Zaccardi, Gonzalo Emanuel

Despliegue de cloud computing en el ámbito universitario

Instituto: Ingeniería y Agronomía

Carrera: Ingeniería en Informática

Informe de final de carrera

Práctica Profesional Supervisada

Gonzalo Emanuel Zaccardi

Ingeniería en Informática

N° Legajo: 4036

Director: Diego Omar Encinas

29/08/2016

Universidad Nacional
ARTURO JAURETCHE

Universidad Nacional ARTURO JAURETCHE

Índice

Universidad Nacional ARTURO JAURETCHE

Agradecimientos

"A mi familia por estar conmigo durante toda la carrera y a cada una de las personas que me apoyaron a lo largo este camino, algunas de ellas hoy ausentes, siempre las tendré presentes."

1. Introducción

Cloud Computing [1] es un paradigma que está en constante crecimiento durante estos últimos años, cada vez más compañías y grupos de investigación trabajan en conjunto con el fin de explotar las oportunidades ofrecidas por el mismo. Dicho paradigma ofrece muchas ventajas, tales como el bajo costo de implementación, ya que no se necesitan computadoras de última tecnología debido a que éstas trabajan conjuntamente (Clustering) con la posibilidad de escalar horizontalmente[1] de manera sencilla. Además, hay software Open Source disponible para los nodos[2] en el clúster[3]como las infraestructuras Eucalyptus, OpenNebula, CloudStack u OpenStack[2] integradas con GNU/Linux y compatibles, por ejemplo, con Amazon WebServices.
Durante este trabajo, se ha utilizado la herramienta Fuel [3] desarrollada por Mirantis [4] con el fin de facilitar la instalación de la infraestructura debido a la complejidad y a los cambios constantes que sufre OpenStack diariamente.

1.1 Cloud Computing

Cloud computing es un paradigma de computación en franco avance y tiene muchos servicios que aún no han sido desarrollados. La capacidad de ofrecer servicios a usuarios de manera sencilla y económica propendió que empresas como Google, Facebook, Outlook, Yahoo, Hotmail, etc. utilizaran sus recursos en hardware y software para desarrollar "nubes" que almacenan todos los datos de los usuarios ofreciendo distintos tipos de servicios [5]:
IaaS (Infraestructure as a Service): Se contrata capacidad de proceso (CPU) y almacenamiento. En este caso sólo se paga por lo utilizado. Ejemplos: EC2 de Amazon y Azure de Microsoft.
PaaS (Platform as a Service): Se proporciona almacenamiento, un servidor de aplicaciones (en donde se ejecutarán las aplicaciones requeridas) y una base de datos. Ejemplo: Google App Engine.
SaaS (Software as a Service): Es una aplicación para el usuario final en donde se paga un alquiler por el uso de software. Muchas de estas aplicaciones son actualmente gratuitas. Ejemplos: Google Docs, Office 365, Dropbox.
HaaS (Hardware as a Service): Consiste en ofrecer hardware de manera que sea utilizada a través de internet logrando que el sistema utilice los recursos distribuidos geográficamente como si estuvieran localmente. Ofreciendo una mayor eficiencia de los mismos.
A su vez cada uno de ellos puede ser:

- *Público:* Una empresa ofrece IaaS a terceros, encargándose de toda la gestión del Cloud. El caso más conocido es Amazon Elastic Compute Cloud (EC2).
- *Privado:* Una organización configura sus propios recursos como IaaS para tener más flexibilidad y control total sobre sus recursos.
- *Híbrido:* Algunos servicios se gestionan en el cloud privado y otros se transfieren a uno público, normalmente utilizan una API común que permita una buena integración.

---

1 **La escalabilidad horizontal consiste en añadir más máquinas a la aplicación**, aumentando su número aunque no necesariamente su potencia.
2 **Nodos: Cada una de las computadoras en una red informática.**
3 **Clúster: conjuntos o conglomerados de computadoras unidos entre sí por una red informática.**

2. Objetivos

- Implementar de manera práctica lo aprendido durante la carrera y en los trabajos de investigación realizados.
- Comprobar la viabilidad de implementación en un ambiente de producción de Fuel + OpenStack.
- Lanzar instancias de manera exitosa sobre el cloud implementado.
- Comprobar la posibilidad de instalación, o no, de aplicaciones sobre la instancia lanzada en el cloud.
- Maximizar el uso de los equipos utilizando la menor cantidad de placas de red posible y así poder realizar implementaciones útiles sin la necesidad de tener una cantidad excesiva de hardware.

3. Plan de trabajo

La PPS tendrá lugar en la sede central de la Universidad Nacional Arturo Jauretche y se llevará a cabo principalmente en el aula 219 en donde se encuentra el hardware y los recursos a utilizar. El estudiante realizó una implementación práctica de una nube informática (cloudcomputing).

La carga horaria total fue de 260 hs aproximadamente, superando las 200 hs establecidas en el plan de estudio de la carrera. Y se llevó a cabo entre Marzo y Agosto del año 2016.

En el Gráfico 3 puede verse una descripción del cronograma de trabajo.

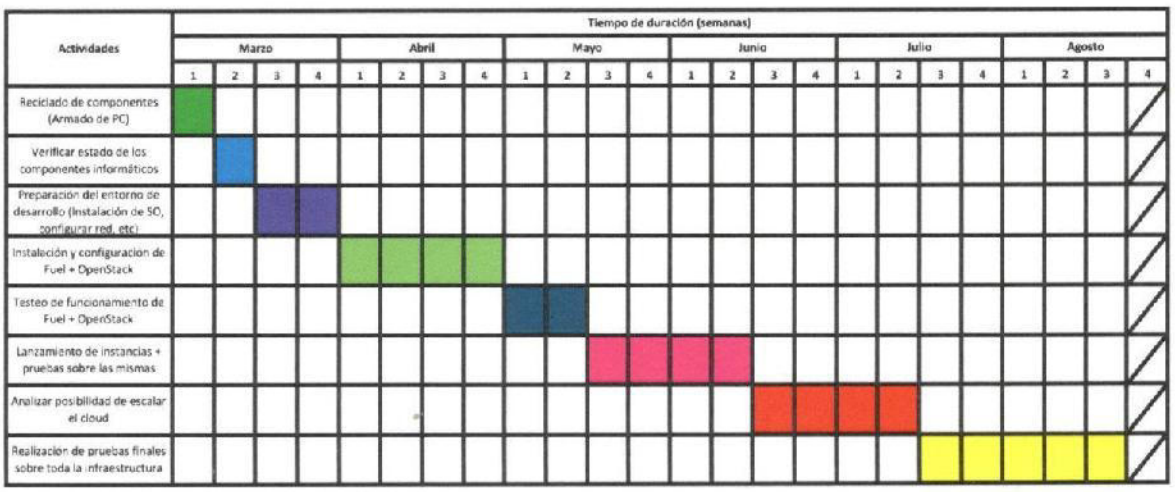


Gráfico1: Diagrama de GANTT de actividades realizadas.

4. Herramientas utilizadas

En primera instancia, el equipamiento a utilizar estaba constituido por:

- 1 Servidor HP Modelo Proliant DL360 G5, 12GB de RAM (4x2GB + 4x1GB) DDR2 667Mhz, microprocesador Intel Xeon E5450 3.0Ghz QuadCore. 3 HDD SAS x 146GB 10000RPM + 3 HDD SAS 72GB 10000 RPM.
- 1 Switch Cisco modelo SRW224G4-K9-AR, de 24 bocas (24 x 10/100+ 4 gigabit 10/100/1000).
- Direccionamiento directo de puertos basados en una IP estática lo que permitirá el acceso desde el exterior al cloud para control y/o mantenimiento (acceso remoto).
- Una red pública propia de la Universidad (192.168.0.0/24).
- Una red privada para control de OpenStack (172.16.0.0/16).
- Una red virtual determinada por placas de red cuya dirección de red será 10.20.0.0/8. La misma servirá para proveer arranque LAN PXE [6] a cada uno de los nodos que se deseen adicionar.
- Virtualbox con sus respectivas extensiones (VirtualBox + Virtualbox Extension Pack v5.0.16 r105871, la cual servirá como soporte para arranque de PXE).
- Sistema operativo Debian [7] AMD64 8.1 Jessie.

