Redes en Proxmox I

Redes en nodos Proxmox y SDN para MV y contenedores

1.- Introducción a las Redes en Proxmox VE 8.2

1.1.- Gestión de Redes en Proxmox

Proxmox VE nos ofrece, de una manera muy sencilla, la gestión de las redes con las que va a trabajar. Esta gestión la podemos hacer en dos niveles:

- Podemos configurar **la configuración de red del servidor Proxmox VE** para determinar el tipo de conexión que tendrá el servidor con el exterior.
- Podemos configurar **la configuración de red que tendrán las máquinas virtuales y contenedores** que gestionemos en nuestro servidor Proxmox.

Antes de estudiar detenidamente cada una de estos niveles, vamos a introducir un concepto de redes con el que vamos a trabajar: **un puente o bridge/switch** es un dispositivo de interconexión de redes. Un **virtual network switch o bridge** es equivalente a un switch físico con la diferencia de que un Linux Bridge posee un número ilimitado de puertos virtuales. Podemos conectar MV y contenedores a estos puertos virtuales. Del mismo modo que un switch físico, el Linux Bridge aprende direcciones MAC de paquetes recibidos y los guarda en una MAC table, la cual usa para tomar decisiones de forwarding de tramas.

En Proxmox ya vienen por defecto instalados los paquetes necesarios para utilizar Linux Bridge, pero aun así vamos a indicar los necesarios por si en algún momento se corrompe el sistema:

```
apt install -y bridge-utils
```

Con el paquete de administración opcional de red **ifupdown2** , también puede volver a cargar la configuración de red en vivo, sin necesidad de reiniciar. Si instaló Proxmox VE sobre Debian o actualizó a Proxmox VE 7.0 desde una instalación anterior de Proxmox, asegúrate de que ifupdown2 está instalado:

```
apt install ifupdown2
```

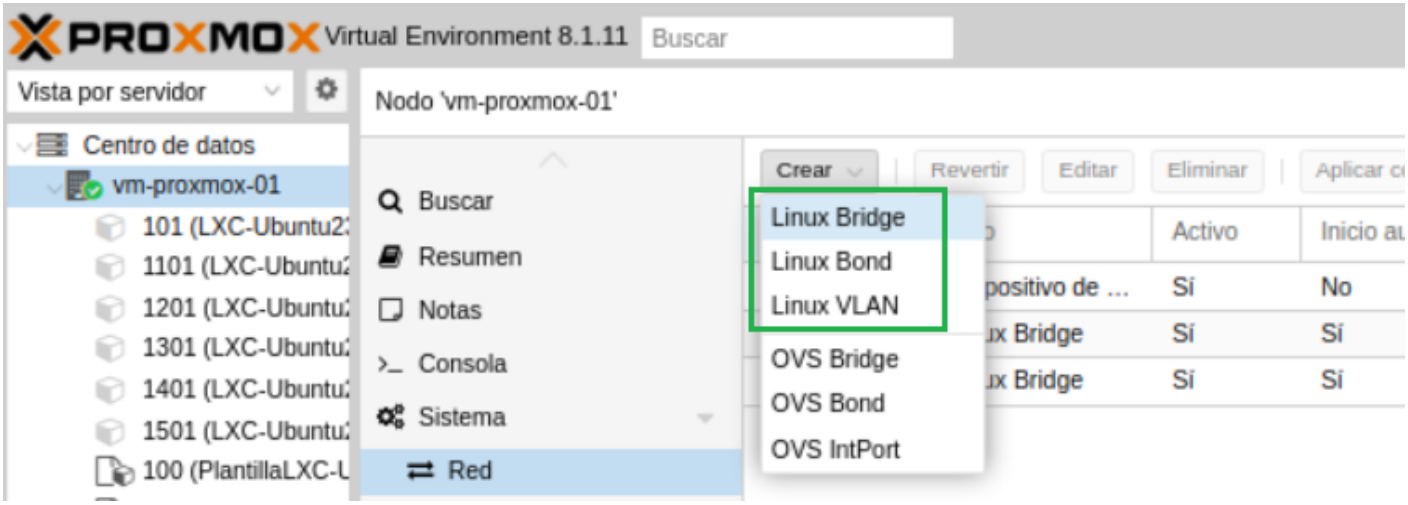


Imagen de elaboración propia *Crear una configuración de red para las MV y contenedores de Proxmox* (CC BY-NC-SA)

Tenemos más opciones para implementar un bridge (puente) por software, pero la más fácil de usar y la que vamos a usar nosotros en Proxmox VE es **Linux Bridge**.

> Por su simplicidad y al estar **integrado en el Kernel de Linux**, son las rozones principales por la que se utiliza Linux Bridge en Proxmox por defecto. Linux Bridge ha incluido soporte básico para **STP (Spanning Tree Protocol)**, multicast (multidifusión) y Netfilter desde las series de kernel 2.4 y 2.6 de Linux.

1.2.- Configuración de la red con Linux Bridge

La configuración de la red se puede realizar a través de la GUI o editando manualmente el archivo **/etc/network/interfaces**, que contiene toda la configuración de la red. Es preferible utilizar la GUI para evitar errores. Una vez que la red está configurada, puede usar las herramientas tradicionales de Debian `ifup` e `ifdown` para subir y bajar las interfaces.

Aplicar cambios de red

Proxmox VE no escribe los cambios directamente en **/etc/network/interfaces**, lo hace en un archivo temporal llamado **/etc/network/interfaces.new**, de esta manera puede hacer muchos cambios relacionados a la vez. Esto también permite asegurarse de que los cambios sean correctos antes de aplicar, ya que una configuración de red incorrecta puede hacer que un nodo sea inaccesible.

Convenciones de nombres

Actualmente Proxmox utiliza las siguientes convenciones de nomenclatura para nombres de dispositivos:

- **Dispositivos Ethernet**: **en***, nombres de interfaz de red systemd. Este esquema de nomenclatura se utiliza para las nuevas instalaciones de Proxmox VE desde la versión 5.0. (Dispositivos Ethernet: **eth**[N], donde N>=0 (eth0, eth1, ...) Este esquema de nomenclatura se utiliza para los hosts Proxmox VE que se instalaron antes de la versión 5.0. Al actualizar a 5.0, los nombres se mantienen tal cual.)
- **Nombres de puente**: **vmbr[N],** donde 0≤N≤4094 (vmbr0 - vmbr4094)
- **Bonds: bond[N]**, donde N>=0 (bond0, bond1, ...)
- **VLAN:** simplemente agregue el número de VLAN al nombre del dispositivo, separado por un punto (eno1.50, bond1.30)

Elegir una configuración de red

Dependiendo de su organización de red actual y sus recursos, puede elegir una configuración de red **puenteada(bridged)**, **enrutada(routed)** o **enmascarada(masquerading).**

- **Servidor Proxmox VE en una LAN privada, utilizando una puerta de enlace externa para llegar a Internet.**
  - El modelo Bridged es el modo predeterminado en las nuevas instalaciones de Proxmox VE. Cada una de las MV o contenedores (Guest) tendrá una interfaz virtual conectada al puente Proxmox VE. Esto es similar en efecto a tener la tarjeta de red de la MV o del contenedor conectada directamente a un nuevo conmutador en su LAN, el puente de Proxmox VE (vmbr0) desempeña el papel del **switch**.
- **Servidor Proxmox VE como proveedor de alojamiento, con rangos de IP pública para invitados.**
  - Para esta configuración, puede usar un modelo en puente o enrutado, según lo que permita su proveedor.
- **Servidor Proxmox VE como proveedor de alojamiento, con una sola dirección IP pública.**
  - En ese caso, la única forma de obtener accesos de red salientes para sus sistemas invitados es usar Masquerading (**NAT**). Para el acceso de red entrante a sus invitados, deberá configurar el reenvío de puertos.

Configuración predeterminada usando un puente (bridged)

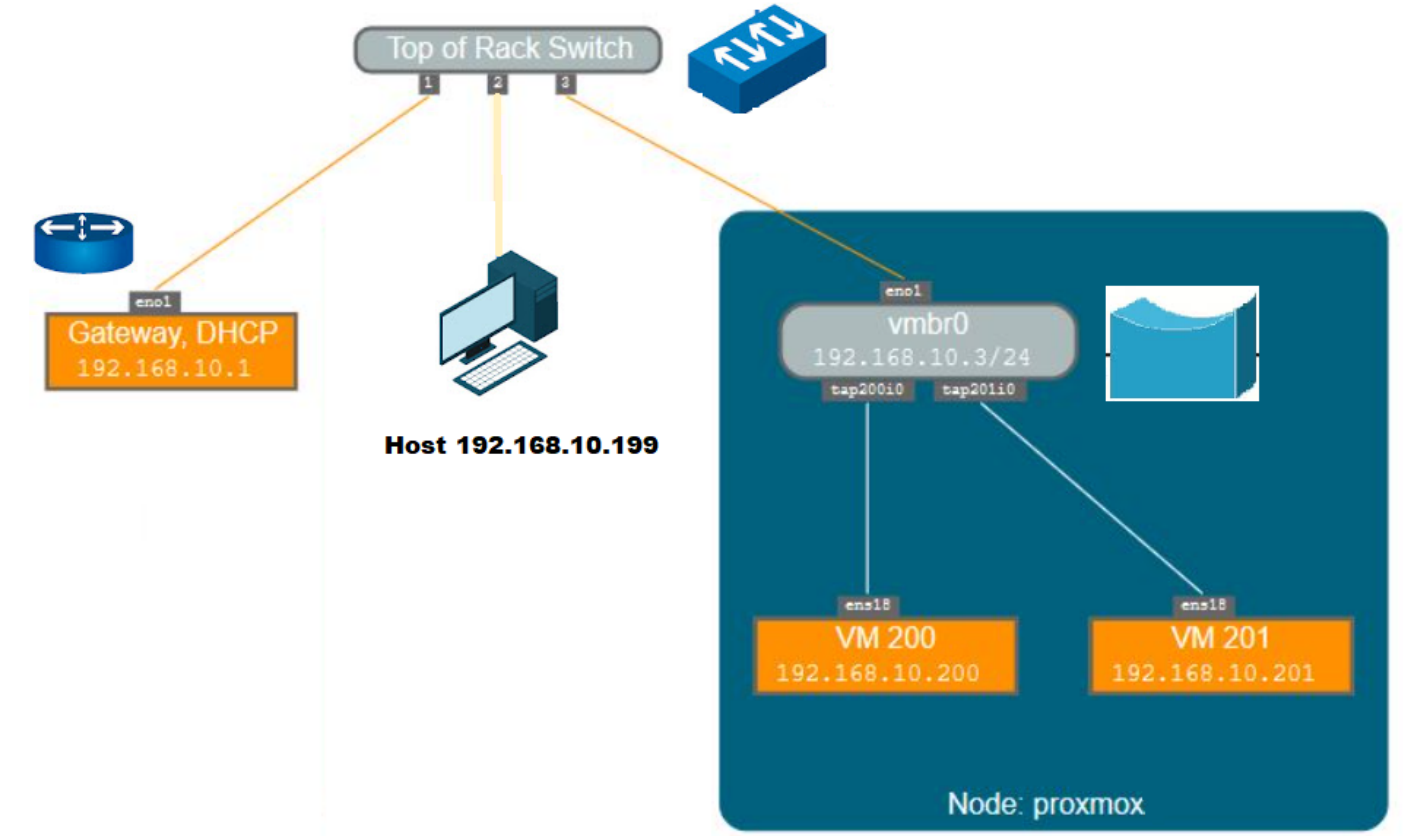


Imagen de elaboración propia. *Configuración Linux Bridge (predeterminada)* (CC BY-NC-SA)

Los puentes son como los **switches** de red físicos implementados en software. Todas las máquinas virtuales pueden compartir un solo puente, o puede crear múltiples puentes para separar dominios de red. Cada nodo de Proxmox puede tener hasta 4094 puentes, como máximo.

