The background of the entire page is a photograph of a city skyline at sunset. On the left side, a tall, modern skyscraper with a curved, metallic facade is illuminated from within, showing a warm golden glow. The rest of the city is visible in the background, with various buildings and structures silhouetted against the colorful sky. The sky transitions from a deep blue at the top to a mix of orange, yellow, and purple near the horizon, with scattered clouds catching the low light.

Despliegue de Arquitectura de nube con OpenStack

PROYECTO FINAL

Ciclo Formación Profesional Superior de
Administración de Sistemas informáticos en Red

Pedro Vicente López Bañón

Tutor: Francisco Javier López Mota

Proyecto final Ciclo Formación Profesional Superior de
Administración de Sistemas informáticos en Red

Despliegue de Arquitectura de nube con OpenStack

Pedro Vicente López Bañón ^{1 2}

Junio 2018

¹Tutor: Francisco Javier López Mota

²I.E.S. José Luís Castillo Puche

PROYECTO FINAL DE CICLO DE FORMACIÓN PROFESIONAL SUPERIOR DE ADMINISTRACIÓN DE SISTEMAS INFORMÁTICOS EN RED.

Este Proyecto final fue realizado bajo la supervisión del profesor Francisco Javier López Mota y el soporte del I.E.S. José Luís Castillo Puche de Yecla durante marzo, abril, mayo y junio de 2018.

Edición única, Junio 2018



https://github.com/pvlopez/Proyecto_Final_FPS_ASIR_Pedro_V_Lopez_Banon

©2018 * PEDRO VICENTE LÓPEZ BAÑÓN

pedrovicenteloba@gmail.com

ESTA OBRA ESTÁ SUJETA A LA LICENCIA RECONOCIMIENTO 4.0 INTERNACIONAL DE CREATIVE COMMONS.



PARA VER UNA COPIA DE ESTA LICENCIA, VISITE: <https://creativecommons.org/licenses/by-sa/4.0/legalcode>

*”La única manera de hacer
un trabajo genial es amar lo
que haces.”*

Steve Jobs

Sinópsis

Resumen

En el desarrollo de este proyecto se ha realizado una completa implementación de un cloud computing utilizando el sistema de OpenStack sobre CentOS. He realizado un pequeño estudio de las diferentes opciones de implementación de sistemas de nube. Durante el informe se explican algunas de las partes principales del sistema OpenStack y más tarde, se realiza una implementación multinodo y sus pasos con RDO Packstack.

Abstract

In this work I have developed a complete implementation of a cloud computing system using OpenStack Platform over CentOS. I have perform a small study of the different options of cloud systems. The report indexes some of the main parts and services of the OpenStack cloud and later, uses RDO Packstack to perform a multi-node deployment..

Índice general

Glosario	11
Siglas	13
Índice de figuras	15
Índice de cuadros	17
1 Introducción (CE 1,2)	19
1.1 Motivación	19
1.2 La computación en la nube	20
1.3 Tipos de nube	21
1.3.1 Tipos de servicio	21
1.4 Objetivo del proyecto	22
1.5 Porqué OpenStack	22
1.5.1 Historia y alcance de OpenStack	23
2 Punto de partida y planificación inicial (CE3,4,7,8)	25
2.1 Situación de partida	25
2.2 Recursos disponibles	25
2.2.1 Recursos de Hardware	25
2.2.2 Recursos de software	26
2.2.3 Subvenciones y ayudas	26
2.3 Entorno de desarrollo	26
2.3.1 Oracle VirtualBox	26
2.3.2 CentOS 7	27

2.3.3	Python	27
2.3.4	Git y Github	27
2.3.5	IDEs y editores	28
2.3.6	Control de proyectos	28
2.3.7	Documentación	29
3	OpenStack	31
3.1	Introducción a OpenStack	31
3.2	Componentes básicos de OpenStack	31
3.2.1	Servicio de identidad (Keystone)	31
3.2.2	Servicio de cómputo (Nova)	33
3.2.3	Servicio de red (Neutron)	34
3.2.4	Servicio de imagen (Glance)	34
3.2.5	Servicio de almacenamiento de objetos (Swift)	35
3.2.6	Servicio de almacenamiento en bloque (Cinder)	36
3.2.7	OpenStack dashboard (Horizon)	36
3.3	Componentes adicionales de OpenStack	36
3.3.1	Servicio de orquestación (Heat)	36
3.3.2	Servicio de telemetría (Ceilometer)	37
3.3.3	Servicio de bases de datos (Trove)	37
3.3.4	Servicio de procesamiento de datos (Sahara)	38
3.3.5	Otros componentes	38
3.4	OpenStack en entornos de producción	39
3.4.1	RDO	39
3.4.2	Triple-O	39
3.4.3	Otros	40
4	Seguridad y Notas Legales (CE 11)	43
5	Escenario y configuración inicial (CE 6, 7, 8)	45
5.1	Configuración de red	45
5.2	Preconfiguración de los equipos	45
5.2.1	Configurar autenticación sin contraseña en el nodo controller	46
6	Instalación de OpenStack (CE 10)	47
6.1	Requisitos	47
6.2	Añadir los repositorios de RDO	47
6.2.1	Instalar OpenStack	48
6.2.2	Instalación de OpenStack Sahara	50
7	Inicialización y puesta en marcha del sistema (CE 14,15, 16) ..	55
7.1	Login	55
7.2	Usuarios y grupos	55

7.3	proyectos	56
7.3.1	Instancias	56
7.3.2	imagenes	57
7.3.3	volumenes	58
7.3.4	redes	58
8	Línea temporal	63
8.1	Línea temporal (CE 12)	63
9	Epílogo	65
9.1	Conclusión (CE 15)	65
10	Anexos (CE 13)	69
10.1	Anexo A	69
10.2	Anexo B	72

Glosario

Big Data Big data, macrodatos, datos masivos, inteligencia de datos o datos a gran escala es un concepto que hace referencia a un conjunto de datos tan grandes que aplicaciones informáticas tradicionales de procesamiento de datos no son suficientes para tratar con ellos y los procedimientos usados para encontrar patrones repetitivos dentro de esos datos. Los textos científicos en español con frecuencia se usa directamente el término en inglés big data, tal como aparece en el ensayo de Viktor Schönberger: La revolución de los datos masivos.[14]. 19

Business Intelligence Se denomina inteligencia empresarial, inteligencia de negocios o BI (del inglés business intelligence), al conjunto de estrategias, aplicaciones, datos, productos, tecnologías y arquitectura técnicas, los cuales están enfocados a la administración y creación de conocimiento sobre el medio, a través del análisis de los datos existentes en una organización o empresa.[13]. 19

CentOS 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. Desde la versión 5, cada lanzamiento recibe soporte durante diez años, por lo que la actual versión 7 recibirá actualizaciones de seguridad hasta el 30 de junio de 2024.[5]. 22

Ciencia de datos La ciencia de datos es un campo interdisciplinario que involucra métodos científicos, procesos y sistemas para extraer conocimiento o un mejor entendimiento de datos en sus diferentes formas, ya sea estructurados o no estructurados, lo cual es una continuación de algunos campos de análisis de datos como la estadística, la minería de datos, el aprendizaje automático y la analítica predictiva.[6]. 19

fully qualified domain name Un FQDN (sigla en inglés de fully qualified domain name) es un nombre que incluye el nombre de la computadora y el nombre de dominio asociado a ese equipo. Por ejemplo, dada la computadora llamada «serv1» y el nombre de dominio «bar.com.», el FQDN será «serv1.bar.com.»; a su vez, un FQDN asociado a serv1 podría ser «post.serv1.bar.com.». En los sistemas de nombre de dominio de zonas, y más especialmente en los FQDN, los nombres de dominio se especificarán con un punto al final del nombre, aunque se puede omitir.. 45

Instituto de Fomento de la Región de Murcia El info es más que una agencia de desarrollo

de la Comunidad Autónoma de la Región de Murcia. En 30 años cumplidos de historia, hemos sumado talento y experiencia al espíritu emprendedor de los ciudadanos para avanzar juntos en los nuevos retos de un entorno en constante evolución.. 26

Machine Learning El aprendizaje automático o aprendizaje de máquinas (del inglés, "Machine Learning") es el subcampo de las ciencias de la computación y una rama de la inteligencia artificial cuyo objetivo es desarrollar técnicas que permitan a las computadoras aprender. De forma más concreta, se trata de crear programas capaces de generalizar comportamientos a partir de una información suministrada en forma de ejemplos. Es, por lo tanto, un proceso de inducción del conocimiento. En muchas ocasiones el campo de actuación del aprendizaje automático se solapa con el de la estadística computacional, ya que las dos disciplinas se basan en el análisis de datos. Sin embargo, el aprendizaje automático también se centra en el estudio de la complejidad computacional de los problemas. Muchos problemas son de clase NP-hard, por lo que gran parte de la investigación realizada en aprendizaje automático está enfocada al diseño de soluciones factibles a esos problemas. El aprendizaje automático puede ser visto como un intento de automatizar algunas partes del método científico mediante métodos matemáticos.[4]. 19

OpenStack OpenStack es una plataforma de software de código abierto para nubes privadas y públicas. OpenStack, una plataforma que ofrece infraestructura como servicio (IaaS), permite a las empresas agregar servidores, y componentes de redes y de almacenamiento de manera fácil y eficiente a su nube.. 22, 23

Red Hat Enterprise Linux Red Hat Enterprise Linux también conocido por sus siglas RHEL es una distribución comercial de GNU/Linux desarrollada por Red Hat. Es la versión comercial basada en Fedora que a su vez está basada en el anterior Red Hat Linux, de forma similar a como Novell SUSE Enterprise (SUSE Linux Enterprise Desktop y SLE Server) lo es respecto de OpenSUSE o Mandriva Corporate respecto de Mandriva Linux One.[18]. 22

Replicación La replicación es el proceso de copiar y mantener actualizados los datos en varios modos de bases de datos ya sean estos persistentes o no. Éste usa un concepto donde existe un nodo amo o maestro (master) y otros sirvientes o esclavos (slaves).[1]. 19

Virtualización virtualización es la creación a través de software de una versión virtual de algún recurso tecnológico, como puede ser una plataforma de hardware, un sistema operativo, un dispositivo de almacenamiento u otros recursos de red. Dicho de otra manera, se refiere a la abstracción de los recursos de una computadora, llamada Hypervisor o VMM (Virtual Machine Monitor) que crea una capa de abstracción entre el hardware de la máquina física (host) y el sistema operativo de la máquina virtual (virtual machine, guest), dividiéndose el recurso en uno o más entornos de ejecución.[wiki-virtualizacion]. 19

Siglas

- aaS** as a Service. 19
- AMQP** Advanced Message Queuing Protocol. 31
- AWS** Amazon Web Services. 26
- CLI** Command Line Interface. 50, 55
- CMDB** configuration management database. 36
- DHCP** Dinamic Host Configuration Protocol. 19
- FTP** File Transfer Protocol. 19
- IaaS** Infrastructure. 20, 21, 23
- IoT** Internet of Things. 20
- PaaS** Platform as a Service. 20, 21
- REST** Representation State Transfer. 31
- SaaS** Software as a Service. 20, 21
- Triple-O** OpenStack On OpenStack. 39

Índice de figuras

1.1	Hadoop y su simpático logo, permite a las aplicaciones trabajar con miles de nodos y petabytes de datos. http://hadoop.apache.org/	22
1.2	OpenStack, https://www.openstack.org/	22
1.3	Logos de NASA y Rackspace.	23
2.1	Captura de pantalla del código L ^A T _E X en Overleaf de este documento.	29
3.1	Diagrama OpenStack.	32
3.2	OpenStack Keystone	32
3.3	Flujo de peticiones de RabbitQM.	33
3.4	Diagrama de Neutron.	34
3.5	Estados de una imagen de Glance.	35
3.6	Esquema de Heat.	37
3.7	Sahara: Big Data en OpenStack	38
3.8	Esquema de OpenStack Sahara.	39
3.9	Mapa conceptual de servicios de OpenStack. fuente: openstack.org	40
3.10	Despliegue conceptual del Overcloud desde el Undercloud.	40
6.1	Procesos durante la instalación de OpenStack. Fuente propia.	51
6.2	Vista de recursos y procesos con htop durante la instalación. Pueden apreciarse los distintos servicios de OpenStack ejecutandose. Fuente propia.	52
6.3	Confirmaciones correctas durante la instalación de OpenStack. fuente propia.	53
6.4	Dependencias durante la instalación de sahara. fuente propia.	53
6.5	Instalación de sahara. fuente propia.	54
7.1	Archivo keystone_admin. fuente propia.	55
7.2	Login de Horizon. fuente propia.	56
7.3	Panel de usuarios. fuente propia.	57
7.4	Creación de un nuevo grupo desde Horizon Identity. fuente propia.	57
7.5	Comprobación de grupos creados. fuente propia.	58
7.6	Panel de proyectos. fuente propia.	58

7.7	Momento en el que están levantandose 10 instancias en bloque sobre imagen debian [8]. fuente propia.	59
7.8	Panel de control de compute. fuente propia.	59
7.9	Panel de sabores. fuente propia.	60
7.10	Panel de control de compute. fuente propia.	60
7.11	Panel de volúmenes. fuente propia.	60
7.12	Propiedades de un router creado.fuente propia.	61
7.13	Redes. fuente propia.	61

Índice de cuadros

2.1	Especificación de los PCs prestados por el I.E.S. Castillo Puche y mi ordenador personal.	26
5.1	Especificaciones de los valores de red en los dispositivos del proyecto.	45
8.1	Tabla de Disposición temporal.	63

1. Introducción (CE 1,2)

1.1 Motivación

Uno de los grandes retos, sin duda, a los que se enfrentan en la actualidad los profesionales de las tecnologías de la información, es la dificultad de implementar entornos de producción de forma rápida y efectiva, adquiriendo equipos o infraestructuras, ajustando presupuestos de material y personal, resolviendo problemas de conectividad o incompatibilidad de software o hardware y lo más importante, quitando tiempo al desarrollo en sí del núcleo del proyecto. En adición, con la generación tan inmensa de datos que se generan cada día y la necesidad que existe por analizarla a tiempo real con la ambición de dar una mejor experiencia al usuario, la dificultad de implementar plataformas capaces de utilizar la ciencia de datos [6], el Machine Learning o la Replicación y almacenamiento masivo en lagos de datos requiere de una gran infraestructura que, físicamente, es difícil implementar.

La necesidad de abordar estos problemas hizo surgir el concepto de nube, la capacidad de separar la infraestructura y configuración "física" de la arquitectura conceptual, dando alta disponibilidad multiplataforma y dando a los desarrolladores una gran ventaja: dejar de preocuparse tanto por la implementación física y utilizando el tiempo para el núcleo del proyecto. Sin duda la Virtualización es uno de los pilares de este desarrollo, ofreciendo abstracción y orquestación de servicios, haciendo que con unas pocas líneas de código sea posible implementar cientos o miles de máquinas virtuales, con sus propias redes virtualizadas (switches, routers, IP's flotantes, SO, librerías preinstaladas, servicios de servidor web, File Transfer Protocol (FTP), Dinamic Host Configuration Protocol (DHCP) y un largo etcétera), consiguiendo en cuestión de minutos tener una arquitectura base para cualquier proyecto.

Hacia ya un tiempo que me había interesado en el tema de la implementación de nubes, infraestructura de clusters, bare metal, contenedores y los distintos conceptos de as a Service (aaS), y sin duda mis conocimientos matemáticos junto con los informáticos me llevaron inevitablemente a introducirme en el mundo del Big Data, Ciencia de datos, Machine Learning y Business Intelligence. Por ello tomé la decisión de sumergirme de cabeza en el mundo de la implementación de arquitectura de nubes y Virtualización, eligiendo este tema inmenso, arduo y sin duda maravilloso.

1.2 La computación en la nube

Según [9] la Computación en la Nube (Cloud computing) es el desarrollo y la utilización de capacidad de procesamiento computacional basado en Internet, denominado habitualmente por "la nube.^{es}", dicho de otra forma, tomar los equipos que están conectados a una red (ordenadores, servidores, smartphones, dispositivos Internet of Things (IoT), etc.), en este caso la mayor de todas "Internet", y hacer que trabajen todos juntos como un mismo organismo que, independientemente de sus partes, trabaja dinámicamente y desde cualquier lugar.

Según IBM [20], la computación en la nube proporciona una serie de ventajas:

- *Flexibilidad:* Los usuarios pueden escalar los servicios para ajustarlos a sus necesidades, personalizar las aplicaciones y acceder a los servicios cloud desde cualquier sitio, con una conexión a Internet.
 - *Escalabilidad:* La infraestructura cloud se escala on demand para dar soporte a las cargas de trabajo fluctuantes.
 - *Opciones de almacenamiento:* Los usuarios pueden elegir soluciones de almacenamiento público, privado o híbrido, en función de las necesidades de seguridad y otras consideraciones.
 - *Opciones de control:* Las organizaciones pueden determinar su nivel de control con opciones como servicio, las cuales incluyen Software as a Service (SaaS), Platform as a Service (PaaS) e Infrastructure (IaaS).
 - *Selección de herramientas:* Los usuarios pueden seleccionar entre un menú de herramientas y características predefinidas para crear una solución que se adapte a sus necesidades específicas.
 - *Características de seguridad:* Cloud privado virtual, cifrado y claves de API ayudan a proteger los datos.
- *Eficiencia:* Los usuarios empresariales pueden lanzar sus aplicaciones al mercado rápidamente, sin preocuparse por los costes de infraestructura subyacentes o el mantenimiento.
 - *Accesibilidad:* Las aplicaciones basadas en cloud y los datos son accesibles desde prácticamente cualquier dispositivo conectado a Internet.
 - *Acelerar la comercialización:* El desarrollo en cloud permite a los usuarios lanzar sus aplicaciones al mercado rápidamente.
 - *Seguridad de datos:* Los fallos de hardware no provocan la pérdida de datos gracias a las copias de seguridad en red.
 - *Ahorro en el equipo:* Cloud computing utiliza recursos remotos, de esta manera las organizaciones se ahorran el coste de los servidores y otros equipos.
 - *Estructura de pago:* Una estructura de pago "utility" significa que los usuarios solo pagan por los recursos que utilizan.

- *Valor estratégico:* Los servicios cloud dan a las empresas una ventaja competitiva a través de la tecnología más innovadora disponible.
 - *Trabajo optimizado:* Los proveedores de servicios cloud (CSP) gestionan la infraestructura subyacente, permitiendo a las organizaciones centrarse en el desarrollo de aplicaciones y otras prioridades.
 - *Actualizaciones regulares:* Los proveedores de servicios actualizan las soluciones con regularidad para ofrecer a los usuarios la tecnología más actualizada.
 - *Colaboración:* El acceso mundial significa que los equipos pueden colaborar desde muy diversas ubicaciones.
 - *Ventaja competitiva:* Las organizaciones se mueven con mayor agilidad que la competencia, que tiene que destinar los recursos de TI a la gestión de la infraestructura.

Debido a la gran cantidad de opciones que se pueden construir con el cloud computing, es habitual clasificar las nubes según los servicios o el dominio para el que tienen fin, esto lo vemos en el apartado ??

1.3 Tipos de nube

Los distintos modelos de cloud computing pueden dividirse en dos subgrupos bien por tipo de servicio o por otra parte por disponibilidad y privacidad.

