

PROTOTIPO DE COMPUTACIÓN EN LA
NUBE PARA LA ESCUELA DE INGENIERÍA
DE SISTEMAS E INFORMÁTICA.

por

JOSE DAVID SILVA LORA
JERSON ALBEY VILLAMIZAR JAIMES

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2016

PROTOTIPO DE COMPUTACIÓN EN LA NUBE PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA.

por

JOSE DAVID SILVA LORA
JERSON ALBEY VILLAMIZAR JAIMES

Trabajo de grado presentado como requisito
para optar al título de

INGENIERO DE SISTEMAS

Director: MSc. Manuel Guillermo Flórez Becerra

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2016

*A mi madre Ledys del Carmen y mi padre Alberto Santiago,
que me dieron ejemplo y quienes hicieron quien soy.*

José David Silva Lora

Primeramente, le dedico este trabajo a Dios por ser mi guía espiritual que me conduce siempre hacia el camino del bien y éxito. Por darme el mejor regalo que es tener a mis padres con vida y mucha salud, solo él sabe lo importante que son ellos para mí.

También dedico este trabajo a mi familia, por acompañarme incondicionalmente a lograr las metas y objetivos propuestos. A mis padres por todo lo que me han dado en esta vida, especialmente cada uno de sus consejos y por estar a mi lado en los momentos más difíciles de mi vida brindándome todo su apoyo.

A mi tío John Jairo Carrillo porque estuvo cuando más lo necesitaba y me ayudó a tomar conciencia de la vida, sé que donde se encuentre va a estar orgulloso de mí.

A los docentes que me han acompañado durante el largo camino, brindándome siempre su orientación con profesionalismo ético en la adquisición de conocimientos y afianzando mi formación como estudiante universitario.

A mis amigos, quienes se convirtieron en mi familia Cesar David Vasquez, Julian Andres Martinez, Jose David Silva, Yeny Portilla, Jerson Florez, Leidy Carrillo que poco a poco me enseñaron valores y costumbres que me han formado como persona, muchas gracias por hacerme sentir como en casa cuando los necesité.

Jerson Albey Villamizar Jaimes

Índice general

1	Introducción	12
1.1.	La computación en la nube	12
1.2.	Información del proyecto	13
1.2.1.	Descripción del problema	13
1.2.2.	Justificación	14
1.2.3.	Viabilidad	14
1.2.4.	Metodología	14
1.2.5.	Objetivos	15
2	Marco teórico	17
2.1.	Virtualización	17
2.1.1.	Historia de la virtualización	17
2.1.2.	Objetivo de la virtualización	18
2.1.3.	Tipos de virtualización	18
2.1.4.	Ventajas de la Virtualización	19
2.1.5.	KVM	19
2.2.	Computación en la Nube	21
2.2.1.	Modelos de Implementación	21
2.2.2.	Modelos de Servicio	22
2.2.3.	Beneficios de la computación en la nube	23
2.2.4.	Alta disponibilidad	24
2.2.5.	Clúster de alta disponibilidad	24
2.2.6.	Configuraciones de alta disponibilidad	24

2.3.	Openstack	26
2.3.1.	Tipos de recurso en OpenStack	30
2.3.2.	Instancias	31
2.3.3.	Volúmenes	33
2.4.	HAProxy	34
2.5.	Linux-HA: Heartbeat	35
2.6.	Bacula	35
2.7.	Análisis y diseño	38
2.7.1.	Estudio de la situación actual	38
2.7.2.	Modelo de alta disponibilidad	40
2.7.3.	Consideraciones iniciales de diseño	40
2.7.4.	Diseño del modelo	41
2.8.	Implementación del modelo	41
2.8.1.	Selección del hipervisor	42
2.9.	Puesta en marcha del modelo	49
2.9.1.	Seguridad	50
2.9.2.	Documentación administrativa	50
2.9.3.	Pruebas de funcionamiento	51
2.10.	Conclusiones	56
2.11.	Recomendaciones	56
	Bibliografía	58
	A Términos y Condiciones de Uso	62
	B GID-CONUSS	65

RESUMEN

TÍTULO: PROTOTIPO DE COMPUTACIÓN EN LA NUBE PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

JOSE DAVID SILVA LORA

JERSON ALBEY VILLAMIZAR JAIMES

PALABRAS CLAVES: Virtualización, Computación en la Nube, Software Libre, IaaS (Infraestructura como Servicio), openstack, kvm, replicación de datos, base de datos distribuida.

DESCRIPCIÓN: La computación en la nube representa un cambio del paradigma de la computación hacia un modelo de prestación de múltiples servicios sobre hardware compartido, así como también un cambio representado en el desplazamiento de las labores de cómputo a dispositivos diferentes a los equipos finales de un usuario, dicho modelo se apoya en el uso de la virtualización para la utilización eficiente de los recursos de hardware con los cuales cuenta un centro de datos. La infraestructura presente en los entornos académicos y las limitaciones en la cantidad de los recursos con los que cuentan, hace que sea necesario buscar formas que permitan utilizar la capacidad no utilizada del hardware para el beneficio de los miembros de estos entornos, es allí, donde la computación en la nube permite un eficiente aprovechamiento de esta capacidad inactiva. En el presente trabajo de investigación se presenta la construcción y posterior implementación de un prototipo de computación en la nube que busca prestar servicios a la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, utilizando el modelo de computación en la nube conocido como IaaS o Infrastructure as a Service, de igual manera, también se presenta un modelo de alta disponibilidad para aplicaciones web, que garantiza la alta disponibilidad del servicio y redundancia de datos, acompañado de un software que permite monitorear la

salud de los servidores en cualquier momento. Es importante resaltar que todo el software utilizado en los procesos anteriormente mencionados es de carácter libre, a pesar de la existencia de software privativo que tiene la misma funcionalidad.

ABSTRACT

**TITLE: CLOUD COMPUTING INFRASTRUCTURE PROTOTYPE FOR
SCHOOL OF SYSTEMS ENGINEERING AND COMPUTING**

JOSE DAVID SILVA LORA

JERSON ALBEY VILLAMIZAR JAIMES

KEYWORDS: Virtualization, High Availability, Cloud Computing, IaaS (Infrastructure as a Service), Open Source, OpenStack, kvm, data replication, distributed database.

DESCRIPTION: The cloud computing represents a paradigm shift in computing towards a model of providing multiple services on shared hardware, as well as a shift represented in the displacement of the computing tasks to different user end devices equipment, this model is based on the use of virtualization for efficient use of hardware resources with which has a data center. This in academic environments and infrastructure constraints on the amount of resources at their disposal, makes it necessary to find ways to allow use the unused hardware capacity for the benefit of members of these environments, it is there where cloud computing allows efficient use of this idle capacity. In the present research work it presents construction and subsequent implementation of a cloud computing prototype that seeks to serve the School of Systems Engineering of the Universidad Industrial de Santander, using the cloud computing model known as IaaS or Infrastructure as a Service, similarly, a high availability model for web applications, ensuring high availability service and redundancy data with health software monitor that allows you to see the health of servers at any time. It is important to note that all software used in the above process is open source, even though the existence of proprietary software that has the same functionality.

Introducción

1.1. La computación en la nube

La computación en la nube se ha convertido en un concepto muy importante en los últimos años, ya que los servicios prestados a través de hardware, software están siendo utilizados por una amplia gama de clientes[1], éste, representa un cambio en la forma en la cual se utilizan los centros de datos y un cambio de paradigma en desarrollo acerca de la computación y la manera en la cual se usa y se accede a ésta [2].

Los modelos de prestación de servicios apoyados en la computación en la nube han generado una industria que mueve miles de millones de dólares al año y a pesar de ser un paradigma con un par de años en el mercado, prácticamente considerado en su infancia, se ha posicionado con sus prestaciones, alta usabilidad y la posibilidad de usar servicios bajo demanda, como una de las tecnologías más prometedoras a corto plazo[3].

El campo de la computación en la nube, ha sido investigado por la academia desde sus inicios, pero debido a la etapa en la cual se encuentra, en la literatura existente es fácil hallar información contradictoria acerca de los conceptos fundamentales que lo rodean [4]; a pesar de eso, hay cada vez un mayor consenso en dichos conceptos gracias al gran potencial futuro de esta tecnología y a los esfuerzos realizados por desarrollar estándares por parte de múltiples entidades.

Las soluciones de código abierto presentan una oportunidad para que organizaciones interesadas en experimentar con la computación en la nube puedan hacerlo, sin necesidad de grandes inversiones iniciales de capital, ni infraestructura, ya que pueden hacer uso de sus equipos existen-

tes y determinar cuáles son sus necesidades respecto a este nuevo paradigma, que ofrece ventajas como proporcionar enormes cantidades de recursos informáticos, y la capacidad para almacenar grandes cantidades de datos.[2]

En el presente trabajo de investigación se presenta la construcción y posterior implementación de un prototipo de computación en la nube que busca prestar servicios a la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander, utilizando el modelo de computación en la nube conocido como IaaS o Infrastructure as a Service, de igual manera, también se presenta un modelo de alta disponibilidad para aplicaciones web, que garantiza la alta disponibilidad del servicio y redundancia de datos, acompañado de un software que permite monitorear la salud de los servidores en cualquier momento.

Con los parámetros mencionados se busca en general garantizar la optimización de la labor administrativa en los nuevos equipos de servidores, haciendo posible que se hagan relevos de administradores de manera oportuna y correcta, teniendo en cuenta que la labor administrativa se lleva a cabo por estudiantes de pregrado de último nivel de Ingeniería de Sistemas.

1.2. Información del proyecto

1.2.1. Descripción del problema

El grupo de investigación GID-CONUSS cuenta con una infraestructura de servidores; La cual era necesario reestructurar y reconfigurar debido a deficiencias en el rendimiento, por tal motivo, los servicios que estos servidores brindaban a la comunidad universitaria se vieron afectados, presentando problemas en el respaldo de la información almacenada, ocasionando en algunos casos pérdida completa o parcial de dicha información.

En el desarrollo del proyecto se diseñó y construyó un prototipo de computación en la nube basado en OpenStack (tecnología que se impone sobre la existente) y un modelo de alta disponibilidad para aplicaciones web, que permite mejorar la funcionalidad y seguridad de la infraestructura con miras a establecer un sistema que soporte la extensión de nuevo hardware y servicios, creando en el proceso un modelo que sea flexible y escalable.

1.2.2. Justificación

En el desarrollo del proyecto se construyó un modelo de Computación en la Nube y un otro de alta disponibilidad para aplicaciones WEB que lo requieran, logrando así que la Escuela de Ingeniería de Sistemas e Informática preste servicios de IaaS a la comunidad estudiantil, aprovechando las ventajas de la computación en la nube y la alta disponibilidad en sus diferentes proyectos. Se busca que la Escuela de Ingeniería de Sistemas e Informática cuente con una plataforma en la cual se puedan desarrollar proyectos de diversas áreas sobre un nuevo paradigma de trabajo para estar al día con tecnologías que le permitan cumplir su misión institucional.

1.2.3. Viabilidad

La Escuela de Ingeniería de Sistemas e Informática cuenta actualmente con equipos disponibles para la realización de diferentes proyectos, entre los cuales se encuentra el presente trabajo de investigación, con el cual se busca hacer un uso eficiente de los recursos tecnológicos disponibles. Respecto al software disponible para el desarrollo del presente proyecto, se cuenta con múltiples herramientas de software libre y comunidades de desarrollo en las cuales se puede apoyar la realización del prototipo; estas herramientas tienen la ventaja de no originar mayores costos económicos. Las licencias de software de los diferentes sistemas operativos se implementarán de acuerdo a los acuerdos que maneja actualmente la universidad. En cuanto al personal, el grupo GID-CONUSS (ver Anexo B) cuenta con un alto grado de experiencia en el manejo de servidores, el cual será fundamental para optimizar muchas de las características de las cuales se espera disponer al final del presente proyecto.

1.2.4. Metodología

Para el desarrollo de este proyecto se llevará a cabo la investigación de la infraestructura como servicio (IaaS) y modelos de alta disponibilidad, para realizar un análisis de la documentación de diferentes autores confiables de computación en la nube, después de este proceso iniciaremos con el ciclo desde la recolección de requisitos, diseño, construcción y evaluación de la infraestructura en el cual se realizarán las respectivas pruebas con el fin de optimizar los servicios y procesos prestados, hasta llegar al punto donde sea viable su implementación en CloudEISI para mejorar la funcionalidad de la infraestructura.

1.2.5. Objetivos

Objetivo general

DESARROLLAR E IMPLEMENTAR UN PROTOTIPO DE COMPUTACIÓN EN LA NUBE PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA, HACIENDO USO DE LA TECNOLOGÍA OPENSTACK Y KVM PARA PRESTAR SERVICIOS VIRTUALES BAJO DEMANDA.

Objetivos específicos

1. Administrar la actual infraestructura de alta disponibilidad y computación en la nube del grupo GID-CONUSS. Entre las labores a cumplir se encuentran:
 - Entrenamiento en el funcionamiento y uso de la infraestructura.
 - Realizar tareas de mantenimiento programadas y copias de seguridad.
 - Recuperación del sistema o servicios en caso de fallo.
 - Mantenimiento y administración de las unidades de almacenamiento.
 - Realizar tareas relacionadas con atención a los usuarios de los diferentes servicios.
 - Registro de las solicitudes de servicios usando formularios del grupo.
 - Implementar servicios virtuales solicitados por la comunidad.
 - Monitoreo y mantenimiento de los servicios virtualizados.
 - Comunicación y Asesoría a usuarios del CloudEisi.
2. Realizar las actualizaciones necesarias en todos los servidores de GID-CONUSS que permitan el mejoramiento de las funcionalidades del CloudEISI.
 - Cambiar la estructura de la tabla de particiones.
 - Instalación de Ubuntu Server.
 - Instalación, prueba de Herramientas y software de virtualización.
3. Implementar en los servidores del grupo CONUSS una infraestructura de computación en la nube usando OpenStack y KVM.
4. Garantizar para los servicios virtualizados su funcionalidad, seguridad, eficiencia en la nueva infraestructura.

5. Diseñar e implementar un modelo de máquinas virtuales con alta disponibilidad para aplicaciones web , probar su funcionalidad y eficiencia aplicado a las del Aula Virtual MEIWEB.
6. Hacer análisis de las medidas de seguridad actuales del CloudEISI, y buscar soluciones que permitan que sean más confiables para los usuarios, y disminuir los riesgos causados por ataques externos.
7. Realizar la documentación de la infraestructura del CloudEISI.
8. Escribir un artículo con los resultados del proyecto y las recomendaciones para próximas investigaciones y desarrollos.
9. Entrenar y asesorar los relevos administrativos para garantizar la continuidad del proyecto, mediante un curso de mínimo 20 horas durante 5 días.

Marco teórico

Para entender y comprender este proyecto, desde su enfoque hasta su funcionamiento se definirán los siguientes conceptos.

2.1. Virtualización

2.1.1. Historia de la virtualización

Durante la década de los 60 los equipos de informática de muchas empresas y entidades tenían un problema similar: contaban con super-computadoras o mainframes de alto rendimiento que deseaban particionar lógicamente, o utilizar para múltiples tareas simultáneas. Es por esto que IBM desarrolló un método para crear múltiples particiones lógicas (similar a lo que conocemos hoy como máquinas virtuales) las cuales trabajaban independientemente una de las otras, y cada una utilizando los recursos provistos por el mainframe. Ya para la década de los 80 y con la llegada de las relativamente económicas maquinas x86, comenzó una nueva era de microcomputadoras, aplicaciones cliente-servidor, y computación distribuida; en donde los enormes y potentes mainframes con mil y una tareas y utilidades en una sola caja gigantesca se comenzaron a cambiar por relativamente pequeños servidores y computadoras personales de arquitectura x86, con una caja diferente para cada uso, lo que se convirtió rápidamente en el estándar de la industria. Debido a esto, una vez más, el tema de la virtualización vuelve a quedar prácticamente en el olvido y no es hasta finales de la década de los 90 que gracias al alto desarrollo del hardware volvemos a caer en un predicamento similar al que estábamos en los años 60: el hardware existente es altamente eficiente, y utilizar cada caja para una sola aplicación sería un desperdicio de recursos, espacio, energía y dinero; y tampoco es conveniente asignar múltiples usos o instalar varias aplicaciones en un solo servidor convencional, por más de una razón. Es por esto que vuelve a resurgir la idea

de dividir el hardware, de manera tal que funcione como múltiples servidores independientes pero compartiendo los recursos de un mismo servidor físico. Y es de aquí que nace lo que hoy todos conocemos como Virtualización.[9]

2.1.2. Objetivo de la virtualización

Su objetivo es la simulación de varias computadoras lógicas, a partir de una computadora real. Para ello existen varias técnicas, que van desde la virtualización del hardware (CPU, memoria y dispositivos) hasta la separación de los procesos utilizando un único kernel. [10]

La virtualización nos permite, entre otras cosas:

- Alojarse varios servidores en una única computadora física. Esto nos permite optimizar el uso de recursos (CPU, memoria, almacenamiento).
- Simplificar la realización de copias de respaldo (backup) y su restablecimiento.
- Todo un servidor virtual puede ser un único archivo.
- Migrar fácilmente servidores entre distintas computadoras.
- Incrementar la seguridad, utilizando servidores aislados para tareas diferentes.

