

GRADO EN INGENIERÍA EN TECNOLOGÍA DE
TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Alumno: Ruiz García, Daniel

Directora: Astorga Burgo, Jasone

Curso: 2018-2019

Fecha: Bilbao, 18 de julio de 2019

ÍNDICES

Índice General

ÍNDICES.....	2
RESÚMENES.....	9
MEMORIA	11
1. Introducción	11
2. Contexto	13
2.1. NFV.....	13
2.2. Open Source MANO (OSM)	14
2.3. OpenStack	18
2.4. OSM y OpenStack	22
3. Objetivos y alcance del trabajo	24
4. Beneficios.....	26
4.1. Técnicos	26
4.2. Económicos	26
4.3. Sociales	26
5. Estado del arte	28
5.1. Cloud Computing.....	28
5.2. Openstack.....	30

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

5.3. Open Source MANO.....	31
6. Análisis de las alternativas.....	33
6.1. Alternativas de para la instalación de OpenStack.....	33
6.2. Alternativas de instalación de OSM	35
7. Descripción de la solución.....	38
7.1. Diseño de la solución.....	38
7.2. Especificaciones técnicas del proyecto.....	40
7.3. Instalación OpenStack	41
7.4. Instalación directa de OSM.....	43
7.5. Integración OSM y OpenStack	43
METODOLOGÍA.....	48
1. Descripción de los resultados.....	48
1.1. Funcionamiento OpenStack.....	48
1.2. Funcionamiento de OSM sobre OpenStack.....	50
2. Descripción de tareas y entregables	54
3. Diagrama de Gantt / Cronograma	56
ASPECTOS ECONÓMICOS.....	58
1. Horas internas	58
2. Gastos.....	59
3. Amortizaciones	60
4. Coste total	60
CONCLUSIONES.....	62

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

BIBLIOGRAFÍA 65

ANEXO 68

Índice de Figuras

MEMORIA	11
Ilustración 2.1 - Arquitectura NFV	14
Ilustración 2.2 - NFV MANO architectural framework by ETSI.....	16
Ilustración 2.3 - OSM Project Architecture	17
Ilustración 2.4 - Servicios OpenStack	18
Ilustración 2.5 - Esquema General de la Arquitectura.....	23
Ilustración 5.1 - Tipos de servicios Cloud Computing	29
Ilustración 7.1 - Arquitectura de la solución	38
Ilustración 7.2 - Componentes de la implementación NFV	39
Ilustración 7.3 - OpenStack UI	42
Ilustración 7.4 - OSM UI	43
Ilustración 7.5 - Archivo de autenticación.....	44
Ilustración 7.6 - /etc/neutron/plugins/ml2/ml2_conf.ini configuration	45
Ilustración 7.7 - OpenStack KeyPairs	45
Ilustración 7.8 – Link de autenticación OpenStack	46
METODOLOGÍA.....	48
Ilustración 1.1 - Instancia lanzada	49
Ilustración 1.2 - Instancia 1 con IP 192.168.233.133	49
Ilustración 1.3 - 2 instancias en ejecución	49
Ilustración 1.4 - Topología con 2 instancias	50
Ilustración 1.5 - Instancia 2 realizando ping a 192.168.233.133.....	50

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Ilustración 1.6 - Network Service: Dos VNFs Cirros.....	51
Ilustración 1.7 - NS running and configured	51
Ilustración 1.8 - Topología NS OSM.....	52
Ilustración 1.9 - Topología NS OpenStack	52
Ilustración 1.10 - Instancia 1 IP 192.168.32.39	53
Ilustración 1.11 - Instancia 2 IP 192.168.32.254	53
Ilustración 1.12 - Ping de instancia 1 a instancia 2.....	53
Ilustración 3.1 - Diagrama de Gantt.....	56
ANEXO	68
Ilustración 1 - Añadir imágenes a OpenStack	68
Ilustración 2 – Parámetros de configuración de las imágenes	68
Ilustración 3 - Agregación de paquetes NFV	69
Ilustración 4 - VNF composer	70
Ilustración 5 - Ejemplo NFV	70
Ilustración 6 - Diseño VNF en modo texto.....	71
Ilustración 7 - Agregación paquetes NS	72
Ilustración 8 - NS composer	72
Ilustración 9 - Ejemplo NS	73
Ilustración 10 - Ejemplo en modo texto	73
Ilustración 11 - Lanzar instancia.....	74

Índice de Tablas

MEMORIA	11
Tabla 6.1 - Tabla de comparación métodos de instalación OpenStack	35
Tabla 6.1 - Tabla de comparación métodos de instalación OSM	37
ASPECTOS ECONÓMICOS	58
Tabla 1.1 - Horas de los Ingenieros	59
Tabla 1.2 - Coste de las Horas Internas	59
Tabla 2.1 - Coste de los Gastos.....	59
Tabla 3.1 - Coste de las Amortizaciones	60
Tabla 4.1 - Coste Total del Proyecto	60

RESÚMENES

El objetivo de este documento es implementar un escenario NFV, en el cual las funciones de red se encuentran virtualizadas, sobre una plataforma de Cloud Computing. Para ello se va a hacer uso especialmente de dos proyectos open source como son OSM para las funcionalidades NFV, y OpenStack para la capa de gestión de infraestructura en la nube. Además, se pretende implementar el escenario en un equipo personal sin la necesidad de grandes servidores, para fines didácticos y de experimentación, ya que estas tecnologías están orientadas a infraestructuras con gran cantidad de equipamiento y recursos.

This document will implement an NFV scenario, which is a scenario in which the network functions are virtualized, on a Cloud Computing platform. For this purpose, we will make use of 2 open source projects such as OSM for the NFV functionalities and OpenStack for the infrastructure management layer in the cloud. In addition, it is intended to implement the scenario in a personal equipment without the need for large servers, for didactic and experimental purposes, since these technologies are oriented to infrastructures with a large amount of equipment and resources.

Dokumentu honek NFV eszenatoki bat ezartzen saiatuko da. Eszenatoki honetan sare funtzioak birtualizatuta aurkitzen dira Cloud Computing plataforma baten gainean. Hau lortzeko bi open source proiektuak erabiliko dira: OSM, NFV funtzionalitateetarako eta OpenStack, hodei infraestruktura arduratzeko. Horrez gain, eszenatoki hau ordenagailu pertsonal batean ezarri nahiko dugu zerbitzariak erabili gabe.

MEMORIA

1. INTRODUCCIÓN

Recientemente en la industria actual de telecomunicaciones, está surgiendo un interés especial por el despliegue de infraestructuras donde la flexibilidad y la escalabilidad sean factores muy importantes a tener en cuenta; como por ejemplo en la movilidad con grandes anchos de banda que se prometen en las 5G. Es aquí donde la tecnología de Virtualización de Funciones de Red (NFV), y las Red Definidas por Software SDN cobran especial importancia, ya que permiten ofrecer de manera virtual y con mayor independencia del hardware, funcionalidades de red.

Actualmente hay varias tecnologías de las que cada vez se está oyendo más hablar y las cuales se prevé que en pocos años formen parte de nuestro día a día. De entre todas las tecnologías que están por llegar quizás las tecnologías 5G (Tecnologías inalámbricas de 5ª Generación), sean de las que más se oye hablar, o al menos son las que se espera que tengan un mayor impacto en nuestras vidas. Todas ellas tendrán unas especificaciones muy mejoradas con respecto a generaciones anteriores en términos de ancho de banda, consumo, seguridad, latencia, y movilidad entre otras. Además, independientemente de las tecnologías que están por venir, los usuarios cada vez están demandando mejores servicios de red, hasta puntos en los que es inevitable que la arquitectura actual se quede obsoleta. Y para lograr dicho rendimiento es necesario un rediseño completo de la red que permita afrontar dicha demanda.

Para satisfacer el incremento de la demanda las redes van evolucionando con cada vez mayor o mejor equipamiento, lo que supone mucha dependencia del hardware. Esto por un lado complica el diseño de la red porque hay gran variedad de alternativas en el mercado ante unas determinadas funcionalidades, cada una con diferentes maneras de integrarse, y muchas veces requiere de deshacerse del equipamiento previo, lo que supone pérdidas adicionales. Y por otro lado no es una solución óptima en términos de eficiencia en comparación con soluciones como las que proponen las redes basadas en NFV.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

De manera resumida NFV promete una nueva arquitectura en la que el despliegue de servicios será muy sencillo y rápido, además con mucha interoperabilidad debido al uso de equipamiento estandarizado, un muy alto grado de escalabilidad y de flexibilidad, y una óptima utilización de los recursos hardware que se están empleando, puesto que múltiples servicios podrán compartir equipamiento hardware.

De manera independiente, hoy en día se está observando una tendencia hacia lo que se conoce como Cloud Computing o computación en la nube. El Cloud Computing explicado de una manera sencilla, es una tecnología que permite acceder de forma remota a recursos alojados en equipos ajenos ya sea para ofrecer un software, almacenamiento de archivos o procesamiento de datos entre otros muchos servicios. El Cloud Computing es una alternativa al modelo tradicional de computadoras personales en las que instalamos nuestras aplicaciones y guardamos nuestros datos, a un modelo en el que a estas aplicaciones y datos se alojan en equipos externos, y se accede a ellos de manera remota. La nube, como se conoce a este término de manera más popular, ofrece a las personas y las empresas la capacidad tener a su alcance recursos de computación, con servicios de mantenimiento, de manera segura y con un fácil acceso bajo demanda.

Con todo esto, el objetivo de este proyecto es implementar un escenario NFV con una finalidad didáctica y de experimentación, ya que es un concepto que engloba muchas tecnologías que todavía se encuentran en desarrollo, con objeto de estudiar, analizar, y probar el comportamiento de dichas tecnologías. Las tecnologías involucradas en arquitecturas NFV estarán orientadas a equipamientos que manejen gran cantidad de recursos, es por ello que resulta de especial interés conseguir un escenario funcional en un equipo personal, y así no depender de estos recursos para lograr esta finalidad didáctica.

Concretamente se va a necesitar un software para la cata de Management and Orchestration (MANO). Y junto a este un gestor de los recursos virtualizados (VIM). Aprovechando la tendencia que está surgiendo hacia los servicios en Cloud Computing, se va a emplear un gestor VIM que está teniendo mucho éxito en la actualidad llamado OpenStack. Y para la capa MANO se va a emplear Open Source MANO, que es un proyecto que ofrece una solución para esta capa alojado en el ETSI.

2. CONTEXTO

En este apartado se va a explicar, en mayor detalle, algunas de las tecnologías mencionadas en la introducción que resultarán de especial importancia para el desarrollo del proyecto.

2.1. NFV

Estas tecnologías NFV que se han mencionado, consisten en un enfoque de red que permite la sustitución de dispositivos de hardware dedicados a funcionalidades específicas, como routers o firewalls entre otros, a dispositivos de red basados en software que se ejecutan de manera virtualizada. Esto permite a las operadoras una implantación de servicios mucho más sencilla, debido a que no depende de un determinado hardware, y en definitiva abaratar los costes del servicio como tal para la operadora. También supone una reducción de costes el hecho de que se reduce la complejidad de intercambio de información entre los servicios que se ofrecen y las aplicaciones de usuario como tal. Esto permitirá a las operadoras ofrecer una mayor velocidad a los usuarios, y por lo tanto ofrecer un mejor servicio.

Además, con la virtualización se consigue trasladar las funciones de un equipo que las realiza de manera específica, y que ha sido diseñado para proporcionar tales funciones, a otro equipo que no haya sido diseñado exclusivamente para tales funcionalidades, diseñado para fines de propósito general, que además sería de menor coste. En otras palabras, el software que se ejecuta en los equipos diseñados para unas determinadas funciones de red, ahora puede ejecutarse en otro equipo capaz de ejecutar muchos tipos de software y, además, tener en marcha varios simultáneamente.

En las tecnologías 5G las redes que implementen NFV, se espera que desempeñen un papel crucial, debido a que estas tecnologías prometen unas especificaciones que no se pueden lograr sin un rediseño basado en tecnologías como NFV que permitan mejorar el rendimiento de la red. En 5G, las NFV permitirán que una red física se separe en múltiples redes virtuales que puedan admitir diferentes redes de acceso de radio o varios tipos de servicios para ciertos segmentos de clientes. Los segmentos de red se aislarán entre sí en el plano de control y en el plano de usuario, por lo que la experiencia del usuario será la misma que si fuera una red físicamente separada. Esta división de la red jugará un papel crucial en las redes 5G debido a la multitud de casos de uso y los nuevos servicios que soportará 5G. Uno de los casos de uso principales del 5G NFV, será la banda ancha móvil más potente con menor latencia, pero

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

también brindará grandes beneficios en ancho de banda, movilidad, flexibilidad, seguridad y disponibilidad. Las futuras redes 5G ofrecerán a los operadores la flexibilidad de asignar velocidad, capacidad y cobertura en segmentos lógicos de acuerdo con las demandas de cada caso de uso.

La arquitectura propuesta por la ETSI, para las NFV es la mostrada a continuación en la ilustración. Esta consta de tres bloques principales que se van a explicar a continuación.

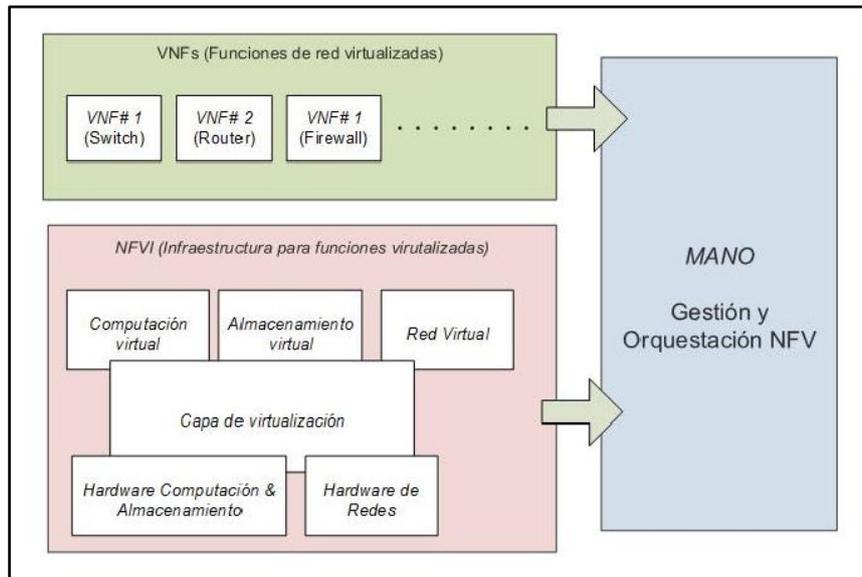


Ilustración 2.1 - Arquitectura NFV

Network Functions Virtualization Infrastructure – NFVI: Es la base de la arquitectura NFV completa y es la que contiene el hardware sobre el que se apoya la misma. Además, posee un software que posibilita la virtualización de los recursos, ofreciéndoselos como máquinas virtualizadas a la capa superior.

Virtualized Network Function – VNF: Utiliza los recursos que ofrece la capa NFVI para sobre estos construir funciones de red.

Management and Orchestration – MANO: Es un bloque separado, que interactúa con los otros dos bloques y que se encarga del control de los mismos. Este gestiona todos los recursos de la capa de infraestructura ya sea creando, borrando, o reservando espacio para las máquinas virtuales.

2.2. Open Source MANO (OSM)

Estas NFV necesitan de un cambio en la manera de administrar las redes en las que se hace uso de las mismas, y a esta solución es a la que se denomina

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

NFV Management and Network Orchestration o como más comúnmente se conoce, MANO. Este nuevo modo de administración que gestiona MANO abarca desde las configuraciones iniciales en la red, hasta sus operaciones del día a día. NFV MANO es un proyecto desarrollado por un grupo de trabajo de la ETSI (European Telecommunications Standards Institute) dentro del grupo ISG (Industry Specification Group) enfocados en el desarrollo de NFV o de manera resumida el grupo ETSI ISG NFV.

MANO es responsable de la orquestación y la administración completa del ciclo de vida de los recursos de hardware y las funciones de red virtual (VNF). En otras palabras, la capa MANO coordina los recursos de la infraestructura NFV (NFVI) y los asigna de manera eficiente a varios VNF.

OSM es un proyecto que se lanzó en 2016 que se ofrece como una solución MANO y que además está alojado por la propia ETSI. Hay varias opciones disponibles como pilas de software para MANO. Sin embargo, OSM se prefiere en gran medida debido a la gran actividad a nivel de la comunidad, así como a su marco altamente maduro, la preparación para la producción, la facilidad de implementación y la alimentación constante con los casos de uso por parte de los miembros.