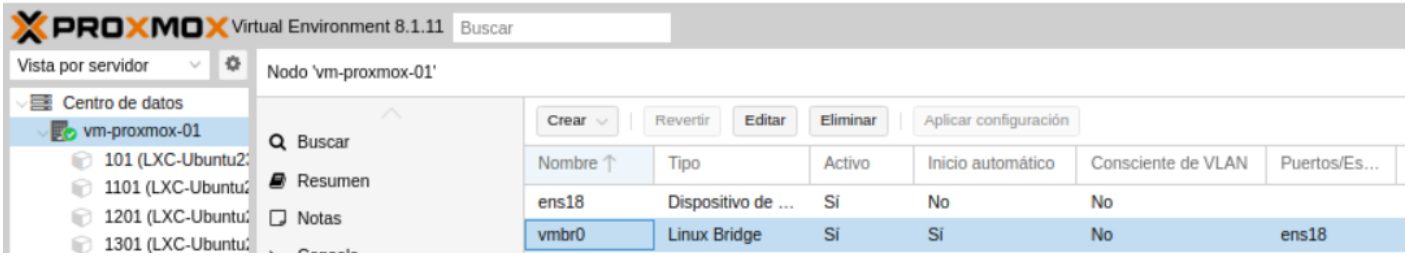


Imagen de elaboración propia. *El Linux Bridge "vmbr0" haciendo de puente con la interfaz de red "ens18"* (CC BY-NC-SA)

El programa de instalación crea un único puente llamado vmbr0, que está conectado a la primera tarjeta Ethernet de las 4 existentes en el nodo Proxmox. La configuración correspondiente en **/etc/network/interfaces** podría ser:

```
auto lo
iface lo inet loopback
iface eno1 inet manual
```

```
iface eno2 inet manual
iface eno3 inet manual
iface eno4 inet manual

auto vmbr0
iface vmbr0 inet static
address  192.168.10.3
netmask  255.255.0.0
gateway  192.168.10.1
bridge-ports eno1
bridge-stp off
bridge-fd 0
```

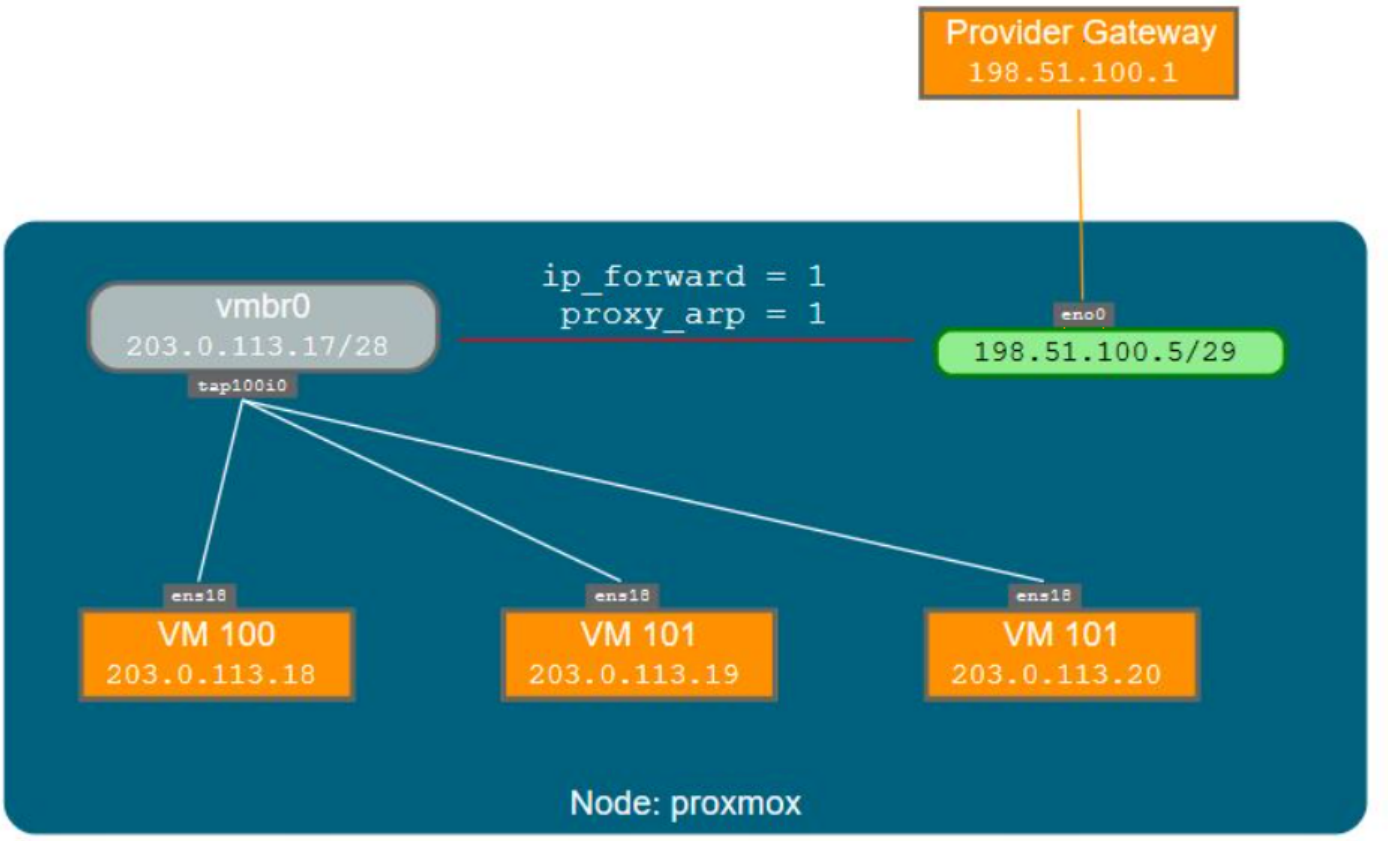
> **Las máquinas virtuales se comportan como si estuvieran conectadas directamente a la red física**. La red, a su vez, considera que cada máquina virtual tiene su propia MAC, a pesar de que sólo hay un cable de red que conecta todas estas máquinas virtuales a la red.

Aunque no es lo habitual podemos **asignar una IP mediante DHCP**:

```
auto lo
iface lo inet loopback
iface ens18 inet manual
auto vmbr0
iface vmbr0 inet dhcp
    bridge_ports ens18
    bridge_stp off
    bridge_fd 0
```

> Es útil para crear **MV de Proxmox anidadas** (un Proxmox virtualizando dentro de otro Proxmox)

Configuración enrutada



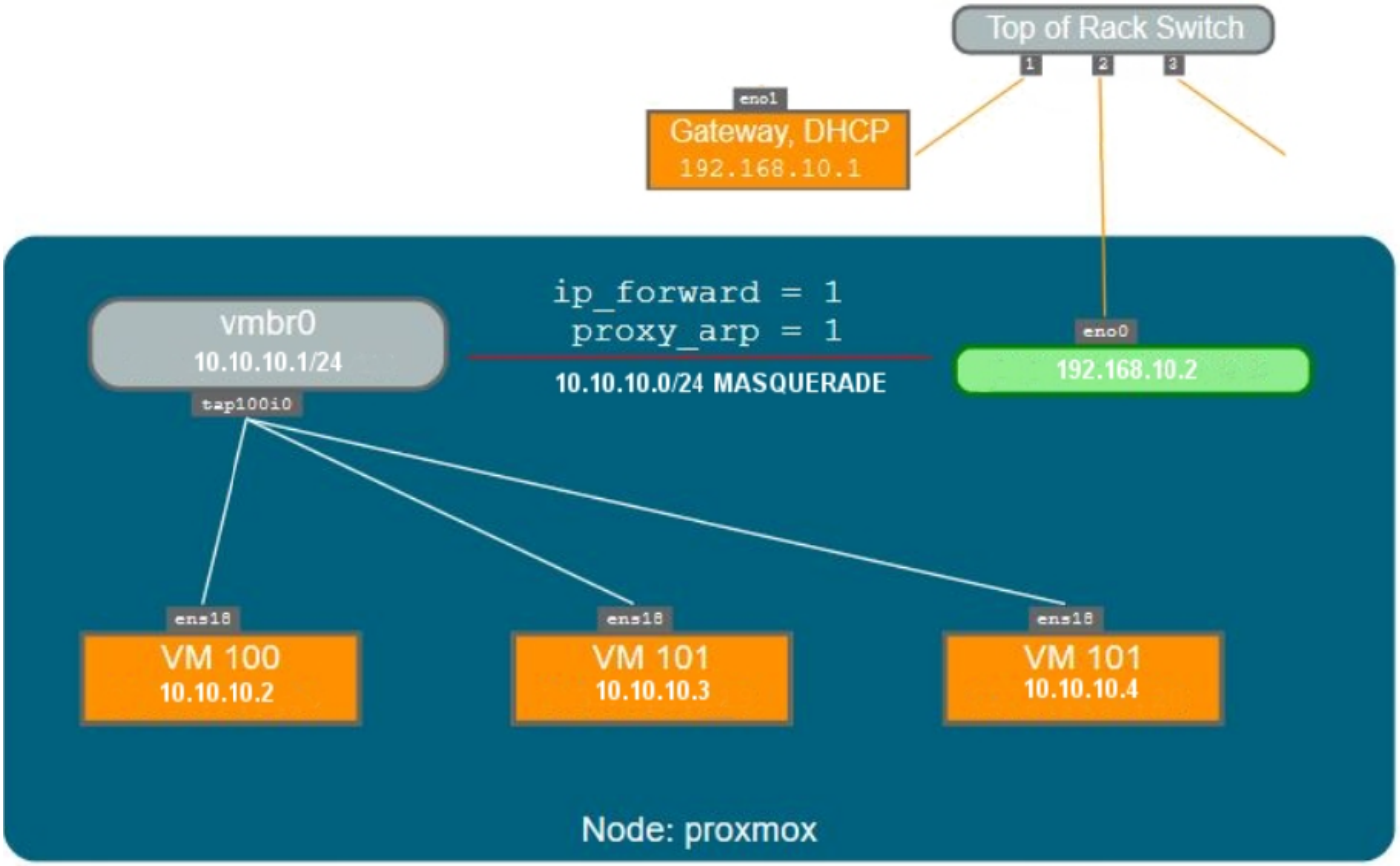
La mayoría de los proveedores de internet no admiten múltiples direcciones MAC en una sola interfaz. Por razones de seguridad, deshabilitan las redes tan pronto como detectan.
Podemos evitar el problema enrutando todo el tráfico a través de una única interfaz. Esto asegura que todos los paquetes de red usen la misma dirección MAC.

Un escenario común es tener una IP pública (198.51.100.5) y un bloque de IPs para sus máquinas virtuales (203.0.113.16/29).

```
auto lo
iface lo inet loopback

auto eno1
iface eno1 inet static
address 198.51.100.5
netmask 255.255.255.0
gateway 198.51.100.1
```

```
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up echo 1 > /proc/sys/net/ipv4/conf/eno1/proxy_arp

auto vmbr0
iface vmbr0 inet static
address 203.0.113.17
netmask 255.255.255.248
bridge_ports none
bridge_stp off
bridge_fd 0
```

Configuración Masquerading (NAT) con iptables



El enmascaramiento permite a las MV y contenedores, que sólo tienen una dirección IP privada, acceder a la red pública utilizando la dirección IP del nodo Proxmox para el tráfico saliente. Cada paquete saliente es reescrito por iptables para aparecer como originario del host, y las respuestas se reescriben en consecuencia para ser enrutadas al remitente original.