Debido a inconvenientes surgidos durante la ejecución de la práctica (ver Capítulo 2 "*Infraestructura de red*" para más información), el servidor debió ser reemplazado por equipamiento obtenido como parte del proyecto "Desarrollo de distribución Linux" en marco del programa "Universidad, Diseño y Desarrollo Productivo" del año 2013 en el cual se obtuvo financiamiento para la compra de, entre otras cosas, una PC con las siguientes características:

- Procesador socket 1155 Intel Quad Core I7 3770 3.4 Ghz
- Memoria DDR3 1333 Mhz Kingston 2 x 4 GB
- Disco rígido S-ATA 3 7200 rpm Western Digital Caviar Blue 1 Tb 64 Mb
- Motherboard Gigabyte GA-H61M-DS2
- Placa de red PCI Gigabit TP-Link

5. Tareas a ejecutadas

- Ensamblado de PC + Instalación de sistema operativo.
- Relevar los requerimientos y necesidades del sistema a implementar.
- Administrar de manera eficiente los recursos disponibles.
- Configurar la red necesaria para el correcto funcionamiento del cloud.
- Controlar el correcto funcionamiento del cloud.
- Pruebas de funcionamiento.
- Instalación e implementación correcta de Fuel y OpenStack.
- Lanzamiento de Instancia ejecutando Hetnux en la misma.

6. Resultados previos

La investigación de Cloud Computing por parte del autor del presente informe comenzó en el año 2013 a partir de una propuesta del docente Diego Encinas de la cátedra de Redes de Computadoras 1. Desde ese entonces se realizaron varios trabajos presentados en diversos congresos tales como CoNaIISI, CACIC y WICC (ver Anexo 1).

Durante los primeros informes se realizó la investigación de forma teórica de OpenStack, su funcionamiento y posibles aplicaciones. Con el paso del tiempo se comenzó a analizar dicha herramienta con mayor profundidad hasta que gracias a la PC armada con los fondos adquiridos a partir del Proyecto presentado en el programa "Universidad, diseño y desarrollo productivo", el desarrollo de un Sistema Operativo GNU/Linux orientado al uso Universitario (Hetnux), fue posible comenzar con el despliegue del cloud de manera física y así también las pruebas correspondientes de instalación y funcionamiento.

En primer instancia la implementación parecía ser sencilla debido a la existencia de documentación abundante y detallada oficial de OpenStack pero tras varios intentos sin éxito, además que en los foros oficiales se encuentran planteados muchos problemas pero ninguna respuesta, fue necesario recurrir a otra herramienta que permitiera realizar la configuración inicial del mismo por lo que finalmente se recurrió a Fuel (ver Capítulo 1).

7. Organización del trabajo

En el Capítulo 1 se encuentra la descripción correspondiente al software utilizado tanto como de OpenStack como también Fuel. Luego, en el Capítulo 2 está descripta de forma detallada la infraestructura de red utilizada para el despliegue del cloud incluyendo la configuración del router y del servidor DHCP. Finalmente, en el Capítulo 3, se explica el funcionamiento de OpenStack haciendo referencia al despliegue de red indicando sus interfaces, tanto físicas como virtuales, así como también al direccionamiento y flujo de paquetes de comunicación para concluir exponiendo los resultados obtenidos y expresando las conclusiones correspondientes en los capítulos 4 y 5 respectivamente.

Capítulo 1. Descripción del software

1.1 OpenStack

OpenStack es un software de código abierto que permite la implementación de una Infraestructure as a Service (IaaS) a través de múltiples servicios que, de manera coordinada, cumplen diferentes propósitos para lograr el correcto funcionamiento de dicha infraestructura. Cada servicio ofrece una interfaz de programación de aplicación (API) que facilita la integración de las mismas. Dependiendo de las necesidades de cada usuario, puede instalar algunos o todos los servicios disponibles. Estos servicios disponibles se describen en la Figura 1.

Servicio	Nombre del proyect	Descripción
Dashboard	Horizon	Proporciona una interfaz gráfica web basado en los servicios subadyacentes de OpenStack permitiendo el lanzamiento de una instancia, la asignación de direcciones IP y la configuración de los controles de acceso, etc.
Compute	Nova	Gestiona el ciclo de vida de las instancias de proceso en un ambiente de OpenStack. Las responsabilidades incluyen la descomposición, programación y el cierre definitivo de las máquinas en la demanda.
Networking	Neutron	Permite la conectividad de la red para uno o varios servicios de OpenStack, como OpenStack Compute. Proporciona una API para que los usuarios definan las redes y los archivos adjuntos en ellos. Tiene una arquitectura conectable que soporta varios proveedores y tecnologías de redes populares.
Almacenamiento		
Object Storage	Swift	Almacena y recupera los objetos de datos no estructurados de su elección mediante una REST, es una API basada en HTTP. Es muy tolerante a fallos con su réplica de datos y su posible escalabilidad. Su puesta en práctica no es como un servidor de archivos con los directorios montables.
Block Storage	Cinder	Proporciona almacenamiento persistente en bloques para las instancias en ejecución. Soporta dispositivos plug and play y permite gestionar, por lo tanto, todos aquellos dispositivos de almacenamiento en bloques.
Servicios compartidos		
Identity Service	Keystone	Proporciona un servicio de autenticación y autorización para otros servicios de OpenStack. Proporciona también un catálogo de endpoints para todos los servicios de OpenStack.
Image Service	Glance	Permite almacenar y recuperar imágenes de disco de máquinas virtuales. OpenStack Compute hace uso de esta instancia durante la instancia de provisión.
Telemetry	Ceilometer	Monitorea la nube de OpenStack para la evaluación comparativa, la escalabilidad y con fines estadísticos.
Servicios alto nivel		
Orchestration	Heat	Organiza múltiples aplicaciones en la nube de material compuesto utilizando ya sea el formato de la plantilla HOT nativo o el formato de la plantilla AWS, tanto a través de una API REST nativa de OpenStack nativo y una forma de nube compatible con la API de querys.

Figura 1. Servicios de OpenStack.

La arquitectura conceptual de OpenStack puede ser apreciada en la Figura 2:

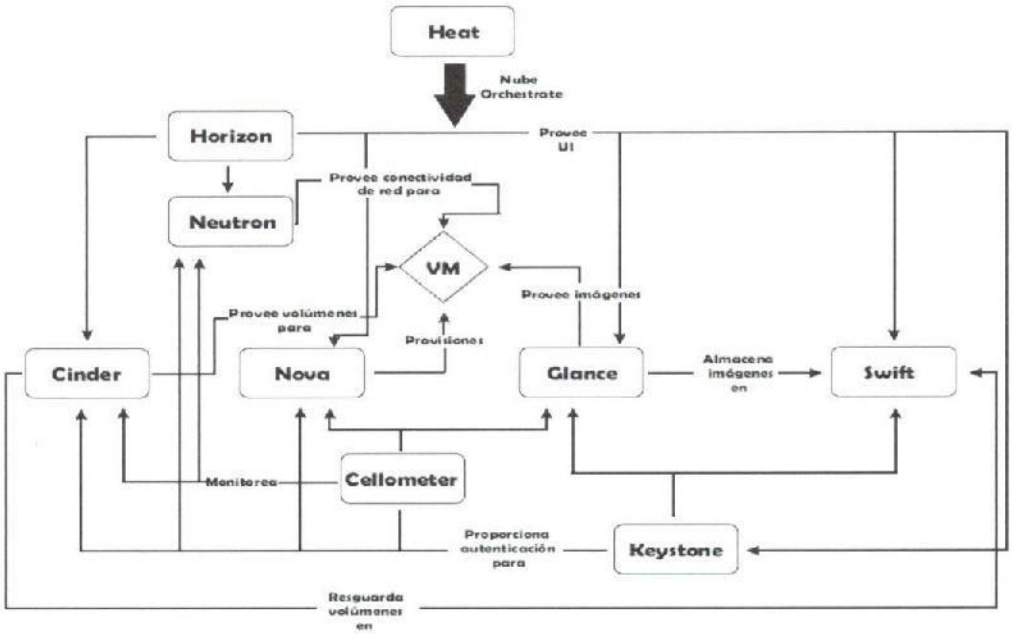


Figura 2. Arquitectura conceptual de OpenStack.

