

**IMPLEMENTACIÓN DE UN SERVICIO OPENSTACK EN EL PROGRAMA DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD DEL SINÚ**

LUIS FERNANDO BELLIO MEZA

**PROGRAMA DE INGENIERIA DE SISTEMAS
FACULTAD DE INGENIERIAS**



Año de 2017

**IMPLEMENTACIÓN DE UN SERVICIO OPENSTACK EN EL PROGRAMA DE
INGENIERÍA DE SISTEMAS DE LA UNIVERSIDAD DEL SINÚ**

LUIS BELLIO MEZA

**Trabajo de grado presentado como requisito para optar al título de
INGENIERO DE SISTEMAS**

Director: Rafael Monterrosa

Título del director

Co-Director: Carlos Arenas

Título del co-director

**PROGRAMA DE INGENIERIA DE SISTEMAS
FACULTAD DE INGENIERIAS
UNIVERSIDAD DEL SINÚ ELIAS BECHARA ZAINÚM**

Seccional Cartagena, Noviembre 14 de 2017

AGRADECIMIENTOS

Agradezco a Dios por esta bendición y entregarnos la sabiduría suficiente para el entendimiento y culminación de nuestra carrera universitaria. A mi familia, por ser el apoyo incondicional que siempre ha estado presente en cada momento y lugar que se los necesita y a nuestros tutores Rafael Monterrosa y Carlos Arenas por el soporte brindado en el transcurso del proyecto y su recomendación de OpenStack como temas del proyecto.

TABLA DE CONTENIDO

1. INTRODUCCIÓN.....	6
2. EL PROBLEMA DE INVESTIGACION	7
2.1 DESCRIPCION DEL PROBLEMA	7
2.2 JUSTIFICACION	¡Error! Marcador no definido.
2.3 OBJETIVOS DEL PROYECTO	9
2.3.1 Objetivo general	9
2.3.2 Objetivos específicos.....	9
3. MARCOS DE REFERENCIA.....	10
3.1 ANTECEDENTES	10
3.2 MARCO TEORICO	11
3.3 MARCO CONCEPTUAL	19
3.4 MARCO LEGAL	21
4. DISEÑO DEL PROYECTO.....	25
4.1 LINEA DE INVESTIGACION.....	26
4.2 ALCANCES.....	26
4.2.1 Alcance del proyecto	26
4.3 PRESUPUESTO.....	27
4.4 DIAGRAMAS Y TABLAS	¡Error! Marcador no definido.
5. Capítulos.....	28
6. CONCLUSIONES	¡Error! Marcador no definido.
7. REFERENCIAS BIBLIOGRÁFICAS	39

LISTA DE FIGURAS

Figura 1. Estructura de Openstack.....	¡Error! Marcador no definido.	5
Figura 2. Proyectos de Openstack	¡Error! Marcador no definido.	7
Figura 3. Modelos de servicios de cloud computing.....		19
Figura 4. Tipos de servicios de Openstack		21
Figura 5. Diseño logico de nuestro sistema de dos nodos		26
Figura 6. Diseño fisico de nuestro sistema.....		26
Figura 7. Cronograma de actividades		27

1. INTRODUCCIÓN

El siguiente proyecto trata de la implementación de un prototipo de *Cloud Computing* basado en OpenStack que permite la creación de máquinas virtuales. Existen tres modelos de Servicios los cuales son Software como Servicio (SaaS), plataforma como Servicio (PaaS), Infraestructura como Servicio (IaaS).

OpenStack es un sistema que ofrece una infraestructura como servicio (IaaS), en la que los usuarios podrán albergar sus sistemas virtualizados. Por esta razón lo escogimos para la realización de nuestro proyecto el cual tiene como objetivo implementar un recurso para la virtualización, la comunidad educativa de la Escuela de Ingeniería de Sistemas ya que no se pueden desarrollar algunas competencias académicas porque los computadores no cuentan con las características apropiadas, además, la escuela de ingeniería de sistemas no es la única que utiliza los equipos, por lo cual se dificulta más el proceso ya que no se pueden personalizar para nuestro aprendizaje. Por esto se pretende crear un sistema de virtualización con el fin de mejorar el desarrollo del manejo de los equipos y así el área de sistema pueda desarrollar los ejercicios en dichos momentos.

2. EL PROBLEMA DE INVESTIGACION

2.1 DESCRIPCION DEL PROBLEMA

En el programa de ing. De sistemas de la institución universitaria del Sinú se ha podido observar que algunas competencias académicas del área de redes y servicios no se desarrollan de la forma más óptima debido a los PC no cuentan con las características de hardware apropiadas, para la configuración de servicios y virtualización de sistemas operativos.

Los laboratorios de sistemas de la institución universitaria del Sinú se encuentran divididos por áreas y son utilizados por las diferentes carreras y estudiantes que se tiene en la universidad, , por lo tanto, no pueden ser utilizados por cualquier estudiante, ya que al estar libre el estudiante tendría que verificar si el tipo de software instalado es el mismo que necesita para realizar sus tareas o prácticas.

Otra situación a la que se enfrenta los estudiantes de la universidad del Sinú es la seguridad de su información, los dispositivos físicos corren riesgo como pueden ser, extravío, robo, daño por negligencias o catástrofes naturales, usuarios mal intencionados, etc., estos riesgos podrían causar la pérdida de la información valiosa y que muy pocas veces puede salvar. Los usuarios a veces requieren acceder a su información desde otros lugares, para esto tendrían que llevar con ellos sus dispositivos físicos como computadoras o memorias, esto a veces resulta complejo, engorroso y no es muy práctico.

2.2 JUSTIFICACIÓN

La presente justificación se basa en un sistema de virtualización el cual permite la creación de máquinas virtuales, las cuales se utilizarán para mejorar el desarrollo académico de las asignaturas de redes y servicios.

Entre los mayores beneficios que se puede observar se encuentra la reducción de costos de hardware, debido a que se implementará la Infraestructura como Servicio (IaaS), con lo cual se pretende eliminar los servidores físicos e implementarlos virtualmente en la nube.

Otro beneficio es la movilidad que nos permite este servicio, ya que los datos y servicios se montan en la nube privada y pueden ingresar a las máquinas virtuales y ser buscados por los estudiantes desde cualquier lugar de la institución que tengan acceso de a la red, Sede plaza colon

3. OBJETIVOS DEL PROYECTO

3.1 Objetivo general

Implementar un prototipo con el sistema de virtualización de OpenStack como recurso para la virtualización, de sistemas operativos tipo servidores de la Escuela de Ingeniería de Sistemas.

.

3.2 Objetivos específicos

- Diseño de topología de red y el servicio OpenStack
- Implementar los servicios de virtualización en servidor
- Instalación de s.o
- Pruebas de conectividad

4. MARCOS DE REFERENCIA

4.1 Antecedentes

En las últimas décadas los procesos de deslocalización e internacionalización de las grandes empresas, unidos a la explosión en el uso de tecnologías de información y procesamiento de datos, han hecho que las necesidades de cómputo hayan crecido a un ritmo superior al que lo hacía la capacidad de cálculo de los ordenadores personales.

Debido a esto, se ha venido realizando un importante esfuerzo en la investigación de capacidades para la ejecución de procesos en múltiples computadores. Esta tendencia fue impulsada originalmente por la utilización de sistemas abiertos, interoperables y protocolos de comunicación estándar que permitían la comunicación eficiente entre sistemas y tecnologías heterogéneas.

Como primer paso Unix permitió la configuración de clusters, es decir, agrupaciones de ordenadores con componentes de hardware comunes que se comportan como una única computadora.

Más tarde el sistema operativo Linux mediante sus estándares abiertos permitió implementar clusters basados en la arquitectura estándar de los PC, consiguiendo instalaciones de cálculo de alto rendimiento a bajos precios y popularizando esta solución durante la década de 1990.

Lo que permitió ofrecer servicios a través de protocolos estándar, constituyendo la denominada arquitectura de computación grid, orientada al procesamiento en paralelo o al almacenamiento de gran cantidad de información, pero la complejidad para utilizar la infraestructura, las dificultades para utilizarla, y los problemas de portabilidad entre

ellas, hicieron que nunca se popularizara fuera del ámbito de la investigación y académico.