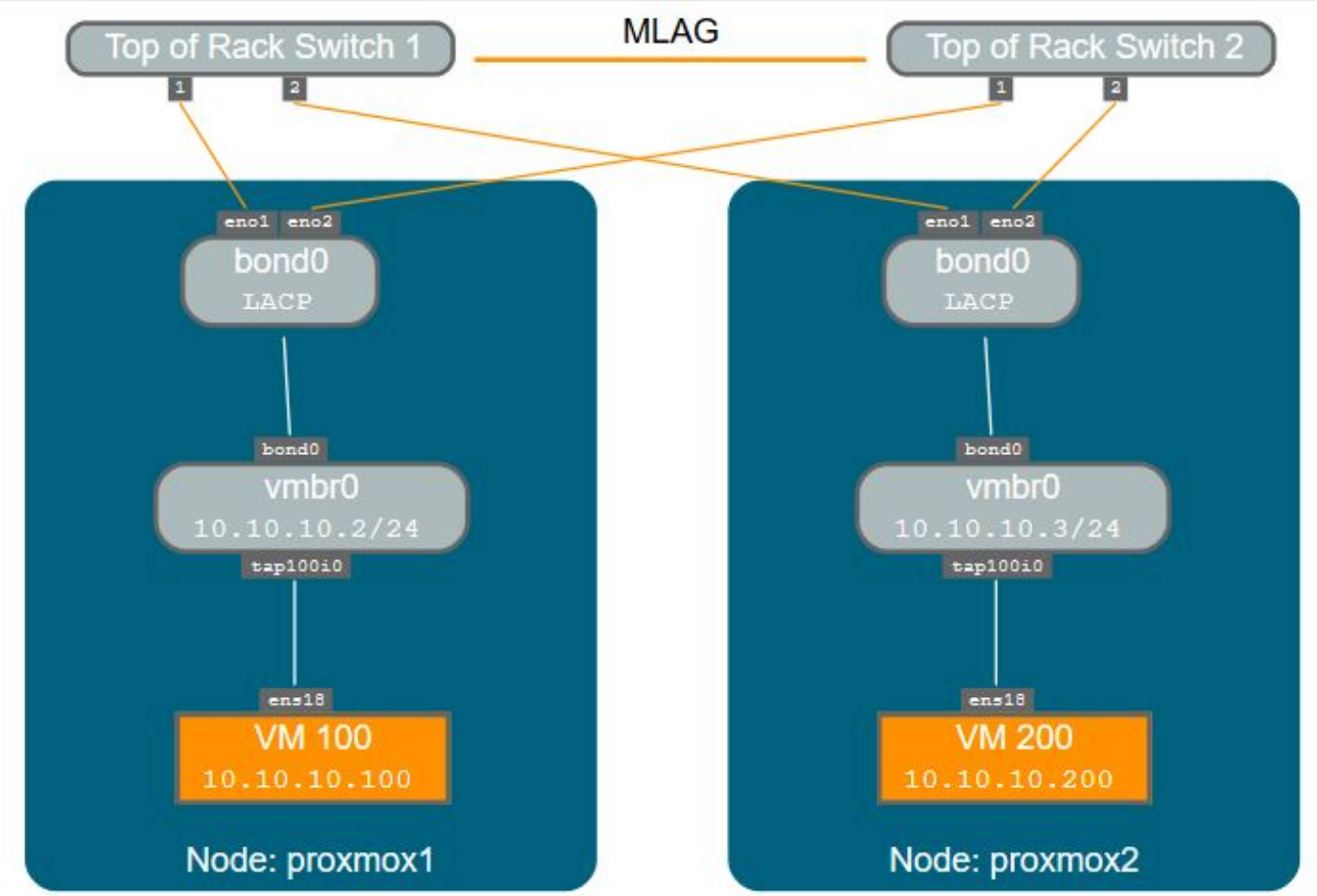
```
auto lo
iface lo inet loopback
auto eno0

iface eno1 inet static
Address 192.168.10.2
netmask 255.255.255.0
Gateway 192.168.10.1

auto vmbr0
#private sub network
iface vmbr0 inet static
address 10.10.10.1
netmask 255.255.255.0
bridge_ports none
bridge_stp off
bridge_fd 0

post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -s '10.10.10.0/24' -o eno1 -j MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '10.10.10.0/24' -o eno1 -j MASQUERADE
```

Configuración Linux Bond (o modo balanceador de red)

Bonding (también llamado agrupación de NIC o **agregación de enlaces**) es una técnica para vincular varias NIC a un sólo dispositivo de red. Es posible lograr diferentes objetivos, como hacer

que la red sea tolerante a fallos, aumentar el rendimiento (ancho de banda) o ambos juntos.

El hardware de alta velocidad como Fibre Channel y el hardware de conmutación asociado pueden ser bastante caros. Al hacer bonding, dos NIC pueden aparecer como una interfaz lógica, lo que resulta en una velocidad doble. Esta es una característica nativa del kernel de Linux que es compatible con la mayoría de los switches.



> Esta configuración es habitual en los CPD (Centro de Procesado de Datos) para protegerse de fallos o caídas en los switch que conectan a distintos servidores Proxmox.

Un ejemplo de bond con IP fija:

```
auto lo
iface lo inet loopback

iface eno1 inet manual

iface eno2 inet manual

auto bond0
iface bond0 inet static
slaves eno1 eno2
address 192.168.1.2
netmask 255.255.255.0
bond_miimon 100
bond_mode 802.3ad
bond_xmit_hash_policy layer2+3

auto vmbr0
iface vmbr0 inet static
address 10.10.10.2
netmask 255.255.255.0
gateway 10.10.10.1
bridge_ports eno1
bridge_stp off
bridge_fd 0
```

Otra posibilidad es utilizar el **bond directamente como puerto de puente.** Esto se puede usar para hacer que la red de invitados sea tolerante a fallos.

```
auto lo
iface lo inet loopback

iface eno1 inet manual

iface eno2 inet manual
```

```
auto bond0
iface bond0 inet manual
slaves eno1 eno2
bond_miimon 100
bond_mode 802.3ad
bond_xmit_hash_policy layer2+3

auto vmbr0
iface vmbr0 inet static
address 10.10.10.2
netmask 255.255.255.0
gateway 10.10.10.1
bridge_ports bond0
bridge_stp off
bridge_fd 0
```

VLAN 802.1Q

Una LAN virtual **(VLAN)** es un dominio de difusión que es particionado y aislado en la red en la capa 2. Por lo tanto, es posible tener múltiples redes (**4096**) en una red física, cada una independiente de las demás.

Cada red VLAN se identifica por un **número llamado tag**. Los paquetes de red se etiquetan para identificar a qué red virtual pertenecen.

Ejemplo: usamos la VLAN 5 para la IP de administración Proxmox VE con el tradicional de Linux bridge.

```
auto lo
iface lo inet loopback

iface eno1 inet manual

iface eno1.5 inet manual

auto vmbr0v5
iface vmbr0v5 inet static
```

```
address 10.10.10.2
netmask 255.255.255.0
gateway 10.10.10.1
bridge_ports eno1.5
bridge_stp off
bridge_fd 0

auto vmbr0
iface vmbr0 inet manual
bridge_ports eno1
bridge_stp off
bridge_fd 0
```

> Este material ha sido elaborado a partir de los manuales del Profesor **Antonio López Téllez (2020)** sobre Proxmox VE 6.0

1.3.- Configuración de la red con Open vSwitch

Open Virtual Switch

Logo Open vSwitch (GNU/GPL)

Open vSwitch es un conmutador virtual multicapa con calidad de servicio y de licencia de **código abierto Apache 2.0**. Está diseñado para permitir la automatización masiva de la red a través de extensión programadas, sin dejar de admitir interfaces y protocolos de administración estándar

> Por defecto, Proxmox usa Linux Bridges (switch virtual al igual que Open vSwitch).

Open vSwitch es un switch virtual, al igual que Linux Bridge y reenvía paquetes entre las interfaces que están conectadas a él (capa 2 de TCP/IP). Generalmente se usa para reenviar paquetes en enrutadores, puertas de enlace o entre máquinas virtuales y espacios de nombres de red en un host (contenedores).

> La gran ventaja de Open vSwitch era que soportaba STP (Spanning Tree Protocol) pero Linux Bridge ha incluido soporte básico para STP, multicast (multidifusión) y Netfilter desde las series de kernel 2.4 y 2.6 de Linux.

Sin embargo, Linux Bridge, no es del todo compatible con todos los componentes de red y protocolos del resto de la industria del hardware de red.

Linux Bridge queda atrás de Open vSwitch en la mayoría de las pruebas de rendimiento y tiempo de transacción (latencia), donde la diferencia más notable se observó con grandes cargas transaccionales.

Requisito previo: instalar dependencias

```
apt update
apt install openvswitch-switch -y
```

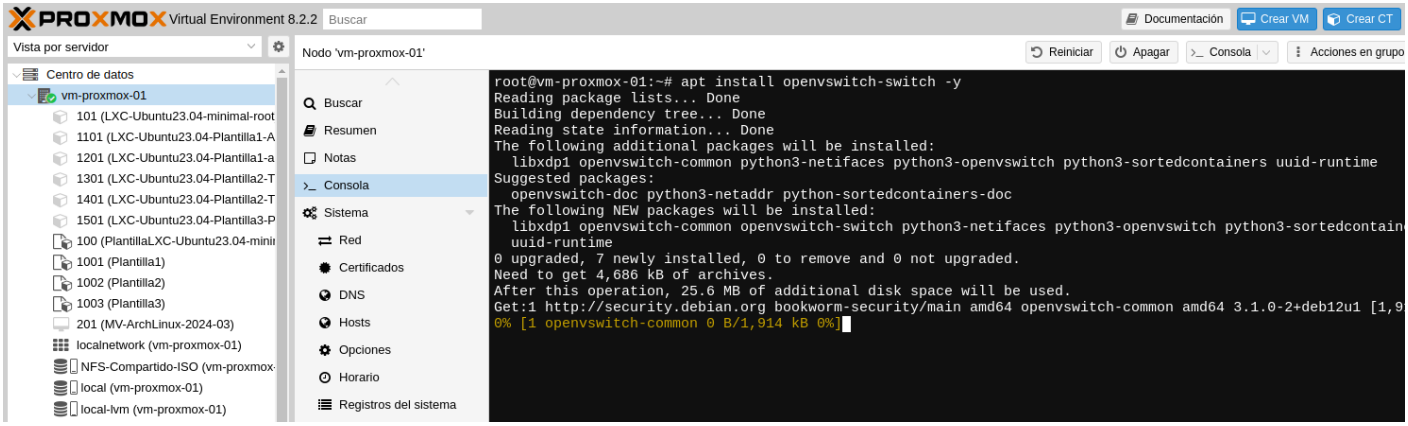


Imagen de elaboración propia· *Instalación de los paquetes necesarios para habilitar Open vSwitch* (CC BY-NC-SA)

> **Importante:** crea una copia de seguridad de la configuración de red actual para cambiar

Abre el shell desde la consola web y ejecuta este comando:

```
cp /etc/network/interfaces /etc/network/interfaces.bak
```

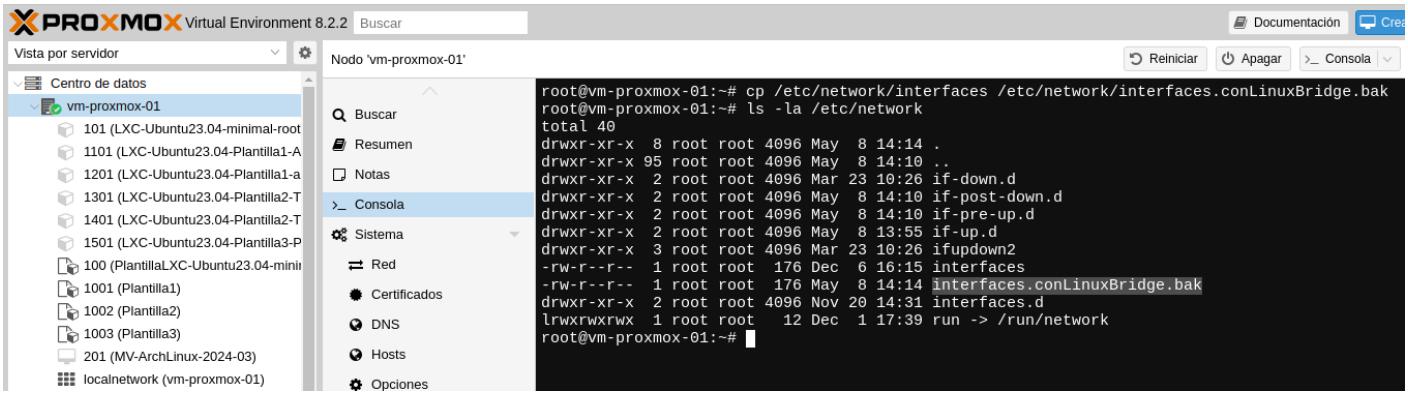


Imagen de elaboración propia· *Copia de seguridad del fichero de configuración de red del nodo Proxmox* (CC BY-NC-SA)

Los OVS Bridge son más recomendables que los Linux Bridge para realizar un cluster de Proxmox.

