor escala, presentando máquinas virtuales con recursos elásticos que pueden ser rápidamente desplegadas sobre redes virtuales. Este enfoque es muy común actualmente, el sistema mantiene las máquinas virtuales en funcionamiento y cede el control sobre estas máquinas a los consumidores, permitiendo el acceso conveniente, ubicuo y por demanda a los diferentes recursos de cómputo.

2.2. Seguridad en la nube

Con toda nueva tecnología, sus posibles amenazas y brechas de seguridad crecen con su mismo desarrollo. La computación en la nube está en crecimiento y ha habido distintas discusiones en cuanto a su seguridad. Algunas de las posibles brechas de seguridad son inherentes a cualquier tipo de sistema que utilice redes de comunicación de datos; con esto en mente, abordaremos en ésta sección la seguridad en la nube desde diferentes perspectivas.

Los problemas de seguridad en la nube están aparte de los inconvenientes con las redes. Cada nube sirve a un conjunto de usuarios con recursos disponibles, primero se debe garantizar que sólo los usuarios tengan acceso administrativo a estos recursos. Desde ese punto en adelante algunos aspectos de seguridad dependen de la implementación. En la actualidad hay intentos constantes de alcanzar estándares, pero todavía se encuentran diferencias integrales en las formas de implementación del modelo de nube. También hay un alto desarrollo de tecnologías que se involucran en el funcionamiento de la nube,

esto hace que exista más de una forma correcta de implementación, cada una con sus propias tecnologías de bajo nivel.

Entre las principales versiones de sistemas de nube es muy común encontrar tecnologías de virtualización, que permiten crear máquinas virtuales con recursos elásticos. En teoría, de este tipo de sistemas se podría esperar un comportamiento que comprometa la seguridad en distintos niveles. Habría distintas máquinas físicas y virtuales, distintos sistemas operativos a tomar en cuenta, distintas formas de almacenamiento y distintos consumidores, cada uno con necesidades que deben ser satisfechas y datos que deben estar protegidos. Todo esto se complica si consideramos los volúmenes masivos de datos en movimiento. Además, un sistema de este tipo se presta para diferentes modelos de funcionamiento nuevos y con problemas de seguridad no resueltos. Sin embargo, los posibles ataques a una nube son parecidos a los de un centro de datos estándar, aunque se incluye el peso extra de la configuración inherente a la implementación y las características físicas del sistema.

Actualmente, en cuanto a seguridad, los proveedores de servicio utilizan algoritmos de cifrado robustos para la confidencialidad de datos almacenados. Dependen de protocolos de seguridad como SSL (Secure Sockets Layer), IPSec (Internet Protocol Security) y otros, para proteger los datos en transmisiones de red. Para la disponibilidad y alto desempeño, utilizan tecnologías de virtualización y aplican esquemas fuertes de autenticación y autorización en sus dominios. De cualquier forma, como una nueva infraestructura/plataforma que lleva a nuevos modelos de aplicaciones y servicios, los requerimientos de seguridad en una nube son diferentes de los tradicionales. Como señala el Dr. K. M. Khan [4], cifrado, firmas digitales, seguridad a nivel de red, firewalls, y el aislamiento de los ambientes virtuales son todos importantes para la seguridad de una nube, pero estos solos no harán a la computación en la nube confiable para los consumidores.

Cuando se necesita asegurar un bien, se lleva a cabo un proceso de análisis de seguridad. Esto incluye qué tipo de bien se va a proteger, qué amenazas posibles existen y qué medidas pueden ser tomadas para detener la ocurrencia de ataques. Además se deben tomar en cuenta las propiedades del bien que deben mantenerse intactas, y el nivel de tolerancia al ver comprometida alguna de estas propiedades. Este análisis se puede aplicar a cualquier sistema, como se trata de una nube, puede llegar a ser particularmente difícil,

serían muchos aspectos a tomar en cuenta. Además, existe la perspectiva del proveedor y la del consumidor, que no siempre están de acuerdo. Sin embargo, es un proceso de recolección de experiencias a gran escala que brinda resultados a largo plazo.

2.2.1. Amenazas identificadas

Algunos problemas nuevos surgen en la computación en la nube, ligados a las deficiencias en la seguridad y relacionados con el modelo de negocio utilizado comúnmente. En este modelo, múltiples consumidores residen en la misma nube. En un sistema de este tipo, un consumidor o usuario que logre atacar al sistema, puede poner en peligro a muchos otros usuarios o al proveedor. No se trata de un sistema de elementos independientes sino en conjuntos. Las fallas en cuanto a seguridad tienen peores consecuencias en este modelo, es una cuestión de alcance de la seguridad. A continuación se comentan algunos de los problemas de seguridad que se han encontrado al implementar [5] una nube.

- **Incompatibilidad**. Entre una nube y otra, es necesario hacer cambios en aplicaciones y datos para lograr que sean compatibles. Actualmente hay poco que ofrecer en forma de herramientas, procedimientos o formatos estándar de datos que permitan garantizar la portabilidad de los datos, aplicaciones y servicios. Es difícil migrar de un proveedor a otro, o a un esquema distinto al de la nube, una vez que el sistema está en uso. Esto tiene como consecuencia una dependencia de un proveedor particular por parte de sus consumidores y la falta de portabilidad de sus aplicaciones y datos. Para corregir esto, claramente se necesita una estándarización a gran escala de las APIs que se utilizan en sistemas de cómputo en la nube, es un trabajo a largo plazo.
- **Eliminación de datos**. Los consumidores de una nube no tienen la certeza en cuanto al manejo de los datos de su proveedor, pudiendo causar problemas. Cuando se hace una petición para eliminar un recurso de la nube, esto podría no cumplirse inmediatamente. La eliminación de datos certera o en el tiempo correcto puede ser imposible, sea porque los datos están almacenados pero no disponibles o porque el disco a ser destruido también almacena datos de otros clientes. Cuando hay múltiples inquilinos y se reutilizan los recursos de hardware, existe un riesgo

extra para el consumidor de la nube comparado con los que utilizan hardware dedicado.

- **Ataques a nivel de Máquina Virtual**. El problema es que existen ciertas brechas de seguridad inherentes a cada hipervisor porque es costoso adaptarlos a su utilización en la nube. Las vulnerabilidades pueden ser mitigadas utilizando un sistema de detección/prevención de intrusiones e implementando un firewall adecuado.

- **Abuso y utilización maliciosa de una nube**. Así como en otro tipo de sistemas, en algunos sistemas de computación en la nube se utilizan procesos de registro y validación muy pobres, permitiendo la anonimidad y haciendo más fácil el acceso a usuarios indeseables. Para reducir estos riesgos, es necesario implementar procesos de registro y validación estrictos, además de formas de monitorización de tráfico de usuario.

- **APIs inseguras**. Los consumidores utilizan un conjunto de interfaces de software para interactuar con los servicios de la nube. El aprovisionamiento, la administración, la coordinación y la monitorización, de los servicios de la nube, son llevados a cabo usualmente a través de estas interfaces. Si existen faltas en aspectos de seguridad de estas interfaces, la nube completa está expuesta a amenazas de seguridad como acceso anónimo, tokens de identificación o contraseñas reusables, autenticaciones y autorizaciones incorrectas, capacidades de monitorización limitadas y otros problemas conocidos. Para tratar con este problema, las interfaces que son utilizadas en el modelo de nube del proveedor deben ser analizadas para intentar encontrar brechas de seguridad. Es necesario utilizar procesos de autenticación y control de acceso seguros. Es recomendable usar cifrado para las transmisiones de contenido y entender claramente la cadena de dependencias asociada con cada API.

- **Fallas de aislamiento**. Los servicios en una nube son provistos por una infraestructura compartida. Los componentes utilizados para construir algunos recursos en la nube (como particiones de discos, caché de procesador o unidades de procesamiento gráfico) no están diseñados para ofrecer propiedades de aislamiento fuerte ni en diferentes niveles. Los hipervisores, que son los bloques básicos para la computación en

la nube, pueden fallar en algunos casos, entregando el control al sistema operativo huésped. Los atacantes podrían enfocarse en esta falla de aislamiento de cualquier consumidor de la nube para ganar acceso no autorizado a los datos y aplicaciones. Para limitar los riesgos se deben utilizar medidas estratégicas de aislamiento, mientras menos componentes tengan acceso a la información, menos probabilidades existen de que se filtre. Esto se puede mejorar implementando mejores prácticas en las actividades de instalación, configuración y monitorización, y utilizar un control de acceso y de autenticación fuertes. Además, manejar las vulnerabilidades restantes individualmente y hacer escaneos de vulnerabilidades periódicos.

- **Pérdida o filtrado de datos**. Los servicios en Internet necesitan garantizar la protección de los datos de sus consumidores. Es tan importante que sin esta característica, cualquier servicio está destinado al fracaso. Puede ser causado por control insuficiente de autenticación, autorización y auditoría, uso inconsistente de cifrado y otras situaciones parecidas. Estas amenazas pueden ser tratadas utilizando cifrado y protegiendo la integridad de los datos en tránsito, analizando la protección de los datos en diseño y ejecución, manejo correcto del almacenamiento y administración proactiva del sistema en general.

- **Secuestro de cuenta o servicio**. Los atacantes pueden robar credenciales y ganar acceso a áreas críticas de servicios desplegados en una nube, esto pasa a través de técnicas que se enfocan en hacer que los usuarios entreguen su información y también por vulnerabilidades de seguridad en el software, resultando en un compromiso de la confidencialidad, integridad y disponibilidad de los servicios. Para mitigar los riesgos mencionados, no debe estar permitida la compartición de credenciales entre usuarios o servicios, se deben usar técnicas de autenticación de múltiples factores donde sea posible. Para mantener un control y registro de la actividad en el sistema, es recomendable una monitorización fuerte. Finalmente un entendimiento completo de las políticas de seguridad y del acuerdo de nivel de servicio es necesario para tratar eventualidades de este tipo.

- **Compromiso de interfaces de administración**. La interfaz de administración de los consumidores en la nube es accesible a través de Internet. En ambientes de computación en la nube, un mayor número

de recursos son accedidos utilizando estas interfaces que en sistemas tradicionales. Esto puede ser un problema serio si existen vulnerabilidades en el navegador web. Para tratar este punto, se deben usar protocolos seguros para proveer el acceso, también se deben analizar y corregir las vulnerabilidades del navegador antes de permitir el acceso remoto.

- **Riesgos de cumplimiento**. Los proveedores no necesariamente cumplen con los acuerdos de servicio y no hay forma de comprobarlo. Debido a la falta de control sobre las auditorías, los consumidores de servicios de nube no tienen una vista de los procesos, procedimientos y prácticas del proveedor en las áreas de acceso, manejo de identidad y segregación de responsabilidades. Las organizaciones que buscan obtener un certificado se pueden poner en riesgo porque los proveedores de servicio no necesariamente pueden proveer evidencia del cumplimiento o puede que no se permita la auditoría por parte del consumidor. Para evitar problemas en de este tipo, el proceso interno de auditoría debe ser revisado. Debe quedar claro qué tan seguido el sistema será auditado por agencias externas y si estará abierto o no a auditoría de cumplimiento.

- **Ataques internos**. Este riesgo es muy conocido, una cosa son los ataques por entes ajenos al sistema, pero los ataques por parte de personas cercanas a las organizaciones tienen consecuencias considerables. Son particularmente peligrosos porque los atacantes tienen algún nivel de acceso garantizado, conocimiento del sistema y un punto de partida para casi cualquier tipo de ataque. Normalmente no intentan perjudicar el funcionamiento del sistema, sino aprovechar el acceso para filtrar información que no debería ser pública. Entre lo que se puede hacer es utilizar una jerarquía de usuarios con acceso privilegiado adecuada al sistema, de manera que cada uno sepa lo que necesite para hacer su trabajo, lo demás es directamente con las personas involucradas.

- **Pérdida de control**. En la infraestructura de una nube, es una situación que está propuesta desde la perspectiva del consumidor. Este cede necesariamente el control al proveedor en varios aspectos que pueden afectar su seguridad. El acuerdo de nivel de servicio podría no ofrecer el compromiso para garantizar la seguridad. Esto puede llevar a una falta

de confidencialidad, integridad y disponibilidad de los datos. Es un problema muy complejo como para tener soluciones genéricas o estándar por ahora. Cada organización a la que se presenta este problema, debe aplicar esfuerzos permanentes para cumplir los acuerdos de nivel de servicio.

- **Protección de los datos**. Es uno de los puntos más importantes, tiene riesgos tanto para consumidores como para proveedores. Puede ser difícil para un consumidor verificar efectivamente las prácticas en cuanto a manejo de datos de su proveedor y, por supuesto, el proveedor no debe revelar información importante en cuanto a seguridad en el manejo de los datos, aunque algunos entregan cierta información a sus clientes.

Para algunos de estos problemas se ha encontrado una solución satisfactoria, y es posible que funcionen en distintos escenarios. Muchos de los problemas que se habían encontrado en otros modelos o sistemas, sobre todo los que presentan las mayores dificultades, deben reconsiderarse para adaptar las soluciones a los requerimientos particulares de una nube y los volúmenes que plantea en cuanto a transferencia y almacenamiento de datos, poder de procesamiento, consumo de energía, números manejables de usuarios, etc.

En algunos casos, para resolver un riesgo de seguridad en un sistema, se eliminan o limitan características extras del mismo sistema. Es posible que siendo más estricto en cuanto a políticas y mecanismos, se consiga minimizar los riesgos. Por ejemplo, el firewall de un data center dedicado para un servicio web puede bloquear todos los puertos que no serán utilizados para brindar ese servicio. En una nube se podría aplicar a cada aparato de cómputo de cada consumidor, se complica cuando se lleva a todo el sistema porque es necesario mantener la coherencia. Se deben satisfacer las necesidades de todos los consumidores y es necesario un comportamiento dinámico, que es ideal para soportar los requerimientos de una nube. Además se incluyen los aspectos de la seguridad del proveedor. Muchos datos estarían en movimiento fluido por distintas partes físicas y virtuales de una nube. Esto trae problemas particulares en cuanto a seguridad.

2.2.2. Seguridad de los datos

Tomar información y hacerla segura, es decir, que sólo pueda ser vista y modificada por un grupo de usuarios, es algo en lo que se ha estado trabajando desde hace tiempo y es un problema particular. En una organización típica, la necesidad de seguridad de los datos varía desde la información que está en el dominio público, la información que necesita alguna protección (como control de acceso), hasta la información extremadamente importante, con consecuencias catastróficas si se filtra pero que igualmente debe ser accedida y utilizada por un conjunto selecto de usuarios.

Los datos en un data center de propósito específico se almacenan en lugares estáticos. Es suficiente con aplicar medidas estándares, tomando en cuenta parámetros como la localidad de los datos y los puntos de acceso a los mismos para garantizar ciertos niveles de seguridad. En estos casos, la seguridad de los datos está basada en el acceso a los mismos. En una nube, estos parámetros no son estáticos y pueden cambiar frecuentemente. Los datos pueden recorrer distintas áreas de todo el sistema, son accesibles desde múltiples nodos y localidades geográficas. La información, en un ambiente de computación en la nube, tiene mucha más fluidez y dinamismo. Al asegurarla se debe tener en cuenta que éstas características no pueden ser comprometidas.

Un escenario de mucho riesgo tiene que ver con el almacenamiento de los datos. Para un usuario que está subiendo datos a la nube o creándolos en la misma, se deben proteger estos datos en la subida para evitar que sean interceptados o modificados en el camino. Es necesario también proteger los datos mientras están en la nube y no han sido almacenados y, finalmente, se necesita controlar el acceso a esos datos almacenados en la nube. El control de acceso se debe aplicar a todos los entes humanos involucrados, al proveedor de la infraestructura de la nube, al proveedor del servicio de almacenamiento, que puede no ser el mismo proveedor de infraestructura y, además, a los otros usuarios de la nube. Los riesgos en cuanto a la seguridad de los datos en la nube están agravados por la naturaleza abierta de la misma. El control de acceso es un problema fundamental en ambientes basados en la nube por la amplia accesibilidad que deben tener los datos.