Varias compañías se han unido al proyecto: Red Hat, AT&T, Canonical, Cisco, Dell, GoDaddy, HP, IBM, Intel, Rackspace Hosting, Nexenta, AMD, Suse, VmWare, Oracle, Yahoo, entre otras.

1.2 Fuel

Fuel (Figura 3) es una herramienta Desarrollada por Mirantis en la cual se ejecuta un script que permite configurar de forma amigable los recursos que se desean otorgar a la infraestructura, desde la cantidad de nodos, los núcleos de procesador, la memoria RAM, etc.

FUEL

Figura 3: Logotipo de Fuel.

Fuel trabaja con un nodo máster el cual es el encargado de controlar a los nodos slaves que son los que van a contener la infraestructura OpenStack. Es decir, desde el nodo Fuel Máster se indican qué paquetes se van a instalar en cada nodo slave (Glance, Nova-Compute, Keystone, etc.) para luego en los slaves poder tener armados los nodos compute y controller, sin necesidad de tener que realizar configuraciones manuales en cada uno de los mismos.

Capítulo 2. Infraestructura de red

La infraestructura de red implementada puede apreciarse en la Figura 4. Cabe mencionar que en primera instancia el servidor de entorno de trabajo de Openstack (que incluye las máquinas virtuales [8]del nodo máster junto con las de los nodos esclavos) fue conectado de forma directa a la red de la Universidad por medio de la placa de red onboard (eth0). Esto en un principio funcionó correctamente pero luego generó conflictos ya que las máquinas virtuales, basadas en QEMU [9], requerían IP's estáticas (y en algunos casos dinámicas). Pero al ser una red pública la cual asigna IP por medio de Dynamic Host ConfigurationProtocolDHCP, sucedía que al momento de realizar la instalación de OpenStack se asignaba una IP a los nodos. Y al reiniciar los servicios, dichos nodos no funcionaban ya que las IP habían sido vinculadas a otros dispositivos. Es por ello que se decidió realizar la implementación y configuración de un servidor DHCP para así poder controlar las IP asignadas a cada dispositivo. Además, se agregó un switch para poder escalar la nube de tal forma que puedan agregarse otras computadoras a la nube en un futuro.

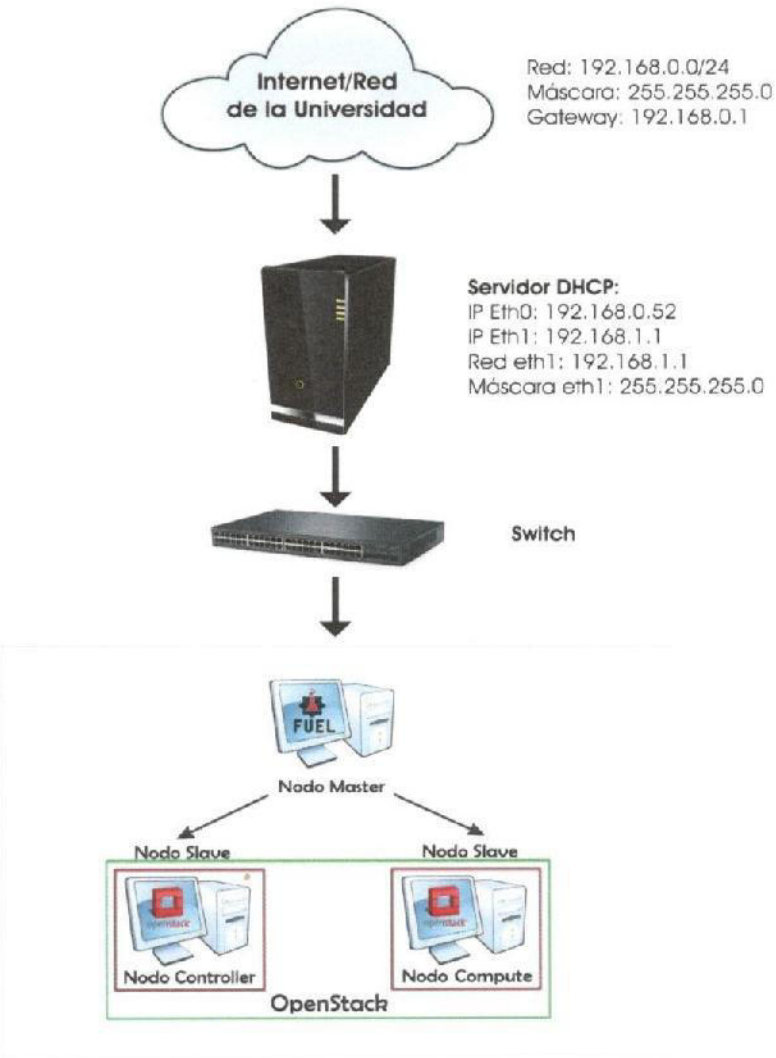


Figura 4: Infraestructura de red.

2.1 Implementación del servidor DHCP

2.1.1 DHCP

El protocolo de configuración dinámica de host (DHCP, Dynamic Host Configuration Protocol [10] es un estándar TCP/IP diseñado para simplificar la administración de la configuración IP de los equipos de una red.
Si se dispone de un servidor DHCP, la configuración IP de los PCs puede hacerse de forma automática, evitando así la necesidad de tener que realizar manualmente uno por uno la configuración TCP/IP de cada equipo.
Un servidor DHCP es un servidor que recibe peticiones de clientes solicitando una configuración de red IP. El servidor responderá a dichas peticiones proporcionando los parámetros que permitan a los clientes autoconfigurarse al tener habilitada la opción de obtención de IP automática.

El servidor proporcionará al cliente al menos los siguientes parámetros:

- Dirección IP
- Máscara de subred

Opcionalmente, el servidor DHCP podrá proporcionar otros parámetros de configuración tales como:

- Puerta de enlace
- Servidores DNS
- Muchos otros parámetros más

El servidor DHCP proporciona una configuración de red TCP/IP segura y evita conflictos de direcciones repetidas. Utiliza un modelo cliente-servidor en el que el servidor DHCP mantiene una administración centralizada de las direcciones IP utilizadas en la red. Los clientes podrán solicitar al servidor una dirección IP y así poder integrarse en la red.

El principio de funcionamiento puede observarse en la Figura 5.

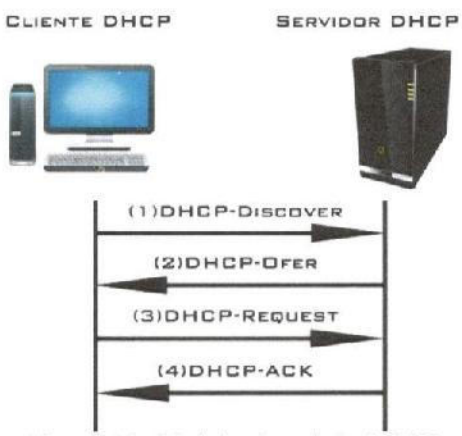


Figura 5: Principio de funcionamiento de DHCP.

Cuando un dispositivo configurado con DHCP e IPv4 se inicia o se conecta a la red, el cliente transmite un mensaje de detección de DHCP (DHCPDISCOVER) para identificar cualquier servidor de DHCP disponible en la red. Un servidor de DHCP responde con un mensaje de oferta de DHCP (DHCPOFFER), que ofrece una concesión al cliente. El mensaje de oferta contiene la dirección IPv4 y la máscara de subred que se deben asignar, la dirección IPv4 del servidor DNS y la dirección IPv4 del gateway predeterminado. La oferta de concesión también incluye la duración de esta.

El cliente puede recibir varios mensajes DHCPOFFER si hay más de un servidor de DHCP en la red local. Por lo tanto, debe elegir entre ellos y enviar un mensaje de solicitud de DHCP (DHCPREQUEST) que identifique el servidor explícito y la oferta de concesión que el cliente acepta. Un cliente también puede optar por solicitar una dirección previamente asignada por el servidor.