Eliminaremos el Linux Bridge "vmbr0" creado por defecto en Proxmox VE, pero no le daremos a "Aplicar configuración" hasta haber terminado con los cambios:

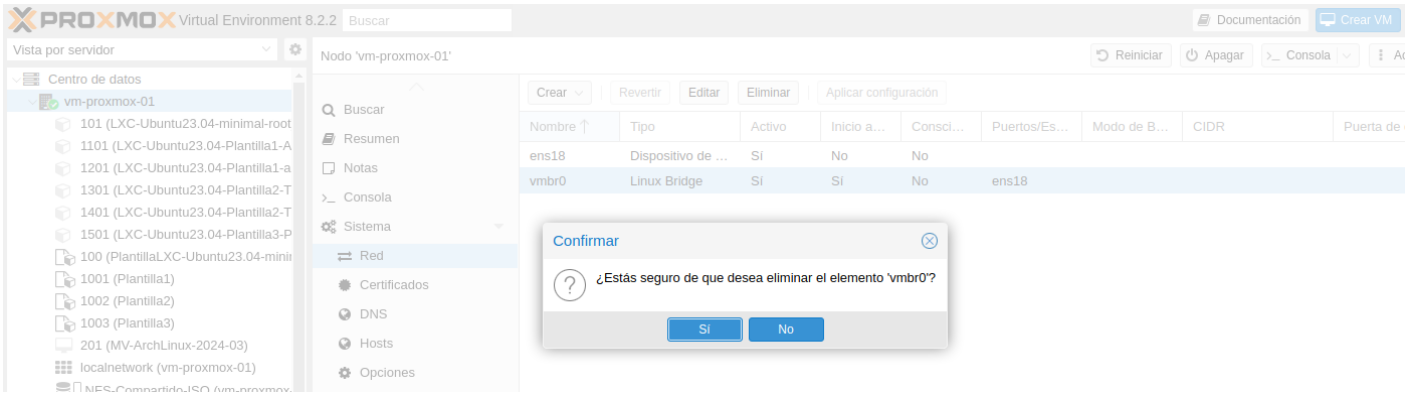


Imagen de elaboración propia· *Eliminar el Linux Bridge "vmbr0"* (CC BY-NC-SA)

Sin actualizar los cambios, daremos a "Crear" un OVS Bridge:

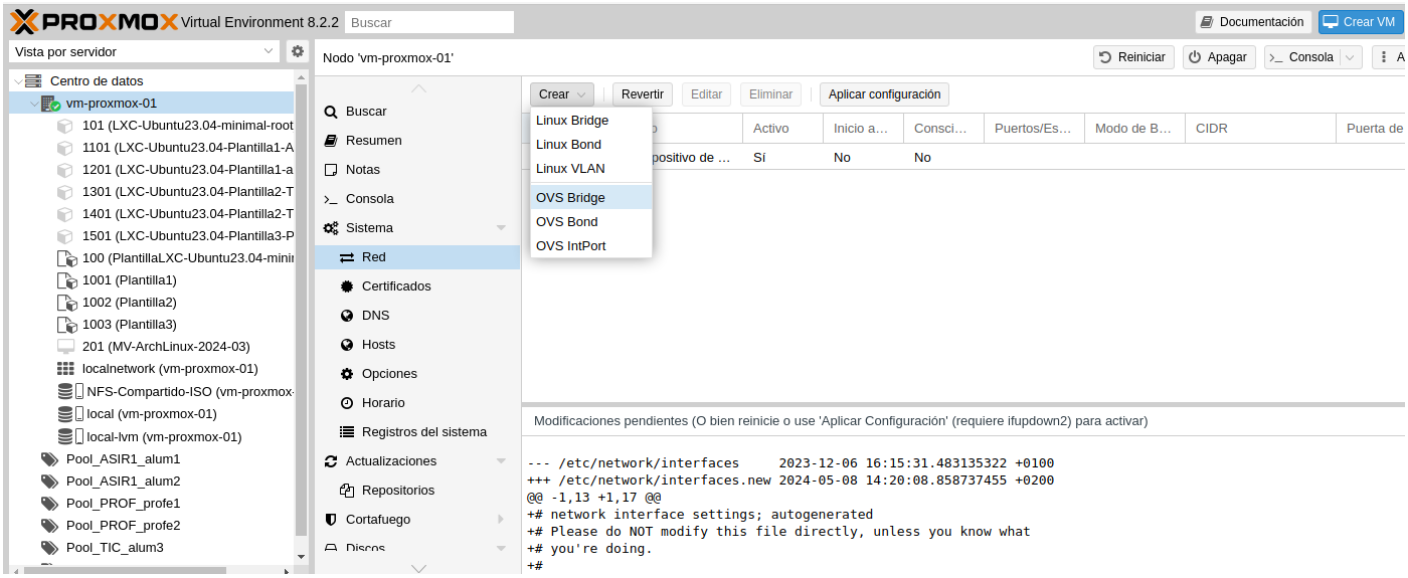


Imagen de elaboración propia· *Crear un Open Virtual Switch* (CC BY-NC-SA)

Llamaremos al nuevo Open vSwitch como "vmbr0" para que coincida con el anterior nombre de Linux Bridge y de esta manera no se vean afectadas las interfaces de red de las MV y contenedores creados previamente:

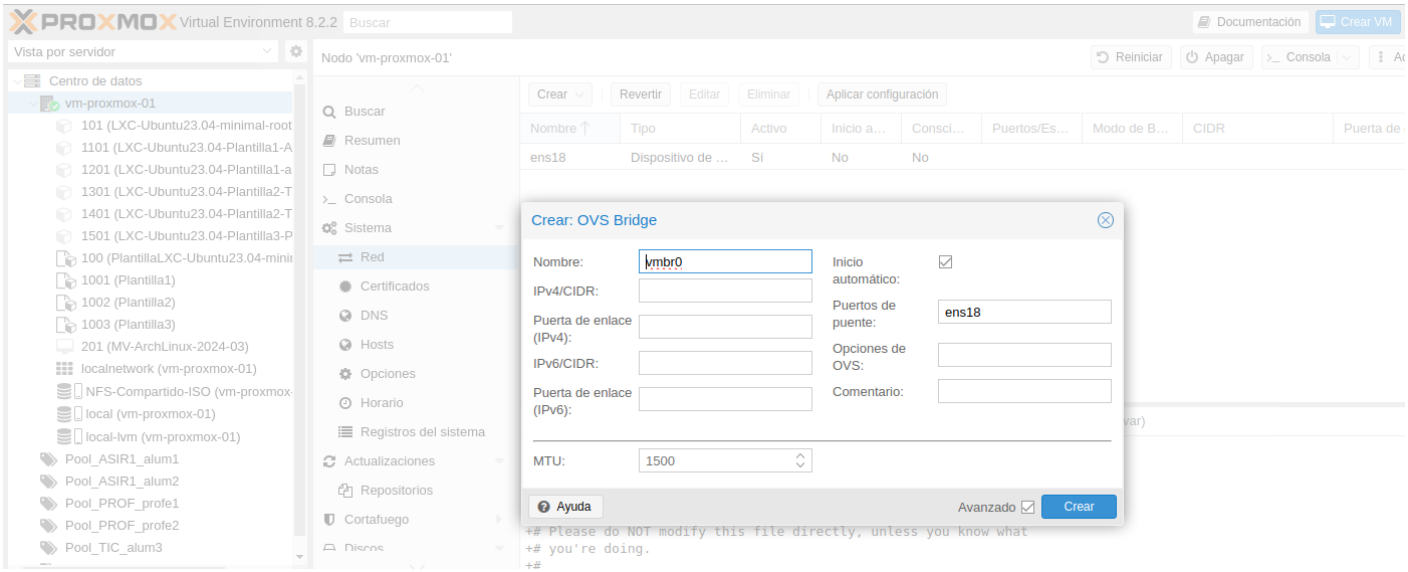


Imagen de elaboración propia· *Creación del OVS Bridge* (CC BY-NC-SA)

> **¡ATENCIÓN!** a diferencia de una configuración con Linux Bridge, debemos de habilitar un "OVS IntPort" para acceder a GUI de Proxmox:

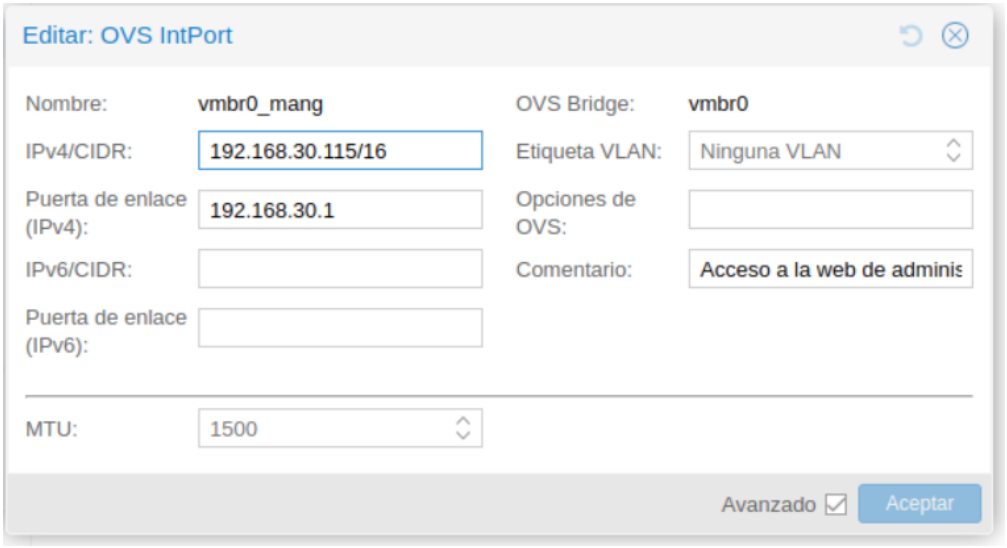


Imagen de elaboración propia· *Crear un OVS IntPort para el acceso a web administrativa de Proxmox* (CC BY-NC-SA)

Ahora sí aplicamos los cambios:

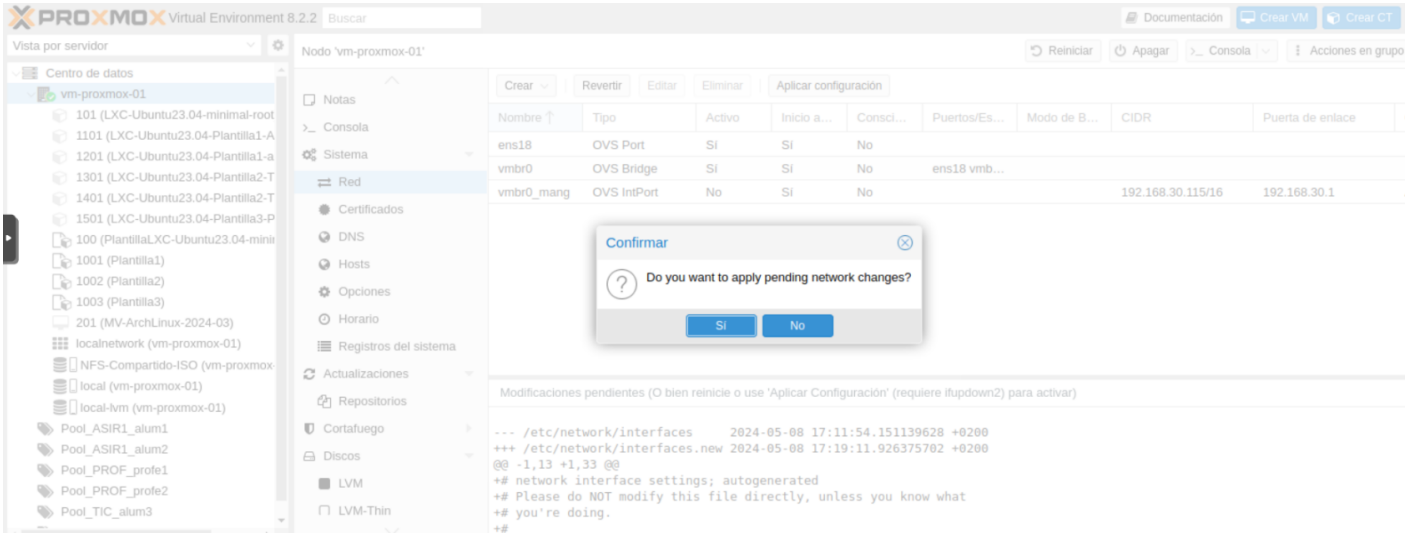


Imagen de elaboración propia· *Aplicar cambios para reiniciar el servicio de red* (CC BY-NC-SA)

Si todo ha salido bien en un momento tendremos preparada nuestro OVS Bridge. Si queremos que la IP del OVS IntPort se obtenga por DHCP tendremos que modificar el fichero /etc/network/interfaces de forma manual:

```
  GNU nano 7.2                                                /etc/network/interfaces *
# network interface settings; autogenerated
# Please do NOT modify this file directly, unless you know what
# you're doing.
#
# If you want to manage parts of the network configuration manually,
# please utilize the 'source' or 'source-directory' directives to do
# so.
# PVE will preserve these directives, but will NOT read its network
# configuration from sourced files, so do not attempt to move any of
# the PVE managed interfaces into external files!

auto lo
iface lo inet loopback

auto ens18
iface ens18 inet manual
        ovs_type OVSPort
        ovs_bridge vmbr0

auto vmbr0_mang
iface vmbr0_mang inet dhcp
#iface vmbr0_mang inet static
#       address 192.168.30.115/16
#       gateway 192.168.30.1
        ovs_type OVSIntPort
        ovs_bridge vmbr0
#Acceso a la web de administración de Proxmox

auto vmbr0
iface vmbr0 inet manual
        ovs_type OVSBridge
        ovs_ports ens18 vmbr0_mang

source /etc/network/interfaces.d/*
```

Imagen de elaboración propia· *Modificación del fichero /etc/network/interfaces para que OVS IntPort obtenga una configuración de red por DHCP* (CC BY-NC-SA)

Si por alguna razón cometiste un error tipográfico o algún otro error en tu configuración y tienes problemas para conectarte a través del navegador web:

Vaya a su servidor Proxmox e inicie sesión localmente y ejecute estos comandos:

```
cp /etc/network/interfaces.bak /etc/network/interfaces
ifreload -a
```

Intenta seguir otra vez estos mismos pasos.

