

**ANALISIS DE LA ARQUITECTURA DE COMPUTACION EN LA NUBE
CLOUDEISI CON ENFASIS EN EL MEJORAMIENTO DE LA
ADMINISTRACION, FUNCIONALIDAD Y SEGURIDAD.**

**YONATHAN NAHÚN CARREÑO ORTIZ
JOSÉ ALONSO PÁEZ CRUZ**

**UNIVERSIDAD INSUSTRIAL DE SANTANDER
FACULTAD DE INGENERÍAS FÍSICO MECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2016

**ANALISIS DE LA ARQUITECTURA DE COMPUTACION EN LA NUBE
CLOUDEISI CON ENFASIS EN EL MEJORAMIENTO DE LA
ADMINISTRACION, FUNCIONALIDAD Y SEGURIDAD.**

**YONATHAN NAHÚN CARREÑO ORTIZ
JOSÉ ALONSO PÁEZ CRUZ**

**Proyecto de Grado 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

DEDICATORIA

“A todas las personas que de una u otra manera me apoyaron en este camino que no hubiera podido terminar sin su ayuda, muchas gracias de todo corazón, pues si bien este logro es mio el merito es completamente de ustedes”

Yonathan Nahún Carreño Ortiz

“Agradezco especialmente a mis padres por brindarme su apoyo y creer en mi en este largo proceso que al fin termina. A mi pequeña hermana por siempre estar a mi lado dandome animos y regañandome cuando era necesario. Y finalmente a mis mejores amigos (clan SYDE) por hacer de mi vida universitaria y personal una aventura constante llena de inolvidables recuerdos. Gracias”

José Alonso Páez Cruz

AGRADECIMIENTOS

Los autores desean expresar sus mas sinceros agradecimientos a:

El profesor Manuel Guillermo Florez Becerra, por la confianza depositada en este proyecto.

A la universidad Industrial se Santander por permitir espacios que permiten el desarrollo de este tipo de proyectos.

A nuestro compañeros y amigos John Gonzalez, Helmer Acuña cuya presencia y ayuda fue de vital importancia durante este proceso.

CONTENIDO

	Pág.
INTRODUCCION	16
1. JUSTIFICACIÓN	17
2. OBJETIVOS	18
2.1 OBJETIVO GENERAL	18
2.2 OBJETIVOS ESPECÍFICOS	18
3. MARCO DE REFERENCIA	19
3.1 CLOUD COMPUTING	19
3.2 VIRTUALIZACIÓN	22
3.2.1 Tipos de virtualización	22
3.3 KVM (KERNEL-BASED VIRTUAL MACHINE)	23
3.4 ALTA DISPONIBILIDAD	24
3.4.1 Cluster de alta disponibilidad	24
3.5 COPIAS DE SEGURIDAD (BACKUP)	29
3.5.3 Backup diferencial	31
3.6 OPENNEBULA	31
3.7 OPENSTACK	33
3.7.1 Compute (NOVA).	35
3.7.2 Object Storage (SWITCH)	35
3.7.3 Block Storage (CINDER)	35
3.7.4 Networking (NEUTRON)	36
3.7.5 Dashboard (HORIZON)	37
3.7.6 Servicio de identidad (KEYSTONE)	37
3.7.7 Servicio de imagen (GLANCE)	37
3.7.8 Telemetria (CEILOMETER)	38
3.7.9 Orquestación (HEAT)	38
3.7.10 Base de datos (TROVE).	38
3.8 KEEPALIVED	38

4. ANALISIS Y DISEÑO	40
4.1 SITUACION ACTUAL	40
4.1.1 Esquema general	41
4.1.2 Estructura general de cada servidor	42
4.1.4 Hipervisor	44
4.1.5 Opennebula	45
4.2 COPIAS DE SEGURIDAD	45
5. PROTOTIPO	47
5.1 SELECCIÓN DE LA PLATAFORMA	48
5.2 OPENSTACK	52
5.3 DISEÑO DE LA INFRAESTRUCTURA	52
5.4 GESTOR DE BASES DE DATOS	55
5.5 SISTEMA OPERATIVO	56
5.6 INSTALACION Y CONFIGURACION	57
5.6.1 Nodo frontend.	57
5.6.1 Nodo controller.	58
5.6.1 Nodo Neutron	59
5.6.1 Nodo databases	60
5.6.1 Nodo compute (Servidor Físico).	60
5.7 COPIAS DE SEGURIDAD	60
5.7.1 Bacula	61
5.8 PRUEBAS Y VERIFICACION	63
5.8.1 Pruebas keepalived y haproxy	64
5.8.2 Pruebas gestor base de datos y plataforma openstack	66
5.8.3 Verificación funcionamiento plataforma Cloudeisi	68
5.8.4 Verificación funcionamiento del Bacula	75
CONCLUSIONES	79
RECOMENDACIONES	81
REFERENCIAS BIBLIOGRÁFICAS	82
BIBLIOGRAFIA	85

LISTA DE FIGURAS

	Pág.
Figura 1. Modelos de Servicio.	21
Figura 2. Clúster de Alta Disponibilidad de dos Nodos.	25
Figura 3. Esquema de clúster con infraestructura Activo/Pasivo.	26
Figura 4. Esquema de clúster con infraestructura Activo/Activo.	27
Figura 5. Esquema de clúster con infraestructura N+1.	28
Figura 6. Esquema de clúster con infraestructura Split-site.	29
Figura 7. Esquema infraestructura básica OpenNebula.	32
Figura 8. Esquema infraestructura básica OpenStack.	34
Figura 9. Esquema funcionalidad Keepalived.	39
Figura 10. Esquema general de la infraestructura.	41
Figura 11. Estructura general de cada servidor.	42
Figura 12. Componentes Servicio Computación en la Nube.	43
Figura 13. Actividad de Usuarios de cada Comunidad.	49
Figura 14. Participación activa de la Comunidad en los Foros.	50
Figura 15. Población que Contribuye en los Proyectos.	51
Figura 16. Modelo Plataforma CloudEisi.	53
Figura 17. Disposición Plataforma CloudEisi.	54
Figura 18. Estructura Clúster HA MariaDB.	56
Figura 19. Interacción de los elementos del Bacula.	62
Figura 20. Modelo de copias de seguridad	63
Figura 21. Lista de VM's de administración.	64
Figura 22. Nodo activo en el Frontend.	65
Figura 23. Caída del nodo 1 del Frontend.	66
Figura 24. Caída de uno de los nodos de Database y Controller.	67
Figura 25. Prueba de funcionamiento con nodos caídos.	68
Figura 26. Página principal de OpenStack.	69
Figura 27. Modelo de red virtual administrado por Neutron.	70

Figura 28. Acceso a la interfaz de una VM.	71
Figura 29. Instanciación de VM en OpenStack.	72
Figura 30. Lista de instancias disponibles.	72
Figura 31. Ping Instancia Apagada.	73
Figura 32. Encendido maquina DeJong.	74
Figura 33. Ping Instancia Encendida.	74
Figura 34. Backups realizados.	75
Figura 35. Estadística backups realizados.	76
Figura 36. Estadística backups realizados en los diferentes volúmenes de archivos.	76
Figura 37. Estadística tamaño de archivos salvaguardados.	77
Figura 38. Copias de seguridad realizados al MeiWEB.	78

LISTA DE ANEXOS

	Pág.
ANEXO A. ARTÍCULO	88

RESUMEN

TITULO: Análisis de la arquitectura de computación en la nube CloudEisi con énfasis en el mejoramiento de la administración, funcionalidad y seguridad.*

AUTORES:

CARREÑO ORTIZ, Yonathan Nahún **

PÁEZ CRUZ, José Alonso**

PALABRAS CLAVE: Virtualización, computación en la nube, OpenStack.

DESCRIPCIÓN: La computación en la nube es un paradigma cuyo uso que se ha venido implementando en los últimos años de forma incrementada y se espera que siga esta tendencia en los siguientes años. Este paradigma básicamente consiste en la prestación de servicios informáticos a través de internet sin necesidad de que los usuarios tengan conceptos avanzados sobre los recursos que utilizan, para ello se requieren de servidores en internet que atiendan peticiones en cualquier momento.

En la actualidad existen numerosas plataformas que prestan servicios de computación en la nube tales como las de Google, Amazon y Microsoft, las cuales son plataformas a gran escala, estas por mencionar algunas, pero existen una gran cantidad de plataformas de menor magnitud; esto debido a los numerosos software de aplicación existentes para gestionar dicho tipo de plataformas, lo cual permite que con un pequeño número de servidores, la tecnología adecuada y acceso a internet se pueda ofrecer servicios en la nube, como ejemplo de esto se tiene la plataforma CloudEisi del grupo CONUSS objeto de este trabajo.

En el presente trabajo se hace uso de tecnologías emergentes relacionadas a la computación en la nube, la cual permita generar un nuevo modelo con el fin de optimizar la funcionalidad de la plataforma CloudEisi la cual alberga un conjunto de servicios que benefician estudiantes a y profesores de la comunidad académica. Este objetivo se logra mediante la redefinición de sus sistemas base de alta disponibilidad y de computación en la nube.

* Trabajo de grado modalidad de investigación.

** Facultad de ingenierías físico-mecánicas. Escuela de ingeniería de sistemas e informática. Director: MSc. Manuel Guillermo Flórez Becerra

ABSTRACT

TITLE: Analysis of the architecture of cloud computing CloudEisi with emphasis on improving management, functionality and safety^{*}

AUTHORS:

CARREÑO ORTIZ, Yonathan Nahún^{**}

PÁEZ CRUZ, José Alonso^{**}

KEYWORDS:

Virtualization, cloud computing, OpenStack.

DESCRIPTION:

Cloud computing is a paradigm that has been implemented recently and its use is expected to increase over the next years. This paradigm is basically the provision of computer services over the Internet without users having advanced concepts about the resources they use, this will require servers that attend requests at any time.

Currently there are numerous platforms, on a large scale, that provide cloud computing services such as: Google, Amazon y Microsoft. In addition, there is a lot of platforms, of lesser magnitude, due to the numerous application software available to manage them, allowing with a small number of servers, the right technology and internet access can offer cloud services such as the group CloudEisi CONUSS platform, object of this work.

In this work, emerging technologies related to cloud computing are used, which allow the generation of a new model, in order to optimize the functionality of the CloudEisi platform that offers a set of services that benefit the academic community. This is achieved by redefining its basis high-availability systems and cloud computing.

^{*} Undergraduate Final Project

^{**} Physical-Mechanical Engineering Faculty. Systems and Informatics Engineering School. Project director MSc. Manuel Guillermo Flórez Becerra

INTRODUCCION

La computación en la nube es una tecnología en continuo crecimiento, muy acogida por grandes empresas por su eficiente administración de la información y recursos. Según el IEEE Computer Society, es un paradigma en el que la información se almacena de manera permanente en servidores de Internet y se envía a cachés temporales de cliente, lo que incluye equipos de escritorio, centros de ocio, portátiles, etc. [1] Inspirado su modelo de los cajeros automáticos, en el cual está disponible el dinero desde cualquier punto de servicio, su implementación en la red origino “CloudComputing”, tecnología que gracias a la virtualización y la computación paralela es capaz de brindar servicios a usuarios en tiempo real, totalmente estable y seguro gracias a su infraestructura de alta disponibilidad.

Esta tecnología se clasifica en tres clases, nube publica, nube privada y nube hibrida; Amazon, Windows Azure y Google App Engine se consideran nubes públicas. Un ejemplo de nube privada es el CloudEisi, la cual es administrada por el grupo GID-CONUSS de la Universidad Industrial de Santander.

El presente trabajo busca evaluar la arquitectura implementada de esta tecnología en el CloudEisi, con el fin de proponer un modelo de computación en la nube el cual mejore en primera instancia aspectos de seguridad y funcionalidad de dicha nube.

1. JUSTIFICACIÓN

El grupo de investigación GID-CONUSS cuenta con una infraestructura de computación en la nube de alta disponibilidad; La cual presta servicios de instancias virtuales a la comunidad universitaria.

La infraestructura existente posee ineficiencias debido a la mal utilización de los recursos de cómputo, manifestado en la lentitud del servicio prestado; Problemas en el respaldo de la información almacenada en la nube, ocasionando en algunos casos pérdida completa o parcial de dicha información.

En el desarrollo del proyecto se evaluara y construirá un modelo de computación en la nube que permita 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.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Análisis de la arquitectura de computación en la nube CloudEisi con énfasis en el mejoramiento de la administración, funcionalidad y seguridad.

2.2 OBJETIVOS ESPECÍFICOS

- Continuar con el proceso actual de administración, mantenimiento, configuración, actualización y monitoreo de los servicios de computación en la nube del CloudEisi.
 - ✓ Entrenamiento completo del funcionamiento de dicha infraestructura.
 - ✓ Monitoreo del sistema.
 - ✓ Monitoreo de los servidores, clúster de alta disponibilidad y servicios de computación en la nube.
 - ✓ Realizar tareas de mantenimiento programadas y copias de seguridad.
 - ✓ Recuperación del sistema o servicios en caso de fallo.
 - ✓ Realizar tareas relacionadas con atención a los usuarios de los diferentes servicios.
- Comparar las plataformas Open Source de computación en la nube con la implementada en el CloudEisi.
- Crear Un modelo de computación en la nube que sea flexible y escalable, basado en el análisis de las plataformas Open Source.
- Realizar las actualizaciones necesarias de software y corregir posibles fallos de seguridad en los servidores.
- Analizar e implementar un sistema de Backups en los servidores del CloudEisi.
- Socializar los resultados del proyecto mediante un artículo con los resultados y las recomendaciones para próximas investigaciones y desarrollos

3. MARCO DE REFERENCIA

3.1 CLOUD COMPUTING

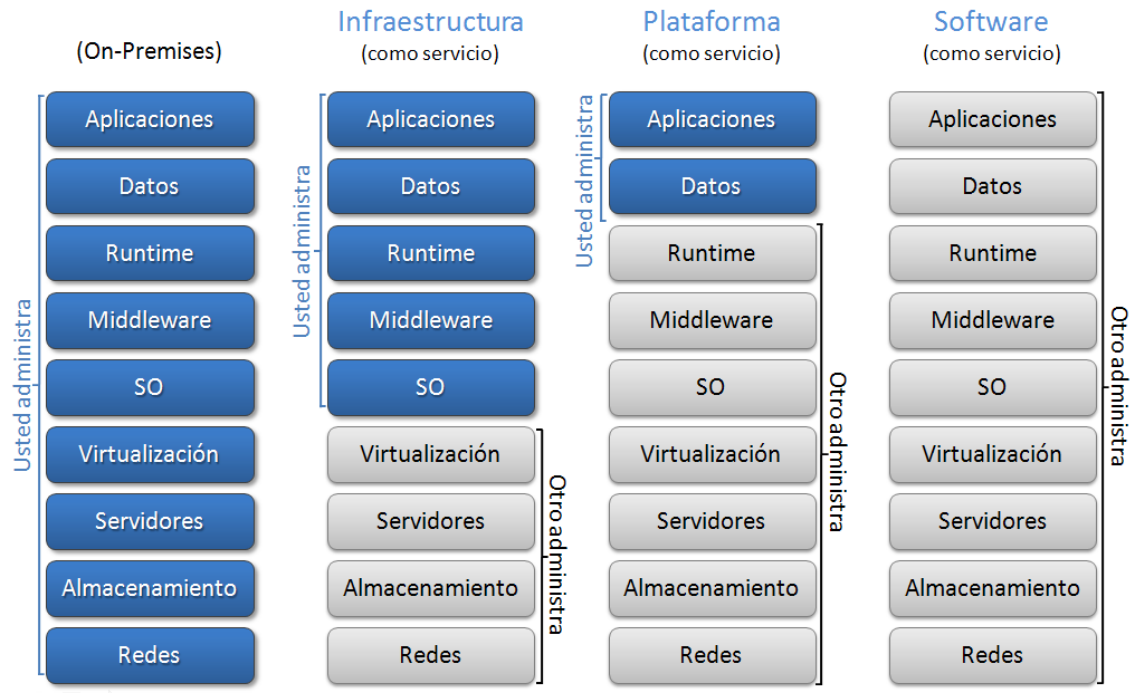
El concepto de computación en la nube se empezó a utilizar en 2006 por proveedores como Google, Amazon, Microsoft las cuales habían construido sistemas de recursos distribuidos de manera horizontal, introducidos como servicios virtuales de TI, escalados de manera continua el cual se reflejó en el artículo “las fábricas de información” de la revista Wired Magazine, escrito por George Gilder; en el cual plantea un modelo de nube virtual, similar en estructura a la computación grid [3].

El NIST (Instituto Americano de Estándares y Tecnología) como un modelo que permite el acceso bajo la demanda a través de la red a un conjunto compartido de recursos de computación configurables que pueden rápidamente aprovisionados con el mínimo esfuerzo de gestión e iteración del proveedor de servicio enfocado a su uso en la web permitiendo el acceso a un conjunto compartido de recursos de computación configurables que pueden proveerse y liberarse rápidamente de administración basándose en la virtualización, ofreciendo los siguientes capas generales de servicios:

- **Software como servicio (SaaS):** Consiste en el suministro de aplicaciones software del proveedor, utiliza una arquitectura web accesible desde cualquier dispositivo y el usuario no tiene ningún control sobre la infraestructura; en este tipo de servicio el usuario se ve comprometido a mantener actualizadas las licencias entre las empresas que ofrecen este tipo de servicios se encuentran SaleForce y BaseCamp.

- **Infraestructura como servicio (IaaS):** Ofrece servicios de computación y almacenamiento en lo relativo al aprovisionamiento de servidores y discos donde el usuario alquila recursos de computación comportándose como la capa física de la nube el proveedor se encarga de toda el mantenimiento y optimización de los recursos del equipamiento hardware, entre las cuales se destacan empresas como Amazon Web Service, EC2 y GoGrid.
- **Plataforma como Servicios(PaaS):** El proveedor ofrece un entorno para facilitar el trabajo de los programadores, encapsulando el entorno de desarrollo facilitando el ciclo de construcción y puesta en marcha de aplicaciones, manteniendo la seguridad y escalabilidad de los programas, la plataforma como servicio (PaaS) está muy ligada a la capa software como servicio (SaaS) porque esta es la plataforma donde se desenvuelve el software que está a disposición del proveedor y es el medio de virtualización para el hardware que el cliente alquila y todo se ejecuta desde la plataforma externa y el usuario no necesita descargar nada, una de las empresas que ofrece este servicio el Google App Engine.

Figura 1. Modelos de Servicio.



Fuente: Carreño Díaz, Emmanuel. 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 informática, 2012. 25 p. [7]

En la Figura 1 se puede observar una comparación directa entre los diferentes tipos de modelos de servicios de computación en la nube y se indican los niveles de administración permitida al cliente y los que son responsabilidad del proveedor del servicio, junto al modelo tradicional servicio. On-premises quiere decir que el servicio de cómputo es alojado en las instalaciones del usuario, por lo tanto este administra la totalidad de los componentes, algo que puede ser innecesario según el tipo de empresa, la administración de redes, almacenamiento, servidores, hipervisores, sistemas operativos, middleware (software intermediario), runtimes (entornos de ejecución), aplicaciones y datos pueden ser manejados por un tercero según el modelo seleccionado.

3.2 VIRTUALIZACIÓN

Virtualización es uno de los pilares de la computación en la nube, es el uso de los recursos de las computadoras para simular o imitar a otros recursos de ellas o las propias computadoras en su totalidad. La virtualización es un mecanismo fundamental para la entrega de servicios; proporciona una plataforma para la optimización de recursos de TI complejos de un modo escalable (crecimiento de recursos de modo eficiente y adaptable a los servicios a entregar) [2].

3.2.1 Tipos de virtualización. El concepto de virtualización permite su uso sobre múltiples recursos, para efectos de este proyecto se utilizó la virtualización por hardware, se ha conseguido por dos métodos distintos, virtualización parcial y virtualización completa.

Paravirtualización es una técnica en la cual el hipervisor presenta un API al sistema operativo invitado y éste 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 parcial es una técnica que simula parte del entorno de hardware, a diferencia del tipo de virtualización detallado a continuación. Es conocida como "Address Space Virtualization" debido a que simula espacios de direcciones. Se comparten recursos y procesos pero no es posible crear instancias separadas de sistemas operativos.

La **virtualización completa** está diseñada para proporcionar abstracción total del sistema físico fundamental y crea un sistema virtual completo en que se puede ejecutar el sistema operativo huésped (guest). No se requiere ninguna modificación en el SO o aplicación huésped. El SO o aplicación cliente no aprecia el entorno virtualizado, de modo que se puede ejecutar en la máquina virtual (VM)

como si ella fuera un sistema físico. Este enfoque puede ser ventajoso ya que facilita el desarrollo completo del software del hardware.

La virtualización completa puede racionalizar la migración de aplicaciones y cargas de trabajo entre diferentes sistemas físicos y también ayuda a hacer este enfoque muy seguro.

VMware Workstation/Server, Windows Server 2008 R2 Hyper-V, VirtualBox de Oracle, Linux KVM y Microsoft Virtual PC² son ejemplos 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.

3.3 KVM (KERNEL-BASED VIRTUAL MACHINE)

KVM es una solución para implementar virtualización completa con Linux. Está formada por un módulo del núcleo (con el nombre `kvm.ko`) y herramientas en el espacio de usuario, siendo en su totalidad software libre. El componente KVM para el núcleo está incluido en Linux desde la versión 2.6.20.

KVM permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio hardware virtualizado: Una tarjeta de red, discos duros, tarjeta gráfica, etc. [5] KVM será la base de virtualización implementada por ser software libre y además contar con una vasta documentación gracias a la comunidad Linux.

