

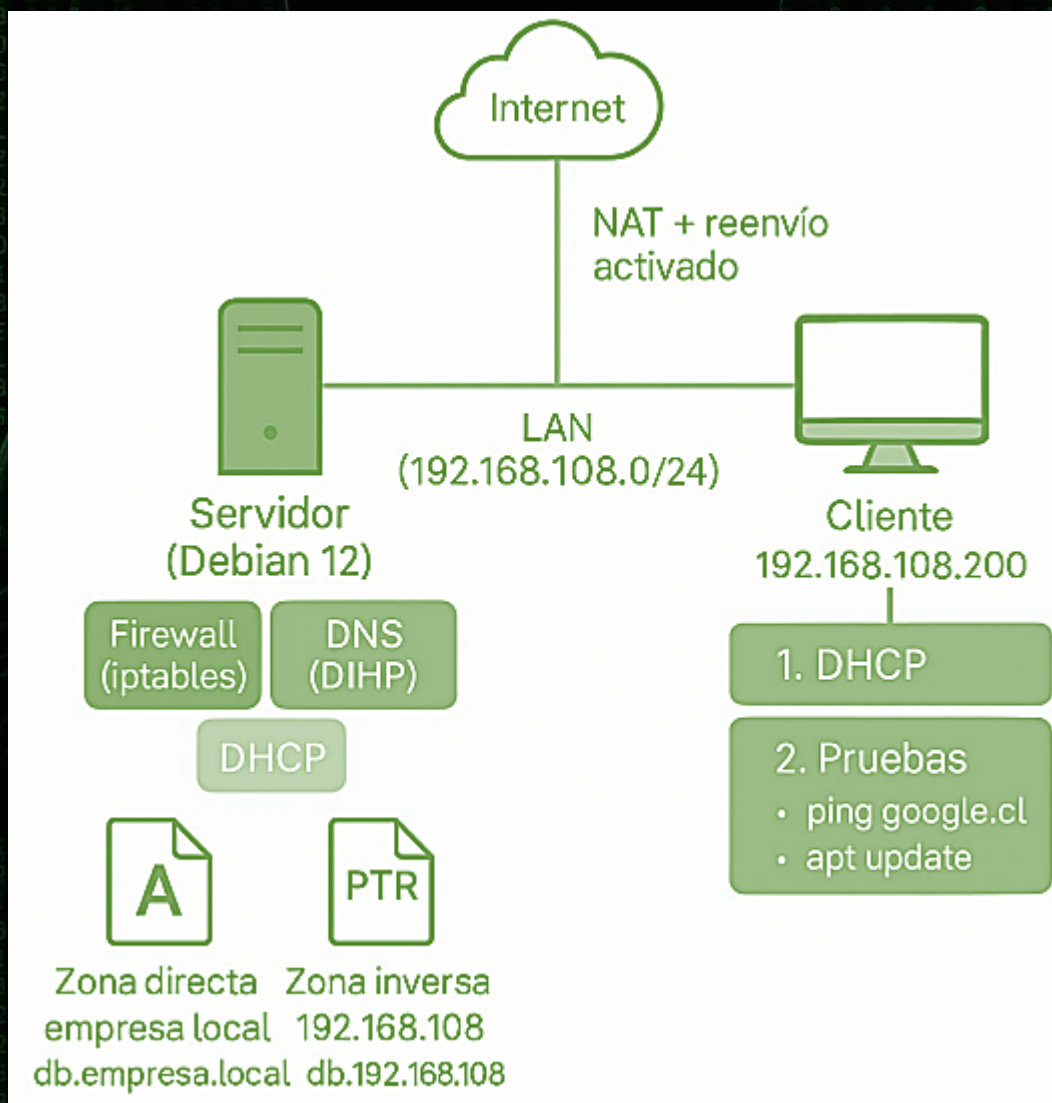
# Configuración de Servidor DNS y DHCP en Debian 12

Ruben Apablaza Muñoz  
—OZone—



## INTRODUCCIÓN

El presente informe tiene por objetivo mostrar el paso a paso que seguí para la configuración de una máquina que haga de servidor DHCP, DNS, y Firewall y otra que actúe como cliente. El documento abarca desde la preparación del laboratorio como lo muestra la imagen hasta la solución de problemas que se presentaron durante la configuración.





## PREPARACION DEL LABORATORIO



## CONSIDERACIÓN IMPORTANTE:

Las maquinas Debian 12 con las que se trabajará en este laboratorio, son maquinas “limpias”; esto quiere decir que se instalaron solo como CLI, ni siquiera tienen sudo instalado. Se instalo una y después se clonó para tener 2 máquinas iguales inicialmente.

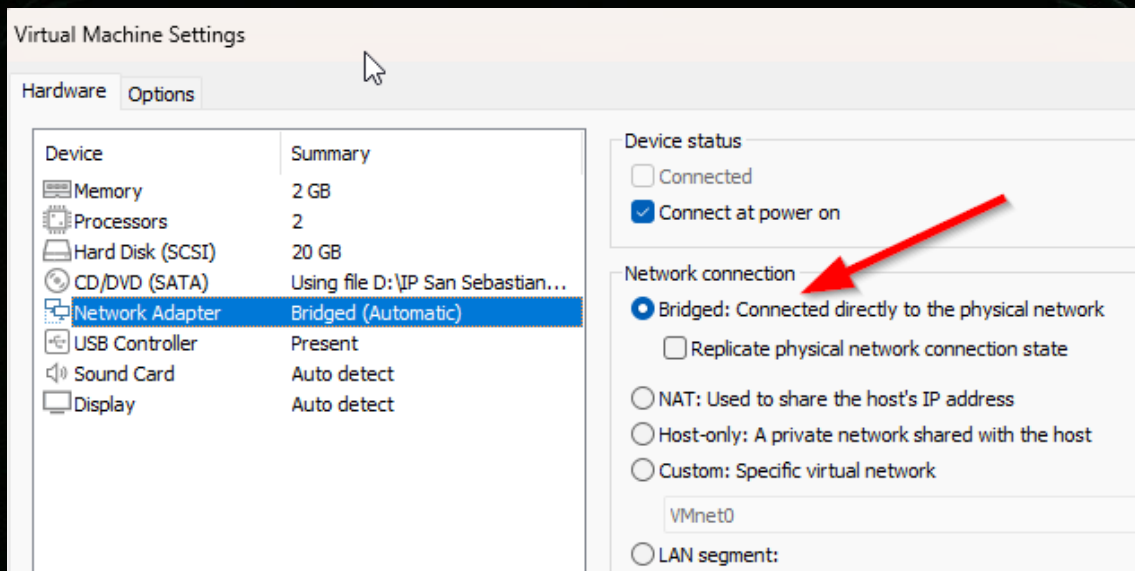
Una de las maquinas actuará como “Servidor” y proporcionará los servicios de Router, DNS y DHCP para la otra máquina que será el equipo “Cliente”.

Antes de comenzar el laboratorio recomiendo actualizar las máquinas; para esto desde vmware, dejaremos ambas maquinas con una conexión bridge para que tengan salida a internet.