La privacidad es un problema que acompaña a muchos otros en la computación en la nube. Hay que proteger mucha información para el funcionamiento correcto de una nube. Muchas organizaciones no se sienten cómodas almacenando sus datos y aplicaciones en sistemas que residen fuera de sus

límites, este puede ser el miedo principal de los consumidores de servicios en la nube. Al migrar cargas de trabajo a infraestructuras compartidas, la información privada de los clientes enfrenta un riesgo mayor de acceso no autorizado, los proveedores de servicios en la nube deben asegurar los datos de sus clientes y proveer un alto nivel de transparencia en sus operaciones, es recomendable que existan mecanismos de protección de seguridad embebidos en todas las soluciones de seguridad. En un tema relacionado, cada vez se hace más importante saber quién creó algún dato, quién lo modificó, cómo, etc. La información de proveniencia puede ser utilizada para diversos propósitos como control de acceso basado en historial. El balance entre la proveniencia de los datos y la privacidad es un reto importante en nubes que no tienen perímetros físicos.

Muchos de los datos de usuario en una nube son accedidos casi permanentemente. Los datos que están almacenados necesitan ser utilizados, pueden ser accedidos y modificados por distintos usuarios, eliminados, y enviados a distintos lugares, también pueden ser almacenados por algún servicio de almacenamiento en otro data center de la misma nube, o en otra nube, y como pasa con otros modelos distinto al de la nube, los datos están en movimiento constantemente con paradas intermitentes, en pocos casos son realmente estáticos. Si pensamos en los datos como entidades, sin barreras de red y que puedan ser accedidos por múltiples usuarios de manera distribuida, entonces podemos comenzar a ver un modelo de seguridad basado en los datos. No estaría basado principalmente en el lugar donde se almacenan ni en los usuarios que los acceden, aunque estos siguen siendo parámetros importantes. Si la seguridad se convierte en una característica inherente a los datos, es menos importante dónde residen. Esto hace posible aplicar un control de acceso en toda la nube, a los proveedores y consumidores, y a todos los involucrados fuera de la nube. Es una idea que comenzó recientemente como un intento de romper las barreras entre una organización y su entorno, en cuanto a comunicación de información. De cualquier forma, recaer en un modelo nuevo para garantizar la seguridad de un sistema, no es una opción viable. Por ahora, la seguridad de los datos en sistemas de computación en la nube queda en manos de la experiencia con sistemas reales.

2.2.3. Niveles de abstracción

El ambiente de computación en la nube tiene tres niveles de abstracción [6].

- El **proveedor de infraestructura**. Administra la infraestructura de red y los recursos, incluyendo el hardware y el software del sistema.
- El **proveedor de servicio**. Ofrece servicios como computación por demanda, procesamiento de datos, servicios de software y plataformas para el desarrollo de aplicaciones.
- El **consumidor**. Hay dos grandes categorías, (a) *desarrollador*, quien aprovecha las ventajas de la infraestructura del hardware y las plataformas de software para construir aplicaciones. (b) *usuarios finales* que utilizan los servicios y aplicaciones disponibles.

Con respecto a la seguridad de los datos o información, los usuarios en diferentes niveles tienen distintas espectativas y preocupaciones debido al rol que toman en el ciclo de vida de los datos.

Desde la perspectiva de los consumidores, normalmente los dueños o la fuente de los datos, las preocupaciones son levantadas por la pérdida de control sobre los datos cuando están en una nube. Recordando que lo importante es la posibilidad de que sean tratados póbremente por el proveedor de servicio o de infraestructura, con o sin intención. El proveedor de infraestructura tiene acceso a todos los aparatos de cómputo de sus clientes, que usualmente son proveedores de algún servicio. Los proveedores de servicio controlan los datos de los usuarios en la nube. Además, un tercero puede atacar a cualquiera de los tres. Es una cadena de riesgos entre todos los actores. Como los datos pueden ser almacenados en una infraestructura desconocida de un tercero, el dueño de los datos pierde algunas ventajas que tendría en otros modelos. La incertidumbre sobre las privacidad, o la duda sobre las vulnerabilidades que supone la entrega de los datos a otro, es la consecuencia de la cadena de riesgos mencionada.

Los riesgos principales del usuario final incluyen confidencialidad, pérdida de datos y los perfiles de seguridad desconocidos de los proveedores. Los datos son transmitidos entre la máquina local y el proveedor para distintas operaciones, además son almacenados permanentemente en la infraestructura del proveedor. Durante este procedimiento, los datos pueden no ser adecuadamente protegidos mientras se mueven entre diferentes sitios del sistema del proveedor.

El problema se hace más complicado cuando el proveedor de infraestructura y el de servicio no son el mismo, esto implica más enlaces de comunicación.

Involucrar un tercero en los servicios introduce otro vector de ataque. En la práctica hay escenarios con más riesgos, en el caso de que múltiples usuarios finales tengan distintos requerimientos de seguridad utilizando el mismo servicio ofrecido por un proveedor de servicio individual. Es un escenario muy complejo para el proveedor de servicio, que además debe tener un proveedor de infraestructura capaz de soportar múltiples niveles de requerimientos de seguridad.

Desde la perspectiva de los proveedores de servicio, el riesgo principal al proteger los datos de usuario es al momento de la transferencia en la que los datos del usuario se reciben y los movimientos de los datos para almacenarlos en la nube. Los datos se almacenan en múltiples máquinas del proveedor de servicio, en dispositivos que pertenecen al proveedor de infraestructura. El proveedor de servicio necesita asegurar a sus usuarios que los datos están manejados adecuadamente entre las partes, que sus ambientes virtuales están aislados con suficiente protección, y que la limpieza de imágenes desactualizadas es manejada correctamente en las máquinas del proveedor de servicio y el de infraestructura.

Los problemas en los tres niveles tienen la misma importancia. El proveedor de infraestructura sabe que sólo un punto de falla en los mecanismos de seguridad de su infraestructura puede permitir que mucha información de sus clientes sea obtenida, afectando probablemente a otros proveedores.

2.3. Implementaciones de nube

Desde un punto de vista general, es difícil decir qué necesita una nube para funcionar. Se sabe que es posible implementar las mismas funcionalidades de distintas formas, y para lograr que se cumplan las características esenciales de la definición del NIST, lo que hace falta es un esquema en el que se organicen los elementos de manera que cumplan con los requerimientos funcionales de un data center convencional, pero aprovechando las ventajas de una nube. Lo común es que se encuentre una solución de virtualización especial que controla el hardware, sobre la que se instalan las funciones y los servicios necesarios, formando una plataforma de nube. Sobre esta plataforma está lo que se conoce como sistema de nube, todo lo necesario para hacer la entrega de los recursos de hardware y software como un servicio a través de una red.

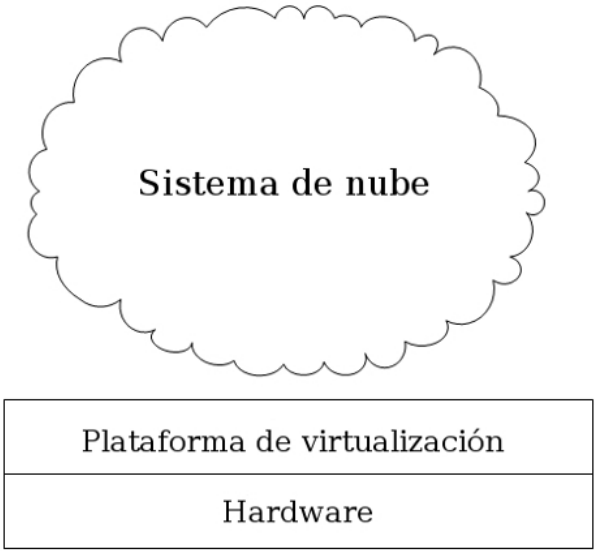

Figura 2.2: Esquema básico de implementación de nube

En la Figura 2.2 se muestra el esquema básico de una implementación de nube actual, en la que el hardware está controlado por una plataforma de virtualización especial y, sobre esta, está instalado el sistema de nube.

En la presente sección se reseñan las plataformas de virtualización y sistemas de nube (ver Figura 2.2); mostrando algunas de sus características.

2.3.1. Plataformas

Actualmente existen varios manejadores de máquinas virtuales con características especiales, que los hacen ideales para su uso en nubes, VMware vSphere, QEMU (Quick EMUlator), Citrix XenServer y Xen Cloud Platform (XCP) son algunos de ellos. Estos son la base sobre las que se construyen los ambientes de nube, proveen de las funcionalidades básicas necesarias para el funcionamiento de un sistema de nube; son a un sistema de nube lo que un kernel es a un sistema operativo. Una plataforma de este tipo se usa para construir nubes de modelo IaaS, en la que la infraestructura virtualizada es transformada en un ambiente elástico de nube, permitiendo la entrega de

recursos virtuales por demanda.

En esta sección se detallan algunas plataformas de nube actuales.

Xen Cloud Platform

XCP [17] es una solución de virtualización que provee de funcionalidades particulares de computación en la nube. Incluye el Xen Hypervisor, su monitor de máquina virtual, el Xen API Toolstack, y tiene pre-integradas funcionalidades de red y disco como Open vSwitch. El Xen API Toolstack es una pila de herramientas con un conjunto de funcionalidades como capacidad de manejar conjuntos de sistemas anfitrión, soporte para repositorios avanzados de almacenamiento, soporte para garantías de acuerdos de nivel de servicio, métricas detalladas de consumo y otras.

Se puede instalar de dos formas:

- **XCP ISO**. Muy parecido a XenServer, la distribución comercial de XCP de Citrix, está basado en una versión específica de Xen y un Kernel Dom0 CentOS 5 optimizado. Reemplaza todo el software instalado en la máquina, soporta la mayoría de las características de XenServer y la mayoría de los tipos de repositorios de almacenamiento.
- **Paquetes XCP-XAPI**. Son paquetes XCP para distribuciones Linux (actualmente para Debian y Ubuntu 12.04 LTS o más reciente) y se instala con el manejador de paquetes del sistema operativo. Queda ensamblado con el sistema operativo y utiliza sus componentes, se administra utilizando XAPI, soporta un subconjunto de las características de XenServer y algunos tipos de repositorios de almacenamiento, y es de fácil configuración.

Una de las diferencias principales entre Xen y XCP es el XCP Toolstack (XAPI) y la consola de línea de comandos XE (por defecto en XCP); Xen puede ser ejecutado utilizando su pila de herramientas por defecto, con Libvirt y con XAPI, y cuando se ejecuta con XAPI se llama XCP a la pila resultante.

XCP permite la integreación con diferentes pilas de coordinación de nube (Cloud Orchestration Stacks). Entre estas últimas está Apache CloudStack, un software de código abierto escrito en Java, diseñado para desplegar y administrar grandes redes de máquinas virtuales como una plataforma de

computación en la nube de alta escalabilidad y alta disponibilidad; CloudStack ofrece tres formas de manejar los ambientes de computación en la nube: una interfaz web fácil de usar, una línea de comandos y una API con todas las funciones. XCP también permite integración con OpenNebula, un proyecto enfocado en lograr una herramienta estándar para manejar la complejidad y heterogeneidad de infraestructuras de data centers distribuidos, y OpenStack, una colección de tecnologías que proporcionan un sistema operativo de nube altamente escalable.

Actualmente existen otras pilas de coordinación de nube, son soluciones que establecen el ambiente de computación en la nube en una organización, es decir, manejan los recursos de manera que se cumplan las características esenciales y permitan soportar servicios que serán prestados desde la nube.

Citrix XenServer

XenServer [18] es una plataforma de virtualización de servidores construida sobre el manejador de máquinas virtuales Xen, provee de lo necesario para crear y administrar una infraestructura virtualizada. Está disponible en una versión gratis límitada en funciones de administración y automatización, también las versiones Advance, Enterprise y Platinum. Está diseñado para ser escalable, soporta Windows y Linux, permite la administración centralizada de múltiples servidores, migración de máquinas virtuales en ejecución y otras funcionalidades.

Quick EMUlator

QEMU [**?**] (Emulador rápido) es una solución software libre para emular procesadores basado en la traducción dinámica de código binario. Provee un conjunto de modelos de aparatos de cómputo, permitiendo ejecutar una variedad de sistemas operativos huéspedes sin modificar.

VMware vSphere

VMware vSphere [20] es una plataforma de virtualización para construir infraestructuras de nube. Permite transformar la infraestructura de tecnologías

de información en una nube privada, entregándola como un servicio de fácil acceso.

Sus principales servicios son:

- **Cómputo**. Permite virtualizar recursos de servidores y agregarlos a conjuntos lógicos que pueden ser entregados a diferentes cargas de trabajo, con balanceo automático. Optimiza el consumo de energía y elimina el tiempo de inactividad por manenimiento, al permitir la migración de máquinas virtuales en ejecución.

- **Red**. Provee de servicios de red optimizados para ambientes virtualizados, así como una forma de administración simplificada, permite definir acceso a la red por prioridad y tiene aprovisionamiento, administración y monitorización centralizado.

- **Almacenamiento**. Los servicios de almacenamiento abstraen la complejidad del sistema de almacenamiento y están enfocados en hacer una utilización eficiente del almacenamiento virtualizado.

- **Seguridad**. Hace énfasis en la robustés y seguridad de la plataforma de virtualización completa.

- **Disponibilidad**. Incrementa la disponibilidad de aspectos de la infraestructura como recursos de almacenamiento y aplicaciones. Provee de alta disponibilidad en todo el ambiente virtualizado, disponibilidad continua para aplicaciones, asegurando que no se pierda ningún dato en caso de fallas en el servidor.

- **Automatización**. Los servicios de automatización proveen soluciones precisas y consistentes para ahorrar tiempo a administradores, incluye actualizaciones automáticas.

- **Administración**. Permite administrar centralmente con Operations Management, ensamblar flujos de trabajo utilizando una interfaz drag-and-drop y entrega datos de flujos de trabajo, monitorización y rendimiento.

2.3.2. Sistemas de nube

La idea básica de un sistema operativo de nube consiste en un software que se comporte como un sistema operativo pero sobre las características de una nube, permitiendo control sobre los recursos físicos comunes y los recursos virtuales especiales, y la ejecución de varias aplicaciones simultáneamente con seguridad; como boceto es viable, pero es una discusión para otro tema y no es la forma en la que se presentan estos sistemas actualmente.

En esta sección se muestra una reseña de algunos sistemas de nube disponibles, en la que se resumen sus componentes y características principales. Todos permiten el despliegue de nubes de modelo IaaS, en el que se permite el acceso a los recursos de nube, a través de máquinas virtuales.

VMWare vCloud Suite

VMware vCloud Suite [21] es una solución integrada para la creación y administración de una infraestructura de nube completa. Agrupa el hardware y ejecuta cada capa del data center como un servicio definido por software siguiendo la idea del data center definido por software, es decir, una plataforma de data center unificada que provee de automatización y flexibilidad, en la que los servicios de cómputo, almacenamiento, red, seguridad y disponibilidad son agrupados, agregados, prestados como un software, y son administrados basándose en políticas. Para cada conjunto de recursos virtuales (servidores, almacenamiento, red) provee de servicios configurables de seguridad, disponibilidad y administración.

Sus componentes son:

- ***vSphere***. Infraestructura virtualizada con automatización basada en políticas.
- ***vCloud Director***. Data centers virtualizados con extensibilidad para nubes públicas.
- ***vCloud Connector***. Vista integrada y transferencia dinámica de cargas de trabajo entre nubes privadas y públicas.
- ***vCloud Networking and Security***. Red y seguridad definidas por software, permite acceder a los todos los recursos de la nube.

- ***vCenter Site Recovery Manager***. Recuperación automatizada en caso de desastre.
- ***vCenter Operations Management Suite***. Administración integrada de rendimiento, capacidad y configuración para ambientes dinámicos de nube.
- ***vFabric Application Director***. Despliegue y aprovisionamiento de aplicaciones sobre nubes híbridas.
- ***vCloud Automation Center***. Despliegue de servicios con aprovisionamiento automático y basado en políticas.

vCloud Suite entrega el rendimiento correcto para todas las aplicaciones, incluyendo cargas de trabajo críticas y sensibles a operaciones de entrada/salida. Se encarga de balancear las cargas de trabajo, permite la monitorización de sistemas operativos Windows y Linux, así como para servidores de aplicaciones y tecnologías de base de datos. Todos los componentes trabajan en conjunto para proveer una plataforma de nube.

Eucalyptus Cloud

La plataforma de nube de Eucalyptus [22] es un software de código abierto para construir nubes híbridas y privadas. Permite compatibilidad con Amazon Web Services (AWS), lo que significa que los usuarios pueden administrar instancias de Eucalyptus y de AWS, y pueden mover instancias entre las dos nubes, creando una nube híbrida. Agrupa la infraestructura virtualizada existente para crear recursos de nube para cómputo, red y almacenamiento. Aprovecha las ventajas de la virtualización para crear conjuntos elásticos que pueden escalar dinámicamente y está enfocada en nubes empresariales.