Suponiendo que la dirección IPv4 solicitada por el cliente, u ofrecida por el servidor, aún está disponible, el servidor devuelve un mensaje de reconocimiento de DHCP (DHCPACK) que le informa al cliente que finalizó la concesión. Si la oferta ya no es válida, el servidor seleccionado responde con un mensaje de reconocimiento negativo de DHCP (DHCPNAK). Si se devuelve un mensaje DHCPNAK, entonces el proceso de selección debe volver a comenzar con la transmisión de un nuevo mensaje DHCPDISCOVER. Una vez que el cliente tiene la concesión, se debe renovar mediante otro mensaje DHCPREQUEST antes de que expire.

El servidor DHCP asegura que todas las direcciones IP sean únicas (no se puede asignar la misma dirección IP a dos dispositivos de red diferentes de forma simultánea).

2.1.2 Configuración de PC como router.

En primer instancia, para que el servidor DHCP funcione correctamente es necesario configurar la PC, que cumplirá dicha tarea, como router. Esto es, permitir el pasaje de datos de una interfaz de red hacia la otra. Para ello (siempre como usuario root) es necesario:

1) Habilitar el bit de redireccionamiento:

#echo 1 > /proc/sys/net/ipv4/ip_forward

Para habilitar este bit de forma definitiva entonces es necesario editar el archivo /etc/sysctl.conf descomentando la línea:

net.ipv4.ip_forward=1

De esta manera, al reiniciar el equipo el cambio estará guardado.

2) Modificar el archivo /etc/network/interfaces de forma tal que su contenido sea:

```
auto lo
iface lo inetloopback
allow-hotplug eth1
iface eth1 inet static
```

address 192.168.1.1
netmask 255.255.255.0

Así queda como estática la IP de la placa de red eth1 indicando también su máscara de subred.

3) Crear un script bash llamado script.sh con el siguiente contenido:

#!/bin/bash
iptables –Z
iptables –t nat –F
iptables –t nat –A POSTROUTING –s 192.168.1.0/24 –o eth0 –j MASQUERADE

iptables –Z Establece en 0 los contadores de las reglas.
iptables –t nat –F Borra todas las reglas de la tabla nat.
POSTROUTING: Modifica los paquetes justo antes de salir del equipo así los mismos son enmascarados y todo el tráfico de la red 192.168.1.0/24 se envía a la interfaz eth0.

Y se le da permisos para poder ejecutarlo:

#chmod 755 script.sh
#chmod +x script.sh

4) Se agrega el script para que sea ejecutado al iniciar el sistema operativo:
#cd /etc/init.d
#update-rc.d router defaults

Una vez realizados estos pasos y reiniciado el sistema entonces la PC ya funcionará como router, es decir que si se conectara un switch o una PC a la interfaz eth1 estos dispositivos podrán conectarse a internet a través de la interfaz eth0 previo a haberse configurado de forma manual una IP y una submáscara de red.
Para ahorrar esos pasos lo configurado fue el servidor DHCP.

2.1.3 Configuración del servidor DHCP.

El servidor DHCP implementado para este trabajo fue el paquete ISC-DHCP-SERVER [11] de Debian 8 Jessie el cual fue instalado directamente desde los repositorios del sistema operativo. Para la instalación y configuración del mismo (luego de haber configurado los repositorios):

1) Se instala el paquete:
#apt-get install isc-dhcp-server

2) Se agrega el siguiente contenido al archivo /etc/network/interfaces:

allow-hotplug eth0
iface eth0 inet static
address 192.168.0.130
netmask 255.255.255.0
gateway 192.168.0.254

nameserver 192.168.0.254

Así se configura estáticamente la interfaz que tendrá salida a internet. En este caso fue necesario configurar la misma para que pueda vincularse con la red de la Universidad.

3)Al archivo /etc/default/isc-dhcp-server se agrega:

INTERFACES="eth1"

Indicando así la interfaz de red que funcionará como servidor DHCP.

4) Al archivo /etc/dhcp/dhcpd.conf se agrega:

```
ddns-update-style none;
option doman.name-servers 192.168.0.254;
default-lease-time 7200;
max-lease-time 21600;
authoritative;
log-facility local7;

subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.2 192.168.1.100;
        optionrouters 192.168.1.1
}
```

ddns-update-stylenone: Indica actualización de IP DHCP automática.
option doman.name-servers 192.168.0.254: Define los servidores de dominio de la red.
default-lease-time: 7200: Tiempo de asignación de direcciones en segundos.
max-lease-time: Tiempo máximo de asignación de una dirección en segundos.
authoritative: Intenta reasignar datos a clientes mal configurados suponiendo que su configuración es la correcta.
log-facility local7: Indica que se desea guardar en los logs todo el debugging.
subnet 192.168.1.0 netmask 255.255.255.0: se define la red de difusión como así también su máscara.
Range192.168.1.2 192.168.1.100: Indica el rango de IP's a asignar.
optionrouters 192.168.1.1: Indica la IP del router de la red, recordar que éste es el que se ha configurado en la sección anterior.

5) Finalmente se reinicia el servicio:

#/etc/init.d/isc-dhcp-server restart

Habiendo seguido todos estos pasos, el servidor DHCP ya se encuentra funcionando y asignando direcciones IP dinámicas por su interfaz eth1. Asimismo esto permitirá asignar direcciones IP estáticas dentro de la red de OpenStack, asegurando que al reiniciar las mismas no estarán ocupadas como sí sucedía en la red de la Universidad.

Capítulo 3.Infraestructura de OpenStack

OpenStack permite el montaje de diferentes tipos de escenarios en función del hardware disponible y de las necesidades a cubrir. En el caso de este trabajo se optó por la infraestructura Legacy que es la más básica y la que menos requisitos de hardware solicita. Éstos pueden apreciarse en la Figura 6.

Se dispondrá de 3 nodos: Network (encargado del servicio de DHCP, router, etc), Compute (encargado principalmente de gestionar las instancias (tiene como funcionalidad garantizar la correcta comunicación entre los diferentes nodos y servicios que cada uno de ellos ofrece) y Controller (posee las bases de datos de usuarios y configuraciones). La arquitectura del nodo Compute y del nodo Controller puede apreciarse en la Figura 6.

El escenario legacy [13] permite proveer un "auto servicio" dentro de la infraestructura. Es decir permite que tanto administradores como usuarios sin privilegios (o "tenants", "inquilinos") puedan coexistir dentro del mismo entorno (sus requisitos de hardware pueden apreciarse en la Figura 7.)

En esta infraestructura coexisten los siguientes componentes:

- Redes tenants: Permiten conectividad a proyectos específicos dentro de ciertas redes. Así, los usuarios invitados pueden administrar en su totalidad proyectos que el administrador define para ellos. Generalmente este tipo de proyectos utilizarán redes privadas.
- Redes externas: Estas redes son las que permiten la conectividad con el exterior.

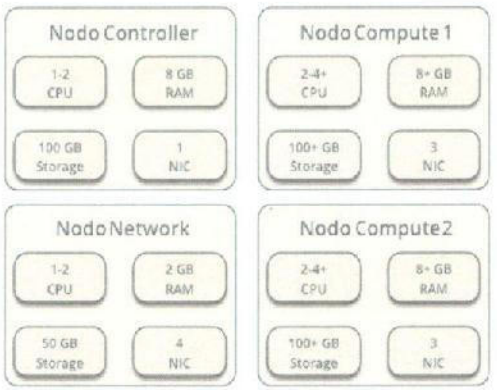


Figura 7: Requisitos de hardware, escenario Legacy.

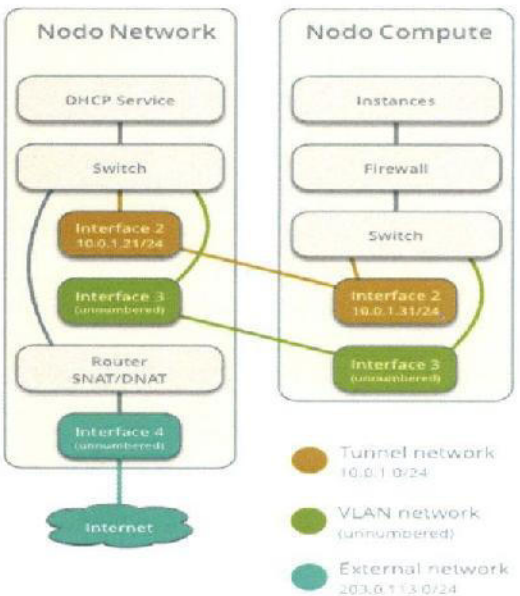


Figura 6: Arquitectura general entre el nodo Compute y el Network.