Además, en cuanto a la arquitectura de OSM, como el proyecto está alojado en la ETSI, la comunidad de OSM intenta respetar las especificaciones del mismo. La arquitectura que propone la ETSI es la que se muestra en la siguiente Ilustración, en ella se puede apreciar que cuenta con 3 bloques principales que son los que se explican a continuación.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

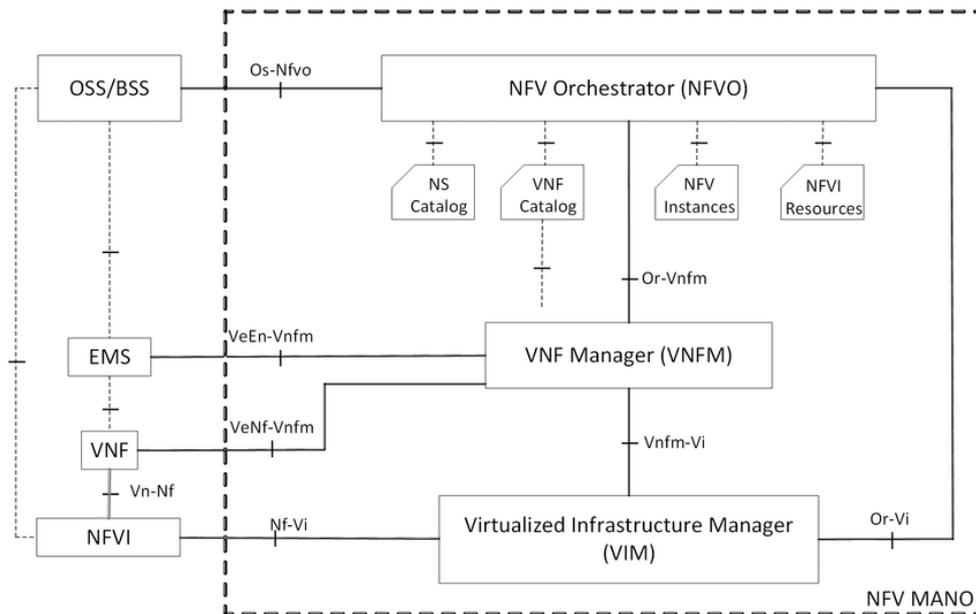


Ilustración 2.2 - NFV MANO architectural framework by ETSI

VNF Manager: Este bloque se encarga de supervisar la administración del ciclo de vida de las instancias de VNF; cumple la función de coordinación y adaptación para la configuración y el reporte de eventos entre la infraestructura NFV.

NFV Orchestrator: Este bloque es el responsable de la incorporación de nuevos paquetes de servicios de red (NS) y función de red virtual (VNF); Gestión de los ciclos de vida de los NS; gestión global de recursos; validación y autorización de las solicitudes de recursos por parte de la infraestructura de virtualización de funciones de red (NFVI).

Virtualized Infrastructure Manager (VIM): Se encarga de controlar y administrar los recursos de computación, de almacenamiento y de red de NFVI.

Y el proyecto El proyecto OSM consta de tres componentes básicos. Que se muestran en la siguiente ilustración.

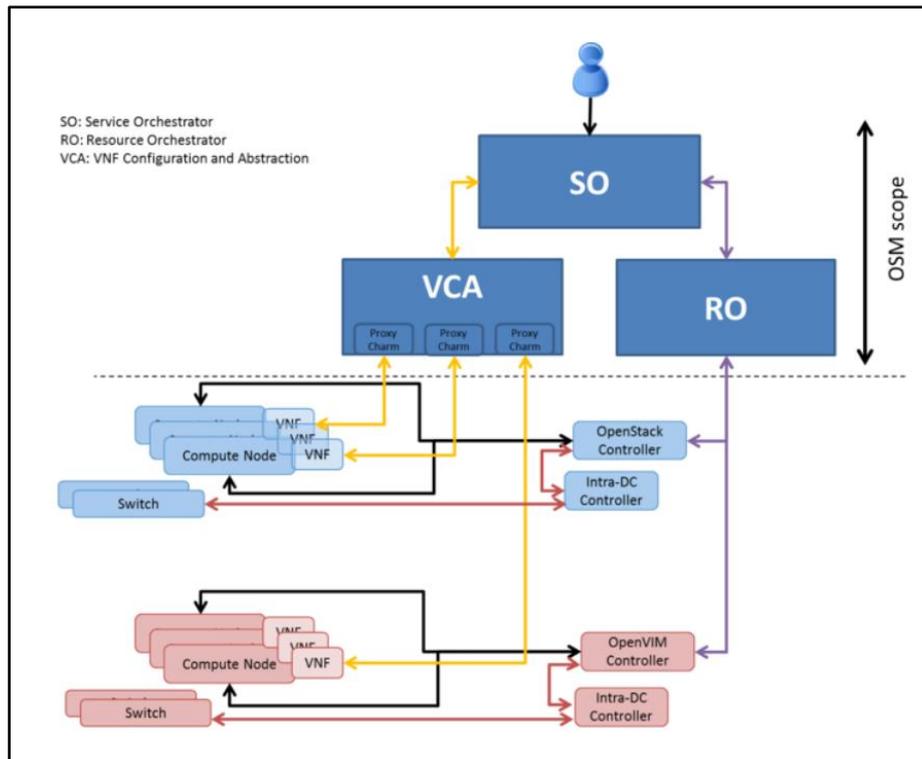


Ilustración 2.3 - OSM Project Architecture

El orquestador de servicios (SO): Este componente se encarga de la orquestación y el aprovisionamiento extremo a extremo. Además, almacena los VNF definidos, el catálogo de NS (Network Services), gestiona el flujo de trabajo de la implementación de los servicios, y consulta el estado de los servicios ya implementados.

El Resource Orchestrator (RO): Este componente se encarga del aprovisionamiento de servicios haciendo uso de un proveedor IaaS. Actualmente OSM tiene un soporte para OpenStack, OpenVIM, VMware y recientemente también han dado soporte a AWS (Amazon Web Service). Estos componentes SO y RO se encontrarían en la entidad NFVO de la arquitectura del ETSI.

El VNF Configuration and Abstraction (VCA): Este módulo realiza la configuración inicial del VNF. Este bloque se podría considerar como un VNFM con funcionalidades limitadas dentro de la arquitectura de la ETSI.

2.3. OpenStack

Como se acaba de mencionar en el apartado previo el bloque VIM forma parte de uno de los tres bloques funcionales principales de NFV MANO. El VIM es responsable de controlar y administrar: la NFVI (NFV Infraestructura, fijarse en la Ilustración 1), las capacidades de computación, el almacenamiento y los recursos de red. Y en este proyecto este bloque va a funcionar con OpenStack.

OpenStack es muy conocido hoy en día por ser el grupo más grande de proyectos de código abierto que forman una plataforma de software de manera colectiva para montar una infraestructura de Cloud Computing. Esta infraestructura es muy usada para casos de uso de nube privada, por muchas empresas. Después de la introducción de las NFV por la ETSI, OpenStack se ha convertido en una plataforma de mucha importancia para proporcionar una infraestructura para NFV. En la mayoría de las implementaciones de NFV, OpenStack se usa en la capa VIM, para brindar una interfaz estandarizada para administrar y monitorizar los recursos dentro de la infraestructura NFV.

OpenStack se compone de una serie de proyectos aliados que controlan grandes conjuntos de recursos de computación, almacenamiento y red en un centro de datos mientras se administra a través de un cuadro de mando. OpenStack se compone de nueve servicios principales, y está diseñado para añadir más si es necesario en un futuro.

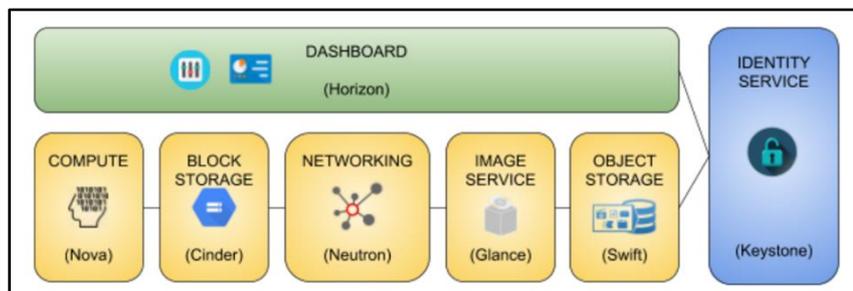


Ilustración 2.4 - Servicios OpenStack

Compute (Nova) - Es el elemento principal del sistema, y se encarga de controlar la estructura Cloud Computing. Se encarga de gestionar los pools de recursos y puede trabajar con diferentes tecnologías de virtualización como: KVM y Xen para las funciones de hypervisor, junto con Hyper-V, vSphere o LXC.

Object Storage (Swift) - Es el sistema de almacenamiento. Este permite un almacenamiento redundante y escalable, y se asegura de la replicación y de la

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

integridad de los datos. Permite el escalado horizontal del almacenamiento, añadiendo nuevos servidores. Para la redundancia del contenido, se encarga de que siempre haya una copia de los datos en otro servidor, y si una copia falla, réplica el contenido a otra nueva ubicación.

Block Storage (Cinder) - Proporciona un almacenamiento basado en bloques para utilizar con instancias de OpenStack Compute. Gestiona la creación, aplicación y eliminación de estos bloques, y permite a los usuarios gestionar sus necesidades de almacenamiento a través de la UI. Puede hacer uso tanto del sistema de almacenamiento local de linux, como de otras muchas plataformas de almacenamiento que se usan hoy en día.

Dashboard (Horizon) - Proporciona una interfaz gráfica para la administración del sistema por parte de los usuarios o administradores del sistema. Permite entre otras cosas automatización y provisión de recursos basados en el cloud. Además, permite servicios como facturación, monitorización o herramientas de gestión adicionales.

Servicio de Identidad (Keystone) - Es el servicio de Identificación de OpenStack ofrece a los usuarios un directorio a los servicios que estos pueden acceder. Y es común para todo el sistema OpenStack. Es también compatible con diferentes formas de autenticación como por credenciales o a través de nombre y usuario...etc.

Servicio de Imagen (Glance) - Proporciona servicios de gestión de imágenes de disco, así como los servicios de descubrimiento registro y entrega de las mismas al servicio de cómputo. Estas imágenes se pueden emplear como plantilla para las instancias o también para realizar copias de seguridad.

Telemetría (Ceilometer) - Permite la facturación de todos los recursos del sistema mediante el uso de los contadores, necesarios para monitorizar el consumo de los componentes de OpenStack. El sistema de motorización por contadores es fácilmente extensible para admitir posibles futuros proyectos.

Orquestación (Heat) - Permite la orquestación de múltiples aplicaciones instaladas en la nube a través del uso de plantillas.

Bare metal (Ironic) - Permite la integración de las maquinas físicas al sistema. Se comporta como una API de hipervisor para el hipervisor del equipo físico.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Base de datos (Trove) - Es una base de datos del sistema que se emplea para dar soporte a los diferentes servicios mediante el uso de bases de datos relacionales y no relacionales.

Almacenamiento seguro (Barbican) - Es una API empleada para el almacenamiento de manera segura. Permite almacenar secretos, contraseñas claves de cifrado y certificados.

Networking (Neutron) - Es un sistema que se encarga de la gestión de las redes y de las direcciones IP, se asegura de que la red no sea el factor limitante del sistema, y además permite a los usuarios configurarlas creando sus propias redes, controlando el tráfico y conectando los servidores y los dispositivos en la UI. Permite agregar servicios adicionales como sistemas de detección de intrusos, balanceos de carga, cortafuegos, redes privadas virtuales (VPN), o el uso de redes definidas por software apoyándose en software adicional como OpenFlow.

También permite asignación de IPs por DHCP, y asignación de IPs flotantes. Y, además, proporciona diferentes tipos de redes para adaptarse a las diferentes aplicaciones en función de las necesidades:

- Modelos de red plana que denomina FLAT, sin ningún tipo de segmentación.
- Modelos de red VLAN que permiten utilizar VLAN IDs usando el protocolo 802.1Q.
- Redes con protocolos de overlay como GRE y VXLAN que permiten la comunicación entre las diferentes instancias de computación.

También distingue otros dos tipos de redes dentro de OpenStack para garantizar un correcto funcionamiento del mismo:

Provider networks. Estas permiten la comunicación a nivel 2 con todas las instancias. Normalmente hacen uso de VLANs para la separación de las mismas.

Self-service networks. Que permiten la creación de redes en proyectos sin permisos de administrador, son redes de nivel 2 y que para acceder a redes externas o a provider networks necesitan de conectividad a nivel 3 a través de un routers. En muchos casos estas redes utilizan protocolos de overlay por que estos son compatibles con más tipos de redes.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Management networks. Es la red que se emplea para la comunicación entre los propios servicios de OpenStack. Esta red debe ser accesible solo desde los propios data center.

Guest networks. Se emplea para la comunicación de las máquinas virtuales implementadas en la nube.

External networks. Utilizadas para proporcionar conexión a internet a las VMs en determinados escenarios.

API networks. Se utilizan para acceder a las diferentes APIs de OpenStack incluyendo la propia API del servicio Networking.

2.4. OSM y OpenStack

Tanto OpenStack como OSM tienen una gran comunidad conocida por su rápido ritmo de innovación para NFV y la mayor contribución de grandes compañías para mejorar las características actuales y desarrollar los proyectos que son necesarios para el ecosistema de ambos.

Entre los proveedores de servicios de telecomunicaciones de hoy en día, y las empresas que han implementado su entorno NFV haciendo uso de OpenStack se incluyen: AT&T, China Mobile, SK Telecom, Ericsson, Deutsche Telekom, Comcast y Bloomberg.

Con las tres últimas versiones (3, 4 y 5), OSM ha evolucionado para admitir la integración para el enfoque nativo de la nube al habilitar los marcos de CI / CD en capas de orquestación. OSM orienta su trabajo para funcionar en la nube, lo cual lo hace idóneo para funcionar junto con OpenStack, que tiene una arquitectura probada para las nubes privadas y públicas.

La implementación de OSM en la infraestructura NFV se ha vuelto muy ágil y sencilla y, por otro lado, OpenStack es conocido por permitir la simplicidad para administrar infraestructuras virtualizadas y basadas en contenedores. Las organizaciones pueden obtener todos los beneficios de la integración como NFV MANO utilizando OSM y OpenStack debido a una administración y despliegue sencillos.

Un esquema general de la solución que se propone es la que se muestra a continuación.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

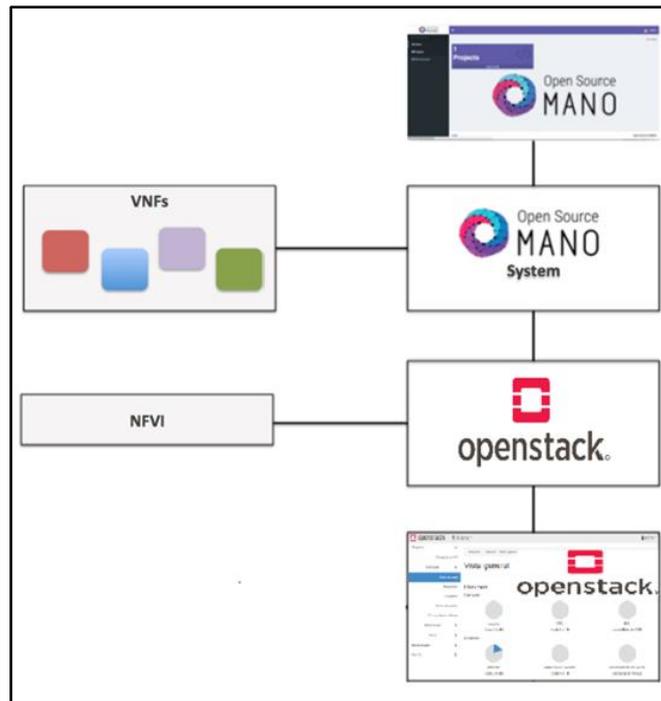


Ilustración 2.5 - Esquema General de la Arquitectura

3. OBJETIVOS Y ALCANCE DEL TRABAJO

Este proyecto tiene como objetivo principal desplegar una plataforma que permita implementar un escenario NFV haciendo uso de OSM para la capa MANO, y OpenStack para la capa VIM, en equipos convencionales, sin necesidad de grandes servidores. Ya que, la mayoría de las opciones del mercado están diseñadas para trabajar haciendo uso de gran cantidad de recursos de computación, y estas alternativas son imposibles de utilizar por las limitaciones del hardware, es por ello que para fines didácticos y de experimentación puede resultar de especial interés.

