



# ROUTER

## ☐ Configuración Hardware VM

- Antes de empezar a configurar el netplan es importante ir al apartado de hardware y agregar una tarjeta de red, ya que por defecto las máquinas virtuales solo tiene uno

Add ▾

Virtual Machine 100 (Router) on node 'mlb' No Tags

Summary

> Console

**Hardware**

Cloud-Init

Options

Task History

Monitor

Backup

Replication

Snapshots

Firewall

Permissions

Add ▾ Remove Edit Disk Action ▾ Revert

Memory	2.00 GiB
Processors	1 (1 sockets, 1 cores) [x86-64-v2-AES]
BIOS	Default (SeaBIOS)
Display	Default
Machine	Default (i440fx)
SCSI Controller	VirtIO SCSI single
CD/DVD Drive (ide2)	local:iso/ubuntu-22.04.5-live-server-amd64.iso,media=cdrom,size=2086842K
Hard Disk (scsi0)	local-lvm:vm-100-disk-0,ioread=1,size=14G
Network Device (net0)	virtio=BC:24:11:D8:EE:AE,bridge=vbr0,firewall=1
Network Device (net1)	virtio=BC:24:11:B6:F4:97,bridge=vbr1,firewall=1

Network Device (net0)

virtio=BC:24:11:D8:EE:AE,bridge=vbr0,firewall=1

Network Device (net1)

virtio=BC:24:11:B6:F4:97,bridge=vbr1,firewall=1

Ambos son puentes también llamados **“linux bridge”**

vmr0 > 100.77.20.0/24 > Conecta a la red de fuera a través del router mediante el host

vmr1 > 192.168.1.0/24 > Conecta a la red interna para que las máquinas virtuales se puedan comunicar entre sí y a la vez salir.

```
QEMU (Router) - noVNC - Google Chrome
No es seguro https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=100&vmname=Router&node=mlb&resize=off&cmd=
ping 8.8.8.8
apt install qemu-guest-agent
```

Comprobamos que tengamos conexión a internet para poder instalar qemu-guest-agent (sirve para poder visualizar dentro de proxmox que direcciones ip tiene las mv)

```
apt install net-tools
```

\*\*\*Opcional instalar net-tools para poder monitorear la red, supervisar servicios, máquinas, tráfico de red y dispositivos de red y no tener porqué utilizar ip -a todo el tiempo  
La manera de poder configurar direcciones ips cuando no tenemos dhcp lo suyo es hacerlo usando NETPLAN :D

```
vim /etc/netplan/50-cloud-init.yaml
```

vim: Editor de texto

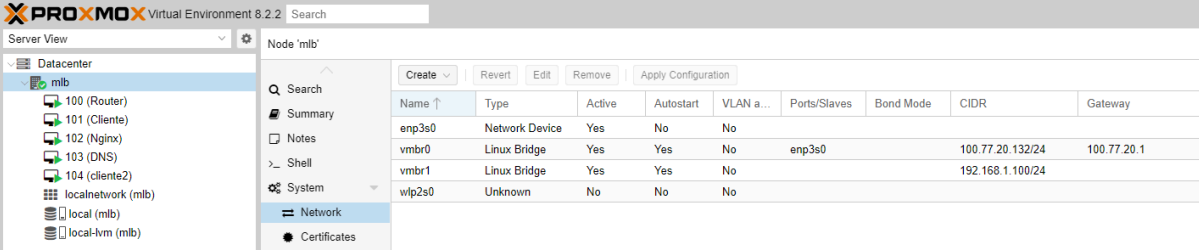
Hacemos una copia

```
cp 50-cloud-init.yaml.bkp 00-installer-config.yaml
```

Es muy importante en caso de tener el archivo con nombre 50-cloud-init.yaml (viene por defecto) hay que renombrarlo ya que en caso de reiniciar la máquina, dicho archivo no se guarda, se vuelve a generar.