2.1.3. Tipos de virtualización

Si bien el concepto de virtualización es abstracto y permite su uso sobre múltiples recursos, para la computación en la nube es clave la virtualización de hardware, esta se ha conseguido por dos métodos, virtualización completa y paravirtualización.

Paravirtualización es una técnica en la cual el hipervisor presenta un API al sistema operativo invitado y este realiza las peticiones a través de él al sistema operativo huésped, esto limita la cantidad de sistemas operativos virtualizables a solo aquellos que tengan soporte desde su núcleo.

Virtualización completa es una técnica que se basa en translación binaria para atrapar y virtualizar la ejecución de cierto tipo de instrucciones sensibles que no pueden ser virtualizadas, de esta manera las instrucciones críticas son reemplazadas por el hipervisor y emuladas por software. [11]

KVM es un ejemplo de virtualización completa. Este método de virtualización se adapta muy

bien a los entornos de la nube. Desde un punto de vista práctico la virtualización completa es una técnica en que una instalación completa de una máquina se ejecuta en otra. El resultado es un sistema en que todo el software corre sobre el servidor que está dentro de una máquina virtual.

2.1.4. Ventajas de la Virtualización

1. Reutilización de hardware existente y optimizar el aprovechamiento de todos los recursos de hardware
2. Rápida incorporación de nuevos recursos para los servidores virtualizados
3. Reducción de los costes de espacio y consumo necesario de forma proporcional al índice de consolidación logrado
4. Administración global centralizada y simplificada
5. Mejora en los procesos de clonación y copia de sistemas
6. Aislamiento: un fallo general de sistema de una máquina virtual no afecta al resto de máquinas virtuales
7. Balanceo dinámico de máquinas virtuales entre los servidores físicos que componen el pool de recursos, garantizando que cada máquina virtual ejecute en el servidor físico más adecuado y proporcionando un consumo de recursos homogéneo y óptimo en toda la infraestructura.
8. Contribución al medio ambiente por menor consumo de energía en servidores físicos.[12]

2.1.5. KVM

Es la solución de virtualización completa de código abierto líder en hardware x86 y es compatible con los principales sistemas operativos incluyendo Linux y Windows. KVM permite a las empresas ser ágil, proporcionando flexibilidad y escalabilidad robusta que se ajusten a sus demandas específicas del negocio. KVM convierte el núcleo de Linux en un hipervisor aprovechando las características avanzadas de Intel VT-x y el hardware x86 AMD-V, ofreciendo así los niveles de rendimiento sin igual. Además, KVM incorpora características de seguridad Linux incluyendo SELinux (Security-Enhanced Linux) desarrollado por la Agencia de Seguridad de Estados Unidos para añadir controles de acceso, multi-nivel y de seguridad multi-categoría, así como la aplicación

de políticas. Como resultado, las organizaciones están protegidos de máquinas virtuales comprometidas que son aislados y no se puede acceder por cualquier otro proceso. Este progreso se ha visto favorecido por el hecho de que Industria- procesadores x86 estándar y sistemas se han vuelto cada vez más potente, en términos de procesamiento, memoria y E / S, y ahora representan recursos compartidos atractivos. Además, la integración de la tecnología clave de virtualización a nivel de procesador, por tanto Intel (Intel® VT) y AMD® (AMD-V) ha permitido la virtualización estar profundamente integrada a nivel del kernel Linux, proporcionando beneficios significativos en términos de rendimiento, escalabilidad , y la seguridad. [13]

Beneficios de KVM

Costos más bajos

Un análisis hecho por LetterGen, un proveedor de servicio T.I. en Belgica muestra que KVM puede ser un 60-90 % más económico que otras soluciones, al tiempo que ofrece la misma funcionalidad básica. KVM representa una solución verdaderamente abierta que ofrece un valor real en términos de funcionalidad. [27]

Rendimiento y escalabilidad de la empresa Superior

En las últimas evaluaciones de rendimiento, KVM ha igualado consistentemente con las principales tecnologías patentadas, lo que demuestra los más altos parámetros de rendimiento. SPECvirt diseñado para medir el desempeño de los servidores del centro de datos que se utilizan para la virtualización de servidores, muestra que KVM ofrece un nivel de rendimiento más alto que VMware en los 2, 4, 8 categorías de socket. En términos sencillos, cuanto mayor sea el rendimiento, más cargas de trabajo que se pueden ejecutar en esa solución de virtualización. [14]

Seguridad avanzada

Debido a que está construida encima de Linux, KVM es capaz de utilizar las capacidades avanzadas de seguridad de Security Enhanced Linux (SELinux). Esto permite KVM para proporcionar seguridad control de acceso obligatorio entre las máquinas virtuales. [14]

Alta Calidad de Servicio (QoS)

Linuxó incluye grupos C '(grupos de control), que permiten a las políticas de QoS de grano fino para procesos de Linux. KVM es parte del núcleo de Linux, una máquina virtual no es diferente de cualquier otro programa que se ejecuta en Linux. Así que los administradores pueden establecer

umbrales definidos para la CPU, memoria, red y disco I/O, lo que garantiza la calidad de servicio para máquinas virtuales dada. [14]

Ecosistema abierto

arquitectura de código abierto de KVM le da más opciones, a diferencia de las soluciones propietarias que pueden limitar el tipo de socios y, por lo tanto, la elección de soluciones interoperables disponibles. [14]

2.2. Computación en la Nube

La computación en la nube se refiere a las aplicaciones y los servicios que se ejecutan en una red distribuida utilizando recursos virtualizados, a los que se accede a través de protocolos de internet y estándares de uso de red. Según la IEEE computer society, es un paradigma en el que la información se almacena de manera permanente en servidores de internet y se envía a caché temporales de cliente, lo que incluye equipos de escritorio, centros de ocio, portátiles, etc. Esto se debe a que, pese a que las capacidades de las computadoras han mejorado, parte de su potencia se desaprovecha, al ser máquinas de propósito general.[5]

Cloud Computing es un nuevo modelo de prestación de servicios de negocio y tecnología, que permite al usuario acceder a un catálogo de servicios estandarizados y responder a las necesidades de su negocio, de forma flexible y adaptativa, en caso de demandas no previsibles, pagando únicamente por el consumo efectuado.

2.2.1. Modelos de Implementación

Como modelos de implementación se refiere a la ubicación y administración de la infraestructura de la nube, es decir, define el propósito de la nube y la naturaleza del modo en que se ubica, la definición de los cuatro modelos de implementación son los siguientes:

Nube pública: la infraestructura de nube pública está disponible para el uso público o, de manera alternativa, para un grupo industrial grande y es propiedad de una organización que vende servicios en la nube. [6]

Nube privada: la infraestructura de la nube privada funciona para el uso exclusivo de una organización. [6]

Nube híbrida: una nube híbrida combina varias nubes (privada, comunitaria o pública), donde se mantienen sus identidades pero se vinculan como una unidad.[4]



Figura 2.1: Modelos de implementación de cloud computing.

2.2.2. Modelos de Servicio

Infraestructura como servicio (IaaS, Infrastructure as a Service) ofrece hardware y software, y equipos para ofrecer entornos de aplicaciones de software con un modelo de precios basado en el uso de los recursos.[7]



Figura 2.2: Identidad visual de algunas empresas que ofrecen servicios IaaS.

Plataforma como servicio (PaaS, Platform as a Service) ofrece un entorno integrado de alto nivel para desarrollar, probar e implementar aplicaciones personalizadas. [7]



Figura 2.3: Identidad visual de algunas empresas que ofrecen servicios PaaS.

Software como servicio (SaaS, Software as a Service) es un modelo de desarrollo de software, donde se alojan las aplicaciones de forma remota por un proveedor de la aplicación o servicio

y puestos a disposición de los clientes a través de internet. SaaS ofrece al cliente importantes beneficios, como la mejora de la eficiencia operativa y reducción de costos. Proporciona software de propósito especial que es accesible de forma remota por los consumidores a través de internet con un modelo de precios basado en el uso. [7]



Figura 2.4: Identidad visual de algunas empresas que ofrecen servicios SaaS.

2.2.3. Beneficios de la computación en la nube

Características que los sistemas de computación en la nube deben ofrecer:

Autoservicio bajo demanda: Un cliente puede aprovisionar recursos informáticos sin necesidad de interactuar con el personal del proveedor del servicio de nube.

Acceso amplio desde la red: El acceso a los recursos de la nube se encuentra disponible sobre la red utilizando métodos estándar de una manera que proporciona acceso independiente de la plataforma a los clientes de todo tipo.

Fondo de recursos: Un proveedor de servicios de nube crea recursos que se reúnen en un sistema compatible con el uso en multitenencia. Quiere decir que los sistemas físicos y virtuales se asignan o reasignan de forma dinámica según sea necesario.

Elasticidad rápida: Los recursos se pueden suministrar de forma rápida y elástica. El sistema puede añadir recursos escalando sistemas o ajustando el número de sistemas, y estas aplicaciones o ajustes pueden ser automáticas o manuales.

Servicio medido: El uso de los recursos del sistema de nube se mide, se audita y se recoge en un informe para el cliente basándose en un sistema de contador.

2.2.4. Alta disponibilidad

Tomando la definición expuesta por el IEEE alta disponibilidad se refiere a la disponibilidad de los recursos en un sistema computacional en el evento de fallo de componentes del sistema. Esto se puede lograr de varias maneras, abarcando todo el espectro que va desde un extremo con soluciones que utilizan hardware personalizado y redundante para garantizar la disponibilidad, hasta el otro extremo con soluciones que ofrecen las soluciones de software que utilizan componentes de hardware genéricos.[8]

2.2.5. Clúster de alta disponibilidad

Es un conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitoreándose entre sí.[15] Su finalidad es poder ser capaz de detectar cualquier fallo de hardware o de software, reiniciar la aplicación en otro nodo y mantener el servicio sin intervención de operador alguno, garantizando la integridad de los datos del clúster. [16]

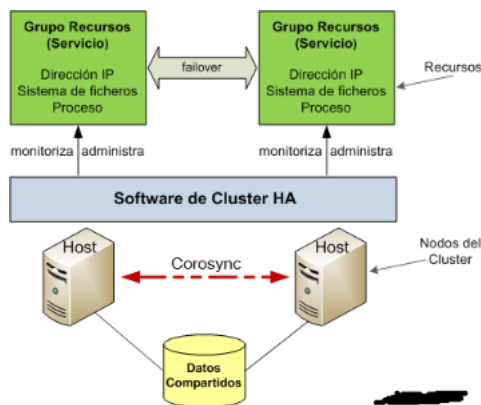


Figura 2.5: Clúster de Alta Disponibilidad de dos Nodos. Fuente: <http://www.lintips.com/>

2.2.6. Configuraciones de alta disponibilidad

Las configuraciones más comunes en entornos de clúster de alta disponibilidad son la configuración activo/activo y la configuración activo/pasivo.

Configuración Activo/Activo

En una configuración activo/activo, todos los servidores del clúster pueden ejecutar los mismos recursos simultáneamente. Es decir, los servidores poseen los mismos recursos y pueden acceder a estos independientemente de los otros servidores del clúster. Si un nodo del sistema falla y deja de

estar disponible, sus recursos siguen estando accesibles a través de los otros servidores del clúster. La ventaja principal de esta configuración es que los servidores en el clúster son más eficientes ya que pueden trabajar todos a la vez. Sin embargo, cuando uno de los servidores deja de estar accesible, su carga de trabajo pasa a los nodos restantes, lo que produce una degradación del nivel global de servicio ofrecido a los usuarios. En la siguiente figura se muestra como ambos servidores están activos, proporcionando un mismo servicio a los diferentes usuarios. Los clientes acceden al servicio o recursos de forma transparente y no tienen conocimiento de la existencia de varios servidores formando un clúster. [17]

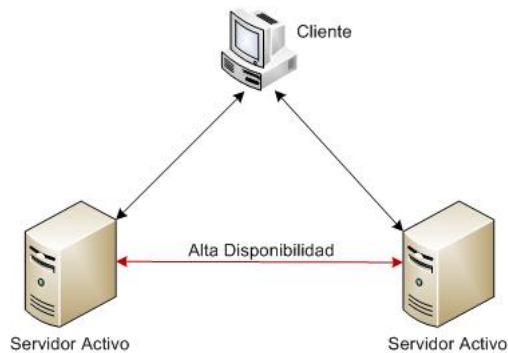


Figura 2.6: Configuración Activo/Activo. Fuente: <http://goo.gl/kdZeyJ>

Configuración Activo/Pasivo

Un clúster de alta disponibilidad, en una configuración activo/pasivo, consiste en un servidor que posee los recursos del clúster y otros servidores que son capaces de acceder a esos recursos, pero no los activan hasta que el propietario de los recursos ya no esté disponible. Las ventajas de la configuración activo/pasivo son que no hay degradación de servicio y que los servicios solo se reinician cuando el servidor activo deja de responder. Sin embargo, una desventaja de esta configuración es que los servidores pasivos no proporcionan ningún tipo de recurso mientras están en espera, haciendo que la solución sea menos eficiente que el clúster de tipo activo/activo. Otra desventaja es que los sistemas tardan un tiempo en migrar los recursos (failover) al nodo en espera. [17]

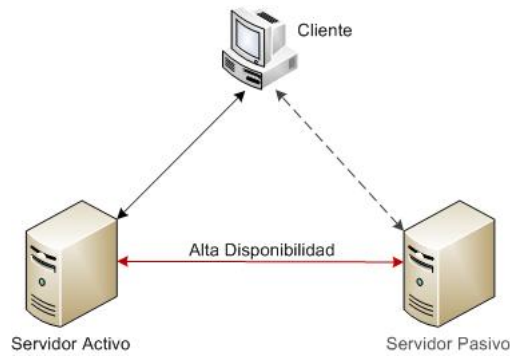


Figura 2.7: Configuración Activo/Pasivo. Fuente: <http://www.ecured.cu/Archivo:Leo3Cluster.JPG>

Las razones para implementar un clúster de alta disponibilidad son

1. Aumentar la disponibilidad.
2. Mejorar el rendimiento.
3. Escalabilidad.
4. Tolerancia a fallos.
5. Recuperación ante fallos en tiempo aceptable.
6. Reducir costes.
7. Consolidar servidores.
8. Consolidar el almacenamiento. [17]

2.3. Openstack

OpenStack [18] es un proyecto de computación en la nube libre y de código abierto que tiene como finalidad ofrecer un conjunto de servicios para proporcionar IaaS. El proyecto es gestionado por la Fundación OpenStack y respaldado por algunas de las empresas más grandes del sector como Google, IBM, Yahoo!, HP, Rackspace o AMD. OpenStack es desarrollado por una comunidad muy activa y tiene un ciclo de lanzamiento de versiones semestral. En Kilo, la versión más reciente de OpenStack (abril 2015), los principales servicios ofrecidos son los siguientes:

Nova (Computación) Este módulo es el núcleo de OpenStack, un controlador de estructura de computación. Se usa para crear máquinas virtuales (instancias) y manejar tareas computacionales. La arquitectura de Nova está diseñada para escalar horizontalmente en hardware estándar y proporcionar integración con sistemas antiguos y tecnologías de terceros.

Cinder (Almacenamiento de bloques)

Es el módulo de almacenamiento de bloques, análogo a la tradicional idea de acceder localizaciones específicas en un disco duro. Cinder maneja lo que se llaman volúmenes de datos persistentes que se pueden usar contra instancias de Nova. El almacenamiento de bloques es apropiado para escenarios donde un alto rendimiento es importante.

Glance (Imágenes)

Proporciona servicios de imágenes. En este caso imagen se refiere a copias virtuales de discos. Glance permite usar dichas imágenes como plantilla a la hora de desplegar una nueva instancia en Nova. Por ejemplo, se podría utilizar una imagen del sistema operativo Ubuntu para crear una máquina virtual (instancia de Nova) con el mencionado sistema instalado.

