

HLC

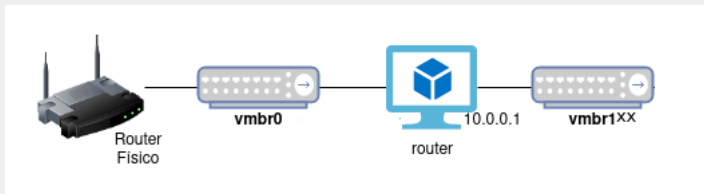
PRÁCTICA 7: TRABAJANDO CON REDES EN PROXMOX

PRÁCTICA 7

- Un **punto o bridge/switch** es un dispositivo de interconexión de redes.
- **Linux Bridge** es un software que tiene la misma funcionalidad que un bridge físico.
- Nuestras MV y contenedores se han conectado al puente **vmbro** que nos da salida a internet.
- Podemos conectar nuestras MV y contenedores a otros puentes para hacer escenarios más complejos.
- Cada alumno tendrá asignado un puente **vmbri1XX** para conectar sus MV y contenedores.

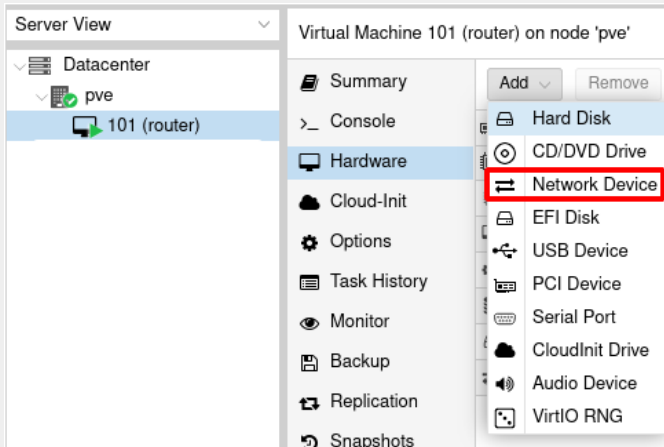
CONEXIÓN DE UNA MÁQUINA VIRTUAL INSTALADA DESDE UNA ISO A LA RED INTERNA

CONEXIÓN DE UNA MÁQUINA VIRTUAL A LA RED INTERNA



- La máquina **router** ya está creada y está conectada al bridge **vmbr0**, le vamos a añadir una nueva interfaz de red conectada al bridge **vmbr100** y la vamos a configurar con la ip **10.0.0.1/24**.

AÑADIR UNA NUEVA INTERFAZ DE RED A UNA MÁQUINA



AÑADIR UNA NUEVA INTERFAZ DE RED A UNA MÁQUINA

Elegimos al bridge al que va a estar conectada esta nueva interfaz (recuerda elegir el **driver VirtIO**):

Agregar: Dispositivo de red

Puente:

vmbr100

Modelo:

VirtIO (paravirtualizado)

Etiqueta VLAN:

no VLAN

Dirección MAC:

auto

Cortafuego:

☒

Ayuda

Avanzado ☐

Agregar

CONFIGURACIÓN DE LA NUEVA INTERFAZ

- En la red interna a la que hemos conectado la máquina no existe ningún **servidor DHCP**.
- Tendremos que hacer una configuración manual del direccionamiento de esta interfaz de red.

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default  
    link/ether fe:23:f9:26:c3:7e brd ff:ff:ff:ff:ff:ff  
    altname enp0s18  
    altname ens18  
    inet 172.22.0.249/16 metric 100 brd 172.22.255.255 scope global dynamic eth0  
        valid_lft 86171sec preferred_lft 86171sec  
    inet6 fe80::fc23:f9ff:fe26:c37e/64 scope link  
        valid_lft forever preferred_lft forever  
3: ens19: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000  
    link/ether 5e:bf:7f:3b:fb:6a brd ff:ff:ff:ff:ff:ff  
    altname enp0s19
```


CONFIGURACIÓN DE LA NUEVA INTERFAZ

- La nueva interfaz se llama **ens19** y es la que tenemos que configurar.
- Para ello modificamos el fichero **/etc/network/interfaces** y añadimos el siguiente contenido:

```
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
allow-hotplug ens18  
iface ens18 inet dhcp  
  
allow-hotplug ens19  
iface ens19 inet static  
    address 10.0.0.1  
    netmask 255.255.255.0
```

CONFIGURACIÓN DE LA NUEVA INTERFAZ

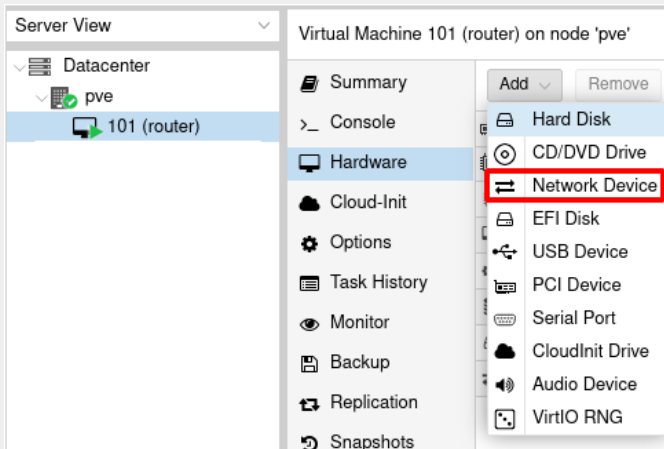
- Levantamos la interfaz con el comando **ifup ens19**

```
usuario@router:~$ sudo ifup ens19
usuario@router:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 22:34:88:db:44:d4 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 172.22.8.181/16 brd 172.22.255.255 scope global dynamic ens18
        valid_lft 86245sec preferred_lft 86245sec
    inet6 fe80::2034:88ff:fedb:44d4/64 scope link
        valid_lft forever preferred_lft forever
3: ens19: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 7e:1f:ed:13:df:6e brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    inet 10.0.0.1/24 brd 10.0.0.255 scope global ens19
        valid_lft forever preferred_lft forever
    inet6 fe80::7c1f:edff:fe13:df6e/64 scope link
        valid_lft forever preferred_lft forever
```

CONEXIÓN DE UNA MÁQUINA VIRTUAL CREADA DESDE UNA PLANTILLA DE “IMÁGENES” A LA RED INTERNA

AÑADIR UNA NUEVA INTERFAZ DE RED A UNA MÁQUINA

- Creamos una MV desde la imagen **debian-12-cloud**.
- Antes de arrancar añadimos una nueva interfaz.



CONFIGURACIÓN DE LA NUEVA INTERFAZ

- Nos vamos a **cloud-init** y configuramos la máquina.
- En **cloud-init** configuramos la segunda interfaz de red (**net1**):

Editar: Configuración de la red ⓧ

Dispositivo de red: **net1**

IPv4: ☒ Estático ☐ DHCP

IPv4/CIDR:

Puerta de enlace (IPv4):

IPv6: ☒ Estático ☐ DHCP ☐ SLAAC

IPv6/CIDR:

Puerta de enlace (IPv6):

CONFIGURACIÓN DE LA NUEVA INTERFAZ

- Compramos que ha sido configurada:

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 22:c1:62:43:00:91 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    altname ens18
    inet 172.22.8.186/16 metric 100 brd 172.22.255.255 scope global dynamic eth0
        valid_lft 86285sec preferred_lft 86285sec
    inet6 fe80::20c1:62ff:fe43:91/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 7a:26:85:b9:cb:0f brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    altname ens19
    inet 10.0.0.1/24 brd 10.0.0.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::7826:85ff:feb9:cb0f/64 scope link
        valid_lft forever preferred_lft forever
```

CONEXIÓN DE UN CONTENEDOR A LA RED INTERNA

AÑADIR UNA NUEVA INTERFAZ DE RED A UN CONTENEDOR

- Tenemos un contenedor conectado a **vmbro**, vamos a añadir una nueva red.

Contenedor 283 (cliente) en el nodo proxmox1 No Tags ▶ Iniciar

Resumen

>_ Consola

Recursos

Red

DNS

Opciones

Agregar

Eliminar

Editar

ID ↑	Nombre	Puente	Cortafu...	Etiquet...	Dirección MAC	Dirección IP	Puerta de enlace	MTU	Disconnected
net0	eth0	vmbro	Sí		4E:5B:E6:45:...	dhcp			No

AÑADIR UNA NUEVA INTERFAZ DE RED A UN CONTENEDOR

- Indicamos el **nombre de la interfaz**, el **punto de puente** y la **configuración estática**.

Agregar: Dispositivo de red (veth) ✕

Nombre:	eth1	IPv4:	<input checked="" type="radio"/> Estático <input type="radio"/> DHCP
Dirección MAC:	auto	IPv4/CIDR:	10.0.0.1/24
Punto de puente:	vmbr100 ▾	Puerto de enlace (IPv4):	
Etiqueta VLAN:	no VLAN ▴ ▾	IPv6:	<input checked="" type="radio"/> Estático <input type="radio"/> DHCP <input type="radio"/> SLAAC
Cortafuego:	<input checked="" type="checkbox"/>	IPv6/CIDR:	None
		Puerto de enlace (IPv6):	