El auge de los hipervisores libres como KVM o Xen, hacen que la integración con los sistemas Linux esté muy optimizada y los resultados de muy alta calidad siendo un mercado en donde Linux está bien posicionado.

Como emblemas actuales de plataformas en la nube tenemos en la actualidad los siguientes productos: OpenStack, CloudStack, OpenNebula o Eucaliptus.

Estos productos ofrecen la flexibilidad de la comunidad: acceso a la comunidad, recursos compartidos, actualizaciones rápidas y libertad en la modificación y adaptación de los productos a nuestras necesidades actuales así como una completa interoperabilidad entre plataformas y una estandarización al acceso de los recursos.

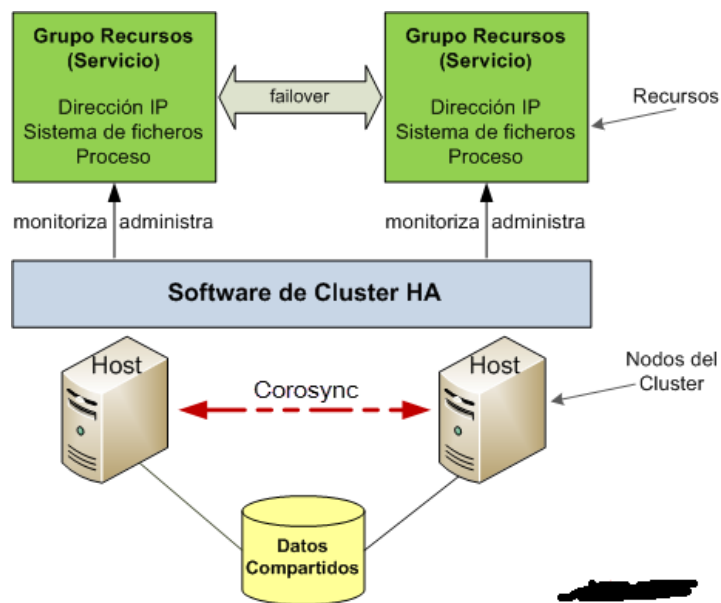
3.4 ALTA DISPONIBILIDAD

Tomando la definición expuesta por el IEEE TFCC¹, 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 otro extremo con soluciones que ofrecen las soluciones de software que utilizan componentes de hardware genéricos.”[10]

3.4.1 Cluster de alta disponibilidad. Un clúster de alta disponibilidad es una arreglo de 2 o más equipos servidores que están continuamente monitoreándose entre sí. Comparten ciertos recursos y reaccionan frente a eventos inesperados aplicando acciones que aseguran la disponibilidad de los servicios. La finalidad de este tipo de clúster es eliminar puntos de fallo mediante redundancia en niveles como hardware, el almacenamiento, la red, las aplicaciones etc.

Existe una gran diferencia, para evitar confusiones, entre clúster de alta disponibilidad y clúster de alto rendimiento. El segundo es una configuración de equipos diseñado para proporcionar capacidades de cálculo mayores que la que proporcionan los equipos individuales, mientras que el primer tipo de clúster está diseñado para garantizar el funcionamiento ininterrumpido de ciertas aplicaciones. El siguiente esquema presenta un modelo de clúster de alta disponibilidad con dos nodos (equipos servidores), donde muestra el funcionamiento básico del clúster.

Figura 2. Clúster de Alta Disponibilidad de dos Nodos.



Fuente: <http://www.lintips.com/>

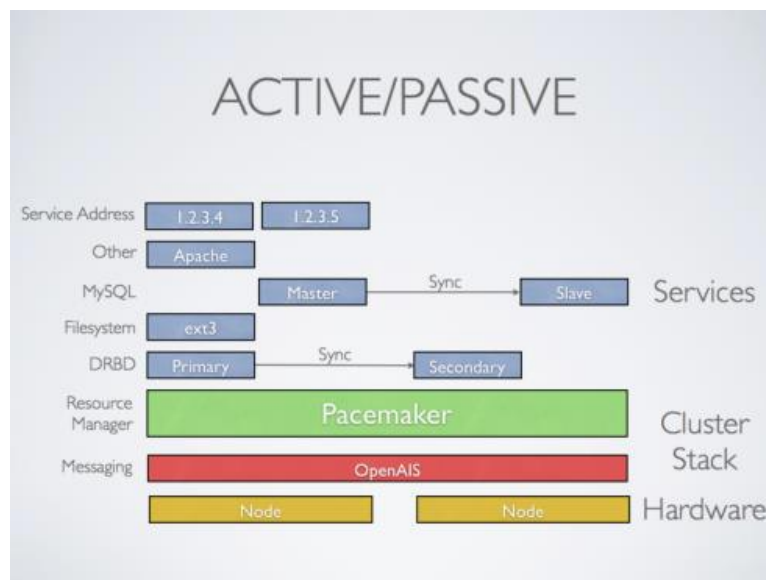
En la figura se observan dos nodos (Hosts), los cuales comparten datos que se encuentran dentro de un disco o partición específica de sincronización, que deben tener el mismo tamaño y características, para que dichos datos estén sincronizados constante y correctamente. Además de datos, se puede configurar y asegurar la disponibilidad de los recursos (servicios) instalados en ambos nodos. Los datos y los recursos se sincronizan a través del software de gestión del clúster y del uso de un demonio de comunicación que pasa información constantemente

entre los nodos del clúster acerca del estado de los servicios y recursos compartidos.

La infraestructura base del clúster se puede implementar de diferentes maneras y con diferente software según sea el propósito o la disposición requerida de los recursos. A continuación se muestra algunas de las infraestructuras más utilizadas en el diseño de un clúster:

- **Activo/Pasivo:** Uno de los nodos del clúster tiene activos los servicios y los sistemas de archivos compartidos, el otro, los mantiene en espera en caso de un fallo.

Figura 3. Esquema de clúster con infraestructura Activo/Pasivo.

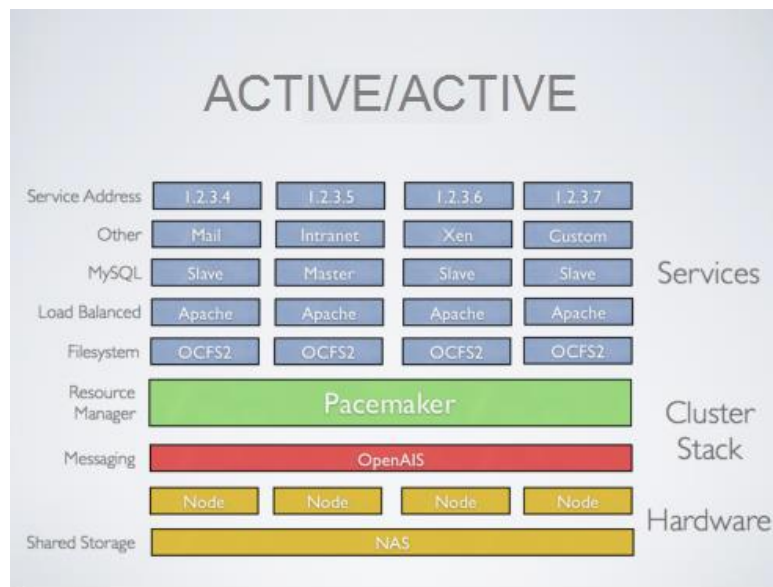


Fuente: <http://www.clusterlabs.org/>

- **Activo/Activo:** Todos los nodos del clúster se encuentran en estado activo, esto permite que todos los nodos sean utilizados potencialmente en caso de

fallo, corriendo simultáneamente múltiples copias de los servicios y permitiendo así el balanceo de carga de trabajo entre los distintos nodos.

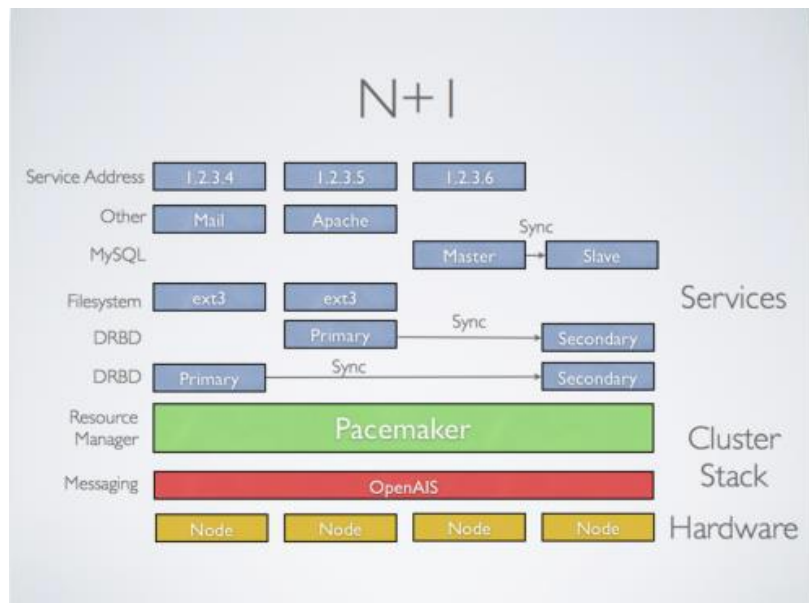
Figura 4. Esquema de clúster con infraestructura Activo/Activo.



Fuente: <http://www.clusterlabs.org/>

- **N+1:** Esta infraestructura puede contar con más de dos nodos. Cuenta con varios nodos Activos/Pasivos, teniendo un nodo de backup común compartido, lo cual ayuda a reducir costos de hardware en los equipos.

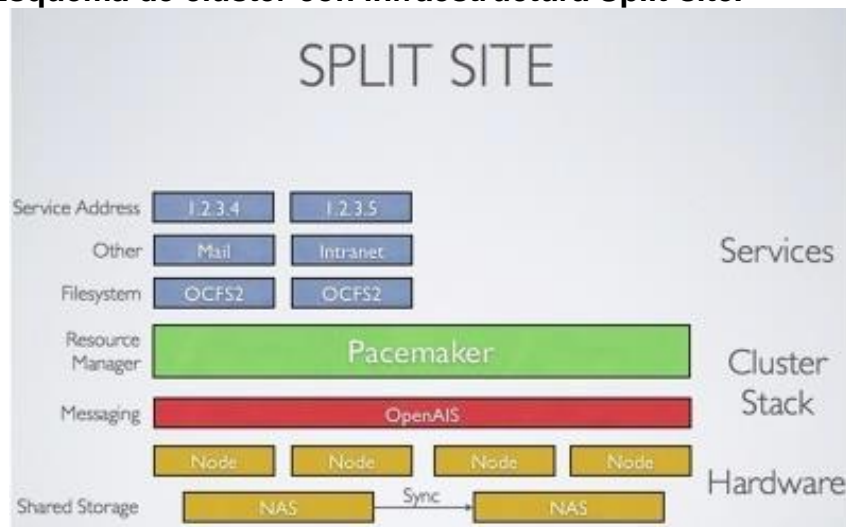
Figura 5. Esquema de clúster con infraestructura N+1.



Fuente: <http://www.clusterlabs.org/>

- **Split-Site Clúster:** Un clúster de esta característica tiene compartido el almacenamiento de datos por red en los diferentes nodos del clúster, mientras que los servicios se encuentran activos en uno o más nodos y los omite en los otros.

Figura 6. Esquema de clúster con infraestructura Split-site.



Fuente: <http://www.clusterlabs.org/>

3.5 COPIAS DE SEGURIDAD (BACKUP)

Es una copia de los datos originales que se realiza con el fin de disponer de un medio para recuperarlos en caso de su pérdida. Las copias de seguridad son útiles ante distintos eventos y usos: recuperar los sistemas informáticos y los datos de una catástrofe informática, natural o ataque; restaurar una pequeña cantidad de archivos que pueden haberse eliminado accidentalmente, corrompido, infectado por un virus informático u otras causas.

El proceso de copia de seguridad se complementa con otro conocido como restauración de los datos (en inglés restore), que es la acción de leer y grabar en la ubicación original u otra alternativa los datos requeridos.

El objeto de la mayor parte de los backups consiste en crear una copia de los datos, de forma que se pueda restaurar un archivo o aplicación concretos tras la pérdida o supresión de los datos o debido a un desastre. Por lo tanto, el backup no

es el objetivo, sino un medio de cumplir el objetivo de proteger los datos. La comprobación de los backups es tan importante como copiar y restaurar los datos.

Hace mucho que las aplicaciones de backup ofrecen varios tipos de operaciones de backup. Los tipos de backup más corrientes son el backup completo, el backup incremental y el backup diferencial.

3.5.1 Backups completos. Como su propio nombre indica, este tipo de backup copia la totalidad de los datos en otro juego de soportes, que puede consistir en cintas, discos, o en un DVD o CD. La ventaja principal de la realización de un backup completo en cada operación es que se dispone de la totalidad de los datos en un único juego de soportes. Esto permite restaurar los datos en un tiempo mínimo, lo cual se mide en términos de objetivo de tiempo de recuperación (RTO). No obstante, el inconveniente es que lleva más tiempo realizar un backup completo que de otros tipos, y requiere más espacio de almacenamiento.

Por lo tanto, sólo se suelen realizar backups completos periódicamente. Los centros de datos que manejan un volumen de datos (o de aplicaciones críticas) reducido pueden optar por realizar un backup completo cada día, o más a menudo aún en ciertos casos. Lo normal es que en las operaciones de backup se combine el backup completo con backups incrementales o diferenciales.

3.5.2 Backup incremental. Una operación de backup incremental sólo copia los datos que han variado desde la última operación de backup de cualquier tipo. Se suele utilizar la hora y fecha de modificación estampada en los archivos, comparándola con la hora y fecha del último backup. Las aplicaciones de backup identifican y registran la fecha y hora de realización de las operaciones de backup para identificar los archivos modificados desde esas operaciones.

Como un backup incremental sólo copia los datos a partir del último backup de cualquier tipo, se puede ejecutar tantas veces como se desee, pues sólo guarda los cambios más recientes. La ventaja de un backup incremental es que copia una menor cantidad de datos que un backup completo. Por ello, esas operaciones se realizan más deprisa y exigen menos espacio para almacenar el backup.

3.5.3 Backup diferencial. Una operación de backup diferencial es similar a un backup incremental la primera vez que se lleva a cabo, pues copiará todos los datos que hayan cambiado desde el backup anterior. Sin embargo, cada vez que se vuelva a ejecutar, seguirá copiando todos los datos que hayan cambiado desde el anterior completo. Por lo tanto, en las operaciones subsiguientes almacenará más datos que un backup incremental, aunque normalmente muchos menos que un backup completo. Además, la ejecución de los backups diferenciales requiere más espacio y tiempo que la de los backups incrementales, pero menos que la de los backup completos.

3.6 OPENNEBULA

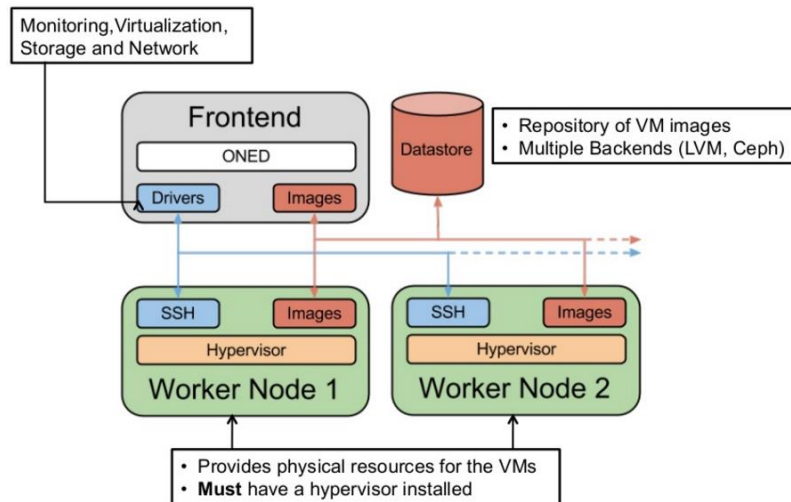
OpenNebula es una solución Open Source (bajo licencia Apache v2) que permite implementar fácilmente infraestructuras Cloud Computing privadas (también híbridas) según el modelo IaaS. Su parte principal consiste en software que permite desplegar máquinas virtuales sobre un pool de máquinas físicas. Además está diseñado para integrarse con otras soluciones de almacenamiento y de red. Así, maneja tanto la transferencia de máquinas virtuales como la configuración de la red, el almacenamiento y su gestión.

OpenNebula fue inicialmente desarrollado por la Universidad Complutense de Madrid en 2008. Más adelante el número de participantes ha ido creciendo y más

organizaciones se han unido al desarrollo del proyecto. Algunos destacados contribuidores son IBM, Suse o AT&T.

Proporciona una plataforma de Cloud Computing escalable, segura y rápida de desplegar y permite manejar y construir infraestructuras públicas, privadas e híbridas siguiendo el modelo IaaS. Es una solución que se ajusta muy bien a los data-centers actuales.

Figura 7. Esquema infraestructura básica OpenNebula.



Fuente: <http://www.opennebula.org/>

Los siete pilares básicos de OpenNebula son:

- Almacenamiento: Permite almacenar las imágenes de discos virtuales en repositorios desde donde serán usadas para desplegar rápidamente máquinas virtuales o compartidas con otros usuarios. Estas imágenes de disco pueden ser tanto de Sistemas Operativos o de datos.

- Repositorio de plantillas: Es donde se almacenan plantillas de máquinas virtuales, con sus características para ser instanciadas más tarde en el hipervisor elegido.
- Redes virtuales: Soporta el manejo de redes virtuales que interconectará las diferentes máquinas virtuales, pudiendo definir una IP concreta o rangos de IP para cada red.
- Manejo de máquinas virtuales: Una vez que se ha desplegado una instancia de una plantilla en un hipervisor (host) se puede controlar todo su ciclo de vida como el arranque, parada, clonación y apagado.
- Clústeres: Son pools de host que comparten almacenamiento y redes virtuales. Se utilizan para el balanceo de carga y alta disponibilidad y rendimiento. OpenNebula y Hadoop: Cloud Computing con herramientas Open Source 23
- Usuarios y grupos: Soporta la definición de usuarios y grupos para el acceso a los recursos así como mecanismo de autenticación. También implementa ACL para la asignación permisos.
- API: Proporciona interfaces de comunicación con las diferentes funcionalidades ofrecidas a través desde herramientas de línea de comando o través del GUI Web Sunstone. Además proporciona interfaces para interactuar con otras infraestructuras Cloud Computing públicas como OCCi7 y EC28 con lo que permite el despliegue de nubes híbridas.

3.7 OPENSTACK

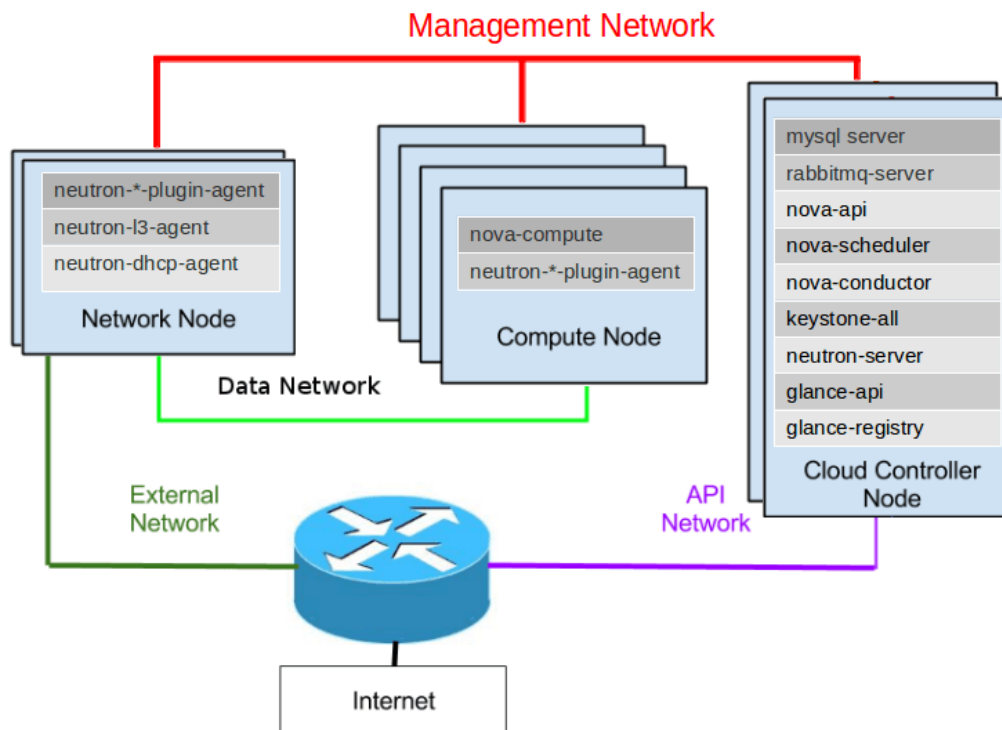
OpenStack es un proyecto de computación en la nube para proporcionar una infraestructura como servicio (IaaS). Es un software libre y de código abierto

distribuido bajo los términos de la licencia Apache. El proyecto está gestionado por la Fundación OpenStack, una organización sin fines de lucro creada en septiembre de 2012 para promover el software OpenStack y su comunidad.

La tecnología consiste en una serie de proyectos relacionados entre sí que controlan estanques de control de procesamiento, almacenamiento y recursos de red a través de un centro de datos, todos administrados a través de un panel de control que permite a los administradores controlar mientras potencia a sus usuarios proveyendo los recursos a través de una interfaz web.