1.3.1 Tipos de servicio

Los proveedores de Cloud ofrecen una serie de servicios adicionales que ajusta la oferta a las necesidades del cliente. En función de lo completa que sea la solución que se ofrece tenemos Infraestructura como Servicio, Plataforma como Servicio o Software como Servicio:

- *SaaS:* Podemos hablar de este sistema cuando el usuario se encuentra con todas las herramientas para implementar todos los procesos necesarios para la empresa.
- *IaaS:* No existe valor añadido, el proveedor ofrece capacidad de almacenamiento y proceso en bruto, siendo el usuario el que construye las aplicaciones que necesita desde cero.
- *PaaS:* En este caso, se ofrece herramientas y utilidades para facilitar la construcción de aplicaciones, como puede ser base de datos o entornos de programación.

según [19].

También se suelen clasificar por:

- *Nube Pública:* Un servicio de Nube Pública es cuando el proveedor proporciona sus recursos de forma abierta a todas las entidades que lo deseen, desde particulares a grandes corporaciones. Este tipo de servicios son los que ofrecen Amazon, Azure de Microsoft o Google Engine.
- *Nube Privada:* En la otra parte de la balanza se encuentra el Cloud Privado, que es una

cuando el proveedor realiza la implementación y administración del sistema para la entidad que forma parte de ella. Las entidades que optan por este tipo de sistemas son aquellas que tienen un alto nivel de complejidad y necesitan centralizar sus recursos, como pueden ser grandes corporaciones o administraciones públicas. El sistema que más está destacando para este tipo de servicios es Openstack, solución OpenSource.

- *Nube Híbrida:* Como su propio nombre indica esta solución esta compuesta por las dos anteriores , donde una parte de los servicios y la información se ofrece de manera pública y otra de manera privada. Este tipo de soluciones tienen mucho potencial, ya que permiten hacer crecer tu sistema contratando a terceros lo que vayas necesitando. Este tipo de servicios también se pueden realizar con Openstack.

1.4 Objetivo del proyecto

El objetivo de este proyecto es el realizar la implantación de una arquitectura de Nube, concretamente de OpenStack, que es sin duda kit de soluciones de cloud computing más expandido a nivel mundial. Dicha implementación se decide realizarla a través del sistema operativo CentOS en su versión 7, perteneciente a la familia de distribuciones de Red Hat Enterprise Linux.

La instalación y configuración debe ser completamente funcional y permitir el levantamiento de instancias, redes y todos los recursos básicos de openstack.

Como segundo objetivo, limitado por el tiempo, se quiere implementar el servicio de análisis de datos de openstack SAHARA, para su utilización en desarrollo de cluster de Hadoop.

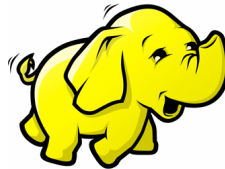


Figura 1.1: Hadoop y su simpático logo, permite a las aplicaciones trabajar con miles de nodos y petabytes de datos.<http://hadoop.apache.org/>

1.5 Porqué OpenStack



Figura 1.2: OpenStack, <https://www.openstack.org/>

Según Rackspace (uno de los fundadores del proyecto) en [17], OpenStack es:

®OpenStack es una plataforma de software de código abierto para nubes privadas y públicas. OpenStack, una plataforma que ofrece IaaS, permite a las empresas agregar servidores, y componentes de redes y de almacenamiento de manera fácil y eficiente a su nube.

Elegí OpenStack precisamente por ser de código abierto, lo que ha convertido al proyecto en uno de los que mayor desarrollo tienen a nivel mundial, con una inmensa cantidad de contribuidores, tanto particulares como empresas u organismos públicos. La documentación sobre el proyecto es extensa, y tal es la capacidad de desarrollo que sale una versión nueva cada seis meses. Puedes implementarlo sobre casi cualquier sistema con las distintas distribuciones y soportes que existen, o directamente en Bare metal a través de herramientas de orquestación e implementación multinodo.

1.5.1 Historia y alcance de OpenStack

Rackspace y la NASA lanzaron juntos OpenStack como un proyecto de software de nube de código abierto en 2010. Hoy en día, OpenStack es administrado por la organización sin fines de lucro OpenStack Foundation. Rackspace es un socio platino de la organización y está muy involucrado en la comunidad OpenStack.



Figura 1.3: Logos de NASA y Rackspace.

OpenStack es la plataforma de cómputo en la nube predilecta de las mayores empresas del mundo. Esta plataforma impulsa el sitio de comercio electrónico de Walmart, que recibe más de 80 millones de visitas cada mes. CERN, una organización europea de investigación, también utiliza OpenStack para su nube privada: la mayor nube de OpenStack del mundo, que cuenta con casi 200,000 núcleos.

2. Punto de partida y planificación inicial (CE3,4,7,8)

2.1 Situación de partida

El comienzo del proyecto fue un tanto desbordante. Hay una gran cantidad de documentación y recursos sobre OpenStack en la nube, como no podía ser de otra forma. Buscar entre tanta cantidad de información y aclararme las ideas sobre qué era OpenStack implicó muchas horas de estudio, leyendo documentación en inglés y aprendiendo nuevos conceptos. En definitiva, OpenStack se presentó ante mí como todo un universo lleno de posibilidades.

Una vez hube investigado lo suficiente - máquinas necesarias, redes, presupuesto, recursos de todo tipo - comenté la disponibilidad de recursos con los que podía contar por parte del centro educativo I.E.S. José Luís Castillo Puche, pensando en hacer un desarrollo multinodo necesitaba por lo menos un par de nodos de computación (ordenadores), y algún dispositivo de interconexión. Gracias a la ayuda de los profesores y de mi tutor conseguí un par de equipos y un router, con los que hacer las pruebas de desarrollo necesarias.

Mis recursos personales, por otra parte, no resultaban significativos para el desarrollo de este proyecto, pudiendo aportar un viejo portátil como nodo de la nube.

De entre todos los sistemas que valoré utilizar (Ubuntu, debian, fedora, windows, mac, Archlinux..) al final me decidí a utilizar CentOS Linux en su versión 7, puesto que tiene un muy buen soporte de OpenStack a través de RDO.

2.2 Recursos disponibles

2.2.1 Recursos de Hardware

Dentro de los recursos de hardware que tenía a mi disposición contaba con dos equipos del instituto con la disposición que se encuentra en 2.1 junto con un viejo portátil.

También se me procuró un router Linksys que se utilizará durante el proyecto como switch y como puerta de enlace.

	PC-1	PC-2	portátil
CPU	Intel-i5	Intel-i5	Intel Core 2
RAM	12Gb	4Gb	2 Gb
DISCO	1TB	1TB	120 GB

Cuadro 2.1: Especificación de los PCs prestados por el I.E.S. Castillo Puche y mi ordenador personal.

Para las conexiones a través de cable Ethernet, utilicé un rollo de cable de cat.6 S/FTP, y crimpé personalmente todos los cables de conexión de la red.

2.2.2 Recursos de software

Este proyecto está basado en OpenSource, por tanto todos los recursos de software a utilizar son aquellos que estén disponibles para su libre distribución.

Tanto OpenStack, CentOS, RDO y demás software utilizado para el proyecto está sacada directamente de fuentes de libre distribución.

También se cuenta como recurso Amazon Web Services (AWS) en donde tengo un dominio y suelo utilizar instancias ec2. Debido a que es un coste adicional, se intentará no utilizar mucho, solo en pruebas puntuales si fuera necesario.

2.2.3 Subvenciones y ayudas

Dentro de este apartado no puedo especificar mucho más allá de la ayuda aportada por el centro. A pesar de buscar ayudas o subvenciones para el proyecto en plataformas como el Instituto de Fomento de la Región de Murcia, distintos ministerios y/o Consejerías, la gran mayoría no encajaba con las necesidades del proyecto, estando entre los criterios de concesión, criterios tales como: "ser autónomo"; "ser desempleado"; "investigador" u otras en las que no encajaba mi perfil. Otras por el contrario, tenían un calendario no compatible con el desarrollo de este proyecto.

2.3 Entorno de desarrollo

Durante el desarrollo del proyecto he utilizado una serie de herramientas transversales para llevar a cabo cada una de las partes. Se especifican a continuación:

2.3.1 Oracle VirtualBox

Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64. Actualmente es desarrollado por Oracle Corporation como parte de su familia de productos de virtualización. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, conocidos como «sistemas invitados», dentro de otro sistema operativo «anfitrión», cada uno con su propio ambiente virtual. Entre los sistemas operativos soportados (en modo anfitrión) se encuentran GNU/Linux, Mac OS X, OS/2 Warp, Microsoft Windows, y Solaris/OpenSolaris, y dentro de ellos es posible virtualizar los sistemas operativos FreeBSD, GNU/Linux, OpenBSD, OS/2 Warp, Windows, Solaris, MS-DOS y muchos otros.

He utilizado este sistema para simular el desarrollo del proyecto, creando máquinas virtuales sobre un host. Requiere demasiados recursos como para tratarlo fuera de simples pruebas.

 Otra alternativa podría ser VMware, QEMU, Xen o PROXMOX

2.3.2 CentOS 7

CentOS (Community ENTERprise Operating System) es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux RHEL, compilado por voluntarios a partir del código fuente publicado por Red Hat, siendo la principal diferencia con este la eliminación de todas las referencias a las marcas y logos propiedad de Red Hat.

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. Desde la versión 5, cada lanzamiento recibe soporte durante diez años, por lo que la actual versión 7 recibirá actualizaciones de seguridad hasta el 30 de junio de 2024.

Red Hat Enterprise Linux se compone de software libre y código abierto, pero su compilación se distribuye a través de medios (CD-ROM o DVD-ROM) solamente a suscriptores de pago. Red Hat libera todo el código fuente del producto de forma pública bajo los términos de la Licencia pública general de GNU y otras licencias. Los desarrolladores de CentOS usan ese código fuente para crear un producto final que es compatible a nivel binario y funcional con Red Hat Enterprise Linux y está libremente disponible para ser descargado, utilizado y redistribuido libremente por el público, pero sin que sea mantenido directamente por Red Hat. CentOS es una de las distribuciones consideradas clones de Red Hat Enterprise Linux [5].

2.3.3 Python

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.[16]

2.3.4 Git y Github

■ *Git*:

Git (pronunciado "guit") es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

Al principio, Git se pensó como un motor de bajo nivel sobre el cual otros pudieran escribir la interfaz de usuario o front end como Cogito o StGIT. Sin embargo, Git se ha convertido desde entonces en un sistema de control de versiones con funcionalidad plena. Hay algunos proyectos de mucha relevancia que ya usan Git, en particular, el grupo de programación del núcleo Linux.

El mantenimiento del software Git está actualmente (2009) supervisado por Junio Hamano, quien recibe contribuciones al código de alrededor de 280 programadores. En cuanto a derechos de autor Git es un software libre distribuible bajo los términos de la versión 2 de la Licencia Pública General de GNU. [11]

■ **GitHub:**

GitHub es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de computadora. El software que opera GitHub fue escrito en Ruby on Rails. Desde enero de 2010, GitHub opera bajo el nombre de GitHub, Inc. Anteriormente era conocida como Logical Awesome LLC. El código de los proyectos alojados en GitHub se almacena típicamente de forma pública, aunque utilizando una cuenta de pago, también permite hospedar repositorios privados.

El 4 de junio de 2018, Microsoft compró GitHub por un monto de 7.500 millones de dólares, lo que hizo que muchos desarrolladores empezaran a migrar al competidor de código abierto GitLab. [12]

2.3.5 IDEs y editores

A la hora de manejar códigos de todo tipo en distintos lenguajes de programación (Shell Script, python, YAML, txt, etc.) me gusta utilizar el editor Atom. A pesar de ello cuando trabajas en terminal los recursos gráficos pasan a segundo plano y llegas a utilizar otros editores de terminal:

■ **nano:**

Este es un editor muy sencillo, bastante fácil de manejar si eres novicio en terminal, pero con la práctica puede volverse una herramienta fundamental.

■ **Vim:**

Vim sin duda es una verdadera "máquina" de teclear código en terminal. Tiene una curva de aprendizaje que no deja indiferente a nadie, pero su capacidad para adjuntarle plugins o sus capacidades avanzadas de manejo de código hacen a este editor una verdadera navaja suiza.

2.3.6 Control de proyectos

Todo proyecto debe llevar un control de las actividades que se realizan, bien por tener en cuenta todo lo que queda por hacer o simplemente por logística. Personalmente creo que trabajar con herramientas de control de proyectos hace que el trabajo en equipo sea mucho más fluido y que en todo momento sea mucho más sencillo determinar el estado actual de desarrollo.

Particularmente en este proyecto he utilizado Trello como herramienta de control para el proyecto. Es muy sencillo crear tareas y agregar comentarios, pasando de un estado a otro de una tarea con simplemente arrastrar en un panel gráfico. He utiliza esta herramienta porque para uso personal siempre le he sacado mucho rendimiento, pero para proyectos más avanzados hay plataformas mucho más avanzadas y adaptadas a ello , como JIRA.

2.3.7 Documentación

Esta documentación está realizada utilizando \LaTeX con un editor online llamado Overleaf <https://v2.overleaf.com> tal como se puede observar en la figura 2.1.

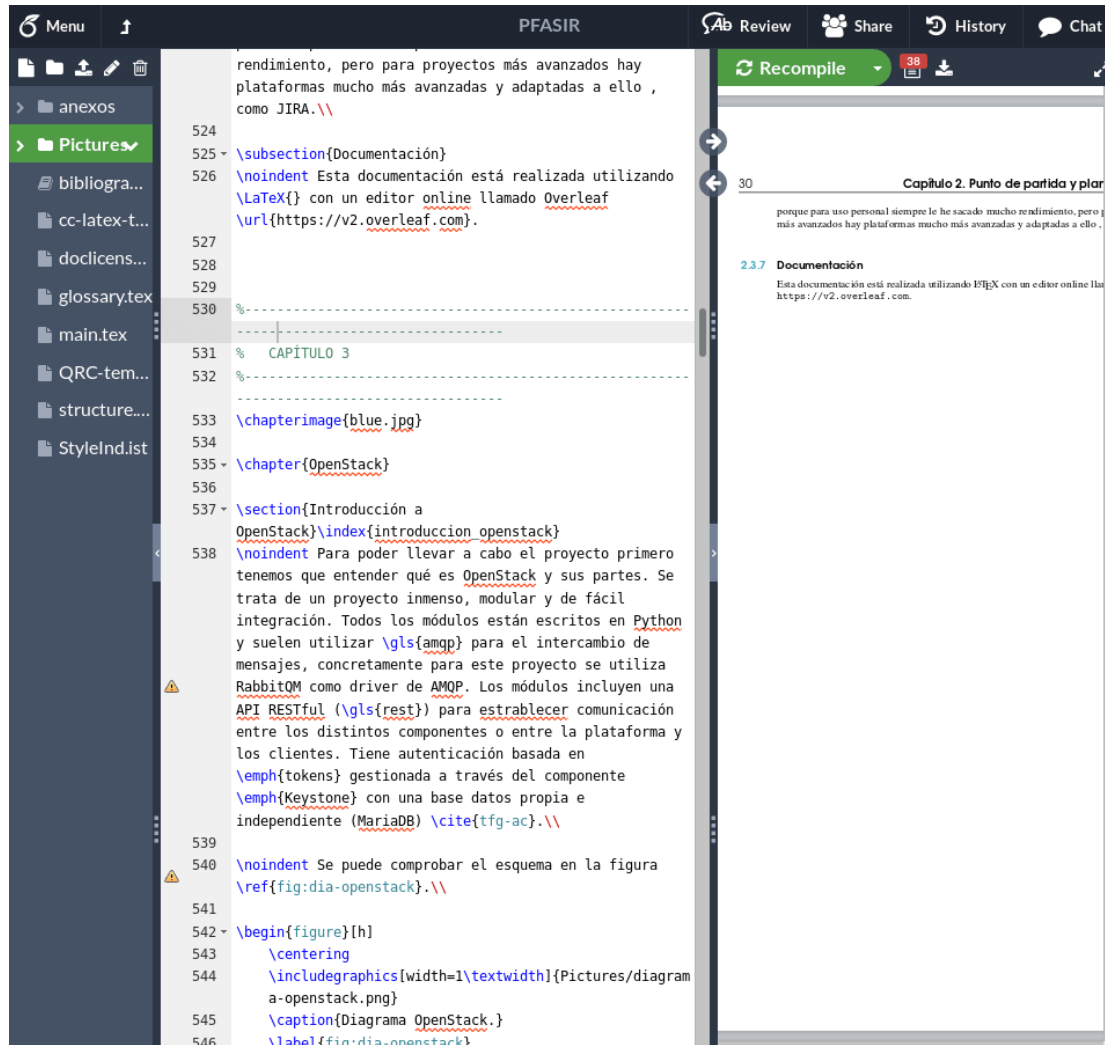


Figura 2.1: Captura de pantalla del código \LaTeX en Overleaf de este documento.

3. OpenStack

3.1 Introducción a OpenStack

Para poder llevar a cabo el proyecto primero tenemos que entender qué es OpenStack y sus partes. Se trata de un proyecto inmenso, modular y de fácil integración. Todos los módulos están escritos en Python y suelen utilizar Advanced Message Queuing Protocol (AMQP) para el intercambio de mensajes, concretamente para este proyecto se utiliza RabbitMQ como driver de AMQP. Los módulos incluyen una API RESTful (Representation State Transfer (REST)) para establecer comunicación entre los distintos componentes o entre la plataforma y los clientes. Tiene autenticación basada en *tokens* gestionada a través del componente *Keystone* con una base de datos propia e independiente (MariaDB) [9].

Se puede comprobar el esquema en la figura 3.1.

3.2 Componentes básicos de OpenStack

3.2.1 Servicio de identidad (Keystone)

KeyStone es el servicio de identidad que utiliza OpenStack para la autenticación, dicho sistema puede ser integrado con ldap, además puede ser usado a través de nombre de usuario y contraseña estándar, sistemas basados en tokens e inicios de sesión. Dispone de una API de identidad (actualmente la versión 3.0) la cual puede ser usada por herramientas de terceros para determinar la accesibilidad de los recursos. KeyStone nos va a permitir la autenticación y autorización entre los servicios, siendo instalado en el nodo Controlador, el cual suele usar como motor de base de datos tanto MySQL como MariaDB. [15] 3.2

Para entender cómo funciona KeyStone o Servicio de identidad debemos tener en cuenta los siguientes componentes:

- **Usuario:**
Pueden ser personas, sistemas o servicios que usen la nube de OpenStack, KeyStone será el encargado de validar las peticiones de éste. Los usuarios disponen de un login.
- **Credenciales:**
Es el dato que va a usar el usuario para su autenticación, puede ser un login y contraseña, un token, una clave de la API.
- **Autenticación:**

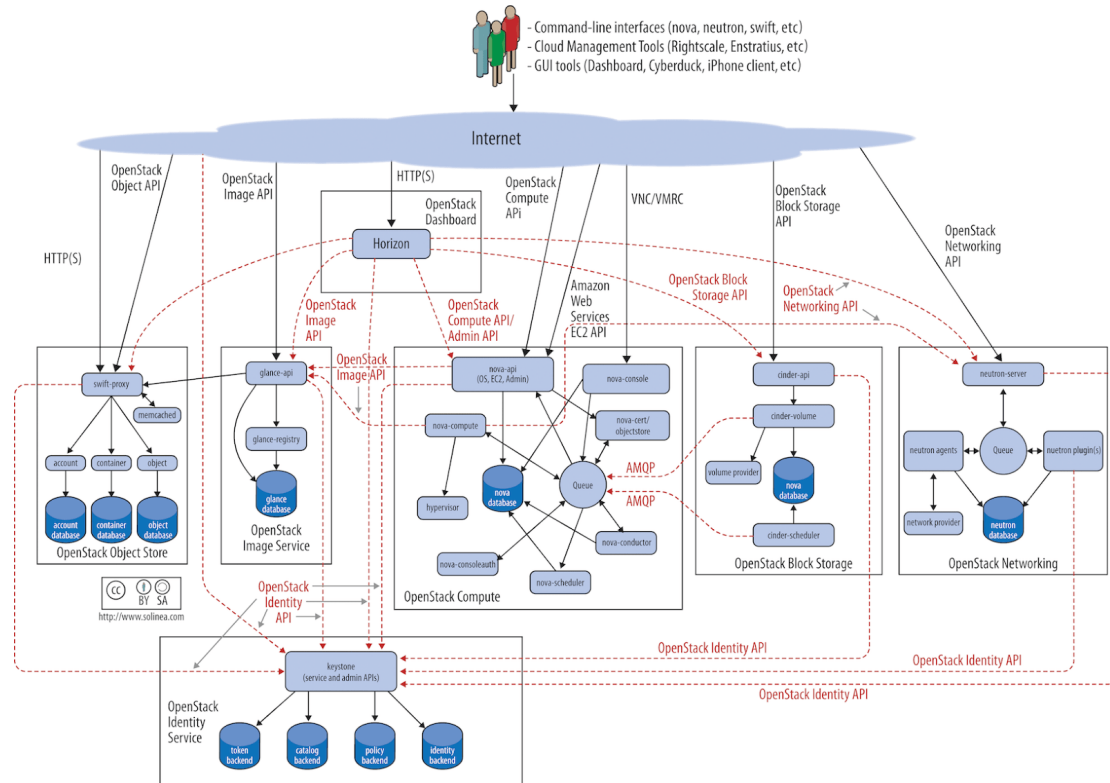


Figura 3.1: Diagrama OpenStack.



