

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN ESCENARIO
DIDÁCTICO NFV UTILIZANDO HERRAMIENTAS OPEN SOURCE**

TRABAJO FIN DE MÁSTER

Jessica Avilés Armijos

2017

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

Máster Universitario en Ingeniería de Redes y Servicios Telemáticos

TRABAJO FIN DE MÁSTER

**ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN ESCENARIO
DIDÁCTICO NFV UTILIZANDO HERRAMIENTAS OPEN SOURCE**

Autor
Jessica Avilés Armijos

Director
David Fernández Cambrero

Departamento de Ingeniería de Sistemas Telemáticos

2017

Resumen

Las redes de datos tradicionalmente han estado formadas por múltiples equipos y enlaces de comunicación que conectan a sus usuarios con los servicios, se trata de un conjunto de tecnología de infraestructura de propósito bastante específico. Una red de comunicación es por tanto una infraestructura compleja y muy rígida, en la que cualquier cambio que se quiera introducir implica largos periodos de tiempo y a la que habría que involucrar varios actores para que intervengan en su construcción. Por otra parte la industria proveedora de equipos se ha centrado en el desarrollo de hardware para funciones específicas, lo que ha favorecido un modelo en el que las tecnologías proporcionadas por diferentes suministradores, son exclusivas, y difícilmente interconectables con las de otros proveedores. Esto contribuye a que las redes acaban coexistiendo forzosamente con un número elevado de diferentes tecnologías. Toda esta complejidad dificulta además enormemente la operación y hace que la red sea inflexible y que no pueda adaptarse a cambios. Frente a esto las operadoras de telecomunicación se han puesto como objetivo la transformación de la red en una infraestructura más eficiente, flexible, moldeable e inteligente.

Dicho lo anterior, el objetivo de este trabajo es demostrar la funcionalidad y beneficios de una implementación de NFV (Network Function Virtualization), a través de la creación de un caso de uso funcional, utilizando como marco de referencia la arquitectura NFV, según los estándares de la ETSI (European Telecommunications Standards Institute) GS NFV 002, más específicamente en la capa de Gestión y Orquestación MANO. Para ello es preciso investigar sobre las distintas tecnologías que se implementan dentro de NFV, sus componentes y roles dentro de la arquitectura NFV. De lo anterior, debido al rápido crecimiento de estas tecnologías, se han ido desarrollando algunos proyectos y herramientas para la implementación de escenarios de funciones de red virtualizadas.

De ahí que, este trabajo se dividirá en fases; la primera de ellas será la realización de un estado del arte sobre NFV, sobre su arquitectura y las tecnologías en las que se cimienta. En esta etapa es importante conocer sobre los proyectos y herramientas que vienen trabajando esta tecnología y como se van complementando y tienen su funcionalidad específica dentro de la arquitectura

NFV. Para ello, se realizará una comparación de cada una de las herramientas, esta será una parte importante para la posterior elección de una de ellas para el desarrollo e implementación de un escenario NFV. Como siguiente fase, después del análisis de las herramienta NFV, se escoge la que mejor cumplan con los requerimientos para alcanzar la funcionalidad del escenario propuesto. Una vez implementadas las herramientas necesarias se realiza el desarrollo del caso de uso NFV.

Abstract

Data networks have traditionally have been formed by multiple hardware and communication links that connect their users with services, it is a set of infrastructure technology of specific purpose. A communication network is therefore a complex and very rigid infrastructure, in which any change that is wanted to introduce involves long periods of time and which would have to involve several actors to intervene in its construction. On the other hand, the equipment supplier industry has focused on the development of hardware for specific functions, which has favored a model in which the technologies provided by different suppliers are exclusive and difficult to interconnect with those of other suppliers. This contributes to the fact that networks end up coexisting with a large number of different technologies. All this complexity also complicates the operation and makes the network inflexible and unable to adapt to changes. As a result, telecommunications operators have focused on transforming the network into a more efficient, flexible, moldable and intelligent infrastructure.

The aims of this work is to demonstrate the functionality and benefits of an NFV implementation, through the creation of a functional use case, using the NFV architecture as reference framework, according to ETSI GS NFV standards 002, more specifically in the MANO Management and Orchestration layer. For this, it is necessary to investigate the different technologies that are implemented within NFV, its components and roles within the NFV architecture. Due to the rapid growth of these technologies, some projects and tools have been developed for the implementation of scenarios of virtualized network functions.

Hence, this work will be divided into phases; the first of them will be the realization of a state of the art on NFV, its architecture and the technologies in which it is based. At this stage it is important to know about the projects and tools that are working this technology and how they complement each other and have their specific functionality within the NFV architecture. To do this, a comparison of each of the tools will be done, this will be an important part for the subsequent choice of one of them for the development and implementation of an NFV scenario. As a next step, after the analysis of the NFV tool, the one that best meets the requirements to reach the functionality of the proposed scenario is chosen. Once the necessary tools are implemented, the NFV use case is developed.

Índice General

Resumen	II
Abstract	IV
Índice General	V
Índice de Figuras	IX
Siglas	XI
1 Introducción	1
1.1 Motivación	2
1.2 Objetivos	3
2 Tecnologías implementadas en NFV	5
2.1 Virtualización	5
2.1.1 Beneficios de la virtualización	6
2.1.2 Componentes de una máquina virtual	6
2.1.3 Virtualización de Servidores y Virtualización de Redes	7
2.2 Computación en la Nube	9
2.2.1 Características	10
2.2.2 Modelos de Servicio de Computación en la Nube	11

2.3	Virtualización en la Nube	12
2.3.1	Tipos de nubes	13
3	Virtualización de Funciones de Red (NFV)	14
3.1	Estandarización de NFV	15
3.2	Principios de NFV	17
3.2.1	Encadenamiento de Servicios	17
3.2.2	Gestión y Orquestación MANO	17
3.2.3	Arquitectura Distribuida	18
3.3	Arquitectura de referencia NFV	18
3.3.1	Capa de Infraestructura NFV	19
3.3.2	VNF / EMS	20
3.3.3	Gestión y orquestación de NFV (NFV-MANO)	21
3.3.4	OSS / BSS	22
4	Proyectos y Plataformas NFV	24
4.1	OpenStack	24
4.1.1	Componentes	25
4.2	OPNFV	27
4.2.1	Objetivos de OPNFV	27
4.3	CloudNFV	29
4.3.1	Arquitectura	30
4.3.2	Especificaciones	31

4.4	CloudBand	31
4.4.1	Componentes	32
4.5	Open Source MANO OSM	33
4.5.1	Objetivos	34
4.5.2	Principios Arquitectónicos	34
4.6	Cloud4NFV	37
4.6.1	Características	37
4.6.2	Arquitectura	38
4.7	Open-O	39
4.7.1	Arquitectura	40
4.8	Cloudify	41
4.8.1	Arquitectura	42
5	Implementación del Escenario NFV	44
5.1	Elección de las Herramientas y Plataformas	44
5.1.1	Orquestador NFV	44
5.1.2	VNF Manager	45
5.1.3	Virtualized Infrastructure Manager (VIM)	45
5.2	Arquitectura	46
5.2.1	Open Source Mano	46
5.2.2	OpenStack	48
5.3	Despliegue	49

5.3.1	NSD	50
5.3.2	VNFD	51
5.3.3	Lenguaje y formato de modelado de datos	52
5.3.4	Servicio RO	53
5.3.5	Acoplamiento Open Source Mano y Open Stack	53
5.3.6	Catálogos NFV	58
5.4	Resultados	61
6	Conclusiones y Lineas Futuras de Investigación	66
6.1	Conclusiones	66
6.2	Lineas Futuras de Investigación	68
7	Bibliografía	70
8	Anexos	72

Índice de Figuras

List of Figures

1	Beneficios de la Virtualización	6
2	Virtualización de redes con VLAN	8
3	Capas de la Nube	11
4	Principio de Encadenamiento de Servicios	18
5	Arquitectura NFV	19
6	Infraestructura NFVI	20
7	Capa de VNF/EMS	21
8	Arquitectura NFV-MANO	22
9	Gestión y Orquestación MANO de VNF	23
10	OSS/BSS	23
11	Arquitectura Conceptual de OpenStack	27
12	Diagrama OPNFV	29
13	Arquitectura OPNFV ETSI	29
14	Arquitectura CloudNFV	30
15	Arquitectura CloudBand	33
16	Arquitectura Open Source Mano - ETSI	34
17	Arquitectura OSM ETSI	35
18	Componentes principales OSM	37

19	Arquitectura Cloud4NFV	38
20	Arquitectura OPEN-O	40
21	Arquitectura Cloudify	42
22	Arquitectura Implementación MANO	47
23	Escenario Open Source MANO	48
24	GUI Open Source MANO	48
25	Escenario OpenStack	50
26	Modelo de Objetos de NSD	51
27	Conexiones VNF	52
28	Estructura de un despliegue de Open Source MANO	53
29	Carga de paquetes ejemplo NSD y VNFD en OSM	54
30	Servicio RO	55
31	Plugin Model	55
32	Ciclo de vida NFV	56
33	Despliegue de Instancia en OSM	57
34	Flujo de trabajo OpenStack	58
35	Servicio de Catálogo NFV	59
36	Ejemplo de implementación NFV	61
37	Instancia OSM desplegada en OpenStack	63
38	Estado del Hypervisor de Openstack	64
39	Topología de Red creada	64
40	Instancia NFV	65

Siglas

5G	Quinta Generación de Telefonía Móvil
API	Interfaz de programación de aplicaciones
ASIC	Application Specific Integrated Circuit
BSS	Sistemas de Apoyo Empresarial
COTS	Commercial off the shelf
CPE	Customer Premises Equipment
CPs	Connection Points
CPs	PNF Descriptor
DC	Domain controlles
EMS	Sistema de Gestión de Elementos
ETSI	European Telecommunications Standards Institute
IaaS	Infraestructura como Servicio
IND	Infraestructura de Dominio de red
IOT	Internet of Things
ISG	Grupo de Especificación de la Industria
ISP	Proveedor de Servicios de Internet
LXC	Linux Container
MANO	Gestión y Orquestación
MWC	Mobile World Congress
NFV	Virtualización de Funcione de Red
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NMS	Network Management System
NOC	Network Operators Council

NS	Network Services
NSD	Network Services Descriptor
NSS	Network Switching Subsystem
OPNFV	Open Platform for NFV
OSS	Sistemas de Soporte de Operaciones
PaaS	Plataforma como Servicio
PoC	Prueba de Concepto
REST	Representational State Transfer
RO	Orquestador de Recursos
SDN	Software-defined networking
SFC	Service Function Chaining
SO-ub	Orquestador de Servicios de Red
TSC	Technical Steering Committee
VCA	Configuración y Abstracción de VNF
VDU	Virtual Deployment Unit
VIM	Virtualized Infrastructure Manager
VL	Virtual Link
VLAN	Red de área local virtual o LAN virtual
VLD	Virtual Link Descriptor
VM	Virtual Machine
VNF	Función de Red Virtualizada
VNF-FGD	VNF Forwarding Graph Descriptor
VNFC	VNF Components
VNFD	VNF Descriptor
VNFM	VNF Manager
WG	Working Groups

1 Introducción

Hoy en día la tendencia de los usuarios es obtener la mayor velocidad posible pero sin pagar más por el servicio; también están las operadoras que están dispuestas a invertir en infraestructura nueva pero sólo cuando se garantiza el retorno de la inversión. Ahí es donde la escalabilidad, flexibilidad y confiabilidad indican el nuevo camino hacia el futuro. Por esta razón, la virtualización juega un papel fundamental para crear un ambiente totalmente disponible que garantiza el acceso a aplicaciones en todo momento. La virtualización de Funciones de Red es un paso gigante para construir una red más rápida y escalable, que explote las funcionalidades de las redes definidas por software SDN.

NFV puede acelerar a gran escala la implementación de nuevos servicios a menor coste, con menor esfuerzo y de una forma más controlada. De la misma manera la capacidad de compartir recursos entre servicios, aplicaciones y usuarios, esto implica un aumento de velocidad que puede ser aprovechado por operadoras y proveedores de servicios y brindar a sus usuarios una mejor experiencia y satisfacción. NFV cambia el modelo de negocio, los servicios tienen un cargo por uso, lo cual significa que los suscriptores pagan por lo que consumen.

La finalidad de esta virtualización es la de trasladarla a los dispositivos de red destinados a realizar funciones específicas. NFV rompe con el esquema: “un dispositivo, una función en hardware personalizado” y empieza a permitir que sus sistemas operativos de red se ejecuten en un ambiente virtualizado. Uno de los principales objetivos de la virtualización de red es integrar todas las políticas de red en un esquema automatizado que permita gestionar redes virtuales. El reto de todo esto radica en la redefinición de procesos para simplificar el funcionamiento y acelerar la red. Esto conlleva a la necesidad de que la red funcione como una plataforma abierta y flexible. Esta capacidad es la que tienen en sus manos los proveedores para dejar de ser un proveedor de acceso y convertirse en un proveedor de una amplia gama de servicios y con esto mejorar significativamente la satisfacción del cliente.

Las operadoras, proveedores de servicios y empresas quieren tener el mejor servicio con la mejor red; y los usuarios no desean saber de megabits por segundo sino de un servicio veloz y confiable, por lo que la virtualización de funciones de red representa una de las soluciones más prometedoras. Para conseguir este objetivo se plantean cambios en la arquitectura de la red, en las infraes-

estructuras que soportan los equipos, en la centralización-descentralización de las infraestructuras, etc. El objetivo de esta transformación es conseguir unas infraestructuras moldeables, superando las limitaciones que impone la infraestructura física actual. La clave está en conseguir que la red sea programable, y por ello la virtualización de red es uno de los ingredientes principales.

Por otro lado, en los últimos años las Telcos más importantes tanto de Europa como de Asia han venido trabajando el tema de las NFV conjuntamente con la ETSI (Instituto Europeo de Normas de Telecomunicaciones), esto con el apoyo de comunidades académicas, mismos proyectos que serán discutidos más adelante en el desarrollo de este trabajo.

1.1 Motivación

Tal y como se encuentra la estructura de red actual requiere con urgencia un cambio y la necesidad de implementación de un nuevo enfoque. Los nuevos servicios de Internet de Cosas (IoT) y 5G amenazan con llevar los costos de red a niveles insostenibles. Hoy en día las redes han tendido a crecer desmesuradamente y necesitan cada vez una mayor cantidad y variedad de aparatos hardware patentados, lo que conduce a consecuencias negativas como:

- Los nuevos servicios de red requieren diferentes tipos de dispositivos de hardware, lo que conlleva la necesidad de mayor espacio físico y potencia en los centros de datos.
- Nuevo hardware significa gastos adicionales de capital.
- Una vez que se adquieren nuevos aparatos de hardware, los operadores se enfrentan a la dificultad de diseñar, integrar y operar dispositivos cada vez más complejos basados en hardware.
- Los aparatos basados en hardware alcanzan rápidamente el final de su vida útil.

El enfoque NFV trata de dejar atrás la dependencia de una variedad de plataformas de hardware para dar paso al uso de un pequeño número de tipos de plataformas estandarizadas, con técnicas de virtualización utilizadas para proporcionar la funcionalidad de red necesaria. En el documento de especificación NFV definido por la ETSI (White Paper ETSI NFV), se expone la convicción de que el enfoque NFV es aplicable a cualquier procesamiento de paquetes de

plano de datos y la función de plano de control en las infraestructuras de red fija y móvil.

Si el NFV se implementa de manera eficiente y eficaz, puede proporcionar una serie de beneficios en comparación con los enfoques de redes tradicionales. Entre los múltiples beneficios de NFV se recalcan:

- La reducción de CapeX y OpEx son quizás los principales impulsores de NFV.
- Capacidad de innovar y desplegar servicios rápidamente.
- Facilidad de interoperabilidad gracias a las interfaces estandarizadas y abiertas.
- Uso de una plataforma única para diferentes aplicaciones, usuarios y arrendatarios.
- Los servicios se pueden escalar rápidamente hacia arriba o hacia abajo según sea necesario.

Todo lo mencionado hace que sea necesario un cambio para como son concebidas las redes actualmente. Esto se hace posible a través de la implementación de NFV, tal como se explica a continuación.

1.2 Objetivos

Como objetivo general, se plantea en este trabajo realizar un análisis del estado del arte de NFV, presentando sus características, principios, arquitectura y beneficios. Para mejor comprensión del estado del arte de NFV es necesaria la implementación de un escenario práctico-didáctico, mismo que se desarrollará con una o varias herramientas Open Source existentes para la Gestión y configuración de NFV.

Como objetivos específicos del trabajo se tiene:

- Investigar el estado del arte referente a NFV con la finalidad de describir las principales motivaciones de NFV y los beneficios de su implementación. En esta parte es necesario analizar la arquitectura de NFV y las tecnologías que aquí se despliegan.

- Realizar un análisis sobre los proyectos y plataformas de código abierto existentes en el tema de NFV y exponer la función que cumplen dentro de esta arquitectura.
- Analizar cuál de los proyectos de NFV descritos es el más adecuado para el diseño de un escenario práctico y didáctico para el Manejo de NFV.
- Experimentar con el escenario didáctico la implementación de NFV, probar su funcionamiento y utilizar el orquestador de recursos para automatizar el despliegue de un escenario de red.

2 Tecnologías implementadas en NFV

En esta sección se describe el estado del arte de cada una de las tecnologías que se utilizan para el despliegue del sistema planteado. Lo que se muestra es una descripción de cada tecnología, sus características más importantes y la arquitectura que implementa.

2.1 Virtualización

La virtualización es la tecnología principal en la que se fundamenta NFV, por lo que es necesario adentrarnos en este tema y conocer conceptos y características de la misma.

La virtualización es la tecnología que se utiliza para ejecutar múltiples sistemas operativos o aplicaciones sobre una sola infraestructura física, proporcionando una vista abstracta del hardware. Estos sistemas operativos o aplicaciones se ejecutarán de forma aislada mientras comparten recursos de hardware. La virtualización no es un concepto nuevo, en la década de 1960, fue IBM quién desarrollo el sistema operativo CP-40 con el objetivo de compartir memoria entre usuarios y aplicaciones, fueron ellos quienes sentaron las bases para los conceptos de virtualización de hoy en día. El primer producto comercial para habilitar la virtualización en plataformas x86 fue lanzado por VMWare en 1999 y llamado VMWare Workstation. Pronto fue seguida por "VMWare ESX.^{en} 2001 para el mercado de servidores. Pronto siguieron otras implementaciones, como Hyper-V (por Microsoft), VirtualBox (por Oracle) y soluciones de virtualización de código abierto como Xen y KVM. En 2005, Intel y AMD anunciaron capacidades de procesador que ofrecerían soporte de CPU para la virtualización asistida por hardware [4]. El objetivo de la virtualización es ofrecer un mecanismo para ejecutar múltiples aplicaciones sin dependencia, compartiendo un mismo sistema operativo y recursos de hardware. A cada aplicación se le hace creer que posee recursos de hardware pero no es consciente de que este hardware es un subconjunto abstraído de un grupo de hardware más grande. Las aplicaciones tienen segregación entre sus procesos, el uso del disco y del sistema de archivos, la gestión de usuarios y la creación de redes.

2.1.1 Beneficios de la virtualización

El objetivo principal de la virtualización es ofrecer un mecanismo para ejecutar múltiples aplicaciones sin que dependan la una de la otra ni tengan la necesidad de conocimiento mutuo, teniendo en común un sistema operativo y recursos de hardware. Cada aplicación cree que posee recursos de hardware propios, las aplicaciones tienen segregación entre sus procesos, el uso del disco y del sistema de archivos, la gestión de usuarios y la creación de redes. Con la llegada de la virtualización, el número de servidores en un mismo centro de datos se redujo radicalmente, con menos servidores haciendo el mismo trabajo y alojando múltiples aplicaciones simultáneamente. Los beneficios resultantes son indiscutibles: ahorro de energía, espacio y costos operacionales; reducción del costo total de propiedad; y resiliencia del negocio. Fig. 1. [4].

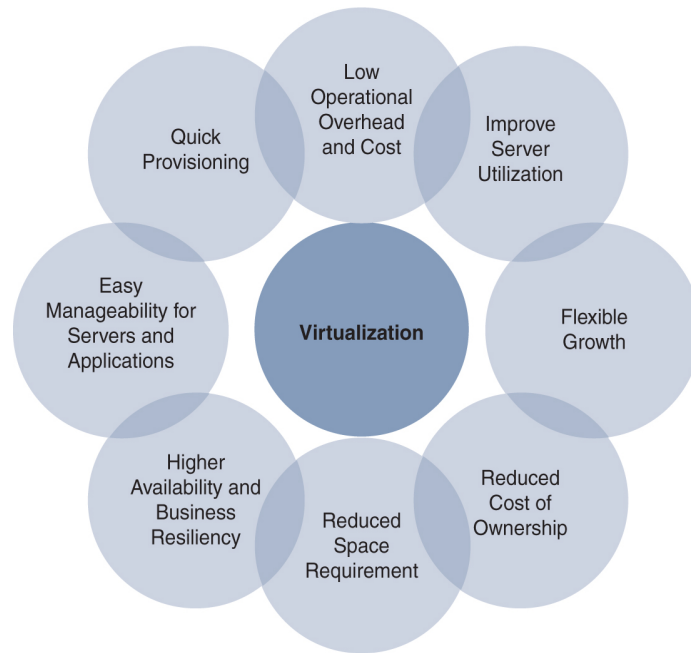


Figura 1: Beneficios de la Virtualización

2.1.2 Componentes de una máquina virtual

Las máquinas virtuales tienen tres componentes principales:

- Sistema operativo host : es el sistema operativo que se ejecuta directamente en el hardware, este tiene la visibilidad del conjunto completo

de recursos de hardware que están disponibles para ser asignados a las máquinas virtuales.

- Administrador de máquina virtual (VMM o hypervisor) : La función del hypervisor es permitir la creación de la máquina virtual, la asignación de sus recursos, modificación de sus parámetros y eliminación de la misma. Los hipervisores son clasificados en dos tipos: de tipo 1; son autónomos y, debido a su interacción directa con el hardware, tienen más flexibilidad y son más seguros. Los hipervisores de tipo 2 son más fáciles y rápidos de desarrollar y son más flexibles porque no tienen que ser desarrollados o probados contra hardware específico.
- Sistema operativo invitado: El sistema operativo que se ejecuta en este servidor virtual se denomina sistema operativo invitado. Al igual que cualquier otro servidor, necesita un sistema operativo para arrancar, administrar los dispositivos y ejecutar aplicaciones en él.

2.1.3 Virtualización de Servidores y Virtualización de Redes

La virtualización de la que se ha mencionado hasta ahora es la virtualización de servidores. No obstante, la virtualización en los últimos tiempos se ha implementado no sólo en los servidores sino en otros lugares como redes y dispositivos de almacenamiento, por lo que el concepto de virtualización tomó un significado diferente dependiendo de su contexto y uso. Entre las áreas más significativas en las que se ha implementado la virtualización están:

- Virtualización de servidores
- Virtualización de red
- Virtualización de funciones de red (NFV) [4]

Virtualización de servidores

Varios servidores físicos que proporcionan servicios como de correo electrónico, base de datos, administración y web se pueden virtualizar en un número mucho menor de servidores. Aún así, se debe tomar en cuenta la redundancia, ya que no es un buen diseño el virtualizar servidores como los de bases de datos primarios y de respaldo en el mismo hardware, por ejemplo. La virtualización

de servidores es una tecnología madura y ha demostrado ser bastante eficiente. Hoy en día se han desarrollado una gran cantidad de herramientas de software con el objeto de proporcionar un despliegue más rápido y efectivo para la virtualización de servidores y también para que los administradores tengan la capacidad de administrar y optimizar su rendimiento.