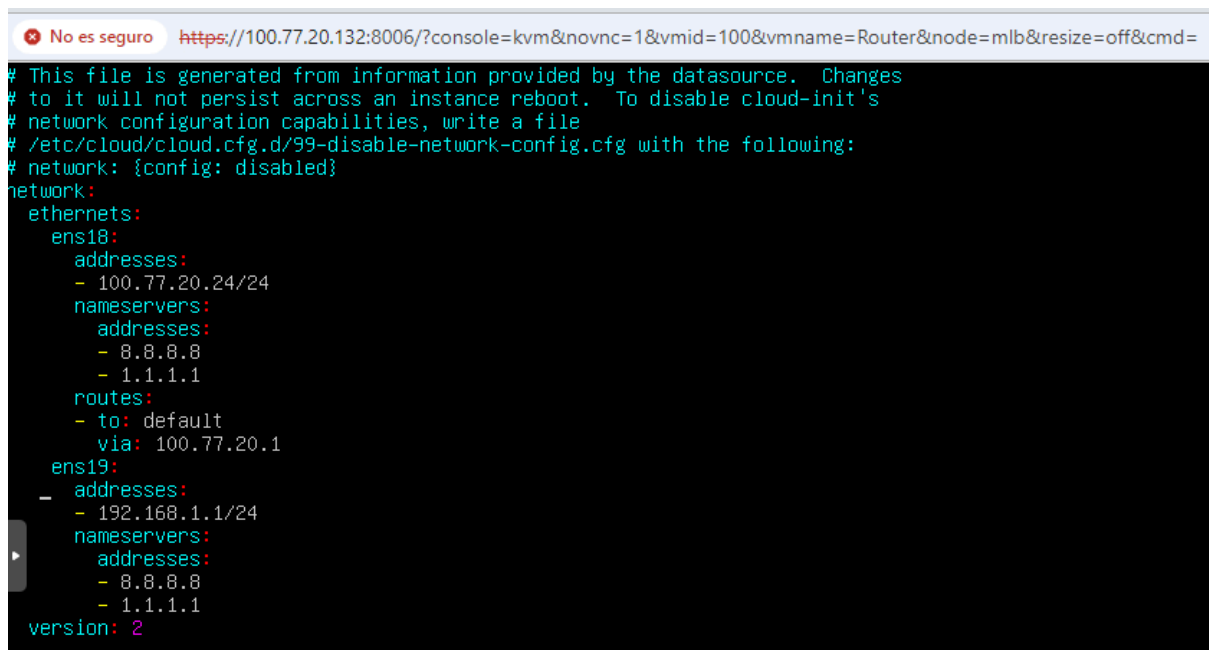
Deberíamos de editar el archivo de configuracion de esta manera:

OJO: Teniendo en cuenta los rangos de ip que tenemos establecidos en proxmox



The screenshot shows the Proxmox Virtual Environment 8.2.2 interface. On the left, the 'Server View' sidebar shows a tree structure with 'Datacenter' expanded, containing 'mlb' and 'local'. The 'mlb' node is selected. The main panel displays the configuration for the 'mlb' node, specifically the 'Network' tab. It shows a table of network interfaces with columns: Name, Type, Active, Autostart, VLAN a..., Ports/Slaves, Bond Mode, CIDR, and Gateway.

Name	Type	Active	Autostart	VLAN a...	Ports/Slaves	Bond Mode	CIDR	Gateway
enp3s0	Network Device	Yes	No	No				
vmbr0	Linux Bridge	Yes	Yes	No	enp3s0		100.77.20.132/24	100.77.20.1
vmbr1	Linux Bridge	Yes	Yes	No			192.168.1.100/24	
wlp2s0	Unknown	No	No	No				



```
# This file is generated from information provided by the datasource. Changes
# to it will not persist across an instance reboot. To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    ens18:
      addresses:
        - 100.77.20.24/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
      routes:
        - to: default
          via: 100.77.20.1
    ens19:
      addresses:
        - 192.168.1.1/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 1.1.1.1
  version: 2
```

Como podemos ver en la imagen superior estamos editando ambas interfaces de red. Esta edición de interfaces nos permitirá la comunicación entre máquinas para que tengan comunicación con el host a partir del mismo router.

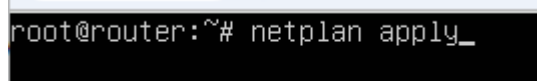
En la interfaz ens18 asignamos la dirección ip 100.77.20.24/24

Así mismo definimos los dns, como las rutas que en este caso sería la salida (gateway) de esa interfaz

Por otra parte en la interfaz ens 19 asignamos la dirección ip 192.168.1.1/24, aquí no habrá salida ya que ens 18 será la que nos la brinde.