? Ayuda

Avanzado ☐

Agregar

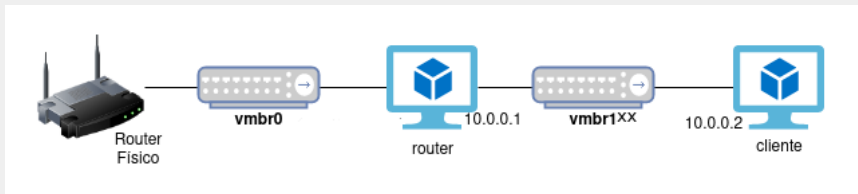
AÑADIR UNA NUEVA INTERFAZ DE RED A UN CONTENEDOR

- Accedemos al contenedor y ya tenemos la interfaz configurada:

```
root@router3:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0@if358: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 0a:4c:d2:ee:86:cf brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.22.9.105/16 brd 172.22.255.255 scope global dynamic eth0
        valid_lft 86394sec preferred_lft 86394sec
    inet6 fe80::84c:d2ff:feee:86cf/64 scope link
        valid_lft forever preferred_lft forever
3: eth1@if362: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether 96:23:ff:b2:1c:7b brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.0.0.1/24 brd 10.0.0.255 scope global eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::9423:ffff:feb2:1c7b/64 scope link
        valid_lft forever preferred_lft forever
```

CREACIÓN DE UNA MÁQUINA VIRTUAL/CONTENEDOR EN LA RED INTERNA

CREACIÓN DE UNA MV/CONTENEDOR EN LA RED INTERNA



- La máquina **router** ya la hemos creado.
- Ahora vamos a crear la máquina/contenedor **cliente** conectada sólo a la red interna al puente **vmbr100** con una **ip estática 10.0.0.X/24..**

- Si estamos creando una máquina virtual **desde una ISO** tendremos que tenerla conectada a **vmbro** para que tenga acceso a internet, y podemos realizar la instalación:
- Si estamos creando una máquina virtual desde las plantillas:
 - ▶ Una vez creada, **antes de ejecutarla**,
 - ▶ Elimino la interfaz conectada a **vmbro**.
 - ▶ Añado una interfaz conectada a **vmbro100**.
 - ▶ Y la configuro de **forma estática** desde el **cloud-init**.

CREACIÓN DE UNA MÁQUINA VIRTUAL EN LA RED INTERNA

- Si estamos creando un contenedor, en su creación elegimos el puente al que nos conectamos y la configuración estática:

Crear: Contenedor LXC

General Plantilla Discos CPU Memoria **Red** DNS Confirmar

Nombre:

Dirección MAC:

Puente:

Etiqueta VLAN:

Cortafuego: ☒

IPv4: ☒ Estático ☐ DHCP

IPv4/CIDR:

Puerta de enlace (IPv4):

IPv6: ☒ Estático ☐ DHCP ☐ SLAAC

IPv6/CIDR:

Puerta de enlace (IPv6):

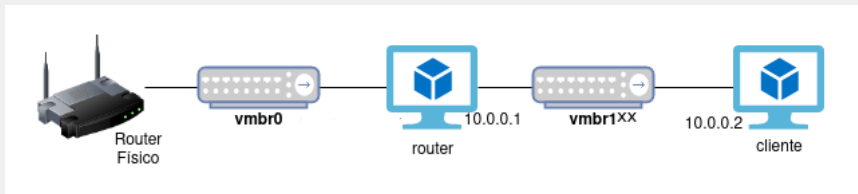
Ayuda Avanzado ☐ Atrás Siguiente

COMPROBACIÓN DE QUE TENEMOS CONECTIVIDAD ENTRE LOS EQUIPOS CONECTADOS AL MISMO PUENTE

- Por ejemplo desde el contenedor que hemos creado podemos hacer ping a otra máquina:

```
root@c2:~# ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.115 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.107 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.099 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=0.090 ms
```

CONFIGURACIÓN DE ESCENARIO ROUTER/NAT



- **¿Qué tenemos que hacer para que las mv/contenedores conectadas a la red interna tengan salida a internet?:**
 - ▶ La puerta de enlace de **cliente** tiene que ser la dirección del **router**, en nuestro caso la **10.0.0.1/24**.
 - ▶ Configura la máquina **router** para que haga SNT y enrrote los paquetes.