### Paso 0: Preparación inicial de las máquinas virtuales

#### 1. Configurar máquinas virtuales en modo Bridge:

- Para la máquina "Servidor" y la máquina "Cliente", configuramos por ahora solo 1 interfaz de red en modo Bridge. Esto les permitirá obtener una dirección IP de nuestra red local y acceder a Internet.



Cambiar nombre y actualizar los sistemas:

- En ambas máquinas; abrimos una terminal con acceso root.
- Ejecutamos el comando

`hostnamectl set-hostname Servidor`  *#(para el otro equipo será cliente)*

`apt update && apt upgrade` para actualizar la lista de paquetes disponibles.

```
ruben1@maquina1:~$ su - root
Contraseña:
root@maquina1:~# hostnamectl set-hostname Servidor
root@maquina1:~# su - ruben1
ruben1@Servidor:~$ su - root
Contraseña:
root@Servidor:~# apt update && apt upgrade
```



Para iniciar el laboratorio vamos a cambiar la conexión de las máquinas y quedaran de la siguiente forma

Servidor: Bridge + Host only

Cliente: Host only

La configuración de vmware Virtual Network Editor quedará así:

The screenshot shows the VMware Virtual Network Editor window. At the top, there is a table listing the virtual networks. Below the table are buttons for 'Add Network...', 'Remove Network', and 'Rename Network...'. The 'VMnet Information' section is expanded, showing options for 'Bridged', 'NAT', and 'Host-only'. The 'Host-only' option is selected. Below this, there are checkboxes for 'Connect a host virtual adapter to this network' and 'Use local DHCP service to distribute IP address to VMs', both of which are checked. At the bottom, there are input fields for 'Subnet IP' and 'Subnet mask'. The 'Subnet IP' field is highlighted with a red box and contains the value '192 . 168 . 108 . 0'. The 'Subnet mask' field contains the value '255 . 255 . 255 . 0'. At the very bottom, there are buttons for 'Restore Defaults', 'Import...', 'Export...', 'OK', 'Cancel', 'Apply', and 'Help'.

Name	Type	External Connection	Host Connection	DHCP	Subnet Address
VMnet0	Bridged	Auto-bridging	-	-	-
VMnet1	Host-only	-	Connected	Enabled	192.168.108.0
VMnet8	NAT	NAT	Connected	Enabled	192.168.77.0

Buttons: Add Network..., Remove Network, Rename Network...

VMnet Information

☐ Bridged (connect VMs directly to the external network)

Bridged to: Automatic Automatic Settings...

☐ NAT (shared host's IP address with VMs) NAT Settings...

☒ Host-only (connect VMs internally in a private network)

☒ Connect a host virtual adapter to this network

Host virtual adapter name: VMware Network Adapter VMnet1

☒ Use local DHCP service to distribute IP address to VMs DHCP Settings...

Subnet IP: 192 . 168 . 108 . 0 Subnet mask: 255 . 255 . 255 . 0

Buttons: Restore Defaults, Import..., Export..., OK, Cancel, Apply, Help

Problema: “la interfaz de host-only no tiene IP”

Puede ser que te ocurra lo siguiente:

que al hacer

`ip -a`

la interfaz correspondiente al host no tenga una dirección ip asignada,

```
ruben1@Servidor:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1a:3d:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.36/24 brd 192.168.0.255 scope global dynamic ens33
        valid_lft 604665sec preferred_lft 604665sec
    inet6 2800:150:119:815:214:5dff:fe00:4243/64 scope global dynamic mngtmpaddr
        valid_lft 3600sec preferred_lft 3600sec
    inet6 fe80::20c:291a:3d00:4243/64 scope link
        valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:0c:29:99:46:80 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
ruben1@Servidor:~$ _
```

Esto ocurre porque Debian no está solicitando la IP de forma automática.

Si ese es el caso, ingresa en modo root el siguiente comando:

`dhclient -v`

(si tienes sudo instalado lo puedes hacer con `sudo dhclient -v`)

Y si vuelves a llamar `ip -a`, comprobarás que ya tienes ip en esa interfaz.

```
Contraseña.  
root@Servidor:~# dhclient -v  
Internet Systems Consortium DHCP Client 4.4.3-P1  
Copyright 2004-2022 Internet Systems Consortium.  
All rights reserved.  
For info, please visit https://www.isc.org/software/dhcp/  
  
Listening on LPF/ens37/00:0c:29:99:46:80  
Sending on LPF/ens37/00:0c:29:99:46:80  
Listening on LPF/ens33/00:0c:29:99:46:76  
Sending on LPF/ens33/00:0c:29:99:46:76  
Sending on Socket/fallback  
DHCPREQUEST for 192.168.108.128 on ens37 to 255.255.255.255 port 67  
DHCPREQUEST for 192.168.0.36 on ens33 to 255.255.255.255 port 67  
DHCPCACK of 192.168.108.128 from 192.168.108.254  
bound to 192.168.108.128 -- renewal in 789 seconds.  
root@Servidor:~# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:99:46:76 brd ff:ff:ff:ff:ff:ff  
    altname enp2s1  
    inet 192.168.0.36/24 brd 192.168.0.255 scope global dynamic ens33  
        valid_lft 604260sec preferred_lft 604260sec  
    inet6 2800:150:119:815:20c:29:99:46:76/64 scope global dynamic mngtmpaddr  
        valid_lft 3600sec preferred_lft 3600sec  
    inet6 fe80::20c:29:99:46:76/64 scope link  
        valid_lft forever preferred_lft forever  
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000  
    link/ether 00:0c:29:99:46:80 brd ff:ff:ff:ff:ff:ff  
    altname enp2s5  
    inet 192.168.108.128/24 brd 192.168.108.255 scope global dynamic ens37  
        valid_lft 1791sec preferred_lft 1791sec  
    inet6 fe80::20c:29:99:46:80/64 scope link  
        valid_lft forever preferred_lft forever  
root@Servidor:~# _
```



Para que inicie de forma automática haremos lo siguiente como root:

```
nano /etc/network/interfaces
```

y agregaremos o ajustaremos las líneas, cambiando ens37 por la interfaz que corresponda a tu configuración de interfaces:

```
auto ens37
```

```
iface ens37 inet dhcp
```

GNU nano 7.2

/E

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
```

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

```
# The loopback network interface
```

```
auto lo
```

```
iface lo inet loopback
```

```
# The primary network interface
```

```
allow-hotplug ens33
```

```
iface ens33 inet dhcp
```

```
# This is an autoconfigured IPv6 interface
```

```
iface ens33 inet6 auto
```

```
auto ens37
```

```
iface ens37 inet dhcp
```

Guardamos los cambios y reiniciamos el servicio networking con

```
sudo systemctl restart networking
```

y ahora cuando volvamos a iniciar el sistema automáticamente nos asignara una ip a nuestra interfaz.

Otra voz...

A la maquina "Servidor" le asignaremos IP fija (que es como debe ser)

Abriremos como root

```
nano /etc/network/interfaces
```

y agregaremos o ajustaremos las líneas, cambiando ens37 por la interfaz que corresponda:

```
auto ens37
iface ens37 inet static
address 192.168.108.100
netmask 255.255.255.0
```

```
GNU nano 7.2
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet dhcp
# This is an autoconfigured IPv6 interface
iface ens33 inet6 auto

auto ens37
iface ens37 inet static
address 192.168.108.100
netmask 255.255.255.0
```

Guardamos los cambios y reiniciamos el servicio *networking* con

```
sudo systemctl restart networking
```



y ahora cuando volvamos a iniciar el sistema tendremos asignada la ip 192.168.108.100 a nuestra interfaz.

