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)

Licencia Creative Commons BY-NC-SA.

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)

Licencia Creative Commons BY-NC-SA.

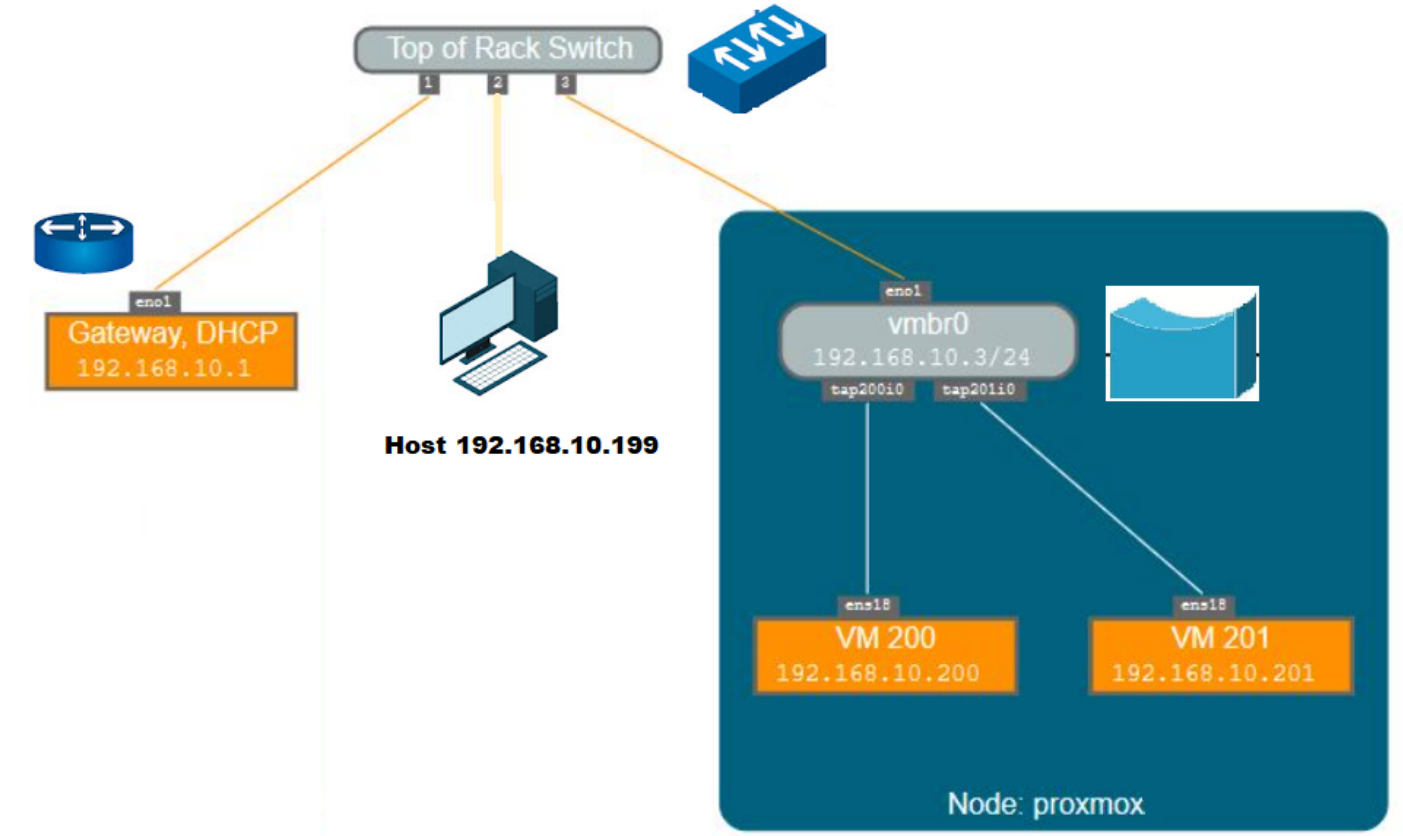


Imagen de elaboración propia. *Configuración Linux Bridge (predeterminada)* (CC BY-NC-SA)

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.