Asimismo, estas redes pueden ser manipuladas sólo por usuarios administradores ya que generalmente poseen interacción directa con las placas físicas de red. Sin embargo, pueden configurarse interfaces virtuales.

- Routers: Son los que permiten la conectividad entre los proyectos y el exterior. Utilizan SNAT para los paquetes salientes y DNAT para los paquetes entrantes. Las redes externas mencionadas con anterioridad hacen referencia a las IP's asignadas a los routers que permitirán el acceso a redes remotas.
- Servicios de soporte: Aquí se encuentra el servidor DHCP de OpenStack, es quien le asignará las IP a los diferentes proyectos creados.

La arquitectura legacy permite la relación entre diferentes proyectos así como también el acceso exterior y la salida a internet dentro de cada uno de ellos. Promueve los recursos básicos para lanzar instancias de manera correcta. Sin embargo el hecho de realizar todas las conexiones de red en el mismo nodo genera un punto centralizado de fallo.

3.1 Nodo network

Los componentes del Nodo Network (apreciables en la Figura 8) son:

1. Puente Linux que administrará switches virtuales y conectividad a través de ellos así como también a través de los diferentes puertos.
2. Un servidor DHCP que administrará además los nombres de los entornos de trabajo. Asimismo le asignará las IP's correspondientes a cada una de las instancias de trabajo.
3. Un agente L3 (level 3) que será el encargado de enrutar los datos entre redes de proyectos y de proyectos con redes externas.
4. Un agente manejador de metadata para instancias.

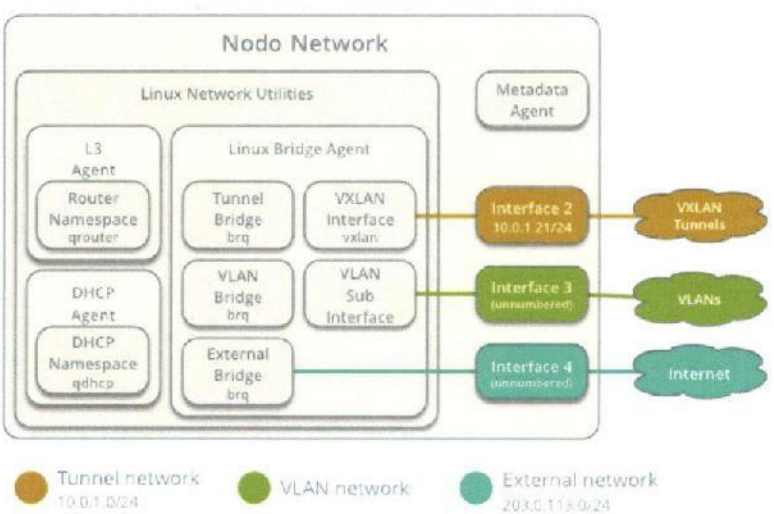


Figura 8: Componentes del nodo Network.

3.2 Nodo Compute

El nodo compute posee un elemento clave (ver Figura 9): el puente Linux (Linux Bridge Agent), éste es el encargado de administrar interfaces virtuales y la conectividad entre ellas así como también la interacción de puertos virtuales con otras redes, grupos de seguridad, nombres de dominio, etc.

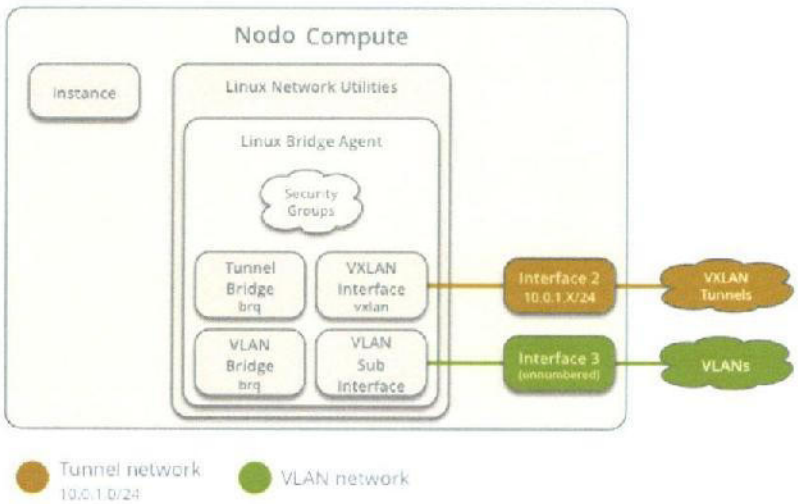


Figura 9: Componentes del nodo Compute.

3.3 Flujo de paquetes

El flujo de paquetes en la infraestructura implementada se denomina norte-sur debido a que la red con infraestructura física administra el direccionamiento y otros servicios entre el proveedor y la red externa. De todas formas el proveedor podría, eventualmente, generar conexiones directas con la red externa (Figura 10).

- Red
  - red 192.168.1.0/24
  - la asignación de direcciones IP de 192.168.1.2/24 a 192.168.1.100/24
  - Proyecto interfaz del router de red TR 192.168.1.1
- Red del proyecto
  - red 172.16.0.0/24
  - 172.16.0.1 Gateway con la dirección MAC TG
- Nodo compute
  - Instancia 1 172.16.0.11 con la dirección MAC I1
- La instancia 1 reside en el nodo compute 1 y utiliza una red del proyecto.
- La instancia envía un paquete a un host en la red externa.

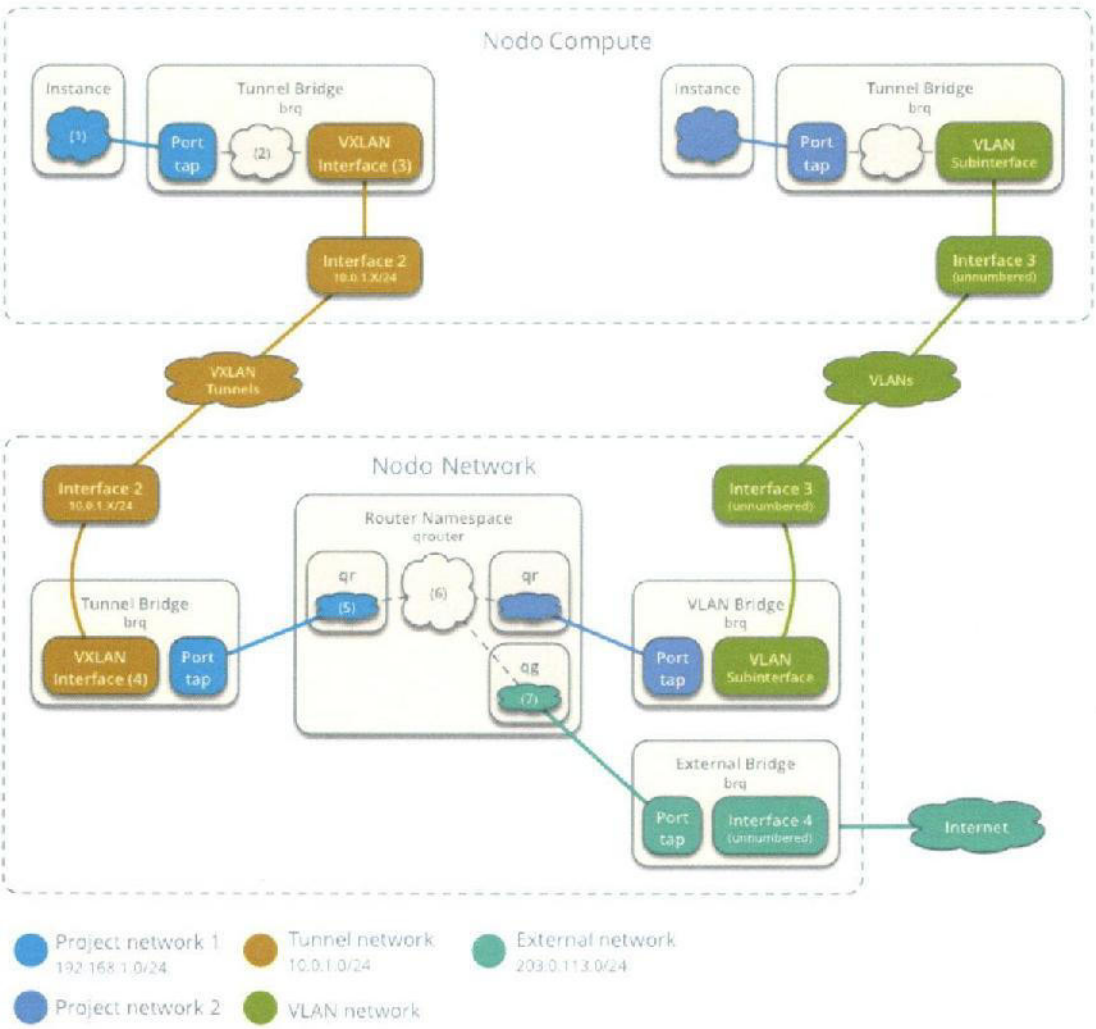


Figura 10: Flujo de paquetes.