El software tiene una arquitectura modular con varios nombres para sus componentes.

Figura 8. Esquema infraestructura básica OpenStack.



Fuente: <http://www.dbigcloud.com/>

3.7.1 Compute (NOVA). Nova es un controlador de estructura cloud computing, que es la parte principal de un sistema de IaaS. Está diseñado para gestionar y automatizar los pools de los recursos del equipo y puede trabajar con tecnologías ampliamente disponibles de virtualización. KVM y Xen son las opciones disponibles para la tecnología de hipervisor, junto con la tecnología Hyper-V, la tecnología vSphere de VMware y la tecnología de contenedores Linux como LXC.

3.7.2 Object Storage (SWITCH)

- Es un sistema de almacenamiento redundante y escalable.
- Los objetos y los archivos se escriben en varias unidades de disco repartidos por los servidores del centro de datos, con el software OpenStack responsable de asegurar la replicación y la integridad de los datos en el clúster. Agrupaciones de almacenamiento escalar horizontalmente simplemente añadiendo nuevos servidores.
- En caso de que un servidor o disco duro falla, OpenStack replica su contenido desde otros nodos activos a nuevas ubicaciones en el clúster.
- Debido a que OpenStack utiliza la lógica del software para asegurar la replicación de datos y la distribución a través de diferentes dispositivos, discos duros y servidores de bajo costo pueden ser utilizados.

3.7.3 Block Storage (CINDER)

- Este módulo proporciona dispositivos de almacenamiento a nivel de bloque persistentes para usar con instancias de OpenStack Compute.
- El sistema de almacenamiento de bloques gestiona la creación, aplicación y el desprendimiento de los dispositivos de bloque a los servidores.
- Volúmenes de almacenamiento de bloque se integran plenamente en OpenStack Compute y el Dashboard que permite a los usuarios en la nube gestionar sus propias necesidades de almacenamiento. Además del

almacenamiento del servidor local de Linux, puede utilizar las plataformas de almacenamiento incluyendo Ceph, CloudByte, Coraid, EMC (VMAX y VNX), GlusterFS, Hitachi Data Systems, IBM Storage (familia Storwize, controlador de volumen SAN, XIV Storage System, y GPFS), Linux LIO, NetApp, Nexenta, Scality, SolidFire, HP (StoreVirtual y 3PAR StoreServ familias) y almacenamiento puro.

- El almacenamiento de bloques es apropiado para escenarios donde el rendimiento es sensible, tales como el almacenamiento de base de datos, sistemas de archivos expandibles, o la prestación de un servidor con acceso al almacenamiento a nivel de bloque en bruto.

3.7.4 Networking (NEUTRON)

- Este módulo proporciona modelos de redes para diferentes aplicaciones o grupos de usuarios.
- Los modelos estándar incluyen redes planas o VLAN para la separación de los servidores y el tráfico. Gestiona las direcciones IP, lo que permite direcciones IP estáticas o DHCP reservados.
- Direcciones IP flotantes permiten que el tráfico se redirija dinámicamente a cualquiera de sus recursos informáticos, que permite redirigir el tráfico durante el mantenimiento o en caso de fracaso.
- Los usuarios pueden crear sus propias redes, controlar el tráfico y conectar los servidores y los dispositivos a una o más redes. Los administradores pueden aprovechar las redes definidas por software de tecnología (SDN) como OpenFlow para permitir altos niveles de multiempresa y escala masiva.
- Neutrón tiene un marco que permite la extensión de servicios de red adicionales, como los sistemas de detección de intrusos (IDS), balanceo de carga, cortafuegos y redes privadas virtuales (VPN) para ser implementada y administrada.

3.7.5 Dashboard (HORIZON)

- Horizon proporciona a los administradores y usuarios una interfaz gráfica para el acceso, la provisión y automatización de los recursos basados en la nube.
- El diseño permite que los productos y servicios de terceros, tales como la facturación, el monitoreo y las herramientas de gestión adicionales. El Dashboard es sólo una forma de interactuar con los recursos de OpenStack.
- Los desarrolladores pueden automatizar el acceso o construir herramientas para gestionar sus recursos mediante la API nativa de OpenStack o la API de compatibilidad EC2.

3.7.6 Servicio de identidad (KEYSTONE)

- Keystone ofrece un directorio central de usuarios asignados a los servicios de OpenStack que pueden acceder.
- Actúa como un sistema de autenticación común en todo el sistema operativo para la nube y se puede integrar con los servicios de directorio backend existentes como LDAP.
- Es compatible con múltiples formas de autenticación, incluyendo nombre de usuario y contraseña de credenciales estándar, sistemas basados en tokens e inicios de sesión (login) de estilo AWS (es decir, Amazon Web Services). Además, el catálogo incluye una lista consultable de todos los servicios existentes en la OpenStack cloud, en un solo registro.
- Los usuarios y las herramientas de terceros mediante programación pueden determinar qué recursos pueden acceder.

3.7.7 Servicio de imagen (GLANCE)

- Glance proporciona servicios de descubrimiento, de inscripción y de entrega de los discos y del servidor de imágenes. Las imágenes almacenadas se pueden utilizar como una plantilla. También se puede utilizar para almacenar y catalogar un número ilimitado de copias de seguridad.

- El servicio de imagen puede almacenar imágenes de disco y de servidores en una variedad de back-ends, incluyendo OpenStack Object Storage.
- La API de servicios de imagen proporciona una interfaz REST estándar para consultar información sobre las imágenes de disco y permite a los clientes transmitir las imágenes a nuevos servidores.

3.7.8 Telemetría (CEILOMETER). Ceilometer proporciona un único punto de contacto para los sistemas de facturación, proporcionando todos los contadores que se necesitan para establecer la facturación del cliente, a través de todos los componentes actuales y futuras de OpenStack. La entrega de los contadores es trazable y auditable, los contadores deben ser fácilmente extensible para apoyar nuevos proyectos, y los agentes que realizan las colecciones de datos deben ser independientes de todo el sistema.

3.7.9 Orquestación (HEAT). Heat es un servicio para orquestar múltiples aplicaciones compuestas en la nube utilizando plantillas, tanto a través de una API REST OpenStack nativa y una API de consultas compatibles con CloudFormation.

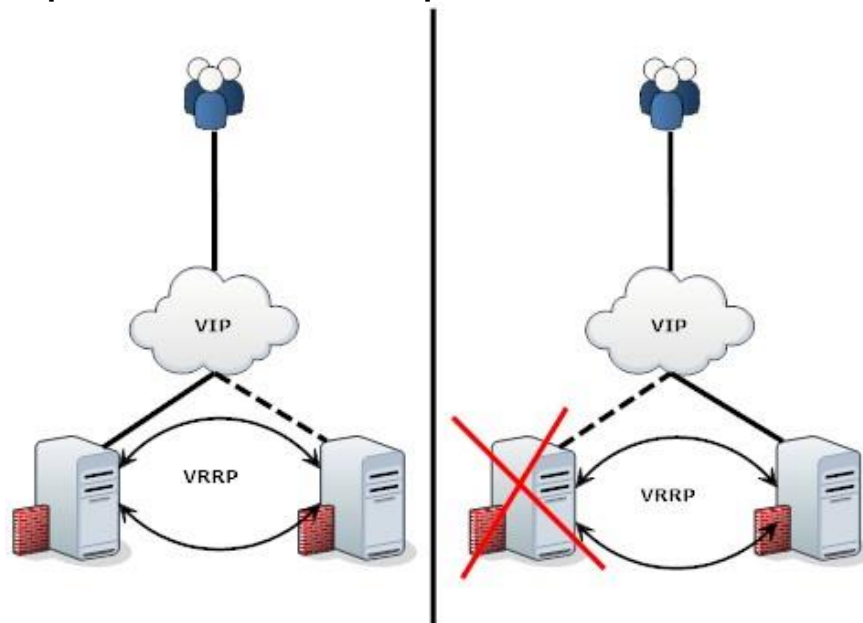
3.7.10 Base de datos (TROVE). Trove es una base de datos que funciona como un servicio de aprovisionamiento de motores de bases de datos relacionales y no relacionales.

3.8 KEEPALIVED

Keepalived es un software de enrutamiento escrito en C. El objetivo principal de este proyecto es proporcionar simples y robustos servicio para balanceo de carga y de alta disponibilidad para los sistemas Linux y las infraestructuras basados en Linux. El Framework de balanceo de carga depende del conocido y ampliamente utilizado servidor virtual de Linux (IPVS) módulo del kernel que proporciona la capa 4 (modelo OSI) al balanceo de carga. Keepalived implementa un conjunto de

herramientas para mantener y administrar de forma dinámica y adaptativa el balanceo de carga de un grupo de servidores de acuerdo a su estado. Por otro lado la alta disponibilidad se consigue mediante protocolo VRRP.

Figura 9. Esquema funcionalidad Keepalived.



Fuente: los autores

4. ANALISIS Y DISEÑO

Este capítulo se encuentra estructurado de la siguiente manera, inicialmente se hace una revisión del estado de los recursos actuales del grupo CONUSS y se definen los diferentes grupos objetivos de este proyecto, luego se realiza una clasificación de los productos finales de la producción académica de los trabajos de los estudiantes de pregrado de los últimos 3 años; finalmente con el apoyo de esta información se determinan los tipos de servicios de computación en la nube requeridos.

4.1 SITUACION ACTUAL

La infraestructura como servicio CloudEisi le presta a la comunidad EISI el servicio de instanciación de máquinas virtuales para el desarrollo de proyectos de grado, actividades de investigación, hosting para asignaturas que implementan aplicaciones web, repositorio de aplicaciones web, repositorio de imágenes digitales, servicios de aulas virtuales LMS (Learning Management System). Está soportada por tres servidores de altas prestaciones, su esquema cuenta con las siguientes características generales:

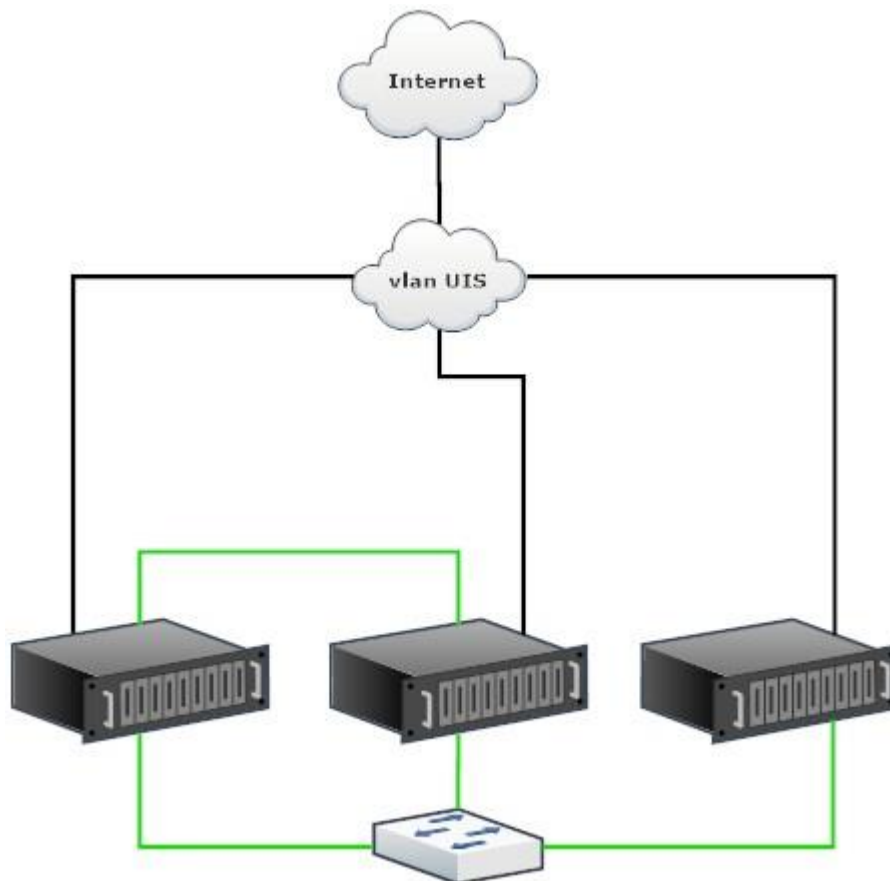
- Sistemas operativos basados en Linux tipo servidor.
- Uso de máquinas virtuales configuradas con los componentes base de la infraestructura.
- Sistema de alta disponibilidad para asegurar el continuo servicio en caso fallas por parte de uno de los nodos que pertenecen al clúster.
- Servicio de asignación de instancias virtuales.
- Repositorio de imágenes virtuales preconfiguradas con servicios listos para ser usados.

- Interfaz administrativa para manejo de recursos virtuales.
- Acuerdo de prestación de servicios con los usuarios.
- Esquema de red interna virtual para la comunicación entre instancias virtuales.
- Punto de salida para la comunicación hacia internet de los servicios virtuales.

Dicha infraestructura ha estado evolucionando desde su creación como un clúster de alta disponibilidad, hoy día es un clúster de alta disponibilidad que ofrece servicios en la nube.

4.1.1 Esquema general. El esquema físico de los equipos está dado como se muestra en la siguiente figura:

Figura 10. Esquema general de la infraestructura.



Fuente: Los Autores

Los hosts se conectan a la red de la universidad, y ésta brinda el servicio de conexión a internet que permite dar salida a los servicios de cada servidor. La conexión dedicada presente entre los dos servidores es usada por el sistema de alta disponibilidad para respaldar el componente de Computación en la Nube. El tercer servidor es empleado para soportar la instanciación de máquinas virtuales en la Nube, junto con los otros dos servidores, y además se emplea como nodo de repositorio de imágenes y almacén de copias de seguridad.

4.1.2 Estructura general de cada servidor. La composición interna de cada servidor está dada por un enfoque basado en componentes.

Figura 11. Estructura general de cada servidor.

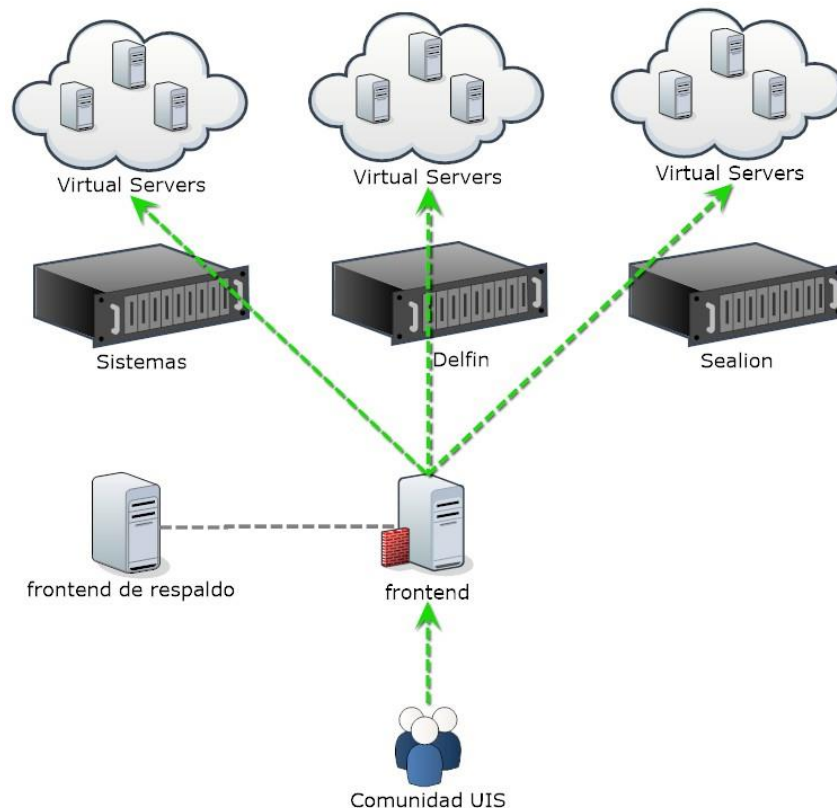


Fuente: Los Autores

Con esta estructura se obtiene un diseño simple y de fácil manejo, ya que los servicios no se están ejecutando de forma nativa en cada servidor, si no en instancias virtuales individuales, por lo que su configuración, respaldo, migración y recuperación no afectan de forma total los demás componentes de la infraestructura.

4.1.3 Estructura computación en la nube. Esta plataforma de computación en la Nube CloudEisi tiene como usuarios finales a los miembros de la comunidad UIS. El punto de acceso del servicio es una máquina virtual configurada como frontend, esta instancia virtual es la encargada de administrar y monitorear las máquinas virtuales usadas por los usuarios. Como las instancias de servicio pueden estar en diferentes servidores, el frontend tiene la tarea de realizar el direccionamiento del tráfico a través de la red interna de la infraestructura. Ante la posible falla del frontend es posible reanudar el servicio de la infraestructura gracias a un frontend de respaldo que es encendido de forma manual y toma el control de la infraestructura, dicho frontend de respaldo se encuentra instanciado en otro servidor y comunicado a través de un sistema de alta disponibilidad de datos que lo mantiene actualizado con relación a su homónimo.

Figura 12. Componentes Servicio Computación en la Nube.



Fuente: Los autores

Otra de las funciones del frontend es la de filtrado de peticiones que van de la red externa a la interna bloqueando a los usuario malintencionados. Como función administrativa tiene la capacidad de crear instancias virtuales para usuarios finales según los requerimientos.

La herramienta usada en el frontend para la administración de la nube es OpenNebula, se trata de una herramienta de administración para centros de datos virtualizados bajo la premisa de infraestructura como servicio para la nube (IaaS). El Frontend al ser una máquina virtual instanciada en uno de los servidores, es susceptible a este. Si por alguna razón el servidor que lo contiene deja de funcionar, el administrador configurara el Frontend de respaldo para mantener activa la infraestructura. Actualmente este proceso se hace de forma manual, lo que obliga al administrador estar pendiente del sistema 24/7 por si este presenta algún fallo.

Adicional a esto, la infraestructura está muy ligada al funcionamiento del tercer servidor. Ya que este hace de DataStore, si falla, el sistema quedara inactivo en lo que se refiere a nuevas instancias solicitadas.

4.1.4 Hipervisor. La infraestructura anterior a este proyecto era soportada en virtualización con el hipervisor VirtualBox y KVM conjuntamente, los hosts se encargaban de instanciar máquinas virtuales para el soporte de los servicios como son:

- Máquina virtual con la base de datos central de la infraestructura.
- Instancia virtual encargada del monitoreo de los host y otras instancias virtuales.
- Máquina virtual para el balanceo de carga en los hosts.
- Máquinas especiales de servicio.

4.1.5 Opennebula. La herramienta OpenNebula en la versión actual implementada instancia máquinas en dos hipervisores (KVM y VirtualBox), pero VirtualBox no es soportado de manera nativa, para ello es necesario la instalación y configuración de un plugin adicional, lo que representa un incremento en la dificultad de la administración y configuración de la infraestructura cuando se requiera realizar cambios significativos.

Cuando es realizada una petición de servicio por parte de un usuario para sus proyectos, por lo general éste no sabe con exactitud los recursos que podría llegar a utilizar durante el desarrollo de su proyecto. Por lo que de necesitar mayor capacidad de procesamiento o memoria en su instancia virtual se debe suspender el servicio para reasignar los recursos. Por otro lado puede ocurrir lo contrario, que el usuario este subutilizando los recursos cuando pueden existir otros usuarios con mayor demanda.

Estas tareas de administración pueden llevar a ser tediosas cuando se trata de un ambiente masivo de producción, además de la interrupción del servicio.

En el caso de la realización de tareas de mantenimiento de un host todas las máquinas deben ser migradas a otro nodo activo, pero para ello es necesaria la suspensión de la ejecución de la instancia para llevar a cabo el traslado de la imagen virtual.

4.2 COPIAS DE SEGURIDAD

Las copias de seguridad son un aspecto fundamental en los sistemas de la actualidad, estas permiten garantizar la seguridad de la información en situaciones imprevistas que pueda experimentar el servicio, en otras palabras es uno de los pilares que hace a un servicio que maneje información sea confiable. La plataforma propuesta en este trabajo no está excepta a tener un sistema de copias

de seguridad, de hecho, el tipo de servicio que esta ofrece la obliga a contar con uno.

Actualmente en la plataforma, en los procesos de copias de seguridad se realizan los diferentes tipos de esta, es decir backups incrementales, diferenciales y totales durante la semana, mediante las herramientas rsync y corosync instaladas en dos de los servidores físicos y el tercer servidor se usa para almacenar dichas copias de seguridad.

Las herramientas mencionadas anteriormente deben ejecutarse de forma manual para que así se pueda realizar el procedimiento de respaldo de la información, lo cual impide que se realice de forma óptima, pues al depender de los administradores de los servidores obligan a que estos estén presentes para así de esa forma poder realizar las copias de seguridad, y aun cuando los responsables de la plataforma se encuentren presentes no garantiza que se realice el procedimiento de salvaguardado ya que existe la posibilidad de que olviden ejecutarlo.

5. PROTOTIPO

Para la implementación del prototipo se seleccionó la plataforma CloudComputing a usar, se realizaron estudios para determinar si era conveniente conservar la infraestructura actual o evaluar una nueva.

Tras una revisión del estado actual de la infraestructura CloudEisi se optó por crear un nuevo diseño, una de las razones por las que se tomó esta decisión es porque actualmente la plataforma CloudEisi presenta muchas falencias, una de ellas es que el nodo de administración OpenNebula no se encuentra balanceado, ocasionando que si el servidor que lo contiene deja de funcionar, la plataforma CloudEisi también. Como no se dispone del material administrativo que detalla la configuración original de toda la infraestructura CloudEisi se decidió evaluar y proponer una nueva que fue avalada por el director de proyecto.