Figura 3.2: OpenStack Keystone

Es el acto de confirmar la identidad del usuario usando las credenciales provistas. Una vez validada la autenticación se provee de un token para ser usado en las peticiones del usuario, de esta forma no necesita autenticarse de nuevo.

- **Token:**
Es un bit de texto usado para acceder a los recursos, es generado cuando se valida la autenticación, cada token tiene un alcance el cual nos permite saber a qué recursos puede acceder un usuario.
- **Tenant:**
Es el contenedor usado para agrupar y aislar los recursos. Los tenants pueden ser clientes, organizaciones, cuentas o proyectos.
- **Servicio:**
Es cada uno de los servicios que conforman OpenStack, como pueden ser Computo (Nova), Almacenamiento (Swift) o Servicio de imágenes (Glance), nos va a proveer de los endpoints o URL para que los usuarios puedan acceder a los recursos.
- **Endpoint:**

Es el direccionamiento IP o URL para acceder a los servicios.

- **Roles:**

Para especificar o limitar las operaciones de usuarios KeyStone usa roles, los cuales contienen un grupo de privilegios para determinar el alcance del usuario.

3.2.2 Servicio de cómputo (Nova)

Nova es el componente principal del IaaS (infraestructura como servicio) y es el encargado de administrar los pools de recursos que tenemos disponibles, hay que tener en cuenta que Nova NO es un hypervisor, sino el gestor de recursos de hipervisores tales como: Xen, KVM o VMware. []

La arquitectura del nodo de cómputo esta compuesta de los siguientes servicios:

- **Servidor de API:**

Nos va a permitir la comunicación de los hipervisores con nuestro entorno, gracias a él los distintos fabricantes pueden integrar sus productos de forma eficaz.

- **Colas de mensajes (rabbit queues):**

Se encarga de la comunicación entre componentes (ya sean nodos de cómputos, componentes de red o API de fabricantes) a la hora de provisionar recursos.

- **Worker:**

Es el administrador de instancias en los hosts que nos va a permitir, crear, terminar, reiniciar instancias, añadir o quitar volúmenes.

- **Controlador de red:**

Podremos provisionar recursos de networking, como por ejemplo IP, vLAN o routers.

En la imagen 3.3 podéis observar como se realiza el flujo de peticiones:

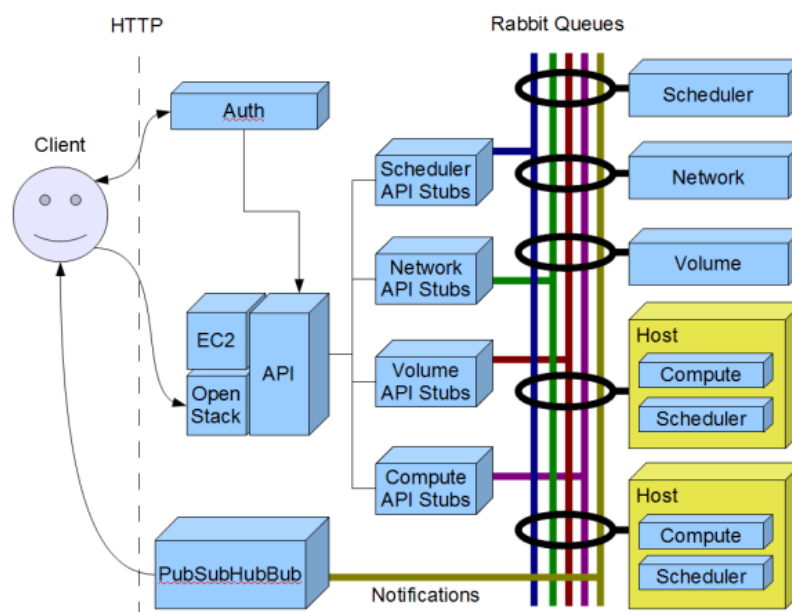


Figura 3.3: Flujo de peticiones de RabbitQM.

3.2.3 Servicio de red (Neutron)

El componente de administración de red en OpenStack es uno de los más complejos debido a la cantidad de posibilidades de configuración que tenemos. Neutron nos permite hacer uso de su API a través de plugins y agentes capaces de conectar desconectar puertos, provisionar redes y subredes, etc, todo ello vinculado al proveedor que vayamos a usar en nuestra nube, como por ejemplo switches físicos o virtuales de Cisco, productos de NEC OpenFlow, Open vSwitch, Linux bridging, Ryu Network Operating System o VMware NSX.

Los tres principales componentes de neutron son:

- **Neutron L3 agent.**
- **DHCP agent.**
- **Neutron plugin agent.**

En el esquema 3.4 podemos ver de forma representativa la comunicación que se realiza entre el servicio de administración de red y los nodos de una arquitectura de OpenStack. Hay que tener en cuenta que en el esquema anterior, se usa un nodo independiente para la administración de red.

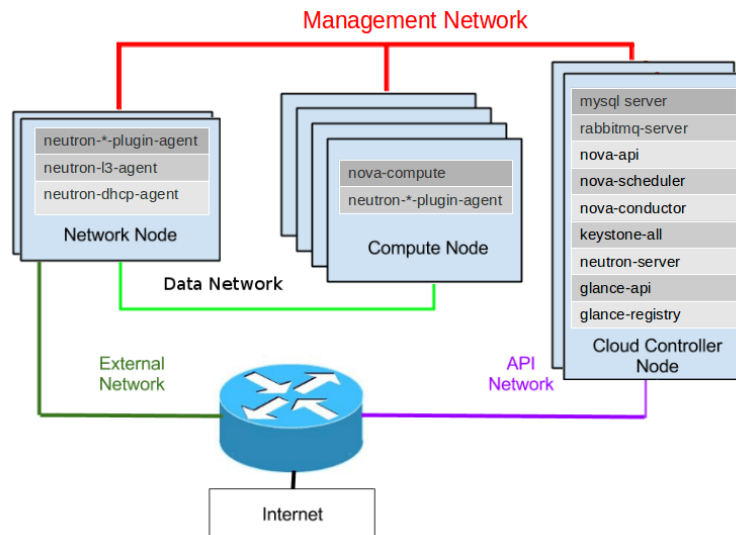


Figura 3.4: Diagrama de Neutron.

3.2.4 Servicio de imagen (Glance)

Glance es el servicio de OpenStack el cual permite registrar, descubrir y recuperar imágenes de máquinas virtuales para ser usadas en nuestro entorno de OpenStack. Dichas imágenes son almacenadas tanto en simples como en distribuidos filesystems y son identificadas de la siguiente forma:

```
<Glance Server Location>/v1/images/<ID>
```

Cuando usamos imágenes en Glance, ya sean para crear nuevas o eliminarlas, éstas pueden pasar por los siguientes estados:

- **Encolada - queued:** Quiere decir que hemos registrado una imagen nueva, asignado un identificador pero aún no ha sido subida a Glance.
- **Guardando - saving:** La imagen está siendo subida a Glance.
- **Activa - active:** La imagen está totalmente disponible para ser usada en Glance.

- **Abortada - killed:** Se produce cuando existe un error durante la subida de una imagen y ésta no es legible.
- **Borrada - deleted:** En este estado aún disponemos de la información de la imagen pero no podemos usarla ya que se encuentra eliminada.
- **Pendiente de borrado - pending_delete:** Estamos a punto de eliminar la imagen pero aún podemos recuperarla.

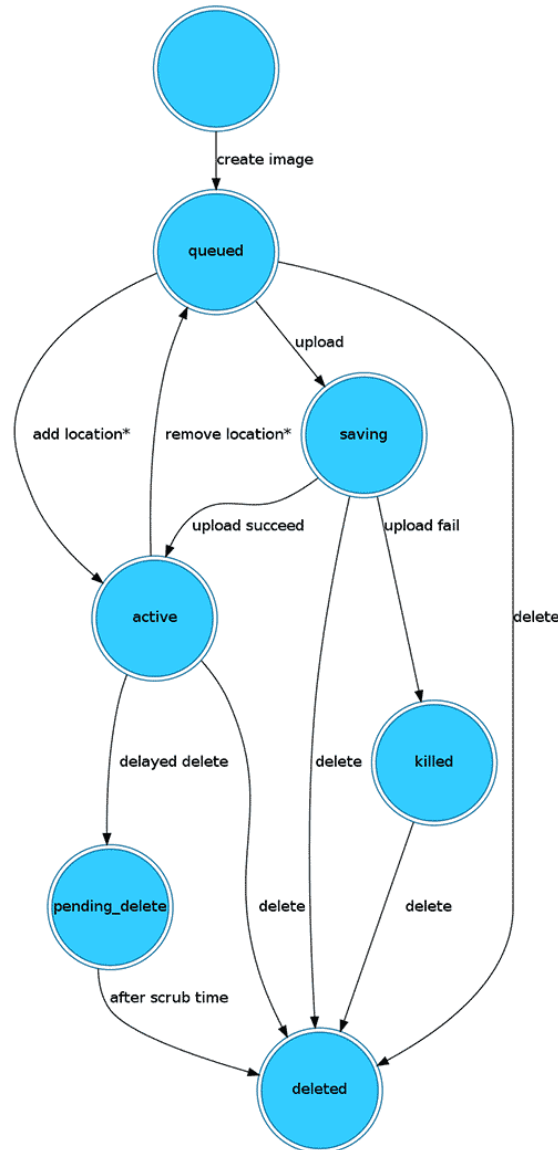


Figura 3.5: Estados de una imagen de Glance.

3.2.5 Servicio de almacenamiento de objetos (Swift)

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

discos duros y servidores de bajo costo pueden ser utilizados.

En agosto de 2009, Rackspace comenzó el desarrollo del precursor de OpenStack Object Storage, como un reemplazo completo para el producto Cloud Files. El equipo de desarrollo inicial consistió en nueve desarrolladores. SwiftStack, una compañía de software de almacenamiento de objetos, es actualmente el líder en el desarrollo de Swift.

3.2.6 Servicio de almacenamiento en bloque (Cinder)

OpenStack Block Storage (Cinder) proporciona dispositivos de almacenamiento a nivel de bloque persistentes para usar con instancias de OpenStack Compute. El sistema de almacenamiento de bloques gestiona la creación, aplicación y el desprendimiento de los dispositivos de bloque a los servidores. Volúmenes de almacenamiento de bloque se integran plenamente en OpenStack Compute y el Dashboard que permite a los usuarios en la nube gestionar sus propias necesidades de almacenamiento. Además del almacenamiento del servidor local de Linux, puede utilizar las plataformas de almacenamiento incluyendo Ceph, CloudByte, Coraid, EMC (VMAX y VNX), GlusterFS, Hitachi Data Systems, IBM Storage (familia Storwize, controlador de volumen SAN, XIV Storage System, y GPFS), Linux LIO, NetApp, Nexenta, Scalify, SolidFire, HP (StoreVirtual y 3PAR StoreServ familias) y almacenamiento puro. El almacenamiento de bloques es apropiado para escenarios donde el rendimiento es sensible, tales como el almacenamiento de base de datos, sistemas de archivos expandibles, o la prestación de un servidor con acceso al almacenamiento a nivel de bloque en bruto. La gestión Snapshot ofrece una potente funcionalidad para realizar copias de seguridad de los datos guardados en volúmenes de almacenamiento en bloque. Las instantáneas pueden restaurarse y utilizarse para crear nuevos volúmenes de almacenamiento en bloque.

3.2.7 OpenStack dashboard (Horizon)

Horizon es una interfaz web que permite a los usuarios administrar los servicios que ofrece OpenStack incluyendo Nova, Swift, Keystone.

Al igual que cualquier componente de OpenStack dispone de su propia API, la cual puede ser accedida de forma sencilla, una utilidad que se le puede dar al uso de las API de OpenStack, es la integración con tu propia configuration management database (CMDB) CMDB, cuando usuarios desplieguen instancias, volúmenes, etc. estos datos pueden ser capturados para la integración con la base de datos.

3.3 Componentes adicionales de OpenStack

3.3.1 Servicio de orquestación (Heat)

Heat es el orquestador de Openstack, y sirve para poder orquestar lanzamientos de varias cosas a la vez escribiendo una serie de código. Para hacerse una idea: podemos lanzar un proyecto entero de cloud (con sus redes, sus instancias, sus volúmenes, sus imágenes, etc.) con sólo ejecutar una pila (stack), es decir, con “ejecutar” un solo fichero.

Cómo funciona:

En las plantillas de Heat, se definen los recursos necesarios para lanzar la aplicación, estos son tomados de los servicios disponibles de Openstack. Heat maneja el ciclo de vida de la app y se proveen características como auto-escalabilidad. Dentro de la misma plantilla se puede definir que aplicaciones lanzar directamente con shell scripting o dejar que herramientas de administración, tales como Ansible o Puppet, se encarguen de ello.

Podemos observar un esquema de heat en la figura 3.6.

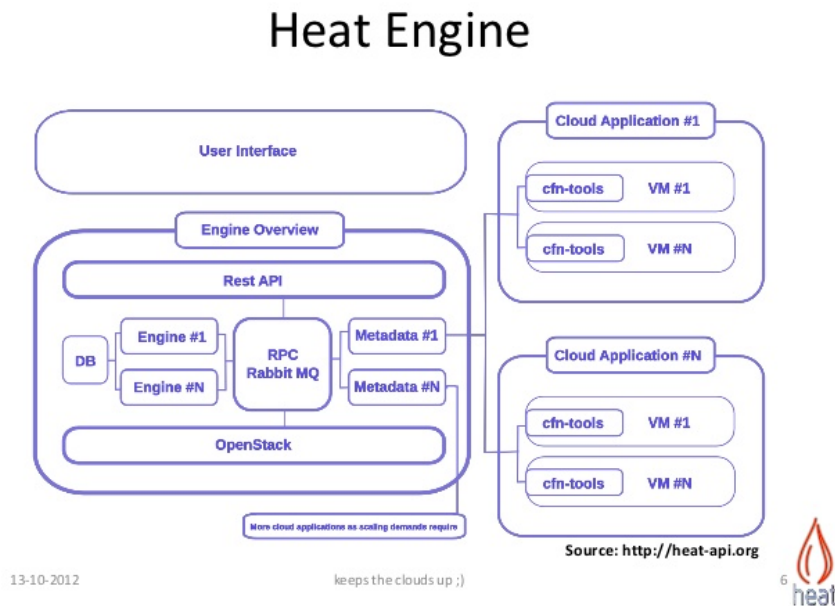


Figura 3.6: Esquema de Heat.

3.3.2 Servicio de telemetría (Ceilometer)

Ofrece servicios de telemetría con los que podremos monitorizar el uso de cada usuario en nuestra infraestructura, así como facturar individualmente por dicho uso.

3.3.3 Servicio de bases de datos (Trove)

Trove sienta las bases para el aprovisionamiento y gestión de bases de datos SQL y NoSQL. (MySQL, Percona Server, MariaDB, PostgreSQL, MongoDB, Cassandra, Couchbase y Redis), es decir, ejerce como director de bases de datos como servicio (DBaaS). [2]

El uso de DBaaS se sustenta en 5 pilares básicos:

- **Reducción de Costes:** Gracias a la estandarización se reducen los costes asociados a la gestión y mantenimiento de las diferentes plataformas. Porporcionando de forma integrada capacidades adicionales (backup, actualizaciones automaticas y gestión de parches, auto-reparación) que ahondan aún más esta reducción de costes.
- **Agilidad:** mayor velocidad de aprovisionamiento y por tanto menos esperas a la hora de entregar nuevos servicios. Tendremos un nuevo servicio de bbdd a tan solo unos clicks de distancia.
- **Facilitar el acceso a nuevas tecnologías:** NoSQL y NewSQL son el futuro de muchos de los entornos que a día de hoy trabajan con bbdd relacionales. El disponer de un entorno preconfigurado, estable y robusto puede ayudar a la adopción de nuevas herramientas que permitan un mayor crecimiento y una mejor prestación de servicios.
- **Mayor poder de innovación:** muy ligado al punto anterior tanto en cuanto va asociado a nuevas tecnologías y nuevas formas de afrontar la problematica del almacenamiento y sobre todo el acceso a la información.
- **Reducción de tiempos y esfuerzos en los pasos a producción:** la posibilidad de autoescalar y dotar de funcionalidades asociadas a la plataforma sobre la que se está trabajando permitirá una mayor agilidad a la hora de transformar proyectos de desa-

rollo a entornos productivos, a la vez que escalamos en capacidad y disponibilidad.

Más allá del aprovisionamiento básico, Trove automatiza la gestión del ciclo de vida de las instancias de base de datos. Cuando llega el momento de lanzar una copia de seguridad, este inicia el proceso y almacena la copia de seguridad en el almacén de objetos asociado a nuestra plataforma OpenStack. Un usuario más adelante podría lanzar una nueva instancia de base de datos a partir de esa copia de seguridad realizada previamente. Cuando un usuario desea cambiar el tamaño de la base de datos, ya sea cambiando los recursos de almacenamiento o de cómputo que se le asignen, Trove organiza todo el proceso, incluida la migración de datos.

3.3.4 Servicio de procesamiento de datos (Sahara)



Figura 3.7: Sahara: Big Data en OpenStack

OpenStack Sahara es un proyecto integrado en OpenStack cuyo objetivo es hacer elástico Hadoop bajo demanda. Hadoop es prácticamente el estándar cuando hablamos de Big Data, las organizaciones típicamente comienzan montando un cluster. Incluso con las distribuciones comerciales este proceso es complejo.

Sahara está preparado para configurar, autodesplegar y escalar clusters Hadoop sobre OpenStack. Sahara ofrece por tanto capacidades Elastic Data Processing (EDP) sobre Hadoop al estilo de Amazon Elastic MapReduce. Sahara permite:

- Crear clusters desde UI o bien integrar Sahara con tu aplicación vía el API.
- legir entre diversas distribuciones, incluidas Hortonworks Data Platform (HDP) y Cloudera Hadoop Distribution (CHD).
- Soporte de trabajos (jobs) de Apache Spark.
- Configurar HDFS de forma sencilla.