[—> En un futuro explicaremos cómo hacerlo.](#)

Una vez que está bien la configuración aplicamos la configuración con el siguiente comando.



```
root@router:~# netplan apply_
```

\*\*\*OJO > En algunos casos es posible que al momento de aplicar la configuración salga un error debido a una “dependencia” llamada openvswitch

Se corrige con el siguiente comando:

- **sudo apt install openvswitch-switch-dpdk**

Ahora sin problemas deberían de poder aplicar la configuración de netplan :D

```

No es seguro https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=100&vmname=Router&node=mlb&resize=off&cmd=
root@router:~# ifconfig
ens18: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 100.77.20.24 netmask 255.255.255.0 broadcast 100.77.20.255
    inet6 fe80::be24:11ff:fed8:eeae prefixlen 64 scopeid 0x20<link>
    ether bc:24:11:d8:ee:ae txqueuelen 1000 (Ethernet)
    RX packets 143241 bytes 42209731 (42.2 MB)
    RX errors 0 dropped 12132 overruns 0 frame 0
    TX packets 3988 bytes 344483 (344.4 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens19: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.1 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::be24:11ff:feb6:f497 prefixlen 64 scopeid 0x20<link>
    ether bc:24:11:b6:f4:97 txqueuelen 1000 (Ethernet)
    RX packets 4213 bytes 367784 (367.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5637 bytes 23154236 (23.1 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 162 bytes 11992 (11.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 162 bytes 11992 (11.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@router:~# _
```

En este punto estamos listos para hacer la salida del tráfico de la red interna a través de la red externa, es decir de **ens19** a **ens18**, para ello tenemos que establecer unas reglas con iptables.

```

root@router:~# sudo apt install iptables
```

Una vez instalado procedemos a setear las reglas de la siguiente manera:

```

root@router:~# iptables -t nat -A POSTROUTING -o ens18 -j MASQUERADE
```

- El comando anterior configura una regla de NAT (Network Address Translation) en la tabla nat de iptables.
- Se añade a la cadena **POSTROUTING**, que se aplica a los paquetes justo antes de salir del sistema.
- La opción **-o ens18** especifica que la regla se aplica a los paquetes que salen por la interfaz de red ens18.
- La acción **-j MASQUERADE** indica que el origen de estos paquetes será reemplazado por la dirección IP de la interfaz, permitiendo que dispositivos en una red privada accedan a internet usando una dirección IP pública compartido

Este comando permite a los dispositivos en una red interna enviar paquetes a través del servidor, donde sus direcciones IP privadas son reemplazadas (enmascaradas) por la dirección IP pública o externa del servidor.

```
root@router:~# iptables -t nat -L
```

Aquí listamos todas las reglas que tenemos de **iptables**, pero la que nos importa es esta.

```
Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE all  --  anywhere              anywhere
```

```
root@router:~# apt install iptables-persistent -y
```

El comando se utiliza para instalar el paquete “iptables-persistent” y permite que las iptables se guarden y se carguen automáticamente sin importar el reinicio de la máquina. La y- confirma la instalación.

En caso de que por alguna casualidad ya tengamos iptables-persistent  
NO PASA NADA

Podemos guardar las reglas de la siguiente manera:

```
root@router:~# iptables-save
# Generated by iptables-save v1.8.7 on Mon Oct 21 17:18:02 2024
*nat
:PREROUTING ACCEPT [57171:5229250]
:INPUT ACCEPT [402:96221]
:OUTPUT ACCEPT [70:4434]
:POSTROUTING ACCEPT [43:2544]
-A PREROUTING -s 100.77.20.132/32 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.1.10:80
-A PREROUTING -s 100.77.20.132/32 -i vmr0 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.1.10:80
-A PREROUTING -s 100.77.20.132/32 -i ens18 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.1.15:80
-A PREROUTING -s 100.77.20.0/24 -i ens18 -p tcp -m tcp --dport 80 -j DNAT --to-destination 192.168.1.15:80
-A POSTROUTING -o ens18 -j MASQUERADE
-A POSTROUTING -d 192.168.1.10/32 -p tcp -m tcp --dport 80 -j MASQUERADE
-A POSTROUTING -d 192.168.1.10/32 -o vmr0 -p tcp -m tcp --dport 80 -j MASQUERADE
-A POSTROUTING -d 192.168.1.15/32 -o ens18 -p tcp -m tcp --dport 80 -j MASQUERADE
COMMIT
# Completed on Mon Oct 21 17:18:02 2024
root@router:~# _
```