Swift (Almacenamiento de objetos)

Ofrece servicio de almacenamiento redundante y escalable. La idea detrás de Swift es referirse a los objetos y archivos mediante un identificador en lugar de hacerlo como se ha hecho tradicionalmente mediante su localización en el disco duro. Esto hace que la escalabilidad sea fácil ya que, al poder almacenar los archivos y objetos en múltiples discos dejando que el servicio se encargue de su identificación, localización y redundancia, se permite que el desarrollador no tenga que preocuparse por problemas de espacio.

Neutron (Redes)

Es el módulo que gestiona las redes y direcciones IP de las máquinas virtuales. Asegura que la red no presente problemas y que las instancias puedan comunicarse entre sí rápida y eficientemente.

Keystone (Identificación)

Proporciona servicio de autenticación y autorización usado por todos los demás servicios de OpenStack para validar las operaciones de un usuario. Es, esencialmente una lista centralizada de todos los usuarios de una instalación de OpenStack mapeada contra todos los servicios. A través del módulo keystone, el usuario adquiere permisos para hacer uso de cada servicio de OpenStack que los administradores le permitan usar.

Ceilometer (Telemetría)

Proporciona servicios de facturación. Mantiene la cuenta del uso que ha hecho cada usuario de cada uno de los servicios de OpenStack a través de contadores. Se utiliza para establecer la facturación del cliente en un período dado.

Heat (Orquestación)

Permite a los desarrolladores almacenar los requerimientos de recursos necesarios para una aplicación dada en la nube. De esta manera, ayuda a definir la infraestructura necesaria para ejecutar un servicio en la nube.

Horizon (Panel de administración)

Horizon es la única interfaz gráfica disponible para OpenStack actualmente. Desde este dashboard los usuarios pueden acceder a la mayoría de los servicios ofrecidos por OpenStack, pero no a todos. Permite administrar los recursos virtualizados de un usuario (o de todo el Cloud en caso de ser administrador). El principal problema de esta interfaz es que no es modificable y personalizable a nivel de funcionalidad sin acceder al código fuente. También tiene la desventaja de que está pensada para gestionar una única instalación de OpenStack y no es compatible con la gestión de múltiples regiones en la manera que están implementadas en FIWARE a través de múltiples instalaciones federadas de OpenStack.

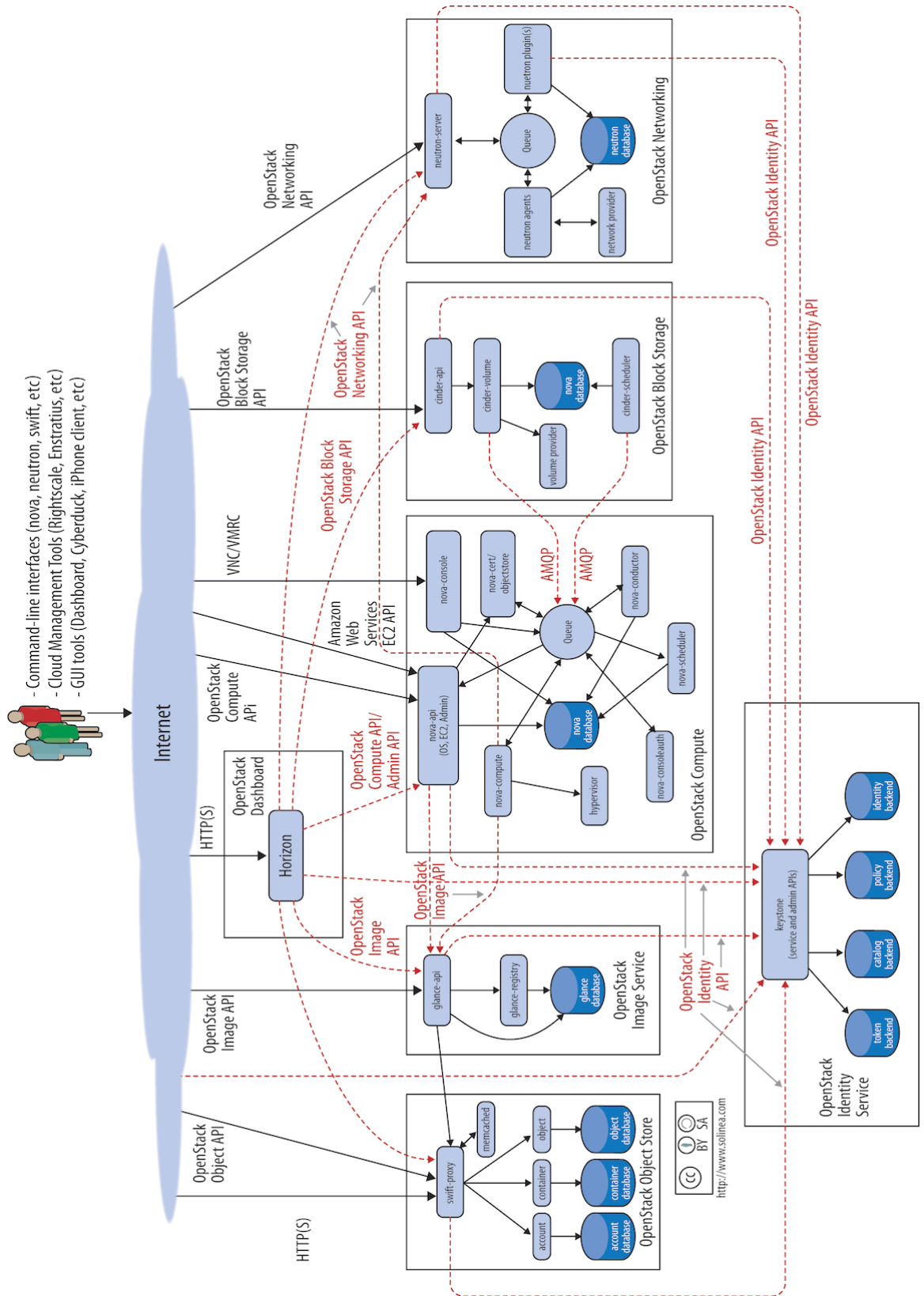


Figura 2.8: Diagrama de la arquitectura OpenStack. Fuente: <https://goo.gl/t35G2s>

2.3.1. Tipos de recurso en OpenStack

En una instalación de OpenStack se pueden gestionar varios tipos de recursos que pueden ser imágenes, instancias, flavors, redes, volúmenes, IPs, grupos de seguridad y otros. En el contexto de este Trabajo de fin de Grado se van a gestionar en mayor medida los recursos de imágenes, instancias y volúmenes.

Imágenes

Una imagen es un tipo de archivo que contiene la estructura de un disco o una unidad de almacenamiento al completo. Una imagen de disco normalmente se produce haciendo una copia completa sector a sector del medio origen, replicando así la estructura y contenidos del dispositivo.

Las imágenes de disco suelen utilizarse con propósitos de virtualización. La virtualización de una imagen es interpretada por un monitor de máquina virtual como un disco. Esto permite la instalación de sistemas operativos a través de la información almacenada en este tipo de archivos.

Formatos

Los formatos más importantes son los siguientes:

- **Raw:** Es el formato más simple, equivalente a un archivo de bloques de un disco, creado a partir de la copia directa del disco por ejemplo con un comando `dd`. Este formato es soportado por los hypervisors KVM y Xen.
- **qcow2:** QEMU Copy-On-Write version 2. Este formato se usa normalmente con el hypervisor KVM y tiene algunas características que no posee el formato Raw como el uso de la representación sparse, que genera ficheros de imagen más pequeños o el soporte para snapshots.

Almacenamiento

Las imágenes en OpenStack son gestionadas por el servicio Glance, que ofrece una API a través de la cual los usuarios y los administradores pueden realizar acciones sobre sus imágenes o crear nuevas. Glance se encarga también de la gestión del almacenamiento de las imágenes.

El servicio de imágenes de Glance soporta modos de almacenamiento en back-end, algunos de ellos son los siguientes:

Sistema de ficheros

Es el modo por defecto y el más simple ya que solo escribe las imágenes en el propio sistema de ficheros.

- **Amazon S3:** En este modo las imágenes se almacenan utilizando el servicio S3 de Amazon, que es el equivalente a Swift en AWS.
- **HTTP:** En este modo se almacenan las imágenes en un servidor en internet y se accede a ellas a través de HTTP. Este almacenamiento solo permite modo lectura.

Estados

Las imágenes gestionadas en Glance tienen un estado asociado que varía en función de la tarea que se esté realizando sobre la imagen o simplemente el estado de almacenamiento en que se encuentran sus datos. Una imagen puede adoptar muchos estados, algunos de ellos son los siguientes:

- **QUEUED:** El identificador de imagen se ha reservado para la imagen pero aún no se han subido datos de imagen a Glance y/o no se indicó explícitamente un tamaño 0 en la creación de la imagen.
- **SAVING:** Los datos de la imagen se están subiendo al servidor de Glance. Una imagen nunca estará en este estado si al crearla se indica una localización en vez de subir un archivo.
- **ACTIVE:** La imagen está completamente lista para ser usada en Glance. Se llega a este estado cuando se acaban de subir los datos de una imagen o si se especifica el tamaño 0 al crearla.
- **DELETED:** Glance aún tiene información de la imagen almacenados pero ya no puede ser usada y será borrada posteriormente.

2.3.2. Instancias

Una instancia en OpenStack es una máquina virtual desplegada en un nodo físico de computación. Cada instancia se lanza desde una imagen de máquina virtual y consigue su capacidad de almacenamiento de volúmenes de almacenamiento de bloques.

Al crearse la instancia, la imagen utilizada como base para la máquina virtual se copia al servidor desde el almacén de imágenes del servicio Glance. Además, al crear la instancia se le pueden asignar volúmenes de almacenamiento o crear volúmenes efímeros en el propio disco del servidor. En

cualquier caso el disco local es el primero al que accede la instancia a través del punto de montaje `/dev/vda`.

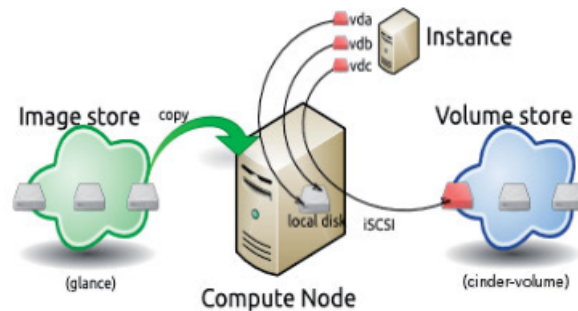


Figura 2.9: Proceso de creación de una instancia. Fuente: http://docs.openstack.org/admin-guide/_images/instance-life-1.png

Las instancias en OpenStack son gestionadas por el servicio Nova, que permite al usuario realizar operaciones sobre sus instancias y crear nuevas a partir de imágenes almacenadas en Glance. Entre las operaciones permitidas por Nova están las de borrado, edición, reinicio software, reinicio hardware, cambio de contraseña, adjunto de un volumen, adjunto de un grupo de seguridad y ajuste del tamaño de una instancia.

Estados

Una instancia en OpenStack tiene tres estados distintos. El primero de ellos es el estado de la máquina virtual (VM State). Este estado refleja la estabilidad de las llamadas a la API por parte del usuario. Algunos de los valores son los siguientes:

- **INITIALIZED:** La máquina virtual acaba de ser creada en la base de datos pero aún no se ha construido.
- **ACTIVE:** La instancia está ejecutándose correctamente con la imagen especificada durante su creación.
- **PAUSED:** La instancia se encuentra pausada.
- **SUSPENDED:** La instancia ha sido suspendida y tiene un snapshot válido de la memoria.
- **ERROR:** Ha ocurrido un error irrecuperable. En este estado la única operación que se puede realizar sobre la instancia es la de borrado.

Seguridad y red

Si se añade la instancia a una red de OpenStack, se le puede asignar una dirección pública a través de la cual el usuario pueda acceder vía SSH con la clave RSA indicada al crear la instancia. Además, una instancia puede tener asignado un grupo de seguridad que denota los puertos que son accesibles y desde qué direcciones. El usuario puede editar este grupo de seguridad a través de la API de Nova.

2.3.3. Volúmenes

Un volumen en OpenStack es una unidad de almacenamiento por bloques que puede ser conectada a una instancia para proporcionar capacidad extra de almacenamiento de la misma manera que lo haría un disco duro externo. Cada volumen solo puede estar conectado a una instancia a la vez. Los volúmenes son gestionados por el servicio Cinder de OpenStack, que permite al usuario realizar operaciones sobre sus volúmenes así como crear nuevos. Algunas de las operaciones permitidas por cinder son: editar, borrar, y hacer un snapshot de un volumen.

Datos

Las soluciones de alta disponibilidad enfocadas a los datos son fundamentales y por lo general es necesaria su integración con los sistemas de alta disponibilidad enfocados a servicios. El objetivo que se persigue con la alta disponibilidad enfocada a los datos es mantener en varias localizaciones una serie de datos, los cuales no presentan problemas de consistencia ni integridad entre ellos. En GNU/Linux se dispone de varias herramientas software que cumplen este cometido.

El proyecto destinado a ofrecer alta disponibilidad en datos es:

DRBD (Distributed Replicated Block Device). Software de replicación de dispositivos de bloque formando un RAID 1 a través de la red, es decir, replica los datos en distintas localizaciones. Datos de un sistema de archivos, de una base de datos, etc. Algunas de sus características son una replicación en tiempo real de manera continua, transparencia en las aplicaciones que estén almacenando datos en la unidad, posibilidad de recuperación de los datos ante un desastre, etc.

Las principales características de DRBD son:

- **Tiempo real.** La replicación se realiza de manera continua.
- **Transparencia.** Las aplicaciones que estén almacenando datos en la unidad no sabrán que está se está replicando en varias localizaciones.

- **Síncrono o asíncrono.** Con replicación síncrona la aplicación que escribe es notificada que la escritura se ha completado solo después de que ésta haya sido replicada. Con replicación asíncrona la notificación de que la escritura se ha completado será entregada a la aplicación cuando ésta haya sido realizada de manera local, esto es, antes de que la escritura haya sido propagada al resto de ordenadores.[19]

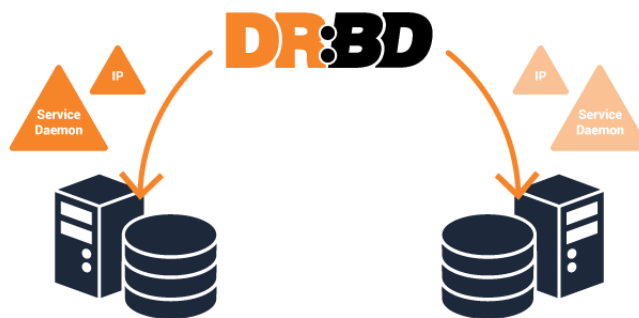


Figura 2.10: Esquema de un sistema que usa DRBD. Fuente <https://goo.gl/MJOiT4>

2.4. HAProxy

Es un popular software de código abierto TCP / HTTP equilibrador de carga y la solución de proxy que se puede ejecutar en Linux, Solaris y FreeBSD. Su uso más común es para mejorar el rendimiento y la fiabilidad de un entorno de servidor mediante la distribución de la carga de trabajo a través de múltiples servidores (por ejemplo web, aplicación, base de datos). Se utiliza en muchos entornos de alto perfil, incluyendo: GitHub, Imgur, Instagram y Twitter.[28]

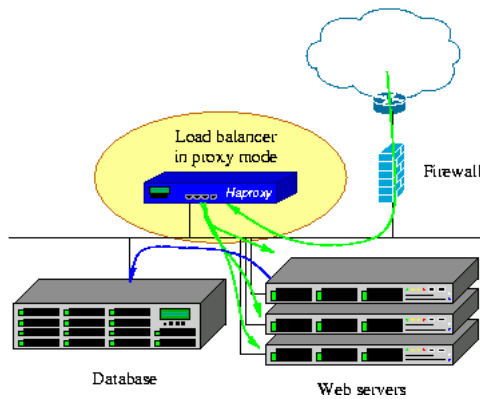


Figura 2.11: Esquema de un sistema que usa HAProxy. Fuente <http://www.haproxy.org/img/haproxy-pmode.png>