Para lograr dicho objetivo se han definido una serie de objetivos secundarios, los cuales juntos conforman el objetivo principal. Estos objetivos se enumeran de manera secuencial acorde con el orden en el que estos deben desarrollarse. Estos objetivos son:

➤ **Definición de los requerimientos**

El primer objetivo será realizar una definición de los requerimientos los cuales queremos que posea la solución que vamos a desarrollar. Para ello se realizará una descripción detallada de las condiciones que ha de cumplir, con sus correspondientes indicadores.

➤ **Estudio del estado del arte y análisis de las alternativas**

Para una óptima solución al objetivo principal del proyecto será necesario realizar un estudio de las diferentes alternativas que hay en el mercado que podrían ser una posible solución para nuestro objetivo. Así mismo será necesario escoger entre todas las posibles opciones en función de unos parámetros que se consideren oportunos.

➤ **Diseño de la solución**

Diseñar la solución al objetivo que se plantea teniendo en cuenta los objetivos que se habrán alcanzado previamente, es decir; prestando especial atención a los requisitos que se hayan definido, y haciendo uso de las alternativas que se hayan considerado óptimas para dar solución a nuestro problema.

➤ **Implementación de la solución**

Desarrollo de la solución que se ha sido diseñada previamente en la fase previa. En este proyecto este punto se compone de 2 subobjetivos que lo componen que son los siguientes:

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

- **Implementación de los módulos de manera independiente:** Para implementar el escenario que se pretende llevar a cabo, por un lado, es necesario implementar una plataforma de orquestación NFV, y por otro lado es necesario una implementar un gestor de infraestructuras virtuales de manera independiente.
- Una vez se tengan ambos módulos completamente funcionales habrá que realizar una **integración de ambos módulos** para que trabajen de manera conjunta.

➤ **Validación de la solución**

Una vez desarrollada la solución en el apartado anterior será necesario realizar una validación de la misma mediante la realización de las pruebas que se consideren oportunas para determinar si el sistema cumple con los requisitos que se han definido.

➤ **Documentación**

Por último, una vez se haya desarrollado y validado la solución, se procederá a realizar la documentación de todo el proyecto que se ha llevado a cabo, para dejar de forma escrita y detallada todos los pasos que han sido necesarios para cumplir con el objetivo principal del proyecto.

4. BENEFICIOS

Por un lado, son muchos los beneficios que aporta la realización de este trabajo, aunque el principal beneficio es el de poder experimentar, realizar pruebas y aprender de una plataforma NFV, cuya implementación está al alcance de cualquier persona con un equipo personal mínimamente potente; sin la necesidad de contar con grandes recursos de computación. Pero, por otro lado, la propia tecnología NFV, es una tecnología puntera hoy en día que aportara muchas mejoras en el ámbito de las telecomunicaciones y cuyos beneficios también aportaría este proyecto de forma indirecta pues dicha plataforma experimental facilitaría el desarrollo de la propia tecnología permitiendo una mejora de las propias NFV y todos los beneficios que estas conllevan.

4.1. Técnicos

- Permite fomentar la innovación en nuevos servicios de virtualización y en general en cualquier prestación de las NFV pues dicho despliegue facilitaría la accesibilidad, y fomentaría la investigación y desarrollo de sus prestaciones.
- También muy relacionado a la virtualización está la flexibilidad. Esta red puede adaptarse de manera sencilla a diferentes situaciones, como podría ser un escalado de la red, o una reducción de la capacidad de la misma.

4.2. Económicos

- Permitir experimentar en NFV ahorrando recursos de computación pues se reduce de manera considerable la infraestructura necesaria, y esto supone un ahorro en los costes
- Se reducirían mucho los gastos en equipamiento debido a que este se compondrá por un hardware estandarizado muy reutilizable, y de propósito general.
- Mayor eficiencia en cuanto a espacio, consumo y refrigerado; ya que ahora los equipos pueden ejecutar varios tipos de sistemas operativos, y esto permite que se aprovechen al máximo sus características y ahorrar directamente costes en equipamiento.

4.3. Sociales

- Otra característica muy importante que se consigue, es que en cuanto al tiempo de implementación de los servicios ya que ahora y todos estos servicios están virtualizados y por lo tanto se consigue realizar de manera

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

mucho más rápida lo que aumentaría el grado de satisfacción de los usuarios.

- La solución que se propone en este trabajo está basada en las soluciones de código abierto muy utilizadas, por lo que habría una amplia comunidad de software independiente, en la cual se estaría fomentando el desarrollo.
- Acercaría los conceptos de NFV a la sociedad, ya que estos serían mucho más accesibles para la gente que no es experta en el tema.
- Y además fomentar el desarrollo de estas tecnologías, supondría una mejora en las prestaciones de tecnologías futuras, las cuales, la sociedad disfrutará el día de mañana.

5. ESTADO DEL ARTE

En este apartado se va a extender la explicación de algunos de los conceptos y proyectos que se mencionan en el documento. Concretamente del Cloud Computing, OpenStack, y OSM.

5.1. Cloud Computing

El Cloud Computing consiste en la utilización de la capacidad de red para, sobre esta, ofrecer recursos de computación. La idea es que sus usuarios puedan acceder a estos recursos desde cualquier parte haciendo uso de Internet. El Cloud Computing es básicamente un modelo de prestación de servicios. Este permite al usuario disponer de las necesidades que exija su negocio, de manera flexible y adaptativa.

El Cloud Computing tiene su origen en los años sesenta. Joseph Carl Robnett Licklider, introdujo una idea de que todo el mundo pudiese estar conectado a través de una red y en la cual las personas pudiesen acceder a sus programas y datos desde cualquier lugar. Y John McCarthy, dijo que algún día los servicios de computación se ofrecerían como un servicio público.

En 1999 llegó Salesforce.com, una página pionera que entregaba aplicaciones a través de una página web, la cual empezó a allanar el camino al Cloud Computing. Posteriormente en 2002 llega una de las páginas de Cloud Computing más conocidas hoy en día Amazon Web Services, la cual ofrecía servicios de almacenamiento y computación entre otros. Y más adelante en 2006 ya lanzó su servicio Elastic Compute Cloud, el cual permitía a pequeñas empresas alquilar equipos que se ejecutaban en sus servidores siendo este el primer servicio de IaaS.

Más adelante en 2009 los gigantes tecnológicos Google y Microsoft se unieron al mundo del Cloud Computing empezando a ofrecer, entre otros servicios, aplicaciones basadas en navegadores. Lo cual empezó a crear un sentimiento de mayor aceptación de estos servicios online en los consumidores. Poco más adelante Microsoft sacó en 2010 al mercado Microsoft Azure para crear, probar, desplegar aplicaciones y servicios a través de data centers gestionados por Microsoft; este junto a AWS se convirtieron en las 2 únicas opciones del mercado siendo ambos líderes en el sector.

Hoy en día el Cloud Computing está muy extendido y numerosas empresas ofrecen estos servicios. Incluso algunas de estas ofrecen ciertos servicios de manera gratuita mientras no se excedan unos límites.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Aun con todo, el concepto de nube puede ser muy amplio, ya que casi cualquier servicio que se proporciona a través de Internet se apoya en equipos de computación. Con lo que podría abarcar casi todos los posibles tipos de servicio en línea, pero lo que hoy en día se está conociendo como Cloud Computing es a lo que se denomina alojamiento en la nube. Este alojamiento tiene 3 modalidades ofreciendo 3 tipos de servicio distinto que son: el Software como Servicio, Plataforma como Servicio, e Infraestructura como Servicio.

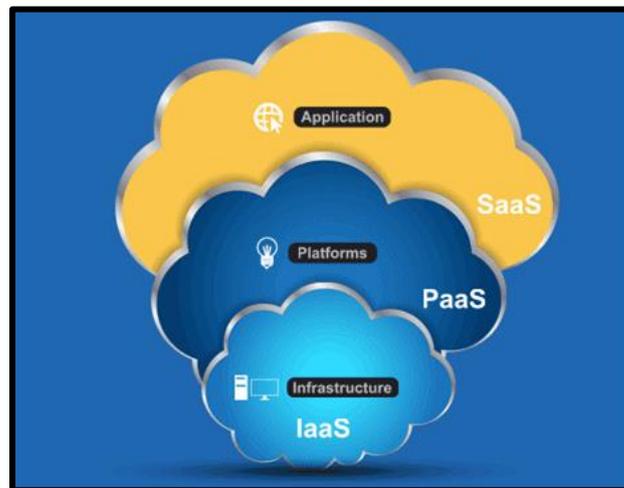


Ilustración 5.1 - Tipos de servicios Cloud Computing

El software como servicio (SaaS) es un modelo de distribución de software en el que las aplicaciones de usuario están alojadas en una compañía o proveedor de servicios. Y estas se sirven para la disposición de los usuarios a través de Internet.

La plataforma como servicio (PaaS) consiste en el ofrecimiento a los usuarios de sistemas operativos en funcionamiento o servicios asociados similares a través de Internet.

La Infraestructura como Servicio (IaaS) se refiere al ofrecimiento de los recursos de computación de los equipos, como pueden ser: el almacenamiento, el hardware, los componentes de red...etc.

El cloud Computing es cada vez más utilizado, y se prevé que en un futuro los dispositivos personales ejecuten todo en la nube directamente. En general esto se debe a las ventajas que esto conllevaría de las cuales a continuación enumeran algunas de las más importantes.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

El costo: Los recursos que se ofrecen en la nube no necesitan de la inversión inicial, cuando vas a comprar un equipo para usarlo de manera local. Esto en muchos casos permite ahorrar costes al que demanda los servicios.

Escalabilidad y elasticidad: Estos servicios permiten asignar o liberar recursos de manera muy dinámica, prácticamente en tiempo real. Esto permite aumentar o reducir los recursos de computación según las preferencias del cliente, y también en caso de estar ofreciendo un servicio, permite emplear los recursos justos que se demanden aumentando o reduciendo los recursos en función de la demanda de los mismos.

En cualquier lugar: Una de las principales ventajas es que puedes acceder a tus datos o aplicaciones en cualquier lugar independientemente de dónde te encuentres. Con el único requisito de disponer de una conexión a Internet.

Rendimiento: Los proveedores de estos servicios cuentan con mecanismos que controlan y optimizan el uso de sus recursos, permitiendo emplear en cada momento, solo los recursos que se necesiten y no malgastar recursos adicionales. Esto en gran medida es gracias a las tecnologías de virtualización que permiten compartir servidores físicos, y migrar de unos a otros con facilidad.

Seguridad: la centralización de todo el equipamiento facilita la seguridad del mismo. Esto es debido a que los proveedores pueden hacer uso de sistemas de seguridad los cuales un cliente de manera local no podría permitirse. El proveedor garantiza la seguridad del equipamiento físico.

Mantenimiento: se facilita el mantenimiento debido a que las aplicaciones no se ejecutan en los equipos de usuario, estos solo acceden de manera remota. El hecho de que las aplicaciones estén instaladas en los servidores de los proveedores permite garantizar su correcto funcionamiento, por parte de los mismos.

5.2. Openstack

OpenStack es un proyecto de Cloud Computing orientado a ofrecer una arquitectura como servicio (IaaS), que es de software libre y código abierto bajo la licencia de Apache 2.0. El proyecto está gestionado por la Fundación OpenStack que se fundó en 2012, y no tiene ánimos de lucro. El software está creado por una próspera comunidad de desarrolladores, en colaboración con los usuarios, y el diseño está completamente abierto al alcance de todos.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

OpenStack se compone de una gran comunidad con más de 9000 colaboradores y casi 500 compañías desde su lanzamiento inicial en 2010, por la NASA y Rackspace. Actualmente cuenta con la colaboración de grandes empresas como IBM, Intel, AMD, o Huawei entre otras. A su vez el proyecto se compone de muchos subproyectos, que se encargan de funciones como el control de procesamiento, control del almacenamiento, de las funciones de red...etc.

OpenStack proporciona soluciones de Cloud Computing muy flexibles, pues es capaz de proporcionar servicios en grandes redes públicas, y al mismo tiempo se puede implementar OpenStack de forma privada en máquinas de apenas unos pocos recursos.

Varios proyectos de OpenStack (Tacker, Neutron, Nova, Astaro, Congress, Mistral y Senlin entre otros) son capaces de administrar componentes de infraestructura virtualizados de entornos NFV. Como ejemplo, Tacker se utiliza para crear un administrador de VNF genérico (VNFM) y un orquestador de NFV (NFVO), que ayudan en la implementación y operación de VNF dentro de la infraestructura de NFV. Además, la integración de los proyectos de OpenStack presenta varias funcionalidades a la infraestructura NFV, como por ejemplo características de rendimiento, asignación de CPU, topología NUMA (para mejorar el rendimiento del acceso a memoria en escenarios multiprocesadores) o SR-IOV (para la virtualización de puertos físicos de entrada o salida en varios puertos lógicos virtuales). También, servicios de encadenamiento de funciones, división de redes, escalabilidad, alta disponibilidad, flexibilidad y habilitación multisitio; entre otras muchas.

OpenStack se usa para ofertas de soluciones comerciales de compañías como Canonical / Ubuntu, Cisco Systems, Ericsson, Huawei, IBM, Juniper, Mirantis, Red Hat, SUSE, VMware o Wind River. Un gran porcentaje de las implementaciones de VIM se basan en OpenStack debido a la simplicidad en el manejo y la operación de varios proyectos, los cuales, requieren del mayor potencial posible en lo que se refiere a almacenamiento, cómputo y NFV, motivo por el cual es una de las mejores opciones a tener en cuenta.

5.3. Open Source MANO

Open Source MANO (OSM) es un proyecto que se creó en 2016 en el ETSI (Telecommunications Standards Institute) para desarrollar un software Open Source NFV y que haga las funciones MANO de Gestión y Orquestación acorde al ETSI NFV. Este proyecto fue creado con el apoyo de 25 empresas como:

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Huawei, VMware, Canonical, o Telefónica entre otros y actualmente cuenta ya con más de 300 empresas colaboradoras.

El objetivo de OSM es la creación de software libre para dar una solución a la capa MANO de la arquitectura estandarizada por la ETSI para las NFV. Además, el propio proyecto OSM está alojado en el propio ETSI, por lo que trata especialmente de cumplir sus estándares. Nace con el fin de satisfacer los requisitos MANO de las redes NFV, impulsado por las empresas de operadores, y como un proyecto open source.

La base del código ya es capaz de gestionar y orquestar casos complejos de uso de NFV, y realizar todas las funcionalidades más significativas de los servicios end-to-end. Actualmente OSM cubre funcionalidades desde la orquestación de recursos y servicios, hasta la gestión de los ciclos de vida de los servicios de red.

Entre algunos de los objetivos de OSM se encuentra el trabajar con la comunidad open source para mejorar la calidad del mismo haciendo uso de la licencia de apache. Un código open source permite crear modelos de información con un alto grado de certeza en su correcto funcionamiento, lo que permite ofrecer soluciones de manera más rápida y económica.

El proyecto aprovecha la sinergia entre los procesos de estandarización y el software open source, accediendo a un conjunto más grande y variado de desarrolladores y participantes que el que se consigue normalmente.

6. ANÁLISIS DE LAS ALTERNATIVAS

En este apartado se van a analizar las diferentes alternativas que podrían ser posibles soluciones para los objetivos que se han planteado. Y se seleccionará de entre todas, la mejor alternativa en función de los criterios que se considere oportunos.

6.1. Alternativas de para la instalación de OpenStack

Son muchas las alternativas con la que se puede lanzar OpenStack. OpenStack es ampliamente utilizada en lo que se refiere al Cloud Computing, tanto que incluso algunos productos, que en principio podrían ser una competencia para OpenStack, como podría ser la virtualización de VMware, en vez de desarrollar su propia solución Cloud Computing, han optado por desarrollar su propia versión de OpenStack.

Hay muchas alternativas, pero si de entre todas ellas escogemos solo las que permiten una instalación con recursos limitados, las opciones se reducen bastante hasta el punto que solo se han encontrado tres opciones que son PackStack, DevStack y MicroStack.

6.1.1. PACKSTACK

PackStack no es una distribución de OpenStack como tal. Si no que es un método de instalación proporcionado por un proyecto denominado RDO. Que no es más que una comunidad de gente que quiere facilitar y promover el uso de OpenStack. Para ello lo que hacen es simplificar mucho el proceso de instalación a través de unos scripts que desarrolla la comunidad a los que se conoce como PackStack. Estos scripts se pueden ejecutar en sistemas operativos Red Hat o similares. Y proporcionan los ser servicios Nova, Keystone, Cinder, Glance, Horizon y Neutron.