Eucalyptus está disponible en una versión software libre y otra sujeta a suscripción. Algunas de las características presentes en ambas versiones son las siguientes:

- Reportes mejorados de uso y cuentas.
- Administración simplificada de nube.
- Robustez mejorada de nube.

- Compatibilidad con la API de AWS.
- Coordinación y administración de recursos elásticos.
- Administración de acceso por grupo y rol de usuario.

Actualmente Euclyptus se encuentra en su tercera versión, esta puede ser configurada como un despliege de alta disponibilidad (HA - High Availability) para maximizar la confiabilidad de la nube, utilizando mecanismos para fallas y reparaciones. Eucalyptus ofrece flexibilidad en el manejo de múltiples formatos de imagen, permitiendo ejecutar distintas versiones de Windows y Linux en la nube. También permite la integración con dispositivos storage area network (SAN) que pueden ser configurados para aprovechar arreglos de almacenamiento, mejorando el rendimiento, y para permitir locación dinámica para lograr elasticidad en el almacenamiento. El manejo de identidad incluye capacidades para controlar conjuntos de recursos virtuales, utilizando mecanismos de control de acceso de grano fino y basados en rol para cada conjunto, además se puede administrar el uso de los recursos por usuario y por grupos de usuarios. El Eucalyptus Dashboard provee a los administradores de la nube de una consola gráfica para ejecutar tareas de administración, incluyendo la administración de recursos físicos y virtuales, y la configuración, aprovisionamiento y reportes de recursos virtuales de la nube. La plataforma provee de una colección de servicios web para que los desarrolladores construyan una interfaz personalizada de auto-servicio.

OpenStack

El proyecto OpenStack comenzó en el año 2010 como una iniciativa de software libre por parte de NASA (National Aeronautics and Space Administration - Agencia Espacial Norteamericana) y Rackspace Hosting, con la intención de brindar servicios de computación en la nube soportados por hardware estándar[29]. Desde el primer lanzamiento oficial, llamado Austin, decidieron mantener actualizaciones de software regulares, la versión actual se llama Havana y fue lanzada el 17 de Octubre de 2013. Desde el principio el proyecto tiene una estructura modular, asignando diferentes proyectos particulares para mantener el control sobre los recursos de cómputo, red y almacenamiento, y alcanzando una alta escalabilidad.

OpenStack se puede considerar como un sistema operativo desde el punto de vista del control sobre los recursos del sistema, estaría en un nivel más alto de abstracción que un sistema operativo común, los recursos que tiene que administrar están en forma de conjuntos (*pools*), y presta servicios a aplicaciones que requieren un acceso elástico a esos recursos. Un centro de datos actual que preste un conjunto de servicios dentro de una organización, como un servicio de correo electrónico o una página web, generalmente está formado por servidores independientes con usos diferentes, conectados en una topología de red que permite una comunicación segura. El centro de datos sería entonces el hardware sobre el que se despliega OpenStack, que integra sus componentes de manera modular y permite una administración centralizada, aportando ciertas ventajas frente al centro de datos simple. El hardware, que está distribuido, se integra lógicamente en lo que se conoce como una nube.

Los servicios de OpenStack permiten el acceso a los recursos de red, cómputo y almacenamiento en forma de máquinas virtuales, las cuales son desplegadas en redes definidas por los usuarios. Esto permite crear centro de datos virtuales con recursos elásticos.

Los módulos de OpenStack son proyectos particulares, cada uno enfocado en brindar un servicio para permitir la utilización correcta de los recursos del sistema. Todos los módulos pueden funcionar de manera independiente y los despliegues de OpenStack requieren una comunicación entre los nodos físicos en los que se ejecuta el software de OpenStack. Los recursos están organizados lógicamente en red, cómputo y almacenamiento, y los diferentes módulos trabajan en conjunto para garantizar el acceso seguro, ubicuo y por demanda a estos recursos. Los módulos interactúan entre ellos y con los usuarios del sistema a través de APIs comunes. A continuación se describen los módulos que actualmente soporta OpenStack:

- **Servicio de cómputo (Nova).** Este servicio está diseñado para administrar y automatizar conjuntos (*pools*) de recursos de cómputo. Puede trabajar con distintos manejadores de máquinas virtuales y su arquitectura le permite escalar en aparatos de hardware estándar. Nova participa en la creación y control de máquinas virtuales, específicamente brinda el soporte necesario para la gestión de instancias de máquinas virtuales.
- **Servicio de almacenamiento de objetos (Swift).** Es un sistema

de almacenamiento redundante, el cual permite la administración de *clusters* de almacenamiento escalables.

- **Servicio de almacenamiento por bloques (Cinder).** Este servicio provee de almacenamiento persistente a las instancias del servicio Nova, es por esto que este participa en la creación de aparatos de almacenamiento por bloques que pueden ser agregados y retirados de las instancias.

- **Servicio de imágenes (Glance).** Este servicio provee de descubrimiento, registro y entrega de imágenes de disco. Las imágenes almacenadas pueden ser utilizadas como plantillas, y el servicio permite utilizar diferentes tecnologías para el almacenamiento de las imágenes, incluyendo Swift.

- **Servicio de red (Neutron).** Es un sistema para administrar redes y direccionamiento IP en infraestructuras de nube. En los despliegues de OpenStack, se encarga de que la red no sea el factor limitante de la nube. El servicio provee a usuarios y administradores de autoservicio sobre las configuraciones de red, permite asignar direcciones IP flotantes a las instancias del servicio Nova y administra los flujos de datos entre las redes del sistema.

- **Servicio de medidas (Ceilometer).** Este servicio provee de un punto único de contacto para sistemas de facturación, provee los contadores necesarios para establecer la facturación de los consumidores, entre todos los componentes de OpenStack.

- **Servicio de Coordinación (Heat).** Es un servicio que permite utilizar plantillas para desplegar infraestructuras virtuales.

- **Servicio de acceso web (Horizon).** Provee a usuarios y administradores de una interfaz gráfica para acceder, aprovisionarse y automatizar los recursos del sistema.

- **Servicio de identidad (Keystone).** Provee de un directorio central de usuarios vinculados con los servicios de OpenStack a los que tienen acceso. Actúa como un sistema común de autenticación y puede integrarse con otros directorios como Lightweight Directory Access Protocol (LDAP).

OpenNebula

OpenNebula [24] es una conjunto de herramientas para administrar infraestructuras de data centers distribuidas y heterogéneas. Manipula la infraesturctura virtual del data center para construir una infraestructura de nube privada, pública o híbrida. OpenNebula coordina tecnologías de almacenamiento, red, virtualización, monitorización y seguridad para desplegar servicios como máquinas virtuales en infraestructuras distribuidas, combinando los recursos del data center con los de una nube remota. El conjunto de herramientas incluye funcionalidades para integración, administración, escalabilidad, seguridad y cuentas de usuario. Está enfocado en la estandarización, interoperabilidad y portabilidad, permitiendo a usuarios y administradores el uso de diferentes interfaces e hipervisores, y una arquitectura flexible que puede funcionar con diferentes combinaciones de hardware y software en el mismo data center.

OpenNebula provee de una plataforma escalable y segura. El sistema de almacenamiento permite almacenar imágenes de discos que pueden ser utilizadas para definir máquinas virtuales o compartirlos con otros usuarios. El sistema de repositorio de plantillas permite registrar definiciones de máquinas virtuales para ser instanciadas luego. La red virtualizada está provista para interconectar máquinas virtuales, pueden ser definidas como fijas o por rango. Una vez que se instancie una plantilla en una máquina virtual, se le puede aplicar un conjunto de operaciones como migración en caliente, detener, resumir, cancelar, etcétera.

Los componentes principales de OpenNebula son:

- **Interfaces y APIs**. OpenNebula provee de diferentes interfaces que pueden ser utilizadas para interactuar con las funcionalidades ofrecidas para manejar recursos físicos y virtuales. Para administrar las instancias hay dos formas principales, la interfaz de línea de comandos y la interfaz gráfica de usuario (GUI) Sunstone. Además tiene APIs de integración para permitir el desarrollo de nuevos componentes.
- **Usuarios y Grupos**. OpenNebula soporta cuentas de usuario y grupos así como varios métodos de autenticación y autorización. Esta última característica puede ser utilizada para crear compartimientos aislados en la misma nube. También tiene un mecanismo de lista de control de acceso para permitir manejo de roles.

- **Servidores**. Se soportan varios hipervisores, con la habilidad de controlar el ciclo de vida de la máquina virtual, y permite la monitorización de máquinas virtuales y servidores físicos.

- **Red**. OpenNebula presenta un subsistema de red personalizable y fácilmente adaptable, para integrar mejor con los requerimientos particulares del data center.

- **Almacenamiento**. OpenNebula es lo suficientemente flexible para soportar varias configuraciones de almacenamiento de imágenes diferentes.

- **Clusters**. Los clusters son conjuntos de servidores que comparten almacenes de datos y redes virtuales, y son utilizados para balanceo de carga, alta disponibilidad y computación de alto rendimiento.

2.3.3. Comparación de características

En esta sección se presenta una tabla de comparación de características de plataformas de nube reseñadas anteriormente, que se muestran en el Cuadro 2.1. Las herramientas de conversión intervienen en el proceso de migración de sistemas operativos, aplicaciones y datos, del servidor físico a la máquina virtual huésped en la plataforma virtualizada. Los conjuntos (pools) heterogéneos hacen referencia a la habilidad de la plataforma de crear y administrar conjuntos virtuales de diferentes recursos físicos, por ejemplo, un conjunto de procesamiento puede estar soportado por procesadores físicos diferentes. Las alertas y reportes de rendimiento permiten automatizar las respuestas ante diferentes eventos. La alta disponibilidad especifica si el sistema tiene un modo de ejecución que garantice alta disponibilidad de recursos. La protección de máquinas virtuales garantiza que no haya ninguna intervención en la ejecución de la máquina virtual. La recuperación de máquinas virtuales es la posibilidad de devolver la máquina virtual a un estado correcto en caso de una falla en el servidor. La administración basada en rol le da a cada usuario los privilegios mínimos que necesita para trabajar. El balanceo dinámico de cargas de trabajo permite la máxima utilización de los recursos, asignando automáticamente conjuntos de recursos a máquinas virtuales que los necesitan.

Característica	**XCP**	**vSphere**	**XenServer**	**QEMU**
Código abierto	Si	No	No	Si
Herramientas de conversión	Si	Si	Si	No
Conjuntos heterogéneos	Si	No especificado	Si	Si
Alertas y reportes de rendimiento	Si	Si	Si	No
Alta disponibilidad	No	Si	Si	No
Protección de MV	Si	Si	Si	Si
Recuperación de MV	Si	Si	Si	No
Administración basada en rol	Si	No especificado	Si	No
Balanceo dinámico de cargas de trabajo	No	Si	Si	No

Cuadro 2.1: Características de plataformas de nube

El Cuadro 2.2 muestra las principales características encontradas en los sistemas de nube comentados anteriormente. El despliegue de nube hace referencia al enfoque que tiene el sistema frente a los modelos de despliegue de la definición del NIST. El soporte para sistemas operativos muestra los sistemas operativos soportados como componentes dentro de la nube, agrupa administración y monitorización. El lenguaje de programación muestra los principales lenguajes utilizados en la implementación de los sistemas. La consola web para autoservicio es un portal que muestra diferentes opciones, a usuarios y administradores, para solicitar recursos en la nube. El repositorio de imágenes provee de servicios relacionados con imágenes de disco, como almacenamiento y permitir el uso de imágenes como plantillas para máquinas virtuales. La administración de máquinas virtuales en ejecución incluye operaciones sobre máquinas virtuales como migración, mientras está siendo ejecutada en la nube. La compatibilidad con la API de Amazon Cloud Service indica si el sistema está diseñado para ser compatible con la API de AWS. El soporte para alta disponibilidad indica si el sistema provee de un modo de ejecución en el que se garantice alta disponibilidad de los servicios en la nube.

Característica	**vCloud Suite**	**Eucalyptus**	**OpenStack**	**OpenNebula**
Código abierto	No	Si	Si	Si
Despliegue de nube	Híbrida y Privada	Híbrida y Privada	Pública y Privada	Híbrida, Pública y Privada
Virtualización	VMware	VMWare, Xen y QEMU	Xen y QEMU	VMware, Xen y QEMU
Soporte para SO	Linux y Windows	Linux y Windows	Linux y Windows	Windows y Linux
Estructura	Modular	Modular	Modular	Modular
Lenguaje de programación utilizado	No especificado	Java, C	Python	C++, C, Ruby, Java
Consola web para autoservicio	Si	Si	Si (Horizon)	Si
Repositorio de imágenes	Si	Si	Si (Glance)	Si
Administración de MV en ejecución	No especificado	No especificado	Si	Si
Compatibilidad con API de AWS	No especificado	Si	Si	Si
Soporte para alta disponibilidad	Si	Si	Si	Si
Recuperación en caso de desastre	Si	No especificado	Si	Si
Control de acceso	No especificado	Rol	Rol y Grupos	Rol y Grupos

Cuadro 2.2: Comparación de Características de Sistemas de Nube.

Basado en las características que se muestran en el Cuadro 2.2, se escogió el sistema de nube OpenStack como caso de estudio, principalmente por ser de código abierto y por ser soportado por una gran comunidad de desarrolladores. Además es un proyecto de mucha actividad, que está enfocado en la administración óptima de los recursos de hardware sin dejar de un lado las cosideraciones relacionadas con los usuarios, permitiendo administrar los recursos a través de una interfaz web fácil de usar, segura y que permite auditorías de las mediciones que muestra.

Capítulo 3

Método de investigación y herramientas utilizadas

En este capítulo se presenta el método de investigación que se siguió en el presente trabajo. Adicionalmente las herramientas de software y hardware utilizadas durante la investigación.

3.1. Método de análisis y síntesis

En este trabajo de investigación se explora el tema de la computación en la nube, es una tecnología en auge actualmente y representa un área de mucho interés en Tecnologías de la Información. El objetivo de la investigación es desplegar una nube para una organización pequeña, logrando un contacto directo con una tecnología nueva. El modelo de nube del NIST, describe el comportamiento esperado de un sistema de nube, que se puede resumir en garantizar el acceso conveniente, ubicuo, seguro y por demanda a conjuntos configurables de recursos de cómputo. Los conjuntos de recursos pueden ser implementados de distintas formas, lo importante es conseguir la elasticidad rápida, que es de las principales ofertas de la nube, y lograr altos niveles de escalabilidad. Con esto quedan identificadas dos partes de todo sistema de nube, por un lado los conjuntos (*pools*) de recursos, que deben estar disponibles para su utilización, y por otro lado la parte que se encarga de administrar estos recursos con el objetivo de hacerlos disponibles a los usuarios. Entonces

es necesario lograr una vista generalizada de varias unidades de recursos, por ejemplo, varios servidores de almacenamiento en un *pool* de almacenamiento, y de igual forma con recursos de cómputo y red, de manera que puedan ser asignados rápidamente y el sistema mantenga un control preciso sobre todos los recursos disponibles.

El método de investigación que se sigue en este trabajo es el de análisis y síntesis que consiste en la separación de las partes de un todo, para permitir un estudio individual, y la reunión racional de las partes en su totalidad. Los componentes que forman una nube están relacionados directamente con los recursos de hardware y su administración, se encuentran nodos de cómputo, de red y de almacenamiento, además de nodos de control. Los nodos de cómputo se encargan específicamente de proveer poder de cómputo a diferentes usuarios y aplicaciones, varios nodos de cómputo forman un *pool* de cómputo. Los nodos de red proveen de recursos de red a los diferentes nodos de cómputo, de manera que los recursos de red puedan ser elásticamente aprovisionados para garantizar un mejor rendimiento. Los nodos de almacenamiento proveen elasticidad de almacenamiento a nodos de cómputo o directamente a los usuarios de la nube. Todos estos nodos son coordinados por conjuntos de nodos de control, que mantienen el estado del sistema y ejecutan las operaciones administrativas.

Las principales implementaciones de nube aprovechan el esquema de la virtualización. Estas implementaciones prestan un servicio de modelo IaaS (Infraestructure as a Service), que permite el despliegue rápido de infraestructuras de hardware virtualizado con recursos elásticos. Todas las máquinas virtuales alojadas en la nube comparten la infraestructura física, pero están aisladas lógicamente. Cada una tiene acceso a diferentes recursos y pertenece a una red virtual configurable. Los recursos que tienen asignados como procesadores virtuales, diferentes tipos de almacenamiento, interfaces de red y otros, pueden ser aumentados o disminuidos bajo demanda; en algunos casos automáticamente y sin la necesidad de reiniciar la máquina virtual. El consumo de los recursos está siendo registrado permanentemente por el sistema de nube, permitiendo auditorías de facturación por parte de los consumidores y monitorización de valores de rendimiento en tiempo real.