```
root@Servidor:~# 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1a:3d:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.0.36/24 brd 192.168.0.255 scope global dynamic ens33
        valid_lft 604793sec preferred_lft 604793sec
    inet6 2800:150:119::/64 scope global dynamic mngtmpaddr
        valid_lft 3599sec preferred_lft 3599sec
    inet6 fe80::20c:291a:3d:00 scope link
        valid_lft forever preferred_lft forever
3: ens37: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:1a:3d:00 brd ff:ff:ff:ff:ff:ff
    altname enp2s5
    inet 192.168.108.100/24 brd 192.168.108.255 scope global ens37
        valid_lft forever preferred_lft forever
    inet6 fe80::20c:291a:3d:00 scope link
        valid_lft forever preferred_lft forever
root@Servidor:~# nano /etc/network/interfaces
```



SERVIDOR



## INSTALACION DE PAQUETES NECESARIOS EN EL SERVIDOR

Ahora instalaremos o actualizaremos los siguientes paquetes en el servidor

### isc-dhcp-server

1. Instalar: Ejecutamos `apt install isc-dhcp-server`
2. Verificar: Revisa el estado del servicio con `systemctl status isc-dhcp-server` . Como vimos antes, es probable que falle al intentar iniciarse sin una configuración, pero el paquete estará instalado.

```
ago 08 11:14:59 Servidor isc-dhcp-server[1458]: Starting ISC DHCPv4 server: dhcpdcheck syslog for diagnostics. ... failed!
ago 08 11:14:59 Servidor isc-dhcp-server[1458]: failed!
ago 08 11:14:59 Servidor systemd[1]: isc-dhcp-server.service: Control process exited, code=exited, status=1/FAILURE
ago 08 11:14:59 Servidor systemd[1]: isc-dhcp-server.service: Failed with result 'exit-code'.
ago 08 11:14:59 Servidor systemd[1]: Failed to start isc-dhcp-server.service - LSB: DHCP server.
Procesando disparadores para man-db (2.11.2-2) ...
root@Servidor:~#
```

### bind9

1. Instalar: Ejecuta `apt install bind9`
2. Verificar: Revisa el estado del servicio con `systemctl status bind9` . Debería estar activo y funcionando.

```
root@Servidor:~# systemctl status bind9
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2025-08-08 11:18:17 -04; 18s ago
     Docs: man:named(8)
    Main PID: 1692 (named)
      Status: "running"
        Tasks: 6 (limit: 2255)
       Memory: 30.3M
          CPU: 26ms
    CGroup: /system.slice/named.service
            └─1692 /usr/sbin/named -f -u bind
```

## iptables

1. Instalar: Ejecuta `apt install iptables`
2. Verificar: Puedes verificar su instalación y la versión con

`iptables --version`

```
root@Servidor:~# iptables --version
iptables v1.8.9 (nf_tables)
root@Servidor:~#
```

## iproute2

1. Instalar: Ejecuta `apt install iproute2`
2. Verificar: Puedes verificar que se instaló correctamente ejecutando uno de sus comandos, como `ip addr` o `ip route`

```
root@Servidor:~# ip route
default via 192.168.0.1 dev ens33
192.168.0.0/24 dev ens33 proto kernel scope link src 192.168.0.36
192.168.108.0/24 dev ens37 proto kernel scope link src 192.168.108.100
root@Servidor:~#
```

El objetivo es que la maquina “Servidor” actúe como Router, DNS y Servidor DHCP para el equipo “Cliente”



Activar el reenvío de paquetes IP

Habilitaremos el reenvío de paquetes para que el servidor pueda actuar como router. Esto se hace agregando:

"net.ipv4.ip\_forward=1" al archivo /etc/sysctl.d/99-ipforward.conf y luego aplicando los cambios con sysctl -p.

```
echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-  
ipforward.conf
```

```
sysctl -p
```

```
root@Servidor:~# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/99-ipforward.conf  
root@Servidor:~# sysctl -p  
root@Servidor:~#
```

## Configuracion de DHCP

Con nano /etc/dhcp/dhcpd.conf vamos a editar el archivo y dejaremos la siguiente información en su interior.

```
default-lease-time 600;
max-lease-time 7200;
authoritative;
ddns-update-style interim;
update-static-leases on;
zone empresa.local. {
    primary 127.0.0.1;
}
zone 108.168.192.in-addr.arpa. {
    primary 127.0.0.1;
}
subnet 192.168.108.0 netmask 255.255.255.0 {
    range 192.168.108.150 192.168.108.200;
    option routers 192.168.108.100;
    option subnet-mask 255.255.255.0;
    option domain-name "empresa.local";
    option domain-name-servers 192.168.108.100;
}
```



GNU nano 7.2

/etc/dhcp/dhcpd.conf \*

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

ddns-update-style interim;
update-static-leases on;

zone empresa.local. {
    primary 127.0.0.1;
}

zone 108.168.192.in-addr.arpa. {
    primary 127.0.0.1;
}

subnet 192.168.108.0 netmask 255.255.255.0 {
    range 192.168.108.150 192.168.108.200;
    option routers 192.168.108.100;
    option subnet-mask 255.255.255.0;
    option domain-name "empresa.local";
    option domain-name-servers 192.168.108.100;
}
```

## CONFIGURACION DEL DNS

### 1. Configuración de zonas en BIND

```
nano /etc/bind/named.conf.local
```

y pegamos lo siguiente en su interior

```
zone "empresa.local" {  
    type master;  
    file "/etc/bind/db.empresa.local";  
    allow-update { 127.0.0.1; };  
};  
  
zone "108.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192.168.108";  
    allow-update { 127.0.0.1; };  
};
```



GNU nano 7.2

/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 "empresa.local" {  
    type master;  
    file "/etc/bind/db.empresa.local";  
    allow-update { 127.0.0.1; };  
};  
zone "108.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192.168.108";  
    allow-update { 127.0.0.1; };  
};  
|
```

Guardamos y vamos al siguiente

## Configuración de opciones globales en BIND

Incluiremos las direcciones IP en que BIND escuchará, quien puede hacer consultas y resolución recursiva además de los servidores externos para reenviar consultas como (8.8.8.8 y 1.1.1.1).