A mediados de 2010 se produjo un 'evento sistémico' en el mundo del software de Infraestructura como Servicio (IaaS). Rackspace y la NASA anunciaban de manera conjunta que estaban desarrollando un nuevo software de IaaS que venía a solucionar los problemas que la NASA le había encontrado a Eucalyptus, y a convertirse en una alternativa real a Amazon Web Services.

La NASA buscaba algo que escalase a lo bestia-bestia: un millón de servidores o 60 millones de máquinas virtuales. Para conseguir eso, es necesario abrir las tripas de Eucalyptus y modificarlo de tal manera que soporte ese nivel de escalado. Pero... resulta que no es 'abierto' de verdad... y la NASA decidió que no podía meter recursos en algo que no controlaba.

Así que OpenStack nació como una alternativa completamente abierta al amparo de Rackspace y gran parte de la industria. A diferencia de Eucalyptus no es un producto, sino un 'Framework'. Es decir, puede manipularse de tal manera que puede adaptarse a las necesidades de los clientes. Y por supuesto todo el código se encuentra disponible bajo licencia Apache 2.0.

4.2 MARCO TEORICO

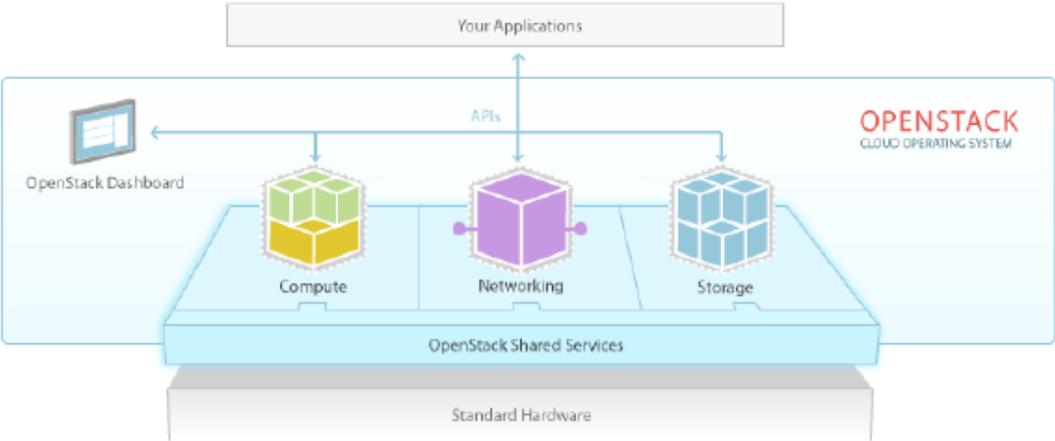
4.2.1 OpenStack

Openstack se define comercialmente como un sistema operativo en la nube que permite controlar recursos de computación, almacenamiento y redes a través de un dashboard o cuadro de mando y APIs estándares de programación. Su arquitectura es distribuida y utiliza diferentes tipos de nodos especializados en distintos servicios y conectados mediante redes de gestión y datos.

OpenStack es una solución de computación en la nube que permite la construcción de nubes públicas, híbridas y privadas que proporcionen Infraestructura como Servicio (IaaS). Nacido de una iniciativa conjunta de Rackspace y la NASA en Julio de 2010, su popularidad y uso desde su lanzamiento se ha disparado. Más de 200 compañías, incluyendo a IBM, Cisco, Canonical o Red Hat3 apoyan el proyecto en la actualidad. Consiste en un conjunto de proyectos que pueden instalarse y configurarse para que funcionen de manera independiente o para que se interrelacionen. Estos proyectos incluyen Servicio de Identidad (Keystone), Servicio de Imágenes (Glance), Servicio de Computación (Nova), Servicio de Red (Neutron), Servicio de Almacenamiento en Bloques (Cinder), Almacenamiento de Objetos (Swift), Procesamiento de Datos (Sahara), Telemetría (Ceilometer), Orquestación (Heat) y Base de Datos (Trove). Debido a que la comunidad de desarrollo es sumamente activa y cuenta con un gran respaldo de financiamiento, en los últimos años han surgido nuevos proyectos que ofrecen la posibilidad de integrar a OpenStack con las últimas tecnologías de virtualización, red y almacenamiento que han tenido éxito en el sector IT.

La arquitectura de OpenStack es totalmente modular y en la actualidad engloba dos proyectos principales: OpenStack Compute (NOVA aportado por la NASA) y OpenStack Object Storage (SWIFT, aportado por Rackspace). Con base en estos dos proyectos, OpenStack, posee tres módulos principales: Compute, Networking y Storage, a partir de los cuales se han desarrollado un conjunto de componentes que permiten la administración y gestión del cloud, de los cuales los principales son: Nova, Glance, Swift, Horizon, Keystone, Neutrón. Además, dependiendo de la versión de la que se trate se cuentan con otros componentes¹, los cuales se comunican mediante APIs provistas por el componente.

Figura 1 Estructura de open stack



4.2.2 Características De Openstack

Según la documentación oficial de la página del proyecto OpenStack cuenta con las siguientes características:

- Administración de los recursos del servidor: como el CPU, memoria, disco, interfaces de red, estos recursos deben ser manejados de una forma rápida a través de la virtualización de servidores para su implementación para lograr que los recursos sean optimizados y manejados como maquinas físicas.
- Administración de la LAN: Permite manejar redes tanto públicas como privadas, pero al tratarse de una nube privada.
- Administración de APIs para gestión de velocidad y servicios de autenticación: Openstack cuenta con API's que permiten realizar controles de telemetría, para el análisis de tráfico y carga en los servidores, también permiten 43 un nivel de seguridad a través de autenticación para los accesos a los servicios que se están brindando.
- Administración de Arquitectura: Este tipo de arquitecturas son soportadas en Openstack para aplicaciones distribuidas en donde las cargas son administradas mediante software especializado que realiza balanceo de carga y brinda escalabilidad de la nube.
- Administración de las Imágenes: Permite mantener un control de imágenes, las cuales representan Sistemas Operativos, en donde se desplegaran los servicios, y posteriormente la información de la organización.
- Administración de Seguridad: Permite a través de llaves de acceso, tener asegurada la infraestructura creada por el administrador. Y a través de estas llaves se asegura la disponibilidad e integridad en los servidores desplegados.
- Administración de Roles: Para el servicio de integridad, se crean roles los cuales según la jerarquía, están creados de acuerdo a las políticas de la empresa para cumplir con los estándares de seguridad de la información.
- Administración de Almacenamiento a través de APIs: Al igual que los servicios antes mencionados, el almacenamiento es uno de los más importantes pues

aquí están los activos de la organización y en el caso que estos se vean afectados, pondrían en peligro la estabilidad de la empresa.

4.2.3 Servicios que proporciona Openstack

Openstack es un conjunto de proyectos que ofrecen servicios interrelacionados que ofrecen una Interfaz de Programación de Aplicaciones (API) para la administración de computación, almacenamiento y red que forman parte de la infraestructura de Nube.

Figura 2. Proyectos de openstack

Proyecto	Servicio	Descripción
Nova	Computación	Gestiona el ciclo de vida de las máquinas virtuales incluyendo su creación, la selección del nodo de computación donde se ejecutarán y el desmantelamiento de las mismas bajo demanda
Glance	Imagen	Almacena y gestiona imágenes de los discos de las máquinas virtuales. Nova hace uso de estas imágenes durante el proceso de creación de las máquinas virtuales
Horizon	Interfaz web	Portal de auto-servicio web que permite a administradores y clientes interactuar con los servicios subyacentes de OpenStack.
Trove	Base de datos	Proporciona Bases de Datos como Servicio de

		<p>manera escalable y fiable. Las bases de datos ofertadas son tanto relacionales como no relacionales</p>
Swift	Almacenamiento de objetos	<p>Almacena y gestiona datos arbitrarios no estructurados en forma de objetos los cuales son accedidos mediante una API RESTful basada en HTTP. Es altamente tolerante a fallos a partir de la implementación de mecanismos de replicación de datos y arquitectura escalable. Su implementación no consiste en un servidor de ficheros compartidos con directorios montados, sino que escribe objetos y ficheros en múltiples dispositivos, garantizando que los datos estén replicados a lo largo del clúster de servidores.</p>
Cinder	Almacenamiento de Bloques	<p>Proporciona almacenamiento permanente a las máquinas virtuales en ejecución. Presenta una arquitectura modular con una extensa variedad de drivers que permiten la creación de volúmenes de almacenamiento en bloques los cuales son acoplados a las máquinas virtuales</p>
Keystone	Identidad	<p>Le proporciona un servicio de autenticación y autorización a otros proyectos de OpenStack. Provee un catálogo con los puntos de acceso o "endpoints" de cada uno de estos proyectos.</p>
Neutron	Red	<p>Proporciona Conectividad de Red como Servicio a otros proyectos de OpenStack como por ejemplo Nova, el cual utiliza la API de Neutron para solicitar la conexión de las máquinas virtuales a un segmento de red determinado. Permite la creación de redes, subredes, enrutadores, cortafuegos, balanceadores de carga y redes</p>

privadas virtuales. Cuenta con una arquitectura modular que se integra con múltiples tecnologías de proveedores de red externos.