Para la selección de la plataforma CloudComputing y demás herramientas que permitieron la obtención de resultados, se fijaron ciertos parámetros:

- Las licencias debían ser gratuitas, sin limitaciones de uso por cantidad de usuario.
- La infraestructura debía correr principalmente en el hipervisor KVM

Por tratarse de un proyecto académico se buscó no incurrir en gastos adicionales, además las herramientas escogidas poseen suficiente documentación y desarrollo para asegurarse de que no solamente se pudiera cumplir satisfactoriamente con el objetivo de la migración, otro aspecto que se tuvo en cuenta fue evitar problemas por defectos en el producto al no tener una larga experiencia en sus desarrollos; otra característica por la cual se escogieron estas herramientas fue la posibilidad de realizar mejoras posteriores al sistema gracias a la implementación de nuevas

características en las herramientas, debido al avance en los desarrollos de los mismos.

5.1 SELECCIÓN DE LA PLATAFORMA

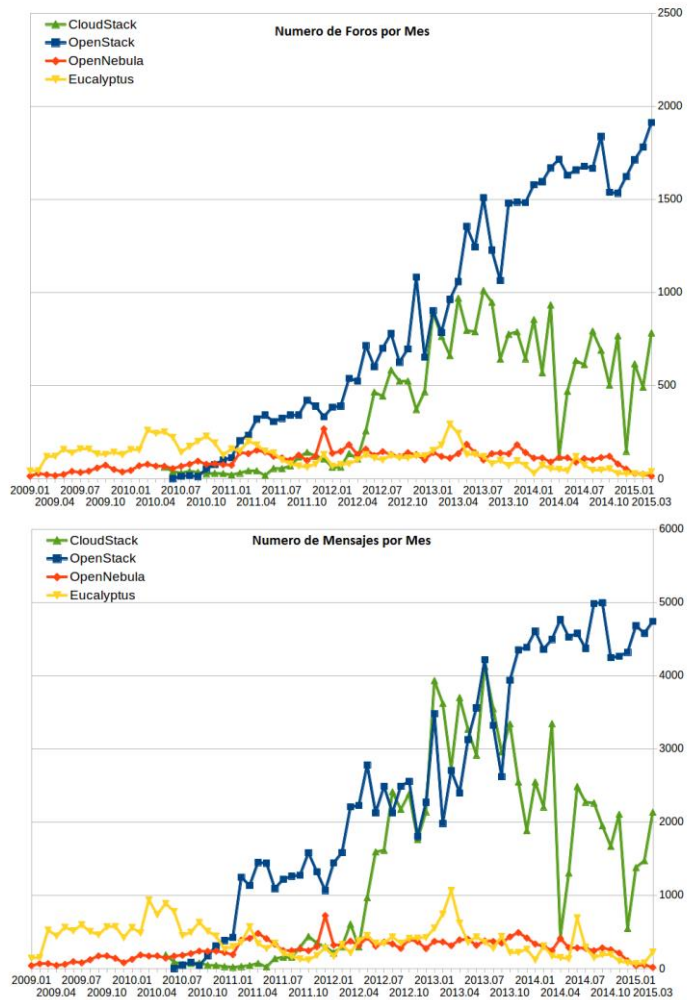
Actualmente las plataformas CloudComputing de código abierto más relevantes son: OpenStack, OpenNebula, Eucalyptus y CloudStack. Seleccionar cualquiera de las mencionadas anteriormente, es una buena decisión debido a que cuentan con una gran acogida en el mercado y documentación suficiente para crear una nube privada.

El criterio que llevó a seleccionar la plataforma que se utilizó, fueron los siguientes:

- Comunidad activa en los foros.
- Contribuyentes que impulsaran el continuo mejoramiento de la plataforma.
- Amplia documentación administrativa referente a sus componentes.

El bloguero chino Qingye Jiang ha elaborado un estudio donde se analizaron y compararon los datos de foros y listas de correo para la actividad de la comunidad OpenStack, OpenNebula, eucalipto y el proyecto CloudStack. Los datos se extrajeron de los foros de debate público y comunicaciones entre los miembros de la comunidad en forma de listas de correo.

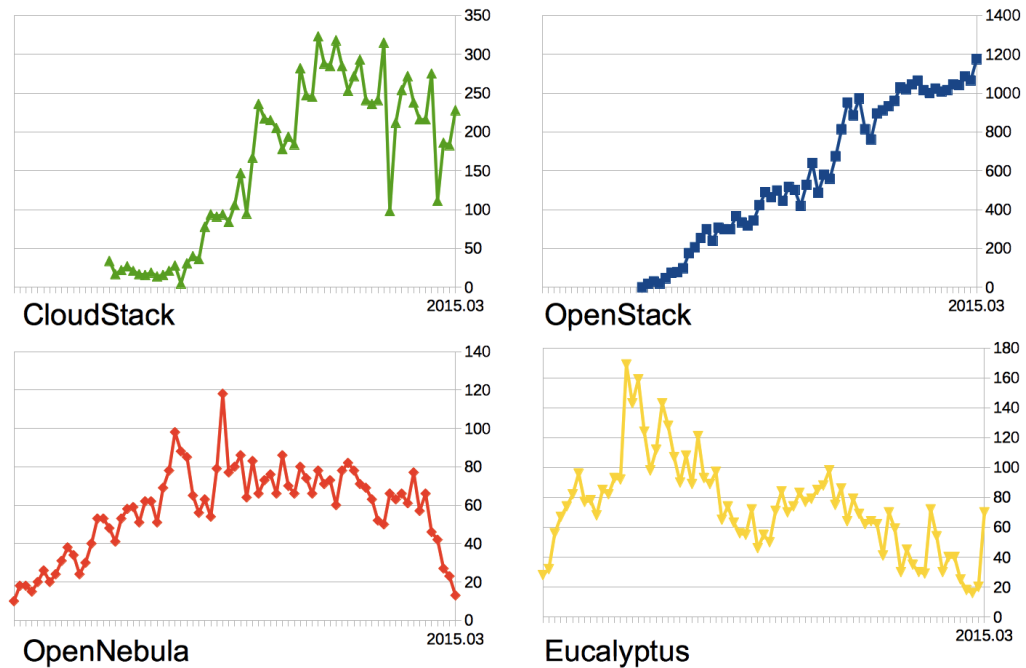
Figura 13. Actividad de Usuarios de cada Comunidad.



Fuente: <http://www.qyjohn.net/>

En las gráficas se aprecia el número mensual de temas y mensajes. Se puede observar que durante los últimos 12 meses, las discusiones relacionadas con OpenStack continuaron mostrando un crecimiento fuerte. Discusiones relacionadas con CloudStack estaban disminuyendo a un ritmo rápido. El volumen de las discusiones en torno a OpenNebula y Eucalyptus eran todavía muy pequeño, ambos mostraron tendencias a disminuir.

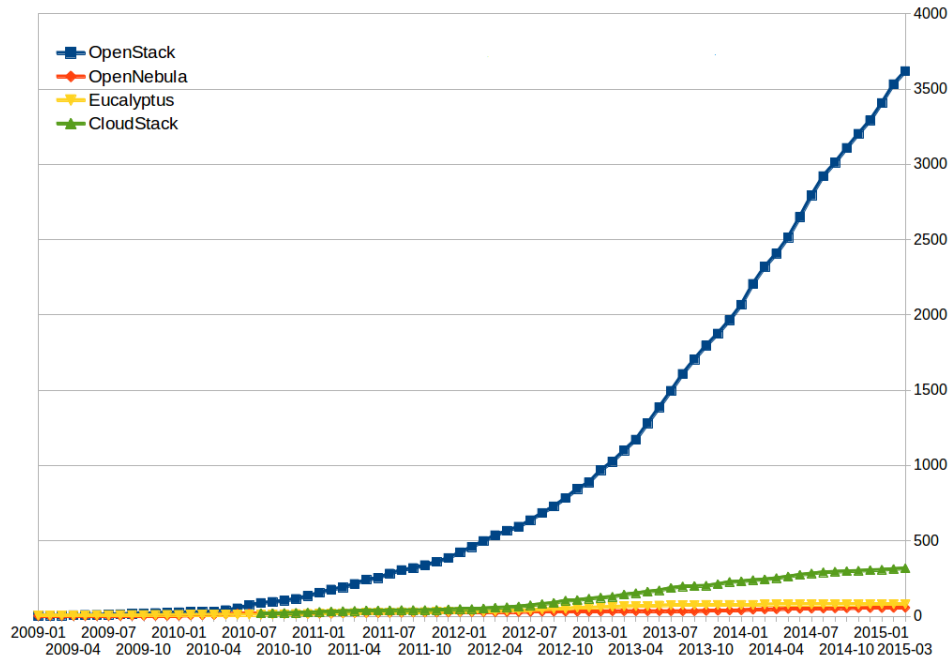
Figura 14. Participación activa de la Comunidad en los Foros.



Fuente: <http://www.qjjohn.net/>

En la figura se muestra los participantes activos de los cuatro proyectos que se discuten. Se puede observar que el número de participantes activos de OpenStack es mucho mayor que los otros tres proyectos. El número de participantes activos de CloudStack también es significativamente más alta que OpenNebula y Eucalyptus. Al observar las cifras de quiebre, el número de participantes activos para OpenStack estaba creciendo de manera constante, mientras que el número de participantes activos para CloudStack, Eucalipto y OpenNebula exhibió disminución significativa.

Figura 15. Población que Contribuye en los Proyectos.



Fuente: <http://www.qjjohn.net/>

En la gráfica se puede apreciar el número total de los desarrolladores que han contribuido código a un proyecto en particular. Se muestra el crecimiento de las poblaciones de desarrolladores acumulados de estos 4 proyectos. Actualmente OpenStack tiene la mayor población desarrolladora acumulada, que es cerca de 10 veces más grande que el distante número 2 CloudStack.

Basado en la información antes presentada, se decidió usar la plataforma OpenStack, ya que esta además de presentar una gran comunidad activa y un número notable de contribuyentes que día a día van mejorando la plataforma, también cuenta con una estructura modular lo que nos permite instalar una plataforma básica y que continuamente puede ser expandida. Haciendo de OpenStack la plataforma adecuada para estudios académicos permitiendo la escalabilidad de un prototipo inicial.

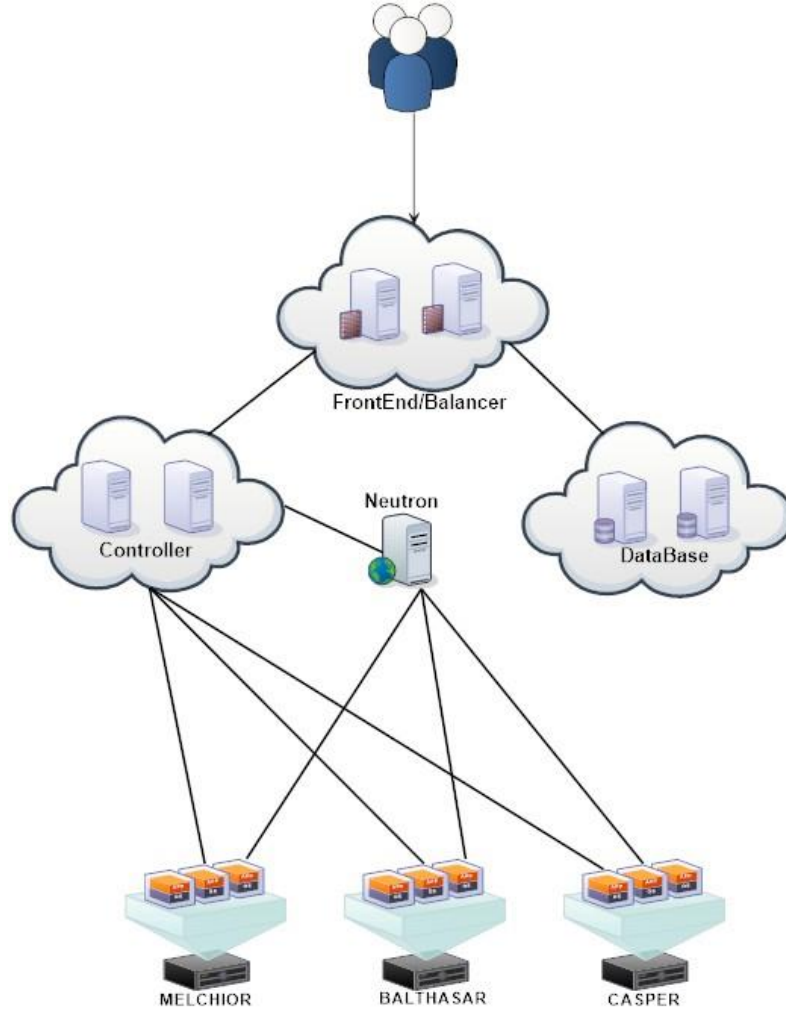
5.2 OPENSTACK

OpenStack es una plataforma con una arquitectura modular con varios nombres para sus componentes: Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove, Sahara. Abarcar todos los módulos con los que cuenta OpenStack actualmente es un trabajo sumamente extenso; gracias a su arquitectura es posible elegir los módulos claves a implementar para conseguir una plataforma funcional básica, tomando esto en cuenta se decidieron implementar en el prototipo: Keystone, Glance, Nova, Neutron y Horizon. De la versión Kilo de OpenStack.

5.3 DISEÑO DE LA INFRAESTRUCTURA

El diseño de la infraestructura tiene como prioridad la alta disponibilidad de la plataforma CloudEisi. Tomando en cuenta el alcance del proyecto, se presentó un modelo de alta disponibilidad aprobado por el director de proyecto, este modelo es una fase inicial de una compleja infraestructura que poco a poco ira mejorando.

Figura 16. Modelo Plataforma CloudEisi.



Fuente: los autores

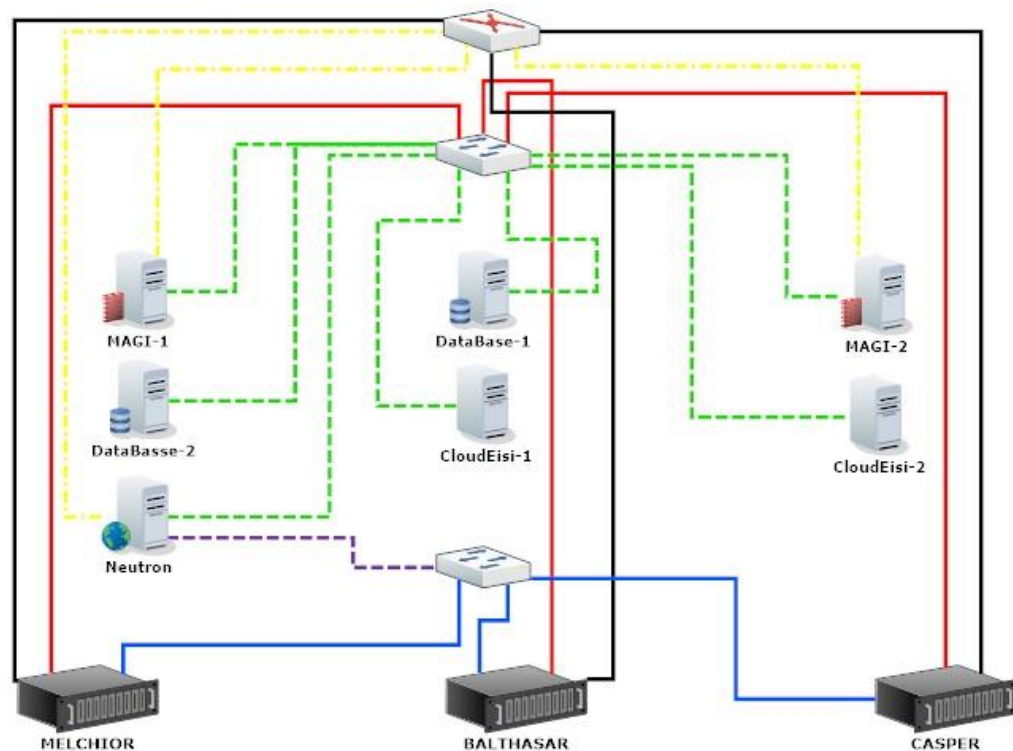
En la figura se puede observar que el nodo controller (administración OpenStack), el FrontEnd y la base de datos se encuentran en alta disponibilidad.

El nodo Neutron encargado de brindar el acceso a la red de las instancias administradas por OpenStack no está en alta disponibilidad, esto se debe a una herramienta de administración (OpenVSwitch) que administra las redes virtuales de la plataforma, dicha herramienta necesita replicar las redes virtuales de un nodo al otro, las pruebas realizadas no consiguieron este proceso así que se optó por crear una copia de respaldo de la máquina virtual Neutron por medio de la

herramienta Bacula. Por cuestiones del alcance del proyecto no se siguieron probando métodos de balanceo y alta disponibilidad para este nodo, pero se dejara documentado en los manuales administrativos las recomendaciones y métodos que logren la alta disponibilidad de dicho nodo.

Como se aprecia en la figura, se contara con dos máquinas virtuales para cada nodo con una configuración Activo/Activo de alta disponibilidad a excepción del Frontend que contara con una configuración Activo/Pasivo.

Figura 17. Disposición Plataforma CloudEisi.



Fuente: los autores

En la figura se aprecia la disposición de los nodos en cada uno de los servidores físicos, esta organización está pensada por si falla un servidor la plataforma no presente interrupción en el servicio, incluso si el nodo Neutron deja de funcionar la

plataforma estará en servicio pero la red de las maquinas no funcionara, aquí entraría a jugar Bacula para restaurar la máquina virtual en otro servidor.

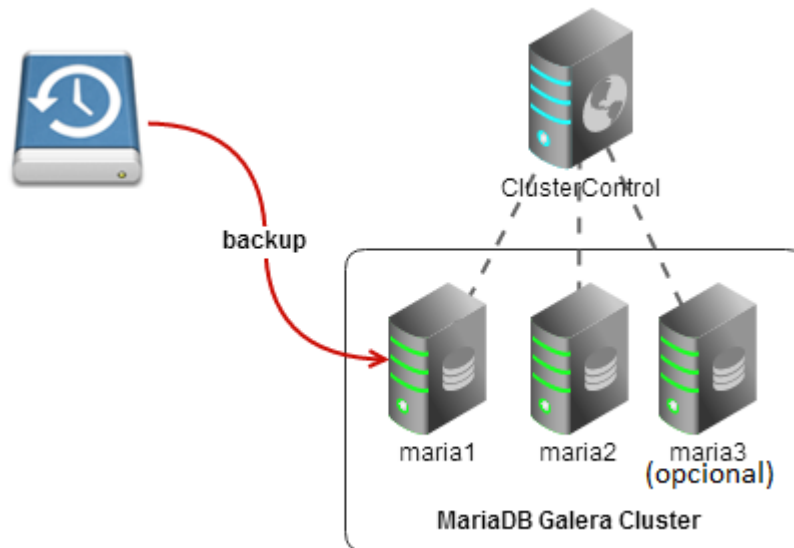
OpenStack hace uso de tres redes para su funcionamiento, dos internas y una externa que se puede apreciar en la Figura 17. una de las redes internas es de uso exclusivo de OpenStack para la administración de las redes virtuales de las instancias y la otra red es para la comunicación de todos los módulos y nodos de la infraestructura, la red externa la tendrán disponible los nodos que necesitan de ella como son, el Frontend y el nodo Neutron. Los servidores físicos que hacen de nodos Compute de la plataforma OpenStack no necesitan red externa, pero ya que la infraestructura está diseñada en máquinas virtuales contenidas en estos Servidores se hace indispensable que cuenten con esta conexión.

5.4 GESTOR DE BASES DE DATOS

Se estudió la manera de crear una base de datos distribuida capaz de soportar los datos de administración de OpenStack, recientemente estudiantes del grupo CONUSS de la Universidad Industrial de Santander [9] investigaron sobre este tema, basaron su investigación sobre MySQL y sus resultados concluyeron que esta herramienta no soportaría la administración de OpenStack. Gracias a esta investigación se buscó una herramienta de administración de base de datos OpenSource que permitiera lograr este fin. Se optó por implementar MariaDB. (MySQL después de ser comprado por Oracle) [14] freno su investigación en el campo de base de datos distribuidas, a raíz de esto Michael (Monty) Widenius (fundador de MySQL) desarrollo MariaDB.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL [15], esta cuenta con una herramienta llamada Galera Clúster que permitió crear la base de datos distribuida.

Figura 18. Estructura Clúster HA MariaDB.



Fuente: <http://severalnines.com/>

En la figura se aprecia el modelo implementado en la plataforma, siendo el ClusterControl el nodo Frontend, la configuración permite N nodos de MariaDB, por motivos de reducir el uso de recursos, se implementaron en el prototipo dos nodos MariaDB con backups hechos por Bacula.

5.5 SISTEMA OPERATIVO

Los servidores, contaban con el S.O Debian 6, ya que los servidores necesitaban actualizar su software y limpiar los discos de información innecesaria, se optó por formatear y cambiar el sistema operativo, por tanto, la labor administrativa se inició desde la instalación misma del S.O. El sistema operativo elegido para el funcionamiento de los servidores fue Ubuntu Server 14.04 de 64 bits, que es un S.O basado en GNU/Linux.

Se optó por la instalación de Ubuntu Server 14.04, no sólo por contar con repositorios oficiales de OpenStack (Debian no cuenta con repositorios oficiales

OpenStack, solo de terceros), sino por ser un S.O estable, por su versatilidad en el mantenimiento e instalación de paquetes software, tiene un bajo consumo de recursos (Memoria Ram, Disco duro, Procesador) de su sistema base, además de tener gran soporte y documentación, entre otros factores que ayudaron decidir su instalación.