Virtualización de Redes

La virtualización de red es un concepto que a menudo se confunde con NFV, pero no tiene una relación directa con las ideas de virtualización que se discutieron anteriormente. En este apartado se utiliza la palabra virtualización en un contexto distinto al de la virtualización de servidores. El enfoque de esta virtualización es que una única red física se divide en varias redes lógicas y comparten infraestructura pero para el usuario final esto no es visible y los protocolos y tecnologías utilizados hacen que estas redes parezcan ser completamente independientes. Las redes lógicas o virtuales proporcionan aislamiento, privacidad y segregación a nivel de red. Un ejemplo de virtualización de red es tal vez el uso de LAN virtual (VLAN), tal como se ilustra, Fig. 2 [4].

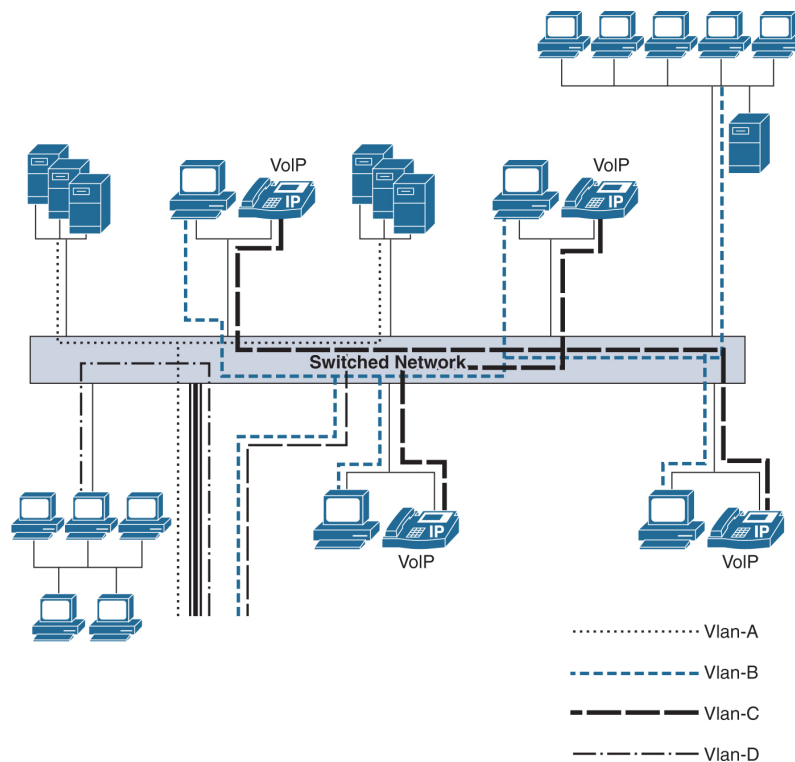


Figura 2: Virtualización de redes con VLAN

Las técnicas de virtualización de red hicieron posible la ejecución simultánea de servicios como Internet de banda ancha, transmisión de vídeo y voz sobre IP, las que están lógicamente separadas en la misma red física. Los principales beneficios incluyen el ahorro en los costos de implementación, administración y mantenimiento de la infraestructura. Para las empresas, la virtualización de red ofrece formas rentables de interconectar sus segmentos de intranet, pequeñas oficinas y trabajadores remotos a través de Internet o de los servicios de redes privadas virtuales que pueden recibir a través de los ISP.

Virtualización de Funciones de Red NFV

La virtualización de las funciones de red, traslada la idea de la virtualización de servidores a dispositivos de red destinados a realizar funciones específicas en la red. Como ya se ha venido tratando, el éxito de la virtualización de servidores radica en poder disminuir costos tanto por hardware como por administración, esto ha atraído a los operadores de redes a mirar a NFV y a los proveedores de equipos y fabricantes a romper con el paradigma de “ un equipo, una función de hardware ” [4]. Como se tratará más adelante, el Whitepaper de NFV, mismo que hace referencia al éxito que ha tenido la virtualización de servidores y propuso hacer lo mismo con funciones de red con el objetivo de lograr beneficios similares. Más adelante se tratará con detenimiento este tema en el estado del arte de NFV.

2.2 Computación en la Nube

Con los recientes avances en programación y tecnologías de información, se han mostrado también nuevas aplicaciones para Internet, en este caso la computación en la nube o Cloud Computing, en la cual los recursos informáticos son ofrecidos y consumidos como servicios a través de Internet sin que los usuarios tengan conocimiento de la infraestructura que hay detrás [16]. La computación en la nube hace posible el aprovechamiento económico de la infraestructura, optimizar recursos según necesidades instantáneas, el ahorro en personal y un ecosistema de comunicaciones y cómputo, con mecanismos de competencia.

2.2.1 Características

No es necesario disponer de un equipo potente, tan sólo de un aparato con conexión a internet; esto debido a que el dispositivo del usuario no realizará ningún proceso complejo y los ficheros pueden guardarse en la nube. Los servidores en donde se hallan los programas que se utilicen son los encargados de las tareas complicadas que antes se realizaba localmente. Con el uso del Cloud Computing no hay necesidad por parte del usuario de conocer la infraestructura detrás de esta, ya que pasa a ser una abstracción, “una nube” donde las aplicaciones y servicios pueden fácilmente crecer, funcionar rápido y con pocas fallas. Este tipo de servicio se puede pagar según alguna métrica de consumo, no por el equipo usado en sí, sino por uso de CPU/hora. Entre otras características podemos mencionar:

- **Auto reparable:** En caso de surgir un fallo, el último respaldo (backup) de la aplicación se convierte automáticamente en la copia primaria y a partir de esta se genera uno nuevo.
- **Escalable:** Todo el sistema y su arquitectura es predecible y eficiente. Se establece un nivel de servicios que crea nuevas instancias de acuerdo a la demanda de operaciones existente de tal forma que se reduzca el tiempo de espera y los cuellos de botella.
- **Virtualización:** las aplicaciones son independientes del hardware en el que corran, incluso varias aplicaciones pueden corren en una misma máquina o una aplicación puede usar varias maquinas a la vez.
- **Alto nivel de seguridad:** El sistema está creado de tal forma que permite a diferentes clientes compartir la infraestructura sin preocuparse de ello y sin comprometer su seguridad y privacidad; de esto se ocupa el sistema proveedor que se encarga de cifrar los datos.
- **Disponibilidad de la información:** No se hace necesario guardar los documentos editados por el usuario en su computadora o en medios físicos propios ya que la información radicaría en Internet permitiendo su acceso desde cualquier dispositivo conectado a la red (con autorización requerida). [13]

2.2.2 Modelos de Servicio de Computación en la Nube

La computación en la nube basa su arquitectura haciendo una separación entre hardware, plataforma y aplicaciones, tal y como indica la Fig. 3:

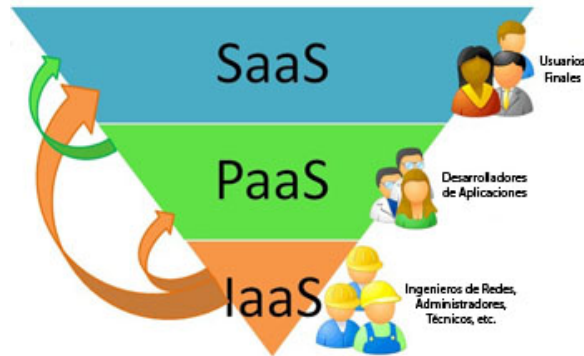


Figura 3: Capas de la Nube

Software como Servicio (SaaS)

Se encuentra en la capa más alta y consiste en la entrega de aplicaciones completas como un servicio. El proveedor ofrece el SaaS (Software as a Service), para esto dispone de una aplicación y es el mismo proveedor el encargado de operar y mantener esta aplicación que frecuentemente es desarrollada por él mismo. Con ella se encarga de dar servicio a multitud de clientes a través de la red, sin que éstos tengan que instalar ningún software adicional. Las actividades son gestionadas desde alguna ubicación central, en lugar de hacerlo desde la sede de cada cliente, permitiendo a los clientes el acceso remoto a las aplicaciones mediante la web, lo mismo para con las actualizaciones, eliminando la necesidad de descargar parches por parte de los usuarios finales [16].

Plataforma como Servicio (PaaS)

Su objetivo se centra en un modelo en el que se proporciona un servicio de plataforma con todo lo necesario para dar soporte al ciclo de planteamiento, desarrollo y puesta en marcha de aplicaciones y servicios web a través de la misma. El proveedor es el encargado de escalar los recursos en caso de que la aplicación lo requiera, de que la plataforma tenga un rendimiento óptimo, de la seguridad de acceso, etc.

Con el uso de PaaS, se abstrae del hardware físico al cliente, lo cual es interesante para muchos desarrolladores web, y es probable que llegue a reemplazar a las empresas de alojamiento tradicionales. Incluso, también a los administradores de sistemas, ya que no se necesita controlar un sistema ni hay optimización posible más allá del código y sus algoritmos [16].

Infraestructura como Servicio (IaaS)

La idea básica es la de hacer uso externo de servidores para espacio en disco, base de datos, ruteadores, swtiches así como tiempo de computo evitando de esta manera tener un servidor local y toda la infraestructura necesaria para la conectividad y mantenimiento dentro de una organización. Con una IaaS lo que se tiene es una solución en la que se paga por consumo de recursos solamente usados: espacio en disco utilizado, tiempo de CPU, espacio para base de datos, transferencia de datos, etc.

Las IaaS permiten desplazar al proveedor la mayor parte de los factores relacionados con la gestión de las máquinas con el ahorro de costos al pagar sólo por lo consumido y olvidarse de tratar con máquinas y su mantenimiento. Por otro lado, IaaS puede permitir una escalabilidad automática o semiautomática, de forma que se puedan contratar más recursos según los se requieran [13].

2.3 Virtualización en la Nube

La virtualización es un elemento fundamental en el desarrollo óptimo de la computación en la nube, y se enfoca principalmente a la plataforma. Puede considerarse como una abstracción de los recursos tecnológicos que permite a los servidores crear dispositivos virtuales los cuales pueden ser usados para aumentar dichos recursos más que como sistemas separados. Mediante la virtualización se permite tratar a un servidor como muchos servidores. Otro método usado es el clustering, que consiste en tratar a muchos servidores como uno solo.

2.3.1 Tipos de nubes

Nubes públicas

Estas se administran externamente por terceros, los contenidos de distintos clientes pueden encontrarse ubicados en los mismos servidores, sistemas de almacenamiento. Los usuarios finales usan la infraestructura de la nube en todas sus capas y no conocen qué trabajos de otros clientes pueden estar corriendo en el mismo servidor o red.

Nubes privadas

Los servicios no se ofrecen al público en general. La infraestructura está totalmente gestionado por una organización [2]. Las nubes privadas están en una infraestructura manejada por un solo administrador que controla qué aplicaciones debe correr y dónde. Son una buena opción para las compañías que necesitan alta protección de datos y manipulaciones a nivel de servicio.

Nubes híbridas

Una combinación de dos o más nubes individuales que pueden ser privado, compartido o público. Esto permite intercambiar datos y aplicaciones. Una combinación de nubes públicas y privadas. El cliente está en posesión de unas partes y comparte otras, esto además puede ser de manera controlada. Las nubes híbridas ofrecen la ventaja del escalado proporcionado externamente, bajo demanda, se añade la posibilidad de determinar cómo distribuir las aplicaciones a través de los ambientes diferentes [13].

3 Virtualización de Funciones de Red (NFV)

NFV se originó en el 2012 a partir de discusiones entre siete operadoras de redes de telecomunicaciones (AT&T, BT, Deutsche Telekom, Orange, Telecom Italia, Telefónica y Verizon), lo lidera ETSI (Instituto Europeo de Estándares de Telecomunicaciones). Estas discusiones dieron lugar a la publicación de NFV white paper, Virtualización de Funciones de Red: Una Introducción, Beneficios, Habilitadores, Desafíos y Llamado a la Acción (ISGN12) [10]. La tecnología clave que hace posible NFV es la virtualización. La idea de esta virtualización es la de romper con el esquema de los dispositivos destinados a realizar funciones de red e implementar estas funciones en software y ejecutarlas en máquinas virtuales.

NFV rompe con el esquema poco flexible de las redes típicas y empieza a permitir que sus sistemas operativos de red se ejecuten en un ambiente virtualizado. NFV es una significativa desviación de los enfoques tradicionales para el diseño, implementación y administración de servicios de red. NFV separa las funciones de red, tales como la traducción de direcciones de red (NAT), firewall, detección de intrusiones, DNS y almacenamiento en caché, desde dispositivos de hardware propietarios para que puedan ejecutarse en software en máquinas virtuales. NFV se basa en las tecnologías VM estándar, ampliando su uso en el dominio de red [25].

El software de los dispositivos de red ha sido históricamente propietario y personalizado, mientras que el hardware consiste en un procesador de bajo a medio, almacenamiento en disco y un gran número de interfaces físicas para la entrada y salida de datos (E/S). Estos dispositivos también utilizan CPU dedicadas para procesar y reenviar tráfico de red, y memoria especializada para búsqueda de direcciones. Estas CPUs de procesamiento de paquetes son altamente personalizadas para implementar funciones de red tales como reenvío, clasificación, colas y control de acceso y se implementan a menudo como ASIC (Application Specific Integrated Circuit) en los dispositivos de red tradicionales. Con NFV, el sistema operativo se virtualiza y se ejecuta en servidores COTS: la cantidad de computación, almacenamiento y recursos de interfaz necesarios para ejecutar varias instancias de estos sistemas operativos en un único servidor pueden ser fácilmente ajustados. Para compensar la falta de CPUs de reenvío y memoria caché de acceso rápido, se han desarrollado técnicas especiales de software para lograr un alto rendimiento mientras se usan CPU de uso general [4].

Con NFV, los diseños de red y la arquitectura pueden adaptarse de la misma manera que la virtualización de TI. Dado que la virtualización de las funciones de red sigue los pasos de la virtualización de servidores la misma que es relativamente madura, aprovecha una serie de herramientas desarrolladas para ello.

Características

Las características y ventajas principales que implementa una arquitectura NFV son:

- **Elasticidad:** La característica principal que distingue a NFV, es lograr que dicha función sea elástica, es decir que pueda ampliar su escalabilidad (scale-out) o reducirla (scale-in) en base al despliegue y remoción automáticos de máquinas virtuales o contenedores.
- **Performance:** La performance de las funciones de red no debe afectarse al pasar a un esquema virtualizado, por lo que la infraestructura de virtualización en NFV cuenta con técnicas avanzadas en el procesamiento de archivos y paquetes.
- **Agilidad:** Una de las principales motivaciones de NFV es vencer la lentitud que típicamente se tiene con equipamiento dedicado en el despliegue de nuevas configuraciones o funcionalidades, por lo que una arquitectura NFV se caracteriza por la rapidez en programar las funciones de red para lograr los cambios requeridos.

3.1 Estandarización de NFV

La estandarización de NFV está siendo liderada por la ETSI. Aunque ETSI es una organización para el desarrollo de estándares, el objetivo de la NFV ISG no es la producción de estándares, sino conseguir el consenso de la industria en los requerimientos técnicos y de negocio de NFV, y en acordar enfoques comunes para alcanzar estos requerimientos bajo una arquitectura común. Los resultados son publicados de forma abierta y compartidos con los principales grupos de estandarización, foros industriales y consorcios, como IETF, NGMN Alliance, ONF, TM Forum etc.; con el fin de conseguir un esfuerzo de colaboración más amplio. La NFV ISG colabora con otros organismos de estandarización en caso de que sea necesaria cualquier tipo de estandarización para cumplir los requisitos.

Además de la dirección del ISG, el NOC y el TSC, el NFV ISG se compone de varios. En concreto, existen cuatro grupos de trabajo centrados en áreas específicas de NFV: infraestructura, arquitectura software, MANO y fiabilidad y disponibilidad. También hay dos EG (Expert Group) de carácter transversal: PER (rendimiento y portabilidad) y SEC (seguridad) [10].

Desde su primera reunión en enero de 2013, el Grupo de Especificación de la Industria (NFV ISG) del ETSI NFV ha crecido a 235 empresas, incluyendo 34 organizaciones proveedoras de servicios. Los primeros resultados publicados en Octubre de 2013 en su el White Paper introductorio, incluía un marco arquitectónico NFV que en la actualidad está siendo ampliamente utilizado en la industria para el desarrollo de productos, la estandarización y las nuevas iniciativas de Open Source para NFV (OPNFV) [10]. El NFV White Paper de la ETSI, trata de un documento guía que fué elaborado por el Instituto Europeo de Normas de Telecomunicaciones (ETSI), quién se encarga de la estandarización de la tecnología NFV. Este documento está creado como una guía de ayuda a los lectores para comprender los objetivos de la implementación y manejo de NFV y de esta forma lo orienten a conocer, diseñar y mejorar los escenarios NFV.

El objetivo específico de estos documentos no son la producción de estándares, sino el de obtener el consenso de la industria en los requerimientos técnicos y de negocio de NFV, y en acordar enfoques comunes para alcanzar estos requerimientos bajo una arquitectura común. Los resultados son publicados de forma abierta y compartidos con los principales grupos de estandarización, foros industriales y consorcios, como IETF ¹. Dentro de las especificaciones logradas, se lanzaron varios documentos en donde se explican con más detalla sus distintos componentes y estructura. Entre los que se destacan por su importancia son: Entorno de arquitectura, Infraestructura (NFVI), Gestión, Orquestación (MANO) y Casos de Uso [10].

En el primer White Paper publicado en octubre de 2012 [10] se presentó el concepto de Funciones de Red Virtualizadas (NFV) y esbozó los beneficios y desafíos para su implementación y despliegue, al igual que realizó un llamado para la acción, que sugiere tanto a la industria como a los académicos a combinar conocimientos para el desarrollo de esta nueva tecnología. NFV transformó la forma en que las redes eran construidas y operadas y aprovechó la tecnología estándar de virtualización de TI y consolidó los equipos "estándar de la industria" de servidores. En este documento también se emitió un llamamiento para que la industria cooperara en estos desafíos y se declaró la intención de

¹<http://www.etsi.org/nfv>

fomentar el crecimiento de un ecosistema abierto para NFV.

Posterior a esto fueron surgiendo las primeras especificaciones de la ETSI ISG en Octubre de 2013: GS NFV 001 “Use Cases”, GS NFV 002 “Architectural Framework”, GS NFV 003 “Terminology for Main Concepts in NFV”, GS NFV 004 “Virtualisation Requirements” y GS NFV-PER 002 “Proofs of Concepts Framework”. Estos cinco documentos establecen un marco de trabajo fundamental para el desarrollo de NFV, cubriendo: casos de uso, requerimientos, arquitectura marco y terminología. Además se establece un marco de trabajo para coordinar y promover demostraciones públicas de los aspectos claves de NFV. Esto hará más sencillo para los operadores de red y proveedores de soluciones NFV trabajar juntos mejorando la interoperabilidad y el desarrollo de un ecosistema abierto [11]. Desde aquello, el NFV ISG se ha convertido en el punto focal para el progreso de la industria en NFV. Este documento técnico proporciona una actualización sobre el trabajo del NFV ISG junto con sus perspectivas futuras para NFV.

3.2 Principios de NFV

Existen tres principios claves del NFV y están involucrados en la creación de servicios de red siendo estos los siguientes [25]:

3.2.1 Encadenamiento de Servicios

Los elementos de Función de Red Virtualizada (VNFs) son modulares y cada VNF proporciona funcionalidad limitada por sí solo. Para un flujo de tráfico dado, dentro de una aplicación, el proveedor de servicios dirige el flujo a través de múltiples VNF para lograr la funcionalidad de red deseada. Esto se conoce como encadenamiento de servicio. Véase Fig. 4. El encadenamiento de servicios permite insertar automáticamente servicios de redes físicas o virtuales, con las políticas de tráfico y las configuraciones adecuadas, formando parte de las ofertas basadas en soluciones de nube.

3.2.2 Gestión y Orquestación MANO

El Manejo y Orquestación de NFV hace referencia a la implementación, orquestación y administración del ciclo de vida tanto de los recursos lógicos

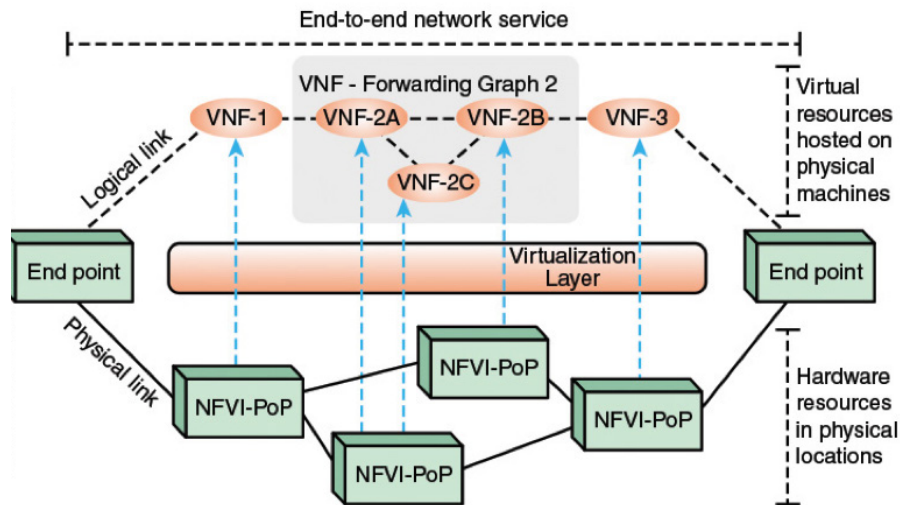


Figura 4: Principio de Encadenamiento de Servicios

como los físicos. Este proceso incluye la creación de instancias de VNF, encadenamiento de servicios VNF, supervisión, reubicación y cierre de instancias. MANO interactúa con (NFV externa) OSS/BSS, la cual permite a la NFV ser integrada en un entorno de gestión más amplio de toda la red ya existente. MANO también administra los elementos de la infraestructura NFV.

3.2.3 Arquitectura Distribuida

Un VNF puede estar compuesto por uno o más componentes VNF ó VNFC, cada uno de los cuales implementa un subconjunto de la funcionalidad del VNF. Cada VNFC puede desplegarse en una o varias instancias. Estas instancias pueden desplegarse en hosts separados y distribuidos para proporcionar escalabilidad y redundancia.

3.3 Arquitectura de referencia NFV

La arquitectura de NFV definido por el ISG NFV se divide en cuatro bloques principales, cada uno de los cuáles se describe a continuación. Fig. 5.