```
nano /etc/bind/named.conf.options
```

y en su interior lo dejamos asi:

```
options {  
    directory "/var/cache/bind";  
    listen-on { 127.0.0.1; 192.168.108.100; };  
    allow-query { 127.0.0.1; 192.168.108.0/24; };  
    allow-recursion { 127.0.0.1; 192.168.108.0/24; };  
    recursion yes;  
    dnssec-validation no;  
    forwarders {  
        8.8.8.8;  
        1.1.1.1;  
    };  
    listen-on-v6 { none; };  
};
```



```
GNU nano 7.2 /etc/bind/named.conf.options
options {
  directory "/var/cache/bind";
  listen-on { 127.0.0.1; 192.168.108.100; };
  allow-query { 127.0.0.1; 192.168.108.0/24; };
  allow-recursion { 127.0.0.1; 192.168.108.0/24; };
  recursion yes;
  dnssec-validation no;
  forwarders {
    8.8.8.8;
    1.1.1.1;
  };
  listen-on-v6 { none; };
};
```

Guardamos y vamos al siguiente.

Creación de archivo de zona directa

nano /etc/bind/db.empresa.local

y en su interior pegamos lo siguiente

\$TTL 604800

@ IN SOA ns.empresa.local. admin.empresa.local. (

2025080601 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800 ) ; Negative Cache TTL

;

@ IN NS ns.empresa.local.

ns IN A 192.168.108.100

```
GNU nano 7.2 /etc/bind/db.empresa.local *
$TTL      604800
@         IN      SOA      ns.empresa.local. admin.empresa.local. (
                                2025080601 ; Serial
                                604800 ; Refresh
                                86400 ; Retry
                                2419200 ; Expire
                                604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.empresa.local.
ns        IN      A        192.168.108.100
```

Guardamos y vamos al siguiente.



Creación de archivo de zona directa

```
nano /etc/bind/db.192.168.108
```

y dentro pegamos lo siguiente

```
$TTL 604800
```

```
@ IN SOA ns.empresa.local. admin.empresa.local. (
```

```
2025080601 ; Serial
```

```
604800 ; Refresh
```

```
86400 ; Retry
```

```
2419200 ; Expire
```

```
604800 ) ; Negative Cache TTL
```

```
;
```

```
@ IN NS ns.empresa.local.
```

```
100 IN PTR ns.empresa.local.
```

```
GNU nano 7.2 /etc/bind/db.192.168.108 *
$TTL      604800
@          IN      SOA      ns.empresa.local. admin.empresa.local. (
                                2025080601 ; Serial
                                604800 ; Refresh
                                86400 ; Retry
                                2419200 ; Expire
                                604800 ) ; Negative Cache TTL
;
@          IN      NS       ns.empresa.local.
100        IN      PTR      ns.empresa.local.
```

Guardamos y cerramos.

## CONFIGURACION DE NAT Y FIREWALL (IPTABLES)

En mi caso el directorio /etc/iptables no existe por lo que primero debo crearlo como root con el siguiente comando

```
mkdir -p /etc/iptables
```

Creamos o modificamos el archivo

```
nano /etc/iptables/rules.v4
```

y pegamos lo siguiente

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A FORWARD -i ens37 -o ens33 -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j
ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j
ACCEPT
-A INPUT -s 192.168.108.0/24 -i ens37 -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A INPUT -p udp --dport 67:68 -j ACCEPT
-A INPUT -p udp --sport 67:68 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
```



```
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o ens33 -j MASQUERADE
COMMIT
```

```
root@Servidor:~# cat /etc/iptables/rules.v4
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A FORWARD -i ens37 -o ens33 -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 192.168.108.0/24 -i ens37 -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A INPUT -p udp --dport 67:68 -j ACCEPT
-A INPUT -p udp --sport 67:68 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o ens33 -j MASQUERADE
COMMIT
root@Servidor:~#
```

Guardamos y listo.

OJO. En este punto tenía otra configuración con otro orden para el archivo de ip tables que no me dejaba conectar desde el cliente a internet. Cuando apliqué estas reglas en este orden pude hacer ping a 8.8.8.8 desde la maquina "Cliente". Lo que quiero decir con esto es que importa el orden en que se aplican las reglas de iptables, una regla "mas arriba" puede simplemente anular a una de mas abajo.

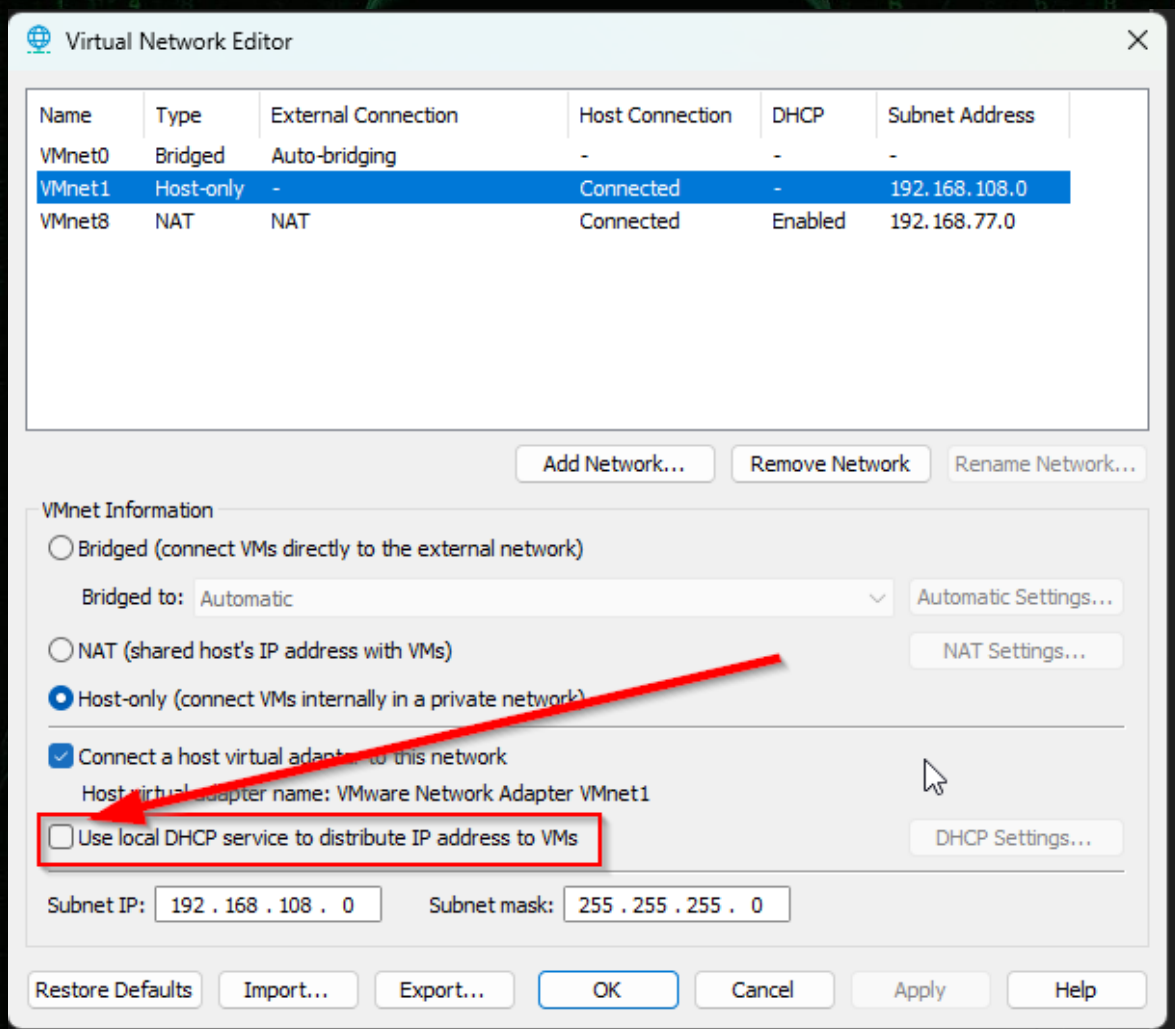


CLIENTE



## CONFIGURAR Y PROBAR

Un punto importante antes de probar la conexión del cliente hacia nuestro “Servidor”, es apagar el servicio de DHCP que esta activado en la configuración de VMWare en Host-only, con el objetivo de que nuestro cliente reciba la dirección ip desde nuestro servidor y no del servicio de VMWare.