CONFIGURACIÓN DE LA PUERTA DE ENLACE EN UNA MV CREADA DESDE UNA ISO

- Modificamos el fichero **/etc/network/interfaces** de esta forma:

```
allow-hotplug ens19
iface ens19 inet static
    address 10.0.0.2
    netmask 255.255.255.0
    gateway 10.0.0.1
```

- Y reiniciamos la red con **systemctl restart networking**.
- Y comprobamos que ya tenemos la ruta por defecto:

```
root@router:~# ip r
default via 10.0.0.1 dev ens19 onlink
10.0.0.0/24 dev ens19 proto kernel scope link src 10.0.0.2
```

CONFIGURACIÓN DE LA PUERTA DE ENLACE EN UNA MV CREADA DESDE UNA PLANTILLA DE “IMÁGENES”

- Antes de **iniciar la máquina** configuramos la red en el **cloud-init** indicando la **puerta de enlace**:

Editar: Configuración de la red

Dispositivo de red: net1

IPv4: ☒ Estático ☐ DHCP

IPv4/CIDR:

Puerta de enlace (IPv4):

IPv6: ☒ Estático ☐ DHCP ☐ SLAAC

IPv6/CIDR:

Puerta de enlace (IPv6):

Aceptar Reset

CONFIGURACIÓN DE LA PUERTA DE ENLACE EN UN CONTENEDOR

- Al crear el contenedor configuramos la red indicando la **puerta de enlace**:

Crear: Contenedor LXC

General Plantilla Discos CPU Memoria **Red** DNS Confirmar

Nombre:	eth0	IPv4:	<input checked="" type="radio"/> Estático <input type="radio"/> DHCP
Dirección MAC:	auto	IPv4/CIDR:	10.0.0.2/24
Puente:	vmbr100	Puerta de enlace (IPv4):	10.0.0.1
Etiqueta VLAN:	no VLAN	IPv6:	<input checked="" type="radio"/> Estático <input type="radio"/> DHCP <input type="radio"/> SLAAC
Cortafuego:	<input checked="" type="checkbox"/>	IPv6/CIDR:	None
		Puerta de enlace (IPv6):	

Ayuda Avanzado ☐ Atrás Siguiente

- En el **router**:
- Activamos el **bit de forwarding**.
- Modificamos el fichero **/etc/sysctl.conf** y descomentamos la línea **net.ipv4.ip_forward=1**.
- Para confirmar el cambio ejecutamos **sysctl -p**.

- En el **router**, añadimos una regla en el cortafuego para realizar el SNAT. Para ello instalamos iptables:

apt install iptables

- Añadimos en el fichero **/etc/network/interface** la regla:

**post-up iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o ens18 -j
MASQUERADE**

Los paquetes que vengan de la red interna 10.0.0.0/24, y saliendo por la primera interfaz se enmascaran, es decir su dirección de origen se cambian por la dirección de la interfaz de salida.

- Reiniciamos la red para activar la configuración. **Nota: Cambia el nombre de la interfaz de red o cualquier otro dato si en tu escenario es necesario.**

ifdown ens19 ifup ens19

- Comprobamos que la regla se ha ejecutado:

```
# iptables -L -n -t nat
...

Chain POSTROUTING (policy ACCEPT)
target      prot opt source                destination
MASQUERADE  all  --  10.0.0.0/24            0.0.0.0/0
```

- Ahora podemos comprobar que el cliente tiene acceso al exterior:

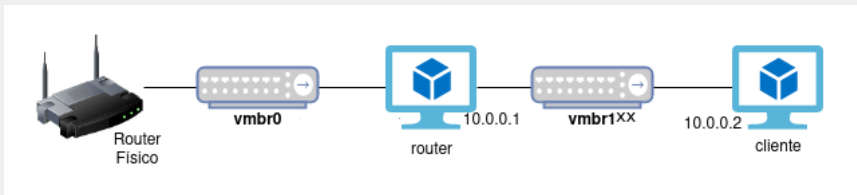
```
cliente:~$ ping 1.1.1.1  
PING 1.1.1.1 (1.1.1.1) 56(84) bytes of data.  
64 bytes from 1.1.1.1: icmp_seq=1 ttl=56 time=9.05 ms
```


- Para que los **clientes** tengan resolución DNS:
- Añadimos al fichero **/etc/resolv.conf** esta línea:
nameserver 172.22.0.1
- Ya ahora puedes ejecutar un **ping dit.gonzalonazareno.org**.
- Ejecuta un **apt update** para comprobar que todo funciona.

PRÁCTICA 7

¿QUÉ TIENES QUE HACER?

- Vamos a crear el siguiente escenario:



- Donde la máquina **router** se crea desde la imagen **debian-12-cloud**.
- La máquina cliente es un contenedor.
- Realiza la configuración mostrada en el esquema y comprueba que la máquina cliente le puede hacer ping al router.
- **VOLUNTARIA:** Configura la máquina **router** para que el cliente tenga salida al exterior.

¿QUÉ TIENES QUE ENTREGAR?

1. La configuración ip de la máquina router.
2. La configuración ip del contenedor.
3. Una captura de pantalla haciendo ping al router desde el cliente.
4. Si has hecho la parte voluntaria:
 - La ruta por defecto del cliente.
 - Una comprobación de que el cliente tiene acceso al exterior usando un nombre (un ping a un nombre de dominio).