Los requerimientos mínimos para ejecutar PackStack en un solo nodo son los que se muestran a continuación. Y se recomienda que el proceso de instalación se realice en una maquina cuyo objetivo sea únicamente el uso de OpenStack, pero no es obligatorio.

- 2GB de RAM.
- 50GB de almacenamiento en disco.
- Procesador 64 bits.
- Sistema Operativo Red Hat, CentOS, o Fedora.

6.1.2. DEVSTACK

DevStack es muy similar a la alternativa de RDO proyect con PackStack. No es una distribución como tal si no que está orientada a poder usar OpenStack de manera sencilla. Es por eso que DevStack se compone también unos scripts que lo que hacen es facilitar su instalación al igual que PackStack. Pero con la principal diferencia de que estos se realizan para sistemas operativos basados en Debian, como por ejemplo Ubuntu, motivo por el cual también posee una amplia comunidad detrás. Al igual que PackStack, DevStack proporciona los servicios Nova, Keystone, Cinder, Glance, Horizon y Neutron.

Los requerimientos mínimos para el funcionamiento de DevStack en un solo nodo son los que se muestran a continuación. Y también se recomienda que la maquina en la que se instala sea para uso exclusivo de OpenStack.

- 1GB de RAM.
- 1 núcleos de procesador.
- 60GB del almacenamiento en disco.
- Sistema Operativo Ubuntu, o LinuxMint.

6.1.3. MICROSTACK

MicroStack es una solución que permite instalar OpenStack de una manera muy sencilla, a través de la instalación de un paquete snap. MicroStack puede ejecutarse localmente en una sola máquina, actualmente proporciona los servicios Nova, Keystone, Glance, Horizon y Neutron. Por lo que de manera muy sencilla puedes conseguir tener OpenStack completamente funcional.

No se cuenta con mucha documentación sobre los requisitos o las recomendaciones para la instalación del mismo.

6.1.4. SELECCIÓN

En cuanto a la selección de la mejor alternativa se ha escogido en base a 3 aspectos principales que son:

- Sencillez de instalación: Se valorará métodos de instalación más sencillos con respecto a métodos más complejos.
- Compatibilidad Ubuntu: La compatibilidad con el sistema operativo Ubuntu se valorará con respecto a propuestas que necesiten sistemas operativos distintos.
- Exigencia de recursos: Se valorarán las propuestas que mejor funcionen mejor en equipos personales.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

A continuación, se procederá a evaluar las diferentes alternativas en base a los 3 criterios mencionados, ponderando cada uno de ellos en función de la importancia que se le ha asignado a cada uno.

Criterio	PackStack	DevStack	MicroStack
Sencillez de instalación (30%)	4	4	8
Compatibilidad Ubuntu (30%)	No	Si	Si
Exigencias de recursos (40%)	6	8	3
Total	3,6	7,4	6,6

Tabla 6.1 - Tabla de comparación métodos de instalación OpenStack

La elección final como se aprecia en la tabla fue DevStack. Esto es debido principalmente a que era el que menos exigía de los 3. Ya que, aunque la instalación de MicroStack es muy sencilla este exige demasiados recursos.

6.2. Alternativas de instalación de OSM

En cuanto a las alternativas de instalación de OSM, solo se cuenta con las que se proporcionan en la página oficial a través de dos métodos diferentes. Aunque es cierto que la página sigue estando la posibilidad de descargar alguna de las versiones anteriores.

Inicialmente se optó por trabajar con la versión 5 de OSM ya que es la más actualizada, y si hubiera algún problema se podría probar con otra versión las cuales tienen un poco más de información en internet, debido a que OSM5 lleva menos tiempo disponible.

Ahora en la página ofrecen 2 maneras de instalación de OSM que son las que se describirá a continuación.

6.2.1. INSTALACIÓN OSM A TRAVÉS DE VAGRANT

Vagrant, que no es más que un gestor de clientes de virtualización, el cual te puede crear una máquina virtual con OSM, a través de VirtualBox u otros gestores de virtualización, ya previamente configurada. La principal ventaja de Vagrant es que proporciona la seguridad de que la MV que te va a crear va a estar correctamente configurada, ya que esta se va a realizar en base a una “box” configurada por los desarrolladores de OSM, de manera que se garantiza que la MV es igual a como los desarrolladores quieren que sea.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Además, la máquina que se crea es muy liviana, esta cuenta con apenas 512MB de RAM, y una CPU. Aunque en la propia página recomiendan más recursos para el uso de osm5 que se mencionaran más adelante.

6.2.2. INSTALACIÓN DIRECTA

El segundo modo de instalación que se ofrece, es una instalación directamente en la máquina en la que se trabaja. Es también bastante sencilla ya que se realiza en apenas 2 pasos. Además, si se realiza en una MV es prácticamente igual al método anterior. Pero con la ventaja de que la máquina se puede configurar como se desee conveniente. La desventaja de este método es que es más sencillo que entre en conflicto con alguna aplicación del sistema y haya que solucionarlo.

En cuanto a los requisitos mínimos recomendados que se nos proporciona en la página oficial para la RELEASE FIVE son:

- 4GB de RAM.
- 2 núcleos de procesador.
- 20GB de almacenamiento en disco.
- Sistema Operativo Ubuntu 16.04 LTS en su versión de 64 bits.

6.2.3. SELECCIÓN

La selección de la mejor alternativa se ha hecho en base a los siguientes tres criterios.

- Sencillez de la instalación: Se valorará métodos de instalación más sencillos con respecto a métodos muy complejos.
- Exigencia de recursos: Se valorarán las propuestas que mejor funcionen mejor en equipos personales.
- Grado de configuración: Se valorará el grado de configuración de la máquina de instalación.

A continuación, se procederá a evaluar las diferentes alternativas en base a los 3 criterios mencionados, ponderando cada uno de ellos en función de la importancia que se le ha asignado a cada uno.

Criterio	Con Vagrant	Directamente
Sencillez de instalación (30%)	9	7
Exigencias de recursos (20%)	8	6
Grado de configuración (50%)	5	9

Total	6,8	7,8
-------	-----	-----

Tabla 6.1 - Tabla de comparación métodos de instalación OSM

La elección final fue el método de instalación directo. Esto se debió principalmente a que permitía una mayor configuración de la maquina, especialmente en cuanto a la configuración de interfaces. Puesto que Vagrant proporciona una configuración mediante NAT que no facilita la comunicación con otras máquinas.

7. DESCRIPCIÓN DE LA SOLUCIÓN

Como ya se ha ido mencionando a lo largo del proyecto la solución que se ha realizado para conseguir implementar un escenario NFV, ha sido con OSM para la capa MANO, y OpenStack para realizar las funciones de VIM.

7.1. Diseño de la solución

La arquitectura general que se ha planteado para implementar la solución propuesta es la que se muestra en la siguiente Ilustración.

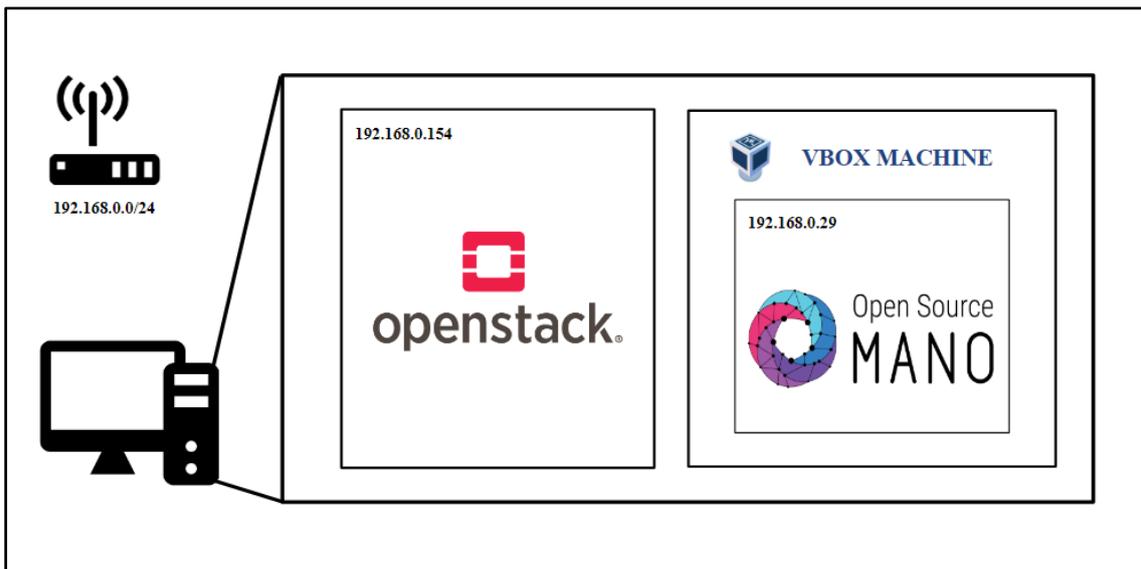


Ilustración 7.1 - Arquitectura de la solución

Se empleó un equipo de sobremesa que corría Ubuntu el cual ejecutaba OpenStack por un lado y una máquina virtual box Ubuntu con OSM por otro lado.

Y analizando la arquitectura NFV que se ha montado, para entender mejor como está funcionando todo y con ayuda del esquema NFV del ETSI se tiene lo siguiente.

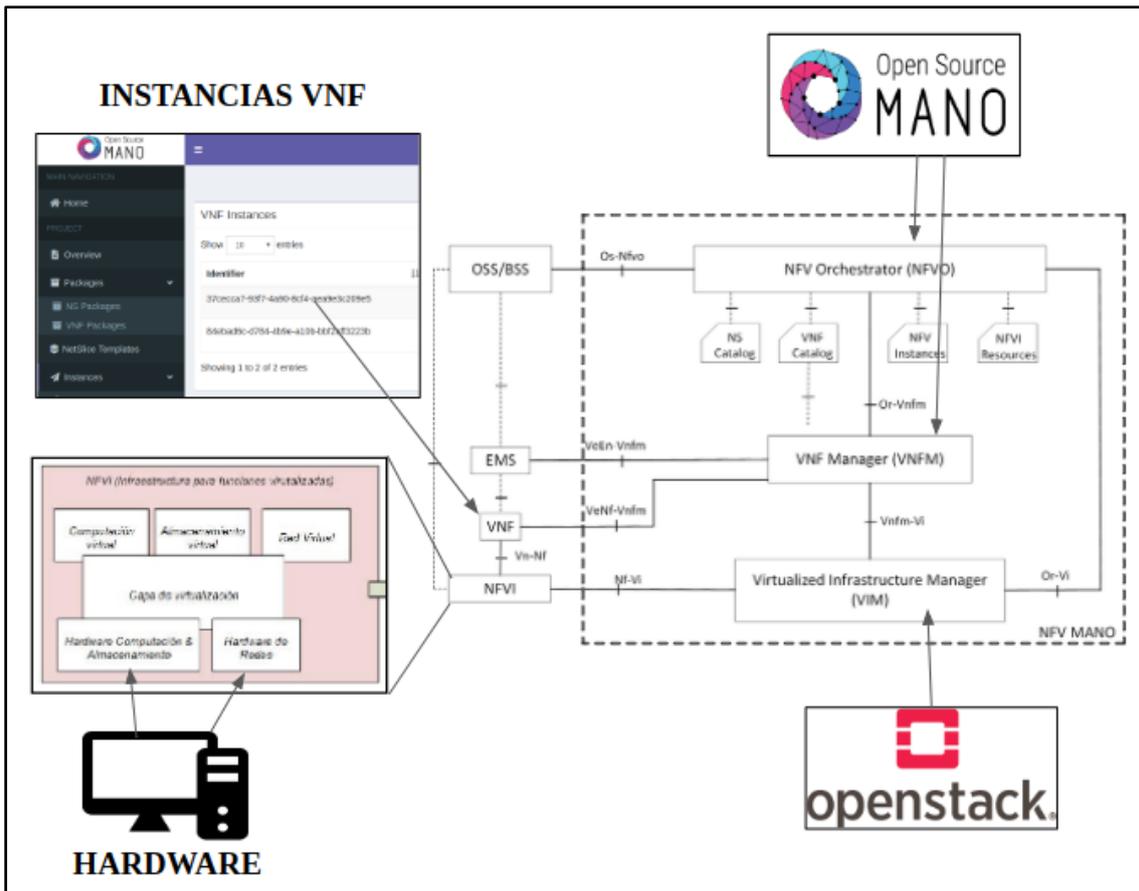


Ilustración 7.2 - Componentes de la implementación NFV

Aquí se puede ver, qué lugar ocupan en la implementación de un escenario NFV cada uno de los componentes de los que estamos haciendo uso.

OpenStack es el gestor de infraestructuras virtuales o VIM. Este se encarga del bloque NFVI de la arquitectura. Concretamente se encarga de virtualizar el hardware y ofrecérselo a al bloque superior en forma de recursos virtualizados para ello emplea el su componente o servicio Ironic que es el que se encarga de agregar máquinas físicas.

Y después OSM ocupa los otros dos bloques de la arquitectura MANO. Por un lado, la función del VNF Manager la realiza el componente VCA (VNF Configuration and Abstraction), ya mencionado previamente; este se encarga de la gestión de las diferentes VNFs. Y, por otro lado, las funcionalidades del NFV Orchestrator, las realiza el componente RO (Resource Orchestrator) que se encarga del aprovisionamiento de recursos que son ofrecidos por OpenStack en este caso, y también el componente SO (Service Orchestrator) que se encarga de funcionalidades de más alto nivel.

7.2. Especificaciones técnicas del proyecto

Para conseguir implementar el sistema que se propone es necesario contar con un equipo que cumpla ciertos requerimientos necesarios para llevar a cabo el proyecto. Por un lado, están los requerimientos de software y por otro lado los de hardware.

Los requerimientos de Software serán los siguientes:

- Por un lado, se van a necesitar 2 máquinas virtuales o reales para ejecutar OSM por una parte y Openstack por otra. Esto es debido a que para el funcionamiento de Openstack se recomienda que la máquina sea para uso exclusivo de Openstack.
- Después necesitaremos de algún software de virtualización que nos permita trabajar con varias máquinas en un mismo equipo, ya que es el objetivo del proyecto.
- Y por último necesitaremos el software como tal de OSM y Openstack, junto con todas las dependencias que tengan estos programas, como es el caso de Docker con OSM.

Y en cuanto a los requerimientos de Hardware.

Por un lado, está OSM que en la página oficial los requerimientos mínimos que se mencionan para el correcto funcionamiento son los listados a continuación:

- 2 CPUs.
- 4GB de RAM.
- 20GB de espacio en disco.
- Y una interfaz con acceso a internet.

Y por otro lado está Openstack cuyos requerimientos varían en función de la distribución de Openstack que se decida escoger. Pero los mínimos encontrados son empleando DevStack que son los mostrados a continuación:

- 1 CPU.
- 1GB de RAM.
- 10GB de espacio en disco.

7.3. Instalación OpenStack

Para la instalación de OpenStack, como se mencionaba en el análisis de alternativas, había varias opciones, pero finalmente se decidió emplear DevStack.

Para realizar la instalación de DevStack hay que instalarlo en un sistema operativo Ubuntu, o LinuxMint. En este caso se realizó con Ubuntu 18.04.

Comenzando con el proceso de instalación, inicialmente es recomendable crear un usuario dedicado al uso de OpenStack, y asignarlo al grupo sudo. Para ello se facilitan los siguientes comandos:

```
sudo useradd -s /bin/bash -d /opt/stack -m stack  
echo "stack ALL=(ALL) NOPASSWD: ALL" | sudo tee /etc/sudoers.d/stack
```

Después hay que descargar los scripts de devstack, y para ello se facilita un repositorio git. Para descargar los scripts ejecutar lo siguiente:

```
sudo su - stack  
sudo apt -y install git  
git clone https://git.openstack.org/openstack-dev/devstack
```

Después hay que crear un archivo dentro de la carpeta devstack que se ha descargado mediante git. Este archivo permitirá configurar el proceso de instalación automáticamente desde el inicio. Este debe tener el nombre "local.conf" y hay que tener cuidado en no crearlo con permisos de administrados. Ejecutar:

```
cd devstack  
nano local.conf
```

Después hay que incluir la configuración que se va a realizar en la instalación. Para ello agregar el siguiente texto, y cambiar tanto el ADMIN_PASSWORD como el HOST_IP. Este último ha de ser la IP de la interfaz a través de la cual se va a acceder a OpenStack.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

```
[[local|localrc]]
```

```
ADMIN_PASSWORD=<password>  
DATABASE_PASSWORD=$ADMIN_PASSWORD  
RABBIT_PASSWORD=$ADMIN_PASSWORD  
SERVICE_PASSWORD=$ADMIN_PASSWORD  
  
HOST_IP=<IP>
```

Después para iniciar el proceso de instalación ejecutar

```
./stack.sh
```

Si todo va bien al final se mostrará un resumen de la instalación.

