

# Open Source MANO

Eduardo.Jacob@ehu.eus  
jorge.sasiain@ehu.eus  
david.franco@ehu.eus



eman ta zabal zazu



Universidad  
del País Vasco

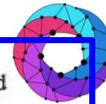
Euskal Herriko  
Unibertsitatea

# Contenido

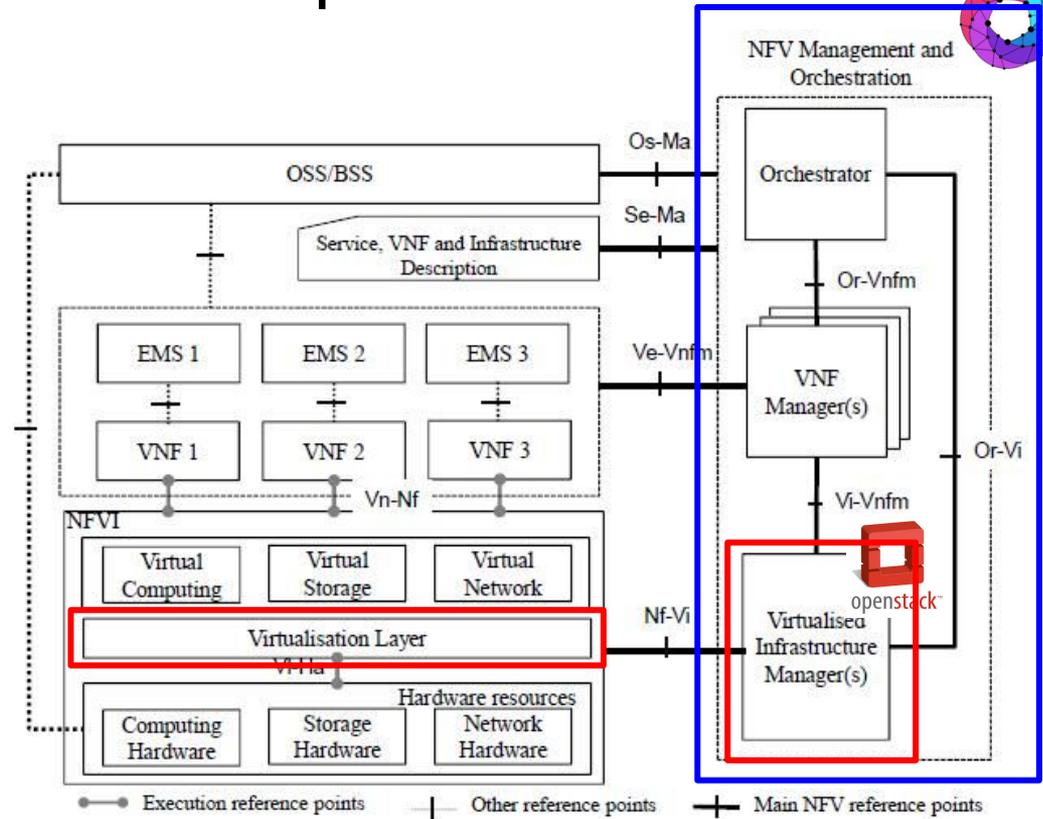
- Introducción
  - Conceptos básicos
  - Virtual Network Function (VNF)
  - Network Service (NS)
  - Despliegue de servicios en OSM
  - Módulos principales de OSM
- Instalación y configuración
- Labs
  - Lab 0 - Registro de VIM en OSM
  - Lab 1 - Servicio simple
  - Lab 2 - Servicio compuesto
  - Lab 3 - Servicio configurable
- Documentación



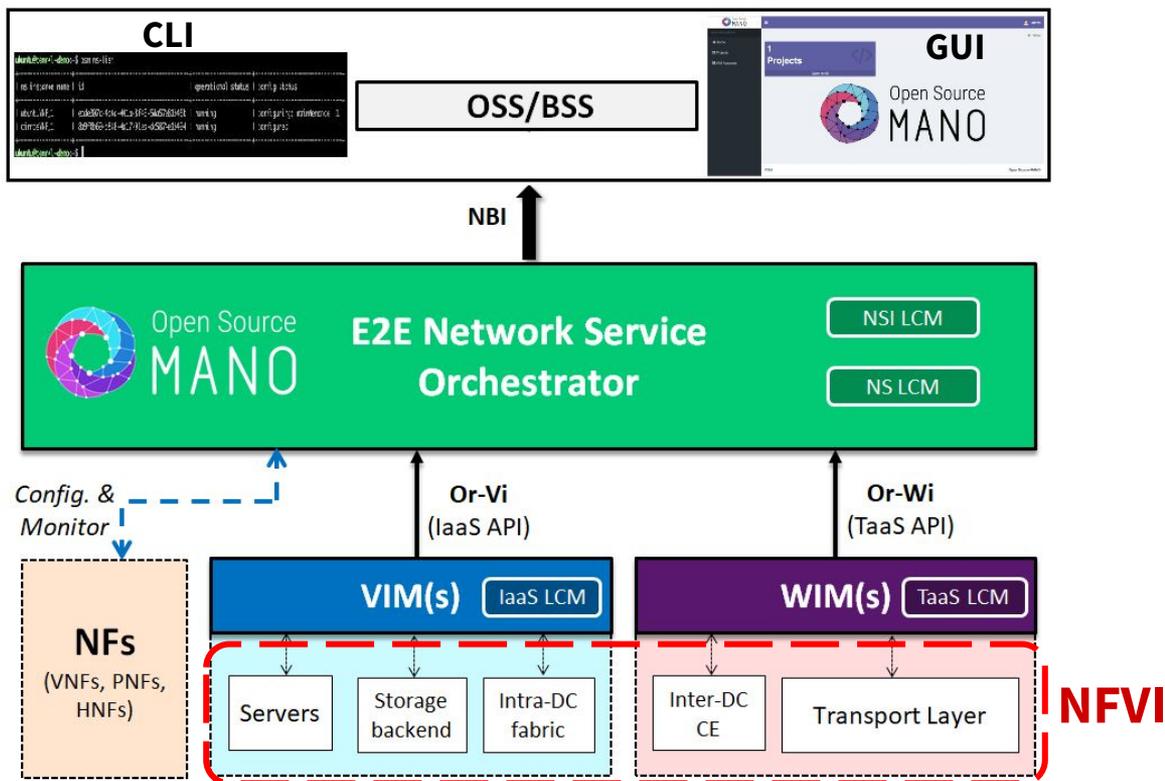
# Introducción - Conceptos básicos



Open Source MANO



# Introducción - Conceptos básicos



# Introducción - Conceptos básicos

- Virtual Deployment Unit (**VDU**)
  - Describe las capacidades de una máquina virtual (imagen S.O., CPUs, memoria RAM, almacenamiento, interfaces de red, etc.)
- Virtual Network Function Descriptor/Package (**VNFD/VNFP**)
  - Formado por al menos una VDU
  - Describe una **VNF** o función de red (ej: firewall, IDS, balanceador de carga, servicio web/correo/DNS/datos, vEPC/eNodeB/gNodeB en LTE/5G, etc.)
  - El proveedor de la VNF diseña su topología, incluyendo las VDUs y sus conexiones
- Network Service Descriptor (**NSD**)
  - Formado por al menos una VNF
  - Describe un **NS** o servicio de red
  - El administrador de la red diseña la topología del NS, indicando que VNFs lo conforman y diseñando cómo se conectan entre sí y se integran en la red

# Introducción - Conceptos básicos

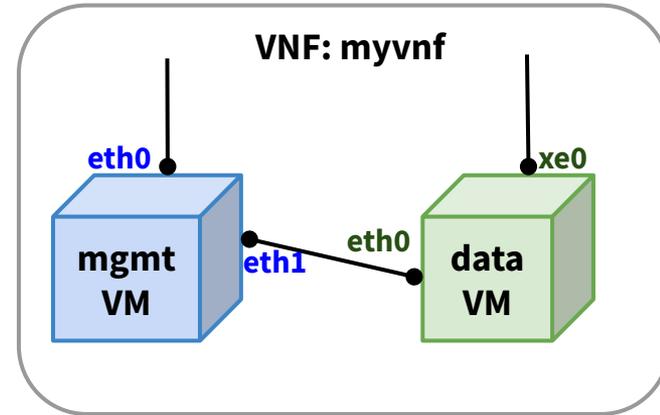
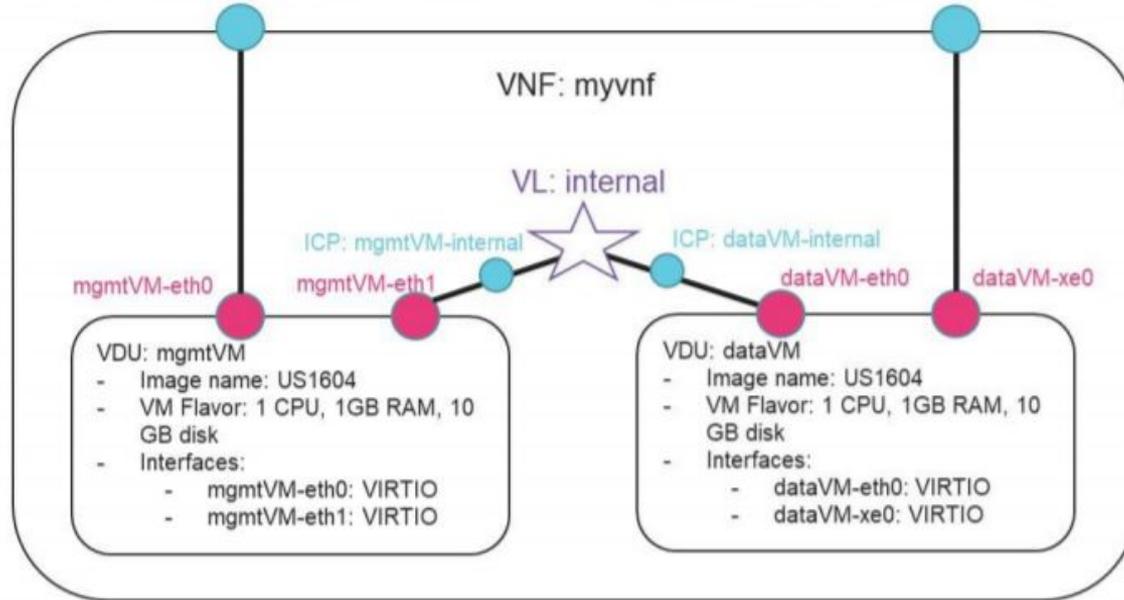
- **Virtual Link (VL)**
  - Es interno si conecta VDUs de una misma VNF y por tanto se define en el VNFD
  - Es externo si conecta VDUs entre distintas VNFs y por tanto se define en el NSD
- **Connection Point (CP)**
  - Extremos de un VL en cada VDU que conecta (puede ser una VDU o más)
    - Al igual que los VLs, pueden ser internos o externos
  - Se corresponden con los interfaces de red de una VDU (es decir, máquina virtual)
- **Onboarding**
  - Proceso de diseñar, crear y “subir” al catálogo de OSM un nuevo VNFD o NSD
- **Instantiation**
  - Proceso de instanciar (desplegar) un NSD
    - Un NSD puede estar instanciado de 0 a N veces
    - Cuando se instancia un NSD, se instancian las VNFDs que lo componen y se crean las máquinas virtuales y conexiones de red correspondientes