Solo nos falta un último paso para poder permitir que las futuras máquinas o las que ya tengamos, puedan conectarse a internet sería editar el siguiente archivo el cual permite varias opciones Sin embargo la que nos interesa es aquella que permite el reenvío de paquetes por ipv4. Basta con descomentar esta línea.

```
root@router:~# vim /etc/sysctl.conf _
```

No es seguro <https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=100&vmname=Router&node=mlb&resize=off&cmd=>

```
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
#
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
#
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lun.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
#
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1 #### DESCOMENTAMOS ESTA LINEA####_
#
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
#
#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
# settings are disabled so review and enable them as needed.
#
# Do not accept ICMP redirects (prevent MITM attacks)
#net.ipv4.conf.all.accept_redirects = 0
#net.ipv6.conf.all.accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
#net.ipv4.conf.all.secure_redirects = 1
#net.ipv6.conf.all.secure_redirects = 1
```

Después de editar este archivo falta recargarlo, de lo contrario no se podrá realizar el reenvío de paquetes > : D

Usamos el comando para editar el archivo usando vim, este archivo se usa para configurar los parámetros del kernel que persisten después de reiniciar.

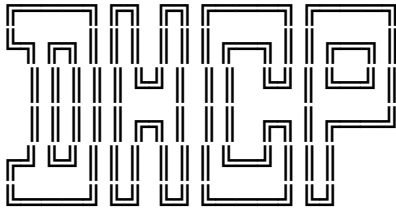
```
root@router:~# sysctl -p .
```

Guardamos el archivo y aplicamos las nuevas configuraciones usando el comando “sysctl -p” para cargar los nuevos cambios sin reiniciar

```
root@router:~# sysctl -p
net.ipv4.ip_forward = 1
root@router:~#
```

Y como vemos el output se aplica la regla sin necesidad de reiniciar.






Para empezar a configurar el servicio de dhcp en nuestro router es evidente que necesitamos haber realizado toda la configuración anterior.

Necesitaremos instalar el siguiente paquete como se muestra a continuación

```
root@router:~# apt install isc-dhcp-server
```

Una vez instalado es necesario comprobar el estado del servicio...

```
root@router:~# systemctl status isc-dhcp-server
```



```
root@router:~# systemctl status isc-dhcp-server
• isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-21 14:18:25 UTC; 3h 14min ago
     Docs: man:dhcpd(8)
    Main PID: 828 (dhcpd)
      Tasks: 4 (limit: 2226)
     Memory: 4.8M
        CPU: 97ms
    CGroup: /system.slice/isc-dhcp-server.service
            └─828 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcp/dhcpd.conf ens19
```

```
root@router:~# cp /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.bkp
```

Este comando nos crea una copia de seguridad del archivo de configuración del server DHCP en el mismo directorio.

[alinagchacon@gmail.com](mailto:alinagchacon@gmail.com)

```
root@router:~# vim /etc/dhcp/dhcpd.conf_
```

Con este comando buscamos abrir el archivo dhcp.conf utilizando el editor de texto vim como lo mostramos en la siguiente imagen

```
No es seguro https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=100&vmname=Router&node=mlb&resize=off&cmd=

#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

#default-lease-time 600;
#max-lease-time 7200;

#ddns-update-style none;

#Falta configurar la IP DNS en este archivo
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.10 192.168.1.100;
    option subnet-mask 255.255.255.0;
    option routers 192.168.1.1;
    option broadcast-address 192.168.1.255;
    option domain-name-servers 192.168.1.2, 8.8.8.8;
    default-lease-time 600;
    max-lease-time 7200;

    ddns-update-style interim;
    ddns-updates on;
    ddns-domainname "tas.io.";
    ddns-rev-domainname "in-addr.arpa.";
    update-static-leases on;
    ignore client-updates;

    authoritative;
    option domain-name "tas.io";
}

# host DNS {
#     hardware ethernet bc:24:11:ff:69:59;
#     fixed-address 10.20.40.4;
# }
#
# host Firebase {
#     hardware ethernet bc:24:11:24:01:5e;
#     fixed-address 10.20.40.5;
# }
#
# host NGINX{
#     hardware ethernet bc:24:11:66:26:12;
#     fixed-address 10.20.40.21;
# }
}
```