Una vez se finaliza el proceso de instalación OpenStack ya queda completamente accesible y funcional. Se puede acceder a la UI a través del link “http://<ip_machine>/dashboard”.

Las credenciales que se crean por defecto son:

```
username: admin / demo  
password: la configurada en el archivo local.conf
```

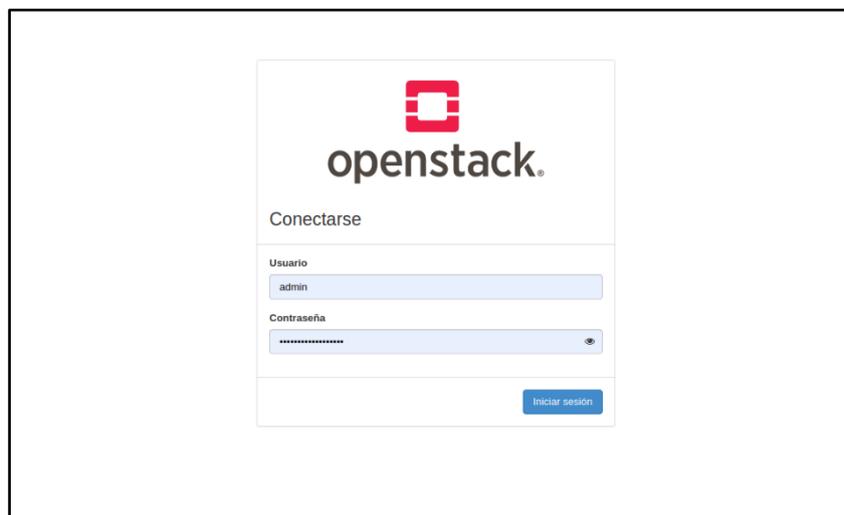


Ilustración 7.3 - OpenStack UI

7.4. Instalación directa de OSM

De manera muy similar se realizó el proceso de instalación de OSM, en este caso realizando una instalación directa sobre la máquina.

Para ejecutar esta instalación es recomendable hacer uso de una maquina Ubuntu 16.04. El proceso solo necesita de los siguiente 3 comandos:

```
wget https://osm-download.etsi.org/ftp/osm-5.0-five/install_osm.sh
chmod +x install_osm.sh
./install_osm.sh -t v5.0.5
```

Una vez instalado queda completamente funcional y accesible como OpenStack. Para acceder a la UI, que se muestra en la siguiente ilustración, se realiza a través del link “http://<ip_machine>”.

Las credenciales por defecto son:

```
username: admin
password: admin
```

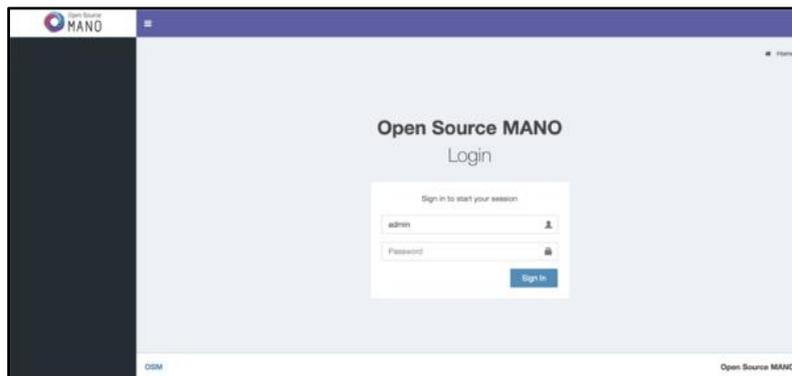


Ilustración 7.4 - OSM UI

7.5. Integración OSM y OpenStack

Por último, es necesario realizar una integración de OSM y OpenStack. Para ello es necesario realizar una configuración previa de OpenStack para funcionar con OSM, y por otro lado hay que agregar de manera correcta OpenStack en la sección de VIMs de OSM.

7.5.1. CONFIGURACIÓN OPENSTACK

En OpenStack hay que crear una red de administración sin la cual OSM no podría integrarse. Esto se realiza fácilmente mediante comandos, pero primero es necesario autenticarse dentro de la consola con la que se va a realizar la configuración.

Para ello primero hay que ir a la UI de OpenStack y loguearse con las credenciales a través de la misma. Una vez realizado esto es necesario descargar un fichero de autenticación del usuario logueado, disponible en un menú en la parte superior derecha como se muestra en la ilustración.

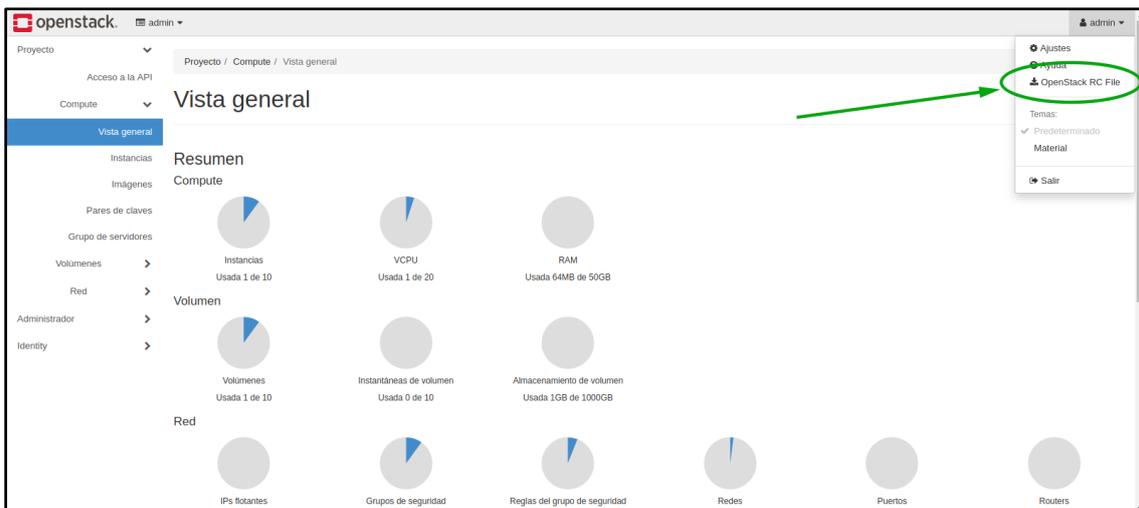


Ilustración 7.5 - Archivo de autenticación

Una vez realizada la descarga hay que abrir una consola y ejecutar el comando

```
source ficheroDescargado.sh
```

Una vez realizado esto se solicitará la contraseña del usuario correspondiente al archivo. Y una vez introducida, la consola ya estaría autorizada para realizar configuraciones.

Para crear la red de administración primero hay que modificar el archivo “/etc/neutron/plugins/ml2/ml2_conf.ini”, y añadir una vlan sobre la que trabajar que se llamara “physnow_em1”.

```
# tunnel IDs that are available for tenant network a
#tunnel_id_ranges =

[m2_type_vlan]
network_vlan_ranges = public,physnet_em1

## From neutron.ml2
#
```

Ilustración 7.6 - /etc/neutron/plugins/ml2/ml2_conf.ini configuration

Una vez realizado esto hay que emplear el siguiente comando para crear la red.

```
neutron net-create mgmt --provider:network_type=vlan --
provider:physical_network=physnet_em1 --provider:segmentation_id=500 --
shared
```

Una vez creada la red hay que asignarle unos dominios de res IP a través del siguiente comando.

```
neutron subnet-create --name subnet-mgmt mgmt 10.208.0.0/24 --allocation-
pool start=10.208.0.2,end=10.208.0.254
```

Después hay que crear un par de claves de seguridad en OpenStack esto se puede realizar de manera sencilla en la UI de OpenStack. Entrar en la “Proyecto -> Par de claves -> Crear Par de Claves” e indicar los parámetros como se muestran en la captura. Guardando el nombre asignado a ese keypar para utilizarlo más adelante.

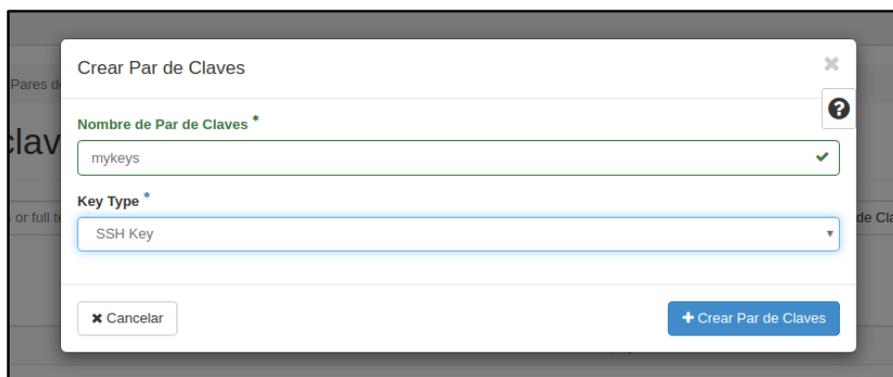


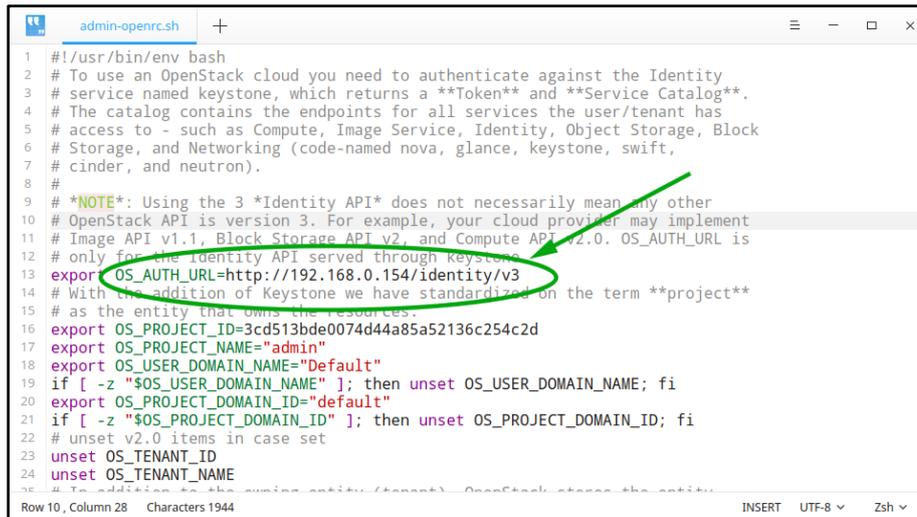
Ilustración 7.7 - OpenStack KeyPars

Con esto OpenStack ya quedaría configurado.

7.5.2. AGREGAR VIM OPENSTACK EN OSM

Ahora hay que agregar el VIM en OSM. Para ello se necesitan por un lado unas credenciales de OpenStack para proporcionárselas a OSM y por otro lado el link de autenticación de OpenStack.

Para obtener este link se tiene que abrir el fichero descargado en el apartado anterior y obtener el link de OpenStack como se muestra en la siguiente Ilustración.



```
1 #!/usr/bin/env bash
2 # To use an OpenStack cloud you need to authenticate against the Identity
3 # service named keystone, which returns a **Token** and **Service Catalog**.
4 # The catalog contains the endpoints for all services the user/tenant has
5 # access to - such as Compute, Image Service, Identity, Object Storage, Block
6 # Storage, and Networking (code-named nova, glance, keystone, swift,
7 # cinder, and neutron).
8 #
9 # *NOTE*: Using the 3 *Identity API* does not necessarily mean any other
10 # OpenStack API is version 3. For example, your cloud provider may implement
11 # Image API v1.1, Block Storage API v2, and Compute API v2.0. OS_AUTH_URL is
12 # only for the Identity API served through keystone.
13 export OS_AUTH_URL=http://192.168.0.154/identity/v3
14 # With the addition of Keystone we have standardized on the term **project**
15 # as the entity that owns the resources.
16 export OS_PROJECT_ID=3cd513bde0074d44a85a52136c254c2d
17 export OS_PROJECT_NAME="admin"
18 export OS_USER_DOMAIN_NAME="Default"
19 if [ -z "$OS_USER_DOMAIN_NAME" ]; then unset OS_USER_DOMAIN_NAME; fi
20 export OS_PROJECT_DOMAIN_ID="default"
21 if [ -z "$OS_PROJECT_DOMAIN_ID" ]; then unset OS_PROJECT_DOMAIN_ID; fi
22 # unset v2.0 items in case set
23 unset OS_TENANT_ID
24 unset OS_TENANT_NAME
25 # In addition to the unique entity (tenant), OpenStack stores the entity
```

Ilustración 7.8 – Link de autenticación OpenStack

Una vez se tiene el link en la maquina donde está instalado OSM ejecutar el siguiente comando, rellenando cada campo.

```
osm vim-create --name openstack-site --user USUARIO --password
PASSWORD --auth_url LINK --tenant PROYECTO --account_type
openstack --config='{security_groups: default, keypair: KEYPAR}'
```

Y con esto OpenStack quedaría correctamente agregado. Y ya se podrían lanzar servicios de red en OSM.

Una vez realizado todo, si se ha hecho de manera correcta ya se tendría la infraestructura montada entera y operativa para trabajar.

METODOLOGÍA

En este apartado se incluyen por un lado la descripción de los resultados obtenidos, y por otro lado se incluirá la planificación del proyecto con la definición de tareas y entregables, y con el diagrama de Gantt.

1. DESCRIPCIÓN DE LOS RESULTADOS

Para validar el correcto funcionamiento de la solución que se ha propuesto se van a realizar pruebas para comprobar su funcionalidad.

1.1. Funcionamiento OpenStack

Para comprobar el funcionamiento de OpenStack se han realizado 2 pruebas que son las siguientes:

- Lanzamiento de una instancia cirros y acceder a ella.
- Establecer comunicación entre dos máquinas conectadas en una red OpenStack.

1.1.1. LANZAMIENTO DE INSTANCIA

La instancia que se va a lanzar va a ser una instancia cirros muy liviana de apenas unos pocos MB, concretamente “cirros 0.4.0”.

Primero se ha agregado la imagen de la instancia fácilmente desde la UI. Después se creó una red para poder trabajar, y sobre ella se procedió a lanzar la instancia a través de una máquina con especificaciones mínimas, pues la imagen es muy ligera. La instancia se ejecutó de manera correcta, como se muestra en la siguiente Ilustración, junto con las especificaciones de la MV.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

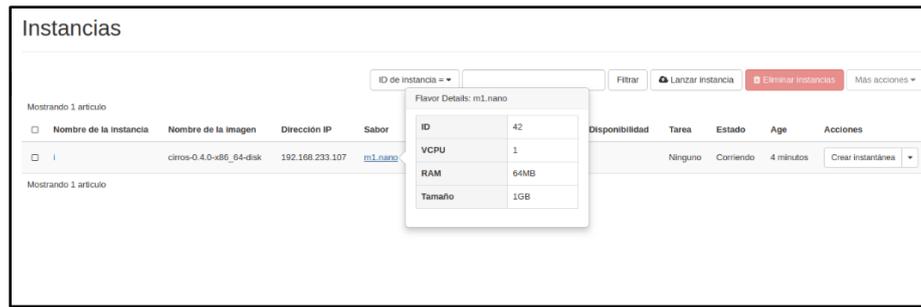


Ilustración 1.1 - Instancia lanzada

Posteriormente se accedió a la maquina también de manera exitosa como se aprecia a continuación.

```
pwd
/home/cirros
$ ifconfig
eth0 Link encap:Ethernet HWaddr FA:16:3E:4A:DD:8E
      inet addr:192.168.233.133 Bcast:192.168.233.255 Mask:255.255.255.0
      inet6 addr: fe80::f816:3eff:fe4a:dd8e/64 Scope:Link
      UP BROADCAST RUNNING MULTICAST MTU:1450 Metric:1
      RX packets:172 errors:0 dropped:0 overruns:0 frame:0
      TX packets:108 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:17147 (16.7 KiB) TX bytes:9749 (9.5 KiB)

lo Link encap:Local Loopback
   inet addr:127.0.0.1 Mask:255.0.0.0
   inet6 addr: ::1/128 Scope:Host
   UP LOOPBACK RUNNING MTU:65536 Metric:1
   RX packets:0 errors:0 dropped:0 overruns:0 frame:0
   TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
   collisions:0 txqueuelen:1
   RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

$
```

Ilustración 1.2 - Instancia 1 con IP 192.168.233.133

1.1.2. ESTABLECER COMUNICACION ENTRE LAS INSTANCIAS

Para realizar un ping entre 2 instancias, se ejecutó una instancia similar a la vista en el proceso anterior, y sobre la misma red como se aprecia en las siguientes ilustraciones.