Cuenta además con licencia GPL (General Public Licence), es de libre distribución y es desarrollado por más de mil voluntarios alrededor del mundo, que colaboran a través de Internet, colocando su documentación, mejoras, características y cambios realizados en las diferentes versiones en el sitio web oficial de Ubuntu (canonical).

5.6 INSTALACION Y CONFIGURACION

La infraestructura CloudEisi seleccionada cuenta con nodos de administración, estos nodos cumplen uno o varios roles específicos que se irán mencionando, se mencionaran los paquetes principales instalados en cada nodo y su configuración quedara documentada en los manuales administrativos.

Cada nodo cuenta con el S.O Ubuntu Server 14.04 y su instalación y configuración se basó en los documentos administrativos referentes a la instalación de Debian 6 para mantener la seguridad ya implementada [8].

5.6.1 Nodo frontend. Encargado de administrar toda la plataforma CloudEisi y hacer de balanceador de cargas de la misma. Cuenta con los siguientes paquetes esenciales:

Keepalived, que le permite administrar su IP virtual que lo mantiene en alta disponibilidad en la red.

```
#apt-get install keepalived -y
```

HAProxy, que será el encargado de balancear toda la arquitectura CloudEisi.

```
#apt-get install haproxy -y
```

RabbitMQ, que será el encargado de todos los mensajes que circulan entre los diferentes módulos de OpenStack.

```
#apt-get install ntp rabbitmq-server
```

MySQL-Client, que le permitirá cumplir el rol de ControlCluster de nuestro gestor de base de datos.

```
#apt-get install mysql-client
```

5.6.1 Nodo controller. Encargado de administrar OpenStack. Cuenta con los siguientes paquetes esenciales:

Keystone, modulo encargado de ofrecer un directorio central de usuarios asignados a los servicios de OpenStack que pueden acceder.

```
# apt-get install keystone python-keystoneclient
```

Glance, que será el encargado de las imágenes almacenadas que se pueden utilizar como una plantilla. También se puede utilizar para almacenar y catalogar un número ilimitado de copias de seguridad.

```
# apt-get install glance python-glanceclient
```

Nova, que será el encargado de administrar los nodos compute de OpenStack para la instanciación de VM.

```
# apt-get install nova-api nova-cert nova-conductor \  
nova-consoleauth nova-novncproxy nova-scheduler python-novaclient
```

Neutron-server, modulo encargado de administrar el nodo de red Neutron.

```
# apt-get install neutron-server neutron-plugin-m12 \  
python-neutronclient
```

Horizon, proporciona a los administradores y usuarios una interfaz gráfica para el acceso, la provisión y automatización de los recursos basados en la nube.

```
# apt-get install openstack-dashboard apache2 \  
libapache2-mod-wsgi memcached python-memcache
```

5.6.1 Nodo Neutron. Este nodo es el encargado de proporciona modelos de redes para diferentes aplicaciones o grupos de usuarios.

Neutron, Modulo encargado del modelo de redes.

```
# apt-get install neutron-plugin-m12 neutron-l3-agent \  
neutron-dhcp-agent
```

OpenVSwitch, herramienta encargada de la comunicación de las redes de las VM's.

```
# apt-get install neutron-plugin-openvswitch-agent
```

5.6.1 Nodo databases. Este nodo es el encargado de la gestión de las bases de datos de toda la infraestructura CloudEisi.

MariaDB Galera Cluster, Herramienta administrativa de bases de datos distribuidas.

```
# apt-get install mariadb-galera-server galera
```

5.6.1 Nodo compute (Servidor Físico). Este nodo es el encargado de la gestión de las instancias virtuales CloudEisi.

KVM, hipervisor predeterminado para las instancias de la plataforma CloudEisi.

```
$ sudo apt-get install qemu-kvm libvirt-bin
```

Nova-Compute, modulo administrativo de las instancias en OpenStack.

```
# apt-get install nova-compute sysfsutils
```

OpenVSwitch, herramienta encargada de la comunicación de las redes de las VM's.

```
# apt-get install neutron-plugin-ml2 \  
neutron-plugin-openvswitch-agent
```

5.7 COPIAS DE SEGURIDAD

Para propósitos del modelo planteado se optó por implementar una herramienta software que realice de forma automática y periódica los procedimientos de copias de seguridad.

Existen numerosos software que realizan copias de seguridad, los hay tanto libres como pagos, ya sea para realizar copias personales o de varios equipos. Para objetos de respaldar la información del modelo planteado se decidió por optar por una herramienta de copias de seguridad en red conocido como Bacula.

5.7.1 Bacula. Bacula es una solución basada en Open Source que gestiona la copia de seguridad centralizada en un servidor, de las diferentes máquinas de una red independientemente del sistema operativo.

Bacula actúa como un director de copias, guardando un catálogo (Base de datos) de todos los ficheros (directorios) relevantes que deben ser copiados y realizando las copias de forma completa o bien sólo de los cambios ocurridos desde una fecha concreta. Para depositar las copias, este producto puede utilizar diversos dispositivos los cuales pueden ser añadidos sin límite y pueden ser desde unidades de discos, volúmenes NAS/SAN, unidades de cinta, etc.

Permite salvaguardar los datos clientes, transmitiéndolos a través de la red local o incluso Internet. La información circula y se almacena de forma totalmente segura gracias a la integración de algoritmos de cifrado y certificados digitales.

DIRECTOR (DIR) Programa servidor que supervisa todas las funciones necesarias para las operaciones de “backup” y “restore”. Es el eje central de la solución y en él se declaran todos los parámetros necesarios. Se ejecuta como un “demonio” en el servidor.

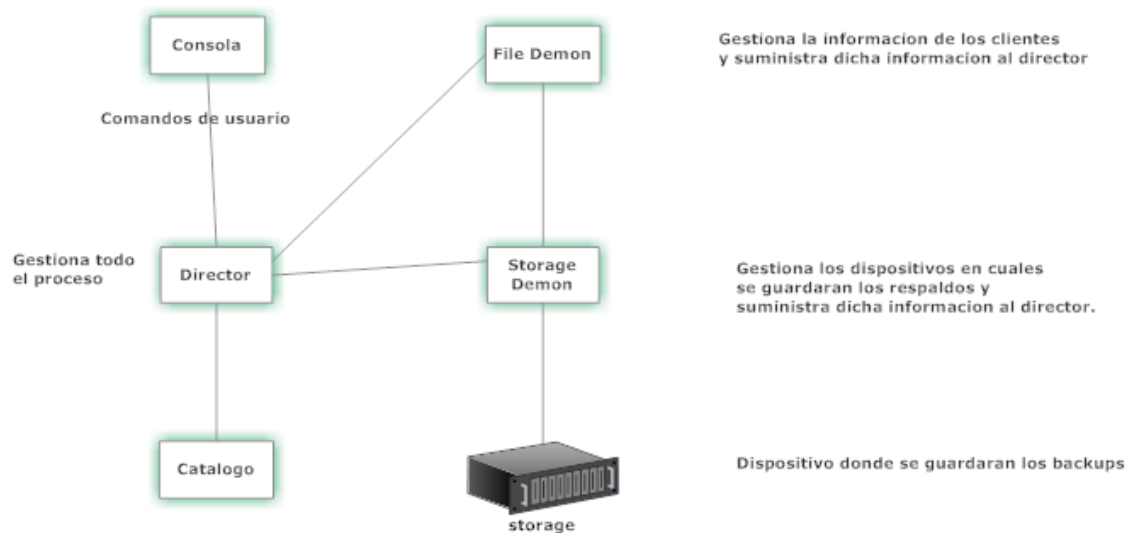
STORAGE (SD) Programa que gestiona las unidades (discos, bandas magnéticas, etc.) donde se almacenarán los datos. Es el responsable de escribir y leer en los medios que utilizaremos para las copias de seguridad. Se ejecuta como un “demonio” en la máquina propietaria de los medios utilizados es decir donde se guardaran la información respaldada. En muchos casos será en el propio servidor.

FILE (FD) Este servicio, conocido como “cliente” está instalado en cada máquina a salvaguardar y es específico al sistema operativo donde se ejecuta. Responsable para enviar al “Director” los datos cuando este lo requiera.

CATALOG Base de datos que almacena la información necesaria para localizar donde se encuentran los datos salvaguardados de cada archivo, de cada cliente, etc.

CONSOLE Programa que permite al administrador la interacción con el “Director” para todas las funciones del servidor. La versión original es una aplicación en modo texto.

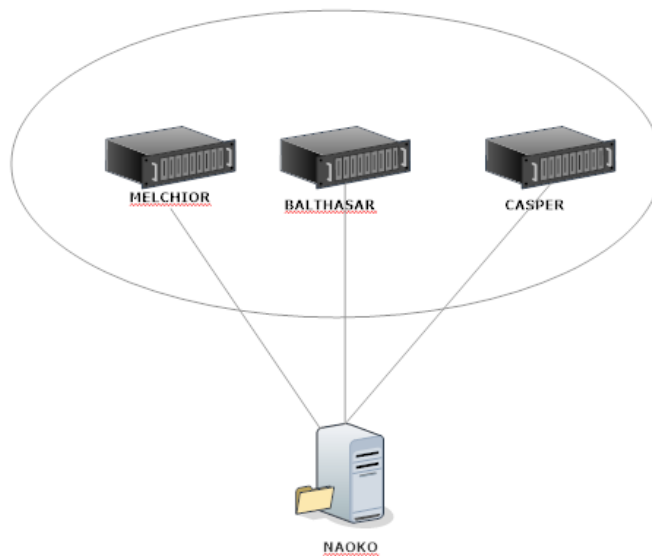
Figura 19. Interacción de los elementos del Bacula.



Fuente: los autores

Esta herramienta se instaló en un equipo ajeno al modelo planteado y realizará el respaldo a los servidores físicos, que hará las veces de servidor es decir en el encargado de gestionar todo el proceso de copias de seguridad y restauración de archivos, con el propósito de dar un respaldo confiable al modelo en caso de que este sufra percances en los que puedan verse afectada información delicada.

Figura 20. Modelo de copias de seguridad



Fuente: los autores

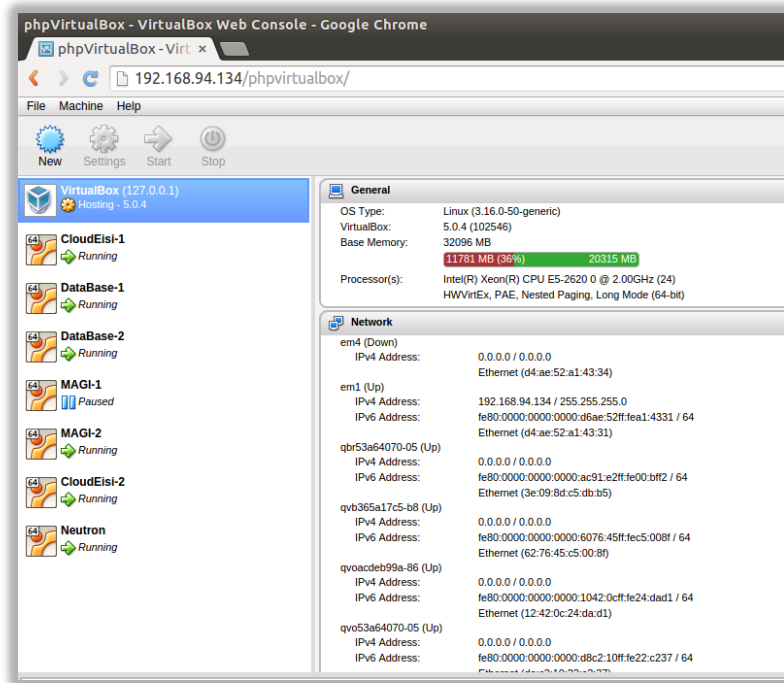
Los clientes no necesariamente deben ser servidores físicos, máquinas virtuales con conexión a una red también pueden ser clientes del Bacula pues solo requiere de un dominio o una IP para ser reconocido por el director y definir directorios de los archivos a salvaguardar.

5.8 PRUEBAS Y VERIFICACION

Todas las maquinas se encuentran encendidas y configuradas para el correcto funcionamiento de la plataforma CloudEisi. Se procederá a probar la alta

disponibilidad del sistema y posterior a esto verificar el correcto funcionamiento de la plataforma.

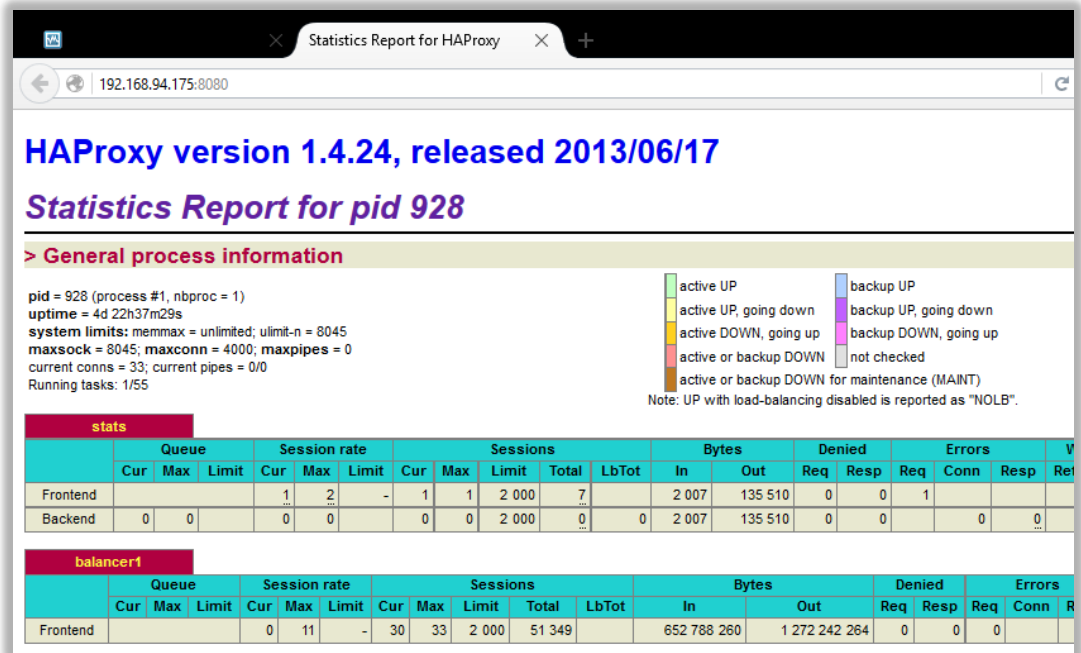
Figura 21. Lista de VM's de administración.



Fuente: los autores

5.8.1 Pruebas keepalived y haproxy. Se ingresará a la dirección IP Virtual antes configurada para el prototipo y se comprobará el funcionamiento de Keepalived y HAProxy.

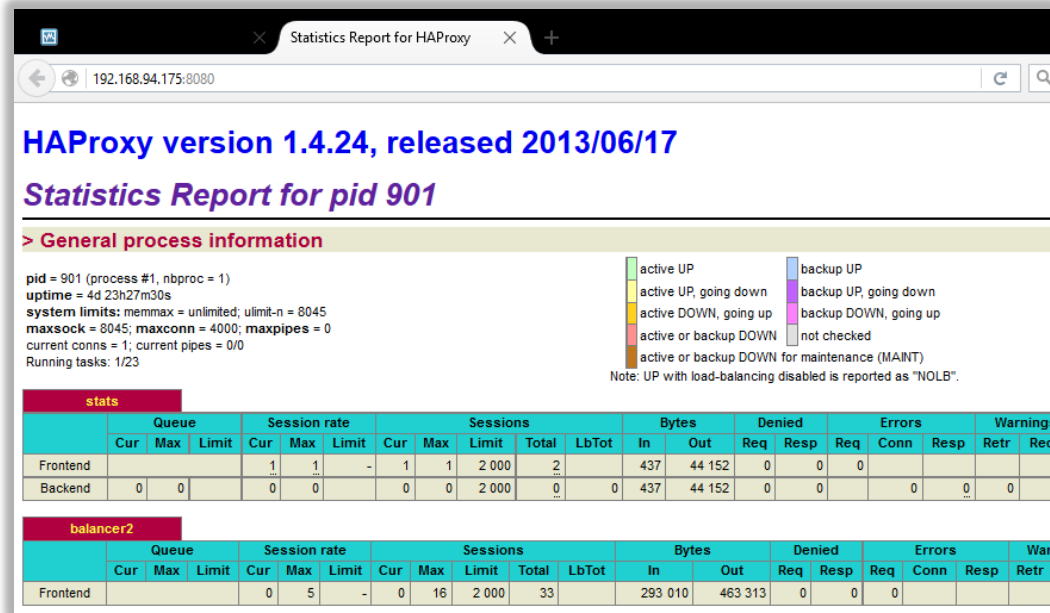
Figura 22. Nodo activo en el Frontend.



Fuente: los autores

En la figura se puede observar que Keepalived está asignando la IP Virtual al nodo balanceador 1 (MAGI-1) que es el Frontend; esta herramienta de HAProxy permite visualizar además el flujo de datos en bytes que presenta el nodo.

Figura 23. Caída del nodo 1 del Frontend.

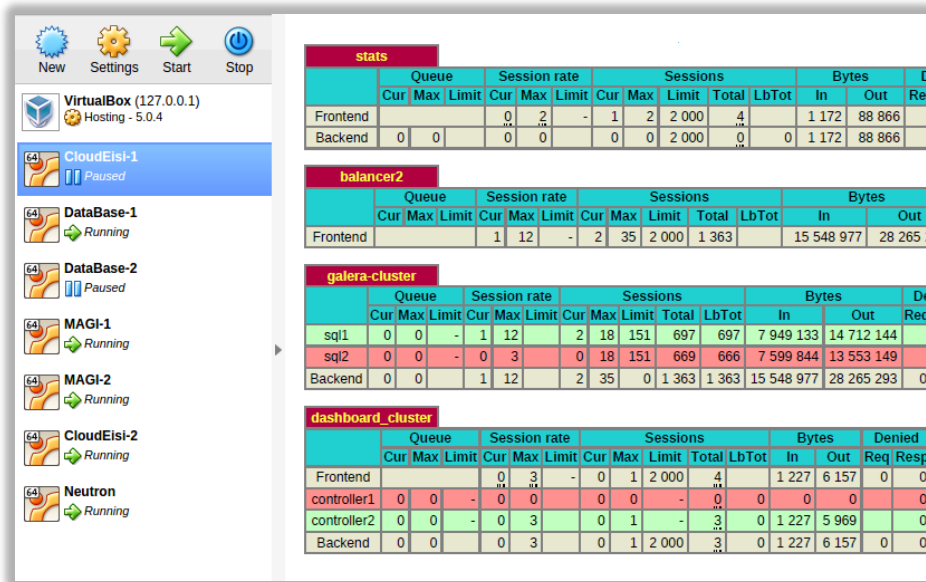


Fuente: los autores

En la figura se puede observar que al apagar en nodo MAGI-1 automáticamente Keepalived asigna la IP Virtual al nodo balanceador 2 (MAGI-2) y será el nuevo Frontend incluso si el nodo MAGI-1 vuelve a la red, hasta que MAGI-2 no pierda la conexión Keepalived no le sedera la IP Virtual a MAGI-1.

5.8.2 Pruebas gestor base de datos y plataforma openstack. Se probará simultáneamente la caída de uno de los nodos de Base de datos y Controller, ya que la plataforma OpenStack no funcionara correctamente si la Base de Datos no tiene conexión.

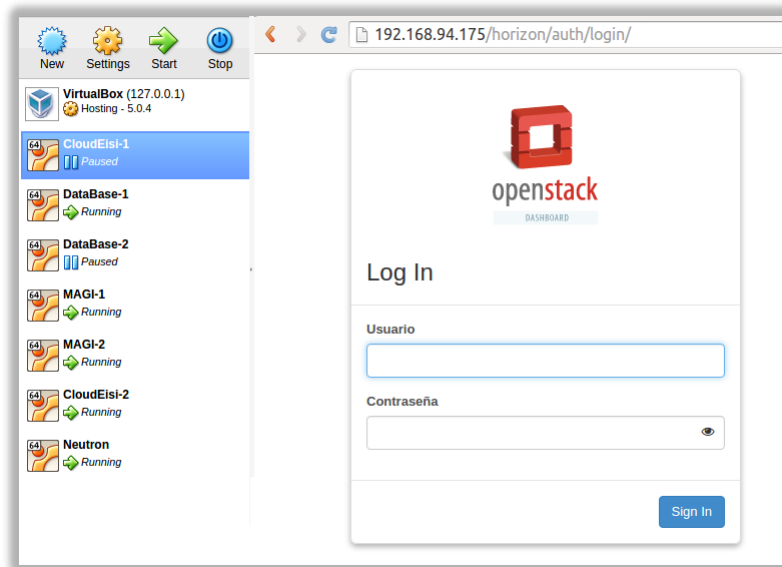
Figura 24. Caída de uno de los nodos de Database y Controller.



Fuente: los autores

HAProxy administra los módulos OpenStack por separado, es decir que uno de los nodos puede tener conexión, pero algunos de sus servicios pueden estar caídos, para agilidad en las pruebas se decidió apagar la maquina completa y colgar todos los servicios de un nodo.

Figura 25. Prueba de funcionamiento con nodos caídos.