Si editamos el OVS Bridge "vmbr0" nos daremos cuenta que está haciendo de puente entre la interfaz de red ens18 y el OVS IntPort "vmbr0_mang":

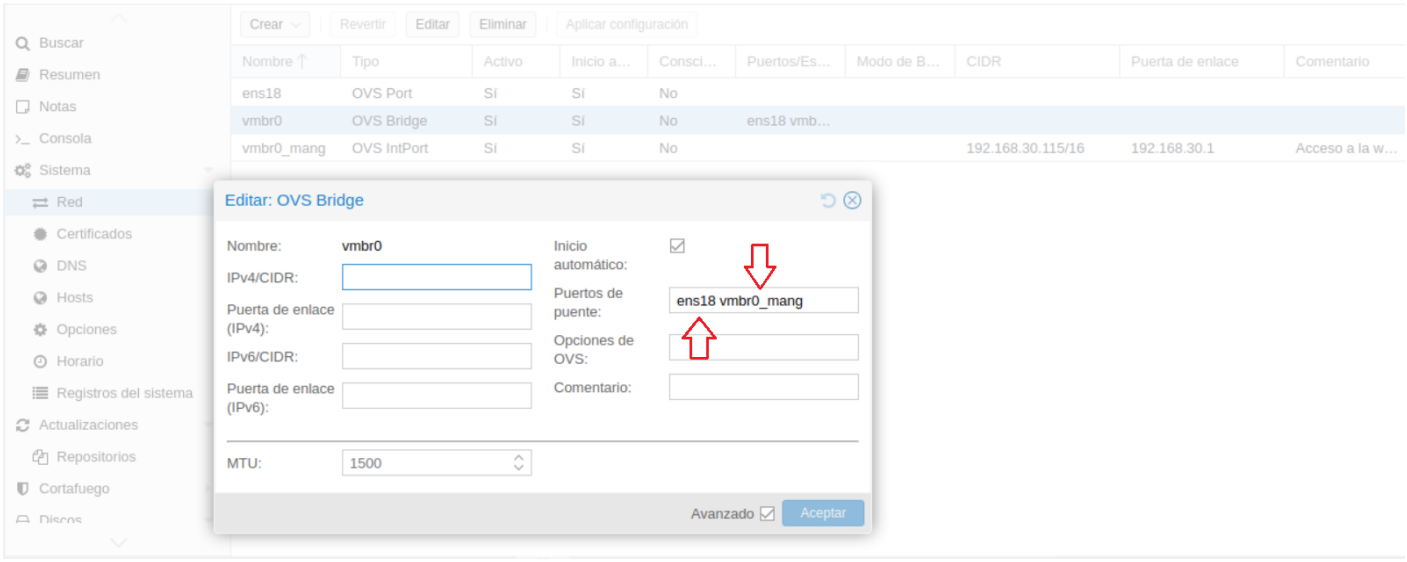


Imagen de elaboración propia· *Edición de OVS Bridge vmbr0* (CC BY-NC-SA)

Comprobaremos ahora si funciona el puente de red de OVS Bridge "vmbr0" arrancando un contenedor y verificando su configuración de red y su conexión a Internet, haciendo un ping a un servidor DNS públicos de Google:

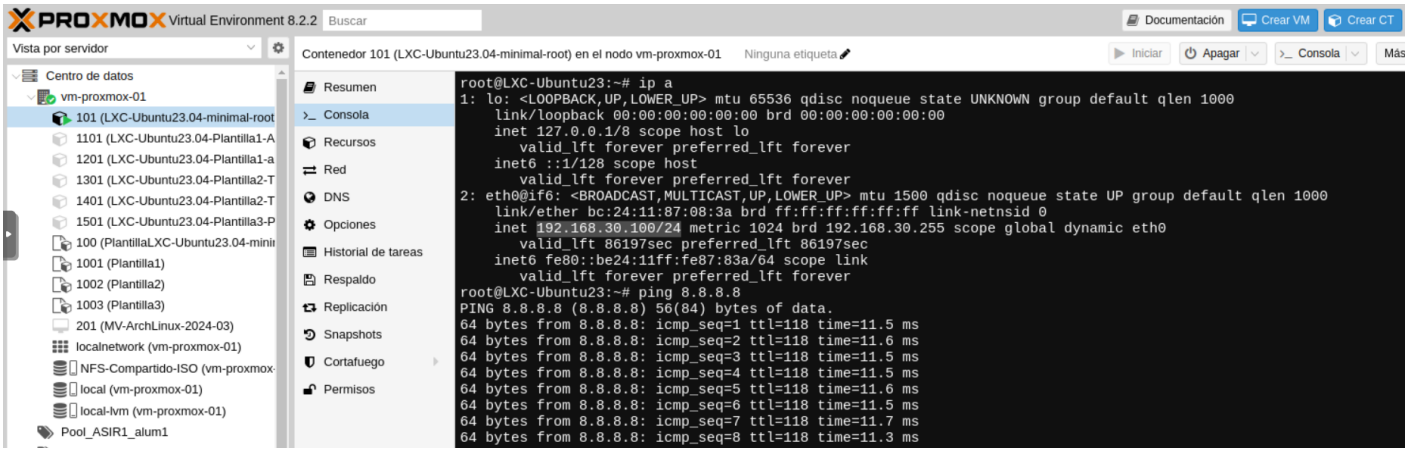


Imagen de elaboración propia· *Configuración de red y ping al DNS de Google 8.8.8.8* (CC BY-NC-SA)

Para saber más

Para seguir profundizando:

https://pve.proxmox.com/wiki/Network_Configuration

2.- Configuración de la red de un nodo Proxmox (utilizando Linux Bridge)

2.1.- Configuración por defecto en un nodo Proxmox

La configuración de red en Proxmox VE se ofrece a nivel de servidor (es decir, cada nodo que forma el clúster tendrá su configuración de red). Podemos obtener la configuración inicial de red a nivel del servidor, en la opción **Sistema - Red** del nodo:

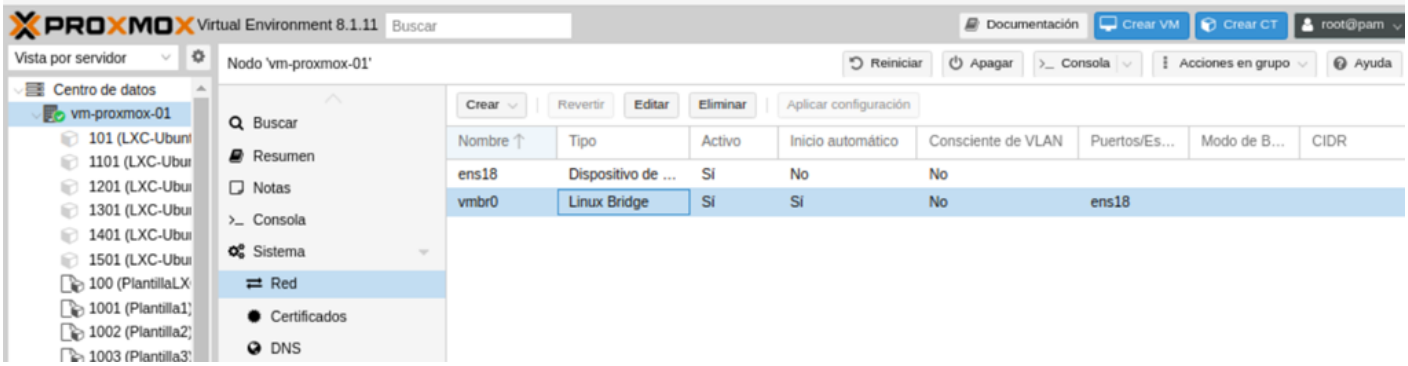


Imagen de elaboración propia *Interfaces de red del servidor Proxmox* (CC BY-NC-SA)

En este manual hemos instalado Proxmox VE sobre una máquina virtual dentro de otro Proxmox, y en nuestro escenario nos aparecen los siguientes recursos:

- La interfaz del servidor Proxmox "física" (en el ejemplo, ens18): Corresponde a la interfaz del servidor real.
- Un Linux Bridge (vmbr0): A este bridge virtual está conectado la interfaz de del servidor y ha tomado del router físico una IP por DHCP

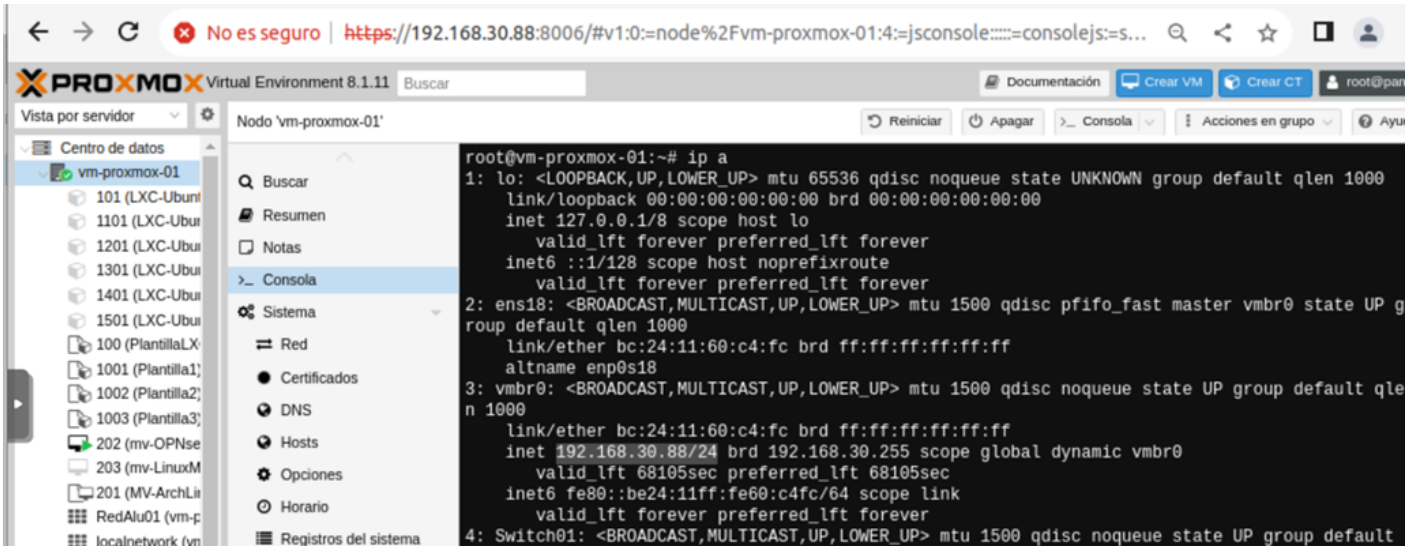


Imagen de elaboración propia. *IP del servidor Proxmox vinculada a la vmbr0 que será el puente virtual que proporciona acceso a las MV y contenedores al exterior* (CC BY-NC-SA)

> Cómo ya vimos en la instalación de Proxmox, **obtener la dirección del servidor Proxmox por DHCP no es lo idóneo**, sino que debería configurarse de forma estática.

Por defecto las máquinas virtuales y contenedores que estamos creando se conectan a este Linux Bridge (vmbr0), y tomarán direccionamiento del router físico, en mi caso el direccionamiento será el 192.168.30.0/24. Por lo tanto, todas las máquinas gestionadas por Proxmox serán accesibles desde cualquier ordenador de mi red local.