# Introducción - Virtual Network Function (VNF)

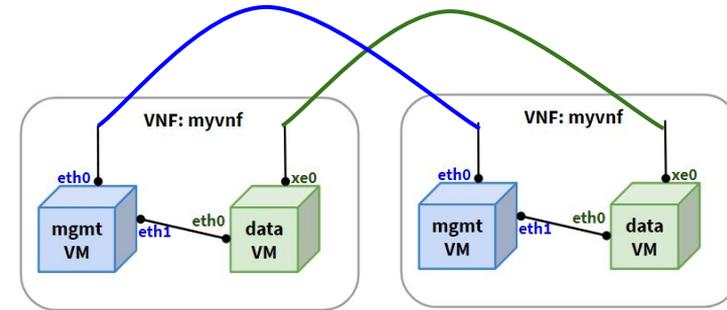
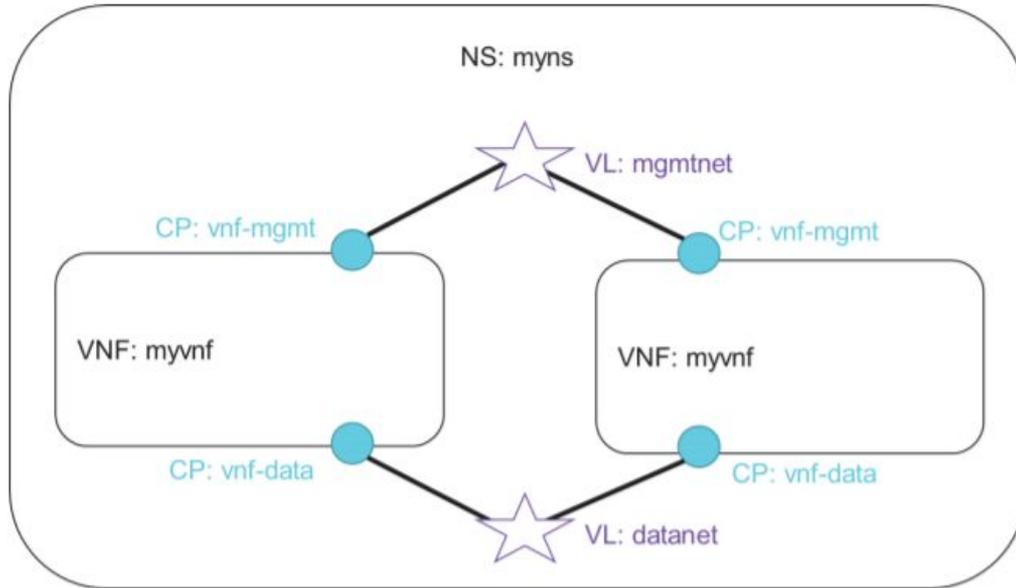


External Connection point: vnf-mgmt

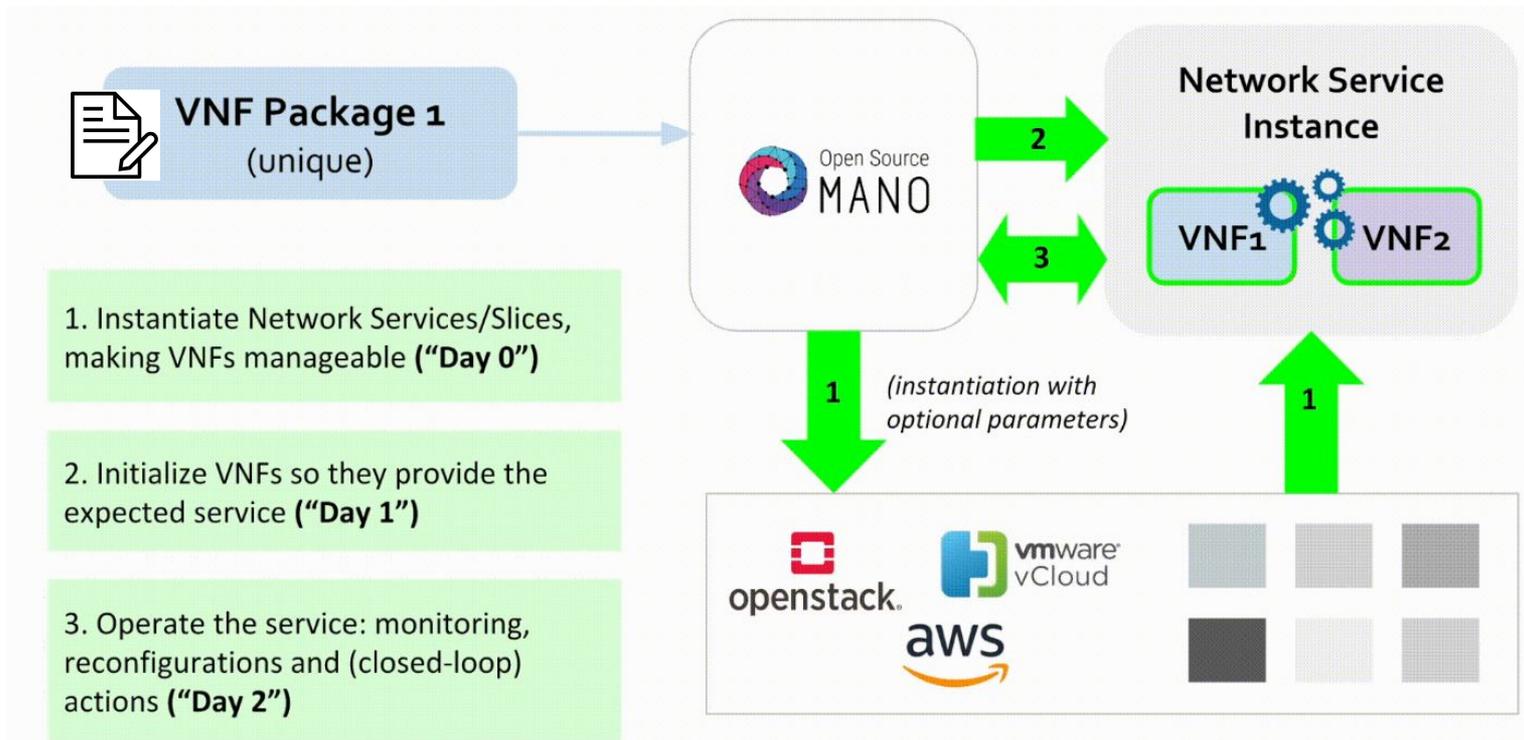
External Connection point: vnf-data



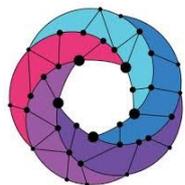
# Introducción - Network Service (NS)



# Introducción - Despliegue de servicios en OSM

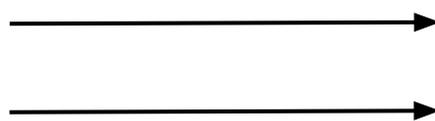


# Introducción - Despliegue de servicios en OSM

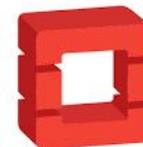


User A

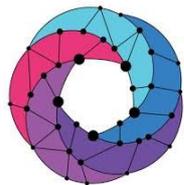
Project A.1  
...  
Project A.N



Project A.1 (Quota A.1)  
...  
Project A.N (Quota A.N)