4.2.4 Modelos de servicios de cloud computing

Cuando nos referimos a cloud computing es un término demasiado amplio y utilizado de muchas formas pero aun así existe aprobación general correspondiente a los tres modelos principales que describe en la siguiente figura



Figura 3. Modelos de servicios de cloud computing

4.2.5 Software como servicio (SaaS)

El modelo de Software como servicios consiste en que una aplicación es proporcionada por un proveedor de servicios a través de Internet y puede ser accedida por diferentes usuarios, generalmente a través de un navegador web.

El cliente usa la aplicación pero no gestiona ni controla la infraestructura subyacente sobre la que esta se ejecuta, como hardware, almacenamiento o red. La empresa que entrega el servicio de software es la encargada de su mantenimiento y entregar el soporte de la aplicación que utilizará el cliente, de acuerdo a las condiciones contractuales que definen la prestación de servicios.

4.2.6 Plataforma como servicio (PaaS)

Es un modelo a través del cual se ofrece todo lo necesario para dar soporte al ciclo de vida aplicaciones, ya sea en la etapa de construcción como en la de puesta en marcha. Para la utilización de este servicio no es necesario descargar ningún tipo de software en los equipos de los desarrolladores, ya que se entregan todas las herramientas necesarias para llevarlos a cabo como una solución integral vía web. El cliente no controla ni gestiona la infraestructura subyacente que incluye la red, servidores, sistemas operativos o almacenamiento, pero tiene control sobre las aplicaciones desplegadas y la posibilidad de controlar las configuraciones de entorno del hosting de aplicaciones

4.2.7 Infraestructura como servicio

Proporciona al cliente una infraestructura de computación como un servicio, usando principalmente la virtualización. El cliente compra recursos a un proveedor externo, para hosting, capacidad de cómputo, mantenimiento y gestión de redes. El cliente puede abastecerse de capacidad de procesamiento, almacenamiento, componentes de red y otros recursos computacionales fundamentales de forma que puede desplegar y controlar software arbitrario, que puede incluir sistemas operativos y aplicaciones. Estos recursos se implementan normalmente mediante una plataforma de virtualización.

Tipo de Servicio	Ejemplos
Procesamiento	Amazon Elastic Compute Cloud (Amazon EC2) (Amazon Web Services, LLC), Sun Network.com (Sun Grid) (SUN Microsystems, Inc.), ElasticHost (ElasticHosts Ltd.), Eucalyptus (Nurmi, y otros, 2009), Nimbus (Alliance), OpenNebula (Grupo de ArquitecturaDistribuida), Enomaly (Enomaly, Inc.).
Distribución de contenido a través de servidores virtuales	Cisco WebEx Weboffice (Cisco Systems, Inc.), Google Docs (Google), Google Talk (Google), IBM BlueHouse (IBM, Corp.), Microsoft Exchange Online (Microsoft), RightNow (RightNow Technologies, Inc.), Gmail (Google), Microsoft Hotmail (Microsoft Hotmail), Yahoo! Mail (Yahoo! Inc.).
Almacenamiento	Amazon Flexible Payments Service (Amazon FPS) (Amazon Web Services, LLC), Amazon DevPay (Amazon Web Services, LLC).
Administración de sistemas	Elastra (Elastra Corporation), Engine Yard (Engine Yard, Inc.), FlexiScalable (XCalibre Communications), Grid Layer (Layered Technologies, Inc.), Joyent (Joyent, Inc.), Mosso (Rackspace, US Inc.), Savvis Virtual Intelligent Hosting (Savvis, Inc.).
Administración de alojamiento	Digital Realty Trust (Digital Realty Trust, Inc.), GoDaddy.com (GoDaddy.com, Inc.), Layered Technology (Layered Technologies, Inc.).
Alojamiento autónomo	Rackspace (Rackspace, US Inc.), Savvis Virtual Intelligent Hosting (Savvis, Inc.), Terremark Worldwide (Terremark Worldwide), FlexiScalable (XCalibre Communications), 1&1 Internet (1&1 Internet, Inc.).
VLAN (Virtual Local Area Network)	CohesiveFT (Cohesive Flexible Technologies, Corp.).

Figura 4. Tipos de servicios de Openstack

4.3 MARCO CONCEPTUAL

- CLOUD COMPUTING: concepto conocido también bajo los términos informática en la nube, nube de cómputo o computación en la nube es un paradigma que permite ofrecer servicios de computación a través de Internet.
- FREEMIUM: es un modelo de negocios que funciona ofreciendo servicios básicos gratuitos, mientras se cobra por otros más avanzados o especiales

- GRID COMPUTING: es una tecnología innovadora que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están sujetos a un control centralizado.
- HOSTING: (El alojamiento web) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web
- HYPERVISOR: es una plataforma que permite aplicar diversas técnicas de control de virtualización para utilizar, al mismo tiempo, diferentes sistemas operativos (sin modificar o modificados en el caso de para virtualización) en una misma computadora.
- LAAS: (infraestructura como servicio, también llamado en algunos casos hardware as a service, HaaS) se encuentra en la capa inferior y es un medio de entregar almacenamiento básico y capacidades de cómputo como servicios estandarizados en la red.
- MIDDLEWARE: es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, software, redes, hardware y/o sistemas operativos.
- NO REPUDIO: es un servicio de seguridad que permite probar la participación de las partes en una comunicación
- PAAS: (plataforma como servicio) es la encapsulación de una abstracción de un ambiente de desarrollo y el empaquetamiento de una carga de servicios
- PLUGINS: un software que permite cambiar, mejorar o agregar funcionalidades en SPIP
- PROTOCOLO SOAP: Se trata de un protocolo que te permite la comunicación entre aplicaciones a través de mensajes por medio de Internet.
- ROOTKIT: es una herramienta o un grupo de ellas, que tiene como finalidad esconderse a sí misma y esconder otros programas, procesos, archivos, directorios, claves de registro, y puertos que permiten al intruso mantener el acceso a un sistema para remotamente comandar acciones o extraer información sensible.RTOs.
- SLA: Hace referencia a los acuerdos de los niveles de sersjuicio

- SSA: (Software como Servicio) es un modelo de distribución de software donde el software y los datos que maneja se alojan en servidores de la compañía de tecnologías de información y comunicación (TIC) y se accede con un navegador web a través de internet
- WORST CASE SCENARIO: es una situación en la que todo lo que podía salir mal, va mal
- WORKFLOWS: consiste en una secuencia de pasos conectados
- WRAPPING: una variable que excede su valor máximo en la informática
- XEN: es un monitor de máquina virtual que proporciona servicios que permiten a múltiples sistemas operativos para ejecutar en el mismo hardware concurrentemente.
- WORKFLOWS: consiste en una secuencia de pasos conectados. Se trata de una representación de una secuencia de operaciones, declaró que el trabajo de una persona, un grupo de personas, una organización de personal, o uno o varios mecanismos simples o complejos.

4.4 MARCO LEGAL

Colombia es uno de los países que cuenta a la fecha con un conjunto de leyes que facilita el desarrollo del Cloud Computing en Colombia. A continuación, se describen estas leyes y se enuncian los temas más importantes que éstas mencionan:

- Ley 1273 de 2009: Por medio de la Ley 1273 de 2009 se modifica el Código Penal, se crea un nuevo bien jurídico tutelado denominado de la protección de la información y de los datos y se preservan integralmente los sistemas que utilicen las tecnologías de la información y las comunicaciones, entre otras disposiciones. Esta ley protege a los sistemas de Información de los atentados contra la confidencialidad, la integridad y la disponibilidad de los datos y de los sistemas informáticos. La Ley penaliza, entre estos atentados, el acceso abusivo a los sistemas informáticos, la interceptación de datos, la ejecución de daños informáticos, el uso de software malicioso, la violación de los datos personales,

la suplantación de sitios web para capturar datos personales, el hurto por medios informáticos y semejantes y la Transferencia no consentida de activos.