Sahara está integrado con los servicios core OpenStack como Keystone, Glance, Horizon y Nova, además soporta las APIs nativa OpenStack lo que permite a los usuarios provisionar entornos Hadoop desde el Horizon GUI. Referencia [3]

3.3.5 Otros componentes

Hay una infinidad más de componentes y módulos de servicio para OpenStack (figura 3.9) que no veremos en este documento, pero hay que destacar que crecen con cada nueva versión.

Para más información sobre módulos se puede consultar: <https://www.openstack.org/software/project-navigator>

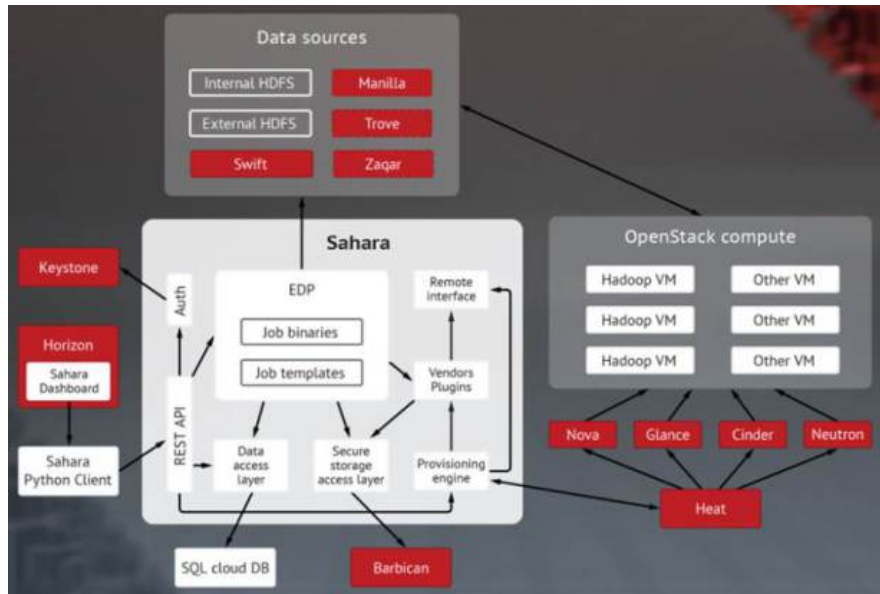


Figura 3.8: Esquema de OpenStack Sahara.

3.4 OpenStack en entornos de producción

3.4.1 RDO

RDO es una comunidad de personas que usan e implementan OpenStack en CentOS, Fedora y Red Hat Enterprise Linux. Desarrollan documentación para ayudar a comenzar, listas de correo donde puede conectarse con otros usuarios y paquetes compatibles con la comunidad de las versiones de OpenStack más actualizadas disponibles para su descarga. OpenStack se basa fundamentalmente en el sistema operativo y el hipervisor, y ¿qué mejor sistema operativo para construir que el sistema operativo líder de la industria? La comunidad RDO es el sitio comunitario integral para todo lo relacionado con el uso de OpenStack en las plataformas basadas en Red Hat.

El proyecto actual se ha desarrollado siguiendo en gran parte esta herramienta y la documentación relacionada que se puede encontrar en <https://www.rdo-project.org/use/> disponible en inglés.

3.4.2 Triple-O

Durante el proceso de diseño de una arquitectura de OpenStack uno de los puntos más importantes a tener en cuenta es el paso hacia la instalación del producto. Para ese paso en la realización del proyecto se utiliza Packstack, de RDO ??, el cual funciona perfectamente para una prueba de concepto y para empezar a conocer el uso y funcionamiento de OpenStack, pero a la hora de trasladar nuestra solución a un entorno más empresarial la primera pregunta que debemos plantearnos es: si yo estoy ofreciendo escalado horizontal a mis clientes de nube, ¿cómo puedo yo escalar horizontalmente mi infraestructura? (En otras palabras: ¿cómo puedo crecer mi capacidad de cómputo, almacenamiento y disponibilidad?).

Para resolver esta cuestión nació el proyecto OpenStack On OpenStack (Triple-O), una plataforma para instalar y administrar OpenStack desde una implementación de OpenStack directamente en baremetal. Es decir, implementar y gestionar una plataforma de producción OpenStack desde otra plataforma OpenStack (más sencilla) que actuará como controlador. El concepto puede resultar confuso al principio.

"Triple-O es la utilización de una infraestructura OpenStack auto-gestionada, es decir, OpenStack baremetal (nova y cinder) junto con Heat, diskimage-

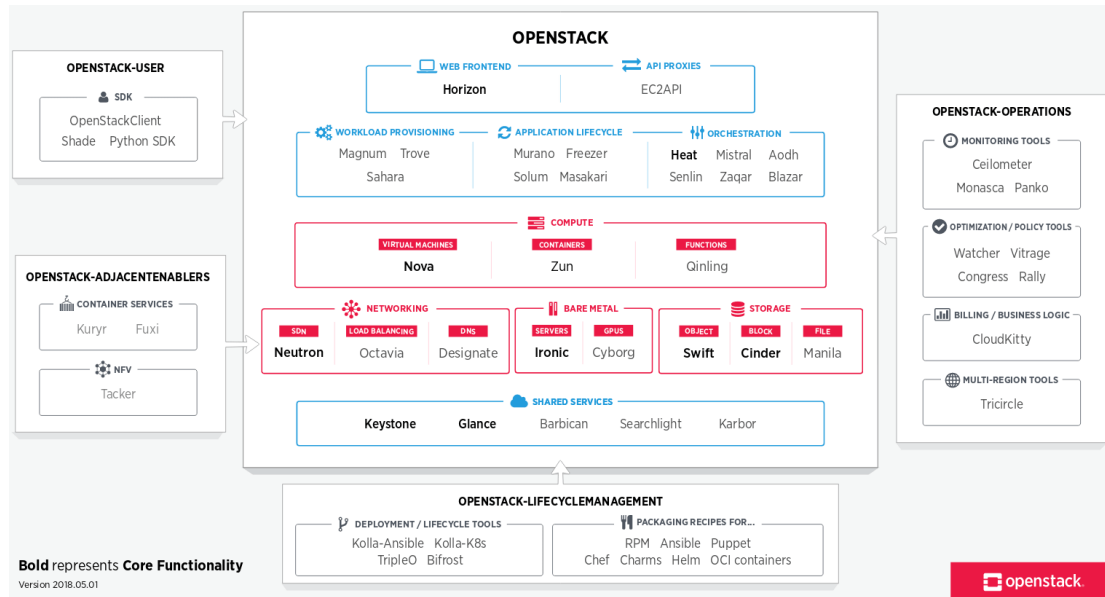


Figura 3.9: Mapa conceptual de servicios de OpenStack. fuente: openstack.org

builder más un orquestador de imágenes como Chef o Puppet para instalarlo, mantenerlo y actualizarlo.”

La estrategia utilizada para llevar a producción dichas implementaciones con Triple-O/Director es la siguiente:

- Implementar una maquina física/virtual que se encargara del despliegue de toda nuestra infraestructura de producción (con Packstack, por ejemplo). A este entorno más sencillo se le llama **undercloud**.
- Desplegar nuestro entorno de producción (llamado **Overcloud** desde el Undercloud).

A este proceso se le denomina *desplegar la Overcloud desde la UnderCloud*. En la figura 3.10 se puede observar. Simple, ¿Cierto?

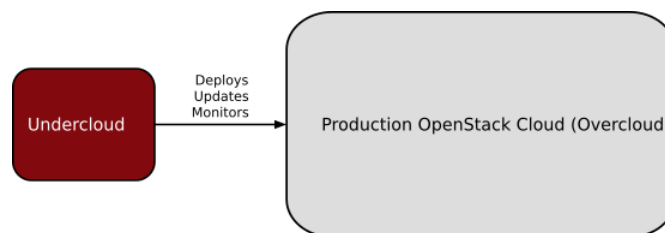


Figura 3.10: Despliegue conceptual del Overcloud desde el Undercloud.

3.4.3 Otros

OpenStack tiene muchos sabores, por así decirlo, debido a que es Open Source y cualquiera puede tomar el código fuente del sistema y hacer su propia implementación. La causalidad lleva por tanto a que surjan distintas distribuciones de OpenStack, está Conjure-up junto con MAAS de Ubuntu, RDO de Red Hat, Mirantis, DevStack, VMware OpenStack, distribuciones para Debian y un lista inmensa de implementaciones. Si tienes curiosidad en implementar OpenStack sobre un sistema determinado o simplemente te gustaría conocer las opciones que existen, te sugiero que eches un vistazo al marketplace de openstack desde donde podrás ver multitud de implementaciones:

<https://www.openstack.org/marketplace/distros/>

4. Seguridad y Notas Legales (CE 11)

La seguridad, el aislamiento y la implementación de políticas de firewall, proxy y de evasión de malware son temas troncales de la administración de OpenStack. La configuración de redes debe ser independiente y las instancias deben poder aislarse efectivamente unas de otras para asegurar la privacidad de los servicios dentro de la infraestructura.

Para este proyecto se han tenido en cuenta las especificaciones básicas de seguridad como la utilización del protocolo de Secure Shell con certificado ssl para la comunicación entre el servidor de despliegue y las máquinas clientes. Asegurar que los puertos del router no permiten las conexiones externas con internet, haciendo que la implementación del proyecto se constituya en una nube privada.

En OpenStack existen todo tipo de entidades como usuarios, grupos, áreas de disponibilidad, claves de seguridad, Roles de permisos y multitud de configuraciones destinadas a la seguridad y la privacidad. Destaco que buscando una implementación hacia la analítica de datos, es fundamental tratar este tema en profundidad en el futuro.

OpenStack implementa muchas de estas reglas de seguridad automáticamente y no permite controlar la configuración desde fuera de los elementos de openstack. Prueba de ello es la gran cantidad de cortafuegos y reglas de firewall con las que estuve varios días buscando configuraciones y no conseguí manualmente abrir los puertos para conseguir enrutar el acceso http para acceder al panel de control desde fuera del router. Seguiré intentándolo en el futuro cuando sean más profundos mis conocimientos sobre la arquitectura lógica de OpenStack.

Otra parte fundamental es llevar unas buenas prácticas en la configuración y almacenamiento de contraseñas y pares de claves, siendo necesario asegurar que no se encuentran al alcance de un ataque de cualquier tipo.

Para más información sobre la seguridad en OpenStack se puede encontrar más información en OpenStack Security Guide (HAVANA) [10] o en OpenStack Security guide Official [7].

5. Escenario y configuración inicial (CE 6, 7, 8)

5.1 Configuración de red

La red se divide en tres nodos u ordenadores, con un router haciendo de switch de red interna, con red ip 172.21.1.0/24. El router hace de punto de acceso y servidor dns. No es necesario configurar DHCP puesto que se va a trabajar con IP estática (aunque por seguridad se emplea DHCP y se hace reserva de IP para cada una de las máquinas dentro del rango). La configuración se puede observar en la tabla 5.1.

	IP	FQDN
PC1	172.21.1.1	viernes
PC1	172.21.1.51	controller.acendata.com
PC2	172.21.1.53	compute.acendata.com
Portatil	172.21.1.52	network.acendata.com

Cuadro 5.1: Especificaciones de los valores de red en los dispositivos del proyecto.

Los nombres Fully qualified domain name utilizados para la resolución DNS de la red (configuración aconsejable de packstack) están directamente relacionados con el propósito al que va dirigido el nodo en cuestión.

Se utiliza un nombre de dominio no existente de propia ocurrencia.

5.2 Preconfiguración de los equipos

Todos los equipos son primariamente formateados con el sistema CentOS 7 y se configuran las interfaces de red y nombres de hosts con los parámetros de la tabla 5.1.

Para ver los nombres de host tecleamos en terminal:

```
1 $ hostnamectl status
2
3 $ hostnamectl set-hostname <nombre>
```

En cada uno de los tres hosts a configurar entrar en el archivo */etc/hosts* e introducimos los siguiente parámetros:

```

1 172.21.1.51 controller.acendata.com controller
2 172.21.1.52 network.acendata.com network
3 172.21.1.53 compute.acendata.com compute

```

y guardamos.

Actualizamos el sistema y los repositorios.

```
1 $ yum -y update ; reboot
```

Tras el reinicio de todos los nodos. Deshabilitamos Selinux en los tres equipos. Para ello ejecutamos el comando:

```
1 $ setenforce 0
```

Si queremos que este cambio sea permanente tendremos que acceder al fichero */etc/sysconfig/selinux* con un editor y asegurar que el parámetro "SELINUX=" está con el valor "disabled".

Tras ello deshabilitamos NetworkManager y Firewalld, en los tres equipos. Con

```

1 $ systemctl stop NetworkManager
2
3 $ systemctl disable NetworkManager
4
5 $ systemctl stop firewalld
6
7 $ systemctl disable firewalld

```

5.2.1 Configurar autenticación sin contraseña en el nodo controller

Vamos a configurar la autenticación sin contraseñas a través de claves ssl desde el controller hacia los nodos compute y network.

Para ello nos situamos en el nodo controller únicamente, y ejecutamos:

```

1 ssh-keygen
2
3 ssh-copy-id -i /root/.ssh/id_rsa.pub root@172.21.1.52
4 ssh-copy-id -i /root/.ssh/id_rsa.pub root@172.21.1.53

```

Tras ello si no hay ningún problema, podremos hacer ssh ha los dos nodos y no nos pedirá contraseña.

```

1 ssh compute
2
3 ssh network

```

para asegurarnos pedimos el hostname dentro de la sesión ssh para comprobar que estamos dentro de las otras máquinas. Con esto ya tendríamos la preconfiguración de los equipos preparada y podríamos pasar a la instalación.

6. Instalación de OpenStack (CE 10)

6.1 Requisitos

La primera intención al realizar una implementación de OpenStack por primera vez, podría ser hacer la instalación de los componentes individualmente, instalando cada una de las dependencias y programas asociados en cada uno de los nodos. Sin embargo, para un novicio, esa tarea es sencillamente desbordante. Se necesita conocer el sistema en mayor profundidad para hacer una instalación "a pelo" de OpenStack.

Por esa razón decidí utilizar RDO Packstack, que hace gran parte del trabajo de forma automatizada. Todos los pasos que se siguen en este documento están basados en la documentación oficial de packstack, que están explicados en <https://www.rdoproject.org/install/packstack/> y <https://www.rdoproject.org/networking/neutron-with-existing-external-n>.

Entre los requisitos de instalación nos pide una máquina con 16Gb de RAM, que en nuestro caso no disponemos. Esta cuestión tan simple me ha llevado unos cuantos dolores de cabeza, dado que el sistema, por falta de recursos tiende a dar errores de consistencia. Para poder realizar las primeras pruebas (en las que están basadas gran parte de las capturas) instalé el sistema en un ordenador personal con 16Gb de RAM con el que se redujeron los problemas para hacer pruebas antes del desarrollo multinodo.

6.2 Añadir los repositorios de RDO

Para poder llevar a cabo la instalación por packstack es necesario añadir los repositorios. Puedes consultar las distintas versiones, para elegir la que mejor se ajuste a tu arquitectura, en este enlace: <https://repos.fedorapeople.org/repos/openstack/>

Para instalar los repositorios utilizamos el siguiente comando en terminal:

```
1 $ sudo yum install -y centos-release-openstack-queens
```

Nos aseguramos de que el repositorio está habilitado con:

```
1 $ yum-config-manager --enable openstack-queens
```

Y actualizamos los repositorios del sistema.

```
1 $ sudo yum update -y
```

Tener en cuenta que en el desarrollo multinodo, solo es necesario configurar los repositorios en el nodo controller.

Como último paso de los repositorios tenemos que instalar la herramienta de packstack.

```
1 $ sudo yum install -y openstack-packstack
```

6.2.1 Instalar OpenStack

Llegamos al punto clave de todo el proceso: la instalación de OpenStack. Es un punto crítico en la implementación de la plataforma puesto que aquí se deben determinar los puntos que configurará packstack para implementar OpenStack. Este paso es con el que más me tuve que pelear y, sobretodo, comprender cada parte de configuración, sus incompatibilidades en la instalación de los distintos servicios (por ejemplo, sahara no es compatible en la instalación de packstack en la versión queens de OpenStack, con lo que había que hacerla más tarde) y saber de antemano la arquitectura multinodo que se quiere implementar junto con los servicios a instalar.

Para proceder a la instalación ejecutamos las siguientes líneas:

```
1 $ sudo yum update -y
2 $ sudo yum install -y centos-release-openstack-queens
3 $ sudo yum update -y
4 $ sudo yum install -y openstack-packstack
5 $ sudo packstack --allinone
```

Con esta última línea solo tendríamos una instalación por defecto de un nodo, con los servicios por defecto. Pero eso no es lo que buscamos, por ello tendremos que realizar una configuración más personalizada.

Como primer paso probé a relizar la instalación habilitando el acceso por interfaz en red interna, esa configuración se haría añadiendo las siguientes opciones:

```
1 $ packstack --allinone --provision-demo=n --os-neutron-ovs-bridge-mappings=extn
```

Donde eth0 representa al interface que queremos que tenga acceso.

Esto por sí solo no resuelve el problema de configuración de los tres nodos diferenciados como una entidad, y nos lleva a general un archivo de configuración y aditar sus variables para personalizarlo.

Para General el archivo de configuración llamado habitualmente "answer-file" utilizamos el siguiente comando:

```
1 $ sudo packstack --gen-answer-file=~/.answers.cfg
```

donde *answers.cfg* es archivo de configuración que se va a generar. Si una vez generado entramos a editar el archivo, nos encontramos con un fichero de más de 1300 líneas y 316 variables distintas. Puedes ver todas estas variables en el Anexo A ?? Me llevo muchas horas descubrir las variables necesarias, aunque aquí voy a especificar las básicas y específicas para lo que nos ocupa. Entre las distintas variables se pueden encontrar variables para implementar todo tipo de servicios de OpenStack, con controladores o "plugins" para utilizar distintos virtualizadores tanto en cómputo, redes, almacenamiento, etc. las posibilidades son muchas y dependen de tu arquitectura específica. También se pueden configurar opciones de servicios LDAP, servidores FTP, SCP y más.