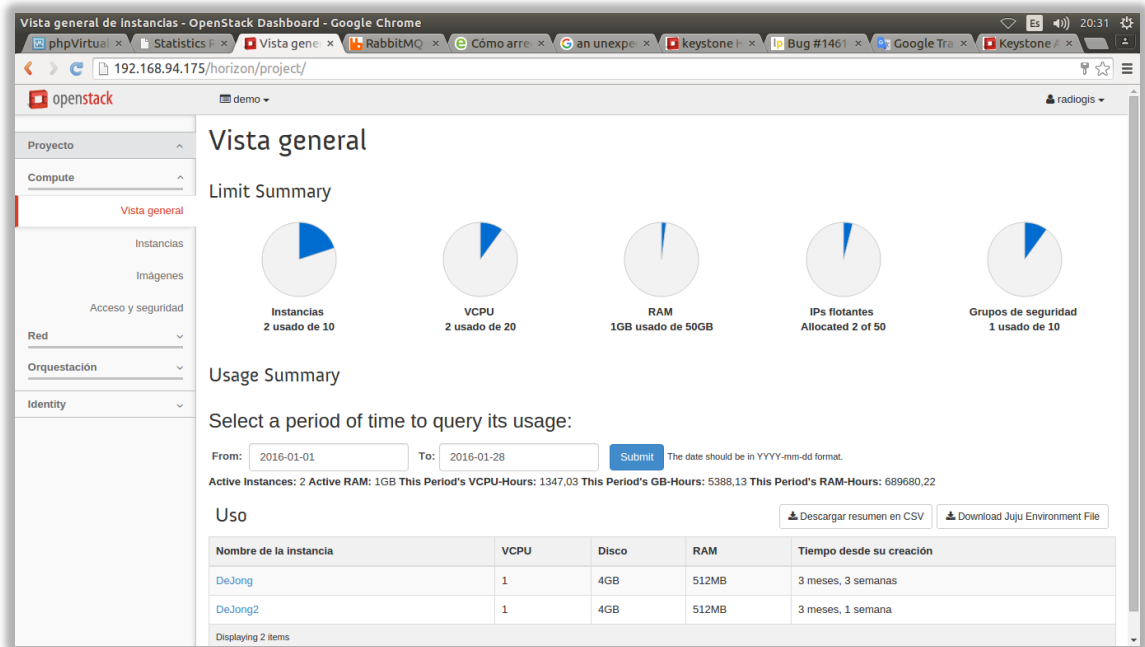


Fuente: los autores

Se aprecia en la figura que a pesar de no tener conexión con uno de los nodos, la plataforma OpenStack sigue en funcionamiento.

5.8.3 Verificación funcionamiento plataforma Cloudeisi. El prototipo implementado en este proyecto realizó una prueba de producción satisfactoriamente. Se asignó una VM administrada por un estudiante de Ingeniería de Sistemas, esta máquina cumplía la función de web service y recibía las peticiones de la página web del grupo RadioGIS de la Escuela de Eléctrica y Electrónica.

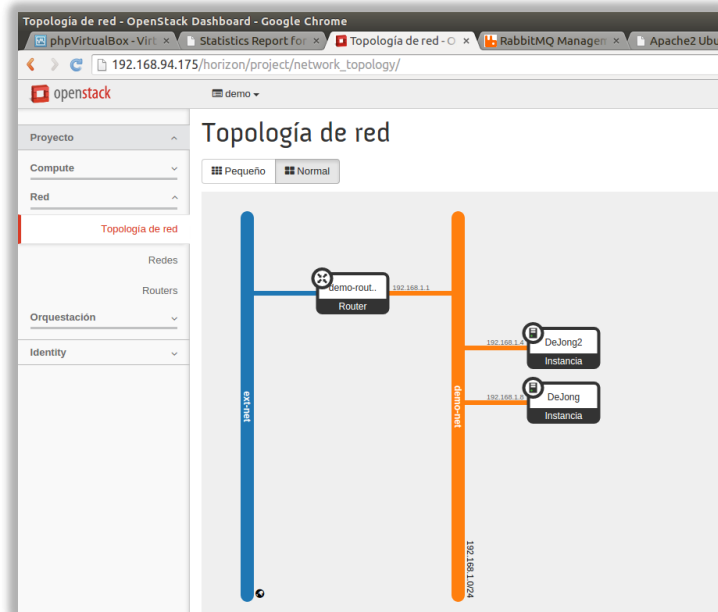
Figura 26. Página principal de OpenStack.



Fuente: los autores

En la figura se aprecia la interfaz de administración de las VM's de los diferentes usuarios, pueden administrar desde la instanciación hasta el modelo de redes que deseen para sus máquinas.

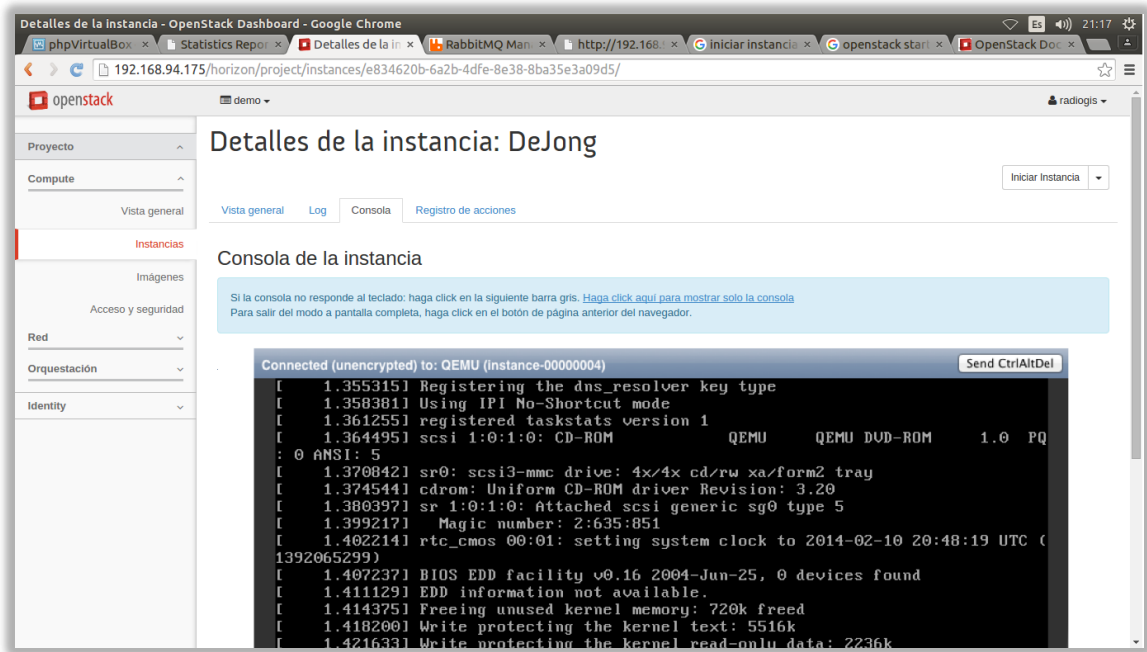
Figura 27. Modelo de red virtual administrado por Neutron.



Fuente: los autores

En esta sección se puede observar el modelo de red con el que cuenta la VM instanciada, se puede escoger si esta máquina necesita acceso a la red externa o no, asignándole una IP flotante previamente creada.

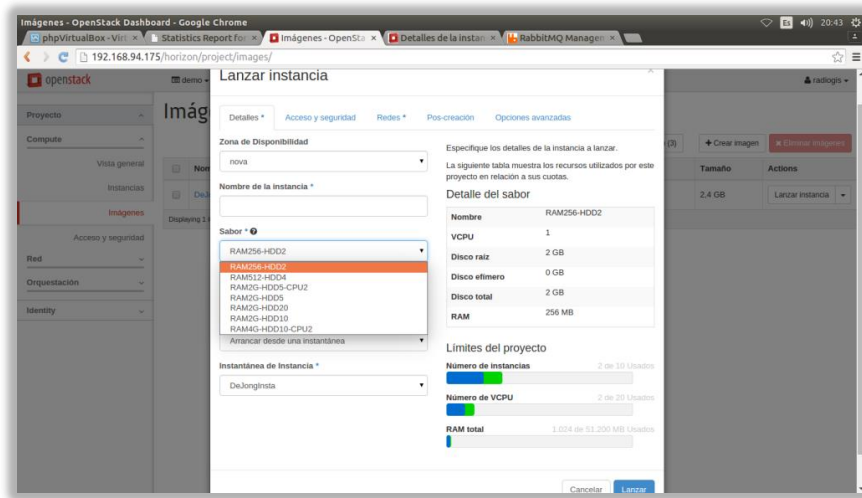
Figura 28. Acceso a la interfaz de una VM.



Fuente: los autores

Desde la plataforma OpenStack se puede acceder a la instancia, si el usuario lo desea puede usar un gestor de escritorio remoto y acceder a la VM por medio de unas llaves que el sistema le entrega para mejorar la seguridad en acceso remoto. Este acceso remoto estará disponible siempre y cuando la maquina cuente con una IP flotante.

Figura 29. Instanciación de VM en OpenStack.



Fuente: los autores

El usuario puede escoger los recursos que su instancia desea utilizar, cabe aclarar, que el usuario no dispondrá de todos los recursos de los nodos Compute, el administrador de la plataforma al momento de asignarle el usuario y rol delimita la cantidad de recursos que este puede utilizar, como también la cantidad de instancias que puede administrar.

Figura 30. Lista de instancias disponibles.

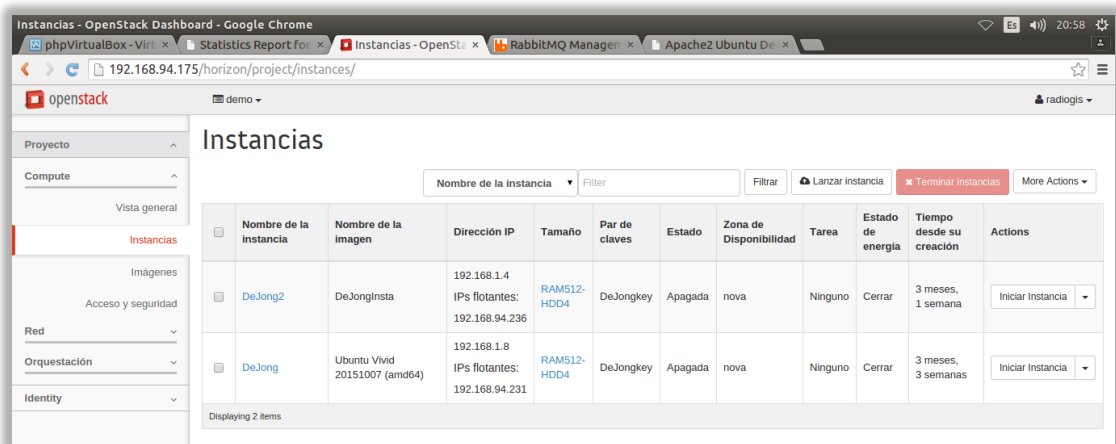
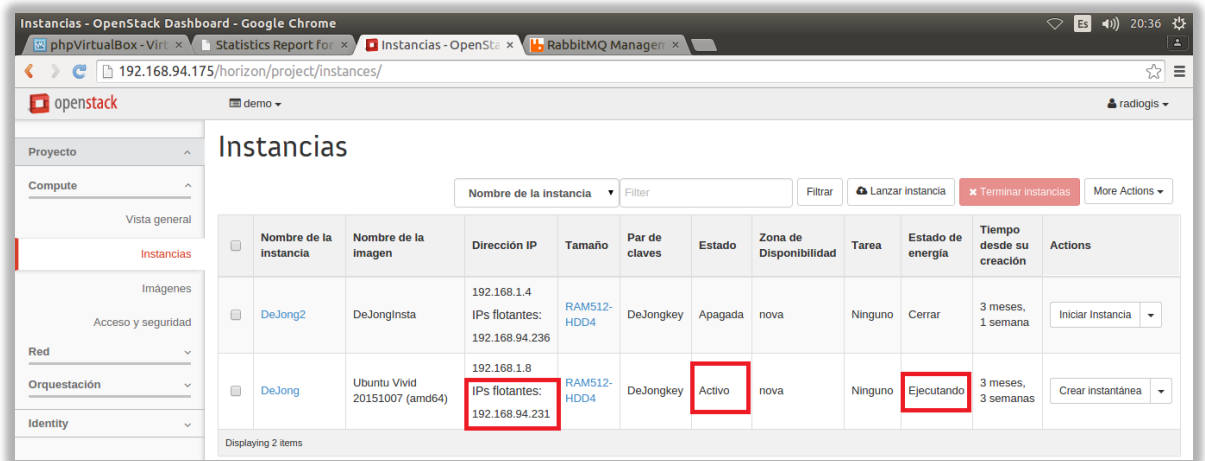
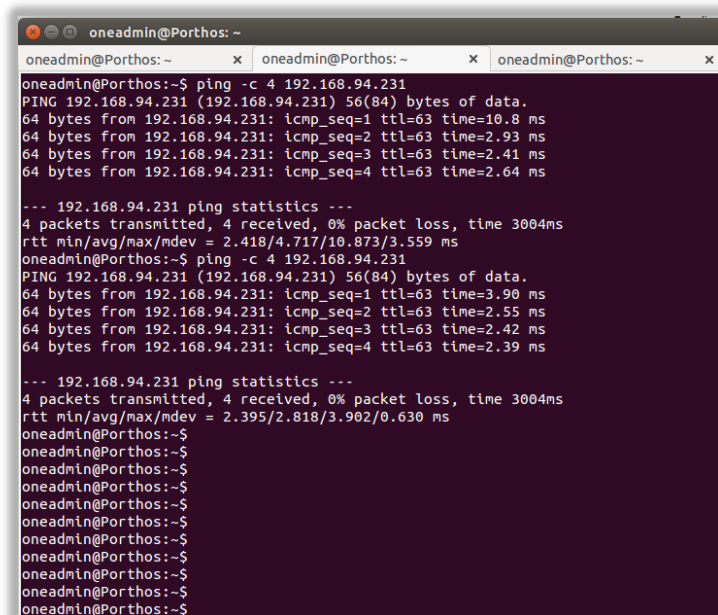


Figura 32. Encendido maquina DeJong.



Fuente: los autores

Figura 33. Ping Instancia Encendida.



Fuente: los autores

En las figuras 32 y 33 se aprecia que la instancia se encuentra encendida y que al hacer ping (DeJong: 192.168.94.231) se encuentra la conexión, comprobando que la maquina tiene reservado y en uso esta dirección de red.

5.8.4 Verificación funcionamiento del Bacula. Para la prueba de la herramienta Bacula se definieron dos clientes a los cuales realizarles backup, la maquina CASPER en la cual el prototipo planteado está en funcionamiento y el otro cliente es la maquina MeiWEB en la que se encuentra alojado el sitio del mismo nombre. Bacula posee un complemento web que permite visualizar datos resumidos sobre las tareas ejecutadas con gráficos. Su instalación se encuentra definida adjunta en el manual del Bacula, para posteriores usos del grupo.

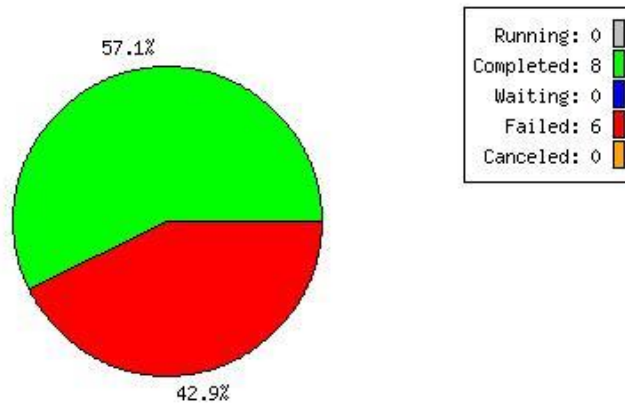
Figura 34. Backups realizados.

Status	Job ID	Name	Type	Start time	End time	Elapsed time	Level	Bytes	Files	Speed	Compression	Pool	Log
✓	233	BackupMeiWEB	B	2016-01-28 12:30:11	2016-01-28 12:30:12	00:00:01	Incr	141.00 B	2	141.00 B/s	0.25	RemoteFileMeiWEB	Q
✓	232	BackupNaoko	B	2016-01-28 12:30:07	2016-01-28 12:30:07	00:00:00	Incr	0.00 B	0	0.00 B/s	n/a	File	Q
✓	231	BackupCASPER	B	2016-01-28 12:30:03	2016-01-28 12:30:04	00:00:01	Incr	197.35 KB	2	197.35 KB/s	0.89	RemoteFileCASPER	Q
✓	230	BackupCatalog	B	2016-01-27 23:10:03	2016-01-27 23:10:03	00:00:00	Full	900.72 KB	1	0.00 B/s	0.00	File	Q
✓	229	BackupMeiWEB	B	2016-01-27 23:05:20	2016-01-27 23:05:21	00:00:01	Incr	0.00 B	0	0.00 B/s	n/a	RemoteFileMeiWEB	Q
✓	228	BackupNaoko	B	2016-01-27 23:05:16	2016-01-27 23:05:17	00:00:01	Incr	1.35 KB	3	1.35 KB/s	0.82	File	Q
✓	227	BackupCASPER	B	2016-01-27 23:05:02	2016-01-27 23:05:13	00:00:11	Incr	43.63 MB	13	3.97 MB/s	0.89	RemoteFileCASPER	Q
✓	226	BackupMeiWEB	B	2016-01-27 12:30:11	2016-01-27 12:30:12	00:00:01	Incr	0.00 B	0	0.00 B/s	n/a	RemoteFileMeiWEB	Q

Fuente: los autores

Gracias a esta herramienta también se pueden ver de forma gráfica es estado de los trabajos de backup, que se han realizado desde la instalación de la herramienta.

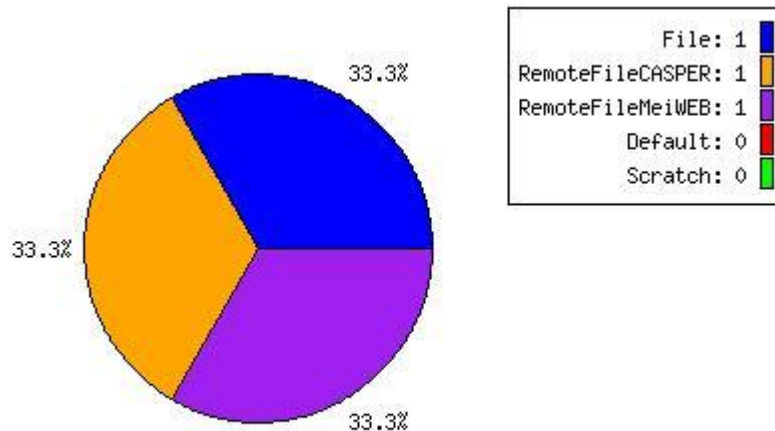
Figura 35. Estadística backups realizados.



Fuente: los autores

Ademas de los “Jobs” realizados tambien es posible hacer un seguimiento a los “POOLS” definidos es decir los destinos en los cuales se almacenas las copias de seguridad, es decir si estos estan siendo usados por algun trabajo de backup definido.

Figura 36. Estadística backups realizados en los diferentes volúmenes de archivos.

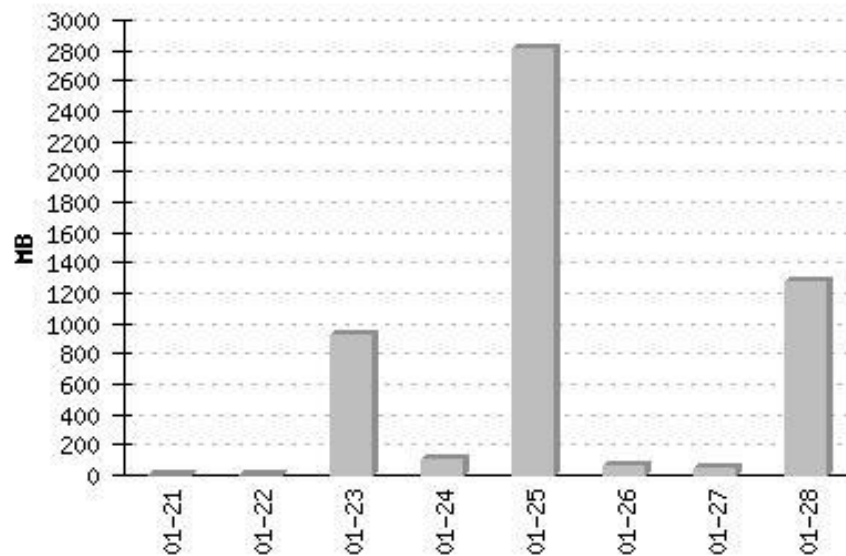


Fuente: los autores

El numero que mostrado en la grafica indica el numero de JOBS que estan usando dicho POOL. En el POOL File se estan guardando las copias de seguridad locales, en RemoteFileCASPER solamente los trabajos definidos para el servidor CASPER y en RemoteFile MeiWEB los realizados a la maquina MeiWEB.

Una vez mostrados los estados de los trabajos de copias de seguridad y de los POOLS respectivamente se verifica si se ha estado salvaguardando archivos.