- Ley 1221 de 2008 – Ley de Teletrabajo: Por medio de esta ley, se establecen normas para promover y regular el Teletrabajo y se provee un marco de seguridad jurídica. Esta ley define el teletrabajo en sus distintas formas, establece una política pública de fomento al teletrabajo y una red nacional de fomento al teletrabajo. De igual manera, menciona que el Gobierno Nacional pondrá en funcionamiento un sistema de inspección, vigilancia y control para garantizar el cumplimiento de la legislación laboral en el marco del teletrabajo y se proveen las garantías laborales, sindicales y de seguridad social para los teletrabajadores.

- Ley 1266 de 2008: Declarado Exequible mediante Sentencia C- 1011 del 16 de octubre de 2008., dictan las disposiciones generales del hábeas data y regula el manejo de la información contenida en bases de datos personales, en especial la financiera, crediticia, comercial, de servicios y la proveniente de terceros países, entre otros.

Esta ley tiene por objeto desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bancos de datos, y los demás derechos, libertades y garantías constitucionales relacionadas con la recolección, tratamiento y circulación de datos personales a que se refiere el artículo 15 de la Constitución Política, así como el derecho a la información establecido en el artículo 20 de la Constitución Política, particularmente en relación con la información financiera y crediticia, comercial, de servicios y la proveniente de terceros países.

Además, establece los principios de la administración de datos: Principio de veracidad, de finalidad, de circulación restringida, de temporalidad de la información, de interpretación integral de derechos constitucionales, de

seguridad y de confidencialidad; establece los derechos de los titulares de la información, los deberes de los operadores, las fuentes y los usuarios de información, la vigilancia de los destinatarios de la ley.

- Ley 1341 de 2009: Por medio de esta ley, se definen principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones –TIC–, se crea la Agencia Nacional de Espectro y se dictan otras disposiciones. Esta ley tiene por objeto determinar el marco general para la formulación de las políticas públicas que regirán el sector de las Tecnologías de la Información y las Comunicaciones, su ordenamiento general, el régimen de competencia, la protección al usuario, así como lo concerniente a la cobertura, la calidad del servicio, la promoción de la inversión en el sector y el desarrollo de estas tecnologías.

El uso eficiente de las redes y del espectro radioeléctrico, así como las potestades del Estado en relación con la planeación, la gestión, la administración adecuada y eficiente de los recursos, regulación, control y vigilancia del mismo y facilitando el libre acceso y sin discriminación de los habitantes del territorio nacional a la Sociedad de la Información. Además, define los siguientes principios orientadores: prioridad al acceso y uso de las tecnologías de la información y las comunicaciones, libre competencia, uso eficiente de la infraestructura y de los recursos escasos, protección de los derechos de los usuarios, promoción de la inversión, neutralidad tecnológica, el derecho a la comunicación, la información y la educación y los servicios básicos de las tic y la masificación del gobierno en línea.

- Resolución CRC 2258 de 2009: Teniendo en cuenta que la protección del ciberespacio es un factor de trascendente importancia para preservar la seguridad de la nación y su economía, la CRC comprendió la necesidad de estudiar los cambios que se han generado sobre estos asuntos, y analizar alternativas de modificación o creación de reglas para contribuir desde la

perspectiva regulatoria interna. Con el fin de cumplir con lo descrito anteriormente se consideraron diferentes tendencias mundiales sobre la materia, así como el estado actual de redes de telecomunicaciones en el país, y los servicios y mecanismos de seguridad que son implementados en las mismas.

5. DISEÑO DEL PROYECTO

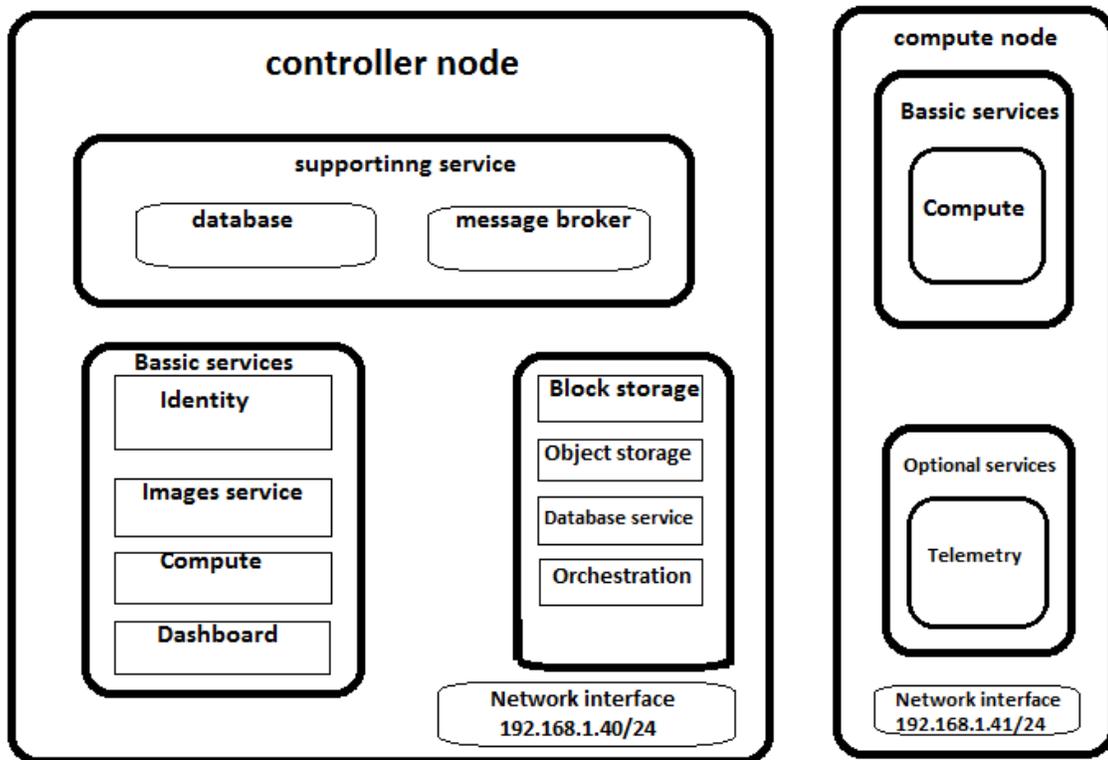


Figura 5. Diseño Lógico de nuestro sistema de dos nodos

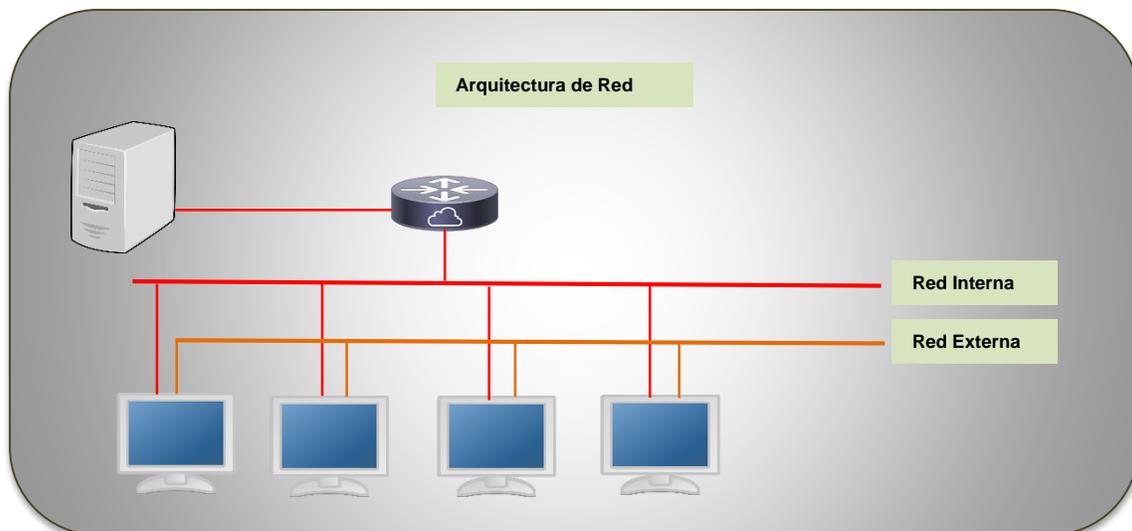


Figura 6. Diseño físico de nuestro sistema

5.1 LINEA DE INVESTIGACION

La línea de investigación que se usa para este proyecto es la de redes.