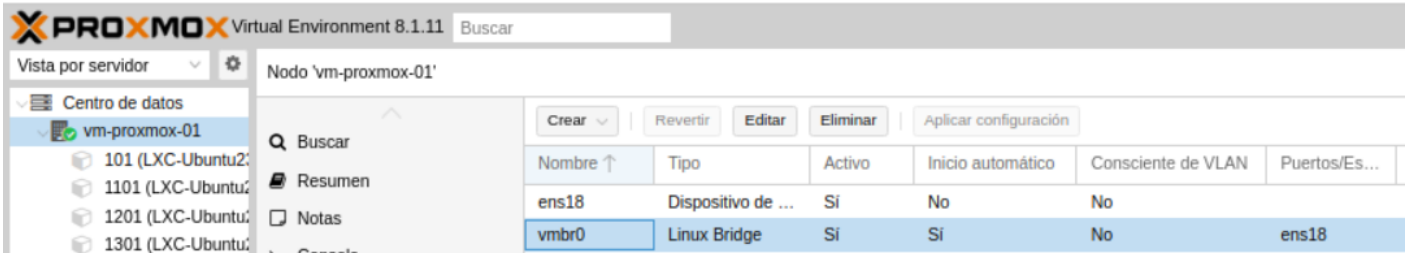


Imagen de elaboración propia. *El Linux Bridge "vmbr0" haciendo de puente con la interfaz de red "ens18"* (CC BY-NC-SA)

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
```

Licencia Creative Commons BY-NC-SA.

```
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.