El esquema que tenemos sería el siguiente:

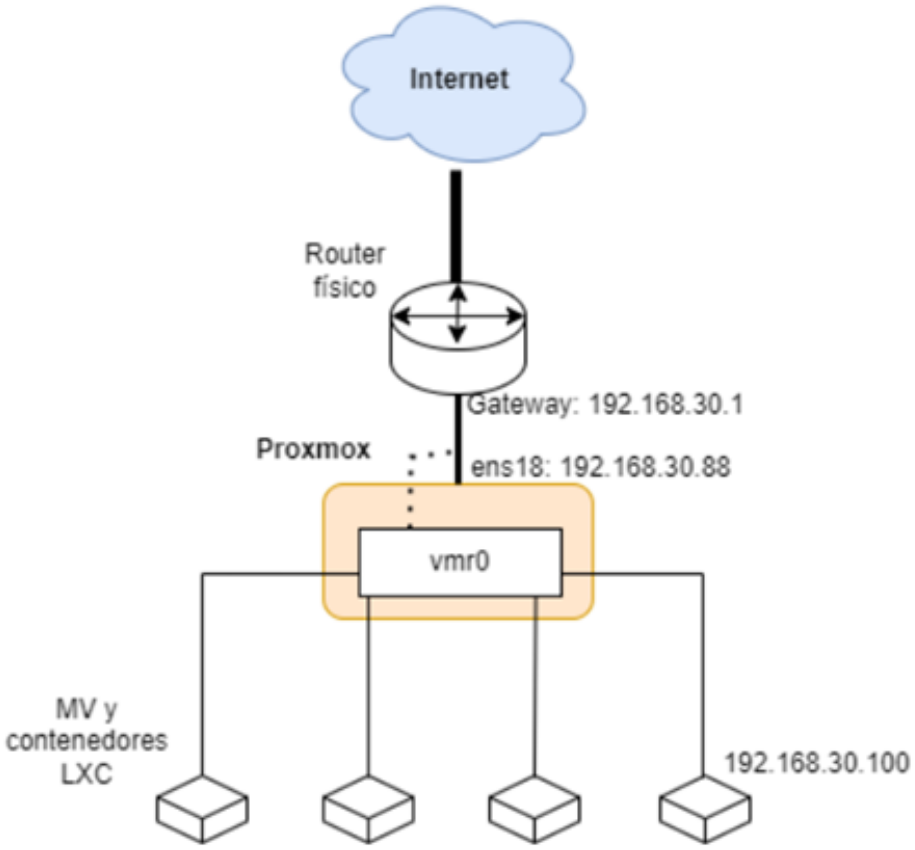


Imagen de elaboración propia. *Esquema de red* (CC BY-NC-SA)

En una instalación real de Proxmox el servidor puede tener más interfaces de red, podría existir un switch físico en la infraestructura y podríamos necesitar una configuración más avanzada usando por ejemplo Bonding (Link Aggregation) o VLAN. Proxmox VE nos permite configurar estas opciones avanzadas usando Linux Bridge o Open vSwitch.

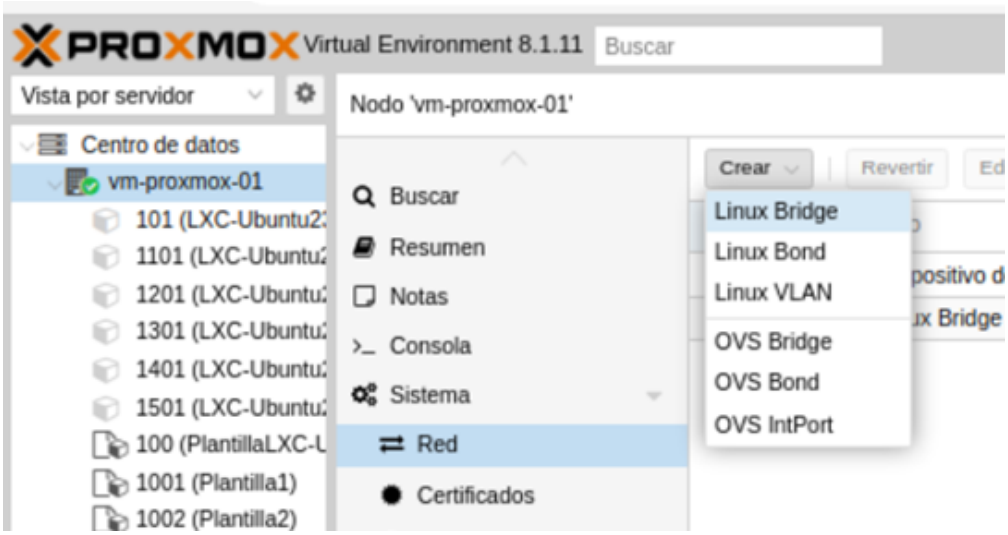


Imagen de elaboración propia *Creación de nuevos Virtual Linux Bridge* (CC BY-NC-SA)

Conexión de las máquinas virtuales/contenedores al puente vmbr0

Como hemos indicado anteriormente, por defecto, las máquinas virtuales y contenedores que hemos creado en nuestro servidor se conecta al Linux Bridge "vmbr0", por lo tanto, se configurarán de forma automática usando el servidor DHCP de nuestra infraestructura, en nuestro caso el del router físico.

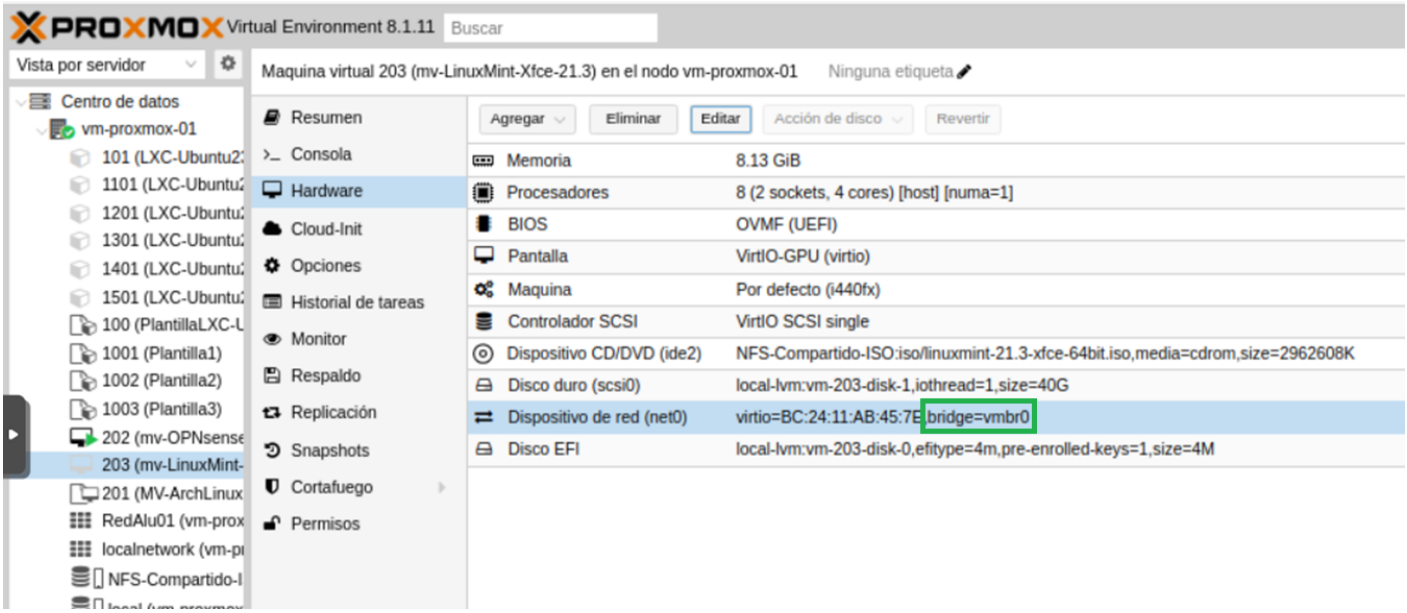


Imagen de elaboración propia. *Conexión de la interfaz de red al Virtual Linux Bridge "vmbr0"* (CC BY-NC-SA)

Por lo que podemos acceder a la MV 203 desde cualquier equipo conectada a nuestra red local.

> Las máquinas virtuales y contenedores conectados al bridge vmbr0 obtendrán direccionamiento automático, tendrán acceso al exterior y podremos acceder a ellos sin ningún problema desde cualquier host de nuestra red local.

2.2.- Configuración de red aislada para las MV y contenedores

Proxmox VE nos permite crear nuevos Linux Bridge donde podremos conectar nuestras máquinas en una red privada. Algunos escenarios donde podríamos usar esta posibilidad podrían ser:

- Varias máquinas conectadas al exterior con una interfaz conectada a vmbr0 y otra interfaz conectada a otro bridge. Estas máquinas tendrían una conexión entre ellas en una red privada.
- Un equipo que funcione como router/nat/cortafuegos que esté conectado al exterior por vmbr0 y a otras redes internas donde tenemos diferentes máquinas.
- Un laboratorio de máquinas que no tengan conectividad al exterior y que estén conectadas a una red interna.

Creación de un nuevo bridge

Para crear un nuevo bridge tenemos que elegir la opción Sistema - Red - Crear - Linux Bridge:

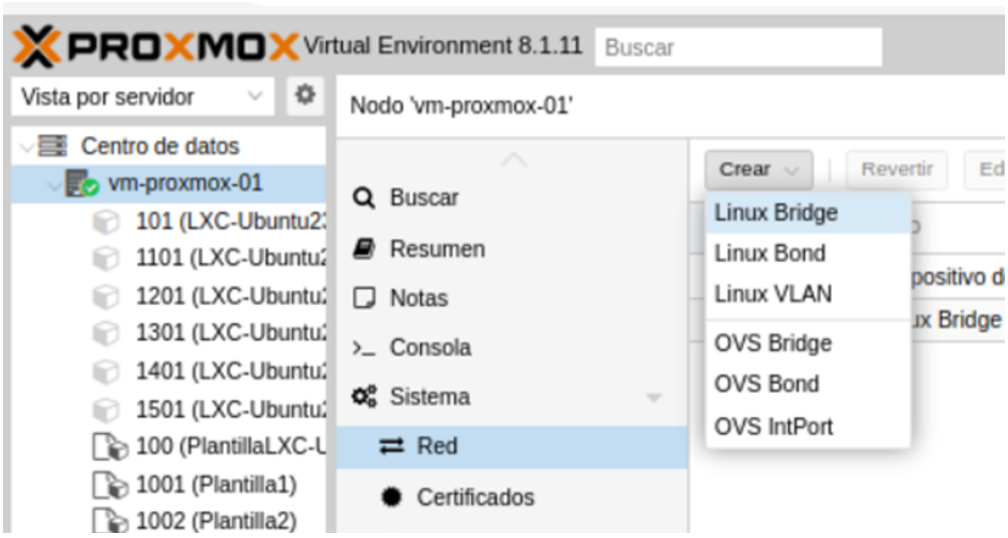


Imagen de elaboración propia· *Crear un nuevo switch virtual para conectar las interfaces de red de las MV y contenedores* (CC BY-NC-SA)

En la creación del bridge tendremos que configurar los siguientes parámetros:

- El nombre del bridge: debe tener el formato de vmbr y un número.
- Aunque no es necesario, podríamos asignarle una IP al bridge (normalmente los switch tienen una IP que nos permiten acceder a ellos para configurarlos).
- Si indicamos la IP del Gateway lo que estaríamos haciendo es conectar una interfaz virtual del servidor Proxmox a este bridge, por lo que desde el servidor Proxmox tendríamos conectividad con las máquinas conectadas a este bridge. En nuestro caso no la vamos a indicar porque no vamos a usar esta funcionalidad.
- Y finalmente es importante activar el "Inicio Automático" para que el bridge se levante tras un reinicio.

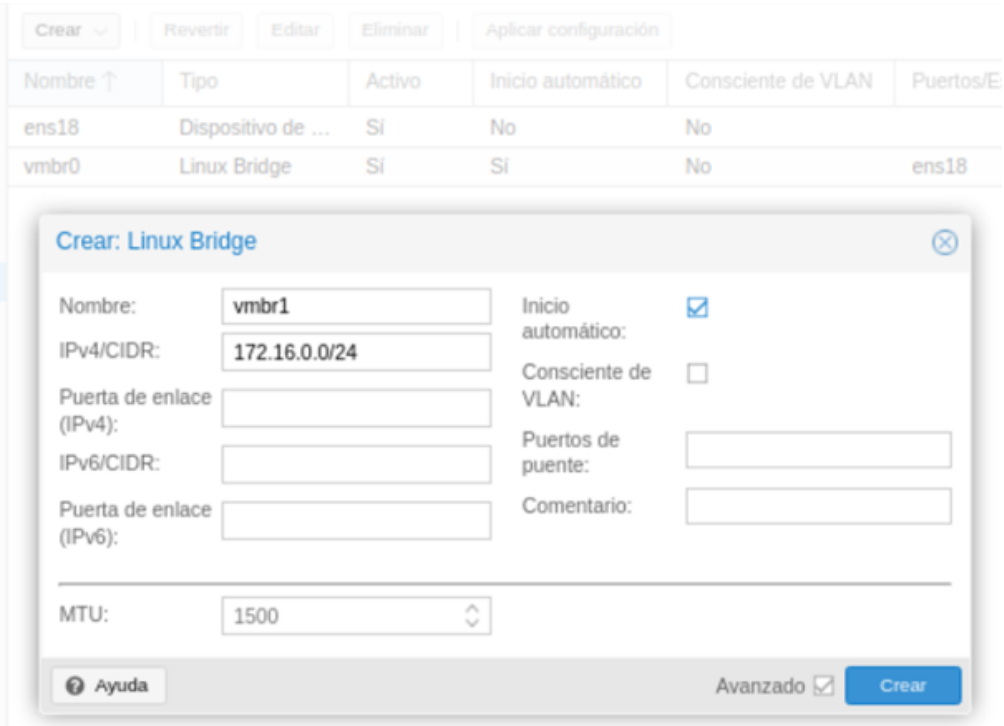


Imagen de elaboración propia· *Creación de un nuevo Linux Bridge* (CC BY-NC-SA)