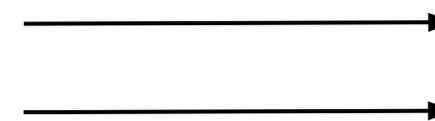


User A

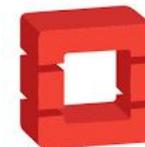


User B

Project B.1  
...  
Project B.N

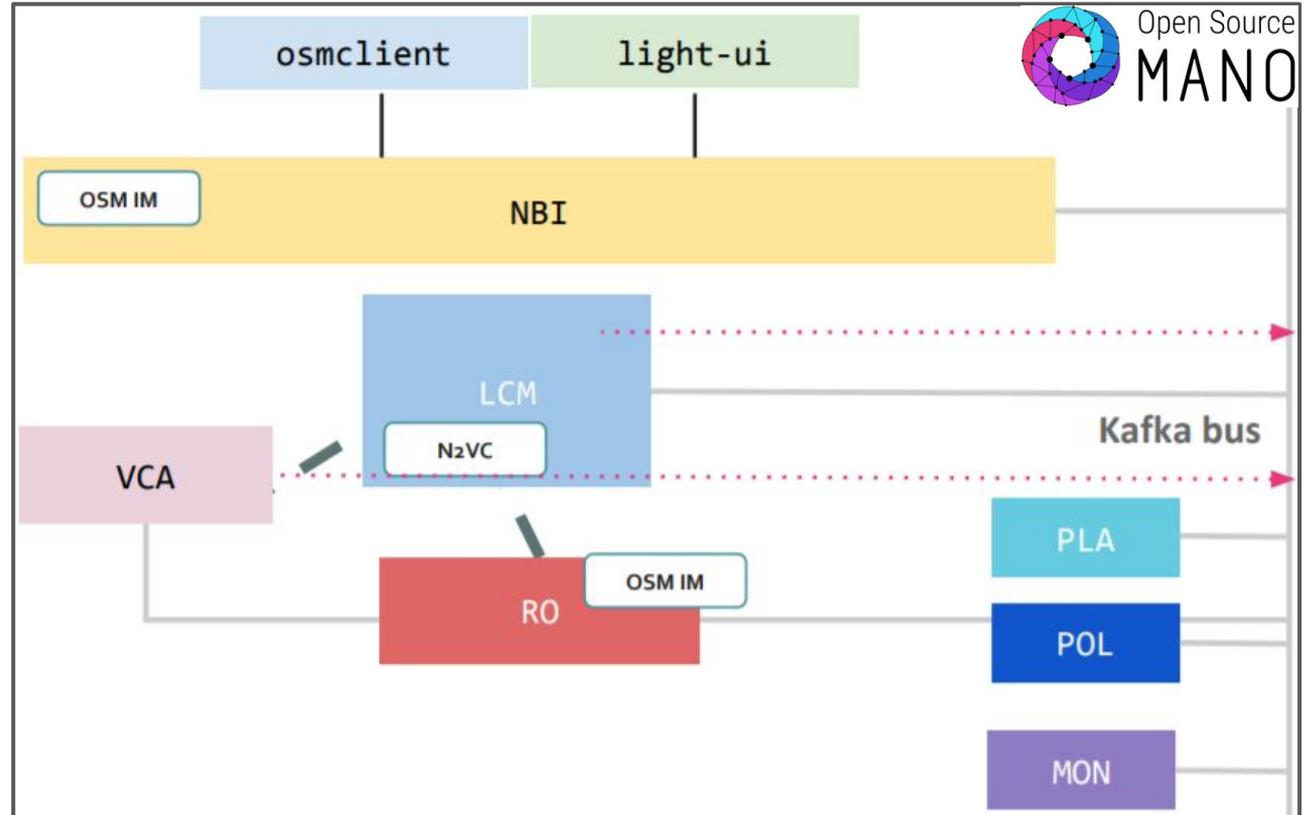


Project B.1 (Quota B.1)  
...  
Project B.N (Quota B.N)

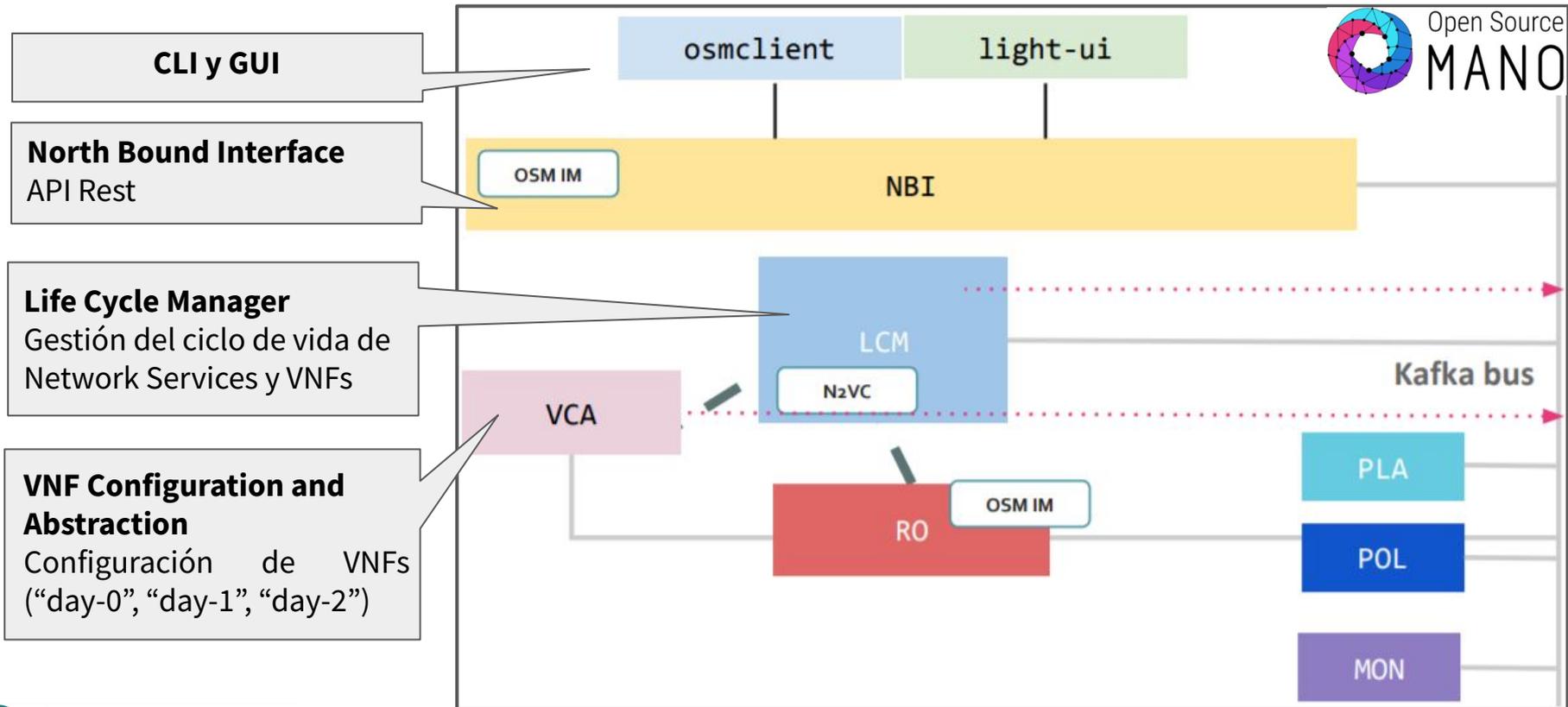


User B

# Introducción - Módulos principales de OSM



# Introducción - Módulos principales de OSM



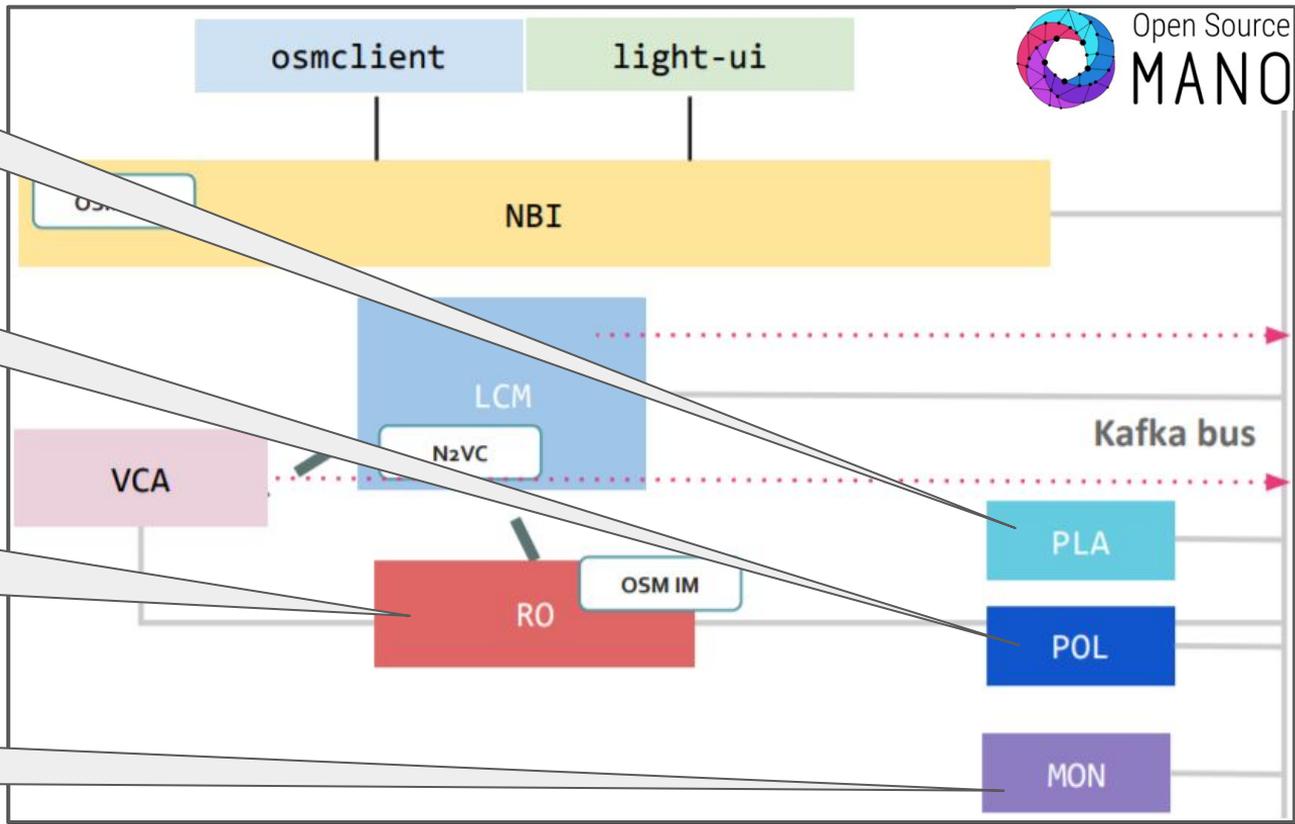
# Introducción - Módulos principales de OSM