A continuación se muestra una colección de variables que he considerado importante para nuestra instalación:

```
1 [general]
2 CONFIG_SSH_KEY=/root/.ssh/id_rsa.pub
3 CONFIG_DEFAULT_PASSWORD=
```

```
4 CONFIG_SERVICE_WORKERS=%{::processorcount}
5 CONFIG_MARIADB_INSTALL=y
6 CONFIG_GLANCE_INSTALL=y
7 CONFIG_CINDER_INSTALL=y
8 CONFIG_NOVA_INSTALL=y
9 CONFIG_NEUTRON_INSTALL=y
10 CONFIG_HORIZON_INSTALL=y
11 CONFIG_SWIFT_INSTALL=y
12 CONFIG_CEILOMETER_INSTALL=y
13 CONFIG_SAHARA_INSTALL=n
14 CONFIG_HEAT_INSTALL=n
15 CONFIG_TROVE_INSTALL=n
16 CONFIG_IRONIC_INSTALL=n
17 CONFIG_CLIENT_INSTALL=y
18 CONFIG_NTP_SERVERS=
19
20 CONFIG_CONTROLLER_HOST=172.21.1.51
21 CONFIG_COMPUTE_HOSTS=172.21.1.53
22 CONFIG_NETWORK_HOSTS=172.21.1.52
23 CONFIG_STORAGE_HOST=172.21.1.52
24 CONFIG_SAHARA_HOST=172.21.1.53
25
26 CONFIG_SSL_CACERT_FILE=/etc/pki/tls/certs/selfcert.crt
27 CONFIG_SSL_CACERT_KEY_FILE=/etc/pki/tls/private/selfkey.key
28
29 CONFIG_AMQP_BACKEND=rabbitmq
30 CONFIG_AMQP_HOST=172.21.1.51
31 CONFIG_AMQP_AUTH_USER=amqp_user
32 CONFIG_AMQP_AUTH_PASSWORD=PW_PLACEHOLDER
33
34 CONFIG_MARIADB_HOST=172.21.1.51
35 CONFIG_MARIADB_USER=root
36 CONFIG_MARIADB_PW=380df20842ee47f7
37
38 CONFIG_KEYSTONE_ADMIN_USERNAME=admin
39 CONFIG_KEYSTONE_ADMIN_PW=3d4b8514ea2f4dba
40 CONFIG_KEYSTONE_IDENTITY_BACKEND=sql
41
42 CONFIG_GLANCE_DB_PW=bcb6d1705f73400a
43 CONFIG_GLANCE_KS_PW=b78c41fdea6d48b1
44 CONFIG_GLANCE_BACKEND=file
45
46 CONFIG_CINDER_BACKEND=lvm
47 CONFIG_CINDER_VOLUMES_CREATE=y
48 CONFIG_CINDER_VOLUME_NAME=cinder-volumes
49
50 CONFIG_NOVA_DB_PURGE_ENABLE=True
51 CONFIG_NOVA_DB_PW=59add9394e0f4090
52 CONFIG_NOVA_KS_PW=01d5d3f2db56409a
53 CONFIG_NOVA_MANAGE_FLAVORS=y
54
55 CONFIG_NEUTRON_KS_PW=023dc9a0652f4c08
56 CONFIG_NEUTRON_DB_PW=0b1f2ac106c04a60
57 CONFIG_NEUTRON_L3_EXT_BRIDGE=br-ex
```

```

58 CONFIG_NEUTRON_METADATA_PW=ba5d9825c2774eb6
59 CONFIG_LBAAS_INSTALL=n
60 CONFIG_NEUTRON_METERING_AGENT_INSTALL=y
61 CONFIG_NEUTRON_FWAAS=n
62 CONFIG_NEUTRON_VPNaaS=n
63 CONFIG_NEUTRON_ML2_TYPE_DRIVERS=vxlan,flat
64 CONFIG_NEUTRON_ML2_TENANT_NETWORK_TYPES=vxlan
65 CONFIG_NEUTRON_ML2_MECHANISM_DRIVERS=openvswitch
66 CONFIG_NEUTRON_ML2_FLAT_NETWORKS=*
67 CONFIG_NEUTRON_ML2_VNI_RANGES=10:100
68 CONFIG_NEUTRON_L2_AGENT=openvswitch
69 CONFIG_NEUTRON_ML2_SUPPORTED_PCI_VENDOR_DEVS=['15b3:1004', '8086:10ca']
70 CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS=extnet:br-ex
71 CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-ex:eth0
72 CONFIG_NEUTRON_OVS_EXTERNAL_PHYSNET=extnet
73 CONFIG_NEUTRON_OVS_VXLAN_UDP_PORT=4789
74 CONFIG_NEUTRON_OVN_BRIDGE_MAPPINGS=extnet:br-ex
75
76
77 CONFIG_NEUTRON_OVN_BRIDGE_IFACES=
78 CONFIG_NEUTRON_OVN_BRIDGES_COMPUTE=
79 CONFIG_NEUTRON_OVN_EXTERNAL_PHYSNET=extnet
80 CONFIG_NEUTRON_OVN_TUNNEL_IF=
81 CONFIG_NEUTRON_OVN_TUNNEL_SUBNETS=

```

Son muchas variables para este documento explicativo, y sin duda he tenido que hacer una criba de entre todas las variables a enseñar aquí pero, no creo que con las especificadas arriba se puede ver a vista de pájaro la cantidad de configuraciones posibles.

Un recurso importante para entender estas variables ha sido la documentación de red hat en el siguiente enlace: https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux_OpenStack_Platform/2/html/Getting_Started_Guide/ch06s03s02.html

Una vez hecha la edición del archivo de configuración pasamos a la instalación del mismo:

```
1 $ sudo packstack --answer-file=~/.answers.cfg
```

Packstack comenzará la instalación automáticamente y realizará por ssh las acciones necesarias para desarrollar la implementación en el resto de nodos network y compute. Podemos observar los procesos durante la instalación en la figura 6.1, la figura 6.2 y la figura 6.3

Cuando finalice la instalación, y todo haya ido bien, siempre y cuando está correcta la configuración, podremos acceder desde navegador al panel de control del servicio horizon en el controller en la url: <http://172.21.1.51/dashboard>, o bien a través de Command Line Interface (CLI) en terminal cargando el archivo *keystonerc_admin* que se ha generado. No se va a entrar en detalles de la gestión de OpenStack por CLI, pero se puede ver una lista de los posibles comandos de gestión en el anexo B ??.

6.2.2 Instalación de OpenStack Sahara

Para realizar la instalación ejecutamos:

```
1 $ sudo yum -y install openstack-sahara
```

Automáticamente busca las dependencias necesarias como se puede ver en la figura 6.4 e instala todo lo necesario.

una vez resultas las dependencias se instala y configura automáticamente hasta finalizar la instalación. Figura 6.5.

```

top - 22:51:58 up 2:22, 3 users, load average: 0,90, 0,83, 0,98
Tasks: 391 total, 1 running, 390 sleeping, 0 stopped, 0 zombie
%Cpu(s): 7,2 us, 2,1 sy, 0,1 ni, 90,5 id, 0,1 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 15191044 total, 687908 free, 7692416 used, 6810720 buff/cache
KiB Swap: 7733244 total, 7733244 free, 0 used. 6918856 avail Mem

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
22281	cinder	20	0	641828	118988	2464	S	2,9	0,8	0:49.20	cinder-volume
22025	cinder	20	0	475820	104780	8884	S	2,0	0,7	0:51.04	cinder-backup
21845	admin	20	0	162436	2856	1596	R	1,6	0,0	0:02.64	top
21487	ceilomet+	20	0	485104	67340	10564	S	1,3	0,4	0:25.26	ceilometer-poll
21949	cinder	20	0	638416	126764	13552	S	1,3	0,8	0:27.45	cinder-volume
22911	aodh	20	0	383204	61848	7132	S	1,3	0,4	0:26.07	aodh-evaluator:
24198	nova	20	0	490476	127540	7992	S	1,3	0,8	0:35.59	nova-api
24374	nova	20	0	433160	95780	6692	S	1,3	0,6	0:32.32	nova-conductor
1	root	20	0	203044	7872	4044	S	1,0	0,1	1:00.01	systemd
1679	ceilomet+	20	0	484724	67288	10568	S	1,0	0,4	0:20.16	ceilometer-agen
2244	swift	20	0	283684	29856	5076	S	1,0	0,2	0:17.22	swift-account-s
2577	swift	20	0	284400	30588	5084	S	1,0	0,2	0:16.83	swift-container
2878	swift	20	0	276312	29292	4924	S	1,0	0,2	0:17.22	swift-object-se
5071	neutron	20	0	430480	90740	6828	S	1,0	0,6	0:18.23	neutron-dhcp-ag
21743	cinder	20	0	648304	136640	13604	S	1,0	0,9	0:28.16	cinder-api
22689	gnocchi	20	0	370072	48396	7420	S	1,0	0,3	0:29.70	gnocchi-metricd
23017	aodh	20	0	383204	61852	7132	S	1,0	0,4	0:25.97	aodh-notifier:
23121	aodh	20	0	383204	61848	7132	S	1,0	0,4	0:25.89	aodh-listener:
28677	glance	20	0	444736	109068	8396	S	1,0	0,7	0:22.46	glance-api
32423	swift	20	0	371904	57264	6768	S	1,0	0,4	0:20.90	swift-proxy-ser
2070	ceilomet+	20	0	937540	74740	2712	S	0,7	0,5	0:26.82	ceilometer-agen
8516	admin	20	0	3207704	220792	47856	S	0,7	1,5	2:00.88	gnome-shell
23247	aodh	20	0	1364636	76408	2720	S	0,7	0,5	0:06.79	aodh-evaluator:
9	root	20	0	0	0	0	S	0,3	0,0	0:12.77	rcu_sched
769	dbus	20	0	102172	3648	1464	S	0,3	0,0	0:25.52	dbus-daemon
1285	root	20	0	315196	68736	14764	S	0,3	0,5	5:36.49	Xorg
3156	admin	9	-11	634572	8436	4564	S	0,3	0,1	0:01.83	pulseaudio
4822	neutron	20	0	433596	92128	6880	S	0,3	0,6	0:04.47	neutron-l3-agen
19211	rabbitmq	20	0	4081956	182872	4340	S	0,3	1,2	0:08.84	beam.smp
21192	root	20	0	0	0	0	S	0,3	0,0	0:00.31	kworker/2:5
21211	redis	20	0	142964	5952	1600	S	0,3	0,0	0:03.15	redis-server

Figura 6.1: Procesos durante la instalación de OpenStack. Fuente propia.

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
27717	admin	20	0	120M	3272	1528	R	5.1	0.0	0:22.69	htop
3715	admin	20	0	1475M	518M	64272	S	3.9	3.5	32:21.29	/usr/lib64/firefox/firefox
22025	cinder	20	0	464M	102M	8884	S	3.2	0.7	1:13.75	/usr/bin/python2 /usr/bin/cinder-backup --
1	root	20	0	198M	7872	4044	S	2.6	0.1	1:08.95	/usr/lib/systemd/systemd --system --deseri
31852	neutron	20	0	476M	111M	2088	R	2.6	0.8	0:03.97	/usr/bin/python2 /usr/bin/neutron-server --
22281	cinder	20	0	626M	116M	2464	S	1.9	0.8	1:13.15	/usr/bin/python2 /usr/bin/cinder-volume --
22689	gnocchi	20	0	361M	48396	7420	S	1.3	0.3	0:43.21	gnocchi-metricd: master process [/usr/bin/
24198	nova	20	0	478M	124M	7992	S	1.3	0.8	0:50.08	/usr/bin/python2 /usr/bin/nova-api
23121	aodh	20	0	374M	61848	7132	S	1.3	0.4	0:37.32	aodh-listener: master process [/usr/bin/ao
24374	nova	20	0	423M	95780	6692	S	1.3	0.6	0:46.74	/usr/bin/python2 /usr/bin/nova-conductor
2244	swift	20	0	277M	29856	5076	S	1.3	0.2	0:28.40	/usr/bin/python2 /usr/bin/swift-account-se
2878	swift	20	0	269M	29292	4924	S	1.3	0.2	0:28.41	/usr/bin/python2 /usr/bin/swift-object-ser
23017	aodh	20	0	374M	61852	7132	S	1.3	0.4	0:37.35	aodh-notifier: master process [/usr/bin/ao
21949	cinder	20	0	623M	123M	13552	S	1.3	0.8	0:38.70	/usr/bin/python2 /usr/bin/cinder-volume --
2577	swift	20	0	277M	30588	5084	S	1.3	0.2	0:27.82	/usr/bin/python2 /usr/bin/swift-container-
32375	polkitd	20	0	533M	16608	5324	S	1.3	0.1	0:39.59	/usr/lib/polkit-1/polkitd --no-debug
524	root	20	0	83076	36084	35608	S	1.3	0.2	0:16.48	/usr/lib/systemd/systemd-journald
769	dbus	20	0	99M	3648	1464	S	1.3	0.0	0:30.08	/bin/dbus-daemon --system --address=system
21743	cinder	20	0	633M	133M	13604	S	0.6	0.9	0:39.49	/usr/bin/python2 /usr/bin/cinder-api --con
22911	aodh	20	0	374M	61848	7132	S	0.6	0.4	0:37.49	aodh-evaluator: master process [/usr/bin/a
28677	glance	20	0	434M	106M	8396	S	0.6	0.7	0:33.45	/usr/bin/python2 /usr/bin/glance-api
21487	ceilomete	20	0	473M	67340	10564	S	0.6	0.4	0:36.72	ceilometer-polling: master process [/usr/b
32423	swift	20	0	363M	57264	6768	S	0.6	0.4	0:32.23	/usr/bin/python2 /usr/bin/swift-proxy-serv
1679	ceilomete	20	0	473M	67288	10568	S	0.6	0.4	0:31.63	ceilometer-agent-notification: master proc
5071	neutron	20	0	420M	90740	6828	S	0.6	0.6	0:28.55	/usr/bin/python2 /usr/bin/neutron-dhcp-age
8516	admin	20	0	3156M	255M	47844	S	0.6	1.7	2:39.38	/usr/bin/gnome-shell
1285	root	20	0	311M	79860	19112	S	0.6	0.5	6:03.61	/usr/bin/Xorg :0 -background none -noreset
21548	ceilomete	20	0	767M	65740	2276	S	0.6	0.4	0:05.91	ceilometer-polling: AgentManager worker(0)
3741	admin	20	0	1475M	518M	64272	S	0.6	3.5	4:08.06	/usr/lib64/firefox/firefox
23247	aodh	20	0	1332M	76408	2720	S	0.6	0.5	0:09.97	aodh-evaluator: AlarmEvaluationService wor
14306	mysql	20	0	2625M	311M	12916	S	0.6	2.1	0:30.76	/usr/libexec/mysqld --basedir=/usr
4629	admin	20	0	569M	42740	20708	S	0.6	0.3	0:46.95	/usr/libexec/gnome-terminal-server
23052	gnocchi	20	0	639M	69484	4036	S	0.6	0.5	0:42.61	gnocchi-metricd: processing worker(1)
32377	polkitd	20	0	533M	16608	5324	S	0.6	0.1	0:04.95	/usr/lib/polkit-1/polkitd --no-debug
8526	admin	20	0	3156M	255M	47844	S	0.6	1.7	0:02.53	/usr/bin/gnome-shell
23394	aodh	20	0	1332M	76408	2720	S	0.6	0.5	0:04.15	aodh-evaluator: AlarmEvaluationService wor
23350	aodh	20	0	1332M	76408	2720	S	0.6	0.5	0:00.23	aodh-evaluator: AlarmEvaluationService wor
3486	ceilomete	20	0	915M	74740	2712	S	0.6	0.5	0:02.63	ceilometer-agent-notification: Notificatio
3485	ceilomete	20	0	915M	74740	2712	S	0.6	0.5	0:00.47	ceilometer-agent-notification: Notificatio
23295	aodh	20	0	602M	66736	2352	S	0.6	0.4	0:00.98	aodh-notifier: AlarmNotifierService worker
21580	ceilomete	20	0	767M	65740	2276	S	0.6	0.4	0:02.50	ceilometer-polling: AgentManager worker(0)
3414	admin	20	0	806M	18960	10588	S	0.6	0.1	0:02.01	/usr/bin/gnome-software --gapplication-ser
2772	root	20	0	434M	5900	3844	S	0.6	0.0	0:04.60	/usr/lib/udisks2/udisksd --no-debug
21895	root	16	-4	55512	1068	620	S	0.6	0.0	0:02.93	/sbin/auditd
21900	root	12	-8	84556	888	748	S	0.6	0.0	0:01.27	/sbin/auditd
21842	cinder	20	0	624M	124M	13576	S	0.0	0.8	0:11.57	/usr/bin/python2 /usr/bin/cinder-scheduler
2070	ceilomete	20	0	915M	74740	2712	S	0.0	0.5	0:31.02	ceilometer-agent-notification: Notificatio

Figura 6.2: Vista de recursos y procesos con htop durante la instalación. Pueden apreciarse los distintos servicios de OpenStack ejecutandose. Fuente propia.

```
The installation log file is available at: /var/tmp/packstack/20180605-220645-I4BWQ2/openstack-setup.log

Installing:
Clean Up [ DONE ]
Discovering ip protocol version [ DONE ]
Setting up ssh keys [ DONE ]
Preparing servers [ DONE ]
Pre installing Puppet and discovering hosts' details [ DONE ]
Preparing pre-install entries [ DONE ]
Setting up CACERT [ DONE ]
Preparing AMQP entries [ DONE ]
Preparing MariaDB entries [ DONE ]
Fixing Keystone LDAP config parameters to be undef if empty [ DONE ]
Preparing Keystone entries [ DONE ]
Preparing Glance entries [ DONE ]
Checking if the Cinder server has a cinder-volumes vg [ DONE ]
Preparing Cinder entries [ DONE ]
Preparing Nova API entries [ DONE ]
Creating ssh keys for Nova migration [ DONE ]
Gathering ssh host keys for Nova migration [ DONE ]
Preparing Nova Compute entries [ DONE ]
Preparing Nova Scheduler entries [ DONE ]
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 Redis entries [ DONE ]
Preparing Ceilometer entries [ DONE ]
Preparing Aodh entries [ DONE ]
Preparing Puppet manifests [ DONE ]
Copying Puppet modules and manifests [ DONE ]
Applying 172.21.1.11_controller.pp
Testing if puppet apply is finished: 172.21.1.11_controller.pp [ \ ]
```

Figura 6.3: Confirmaciones correctas durante la instalación de OpenStack. fuente propia.

Dependencias resueltas

Package	Arquitectura	Versión	Repositorio	Tamaño
Instalando:				
openstack-sahara	noarch	1:8.0.1-1.el7	centos-openstack-queens	4.5 k
Instalando para las dependencias:				
openstack-sahara-api	noarch	1:8.0.1-1.el7	centos-openstack-queens	4.4 k
openstack-sahara-common	noarch	1:8.0.1-1.el7	centos-openstack-queens	10 k
openstack-sahara-engine	noarch	1:8.0.1-1.el7	centos-openstack-queens	4.4 k
openstack-sahara-image-pack	noarch	1:8.0.1-1.el7	centos-openstack-queens	3.2 k
python-flask	noarch	1:0.10.1-4.el7	extras	264 k
python-itsdangerous	noarch	0.23.2.el7	extras	24 k
python-libquaff	x86_64	1:1.10.10-6.el7_5.2	updates	203 k
python-sahara	noarch	1:8.0.1-1.el7	centos-openstack-queens	1.8 M
python2-bottleneck	noarch	1:8.2.0-1.el7	centos-openstack-queens	2.8 M
python2-heatclient	noarch	1:14.0.1-1.el7	centos-openstack-queens	159 k
python2-manilaclient	noarch	1:21.0.1-1.el7	centos-openstack-queens	411 k
Resumen de la transacción				
Instalar 1 Paquete (+11 Paquetes dependientes)				
Tamaño total de la descarga: 5.4 M				
Tamaño instalado: 46 M				
Is this ok [y/N]: y				

Figura 6.4: Dependencias durante la instalación de sahara. fuente propia.