Si es necesario, reiniciamos VMWare, y cuando se vuelva, encender primero la maquina “Servidor” y después que ya este encendida, encendemos la maquina “Cliente”.

Con ip a comprobamos que ya tenemos dirección ip

```
ruben1@Cliente:~$ 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: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:14:95:db brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.108.150/24 brd 192.168.108.255 scope global dynamic ens33
        valid_lft 404sec preferred_lft 404sec
    inet6 fe80::20c:2914:95db:1/64 scope link
        valid_lft forever preferred_lft forever
ruben1@Cliente:~$
```

Con cat /etc/resolv.conf vamos a confirmar que nuestro servidor DNS sea 192.168.108.100 (nuestro "Servidor").

```
ruben1@Cliente:~$ cat /etc/resolv.conf
domain empresa.local
search empresa.local
nameserver 192.168.108.100
ruben1@Cliente:~$
```

Ahora validaremos la resolución de DNS y la conectividad, para esto vamos a realizar una prueba de resolución de nombres con dig google.com o ping google.cl.



```

ruben1@Cliente:~$ ping google.cl
PING google.cl (142.251.0.94) 56(84) bytes of data.
64 bytes from cj-in-f94.1e100.net (142.251.0.94): icmp_seq=1 ttl=106 time=11.6 ms
64 bytes from cj-in-f94.1e100.net (142.251.0.94): icmp_seq=2 ttl=106 time=12.3 ms
64 bytes from cj-in-f94.1e100.net (142.251.0.94): icmp_seq=3 ttl=106 time=10.1 ms
^C
--- google.cl ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 10.148/11.337/12.255/0.881 ms
ruben1@Cliente:~$ dig google.com

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39738
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: a3c33dc7db49a35701000000689663d2494cd1fe841adb41 (good)
;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                262     IN      A      172.217.192.139
google.com.                262     IN      A      172.217.192.101
google.com.                262     IN      A      172.217.192.113
google.com.                262     IN      A      172.217.192.100
google.com.                262     IN      A      172.217.192.102
google.com.                262     IN      A      172.217.192.138

;; Query time: 15 msec
;; SERVER: 192.168.108.100#53(192.168.108.100) (UDP)
;; WHEN: Fri Aug 08 16:53:38 -04 2025
;; MSG SIZE rcvd: 163

```

Ahora verificamos la ruta por defecto usando `ip route` y nos aseguramos de que el gateway sea 192.168.108.100

```

ruben1@Cliente:~$ ip route
default via 192.168.108.100 dev ens33
192.168.108.0/24 dev ens33 proto kernel scope link src 192.168.108.150
ruben1@Cliente:~$

```

Por último hacemos en el equipo cliente un apt update y vemos que todo quedo perfecto

```
ruben1@Cliente:~$ su - root
Contraseña:
root@Cliente:~# apt update
Des:1 http://security.debian.org/debian-security bookworm-security InRelease [48,0 kB]
Obj:2 http://deb.debian.org/debian bookworm InRelease
Des:3 http://deb.debian.org/debian bookworm-updates InRelease [55,4 kB]
Des:4 http://deb.debian.org/debian bookworm-updates/main Sources.diff/Index [20,7 kB]
Des:5 http://deb.debian.org/debian bookworm-updates/main amd64 Packages.diff/Index [20,7 kB]
Des:6 http://deb.debian.org/debian bookworm-updates/main Sources T-2025-08-08-1404.18-F-2025-08-08-1404.18.pdiff [187 B]
Des:6 http://deb.debian.org/debian bookworm-updates/main Sources T-2025-08-08-1404.18-F-2025-08-08-1404.18.pdiff [187 B]
Des:7 http://deb.debian.org/debian bookworm-updates/main amd64 Packages T-2025-08-08-1404.18-F-2025-08-08-1404.18.pdiff [403 B]
Des:7 http://deb.debian.org/debian bookworm-updates/main amd64 Packages T-2025-08-08-1404.18-F-2025-08-08-1404.18.pdiff [403 B]
Descargados 145 kB en 0s (380 kB/s)
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 2 paquetes. Ejecute «apt list --upgradable» para verlos.
root@Cliente:~#
```





SERVIDOR

## INSTALACIÓN DE APACHE

```
apt install apache2
```

Una vez instalado, el servicio se iniciará automáticamente.

Puedes verificar su estado con el comando:

```
systemctl status apache2
```

```
root@Servidor:~#  
root@Servidor:~# systemctl status apache2  
● apache2.service - The Apache HTTP Server  
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enabled)  
   Active: active (running) since Fri 2025-08-08 18:24:31 -04; 1min 38s ago  
     Docs: https://httpd.apache.org/docs/2.4/  
   Main PID: 1569 (apache2)  
     Tasks: 55 (limit: 2255)  
    Memory: 13.0M  
       CPU: 44ms  
    CGroup: /system.slice/apache2.service  
            └─1569 /usr/sbin/apache2 -k start  
              └─1571 /usr/sbin/apache2 -k start  
                └─1572 /usr/sbin/apache2 -k start  
  
ago 08 18:24:31 Servidor systemd[1]: Starting apache2.service - The Apache HTTP Server...  
ago 08 18:24:31 Servidor apachectl[1568]: AH00557: apache2: apr_sockaddr_info_get() failed for Servidor  
ago 08 18:24:31 Servidor apachectl[1568]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name,  
ago 08 18:24:31 Servidor systemd[1]: Started apache2.service - The Apache HTTP Server.  
lines 1-17/17 (END)
```