**Placement**  
 Asignación óptima de VNFs a VIMs

**Policy**  
 Auto-escalado de VNFs en base a métricas de MON

**Resource Orchestrator**  
 Gestión de recursos de NFVI. Comunicación con VIMs, WIMs, SDN controllers, etc.

**Monitoring**  
 Monitorización de VNFs a través de métricas y alarmas



# Instalación y configuración - OSM

- Arrancar la VPN y conectarse por SSH a la máquina virtual asignada
  - Dirección IP: 10.98.1.22 (#1), 10.98.1.11 (#2), 10.98.1.21 (#3), 10.98.1.25 (#4), 10.98.1.14 (#5), 10.98.1.15 (#6), 10.98.1.31 (#7), 10.98.1.10 (#8)
  - Usuario: ubuntu | Password: OSM8DEMO
- Para instalar OSM8, solo hace falta ejecutar los siguientes comandos
  - `sudo apt update -y`
  - `sudo apt upgrade -y` (dar a “Keep the local version currently installed” si os aparece el menú)
  - `wget https://osm-download.etsi.org/ftp/osm-8.0-eight/install\_osm.sh`
  - `chmod +x install_osm.sh`
  - `./install_osm.sh -c k8s --k8s_monitor` (introducir “Y” cuando os pida proceder, luego esperar...)
- Comprobar que OSM8 se ha instalado correctamente accediendo desde el portátil a <http://10.98.1.X> (cada uno la dirección IP de vuestra máquina virtual)
  - User: admin | Password: admin

# Instalación y configuración - OSM

The screenshot displays the Open Source MANO (OSM) web interface. On the left is a dark sidebar with navigation options: Home, Overview (selected), Packages, NS Packages, VNF Packages, NetSlice Templates, Instances, NS Instances, VNF Instances, PDU Instances, NetSlice Instances, SDN Controllers, VIM Accounts, K8s, K8s Clusters, and K8s Repos. The main content area shows the 'admin' project overview. At the top right, there is a user menu for 'admin' and a breadcrumb trail: Home > Projects > admin. Below the breadcrumb is a table with project details:

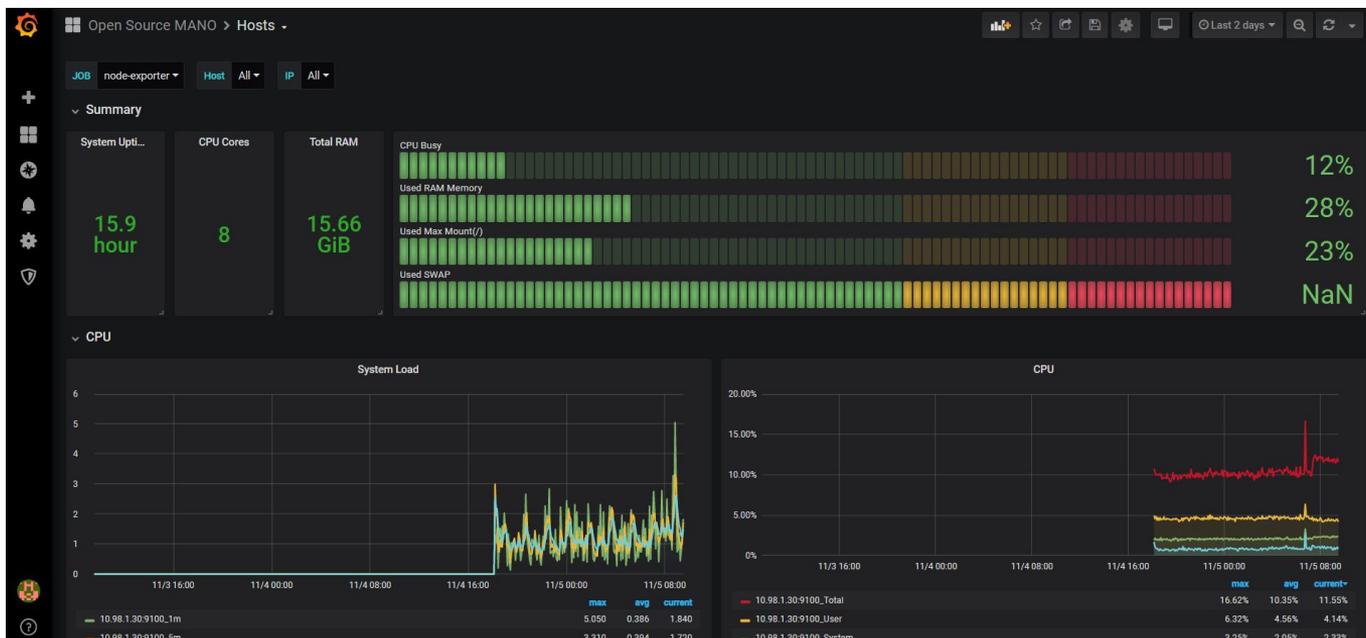
Project Name:	admin
Created:	04-11-2020 17:05:24
Modified:	04-11-2020 17:05:24

Below the table are four summary cards, each with a large '0' and an 'Open list' link with an arrow:

- NS packages (blue card)
- VNF packages (blue card)
- NS Instances (red card)
- VNF Instances (red card)

# Instalación y configuración - OSM

- Grafana para monitorización: <http://<ip>:3000> (admin/admin)



# Instalación y configuración - OpenStack

- Se utilizará un OpenStack ya instalado en el laboratorio, por lo que no hay que instalar nada adicional
  - Cada uno de vosotros tiene un usuario y proyecto propios
- Comprobar que tenéis acceso a dicho OpenStack
  - Con la VPN en marcha, acceder a <http://10.98.1.100/horizon>
  - Entrar con vuestro usuario
    - Domain: default
    - User Name: AlumnoN [N = 1-7]
    - Password: AlumnoN [N = 1-7]

# Instalación y configuración - OpenStack

ubuntu® Default • Jorge ▾ Jorge ▾

Project ^  
Compute ^  
Overview  
Instances  
Volumes  
Images  
Key Pairs  
API Access  
Network ▾  
Admin ▾  
Identity ▾

Project / Compute / Instances

## Instances

Instance ID =  Filter [Launch Instance](#)

Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
No items to display.										



# Labs

- Para realizar los laboratorios propuestos, vamos a trabajar tanto desde la GUI de OSM como desde la CLI. También se va a acceder a la GUI de OpenStack
  - GUI OSM: <http://10.98.1.X> (Usuario: admin, Password: admin)
  - CLI OSM: Hacer SSH a 10.98.1.X (Usuario: ubuntu, Password: OSM8DEMO)
    - `osm --help`
    - `osm <cmd> --help`
  - GUI OpenStack: <http://10.98.1.100/horizon> (Domain: default, Usuario: AlumnoN, Password: AlumnoN)
- En OSM, hay operaciones que se pueden hacer solo desde la CLI y operaciones que se pueden hacer tanto mediante la GUI y la CLI
  - Excepto para el diseño de servicios (Network Services y VNFs) con el editor gráfico de la GUI, es más rápido generalmente usar la CLI

# Lab 0 - Registro de VIM en OSM

- Antes de poder realizar despliegues de servicios, tenemos que indicarle a OSM donde desplegar los componentes (máquinas virtuales y enlaces virtuales) de dichos servicios
- En nuestro caso, tenemos que registrar la cuenta de OpenStack
  - Cada uno su propia cuenta con su usuario y contraseña
- Para ello hay que indicar dos cosas a OSM
  - URL (dirección y puerto) donde OpenStack va a responder a las peticiones de OSM
  - Credenciales de vuestro usuario y proyecto en OpenStack

# Lab 0 - Registro de VIM en OSM

- Mediante la GUI: VIM Accounts → New VIM
  - Name: Openstack
  - Type: Openstack
  - VIM URL: <http://supermicro-1:35357/v3/>
  - Tenant name: AlumnoN
  - Username: AlumnoN
  - Password: AlumnoN
- Mediante la CLI:
  - `osm vim-create --name Openstack --user AlumnoN --password AlumnoN --auth_url \`  
<http://supermicro-1:35357/v3/> `--tenant AlumnoN --account_type openstack`
- Ejecutar script `ro_container.sh` (`sudo sh ro_container.sh`)
  - Necesario para que el OSM sea capaz de alcanzar la URL del OpenStack
- Hacer lo mismo con `mon_container.sh` (`sudo sh mon_container.sh`)