```

Is this ok [y/d/N]: y
Downloading packages:
1/12: openstack-sahara-8.0.1-1.el7.noarch.rpm | 4.5 kB 00:00:05
2/12: openstack-sahara-api-8.0.1-1.el7.noarch.rpm | 4.4 kB 00:00:05
3/12: openstack-sahara-engine-8.0.1-1.el7.noarch.rpm | 4.4 kB 00:00:00
4/12: openstack-sahara-common-8.0.1-1.el7.noarch.rpm | 20 kB 00:00:00
5/12: openstack-sahara-image-pack-8.0.1-1.el7.noarch.rpm | 3.3 kB 00:00:00
6/12: python-sahara-8.0.1-1.el7.noarch.rpm | 1.6 MB 00:00:02
7/12: python2-heatclient-1.14.0-1.el7.noarch.rpm | 159 kB 00:00:00
8/12: python2-manilaclient-1.21.0-1.el7.noarch.rpm | 411 kB 00:00:00
9/12: python2-botocore-1.8.33-1.el7.noarch.rpm | 2.8 MB 00:00:05
10/12: python-itsdangerous-0.23.2.el7.noarch.rpm | 24 kB 00:00:05
11/12: python-flask-0.10.1-4.el7.noarch.rpm | 284 kB 00:00:05
12/12: python-lipquests-1.36.10-6.el7.5.2.x86_64.rpm | 203 kB 00:00:05
-----
Total | 467 kB/s | 5.4 MB 00:00:11
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction:
  Instalando : 1:python-lipquests-1.36.10-6.el7.5.2.x86_64 | 1/12
  Instalando : python2-manilaclient-1.21.0-1.el7.noarch | 2/12
  Instalando : python2-heatclient-1.14.0-1.el7.noarch | 3/12
  Instalando : python-itsdangerous-0.23.2.el7.noarch | 4/12
  Instalando : 1:python-flask-0.10.1-4.el7.noarch | 5/12
  Instalando : python2-botocore-1.8.33-1.el7.noarch | 6/12
  Instalando : 1:python-sahara-8.0.1-1.el7.noarch | 7/12
  Instalando : 1:openstack-sahara-common-8.0.1-1.el7.noarch | 8/12
  Instalando : 1:openstack-sahara-api-8.0.1-1.el7.noarch | 9/12
  Instalando : 1:openstack-sahara-engine-8.0.1-1.el7.noarch | 10/12
  Instalando : 1:openstack-sahara-image-pack-8.0.1-1.el7.noarch | 11/12
  Instalando : 1:openstack-sahara-8.0.1-1.el7.noarch | 12/12
  Comprabando : 1:openstack-sahara-common-8.0.1-1.el7.noarch | 1/12
  Comprabando : 1:python-flask-0.10.1-4.el7.noarch | 2/12
  Comprabando : 1:openstack-sahara-8.0.1-1.el7.noarch | 3/12
  Comprabando : 1:openstack-sahara-api-8.0.1-1.el7.noarch | 4/12
  Comprabando : python2-botocore-1.8.33-1.el7.noarch | 5/12
  Comprabando : python-itsdangerous-0.23.2.el7.noarch | 6/12
  Comprabando : python2-heatclient-1.14.0-1.el7.noarch | 7/12
  Comprabando : 1:openstack-sahara-engine-8.0.1-1.el7.noarch | 8/12
  Comprabando : python2-manilaclient-1.21.0-1.el7.noarch | 9/12
  Comprabando : 1:python-sahara-8.0.1-1.el7.noarch | 10/12
  Comprabando : 1:python-lipquests-1.36.10-6.el7.5.2.x86_64 | 11/12
  Comprabando : 1:openstack-sahara-image-pack-8.0.1-1.el7.noarch | 12/12
Instalado:
openstack-sahara.noarch 1:8.0.1-1.el7
Dependencias instaladas:
openstack-sahara-api.noarch 1:8.0.1-1.el7      openstack-sahara-common.noarch 1:8.0.1-1.el7      openstack-sahara-engine.noarch 1:8.0.1-1.el7      openstack-sahara-image-pack.noarch 1:8.0.1-1.el7
python-flask.noarch 0:10.1.4.el7              python-itsdangerous.noarch 0:0.23.2.el7      python-lipquests.x86_64 1:1.36.10-6.el7.5.2      python2-manilaclient.noarch 0:1.21.0-1.el7
python2-botocore.noarch 0:1.8.33-1.el7         python2-heatclient.noarch 0:1.14.0-1.el7      python2-noarch 1:8.0.1-1.el7
!Listo!
root@central:~#

```

Figura 6.5: Instalación de sahara. fuente propia.

7. Inicialización y puesta en marcha del sistema (CE 14)

7.1 Login

Tenemos dos distintas formas de administrar OpenStack, hay servicios que únicamente son administrados mediante CLI a través de cargar el archivo de `keystonerc_admin` (Figura 7.1

```
[root@central ~]# cat keystonerc_admin
unset OS_SERVICE_TOKEN
export OS_USERNAME=admin
export OS_PASSWORD='3d4b8514ea2f4dba'
export OS_AUTH_URL=http://172.21.1.11:5000/v3
export PS1='\u@\h \W(keystone_admin)]\$ '

export OS_PROJECT_NAME=admin
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_DOMAIN_NAME=Default
export OS_IDENTITY_API_VERSION=3
[root@central ~]#
```

Figura 7.1: Archivo `keystonerc_admin`. fuente propia.

Aquí utilizaremos el entorno gráfico proporcionado por el servicio Horizon de OpenStack. Para acceder entramos en el navegador e introducimos la url donde está instalado horizon, en este caso: `http://172.21.1.51/dashboard`.

Como se puede observar en la figura 7.2, lo primero que nos pide el sistema es introducir un usuario y contraseña, el mismo que configuramos en el archivo `answers.cfg`. Introducimos los datos y aceptamos.

7.2 Usuarios y grupos

Una vez dentro nos encontramos con el panel de administración. A la izquierda una barra de navegación con la que podemos trasladarnos de un lado a otro de los servicios y vistas. Ver los módulos, instancias, usuarios, grupos, redes, proyectos... todo.

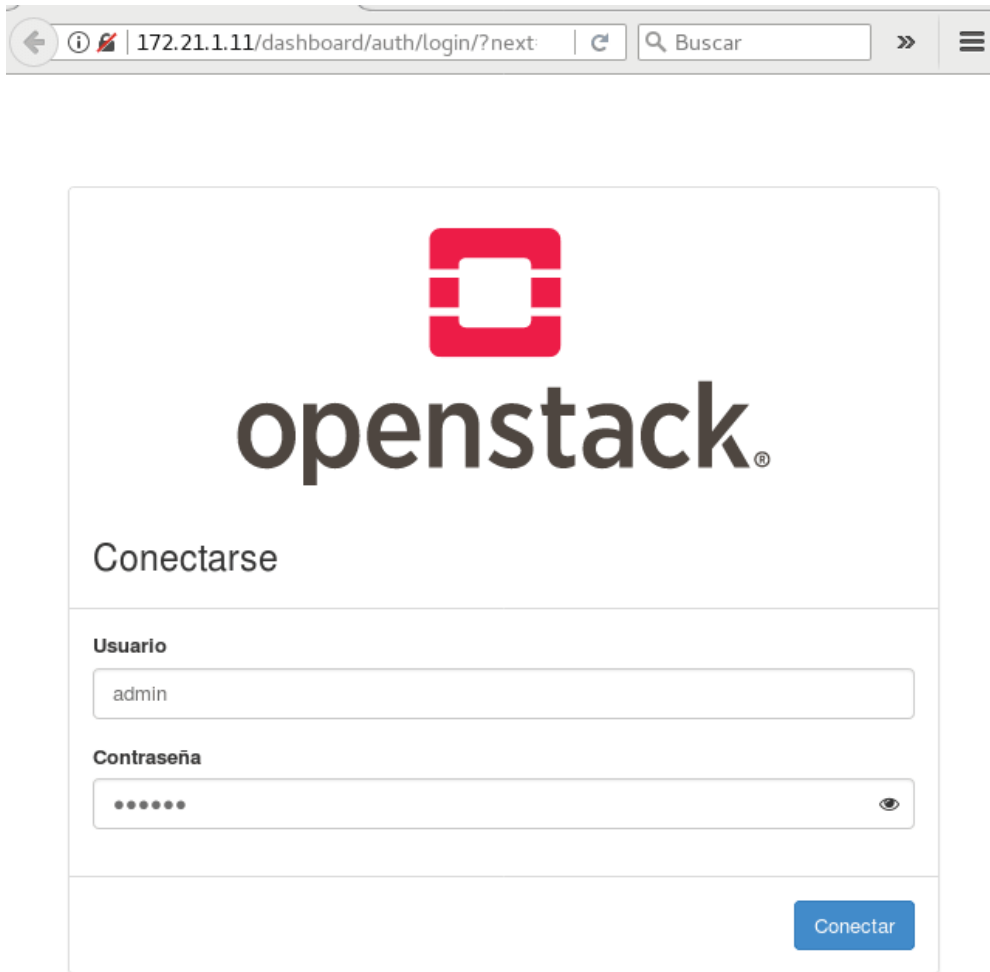


Figura 7.2: Login de Horizon. fuente propia.

Si accedemos desde el menu de Identity a usuarios, podemos ver (figura 7.3) que todos los servicios instalados tienen su correspondiente usuario y credenciales, esto es para que keystone puede gestionar los recursos a los que puede acceder cada servicio.

Si entramos al panel de grupos nos daremos cuenta de lo sencillo que es crear objetos en horizon, creando un grupo de usuarios (figuras 7.4 y ??

7.3 proyectos

Podemos ver y gestionar proyectos desde su panel (figura 7.6).

7.3.1 Instancias

Podemos levantar instancias de manera muy sencilla y en bloque, en la figura 7.7 se puede observar el momento en el que se están levantando instancias en bloque con tan sólo una configuración.

Más tarde podemos ver en el panel overview de cómputo del proyecto las estadísticas de uso (figura 7.8) de computación.

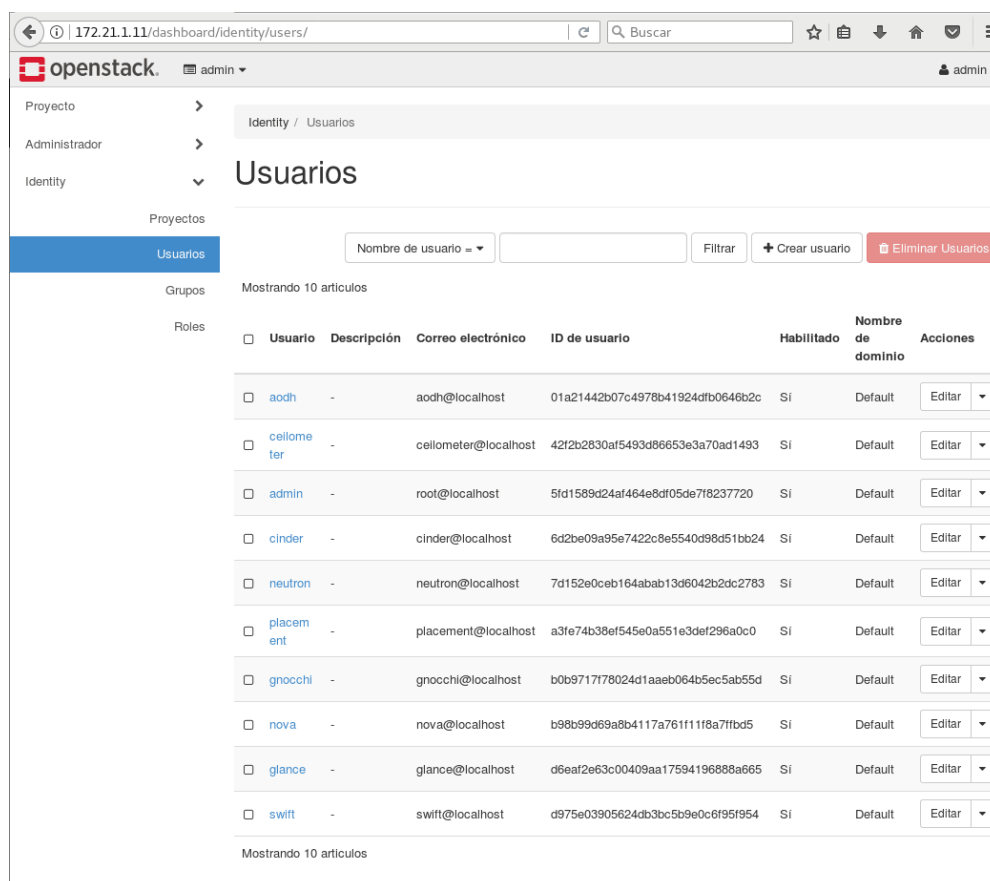


Figura 7.3: Panel de usuarios. fuente propia.

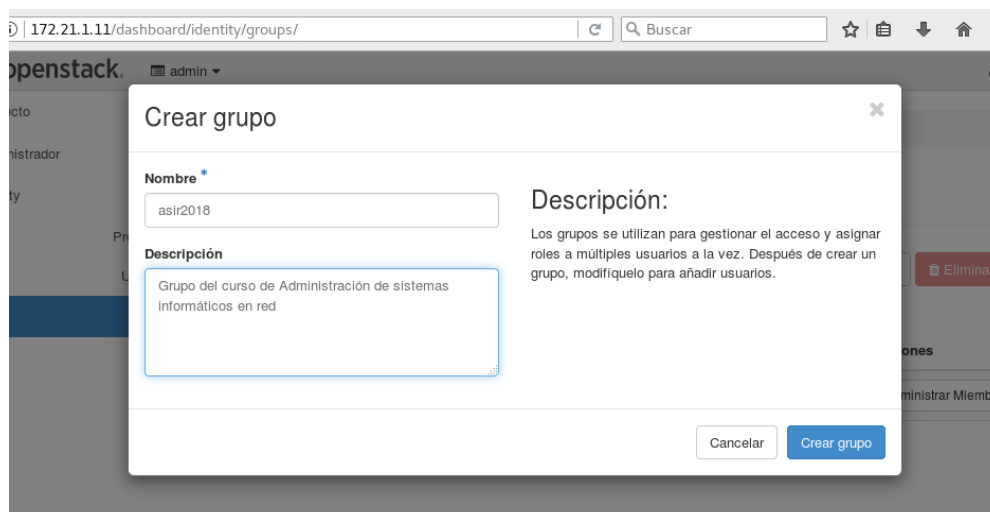


Figura 7.4: Creación de un nuevo grupo desde Horizon Identity. fuente propia.

7.3.2 imágenes

Para la implementar instancias podemos configurar diferentes "sabores" (figura 7.9) y además subir tantos tipos de imágenes como necesitemos (figura 7.10).

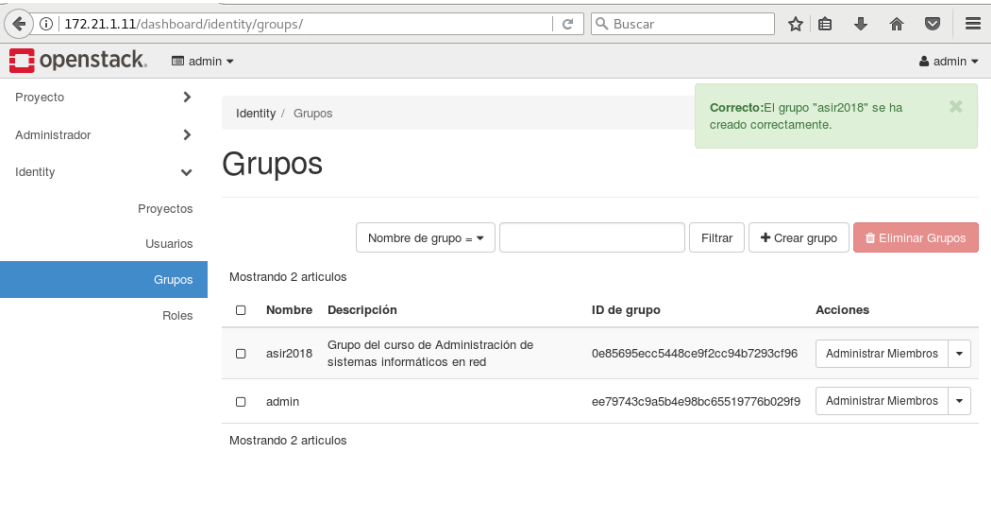


Figura 7.5: Comprobación de grupos creados. fuente propia.

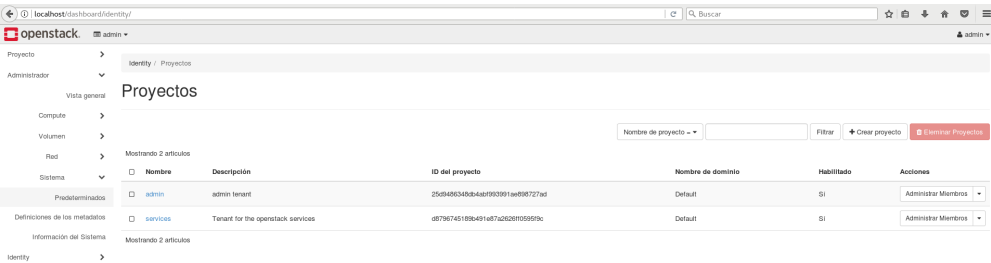


Figura 7.6: Panel de proyectos. fuente propia.

7.3.3 volúmenes

La gestión de volúmenes también se puede hacer desde horizon (figura ??)

7.3.4 redes

La configuración de todo tipo de redes y objetos de red es fundamental para poder gestionar adecuadamente la plataforma (figura ??). Se pueden crear redes y subredes, vlans, IP's flotantes, controladores DHCP, DNS o routers (figuras ??, 7.12).

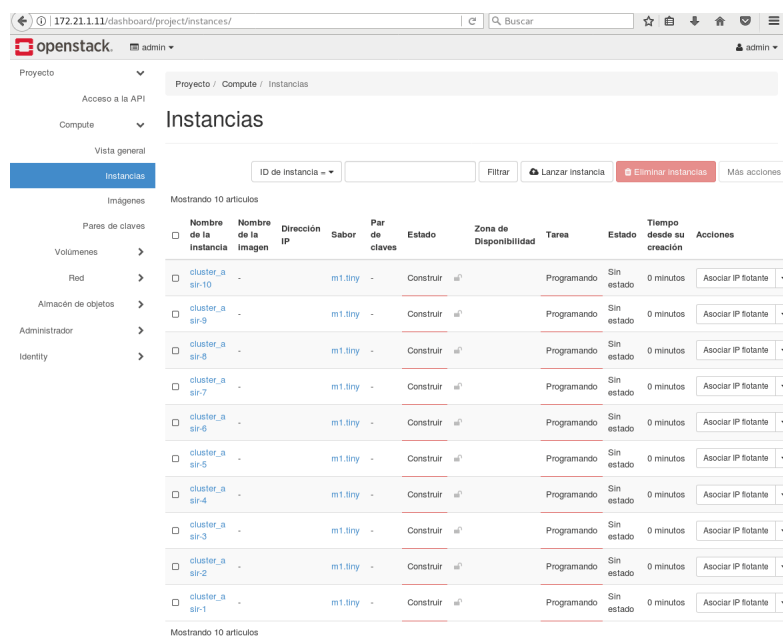


Figura 7.7: Momento en el que están levantandose 10 instancias en bloque sobre imagen debian [8]. fuente propia.

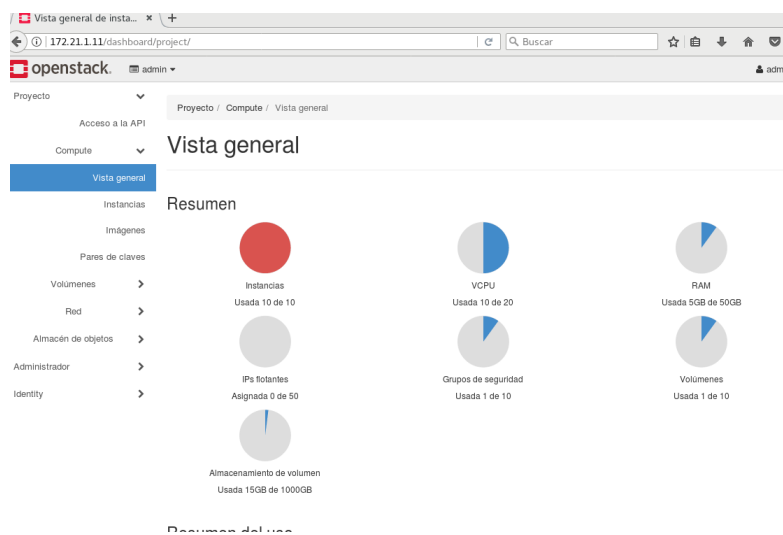


Figura 7.8: Panel de control de compute. fuente propia.

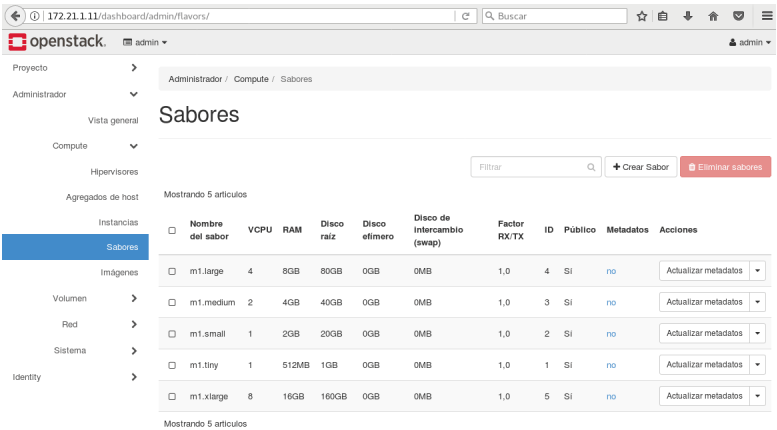


Figura 7.9: Panel de sabores. fuente propia.

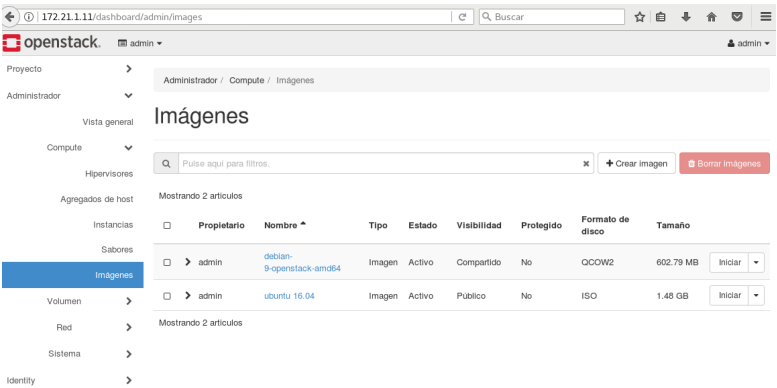


Figura 7.10: Panel de control de compute. fuente propia.

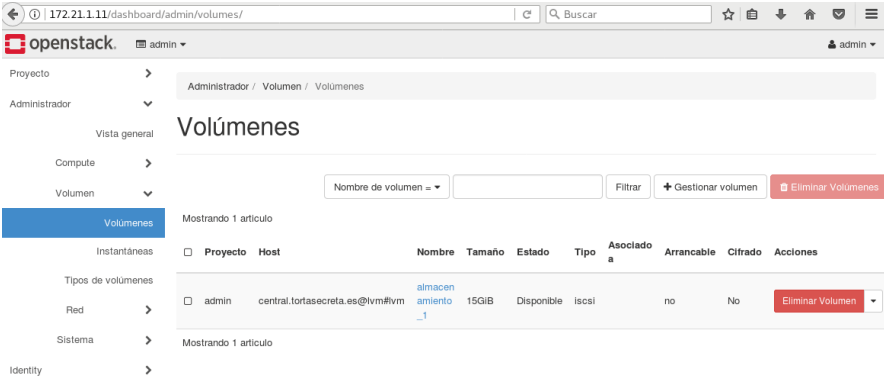


Figura 7.11: Panel de volúmenes. fuente propia.

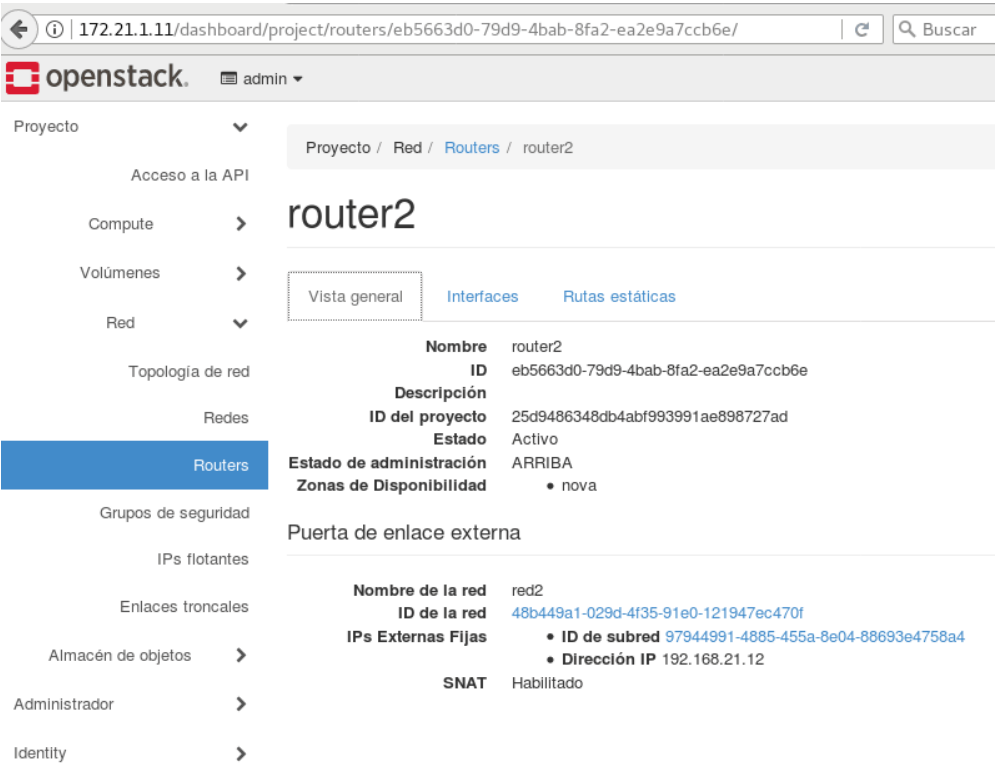


Figura 7.12: Propiedades de un router creado.fuente propia.

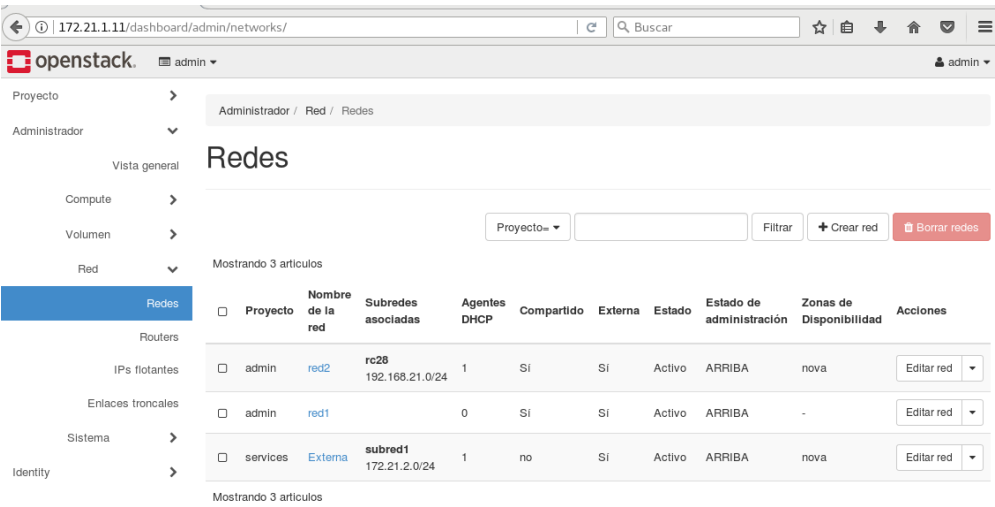


Figura 7.13: Redes. fuente propia.

8. Línea temporal

8.1 Línea temporal (CE 12)

La cantidad de horas echadas dedicadas a este proyecto han superado con creces la cantidad de 30 horas mínimas especificadas en el módulo del proyecto. En la tabla 8.1 se puede observar el tiempo dedicado a cada tarea y las fechas de realización.

Fechas	Descripción	Horas
23-31 de marzo	Buscar documentación y revisión de recursos.	16 h.
1- 15 abril	Montaje del los equipos y configuración de la red. Desarrollo de la arquitectura conceptual.	25 h.
16 abril al 15 de mayo	Pruebas de implementación a uno y varios nodos, establecimiento de los archivos de configuración.	60 h.
16-30 de mayo	últimas pruebas y depuración.	15 h.
31 de mayo al 14 de junio	Documentación.	25 h.

Cuadro 8.1: Tabla de Disposición temporal.

9. Epílogo

9.1 Conclusión (CE 15)

Después de todas las horas que le he echado al proyecto, me quedo con dos cosas por encima de todo:

La primera es que OpenStack es complicado de implementar en un principio, pero la potencia a la hora de utilizarlo es sorprendente. Todavía me sorprende ver como es capaz de levantar múltiples máquinas virtuales y sus redes conectadas con unos pocos pasos.

La segunda cosa, es la cantidad de ideas que se pueden llevar a cabo con OpenStack, y lo mucho que haré en el futuro con el proyecto, porque sin duda alguna me fascina, y continuaré aprendiendo, diseñando y depurando todo aquello que me he dejado en el tintero.

Sin duda es duro y fascinante.

Que en el recuerdo queden los buenos momentos vividos en estos años y todo lo que hemos aprendido. Que nos volvamos a encontrar en un futuro, sea el que sea, como compañeros.

Pedro Vicente López Bañón

Bibliografía

- [1] URL: [https://es.wikipedia.org/wiki/Replicaci%C3%B3n_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Replicaci%C3%B3n_(inform%C3%A1tica)) (véase página 12).
- [2] URL: <http://blog.opencloud.es/database-as-a-service/> (véase página 37).
- [3] URL: <https://unpocodejava.com/2017/05/30/que-es-openstack-sahara/> (véase página 38).
- [4] *Aprendizaje Automático*. URL: https://es.wikipedia.org/wiki/Aprendizaje_autom%C3%A1tico (véase página 12).
- [5] *CentOS*. URL: <https://es.wikipedia.org/wiki/CentOS> (véanse páginas 11, 27).
- [6] *Ciencia de datos*. URL: https://es.wikipedia.org/wiki/Ciencia_de_datos (véanse páginas 11, 19).
- [7] OpenStack Contributors. *OpenStack Security Guide*. openstack.org, 2018. URL: <https://docs.openstack.org/security-guide/> (véase página 43).
- [8] Debian. *Debian Image for OpenStack*. URL: <https://cdimage.debian.org/cdimage/openstack/current/> (véase página 59).
- [9] Antonio Cobos Domínguez. *Despliegue de arquitectura cloud basada en OpenStack y su integración con Chef sobre CentOS*. Sevilla: Universidad de Sevilla, 2014 (véanse páginas 20, 31).
- [10] OpenStack Foundation. *OpenStack Security Guide*. Havana, 2013 (véase página 43).
- [11] *Git*. URL: <https://es.wikipedia.org/wiki/Git> (véase página 28).
- [12] *Git Hub*. URL: <https://es.wikipedia.org/wiki/GitHub> (véase página 28).
- [13] *Inteligencia empresarial*. URL: https://es.wikipedia.org/wiki/Inteligencia_empresarial (véase página 11).

- [14] *Macrodatos*. URL: <https://es.wikipedia.org/wiki/Macrodatos> (véase página 11).
- [15] *OpenStack desde cero - KeyStone*. URL: <https://www.dbigcloud.com/cloud-computing/170-openstack-desde-cero-keystone.html> (véase página 31).
- [16] *Python*. URL: <https://es.wikipedia.org/wiki/Python> (véase página 27).
- [17] *rackspace. ¿Qué es OpenStack?* URL: <https://www.rackspace.com/es/library/what-is-openstack> (véase página 23).
- [18] *Red Hat Enterprise Linux*. URL: https://es.wikipedia.org/wiki/Red_Hat_Enterprise_Linux (véase página 12).
- [19] *Tipos de Cloud Computing*. Open Webinars. URL: <https://openwebinars.net/blog/tipos-de-cloud-computing/> (véase página 21).
- [20] *Ventajas del cloud Computing*. IBM. URL: <https://www.ibm.com/cloud-computing/es-es/learn-more/benefits-of-cloud-computing/> (véase página 20).

10. Anexos (CE 13)

10.1 Anexo A

Fichero de configuración de Packstack sin comentarios