El archivo de configuración `dhcpd.conf` es utilizado por el servidor DHCP para asignar direcciones IP y otros parámetros de red a los clientes dentro de una subred específica. Aquí te explico cada parte del contenido:

- **subnet 192.168.1.0 netmask 255.255.255.0 { ... }:** Se define una subred con la dirección base `192.168.1.0` y una máscara de subred `255.255.255.0`. Y se crea dentro la configuración para los clientes que se conecten a esta subred.
- **range 192.168.1.10 192.168.1.100;** Especificamos el rango de direcciones IP que el servidor DHCP puede asignar dinámicamente a los clientes, desde la 192.168.1.10 hasta 192.168.1.100.
- **option subnet-mask 255.255.255.0;** Define la máscara de subred que se asignará a los clientes, una de tipo C.
- **option routers 192.168.1.1;** Especifica la dirección IP del router predeterminado (gateway) que los clientes deben usar para acceder a otras redes.
- **option broadcast-address 192.168.1.255;** Indica la dirección de broadcast para la subred, -que es utilizada para enviar mensajes a todos los dispositivos en la red-.
- **option domain-name-servers 192.168.1.2, 8.8.8.8;** Define las direcciones IP de los servidores DNS que los clientes deben usar para resolver nombres de dominio, en este caso, `192.168.1.2` y el servidor público de Google DNS `8.8.8.8`.

- **default-lease-time 600**; Establece el tiempo predeterminado (en segundos) durante el cual un cliente puede usar una dirección IP antes de solicitar su renovación, en este caso, 600 segundos (10 minutos).
- **max-lease-time 7200**; Define el tiempo máximo (en segundos) que un cliente puede retener una dirección IP, en este caso, 7200 segundos (2 horas).
- **ddns-update-style interim**; y **ddns-updates on**; Configuran el estilo de actualización del DNS dinámico (DDNS) y habilitan las actualizaciones DDNS automáticas respectivamente.
- **ddns-domainname "tas.io."**; y **ddns-rev-domainname "in-addr.arpa."**; Especifican los nombres de dominio para las actualizaciones DDNS directas e inversas.
- **update-static-leases on**; Permite actualizaciones DDNS para arrendamientos estáticos.
- **ignore client-updates**; Indica al servidor DHCP que ignore cualquier solicitud del cliente para actualizar su información DNS.
- **authoritative**; Declara que este servidor DHCP es autoritativo para la subred especificada, lo que significa que responderá a todas las solicitudes DHCP en esta red.
- **option domain-name "tas.io"**; Establece el nombre de dominio que se asignará a los clientes dentro de esta subred.

La aparte Comentada (en azul) es un ejemplo de cómo podríamos asignar direcciones ips mediante las direcciones macs de las interfaces de red.

Este archivo configura cómo el servidor DHCP asigna direcciones IP y proporciona parámetros importantes de red a los dispositivos dentro de la subred `192.168.1.x`

Dentro de /etc/default/isc-dhcp-server

Antes de cualquier cosa revisamos nuestras ips, que todo cuadre con lo configurado dentro de netplan.

```

No es seguro https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=100&vmname=Router&node=mlb&resize=off&cmd=
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="ens19"
INTERFACESv6=""
#Aquí ponemos ens19 pq es la interfaz de nuestra red interna :D

```

Aquí se especifican las interfaces de red en las que el servidor DHCP (isc-dhcp-server) escuchará y proporcionará servicios DHCP.

El servidor DHCP está configurado para operar en la interfaz de red "ens19".

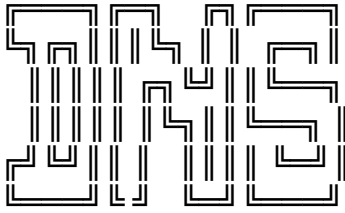
Esto significa que proporcionará direcciones IPV4 a los dispositivos conectados a esta interfaz.

El servicio DHCP solo funcionará para IPv4 en la red conectada a ens19.

No se ofrecerá servicio DHCP para IPv6 en ninguna interfaz.

```
root@router:~# more /var/lib/dhcp/dhcpd.leases
```

El archivo **/var/lib/dhcp/dhcpd.leases** es utilizado por el servidor DHCP para guardar información sobre las concesiones de direcciones IP que ha asignado a los clientes. Este archivo es un registro persistente que se actualiza cada vez que se adquiere, renueva o libera una concesión de IP. +



Necesitamos comprobar nuestra conexión a internet para poder instalar bind9.

```
No es seguro https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=103&vmname=DNS&node=mlb&resize=off&cmd=
dns@dns:~$ sudo apt install bind9_
```

Otro paquete que nos servirá es “**dnsutils**” que contiene instrucciones como nslookup. (no obstante es opcional).