5.2 ALCANCES

5.2.1 Alcance del proyecto

Nuestro proyecto contempla la instalación de un prototipo para la ejecución de la infraestructura computacional en la nube con OpenStack. Se implementa en 1 servidor situado en la Universidad del Sinú con el fin de interactuar con recursos cloud (procesamiento, almacenamiento e instancias) de forma autónoma.

CRONOGRAMA

Actividad	Julio 2016	Agosto 2016	Sept. 2016	Octubre 2016	Febrero 2017	Septiembre 2017	Octubre 2017	Octubre 2017
Recopilación de información								
Análisis de requerimientos								
Desarrollo de objetivos								
Fase de instalación								
Fase de implementación								
Fase de prueba								

Figura 7. Cronograma de actividades

5.2.2 Limitaciones del proyecto

La arquitectura para OpenStack es un prototipo All IN One, el cual compila todos los servicios con recursos de un solo servidor.

De acuerdo a las características del servidor tenemos limitada la creación de instancias, a un máximo de 16.

El tiempo de instalación de sistemas operativos se ve levemente afectada debido a que nuestro servidor actúa como All in One.

5.3 PRESUPUESTO

Todas las tecnologías usadas en el proyecto están licenciadas bajo una licencia Open Source gratuitas (Apache, GPL, etc). Los sistemas operativos utilizados han sido Linux en el sistema host y Centos 7 en los nodos, sistemas que son de Código Abierto y gratuitos. Es decir, el software no añade ningún costo al proyecto a nivel de Software.

6. CAPÍTULOS

6.1 ESTABLECER LOS REQUERIMIENTOS MINIMOS NECESARIOS PARA EL SERVICIO OPENSTACK

Requerimientos mínimos de hardware:

1. 2 equipos donde uno tomara rol de controlador y el otro nodo1
2. Controlador
 - 2.1 Procesador de 2 núcleos con soporte de virtualización
 - 2.2 4 GB de RAM
 - 2.3 Disco Duro capacidad de 40GB
 - 2.4 2 interfaces de red Ethernet
3. Compute
 - 3.1 Procesador con 2 o 4 núcleos con soporte de virtualización
 - 3.2 Memoria RAM de 8 GB
 - 3.3 2 interfaces de red Ethernet
 - 3.4 Disco duro capacidad 100GB

Para la implementación del proyecto utilizamos los siguientes equipos

Servidor Dell R710 Tipo Rack	
Memoria RAM	16 GB
Procesador	2 procesadores de 4 núcleos
Disco duro	1 Terabyte
Interfaces de Red	2

Instalación de OpenStack con RDO y la herramienta PackStack en Centos 7

¿QUE es RDO OpenStack?

RDO es una distribución mantenida por la comunidad de usuarios que permite ejecutar OpenStack en sistemas Red Hat y derivados (fedora y CentOS). RDO utiliza la herramienta Packstack, que mediante el uso de manifiestos de puppet y un simple fichero de configuración es capaz de instalar OpenStack en uno o varios nodos.

PackStack es la utilidad creada por los desarrolladores de esta comunidad que con ayuda de Puppet y Python nos permite hacer la instalación basada en pruebas de conceptos de todos los componentes que conforman OpenStack.

Centos 7 : Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. Se define como robusto, estable y fácil de instalar y utilizar

Pasos previos

Tipo de instalación de servicios OpenStack: la instalación se realiza de modo all in one, debido a que tenemos acceso a 1 equipo.

Dirección IP del servidor:

IP	192.168.8.40
Mascara	255.255.248.0
Gateway	192.168.8.1

Cambiando las Variables de Entorno

Cambiar o agregar las siguientes líneas en el archivo de variables de entorno que se encuentra en la ruta: vi /etc/environment

```
LANG=en_US.utf-8
LC_ALL=en_US.utf-8
```

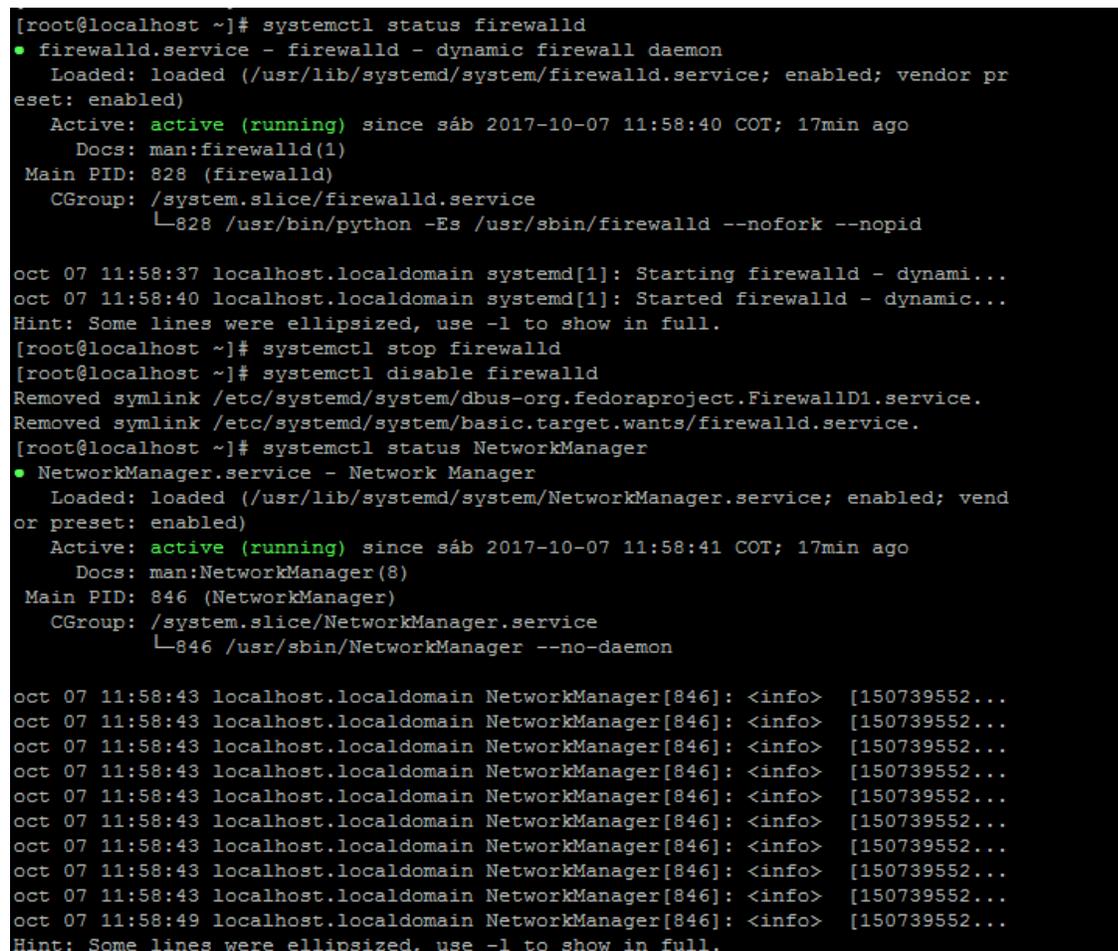


```
LANG=en_US.utf-8
LC_ALL=en_US.utf-8
```

Desactivar los servicios de Firewalld y Network Manager:

Es necesario desactivar los servicios de Firewalld y Network Manager con los siguientes comandos.

```
# systemctl disable firewalld
# systemctl stop firewalld
# systemctl disable NetworkManager
# systemctl stop NetworkManager
```



```
[root@localhost ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since sáb 2017-10-07 11:58:40 COT; 17min ago
     Docs: man:firewalld(1)
    Main PID: 828 (firewalld)
    CGroup: /system.slice/firewalld.service
           └─828 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid

oct 07 11:58:37 localhost.localdomain systemd[1]: Starting firewalld - dynamic...
oct 07 11:58:40 localhost.localdomain systemd[1]: Started firewalld - dynamic...
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl disable firewalld
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
Removed symlink /etc/systemd/system/basic.target.wants/firewalld.service.
[root@localhost ~]# systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since sáb 2017-10-07 11:58:41 COT; 17min ago
     Docs: man:NetworkManager(8)
    Main PID: 846 (NetworkManager)
    CGroup: /system.slice/NetworkManager.service
           └─846 /usr/sbin/NetworkManager --no-daemon

oct 07 11:58:43 localhost.localdomain NetworkManager[846]: <info> [150739552...
oct 07 11:58:49 localhost.localdomain NetworkManager[846]: <info> [150739552...
Hint: Some lines were ellipsized, use -l to show in full.
```

2. Habilitar los servicios de Red

```
# systemctl enable network
# systemctl start network
```

```
[root@localhost ~]# systemctl enable network
network.service is not a native service, redirecting to /sbin/chkconfig.
Executing /sbin/chkconfig network on
[root@localhost ~]# systemctl start network
[root@localhost ~]# ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:26:b9:36:50:76 brd ff:ff:ff:ff:ff:ff
    inet 192.168.8.40/21 brd 192.168.15.255 scope global em1
```

4. Desactivar característica de seguridad selinux en el archivo /etc/selinux/config, cambiando los valores. Al finalizar es necesario reiniciar el sistema.

```
#Editar la línea:
SELINUX=enforcing
```

```
#Cambiar por:
SELINUX=disabled
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected,
#   minimum - Modification of targeted policy. Only selected processes are pro
ected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

5. Instalamos los repositorios de OpenStack y al finalizar la instalación se debe ejecutar yum update -y para actualizar los repositorios.
sudo yum install -y centos-release-openstack-ocata

```
[root@localhost ~]# yum install -y centos-release-openstack-ocata
Failed to set locale, defaulting to C
Loaded plugins: fastestmirror
base                                     | 3.6 kB  00:00:00
extras                                  | 3.4 kB  00:00:00
updates                                 | 3.4 kB  00:00:00
(1/4): base/7/x86_64/group.gz          | 155 kB  00:00:00
(2/4): extras/7/x86_64/primary.db      | 139 kB  00:00:00
(3/4): updates/7/x86_64/primary.db    | 3.8 MB  00:00:04
(4/4): base/7/x86_64/primary.db       | 5.6 MB  00:00:05
Determining fastest mirrors
 * base: mirror.dgn.net.tr
 * extras: ftp.itu.edu.tr
 * updates: ftp.itu.edu.tr
```

6. Instalar los repositorios de PackStack

```
sudo yum install -y openstack-packstack
```

```
[root@localhost ~]# yum install -y openstack-packstack
Loaded plugins: fastestmirror
centos-openstack-ocata
Loading mirror speeds from cached hostfile
 * base: ftp.linux.org.tr
 * extras: mirror.dgn.net.tr
 * updates: ftp.linux.org.tr
Resolving Dependencies
--> Running transaction check
  > Packages openstack-packstack-ocata-9.1.0-0.0.1-0.el7.us1
```

Servicios De Redes Neutrón

1. Instalar los componentes necesarios de openstack, que en este caso son los siguientes:

```
packstack --allinone --provision-demo=n --os-neutron-ovs-bridge-mappings=extnet:br-ex --os-neutronml2-type-drivers=vxlan,flat
```

--provision-demo=n : descartamos que la instalación nos incluya los cuentas demo.

--os-neutron-ovs-bridge-mappings=extnet:br-ex : con Neutron le indicamos que creamos una bridge mappings llamada br-ex y que permitirá el tráfico desde la red física.

--os-neutron-ml2-type-drivers=vxlan,flat : le indicamos los tipos de driver a utilizar del plugin Modular Layer 2 que nos permitirá crear redes virtuales.

Al finalizar la instalación de estos paquetes se obtiene el acceso al Dashboard de OpenStack.

```
Preparing Nova VNC Proxy entries [ DONE ]
Preparing OpenStack Network-related Nova entries [ DONE ]
Preparing Nova Common entries [ DONE ]
Preparing Neutron LBaaS Agent entries [ DONE ]
Preparing Neutron API entries [ DONE ]
Preparing Neutron L3 entries [ DONE ]
Preparing Neutron L2 Agent entries [ DONE ]
Preparing Neutron DHCP Agent entries [ DONE ]
Preparing Neutron Metering Agent entries [ DONE ]
Checking if NetworkManager is enabled and running [ DONE ]
Preparing OpenStack Client entries [ DONE ]
Preparing Horizon entries [ DONE ]
Preparing Swift builder entries [ DONE ]
Preparing Swift proxy entries [ DONE ]
Preparing Swift storage entries [ DONE ]
Preparing Gnocchi entries [ DONE ]
Preparing MongoDB entries [ DONE ]
Preparing Redis entries [ DONE ]
Preparing Ceilometer entries [ DONE ]
Preparing Aodh entries [ DONE ]
Preparing Puppet manifests [ DONE ]
Copying Puppet modules and manifests [ DONE ]
Applying 192.168.8.40_controller.pp [ DONE ]
192.168.8.40_controller.pp: [ DONE ]
Applying 192.168.8.40_network.pp [ DONE ]
192.168.8.40_network.pp: [ DONE ]
Applying 192.168.8.40_compute.pp [ DONE ]
192.168.8.40_compute.pp: [ DONE ]
Applying Puppet manifests [ DONE ]
Finalizing [ DONE ]

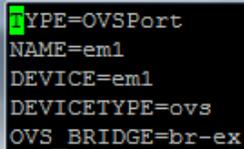
**** Installation completed successfully ****

Additional information:
 * A new answerfile was created in: /root/packstack-answers-20171007-124400.txt
 * Time synchronization installation was skipped. Please note that unsynchronized time on server instances might be problem for some OpenStack components.
 * File /root/keystonerc admin has been created on OpenStack client host 192.168.8.40. To use the command line tools you need to source the file.
 * To access the OpenStack Dashboard browse to http://192.168.8.40/dashboard .
Please, find your login credentials stored in the keystonerc admin in your home directory.
 * Because of the kernel update the host 192.168.8.40 requires reboot.
 * The installation log file is available at: /var/tmp/packstack/20171007-124359-mbogMR/openstack-setup.log
 * The generated manifests are available at: /var/tmp/packstack/20171007-124359-mbogMR/manifests
You have new mail in /var/spool/mail/root
[root@localhost ~]#
```

2. Configurar la interfaz de red ifcfg-m1 en modo bridge que mapea la interfaz virtual br-ex.

```
# vi /etc/sysconfig/network-scripts/ifcfg-m1
```

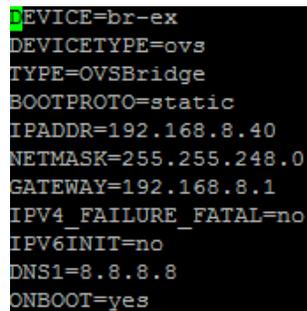
```
TYPE=OVSPort  
NAME=enp0s3  
DEVICE=enp0s3  
DEVICETYPE=ovs  
OVS_BRIDGE=br-ex  
ONBOOT=yes
```



```
TYPE=OVSPort  
NAME=enp0s3  
DEVICE=enp0s3  
DEVICETYPE=ovs  
OVS_BRIDGE=br-ex
```

3. Configurar la interface de puente externa br-ex.