```
1 [general]
2 CONFIG_SSH_KEY=/root/.ssh/id_rsa.pub
3 CONFIG_DEFAULT_PASSWORD=
4 CONFIG_SERVICE_WORKERS=%{:::processorcount}
5 CONFIG_MARIADB_INSTALL=y
6 CONFIG_GLANCE_INSTALL=y
7 CONFIG_CINDER_INSTALL=y
8 CONFIG_MANILA_INSTALL=n
9 CONFIG_NOVA_INSTALL=y
10 CONFIG_NEUTRON_INSTALL=y
11 CONFIG_HORIZON_INSTALL=y
12 CONFIG_SWIFT_INSTALL=y
13 CONFIG_CEILOMETER_INSTALL=y
14 CONFIG_AODH_INSTALL=y
15 CONFIG_PANKO_INSTALL=n
16 CONFIG_SAHARA_INSTALL=n
17 CONFIG_HEAT_INSTALL=n
18 CONFIG_MAGNUM_INSTALL=n
19 CONFIG_TROVE_INSTALL=n
20 CONFIG_IRONIC_INSTALL=n
21 CONFIG_CLIENT_INSTALL=y
22 CONFIG_NTP_SERVERS=
23 EXCLUDE_SERVERS=
24 CONFIG_DEBUG_MODE=n
25 CONFIG_CONTROLLER_HOST=172.21.1.51
26 CONFIG_COMPUTE_HOSTS=172.21.1.53
27 CONFIG_NETWORK_HOSTS=172.21.1.52
28 CONFIG_VMWARE_BACKEND=n
29 CONFIG_UNSUPPORTED=n
30 CONFIG_USE_SUBNETS=n
31 CONFIG_VCENTER_HOST=
32 CONFIG_VCENTER_USER=
33 CONFIG_VCENTER_PASSWORD=
34 CONFIG_VCENTER_CLUSTER_NAMES=
35 CONFIG_STORAGE_HOST=172.21.1.52
36 CONFIG_SAHARA_HOST=172.21.1.53
37 CONFIG_REPO=
38 CONFIG_ENABLE_RDO_TESTING=n
39 CONFIG_RH_USER=
40 CONFIG_SATELLITE_URL=
41 CONFIG_RH_SAT6_SERVER=
42 CONFIG_RH_PW=
43 CONFIG_RH_OPTIONAL=y
44 CONFIG_RH_PROXY=
45 CONFIG_RH_SAT6_ORG=
46 CONFIG_RH_SAT6_KEY=
47 CONFIG_RH_PROXY_PORT=
48 CONFIG_RH_PROXY_USER=
49 CONFIG_RH_PROXY_PW=
50 CONFIG_SATELLITE_USER=
51 CONFIG_SATELLITE_PW=
52 CONFIG_SATELLITE_AKEY=
53 CONFIG_SATELLITE_CACERT=
54 CONFIG_SATELLITE_PROFILE=
55 CONFIG_SATELLITE_FLAGS=
56 CONFIG_SATELLITE_PROXY=
```

```

57 CONFIG_SATELLITE_PROXY_USER=
58 CONFIG_SATELLITE_PROXY_PW=
59 CONFIG_SSL_CACERT_FILE=/etc/pki/tls/certs/selfcert.crt
60 CONFIG_SSL_CACERT_KEY_FILE=/etc/pki/tls/private/selfkey.key
61 CONFIG_SSL_CERT_DIR=/packstackca/
62 CONFIG_SSL_CACERT_SELFSIGN=y
63 CONFIG_SSL_CERT_SUBJECT_C=-
64 CONFIG_SSL_CERT_SUBJECT_ST=State
65 CONFIG_SSL_CERT_SUBJECT_L=City
66 CONFIG_SSL_CERT_SUBJECT_O=openstack
67 CONFIG_SSL_CERT_SUBJECT_OU=packstack
68 CONFIG_SSL_CERT_SUBJECT_CN=central.acendata.com
69 CONFIG_SSL_CERT_SUBJECT_MAIL=admin@central.acendata.com
70 CONFIG_AMQP_BACKEND=rabbitmq
71 CONFIG_AMQP_HOST=172.21.1.51
72 CONFIG_AMQP_ENABLE_SSL=n
73 CONFIG_AMQP_ENABLE_AUTH=n
74 CONFIG_AMQP_NSS_CERTDB_PW=PW_PLACEHOLDER
75 CONFIG_AMQP_AUTH_USER=amqp_user
76 CONFIG_AMQP_AUTH_PASSWORD=PW_PLACEHOLDER
77 CONFIG_MARIADB_HOST=172.21.1.51
78 CONFIG_MARIADB_USER=root
79 CONFIG_MARIADB_PW=380df20842ee47f7
80 CONFIG_KEYSTONE_DB_PW=9baa7cdf7740cb
81 CONFIG_KEYSTONE_DB_PURGE_ENABLE=True
82 CONFIG_KEYSTONE_REGION=RegionOne
83 CONFIG_KEYSTONE_ADMIN_TOKEN=0d7478be89504701a78d42a952ed3380
84 CONFIG_KEYSTONE_ADMIN_EMAIL=root@localhost
85 CONFIG_KEYSTONE_ADMIN_USERNAME=admin
86 CONFIG_KEYSTONE_ADMIN_PW=3d4b8514ea2f4dba
87 CONFIG_KEYSTONE_DEMO_PW=7bbf083c0d9f4227
88 CONFIG_KEYSTONE_API_VERSION=v3
89 CONFIG_KEYSTONE_TOKEN_FORMAT=FERNET
90 CONFIG_KEYSTONE_IDENTITY_BACKEND=sql
91 CONFIG_KEYSTONE_LDAP_URL=ldap://172.21.1.51
92 CONFIG_KEYSTONE_LDAP_USER_DN=
93 CONFIG_KEYSTONE_LDAP_USER_PASSWORD=
94 CONFIG_KEYSTONE_LDAP_SUFFIX=
95 CONFIG_KEYSTONE_LDAP_QUERY_SCOPE=one
96 CONFIG_KEYSTONE_LDAP_PAGE_SIZE=-1
97 CONFIG_KEYSTONE_LDAP_USER_SUBTREE=
98 CONFIG_KEYSTONE_LDAP_USER_FILTER=
99 CONFIG_KEYSTONE_LDAP_USER_OBJECTCLASS=
100 CONFIG_KEYSTONE_LDAP_USER_ID_ATTRIBUTE=
101 CONFIG_KEYSTONE_LDAP_USER_NAME_ATTRIBUTE=
102 CONFIG_KEYSTONE_LDAP_USER_MAIL_ATTRIBUTE=
103 CONFIG_KEYSTONE_LDAP_USER_ENABLED_ATTRIBUTE=
104 CONFIG_KEYSTONE_LDAP_USER_ENABLED_MASK=-1
105 CONFIG_KEYSTONE_LDAP_USER_ENABLED_DEFAULT=TRUE
106 CONFIG_KEYSTONE_LDAP_USER_ENABLED_INVERT=n
107 CONFIG_KEYSTONE_LDAP_USER_ATTRIBUTE_IGNORE=
108 CONFIG_KEYSTONE_LDAP_USER_DEFAULT_PROJECT_ID_ATTRIBUTE=
109 CONFIG_KEYSTONE_LDAP_USER_ALLOW_CREATE=n
110 CONFIG_KEYSTONE_LDAP_USER_ALLOW_UPDATE=n
111 CONFIG_KEYSTONE_LDAP_USER_ALLOW_DELETE=n
112 CONFIG_KEYSTONE_LDAP_USER_PASS_ATTRIBUTE=
113 CONFIG_KEYSTONE_LDAP_USER_ENABLED_EMULATION_DN=
114 CONFIG_KEYSTONE_LDAP_USER_ADDITIONAL_ATTRIBUTE_MAPPING=
115 CONFIG_KEYSTONE_LDAP_GROUP_SUBTREE=
116 CONFIG_KEYSTONE_LDAP_GROUP_FILTER=
117 CONFIG_KEYSTONE_LDAP_GROUP_OBJECTCLASS=
118 CONFIG_KEYSTONE_LDAP_GROUP_ID_ATTRIBUTE=
119 CONFIG_KEYSTONE_LDAP_GROUP_NAME_ATTRIBUTE=
120 CONFIG_KEYSTONE_LDAP_GROUP_MEMBER_ATTRIBUTE=
121 CONFIG_KEYSTONE_LDAP_GROUP_DESC_ATTRIBUTE=
122 CONFIG_KEYSTONE_LDAP_GROUP_ATTRIBUTE_IGNORE=
123 CONFIG_KEYSTONE_LDAP_GROUP_ALLOW_CREATE=n
124 CONFIG_KEYSTONE_LDAP_GROUP_ALLOW_UPDATE=n
125 CONFIG_KEYSTONE_LDAP_GROUP_ALLOW_DELETE=n
126 CONFIG_KEYSTONE_LDAP_GROUP_ADDITIONAL_ATTRIBUTE_MAPPING=
127 CONFIG_KEYSTONE_LDAP_USE_TLS=n
128 CONFIG_KEYSTONE_LDAP_TLS_CACERTDIR=
129 CONFIG_KEYSTONE_LDAP_TLS_CACERTFILE=
130 CONFIG_KEYSTONE_LDAP_TLS_REQ_CERT=demand
131 CONFIG_GLANCE_DB_PW=bc6d1705f73400a
132 CONFIG_GLANCE_KS_PW=b78c41fdea6d48b1
133 CONFIG_GLANCE_BACKEND=file
134 CONFIG_CINDER_DB_PW=53a8317f100842b5
135 CONFIG_CINDER_DB_PURGE_ENABLE=True
136 CONFIG_CINDER_KS_PW=ac91ada1f99c4adc
137 CONFIG_CINDER_BACKEND=lvm
138 CONFIG_CINDER_VOLUMES_CREATE=y
139 CONFIG_CINDER_VOLUME_NAME=cinder-volumes
140 CONFIG_CINDER_VOLUMES_SIZE=20G
141 CONFIG_CINDER_GLUSTER_MOUNTS=
142 CONFIG_CINDER_NFS_MOUNTS=
143 CONFIG_CINDER_NETAPP_LOGIN=
144 CONFIG_CINDER_NETAPP_PASSWORD=
145 CONFIG_CINDER_NETAPP_HOSTNAME=
146 CONFIG_CINDER_NETAPP_SERVER_PORT=80
147 CONFIG_CINDER_NETAPP_STORAGE_FAMILY=ontap_cluster
148 CONFIG_CINDER_NETAPP_TRANSPORT_TYPE=http
149 CONFIG_CINDER_NETAPP_STORAGE_PROTOCOL=nfs
150 CONFIG_CINDER_NETAPP_SIZE_MULTIPLIER=1.0
151 CONFIG_CINDER_NETAPP_EXPIRY_THRES_MINUTES=720

```