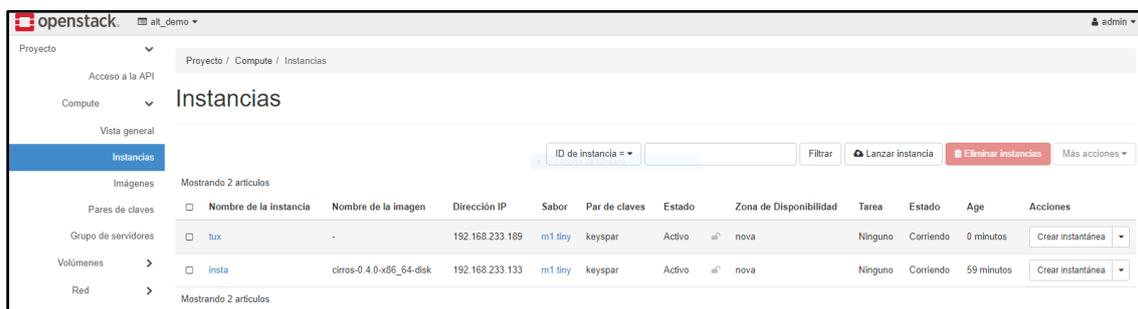


Ilustración 1.3 - 2 instancias en ejecución

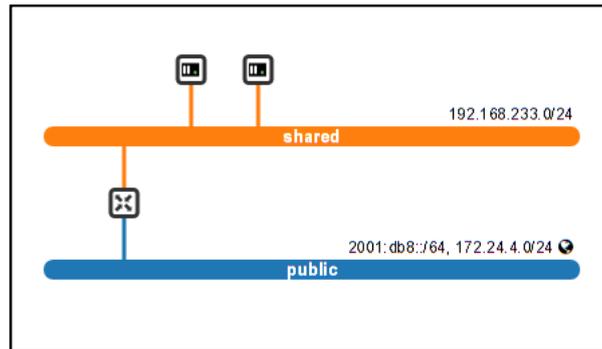


Ilustración 1.4 - Topología con 2 instancias

Después se procedió a acceder a ambas instancias igual que en el apartado anterior y se procedió a realizar un ping como se muestra en las siguientes ilustraciones. Y el ping funciono de manera correcta.

```
inet6 addr: ::1/128 Scope:Host
UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

$ ping 192.168.233.133
PING 192.168.233.133 (192.168.233.133): 56 data bytes
64 bytes from 192.168.233.133: seq=0 ttl=64 time=2.147 ms
64 bytes from 192.168.233.133: seq=1 ttl=64 time=0.635 ms
64 bytes from 192.168.233.133: seq=2 ttl=64 time=0.551 ms
64 bytes from 192.168.233.133: seq=3 ttl=64 time=0.418 ms
64 bytes from 192.168.233.133: seq=4 ttl=64 time=0.509 ms
64 bytes from 192.168.233.133: seq=5 ttl=64 time=0.495 ms
64 bytes from 192.168.233.133: seq=6 ttl=64 time=0.566 ms
64 bytes from 192.168.233.133: seq=7 ttl=64 time=0.495 ms
^C
--- 192.168.233.133 ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max = 0.418/0.727/2.147 ms
$ -
```

Ilustración 1.5 - Instancia 2 realizando ping a 192.168.233.133

1.2. Funcionamiento de OSM sobre OpenStack

Para comprobar el correcto funcionamiento de la plataforma completa, con OSM trabajando sobre OpenStack, se va a lanzar un servicio de red (NS).

Este servicio de red se proporciona en la página oficial de OSM y consta de implementar la red que se muestra en la siguiente ilustración. Que consta de conectar dos máquinas cirros a través de un virtual link (VL).

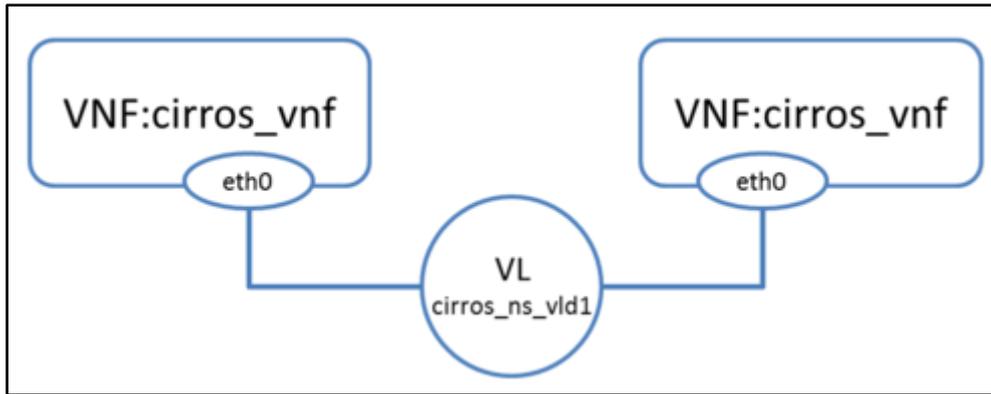


Ilustración 1.6 - Network Service: Dos VNFs Cirros

Una vez agregados los paquetes VNF y NS que son necesarios para ejecutar este servicio de red, y de agregar la imagen correspondiente en OpenStack. Se procedió a lanzar el servicio de red como tal. Este como se aprecia en la siguiente ilustración se ejecutó de manera correcta.

Name	Identifier	Nsd name	Operational Status	Config Status	Detailed Status	Actions
new_ns	02a0d664-b805-4923-b686-183f0c6ca7ce	cirros_2vnf_ns	running	configured	done	i 👤 🗑️ Actions

Ilustración 1.7 - NS running and configured

En la siguiente Ilustración puede apreciarse la topología del NS.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

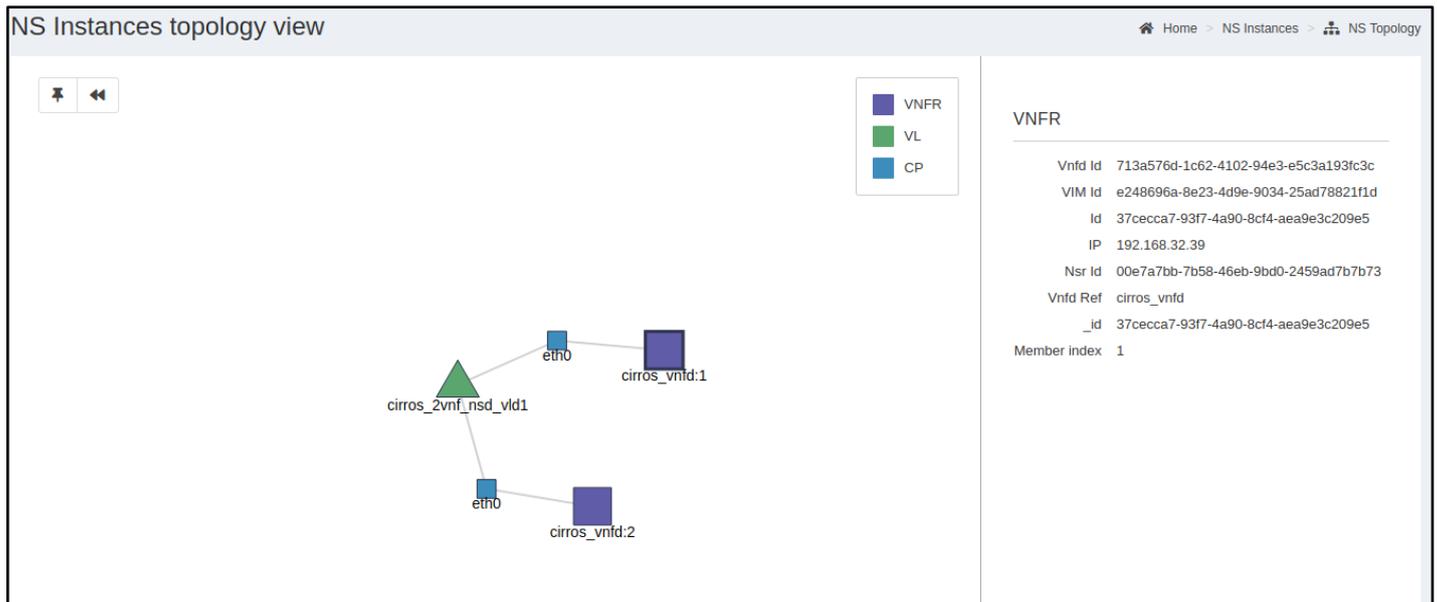


Ilustración 1.8 - Topología NS OSM

Y para comprobar el correcto funcionamiento se han visualizado las instancias y la topología creada en OpenStack. En la ilustración siguiente se muestra la topología de OpenStack con las dos instancias creadas.

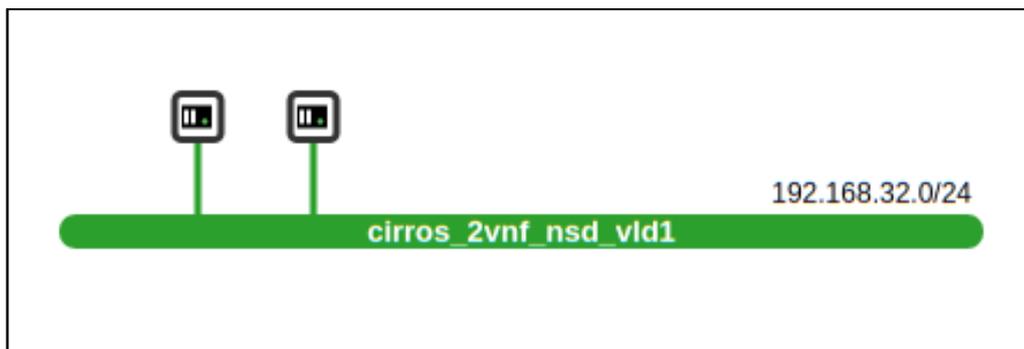


Ilustración 1.9 - Topología NS OpenStack

A continuación, se comprobó que hubiese una correcta comunicación entre ambas instancias, accediendo a ambas y realizando un ping entre ambas.

```

$ sudo
/home/cirros
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:17:F4:EB
          inet addr:192.168.32.39  Bcast:192.168.32.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe17:f4e8/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:58 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5744 (5.6 KiB)  TX bytes:2219 (2.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$
    
```

Ilustración 1.10 - Instancia 1 IP 192.168.32.39

```

$ sudo
/home/cirros
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:2B:15:DF
          inet addr:192.168.32.254  Bcast:192.168.32.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fe2b:15df/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1450  Metric:1
          RX packets:59 errors:0 dropped:0 overruns:0 frame:0
          TX packets:19 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5032 (5.6 KiB)  TX bytes:2219 (2.1 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$
    
```

Ilustración 1.11 - Instancia 2 IP 192.168.32.254

Y se realizó un ping de manera exitosa.

```

UP LOOPBACK RUNNING MTU:16436 Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

$ ping 192.168.32.254
PING 192.168.32.254 (192.168.32.254): 56 data bytes
64 bytes from 192.168.32.254: seq=0 ttl=64 time=1.924 ms
64 bytes from 192.168.32.254: seq=1 ttl=64 time=0.713 ms
64 bytes from 192.168.32.254: seq=2 ttl=64 time=0.492 ms
64 bytes from 192.168.32.254: seq=3 ttl=64 time=0.459 ms
64 bytes from 192.168.32.254: seq=4 ttl=64 time=0.527 ms
64 bytes from 192.168.32.254: seq=5 ttl=64 time=0.519 ms
64 bytes from 192.168.32.254: seq=6 ttl=64 time=0.508 ms
64 bytes from 192.168.32.254: seq=7 ttl=64 time=0.516 ms
64 bytes from 192.168.32.254: seq=8 ttl=64 time=0.487 ms
64 bytes from 192.168.32.254: seq=9 ttl=64 time=0.502 ms
64 bytes from 192.168.32.254: seq=10 ttl=64 time=0.509 ms
64 bytes from 192.168.32.254: seq=11 ttl=64 time=0.504 ms

--- 192.168.32.254 ping statistics ---
12 packets transmitted, 12 packets received, 0% packet loss
round-trip min/avg/max = 0.459/0.638/1.924 ms
$
    
```

Ilustración 1.12 - Ping de instancia 1 a instancia 2

Con lo que se podía asegurar que el escenario que se pretendía implementar funcionaba de manera correcta.

2. DESCRIPCIÓN DE TAREAS Y ENTREGABLES

En este apartado va a recoger las diferentes tareas que se han ido elaborando a lo largo del proyecto. A continuación, se enumeran dichas tareas agrupadas según su funcionalidad:

1. **Definición del proyecto:** en esta tarea se ha definido las especificaciones del proyecto y se han marcado los objetivos claros que pretenden cumplirse en el desarrollo del mismo. Esta tarea se compone de dos subtareas:
 - 1.1. Definición de las especificaciones: se especificarán los requisitos mínimos que ha de cumplir la solución propuesta.
 - 1.2. Definición de los objetivos: Se planificará el proyecto con unos objetivos concretos que se cumplirán preferentemente de manera secuencial.
2. **Análisis de alternativas y estudio del estado del arte:** tarea en la que se han analizado y escogido de entre las diferentes opciones del mercado, y se ha hecho un análisis del estado del arte de las tecnologías con las que se va a trabajar. Se compone de dos subtareas:
 - 2.1. Análisis de alternativas: Se hará un análisis de las distintas opciones que podrían servir como solución al escenario que se plantea como objetivo, y se escogerá entre ella entre en base a unos criterios concretos.
 - 2.2. Estudio del estado del arte: Se realizará un estudio del estado del arte de las tecnologías que se van a emplear.
3. **Diseño de una solución:** donde, con las tecnologías escogidas previamente, se diseñó la solución y posteriormente se realizó una implementación de la misma.
4. **Implementación de la solución:** Tarea en la cual se procederá a implementar la solución que se ha diseñado. Esta tarea se compone de las siguientes 2 subtareas:
 - 4.1.1. Implementación de los módulos independientes: Se implementarán los dos módulos necesarios de manera independiente.
 - 4.1.2. Integración de los diferentes módulos: Se acoplarán ambos módulos para que trabajen de manera conjunta.
5. **Validación de la solución:** tarea en la que se pretende validar que la solución implementada cumpla con las especificaciones realizadas previamente. Para ello se realizaron unas pruebas funcionales. Por ello se compone de la subtaska:
 - 5.1. Diseño de las pruebas funcionales: Se diseñarán unas pruebas para realizar sobre la implementación para la validación de la misma.

- 5.2. Pruebas funcionales: Se procederá a realizar las pruebas diseñadas.
6. **Gestión del proyecto:** Tarea para llevar un control del proyecto, y la vez realizar la documentación del mismo. Se paralelamente al resto de tareas definidas en el proyecto. Se compone de:
- 6.1. Seguimiento del proyecto: Se realizará un seguimiento del proyecto durante toda la realización del mismo.
- 6.2. Documentación del proyecto: Se ira realizando la documentación del proyecto mientras se van cumpliendo las diferentes tareas.

Estas tareas se acompañan de una serie de entregables cuyo objetivo es asegurar el cumplimiento de las diferentes tareas. Estos entregables son los siguientes:

1. Entregable de las especificaciones técnicas. El cual se realizará cuando se haya terminado la tarea 3 de “Definición de las especificaciones”.
2. Entregable del análisis de alternativas. Contendrá un análisis de las diferentes alternativas que se hayan analizado, y la elección final en base a unos criterios que también hay que documentar en el entregable. Se elaborará al acabar la tarea 6 “Análisis de alternativas”.
3. Entregable del diseño. Este recogerá el diseño que se propone, y la implementación de la misma describiendo en detalle cómo se ha llevado a cabo. Se realizará una vez se haya terminado la tarea 8 de “Diseño e implementación de los módulos”.
4. Documento con el diseño del plan de pruebas funcionales. Recogerá las pruebas a realizar. Y se elaborará al acabar la tarea 14 “Diseño de pruebas funcionales”.
5. Documento con el resultado de las pruebas. Contendrá los resultados de las pruebas del documento realizado previamente. Se realizará al acabar la tarea 15 “Pruebas funcionales”.
6. Entregable del documento final. Se realiza al acabar la última tarea. Contendrá diferentes apartados correspondientes a las diferentes tareas que se han realizado durante todo el proyecto.