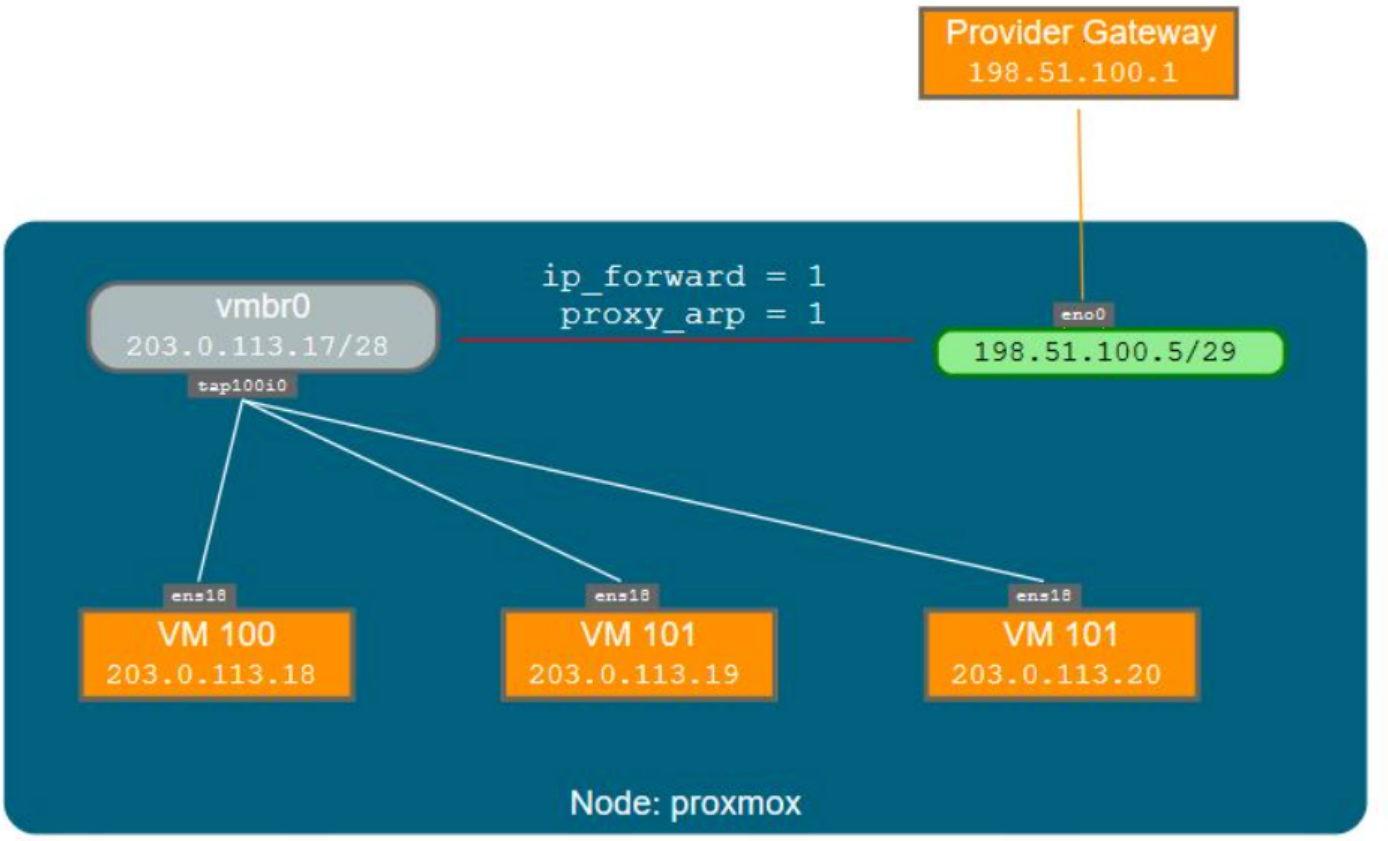
Aunque no es lo habitual podemos **asignar una IP mediante DHCP**:

```
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)

Licencia Creative Commons BY-NC-SA.

Configuración enrutada



Proxmox Server Solutions GmbH. *Configuración de red enrutada* (Todos los derechos reservados)

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

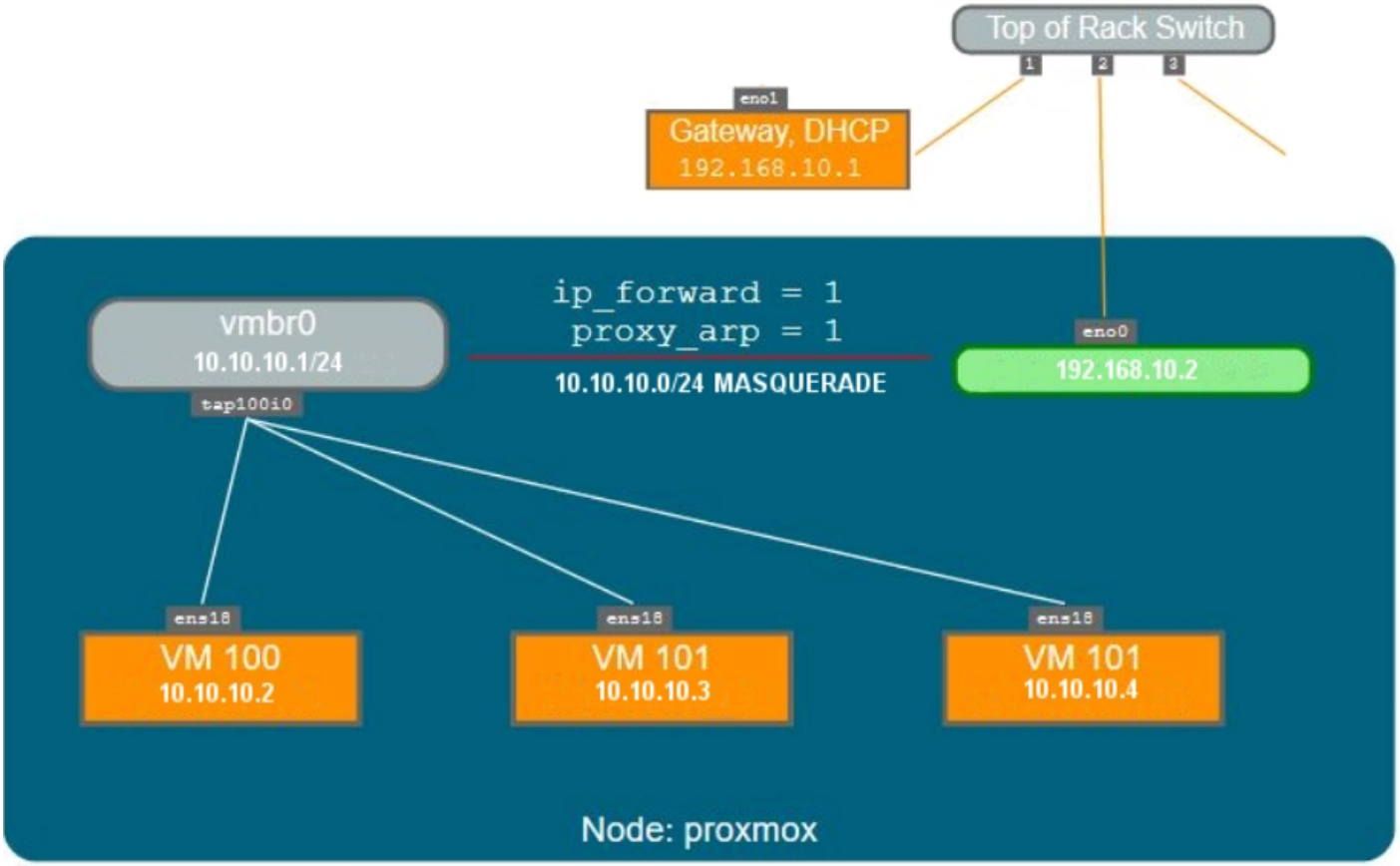
auto eno1
iface eno1 inet static
address 198.51.100.5
netmask 255.255.255.0
gateway 198.51.100.1
```