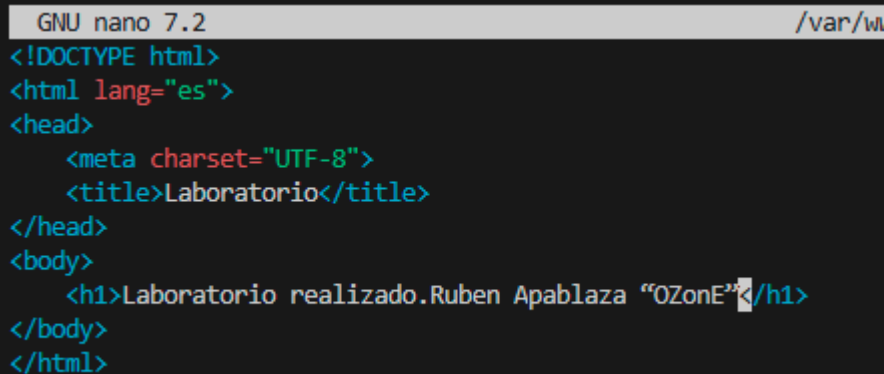


## Configurar la página de inicio

1. La página de inicio predeterminada de Apache se encuentra en el archivo `/var/www/html/index.html`. Debes editar este archivo.
2. Usamos un editor de texto como nano o vi para abrir y modificar el archivo:  
`vi /var/www/html/index.html`
3. Borramos todo el contenido predeterminado y lo reemplazamos con el siguiente código HTML simple:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Laboratorio</title>
</head>
<body>
  <h1>Laboratorio realizado Ruben Apablaza "OZonE"</h1>
</body>
</html>
```

## 4. Guardamos los cambios

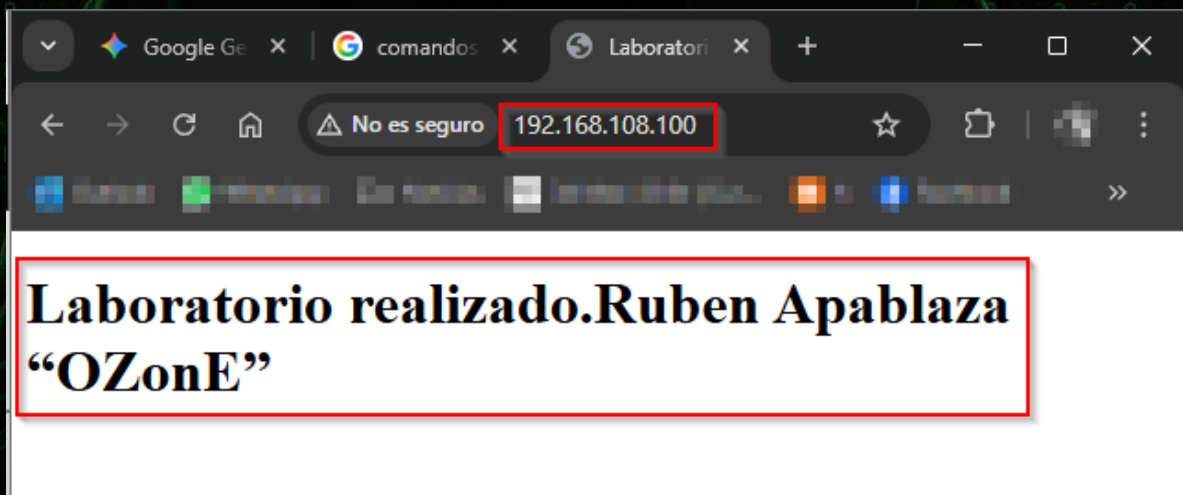
A screenshot of the GNU nano 7.2 text editor interface. The title bar shows 'GNU nano 7.2' on the left and '/var/www/html/index.html' on the right. The editor area contains the following HTML code:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Laboratorio</title>
</head>
<body>
  <h1>Laboratorio realizado.Ruben Apablaza "OZonE"</h1>
</body>
</html>
```

## VERIFICAR EL FUNCIONAMIENTO

Ahora que Apache está instalado y la página de inicio está configurada, podemos verificar que todo funcione.

1. Desde el servidor: Abre un navegador de texto como links o lynx y vamos a la dirección <http://localhost>.
2. Desde el cliente: Abrimos un navegador web y visitamos la dirección <http://192.168.108.100>. Se debería ver la página que creamos con el texto, en mi caso “Laboratorio realizado Ruben Apablaza “OZonE”





ESTO ESTA EXPLICADO MAS ADELANTE EN LA PARTE DONDE DICE PROBLEMAS!!!, SI SE HACEN ESOS PASOS EN ESTE PUNTO NO HABRA TAL PROBLEMA.

```
GNU nano 7.2 /etc/bind/db.empresa.local *
$TTL      604800
@         IN      SOA      ns.empresa.local. admin.empresa.local. (
                                2025080601 ; Serial
                                604800 ; Refresh
                                86400 ; Retry
                                2419200 ; Expire
                                604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.empresa.local.
ns        IN      A        192.168.108.100
www       IN      A        192.168.108.100
```

```
ruben1@Cliente:~$ dig www.empresa.local

; <<>> DiG 9.18.33-1~deb12u2-Debian <<>> www.empresa.local
;; global options: +cmd
;; Got answer:
;; WARNING: .local is reserved for Multicast DNS
;; You are currently testing what happens when an mDNS query is leaked to DNS
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61812
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:: udp: 1232
; COOKIE: b7c2ef1da03bbece0100000068967f0424290268a700bb5d (good)
;; QUESTION SECTION:
;www.empresa.local.      IN      A

;; ANSWER SECTION:
www.empresa.local.      604800 IN      A      192.168.108.100

;; Query time: 0 msec
;; SERVER: 192.168.108.100#53(192.168.108.100) (UDP)
;; WHEN: Fri Aug 08 18:49:41 -04 2025
;; MSG SIZE rcvd: 90

ruben1@Cliente:~$
```

## INSTALACIÓN DE CURL

`apt update`

Al intentar ejecutar el comando se queda colgado

```
root@cliente:~# apt update
0% [Conectando a debian.map.fastlydns.net (2a04:4e42:600::644)] [Conectando a debian.map.fastlydns.net (2a04:4e42:600::644)]
Ign:1 http://security.debian.org/debian-security bookworm-security InRelease
Ign:2 http://deb.debian.org/debian bookworm InRelease
Ign:3 http://deb.debian.org/debian bookworm-updates InRelease
```

El cliente tiene una dirección IPv6 de forma predeterminada, y apt a menudo intenta usar IPv6 primero. Como nuestro servidor no está enrutando el tráfico IPv6, la conexión se queda colgada.

Actualizar las reglas de iptables

Debemos agregar una regla para permitir el tráfico de los puertos 80 (HTTP) y 443 (HTTPS) en la cadena FORWARD de iptables de nuestro servidor. Esto asegurará que las solicitudes de apt y cualquier otro tráfico web del cliente puedan pasar a Internet.

Abrimos el archivo de reglas de iptables en el servidor:

```
nano /etc/iptables/rules.v4
```

En la sección `*filter`, agregamos las siguientes líneas antes de la línea `COMMIT`:

```
-A FORWARD -i ens37 -o ens33 -p tcp --dport 80 -j ACCEPT
```

```
-A FORWARD -i ens37 -o ens33 -p tcp --dport 443 -j ACCEPT
```