```
# vi /etc/sysconfig/network-scripts/ifcfg-br-ex  
DEVICE=br-ex  
DEVICETYPE=ovs  
TYPE=OVSBridge  
BOOTPROTO=static  
IPADDR=<your_IP>  
NETMASK=<your_mask>  
GATEWAY=<your_gateway_IP>  
IPV4_FAILURE_FATAL=no  
IPV6INIT=no  
DNS1=<DNS_Server_IP>  
ONBOOT=yes
```



```
DEVICE=br-ex  
DEVICETYPE=ovs  
TYPE=OVSBridge  
BOOTPROTO=static  
IPADDR=192.168.8.40  
NETMASK=255.255.248.0  
GATEWAY=192.168.8.1  
IPV4_FAILURE_FATAL=no  
IPV6INIT=no  
DNS1=8.8.8.8  
ONBOOT=yes
```

4. Reiniciar el servicio de redes.

service network restart

```
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-em1
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-
ifcfg-em1 ifcfg-em2 ifcfg-em3 ifcfg-em4 ifcfg-lo
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-br-ex
You have new mail in /var/spool/mail/root
[root@localhost ~]# service network restart
Restarting network (via systemctl): [ OK ]
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-em1
[root@localhost ~]# vi /etc/sysconfig/network-scripts/ifcfg-br-ex
[root@localhost ~]#
```

5. este comando te proporciona los privilegios de administrador de openstack.

source keystone_admin

```
anaconda-ks.cfg keystone_admin packstack-answers-20171007-124400.txt
[root@localhost ~]# source keystone_admin
[root@localhost ~(keystone admin)]#
```

6. Crear red de proveedores la cual permite la comunicación entre las instancias.

neutron net-create external_network --provider:network_type flat --provider:physical_network \
extnet --router:external

```
[root@localhost ~(keystone_admin)]# neutron net-create external_network --provid
er:network_type flat --provider:physical_network \
> extnet --router:externa
neutron CLI is deprecated and will be removed in the future. Use openstack CLI i
nstead.
Atributo(s) no reconocido(s) 'router:externa'
Neutron server returns request ids: ['req-d37dc527-13ca-4d12-82b9-e1c174d852c3']
[root@localhost ~(keystone_admin)]# neutron net-create external_network --provid
er:network_type flat --provider:physical_network extnet --router:external
neutron CLI is deprecated and will be removed in the future. Use openstack CLI i
nstead.
Created a new network:
-----
| Field                | Value                |
|-----|-----|
| admin_state_up      | True                 |
| availability_zone_hints |                     |
| availability_zones   |                     |
| created_at           | 2017-10-07T18:31:51Z |
| description          |                     |
| id                   | 3091e0a3-d9e8-4d60-ad15-c77a7235abf5 |
| ipv4_address_scope   |                     |
| ipv6_address_scope   |                     |
| is_default           | False                |
| mtu                  | 1500                 |
| name                 | external_network    |
| project_id           | dc931e9fcf754080904ac86db8d11ef3 |
| provider:network_type | flat                 |
| provider:physical_network | extnet               |
| provider:segmentation_id |                     |
| revision_number      | 3                    |
| router:external      | True                 |
| shared               | False                |
| status               | ACTIVE               |
| subnets             |                     |
| tags                 |                     |
| tenant_id            | dc931e9fcf754080904ac86db8d11ef3 |
| updated_at           | 2017-10-07T18:31:51Z |
|-----|-----|
```

7. Configuración de subred Este comando crea la subred conectada a su red de proveedores la cual permite la comunicación entre la LAN externa y las instancias.

neutron subnet-create --name public_subnet --enable_dhcp=False --allocation-pool
start=192.168.8.42,end=192.168.40.46 --gateway=192.168.40.1 external_network 192.168.8.0/21

```
[root@localhost ~ (keystone_admin)]# neutron subnet-create --name public_subnet --enable_dhcp=False --allocation-pool start=192.168.8.42,end=192.168.8.46 --gateway=192.168.40.1 external_network 192.168.8.0/21
neutron CLI is deprecated and will be removed in the future. Use openstack CLI instead.
Created a new subnet:
+-----+
| Field | Value |
+-----+
| allocation_pools | {"start": "192.168.8.42", "end": "192.168.8.46"} |
| cidr | 192.168.8.0/21 |
| created_at | 2017-10-07T18:44:38Z |
| description | |
| dns_nameservers | |
| enable_dhcp | False |
| gateway_ip | 192.168.40.1 |
| host_routes | |
| id | e5594198-445a-4e98-83ad-0c8e3d0e62c5 |
| ip_version | 4 |
| ipv6_address_mode | |
| ipv6_ra_mode | |
| name | public_subnet |
| network_id | 3091e0a3-d9e8-4d60-ad15-c77a7235abf5 |
| project_id | dc931e9fcf754080904ac86db8d11ef3 |
| revision_number | 2 |
| service_types | |
| subnetpool_id | |
| tags | |
| tenant_id | dc931e9fcf754080904ac86db8d11ef3 |
| updated_at | 2017-10-07T18:44:38Z |
+-----+
```

Instalación De Rally

Rally automatiza y unifica la implementación de varios nodos de OpenStack, la verificación en la nube, la evaluación comparativa y la creación de perfiles. Rally lo hace de una manera conectable, lo que permite verificar si OpenStack funcionará bien.

1. Instalar la herramienta rally.
Yum install -y openstack-rally

```
[root@localhost ~ (keystone_admin)]# yum install -y openstack-rally
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: linorg.usp.br
 * extras: linorg.usp.br
 * updates: centos.xpg.com.br
Resolving Dependencies
--> Running transaction check
---> Package openstack-rally.noarch 0:0.9.1-1.el7 will be installed
--> Processing Dependency: python-muranoclient >= 0.8.2 for package: openstack-rally-0.9.1-1.el7.noarch
--> Processing Dependency: python-manilaclient >= 1.10.0 for package: openstack-rally-0.9.1-1.el7.noarch
--> Processing Dependency: python-magnumclient >= 2.0.0 for package: openstack-rally-0.9.1-1.el7.noarch
--> Processing Dependency: python-zaqarclient for package: openstack-rally-0.9.1-1.el7.noarch
--> Processing Dependency: python-subunit for package: openstack-rally-0.9.1-1.el7.noarch
--> Processing Dependency: python-sphinx for package: openstack-rally-0.9.1-1.el7.noarch
```

2. Crear la base de datos
Rally-manage db recreate

```
[root@localhost ~ (keystone_admin)]# rally-manage db recreate
[root@localhost ~ (keystone_admin)]#
```

3. Registrar una implementación de OpenStack en Rally.

rally create --fromenv --name = existing

```
[root@localhost ~(keystone_admin)]# rally deployment create --fromenv --name=existing
usage: rally [-h] [--config-dir DIR] [--config-file PATH] [--debug]
             [--log-config-append PATH] [--log-date-format DATE_FORMAT]
             [--log-dir LOG DIR] [--log-file PATH] [--nodebug]
             [--norally-debug] [--nouse-syslog] [--noverbose]
             [--nowatch-log-file] [--plugin-paths PLUGIN_PATHS]
             [--rally-debug] [--syslog-log-facility SYSLOG_LOG_FACILITY]
             [--use-syslog] [--verbose] [--version] [--watch-log-file]
             {version,hash-completion,task,plugin,verify,deployment} ...
rally: error: unrecognized arguments: --fromenv
[root@localhost ~(keystone_admin)]# rally deployment create --fromenv --name=existing
2017-10-07 13:51:01.861 29751 INFO rally.deployment.engine [-] Deployment 5d61e30
8-b16e-47a5-94ed-33cdb9fc8358 | Starting: OpenStack cloud deployment.
2017-10-07 13:51:01.901 29751 INFO rally.deployment.engine [-] Deployment 5d61e30
8-b16e-47a5-94ed-33cdb9fc8358 | Completed: OpenStack cloud deployment.
-----+-----+
| uuid                | created_at          | name      |
+-----+-----+-----+-----+
| 5d61e308-b16e-47a5-94ed-33cdb9fc8358 | 2017-10-07 18:51:01.837640 | existing |
| deploy->finished | |
+-----+-----+-----+-----+
Using deployment: 5d61e308-b16e-47a5-94ed-33cdb9fc8358
~/rally/openrc was updated

HINTS:

* To use standard OpenStack clients, set up your env by running:
  source ~/rally/openrc
OpenStack clients are now configured, e.g run:
  openstack image list
[root@localhost ~(keystone_admin)]#
```

4. Finalmente, el comando de verificación de implementación le permite verificar que su implementación actual es saludable y está lista para ser evaluada.

#rally deployment check