```

152 CONFIG_CINDER_NETAPP_THRES_AVL_SIZE_PERC_START=20
153 CONFIG_CINDER_NETAPP_THRES_AVL_SIZE_PERC_STOP=60
154 CONFIG_CINDER_NETAPP_NFS_SHARES=
155 CONFIG_CINDER_NETAPP_NFS_SHARES_CONFIG=/etc/cinder/shares.conf
156 CONFIG_CINDER_NETAPP_VOLUME_LIST=
157 CONFIG_CINDER_NETAPP_VFILER=
158 CONFIG_CINDER_NETAPP_PARTNER_BACKEND_NAME=
159 CONFIG_CINDER_NETAPP_VSERVER=
160 CONFIG_CINDER_NETAPP_CONTROLLER_IPS=
161 CONFIG_CINDER_NETAPP_SA_PASSWORD=
162 CONFIG_CINDER_NETAPP_ESERIES_HOST_TYPE=linux_dm_mp
163 CONFIG_CINDER_NETAPP_WEBSERVICE_PATH=/devmgr/v2
164 CONFIG_CINDER_NETAPP_STORAGE_POOLS=
165 CONFIG_CINDER_SOLIDFIRE_LOGIN=
166 CONFIG_CINDER_SOLIDFIRE_PASSWORD=
167 CONFIG_CINDER_SOLIDFIRE_HOSTNAME=
168 CONFIG_IRONIC_DB_PW=PW_PLACEHOLDER
169 CONFIG_IRONIC_KS_PW=PW_PLACEHOLDER
170 CONFIG_NOVA_DB_PURGE_ENABLE=True
171 CONFIG_NOVA_DB_PW=59add9394e0f4090
172 CONFIG_NOVA_KS_PW=01d5d3f2db56409a
173 CONFIG_NOVA_MANAGE_FLAVORS=y
174 CONFIG_NOVA_SCHED_CPU_ALLOC_RATIO=16.0
175 CONFIG_NOVA_SCHED_RAM_ALLOC_RATIO=1.5
176 CONFIG_NOVA_COMPUTE_MIGRATE_PROTOCOL=ssh
177 CONFIG_VNC_SSL_CERT=
178 CONFIG_VNC_SSL_KEY=
179 CONFIG_NOVA_PCI_ALIAS=
180 CONFIG_NOVA_PCI_PASSTHROUGH_WHITELIST=
181 CONFIG_NOVA_LIBVIRT_VIRT_TYPE=%{::default_hypervisor}
182 CONFIG_NEUTRON_KS_PW=023dc9a0652f4c08
183 CONFIG_NEUTRON_DB_PW=0b1f2ac106c04a60
184 CONFIG_NEUTRON_L3_EXT_BRIDGE=br-ex
185 CONFIG_NEUTRON_METADATA_PW=ba5d9825c2774eb6
186 CONFIG_LBAAS_INSTALL=n
187 CONFIG_NEUTRON_METERING_AGENT_INSTALL=y
188 CONFIG_NEUTRON_FWAAS=n
189 CONFIG_NEUTRON_VPNAAS=n
190 CONFIG_NEUTRON_ML2_TYPE_DRIVERS=vxlan,flat
191 CONFIG_NEUTRON_ML2_TENANT_NETWORK_TYPES=vxlan
192 CONFIG_NEUTRON_ML2_MECHANISM_DRIVERS=openvswitch
193 CONFIG_NEUTRON_ML2_FLAT_NETWORKS=*
194 CONFIG_NEUTRON_ML2_VLAN_RANGES=
195 CONFIG_NEUTRON_ML2_TUNNEL_ID_RANGES=
196 CONFIG_NEUTRON_ML2_VXLAN_GROUP=
197 CONFIG_NEUTRON_ML2_VNI_RANGES=10:100
198 CONFIG_NEUTRON_L2_AGENT=openvswitch
199 CONFIG_NEUTRON_ML2_SUPPORTED_PCI_VENDOR_DEVS=['15b3:1004', '8086:10ca']
200 CONFIG_NEUTRON_ML2_SRIOV_INTERFACE_MAPPINGS=
201 CONFIG_NEUTRON_LB_INTERFACE_MAPPINGS=
202 CONFIG_NEUTRON_OVS_BRIDGE_MAPPINGS=extnet:br-ex
203 CONFIG_NEUTRON_OVS_BRIDGE_IFACES=br-ex:eth0
204 CONFIG_NEUTRON_OVS_BRIDGES_COMPUTE=
205 CONFIG_NEUTRON_OVS_EXTERNAL_PHYSNET=extnet
206 CONFIG_NEUTRON_OVS_TUNNEL_IF=
207 CONFIG_NEUTRON_OVS_TUNNEL_SUBNETS=
208 CONFIG_NEUTRON_OVS_VXLAN_UDP_PORT=4789
209 CONFIG_NEUTRON_OVN_BRIDGE_MAPPINGS=extnet:br-ex
210 CONFIG_NEUTRON_OVN_BRIDGE_IFACES=
211 CONFIG_NEUTRON_OVN_BRIDGES_COMPUTE=
212 CONFIG_NEUTRON_OVN_EXTERNAL_PHYSNET=extnet
213 CONFIG_NEUTRON_OVN_TUNNEL_IF=
214 CONFIG_NEUTRON_OVN_TUNNEL_SUBNETS=
215 CONFIG_MANILA_DB_PW=PW_PLACEHOLDER
216 CONFIG_MANILA_KS_PW=PW_PLACEHOLDER
217 CONFIG_MANILA_BACKEND=generic
218 CONFIG_MANILA_NETAPP_DRV_HANDLES_SHARE_SERVERS=false
219 CONFIG_MANILA_NETAPP_TRANSPORT_TYPE=https
220 CONFIG_MANILA_NETAPP_LOGIN=admin
221 CONFIG_MANILA_NETAPP_PASSWORD=
222 CONFIG_MANILA_NETAPP_SERVER_HOSTNAME=
223 CONFIG_MANILA_NETAPP_STORAGE_FAMILY=ontap_cluster
224 CONFIG_MANILA_NETAPP_SERVER_PORT=443
225 CONFIG_MANILA_NETAPP_AGGREGATE_NAME_SEARCH_PATTERN=(.*)
226 CONFIG_MANILA_NETAPP_ROOT_VOLUME_AGGREGATE=
227 CONFIG_MANILA_NETAPP_ROOT_VOLUME_NAME=root
228 CONFIG_MANILA_NETAPP_VSERVER=
229 CONFIG_MANILA_GENERIC_DRV_HANDLES_SHARE_SERVERS=true
230 CONFIG_MANILA_GENERIC_VOLUME_NAME_TEMPLATE=manila-share-%s
231 CONFIG_MANILA_GENERIC_SHARE_MOUNT_PATH=/shares
232 CONFIG_MANILA_SERVICE_IMAGE_LOCATION=https://www.dropbox.com/s/vi5oeh10q1qkckh/ubuntu_1204_nfs_cifs.qcow2
233 CONFIG_MANILA_SERVICE_INSTANCE_USER=ubuntu
234 CONFIG_MANILA_SERVICE_INSTANCE_PASSWORD=ubuntu
235 CONFIG_MANILA_NETWORK_TYPE=neutron
236 CONFIG_MANILA_NETWORK_STANDALONE_GATEWAY=
237 CONFIG_MANILA_NETWORK_STANDALONE_NETMASK=
238 CONFIG_MANILA_NETWORK_STANDALONE_SEG_ID=
239 CONFIG_MANILA_NETWORK_STANDALONE_IP_RANGE=
240 CONFIG_MANILA_NETWORK_STANDALONE_IP_VERSION=4
241 CONFIG_MANILA_GLUSTERFS_SERVERS=
242 CONFIG_MANILA_GLUSTERFS_NATIVE_PATH_TO_PRIVATE_KEY=
243 CONFIG_MANILA_GLUSTERFS_VOLUME_PATTERN=
244 CONFIG_MANILA_GLUSTERFS_TARGET=
245 CONFIG_MANILA_GLUSTERFS_MOUNT_POINT_BASE=
246 CONFIG_MANILA_GLUSTERFS_NFS_SERVER_TYPE=gluster

```


29		aggregate create	124		recordset list	
30		aggregate delete	125		recordset set	
31		aggregate list	126		recordset show	
32		aggregate remove host	127		tld create	
33		aggregate set	128		tld delete	
34		aggregate show	129		tld list	
35		aggregate unset	130		tld set	
36		compute agent create	131		tld show	
37		compute agent delete	132		tsigkey create	
38		compute agent list	133		tsigkey delete	
39		compute agent set	134		tsigkey list	
40		compute service delete	135		tsigkey set	
41		compute service list	136		tsigkey show	
42		compute service set	137		zone abandon	
43		console log show	138		zone axfr	
44		console url show	139		zone blacklist create	
45		flavor create	140		zone blacklist delete	
46		flavor delete	141		zone blacklist list	
47		flavor list	142		zone blacklist set	
48		flavor set	143		zone blacklist show	
49		flavor show	144		zone create	
50		flavor unset	145		zone delete	
51		host list	146		zone export create	
52		host set	147		zone export delete	
53		host show	148		zone export list	
54		hypervisor list	149		zone export show	
55		hypervisor show	150		zone export showfile	
56		hypervisor stats show	151		zone import create	
57		ip fixed add	152		zone import delete	
58		ip fixed remove	153		zone import list	
59		ip floating add	154		zone import show	
60		ip floating remove	155		zone list	
61		keypair create	156		zone set	
62		keypair delete	157		zone show	
63		keypair list	158		zone transfer accept list	
64		keypair show	159		zone transfer accept request	
65		server add fixed ip	160		zone transfer accept show	
66		server add floating ip	161		zone transfer request create	
67		server add network	162		zone transfer request delete	
68		server add port	163		zone transfer request list	
69		server add security group	164		zone transfer request set	
70		server add volume	165		zone transfer request show	
71		server backup create	166			
72		server create	167		-----+-----	
73		server delete	168		openstack.identity.v3	access token create
74		server dump create	169			catalog list
75		server event list	170			catalog show
76		server event show	171			consumer create
77		server group create	172			consumer delete
78		server group delete	173			consumer list
79		server group list	174			consumer set
80		server group show	175			consumer show
81		server image create	176			credential create
82		server list	177			credential delete
83		server lock	178			credential list
84		server migrate	179			credential set
85		server pause	180			credential show
86		server reboot	181			domain create
87		server rebuild	182			domain delete
88		server remove fixed ip	183			domain list
89		server remove floating ip	184			domain set
90		server remove network	185			domain show
91		server remove port	186			ec2 credentials create
92		server remove security group	187			ec2 credentials delete
93		server remove volume	188			ec2 credentials list
94		server rescue	189			ec2 credentials show
95		server resize	190			endpoint add project
96		server restore	191			endpoint create
97		server resume	192			endpoint delete
98		server set	193			endpoint list
99		server shelve	194			endpoint remove project
100		server show	195			endpoint set
101		server ssh	196			endpoint show
102		server start	197			federation domain list
103		server stop	198			federation project list
104		server suspend	199			federation protocol create
105		server unlock	200			federation protocol delete
106		server unpause	201			federation protocol list
107		server unrescue	202			federation protocol set
108		server unset	203			federation protocol show
109		server unshelve	204			group add user
110		usage list	205			group contains user
111		usage show	206			group create
112			207			group delete
113		-----+-----	208			group list
114		openstack.dns.v2	209			group remove user
115		dns quota list	210			group set
116		dns quota reset	211			group show
117		dns quota set	212			identity provider create
118		dns service list	213			identity provider delete
119		dns service show	214			identity provider list
120		ptr record list	215			identity provider set
121		ptr record set	216			identity provider show
122		ptr record show	217			implied role create
123		ptr record unset	218			implied role delete
		recordset create				implied role list
		recordset delete				

409		network qos rule list	504		firewall group create
410		network qos rule set	505		firewall group delete
411		network qos rule show	506		firewall group list
412		network qos rule type list	507		firewall group policy add rule
413		network qos rule type show	508		firewall group policy create
414		network rbac create	509		firewall group policy delete
415		network rbac delete	510		firewall group policy list
416		network rbac list	511		firewall group policy remove rule
417		network rbac set	512		firewall group policy set
418		network rbac show	513		firewall group policy show
419		network segment create	514		firewall group policy unset
420		network segment delete	515		firewall group rule create
421		network segment list	516		firewall group rule delete
422		network segment set	517		firewall group rule list
423		network segment show	518		firewall group rule set
424		network service provider list	519		firewall group rule show
425		network set	520		firewall group rule unset
426		network show	521		firewall group set
427		network unset	522		firewall group show
428		port create	523		firewall group unset
429		port delete	524		network log create
430		port list	525		network log delete
431		port set	526		network log list
432		port show	527		network log set
433		port unset	528		network log show
434		router add port	529		network loggable resources list
435		router add subnet	530		network subport list
436		router create	531		network trunk create
437		router delete	532		network trunk delete
438		router list	533		network trunk list
439		router remove port	534		network trunk set
440		router remove subnet	535		network trunk show
441		router set	536		network trunk unset
442		router show	537		sfc flow classifier create
443		router unset	538		sfc flow classifier delete
444		security group create	539		sfc flow classifier list
445		security group delete	540		sfc flow classifier set
446		security group list	541		sfc flow classifier show
447		security group rule create	542		sfc port chain create
448		security group rule delete	543		sfc port chain delete
449		security group rule list	544		sfc port chain list
450		security group rule show	545		sfc port chain set
451		security group set	546		sfc port chain show
452		security group show	547		sfc port chain unset
453		subnet create	548		sfc port pair create
454		subnet delete	549		sfc port pair delete
455		subnet list	550		sfc port pair group create
456		subnet pool create	551		sfc port pair group delete
457		subnet pool delete	552		sfc port pair group list
458		subnet pool list	553		sfc port pair group set
459		subnet pool set	554		sfc port pair group show
460		subnet pool show	555		sfc port pair group unset
461		subnet pool unset	556		sfc port pair list
462		subnet set	557		sfc port pair set
463		subnet show	558		sfc port pair show
464		subnet unset	559		sfc service graph create
465			560		sfc service graph delete
466	openstack.neutronclient.v2	bgp dragent add speaker	561		sfc service graph list
467		bgp dragent remove speaker	562		sfc service graph set
468		bgp peer create	563		sfc service graph show
469		bgp peer delete	564		vpn endpoint group create
470		bgp peer list	565		vpn endpoint group delete
471		bgp peer set	566		vpn endpoint group list
472		bgp peer show	567		vpn endpoint group set
473		bgp speaker add network	568		vpn endpoint group show
474		bgp speaker add peer	569		vpn ike policy create
475		bgp speaker create	570		vpn ike policy delete
476		bgp speaker delete	571		vpn ike policy list
477		bgp speaker list	572		vpn ike policy set
478		bgp speaker list advertised routes	573		vpn ike policy show
479		bgp speaker remove network	574		vpn ipsec policy create
480		bgp speaker remove peer	575		vpn ipsec policy delete
481		bgp speaker set	576		vpn ipsec policy list
482		bgp speaker show	577		vpn ipsec policy set
483		bgp speaker show dragents	578		vpn ipsec policy show
484		bgpvpn create	579		vpn ipsec site connection create
485		bgpvpn delete	580		vpn ipsec site connection delete
486		bgpvpn list	581		vpn ipsec site connection list
487		bgpvpn network association create	582		vpn ipsec site connection set
488		bgpvpn network association delete	583		vpn ipsec site connection show
489		bgpvpn network association list	584		vpn service create
490		bgpvpn network association show	585		vpn service delete
491		bgpvpn port association create	586		vpn service list
492		bgpvpn port association delete	587		vpn service set
493		bgpvpn port association list	588		vpn service show
494		bgpvpn port association set	589		
495		bgpvpn port association show	590	openstack.object_store.v1	container create
496		bgpvpn port association unset	591		container delete
497		bgpvpn router association create	592		container list