Una vez instalado es necesario comprobar el estado del servicio.

```
root@dns:/home/dns# systemctl status bind9
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2024-10-21 14:19:54 UTC; 1 day 4h ago
     Docs: man:named(8)
  Process: 818 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 864 (named)
    Tasks: 5 (limit: 2226)
   Memory: 12.9M
      CPU: 5.385s
   CGroup: /system.slice/named.service
           └─864 /usr/sbin/named -u bind -4

oct 22 19:05:14 dns named[864]: timed out resolving 'ntp.ubuntu.com/A/IN': 1.1.1.1#53
oct 22 19:05:14 dns named[864]: timed out resolving 'ntp.ubuntu.com/AAAA/IN': 1.1.1.1#53
oct 22 19:09:16 dns named[864]: timed out resolving 'ntp.ubuntu.com/A/IN': 8.8.8.8#53
oct 22 19:09:16 dns named[864]: timed out resolving 'ntp.ubuntu.com/AAAA/IN': 8.8.8.8#53
oct 22 19:09:21 dns named[864]: timed out resolving 'ntp.ubuntu.com/A/IN': 1.1.1.1#53
oct 22 19:09:21 dns named[864]: timed out resolving 'ntp.ubuntu.com/AAAA/IN': 1.1.1.1#53
oct 22 19:09:28 dns named[864]: timed out resolving 'ntp.ubuntu.com/A/IN': 8.8.8.8#53
oct 22 19:09:28 dns named[864]: timed out resolving 'ntp.ubuntu.com/AAAA/IN': 8.8.8.8#53
oct 22 19:09:32 dns named[864]: timed out resolving 'ntp.ubuntu.com/A/IN': 1.1.1.1#53
oct 22 19:09:32 dns named[864]: timed out resolving 'ntp.ubuntu.com/AAAA/IN': 1.1.1.1#53
```

Si nos fijamos bien, el status nos dice que está volviendo con zonas invertidas predeterminadas y con dirección 8.8.8.8 (google) y 1.1.1.1 (cloudflare).

Antes de empezar tenemos que crear un respaldo de los archivos de configuración originales, para que en caso de que se estropee algo se pueda empezar de nuevo.

```
No es seguro https://100.77.20.132:8006/?console=kvm&novnc=1&vmid=103&vmname=DNS&node=mlb&resize=off&cmd=
root@dns:/home/dns# cp /etc/bind/named.conf.local /etc/bind/named.conf.local.bkp
```

El archivo named.conf.local es un archivo de configuración que se utiliza para definir las zonas de dominio. Una zona de dominio es una parte del espacio de nombres de dominio

que se administra de manera independiente. El archivo `named.conf.local` también se utiliza para definir los servidores DNS secundarios y los servidores de nombres raíz.

```
root@dns:/home/dns# cat /etc/bind/named.conf.local_
```

Para configurar este archivo, se deben seguir algunos pasos clave:

- **1. Crear una zona:** En el archivo `/etc/bind/named.conf.local`, se debe agregar una sección que defina la zona que se va a configurar. Esto incluye el nombre de dominio y la dirección IP del servidor.
- **2. Crear registros:** Una vez creada la zona, se pueden crear registros DNS para los servidores y dispositivos en esa zona. Los registros pueden incluir registros A (que asignan una dirección IP a un nombre de dominio) y registros CNAME (que asignan un nombre de dominio a otro nombre de dominio).
- **3. Configurar la resolución inversa:** También se puede configurar la resolución inversa en el archivo `/etc/bind/named.conf.local`. Esto permite que el servidor resuelva una dirección IP en un nombre de dominio.

Editamos el fichero `/etc/bind/named.conf.local` con tu editor de texto favorito. (nvim en nuestro caso).