2.5. Linux-HA: Heartbeat

Heartbeat es un software que ofrece alta disponibilidad a determinados recursos mediante la creación y mantenimiento de un clúster compuesto por una serie de nodos. Los recursos se ejecutan y se mueven entre los distintos nodos ya sea por motivos de fallo o simplemente por motivos de administración. Por ejemplo un recurso podría ser el servicio Web de una entidad de tal manera que si sucede un evento como el fallo de la fuente de alimentación del nodo que en ese momento ejecuta el servicio, el servicio Web o recurso se mueve hacia otro nodo del cluster por lo que el servicio en cuestión se continúa ofreciendo de manera ininterrumpida a pesar del suceso.[20]

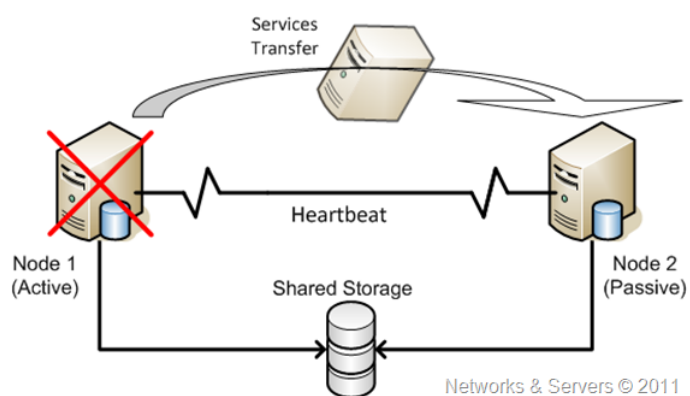


Figura 2.12: Esquema de un sistema que usa Heartbeat. Fuente <http://goo.gl/b3TJWM>

2.6. Bacula

Bacula es un conjunto de código abierto, programas informáticos que le permiten (o el administrador del sistema) para la gestión de backup, recuperación y verificación de los datos informá-

ticos a través de una red de ordenadores de diferentes tipos. Bacula es relativamente fácil de usar y muy eficiente, al tiempo que ofrece muchas características avanzadas de gestión de almacenamiento que hacen que sea fácil de encontrar y recuperar archivos perdidos o dañados. En términos técnicos, es un código abierto, red del programa de copia de seguridad basada.[21]

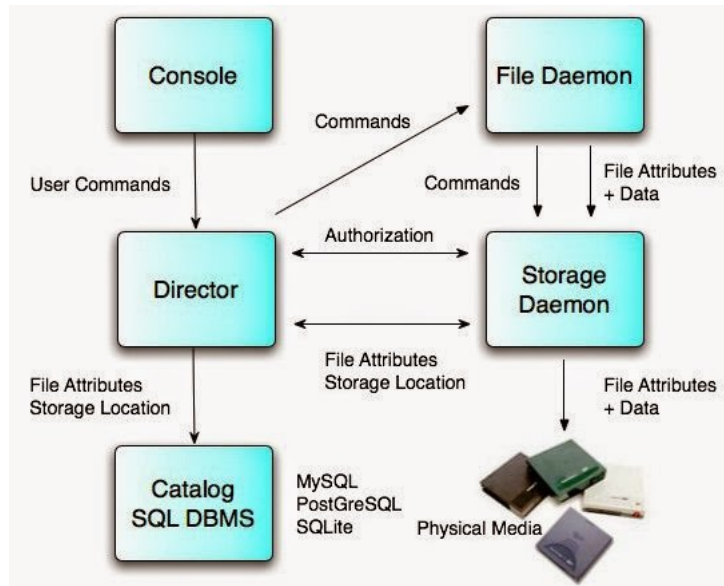


Figura 2.13: Esquema de funcionamiento de Bacula. Fuente <http://goo.gl/b3TJWM>

Componentes

1. Director (DIR)

- Administra y controla de manera centralizada todos los procesos.
- La unidad básica es un job (un cliente, un conjunto de archivos, ...).
- Planifica, inicia y supervisa todos los jobs. -Mantiene el catálogo.
- Generalmente, se configura un Director, excepto en implementaciones muy grandes.

2. File daemon o client (FD)

- Realiza el backup, restauración y verificación solicitada por el director.
- Se instala en cada máquina como un servicio (daemon).
- Se comunica a través de la red con los demonios del director y el storage.
- Requiere acceso a todos los archivos a los cuales se les hará backup.
- Generalmente, existen múltiples File daemons por Director.

3. Storage daemon (SD)

- Lee y escribe data al medio físico Disk, Tape, CD/DVD, USB, ...
- Acepta órdenes y autorizaciones del director.
- Acepta y retorna data hacía/desde File daemons (FD).
- Envía la ubicación del file storage al Director >Catálogo.
- Generalmente, se configura como un director pero con múltiples dispositivos.

4. Consola

- Permite al usuario o administrador controlar Bacula.
- Se comunica con el director vía red.
- Inicia los jobs, revisa la salida del job, permite hacer queries y modificar el catálogo.

5. Catálogo de la base de datos

- Base de datos SQL (MySQL, PostgreSQL, o SQLite).
- Registro de jobs ejecutados, volúmenes usados, ubicaciones de archivos, ...
- Permite recuperaciones rápidas.
- Permite averiguar cuando y donde fueron realizados los backups de los archivos.
- La data muy vieja es reorganizada automáticamente por el director.[22]

Principales características

- **Arquitectura cliente servidor:** Permite la posibilidad de realizar copias de sistemas físicamente alejados pero conectados a una misma red y genera un entorno fácilmente escalable.
- **Uso de formatos libres:** Amanda utiliza herramientas de código abierto como GNUtar o dump.
- **Seguridad:** Las comunicaciones cliente-servidor pueden ser protegidas usando OpenSSH o algoritmos de cifrado.
- **Caché en disco:** Amanda almacena las copias en un disco para evitar pérdidas de datos y mejorar la velocidad de grabación.
- **Programación adaptable:** Se puede utilizar una programación de copias laxa para que Amanda decida el mejor momento para realizarlas aprovechando mejor los recursos del servidor.

- **Gestión de dispositivos:** Capacidad para gestionar cambiadores y diferentes soportes de grabación como cintas, discos duros, CD-ROM y otros medios ópticos.[23]

2.7. Análisis y diseño

2.7.1. Estudio de la situación actual

En la actualidad el grupo GID-CONUSS cuenta con 3 servidores de altas prestaciones que en su momento prestaban servicios de computación en la nube, dichos servidores, con el paso del tiempo, se fueron tornando lentos debido a problemas de software, disminuyendo así su capacidad de prestar un buen servicio a la comunidad universitaria. Por tal motivo se tomó la decisión de desconectarlos de la red.

Al momento de iniciar este proyecto, solo uno de estos tres servidores estaba prestando servicios a la comunidad universitaria, por lo que se ve la necesidad de plantear e implementar un nuevo modelo de computación en la nube basado en Openstack y a demás un modelo de alta disponibilidad para aplicaciones web que hagan uso de estos recursos que no se han estado utilizando.

Esquema general de la infraestructura

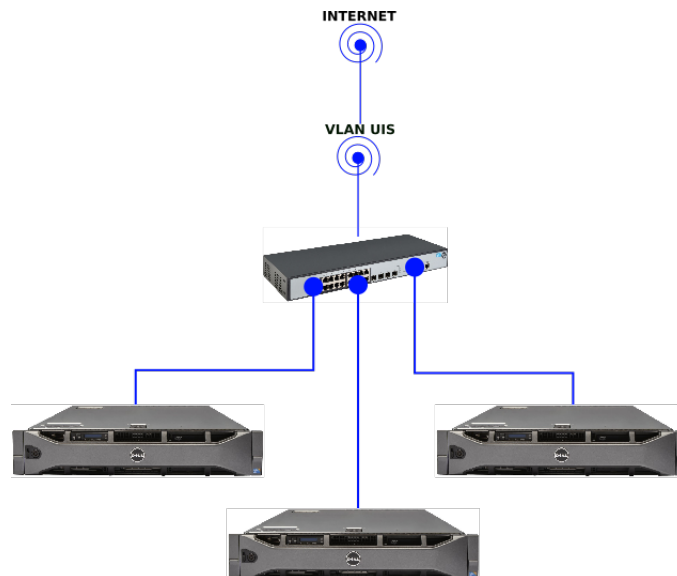


Figura 2.14: Esquema general de la infraestructura propiedad de grupo GID-CONUSS.

Modelo de computación en la nube

Este modelo está basado en la tecnología openstack y se encuentra desplegado sobre los tres servidores propiedad del grupo GID-CONUSS. Los diferentes módulos de openstack que fueron instalados (Cinder, Nova, Neutron, Keystone, Glance, Horizon), y bases de datos distribuidas, fueron desplegados sobre máquinas virtuales que corren dentro de cada uno de los tres servidores que hacen parte de este modelo.

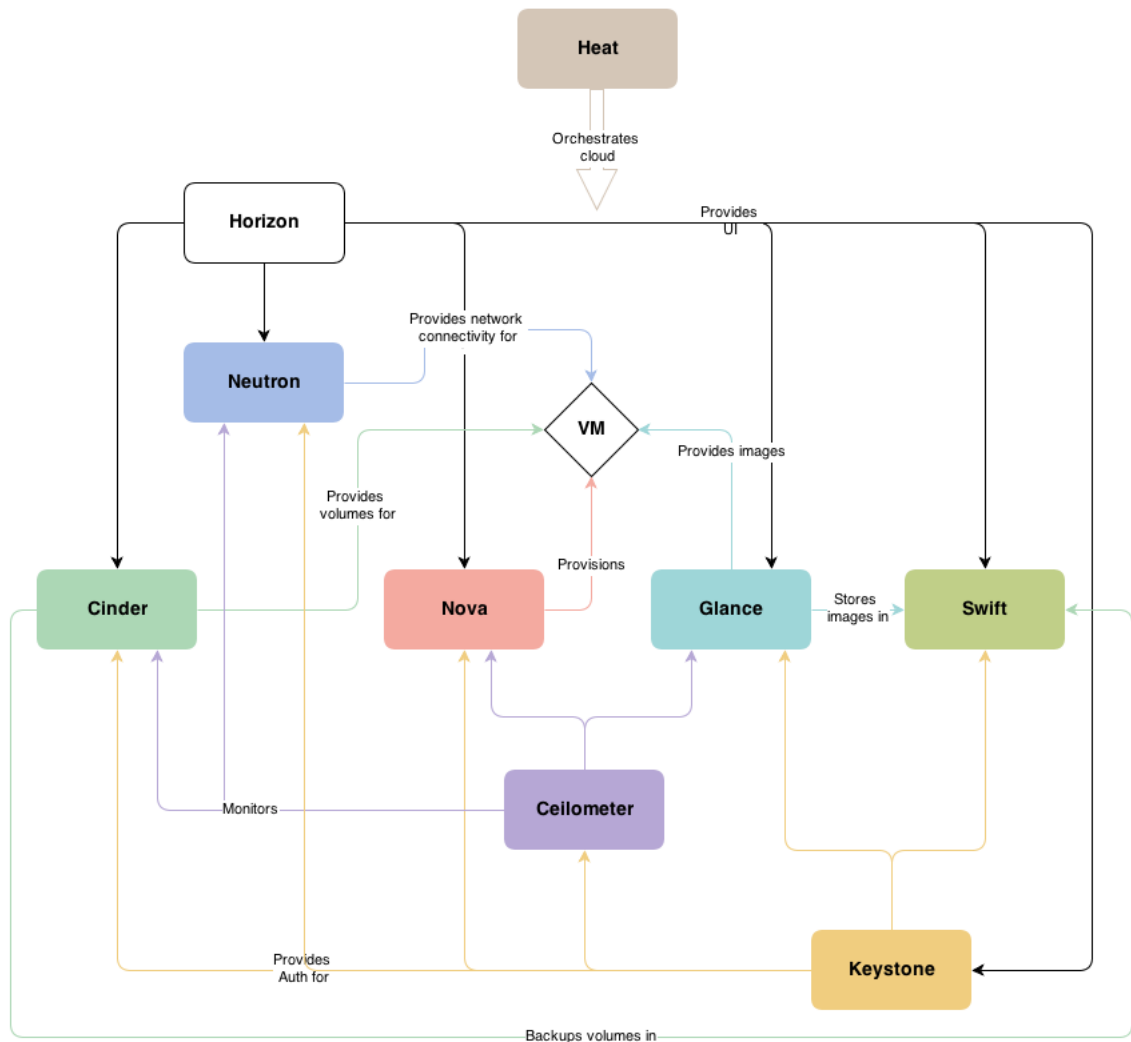


Figura 2.15: Arquitectura conceptual de OpenStack. Fuente <http://goo.gl/1RbDBm>

2.7.2. Modelo de alta disponibilidad

El modelo de alta disponibilidad garantiza el funcionamiento y disponibilidad de la plataforma MEIWEB mediante balanceo de carga, y proporciona la sincronización de datos entre las dos instancias de la plataforma MEIWEB que actualmente están desplegadas sobre la infraestructura. Para evitar pérdidas de datos, el modelo garantiza una copia de respaldo diaria. Teniendo en cuenta más servicios que puedan prestarse a la comunidad universitaria este modelo es flexible hasta cierto punto.

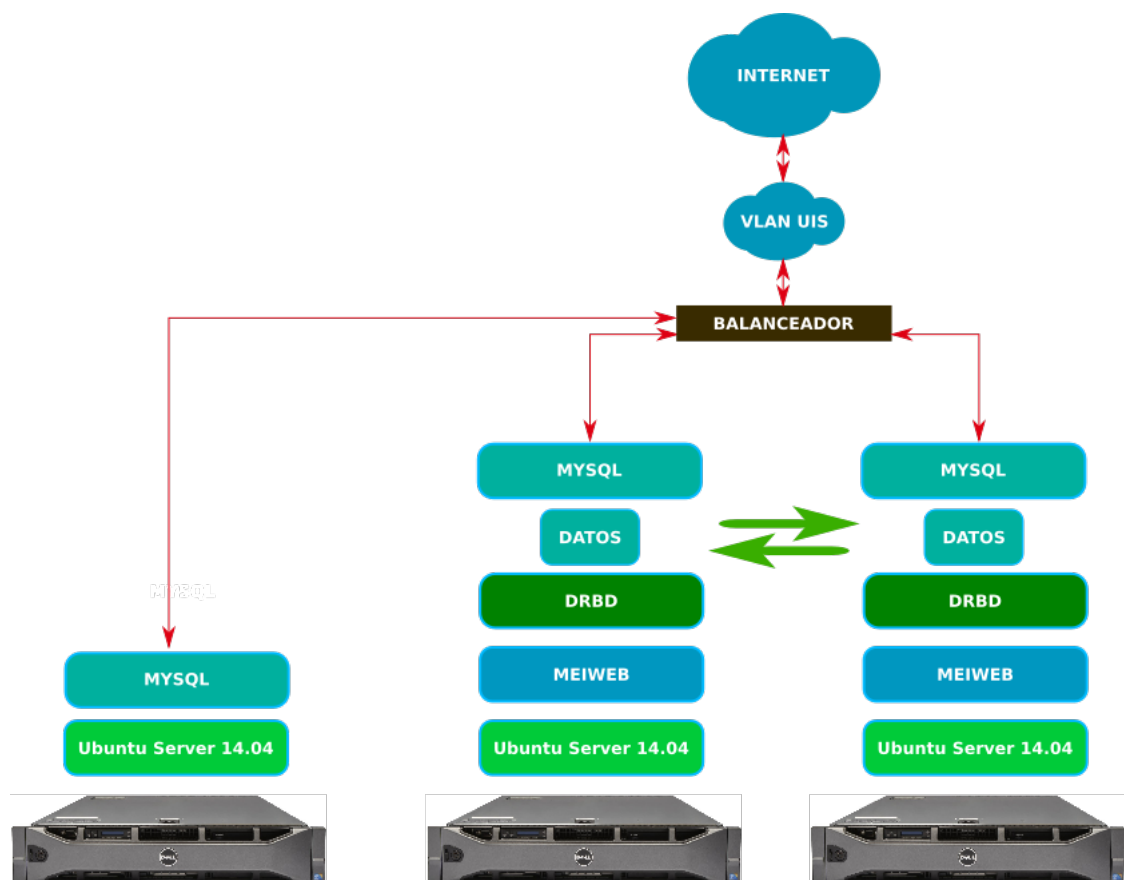


Figura 2.16: Modelo e alta disponibilidad.

2.7.3. Consideraciones iniciales de diseño

Partiendo de las necesidades del grupo GID-CONUSS, en la creación de este nuevo modelo se inicia la investigación de los diferentes requisitos o características que debe cumplir la nueva infraestructura.