Cuando se tiene el control sobre el diseño de un sistema, se busca mantener un orden lógico que permita el funcionamiento correcto de los componentes del mismo, y que facilite el cumplimiento de sus objetivos. Los conceptos o abs-

tracciones permiten agrupar y organizar los diferentes contextos en los que se desarrollan las actividades involucradas en el funcionamiento de un sistema; sirven también para imponer un orden coherente que alcance los límites del sistema, ubicando cada uno de los participantes precisamente donde son necesarios. En una nube de infraestructura como un servicio debe haber formas de agrupar lógicamente grupos de recursos y usuarios, de manera que permitan aislamiento entre las diferentes infraestructuras que alojan. Un usuario típico puede crear instancias con cierta cantidad de recursos, crear redes y conectar las instancias a las redes que tenga disponible. Puede modificar la topología de red, crear routers, asignar direcciones IP a puertos o interfaces de red y modificar los grupos de seguridad de red. Es muy común que una nube ofrezca direcciones IP flotantes, esto para permitir la comunicación desdes Internet a las máquinas virtuales. Estas pertenecen a un rango de direcciones IP válido en una red externa a la red de la infraestructura virtual, y los usuarios las pueden crear y asignar a los puertos de las instancias. Una forma de control para el comportamiento de los usuarios, disponible para el administrador, es limitar los recursos que tienen disponibles los diferentes grupos de usuarios, esto incluye los límites en la utilización de los recursos y en la cantidad, puede limitar el número de redes y el número de reglas en los grupos de seguridad.

Los sistemas de nube integran el hardware que tienen disponible, siempre tomando en cuenta la escalabilidad, permitiendo agregar recursos de hardware fácilmente. La vista lógica de los recursos, en forma de *pools*, va a acompañada de una distribución física coherente en los despliegues de nube. La Figura 3.1 muestra la integración de los recursos de hardware en un centro de datos optimizado para una nube con varios *pools* de procesamiento, coordinados por un *pool* de nodos de control, garantizando alta disponibilidad de los módulos del sistema de nube y de los servicios que prestan.

Los componentes de una nube pueden ser desplegados de diferentes formas para satisfacer diferentes requerimientos. Cada uno cumple con un conjunto de tareas específicas y se comunica con los otros para permitir la función del sistema. La especificación de cada nodo será expuesta en el Capítulo 4.

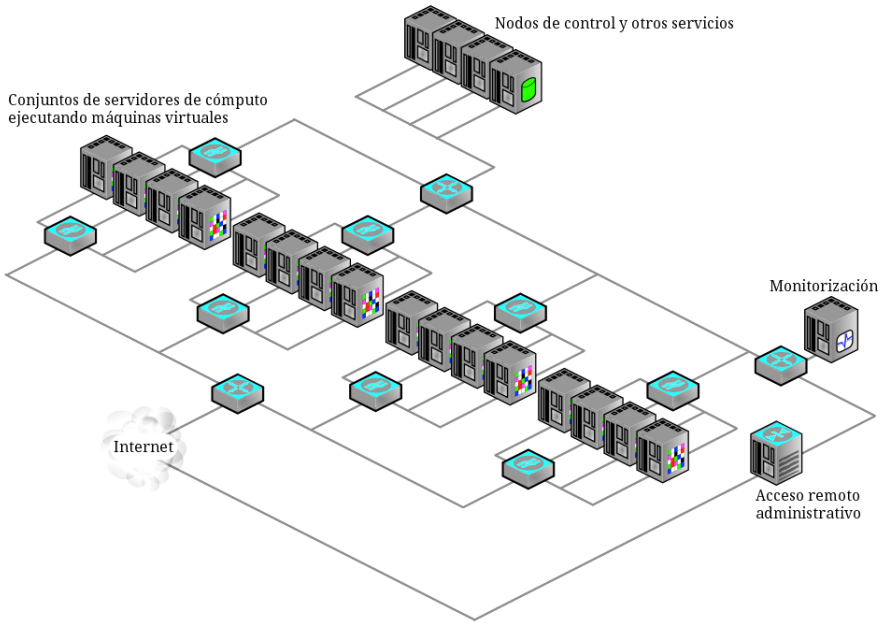

Figura 3.1: Despliegue lógico de un centro de datos de nube

3.2. Herramientas utilizadas

En esta sección se muestran las herramientas de hardware y software que fueron utilizadas durante la investigación. Las herramientas de hardware fueron seleccionadas de un conjunto de servidores disponibles en el Laboratorio ICARO, mientras que el principal criterio para seleccionar las de software es que son de código abierto y compatibles con el sistema de nube escogido.

El el Cuadro 3.1 se muestra un resumen de las tecnologías de software utilizadas en el despliegue del sistema. Havana es la octava versión de OpenStack, la más reciente para el momento en que se escogió. Ubuntu es el sistema operativo anfitrión, sobre el que se desplegaran los módulos de OpenStack en forma de diferentes procesos que se comunican para proveer las funcionalidades necesarias. Ubuntu Cloud Images es el repositorio para Ubuntu que permite descargar los paquetes necesarios. QEMU es el manejador de máquinas virtuales que interactúa con el módulo de OpenStack encargado de la administración de las máquinas virtuales. MySQL permite implementar la base de datos del sistema. RabbitMQ es la cola de mensajes que permite la comunicación entre los diferentes procesos de Ubuntu con los que se implementa OpenStack. Neutron es el módulo que permite proveer el servicio de red en el despliegue, y está desplegado en un nodo simple. ML2 es el plugin del servicio de red que permite la administración de los diferentes agentes. OpenVSwitch provee los agentes del servicio de red, que añaden diferentes funcionalidades al mismo servicio, como el agente L3, que permite crear y utilizar routers virtuales en las infraestructuras alojadas en la nube. SQL es el driver del módulo de identidad Keystone, que administra la autenticación de usuarios y aplicaciones. Las imágenes de sistema operativo disponibles en el sistema se almacenan en archivos. Y los bloques del módulo Cinder se implementan con volúmenes a través del Logical Volume Manager de Ubuntu y se conectan a las máquinas virtuales a través de Internet Small Computer System Interface.

Cada uno de los tres servidres físicos es un HP ProLiant DL380 G5. Cada uno con 4GB de memoria principal, cuatro tarjetas de 1GB (DDR2 667 MHz), y un procesador Intel(R) Xeon(R) CPU E5335 @ 2.00GHz. El nodo de control cuenta con dos discos de almacenamiento de 146 GB en RAID nivel 0, el nodo de red cuenta con un disco de 146 GB, y el de cómputo cuenta con un disco de almacenamiento de 72 GB. Las interfaces de red de los servidores (NetXtreme II BCM5708 Gigabit Ethernet, dos por cada servidor) están co-

Software	**Versión**	**Descripción**
Havana	2013.2	Release de OpenStack
Ubuntu	12.04.02 LTS	Sistema operativo anfitrión
Ubuntu Cloud Archive	Havana	Repositorio de OpenStack
QEMU	1.5.0	Manejador de máquinas virtuales
MySQL	14.14 Dist. 5.5.35	Base de datos del sistema
RabbitMQ	2.7.1	Cola de mensajes
Neutron	N/A	Servicio de red
Nodo simple	N/A	Despliegue del servicio de red
ML2	N/A	Plugin de servicio de red
OpenVSwitch	1.9.3	Agentes del servicio de red
SQL	N/A	Driver de Keystone
Archivos	N/A	Back-end de Glance
LVM/iSCSI	N/A	Back-end de Cinder

Cuadro 3.1: Características de software del despliegue

nectadas a un switch Avantech Switch de 24 Puertos Ethernet, perteneciente a la infraestructura de red del Laboratorio ICARO, y que permite un ancho de banda de 100 Mbps.

	Nodo		
Característica	**Control**	**Red**	**Cómputo**
Almacenamiento	2x146 GB	1x146 GB	1x72 GB
Memoria principal	4x1 GB (DDR2 667 MHz)		
Procesador	Intel(R) Xeon(R) CPU E5335 2.00GHz		
Interfaz de red	2 x NetXtreme II BCM5708 Gigabit Ethernet		

Cuadro 3.2: Características de hardware del despliegue

Capítulo 4

Diseño e implementación de la solución

En este capítulo se describe el diseño de la solución implantada, se muestra la instalación del sistema de nube OpenStack en el Laboratorio ICARO, también la descripción particular de la solución escogida, y finalmente la especificación de los escenarios de utilización.

La solución debe permitir la creación de máquinas virtuales con recursos elásticos en el Laboratorio ICARO, y se explorará la opción de una nube privada de modelo IaaS (Infrastructure as a Service). De los sistemas de nube referenciados anteriormente (Cuadro 2.2), se escogió OpenStack como caso de estudio, por ser de código abierto y por permitir el despliegue de nubes privadas. La versatilidad de esta tecnología podría resultar muy útil en un ambiente de investigación, es un recurso de poder de cómputo de alta escalabilidad y posiblemente una herramienta de difusión académica.

4.1. Despliegue de OpenStack

OpenStack es un sistema de nube completo, es decir, permite la creación, administración y utilización de conjuntos (*pools*) de recursos, y garantiza el acceso seguro y ubicuo a los mismos.

Hay diferentes formas de desplegar OpenStack, los módulos que lo componen exigen una comunicación específica entre ellos pero la instalación puede

ser diferente entre un despliegue y otro. Siguiendo principios básicos, los despliegues de OpenStack involucran una vista de los recursos en forma de *pools*, ubicanco los módulos sobre hardware destinado a procesamiento, almacenamiento o recursos de red. Generalmente el hardware de procesamiento está destinado a la ejecución de las máquinas virtuales de forma exclusiva. El servicio de red también puede ser desplegado en servidores para su uso exclusivo, un servidor de red puede proveer recursos virtuales de red a varios servidores de cómputo.

Cada módulo de OpenStack ejecuta operaciones específicas a su funcionamiento, y también ejecuta operaciones que se pueden considerar como de coordinación, las cuales permiten que los módulos compartan los recursos. Por ejemplo, antes de la ejecución de una instancia en un nodo de cómputo, el servicio Nova la planificó, llenó una entrada en la base de datos del sistema, y reunió los recursos necesarios para la ejecución de la máquina virtual. De manera que los despliegues incluyen *pools* físicos de recursos, para la ejecución de operaciones específicas de los módulos, idealmente servidores con optimizaciones para almacenamiento o poder de cómputo. Estos servidores son administrados por nodos de control, destinados a la ejecución de las operaciones de coordinación.

El despliegue escogido requiere tres nodos físicos para alojar los diferentes módulos de OpenStack, organizando los servicios en Control, Red y Cómputo. El nodo de control se encarga de las actividades de coordinación de todos los módulos de OpenStack, un solo nodo de control puede administrar varios nodos de red y de cómputo. En el nodo de red se alojan los recursos virtuales de red que utilizan las instancias del servicio Nova, las cuales se ejecutan en el nodo de cómputo. Las actividades de coordinación del servicio Nova se ejecutan en el nodo de control, sólo la porción del servicio relacionada directamente con la ejecución de máquinas virtuales es instalada en los nodos de cómputo.

Se escogió este despliegue, común en ambientes de producción, porque permite escalar fácilmente en recursos de cómputo y toma en cuenta que el manejo de recursos de red puede convertirse en un cuello de botella en la nube, dedicando un servidor exclusivo para el servicio de red.

En la Figura 4.1 se muestra la organización de los nodos en el despliegue de OpenStack escogido.

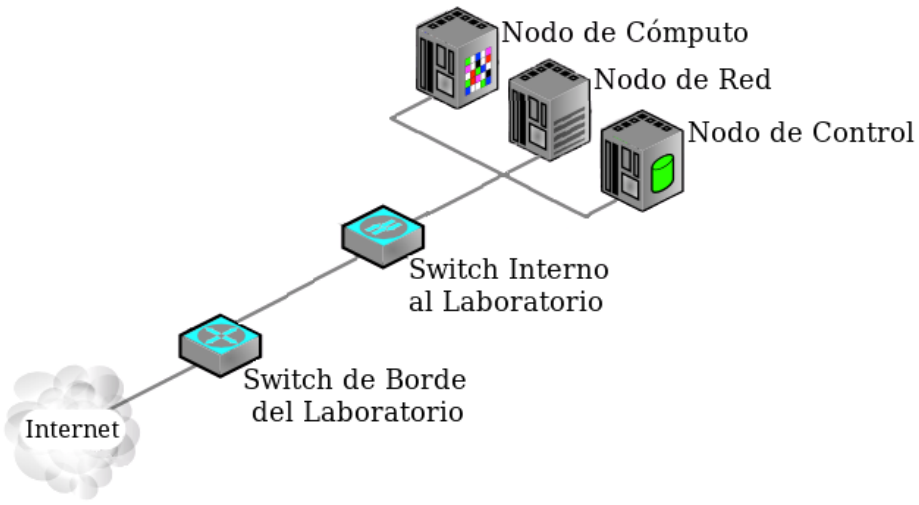

Figura 4.1: Despliegue lógico de OpenStack en el Laboratorio ICARO

4.2. Interacción intermodular

En sistemas de cómputo en general, los componentes de alto nivel brindan la posibilidad a usuarios típicos de interactuar con las funcionalidades provistas por el sistema. En particular, OpenStack ofrece una variedad de componentes de alto nivel destinados a este fin. Los módulos de OpenStack están diseñados para trabajar en conjunto sin importar la localidad física, todos comparten una forma de comunicación que permite una interacción fluida y segura. Un ejemplo interesante de esta comunicación es el de la creación de una instancia por parte de un usuario, esto involucra la interacción de varios componentes de OpenStack, y muestra el comportamiento general de los módulos. Los pasos que sigue un usuario en el ejemplo anterior son muy simples, primero accede a la interfaz web Dashboard con su nombre de usuario y contraseña, y en la vista general de su proyecto (*tenant*), selecciona lanzar una instancia con los parámetros necesarios (imagen de sistema operativo, recursos de hardware virtual, red y nombre). La interfaz web muestra los estados de la máquina virtual hasta que está creada y disponible.

La Figura 4.2 describe la comunicación e interacción de los componentes de OpenStack, durante el proceso de creación de una instancia de maquina

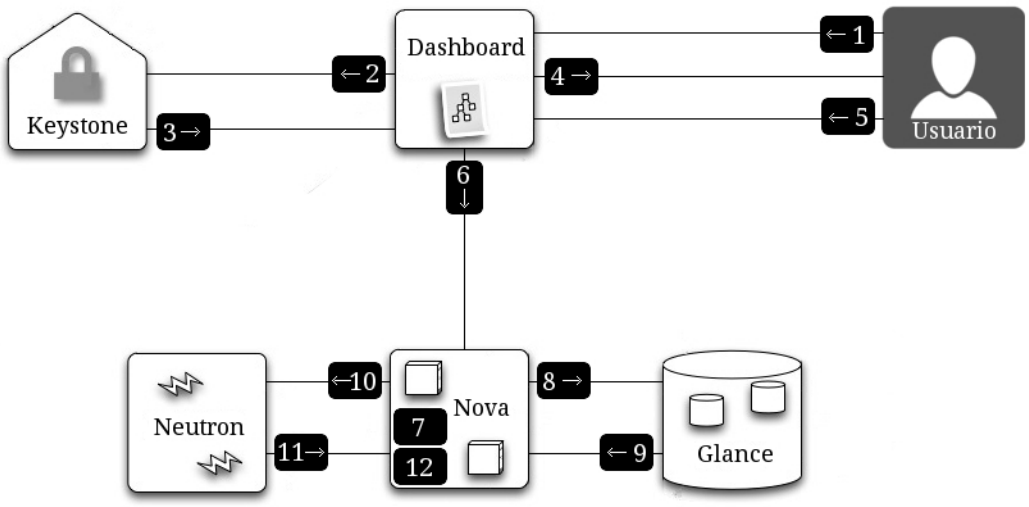

Figura 4.2: Comunicación de los componentes de OpenStack para el lanzamiento de una instancia

virtual por parte de un usuario típico, llamado también lanzar o aprovisionar una instancia. A continuación se describen detalladamente las iteraciones necesarias entre cada módulos involucrados.