# Lab 0 - Registro de VIM en OSM

Registered VIM + New VIM

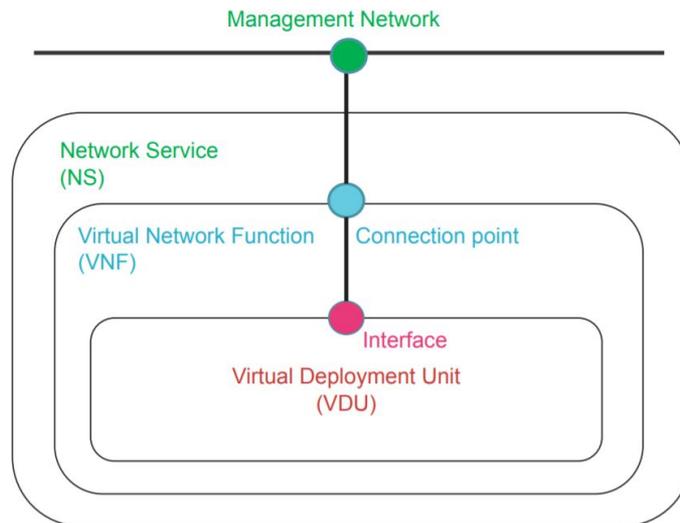
Show 10 entries Search:

Name	Identifier	Type	Operational State	Description	Actions
Openstack	6b92338f-40ec-4219-8b97-622896aadf95	openstack	ENABLED		<a href="#">i</a> <a href="#">🗑️</a>

Showing 1 to 1 of 1 entries Previous **1** Next

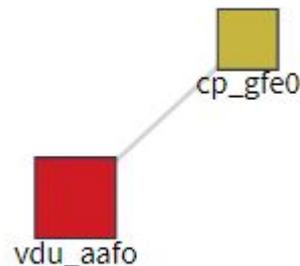
# Lab 1 - Servicio simple

- En primer lugar, se va a diseñar un NS muy sencillo, consistente en una única VNF con una sola VDU
  - Primero se construye el VNFD (descriptor de la VNF) y luego el NSD (descriptor del NS)



# Lab 1 - Servicio simple - VNFD

- VNF Packages → Compose a new VNF
- Package name: simple\_vnf
- Arrastrar elementos del panel situado a la izquierda: una VDU y un CP
- Manteniendo pulsada la tecla *Mayus* hacer clic en el CP seguido de la VDU para asignar el CP a la VDU
- No hace falta ningún VL interno (IntVL), ya que solo existe una VDU
- Pasar al editor de texto (  )



# Lab 1 - Servicio simple - VNFD

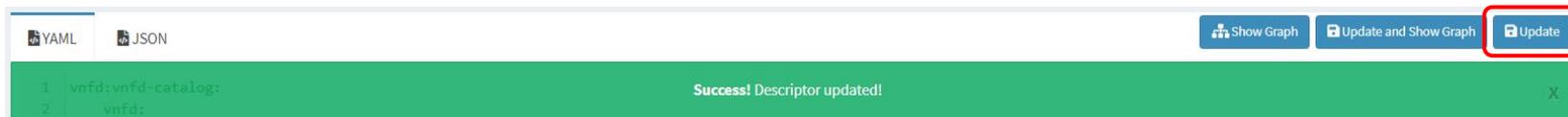
```
1 vnfd:vnfd-catalog:  
2   vnfd:  
3     - connection-point:  
4       - name: cp_gfe0  
5         type: VPORT  
6     description: ''  
7     id: simple_vnf  
8     internal-vld: []  
9     mgmt-interface:  
10      cp: ''  
11     name: simple_vnf  
12     short-name: simple_vnf  
13     vdu:  
14       - count: '1'  
15         description: ''  
16         id: vdu_aafo  
17         image: ubuntu  
18         interface:  
19           - external-connection-point-ref: cp_gfe0  
20             mgmt-interface: true  
21             name: eth_wrxn  
22             type: EXTERNAL  
23             virtual-interface:  
24               type: VIRTIO  
25         internal-connection-point: []  
26         monitoring-param: []  
27         name: vdu_aafo  
28         vm-flavor: {}  
29     version: '1.0'  
30
```

External CP

Descripción de la VDU

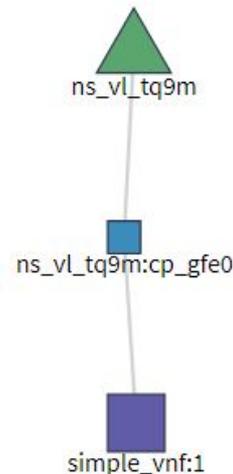
# Lab 1 - Servicio simple - VNFD

- Añadir el external CP como *mgmt-interface* (indicar interfaz de gestión SSH)  
`mgmt-interface:`  
`cp: 'cp_gfe0'` ¡Apuntar el nombre del CP!
- Editar flavor de la VDU (indicar los recursos que queremos asignar a la VDU)  
`vm-flavor:`  
`memory-mb: 64`  
`storage-gb: 1`  
`vcpu-count: 1`
- Editar imagen de sistema operativo de la VDU `image: cirros034`
- Guardar cambios



# Lab 1 - Servicio simple - NSD

- NS Packages → Compose a new NS
- Package name: simple\_ns
- Arrastrar simple\_vnf y un VL
- Manteniendo pulsada la tecla *Mayus* hacer clic en ambos elementos
  - Insertar vnfd-connection-point-ref: cp\_xxxx
- Pasar al editor de texto (  )
- Editar:
  - vim-network-name: provider-lab
- Guardar



# Lab 1 - Servicio simple - Instanciación

- Mediante la GUI: NS Packages → Instantiate NS (primer icono debajo de *Actions*)
  - Name: my\_simple\_ns
  - Description: my simple ns
  - NSD ID: simple\_ns
  - VIM Account ID: Openstack
- Mediante la CLI:
  - `osm ns-create --ns_name my_simple_ns --nsd_name simple_ns --vim_account Openstack`

# Lab 1 - Servicio simple - Instanciación

- Monitorizar progreso desde la GUI (En NS Instances)

NS Instances New NS

Show 10 entries Search:

Name	Identifier	Nsd name	Operational Status	Config Status	Detailed Status	Actions
my_simple_ns	e90c5c5b-0cbd-43bd-8956-0fc7e9884522	simple_ns	running	configured	Done	<span>i</span> <span>👤</span> <span>🗑️</span> <span>Actions</span>

Showing 1 to 1 of 1 entries Previous 1 Next

- Desde CLI
  - `osm ns-list`
  - `osm ns-show my_simple_ns`

# Lab 1 - Servicio simple - Instanciación

- En OpenStack: Project → Compute → Instances, comprobar que se ha desplegado la máquina virtual correspondiente a la única VDU del servicio

Project ^

Compute ^

Overview

**Instances**

Volumes

Images

Key Pairs

API Access

Network v

Admin v

Identity v

Project / Compute / Instances

## Instances

Instance ID = 
Filter
Launch Instance
Delete Instances
More Actions v

Displaying 1 item

<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	my_simple_ns-1-vdu_aafo-1	cirros034	10.98.1.17	vdu_aafo-flv	-	Active	nova	None	Running	1 minute	Create Snapshot v

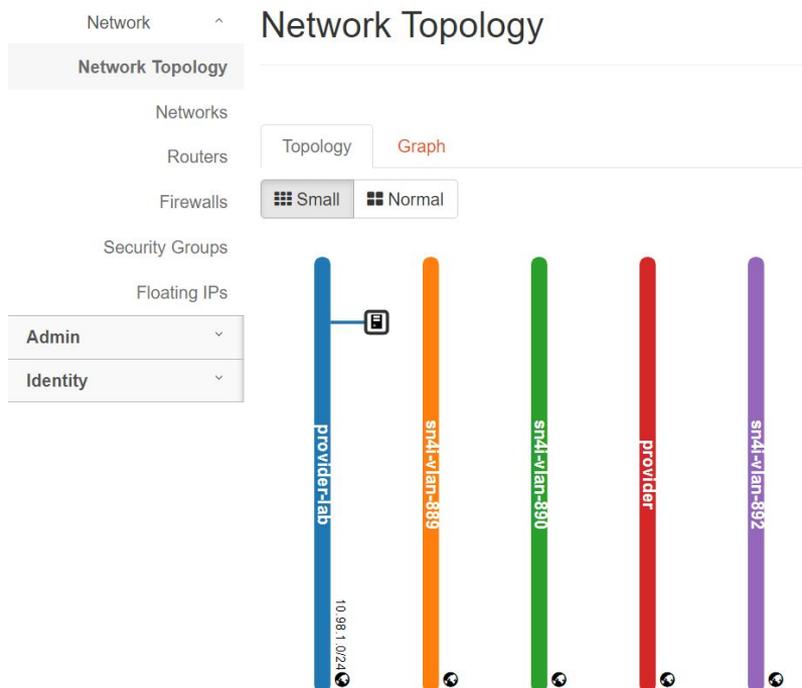
Displaying 1 item

# Lab 1 - Servicio simple - Instanciación

- Hacer ping a la dirección IP de la máquina virtual desplegada
  - No hay conectividad...
- La configuración por defecto del firewall que está asignada a vuestros usuarios no permite pasar el tráfico entrante a las máquinas virtuales
- En OpenStack: Project → Security Groups → Manage Rules → Add Rule
  - Rule: All ICMP, Direction: Ingress, CIDR: 0.0.0.0/23
  - Rule: All TCP, Direction: Ingress, CIDR: 10.98.0.0/23
  - Rule: All UDP, Direction: Ingress, CIDR: 10.98.0.0/23
- Volver a probar el ping
- Se puede entrar a la máquina por SSH también
  - Usuario: cirros, Password: gocubsgo