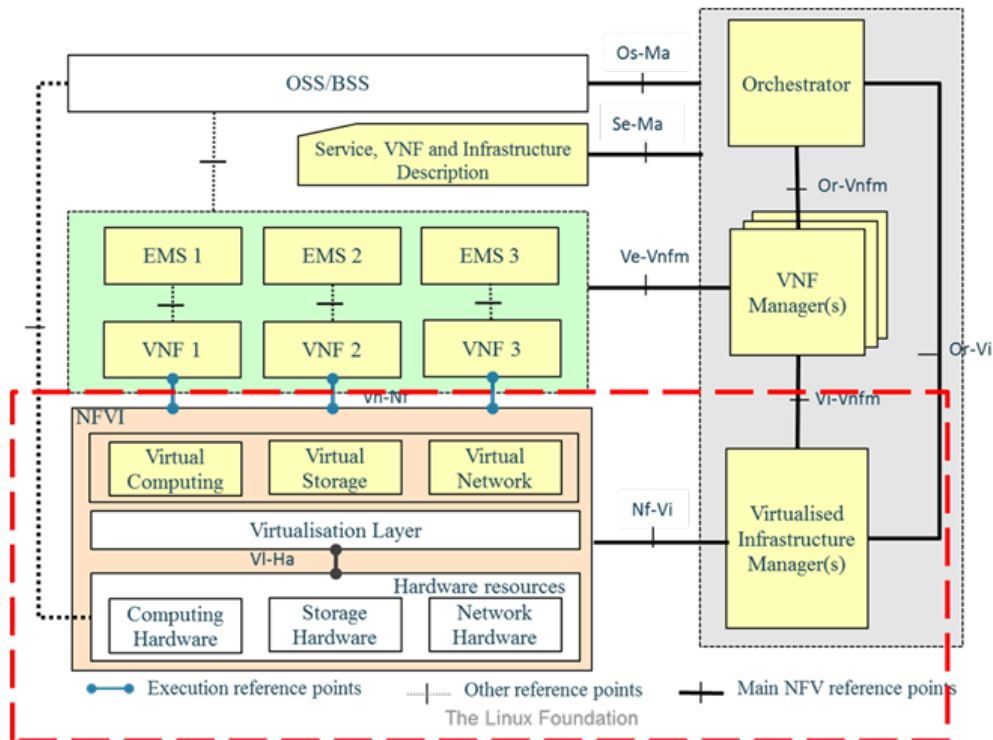


Figura 5: Arquitectura NFV

3.3.1 Capa de Infraestructura NFV

Comprende los recursos de hardware y software que crean el entorno en el que se despliegan los VNF. También llamada NFVI virtualiza la computación física, el almacenamiento y las redes y las coloca en grupos de recursos. Esta proporciona los recursos necesarios para soportar la ejecución de las funciones de red virtualizada. Incluye Hardware comercial-off-the-shelf (COTS), componentes de aceleración donde sea necesario, y una capa de software que virtualiza y abstrae el hardware que hay por debajo [11].

NFVI proporciona una infraestructura de múltiples usuarios aprovechando la tecnología de virtualización de TI estándar que pueden soportar múltiples casos de uso y campos de aplicación de forma simultánea [10]. La Fig. 6 ilustra los distintos bloques donde los proveedores pueden optar por aplicar software con el fin de ofrecer "productos compatibles NFV". Este marco arquitectónico está siendo utilizado para guiar el trabajo futuro en los diferentes grupos de trabajo NFV ISG, donde se analizan los detalles de cada uno de los bloques de construcción con más profundidad, así como un análisis del alcance de los componentes para priorizar el trabajo en curso dentro del ISG NFV todo esto se encuentra publicado y expuesto en el white paper.

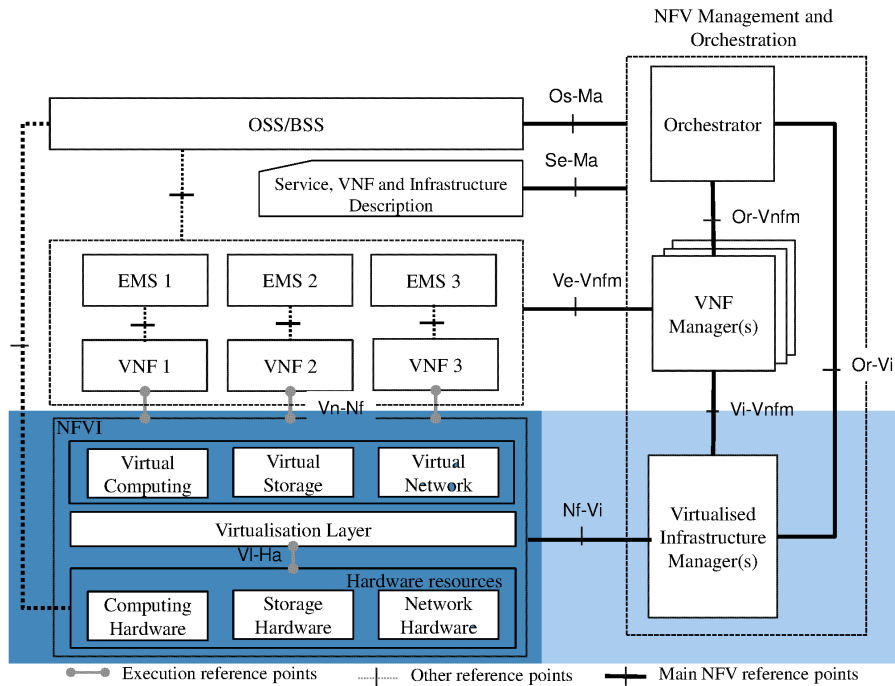


Figura 6: Infraestructura NFVI

El corazón de la arquitectura NFV es una colección de recursos y funciones conocidos como la infraestructura NFV (NFVI). El NFVI abarca tres dominios.

- **Compute domain:** Proporciona servidores y almacenamiento de alto volumen comercial (COTS).
- **Hypervisor domain:** Gestiona los recursos del dominio de cálculo a las máquinas virtuales de los dispositivos de software, proporcionando una abstracción del hardware.
- **Infraestructura de Dominio de red (IND):** Comprende todos los conmutadores genéricos de alto volumen interconectados en una red que se puede configurar para suministrar servicios de red de infraestructura.

3.3.2 VNF / EMS

Colección de VNFs implementados en software para ejecutarse en recursos virtuales de computación, almacenamiento y red, junto con una colección de sistemas de gestión de elementos (EMS) que gestionan los VNF. Es la implementación de software de una función de red que es capaz de correr sobre el

NFVI. Puede ser acompañado por un Element Management System (EMS), siempre y cuando sea aplicable a una función en particular la cuál gestiona un VNF individual y sus características. El VNF es la entidad correspondiente a todos los nodos de red y se espera que se entreguen como solo software dependiente del hardware [11]. Tal como se muestra en la figura Fig. 7, el color azul oscuro indica la capa de VNF y EMS, y el azul claro la gestión asociada con las funciones de orquestación.

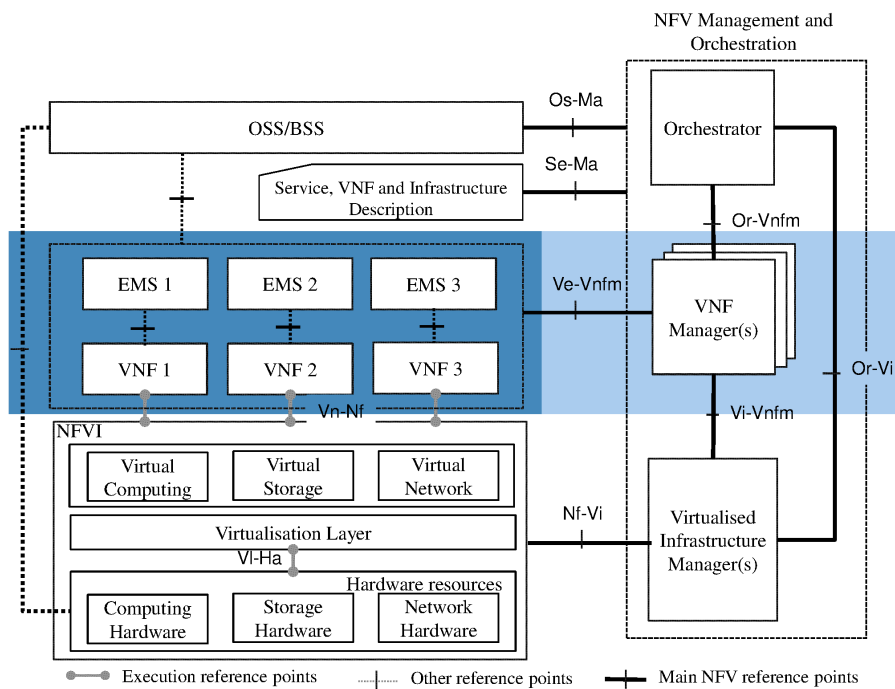


Figura 7: Capa de VNF/EMS

3.3.3 Gestión y orquestación de NFV (NFV-MANO)

Marco para la gestión y orquestación de todos los recursos en el entorno NFV. Se encarga de la gestión del ciclo de vida de los recursos físicos y del software que soportan la virtualización de la infraestructura y la gestión del ciclo de vida de los VNF. Esto incluye computación, red, almacenamiento y recursos de VM. (Véase Fig. 8). La Gestión y Orquestación de NFV se centra en las tareas de gestión específicas de virtualización necesarias en el marco NFV. Esta interactúa con la capa NFV externa OSS/BSS, lo que permite que NFV se integre en la gestión de la red [5].

NFV MANO, como se muestra en la Fig. 9 se divide en tres bloques funcionales:

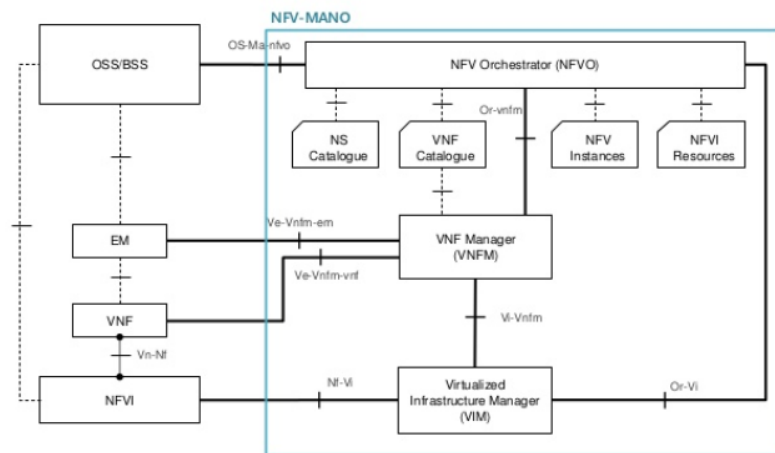


Figura 8: Arquitectura NFV-MANO

- **NFV Orchestrator:** Responsable del despliegue de nuevos servicios de red (NS) y paquetes de las funciones de red virtualizadas (VNF), NS gestión del ciclo de vida, Gestión de recursos globales; Validación y autorización de peticiones de funciones de red de recursos para de infraestructura de virtualización (NFVI)
- **VNF Manager:** Supervisa la gestión del ciclo de vida de las instancias VNF. Se encarga de la coordinación y adaptación para la configuración y la presentación de informes de eventos entre NFVI y E/NMS
- **Virtualized Infrastructure Manager (VIM):** controla y administra los recursos de computación, almacenamiento y red de la NFVI Para que la arquitectura NFV MANO funcione correcta y eficazmente, debe integrarse con interfaces (API) en los sistemas existentes. La capa MANO funciona con plantillas para VNF estándar, y ofrece a los usuarios el poder de escoger y elegir entre los recursos NFVI existentes para desplegar su plataforma o elemento.

3.3.4 OSS / BSS

Sistemas operativos y de soporte empresarial implementados por el proveedor de servicios VNF [25]. ETSI no nombra OSS y BSS como un componente dedicado a NFV. OSS y BSS solicitan servicios de una instancia NFV y consumen servicios producidos por una instancia NFV. Fig. 10.

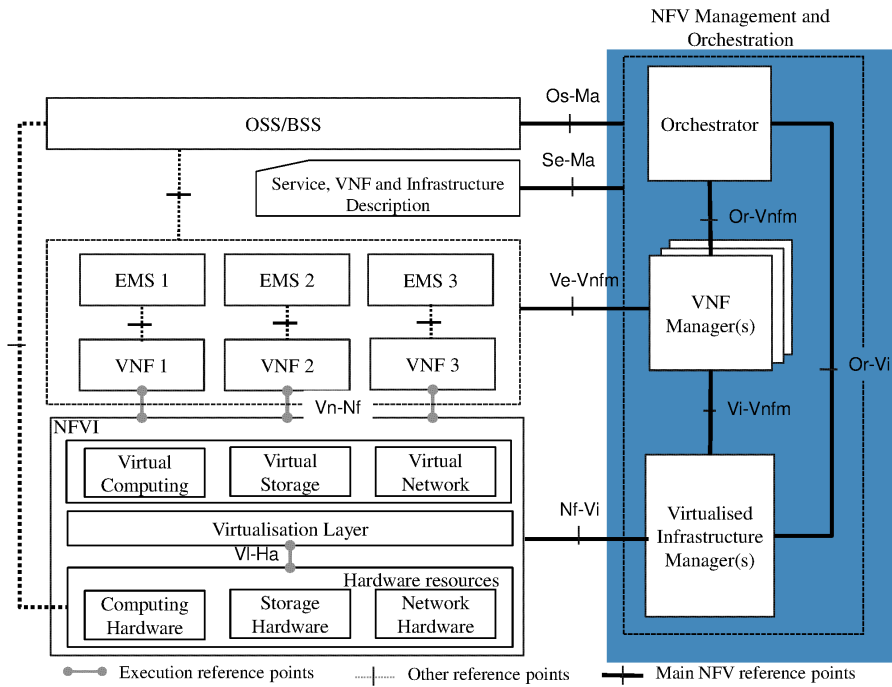


Figura 9: Gestión y Orquestación MANO de VNF

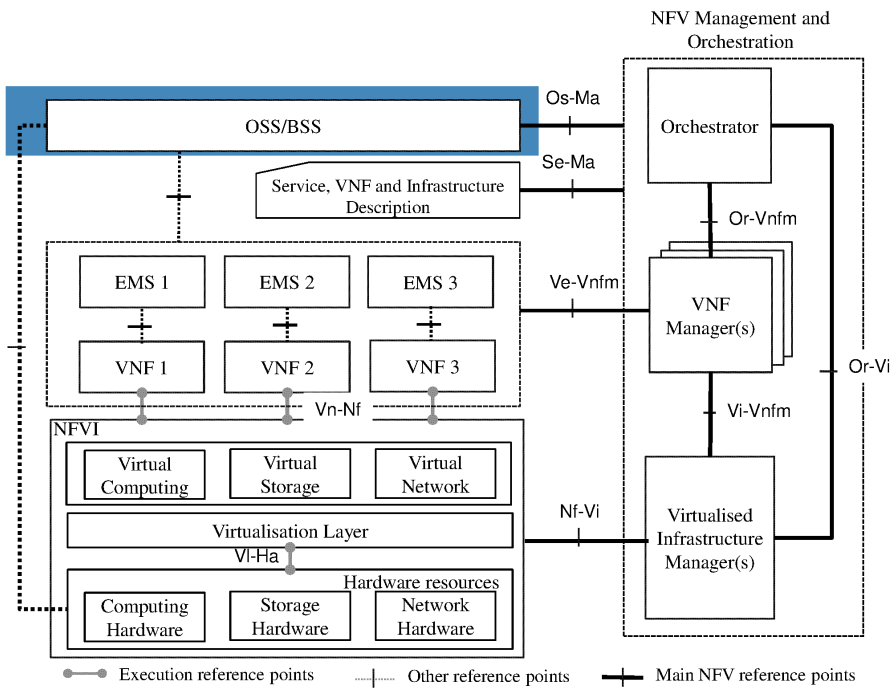


Figura 10: OSS/BSS

4 Proyectos y Plataformas NFV

En esta sección, se analizan algunos de los principales proyectos que se están desarrollando en el tema de NFV, ya sean estos por parte de la industria o de la comunidad académica. Existen varias plataformas creadas con este propósito, tanto comerciales como de código abierto. El código abierto es importante para la producción de implementaciones de referencias abiertas y la producción de estándares. A continuación se describirán algunos de los proyectos y plataformas de código abierto más relevantes, así como la función que cumplen en cada una de las capas de referencia de NFV.

4.1 OpenStack

OpenStack es una plataforma de cloud computing hecha con software libre para desplegar nubes públicas y privadas, desarrollada con la idea de ser sencilla de implementar, masivamente escalable y varias prestaciones. OpenStack proporciona una solución de Infraestructura como servicio (IaaS) a través de un conjunto de servicios interrelacionados. Esta plataforma ha optado por utilizar la licencia de software libre Apache 2.0, una licencia de software libre permisiva y no copyleft. OpenStack ha optado por un desarrollo completamente abierto, en el que se pueden aceptar opiniones y contribuciones de su comunidad o de fuera de ella ².

OpenStack tiene una gran flexibilidad y es capaz de implementar un cloud privado en una pequeña empresa o algunos de los mayores clouds públicos del mundo. Permite a los usuarios desplegar máquinas virtuales y otras instancias que manejan diferentes tareas con el fin de agilizar la gestión de un entorno de nube. Además ayuda a realizar un escalado horizontal fácil gracias a los componentes autónomos que utiliza, quiere decir que las tareas que corren al mismo tiempo pueden servir fácilmente a más o menos usuarios sobre la marcha. También gestiona recursos computacionales, de almacenamiento y de red a través de un datacenter, administrado por medio de una API OpenStack ³. Y lo más importante, OpenStack es un software de código abierto, lo que significa que todo aquel que desee puede acceder al código fuente, y será capaz de realizar cualquier cambio o modificación que crea necesaria, y de esta forma compartir libremente estos cambios de nuevo a la comunidad en general ⁴.

²<https://www.openstack.org/>

³OpenStack Cloud Software: <https://www.openstack.org/>

⁴<https://www.openstack.org/>

4.1.1 Componentes

OpenStack no es un solo producto, sino un conjunto de aplicaciones que pueden combinarse en función de las características y necesidades de cada caso, hay algunos componentes fundamentales y otros opcionales. Cada componente de OpenStack es totalmente autónomo y funcional y utiliza el protocolo AMQP de gestión de colas para comunicarse con el resto de componentes y una API web RESTful para comunicarse con procesos externos o los usuarios.

OpenStack permite la implementación de una Infraestructure as a Service (IaaS) a través de múltiples servicios que, de manera coordinada, cumplen diferentes propósitos para lograr el correcto funcionamiento de dicha infraestructura [8]. Cada servicio ofrece una interfaz de programación de aplicación (API) que facilita la integración de las mismas. Dependiendo de las necesidades de cada usuario, puede instalar algunos o todos los servicios disponibles. Estos servicios son: [12]:

- **Dashboard - Horizon:** Proporciona una interfaz gráfica web basada en los servicios subyacentes de OpenStack permitiendo el lanzamiento de una instancia, la asignación de direcciones IP y la configuración de los controles de acceso, etc.
- **Compute - Nova:** Crea y finaliza las instancias de VMs a través de la API del hipervisor utilizado. Dispone de una base de datos que almacena la mayoría de los datos de estado de compilación y ejecución de estas máquinas virtuales.
- **Networking - Neutron:** Acepta pedidos de la API y los enruta hacia los plugins de Neutron que correspondan. Los plugins y agentes de Neutron son los encargados de realizar las tareas, como enchufar y desenchufar puertos, crear redes y subredes y direccionamiento de IPs. También dispone de una cola para enrutar información entre el neutron-server y los distintos agentes [15].
- **Object Storage - Swift:** Almacena y recupera los objetos de datos no estructurados de su elección mediante un REST, es una API basada en HTTP [12]. El servidor proxy se encarga de aceptar los pedidos entrantes, provee archivos y un listado de los contenedores. El servidor de cuentas maneja las cuentas asociadas con el servicio. El servidor de contenedores realiza un mapeo de los contenedores, carpetas, dentro del servicio. El

servidor de objetos administra los objetos, archivos. También se corren servicios de replicación, para proveer consistencia y disponibilidad a través del cluster, auditoría y actualización [15].

- **Block Storage - Cinder:** Proporciona almacenamiento en bloques para las instancias en ejecución. Soporta dispositivos plug and play y permite gestionarlos. Cinder-volume reacciona leyendo o escribiendo a la base de datos que mantiene el estado, interactúa con otros procesos (como el cinder-scheduler) a través de la cola de mensajes y directamente actúa sobre el almacenamiento proveyendo hardware o software. Cinder-scheduler selecciona el nodo para el almacenamiento por bloques óptimo para la creación de un volumen sobre él. La cola de mensajes enruta información entre los procesos de Cinder. Una base de datos almacena el estado de los volúmenes.
- **Identity Service - Keystone:** Proporciona un servicio de autenticación y autorización para otros servicios de OpenStack y un catálogo de endpoints para estos mismos servicios. keystone se encarga tanto de manejar los pedidos de la API así como también provee un único punto de integración para las políticas de OpenStack, catálogos de los servicios, token y autenticación.
- **Image Service - Glance:** Permite almacenar y recuperar imágenes de disco de máquinas virtuales. Glance-api acepta los pedidos para la búsqueda, obtención y almacenamiento de imágenes. El registro de almacenamiento glance-registry procesa y recupera los metadatos de las imágenes. Posee una base de datos para los metadatos de las imágenes. También se corren servicios de replicación, para proveer consistencia y disponibilidad a través del cluster, auditoría y actualización.
- **Telemetry - Ceilometer:** Monitorea la nube de OpenStack para la evaluación comparativa y escalabilidad con fines estadísticos.
- **Orchestration - Heat:** Organiza múltiples aplicaciones en la nube, utilizando formatos como HOT nativo o formato de la plantilla AWS.

Tal y como se explica en el documento de análisis de despliegue de un IaaS [12], la siguiente imagen ilustra un diagrama conceptual sobre los roles de cada componente de OpenStack y como se relacionan, tal y como muestra la Fig. 11.

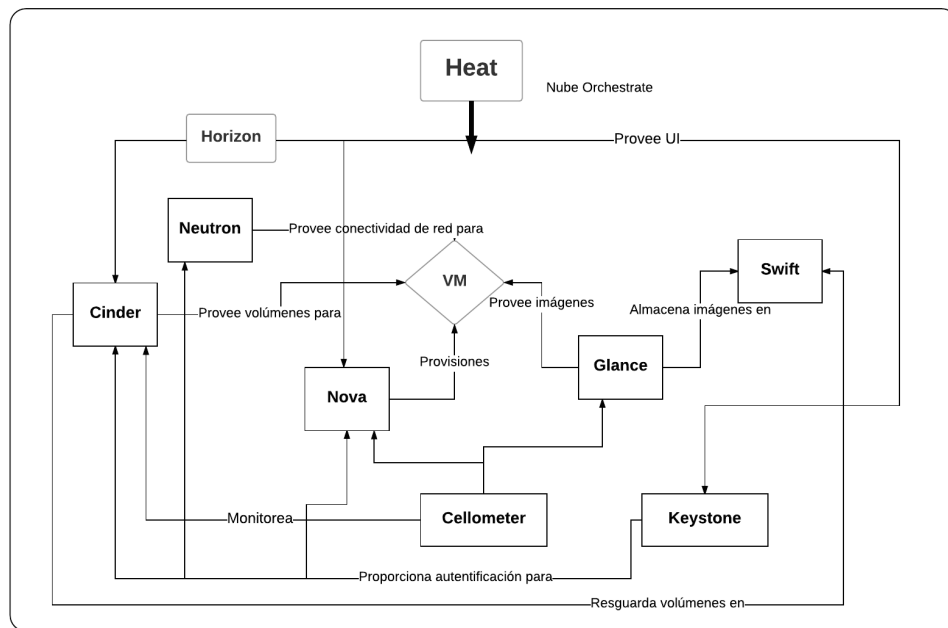


Figura 11: Arquitectura Conceptual de OpenStack

4.2 OPNFV

OPNFV es una nueva iniciativa de la comunidad Open Source lanzada en septiembre del 2014, misma que pretende impulsar la evolución de NFV a través de una plataforma de referencia de acuerdo con el Marco Arquitectónico ETSI NFV ISG [17]. El proyecto fue creado como un framework de código abierto desarrollado en la comunidad y apoyado por la industria de telecomunicaciones ⁵. OPNFV estableció una plataforma integral, de código abierto a nivel de operadores y proveedores de servicios, para avanzar en la evolución de NFV y garantizar la coherencia, rendimiento e interoperabilidad entre varios componentes del código. La comunidad también señaló que debido a que existen múltiples bloques NFV de código abierto ya construidos, entonces OPNFV trabajará con proyectos de exploración, para coordinar su integración y pruebas, mientras se llenan vacíos de desarrollo.