Licencia Creative Commons BY-NC-SA.

```
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

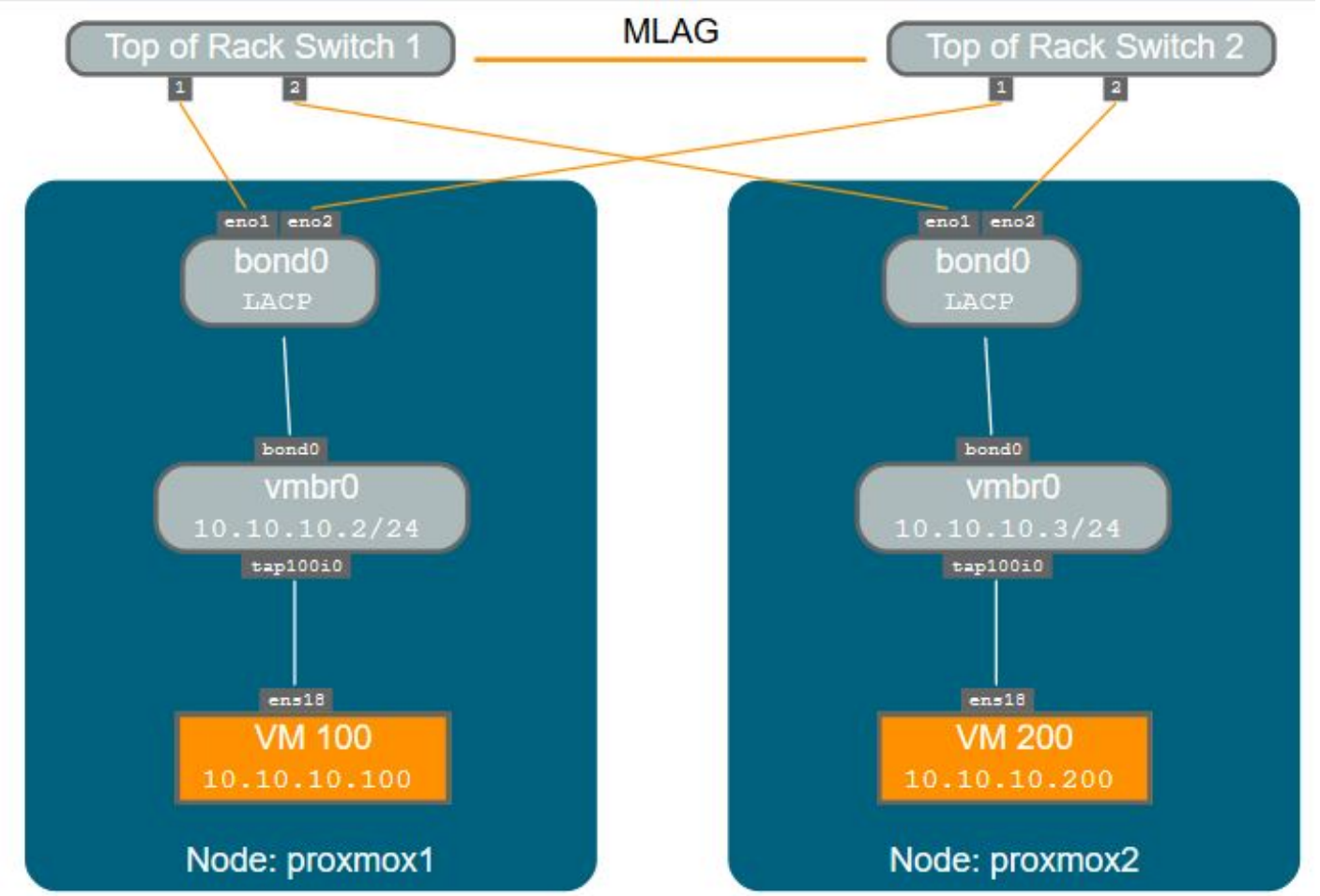
post-up echo 1 > /proc/sys/net/ipv4/ip_forward
post-up iptables -t nat -A POSTROUTING -s '10.10.10.0/24' -o eno1 -j MASQUERADE
post-down iptables -t nat -D POSTROUTING -s '10.10.10.0/24' -o eno1 -j MASQUERADE
```

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.



Proxmox Server Solutions GmbH *Configuración Linux Bonding (o modo balanceador)* (Todos los derechos reservados)

> 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:

Licencia Creative Commons BY-NC-SA.

```
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
```

Licencia Creative Commons BY-NC-SA.

```
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
```

Licencia Creative Commons BY-NC-SA.

```
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

Revisión #5
Creado 11 mayo 2024 11:35:31 por Daniel Cano Verdú
Actualizado 11 mayo 2024 21:13:27 por Daniel Cano Verdú

Licencia Creative Commons BY-NC-SA.