1. El usuario ingresa su nombre y contraseña en el Dashboard.

2. Dashboard comunica los datos de usuario a Keystone, pidiendo una verificación.

3. Keystone compara los datos contra la base de datos del sistema y responde afirmativamente a Dashboard, comunicando metadatos del usuario, como el proyecto al que pertenece y su rol.

4. Dashboard permite el acceso a la interfaz web, con los límites indicados por Keystone.

5. El usuario autenticado selecciona lanzar una instancia de máquina virtual y asigna sus parámetros(imagen y redes en este caso).

6. Dashboard traduce la orden para comunicarla al API del servicio Nova.

7. Nova recibe la orden, verifica sus recursos disponibles y planifica la ejecución de la máquina virtual.

8. Nova comunica la petición de la imagen especificada a Glance.

9. Glance envía la imagen al servicio Nova.

10. Nova comunica la petición de configuración de red para la nueva máquina virtual al servicio Neutron.

11. Neutron envía la información de red para la instancia.

12. Nova crea la instancia, a través del manejador de máquinas virtuales del sistema y actualiza la base de datos del sistema.

El servicio Dashboard actualiza automáticamente la vista del sistema, el usuario tiene acceso a la información en tiempo real, además sólo muestra las opciones correspondientes a los límites del rol y proyecto del usuario, especificado por Keystone y configurado por el administrador. Los pasos que se siguen en el ejemplo anterior son una forma general de ver el caso del lanzamiento de una instancia, cada uno de los módulos ejecuta varias operaciones adicionales que se muestran más adelante. En el momento de la creación de una instancia se pueden aplicar otras opciones de configuración como asignar un volumen, grupos de seguridad de red, una llave privada para acceso remoto seguro, o incluir un conjunto de comandos para que se ejecuten al iniciar la máquina virtual. Nova se encarga de ejecutar todas las operaciones sobre las máquinas virtuales, es por esto que su funcionamiento requiere la comunicación con los otros módulos.

El desarrollo de OpenStack comenzó con la versión Austin, que contaba con un núcleo de cómputo, soporte para almacenamiento de objetos y un portal de control web. En la evolución de este desarrollo se han separado funcionalidades del núcleo de cómputo, haciéndolas independientes, y se han añadido otros módulos de soporte. El núcleo de cómputo se mantiene en el servicio Nova, algunos de los otros servicios están presentes para darle algún soporte a las instancias de Nova (conectividad de red, acceso a almacenamiento elástico y auditorías de consumo de recursos). Los otros servicios están enfocados al usuario, ofrecen una utilización controlada de los recursos, acceso a los recursos por interfaz gráfica web, y la posibilidad de desplegar infraestructuras virtualizadas utilizando plantillas.

4.3. Descripción del despliegue implantado

El despliegue escogido puede ser implementado utilizando diferentes herramientas, en esta sección se describe el despliegue de OpenStack implantado.

La Figura 4.3 muestra un diagrama físico del despliegue de OpenStack en el Laboratorio ICARO. Todos los nodos están ejecutando Ubuntu Server, y sobre Ubuntu se instalan los módulos de OpenStack en forma de distintos procesos que se comunican para proveer los servicios. El **nodo de control** provee de las funcionalidades de la nube excepto por el alojamiento de máquinas virtuales y el servicio de red; en este nodo se aloja el servicio de imágenes de OpenStack (Glance), el servicio de identidad (Keystone) y el servicio de acceso web para usuarios (Dashboard), también ejecuta porciones del servicio de cómputo (Nova) como el planificador, que escoge los nodos de cómputo en los que serán ejecutadas las máquinas virtuales, y el conductor, para comunicación con la base de datos que mantiene el estado del sistema. El **nodo de red** provee del conjunto de servicios de red de OpenStack como DHCP, conmutación (capa 2), enrutamiento (capa 3) y direccionamiento a través de direcciones IPs flotantes. El **nodo de cómputo** aloja las máquinas virtuales de la infraestructura de nube utilizando un manejador de máquinas virtuales compatible con OpenStack, QEMU en este caso, también ejecuta el agente del servicio de red para permitir la conectividad de las máquinas virtuales a las redes de la infraestructura interna.

El servicio de red de OpenStack puede desplegar redes virtuales de distintas formas. Hay diferentes esquemas de despliegue que permiten posibilidades que van desde sólo una red privada por proyecto, hasta varios routers y redes privadas por proyecto. El despliegue de red de la solución sigue el caso de uso "Router privado con redes privadas"que se especifica en el manual de administración de red de OpenStack [30] como "Per-tenant router with private networks". En este caso provee a cada proyecto con la opción de crear routers, que pueden tener como puerta de enlace predeterminada la red externa del Laboratorio. Los usuarios pueden crear redes privadas, y conectarlas a través del router virtual, que es implementado con OpenVSwitch. Este modelo soporta la asignación de direcciones IP flotantes, dando acceso desde Internet bajo demanda a las instancias del servicio Nova.

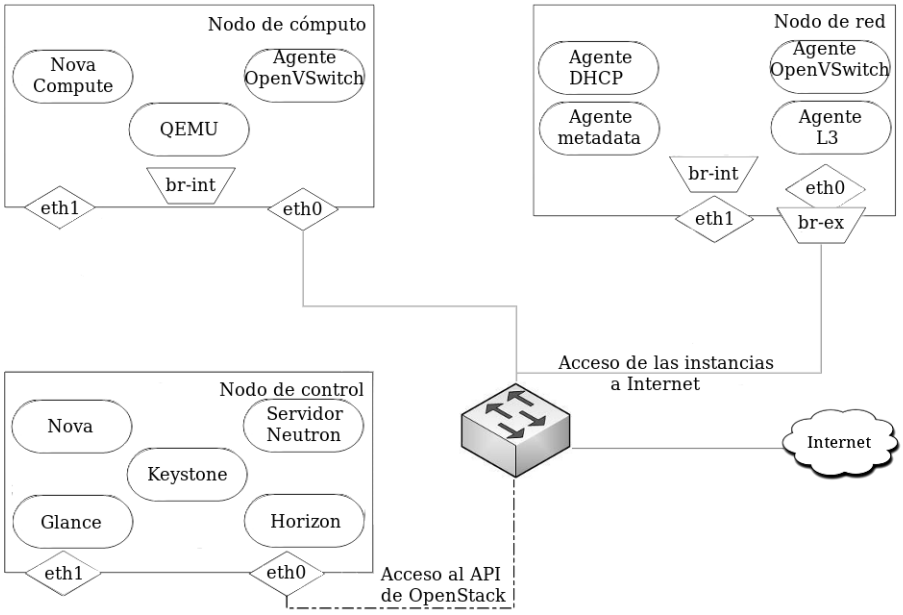

Figura 4.3: Arquitectura y Despliegue de OpenStack en la Solución Implementada

4.4. Especificaciones del despliegue implantado

En esta sección se describen las herramientas de software utilizadas y el hardware sobre el que se desplegó el sistema de nube tomado como caso de estudio.

Todos los nodos que forman la infraestructura física de la nube ejecutan Ubuntu Server 12.04.02 LTS, están conectados a través de un switch físico que pertenece a la infraestructura del Laboratorio, para permitir la coordinación de las operaciones de OpenStack y el acceso a los servicios. Algunas especificaciones importantes se comentan a continuación:

- El **nodo de control** coordina las actividades de los diferentes módulos de OpenStack distribuidos en el despliegue y aloja la base de datos del sistema, implementada con MySQL. A esta base de datos accede Keystone, que controla la información de los usuarios y administradores del sistema, y permite la autenticación de usuarios en el sistema a través de la interfaz de red, conectada a la red del Laboratorio ICARO. El nodo de control también aloja Glance, que almacena las imágenes en archivos, Horizon que mantiene una vista de los recursos recopilando datos de los otros módulos, Neutron Server para coordinar las operaciones de red, y el servicio Nova para coordinar con los diferentes nodos de cómputo. Los diferentes procesos se comunican a través de una cola de mensajes que se implementa con RabbitMQ.
- En el **nodo de cómputo** se ejecutan las máquinas virtuales de la infraestructura interna de la nube, consumiendo los recursos de cómputo disponibles en el nodo físico. Este nodo tiene instalado el paquete Nova Compute, que permite la coordinación con el nodo de control, QEMU para alojar las máquinas virtuales, y el agente OpenVSwitch de Neutron para la comunicación con el nodo de red. También en estos nodos hay un puente de OpenVSwitch, el cual permite la comunicación interna de las máquinas virtuales siendo ejecutadas en el nodo. Nova Compute administra QEMU a través de *libvirt*, un API que permite a Nova Compute utilizar diferentes manejadores de máquinas virtuales.
- El **nodo de red** controla la comunicación entre las máquinas virtuales de la infraestructura interna, incluyendo el tráfico entre las distintas

redes internas y el tráfico con la red física, a través de una de sus interfaces de red. En este nodo está instalado OpenVSwitch, para la implementación de las redes virtuales, incluyendo las abstracciones comunes en conmutación y enrutamiento (subredes, routers, puertos o interfaces de red). El servicio de red de OpenStack (Neutron Server, para esta versión) controla OpenVSwitch a través de un conjunto de agentes que administran los puentes para comunicación de las máquinas virtuales que están siendo ejecutadas en el nodo de cómputo. También están instalados el agente Neutron DHCP y el agente Neutron l3 de OpenVSwitch.

4.5. Interacción intramodular

En el despliegue de la solución cada módulo de OpenStack está formado por distintos demonios, todos los módulos cuentan con un demonio, generalmente llamado <*NombreDelServicio*>-api, que controla la comunicación entre los componentes internos de un módulo y los otros módulos externos. Se seguirá el mismo ejemplo de la creación de una instancia pero con una vista más detallada, además en este caso el usuario también selecciona la asignación de un volumen.

La Figura 4.4 muestra los pasos que se describen a continuación, comenzando por la autenticación del usuario:

1. El Dashboard recibe las credenciales del usuario y hace una llamada a Keystone, pidiendo la autenticación de las mismos.
2. Keystone autentica las credenciales recibidas, también genera y envía un token de autenticación que será utilizado para la comunicación con los otros componentes de OpenStack.
3. El Dashboard, envía el comando para la creación de la instancia a nova-api.
4. nova-api recibe la orden junto con el token de autenticación y hace una llamada a Keystone para que valide el token recibido.
5. Keystone valida el token y envía información actualizada de roles y permisos del usuario.

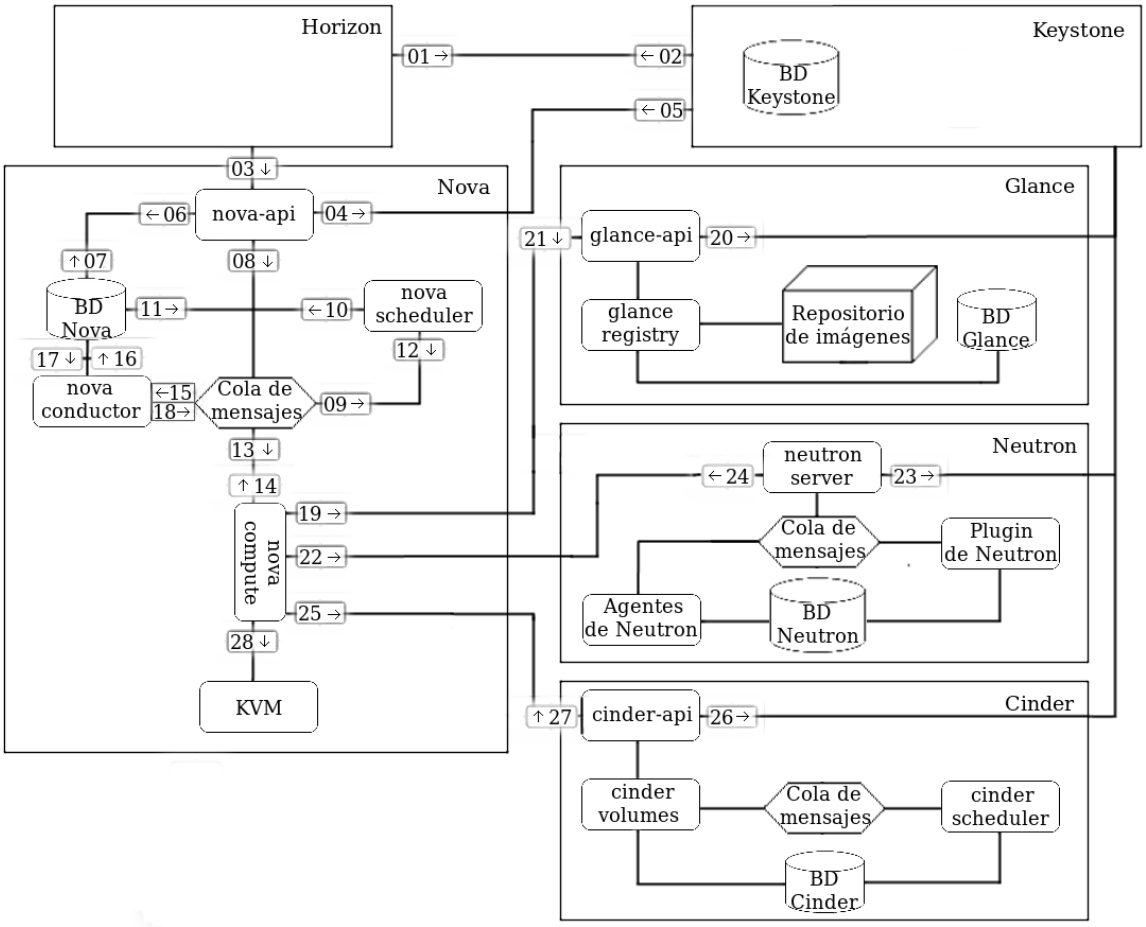

Figura 4.4: Comunicación de los componentes específicos de la solución

6. nova-api interactúa con la base de datos de nova, que contiene el estado del sistema, para verificar los recursos disponibles (*quotas*) para el usuario.

7. Crea una entrada en la base de datos para la instancia nueva.

8. nova-api se comunica con nova-scheduler, el planificador del servicio de cómputo de OpenStack, pidiendo una actualización del identificador de host en la entrada de la instancia nueva en la base de datos.

9. nova-scheduler recoge la petición de la cola.

10. nova-scheduler interactúa con la base de datos de nova para encontrar un host apropiado, esta planificación es completamente configurable en OpenStack.

11. Devuelve la entrada de la instancia actualizada con el identificador del host en el que será ejecutada.

12. nova-scheduler inyecta en la cola información para el lanzamiento de la instancia.

13. nova-compute recibe el mensaje desde la cola.

14. nova-compute se comunica con nova-conductor, que es un intermediario entre nova-compute y la base de datos del sistema implementado por cuestiones de seguridad, para hacer la petición de información sobre los recursos de la instancia.

15. nova-conductor recoge el mensaje de la cola.

16. nova-conductor interactúa con la base de datos del sistema.

17. Devuelve la información de la instancia.

18. nova-compute recoge la información de la instancia en la cola de mensajes.

19. nova-compute hace una llamada a glance-api, con el token de autenticación como parámetro, pidiendo la imagen requerida.

20. glance-api se comunica con Keystone, y este último valida el token de autenticación.

21. nova-compute recibe la imagen junto con sus metadatos.

22. nova-compute hace una llamada a la API de red, con el token de autenticación como parámetro, pidiendo la asignación y configuración de red para la instancia.

23. neutron-server se comunica con Keystone para validar el token de autenticación.

24. nova-compute recibe la información de red.

25. nova-compute hace una llamada a cinder-api, con el token de autenticación como parámetro, para asignar un volumen a la instancia.

26. cinder-api valida el token de autenticación.

27. nova-compute recibe la información de almacenamiento.

28. nova-compute genera los datos y hace la petición para la creación de una máquina virtual en el manejador de máquinas virtuales, a través de libvirt.

El servicio de red de OpenStack está alojado parcialmente en el nodo de control, y parcialmente en el nodo de red; en el nodo de control se llevan a cabo las tareas de coordinación de operaciones y mantiene la base de datos del servicio de red, mientras que en el nodo de red se alojan las redes virtuales y se ejecutan las operaciones inherentes al servicio de red, permite la comunicación entre las máquinas virtuales y la red externa, también los plugins (OpenVSwitch) para crear los aparatos de red (puertos, redes, routers, servidor DHCP) están alojados en el nodo de red. Cada nodo de cómputo tiene instalado un agente del plugin de Neutron, configurado para comunicar metadatos con el nodo de red correspondiente, y un puente que permite la comunicación de datos entre las máquinas virtuales. La mayoría de los otros servicios están alojados en el nodo de control, esto incluye al planificador de Nova, la base de datos del sistema, el repositorio de imágenes de Glance y el manejador de dispositivos de almacenamiento por bloques Cinder. El proceso que se encarga de la manipulación directa de las máquinas virtuales, nova-compute, junto con el manejador de máquinas virtuales, están alojados en el nodo de cómputo.