Con base en la investigación realizada, se determina que la infraestructura como servicio (IaaS) y el modelo de alta disponibilidad implementado deben contar con las siguientes características

mínimas con el fin de brindar un servicio sólido y estable a la comunidad universitaria:

- Despliegue de instancias.
- Redundancia de datos.
- Copias de seguridad.
- Base de datos distribuida.
- Balanceador de carga para distribuir los recursos.

2.7.4. Diseño del modelo

Este diseño parte del formateo y particionado de los 3 servidores del grupo GID-CONUSS, posteriormente se instala el hipervisor kvm y se crean las máquinas pertinentes a cada componente de la infraestructura y modelo de alta disponibilidad. Cada servicio, separado desde la perspectiva de su funcionalidad, será incorporado dentro de una máquina virtual.



Figura 2.17: Esquema base de diseño.

2.8. Implementación del modelo

Durante el proceso de implementación de la nube privada y el modelo de alta disponibilidad, se realizó un análisis de las herramientas necesarias para dar soporte a un servicio de alta disponibilidad y un servicio de computación en la nube de acuerdo a los requerimientos planteados en la fase de análisis y diseño del modelo, y además de ello debían cumplir con unos requisitos planteados desde el inicio del proyecto.

- Las herramientas usadas deben ser gratuitas.
- No debe existir incompatibilidad de licencias o limitaciones de uso.
- Deben funcionar sobre Ubuntu (sistema operativo de los servidores).

2.8.1. Selección del hipervisor

En el análisis de la selección se decidió implementar el esquema de la infraestructura bajo el hipervisor kvm. Esto se hizo pensando en la continuidad de los servicios que hasta el momento funcionaban en el sistema y en la relativa facilidad de hacer la migración y modificación de los módulos que cobija la nueva infraestructura.

KVM V/S Virtualbox

La toma de decisión se basó en los resultados obtenidos de las diferentes pruebas de rendimientos aplicados sobre ambos hipervisores que realizó Michael Larabel sobre Ubuntu 14.04, a continuación se muestran algunas de los resultados obtenidos.

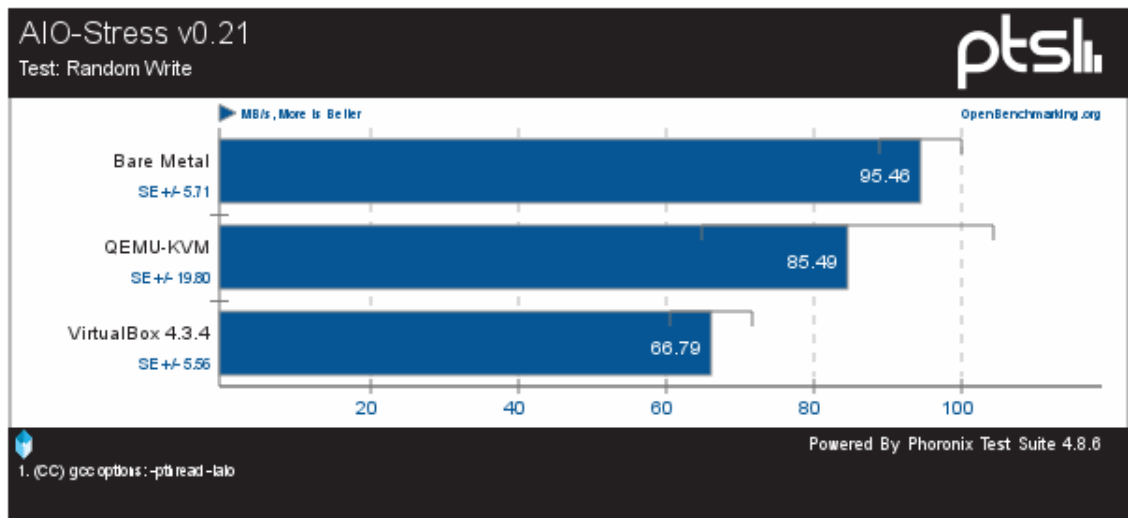


Figura 2.18: Prueba de escritura aleatoria: en ésta prueba QEMU-KVM resulta ser más rápido que VirtualBox.

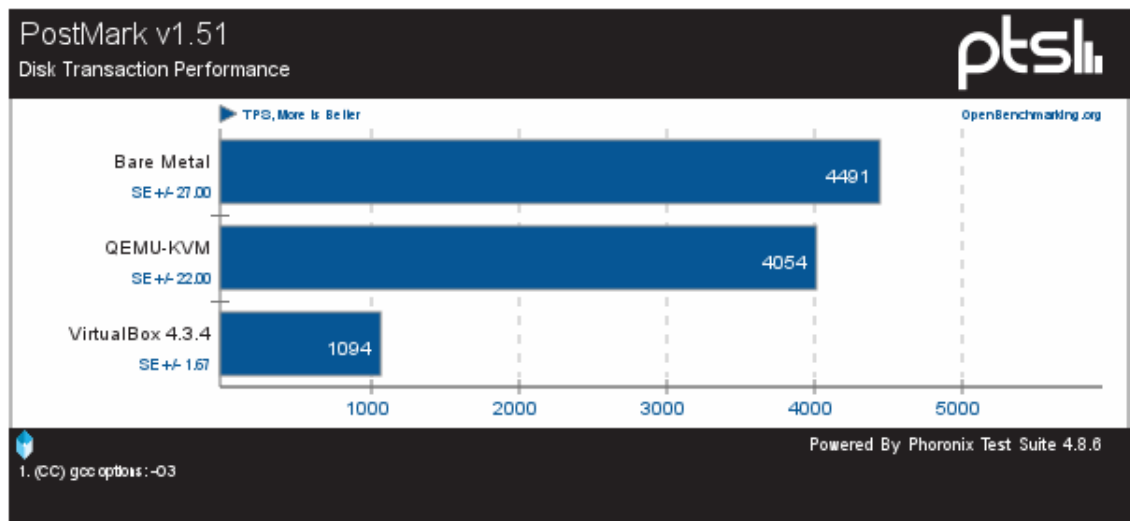


Figura 2.19: Prueba de rendimiento en las transacciones del disco: en esta prueba de rendimiento del disco, VirtualBox corre mucho más lento que QEMU-KVM

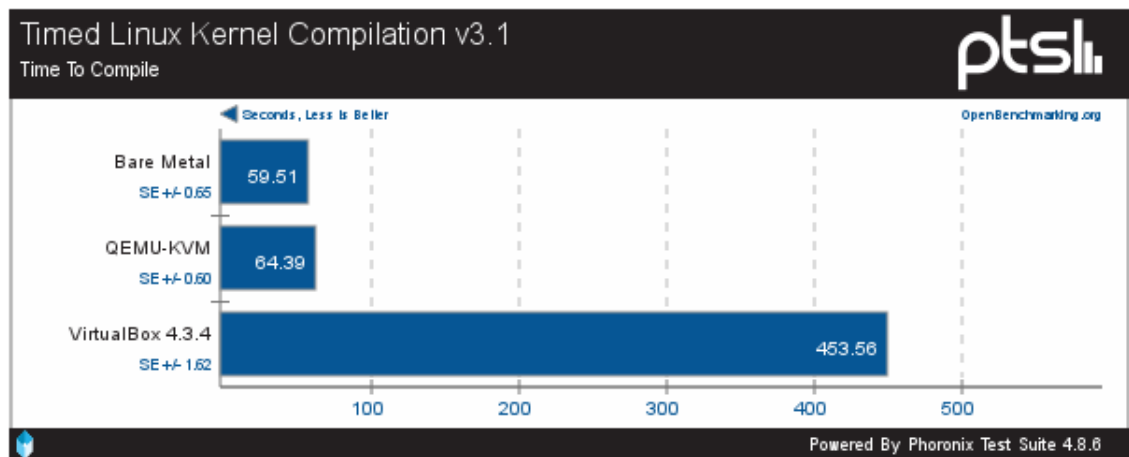


Figura 2.20: Prueba de compilación: compilar el kernel de linux fue mucho más lento con VirtualBox que con QEMU-KVM, a pesar de los resultados, ambas máquinas tienen la misma configuración.

Según estos resultados, las máquinas virtuales sobre QEMU-KVM tienen un mejor rendimiento, comparado con máquinas virtuales que corren sobre VirtualBox y por tal motivo se ha elegido QEMU-KVM como el hipervisor para manejar las máquinas virtuales en este proyecto.

Si desea ver el resultado completo de las pruebas de rendimiento puede hacerlo accediendo al enlace al cual hace referencia en la bibliografía [25].

Herramienta de administración para la nube privada

Las herramientas de administración permiten controlar recursos de computación, almacenamiento y redes a través de un panel de administración provisto por el módulo llamado Horizon. Su arquitectura es distribuida y utiliza diferentes tipos de nodos especializados en distintos servicios y conectados mediante redes de gestión y datos. Usando la terminología Cloud OpenStack se considera una plataforma para Infrastructure as a Service (IaaS por sus siglas en inglés) o Infraestructura como Servicio. Entre sus herramientas para administración están los siguientes módulos:

- Nova (Computación)
- Cinder (Almacenamiento de bloques)
- Glance (Imágenes)
- Swift (Almacenamiento de objetos)
- Neutron (Redes)
- Keystone (Identificación)
- Ceilometer (Telemetría)
- Heat (Orquestación)
- Horizon (Panel de administración)

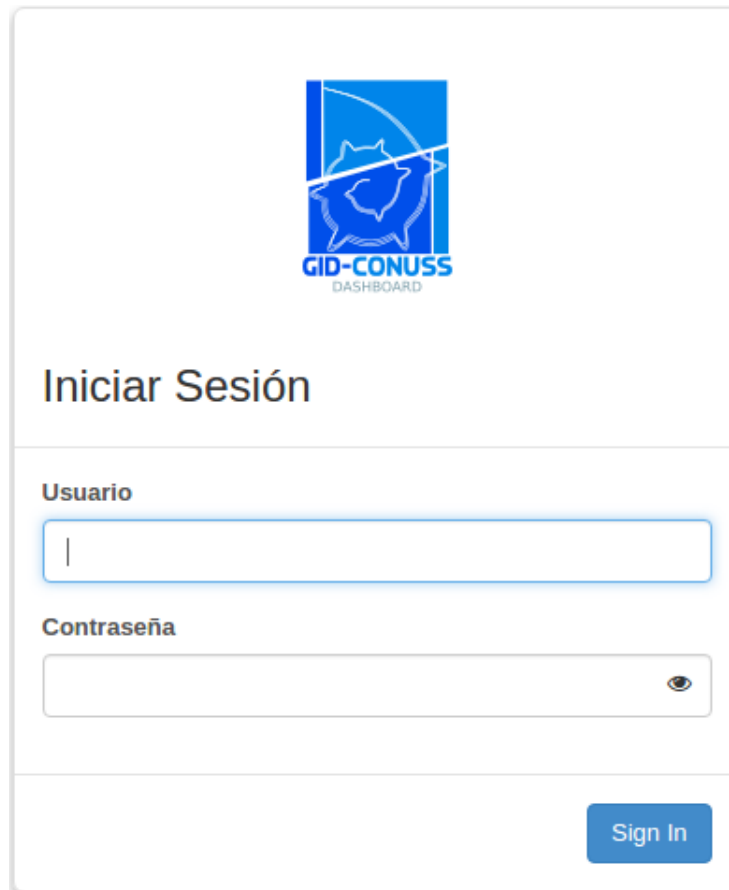


Figura 2.21: Vista de inicio de sesión de Horizon. Fuente: Horizon <https://www.openstack.org/>

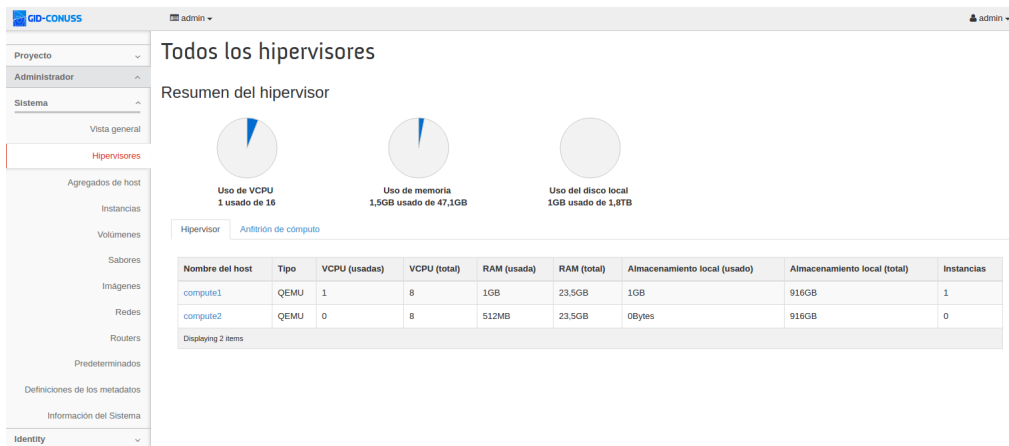


Figura 2.22: Vista principal de Horizon. Fuente: Horizon <https://www.openstack.org/>

Herramientas utilizadas en el modelo de alta disponibilidad

- DRBD (Distributed Replicated Block Device):** Es un software que permite la creación de sistemas de almacenamiento distribuido y opera en sistemas GNU/Linux. Esto se logra mediante la duplicación de un dispositivo (disco duro, partición, volumen lógico, etc.) a

través de una red asignada.[24]

- **Corosync y Pacemaker:** Corosync es un programa de código abierto que proporciona capacidades de mensajería entre miembros de un clúster y, a menudo se hace referencia a él como la capa de mensajería entre servidores cliente. Pacemaker es un gestor de código abierto de recursos de clúster (CRM), es un sistema que coordina los recursos y servicios que son gestionados y hacen la alta disponibilidad en un clúster. En esencia, Corosync permite a los servidores comunicarse como un cluster, mientras que Pacemaker proporciona la capacidad de controlar el comportamiento del cluster. [26]
- **Haproxy:** HAProxy es una solución gratuita y de código abierto, muy rápida y fiable que ofrece alta disponibilidad, balanceo de carga, proxy para TCP y aplicaciones basadas en HTTP. Es especialmente adecuado para los sitios web que presentan mucho tráfico y además se puede ejecutar en Linux, Solaris y FreeBSD.[27]

Antes de la instalación de las herramientas nombradas se llevó a cabo un proceso de investigación determinando las herramientas más convenientes para su utilización, partiendo de varias alternativas de software, se optó por elegir las mencionadas por distintos factores:

- Todas las herramientas software son de carácter libre.
- Tienen documentación completa y soporte actualizado.
- Buenas características de seguridad.
- Adaptables a distintos esquemas de funcionamiento según sea requerido.
- Se complementan muy bien como conjunto.

Implementación modelo computación en la nube

En la implementación del modelo, se tomó el esquema que plantea la guía oficial OpenStack. Gracias a que su arquitectura es distribuida y utiliza diferentes tipos de nodos especializados en distintos servicios y conectados mediante redes de gestión y datos. Lo que caracteriza esta infraestructura es que se puede controlar los recursos de computación, almacenamiento y redes a través de un dashboard. Su arquitectura es distribuida y utiliza diferentes tipos de nodos especializados en distintos servicios y conectados mediante redes de gestión y datos.