4.2.1 Objetivos de OPNFV

Los objetivos generales de OPNFV incluyen:

⁵<http://opnfv.org>

- Desarrollar una plataforma de código abierto integrada y probada que puede usarse para investigar y demostrar la funcionalidad básica de NFV.
- Asegurar la participación proactiva de los principales usuarios finales para validar que OPNFV lanza las necesidades de los operadores participantes.
- Influir y contribuir a los proyectos de código abierto relevantes que serán adoptados en la plataforma de referencia OPNFV.
- Establecer un ecosistema abierto para las soluciones NFV basadas en estándares abiertos y software de código abierto.
- Promover el OPNFV como la plataforma de referencia abierta preferida para evitar la innecesaria y costosa duplicación de esfuerzos [18] [26].
- Acelerar el tiempo de comercialización de las soluciones de NFV, asegurando que la plataforma satisfaga las necesidades de la industria y que permita a los usuarios finales elegir componentes tecnológicos específicos basados en sus casos de uso y arquitecturas de implementación.

OPNFV facilita el desarrollo y la evolución de los componentes NFV a través de diversos ecosistemas de código abierto y de la integración, despliegue y pruebas a nivel de sistema. Como un proyecto de prueba e integración, OPNFV reúne componentes de upstream a través de computación, almacenamiento y virtualización de red para crear una plataforma de extremo a extremo. (Véase Fig. 12) [18]. Las actividades dentro del OPNFV se centran en la integración de componentes y en la creación y despliegue automatizados del entorno integrado. La integración continua y las pruebas automatizadas de la plataforma para casos clave de uso del NFV son fundamentales para asegurar que la plataforma satisfaga las necesidades de la industria NFV. Otro enfoque clave es refinar los nuevos requisitos y características necesarias para la plataforma y trabajar estrechamente con las comunidades de upstream para incorporar estas capacidades dentro de sus bases de código ⁶.

El alcance inicial del OPNFV es proporcionar una Infraestructura NFV (NFVI), Gestión de Infraestructura Virtualizada (VIM) y un API a otros elementos del NFV, que juntos forman la infraestructura básica necesaria para los componentes de Funciones de Red Virtualizadas (VNFs) y Gestión y Orquestación de Redes (MANO) Dentro de los lineamientos de ETSI se puede ubicar a OPNFV en la arquitectura de NFV, tal como se muestra en la Fig. 13.

⁶<http://opnfv.org>

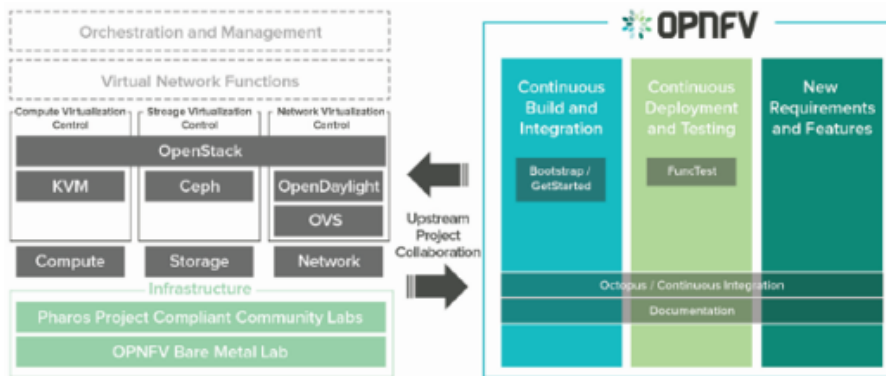


Figura 12: Diagrama OPNFV

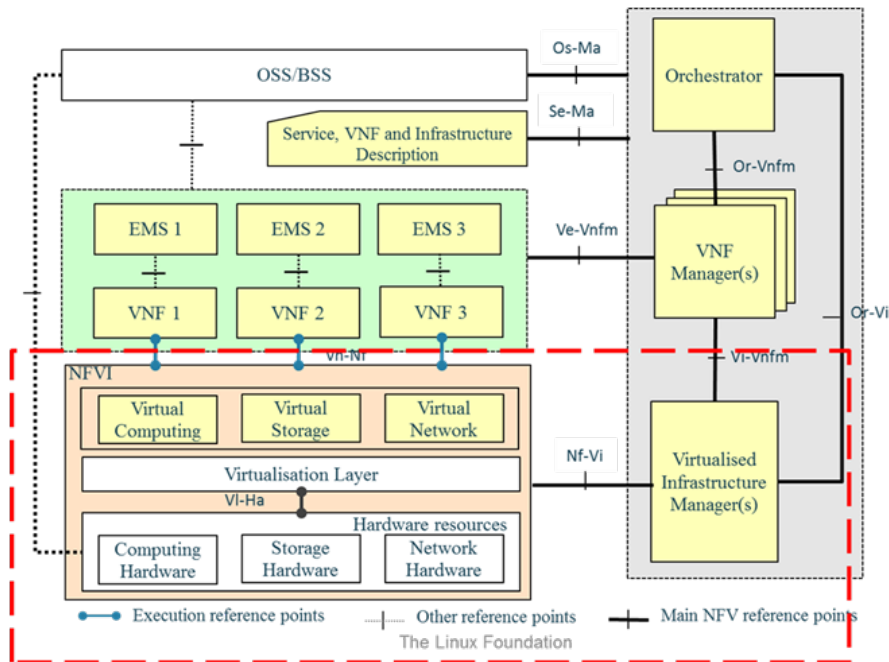


Figura 13: Arquitectura OPNFV ETSI

4.3 CloudNFV

CloudNFV provee una arquitectura abierta para la gestión y despliegue de NFV en un entorno en la nube utilizando estándares abiertos y con la colaboración de múltiples proveedores. Está basada en tecnologías de Cloud Computing y Software Defined Networking (SDN) en un entorno multi-vendedor. CloudNFV es una iniciativa abierta que combina las tres revoluciones de la era: cloud computing, redes definidas por software y virtualización de funciones de red ⁷.

⁷<http://www.cloudnfv.com/>

El grupo oficial de CloudNFV está formada por las empresas 6Wind , CIMI Corp., Dell Inc. , Enterprise Web, Overture Networks y Qosmos en 2013. Otras compañías, incluyendo MetaSwitch , se unieron posteriormente . El objetivo fue basarse en el trabajo de estándares de la ETSI (ISG) en NFV y añadir más capacidades de gestión y orquestación. Los miembros del grupo argumentaron que NFV necesitaba un modelo operativo y de políticas más robusto que fuera adecuado para los proveedores de servicios.

4.3.1 Arquitectura

CloudNFV tiene tres capas básicas: Funciones virtuales, administración y orquestación, y recursos de infraestructura, tal y como muestra la Fig. 14. Para el despliegue de este proyecto se establece un Framework de interfaz general basado en OpenStack, pero para integrarse con los procesos de gestión es necesario integrarlo con recursos activos y Modelos de datos. Esto se hace usando lo que CloudNFV llama "Visualizadores" que resultan ser pequeños objetos de software que proporcionan reglas de información derivación y presentación [27].

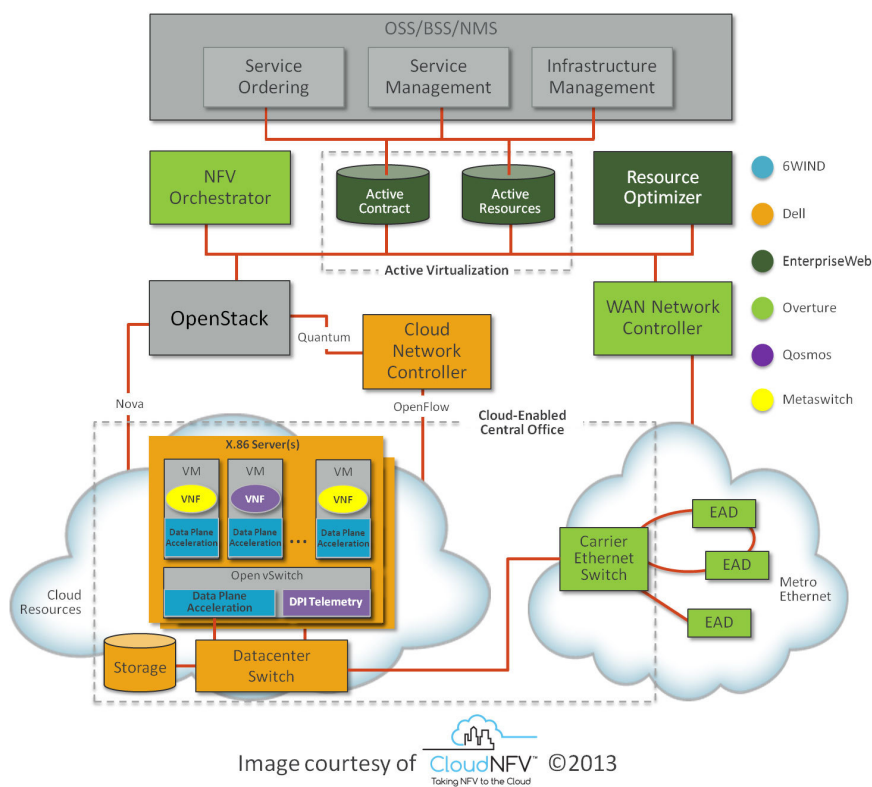


Figura 14: Arquitectura CloudNFV

4.3.2 Especificaciones

Según la especificación del ETSI ISG es necesario definir Interfaces basadas en un visualizador SDOs para poder establecer procesos enlazados a este. Si aún no se han definido, o si necesita algo diferente, se tendrá que desarrollar su propio Visualizador para realizar una conexión a su modelo de datos. Es decir cualquier proyecto de integración requerirá que desarrollar al menos un visualizador.

Cloud NFV sigue el marco del ISG para NFV y define tres capas; Servicios/aplicaciones (funciones de red), Orquestación y gestión (virtualización), e Infraestructura, sobre los cuáles también se basa este proyecto. Para desplegar funciones de red se necesita primero probarlas con OpenStack y una vez comprobada que este funcionando se debe establecer una red Neutron para la conexión entre los componentes.

Según la especificación del White Paper de CloudNFV es indispensable desarrollar este visualizador que se mencionaba antes para poder integrar los servicios por lo que requiere trabajo adicional en esta parte. CloudNFV también cuenta con plantillas propias y ejercicios que se pueden desplegar en su plataforma y realizar PoCs.

4.4 CloudBand

CloudBand, es un proyecto impulsado por Nokia que se encarga de los servicios de Gestión y Orquestación como se especifica en ETSI NFV MANO. La comunidad CloudBand ha estado contribuyendo a la adopción de NFV desde 2013 mediante la implementación de la prueba de conceptos (PoCs) que involucra a sus 60 miembros. En la actualidad, se ha ampliado el programa para promover el despliegue de NFV en un entorno de producción, lo que requiere pruebas más robustas y exhaustivas e integraciones. CloudBand ofrece a los proveedores de Funciones de Red virtualizadas (VNF) un marco riguroso y sistemático para probar y validar sus VNFs a través de la plataforma CloudBand NFV.

Con CloudBand, se puede implementar una plataforma NFV que permite utilizar la nube para trabajar de forma más rápida y rentable. Facilita la compatibilidad entre diferentes soluciones NFV. También utiliza APIs OpenStack estándar, donde el software es independiente del hardware. CloudBand facilita

el hospedaje, orquestación, automatización y administración de funciones de red virtualizadas (VNF) y servicios. Ayudan a reducir el tiempo de ingresos para los nuevos servicios ya que utilizan la automatización y la optimización para mejorar las operaciones de red. CloudBand está disponible como una solución integrada MANO de extremo a extremo o como productos individuales que cubren funciones bien definidas en un despliegue de varios proveedores.

4.4.1 Componentes

Según el whitepaper de CloudBand ⁸ este cuenta con tres componentes principales los que se describen a continuación (Véase Fig. 15) :

CloudBand Infrastructure Software

CloudBand Infrastructure Software, es la capa de infraestructura de NFV (NF-VI) y administrador de infraestructura (VIM). Virtualiza y administra recursos de computación, almacenamiento y red. Permite que los VNFs se ejecuten y garantice que cumplen con rigurosos requisitos de robustez, rendimiento y seguridad. Está construido para la excelencia operacional, con un rendimiento mejorado y capacidades en tiempo real, integradas y probado en una variedad de opciones de Hardware.

CloudBand Administrador de aplicaciones

CloudBand Application Manager es un gestor VNF (VNFM) que automatiza las acciones de gestión del ciclo de vida mediante la gestión de recursos y la aplicación de flujos de trabajo asociados, incluyendo la instanciación y despliegue de VNF, escalado, supervisión y más.

CloudBand Network Director

CloudBand Network Director es un orquestador de recursos y servicios de red NFV. Gestiona los recursos virtuales a través de los nodos de infraestructura NFV que están geo-distribuidos. Visualiza y automatiza el ciclo de vida de

⁸<https://resources.ext.nokia.com/>

los servicios de red, como CPE virtual, incluyendo sus gráficos de reenvío y cadenas de servicios.

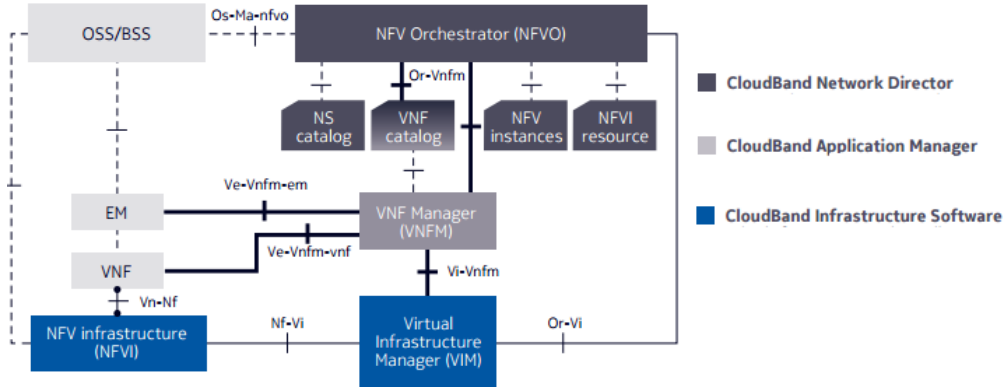


Figura 15: Arquitectura CloudBand

4.5 Open Source MANO OSM

Open Source Mano (OSM) es un proyecto creado por la ETSI para desarrollar una pila de software Open Source NFV que se encargue de la Gestión y Orquestación (MANO) alineada con ETSI NFV. Este proyecto es desarrollado por una comunidad impulsada por la industria de las Telecomunicaciones con el fin de satisfacer los requisitos de orquestación de las redes NFV con soporte a los modelos de información. Los miembros fundadores de OSM incluyen 25 participantes entre los cuales están A10 Networks, Amdocs Ltd., Canonical USA Inc., Cisco Systems, EANTC, Ensemble (ADVA Optical Networking), Ericsson LM, F5 Networks, Fortinet, Fraunhofer FOKUS, Hewlett-Packard Enterprise, Huawei Technologies Co. Ltd, Ixia, Keynetec Technologies, Mahindra Comviva, Mirantis Netrounds, Openet, Palo Alto Networks, Red Hat Limited, RIFT.io, Sandvine Inc., Sonus Networks, VMware, WindyCitySDR y Telefónica.⁹

El proyecto OSM cubre tanto la orquestación de recursos como la orquestación de servicios para permitir el despliegue automático y la interconexión de todos los componentes, tanto de los escenarios NFV como de la gestión de los ciclos de vida de los servicios de red [14]. (Véase Fig. 16).

⁹<https://osm.etsi.org/>

4.5.1 Objetivos

- Trabajar con la comunidad Open Source para poder entregar software de calidad OSM bajo una licencia pública Apache 2.0.
- Orquestar complejos casos de uso NFV usando modelos de información neutros de los proveedores de equipos
- Capturar todas las funcionalidades significativas de los servicios E2E así como los requisitos de los componentes individuales (Funciones Virtualizadas de red, VNFs) ¹⁰.
- Tener un código que sea capaz de orquestar escenarios NFV.

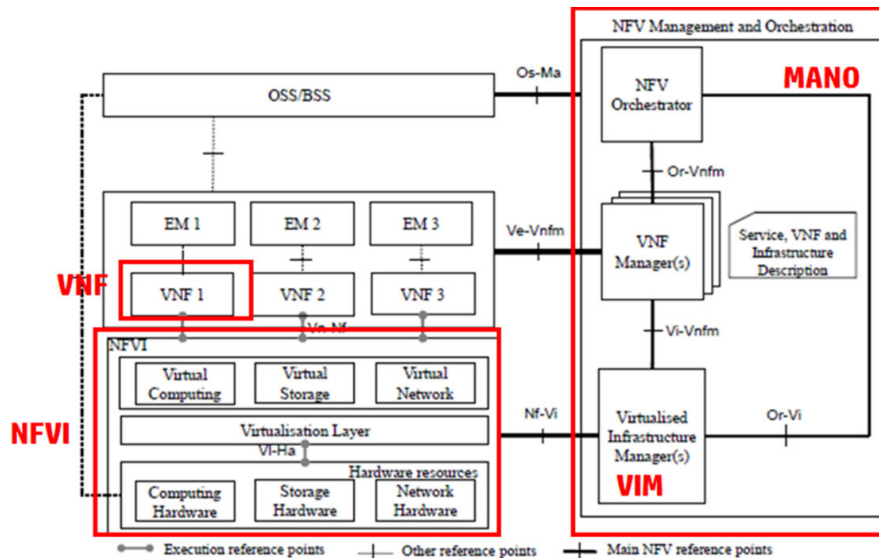


Figura 16: Arquitectura Open Source Mano - ETSI

Este proyecto se enfoca en evolucionar un modelo conjunto ETSI NFV-ISG a un modelo de datos implementable, el cual se convertirá en la base de código de OSM para la dirección de multi usos de NFV. Esto se ha venido trabajando con la mira de alcanzar un modelo de implementación uniforme que puedan ser desplegados por la comunidad de operadores al más bajo costo de integración [14].

4.5.2 Principios Arquitectónicos

- **Layering:** Exigir una delimitación clara entre las capas y los módulos.

¹⁰<http://catedratelefónica.ulpgc.es/open-source-mano>

- **Abstracción:** Subir y bajar las capas debe ofrecer una clara diferenciación en los niveles de abstracción y detalle presentados.
- **Modularidad:** Incluso dentro de las capas, la modularidad se habilita con un modelo de complemento preferido para facilitar el reemplazo de módulos a medida que se desarrolla la comunidad OSM.
- **Sencillez:** La solución debe tener la complejidad mínima necesaria para ser exitosa.

OSM ha lanzado hasta ahora tres versiones Release ZERO, Release ONE, y recientemente el 01 de Junio del 2017 el Release TWO. Este proyecto permite una rápida instalación en los entornos de proveedores, integradores de sistemas y operadores de VNF. OSM Release TWO, como sus predecesores pero con mejoras, alcanza interoperabilidad con otros componentes (VNFs, VIMs, controladores SDN) y proporciona un framework plug-in para el mantenimiento de la plataforma y extensiones [1]. La Fig. 17 muestra el diagrama aproximado del alcance entre los componentes OSM y el ETSI NFV MANO.

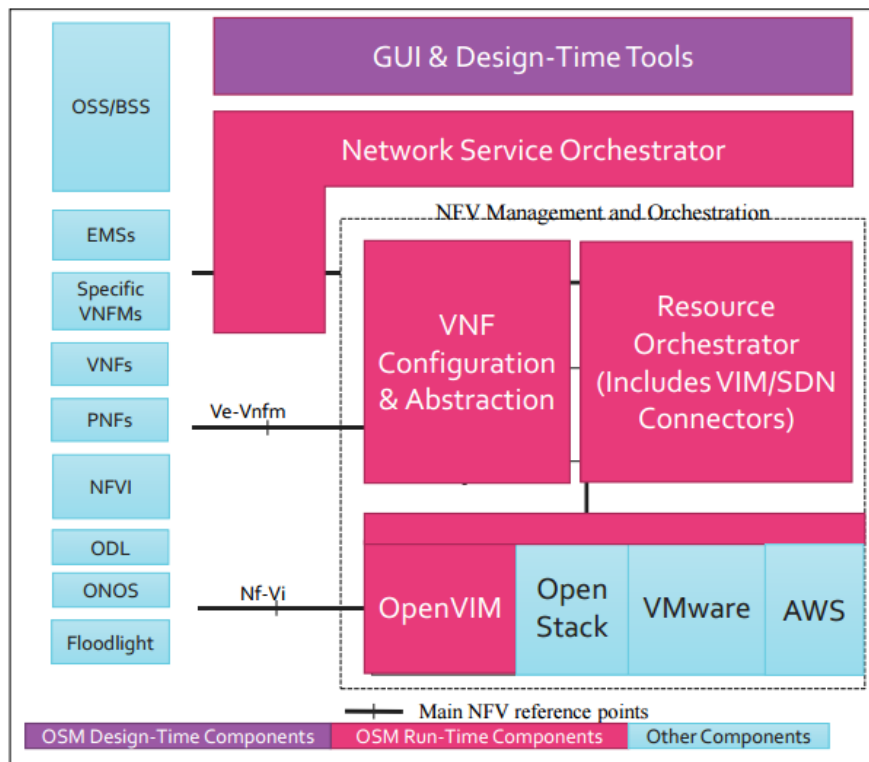


Figura 17: Arquitectura OSM ETSI

La arquitectura de OSM se define con tres componentes principales, VCA, RO, SO-ub. Una vez instalados, siguiendo la receta disponible online [3], ca-

da uno ocupa un contenedor y representan una capa distinta respetando las especificaciones de la arquitectura de referencia ETSI. Estas capas cumplen sus respectivas funciones y servicios, tal y como ilustra la Fig.18. Todos estos tienen conexión a través de un bridge denominado lxdbr0, que representaría un switch de conexión entre los tres componentes, los que se describen a continuación:

RO - Orquestador de Recursos: Este servicio lo desarrolla Telefónica, ésta capa es la responsable de procesar los requerimientos de localización de las instancias VNF y direccionar al VIM a una ubicación apropiada de computación, red y almacenamiento, para desplegar VNFs con sus interconexiones. RO conoce el nivel de uso, estado, estadísticas y resto de parámetros asociados a cada uno de los elementos. Los componentes SO y RO se pueden correlacionar conjuntamente con la entidad NFVO en la arquitectura ETSI MANO.