```
[root@localhost ~(keystone_admin)]# rally deployment check
keystone endpoints are valid and following services are available:
-----+-----+
| services | type      | status |
+-----+-----+-----+
| __unknown__ | alarming | Available |
| __unknown__ | placement | Available |
| __unknown__ | volumev2 | Available |
| __unknown__ | volumev3 | Available |
| ceilometer | metering | Available |
| cinder      | volume   | Available |
| glance     | image    | Available |
| gnocchi    | metric   | Available |
| keystone   | identity | Available |
| neutron    | network  | Available |
| nova       | compute  | Available |
| swift      | object-store | Available |
+-----+-----+-----+
NOTE: '__unknown__' service name means that Keystone service catalog doesn't return name for this service and Rally can not identify service by its type. BUT you still can use such services with api_versions context, specifying type of service (execute 'rally plugin show api_versions' for more details).
```

Proyecto >

Administrador v

Sistema v

Vista general

Hipervisores

Agregados de host

Instancias

Volúmenes

Sabores

Imágenes

Redes

Routers

IPs flotantes

Predeterminados

Definiciones de los metadatos

Información del Sistema

Identity >

Administrador / Sistema / Información del Sistema

Información del Sistema

Servicios Servicios de computación Servicios de almacenamiento de bloques Agentes de red

Filtrar

Mostrando 5 ítem

Tipo	Nombre	Host	Estado	Estado	Actualizada por última vez	Acciones
Open vSwitch agent	neutron-openvswitch-agent	localhost.localdomain	Habilitado	Arriba	0 minutos	
DHCP agent	neutron-dhcp-agent	localhost.localdomain	Habilitado	Arriba	0 minutos	
Metering agent	neutron-metering-agent	localhost.localdomain	Habilitado	Arriba	0 minutos	
Metadata agent	neutron-metadata-agent	localhost.localdomain	Habilitado	Arriba	0 minutos	
L3 agent	neutron-l3-agent	localhost.localdomain	Habilitado	Arriba	0 minutos	Ver enrutadores

Mostrando 5 artículos

Versión: 11.0.3

6.2 CONCLUSION

Como resultado de la implementación de las máquinas virtuales en OpenStack, se pudo profundizar en los sistemas de nubes y obtener los conocimientos en entornos de desarrollo de Computación en la nube tipos de servicios y funcionalidades para cada proyecto.

De acuerdo a lo anterior se concluye que se pudo realizar la implementación del prototipo de sistemas de virtualización de OpenStack, en el cual se pudo realizar la instalación y configuración de 1 instancia de Ubuntu debido a las limitaciones de hardware del servidor brindado por la universidad del Sinú para realizar la implementación.

La solución de Openstack fue realizada sobre 1 servidor y bajo el sistema operativo Centos 7 el cual su rendimiento se vio seriamente afectado por los servicios instalados lo que provoco que solo el 16% de la memoria Ram se usara para la creación de las Instancias.

7. RECOMENDACIONES

Para la implementación de un sistema de virtualización con OpenStack en la Universidad del Sinú, se deben tener en cuenta las siguientes variables las cuales van a permitir el buen funcionamiento del sistema de Openstack y de las máquinas virtuales.

1. La memoria del nodo controlador no puede estar por debajo del 30% de la memoria RAM instalada esto afecta los servicios del sistema de OpenStack, este nodo es el encargado de manejar las base de datos y otros servicios para lo cual el número de cores del procesador debe ser proporcional al número de instancias.
2. Los nodos de computación la memoria Ram no puede estar por debajo del 20% y el número de instancias depende de la cantidad de memoria Ram instalada y el número de procesadores y cores instalados en cada servidor.

Servidor	Procesador	Memoria RAM para el sistema	Memoria Disponible para Instancias	Memoria Total	Numero de maquinas	Características
Controlador	10 cores	7,2 GB	16,8 GB	24 GB		
Nodo 1	8 Cores	6,4 GB	25,6 GB	32 GB	6	4 GB de Ram

8. REFERENCIAS BIBLIOGRÁFICAS

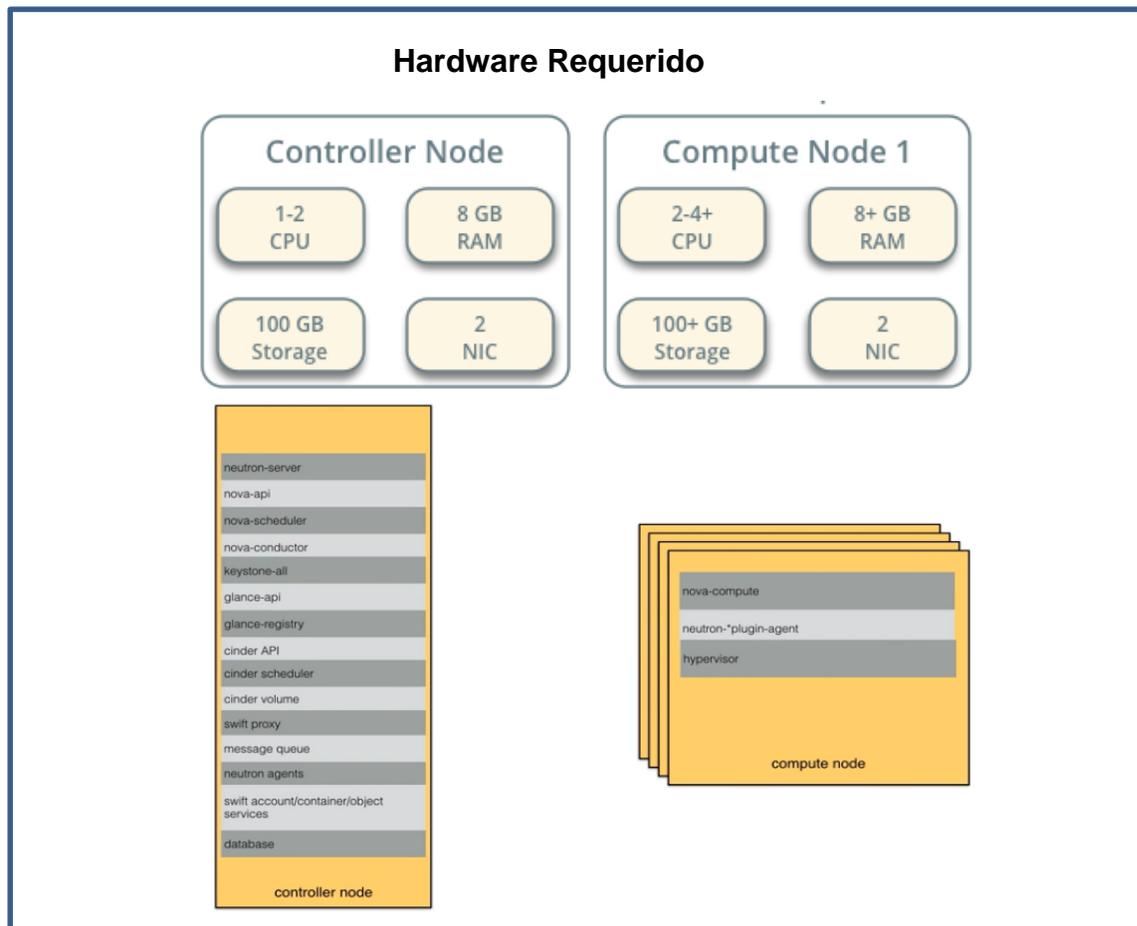
1. Que es OpenStack? - <http://www.openstack.org/>
2. OpenStack API Guide - <http://developer.openstack.org/api-guide/quick-start/>
3. OpenStack Installation Guide for Centos -<https://www.rdoproject.org/>.
4. OpenStack Fundación <http://www.openstack.org/foundation/>
5. <http://neuronadicta.com/post/como-instalar-openstack-en-centos-7-usando-packstack-de-rdo-project/>
6. <https://www.udemy.com/openstackEssential/>

Anexo 1

En la realización del Prototipo se puede observar que los servicios de OpenStack se implementaron en solo 1 equipo, esto genera una sobre carga del servidor y lentitud al momento de lanzar las instancias el valor máximo de instancias es de 16, este factor se ve altamente afectado por 3 componentes:

1. Memoria instalada en el servidor
2. Capacidad y Tipo de Procesador
3. Tecnología de Almacenamiento

EL número de instancias lanzadas depende del tipo de sistema operativo que se utilice en el caso de S.O Linux varia debido a que los requerimientos de Hardware son inferiores a lo que utilizan los S.O desarrollados por Windows, a continuación se describen las Características Mínimas de Hardware según la documentación de OpenStack la implementación básica requiere de 2 Equipos con las siguientes características.



Para realizar la implementación, puesta en marcha de Servidores de Producción para el sistema de virtualización es necesario contar con mínimo 2 equipos con las siguientes características que pueden garantizar un estimado de 30 instancias.