4.6. Utilización de OpenStack en el Laboratorio ICARO

El despliegue de OpenStack es muy versátil y puede ser utilizado de formas muy diversas. Para los fines del laboratorio se pueden aprovechar las siguientes características:

- La creación de topologías de redes virtuales funcionales.
- El alcance de administración de las redes, subredes y routers.
- La facilidad del lanzamiento y configuración de máquinas virtuales.

Estas características permiten la creación de diversos escenarios ideales para la demostración de funcionalidad de ciertas tecnologías y permite el acceso a capacidades de cómputo fácilmente configurables, para cumplir con requerimientos específicos.

Para utilizar los recursos en OpenStack es necesario crear proyectos; los miembros de cada proyecto son capaces de configurar la infraestructura de red, y desplegar diferentes maquinas virtuales. En el despliegue, todos los proyectos cuentan con un router virtual que permite el acceso de las instancias a la red del Laboratorio, y el administrador debe conectar una interfaz del router a cada red interna que requiera acceso a la red externa.

Los proyectos de la solución se describen a continuación:

- **Administrador.** Es el proyecto que tiene el control sobre los recursos del sistema, los administradores pueden aplicar restricciones a todos los proyectos del sistema.
- **Materias.** Sus miembros pueden crear infraestructuras virtualizadas con objetivos académicos.
- **Tesistas.** Sus miembros pueden formar infraestructuras para la ejecución de aplicaciones que involucren diferentes servidores.
- **Investigadores.** Sus miembros pueden aprovisionarse de recursos de cómputo para diferentes proyectos.

La Figura 4.5 muestra el modelo de servicio desplegado en el Laboratorio. Los diferentes usuarios tienen acceso a un servicio de infraestructura en la nube.

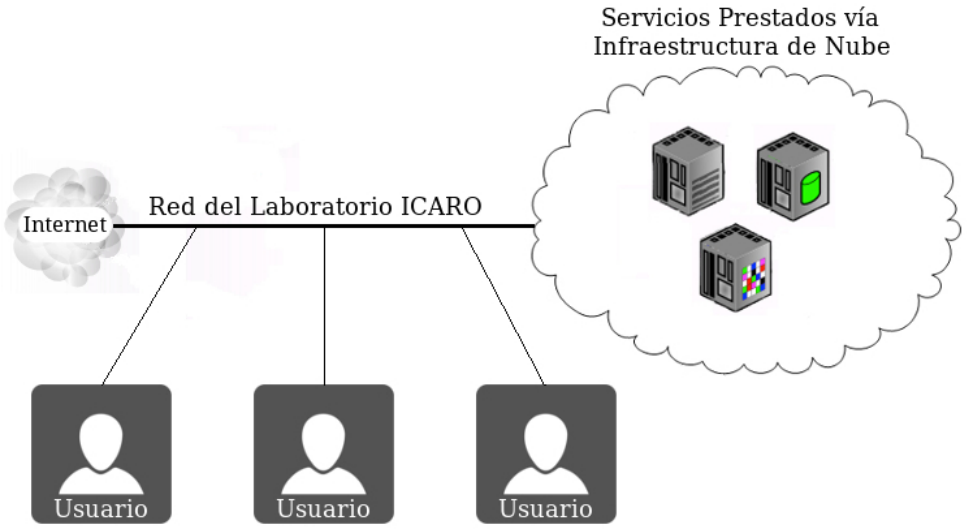

Figura 4.5: Modelo de servicio

Capítulo 5

Pruebas y Resultados

En aras de corroborar el correcto funcionamiento de la implantación del diseño realizado, la solución desplegada se sometió a un ambiente de pruebas. Los módulos Nova y Neutron de OpenStack proveen herramientas para verificar la correcta comunicación de los aparatos de software involucrados en su funcionamiento, esto se cubre en las pruebas de funcionamiento (Sección 5.1). Las pruebas de corrección (Sección 5.2) se enfocan en la interacción que provee el sistema para manipular y configurar las infraestructuras alojadas en la nube, y están basadas sobre las operaciones involucradas en esta interacción del sistema con los usuarios.

El sistema instalado en el Laboratorio ICARO es una nube de modelo IaaS, el mismo permite el despliegue rápido de infraestructuras de hardware virtualizado por demanda, lo cual resume la funcionalidad principal del sistema. Estas infraestructuras están compuestas, básicamente, por máquinas virtuales (instancias) desplegadas sobre redes virtuales configurables. Desde la red física del Laboratorio ICARO, el sistema permite el acceso a las redes virtuales sobre las que se despliegan las instancias. A través de estas redes virtuales, se hace disponible la interacción de los usuarios con las instancias, que es equivalente a acceder remotamente a un servidor físico. La infraestructura es completamente funcional, el acceso a Internet de las instancias se provee a través de la red física del Laboratorio, permitiendo instalar software y hacer actualizaciones, y el acceso desde Internet a las intancias es configurado bajo demanda por el usuario.

La utilización de una infraestructura virtual desplegada en la nube es equiva-

lente a la de una infraestructura física, como un centro de datos, con diferencias notables en tiempos de despliegue y características de administración. Esto cubre otra parte del funcionamiento de la solución, en la que los diferentes módulos interactúan para garantizar que la infraestructura virtual se mantenga funcional, y esté disponible para sus usuarios. Las pruebas sobre esta parte se cubren en la Sección 5.3.

En la Sección 5.5 se presentan conclusiones generales alcanzadas al analizar el resultado de las pruebas.

5.1. Pruebas de funcionamiento

Antes de las pruebas de corrección y estabilidad, se hacen pruebas del funcionamiento de los aparatos de software involucrados en el despliegue del sistema de nube.

El principal módulo involucrado en el funcionamiento del sistema es Nova, que se encarga de ejecutar todas las operaciones sobre las instancias, desde lanzar y terminar una instancia, hasta modificar sus propiedades en tiempo de ejecución. Esto involucra el funcionamiento de varios aparatos del módulo Nova: el planificador (nova-scheduler) escoge un nodo de cómputo apropiado antes de lanzar cada instancia, el encargado de certificar la autenticidad de las operaciones a llevar a cabo (nova-cert), un intermediario para el acceso a la base de datos del sistema (nova-conductor), el encargado de autenticar el acceso por consola desde el Dashboard (nova-consoleauth), y el encargado de alojar las máquinas virtuales (nova-compute), instalado en los nodos destinados a ejecutar las instancias.

La Figura 5.1 muestra el resultado de la ejecución del comando de Nova *nova-manage service list*, que lista los aparatos del módulo Nova y sus localidades y estados. El host *nube* es el nodo de control y el host *computo01* es el nodo de cómputo del despliegue.

Otro módulo importante es Neutron, que administra las redes sobre las que son lanzadas las instancias. Para esto cuenta con un conjunto de agentes: DHCP-agent para asignar automáticamente configuraciones de red a las instancias, L3-agent que habilita el enrutamiento entre diferentes redes, OpenVSwitch-agent que mantiene activas las redes creadas. En la Figura 5.2 se muestra el listado de los agentes de Neutron, junto con su localidad y

```
root@nube:/home/nubeu# nova-manage service list
Binary           Host                                 Zone             Status     State Updated_At
nova-cert        nube                                 internal         enabled    :-)   2014-03-24
nova-conductor   nube                                 internal         enabled    :-)   2014-03-24
nova-consoleauth nube                                 internal         enabled    :-)   2014-03-24
nova-scheduler   nube                                 internal         enabled    :-)   2014-03-24
nova-compute     computo01                            nova             enabled    :-)   2014-03-24
```

Figura 5.1: Pruebas del servicio Nova

```
root@nube:/home/nubeu# neutron agent-list
+--------------------------------------+--------------------+-----------+-------+----------------+
| id                                   | agent_type         | host      | alive | admin_state_up |
+--------------------------------------+--------------------+-----------+-------+----------------+
| 10f4c9f6-dc37-4fc3-bc5d-91780b52d78d | L3 agent           | nodored   | :-)   | True           |
| 88524309-0497-4dfb-8492-54788d4a3dff | DHCP agent         | nodored   | :-)   | True           |
| 8e10cb6b-516b-48be-93e5-588e4459ac88 | Open vSwitch agent | nodored   | :-)   | True           |
| f2a7d078-4da7-4535-994c-ace731669bda | Open vSwitch agent | computo01 | :-)   | True           |
+--------------------------------------+--------------------+-----------+-------+----------------+
```

Figura 5.2: Pruebas del servicio Neutron

estado. El host *nodored* es el nodo de red del despliegue.

5.2. Pruebas de correción

En esta sección se explican las pruebas de correción llevadas a cabo sobre el sistema, y se muestran los resultados.

Para permitir la manipulación de las infraestructuras virtuales alojadas en la nube por parte de los usuarios, el sistema cuenta con un módulo, Horizon, que da acceso a usuarios y administradores. Horizon se comunica con todos los otros módulos del sistema, y cumple la función de intermediario entre los usuarios y los módulos del sistema, que proveen de las funcionalidades principales. Con este objetivo cuenta con una interfaz web que muestra una vista personalizada y en tiempo real de los recursos, y un conjunto de opciones para modificar la infraestructura virtual disponible y hacer diferentes configuraciones de forma segura y ágil. La prueba se basa en los resultados de estas opciones para demostrar la funcionalidad integral del sistema.

Mediante Horizon, un usuario administrador tiene acceso a las siguientes funcionalidades:

- Acceso al sistema por nombre de usuario y contraseña.
- Administrar proyectos (projects o tenants) y usuarios.
  - Crear y eliminar proyectos, añadir o remover usuarios de proyectos.
  - Crear, habilitar, deshabilitar y remover usuarios.
  - Crear, habilitar, deshabilitar y remover grupos de acceso.
  - Crear, habilitar, deshabilitar y remover roles.
- Administrar las instancias en el sistema.
  - Crear y remover copias instantáneas de las instancias.
  - Controlar el estado de las instancias.
- Administrar volúmenes y tipos de volúmenes.
  - Crear y remover tipos de volumen.
  - Remover volumen.
- Administrar imágenes de sistema operativo.
  - Crear imágenes de sistema operativo.
  - Actualizar datos de las imágenes.
  - Remover imágenes.
- Administrar sabores de instancias.
  - Crear sabores.
  - Actualizar datos de los sabores.
  - Remover sabores.
- Administrar cuotas de utilización de recursos.
  - Actualizar cuotas de proyectos.
- Vista de los recursos de la nube.

A un usuario regular, la misma interfaz permite un acceso limitado a los proyectos que tiene asignados, y las siguientes opciones:

- Acceso al sistema por nombre de usuario y contraseña.
- Administrar imágenes de sistema operativo.
  - Crear imágenes de sistema operativo.
  - Actualizar datos de las imágenes.
  - Remover imágenes.
- Configurar acceso y seguridad a sus instancias.
  - Añadir reglas al grupo de seguridad por defecto.
  - Añadir un par de claves.
  - Importar un par de claves.
- Administrar aparatos virtuales de red.
  - Crear, configurar y remover red.
  - Crear, configurar y remover subred.
  - Crear, configurar y remover router.
  - Asignar dirección IP flotante.
- Crear y administrar instancias.
  - Crear, configurar y remover red.
  - Acceso por SSH a las instancias.
  - Crear y remover copias instantáneas de las instancias.
  - Controlar los estados de las instancias.
- Administrar almacenamiento por volúmenes.
  - Crear y remover volumen.
  - Asignar y desasignar volumen a instancia.
  - Crear instantáneas de los volúmenes.

La prueba consiste en la creación de un proyecto de prueba junto con un usuario de prueba. Todas las funcionalidades disponibles al administrador se ejecutan sistemáticamente sobre el proyecto prueba y sobre el usuario prueba, registrando el resultado de los cambios. Luego se sigue el mismo procedimiento desde la perspectiva de un usuario típico.

La lista de las funcionalidades, junto con el resultado de la prueba en cada una, se muestra en el cuadro 5.1.

	Nodos Involucrados			
Prueba	**Control**	**Red**	**Cómputo**	**Resultado**
Usuario administrador				
Acceso usuario/contraseña	**x**			✓
Administrar proyectos	**x**			✓
Administrar usuarios	**x**			✓
Administrar grupos de acceso	**x**			✓
Administrar roles	**x**			✓
Administrar instancias	**x**	**x**	**x**	✓
Administrar volúmenes	**x**			✓
Administrar imágenes	**x**			✓
Administrar sabores	**x**			✓
Administrar cuotas	**x**			✓
Usuario regular				
Acceso usuario/contraseña	**x**			✓
Administrar imágenes	**x**			✓
Configurar acceso y seguridad	**x**			✓
Administrar aparatos de red	**x**	**x**		✓
Crear y administrar instancias	**x**	**x**	**x**	✓
Administrar volúmenes	**x**			✓

Cuadro 5.1: Pruebas de corrección

Todas las operaciones tienen efecto inmediato sobre el estado del sistema. Las operaciones administrativas no modifican la infraestructura virtual, sino que permiten crear un ambiente coherente y seguro. Algunas de las operaciones disponibles a usuarios regulares cambian la infraestructura virtual alojada en la nube, sus resultados se pueden verificar confirmando estos cambios. Por ejemplo, al conectar una interfaz de un router a una subred ya existente, se habilita la comunicación de las instancias pertenecientes a estas subredes,

esto se puede verificar utilizando ICMP (Internet Control Message Protocol) desde las instancias desplegadas sobre esas redes.

5.3. Prueba de estabilidad

En esta sección se explica la prueba de estabilidad llevada a cabo sobre el sistema.

Un sistema de nube no sólo se reconoce por las facilidades de despliegue de infraestructuras virtualizadas en cuanto a tiempo y coste asociados. También es importante la estabilidad en el funcionamiento de estas infraestructuras, lo que denota robustés en la implementación del sistema. En ambientes de producción, el consumidor espera alta confiabilidad y, en muchos casos, alta disponibilidad de las infraestructuras alojadas por un sistema de nube.

La prueba incluye un despliegue de diferentes instancias sobre dos redes virtuales, conectadas por un router. Este despliegue consume una gran parte de los recursos del sistema, y se mantiene levantado por varias horas. Se espera del sistema que mantenga los cambios aplicados sobre la infraestructura antes de comenzar la prueba, y que sus instancias respondan de igual forma luego de varias horas en funcionamiento.

La topología del despliegue se muestra en la Figura 5.3.

Las intancias A y C ejecutan Ubuntu, la instancia B ejecuta Windows XP y las instancias D y E ejecutan Cirros, una imagen de prueba muy ligera. Entre las instancias A y C hay un constante flujo de mensajes ICMP (Internet Control Message Protocol), de manera que el router virtual se vea involucrado en la prueba.