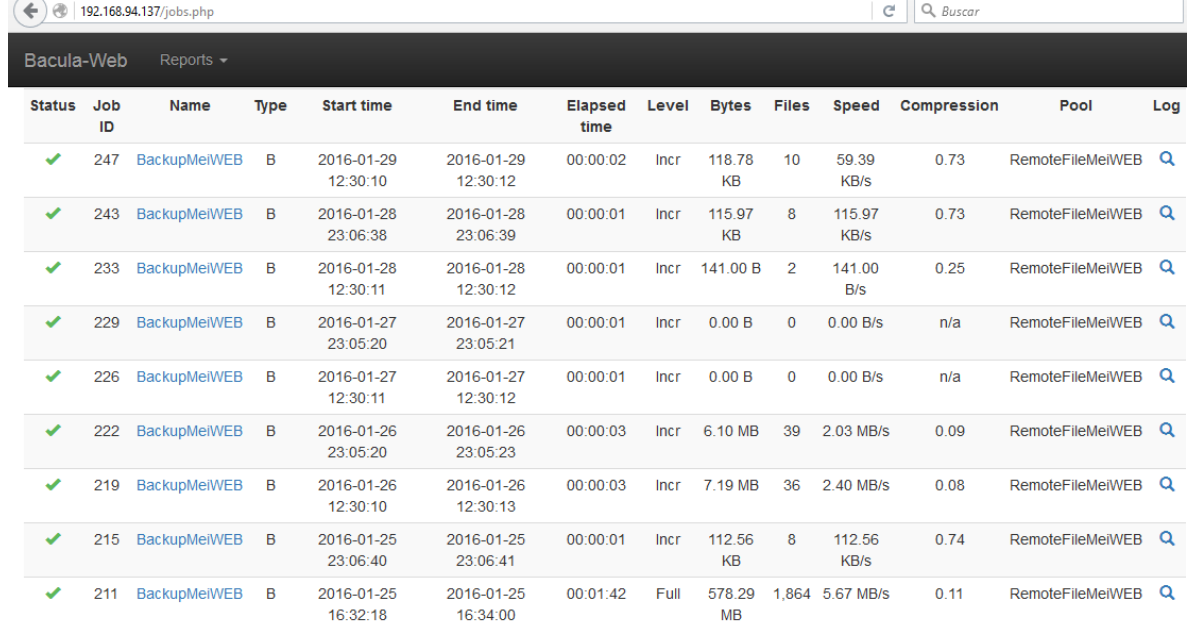
Figura 37. Estadística tamaño de archivos salvaguardados.



Fuente: los autores

Como se puede apreciar en la gráfica se han estado realizando de forma satisfactoria copias de seguridad a la maquina MeiWEB además del tamaño de archivos de dichas copias. Pero la gráfica también es posible mostrar los trabajos de copias de seguridad realizados a un cliente, en este caso MeiWEB de una forma más general como se muestra en la maquina a continuación:

Figura 38. Copias de seguridad realizados al MeiWEB.



Status	Job ID	Name	Type	Start time	End time	Elapsed time	Level	Bytes	Files	Speed	Compression	Pool	Log
✓	247	BackupMeiWEB	B	2016-01-29 12:30:10	2016-01-29 12:30:12	00:00:02	Incr	118.78 KB	10	59.39 KB/s	0.73	RemoteFileMeiWEB	Q
✓	243	BackupMeiWEB	B	2016-01-28 23:06:38	2016-01-28 23:06:39	00:00:01	Incr	115.97 KB	8	115.97 KB/s	0.73	RemoteFileMeiWEB	Q
✓	233	BackupMeiWEB	B	2016-01-28 12:30:11	2016-01-28 12:30:12	00:00:01	Incr	141.00 B	2	141.00 B/s	0.25	RemoteFileMeiWEB	Q
✓	229	BackupMeiWEB	B	2016-01-27 23:05:20	2016-01-27 23:05:21	00:00:01	Incr	0.00 B	0	0.00 B/s	n/a	RemoteFileMeiWEB	Q
✓	226	BackupMeiWEB	B	2016-01-27 12:30:11	2016-01-27 12:30:12	00:00:01	Incr	0.00 B	0	0.00 B/s	n/a	RemoteFileMeiWEB	Q
✓	222	BackupMeiWEB	B	2016-01-26 23:05:20	2016-01-26 23:05:23	00:00:03	Incr	6.10 MB	39	2.03 MB/s	0.09	RemoteFileMeiWEB	Q
✓	219	BackupMeiWEB	B	2016-01-26 12:30:10	2016-01-26 12:30:13	00:00:03	Incr	7.19 MB	36	2.40 MB/s	0.08	RemoteFileMeiWEB	Q
✓	215	BackupMeiWEB	B	2016-01-25 23:06:40	2016-01-25 23:06:41	00:00:01	Incr	112.56 KB	8	112.56 KB/s	0.74	RemoteFileMeiWEB	Q
✓	211	BackupMeiWEB	B	2016-01-25 16:32:18	2016-01-25 16:34:00	00:01:42	Full	578.29 MB	1,864	5.67 MB/s	0.11	RemoteFileMeiWEB	Q

Fuente: los autores

Con lo anterior se puede evidenciar, la implementación de forma satisfactoria de una herramienta efectiva de backups, la cual permite ampliar el número de clientes a los cuales salvaguardarles la información además de igualmente poder aumentar el número de dispositivos o equipos en los cuales poder guardar dichos backups. Los datos solo pueden ser restaurados desde el Bacula lo que garantiza la seguridad de la información pues esta se guarda en un formato encriptado legible solo para Bacula permitiendo que en la restauración, estos archivos vuelvan tal y como estaban al momento de realizar la copia de seguridad.

CONCLUSIONES

El proceso administrativo, no se limita a la instalación de herramientas software y a la realización de tareas de monitoreo de las mismas, es una tarea global, que abarca los diferentes aspectos funcionales desde el hardware hasta la prestación de servicios, incluyendo la automatización de tareas administrativas básicas.

Como resultado de la investigación, se logró implementar de manera exitosa una plataforma en la nube de alta disponibilidad; mediante el uso de herramientas de software libre, generando unos servicios computacionales confiables para su uso en la Escuela de Ingeniería de Sistemas y la comunidad académica en general.

Con el apoyo de tecnologías de software libre para la alta disponibilidad, se logra mantener la información crítica fuera de un punto simple de fallo, donde no sólo la información se mantiene redundante, sino también el enlace de comunicación de los equipos, de esta forma, la prestación de los servicios no se ve alterada por algún evento inesperado en uno de los servidores.

Como resultado de la investigación, se logró diseñar un esquema de red basado en la tecnología OpenVSwitch para la creación y separación adecuada de redes virtuales en el módulo de computación en la nube. Recomendación mencionada en proyectos anteriores que fue una prioridad en el desarrollo de la nueva plataforma CloudEisi.

El desarrollo de una plataforma de prestación de servicios de computación en la nube para comunidades académicas usando soluciones de software libre o abierto, representa una gran oportunidad para el aprovechamiento eficiente de los recursos subutilizados con los que estas cuentan, sin costos de inversión, o de licenciamiento.

Las labores de gestión tecnológica asociadas a la construcción y administración de modelos de prestación de servicios en la nube, requieren de un alto nivel de dedicación laboral y de revisión constante de literatura reciente, al ser éste un modelo en constante evolución y por encontrarse en su etapa inicial.

Los procesos de backup, son más productivos entre menos dependa de ejecuciones manuales para iniciar las labores de respaldo, de esta forma se puede disminuir el daño sufrido en casos de incidentes que implique pérdida de información dando así confianza a sistemas que requieran de este tipo de procesos.

Como resultado de la investigación se logró implementar un sistema de copias de seguridad, con la capacidad de realizar de forma automática respaldo a la plataforma CloudEisi de todos los datos que requieran, y que garantizan la seguridad de estos al momento de salvaguardar y restaurar dicha información.

RECOMENDACIONES

El propósito del entrenamiento de los relevos, es dar a conocer a éstos las diferentes actividades que deben realizar, mostrándole la forma de trabajo que es llevada a cabo en el grupo, evitando inconvenientes a futuro. Por tanto, este proceso de entrenamiento debe ser constante en cada uno de los proyectos subsiguientes relacionados a la labor administrativa.

Enfocar las nuevas investigaciones en el módulo Neutron, para contar con una plataforma 100% a prueba de fallos.

Ya que OpenStack es una plataforma en continuo mejoramiento, se sugiere en los nuevos proyectos enfocados en este tema, ampliar y mejorar los módulos de esta plataforma.

El sistema de copias de seguridad esta implementado y funcionando en un único equipo dedicado a esa labor, cuando se requiera aumentar la capacidad de almacenamiento es recomendable que el nuevo equipo que ayude al respaldo sea igualmente dedicado, al ser un sistema que no consume muchos recursos de maquina puede usarse un equipo que tenga pocos solo con una amplia capacidad de almacenamiento.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Wikipedia, la enciclopedia libre. Disponible: <http://es.wikipedia.org/wiki> [citado 22 de Noviembre de 2014]
- [2] Joyanes Aguilar, Luis. Computación en la Nube: estrategias de Cloud Computing en las empresas. México: Alfaomega Grupo Editor, 2012.
- [3] Fernandez, Leyton. Cloud Computing. Disponible: <http://profesores.elo.utfsm.cl/~agv/elo322/1s10/project/reports/cloudcomputing-10s01.pdf> [citado 1 de Marzo de 2015]
- [4] Sitio web de la comunidad Debian. Disponible: <http://www.debian.org/> [citado 1 de Marzo de 2015]
- [5] Sitio web de KVM. Disponible: <http://www.linux-kvm.org/> [citado 6 de Marzo de 2015]
- [6] ASD (Adaptive Software Development). Disponible: <http://adaptivesoftwaredevelopment.blogspot.com/> [Citado 16 de Marzo de 2015]
- [7] DÍAZ CARREÑO, Emmanuell. 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, 2012.

- [8] BARBOSA AYALA, Alexander; MUÑOZ DUARTE, Elkin Darío. 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.
- [9] ACEVEDO GUTIERREZ, Juan Carlos; DE LAS AGUAS CASTELLON, Daniel David. Implementación de base de datos homogénea distribuida en el aula MeiWEB. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2015.
- [10] MORENO ABAUNZA, César Alberto; LÓPEZ SUÁREZ, Wilmer. Migración De Virtualización En La Infraestructura De Computación En La Nube CloudEisi De La Escuela De Ingeniería De Sistemas E Informática UIS. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2014.
- [11] LIZARAZO TORRES, John Edinson; NOGUERA GIRALDO, Diego Alberto. Administración, mantenimiento, configuración y monitoreo de los equipos servidores del grupo gid-conuss con énfasis en el análisis y reestructuración de los modelos de alta disponibilidad y computación en la nube. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2013.
- [12] Sitio web de Bacula. Disponible: <http://blog.bacula.org/> [citado 2 de Octubre de 2015]

- [13] Sitio web de DebianHelp. Disponible: <http://www.debianhelp.co.uk/index.php>
[citado 8 de Septiembre de 2015]
- [14] Oracle Buys MySQL, Java, and some other stuff. Disponible:
<https://smoothspan.wordpress.com/2009/04/20/oracle-buys-mysql-java-and-some-other-stuff-now-what/> [citado 1 de Julio de 2015]
- [15] MariaDB Galera Cluster Administration. Disponible:
<https://mariadb.com/services/mariadb-mysql-training/mariadb-galera-cluster-administration> [citado 8 de Agosto de 2015]
- [16] OpenStack Installation Guide for Ubuntu 14.04. Disponible:
<http://docs.openstack.org/kilo/install-guide/install/apt/openstack-install-guide-apt-kilo.pdf> [citado 13 de junio de 2015]
- [17] Sitio web de HAProxy. Disponible: <http://www.haproxy.org/> [citado 21 de Agosto de 2015]
- [18] Sitio web de Keepalived. Disponible: <http://keepalived.org/> [citado 25 de Agosto de 2015]
- [19] OpenStack High Availability Guide Disponible: <http://docs.openstack.org/ha-guide/> [citado 27 de Agosto de 2015]

BIBLIOGRAFIA

ACEVEDO GUTIERREZ, Juan Carlos; DE LAS AGUAS CASTELLON, Daniel David. Implementación de base de datos homogénea distribuida en el aula MeiWEB. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2015.

ASD (Adaptive Software Development). [Disponible: <http://adaptivesoftwaredevelopment.blogspot.com/>] [Citado 16 de Marzo de 2015]

BARBOSA AYALA, Alexander; MUÑOZ DUARTE, Elkin Darío. 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.

DÍAZ CARREÑO, Emmanuell. 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, 2012.

FERNANDEZ, LEYTON. Cloud Computing. [Disponible: <http://profesores.elo.utfsm.cl/~agv/elo322/1s10/project/reports/cloudcomputing-10s01.pdf>] [Citado 1 de Marzo de 2015]

JOYANES AGUILAR, Luis. Computación en la Nube: estrategias de Cloud Computing en las empresas. México: Alfaomega Grupo Editor, 2012.

LIZARAZO TORRES, John Edinson; NOGUERA GIRALDO, Diego Alberto. Administración, mantenimiento, configuración y monitoreo de los equipos servidores del grupo gid-conuss con énfasis en el análisis y reestructuración de los modelos de alta disponibilidad y computación en la nube. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2013.

MARIADB GALERA CLUSTER ADMINISTRATION. [Disponible: <https://mariadb.com/services/mariadb-mysql-training/mariadb-galera-cluster-administration>] [citado 8 de Agosto de 2015]

MORENO ABAUNZA, César Alberto; LÓPEZ SUÁREZ, Wilmer. Migración De Virtualización En La Infraestructura De Computación En La Nube CloudEisi De La Escuela De Ingeniería De Sistemas E Informática UIS. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2014.

OPENSTACK HIGH AVAILABILITY Guide [Disponible: <http://docs.openstack.org/ha-guide/>] [citado 27 de Agosto de 2015]

OPENSTACK INSTALLATION GUIDE FOR UBUNTU 14.04. [Disponible: <http://docs.openstack.org/kilo/install-guide/install/apt/openstack-install-guide-apt-kilo.pdf>] [citado 13 de junio de 2015]

ORACLE BUYS MYSQL, JAVA, AND SOME OTHER STUFF. [Disponible: <https://smoothspan.wordpress.com/2009/04/20/oracle-buys-mysql-java-and-some-other-stuff-now-what/>] [citado 1 de Julio de 2015]

SITIO WEB DE BACULA. [Disponible: <http://blog.bacula.org/>] [citado 2 de Octubre de 2015]

SITIO WEB DE DEBIANHELP. [Disponible: <http://www.debianhelp.co.uk/index.php>] [citado 8 de Septiembre de 2015]

SITIO WEB DE HAPROXY. [Disponible: <http://www.haproxy.org/>] [citado 21 de Agosto de 2015]

SITIO WEB DE KEEPALIVED. [Disponible: <http://keepalived.org/>] [citado 25 de Agosto de 2015]

SITIO WEB DE KVM. Disponible: <http://www.linux-kvm.org/> [citado 6 de Marzo de 2015]

SITIO WEB DE LA COMUNIDAD DEBIAN. [Disponible: <http://www.debian.org/>] [citado 1 de Marzo de 2015]

WIKIPEDIA, LA ENCICLOPEDIA LIBRE. [Disponible: <http://es.wikipedia.org/wiki>] [citado 22 de Noviembre de 2014]

ANEXO A. ARTÍCULO

ANALISIS DE LA ARQUITECTURA DE COMPUTACION EN LA NUBE CLOUDEISI CON ENFASIS EN EL MEJORAMIENTO DE LA ADMINISTRACION, FUNCIONALIDAD Y SEGURIDAD.

YONATHAN NAHÚN CARREÑO ORTIZ
JOSÉ ALONSO PÁEZ CRUZ

PALABRAS CLAVE: Virtualización, computación en la nube, OpenStack.

DESCRIPCIÓN: La computación en la nube es un paradigma cuyo uso que se ha venido implementando en los últimos años de forma incrementada y se espera que siga esta tendencia en los siguientes años. Este paradigma básicamente consiste en la prestación de servicios informáticos a través de internet sin necesidad de que los usuarios tengan conceptos avanzados sobre los recursos que utilizan, para ello se requieren de servidores en internet que atiendan peticiones en cualquier momento.

En la actualidad existen numerosas plataformas que prestan servicios de computación en la nube tales como las de Google, Amazon y Microsoft, las cuales son plataformas a gran escala, estas por mencionar algunas, pero existen una gran cantidad de plataformas de menor magnitud; esto debido a los numerosos software de aplicación existentes para gestionar dicho tipo de plataformas.

En el presente trabajo se hace uso de tecnologías emergentes relacionadas a la computación en la nube, las cuales permiten generar un nuevo modelo con el fin de optimizar la funcionalidad de la plataforma CloudEisi la cual alberga un conjunto de servicios que benefician estudiantes a y profesores de la comunidad académica.

KEYWORDS:

Virtualization, cloud computing, OpenStack.

Cloud computing is a paradigm that has been implemented recently and its use is expected to increase over the next years. This paradigm is basically the provision of computer services over the Internet without users having advanced concepts about the resources they use, this will require servers that attend requests at any time.

Currently there are numerous platforms, on a large scale, that provide cloud computing services such as: Google, Amazon y Microsoft. In addition, there is a lot of platforms, of lesser magnitude, due to the numerous application software available to manage them, allowing with a small number of servers, the

right technology and internet access can offer cloud services such as the group CloudEisi CONUSS platform, object of this work.

In this work, emerging technologies related to cloud computing are used, which allow the generation of a new model, in order to optimize the functionality of the CloudEisi platform that offers a set of services that benefit the academic community.

INTRODUCCIÓN

La computación en la nube es una tecnología en continuo crecimiento, muy acogida por grandes empresas por su eficiente administración de la información y recursos. Según el IEEE Computer Society, es un paradigma en el que la información se almacena de manera permanente en servidores de Internet y se envía a cachés temporales de cliente, lo que incluye equipos de escritorio, centros de ocio, portátiles, etc. [1] Inspirado su modelo de los cajeros automáticos, en el cual está disponible el dinero desde cualquier punto de servicio, su implementación en la red origino “CloudComputing”, tecnología que gracias a la virtualización y la computación paralela es capaz de brindar servicios a usuarios en tiempo real, totalmente estable y seguro gracias a su infraestructura de alta disponibilidad.

MARCO DE REFERENCIA

Cloud Computing

El concepto de computación en la nube se empezó a utilizar en 2006 por proveedores como Google, Amazon, Microsoft las cuales habían construido sistemas de recursos distribuidos de manera horizontal, introducidos como servicios virtuales de TI, escalados de manera continua el cual se reflejó en el artículo “las fábricas de información” de la revista Wired Magazine, escrito por George Gilder; en el cual plantea un modelo de nube virtual, similar en estructura a la computación grid [3].

El NIST (Instituto Americano de Estándares y Tecnología) como un modelo que permite el acceso bajo la demanda a través de la red a un conjunto compartido de recursos de computación configurables que pueden rápidamente aprovisionados con el mínimo esfuerzo de gestión e iteración del proveedor de servicio enfocado a su

uso en la web permitiendo el acceso a un conjunto compartido de recursos de computación configurables que pueden proveerse y liberarse rápidamente de administración basándose en la virtualización, ofreciendo los siguientes capas generales de servicios:

- **Software como servicio (SaaS):** Consiste en el suministro de aplicaciones software del proveedor, utiliza una arquitectura web accesible desde cualquier dispositivo y el usuario no tiene ningún control sobre la infraestructura; en este tipo de servicio el usuario se ve comprometido a mantener actualizadas las licencias entre las empresas que ofrecen este tipo de servicios se encuentran Salesforce y BaseCamp.

- **Infraestructura como servicio (IaaS):** Ofrece servicios de computación y almacenamiento en lo relativo al aprovisionamiento de servidores y discos donde el usuario alquila recursos de computación comportándose como la capa física de la nube el proveedor se encarga de toda el mantenimiento y optimización de los recursos del equipamiento hardware, entre las cuales se destacan empresas como Amazon Web Service, EC2 y GoGrid.

- **Plataforma como Servicios(PaaS):** El proveedor ofrece un entorno para facilitar el trabajo de los programadores, encapsulando el entorno de desarrollo facilitando el ciclo de construcción y puesta en marcha de aplicaciones, manteniendo la seguridad y escalabilidad de los programas, la plataforma como servicio (PaaS) está muy ligada a la capa software como servicio (SaaS) porque esta es la plataforma donde se desenvuelve el software que está a disposición del proveedor y es el medio de virtualización para el hardware que el cliente alquila y todo se ejecuta desde la plataforma externa y el usuario no necesita descargar nada, una de las empresas que ofrece este servicio el Google App Engine.

VIRTUALIZACIÓN

Virtualización es uno de los pilares de la computación en la nube, es el uso de los recursos de las computadoras para simular o imitar a otros recursos de ellas o las propias computadoras en su totalidad. La virtualización es un mecanismo fundamental para la entrega de servicios; proporciona una plataforma para la optimización de recursos de TI complejos de un modo escalable (crecimiento de recursos de modo eficiente y adaptable a los servicios a entregar) [2].

ALTA DISPONIBILIDAD

Tomando la definición expuesta por el IEEE TFCC1, 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 otro extremo con soluciones que ofrecen las soluciones de software que utilizan componentes de hardware genéricos.”[6]

KVM (Kernel-based Virtual Machine)

KVM es una solución para implementar virtualización completa con Linux. Está formada por un módulo del núcleo (con el nombre kvm.ko) y herramientas en el espacio de usuario, siendo en su totalidad software libre. El componente KVM para el núcleo está incluido en Linux desde la versión 2.6.20.

KVM permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio hardware virtualizado: Una tarjeta de red, discos duros, tarjeta gráfica, etc. [4] KVM será la base de virtualización implementada por ser software libre y además contar con una vasta documentación gracias a la comunidad Linux.