3. DIAGRAMA DE GANTT / CRONOGRAMA

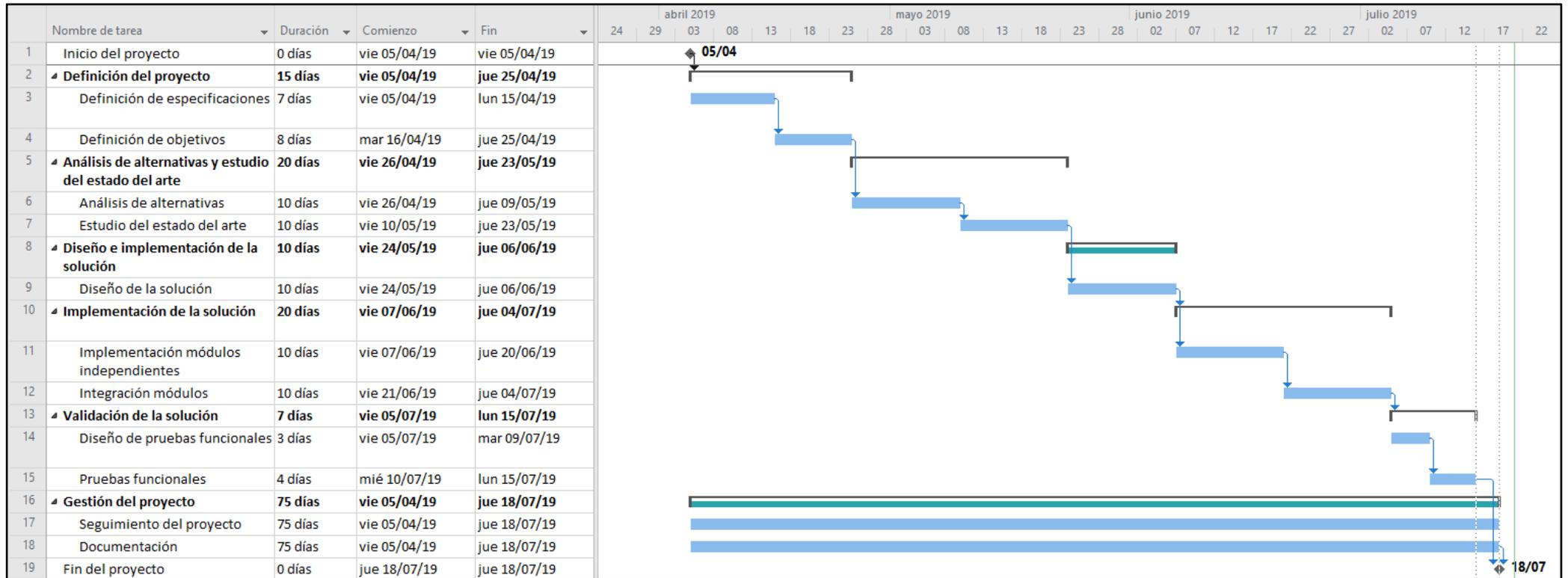


Ilustración 3.1 - Diagrama de Gantt

ASPECTOS ECONÓMICOS

En este apartado se van a recoger todos los costes que ha conllevado la realización de este proyecto durante sus 75 días de duración.

1. HORAS INTERNAS

Las horas internas incluyen las horas invertidas por los participantes en el proyecto. Que en el caso de este proyecto son 2 ingenieros, una ingeniera Senior y un ingeniero Junior.

Para calcular el costo de los ingenieros es necesario contabilizar las horas invertidas en cada apartado del proyecto. Estas horas se recogen en la siguiente tabla.

Paquete de trabajo	Responsable	Nº Horas
Definición del proyecto	Ingeniero Junior	25
	Ingeniera Senior	7
Análisis de alternativas y estudio del estado del arte	Ingeniero Junior	40
	Ingeniera Senior	4
Diseño de la solución	Ingeniero Junior	50
	Ingeniera Senior	3
Implementación de la solución	Ingeniero Junior	75
	Ingeniera Senior	7
Validación de la solución	Ingeniero Junior	25
	Ingeniera Senior	4

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Gestión del proyecto	Ingeniero Junior	85
	Ingeniera Senior	25
Total	Ingeniero Junior	300
	Ingeniera Senior	50

Tabla 1.1 - Horas de los Ingenieros

El ingeniero Junior ha invertido un total de 250h y la ingeniera Senior un total de 50h. Los precios de cada uno junto con el coste total que suponen se recogen a continuación en la siguiente tabla.

Nombre	Rol	Tasa Horaria	Nº Horas	Coste Total
Jasone Astorga	Ingeniera Senior (Directora del proyecto)	50€ / h	50	2500 €
Daniel Ruiz	Ingeniero Junior	20€ / h	300	6000 €
Total				8500 €

Tabla 1.2 - Coste de las Horas Internas

2. GASTOS

A continuación, se recogen los gastos que ha supuesto el proyecto que no son reutilizables.

Concepto	Coste
Electricidad	15 €
Internet	50 €
Total	65 €

Tabla 2.1 - Coste de los Gastos

3. AMORTIZACIONES

En cuanto a los gastos d material reutilizable se tienen los siguientes

Concepto	Valor Inicial	Valor Residual	Vida Util	Tiempo de uso	Coste
Ordenador personal	600 €	100	10 años	3 meses	12,5
Total					12,5

Tabla 3.1 - Coste de las Amortizaciones

4. COSTE TOTAL

El coste total se recogerá en la siguiente tabla, como suma de todos los anteriores, más los costes indirectos.

Concepto	Coste
Horas Internas	8500 €
Gastos	65 €
Amortizaciones	12,5 €
Subtotal	7584,5 €
Costes Indirectos (10%)	857,75€
Total	9435,25 €

Tabla 4.1 - Coste Total del Proyecto

Como se aprecia en la tabla el coste total del proyecto resulta en unos 9435€.

CONCLUSIONES

Este trabajo ha permitido conocer cómo es el estado del arte de las tecnologías como el Cloud Computing o las NFV. Aprendiendo qué nivel de madurez tienen a día de hoy, y especialmente el futuro de las mismas, pues parece muy claro el su futuro debido a las características que ofrecen. También se puede deducir su importancia observando las grandes comunidades apoyando estos proyectos; y no sólo comunidades, sino que también algunas de las empresas más grandes que existen, sobre todo proveedoras de hardware o empresas de telecomunicaciones. Algunas operadoras de telecomunicaciones incluso han empezado a desplegar ya tecnologías NFV para poder proporcionar servicios 5G, como es el caso de Telefónica o British Telecom.

Las tecnologías de virtualización son una clara tendencia de cara al futuro pues garantizan una enorme flexibilidad y con ella mayor eficiencia y menor costo. Es la base de las tecnologías que se han analizado, por un lado, posibilita en gran medida la computación en la nube debido a que permite ofrecer los recursos justos que se demanden; y por otro lado sin ella no se podría virtualizar función de red alguna. Parece que en un futuro todo se encontrará virtualizado, existiendo la menor dependencia posible entre las capas de software y hardware, estando incluso en la nube todos los datos y aplicaciones personales. Esto posibilita escenarios especialmente interesantes como entornos en los que los equipos personales estén orientados exclusivamente a ofrecer una interfaz para el usuario. Permitiendo dispositivos muy livianos y cómodos, y a su vez contando con una enorme potencia, pues sus recursos de computación se encontrarían en la nube.

Con este proyecto se ha podido experimentar con estas tecnologías que trabajan con los conceptos de Cloud y NFV. Además, el principal requerimiento del proyecto era conseguir una plataforma NFV que pudiese ejecutarse en un ordenador personal, ya que son tecnologías novedosas con las que resulta especialmente interesante tener una plataforma accesible, sin necesidad de grandes servidores, para experimentar y aprender sobre las mismas. Es por ello que se han escogido las alternativas que se ha considerado las menos exigentes, y se puede decir que se ha logrado dicho propósito.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

También se ha aprendido mucho sobre el funcionamiento de las tecnologías, puesto que al inicio no se partía con una idea muy clara del funcionamiento de NFV especialmente, y se tenía muchas dudas sobre determinadas características de funcionamiento. Pero tras el estudio y la búsqueda de información en internet los conceptos han quedado mucho más claros.

Personalmente el objetivo era aprender sobre nuevas tecnologías, especialmente si estas parecen prometedoras en un futuro y se ha aprendido bastante sobre las mismas. Aunque también se ha aprendido que hay una gran infraestructura/comunidad detrás de estas tecnologías, con muchos proyectos y desarrolladores que trabajan para que estas tecnologías sean funcionales y que si se quisiese profundizar en su conocimiento aún hay muchísimo por conocer.

BIBLIOGRAFÍA

- [1] SDX Central “What is the Virtualized Infrastructure Manager”. URL: <https://www.sdxcentral.com/networking/nfv/definitions/virtualized-infrastructure-manager-vim-definition/>.
- [2] SDX Central “What is NFV MANO?”. URL: <https://www.sdxcentral.com/networking/nfv/mano-lso/definitions/nfv-mano/>.
- [3] OSM ETSI “OSM Release FIVE”. URL: https://osm.etsi.org/wikipub/index.php/OSM_Release_FIVE.
- [4] OSM ETSI “Open Source MANO WIKI”. URL: https://osm.etsi.org/wikipub/index.php/Main_Page.
- [5] Wikipedia “Network function virtualization”. URL: https://en.wikipedia.org/wiki/Network_function_virtualization.
- [6] Ciena “¿Qué es NFV?”. URL: https://www.ciena.com.mx/insights/what-is/What-is-Network-Functions-Virtualization_es_LA.html .
- [7] Wikipedia “Cloud computing”. URL: https://en.wikipedia.org/wiki/Cloud_computing .
- [8] Salesforce “¿Qué es Cloud Computing?”. URL: <https://www.salesforce.com/mx/cloud-computing/> .
- [9] Manuel Serrano “Distribuciones Openstack – Qué distribución Openstack elegir”. URL: <https://virtualizadesdezero.com/distribuciones-openstack/> .
- [10] Sagar Nangare “OpenStack and Open Source MANO: Technologies for NFV Deployment”. URL: <https://devops.com/openstack-and-open-source-mano-technologies-for-nfv-deployment/> .
- [11] Iricent “Management and Orchestration (MANO)”. URL: <https://iricent.com/solutions/mano/>.
- [12] Opensource “What is OpenStack?”. URL: <https://opensource.com/resources/what-is-openstack> .
- [13] Lee Doyle “What is NFV and what are its benefits”. URL: <https://www.networkworld.com/article/3253118/what-is-nfv-and-what-are-its-benefits.html> .
- [14] SDX Central “How 5G NFV Will Enable the 5G Future”. URL: <https://www.sdxcentral.com/5g/definitions/5g-nfv/>.

- [15] Egyptian Informatics Journal “Cloud based SDN and NFV architectures for IoT infrastructure”. URL: <https://www.sciencedirect.com/science/article/pii/S1110866517303523> .
- [16] Eduardo Ballesteros Martínez “SDN, NFV y los fundamentos de las redes del futuro”. URL: <https://empresas.blogthinkbig.com/sdn-nfv-y-los-fundamentos-de-las-redes-del-futuro/>.
- [17] Josphat Mutal “OpenStack Deployment on Ubuntu 18.04 with DevStack”. URL: <https://computingforgeeks.com/openstack-deployment-on-ubuntu-with-devstack/>.

ANEXO

Manual de usuario para crear Network Services en OSM

Para poder poner en marcha NS (Network Services) en OSM es necesario realizar varios pasos antes de instanciarlo.

Suponiendo que ya se tiene toda la plataforma configurada y en marcha con OpenStack agregado en la sección de VIMs, para lanzar la instancia hay que realizar 3 pasos previos.

Primero es necesario agregar a través de OpenStack las imágenes con las que se va a trabajar. Esto se puede realizar de manera sencilla a través de la UI de OpenStack en la sección de imágenes. Como se aprecia en la ilustración se puede agregar imágenes a través del botón que se resalta.

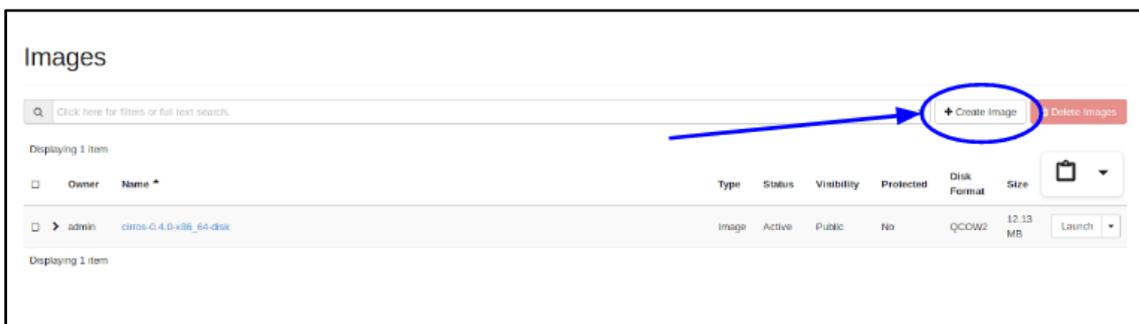


Ilustración 1 - Añadir imágenes a OpenStack

Dentro se piden diferentes parámetros de configuración de la imagen, como la ubicación, el formato, nombre, tamaño...etc.

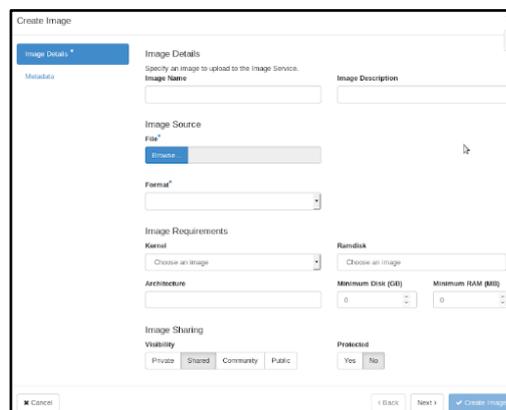


Ilustración 2 – Parámetros de configuración de las imágenes

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Esta imagen que se suba es recomendable que esté diseñada para lanzarse en la nube, por ejemplo, ubuntu ofrece una versión de su sistema operativo para lanzamiento en cloud.

En posteriores versiones de OSM se pretende que la gestión de imágenes también pueda realizarse a través de su UI, pero por ahora es necesario agregarlas manualmente en los diferentes VIM.

OSM ofrece en su página web gran variedad de paquetes tanto NSs, como VNFs junto con sus correspondientes imágenes.

Una vez se ha agregado hay que añadir también los diferentes VNF que se van a necesitar en el lanzamiento del NS. Para agregar los VNF se puede hacer de manera muy sencilla arrastrando los paquetes en la sección de VNF paquetes como se muestra en la ilustración.

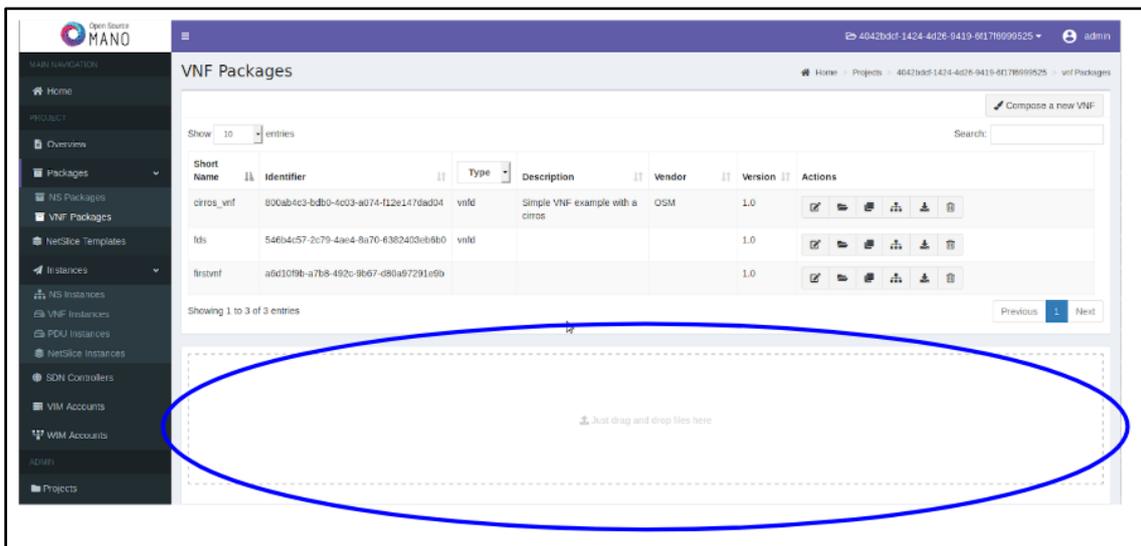


Ilustración 3 - Agregación de paquetes NFV

Además, OSM ofrece gestor gráfico para la creación de estos paquetes en la propia UI. A esta se puede acceder a través del botón “Compose a new VNF”. El creador tiene un formato como el que se muestra en la ilustración.