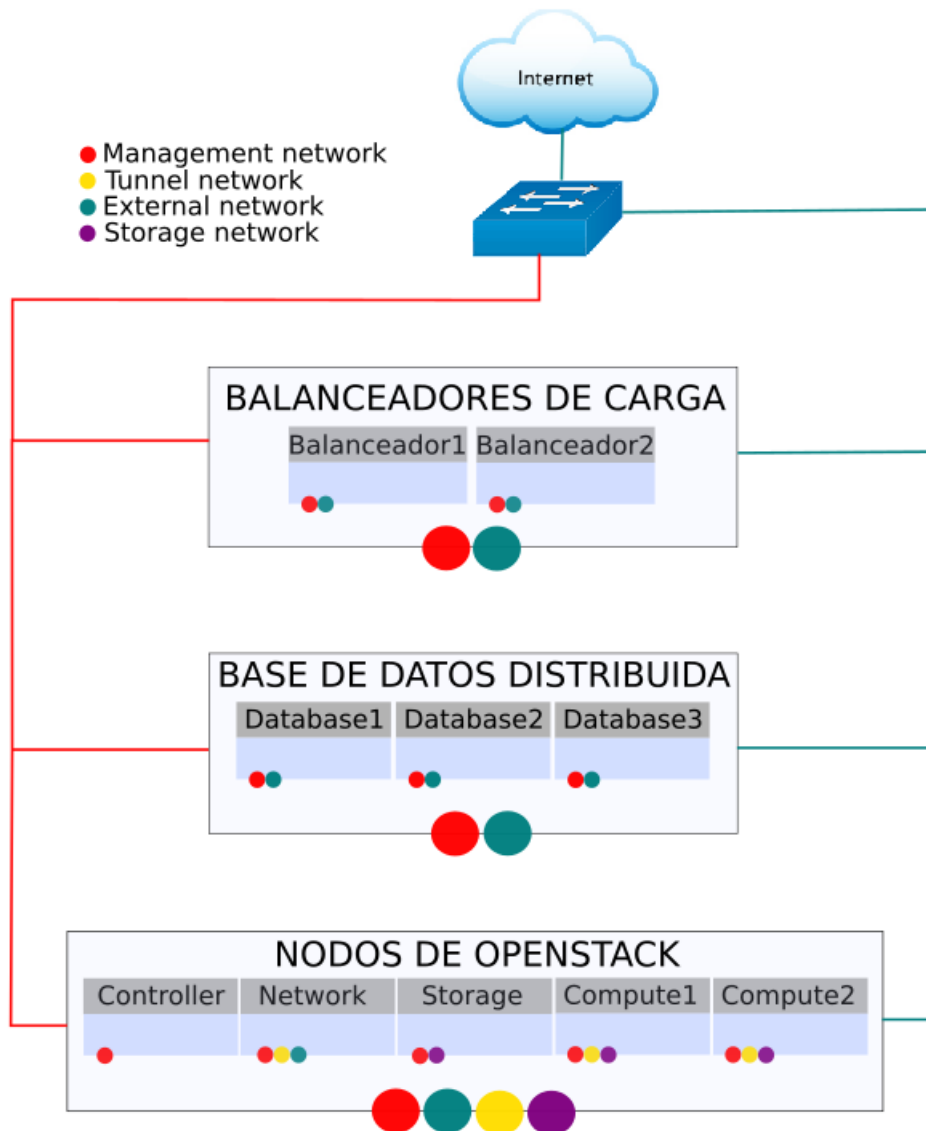


Figura 2.23: Arquitectura del modelo de computación en la nube.

Ventajas de la infraestructura implementada:

1. Se dispuso de dos nodos compute para que la capacidad de creación de máquinas virtuales fuera mucho mayor.
2. La creación de 4 redes para un mejor tráfico de datos en el modelo de computación en la nube.
3. La base de datos distribuida nos permite tener la información del modelo de computación en la nube fiable y disponible en todo momento.
4. Por último, nuestros balanceadores de carga evitan la saturación de los servidores, lo cual

es de gran importancia para toda infraestructura de computación en la nube.

Implementación modelo de alta disponibilidad

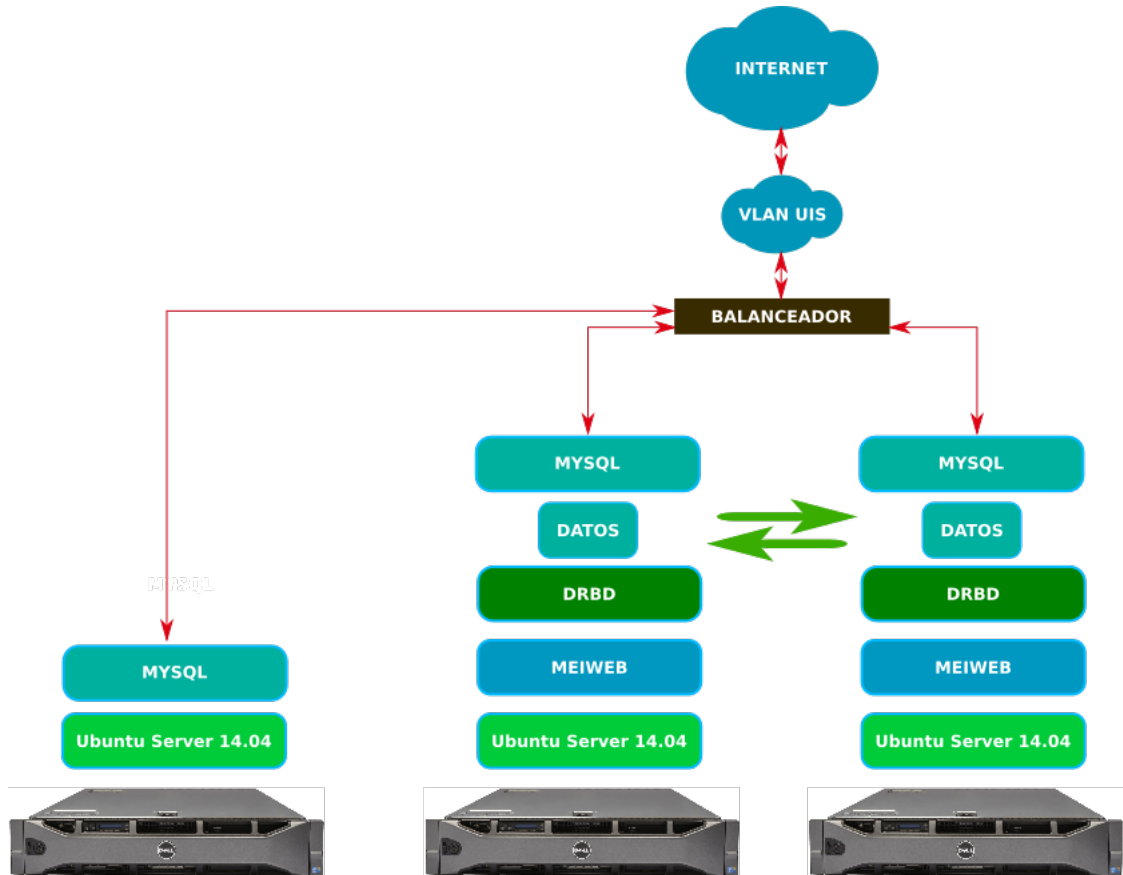


Figura 2.24: Arquitectura del modelo de alta disponibilidad.

Este modelo ha sido probado con éxito utilizando el aula virtual MEIWEB, y de él hacen parte los tres servidores de propiedad del grupo GID-CONUSS, se han desplegado tres nodos (máquinas virtuales) que hacen parte de un cluster de base de datos distribuidas y dos balanceadores (máquinas virtuales) que reparten la carga entre las dos máquinas virtuales que corren MEIWEB, adicionalmente, a cada máquina MEIWEB se le ha configurado una partición de alta disponibilidad manejada por el software DRBD que se encarga de mantener los datos sincronizados en cada una de ellas, permitiendo así tener los mismos datos en ambos servidores virtuales que corren MEIWEB.

Este modelo brinda la posibilidad de tener en alta disponibilidad aplicaciones web que hagan uso de base de datos y datos.

Ventajas

- Evita la saturación de servidores.
- Brinda funcionalidad permanente
- Es transparente para el usuario.
- Brinda alta disponibilidad en datos como en base de datos.
- Modularidad: si un nodo falla simplemente se puede reemplazar.

2.9. Puesta en marcha del modelo

Después de pasar por las fases de análisis, diseño y la implementación de la infraestructura de computación en la nube y el modelo de alta disponibilidad, se presenta en el siguiente apartado la puesta en marcha.

Copias de seguridad

Copias de seguridad de la infraestructura inicial de computación en la nube y el modelo de alta disponibilidad, con el fin de tener un punto de partida en caso de cualquier daño inesperado o inconsistencias del sistema.

Copia de seguridad de máquinas virtuales

Las copias de seguridad de las máquinas virtuales hacen parte de una medida tomada para poder restablecer toda la infraestructura en caso de fallo; si llegase a fallar una o varias máquinas virtuales, estas copias permitirán hacer un reemplazo inmediato de los nodos afectados y un rápido despliegue nuevamente de la infraestructura. Las copias han sido almacenadas en un servidor externo a las infraestructuras desplegadas .

Copia de seguridad de los archivos de configuración

Nos vimos en la tarea de sacar copias de seguridad de archivos de configuración del software instalado de la infraestructura en general. De esta manera tenemos la base documental que nos permita más adelante restablecer el sistema en caso de fallo debido a modificaciones de los miembros del grupo GID-CONUSS.

Copia de seguridad de los datos

Este servicio hace parte del modelo de alta disponibilidad y permite sacar copias completas, incrementales y diferenciales programadas para cada cierta hora del día de los datos sensibles de una aplicación web.

2.9.1. Seguridad

Sistemas de control y monitoreo

Para este fin se dejó instalado el software Observium. Observium es una plataforma que permite supervisar la salud y el estado de las redes, que soporta una amplia gama de tipos de dispositivos, plataformas y sistemas operativos.

Detección ataques de fuerza bruta

A todos los servidores físicos se les dejó instalado el software Fail2ban configurado para enviar notificaciones a un correo electrónico en caso de bloquear a un atacante. Fail2ban bloquea la ip del atacante cada vez que ocurre cierto número de fallos al intentar iniciar sesión haciendo uso del protocolo SSH en algunas de las máquinas, enviando así un email al administrador que contiene datos significativos para identificar el atacante.

En la nueva infraestructura solo los tres servidores físicos y los dos nodos que corren el software MEIWEB y hacen parte del modelo de alta disponibilidad tienen instalado el software Fail2Ban, se recomienda seguir instalando Fail2Ban a los demás servidores que hacen parte de la nueva infraestructura. En la sección de recomendaciones se anuncian algunas sugerencias a la hora de instalar y configurar Fail2Ban.

2.9.2. Documentación administrativa

Con el objetivo de mantener un orden administrativo de los nuevos servicios, se crearon los manuales administrativos. Son documentos individuales acorde a cada servicio implementado, para abstraer las configuraciones por separado. En estos documentos se tienen procesos, configuraciones generales y aplicables a todos los servicios.

2.9.3. Pruebas de funcionamiento

En este apartado se presentan las pruebas que miden el funcionamiento de las máquinas instanciadas y balanceador de carga.

Pruebas de instanciación de imágenes virtuales

Cada instancia de una máquina virtual pertenece a un cliente. Los usuarios pueden crear una o más instancias de máquinas virtuales, en uno o más clientes, y trabajar en ellas. [30]

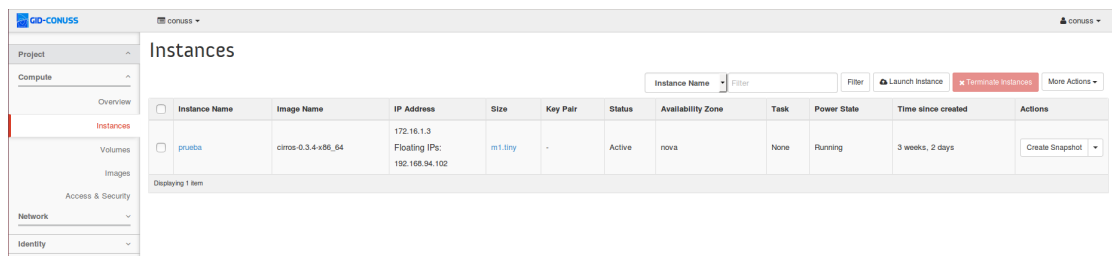


Figura 2.25: Detalles de una máquina virtual de prueba desplegada en la nube de Openstack. Fuente: Horizon <https://www.openstack.org/>

Desde el mismo panel de administración podemos acceder a una terminal y tomar el control de la máquina virtual.

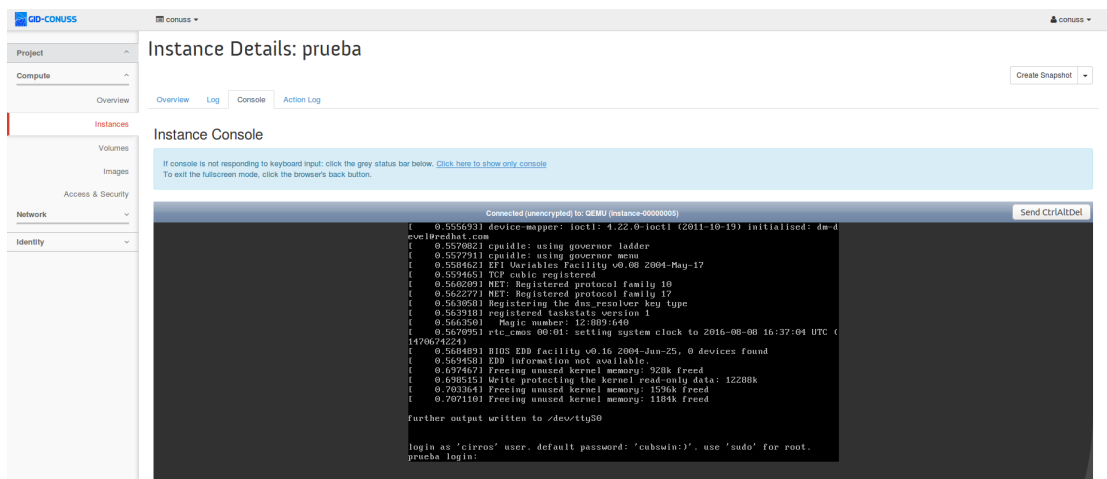


Figura 2.26: Acceso a la terminal de la máquina virtual creada en la nube de OpenStack. Fuente: Horizon <https://www.openstack.org/>

Pruebas del sistema de balanceo de carga

HAProxy es un sistema, software o hardware, que distribuye las peticiones de los clientes de forma equitativa entre distintos servidores, así el balanceador distribuye las peticiones y son

los servidores, quienes arman la respuesta efectiva al cliente. Para balancear la carga entre varios servidores, es necesario que el balanceador detecte servidores sobrecargados para dejar de enviarle peticiones hasta que no baje su carga. Este mismo mecanismo sirve para que un balanceador no envíe peticiones a un servidor caído.

HAProxy version 1.6.4, released 2016/03/13

Statistics Report for pid 1109

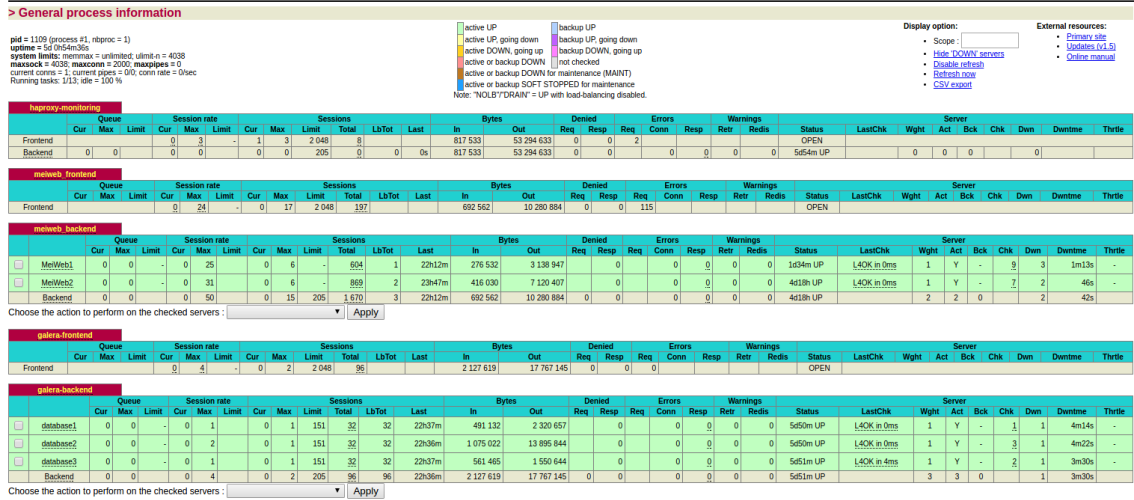


Figura 2.27: Vista principal de HAProxy. Fuente: Software Haproxy www.haproxy.org/

Pruebas topología de red

Una de las grandes ventajas de Neutron es que nos permite ver la topología de red de una manera rápida y sencilla, simplemente haciendo clic en menú de la izquierda del dashboard y seleccionando Topología de red.