GNU nano 7.2

/etc/iptables/rules.v4

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A FORWARD -i ens37 -o ens33 -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 192.168.108.0/24 -i ens37 -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 80 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 443 -j ACCEPT
-A INPUT -p udp --dport 67:68 -j ACCEPT
-A INPUT -p udp --sport 67:68 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o ens33 -j MASQUERADE
COMMIT
```

\*filter

:INPUT DROP [0:0]

:FORWARD DROP [0:0]

:OUTPUT ACCEPT [0:0]

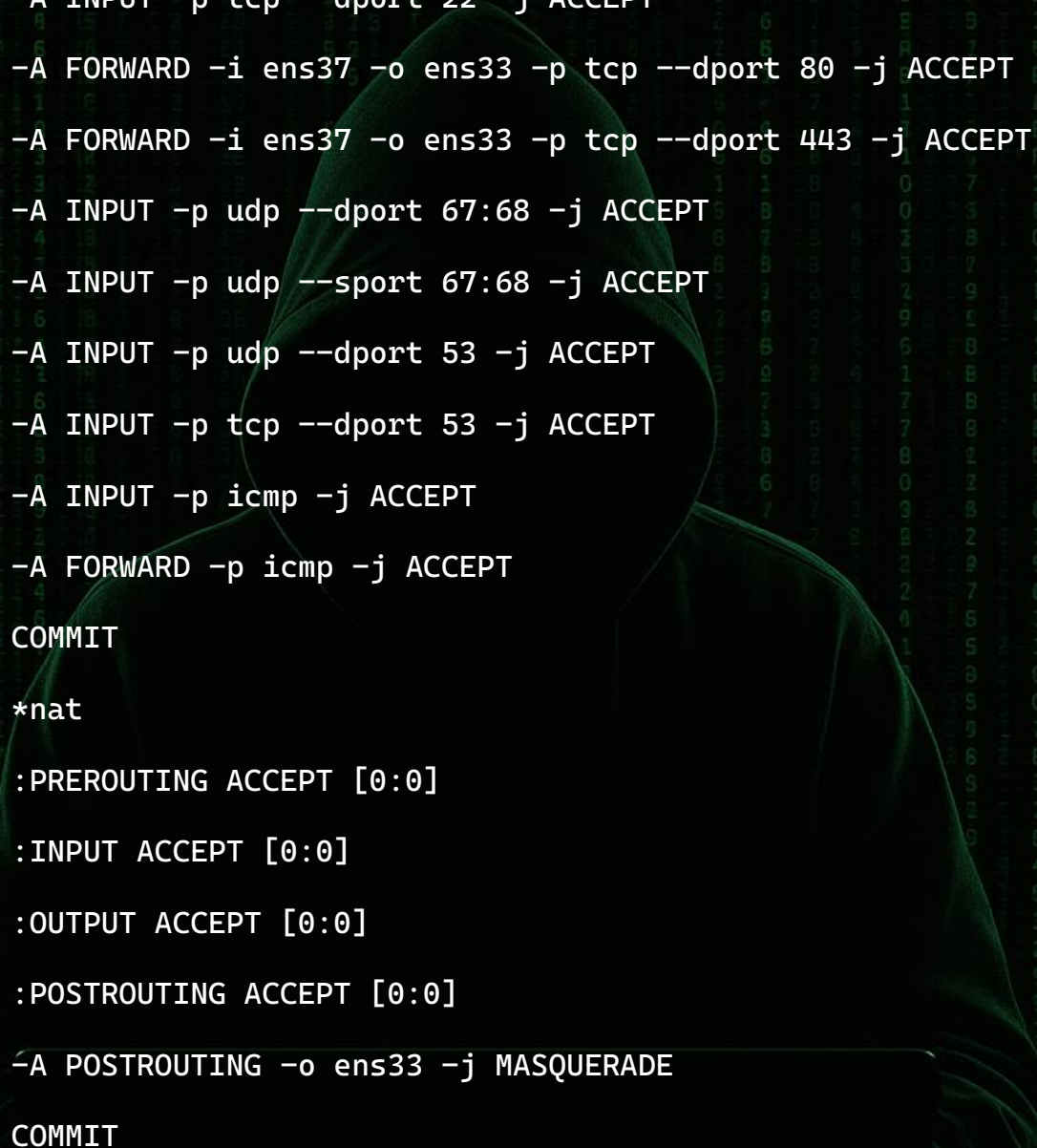
-A INPUT -i lo -j ACCEPT

-A FORWARD -i ens37 -o ens33 -j ACCEPT

-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

-A INPUT -s 192.168.108.0/24 -i ens37 -j ACCEPT



```
-A INPUT -p tcp --dport 22 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 80 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 443 -j ACCEPT
-A INPUT -p udp --dport 67:68 -j ACCEPT
-A INPUT -p udp --sport 67:68 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
COMMIT

*nat

:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]

-A POSTROUTING -o ens33 -j MASQUERADE
COMMIT
```

Estas reglas permiten que el tráfico de los puertos web (80 y 443) sea reenviado desde la interfaz interna (ens37) hacia la externa (ens33).

1. Guardamos los cambios.
2. Recargamos las reglas de iptables para que los cambios surtan efecto:

```
iptables-restore < /etc/iptables/rules.v4
```



La solución es forzar a apt a usar IPv4. Hay dos maneras de hacerlo:

Solución temporal que sirve para nuestro caso

Podemos ejecutar el comando apt update con un parámetro para deshabilitar IPv6 en esa sesión:

```
apt -o Acquire::ForceIPv4=true update
```

y ahora si funciona

```
root@Cliente:~# apt -o Acquire::ForceIPv4=true update
Obj:1 http://security.debian.org/debian-security bookworm-security InRelease
Obj:2 http://deb.debian.org/debian bookworm InRelease
Obj:3 http://deb.debian.org/debian bookworm-updates InRelease
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se pueden actualizar 2 paquetes. Ejecute «apt list --upgradable» para verlos.
root@Cliente:~#
```

Y ahora instalamos curl

```
apt install curl
```

```
root@Cliente:~# apt install curl
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libcurl4
Se instalarán los siguientes paquetes NUEVOS:
  curl libcurl4
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 707 kB de archivos.
Se utilizarán 1.361 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Des:1 http://deb.debian.org/debian bookworm/main amd64 libcurl4 amd64 7.88.1-10+deb12u12 [391 kB]
Des:2 http://deb.debian.org/debian bookworm/main amd64 curl amd64 7.88.1-10+deb12u12 [315 kB]
Descargados 707 kB en 0s (5.536 kB/s)
Seleccionando el paquete libcurl4:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 34587 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar .../libcurl4_7.88.1-10+deb12u12_amd64.deb ...
Desempaquetando libcurl4:amd64 (7.88.1-10+deb12u12) ...
Seleccionando el paquete curl previamente no seleccionado.
Preparando para desempaquetar .../curl_7.88.1-10+deb12u12_amd64.deb ...
Desempaquetando curl (7.88.1-10+deb12u12) ...
Configurando libcurl4:amd64 (7.88.1-10+deb12u12) ...
Configurando curl (7.88.1-10+deb12u12) ...
Procesando disparadores para man-db (2.11.2-2) ...
Procesando disparadores para libc-bin (2.36-9+deb12u10) ...
root@Cliente:~#
```

Y ahora podemos verificar con curl que la pagina es accesible desde el cliente

```
root@Cliente:~# curl http://www.empresa.local
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Laboratorio</title>
</head>
<body>
  <h1>Laboratorio realizado.Ruben Apablaza "OZonE"</h1>
</body>
</html>
root@Cliente:~#
```



## PROBLEMAS!!!