VCA - Configuración y Abstracción de VNF: Este servicio lo provee Canonical (Juju). Realiza la configuración inicial de VNF utilizando las funcionalidades de Juju. Teniendo en cuenta este propósito, el módulo VCA puede considerarse como un VNFM genérico con un conjunto de características limitado. Esta capa ofrece la gestión de la funcionalidad de VNF y es la responsable de los atributos de configuración de procesamiento.

SO-ub Orquestador de Servicios de Red: Esta capa está encargada de las pruebas de interoperabilidad, Interfaz de Usuario (UI) y Orquestación de Servicios (NSO), todo lo brinda RIFT.io. El SO almacena las definiciones de VNF y los catálogos NS, gestiona el flujo de trabajo de la implementación del servicio y puede consultar el estado de los servicios ya implementados. OSM integra el motor de orquestación rift.io como SO. Es este mismo el que ofrece la funcionalidad de interfaz gráfica y visualización al usuario, en donde se puede observar el estado del entorno de orquestación. Este provee una simple y rápida forma para interactuar con NSO, mediante herramientas de fácil despliegue, creado y borrado de entornos VNFs.

Para la integración con un NFVI Open Source Mano Release TWO tiene actualmente disponibles las opciones de OpenStack y Amazon AWS. Cabe destacar que se han realizado ya varios pruebas de concepto (PoC), con Open Source Mano, como encadenamiento de servicios para el desarrollo de 5G.

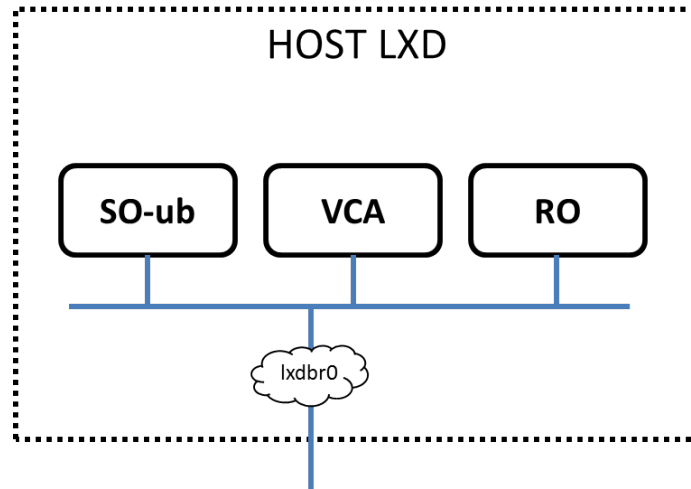


Figura 18: Componentes principales OSM

4.6 Cloud4NFV

La plataforma Cloud4NFV se basa en Cloud, SDN y tecnologías WAN para permitir que los Service Functions (SFs) se gestionen en una base de datos como un Service Basis. La plataforma está dirigida a T elcos para mejorar la gesti on de SFs dentro de su entorno, pero tambi en puede usarse para construir nuevos servicios basados en el concepto de Function as-a-Service (SFaaS), en los que se pueden ofrecer SFs o paquetes que contengan una combinaci on de SFs como un servicio agregado al cliente.

4.6.1 Caracter sticas

Las funcionalidades m as relevantes de Cloud4NFV son: Automatiza la gesti on de la infraestructura de NFV y plataformas SDN Configura y gestiona el ciclo de vida de servicios de red con los dominios web de los clientes. Administraci on y optimizaci on de los recursos de la WAN y de la nube para acomodar SFs Soporte a la composici on SF a trav es de SFC.

Todas las funcionalidades antes mencionadas son esenciales en el  mbito de una plataforma NFV, sin embargo, se destacan las dos  ltimas debido a su novedad. Estas dos funcionalidades se ven como factores de diferenciaci on clave de otras soluciones disponibles, tomando esta plataforma m as cerca de ser plenamente compatible con el grado de portador. La administraci on federada y la optimizaci on de recursos de WAN y cloud, le da a la plataforma un amplio y distribuido alcance. Este permite el establecimiento de servicios de

extremo a extremo sobre una infraestructura física distribuida. La capacidad de realizar SFC da a la plataforma gran flexibilidad con respecto a la Gestión y composición de SF, permitiendo la definición y el establecimiento de servicios avanzados de una manera mucho más eficiente y flexible [24].

4.6.2 Arquitectura

La arquitectura de Cloud4NFV es compatible con la ETSI NFV, y por razones de simplicidad Las diferentes entidades se nombran según la nomenclatura que especifica la ETSI. Las cuatro entidades principales de la plataforma son (véase Fig. 19): el Orquestador; VNF Manager(s); VIM(s); y el plano de Servicio, VNF e Infraestructura. El Orquestador es una entidad única, mientras que habrá múltiples VNF Manager (s), uno por NF o por NF compuesto [24].

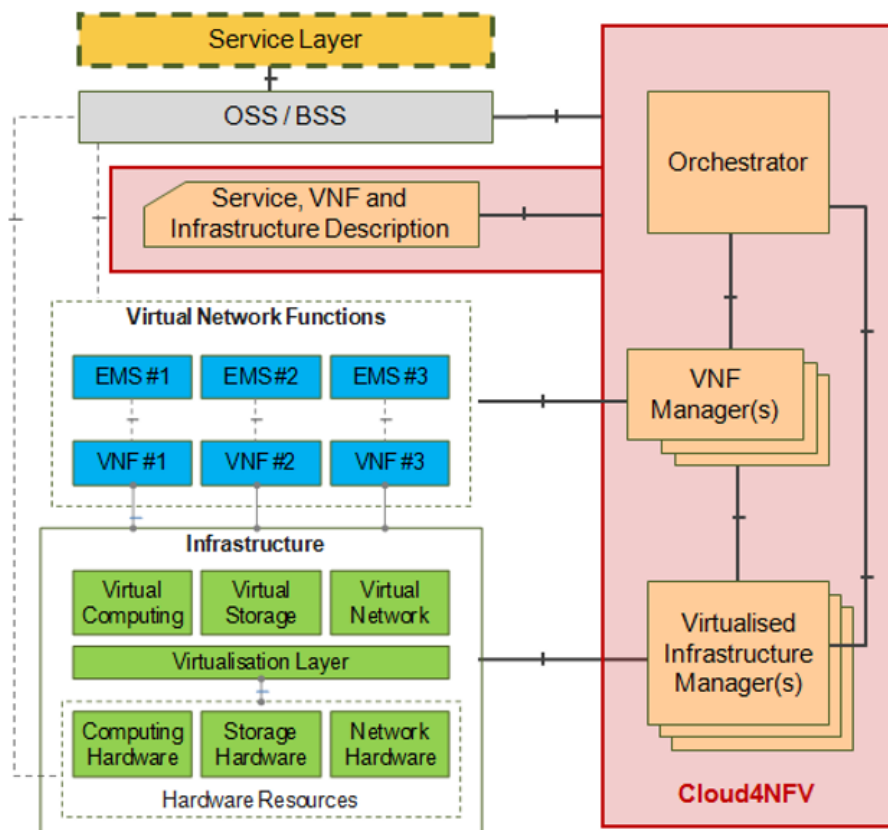


Figura 19: Arquitectura Cloud4NFV

Orquestador

El orquestador es responsable de la provisión, administración y monitoreo automatizados de SFs sobre la infraestructura virtual. Expone la capacidad de crear y eliminar SFs, así como la capacidad de encadenar SFs. Se basa en el Plano de VIM para proporcionar los recursos de infraestructura donde se ejecutan SF (VM, redes virtuales, etc). En cuanto al marco arquitectónico de referencia ETSI NFV , este componente las entidades Orquestador y VNF Manager. El orquestador tiene una interfaz REST (Representational State Transfer), que expone la posibilidad de crear eliminar y encadenar SFs.

Plano de Gestión de Infraestructura virtual

El plano VIM incluye los componentes para la gestión de los recursos de la infraestructura. Incluye DC (Domain Crontrrollers) de cloud (uno por DC) y un controlador WAN que es capaz de establecer servicios de interconexión inter-DC. El Plano VIM puede ser visto como el administrador de infraestructuras virtuales en el marco arquitectónico de referencia ETSI NFV. Sin embargo, la actual especificación ETSI no tiene en cuenta el componente WAN. Esto es considerado por ETSI como sujeto de futuros análisis.

4.7 Open-O

El proyecto OPEN-O se anunció formalmente en el 2016 en el Mobile World Congress (MWC) y planea lanzar lanzamientos futuros en un ciclo de 6 meses. Está alojado en la fundación Linux. Los gestores iniciales del proyecto eran en su mayoría de compañías asiáticas, como Huawei, ZTE y China Mobile. Eventualmente, el proyecto recibió más apoyo de Brocade, Ericsson, GigaSpaces, Intel y otros. El principal objetivo del proyecto es agilizar el servicio de extremo a extremo en múltiples dominios utilizando la plataforma unificada para orquestación NFV y SDN. La Gestión principal de Open-O es dirigida por el Consejo del proyecto ¹¹.

Aunque el proyecto OPEN-O no tiene ningún objetivo de ser compatible con NFV MANO, el componente NFVO de OPEN-O está alineado con un modelo de referencia ETSI, y todos los elementos clave de MANO, tales como VNFM y VIM aquellos que se pueden encontrar en un arquitectura NFVO .

¹¹<https://www.open-o.org/>

Además, el alcance del proyecto OPEN-O va más allá de la simple orquestación del NFV, y más allá del alcance especificado por la arquitectura de referencia ETSI NFV MANO. Una pieza importante de este proyecto se relaciona con la provisión de servicios de red basados en SDN y la orquestación, los cuales podrían ser utilizados en combinación con los servicios NFV o como una característica independiente.

4.7.1 Arquitectura

Desde un punto de vista general, la arquitectura completa del software OPEN-O puede dividirse en 5 grupos de componentes, tal y como ilustra la Fig. 20 [19]:

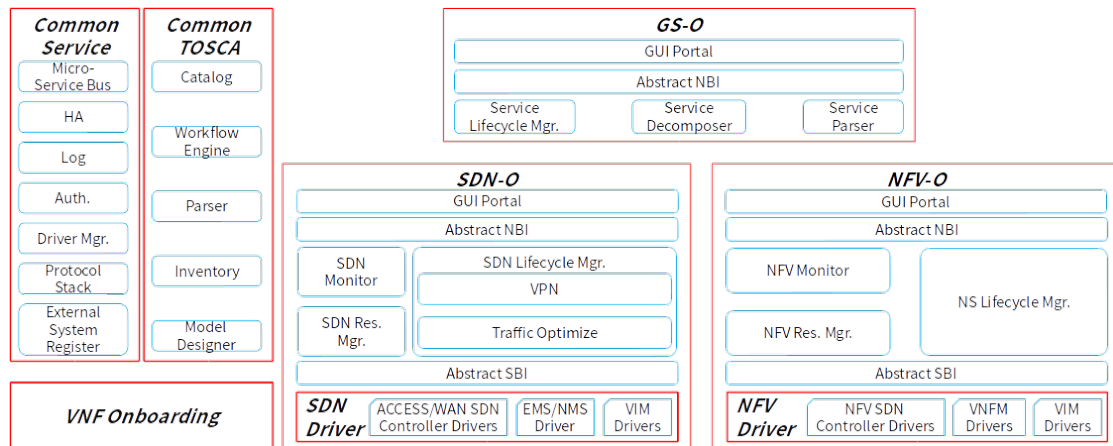


Figura 20: Arquitectura OPEN-O

Common Service : Consiste en servicios compartidos utilizados por todos los demás componentes.

Common TOSCA: Proporciona características relacionadas con TOSCA tales como gestión de catálogos NSD, análisis de definición de NSD, ejecución de flujo de trabajo, etc. Este componente se basa en el proyecto ARIA TOSCA.

Global Service Orchestrator (GSO): Como su nombre indica, este grupo proporciona una gestión general del ciclo de vida del servicio extremo a extremo.

Orquestador SDN (SDN-O): Proporciona abstracción y gestión del ciclo de vida de los servicios SDN; Una parte esencial de este bloque son los

controladores SDN, que proporcionan módulos específicos del dispositivo para la comunicación con un dispositivo particular o un controlador SDN.

Orquestador NFV (NFV-O): Este grupo proporciona la instanciación de los servicios NFV y la gestión del ciclo de vida.

El proyecto OPEN-O utiliza una arquitectura basada en microservicios. El elemento central de la plataforma es el Microservice Bus, que es el microservicio principal del grupo de componentes de Common Service. Cada componente de la plataforma debe registrarse con este bus. Durante el registro, cada microservicio especifica las API expuestas y las direcciones de punto final. Como resultado, la arquitectura general del software es flexible y se puede ampliar fácilmente con módulos adicionales. OPEN-O Rel. 1 consta microservicios basados en Java y python.

El Release 1 del proyecto OPEN-O admite sólo OpenStack como VIM. Este soporte está disponible como un controlador basado en Java para el componente NFVO. Para versiones posteriores, está previsto el soporte para VMware como VIM. Esta plataforma tiene un VNFM genérico que se basa en JuJu Charms. Además, la arquitectura acoplable de la plataforma OPEN-O puede soportar cualquier VNFM específico del proveedor, pero se necesitarán esfuerzos adicionales de desarrollo e integración.

4.8 Cloudify

Cloudscape de Gigaspaces es la plataforma de software de código abierto de orquestación de nube basada en TOSCA, la plataforma se amplió aún más para incluir casos de uso relacionados con NFV, y surgió Cloudify Telecom Edition. Teniendo en cuenta el propósito original de la plataforma, Cloudify tiene una arquitectura extensible y puede interactuar con múltiples proveedores de IaaS / PaaS, como AWS, OpenStack, Microsoft Azure, etc. En general, el software Cloudify es de código abierto bajo la licencia Apache 2 y el código fuente está alojado en un repositorio público. La planificación en general del proyecto se define por Gigaspaces ¹².

El enfoque original de la plataforma Cloudify es la orquestación del despliegue de aplicaciones en la nube. Esta plataforma combina componentes NFVO genéricos y componentes VNFM de las entidades definidas por MANO. Se po-

¹²<http://docs.getcloudify.org/4.0.0/intro/what-is-cloudify/>

dría decir que Cloudify está alineado con la arquitectura de referencia MANO pero no es totalmente compatible.

4.8.1 Arquitectura

Desde una arquitectura de alto nivel se pueden definir los siguientes componentes para Cloudify, véase Fig. 21 ¹³.

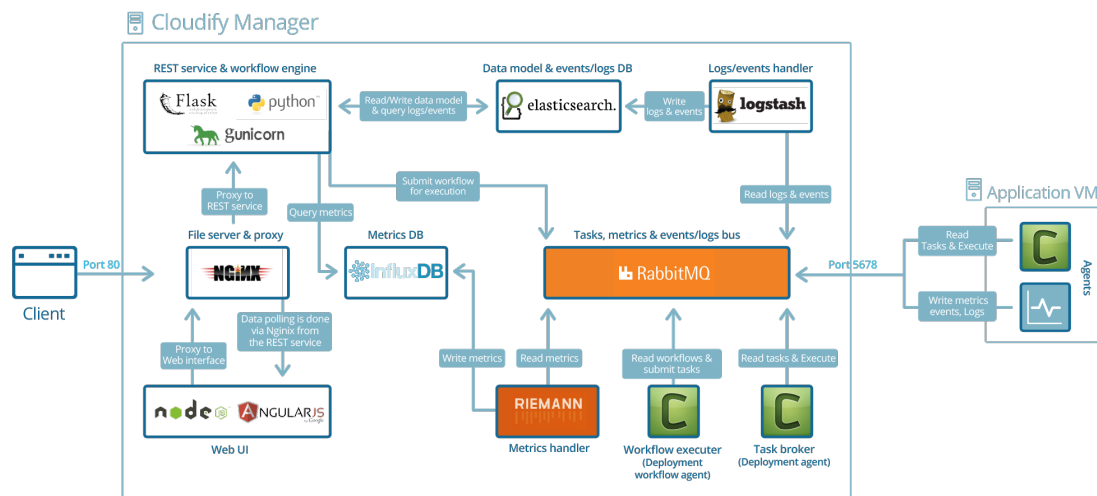


Figura 21: Arquitectura Cloudify

Cloudify Manager: Es el orquestador que realiza el despliegue y la gestión del ciclo de vida de las aplicaciones o NSDs descritos en las plantillas.

Agentes Cloudify: Se utilizan para administrar la ejecución del flujo de trabajo a través de un complemento adecuado. Para proporcionar una gestión general del ciclo de vida, Cloudify integra componentes de terceros como:

- Elasticsearch, utilizado como un almacén de datos del estado de despliegue, incluye datos de tiempo de ejecución y registros de datos procedentes de varios componentes de la plataforma.
- Logstash, utilizado para procesar información de registro procedente de componentes y agentes de plataforma.
- Riemann, utilizado como un motor de políticas para procesar las decisiones de tiempo de ejecución sobre disponibilidad, SLA y monitoreo general.

¹³<http://docs.getcloudify.org/4.0.0/intro/cloudify-manager/>

- RabbitMQ, utilizado como transporte asíncrono para la comunicación entre todos los componentes de la plataforma, incluidos los agentes remotos.

La funcionalidad de orquestación en sí misma es proporcionada por el proyecto ARIA TOSCA, que define el formato de plano de TOSCA y el motor de flujo de trabajo de implementación. Los componentes y complementos "nativos" de Cloudify son aplicaciones python.

Cuando Cloudify fue adaptado para el caso de uso de NFV, heredó plugins para OpenStack, VMware, Azure y otros que ya estaban disponibles para implementaciones de nube de propósito general. Así que se puede decir que Cloudify tiene soporte MultiVIM y cualquier soporte VIM arbitrario puede ser agregado a través del plugin adecuado. Siguiendo el modelo de referencia de Gigaspaces para NFV, existe un VNFM genérico que se puede utilizar con un orquestador Cloudify NFV fuera de la caja". Adicional el proveedor VNFM específico puede estar integrado, pero se requiere desarrollo de complemento adecuado.

5 Implementación del Escenario NFV

La finalidad de este trabajo es la implementación de un escenario que despliegue el modelo de una red NFV siguiendo los lineamientos de la arquitectura planteada por la ETSI enfocándose en el componente para la Gestión y Orquestación de NFV MANO. Para ello es necesario escoger de entre todos los proyectos analizados anteriormente el que mejor cumpla con el objetivo final.

Como primer punto se debe tomar en cuenta que el escenario a desarrollar es una demostración didáctica que cumpla los fines pertinentes, mediante la experimentación con un escenario virtual de los componentes antes descritos, pues bien, esto no representará mayor inconveniente ya que como se había tratado con antelación una de las principales ventajas de NFV es poder minimizar las limitaciones de hardware y software para alcanzar mejores o iguales funcionalidades que una red desplegada de forma tradicional. Después de un análisis con las herramientas y proyectos descritos anteriormente se han escogido los que más se han ajustado al caso de uso propuesto. A continuación se describe el escenario implementado.

5.1 Elección de las Herramientas y Plataformas

El caso de uso elegido se va a enfocar en la implementación del componente MANO de la arquitectura NFV, que es el encargado de la orquestación y administración del ciclo de vida tanto de los recursos lógicos como los físicos. Siguiendo el marco de referencia de la ETSI, para poder desplegar estas funcionalidades este debe cumplir con tres bloques funcionales: NFV Orchestrator, VNF Manager y Virtualized Infrastructure Manager, en el cada uno realiza su función dentro del marco de referencia, tal y como se mostró en la Fig. 7.

5.1.1 Orquestador NFV

Continuando con lo anterior, para el rol de Orquestador se ha escogido el proyecto Open Source Mano (OSM) que se describe en la pag. 21, mismo que será responsable del despliegue de los servicios de red (NS) y VNFs. Las razones por las que se escogió este Orquestador son varias, después de haber analizado algunos de los Orquestadores Open Source disponibles; Open Source Mano destaca por su comunidad e intentos de la misma, de ser ampliamente

compatible con la arquitectura de referencia ETSI NFV MANO, respetando las especificaciones de este grupo de trabajo. Al momento existen varias PoC realizadas bajo este proyecto ¹⁴, lo que respalda la elección tomada.

Ahora bien, otro de los motivos de la elección de esta plataforma de Orquestación es que admite el despliegue de servicios NFV que abarca varios VIM. El proyecto OSM fue inicialmente considerado una plataforma multi-VIM, al momento de realizar este trabajo, OSM Release TWO soporta cuatro plataformas VIM: OpenStack, VMware vCloud Director, OpenVIM y recientemente ha implementado el soporte a Amazon Web Services (AWS). Además, OSM es una plataforma bastante intuitiva que contiene su propia Wiki que posee cuantiosa documentación. La plataforma OSM es adecuada para fines de evaluación y tiene una arquitectura relativamente sencilla. También contiene varios demos de NSD y VNFDs que están disponibles en gerrit report ¹⁵. Como resultado, la plataforma puede instalarse e integrarse fácilmente con el VIM apropiado para evaluar las capacidades básicas del NFV, casos de uso de prueba y PoCs.

Open Source Mano implementa en su orquestador la gestión de requerimientos que ayuda a asegurar que los recursos de la infraestructura NFV se abstraen de forma limpia, es decir independiente del VIM para soportar los servicios que acceden a estos.

5.1.2 VNF Manager

Open Source Mano también implementa un administrador VNF que supervisa el ciclo de vida y la gestión de instancias de las funciones de red virtualizadas (VNF). Como se explicó en secciones anteriores, se supone que cada VNF se asociará con un VNFM que gestionará el ciclo de vida particular de VNF. Open Source Mano puede administrar múltiples instancias del mismo tipo de VNF o de diferentes tipos.

5.1.3 Virtualized Infrastructure Manager (VIM)

Este componente ha recibido un gran enfoque y varias soluciones de código abierto. El propio proyecto Open Source Mano cuenta con su propio VIM,

¹⁴<https://osm.etsi.org/wikipub/index.php>

¹⁵<https://osm.etsi.org/gerrit>

OpenVim, pero al revisar la documentación actual está obsoleta en su página oficial. Ahora bien, la solución más óptima para acoplar como VIM para Open Source Mano resulta ser OpenStack. Según se había revisado en secciones anteriores OpenStack, pag. 27, es una de las soluciones open source más óptimas y robustas que se pueden encontrar y se puede utilizar para realizar la funcionalidad de gestión de infraestructura virtualizada de MANO siendo una de sus características más relevantes su simplicidad.

Otro de los motivos para escoger esta herramienta es que la arquitectura de Openstack es modular, pudiendo instalarlo todos ellos en un único host o por separado. Esto dependerá del tamaño y el objetivo de la cloud. Para entornos de pruebas lo normal es instalarlo todo en un único nodo. Cabe destacar que OpenStack provee interfaces estandarizadas entre elementos NFV e infraestructura.

OpenStack controla y administra los recursos de computación, almacenamiento y red de la NFVI. Este VIM ofrece una arquitectura conectable con API documentadas, servicios compartidos y operaciones. Automatización para VNFs y otras funciones de integración con NFV, automatización y despliegue de los elementos de red.

5.2 Arquitectura

Como se ilustra en la Fig. 22, la arquitectura para el caso de uso escogido implementaría la plataforma OSM en los dos primeros bloques de Orquestación y Gestión y en el tercer bloque OpenStack como VIM. A continuación se explica los requerimientos físicos y lógicos utilizados para esta implementación.

5.2.1 Open Source Mano

OSM necesita requisitos específicos para su despliegue sin necesarios: un servidor o máquina virtual Ubuntu 16.04 como imagen base, 4 CPUs, 8 GB de RAM, disco de 40 GB. El sistema Operativo anfitrión debe estar configurado para ejecutar contenedores LXD [22].