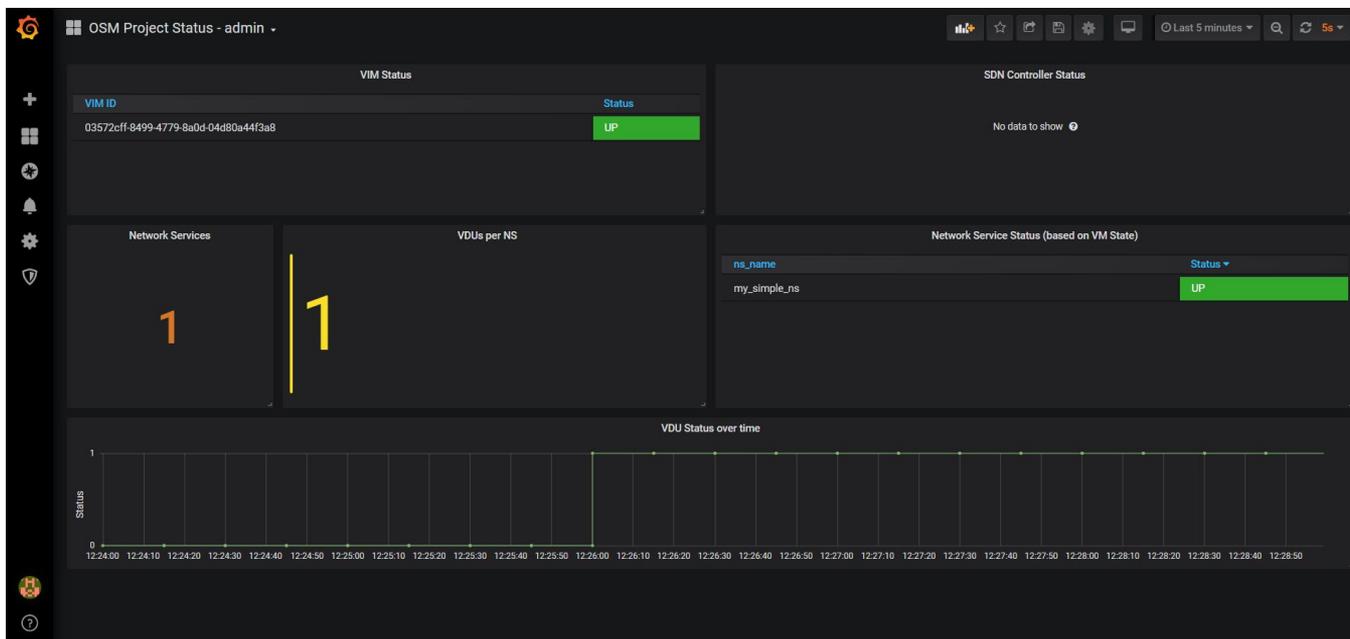
# Lab 1 - Servicio simple - Instanciación

- En OpenStack: Project → Network → Network Topology



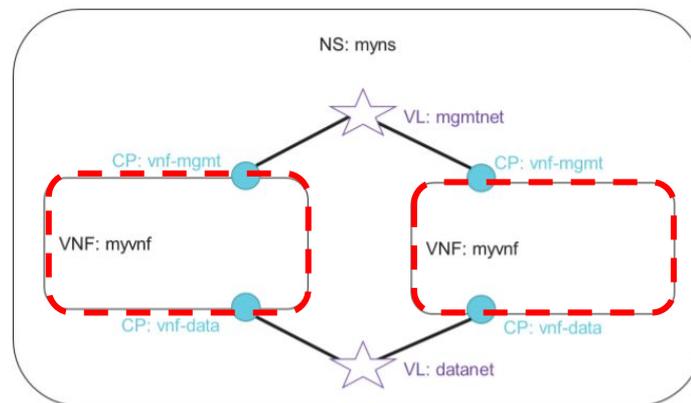
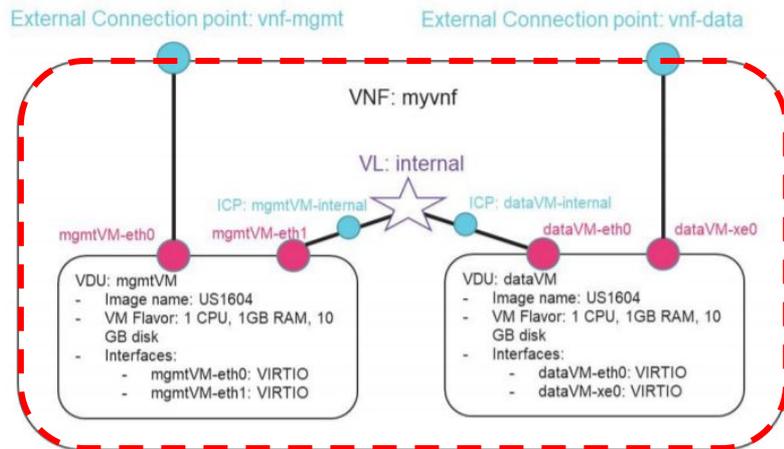
# Lab 1 - Servicio simple - Instanciación

- En Grafana (<http://10.98.1.X:3000>): clic en OSM Project Status - admin



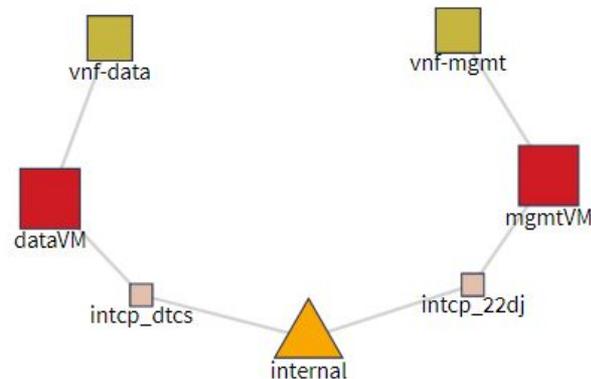
# Lab 2 - Servicio compuesto

- Como segundo ejercicio, se va a diseñar un Network Service más complejo, compuesto por dos VNFs idénticas, con dos VDUs cada una
  - En total, habrá cuatro VDUs, dos VLs externos, y dos VLs internos (uno por VNF)
  - Dado que ambas VNFs son idénticas, solo será necesario diseñar un VNFD, y referenciarlo dos veces en el NSD



# Lab 2 - Servicio compuesto - VNFD

- VNF Packages → Compose a new VNF
- Package name: multi\_vnf
- Arrastrar dos VDUs
  - Cambiar nombres e IDs: mgmtVM, dataVM
  - Cambiar imagen: cirros034
- Arrastrar dos CPs
  - Cambiar nombres: vnf-mgmt y vnf-data
- Click pulsando *Mayus* entre cada par VDU-CP (mgmt y data)
- Arrastrar un IntVL
  - Cambiar nombre e ID: internal
- Click pulsando *Mayus* entre IntVL y cada VDU
  - Se crearán dos CPs.



## Lab 2 - Servicio compuesto - VNFD

- Pasar al editor de texto
- Cambiar nombre de internal-connection-points (no se puede en editor gráfico)
  - mgmtVM-internal (líneas 13, 35, 41-43)
  - dataVM-internal (líneas 14, 59, 65-67)
  - int\_mgmtVM-internal (línea 36)
  - int\_dataVM-internal (línea 60)
- Editar flavors de las dos VDUs (igual que en Lab 1)
- Añadir mgmt-interface: cp: 'vnf-mgmt'
- Guardar cambios

# Lab 2 - Servicio compuesto - NSD

- NS Packages → Compose a new NS
- Package name: multi\_ns
- Arrastrar dos multi\_vnf y dos VL
  - Editar VLs:

Virtual Link SAVE

---

Vim network name

Name

Mgmt network

Type

Id

Virtual Link SAVE

---

Vim network name

Name

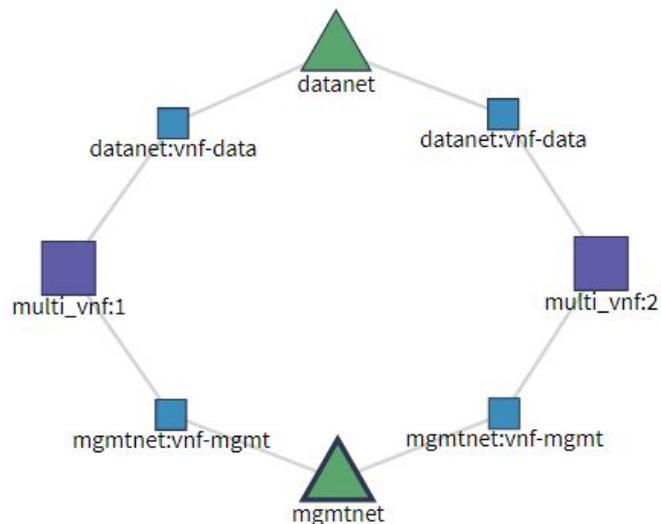
Mgmt network

Type

Id

## Lab 2 - Servicio compuesto - NSD

- Click pulsando *Mayus* entre cada par VDU-VL para crear CPs
  - vnf-mgmt y vnf-data
- Pasar a editor de texto y guardar cambios