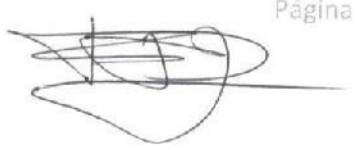
Los siguientes pasos involucran al nodo compute:

1. Para proyectos en redes VXLAN:
    1. La instancia 1 redirecciona el paquete al puente túnel brq. El paquete contiene la MAC del destino, TG, ya que el destino reside en otra red. En este punto cabe recordar que para intercambio de paquetes entre redes, no se modifican los datos de la capa de red hacia arriba pero sí los que se encuentren debajo de ella.
    2. Las reglas de seguridad de los grupos (2) administran el destino del paquete, es decir si podrá continuar o no y si es así hacia dónde.
    3. El túnel brq redirecciona el paquete a la interfaz virtual del túnel *vxlan-sid* (3) donde *sid* contiene la segmentación de la red del proyecto destino.
    4. La interfaz física del túnel redirecciona el paquete al nodo network.
2. Para proyectos en redes VLAN:
    1. La instancia 1 redirecciona el paquete a la VLAN del puente brq. El paquete contiene la MAC del destino, TG, ya que el destino reside en otra red.

2. Las reglas de seguridad de los grupos administran el destino del paquete, es decir si podrá continuar o no y si es así hacia dónde.
3. La VLAN puente brq redirecciona el paquete a la interfaz lógica de la VLAN *device.sid* donde *device* hace referencia a la interfaz física de la VLAN subyacente y *sid* contiene la segmentación de la red del proyecto (Segmentation ID).
4. La interfaz lógica de la VLAN *device.sid* redirecciona el paquete al nodo network a través de la interfaz física de la VLAN.

Los siguientes pasos involucran al nodo network:

1. Para proyectos en redes VXLAN:
    1. La interfaz física del túnel redirecciona el paquete a la interfaz lógica del túnel *vxlan-sid* (4) donde *sid* contiene el ID de segmentación de la red del proyecto.
    2. La interfaz lógica del túnel *vxlan-sid* redirecciona el paquete al puente túnel brq.
    3. El puente túnel brq redirecciona el paquete a la interfaz qr (5) en el router cuyo nombre de entorno es qrouter. La interfaz qr contiene la dirección IP del router de la red del proyecto, TG.
2. Para proyectos en redes VLAN:
    1. la interfaz física de la VLAN redirecciona el paquete a la interfaz lógica de la VLAN device.sid donde device hace referencia a la interfaz física subyacente y sid contiene el ID de segmentación de la red del proyecto.
    2. La interfaz lógica de la VLAN device.sid redirecciona el paquete al puente de la VLAN, brq.
    3. El puente brq de la VLAN redirecciona el paquete a la interfaz qr en el router dentro del entorno qrouter. La interfaz qr contiene la IP de Gateway de la red de proyecto, TG.
3. El servicio IPABLES (6) realiza SNAT al paquete utilizando la interfaz qg (7) la dirección IP de origen. La interfaz qg contiene la dirección IP del router de la red del proyecto, TR.
4. El router con entorno llamado qrouter redirecciona el paquete al puente externo brq.
5. El puente externo brq redirecciona el paquete a la red externa a través de la interfaz física externa.



Capítulo 5. Resultados

Una vez finalizado el despliegue de red, instalación del servidor DHCP e instalación del software necesario es posible ingresar al dashboard de OpenStack (Figura 12) y proceder con el lanzamiento de la instancia que será la utilizada para ejecutar Hetnux.

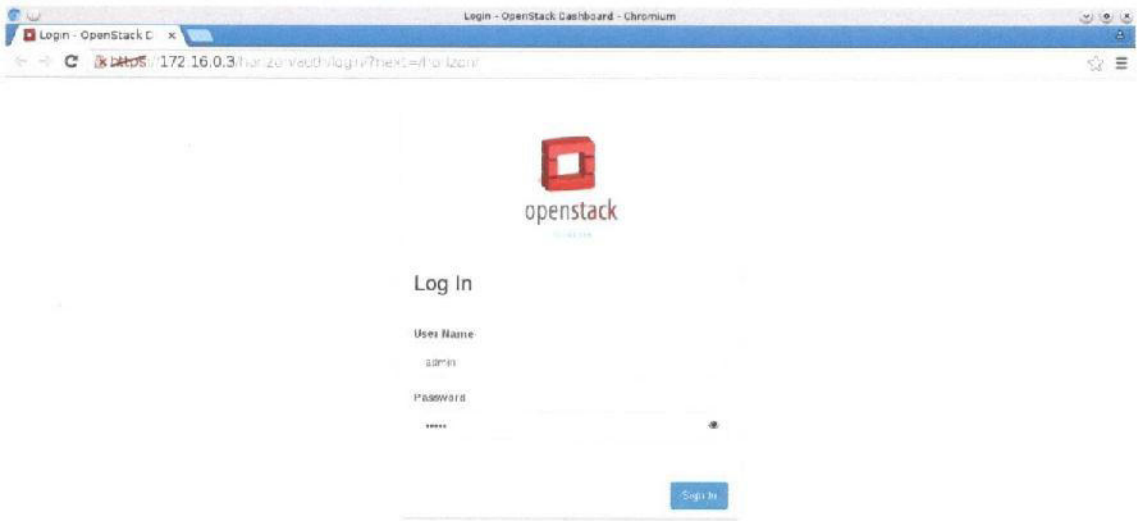


Figura 12: Pantalla de LoginDashboardOpenStack.

Horizon (el Dashboard de OpenStack) ofrece una interfaz amigable para realizar el administración del cloud desplegado permitiendo utilizar de manera gráfica los principales comandos que se encuentran disponibles en una terminal. Asimismo permite observar el uso de recursos, grupos de trabajo, usuarios, contraseñas, servicios en ejecución, entre otros.

Figura 13: Menú de administración de Instancias.

Como puede observarse en la Figura 13, dentro del submenú Instances aparecen las instancias lanzadas y una pequeña descripción de las mismas, esto es muy útil ya que permite observar la IP desde la cual es posible acceder a la misma y dentro de qué red.

Por otra parte, permite observar y administrar las redes e interfaces de red (Figura 14), administrar los diferentes sabores (Figura 15) que son configuraciones predeterminadas para futuras instancias a lanzar, administrar los diferentes volúmenes de almacenamiento, etc.