LXD es es un hypervisor para contenedores Linux, es un complemento para los contenedores de Linux (LXC) que facilita su uso y añade nuevas posibilidades. Este contenedor puede utilizar diferentes storage-backends, por

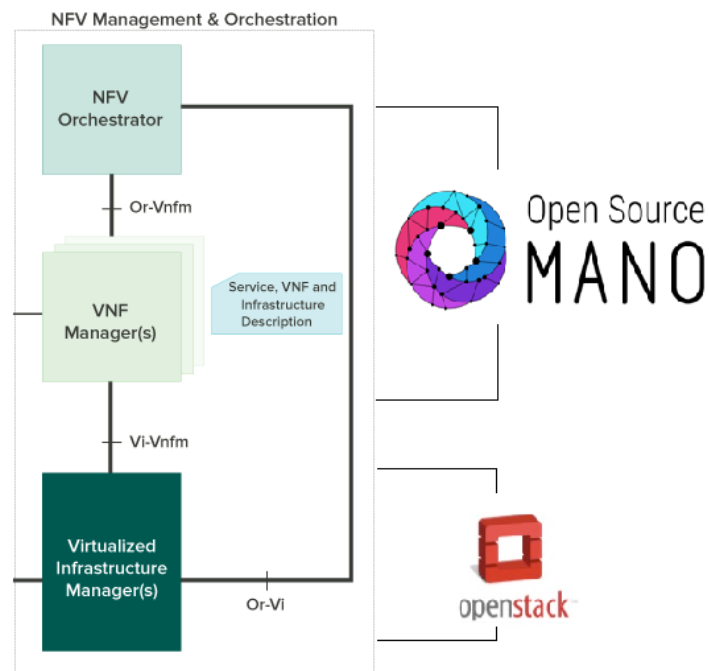


Figura 22: Arquitectura Implementación MANO

defecto se utiliza un directorio en el sistema de archivos pero combinar los contenedores con alguna herramienta que proporcione COW (como ZFS, Btrfs o LVM) permite mejorar la velocidad y economizar espacio al trabajar con snapshots o al copiar contenedores. Aunque para probar LXD bastará cualquier sistema de archivos ¹⁶.

Los contenedores LXD serán los que alojen los tres componentes de Open Source Mano, estos ya se han descrito en secciones anteriores, véase pag. 35, : RO, VCA y SO-ub. El esquema lógico que se siguió el escenario propuesto para esta implementación se muestra en el siguiente esquema de la Fig. 23.

Una vez los contenedores estén listos se puede ingresar a la interfaz de usuario de Open Source MANO, misma que está gestionada por el componente SO-ub que está encargado de las pruebas de interoperabilidad, interfaz de usuario y orquestación de servicios (NSO), este servicio lo provee RIFT.io. La interfaz gráfica de OSM provee una simple y rápida forma para interactuar con NSO, véase Fig. 24, mediante herramientas de fácil despliegue, creado y borrado de entornos VNFs.

¹⁶<https://linuxcontainers.org/lxd/>

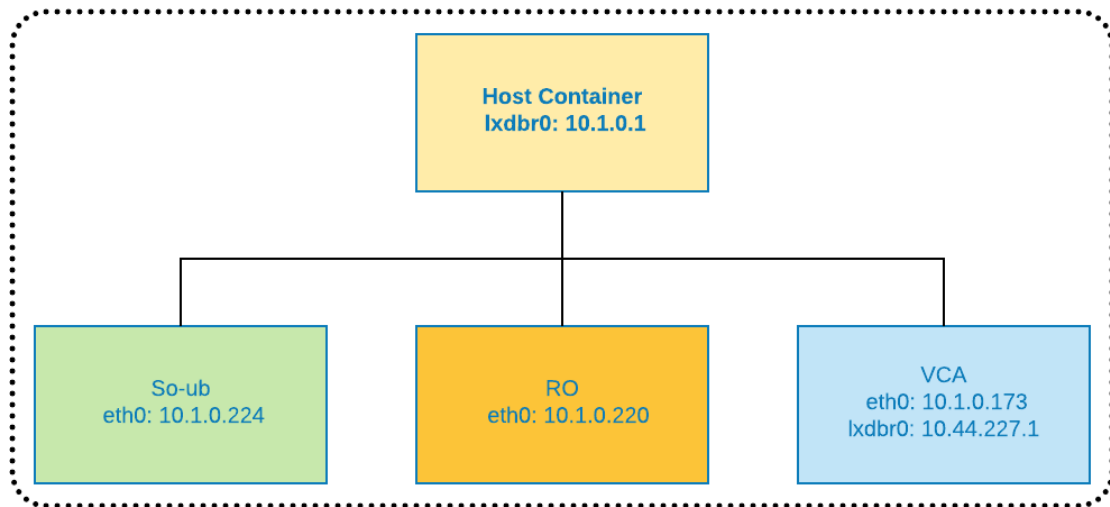


Figura 23: Escenario Open Source MANO

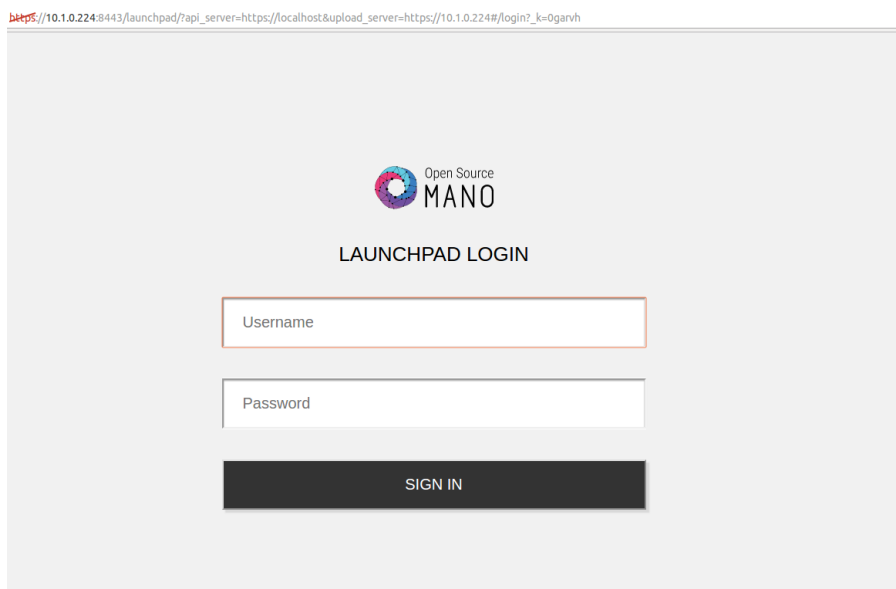


Figura 24: GUI Open Source MANO

5.2.2 OpenStack

Para la ejecución del escenario OpenStack utiliza la plataforma virtual VNX en su versión KVM. VNX es una herramienta de virtualización de código abierto de propósito general diseñada para ayudar a construir pruebas en redes virtuales de forma automática. Permite la definición y despliegue automático de escenarios de red de máquinas virtuales de diferentes tipos (Linux, Windows, FreeBSD, Olive o Dynamips, etc.) interconectados siguiendo una topología definida por el usuario, posiblemente conectada a redes externas [7].

El escenario virtual planteado esta formado por:

- Un nodo controlador, en el cual corren las principales aplicaciones de control y el interfaz de usuario (Dashboard).
- Dos nodos de computación, que dan soporte a la creación de máquinas virtuales utilizando el hipervisor KVM de Linux.
- Un nodo de red, que se ocupa de proporcionar los servicios de red necesarios (routers, NAT, etc.) para que las máquinas virtuales tengan acceso al exterior. Los nodos poseen varios interfaces de red (dos el controlador y los nodos de computación y cuatro el nodo de red) mediante los cuales se conectan a cuatro redes distintas:
- Red de gestión (10.0.0.0/24), que es una red interna utilizada para todo el tráfico de control entre los nodos que componen la infraestructura Openstack (por ejemplo, para el tráfico de mensajes y llamadas REST entre todos los módulos software).
- Red de túneles (10.0.1.0/24), que es la red que soporta todo el tráfico entre máquinas virtuales y el nodo de red. Se denomina “de túneles” ya que el tráfico se intercambia mayormente mediante protocolos tales como VXLAN o GRE.
- Red Exterior (10.0.10.0/24), que es la red que da acceso al exterior (Internet) y que permite que las máquinas virtuales (a través del nodo de red) tengan acceso sean accesibles desde el exterior. También se utiliza para el acceso desde el exterior al controlador, mediante el interfaz gráfica o los APIs REST.
- Red VLAN (no representada en la figura), que permite el acceso de red de las máquinas virtuales directamente a través de la infraestructura de switches del centro de datos (se utilizará en prácticas posteriores). Lo anterior se ilustra en la Fig. 25 [6].

5.3 Despliegue

Para la implementación de dicho escenario es necesario conocer más a detalle el Modelo de Datos que gestiona OSM. OSM sigue la especificación oficial ETSI - MANO, que tiene definiciones tanto para los descriptores NSDs como para VNFD, mismas que se detallan a continuación:

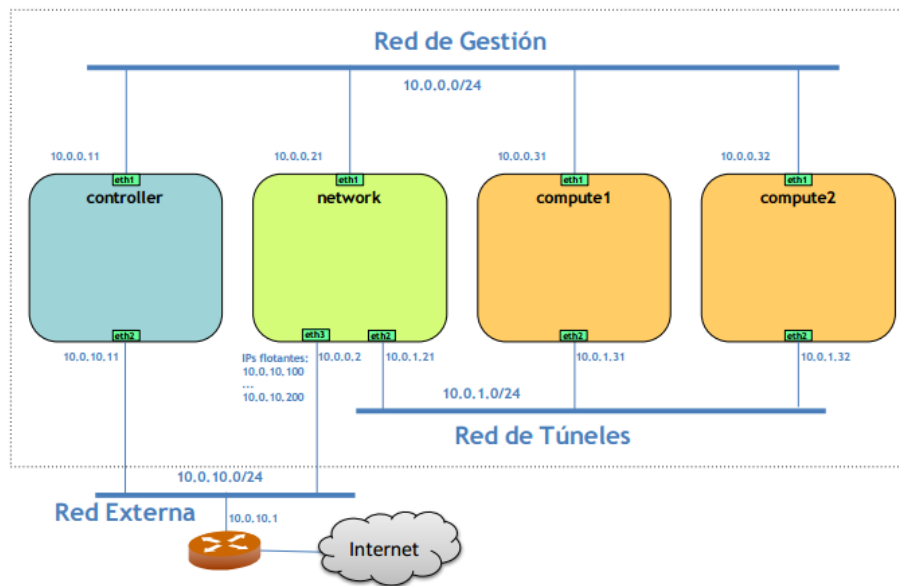


Figura 25: Escenario OpenStack

5.3.1 NSD

NSD o descriptor de servicios de red es el constructor de más alto nivel utilizado para diseñar el encadenamiento de servicios. El NSD se construye usando otros descriptores, que incluye (véase Fig. 26):

- Descriptor VNF (VNF)
- Descriptor PNF (PNFD)
- Descriptor de enlace virtual (VLD)
- Descriptor de reenvío gráfico y VNF (VNF-FGD)

El Descriptor de Servicios de Red define un conjunto de VNFs interconectadas para implementar un servicio de red que abarca múltiples VNFs (encadenamiento de servicios). Fig. 27. Estos VNFs están conectados mediante VLDs , y VNF-FGD y habilita la conectividad a otros servicios de red o al mundo externo. Además, NSD define la información y configuración a nivel de NS. NSD es procesado por el OSM Service Orchestrator para instanciar un Servicio de Red, el cual puede incluir VNFs, reenvío de imágenes y enlaces entre ellas. Los VNF se conectan a los VL usando los Puntos de Conexión

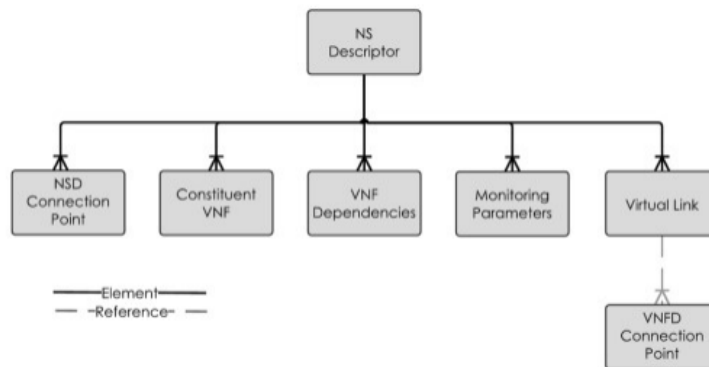


Figura 26: Modelo de Objetos de NSD

(CPs). Así que el NSD captura la lista de VNFs y los VLs que conectan los VNFs [20].

El paquete NS puede contener los siguientes archivos y directorios:

- nsd.yaml: descriptor NS en formato YAML.
- README: Información sobre la NS.
- checksum.txt: contiene suma de comprobación para cada archivo en el paquete.
- scripts: Directorio que contiene los scripts NS lifecycle events y primitivos.
- iconos: Directorio que contiene iconos y registros para la red.
- ns-config: El directorio contiene archivos con valores predeterminados de nivel NS y configuración inicial.
- vnf-config: El directorio contiene archivos con valores predeterminados de nivel VNF y configuración inicial. [20].

5.3.2 VNFD

Un VNFD es una plantilla de despliegue que especifica un VNF en términos de implementación y requisitos de comportamiento operacional. Además, VNFD especifica las conexiones entre las Unidades de Despliegue Virtual (VDU) utilizando los enlaces Virtuales (VLs) internos. Un VDU se refiere a una máquina

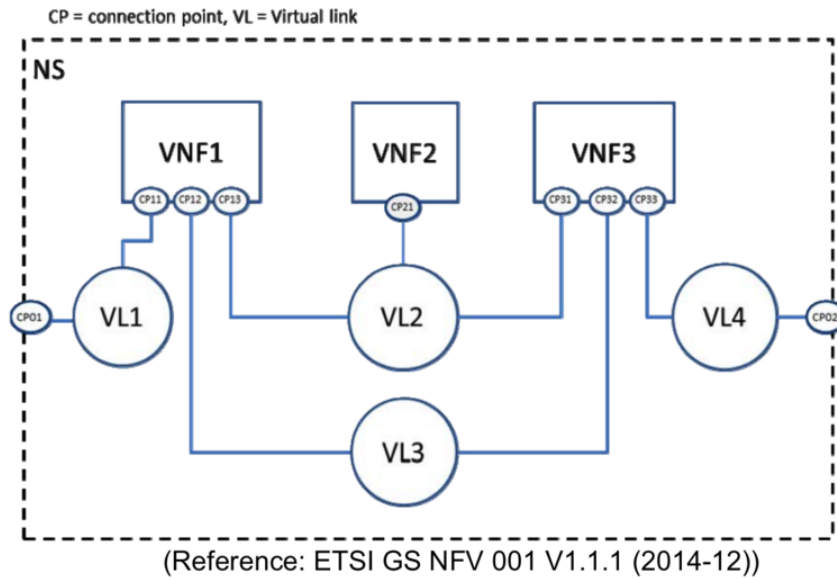


Figura 27: Conexiones VNF

virtual o un contenedor dentro de una sesión OSM. Fig. 28. Para definir plantillas NSD, se utilizan documentos basados en YAML. OSM utiliza el mismo formato de archivos tanto para NSD como para VNFD. Este archivo consiste en la descripción del servicio o VNF, los guiones de configuración iniciales y otros detalles auxiliares [1].

El paquete VNF puede contener los siguientes archivos y directorios:

- vnfd.yaml: archivo descriptor VNF en formato YAML.
- README: contiene información acerca de este paquete VNF.
- checksum.txt: contiene el checksum para cada archivo del paquete.
- images: Directorio que contiene todas las imágenes VM para este VNF.
- scripts: Directorio que contiene scripts personalizados para eventos de ciclo de vida y primitivas de configuración.
- icons: Directorio que contiene los iconos.
- charms: Directorio que contiene los charms/plug-ins de configuración [20].

5.3.3 Lenguaje y formato de modelado de datos

Los descriptores VNF y NS se modelan usando YANG, el formato de entrada para el descriptor es YAML. Los descriptores se validan contra

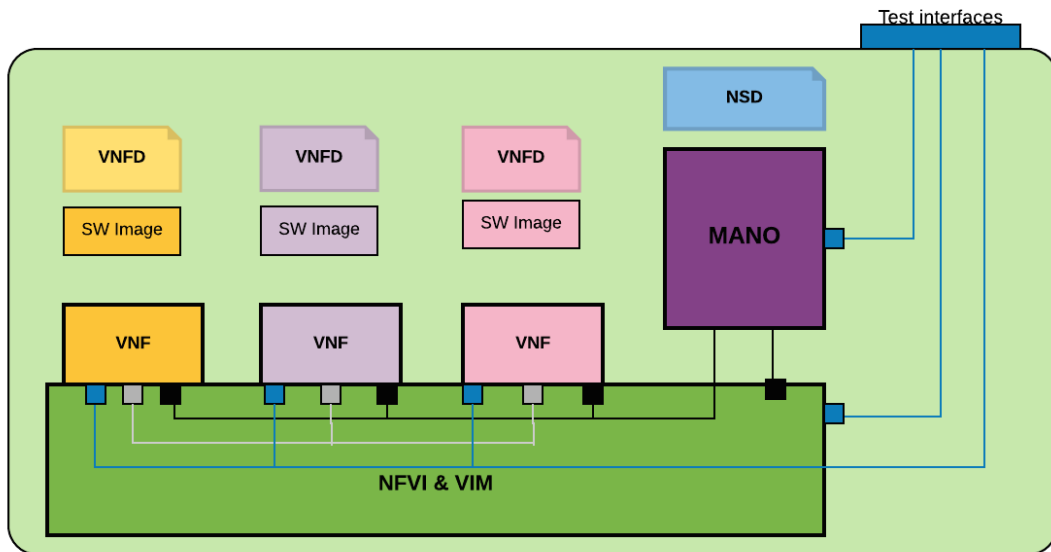


Figura 28: Estructura de un despliegue de Open Source MANO

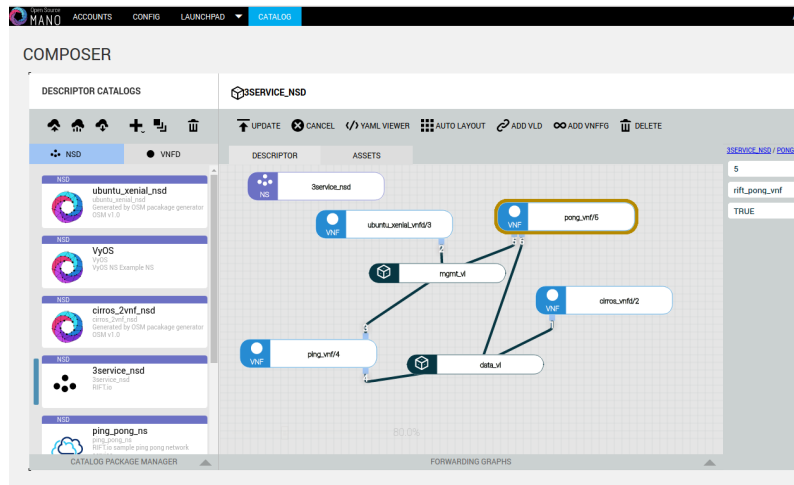
el esquema YANG cuando los paquetes VNF/NS están incluidos. NSD y VNFD están integrados utilizando paquetes NS y VNF respectivamente Fig. 29. En el momento de la instanciación se necesitan objetos adicionales como imágenes, iconos, guiones y otros, además de los descriptores. Para proporcionar una simplicidad, los paquetes se encapsulan en objetos que son necesarios además de los descriptores. Los paquetes se distribuyen en el formato tar.gz.

5.3.4 Servicio RO

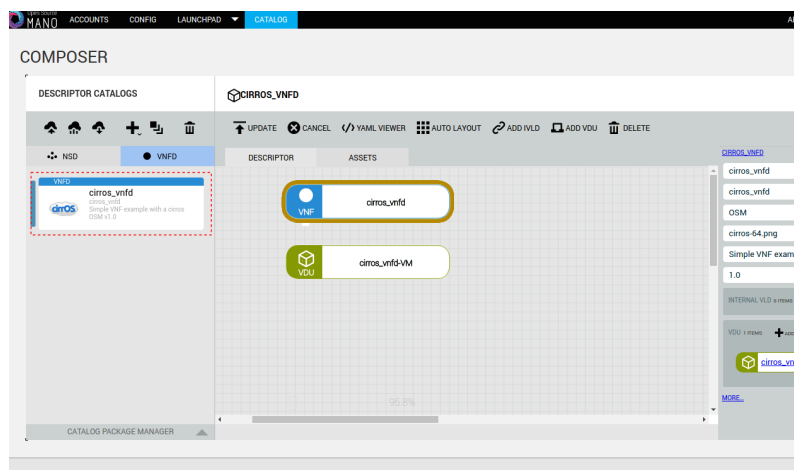
El servicio OSM RO almacena información sobre todos los VIM disponibles para su implementación, mientras que el Service Orchestrator puede usar esta información durante el proceso de instancia NSD. Mientras tanto, la red subyacente entre VIMs debe estar preconfigurada. Además, existe la opción del uso de contenedores de Linux (LXC). No se trata de un modelo que haga uso de un hipervisor, sino que se comparte el Kernel y los recursos entre la máquina anfitriona y los hosts Fig. 30.

5.3.5 Acoplamiento Open Source Mano y Open Stack

Para acoplar ambas plataformas es necesario comprender el flujo de trabajo tanto en la orquestación que realiza Open Source MANO, como en la de



(a) NSD



(b) VNFD

Figura 29: Carga de paquetes ejemplo NSD y VNFD en OSM

aprovisionamiento de infraestructura o el rol de VIM que provee OpenStack.

El orquestador de Open Source MANO contiene una northbound interface que expone las siguientes funcionalidades: anuncio de servicios, despliegue y aprovisionamiento de servicios; Monitoreo y reconfiguración y servicio de desmontaje. Sobre el south side tiene una interfaz con los diferentes gestores de VNF como así como los plugins de conexión para los VIM(s). Estos plugins son los que facilitan la interoperabilidad con distintos VIM y SDN controllers Fig. 31.