2. Añade la siguiente línea al archivo:

```
zone «tas.io» {
```

3. Especifica el tipo de zona que estás configurando. En este ejemplo, estamos configurando una zona directa, por lo que añadimos la siguiente línea:

```
type master;
```

4. Especifica la ruta donde se encuentra el archivo de zona. En este caso, hemos creado un archivo llamado `tudominio.com.zone` en la ruta `/etc/bind/zones/`.

```
file «/etc/bind/zones/db.tas.io»;
```

5. Añade la siguiente línea para establecer los permisos de acceso al archivo de zona:

```
allow-update { none; };
```

6. Cierra el bloque de zona con la siguiente línea:

```
};
```

```
root@dns:/home/dns# cat /etc/bind/named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "tas.io" IN {
    type master;
    file "/etc/bind/zones/db.tas.io";
};
zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/zones/db.1.168.192";
};
```

Con el siguiente sirve para chequear la sintaxis de los ficheros de configuración de BIND. En el chequeo incluye aquellos ficheros de la instrucción *include*.

```
root@dns:/home/dns# named-checkconf
root@dns:/home/dns#
```

Como en este caso no nos devuelve ningún error, podemos proseguir, de lo contrario es muy necesario que corrijamos el error.

Una vez que hayas terminado de configurar `/etc/bind/named.conf.local`, es hora de crear el archivo de zona. Este archivo contiene los registros DNS para el dominio que acabas de configurar. Para crear este archivo, sigue los siguientes pasos:

1. Crea el archivo de zona en la ruta `/etc/bind/zones/`. En este ejemplo, lo hemos llamado `tas.io`.
2. Abre el archivo con tu editor de texto favorito.
3. Añade los siguientes registros DNS para tu dominio:

**\$TTL 86400**

**@ IN SOA ns.tas.io. root.tas.io. (**

**1 ; Serial**

**604800 ; Refresh**

**86400 ; Retry**

**2419200 ; Expire**

**604800 ) ; Minimum TTL**

**@ IN NS ns.tas.io.**

**@ IN A 192.168.1.2**

**ns IN A 192.168.1.2**

```
root@dns:/home/dns# cat /etc/bind/zones/db.tas.io
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns.tas.io. root.tas.io. (
                        2      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.tas.io.
@         IN      A        192.168.1.2
ns        IN      A        192.168.1.2
root@dns:/home/dns#
```



```

root@dns:/home/dns# cat /etc/bind/zones/db.1.168.192
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      tas.io. root.tas.io. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       tas.io.
2         IN      PTR      tas.io.
root@dns:/home/dns# _

```

Después de configurar ambas zonas. Es necesario ,que al igual que con el archivo named.conf.locale, revisar que ambas estén bien configuradas. Para ello solo sigue los siguientes pasos.

```

root@dns:/home/dns# named-checkzone 1.168.192.in-addr-arpa /etc/bind/zones/db.1.168.192

```

```

root@dns:/home/dns# named-checkzone 1.168.192.in-addr-arpa /etc/bind/zones/db.1.168.192
zone 1.168.192.in-addr-arpa/IN: loaded serial 2
OK
root@dns:/home/dns#

```

```

root@dns:/home/dns# named-checkzone tas.io /etc/bind/zones/db.tas.io _

```

```

root@dns:/home/dns# named-checkzone tas.io /etc/bind/zones/db.tas.io
zone tas.io/IN: loaded serial 2
OK
root@dns:/home/dns#

```

Por último quedaría editar en fichero /etc/resolv.conf.

Ojo. Verificar si el archivo tiene un link a otro fichero /run/systemd/resolve/resolv.conf.

Esto es necesario ya que por defecto va a resolver direcciones dns por la ip 127.0.0.53.

```
# This is /run/systemd/resolve/resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.1.2
search tas.io
```

Como se muestra aquí.

```
root@dns:/home/dns# cat /etc/resolv.conf
# This is /run/systemd/resolve/resolv.conf managed by man:systemd-resolved(8).
# Do not edit.
#
# This file might be symlinked as /etc/resolv.conf. If you're looking at
# /etc/resolv.conf and seeing this text, you have followed the symlink.
#
# This is a dynamic resolv.conf file for connecting local clients directly to
# all known uplink DNS servers. This file lists all configured search domains.
#
# Third party programs should typically not access this file directly, but only
# through the symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a
# different way, replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.1.2
search tas.io
root@dns:/home/dns# _
```

```
root@dns:/home/dns# systemctl restart bind9
root@dns:/home/dns#
```

Por último solo queda reiniciar el servicio bind9.