Nombre	**VCPUs**	**RAM(MB)**	**Disco(GB)**	**Uso(Horas/Segs)**
A	1	600	5	8.05/28980)
C	1	600	5	8.15/29340)
E	1	50	1	8.14/29304)
D	1	50	1	8.13/29268)
B	1	1600	10	8.04/28944)

Cuadro 5.2: Consumo de recursos por instancia durante la prueba

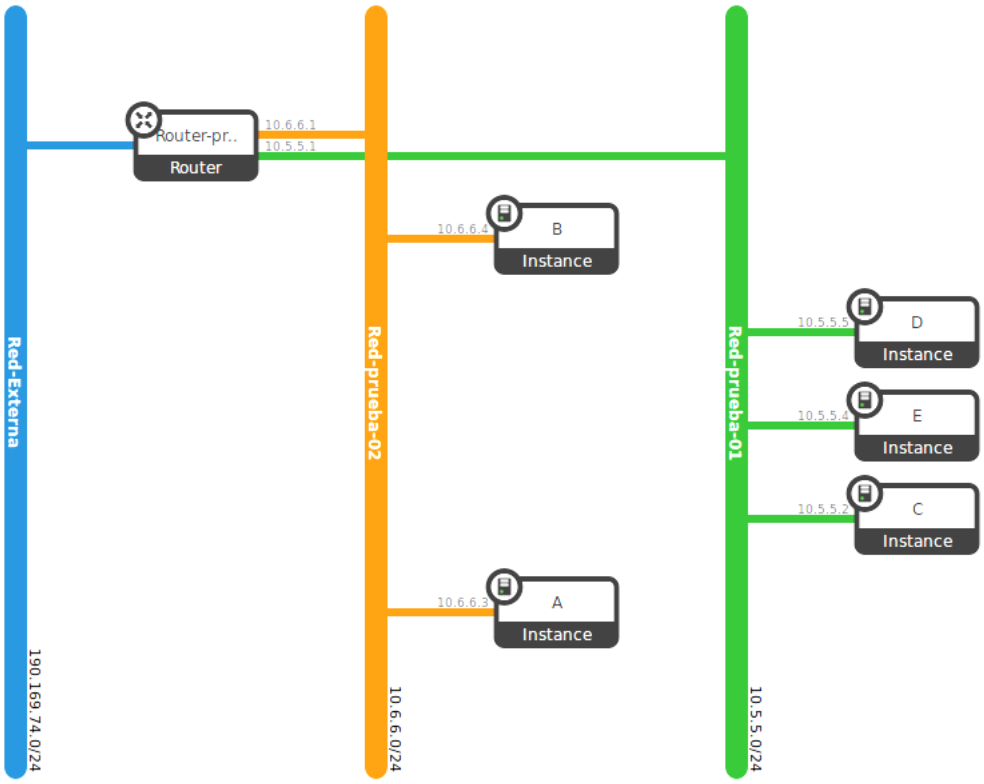

Figura 5.3: Topología de la prueba

Reporte del período:	Mar. 23 2014, Mar. 24 2014
ID del proyecto:	e34c1b1adda44ae69ef34b11bf5fef28
Total VCPUs activos:	5
Horas de CPU usadas:	40.96
Total RAM activa(MB):	2900
Total de almacenamiento(GB):	22
Total de almacenamiento utilizado(GB):	2.30

Cuadro 5.3: Total de consumo de recursos durante la prueba

Los Cuadros 5.2 y 5.3 muestran información obtenida del Dashboard luego de ocho horas de haber desplegado las instancias. Entre estos están incluidos los CPUs (Central Processing Unit) virtuales a los que tiene acceso cada instancia, memoria principal (Random Access Memory), tamaño del disco, y el tiempo que tiene activa cada máquina virtual. El Cuadro 5.3 muestra el total en el consumo de los recursos por el proyecto *prueba*, cuyo identificador se muestra también en el mismo cuadro. Las instancias se mantuvieron activas durante la prueba, cumpliendo las espectativas.

5.4. Pruebas de estrés

Un sistema como este, que involucre el lanzamiento de diferentes máquinas virtuales, plantea una dificultad en la medición precisa del consumo de los recursos en el tiempo. La funcionalidad principal del sistema es el lanzamiento de máquinas virtuales, de manera que las pruebas de estrés estarían enfocadas en esta operación. En particular, no contamos con herramientas que permitan medir con presición el tiempo que transcurre entre la orden de lanzamiento de una máquina virtual, y el momento en el que la instancia queda activa y disponible para el usuario, este proceso involucra una carga de trabajo distribuida entre diferentes módulos del sistema y el manejador de máquinas virtuales.

Durante las pruebas, se intentó utilizar Zenoss, una herramienta que permite medir disponibilidad, desempeño y eventos, en infraestructuras físicas o virtuales. Sin embargo no fue posible utilizar esta herramienta, por cuestiones de compatibilidad entre Zenoss y OpenStack, que no permitió el acceso a las interfaces de comunicación de los módulos del sistema.

De cualquier forma, las prestaciones del despliegue permiten un margen muy pequeño de carga para hacer las pruebas de estrés, en cuanto a lanzamiento de máquinas virtuales se refiere. Al lanzar dos máquinas virtuales de tamaño medio (1 CPU virtual, 600 MB RAM, 10 GB de almacenamiento), el sistema se encuentra bajo una carga moderada, disminuyendo considerablemente el desempeño general. De manera que se llegó a la conclusión de que el despliegue no está en condiciones de ser sometido a una prueba de estrés.

5.5. Resultados generales

El funcionamiento de un sistema de nube involucra una interacción dinámica con sus usuarios. Estos sistemas están diseñados para permitir elasticidad rápida de los recursos virtuales disponibles y un control en tiempo real por parte de sus usuarios, para esto proveen diferentes operaciones administrativas. En las pruebas realizadas se tomaron en cuenta estas operaciones y los efectos que tienen sobre la infraestructura virtualizada alojada en la nube. De la ejecución sin errores de estas operaciones se puede inferir la correcta comunicación y el correcto funcionamiento de los módulos del sistema involucrados en las mismas, permitiendo concluir que el sistema está bien configurado y responde como se espera.

Capítulo 6

Conclusiones

Durante este trabajo de investigación se ha diseñado y desarrollado un sistema de provisionamiento de cómputo vía máquinas virtuales usando una infraestructura de nube. El desarrollo de esta investigación cubre las bases teóricas de esta tecnología, se indaga sobre sus posibles usos y ventajas, además de complementar la información a través de un ejemplo funcional. Del mismo modo esta investigación se reflejan las diferentes formas que puede tomar una nube, haciendo énfasis en el modelo de servicio IaaS.

Tener una infraestructura de nube instalada en un centro de datos local permite el control total sobre la utilización de la misma, cualquier requerimiento puede ser configurado por los administradores. Esto permite incluso errar en el despliegue de las infraestructuras virtualizadas, lo que es ideal para casos de instrucción académica, ya que los cambios que solucionan estos errores son de fácil aplicación. El Laboratorio ICARO, además de recibir estudiantes, también es un centro para investigadores que pueden beneficiarce de las facilidades de despliegue de infraestructuras virtualizadas. La infraestructura de virtualización desplegada en el presente trabajo de investigación, permitirá utilizar sistemas de cómputo virtualizado por demanda, a los miembros de la comunidad del Laboratorio, con fines de probar software, satisfacer necesidades de cómputo y transmitir conocimientos.

OpenStack, el caso de estudio de la investigación, es un proyecto de software libre que cuenta con el interés y apoyo de muchas compañias de tecnologías de la información. Sus módulos están en constante evolución, siempre intentando acercarce a la óptima administración de los conjuntos de recursos de proce-

samiento, almacenamiento y red. El sistema permite amplias posibilidades de despliegue y configuración de sus módulos, abriendo posibles soluciones a problemas diversos. Por lo anterior, esta solución es ideal para el despliegue de infraestructuras virtualizadas en un ambiente de investigación. Como ejemplo de la utilización de OpenStack está la Organización Europea para la Investigación Nuclear (CERN), que tiene requerimientos muy particulares de cálculo y almacenamiento, los resultados de un experimento puntual pueden generar 1 Petabyte por segundo de datos que deben ser registrados para su posterior análisis [32].

La tecnología de la nube resulta ser muy útil para resolver problemas diversos de cálculo, que involucran requerimientos de alta escala en cómputo y almacenamiento. A diferencia de un centro de datos convencional, que administra la información de forma estática, la nube está diseñada para desplegar rápidamente grandes capacidades de cómputo y luego liberarlas, de manera que la infraestructura física que soporta los servicios de nube pueda ser reservada para momentos críticos de utilización, y no reservada permanentemente. Se toma en cuenta de igual forma que lo que hoy son requerimientos de altísima escala, y representan problemas especializados, mañana serán requerimientos regulares. Es la opinión del autor que la nube es una tecnología pivote para el futuro de la administración de la información.

6.1. Contribuciones

El presente trabajo especial de grado hace las siguientes contribuciones:

- Provee de un primer acercamiento a la tecnología de nube, el cual cubre las bases teóricas y las complementa con un ejemplo funcional.
- Una nube de infraestructura como un servicio totalmente funcional en las instalaciones del Laboratorio ICARO, que puede servir para diferentes actividades de investigación.
- Desarrollo de una solución de cómputo como soporte para la instrucción de los laboratorios y prácticas de las materias dictadas en la licenciatura de la Escuela de Computación.

6.2. Limitaciones

A pesar de que la solución de virtualización desplegada se encuentra funcional y representa un aporte importante para las actividades académicas y de investigación del Laboratorio ICARO, es de notar que dicho despliegue es de prestaciones modestas, debido principalmente a los pocos recursos de hardware con que se cuenta en el laboratorio. Para mejorar el rendimiento sería necesario, inicialmente, un switch que permita un mayor ancho de banda, y que el nodo de red cuente con tres interfaces de red, en lugar de dos, para permitir una optimización de recursos red. Además, según la guía oficial de operaciones de OpenStack ([31]), para un ambiente de producción se recomienda que los servidores físicos destinados a control y red cuenten con al menos 32 GB de memoria principal, y 128 GB para el nodo de cómputo, mientras que en el despliegue implantado cada nodo cuenta con 4 GB. Esta misma desproporción de recursos recomendados se nota en almacenamiento secundario y procesamiento.

6.3. Trabajos futuros

Se proponen los siguientes trabajos futuros:

- Despliegues y configuraciones alternativos de los módulos de OpenStack, en particular los tres módulos que no fueron incluídos en el despliegue (Swift, Heat y Ceilometer)
- La integración de la administración de usuarios de Keystone con el sistema de autenticación LDAP (Lightweight Directory Access Protocol) que funciona en el Laboratorio
- Estudiar la posibilidad de añadir nodos de almacenamiento masivo
- Expandir las funcionalidades de los módulos de OpenStack (Ej. autoescalado para Nova)

Bibliografía

[1] Peter Mell , Timothy Grance, *The NIST Definition of Cloud Computing*, Septiembre, 2011.

[2] Michael Armbrust, Armando Fox et al, *A View of Cloud Computing*, Abril, 2010.

[3] Michael Kuperberg, Nikolas Herbst et al, *Defining and Quantifying Elasticity of Resources in Cloud Computing and Scalable Platforms*, 2011.

[4] Khaled M. Khan, *Security Dynamics of Cloud Computing*, 2009.

[5] Alok Tripathi, Abhinav Mishra, *Cloud Computing Security Considerations*, 2010.

[6] Andrzej Goscinski, James Broberg et al, *Cloud Computing: Principles and Paradigms*, 2011.

[7] Hassan Takabi, James B.D. Joshi, Gail-Joon Ahn *Security and Privacy Challenges in Cloud Computing Environments*, Diciembre, 2010.

[8] *iCloud Service* [En línea], [Consultado en Diciembre, 2012.] http://www.apple.com/es/icloud/

[9] *Egnyte HybridCloud* [En línea], [Consultado en Diciembre, 2012.] http://www.egnyte.com/online-storage/fast-local-access.html

[10] *Google Apps* [En línea], [Consultado en Diciembre, 2012.] http://en.wikipedia.org/wiki/Google_Apps

[11] *OpenDrive* [En línea], [Consultado en Diciembre, 2012.] https://www.opendrive.com/

[12] *Dropbox* [En línea], [Consultado en Diciembre, 2012.] http://en.wikipedia.org/wiki/Dropbox_(service)

[13] *Amazon Cloud Drive* [En línea], [Consultado en Diciembre, 2012.] http://en.wikipedia.org/wiki/Amazon_Cloud_Drive

[14] *Proveedores de Nube* [En línea], [Consultado en Diciembre, 2012.] http://searchcloudcomputing.techtarget.com/photostory/2240149038/Top-10-cloud-providers-of-2012/1/Introduction

[15] Mohiuddin Ahmed, Abu Sina Md. Raju Chowdhury et al, *An Advanced Survey on Cloud Computing*, Enero, 2012.

[16] *Amazon Cloud Service Cloud* [En línea], [Consultado en Enero, 2013.] http://aws.amazon.com/

[17] *Xen Cloud Platform* [En línea], [Consultado en Enero, 2013.] http://www.xen.org/products/cloudxen.html

[18] *Citrix XenServer* [En línea], [Consultado en Enero, 2013.] http://www.citrix.com/products/xenserver/

[19] *Quick EMUlator* [En línea], [Consultado en Enero, 2013.] http://en.wikipedia.org/wiki/QEMU

[20] *VMware vSphere* [En línea], [Consultado en Enero, 2013.] http://www.vmware.com/products/datacenter-virtualization/vsphere

[21] *VMware vCloud Suite* [En línea], [Consultado en Enero, 2013.] http://www.vmware.com/products/datacenter-virtualization/vcloud-suite/

[22] *Eucalyptus Cloud* [En línea], [Consultado en Enero, 2013.] http://www.eucalyptus.com/eucalyptus-cloud/

[23] *OpenStack* [En línea], [Consultado en Enero, 2013.] http://www.openstack.org/

[24] *OpenNebula* [En línea], [Consultado en Enero, 2013.] http://opennebula.org/

[25] Rohit Bhadauria, Nabendu Chaki et al, *A Survey on Security Issues in Cloud Computing*, 2011.

[26] Danish Jamil, Hassan Zaki, *Cloud Comuting Security*, Abril, 2011.

[27] Yanpei Chen, Vern Paxson et al, *What's New About Cloud Computing Security?*, Enero, 2010.

[28] Jay Heiser, Mark Nicolett, *Assessing the Security Risks of Cloud Computing*, Junio, 2008.

[29] *OpenStack* [En línea], [Consultado en Octubre, 2013.] http://en.wikipedia.org/wiki/OpenStack

[30] *OpenStack Networking Administration Guide* [En línea], [Consultado en Septiembre, 2013.] docs.openstack.org

[31] *OpenStack Operations Guide* [En línea], [Consultado en Septiembre, 2013.] docs.openstack.org

[32] *OpenStack - Usuarios* [En línea], [Consultado en Abril, 2014.] www.openstack.org/user-stories/cern/

Capítulo 7

Anexos

De forma detallada en el presente capítulo se describe las actividades relacionadas con la instalación y configuración de la solución desarrollada en este trabajo de investigación. Se instalaron los siguientes módulos de OpenStack: Nova, Neutron, Cinder, Keystone, Glance y Horizon. Estos módulos se distribuyen en los siguientes nodos físicos: Control, Red y Cómputo, que en este caso, pertenecen a la misma red.

7.1. Preparación de los nodos físicos

En esta sección se muestran los pasos a seguir en todos los nodos físicos, para prepararlos antes de la instalación de OpenStack.

Instalar el sistema operativo base en los tres servidores. La versión de Ubuntu Server que se utiliza en el despliegue es 12.04.02. Entre las opciones que muestra Ubuntu en la instalación, seleccionar Çonfigurar LVM". Cada servidor cuenta con dos interfaces de red, que serán configuradas acontinuación.