OSM se desplegará sobre una red Neutron de OpenStack, por lo que el éxito del acoplamiento depende de configurar correctamente el módulo de red Neutron, lo que es fundamental para la gestión del tráfico entre ambas pla-

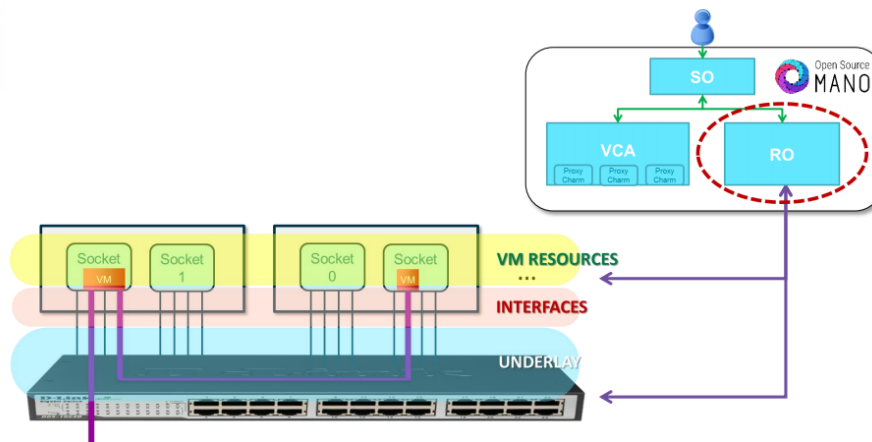


Figura 30: Servicio RO

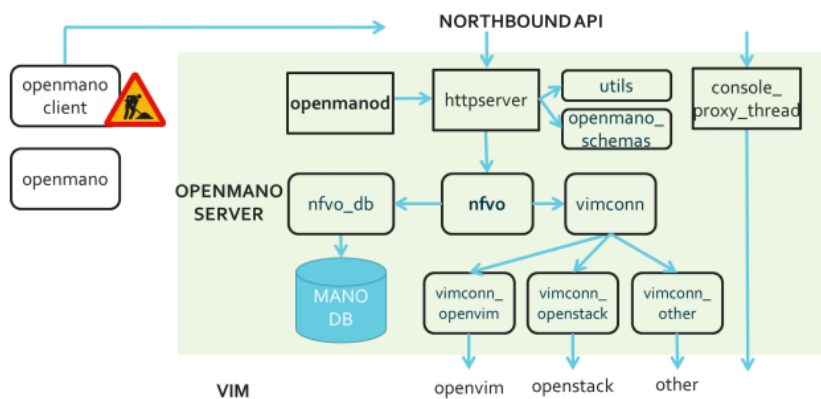


Figura 31: Plugin Model

taformas. OpenStack es el que provee el API de Neutron para la conexión entre este y OSM. OSM por su parte también ha desarrollado dicho plugin para la comunicación con Neutron. Utilizando plugins y servicios del API de Neutron y OSM, Openstack debería poder configurar varios agentes (Open vs-switch/vRouter), residentes en hypervisor. Estos agentes de red configuran el tráfico entre las máquinas virtuales y el tráfico a un gateway externo. Una vez creada la red, es necesario que en OSM configure las credenciales necesarias para la autenticación con OpenStack.

OSM gestiona el ciclo de vida de una NFV, para la instanciación de un nuevo servicio de red se pueden describir los siguientes pasos:

- 1) El módulo SO-ub de OSM gestiona el catálogo los paquetes descriptores NSD y VNFD de los servicios y los catálogos, es el encargado de la interfaz de usuario (UI) y de la orquestación de servicios.

2) El usuario desde el LaunchPad en el Dashboard de OSM Mano escogerá y ejecutará la nueva instancia de servicio de red que desee crear (NS) Fig.33.

3) El módulo VCA valida la solicitud de la nueva instancia y verifica la existencia de los paquetes necesarios.

4) Una vez que haya sido validada la solicitud se realiza la traducción de VNFD.

5) VCA envía un ACK al módulo So-ub confirmando la creación de la nueva VNF.

6) Se crea la instancia del nuevo servicio de red.

7) Se despliega el escenario cargando el NS.

8) El módulo Orquestador de Recursos (RO), se encarga de procesar los requerimientos de localización de las instancias VNF y direccionar al VIM a una ubicación apropiada de computación, red y almacenamiento, para desplegar VNFs con sus interconexiones. Fig.32.

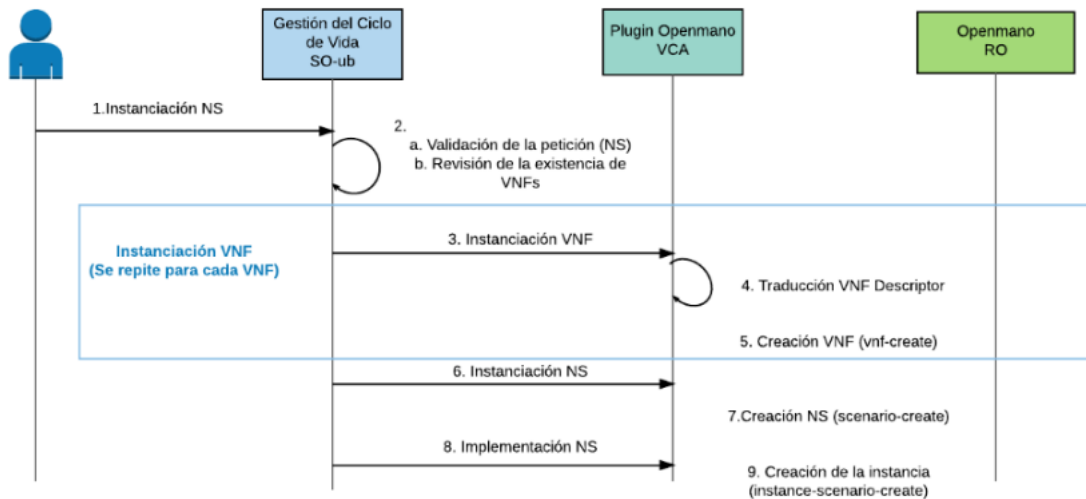


Figura 32: Ciclo de vida NFV

Una vez instancia el nuevo servicio de red, OSM realiza una petición a OpenStack solicitando recursos VIM. Entonces OpenStack ejecuta una secuencia de flujo de trabajo, que se detalla a continuación:

1) OSM envía las credenciales y estas son autenticadas por el servicio de Keystone.

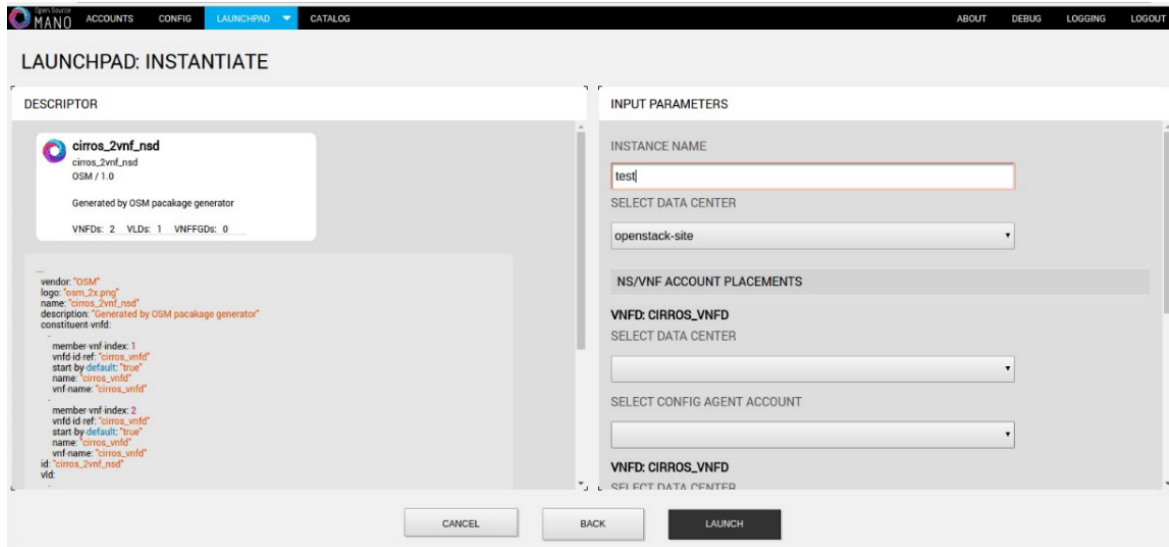


Figura 33: Despliegue de Instancia en OSM

2) El servicio Nova interactúa con el hipervisor del nodo de computación para reservar los recursos virtuales requeridos.

3) El servicio Nova interactúa con el nodo Cinder para reservar la capacidad de almacenamiento necesaria para VM

4) El servicio Nova interactúa con el plugin node/SDN de Neutron para crear una red para la máquina virtual

6) El servicio de Neutron se coordina con el agente L2&DHCP, para obtener detalles de la red. Los detalles de la red pasan a Nova a través de MQ.

7) El servicio Nova interactúa con Glance para descargar la imagen requerida del sistema operativo invitado para VM

8) El servicio Nova notifica al usuario sobre la instanciación de VM exitosa

La plataforma de orquestación de Openstack llevará a cabo estas actividades de manera secuencial y armonizada para proveer una máquina virtual, véase Fig. 34. Open Source Mano interactuará con Openstack en el nivel OS invitado. La imagen del SO para la VNF debe estar disponible en repositorio de imágenes de OpenStack y este lo descargará durante el aprovisionamiento de VM. En otras palabras, la conexión entre Openstack Glance es a través de la imagen de OS invitado.

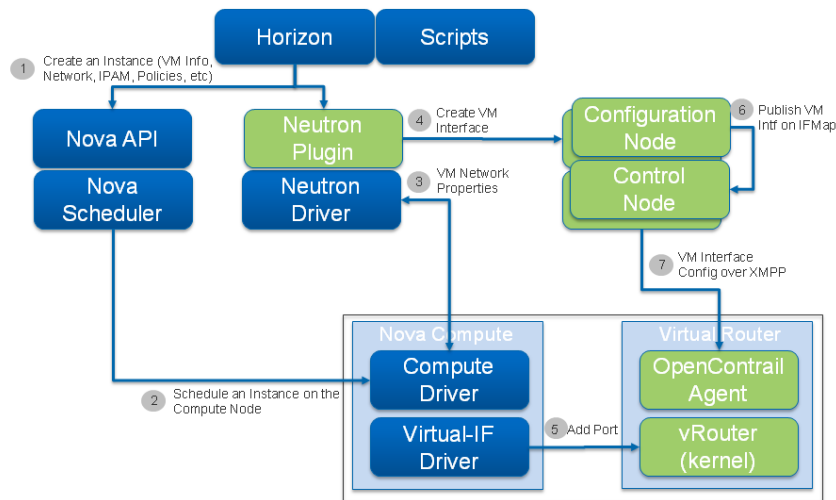


Figura 34: Flujo de trabajo OpenStack

5.3.6 Catálogos NFV

Los descriptores VNF y NS son la base para proporcionar a los Proveedores de Servicios (SP) un sistema abierto y con un ecosistema multi-fabricante de capacidades robustas de red y la posibilidad de automatizar su despliegue y administración. El catálogo de servicios de red es un conjunto de plantillas predefinidas, que definen cómo los servicios pueden ser creados y desplegados, así como las funciones necesarias para el servicio y su conectividad, también describen su despliegue y las características operativas disponibles de VNFs. Los catálogos NFV están compuestos por paquetes de VNF Descriptors, Network Services Descriptors y VNF Forwarding Graph Descriptors, todos estos necesarios según la ETSI para el despliegue de NFV y gestionados y orquestados por MANO [9], véase Fig. 35.

El repositorio de recursos NFVI contiene información acerca de los recursos disponibles NFVI / asignados. El repositorio de instancias NFV contiene información sobre todas las funciones y servicios durante su existencia. El catálogo de VNF que tengamos disponibles incrementará la funcionalidad de la implementación de NFV. Hoy en día existen muchos fabricantes que han volcado esfuerzos en el desarrollo de catálogos NFV, fabricantes como APERI, ARISTA, BROCADE, CISCO, DELL, DIALOGIC, F5, HP, HUAWEI, JUNIPER, NEC, NOKIA, SONUS, etc, tienen a disposición de sus clientes amplios catálogos de VNFs para ser implementados bajo un orquestador.

Para el caso específico de Open Source Mano, este cuenta con algunos

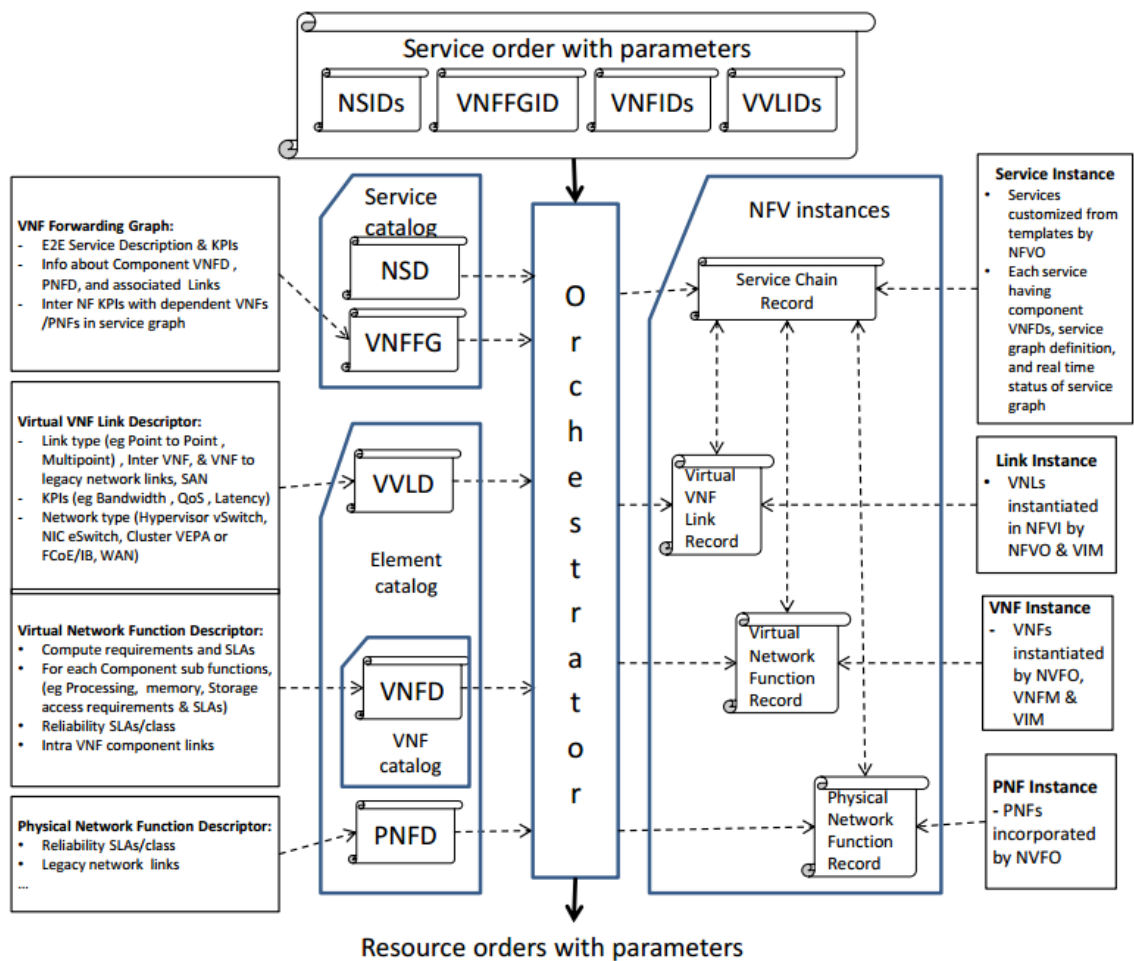


Figura 35: Servicio de Catálogo NFV

paquetes demostrativos VNF. OSM también propone la creación de propias NFV proporcionando ejemplos de referencia que proporcionan ayuda a los proveedores de VNF para crear sus propios descriptores (NSD, VNFD) utilizables en OSM [23]. La creación de paquetes VNF compatibles con OSM se lo puede desarrollar con cualquiera de estas tres opciones:

- Utilizando el Generador de Paquetes basado en la Web
- Empleando el utilitario generador de paquetes
- Aplicando el Composer del propio OSM [21]

Para cualquiera de las opciones que se escojan es necesario tomar en cuenta los Pro y Contras que se pueden presentar sabiendo la clase de paquete de VNF

que queremos desarrollar y características como su tamaño y complejidad. A continuación se presentan los Pro y Contras de las opciones presentadas [21].

Generador de Paquetes basado en la Web

Pros

- Extremadamente fácil de usar y ayuda a comenzar con OSM.

Contras

- Por ahora solo admite VNF simples.
- No hay soporte para objetos adicionales dentro del paquete.

Utilitario generador de paquetes

Pros

- Fácil de usar para VNF simples.
- Capacidad para realizar actualizaciones del descriptor en un paquete existente.
- Soporte para características adicionales.
- Fácil de obtener una plantilla inicial para VNFs más complejos.

Contras

- Se vuelve difícil si se quiere utilizar para paquetes complejos.
- Requiere edición manual de la plantilla de descriptor inicial utilizando editores de texto normales.

Composer del propio OSM

Pros

- Interface gráfica fácil de usar para generar paquetes de descriptores únicamente.
- Más fácil generar descriptores para VNFs y NSes complejos.

Contras

- No admite todavía la gestión de objetos (íconos, charms, scripts).
- Requiere usar el utilitario generador de paquetes para agregar objetos a un paquete.

5.4 Resultados

Con la creación de este escenario didáctico para el manejo y orquestación NFV, es posible experimentar con un ejemplo despliegue de NFV. De esta forma, se pueden demostrar los conceptos de una forma más concreta. Ya se ha analizado el flujo de trabajo de OSM y OpenStack y la comunicación entre ambas plataformas, a continuación se presenta los resultados de este proceso. A modo ejemplo se implementará un servicio de red, que consiste en cuatro VNF simples basadas en Linux bajo sistemas CirrOS y VyOS que están conectados por virtual links descriptors (VLDs) simulando un red típica MPLS. En la demostración se simularán routers CE y PE dentro de una red WAN. Tal y como se muestra en la Fig. 36.

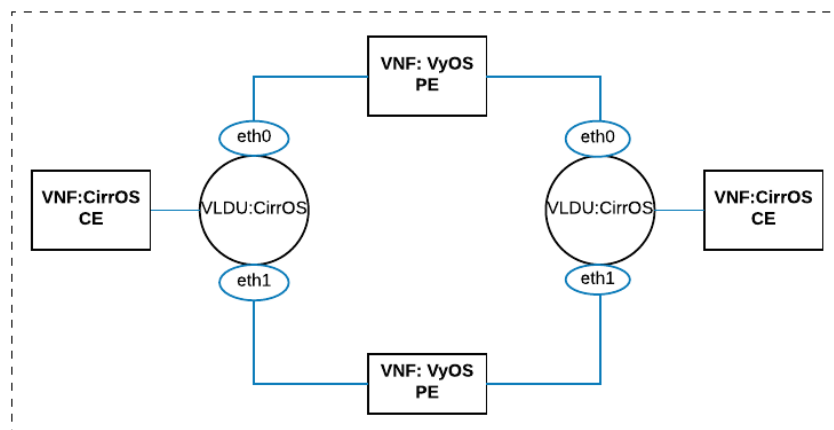


Figura 36: Ejemplo de implementación NFV

Para ello los paquetes descriptores tanto de la VNF como del VLD deben ser descargados y agregados al catálogo de OSM previamente. Para esta demo el catálogo contienen un paquete VNF (`cirros_vnf.tar.gz`) basado en una imagen CirrOS con interfaces de red para cada conexión sola interfaz y un paquete NS (`cirros_2vnf_ns.tar.gz`) que modela un NS con dos VNF de CirrOS interconectados por un enlace virtual, los que representan los CE o routers del cliente tal como se muestra en la Fig. 36. La imagen de CirrOS también ya ha sido cargado previamente en OpenStack, pues está es la imagen base sobre la que se van a montar las VNFs. Por otro lado, también se cargan paquetes VNF de tipo VyOS lo que para este ejemplo representan los routers del proveedor o PE para lo cuál también deberán ser cargadas previamente los paquetes vnf y descriptores de esto. También se puede observar que es necesario configurar VLD para interconectar las VNF, para esto se deben configurar las interfaces de red y de ser necesarios otros parámetros para su conexión.

A continuación, ya es posible crear una nueva instancia donde OSM a través del componente RO comunica al VIM (OpenStack) que necesita recursos para el despliegue de la nueva NFV. Mediante los descriptores el orquestador solicita la creación de nuevas instancias en los nodos de cómputo una para cada VNF, tal y como se muestra en la imagen Fig. 37.

En los hipervisores en OpenStack se despliega la Función de Red Virtualizada, en cada plataforma se crea una instancia y se reservan recursos de almacenamiento y cómputo Fig. 38. En la imagen se puede apreciar que tanto la memoria como los procesadores se ven afectados.

Otro de los recursos solicitados desde OSM a OpenStack son los recursos de red, además de la red compartida "mgmt" que se creó inicialmente para el proceso de acoplamiento, se solicita una red para comunicación entre las VNFs, lo que se ve representado en OSM como componentes VLD. La nueva red que pide el orquestador es una red no compartida que se crea según las definiciones del descriptor VLD tal y como se especifica en el ejemplo con dos puertos, uno para cada VNF, véase Fig. 39.

Cuando la instancia se despliega genera tokens para acceder a las máquinas virtuales de las VNF que representa un recurso virtual por cada una Fig. 40 . A través de estos tokens y de manera gráfica se puede acceder a las máquinas virtuales que representan los servicios de red. Tal y como se había mencionado, se pueden agregar más paquetes VNF y encadenarlos para implementar servicios más complejos, lo que sin duda representa mayor consumo de recursos físicos como lógicos dependiendo del escenario a implementar. OSM por

```

root@RO:~# openmano instance-scenario-list test
c383f43e-55b5-11e7-95ff-00163e1badde test 2017-06-20T12:41:21
Description: a description for 1e78ba05-18a5-4bbd-9b57-160a699f9a21
Template scenario id: c0f9e60b-55b5-11e7-95ff-00163e1badde
Template scenario name: cirros_vnfd_nsd
-----
VNF instances: 2
c383f440-55b5-11e7-95ff-00163e1badde cirros_vnfd_1 Template vnf id: c0f9e608-55b5-11e7-95ff-00163e1badde
c383f443-55b5-11e7-95ff-00163e1badde cirros_vnfd_2 Template vnf id: c0f9e608-55b5-11e7-95ff-00163e1badde
-----
Internal nets:
c383f43f-55b5-11e7-95ff-00163e1badde ACTIVE VIM ID: 3054df78-0a5e-4089-8f29-3859f2aae411
-----
External nets:
-----
VM instances:
c383f441-55b5-11e7-95ff-00163e1badde cirros_vnfd_1 cirros_vnfd-VM ACTIVE VIM ID: dc0bc71b-058d-4ce8-810d-d3dd5ad94399
c383f444-55b5-11e7-95ff-00163e1badde cirros_vnfd_2 cirros_vnfd-VM ACTIVE VIM ID: 714c8f74-8e67-48d9-9b55-320d0bb1c8720

```

(a) Instancias VNF OSM

Instancias

Proyecto= Filtrar

<input type="checkbox"/>	Proyecto	Host	Nombre	Nombre de la imagen	Dirección IP	Tamaño	Estado	Tarea	Estado de energía	Tiempo desde su creación	Actions
<input type="checkbox"/>	admin	compute2	test5.pong_vnf_5.a	Fedora-x86_64-20-20131211.1-sda-pong.qcow2	data_vl 31.31.31.5 test5.mgmt_vl 192.168.31.5	iovedu_0-flv	Activa	Ninguno	Ejecutando	3 minutos	<input type="button" value="Editar instancia"/>
<input type="checkbox"/>	admin	compute2	test5.cirros_vnfd_2.a	cirros034	31.31.31.4	m1.tiny	Activa	Ninguno	Ejecutando	3 minutos	<input type="button" value="Editar instancia"/>
<input type="checkbox"/>	admin	-	test5.ubuntu_xenial_vnfd_3.a	ubuntu1604		ubuntu_xenial_vnfd-VM-flv	Error	Ninguno	Sin estado	3 minutos	<input type="button" value="Editar instancia"/>
<input type="checkbox"/>	admin	compute2	test5.ping_vnf_4.a	Fedora-x86_64-20-20131211.1-sda-ping.qcow2	data_vl 31.31.31.3 test5.mgmt_vl 192.168.31.2	iovedu_0-flv	Activa	Ninguno	Ejecutando	4 minutos	<input type="button" value="Editar instancia"/>
<input type="checkbox"/>	admin	compute1	test2.ubuntu_xenial_vnfd_1.a	ubuntu1604	10.208.0.9	ubuntu_xenial_vnfd-VM-flv	Activa	Ninguno	Ejecutando	1 hora, 5 minutos	<input type="button" value="Editar instancia"/>
<input type="checkbox"/>	admin	compute2	test.cirros_vnfd_2.a	cirros034	192.168.201.4	m1.tiny	Activa	Ninguno	Ejecutando	1 hora, 14 minutos	<input type="button" value="Editar instancia"/>
<input type="checkbox"/>	admin	compute1	test.cirros_vnfd_1.a	cirros034	192.168.201.3	m1.tiny	Activa	Ninguno	Ejecutando	1 hora, ...	<input type="button" value="Editar instancia"/>

(b) Instancias VNF OpenStack

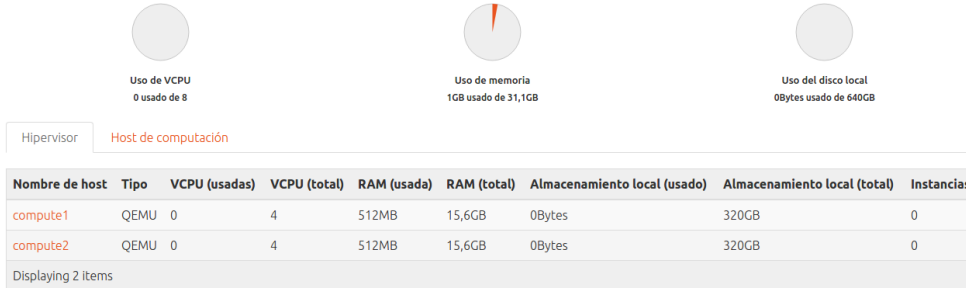
Figura 37: Instancia OSM desplegada en OpenStack

su parte cumplió con su objetivo, orquestó el despliegue automático y la interconexión de todos los componentes, tanto de los escenarios NFV como de la gestión de los ciclos de vida de los servicios de red y el acoplamiento con OpenStack y este último sirvió de VIM para los recursos físicos y lógicos que necesitó OSM.