Una vez creado o modificado algún elemento de la red de Proxmox VE debemos aplicar los cambios para que verdaderamente se produzca el cambio de configuración:

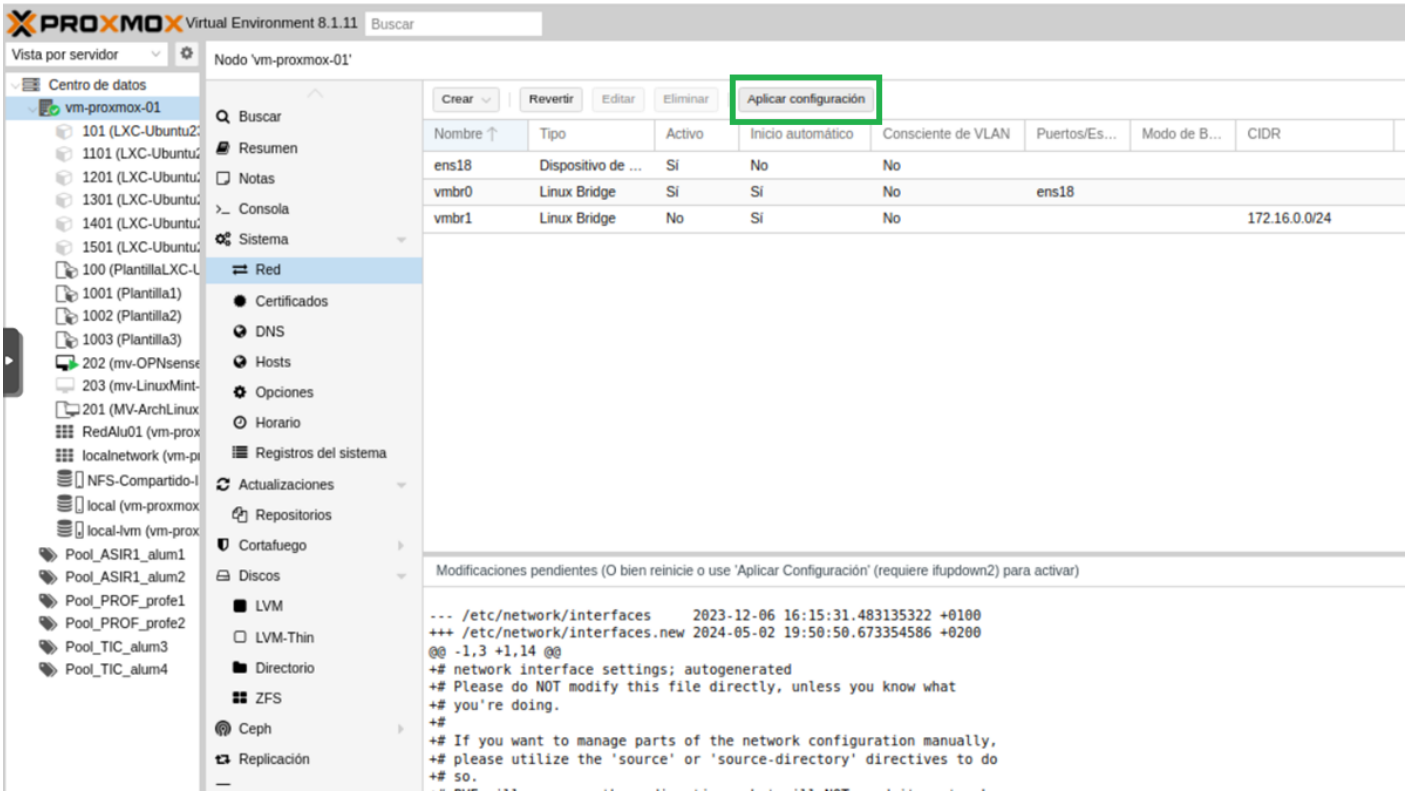


Imagen de elaboración propia· *Aplicar cambios en la Red* (CC BY-NC-SA)

3.- Contafuegos en Proxmox

3.1.- Firewall gestionado por Proxmox

> El cortafuegos hay que activarlo a tres niveles

A nivel de Centro de datos

Para activar el cortafuegos a nivel del clúster de servidores, tenemos que activar en la opción Centro de datos - Cortafuego - Opciones:

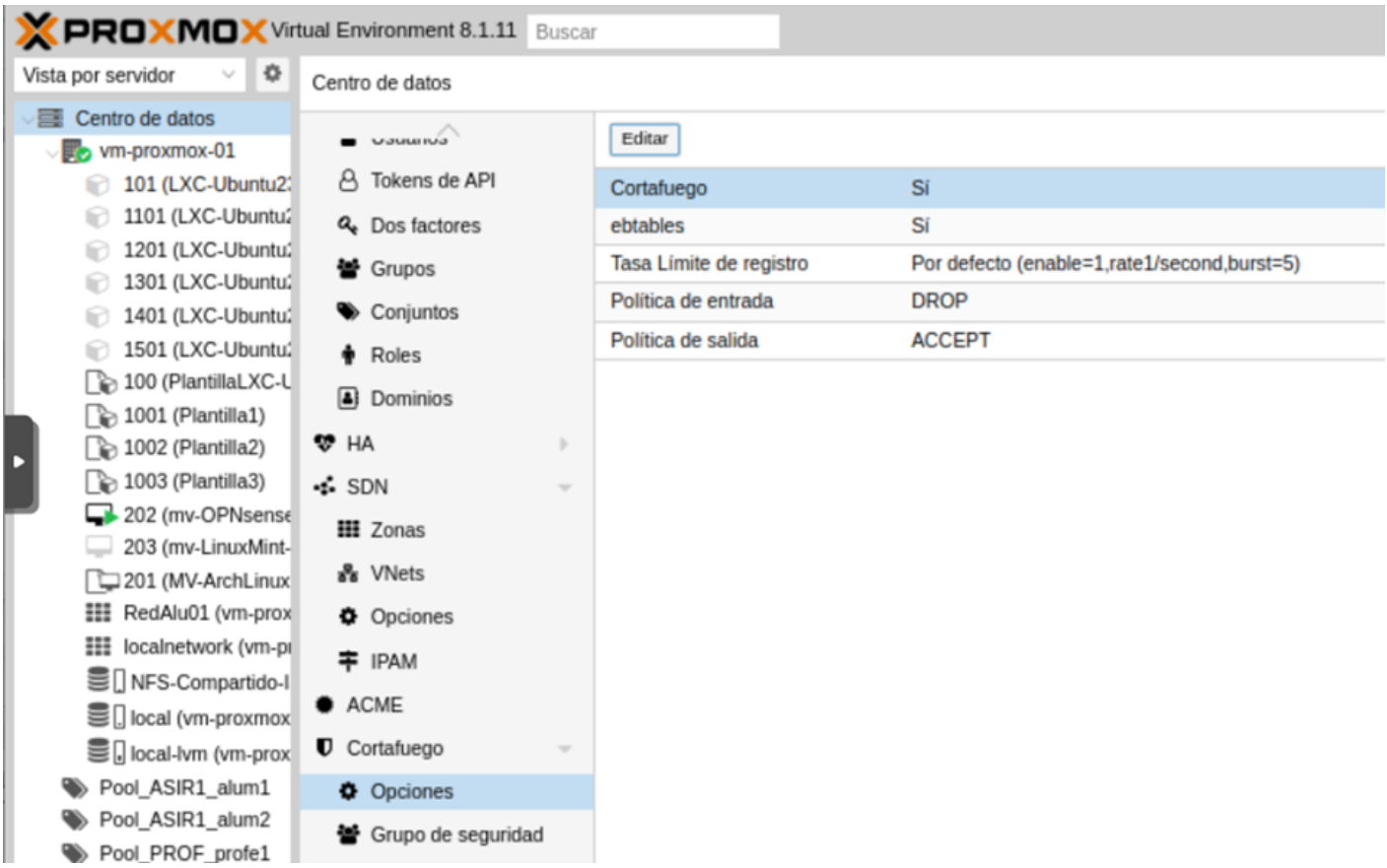


Imagen de elaboración propia· *Activar el firewall a nivel de Centro de Datos* (CC BY-NC-SA)

Aunque no es necesario, para obtener más seguridad en el acceso de los servidores del clúster, podemos cambiar la política para denegar por defecto todo el tráfico de salida, para ello cambiamos la **Políticas de entrada (Output Policy) a DROP**.

En este nivel también puedes configurar:

- Security Group: Conjuntos de reglas de cortafuegos que posteriormente podemos asignar a un cortafuegos de una máquina.
- Alias: Nos permite nombrar direcciones IP para que sea más sencillo crear las reglas de cortafuegos.
- IPSec: Nos permite crear grupos de IP para facilitar la asignación de reglas de cortafuegos a varias IP.

A nivel de Servidor

En este caso volvemos a activar el cortafuegos eligiendo el nombre del nodo, en mi caso la opción vm-proxmox-01 - Cortafuegos - Opciones.

Al activar el cortafuegos a nivel del servidor, se utilizan las políticas de entrada y salida por defecto que se habían configurado en el nivel de Centro de datos: todo el tráfico (de entrada y de salida bloqueado, pero se mantienen abierto el puerto 8006 (para acceder a la página web) y el 22 (para el acceso por ssh al servidor).

Nivel de máquina/contenedor

Para activar el cortafuegos para una máquina/contenedor nos vamos a la opción Cortafuegos - Opciones del recurso:

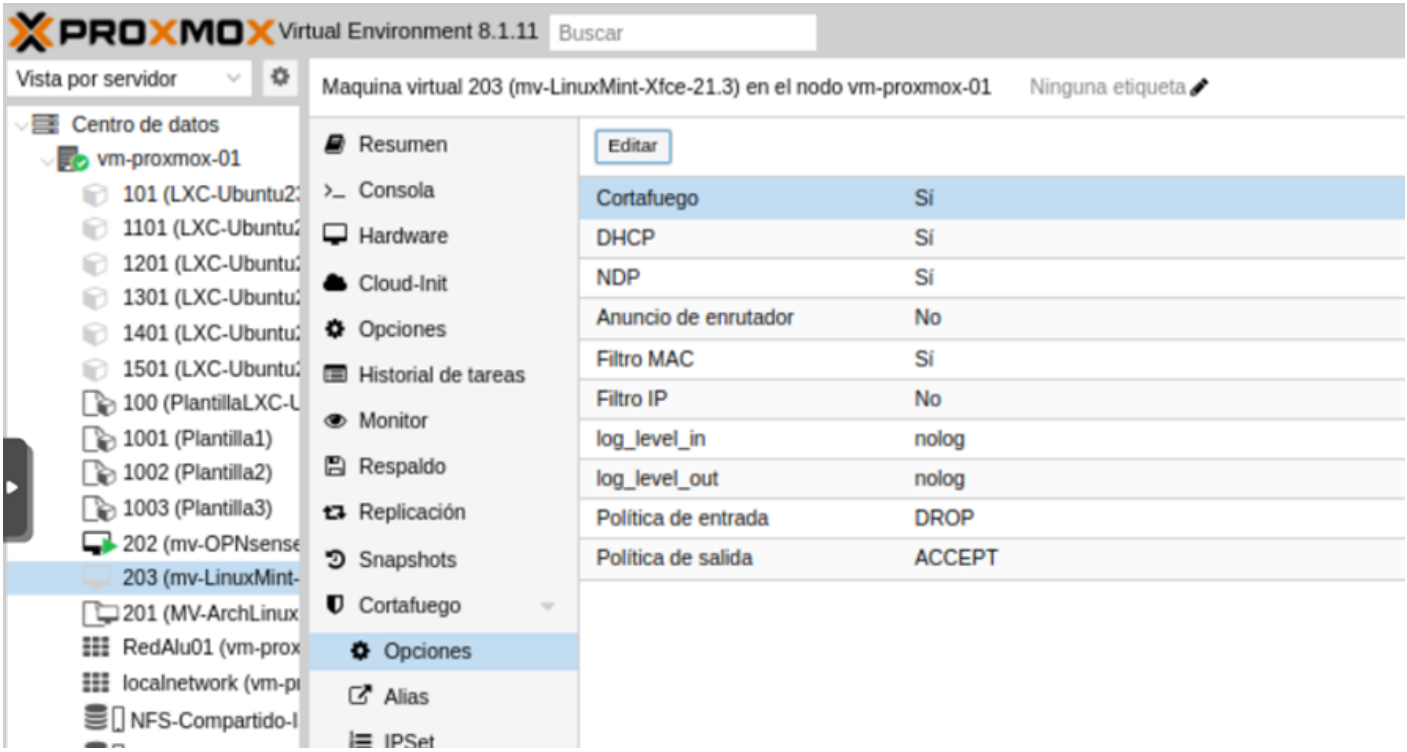


Imagen de elaboración propia: *Activar el cortafuego en una MV* (CC BY-NC-SA)