# Lab 2 - Servicio compuesto - Instanciación

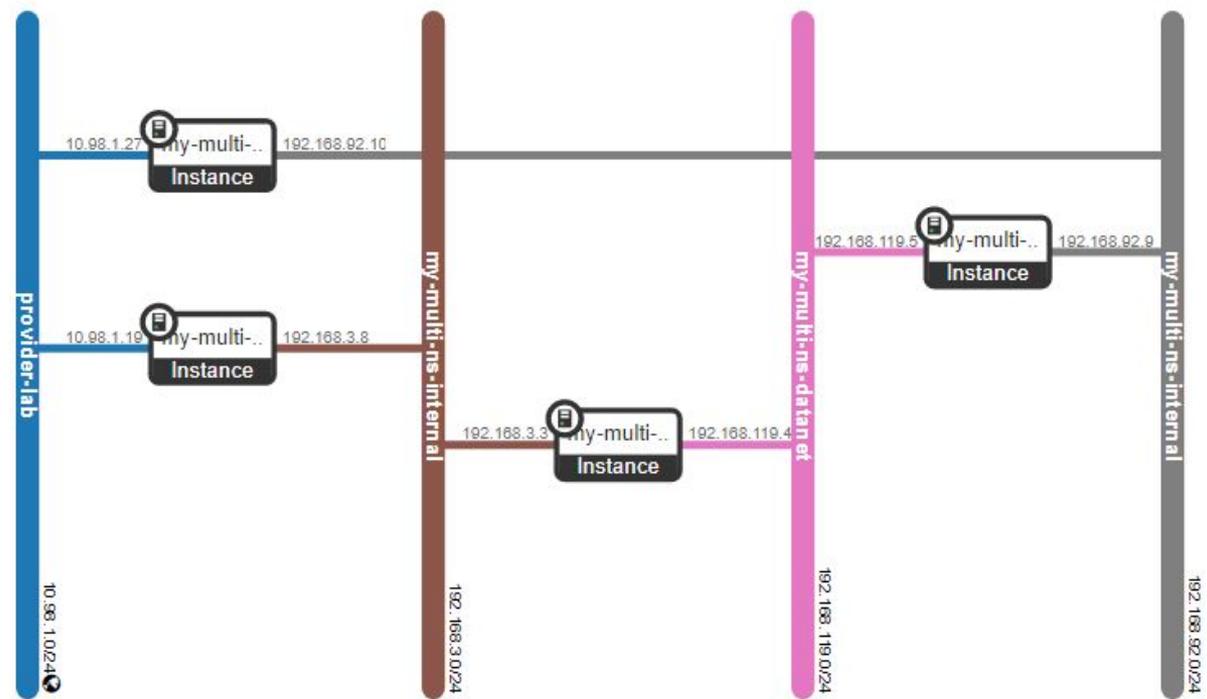
- Eliminar primero la instancia NS desplegada en Lab 1 (desde NS Instances)
- Desplegar NSD multi-ns

Displaying 4 items

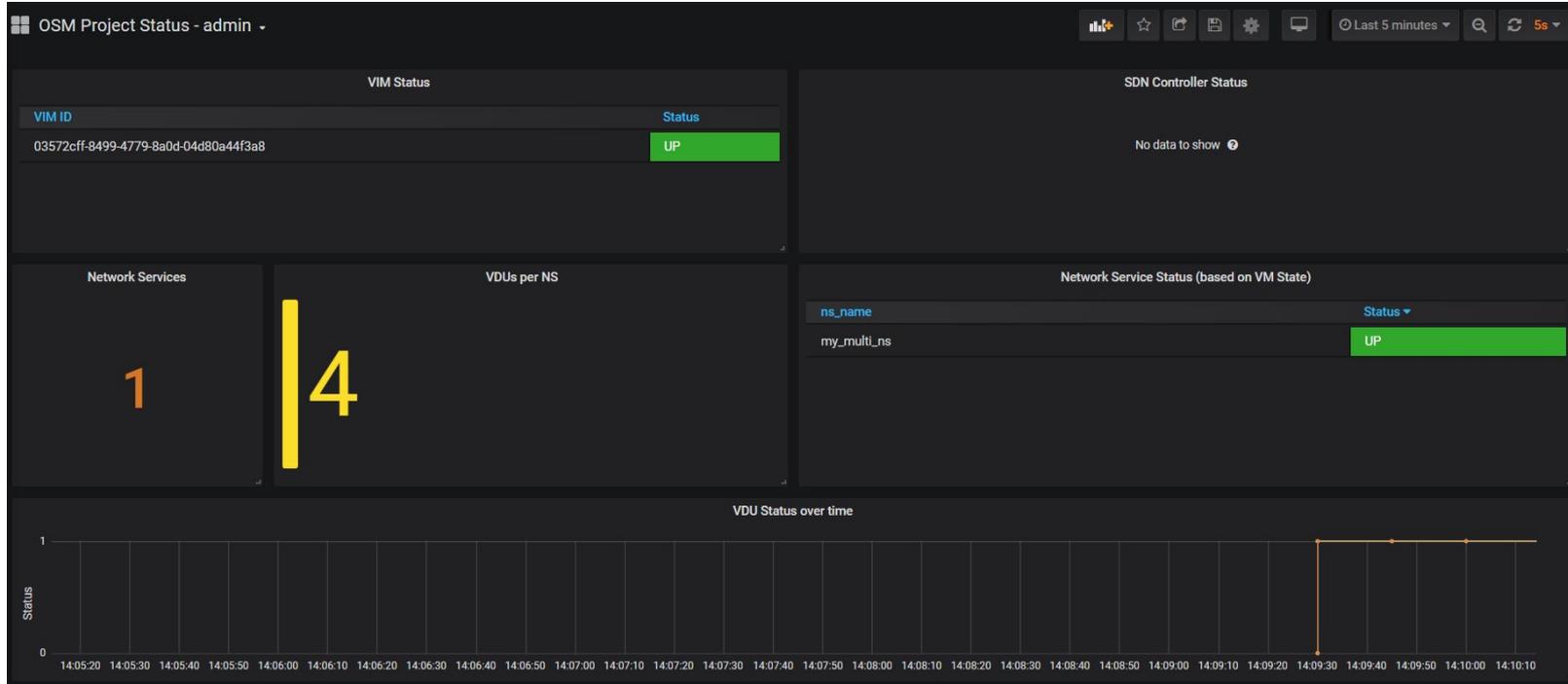
<input type="checkbox"/>	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
<input type="checkbox"/>	my-multi-ns-2-mgmtVM-1	cirros034	my-multi-ns-internal 192.168.92.10 provider-lab 10.98.1.27	dataVM-flv	-	Active	nova	None	Running	1 minute	Create Snapshot ▾
<input type="checkbox"/>	my-multi-ns-2-dataVM-1	cirros034	my-multi-ns-internal 192.168.92.9 my-multi-ns-datanet 192.168.119.5	dataVM-flv	-	Active	nova	None	Running	1 minute	Create Snapshot ▾
<input type="checkbox"/>	my-multi-ns-1-mgmtVM-1	cirros034	my-multi-ns-internal 192.168.3.8 provider-lab 10.98.1.19	dataVM-flv	-	Active	nova	None	Running	1 minute	Create Snapshot ▾
<input type="checkbox"/>	my-multi-ns-1-dataVM-1	cirros034	my-multi-ns-internal 192.168.3.3 my-multi-ns-datanet 192.168.119.4	dataVM-flv	-	Active	nova	None	Running	1 minute	Create Snapshot ▾



# Lab 2 - Servicio compuesto - Instanciación



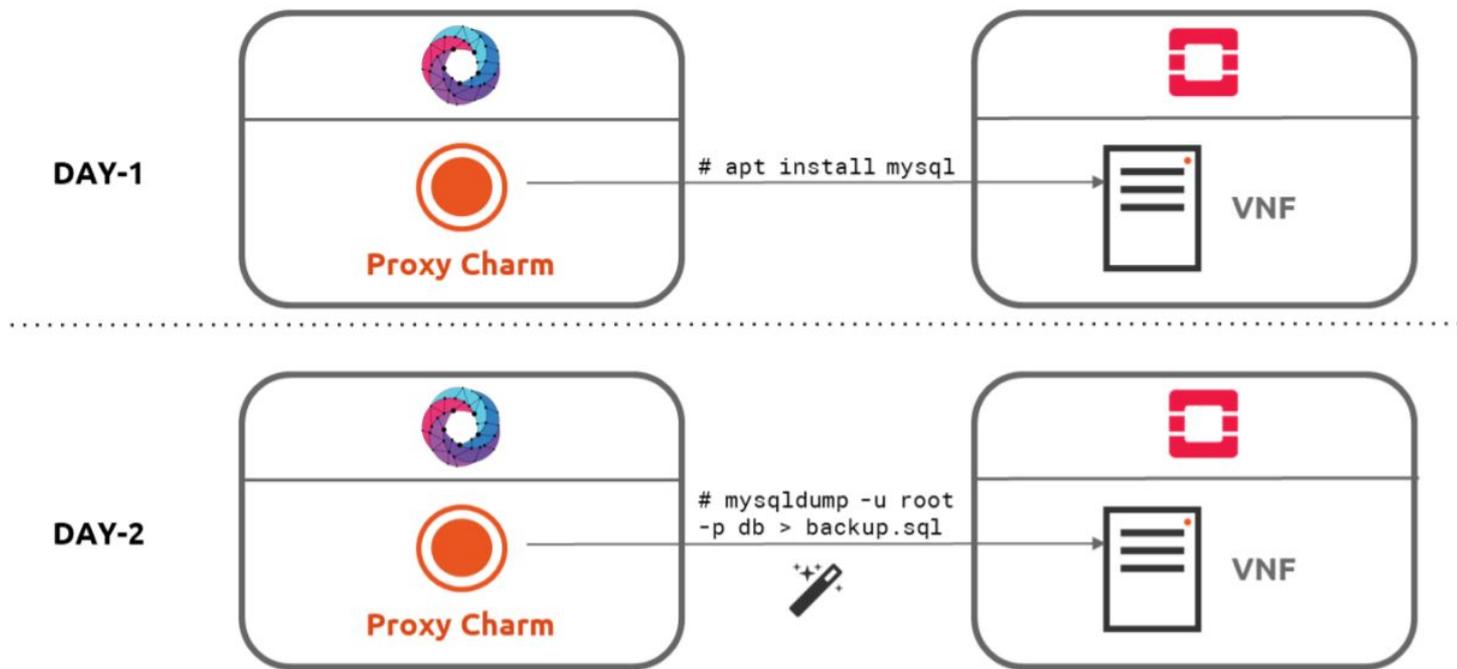
# Lab 2 - Servicio compuesto - Instanciación