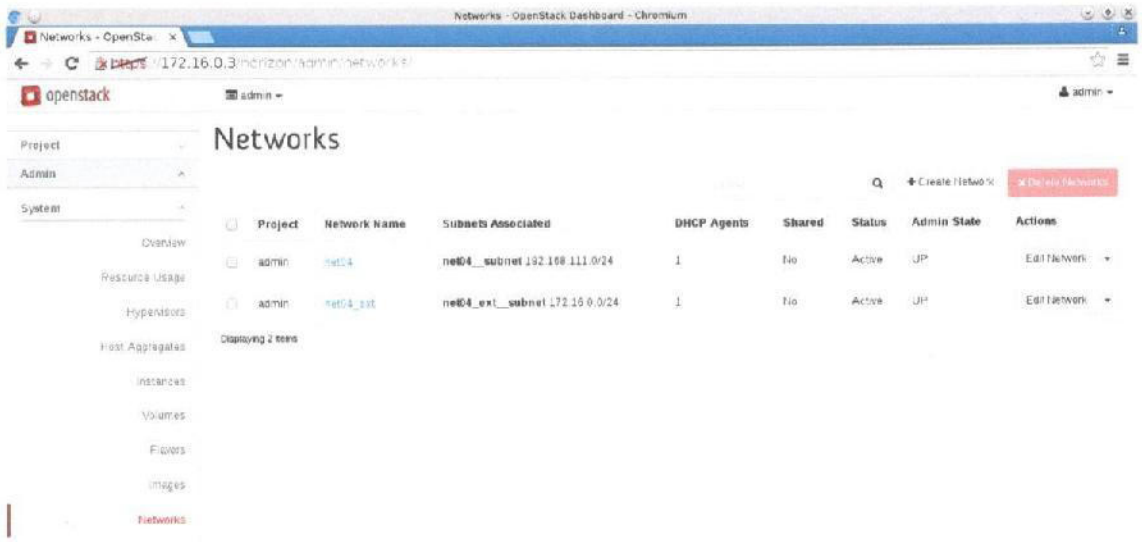


Figura 14: Menú de administración de redes.

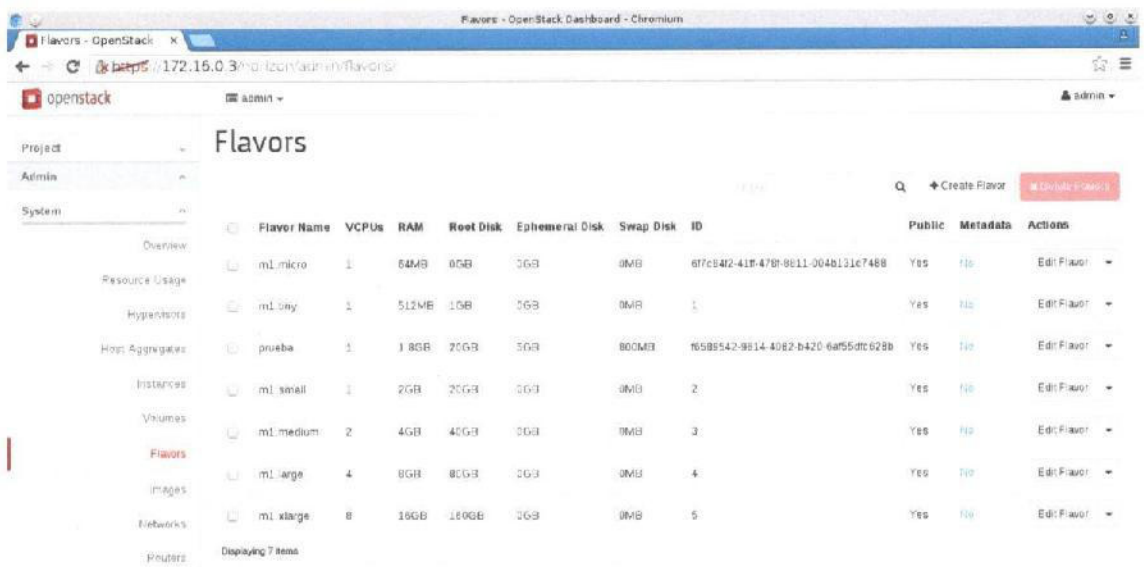


Figura 15: Menú de administración de sabores.

Finalmente, en la Figura 16 puede apreciarse una captura de pantalla de Hetnux ejecutándose sobre la Instancia de OpenStack.

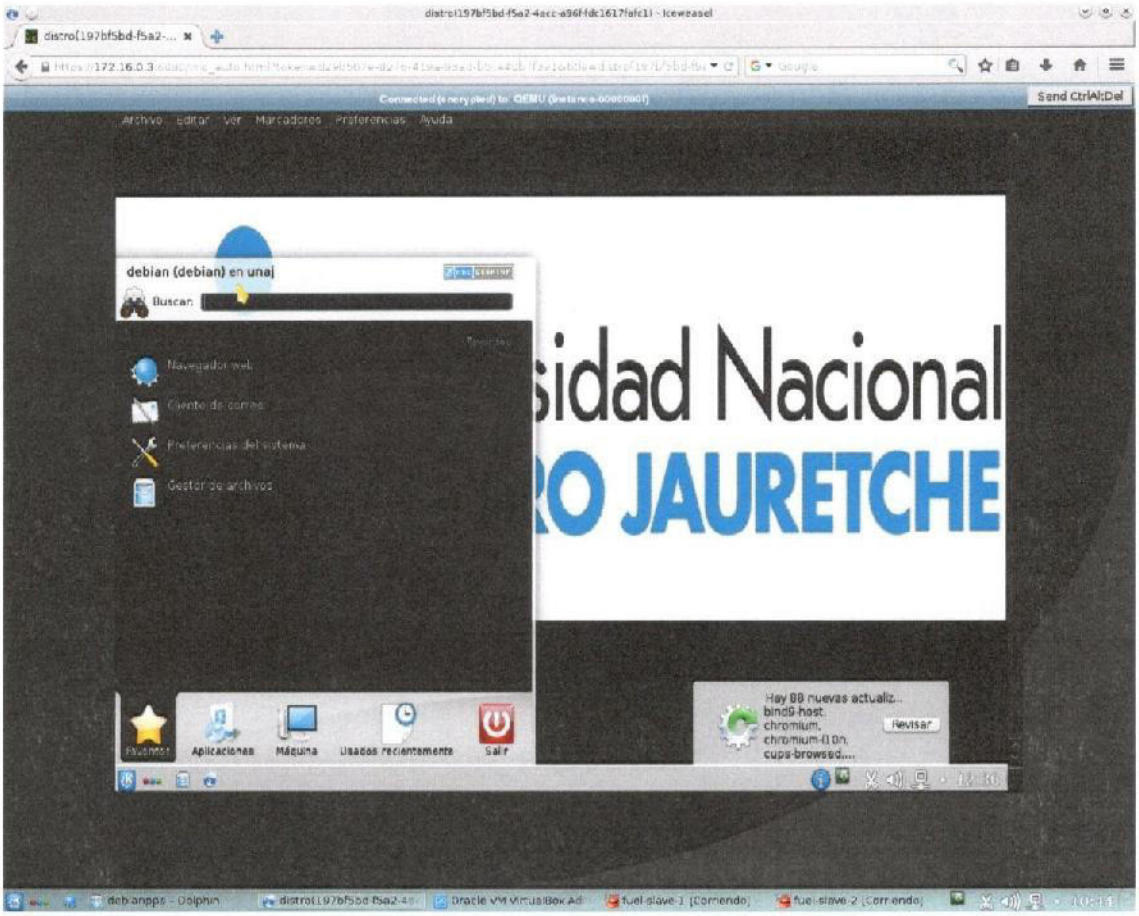


Figura 16: Captura de pantalla de Hetnux funcionando sobre la instancia lanzada.

Capítulo 6. Conclusiones

Si bien OpenStack no es sencillo de implementar y hay que tener conocimientos básicos de redes una vez iniciada la expansión de la nube o clúster, es muy sencillo escalarlo. Se puede decir que es posible hacerlo de manera exponencial, ya que es un sistema muy modulado lo que permite tanto expandirse como reducirse de manera rápida y segura, una vez implementado correctamente los cimientos del sistema.

Queda descartada la posibilidad de reciclar equipamiento obsoleto ya que se requiere de hardware relativamente actualizado y principalmente gran cantidad de RAM (mínimo 16 GB) para que sea utilizable ya que de lo contrario la interfaz se torna muy lenta.

No se logró realizar pruebas de benchmarking ya que con 8 GB de RAM el sistema se torna inestable tendiendo a congelarse (frezarse), al exigirlo mínimamente.

Queda desplegada una infraestructura que servirá a futuro para realizar nuevas pruebas y, en caso de disponer los recursos, expandir el cloud permitiendo hacer de éste un ambiente en producción ya sea para proveer servicios a los alumnos, docentes o a la Universidad en general, incluso a otras Universidades.