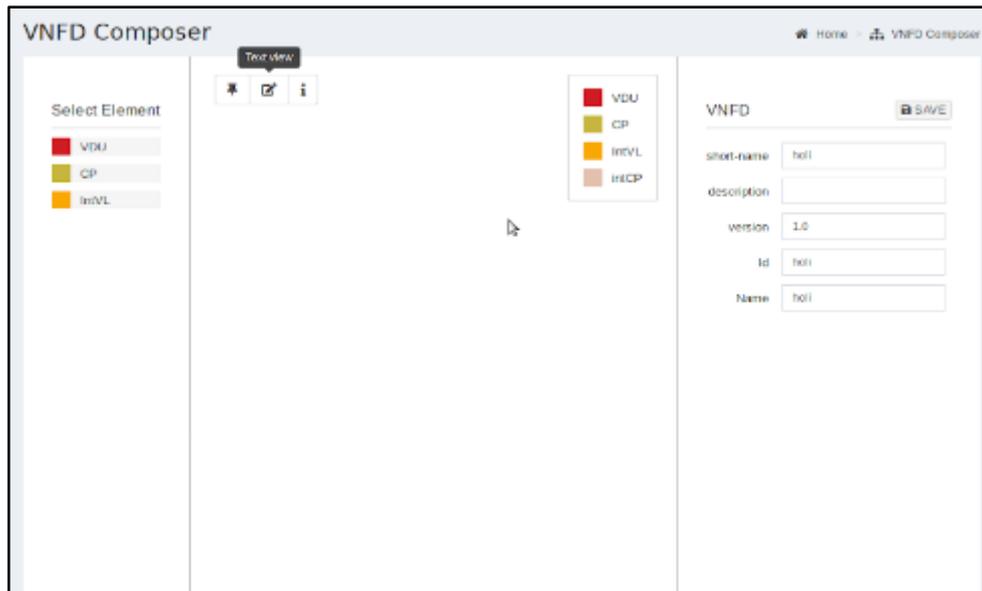


Ilustración 4 - VNF composer

Los elementos necesarios para la creación del paquete están en el panel de la izquierda. Estos se pueden arrastrar al centro y unir entre sí para crear el paquete como tal. Como por ejemplo en los VNF empleados para realizar la validación en este proyecto que tienen el siguiente formato.

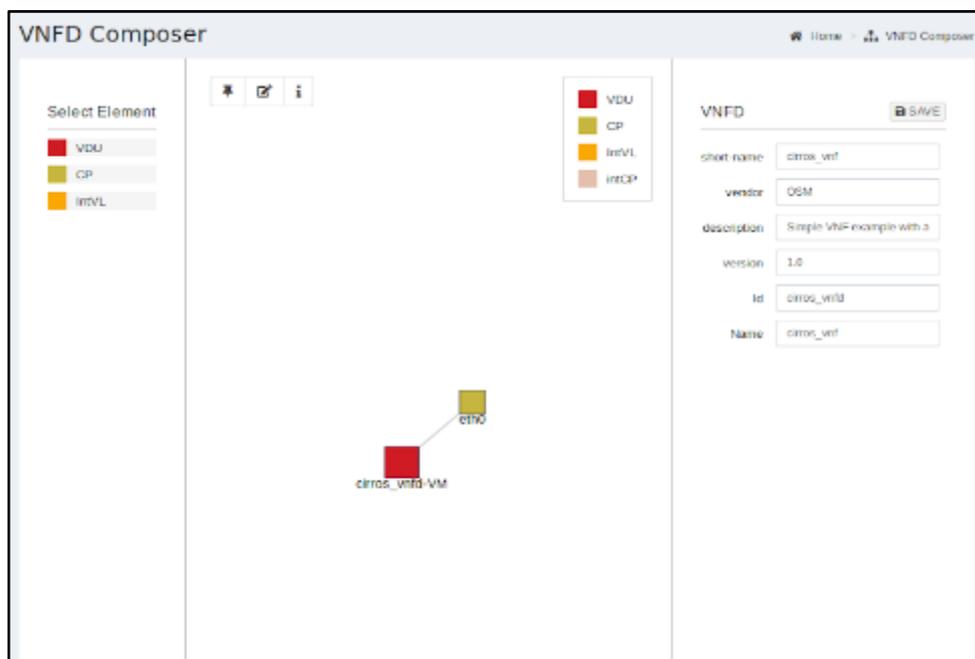


Ilustración 5 - Ejemplo NFV

Pero aunque el editor gráfico facilite mucho la elaboración de paquetes es necesario acceder también al modo texto en la parte superior para

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

configuraciones más detalladas. Por ejemplo, en el caso de las VNF mencionadas, se necesita indicar la imagen del VIM (OpenStack) que se desea emplear como se aprecia en la ilustración.

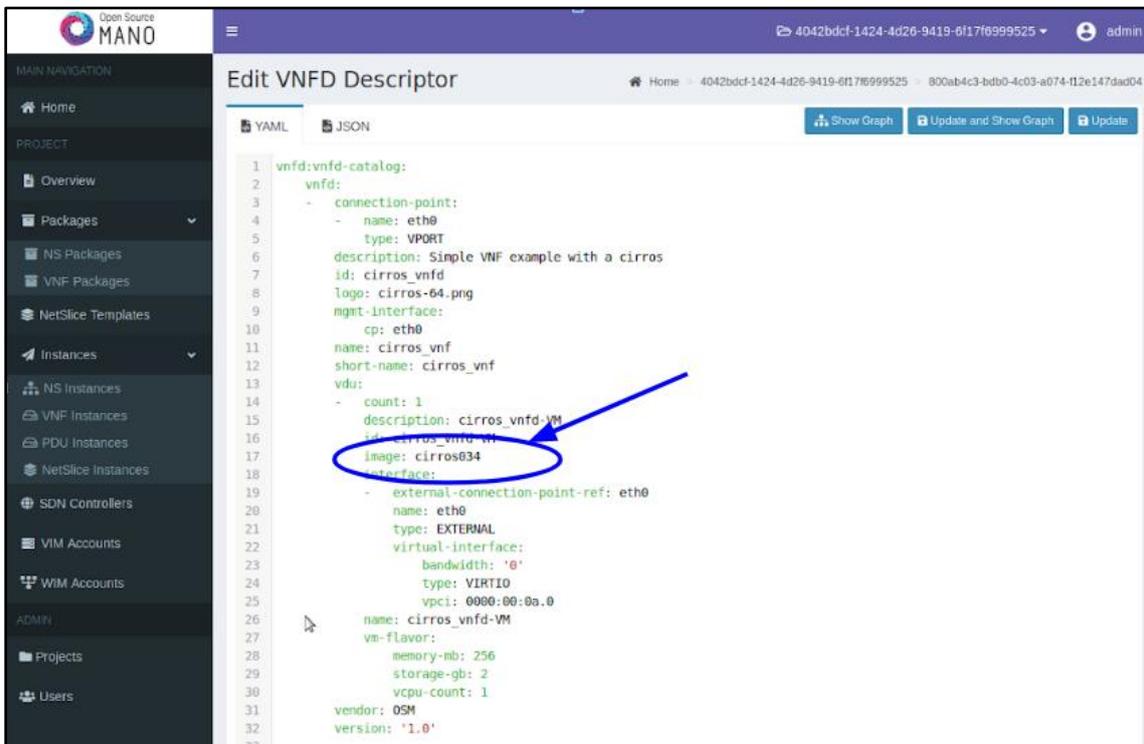


Ilustración 6 - Diseño VNF en modo texto

De manera muy similar se realiza la agregación de paquetes NS. Que se puede realizar arrastrando el paquete en la sección de paquetes NS. Que se muestra en la siguiente ilustración.

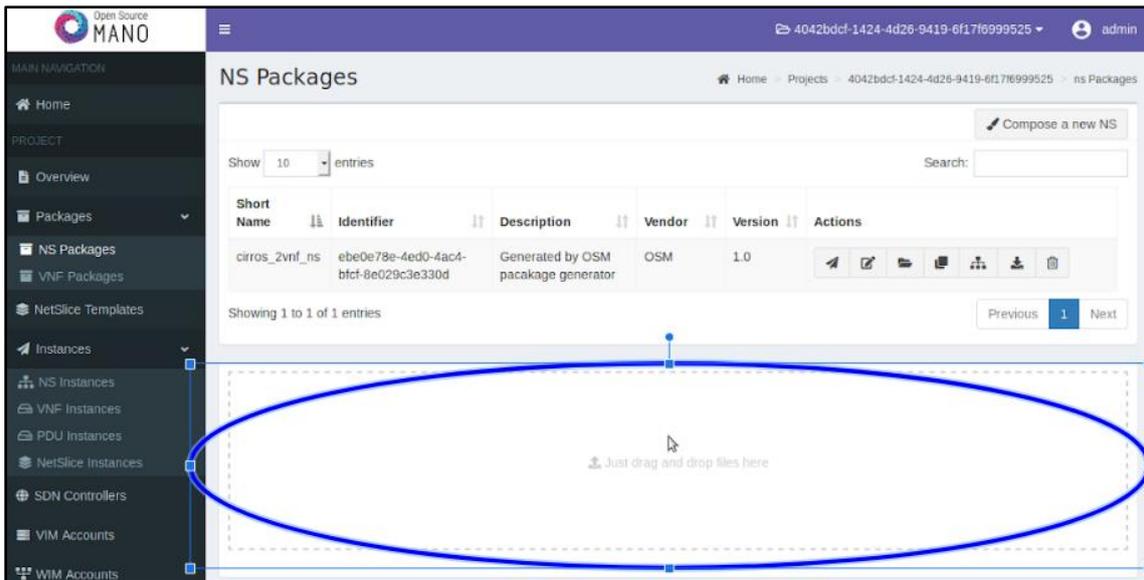


Ilustración 7 - Agregación paquetes NS

Adeams al igual que con VNF tambien proporciona un gestor grafico para la creacion de los mismos el cual es muy similar al de las VNF. Es el que se muestra en la siguiente Ilustracion.

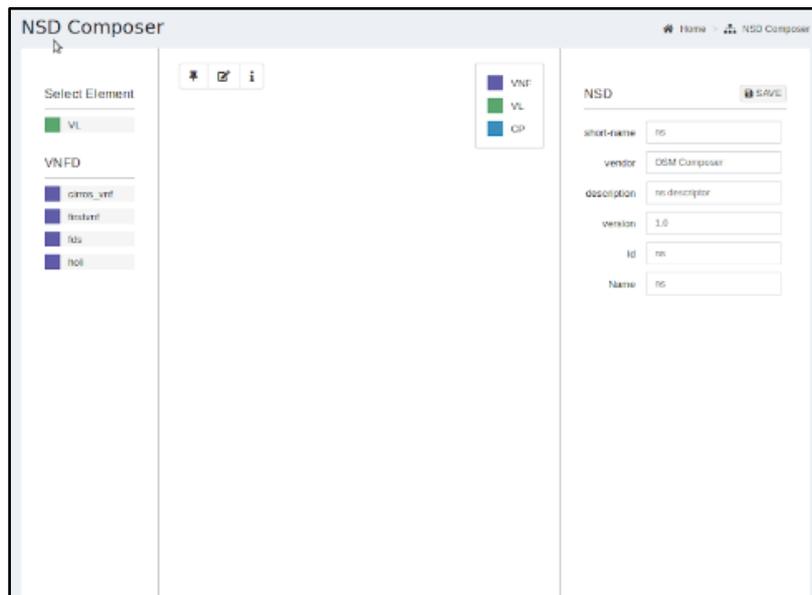


Ilustración 8 - NS composer

Por ver un ejemplo, el caso de NS lanzado en este proyecto consta de la siguiente topología.

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

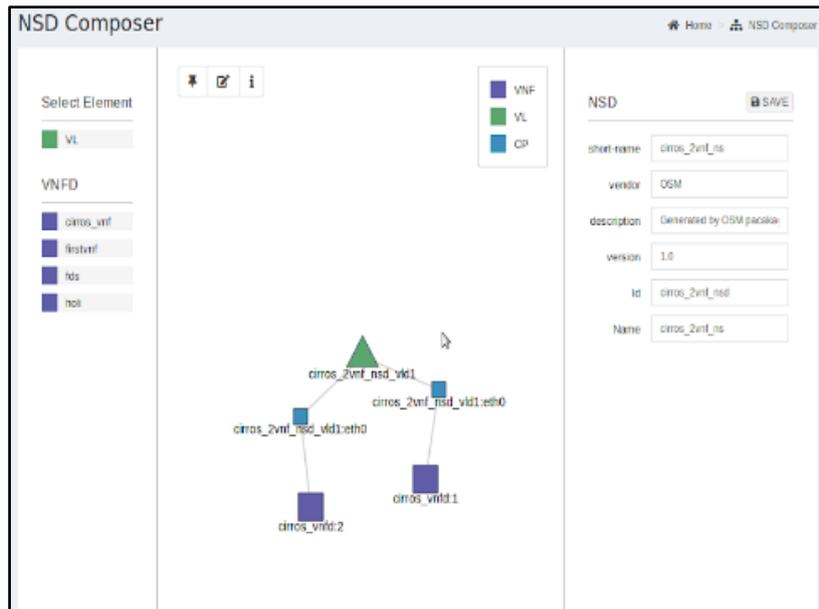


Ilustración 9 - Ejemplo NS

Y del siguiente código.

```
1 nsd:nsd-catalog:
2   nsd:
3     - constituent-vnfd:
4       - member-vnf-index: 1
5         vnf-id-ref: cirros_vnfd
6       - member-vnf-index: 2
7         vnf-id-ref: cirros_vnfd
8     description: Generated by OSM package generator
9     id: cirros_2vnf_nsd
10    logo: osm_2x.png
11    name: cirros_2vnf_ns
12    short-name: cirros_2vnf_ns
13    vendor: OSM
14    version: '1.0'
15    vld:
16      - id: cirros_2vnf_nsd_vld1
17        mgmt-network: 'true'
18        name: cirros_2vnf_nsd_vld1
19        short-name: cirros_2vnf_nsd_vld1
20        type: ELAN
21        vnf-connection-point-ref:
22          - member-vnf-index-ref: 1
23            vnf-connection-point-ref: eth0
24            vnf-id-ref: cirros_vnfd
25          - member-vnf-index-ref: 2
26            vnf-connection-point-ref: eth0
27            vnf-id-ref: cirros_vnfd
28
```

Ilustración 10 - Ejemplo en modo texto

DISEÑO E IMPLEMENTACIÓN DE PLATAFORMA PARA EXPERIMENTACIÓN EN NFV

Por último, solo quedaría lanzar la instancia NS. Para ello solo hay que localizar el paquete NS y lanzar la instancia a través del botón que se muestra en la ilustración.

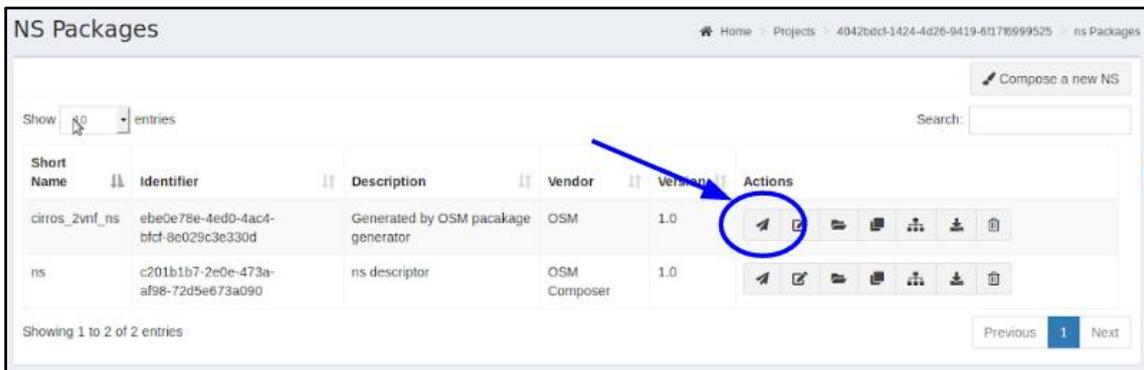


Ilustración 11 - Lanzar instancia

Una vez seleccionado se abre una ventana en la que tenemos que seleccionar que se quiere lanzar sobre OpenStack.

Con todo ello en NS se podría ejecutar de manera correcta.