Hacer las configuraciones pertinentes de red. La primera de las interfaces de cada servidor está conectada a una red, de manera que puedan comunicarse entre ellos. Adicionalmente la segunda interfaz del nodo de red, tiene una configuración promiscua como se muestra:

/etc/network/interfaces

auto eth1
iface eth1 inet manual
up ifconfig $IFACE 0.0.0.0 up
up ip link set $IFACE promisc on
down ip link set $IFACE promisc off
down ifconfig $IFACE down

Añadir el repositorio de OpenStack Havana para Ubuntu y actualizar el software del sistema. Esto se logra con el siguiente comando:

apt-get update && apt-get -y install python-software-properties && add-apt-repository -y cloud-archive:havana && apt-get update && apt-get -y upgrade dist-upgrade && apt-get -y autoremove && reboot

Instalar NTP y configurar los servidores para que se sincronicen con el nodo de control. Esto se logra modificando el archivo de configuración de NTP:

apt-get install -y ntp

/etc/ntp.conf
#server 0.ubuntu.pool.ntp.org
#server 1.ubuntu.pool.ntp.org
#server 2.ubuntu.pool.ntp.org
#server 3.ubuntu.pool.ntp.org

Use Ubuntu's ntp server as a fallback.
server *dirección-IP-nodo-de-control*

7.2. Nodo de control

Instalar todos los paquetes del nodo de control.

apt-get -y install mysql-server python-mysqldb rabbitmq-server ntp keystone python-keystone python-keystoneclient glance nova-api nova-cert novnc nova-consoleauth nova-scheduler nova-novncproxy nova-doc nova-conductor nova-ajax-console-proxy python-novaclient openstack-dashboard memcached libapache2-mod-wsgi cinder-api cinder-scheduler cinder-volume iscsitarget open-

iscsi iscsitarget-dkms

Crear las bases de datos

mysql -u root -p

CREATE DATABASE cinder;
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON cinder.* TO 'cinder'@'%' IDENTIFIED BY 'password';

CREATE DATABASE glance;
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON glance.* TO 'glance'@'%' IDENTIFIED BY 'password';

CREATE DATABASE keystone;
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%' IDENTIFIED BY 'password';

CREATE DATABASE neutron;
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON neutron.* TO 'neutron'@'%' IDENTIFIED BY 'password';

GRANT ALL PRIVILEGES ON keystone.* TO 'neutron'@'%' IDENTIFIED BY 'password';

CREATE DATABASE nova;
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON nova.* TO 'nova'@'%' IDENTIFIED BY

'password';

FLUSH PRIVILEGES;
QUIT;

Configurar Keystone y crear los usuarios del sistema

Modificar /etc/keystone/keystone.conf con lo siguiente:
admin_token = *numero-aleatorio*

Crear un archivo admin.token.creds con lo siguiente:
export OS_SERVICE_TOKEN=*numero-aleatorio*
export OS_SERVICE_ENDPOINT=http://190.169.74.141:35357/v2.0

Crear un archivo admin.user.creds con lo siguiente:
export OS_AUTH_URL=http://190.169.74.141:5000/v2.0
export OS_TENANT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=password

source admin.token.creds

Modificar /etc/keystone/keystone.conf con lo siguiente:
connection = mysql://keystone:password@190.169.74.141/keystone

keystone-manage db_sync

restart keystone

keystone tenant-create –name=admin
keystone tenant-create –name=service

keystone user-create –name=admin –pass=password –email=admin@ciens.ucv.ve

keystone role-create –name=admin
keystone role-create –name=KeystoneAdmin
keystone role-create –name=KeystoneServiceAdmin
keystone role-create –name=Member

keystone user-role-add –tenant=admin –user=admin –role=admin
keystone user-role-add –tenant=admin –user=admin –role=KeystoneAdmin
keystone user-role-add –tenant=admin –user=admin –role=KeystoneServiceAdmin

En el despliegue, al crear los proyectos se asiganaron automáticamente los siguientes identificadores:
admin id = 30960bd07fac47f5aff5c37fb72019a2
service id = b8ca97e1eae84cc898cf6222e163ee5f

keystone user-create –name=cinder –pass=password –tenant-id=b8
ca97e1eae84cc898cf6222e163ee5f –email=cinder@ciens.ucv.ve

keystone user-create –name=glance –pass=password –tenant-id=b8
ca97e1eae84cc898cf6222e163ee5f –email=glance@ciens.ucv.ve

keystone user-create –name=neutron –pass=password –tenant-id=b8
ca97e1eae84cc898cf6222e163ee5f –email=neutron@ciens.ucv.ve

keystone user-create –name=nova –pass=password –tenant-id=b8
ca97e1eae84cc898cf6222e163ee5f –email=nova@ciens.ucv.ve

keystone user-role-add –tenant=service –user=cinder –role=admin
keystone user-role-add –tenant=service –user=glance –role=admin
keystone user-role-add –tenant=service –user=neutron –role=admin
keystone user-role-add –tenant=service –user=nova –role=admin

keystone service-create –name=cinder –type=volume

keystone endpoint-create –region=RegionOne –service-id=cf3c2f17ab8744

489b9e0c326f5ef3aa –adminurl='http://190.169.74.141:
8776/v1/%(tenant_id)s' –internalurl='http://190.169.74.141:8776/v1/%(tenant_id)s'
–publicurl='http://190.169.7
4.141:8776/v1/%(tenant_id)s'

keystone service-create –name=glance –type=image

keystone endpoint-create –region=RegionOne –service-id=98b81f3cc6f14c7ca
b8c0187fd0e2857 –adminurl=http://190.169.74.141:9292/ –internalurl=http://190.169.74.1
41:9292/ –publicurl=http://190.169.74.141:9292/

keystone service-create –name=keystone –type=identity

keystone endpoint-create –region=RegionOne –service-id=5a3d404f686843299d9495
717a97ff27 –adminurl=http://190.169.74.141:35357/v2.0 –internalurl=http://190.1
69.74.141:5000/v2.0 –publicurl=http://190.169.74.141:5000/v2.0

keystone service-create –name=neutron –type=network

keystone endpoint-create –region=RegionOne –service-id=bbaff56d1631487
6a3c94892e391ee01 –adminurl=http://190.169.74.142:9696/ –internalurl=htt
p://190.169.74.142:9696/ –publicurl=http://190.169.74.142:9696/

keystone service-create –name=nova –type=compute

keystone endpoint-create –region=RegionOne –service-id=10083dbd13764475b26
c6ccd0407698f –adminurl='http://19
0.169.74.141:8774/v2/%(tenant_id)s' –internalurl='http://190.169.74.141:8774/v2/%(tenant_
–publicurl='http:/
/190.169.74.141:8774/v2/%(tenant_id)s'

unset OS_SERVICE_TOKEN
unset OS_SERVICE_ENDPOINT

Configurar Glance y añadir la imagen de prueba Cirros

Modificar /etc/glance/glance-api.conf con lo siguiente:
sql_connection = mysql://glance:password@190.169.74.141/glance
rabbit_host = 190.169.74.141
auth_host = 190.169.74.141
admin_tenant_name = service
admin_user = glance
admin_password = password
flavor = keystone

Modificar /etc/glance/glance-api-paste.ini con lo siguiente:
auth_host = 190.169.74.141
admin_tenant_name = service
admin_user = glance
admin_password = password

Modificar /etc/glance/glance-registry.conf con lo siguiente:
sql_connection = mysql://glance:password@190.169.74.141/glance
auth_host = 190.169.74.141
admin_tenant_name = service
admin_user = glance
admin_password = password
flavor = keystone

Modificar /etc/glance/glance-registry-paste.ini con lo siguiente:
auth_host = 190.169.74.141
admin_tenant_name = service
admin_user = glance
admin_password = password

glance-manage db_sync

restart glance-api && restart glance-registry

glance image-create –name=cirros –disk-format=qcow2 –container-format=bare

–is-public=true –location=https://launchpad.net/cirros/trunk/0.3.0/+download/cirros-0.3.0-x86_64-disk.img

Configurar Cinder y crear el volumen principal

sed -i 's/false/true/g' /etc/default/iscsitarget
service iscsitarget start
service open-iscsi start

Modificar /etc/cinder/api-paste.ini con lo siguiente:
paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = 190.169.74.141
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = cinder
admin_password = password

Modificar /etc/cinder/cinder.conf con lo siguiente:
rootwrap_config=/etc/cinder/rootwrap.conf
sql_connection = mysql://cinder:password@190.169.74.141/cinder
api_paste_config = /etc/cinder/api-paste.ini
iscsi_helper=ietadm
volume_name_template = volume-%s
volume_group = cinder-volumes
verbose = True
auth_strategy = keystone
iscsi_ip_address=190.169.74.141
rpc_backend = cinder.openstack.common.rpc.impl_kombu
rabbit_host = 190.169.74.141
rabbit_port = 5672
rabbit_userid = guest
rabbit_password = guest

cinder-manage db sync

dd if=/dev/zero of=cinder-volumes bs=1 count=0 seek=10G
losetup /dev/loop2 cinder-volumes
fdisk /dev/loop2
Escribir los siguientes parámetros:
n
p
1
ENTER
ENTER
t
8e
w

pvcreate /dev/loop2
vgcreate cinder-volumes /dev/loop2

Modificar /etc/rc.local con lo siguiente:
losetup /dev/loop2 /home/nubeu/cinder-volumes

Modificar /etc/lvm/lvm.conf con lo siguiente:
devices
filter = .ª/sda1/", .ª/loop2/", r/.*/"

Configurar Nova

Modificar /etc/nova/nova.conf con lo siguiente:
api_paste_config = /etc/nova/api-paste.ini
auth_strategy = keystone
novncproxy_base_url=http://190.169.74.141:6080/vnc_auto.html
vnc_enabled = true
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = 190.169.74.141
rpc_backend = nova.rpc.impl_kombu
rabbit_host = 190.169.74.141
network_api_class=nova.network.neutronv2.api.API
neutron_url=http://190.169.74.142:9696

neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=password
neutron_admin_auth_url=http://190.169.74.141:5000/v2.0
firewall_driver=nova.virt.firewall.NoopFirewallDriver
security_group_api=neutron
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterfaceDriver
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtGenericVIFDriver
service_neutron_metadata_proxy = True
neutron_metadata_proxy_shared_secret = helloOpenStack
connection = mysql://nova:password@190.169.74.141/nova

Modificar /etc/nova/api-paste.ini con lo siguiente:
auth_host = 190.169.74.141
auth_tenant_name = service
auth_user = nova
auth_password = password

nova-manage db sync

rm /var/lib/nova/nova.sqlite

Modificar /etc/nova/nova.conf con lo siguiente:
my_ip=190.169.74.141
vncserver_listen=190.169.74.141
vncserver_proxyclient_address=190.169.74.141

Modificar /etc/nova/api-paste.ini con lo siguiente:
auth_uri = http://190.169.74.141:5000/v2.0

service nova-api restart
service nova-cert restart
service nova-consoleauth restart
service nova-scheduler restart

service nova-conductor restart
service nova-novncproxy restart

Configurar Horizon
Modificar /etc/openstack-dashboard/local_settings.py con lo siguiente:
TIME_ZONE = ÜTC"
OPENSTACK_HOST = "190.169.74.141"

apt-get purge openstack-dashboard-ubuntu-theme

7.3. Nodo de red

Instalar los paquetes del nodo de red y los agentes
apt-get install -y neutron-server neutron-plugin-openvswitch-agent neutron-dhcp-agent neutron-l3-agent neutron-metadata-agent openvswitch-switch openvswitch-datapath-dkms ntp python-mysqldb

Configurar Neutron y sus agentes

Modificar /etc/neutron/neutron.conf con lo siguiente:
core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin
service_plugins = neutron.services.l3_router.l3_router_plugin.L3RouterPlugin
api_paste_config = /etc/neutron/api-paste.ini
allow_overlapping_ips = True
rabbit_host = 190.169.74.141
auth_host = 190.169.74.141
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = neutron
admin_password = password
signing_dir = /var/lib/neutron/keystone-signing

Modificar /etc/neutron/api-paste.ini con lo siguiente:

paste.filter_factory = keystoneclient.middleware.auth_token:filter_factory
auth_host = 190.169.74.141
admin_tenant_name = service
admin_user = neutron
admin_password = password

mkdir /etc/neutron/plugins/ml2

Modificar /etc/neutron/plugins/ml2/ml2_conf.ini con lo siguiente:
type_drivers = gre
tenant_network_types = gre
mechanism_drivers = openvswitch,linuxbridge
tunnel_id_ranges = 1:1000
sql_connection = mysql://neutron:password@190.169.74.141/neutron
enable_tunneling = True
local_ip = 190.169.74.142
tunnel_types = gre
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFire
wallDriver

chgrp -R neutron /etc/neutron/plugins

Modificar /etc/default/neutron-server con lo siguiente:
NEUTRON_PLUGIN_CONFIG=/etc/neutron/plugins/ml2/ml2_conf.ini"

Modificar /etc/init/neutron-plugin-openvswitch-agent.conf con lo siguiente:
exec start-stop-daemon –start –chuid neutron –exec /usr/bin/neutron-openvswitch-
agent – –config-file=/etc/neutron/neutron.conf –config-file=/etc/neutron/plugins
/ml2/ml2_conf.ini –log-file=/var/log/neutron/openvswitch-agent.log

Modificar /etc/neutron/dhcp_agent.ini con lo siguiente:
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
use_namespaces = True

Modificar /etc/neutron/l3_agent.ini con lo siguiente:
interface_driver = neutron.agent.linux.interface.OVSInterfaceDriver
use_namespaces = True

Modificar /etc/neutron/metadata_agent.ini con lo siguiente:
auth_url = http://190.169.74.141:5000/v2.0
auth_region = RegionOne
admin_tenant_name = service
admin_user = neutron
admin_password = password
nova_metadata_ip = 190.169.74.141
nova_metadata_port = 8775
metadata_proxy_shared_secret = helloOpenStack

Crear los puentes de acceso a las máquinas virtuales
ovs-vsctl add-br br-int
ovs-vsctl add-br br-ex
ovs-vsctl add-port br-ex eth1

7.4. Nodo de cómputo

Instalar los paquetes del nodo de cómputo apt-get install -y kvm libvirt-bin pm-utils openvswitch-datapath-dkms nova-compute-kvm neutron-plugin-openvswitch-agent python-mysqldb

Configurar Nova Modificar /etc/nova/nova.conf con lo siguiente:
api_paste_config = /etc/nova/api-paste.ini
auth_strategy = keystone
novncproxy_base_url=http://190.169.74.141:6080/vnc_auto.html
vnc_enabled = true
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = 190.169.74.141
rpc_backend = nova.rpc.impl_kombu
rabbit_host = 190.169.74.141

network_api_class=nova.network.neutronv2.api.API
neutron_url=http://190.169.74.142:9696
neutron_auth_strategy=keystone
neutron_admin_tenant_name=service
neutron_admin_username=neutron
neutron_admin_password=password
neutron_admin_auth_url=http://190.169.74.141:5000/v2.0
firewall_driver=nova.virt.firewall.NoopFirewallDriver
security_group_api=neutron
linuxnet_interface_driver=nova.network.linux_net.LinuxOVSInterfaceDriver
libvirt_vif_driver = nova.virt.libvirt.vif.LibvirtGenericVIFDriver
service_neutron_metadata_proxy = True
neutron_metadata_proxy_shared_secret = helloOpenStack
connection = mysql://nova:password@190.169.74.141/nova

Modificar /etc/nova/api-paste.ini con lo siguiente:
auth_host = 190.169.74.141
auth_tenant_name = service
auth_user = nova
auth_password = password

Crear los puentes de acceso a las máquinas virtuales

Configurar el agente de Neutron

Modificar /etc/neutron/neutron.conf con lo siguiente:
core_plugin = neutron.plugins.ml2.plugin.Ml2Plugin
service_plugins = neutron.services.l3_router.l3_router_plugin.L3RouterPlugin
api_paste_config = /etc/neutron/api-paste.ini
allow_overlapping_ips = True
rabbit_host = 190.169.74.141
auth_host = 190.169.74.141
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = neutron

admin_password = password
signing_dir = /var/lib/neutron/keystone-signing

Modificar /etc/neutron/api-paste.ini con lo siguiente:
paste.filter_factory =
keystoneclient.middleware.auth_token:filter_factory
auth_host = 190.169.74.141
admin_tenant_name = service
admin_user = neutron
admin_password = password

mkdir /etc/neutron/plugins/ml2

Modificar /etc/neutron/plugins/ml2/ml2_conf.ini con lo siguiente:
tenant_network_types = gre
mechanism_drivers = openvswitch,linuxbridge
tunnel_id_ranges = 1:1000
sql_connection = mysql://neutron:password@190.169.74.141/neutron
enable_tunneling = True
local_ip = 190.169.74.144
tunnel_types = gre
root_helper = sudo /usr/bin/neutron-rootwrap /etc/neutron/rootwrap.conf
firewall_driver = neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver

chgrp -R neutron /etc/neutron/plugins

Modificar /etc/init/neutron-plugin-openvswitch-agent.conf con lo siguiente:
exec start-stop-daemon –start –chuid neutron –exec /usr/bin/neutron-openvswitch-
agent – –config-file=/etc/neutron/neutron.conf –config-file=/etc/neutron/plugins
/ml2/ml2_conf.ini –log-file=/var/log/neutron/openvswitch-agent.log

restart neutron-plugin-openvswitch-agent

7.5. Configuración lógica

Cargar las credenciales del administrador y ejecutar los siguientes comandos en el nodo de control.

Crear la red externa y el conjunto de direcciones IP flotantes

neutron net-create Red-Externa – –router:external=True

neutron subnet-create Red-Externa –name Subred-Externa –disable-dhcp –allocation-pool start=190.169.74.32,end=190.169.74.63 –gateway 190.169.74.254 190.169.74.0/24 –dns_nameservers list=true 190.169.30.2 190.169.94.5

Crear un proyecto, usuario, y topología de red interna

keystone tenant-create –name prueba

neutron router-create Router-prueba –tenant-id ID-prueba

neutron router-gateway-set Router-prueba Red-Externa

neutron net-create –tenant-id ID-prueba Red-prueba-01

neutron subnet-create –tenant-id ID-prueba Red-prueba-01 10.5.5.0/24 –gateway 10.5.5.1 –name Subred-prueba-01

neutron router-interface-add ID-Router-prueba ID-Subred-prueba-01

keystone user-create –name=prueba –pass=prueba –tenant-id ID-prueba –email=prueba@ucv.ciens.ve

keystone user-role-add –tenant=prueba –user=prueba –role=Member