En la siguiente captura se muestra la topología de red implementada por el módulo de red de Openstack:

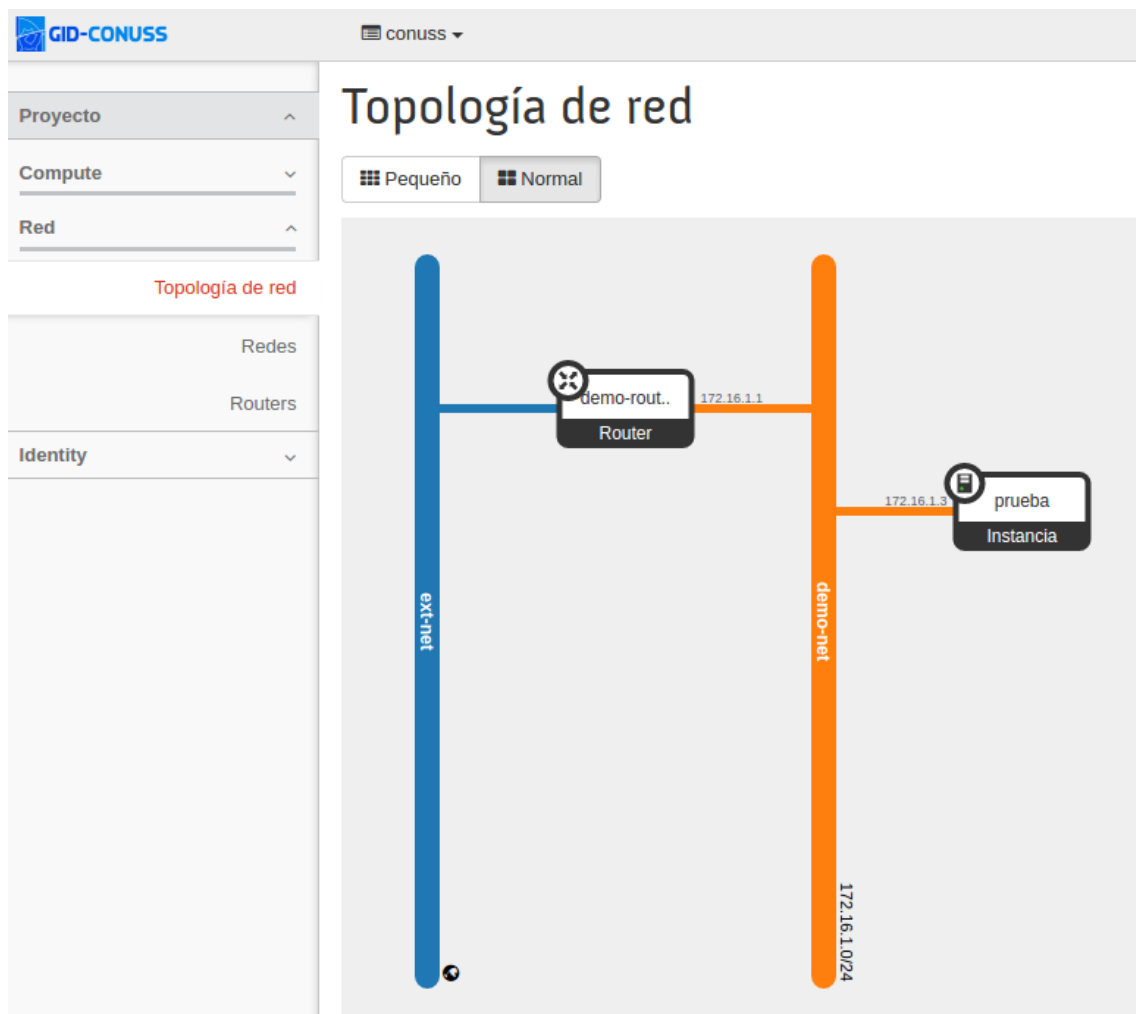


Figura 2.28: Vista de la topología de red de OpenStack. Fuente: Horizon <https://www.openstack.org/>

Pruebas de Monitoreo a servidores Meiweb

Nuestro sistema de control y monitoreo es administrado por el software Obsevium, algunas de las características de este software ya han sido mencionadas en el apartado **Sistemas de control y monitoreo**. A continuación se observan algunas gráficas que muestran el rendimiento de dos máquinas que corren el software MEIWEB.

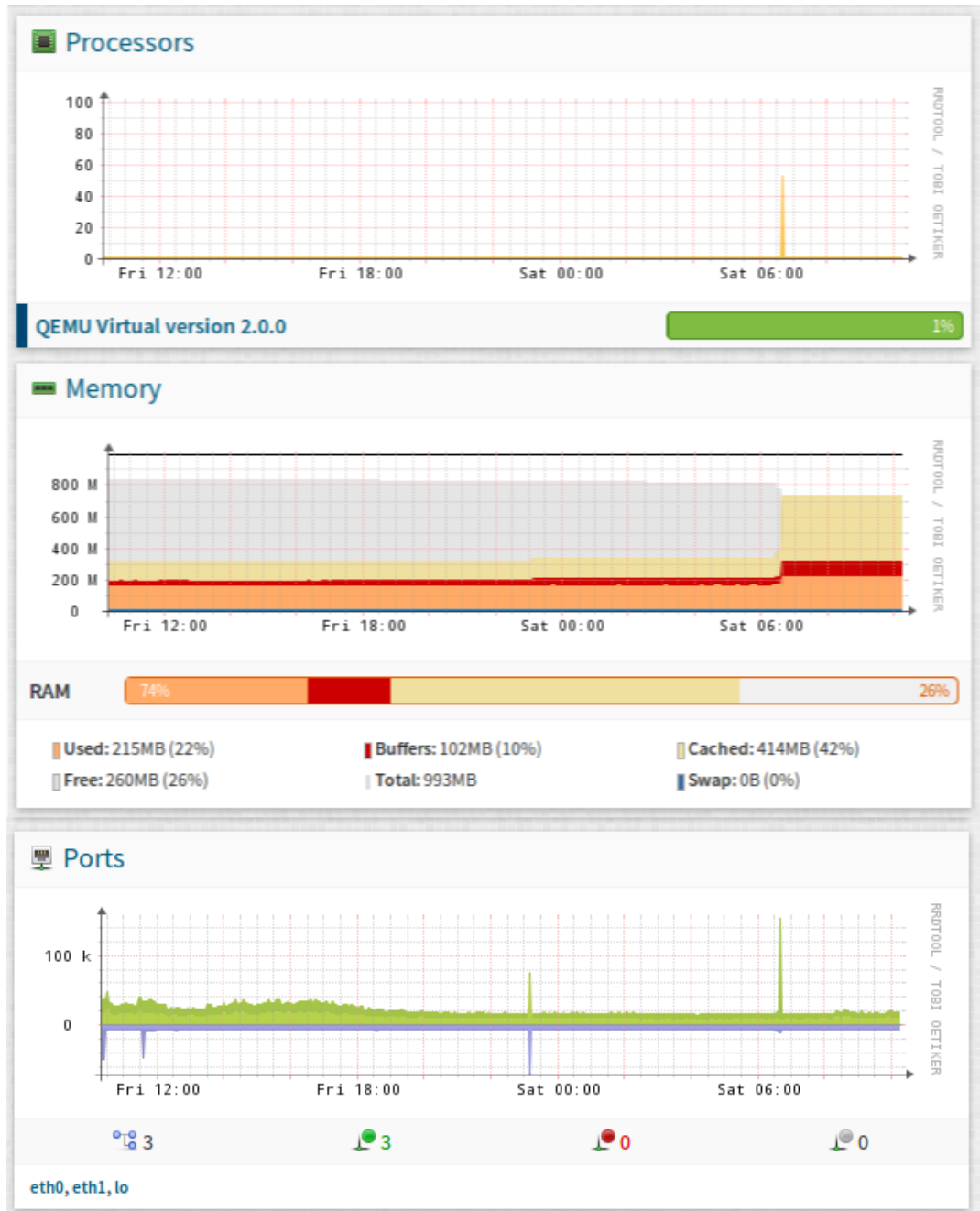


Figura 2.29: Consumo de recursos del procesador, uso de memoria y puertos de red para MEIWEB 1. Fuente: Software Observium www.observium.org

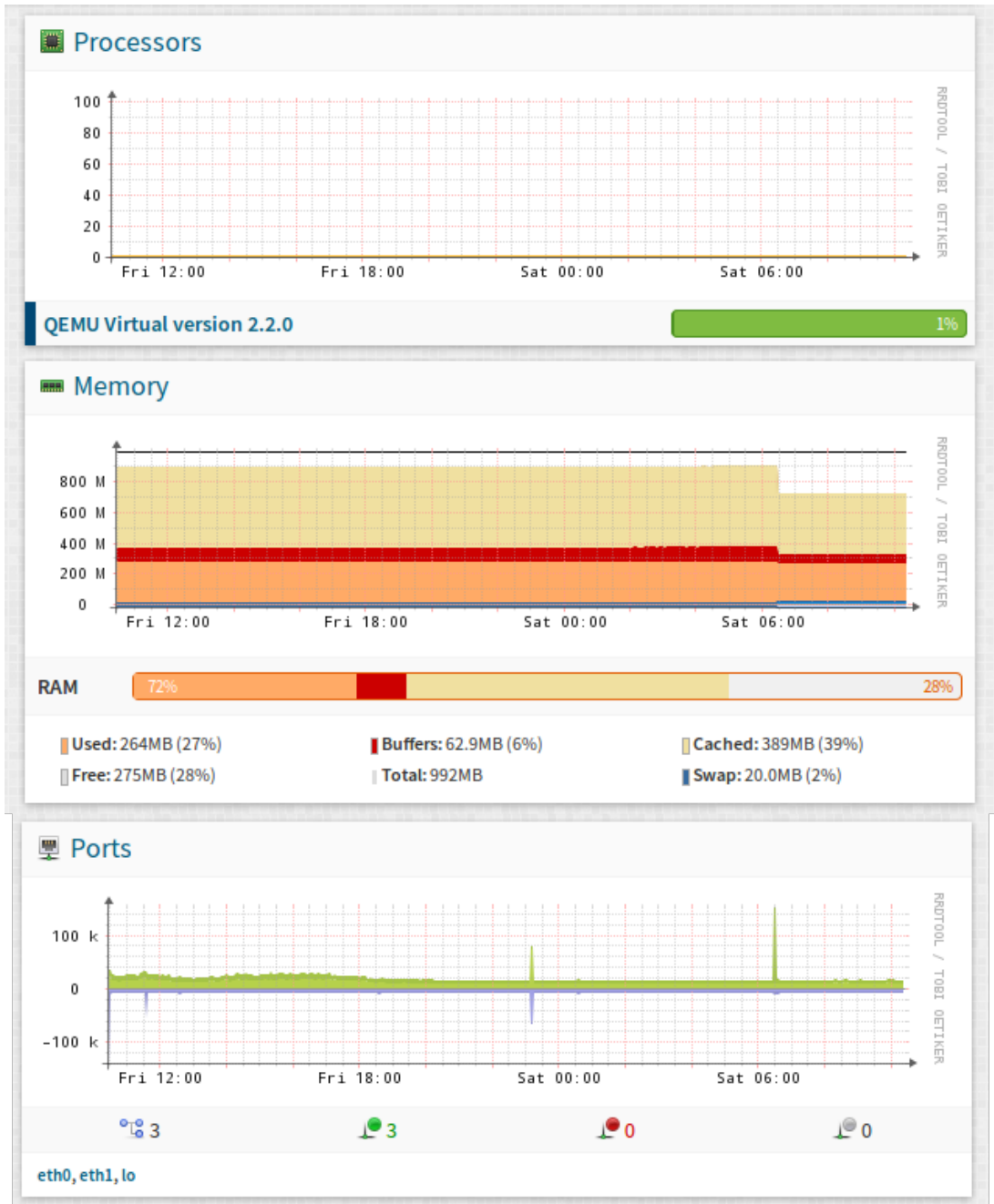


Figura 2.30: Consumo de recursos del procesador, uso de memoria y puertos de red para MEIWEB 2. Fuente: Software Observium www.observium.org

2.10. Conclusiones

Los modelos creados de alta disponibilidad y computación en la nube, se convierten en un aporte en el campo de diseño de infraestructuras de IT, permitiendo al grupo GID-CONUSS establecer nuevos proyectos de investigación y siendo puntos de partida para futuros proyectos de la comunidad académica en general.

Se actualizó toda la infraestructura de servidores, logrando así el mejoramiento de los servicios prestados por el ClousEISI.

Se logró implementar un modelo de alta disponibilidad que garantiza la disponibilidad de los servicios y datos de aplicaciones web.

EL nuevo modelo de computación en la nube permite brindar servicios de máquinas virtuales bajo demanda a estudiantes que hacen parte de la comunidad universitaria.

El uso de tecnologías de software libre como OpenStack proporcionan múltiples ventajas, entre ellas reducir costos de licencias.

2.11. Recomendaciones

1. Se recomienda crear un servidor de archivos con el fin de que la información se almacene en él y se pueda realizar las copias de seguridad de una forma más sencilla.
2. Con fines en la mejora de la seguridad de los servidores se recomienda instalar y configurar el software Fail2Ban en todas las máquinas virtuales que se desplieguen sobre la infraestructura. Actualmente las únicas máquinas que tienen instalado el software Fail2Ban son los tres servidores físicos y las dos máquinas que corren el software MEIWEB y hacen parte del modelo de alta disponibilidad.
3. Terminar de instalar los módulos restantes de OpenStack (Telemetry, Swift y Orchestration).
4. Realizar un estudio completo de la nueva infraestructura con miras a implementar un firewall que permita tener una seguridad más robusta sobre la infraestructura. Es muy importante crear reglas de seguridad sobre los puertos utilizados por esta nueva infraestructura.

5. Se sugiere continuar con la filosofía de uso de tecnologías de software libre en el desarrollo de nuevos proyectos.

Bibliografía

- [1] V. Autores, “Gartner says cloud computing will be as influential as e-business.” [online]. url: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6745284>, Junio 26, 2008 (Consultado el 10 de diciembre de 2015).
- [2] R. Maggiani, “Cloud computing is changing how we communicate.” [online]. url: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5208703&isnumber=5208666>, 2009 (Consultado el 10 de diciembre de 2015).
- [3] F. Research, “Is cloud computing ready for the enterprise?.” [online]. url: <http://www.forrester.com/Research/Document/Excerpt/0,7211,44229,00.html>, (Consultado el 10 de diciembre de 2015).
- [4] J. Geelan, “Twenty one experts define cloud computing. virtualization.” [online]. url: <http://virtualization.sys-con.com/node/612375>, Agosto de 2008.(Consultado el 10 de diciembre de 2015).
- [5] M. Z. Meetei, “Cloud computing and security measure.” [online]. url: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6745284>, 2013 (Consultado el 10 de diciembre de 2015).
- [6] B. Sosinsky, *¿Qué es la nube? futuro de los sistemas de información*. Juan Ignacio Luca de Tena, 15. 28027 Madrid: Grupo Anaya, 2012.
- [7] M. Z. Meetei, “Cloud computing and security measure.” [online]. url: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6745284>, Diciembre de 2013 (Consultado el 10 de diciembre de 2015).

- [8] V. Autores, “V. high availability (ha).” [online]. url: <http://www.cloudbus.org/raj/tfcc/high-availability.html>, Septiembre 24 de 2012 (Consultado el 15 de diciembre de 2015).
- [9] V. Autores, “Introducción al tema de la virtualización.” [online]. url: <http://goo.gl/KCGKM3>, Septiembre 24 de 2009 (Consultado el 15 de diciembre de 2015).
- [10] J. Smaldone, “Virtualización de hardware.” [online]. url: <https://goo.gl/YZ6WaH>, Septiembre 20 de 2008 (Consultado el 15 de diciembre de 2015).
- [11] E. D. Carreño, “Modelo y prototipo de servicios de computación en la nube para estudiantes y profesores de la escuela de ingeniería de sistemas e informática de la universidad industrial de santander. bucaramanga. universidad industrial de santander. facultad de ingenierías fisicomecánicas. escuela de ingeniería de sistemas e informáticas.” [online]. url: <https://blog.smaldone.com.ar/2008/09/20/virtualizacion-de-hardware/>, 2012 (Consultado el 15 de diciembre de 2015).
- [12] V. Autores, “Virtualización.” [online]. url: <https://goo.gl/Rlepcd>, (Consultado el 10 de diciembre de 2015).
- [13] V. Autores, “What is kvm.” [online]. url: <https://openvirtualizationalliance.org/what-kvm>, (Consultado el 10 de diciembre de 2015).
- [14] V. Autores, “Benefits of kvm.” [online]. url: <https://openvirtualizationalliance.org/what-kvm/benefits-kvm>, (Consultado el 10 de diciembre de 2015).
- [15] V. Autores, “Clúster de alta disponibilidad.” [online]. url: <https://goo.gl/loeCZr>, (Consultado el 10 de diciembre de 2015).
- [16] V. Autores, “Seguridad y alta disponibilidad: Clúster ha.” [online]. url: <http://es.slideshare.net/jmorenol/clsters-alta-disponibilidad>, (Consultado el 10 de diciembre de 2015).
- [17] V. Autores, “Clúster de alta disponibilidad.” [online]. url: <http://goo.gl/4irhTT>, (Consultado el 10 de diciembre de 2015).
- [18] V. Autores, “Openstack.” [online]. url: <https://www.openstack.org/>, (Consultado el 10 de diciembre de 2015).
- [19] V. Autores, “Drbd.” [online]. url: <http://goo.gl/Tp8nQN>, (Consultado el 10 de diciembre de 2015).