SITUACION ACTUAL

La infraestructura como servicio CloudEisi le presta a la comunidad EISI el servicio de instanciación de máquinas virtuales para el desarrollo de proyectos de grado, actividades de investigación, hosting para asignaturas que implementan aplicaciones web, repositorio de aplicaciones web, repositorio de imágenes digitales, servicios de aulas virtuales LMS (Learning Management System). Está soportada por tres servidores de altas prestaciones, su esquema cuenta con las siguientes características generales:

- Sistemas operativos basados en Linux tipo servidor.
- Uso de máquinas virtuales configuradas con los componentes base de la infraestructura.
- Sistema de alta disponibilidad para asegurar el continuo servicio en caso fallas por parte de uno de los nodos que pertenecen al clúster.
- Servicio de asignación de instancias virtuales.
- Repositorio de imágenes virtuales preconfiguradas con servicios listos para ser usados.
- Interfaz administrativa para manejo de recursos virtuales.

- Acuerdo de prestación de servicios con los usuarios.
- Esquema de red interna virtual para la comunicación entre instancias virtuales.
- Punto de salida para la comunicación hacia internet de los servicios virtuales.

Dicha infraestructura ha estado evolucionando desde su creación como un clúster de alta disponibilidad, hoy día es un clúster de alta disponibilidad que ofrece servicios en la nube.

ESQUEMA GENERAL

El esquema físico de los equipos está dado como se muestra en la siguiente figura:

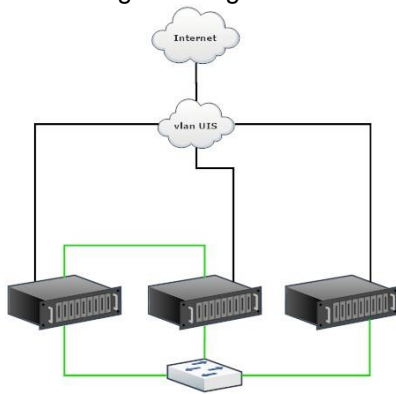


Figura 1. Esquema general de la infraestructura. Fuente: Los Autores

Los hosts se conectan a la red de la universidad, y ésta brinda el servicio de conexión a internet que permite dar salida a los servicios de cada servidor. La conexión dedicada presente entre los dos servidores es usada por el sistema de alta disponibilidad para respaldar el componente de Computación en la Nube. El tercer servidor es empleado para soportar la instanciación de máquinas virtuales en la Nube, junto con los otros dos servidores, y además se emplea como nodo de repositorio de imágenes y almacén de copias de seguridad.

ESTRUCTURA GENERAL DE CADA SERVIDOR

La composición interna de cada servidor está dada por un enfoque basado en componentes.

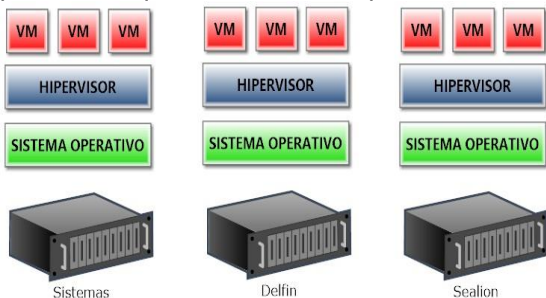


Figura 2. Estructura general de cada servidor. Fuente: Los Autores

Con esta estructura se obtiene un diseño simple y de fácil manejo, ya que los servicios no se están ejecutando de forma nativa en cada servidor, si no en instancias virtuales individuales, por lo que su configuración, respaldo, migración y recuperación no afectan de forma total los demás componentes de la infraestructura.

ESTRUCTURA COMPUTACIÓN EN LA NUBE

Esta plataforma de computación en la Nube CloudEisi tiene como usuarios finales a los miembros de la comunidad UIS. El punto de acceso del servicio es una máquina virtual configurada como frontend, esta instancia virtual es la encargada de administrar y monitorear las máquinas virtuales usadas por los usuarios. Como las instancias de servicio pueden estar en diferentes servidores, el frontend tiene la tarea de realizar el direccionamiento del tráfico a través de la red interna de la infraestructura. Ante la posible falla del frontend es posible reanudar el servicio de la infraestructura gracias a un frontend de respaldo que es encendido de forma manual y toma el control de la infraestructura, dicho frontend de respaldo se encuentra instanciado en otro servidor y comunicado a través de un sistema de alta disponibilidad de datos que lo mantiene actualizado con relación a su homónimo.

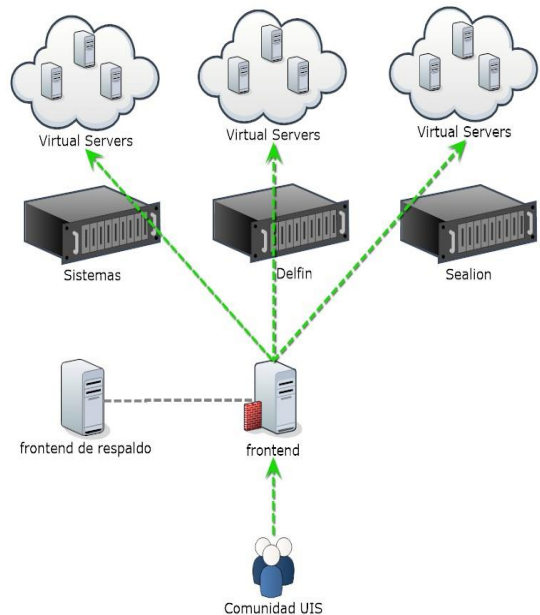


Figura 3. Componentes Servicio Computación en la Nube. Fuente: Los autores

Otra de las funciones del frontend es la de filtrado de peticiones que van de la red externa a la interna bloqueando a los usuario malintencionados. Como función administrativa tiene la capacidad de crear

instancias virtuales para usuarios finales según los requerimientos.

La herramienta usada en el frontend para la administración de la nube es OpenNebula, se trata de una herramienta de administración para centros de datos virtualizados bajo la premisa de infraestructura como servicio para la nube (IaaS).

El Frontend al ser una máquina virtual instanciada en uno de los servidores, es susceptible a este. Si por alguna razón el servidor que lo contiene deja de funcionar, el administrador configurara el Frontend de respaldo para mantener activa la infraestructura. Actualmente este proceso se hace de forma manual, lo que obliga al administrador estar al pendiente del sistema 24/7 por si este presenta algún fallo.

Adicional a esto, la infraestructura está muy ligada al funcionamiento del tercer servidor. Ya que este hace de DataStore, si falla, el sistema quedara inactivo en lo que se refiere a nuevas instancias solicitadas.

PROTOTIPO

El planteamiento del prototipo inicio con la selección de la plataforma CloudComputing a usar, se realizaron estudios para determinar si era conveniente conservar la infraestructura actual o evaluar una nueva.

Tras una revisión del estado actual de la infraestructura CloudEisi se optó por crear un nuevo diseño, una de las razones por las que se tomó esta decisión es porque actualmente la plataforma CloudEisi presenta muchas falencias, una de ellas es que el nodo de administración OpenNebula no se encuentra balanceado, ocasionando que si el servidor que lo contiene deja de funcionar, la plataforma CloudEisi también. Como no se dispone del material administrativo que detalla la configuración original de toda la infraestructura CloudEisi se decidió evaluar y proponer una nueva que fue avalada por el director de proyecto.

Para la selección de la plataforma CloudComputing y demás herramientas que permitieron la obtención de resultados, se fijaron ciertos parámetros:

- Las licencias debían ser gratuitas, sin limitaciones de uso por cantidad de usuario.
 - La infraestructura debía correr principalmente en el hipervisor KVM
- Por tratarse de un proyecto académico se buscó no incurrir en gastos adicionales, además las herramientas a escoger deben poseer suficiente documentación y desarrollo para asegurarse de que no solamente se pueda cumplir satisfactoriamente

con el objetivo de la migración, también que permita las mejoras y escalabilidad en un futuro.

SELECCIÓN DE LA PLATAFORMA

Actualmente las plataformas CloudComputing de código abierto más relevantes son: OpenStack, OpenNebula, Eucalyptus y CloudStack. Seleccionar cualquiera de las mencionadas anteriormente, es una buena decisión debido a que cuentan con una gran acogida en el mercado y documentación suficiente para crear una nube privada.

El criterio que llevó a seleccionar la plataforma que se utilizó, fueron los siguientes:

- Comunidad activa en los foros.
- Contribuyentes que impulsaran el continuo mejoramiento de la plataforma.
- Amplia documentación administrativa referente a sus componentes.

El bloguero chino Qingye Jiang ha elaborado un estudio donde se analizaron y compararon los datos de foros y listas de correo para la actividad de la comunidad OpenStack, OpenNebula, eucalipto y el proyecto CloudStack. Los datos se extrajeron de los foros de debate público y comunicaciones entre los miembros de la comunidad en forma de listas de correo.

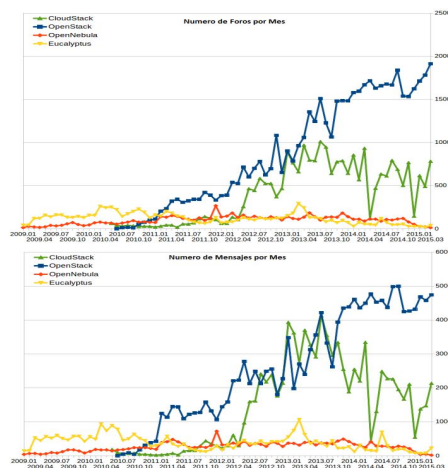


Figura 39. Actividad de Usuarios de cada Comunidad. Fuente: <http://www.qyjohn.net/>

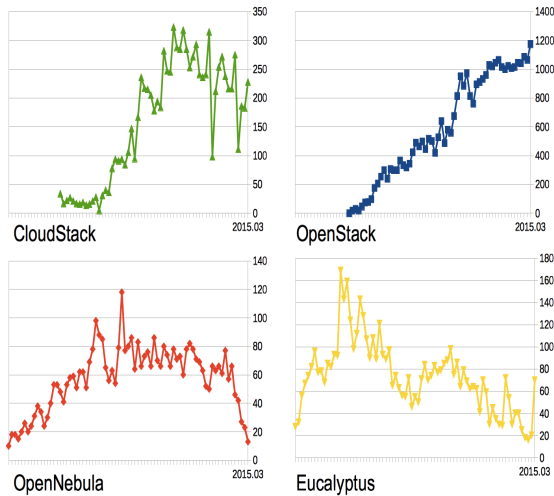


Figura 5. Participación activa de la Comunidad en los Foros. Fuente: <http://www.qyjohn.net/>

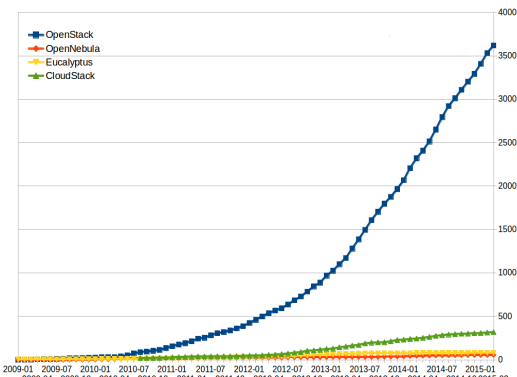


Figura 6. Población que Contribuye en los Proyectos. Fuente: <http://www.qyjohn.net/>

Basado en la información antes presentada, se decidió usar la plataforma OpenStack, ya que esta además de tener una gran comunidad activa y un número notable de contribuyentes que día a día van mejorando la plataforma, también cuenta con una estructura modular lo cual permite instalar una plataforma básica que puede ser expandida de forma continua. Lo mencionado anteriormente hace de OpenStack la plataforma adecuada para estudios académicos permitiendo la escalabilidad de un prototipo inicial.

OPENSTACK

OpenStack es una plataforma con una arquitectura modular con varios nombres para sus componentes: Nova, Glance, Swift, Horizon, Keystone, Neutron, Cinder, Heat, Ceilometer, Trove, Sahara. Gracias a su arquitectura es posible elegir los módulos claves a implementar para conseguir una plataforma funcional básica, tomando esto en cuenta se decidieron implementar

en el prototipo: Keystone, Glance, Nova, Neutron y Horizon, de la versión Kilo de OpenStack.

DISEÑO DE LA INFRAESTRUCTURA

El diseño de la infraestructura tiene como prioridad la alta disponibilidad de la plataforma CloudEisi. Se planteó un modelo de una fase inicial el cual cumple con dicha característica ilustrado en la siguiente gráfica.

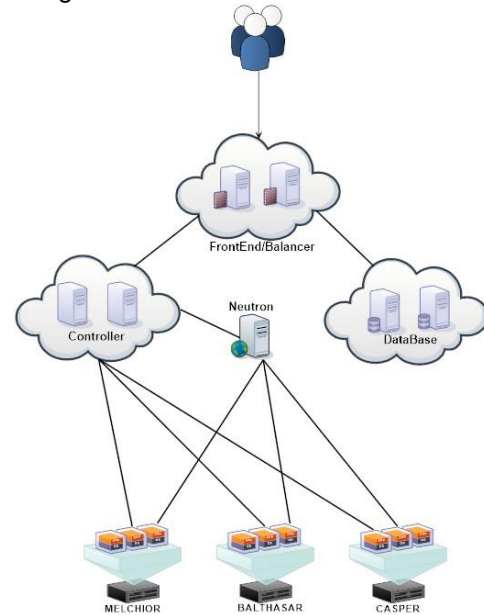


Figura 7. Modelo Plataforma CloudEisi. Fuente: los autores

De acuerdo a la figura se puede observar que el nodo controller (administración OpenStack), el FrontEnd y la base de datos se encuentran en alta disponibilidad.

El nodo Neutron encargado de brindar el acceso a la red de las instancias administradas por OpenStack no se plantea en alta disponibilidad, pero por medio de la investigación se encontró que es posible realizarse mediante una herramienta de administración (OpenVSwitch) que administra las redes virtuales de la plataforma, dicha herramienta necesita replicar las redes virtuales de un nodo al otro, pese a que dicho nodo no se plantea en alta disponibilidad el diseño de la plataforma contempla la creación de una copia de respaldo de la máquina virtual Neutron por medio una herramienta de backups.

El modelo contará con dos máquinas virtuales para cada nodo con una configuración Activo/Activo de alta disponibilidad a excepción del Frontend que contará con una configuración Activo/Pasivo.

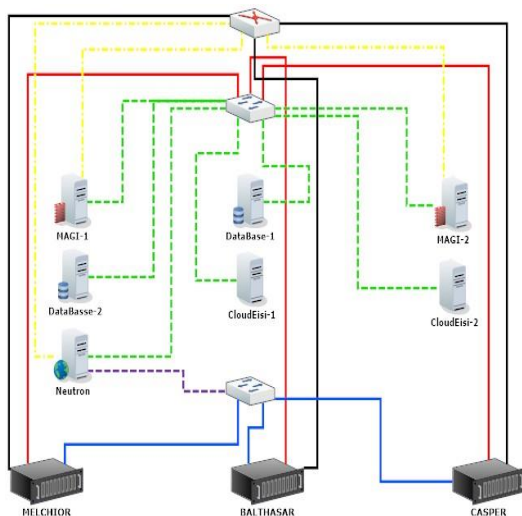


Figura 8. Disposición Plataforma CloudEisi. Fuente: los autores

En la figura se aprecia la disposición de los nodos en cada uno de los servidores físicos, esta organización está pensada por si falla un servidor la plataforma no presente interrupción en el servicio, incluso si el nodo Neutron deja de funcionar la plataforma estará en servicio pero la red de las maquinas no funcionara, es aquí donde se usa una herramienta de backups para restaurar la máquina virtual en otro servidor.

OpenStack hace uso de tres redes para su funcionamiento, dos internas y una externa que se puede apreciar en la Figura 17. una de las redes internas es de uso exclusivo de OpenStack para la administración de las redes virtuales de las instancias y la otra red es para la comunicación de todos los módulos y nodos de la infraestructura, la red externa la tendrán disponible los nodos que necesitan de ella como son, el Frontend y el nodo Neutron.

4 GESTOR DE BASES DE DATOS

Se estudió la manera de crear una base de datos distribuida capaz de soportar los datos de administración de OpenStack, recientemente estudiantes del grupo CONUSS de la Universidad Industrial de Santander [5] investigaron sobre este tema, basaron su investigación sobre MySQL y sus resultados concluyeron que esta herramienta no soportaría la administración de OpenStack. Gracias a esta investigación se buscó una herramienta de administración de base de datos OpenSource que permitiera lograr este fin. Se optó por implementar MariaDB. (MySQL después de ser comprado por Oracle) [7] freno su investigación en el campo de base de datos distribuidas, a raíz de esto Michael (Monty) Widenius (fundador de MySQL) desarrollo MariaDB.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL con licencia GPL [8], esta cuenta con una herramienta llamada Galera Clúster que permitió crear la base de datos distribuida.

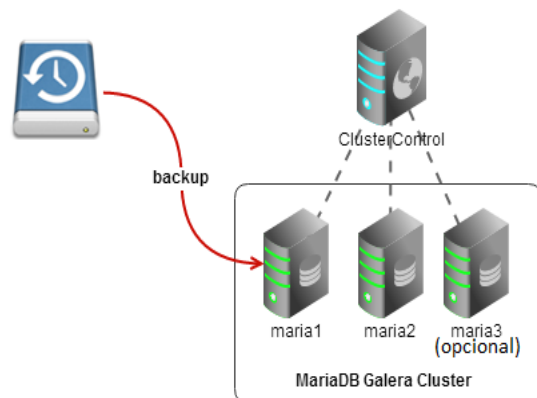


Figura 9. Estructura Clúster HA MariaDB. Fuente: <http://severalnines.com/>

En la figura se aprecia el modelo planteado para la plataforma, siendo el ClusterControl el nodo Frontend, dicho planteamiento permite N nodos de MariaDB.

CONCLUSIONES

Las labores de gestión tecnológica asociadas a la construcción y administración de modelos de prestación de servicios en la nube, requieren de un alto nivel de dedicación laboral y de revisión constante de literatura reciente, al ser éste un modelo en constante evolución y por encontrarse en su etapa inicial.

Como resultado de la investigación, se plantea un modelo de plataforma en la nube de alta disponibilidad, mediante el uso de herramientas de software libre, unos servicios computacionales confiables para su uso en la Escuela de Ingeniería de Sistemas y la comunidad académica en general.

El desarrollo de una plataforma de prestación de servicios de computación en la nube para comunidades académicas usando soluciones de software libre o abierto, representa una gran oportunidad para el aprovechamiento eficiente de los recursos subutilizados con los que estas cuentan, sin costos de inversión, o de licenciamiento.

BIBLIOGRAFÍA

- [20] Wikipedia, la enciclopedia libre. Disponible: <http://es.wikipedia.org/wiki> [citado 22 de Noviembre de 2014]
- [21] Joyanes Aguilar, Luis. Computación en la Nube: estrategias de Cloud Computing en las

- empresas. México: Alfaomega Grupo Editor, 2012.
- [22] Fernandez, Leyton. Cloud Computing. Disponible: <http://profesores.elo.utfsm.cl/~agv/elo322/1s10/project/reports/cloudcomputing-10s01.pdf> [citado 1 de Marzo de 2015]
- [23] Sitio web de la comunidad Debian. Disponible: <http://www.debian.org/>[citado 1 de Marzo de 2015]
- [24] Sitio web de KVM. Disponible: <http://www.linux-kvm.org/> [citado 6 de Marzo de 2015]
- [25] ASD (Adaptive Software Development). Disponible: <http://adaptivesoftwaredevelopment.blogspot.com/> [Citado 16 de Marzo de 2015]
- [26] DÍAZ CARREÑO, Emmanuell. 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, 2012.
- [27] BARBOSA AYALA, Alexander; MUÑOZ DUARTE, Elkin Darío. 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.
- [28] ACEVEDO GUTIERREZ, Juan Carlos; DE LAS AGUAS CASTELLON, Daniel David. Implementación de base de datos homogénea distribuida en el aula MeiWEB. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2015.
- [29] MORENO ABAUNZA, César Alberto; LÓPEZ SUÁREZ, Wilmer. Migración De Virtualización En La Infraestructura De Computación En La Nube CloudEisi De La Escuela De Ingeniería De Sistemas E Informática UIS. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2014.
- [30] LIZARAZO TORRES, John Edinson; NOGUERA GIRALDO, Diego Alberto. Administración, mantenimiento, configuración y monitoreo de los equipos servidores del grupo gid-conuss con énfasis en el análisis y reestructuración de los modelos de alta disponibilidad y computación en la nube. Bucaramanga. Universidad industrial de Santander. Facultad de Ingenierías Fisicomecánicas. Escuela de ingeniería de sistemas e informáticas, 2013.
- [31] Oracle Buys MySQL, Java, and some other stuff. Disponible: <https://smoothspan.wordpress.com/2009/04/20/oracle-buys-mysql-java-and-some-other-stuff-now-what/> [citado 1 de Julio de 2015]
- [32] MariaDB Galera Cluster Administration. Disponible: <https://mariadb.com/services/mariadb-mysql-training/mariadb-galera-cluster-administration> [citado 8 de Agosto de 2015]
- [33] OpenStack Installation Guide for Ubuntu 14.04. Disponible: <http://docs.openstack.org/kilo/install-guide/install/apt/openstack-install-guide-apt-kilo.pdf> [citado 13 de junio de 2015]
- [34] Sitio web de HAProxy. Disponible: <http://www.haproxy.org/> [citado 21 de Agosto de 2015]
- [35] Sitio web de Keepalived. Disponible: <http://keepalived.org/> [citado 25 de Agosto de 2015]
- [36] OpenStack High Availability Guide Disponible: <http://docs.openstack.org/ha-guide/> [citado 27 de Agosto de 2015]