Una vez iniciada las instancias del escenario se puede configurar los parámetros de red necesarios para la interconexión de los routers, según sea el servicio requerido. Esta es una implementación típica de un proveedor de servicios, que ofrece una red virtual a través de la tecnología MPLS. Sin duda existen infinitas posibilidades de servicios implementables con NFV.

Todos los hipervisores

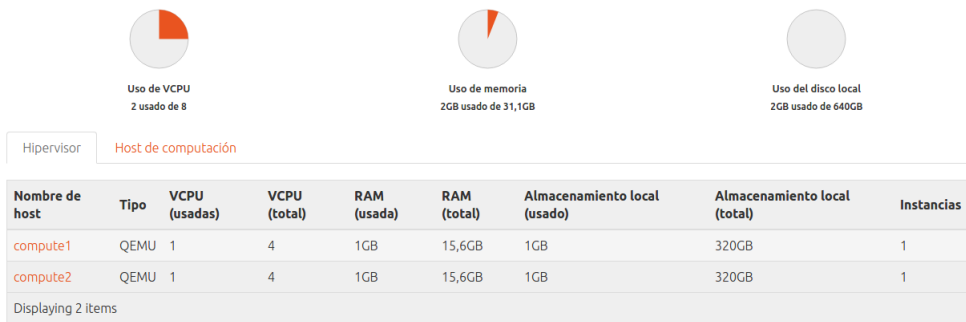
Resumen del hipervisor



(a) Antes del despliegue

Todos los hipervisores

Resumen del hipervisor



(b) Después del despliegue

Figura 38: Estado del Hypervisor de Openstack

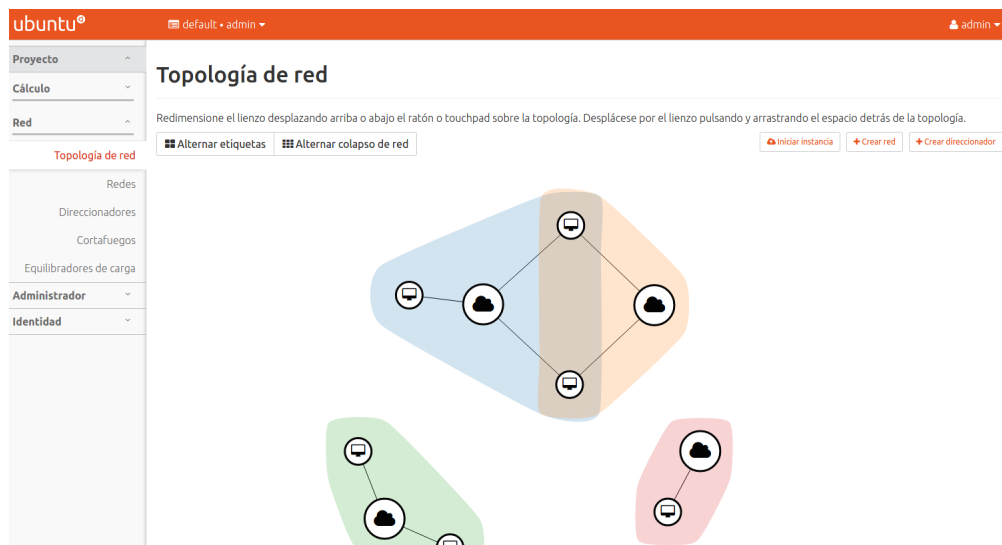


Figura 39: Topología de Red creada

NETWORK SERVICE DETAILS

The screenshot shows a web interface titled "Application Dashboard Links" and "VM Console Links". The "Application Dashboard Links" section contains two entries: "cirros_vnfd-0f64" and "cirros_vnfd-34db", both with a status of "None". The "VM Console Links" section contains two entries: "test_cirros_vnfd_1-c383" and "test_cirros_vnfd_2-c383", both with a link labeled "Obtain Token And Open VM Console *". A note at the bottom states: "* If a separate browser window does not open, please check if the popup was blocked and allow it."

Link Name	Status
cirros_vnfd-0f64	None
cirros_vnfd-34db	None

Link Name	Action
test_cirros_vnfd_1-c383	Obtain Token And Open VM Console *
test_cirros_vnfd_2-c383	Obtain Token And Open VM Console *

(a) NFV desplegada en OSM

```
controller:6080/vnc_auto.html?token=77e71aa9-4adf-46b0-b8d2-fd2c4c0da5ba
Connected (unencrypted) to: QEMU (Instance-00000001)
user name:#uid1 file ...

ifconfig
eth0  Link encap:Ethernet  HWaddr FA:16:3E:7A:5B:D6
      inet addr:192.168.196.2  Bcast:192.168.196.255  Mask:255.255.255.0
      inet6 addr: fe80::f816:3eff:fe7a:5bd6/64  Scope:Link
      UP BROADCAST RUNNING MULTICAST  MTU:1400  Metric:1
      RX packets:94 errors:0 dropped:0 overruns:0 frame:0
      TX packets:119 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:1000
      RX bytes:9051 (8.8 KiB)  TX bytes:11204 (11.0 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:6 errors:0 dropped:0 overruns:0 frame:0
      TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
      RX bytes:672 (672.0 B)  TX bytes:672 (672.0 B)

$ _
```

(b) Instancia Desplegada

Figura 40: Instancia NFV

6 Conclusiones y Lineas Futuras de Investigación

6.1 Conclusiones

Se evidencia el gran potencial de la virtualización y de sobremanera de la virtualización de funciones de red. Existe una tendencia actual y futura para la creación e implementación de redes más programables y flexibles. Es fascinante como NFV se apodera de las principales tecnologías como la virtualización y la computación en la nube para adaptarlas al mundo de las tecnologías de la información y comunicación. NFV se ve bien definida como la tecnología cloud que los operadores de telecomunicaciones están aplicando para aumentar el beneficio y calidad en sus operaciones.

Tanto las compañías proveedores de servicios, como fabricantes están optando por nuevos modelos de negocio y servicios, y a la vez creando estrategias de virtualización, pues los operadores de telecomunicaciones están convencidos de que las técnicas de virtualización han madurado lo suficiente para virtualizar también funciones de red. Uno de los principales motivos por los que esta tecnología es tan atractiva para los proveedores de servicios es la capacidad para redefinir la red y agregar funcionalidades a sus redes bajo demanda utilizando una infraestructura convergente.

Las múltiples bondades de NFV han hecho que los proveedores de servicios y fabricantes trabajen conjuntamente para sacar adelante esta propuesta, a través de proyectos y plataformas, que han puesto en manifiesto el interés para con esta tecnología; muchos fabricantes se han unido y han desarrollado comunidades Open Source y proyectos conjuntos, lo que potencia aún más la tecnología NFV. Aunque NFV está planteado para los grandes carriers a nivel mundial, se pueden proponer escenarios, que se están dando ya, como implementaciones para entornos empresariales, datacenters e incluso para el usuario final. Es importante que los fabricantes consideren fundamental que al empezar un proyecto NFV es conveniente alinearlos con los marcos de referencia definidos por la ETSI, esto hará posible que sus proyectos sean más robustos, comerciales y fáciles de integrar. Así mismo, es importante que los fabricantes sigan manteniendo estos proyectos como Open Source pues esto influye a que este conocimiento este al alcance de todos y puedan ofrecer alternativas de negocios para telcos, fabricantes y usuarios finales.

Es importante resaltar las funciones que está desempeñando la ETSI, pues se ha realizado un gran trabajo de difusión a través de diferentes portales y blogs así como con múltiples webinars, presentaciones, congresos y encuentros con analistas y especialistas en este ámbito. Están disponibles y al alcance de todo el mundo documentos, ejemplos, wikis y PoC, que abarcan más de la mitad de los casos de uso NFV definidos por el ISG. Estas pruebas de concepto ejemplifican la transición del ISG NFV desde la especificación hasta la implementación, lo que hace posible un mayor entendimiento a quienes estamos intentado entender más a fondo esta tecnología que tanto promete.

Ahora bien, en cuanto al Orquestador utilizado para el desarrollo de este escenario Open Source Mano cumple perfectamente con los objetivos propuestos. Mediante este escenario es posible aprender más a fondo sobre NFV, sobre su arquitectura y su funcionamiento debido a que la plataforma OSM es adecuada para fines de evaluación y experimentación; y tiene una arquitectura relativamente sencilla. Teniendo en cuenta la estrecha relación que tiene el proyecto OSM y el grupo ETSI NFV, este proyecto podría surgir como una opción viable para la producción de mayor grado como Orquestador NFV. El proyecto es relativamente joven, sin embargo, existen una serie de características que siguen fortaleciéndose a través de su comunidad y contribuyentes. OSM todavía requiere más desarrollo pero según su documentación estarán disponible una serie de nuevas características en los próximos lanzamientos.

Por otro lado, la plataforma Open Source Mano se pudo integrar con el escenario virtual de OpenStack, que cumple con rol de VIM dentro de la arquitectura NFV. Es necesario destacar que casi todos los despliegues del NFV se originan en una nube, OpenStack, es extremadamente popular en el manejo de tal infraestructura virtualizada, lo que fue conveniente en su uso como escenario didáctico. OpenStack podría presentar un camino más rápido para abrir y potenciar la capa MANO de NFV, en particular si los esfuerzos actuales para iniciar una iniciativa NFV dentro de OpenStack lo llevan a un conjunto de capacidades de tipo MANO más amplio. En resumen, aunque la tendencia es desarrollar orquestadores y administradores adicionales para trabajar con OpenStack, existen grandes ánimos de que OpenStack desarrolle su propio Orquestador MANO. Aquí nuevamente reluce la importancia de mantener a estos proyectos verdaderamente abiertos.

Finalmente, el escenario virtual creado en este trabajo implicó la investigación a fondo sobre la tecnología NFV. El tema de NFV resulta abstracto al inicio, por lo que se debe resaltar la importancia de conocer ampliamente la arquitectura ETSI-NFV y su modelo de datos, para su previa implementación

y el estudio tanto de las plataformas como el desarrollo y despliegue de NFV. Este escenario, didáctico como se lo ha llamado, ofrece múltiples funcionalidades en el manejo de NFV, pues al tener virtualizadas sus plataformas se pueden desplegar en cualquier ambiente. También esto hace posible probar características importantes de la tecnología NFV y profundizarlas, haciendo que cualquiera que este dispuesto a aprender sobre el tema pueda hacerlo de una manera más “cómoda”, pues no habrá que preocuparse por la implementación previa de alguna herramienta o plataforma NFV, que a la larga resulta pesado por el tiempo y los recursos que conlleva dicha implementación y más bien, directamente proponerse la implementación de un caso práctico de NFV, sea este sencillo o complejo y desplegarlo sobre este escenario.

6.2 Líneas Futuras de Investigación

Después de la investigación y elaboración de este trabajo, quedan aún muchos áreas interesantes por tocar, las mismas que serían atractivas para la realización de un futuro trabajo de investigación, éstas serán tratadas a continuación.

Una de las temáticas de gran importancia, es la integración de NFV con SDN, pues una combinación de estas tecnologías supone una utilidad y un mejoramiento tremendo de eficiencia. El potencial para SDN y NFV es interminable. La superposición virtual creada por SDN ayuda a proporcionar y administrar las funciones de red virtual con NFV. SDN también ayuda a manejar cargas de tráfico de manera más eficiente, por lo que la red puede reaccionar cuando las cosas necesitan cambiar a niveles micro y macro. Aún más, trabajando con el escenario didáctico ya implementado y propuesto en este trabajo, se puede integrar uno de los controladores SDN con los que es compatible OSM como OpenDaylight, FloodLight o el mismo OpenStack. Actualmente el proyecto OSM está mejorando características de integración con estos controladores por lo que ya es posible experimentar un posible escenario y desarrollar una red definida por software que impulse NFV haciendo de esta una combinación poderosa.

Otro tema que se puede desarrollar ampliamente es la creación de paquetes de funciones de red NSD para los catálogos NFV. Aunque al momento existan varios fabricantes desarrollándolos, aún no existe algún proyecto importante que se haya propuesto liberar estos proyectos como open source. Open Source Mano propone algunas utilidades para la generación de NSD pero que no se encuentran desplegadas completamente y permiten la construcción de pa-

quetes VNFD para servicios de red sencillos. Como criterio personal, resulta bastante interesante los estándares propuestos para estos descriptores e implementar paquetes descriptores robustos que cumplan funciones de red complejas produce una línea de investigación extensa en esta materia.

En el mes de junio del presente, se ha lanzado el Release TWO de Open Source Mano, resultaría interesante una actualización de este escenario con la nueva versión, pues se ha demostrado que es posible el acoplamiento con OpenStack y así experimentar con las nuevas funcionalidades que ofrece el nuevo lanzamiento de este orquestador.

7 Bibliografía

Referencias

- [1] ADRIAN HOBAN, CHAIR OSM TSC, T. S. A. R. S. F. J. G. D. B. G. K. K. M. C. M. S. M. H. R. V. S. A. V. L. Osm release two a technical overview, tipo @ONLINE, 2017.
- [2] AGUILAR, L. J. Computación en la nube: Notas para una estrategia española en cloud computing. *Revista del Instituto Español de Estudios Estratégicos* 1, 1 (2013).
- [3] AUTORES, V. Osm release two, tipo @ONLINE, 2017.
- [4] CHAYAPATHI, R., HASSAN, S. F., AND SHAH, P. *Network Functions Virtualization (NFV) with a Touch of SDN*. Addison-Wesley Professional, Nov. 2016.
- [5] CUI, C., DENG, H., CHEN, C., BAOHUA, L., ZHENQIANG, S., ZHOU, X., JIA, C., TELEKOM, D., MICHEL, U., MARTINY, K., ET AL. Network functions virtualisation (nfv). In *SDN & OpenFlow World Congress, Düsseldorf* (2014).
- [6] DAVID, F. Escenario virtual openstack, tipo @ONLINE, 2017.
- [7] DIT-UPM. Virtual networks over linux (vnx), tipo @ONLINE, 2016.
- [8] ENCINAS, D., KUNYSZ, E., GALARZA, B., ZACCARDI, G., AND MORALES, M. D. Performance de arquitecturas multiprocesador: técnicas de modelado y simulación, plataformas reconfigurables y cloud computing. In *XVII Workshop de Investigadores en Ciencias de la Computación (Salta, 2015)* (2015).
- [9] ERSUE, M. Etsi nfv management and orchestration-an overview. In *Proc. of 88th IETF meeting* (2013).
- [10] ETSI, V. A. Network functions virtualisation – introductory white paper, tipo @ONLINE, 2012.
- [11] ETSI, V. A. Network functions virtualisation – network operator perspectives on industry progress white paper, tipo @ONLINE, 2013.

- [12] GALARZA, B., ZACCARDI, G., ENCINAS, D., AND MORALES, M. D. Análisis de despliegue de una iaas utilizando openstack. In *XXI Congreso Argentino de Ciencias de la Computación (Junín, 2015)* (2015).
- [13] HERNANDEZ, N. L., AND FLOREZ-FUENTES, A. S. Computación en la nube. *Revista MundoFesc* 2, 8 (2014), 46–51.
- [14] LOPEZ, D. Openmano: The dataplane ready open source nfv mano stack. In *IETF Meeting Proceedings, Dallas, Texas, USA* (2015).
- [15] MARTÍNEZ DE LA CRUZ, V. ¿Cómo funciona OpenStack? | En pocas de la palabras, 2017.
- [16] MEJIA, O. Computación en la nube. *ContactoS* 80 (2011), 45–52.
- [17] PRICE, C., AND RIVERA, S. Opnfv: An open platform to accelerate nfv. *White Paper. A Linux Foundation Collaborative Project* (2012).
- [18] PRICE, CHRISTOPHER, R. S. Opnfv an open platform to accelerate nfv. *Linux Foundation Collaborative Project* 1, 2 (2015), 1–23.
- [19] PROJECT, O.-O. Open-o architecture, tipo @ONLINE, 2013.
- [20] PROJECT, O. S. M. Osm information model , tipo @ONLINE, 2016.
- [21] PROJECT, O. S. M. Vnf package creation & onboarding , tipo @ONLINE, 2016.
- [22] PROJECT, O. S. M. Osm release two, tipo @ONLINE, 2017.
- [23] PROJECT, O. S. M. Reference vnf and ns descriptors , tipo @ONLINE, 2017.
- [24] SOARES, J., DIAS, M., CARAPINHA, J., PARREIRA, B., AND SARGENTO, S. Cloud4nfv: A platform for virtual network functions. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on* (2014), IEEE, pp. 288–293.
- [25] STALLINGS, W. *Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud*. Addison-Wesley Professional, 2015.
- [26] VEERARAGHAVAN, M., SATO, T., BUCHANAN, M., RAHIMI, R., OKAMOTO, S., AND YAMANAKA, N. Network function virtualization: A survey. *IEICE Transactions on Communications*, 2016NNI0001.
- [27] VOORHEES, C. C. Taking nfv to the cloud, tipo @ONLINE, 2013.

8 Anexos

Arranque del escenario Openstack

Para arrancar el escenario virtual siga los siguientes pasos:

- Descargar el escenario:

```
http://idefix.dit.upm.es/download/cnvr/openstack_tutorialmitaka_4n_classic_ovs-v04-with-rootfs.tgz
```

- Copie y descomprima el escenario virtual de Openstack:

```
cp /mnt/vnx/repo/cnvr/openstack_tutorial-mitaka_4n_classic_ovs-v04-withrootfs.tgz
sudo vnx --unpack openstack_tutorial-mitaka_4n_classic_ovs-v04-with-rootfs.tgz
```

- Arranque el escenario mediante:

```
cd openstack_tutorial-mitaka_4n_classic_ovs-v04
sudo vnx -f openstack_tutorial.xml -v --create
```

- Una vez arrancado el escenario deberían aparecer cuatro ventanas de consola: una del controlador, una del nodo de red y dos de los dos nodos de computación. Si alguna de ellas no aparece (el primer arranque falla siempre en el laboratorio), re arranque el escenario mediante:

```
sudo vnx -f openstack_tutorial.xml -v --destroy
sudo vnx -f openstack_tutorial.xml -v --create
```

- Configure el escenario mediante:

```
sudo vnx -f openstack_tutorial.xml -v -x start-all
```

- A continuación, arranque una ventana de navegador y acceda al URL:

```
http://10.0.10.11/horizon
```

- Si el escenario ha arrancado correctamente, debería ver la ventana de login de Openstack. Acceda a la misma como usuario sin privilegios de administración utilizando los datos siguientes:

```
Dominio: default
Usuario: admin
Clave: xxxx
```

Creación Escenario Open Source Mano Release ONE

- Instalar y configurar LXD para OSM Release ONE

```
sudo apt-get update
sudo apt-get install zfs lxd
newgrp                                lxd
# Necesario para registrar el usuario en el grupo lxd
```

- Ejecutar el comando siguiente para configurar lxd de la siguiente manera:

```
sudo lxd init
Name of the storage backend to use (dir or zfs): zfs
Create a new ZFS pool (yes/no)? : yes
Name of the (existing/new) ZFS pool or dataset: lxdpool
Would you like to use an existing block device (yes/no)? no
Size in GB of the new loop device (1GB minimum): 50
Would you like LXD to be available over the network (yes/no)? no
Do you want to configure the LXD bridge (yes/no)? yes
```

- Preparados los requerimientos anteriores procedemos, descargamos y ejecutamos el script:

```
wget https://osm-download.etsi.org/ftp/osm-1.0-one/install_from_source.sh
chmod +x install_from_source.sh
./install_from_source.sh
```

- Ahora es posible acceder mediante el navegador con la ip asignada al contenedor de SO-ub mediante el URL:

```
https://10.1.0.224:8443
```

Acoplamiento Open Source Mano y OpenStack

Para el acoplamiento de Open Source Mano Release ONE y OpenStack se realizó los siguientes pasos:

En OpenStack:

- Editar el archivo: `/etc/keystone/keystone.conf`, con el ID del dominio por default OpenStack:

```
default_domain_id = cfa1aab7f403414ea77cbb01f4d11702
```

- Creación de una management network, con DHCP alcanzable para OSM y especialmente para el contenedor VCA:

```
cat /root/bin/admin-openrc.sh #Vemos las variables
source /root/bin/admin-openrc.sh
neutron net-create mgmt --provider:network_type=vlan --
provider:physical_network=vlan --provider:segmentation_id=500 --shared

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

- Descargar cirros image en el siguiente directorio y si no existe crearlo: `/mnt/powervault/virtualization/osm/CirrOS`:

```
openstack image create --file="./cirros-0.3.4-x86_64-disk.img" --container-format=bare -
-disk-format=qcow2 --public --property
location="/mnt/powervault/virtualization/osm/CirrOS/cirros-0.3.4-x86_64-disk.img"
cirros034
```

- Editar el fichero `/etc/neutron/plugins/ml2/ml2_conf.ini` con los parámetros:

```
[ml2_type_vlan] network_vlan_ranges = physnet_sriov:3000:3100
```