Tendremos que asegurar en la interfaz de red de la MV que el cortafuegos se encuentra activado:

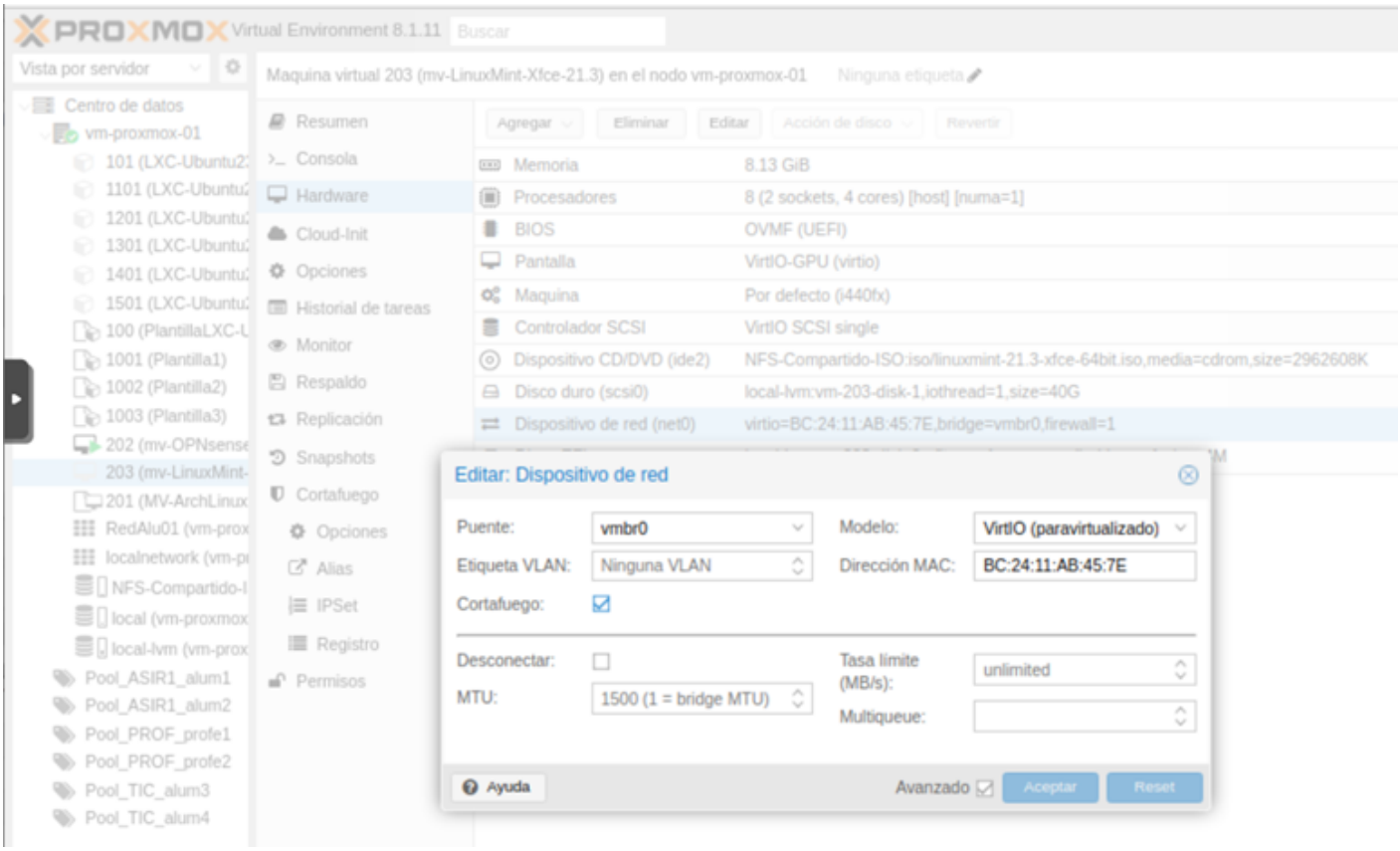


Imagen de elaboración propia: *Activar el cortafuegos en la interfaz de red de la MV* (CC BY-NC-SA)

Vemos las políticas por defecto para esta máquina:

- **Políticas de entrada** (Input policy): DROP, es decir se deniega todo el tráfico de entrada (y tenemos que crear reglas de cortafuegos para permitir el tráfico que nos interese).
- **Políticas de salida** (Output Policy): ACCEPT, se acepta todo el tráfico de salida de la máquina (y tenemos que indicar las reglas de cortafuegos para denegar el tráfico que no permitamos).

Si quisiéramos un cortafuegos más restrictivo pondríamos las dos políticas por defecto a DROP, es decir, tanto el tráfico de entrada como el de salida estarían bloqueados, y tendríamos que ir creando reglas de cortafuegos para aceptar el tráfico que deseáramos permitir.

Además, cómo una máquina o contenedor pueden tener más de una interfaz podemos activar o desactivar el cortafuegos para cada interfaz de red. Por defecto, el cortafuegos está activo en cada interfaz de red. Podemos modificar las características del interfaz de red para desactivar el cortafuego.

En resumen, para poder habilitar el cortafuegos para una máquina virtual y/o contenedor, debemos habilitar el cortafuegos tanto a nivel de Centro de datos como a nivel del servidor, finalmente podemos activar o desactivar el cortafuegos para cada una de las interfaces de red de una máquina o contenedor.

Ejemplo de creación de reglas en el cortafuego

Como hemos visto anteriormente, si habilitamos el cortafuegos para una máquina tendrá permitido el tráfico hacia el exterior (Output Policy: ACCEPT) y tendrá denegado el tráfico desde el exterior a la máquina (Input policy: DROP).

Partimos de una máquina que tiene un servidor ssh instalado. Está máquina tendrá conectividad al exterior, pero no tendrá conectividad desde el exterior. Vamos a poner dos ejemplos de reglas:

Regla para denegar que la máquina haga ping al exterior

Todo el tráfico está permitido hacía el exterior, pero vamos a denegar el ping. Para ello debemos crear una regla de salida para denegar el protocolo ICMP, para ello, a nivel de máquina virtual, vamos a añadir una regla al cortafuego, eligiendo la opción Cortafuegos - Añadir:

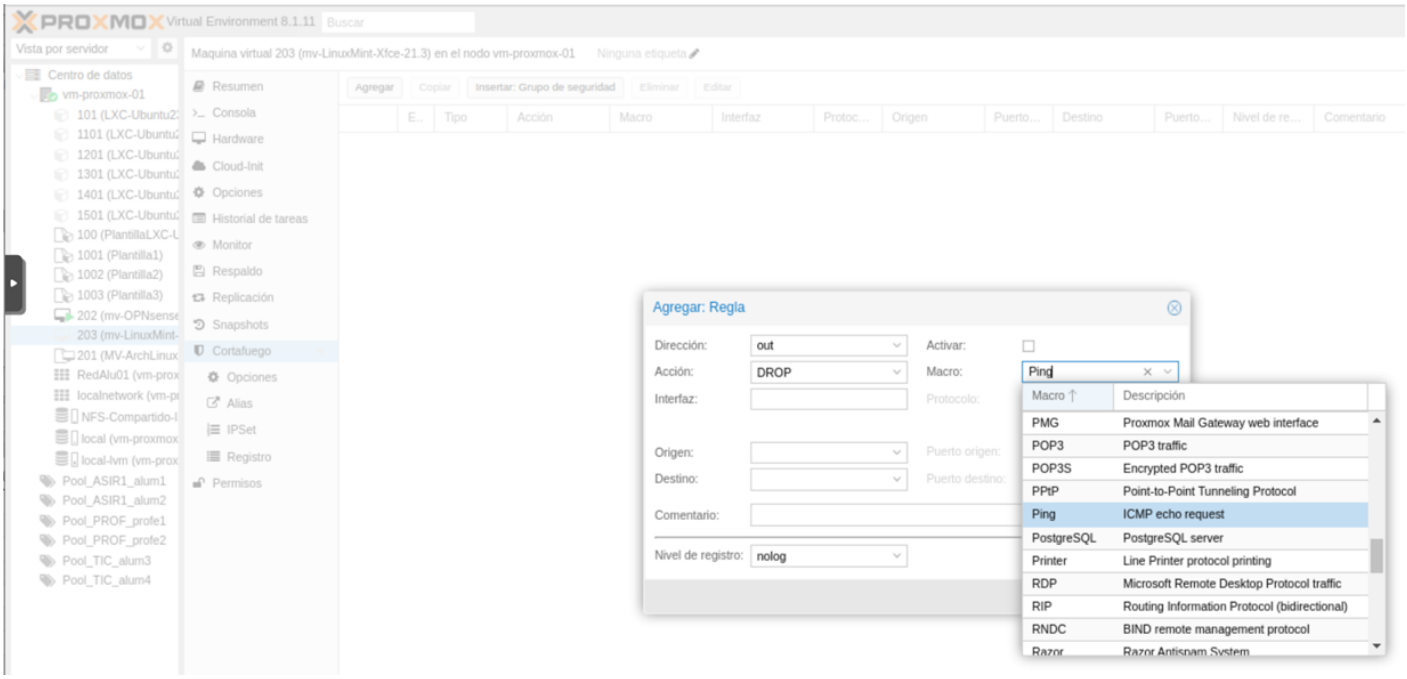


Imagen de elaboración propia· *Regla en el cortafuegos para impedir hacer ping a host fuera de la red local* (CC BY-NC-SA)

> Debemos de activar esta regla para que sea efectiva.

Regla para permitir el acceso por ssh a la máquina

En esta ocasión tenemos que crear una regla que permita (acción ACCEPT) la entrada (dirección in) por el puerto de destino 22 del protocolo TCP. En esta ocasión no vamos a elegir el servicio de la lista de Macro, lo vamos a indicar directamente. Quedaría:

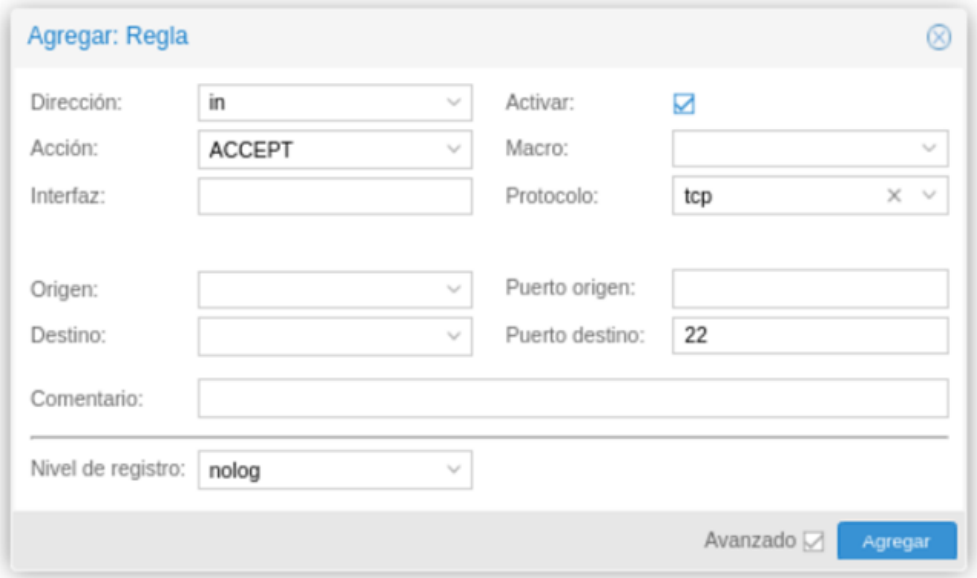


Imagen de elaboración propia· *Permitir el puerto de escucha 22 por TCP en la MV* (CC BY-NC-SA)

Para saber más

- Firewall
- Proxmox VE Firewall

4.- SDN (Software Defined Network)

Se encuentra en el libro: Resdes en Proxmox II

5.- Licencia y autoría de este material

Materiales desarrollados inicialmente por **Daniel Cano Verdú (2024)** profesor de FP de la Junta de Andalucía y actualizados por el profesorado de la Junta de Andalucía bajo licencia **Creative Commons BY-NC-SA**.

Antes de cualquier uso leer detenidamente el siguiente Aviso legal