# Lab 3 - Servicio configurable

- OSM permite realizar operaciones de day-0, day-1 y day-2 para automatizar la instalación inicial del servicio de la VNF y posteriormente gestionar dicho servicio
- Esto es posible mediante la herramienta **Juju**, diseñando lo que se conoce como Juju **charms**.
- Un charm es un conjunto de scripts que implementa **acciones** que pueden ser posteriormente ejecutadas por OSM sobre las VNFs
  - Ejemplo de acciones: añadir una ruta de red, instalar un paquete, arrancar o parar un servicio, editar un fichero de configuración o configurar un servicio, etc.
- El código de los charms se añade como parte del VNF Package (VNFP), y, además, se describen en el VNFD las acciones implementadas por los charms
  - Cada VNF tiene sus charms y sus acciones

# Lab 3 - Servicio configurable



# Lab 3 - Servicio configurable

- Para este ejercicio, se proporciona un VNFP y NSD basados en los de Lab 1 (1 VNF, 1 VDU), pero...
  - Con funcionalidades adicionales que permiten la configuración de la VNF
  - La imagen de S.O. es Ubuntu 18 en vez de Cirros
- Por seguridad, las imágenes cloud de Ubuntu no tienen contraseña por defecto
  - Hace falta clave SSH o configuración de day-0 → Cloud-init
    - Por ejemplo, crear contraseña para el usuario por defecto (ubuntu) y habilitar SSH con contraseña:

```
#cloud-config
password: OSM8DEMO
chpasswd: { expire: False }
ssh_pwauth: True
```

# Lab 3 - Servicio configurable

- El VNFP contiene un charm que implementa las siguientes acciones
  - Day-1
    - Instalar servidor apache (`sudo apt install apache2 -y`)
  - Day-2
    - Gestionar servidor apache (`sudo service apache2 <start/stop/etc>`)
    - Bloquear o permitir tráfico entrante de una dirección IP (mediante iptables)
    - Crear un fichero de texto con el nombre especificado (`touch <filename>`)
    - Ejecutar un comando bash arbitrario

# Lab 3 - Servicio configurable

- La forma más común de diseñar un NSD o un VNFP es construirlos externamente y después subirlos a nuestro catálogo de OSM
- En vuestra máquina virtual, podéis encontrar dos directorios con el contenido del NSD y del VNFP (apache\_ns y apache\_vnf)
- `osm nfpkg-create apache_vnf/`
  - Valida el VNFP, lo comprime, y lo sube a OSM
- `osm nspkg-create apache_ns/`
  - Valida el VNFP, lo comprime, y lo sube a OSM
- Comprobar que se han subido correctamente
  - `osm nsd-list` y `osm-vnfd-list` en CLI, o en NS Packages y VNF Packages en la GUI

# Lab 3 - Servicio configurable

- Instanciar el NS (tardará más que los anteriores...)
  - Nombre: my\_apache\_ns
- Esperar a que se termine de desplegar el NS e instalar el charm
  - Se crea un nuevo modelo de Juju por cada NS que tenga charms
    - `juju switch $(osm ns-list | grep my_apache_ns | awk '{print $4}')`
- Monitorizar progreso
  - `juju status`
  - `juju debug-log` (parar con Control+C)

# Lab 3 - Servicio configurable

- Acceder la `http://<ip>` (dirección IP de la máquina virtual desplegada) y comprobar que aparece la página por defecto de apache
  - Quiere decir que la acción de day-1 (instalar apache) se ha ejecutado correctamente
  - Ahora probar acciones de day-2
    - `osm ns-action my_apache_ns --vnf_name "1" --action_name manage-apache \ --params '{operation: "stop"}'`
      - Comprobar que ya no está el servidor apache escuchando
    - `osm ns-action my_apache_ns --vnf_name "1" --action_name manage-apache \ --params '{operation: "start"}'`
      - Comprobar que el servidor apache vuelve a estar en marcha
  - Ver historial de acciones: `osm ns-op-list my_apache_ns`

# Lab 3 - Servicio configurable

- Acceder la `http://<ip>` (dirección IP de la máquina virtual desplegada) y comprobar que aparece la página por defecto de apache
  - Quiere decir que la acción de day-1 (instalar apache) se ha ejecutado correctamente
  - Ahora probar acciones de day-2
    - `osm ns-action my_apache_ns --vnf_name "1" --action_name block-user \ --params '{user: "10.98.0.X"}'`
      - Comprobar que no podéis alcanzar el servidor, o hacer ping o SSH a la VM
    - `osm ns-action my_apache_ns --vnf_name "1" --action_name allow-user \ --params '{user: "10.98.0.X"}'`
      - Comprobar que volvéis a tener conectividad con la VM
  - Ver historial de acciones: `osm ns-op-list my_apache_ns`

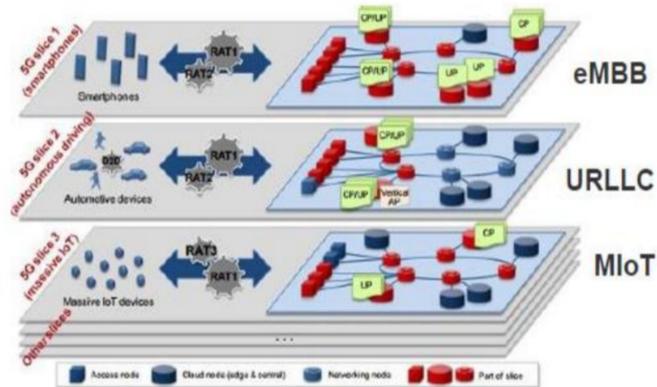
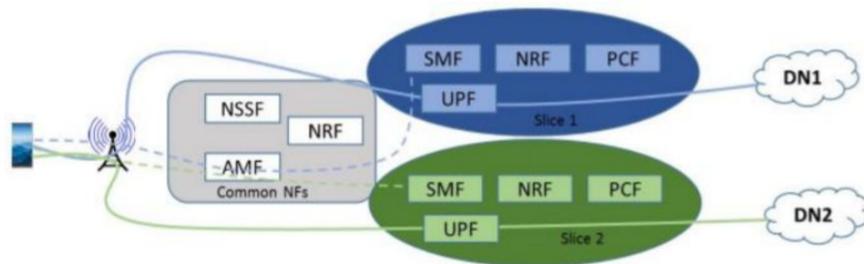
# Más posibilidades con OSM

- Monitorización de VNFs
  - Obtención de métricas de cada VM como uso de CPU, de memoria, de ancho de banda, número de usuarios/peticiones/operaciones por segundo, etc.
- Auto-escalado de VNFs
  - Automáticamente crear/destruir VDUs si las métricas recogidas alcanzan cierto umbral
- Despliegue de contenedores en vez de VMs sobre Kubernetes
- Además de VIMs, orquestación de WIMs (Wide-Area Infrastructure Managers) y SDNs
- Integración con OSS/BSS

# Más posibilidades con OSM

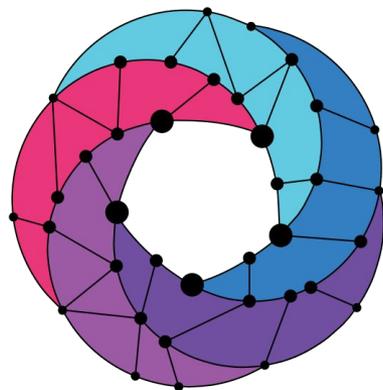
- Network Slicing

- In the scope of 5G, a Network Slice is a logical network that provides specific network capabilities and network characteristics, through a set of Network Function instances and the required resources (e.g. compute, storage and networking resources).
- Different network slices addressing different types of usage requiring different levels of functionality, performance and reliability.



# Documentación

- Documentación oficial
  - <https://osm.etsi.org/docs/user-guide>
  - <https://osm.etsi.org/docs/vnf-onboarding-guidelines>
  - <https://osm-download.etsi.org/ftp/Documentation/201902-osm-scope-white-paper>
- Material de la última OSM Hackfest
  - <https://osm-download.etsi.org/ftp/osm-8.0-eight/OSM-MR9-hackfest/presentations/>
  - [https://osm.etsi.org/wikipub/index.php/OSM9\\_Hackfest#OSM\\_Hackfest\\_Sessions](https://osm.etsi.org/wikipub/index.php/OSM9_Hackfest#OSM_Hackfest_Sessions)



# Open Source MANO

Eduardo.Jacob@ehu.eus  
jorge.sasiain@ehu.eus  
david.franco@ehu.eus



eman ta zabal zazu



Universidad  
del País Vasco

Euskal Herriko  
Unibertsitatea