- [20] V. Autores, “Linux-ha.” [online]. url: www.linux-ha.org, (Consultado el 10 de diciembre de 2015).
- [21] V. Autores, “Bacula.” [online]. url: <http://blog.bacula.org/>, (Consultado el 10 de diciembre de 2015).
- [22] V. Autores, “Respaldo y recuperación con bacula.” [online]. url: <https://goo.gl/aTM1GX>, (Consultado el 10 de diciembre de 2015).
- [23] C. C. Certad, “Soluciones de backup remoto en gnu/linux.” [online]. url: <http://fraterneo.blogspot.com.co/2014/09/soluciones-de-backup-remoto-en-gnulinux.html>, Septiembre 29 de 2014 (Consultado el 15 de diciembre de 2015).
- [24] A. B. AYALA and E. MUÑOZ, “Instalación, administración, configuración e implementación de servidores linux con énfasis en el desarrollo de un modelo administrativo y la creación de un prototipo de clúster de alta disponibilidad. bucaramanga. universidad industrial de santander. facultad de ingenierías fisicomecánicas. escuela de ingeniería de sistemas e informáticas,” 2012 (Consultado el 15 de diciembre de 2015).
- [25] M. Anicas, “How to create a high availability setup with corosync, pacemaker, and floating ips on ubuntu 14.04.” [online]. url: <https://www.digitalocean.com/community/tutorials/how-to-create-a-high-availability-setup-with-corosync-pacemaker-and-floating-ips-on-ubuntu-14-04>, Enero de 2014 (Consultado el 15 de Octubre de 2015).
- [26] V. Autores, “Haproxy.” [online]. url: <http://www.haproxy.org/>, (Consultado el 05 de Agosto de 2015).
- [27] M. Anicas, “An introduction to haproxy and load balancing concepts.” [online]. url: <https://www.digitalocean.com/community/tutorials/an-introduction-to-haproxy-and-load-balancing-concepts>, Enero de 2014 (Consultado el 07 de Agosto de 2015).
- [28] V. Autores, “Kvm: The rise of open enterprise-class virtualization.” [online]. url: <https://goo.gl/Wv2fRk>, (Consultado el 07 de Octubre de 2015).
- [29] J. Ellingwood, “How to protect an apache server with fail2ban on ubuntu 14.04.” [online]. url: <https://www.digitalocean.com/community/tutorials/how-to-protect-an-apache-server-with-fail2ban-on-ubuntu-14-04>, Agosto 14 de 2015 (Consultado el 25 de Abril de 2016).

[30] V. Autores, “Creación de instancias de máquinas virtuales.” [online]. url: <https://goo.gl/Q4O6Fx>, (Consultado el 28 de Noviembre de 2015).

Apéndice A

Términos y Condiciones de Uso

Este acuerdo establece los términos y condiciones de uso aplicables para usted (el "**USUARIO**") en su uso de los servicios prestados (los **SERVICIOS**) por el Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios ("**GID-CONUSS**") para la plataforma de infraestructura como servicio(IaaS) de computación en la nube CloudEISI.

Al usar hacer uso de los **SERVICIOS**, el **USUARIO** automáticamente acepta estos términos y Condiciones de Uso y acuerda regirse por ellos.

1. Usuarios elegibles. Los servicios están abiertos para el uso de cualquier miembro de la comunidad académica de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander.
2. Cuenta de usuario, contraseña y seguridad. Como parte de la creación de una cuenta de usuario se otorgará un nombre de cuenta y una contraseña. El **USUARIO** es completamente responsable por mantener la confidencialidad de su cuenta o contraseña. Si el **USUARIO** tiene conocimiento de que la privacidad de su contraseña ha quedado expuesta, debe notificar inmediatamente a **GID-CONUSS** e informar acerca de cualquier fallo de seguridad.
3. Contenidos del usuario. El **USUARIO** acepta usar los **SERVICIOS** sólo para fines legales y queda prohibida la publicación, distribución y alojamiento en los **SERVICIOS** de: (a) cualquier material ilegal, perjudicial, amenazante, abusivo, acosador, difamatorio u obsceno de cualquier tipo, incluyendo, entre otros, cualquier material que fomente conductas que constituirían una ofensa penal, que produjera responsabilidad civil o que de otro modo infrinja cualquier ley aplicable local, estatal, nacional o internacional. (b) del cual el **USUARIO** no tenga el derecho de poner a disposición conforme a ninguna ley o relación contractual; o

(c) infrinja cualquier derecho de patente, marca comercial, secreto de mercado, derecho de autor u otros derechos de propiedad de cualquier parte. El **USUARIO** acepta no transmitir, subir, publicar, enviar por email o poner a disposición de otro modo virus informáticos, publicidad no solicitada o no autorizada, material promocional o de venta, lo que incluye cadenas de correspondencia, correos masivos o cualquier otra forma de spam".

El **USUARIO** acepta NO:

- (I) falsificar encabezados o manipular de otro modo los identificadores de archivos con el objeto de ocultar el origen de cualquier contenido del **USUARIO**;
- (II) recopilar o almacenar información de identificación personal acerca de otros usuarios sin la debida autorización y consentimiento de estos;
- (III) utilizar herramientas de interceptación de datos;
- (IV) hacer uso de software malicioso o de obstaculización de redes o, (v) realizar accesos abusivos a los equipos informáticos que permiten la prestación de los **SERVICIOS** o la prestación los **SERVICIOS** de los demás usuarios.

El **USUARIO** reconoce que **GID-CONUSS** tiene el derecho (pero no la obligación), de eliminar el contenido del **USUARIO** que viole estos Términos y Condiciones, y nos reservamos el derecho de denegar el servicio y/o cancelar cuentas sin previo aviso para cualquier usuario que viole estos Términos y Condiciones o infrinja los derechos de terceros.

- 4. Seguridad de los servicios. La seguridad de las instancias asignadas por parte de los **SERVICIOS** prestados recaen bajo la administración del **USUARIO**, así como el manejo de las copias de seguridad de los datos, código fuente, aplicaciones y bases de datos que hagan parte de sus contenidos, **GID-CONUSS** no se responsabiliza en casos de falla total o parcial de los servidores y que imposibiliten la recuperación de los almacenamiento implementados.
- 5. Quejas por derecho de autor. **GID-CONUSS** respeta la propiedad intelectual de terceros. Si usted cree que su trabajo ha sido copiado en este sitio de manera que constituya una infracción a sus derechos de autor, puede escribir a cloudeisi@gmail.com para notificarnos acerca de la situación.
- 6. Descargo de responsabilidad. El **USUARIO** reconoce y acepta que utiliza este servicio sin cargo alguno y que por lo tanto este uso se ofrece tal como está, **GID-CONUSS** expresamente niega toda garantía de cualquier tipo, ya sea expresa o implícita. Como **USUARIO**

acepta de forma expresa que todo uso de la información disponible en las los **SERVICIOS** asignados son para su uso personal o académico y que dicha información se utilizará bajo su propio riesgo y responsabilidad. **GID-CONUSS** expresamente niega toda responsabilidad por errores u omisiones en la información, los materiales y las funciones que se incluyen en nuestros servidores o en los sitios de terceros enlazados desde o hacia nuestros servicios por parte del **USUARIO**.

7. Indemnización. El **USUARIO** acepta indemnizar y liberar de toda responsabilidad a **GID-CONUSS** sus directivos, desarrolladores, investigadores, y administradores en cuanto a cualquier reclamación o demanda, lo que incluye los honorarios razonables de representación legal, entablada por terceros debida a, o que resulte de, el uso por parte del **USUARIO** de los **SERVICIOS** prestado.
8. Potestad jurídica. Los **SERVICIOS** son administrados por **GID-CONUSS** desde sus oficinas dentro de la República de Colombia. Al hacer uso de los **SERVICIOS**, tanto el **USUARIO** como **GID-CONUSS** aceptan que todos los asuntos relacionados con su uso de los **SERVICIOS** estarán regidos e interpretados de acuerdo con la legislación de la República de Colombia, sin considerar los principios de conflictos de leyes, como si se realizarán enteramente dentro de la República de Colombia.
9. Modificación de los términos de uso del servicio. **GID-CONUSS** puede cambiar los términos y condiciones de los Términos de Uso del Servicio. **GID-CONUSS** hará las notificaciones de los cambios y publicará la versión más reciente en la sección Términos de Uso del sitio <http://cloudeisi.uis.edu.co>, si el **USUARIO** no está de acuerdo con las modificaciones puede dejar de usar los **SERVICIOS** informando a **GID-CONUSS**.
10. Confidencialidad de la Información. **GID-CONUSS** no compartirá ni revelará la información confidencial con terceros, excepto que tenga expresa autorización de parte del **USUARIO**, o cuando ha sido requerido por orden judicial o legal.

Apéndice B

GID-CONUSS

El Grupo de Investigación y Desarrollo en Computación en la Nube, Seguridad, Servidores y Servicios (GID-CONUSS), surgió como respuesta a las diferentes áreas hacia las cuales el grupo de desarrollo MEIWEB enfocó algunos de sus proyectos de grado a partir del año 2011.

El grupo MEIWEB, a partir de su creación en el año 2006 por el Magíster Ingeniero Manuel Guillermo Flórez Becerra, ha realizado el desarrollo de un entorno virtual de aprendizaje que en su constante evolución se ha convertido en una excelente herramienta para los docentes y estudiantes que hacen usos de sus servicios.

Este grupo, en el año 2010, gracias a la donación por parte de la fundación Raúl D. Ocazonez de dos servidores para el apoyo a proyectos en la Escuela de Ingeniería de Sistemas e Informática, decidió tomar la responsabilidad de desarrollar proyectos que hicieran usos de los equipos donados y a su vez, pudiesen prestar servicios a toda la comunidad EISI.

Para este fin se desarrollaron dos proyectos paralelos de investigación, un proyecto que desarrollara el modelo administrativo para los equipos e incluyera un prototipo de clúster de alta disponibilidad 60 y un proyecto de prestación de servicios de computación en la nube para la EISI 61 . Con el desarrollo de estos dos proyectos surgió la necesidad de unificar los nuevos campos de investigación de MEIWEB con la rama existente de desarrollo, es ahí donde nace GID-CONUSS.

GID-CONUSS representa la unificación de las ramas de investigación y acción del grupo, la descripción del acrónimo y de las ramas mencionas es la siguiente:

GID-CONUSS: Grupo de Investigación y Desarrollo, para indicar que el grupo incluye una rama de investigación, sin abandonar su área de desarrollo.

GID-CONUSS: La rama de investigación y desarrollo en computación en la nube y virtualización.

GID-CONUSS: Seguridad, Servidores y Servicios. Las múltiples S del acrónimo hacen referencia, a los diferentes campos de acción y a que no están limitados a estos.

La rama de Seguridad, está encargada de definir estándares y políticas dentro de los servicios ofrecidos por el grupo. La rama de Servidores, es un área de apoyo a los diferentes procesos de administración de éstos, y presta también sus servicios a los usuarios de la plataforma de computación en la nube. La rama de Servicios, está enfocada a la prestación de servicios y desarrollo de aplicaciones y ésta, a su vez, es un área de trabajo de la rama de Computación en la Nube.

Como se puede observar, la vinculación y el trabajo en equipo de las diferentes áreas es lo que conforma a GID-CONUSS. Como parte del proceso de unificación del grupo, se creó también un logotipo que permita identificar los diferentes productos y servicios a los cuales esté vinculado GID-CONUSS, en este, se abrevia el nombre, quedando el acrónimo CONUSS.

El logotipo representado por un caparazón, hace referencia lo delicada que es la prestación de servicios en los cuales los datos de los usuarios deben ser prioritarios, así como la importancia que tiene la protección y seguridad de los servicios para el grupo.

Las líneas que conforman la silueta del caparazón, son múltiples líneas juntas simbolizando el uso de las redes de datos para la interconexión de áreas independientes y también para la prestación de los servicios. Esta es, a grandes rasgos, la descripción de las diferentes ramas de investigación y acción del grupo y una pequeña reseña del nacimiento del mismo.

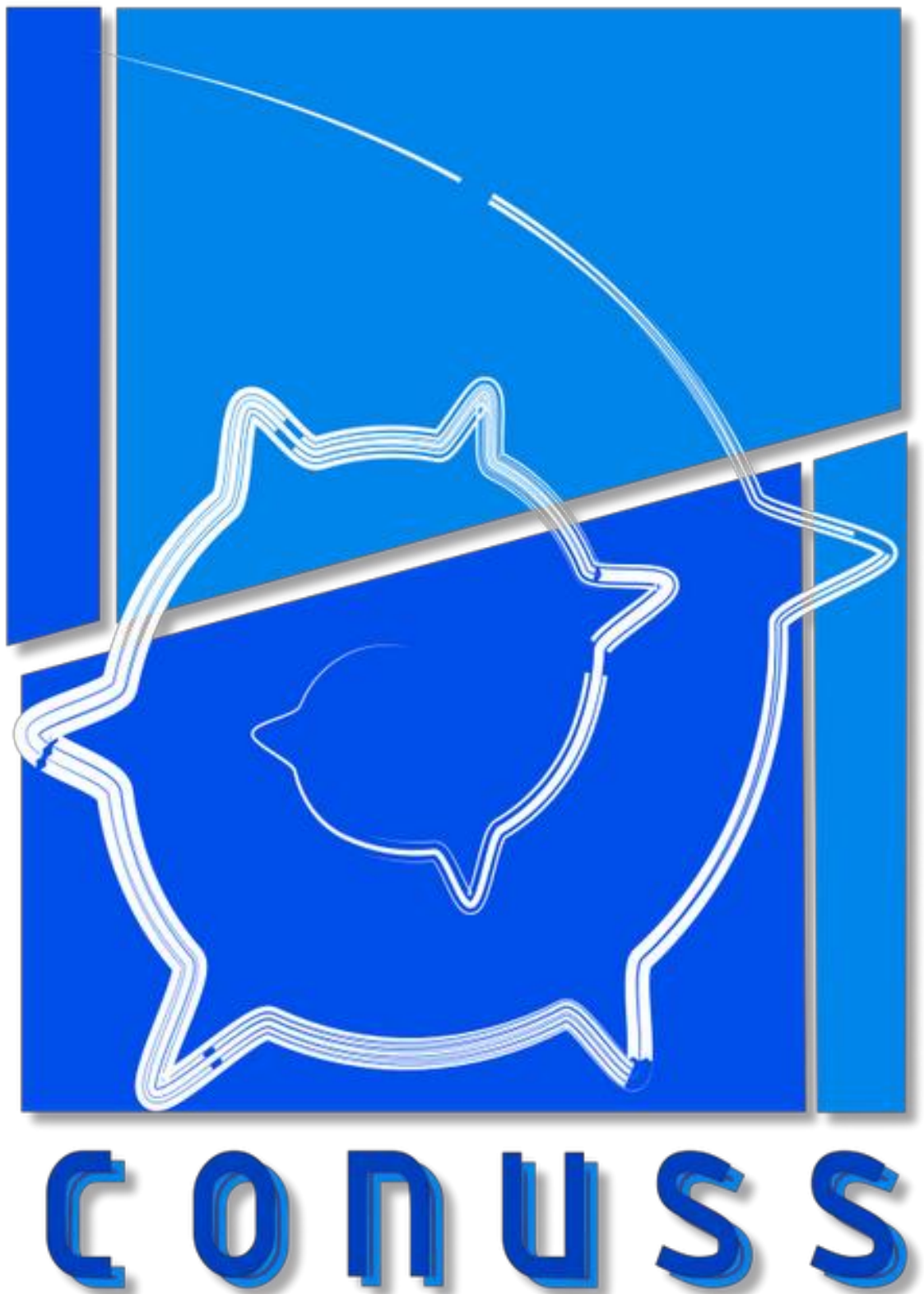


Figura B.1: Identidad visual grupo GID-CONUSS.