```
oot@Servidor:~# ping miweb.empresa.local
ping: miweb.empresa.local: Nombre o servicio desconocido
```

El error "*Nombre o servicio desconocido*" significa que tu servidor no puede resolver el nombre `miweb.empresa.local`. Hay dos razones principales para esto:

1. El registro DNS no existe: El servidor no tiene un registro para `miweb` en el archivo de zona DNS. Solo agregamos `www` a la configuración anterior.
2. El servidor no se está usando a sí mismo como DNS: El archivo `/etc/resolv.conf` del servidor no está apuntando a su propio servicio DNS.

## Solución

Seguir estos pasos para solucionar ambos problemas:

1. Agregar el registro DNS para `miweb`
  - Editamos el archivo de zona directa en el servidor:  

```
nano /etc/bind/db.empresa.local
```
  - Agregar la siguiente línea debajo del registro `www` para mapear `miweb.empresa.local` a la dirección IP del servidor:

<code>www</code>	<code>IN</code>	<code>A</code>	<code>192.168.108.100</code>
<code>miweb</code>	<code>IN</code>	<code>A</code>	<code>192.168.108.100</code>

```
GNU nano 7.2 /etc/bind/db.em
$TTL      604800
@         IN      SOA      ns.empresa.local. admin.empresa.local. (
                                2025080601 ; Serial
                                604800 ; Refresh
                                86400 ; Retry
                                2419200 ; Expire
                                604800 ) ; Negative Cache TTL
;
@         IN      NS       ns.empresa.local.
ns        IN      A        192.168.108.100
www       IN      A        192.168.108.100
miweb     IN      A        192.168.108.100
```

- Guardar los cambios y cerrar el archivo.
- Reiniciar el servicio BIND para que los cambios surtan efecto:

```
systemctl restart bind9
```

## 2. Configurar el servidor para usar su propio DNS

- Editar el archivo `/etc/resolv.conf` del servidor:

```
nano /etc/resolv.conf
```

- Asegurarse de que la única entrada para `nameserver` sea `192.168.108.100` (la dirección del “Servidor”), para que el servidor use su propio servicio DNS.

```
nameserver 192.168.108.100
```

```
GNU nano 7.2
nameserver 192.168.108.100
```

- Guardar el archivo y salir.

Ahora, cuando hagamos `ping` a `miweb.empresa.local` desde el servidor, debería funcionar correctamente.



## Desde el servidor

```
root@Servidor:~# ping miweb.empresa.local
PING miweb.empresa.local (192.168.108.100) 56(84) bytes of data.
64 bytes from ns.empresa.local (192.168.108.100): icmp_seq=1 ttl=64 time=0.028 ms
64 bytes from ns.empresa.local (192.168.108.100): icmp_seq=2 ttl=64 time=0.045 ms
64 bytes from ns.empresa.local (192.168.108.100): icmp_seq=3 ttl=64 time=0.159 ms
^C
--- miweb.empresa.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2036ms
rtt min/avg/max/mdev = 0.028/0.077/0.159/0.058 ms
root@Servidor:~#
```

## Desde el cliente

```
root@Cliente:~# ping miweb.empresa.local
PING miweb.empresa.local (192.168.108.100) 56(84) bytes of data.
64 bytes from ns.empresa.local (192.168.108.100): icmp_seq=1 ttl=64 time=0.163 ms
64 bytes from ns.empresa.local (192.168.108.100): icmp_seq=2 ttl=64 time=0.247 ms
64 bytes from ns.empresa.local (192.168.108.100): icmp_seq=3 ttl=64 time=0.321 ms
^C
--- miweb.empresa.local ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2043ms
rtt min/avg/max/mdev = 0.163/0.243/0.321/0.064 ms
root@Cliente:~#
```

¿Este error hubiese tenido alguna consecuencia en el funcionamiento?

Sí, ese error de configuración en el archivo `resolv.conf` del servidor habría tenido consecuencias directas en el funcionamiento de la red.

### Consecuencias del error en `/etc/resolv.conf`

La configuración incorrecta de los servidores DNS en la máquina del servidor significaba que:

- No podía resolver nombres de dominio internos: Cuando el servidor intentaba acceder a `miweb.empresa.local`, la consulta se enviaba a los servidores DNS de nuestro ISP. Esos servidores no tienen información sobre nuestro dominio local (`empresa.local`), por lo que la consulta fallaba.
- Problemas con `apt` y servicios externos: Si el servidor hubiera necesitado descargar actualizaciones o conectarse a otros servicios de Internet que usaran un nombre de dominio (por ejemplo, `debian.org`), la consulta DNS habría fallado. Como la máquina cliente depende del servidor para el acceso a Internet, esto también afectaría directamente al cliente, como vimos con el error de `apt`.

La solución de apuntar el `nameserver` a `192.168.108.100` en el servidor es fundamental. De esa manera, el servidor usa su propio servicio `BIND9`, que puede resolver los nombres de nuestra red local y, al mismo tiempo, reenviar las consultas de nombres externos (como `google.com` o `debian.org`) a los servidores DNS públicos que configuramos (`8.8.8.8`).



BLOQUEO DESDE 172.16.200.0/24

Para el caso que necesitemos bloquear el trafico desde una red en particular, haremos lo siguiente:

Actualizamos ip tables

```
nano /etc/iptables/rules.v4
```

Y pegamos lo siguiente

```
*filter
```

```
:INPUT DROP [0:0]
```

```
:FORWARD DROP [0:0]
```

```
:OUTPUT ACCEPT [0:0]
```

```
# Bloquear tráfico de la red 172.16.200.0/24
```

```
-A INPUT -s 172.16.200.0/24 -j DROP
```

```
-A FORWARD -s 172.16.200.0/24 -j DROP
```

```
-A INPUT -i lo -j ACCEPT
```

```
-A FORWARD -i ens37 -o ens33 -j ACCEPT
```

```
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

```
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```



```
-A INPUT -s 192.168.108.0/24 -i ens37 -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 80 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 443 -j ACCEPT
-A INPUT -p udp --dport 67:68 -j ACCEPT
-A INPUT -p udp --sport 67:68 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
```

```
COMMIT
```

```
*nat
```

```
:PREROUTING ACCEPT [0:0]
```

```
:INPUT ACCEPT [0:0]
```

```
:OUTPUT ACCEPT [0:0]
```

```
:POSTROUTING ACCEPT [0:0]
```

```
-A POSTROUTING -o ens33 -j MASQUERADE
```

```
COMMIT
```



```
GNU nano 7.2 /etc/iptables/rules.v4
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]

# Bloquear tráfico de la red 172.16.200.0/24
-A INPUT -s 172.16.200.0/24 -j DROP
-A FORWARD -s 172.16.200.0/24 -j DROP

-A INPUT -i lo -j ACCEPT
-A FORWARD -i ens37 -o ens33 -j ACCEPT
-A FORWARD -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -s 192.168.108.0/24 -i ens37 -j ACCEPT
-A INPUT -p tcp --dport 22 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 80 -j ACCEPT
-A FORWARD -i ens37 -o ens33 -p tcp --dport 443 -j ACCEPT
-A INPUT -p udp --dport 67:68 -j ACCEPT
-A INPUT -p udp --sport 67:68 -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A FORWARD -p icmp -j ACCEPT
COMMIT
*nat
:PREROUTING ACCEPT [0:0]
:INPUT ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A POSTROUTING -o ens33 -j MASQUERADE
COMMIT
```

Guardamos y aplicamos los cambios con

```
iptables-restore < /etc/iptables/rules.v4
```