En línea con lo trabajado y expuesto, se busca a futuro:

- Implementación de un IaaS encargado de realizar operaciones en procesamiento paralelo aumentando la eficiencia y reduciendo los costes generados.
- Implementación de una nube académica en la cual se puedan ofrecer distintos servicios personalizados y atender a las necesidades de cómputo y procesamiento de los distintos departamentos de la universidad.
- Creación de una red entre Universidades para compartir recursos didácticos así como también tecnológicos y más específicamente de procesamiento.

A partir de lo que se ha mostrado en el presente trabajo, puede decirse que OpenStack ofrece una gran variedad de servicios que permiten adaptarse a las necesidades específicas de diversos usuarios y empresas permitiendo explorar, también, nuevos horizontes.

Anexo I. Trabajos de investigación.

A continuación se detallan los trabajos de investigación de los que fue integrante el autor del presente informe:

- Zaccardi Gonzalo, Galarza Brian, Morales Martín, Encinas Diego: "Despliegue y ejecución en un cloud privado." en el 4° Congreso Nacional de Ingeniería en Informática y Sistemas de Información (CoNaIISI 2016).
- Galarza Brian, Zaccardi Gonzalo, Rossatto Daniel, Bond Román, Morales Martín, Encinas Diego: "Performance de cloudcomputing para HPC: despliegue y simulación" en el 19°Workshop de Investigadores en Ciencias de la Computación (WICC 2016).
- Galarza Brian, Zaccardi Gonzalo, Encinas Diego, Morales Martín: "Análisis de despliegue de una IaaS utilizando Openstack" en el 19° Congreso Argentino de Ciencias de la Computación (CACIC 2015).
- Encinas Diego, Kunysz Eduardo, Galarza Brian, Zaccardi Gonzalo, Morales Martín: "Performance de arquitecturas multiprocesador: técnicas de modelado y simulación, plataformas reconfigurables y cloudcomputing" en el 17° Workshop de Investigadores en Ciencias de la Computación (WICC 2015).
- Zaccardi Gonzalo, Galarza Brian, Encinas Diego, Morales Martín: "Implementación de Cloud Computing utilizando OpenStack" en el 2° Congreso Nacional de Ingeniería en Informática y Sistemas de Información (CoNaIISI 2014).
- Galarza Brian, Tuanmá Cristian, Zaccardi Gonzalo, Encinas Diego, Morales Martín: "Implementaciones de Cloud Computing y aplicaciones en el ámbito universitario" en el 1° Congreso Nacional de Ingeniería en Informática y Sistemas de Información (CoNaIISI2013).

  También se realizaron diversas presentaciones de Hetnux (ver Anexo II) tanto en Tecnópolis en el stand "Yo quiero estudiar" [12] durante el mes de Septiembre de 2014 así como también en la radio de la Universidad Nacional Arturo Jauretche durante el mismo período.

Anexo II. Hetnux

En el año 2013 se propuso desarrollar un Sistema Operativo empleando herramientas de software libre con el fin de proveer un sistema operativo gratuito a las redes universitarias y escolares de la región.

Hetnuxes un sistema operativo basado en la distribución de GNU/Linux Debian e intenta satisfacer las necesidades de aquellos estudiantes de la UNAJ. La distribución fue modificada (y es modificada continuamente) agregándole aquellas funcionalidades que necesitan los estudiantes como pueden ser IDE's de desarrollo en el caso de estudiantes del área de informática o software útil para materias comunes a todas las carreras como lo son Matemática, Química, etc. Asimismo se intenta dar una visión agradable e intuitiva para motivar a todos aquellos que la vean y/o utilicen por primera vez. Posee un menú (Figura 11) que permite seleccionar entre un modo instalable y uno live, este último permite utilizar el sistema operativo desde un dispositivo externo cargándose el mismo en RAM sin la necesidad de instalarlo en los dispositivos de almacenamiento interno de la PC.

Además, se proyecta establecer la interacción de alumnos y docentes/investigadores de la carrera de Ingeniería en Informática de la UNAJ con escuelas secundarias de educación técnica en la zona de influencia de la Universidad. El desarrollo e interacción se basan en tres ejes: a) emprender la implementación de una distribución GNU/Linux por parte de los alumnos de la carrera de Ingeniería en Informática de la UNAJ, b) articular contenidos y metodologías de enseñanza entre ambos niveles educativos y, c) aumentar las expectativas de los estudiantes secundarios, desarrollando intereses que favorezcan su inserción en la Universidad y que respondan a las necesidades del sector "Software y Hardware" de la región.

El proyecto nació el en marco del programa "Universidad, Diseño y Desarrollo Productivo 2013" y en el mes de Octubre de 2016 se convirtió en el primer registro de Propiedad Intelectual de la Universidad Nacional Arturo Jauretche [13] mediante el expediente 5313703 en el boletín oficial número 33.483 [14].

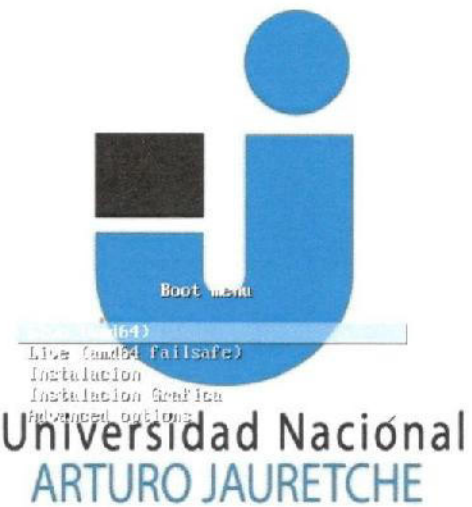


Figura 11: Menú de booteo de Hetnux.

Universidad Nacional ARTURO JAURETCHE

Referencias.

[1] Xing, Y., Zhan, Y.: "Virtualization and Cloud Computing". En: Proceedings pp.305-312, Springer Link. ISBN 978-3- 642-27323- 0. 2012.

[2] "OpenStack Cloud Software: Open source software for building private and public clouds."http://www.openstack.org. Agosto 2016.

[3] "Documentación Fuel." https://docs.mirantis.com/openstack/fuel/fuel-7.0/.Agosto 2016.

[4] "Mirantis Inc. Pure playOpenStack."https://www.mirantis.com.Agosto 2016.

[5]Galarza B.; Zaccardi G.; Encinas D.; Morales M."Implementación de Cloud Computing utilizandoOpenStack" (CoNaIISI 2014). San Luis, Argentina.

[6] "Intel Inside: 'Preboot Execution Environment (PXE) Specification'."ftp://download.intel.com/design/archives/wfm/downloads/pxespec.pdf . Septiembre 1999.

[7]"Debian." https://www.debian.org.Agosto 2016.

[8]Popek, G.J., Goldberg, R.P.: Formal Requirements for Virtualizable Third Generation Architectures. In: Communications in the ACM, Volume 17, Number 7, pp. 412-- 421. USA. (1974).

[9] Nussbaum, L., Anhalt, F., Olivier, M., Gelas, J.: Linux-based virtualization for HPC clusters. En: Montreal Linux Symposium (2009), pp. 221—234. Canada. (2009).

[10]"¿Qué es el DHCP?."http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m2/servidor_dhcp.html. Julio 2013.

[11] "isc-dhcp-server." https://www.isc.org/downloads/dhcp/. Septiembre 2016.

[12] "Participantes del Programa 'Universidad, Diseño y Desarrollo Productivo' en 'Yo quiero estudiar'. https://www.youtube.com/watch?v=rCqz4l6yWPc . Octubre 2014.

[13] "La UNAJ realizó el primer registro de Propiedad Intelectual." https://www.unaj.edu.ar/la-unaj-realizo-el-primer-registro-de-propiedad-intelectual/. Octubre 2016.

[14] "Boletín Oficial de la República Argentina. Año CXXIV. Número 33.483." https://www.eldial.com/nuevo/boletin/2016/BO161017.pdf

Esp. Ing. Martín Morales
Coordinador Ingeniería en Informática
Instituto de Ingeniería y Agronomía
U.N.A.J.

DIEGO ENCINAS