Revisamos el estatus.

```
root@dns:/home/dns# systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-10-22 19:27:24 UTC; 15s ago
     Docs: man:named(8)
   Process: 13901 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
   Main PID: 13902 (named)
    Tasks: 4 (limit: 2226)
   Memory: 5.1M
      CPU: 35ms
   CGroup: /system.slice/named.service
           └─13902 /usr/sbin/named -u bind -4

oct 22 19:27:24 dns named[13902]: zone 255.in-addr.arpa/IN: loaded serial 1
oct 22 19:27:24 dns named[13902]: zone 0.in-addr.arpa/IN: loaded serial 1
oct 22 19:27:24 dns named[13902]: zone localhost/IN: loaded serial 2
oct 22 19:27:24 dns named[13902]: zone 1.168.192.in-addr.arpa/IN: loaded serial 2
oct 22 19:27:24 dns named[13902]: zone tas.io/IN: loaded serial 2
oct 22 19:27:24 dns named[13902]: zone 127.in-addr.arpa/IN: loaded serial 1
oct 22 19:27:24 dns named[13902]: all zones loaded
t 22 19:27:24 dns named[13902]: running
t 22 19:27:24 dns named[13902]: managed-keys-zone: Key 20326 for zone . is now trusted (acceptance timer complete)
t 22 19:27:24 dns named[13902]: resolver priming query complete: success
root@dns:/home/dns#
```

```
network:
  ethernets:
    ens18:
      dhcp4: false
      addresses:
        - 192.168.1.2/24
      nameservers:
        addresses:
          - 192.168.1.2
          - 1.1.1.1
        search: [tas.io]
      routes:
        - to: default
          via: 192.168.1.1
  version: 2
```

```
"/etc/netplan/00-installer-config.yaml" 15L, 270B
```

Por último para ver si está bien configurado y funcionando como servidor dns.

Aquí es donde usamos las herramientas del paquete dnstools, nslookup en este caso.

```
root@dns:/home/dns# nslookup
> ifp.es
Server:          192.168.1.2
Address:         192.168.1.2#53

Non-authoritative answer:
Name:   ifp.es
Address: 104.18.15.196
Name:   ifp.es
Address: 104.18.14.196
> tas.io
Server:          192.168.1.2
Address:         192.168.1.2#53

Name:   tas.io
Address: 192.168.1.2
> google.com
Server:          192.168.1.2
Address:         192.168.1.2#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.184.174
Name:   google.com
Address: 2a00:1450:4006:812::200e
> _
```

Vemos que efectivamente estamos resolviendo con la dirección de nuestro dns. nos salimos

Lo único que faltaría es probar es probar el servidor dns desde otra maquina como si fuera otro cliente.

```
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: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:42:0f:a9 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    inet 192.168.1.11/24 metric 100 brd 192.168.1.255 scope global dynamic ens18
        valid_lft 589sec preferred_lft 589sec
    inet6 fe80::be24:11ff:fe42:fa9/64 scope link
        valid_lft forever preferred_lft forever
cliente@cliente:~$ _
```

Vemos que nuestro cliente está dentro de la misma red y puesto que resuelva con la 192.168.1.2 que es nuestro dns.

Comprobamos haciendo un nslookup

```
cliente@cliente:~$ nslookup
> google.com
Server:      192.168.1.2
Address:     192.168.1.2#53

Non-authoritative answer:
Name:   google.com
Address: 142.250.200.78
Name:   google.com
Address: 2a00:1450:4003:803::200e
> ifp.es
Server:      192.168.1.2
Address:     192.168.1.2#53

Non-authoritative answer:
Name:   ifp.es
Address: 104.18.14.196
Name:   ifp.es
Address: 104.18.15.196
> tas.io
Server:      192.168.1.2
Address:     192.168.1.2#53

Name:   tas.io
Address: 192.168.1.2
```

Perfecto, tenemos SERVIDOR DNS FUNCIONANDO !!!

```
      .--.
      |o_o |
      |:_/ |
      //  \ \
      (|    |)
      /\_ _/\
      \__)=(__
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BMB-----B B
BBB-----BBB
BBB-----BBB
BBB-----BBB
BBB-----BBB
BBB-----BBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBB++++BBBBB.BBBBBBB
BBBBB++BBBBB.++++BBBBB
BBBBB++BBBBB.++++BBBBB
BBBBB++BBBBB.++++BBBBB
BBBBB++++BBBBB.BBBBBBB
```