

# **Instalacion y Configuracion Router en Ubuntu Server**

**Marcos Alonso Cabral, ASIR2**

## Índice

Instalación y configuración de Linux Server con dos adaptadores de red (Adaptador puente y red interna):

- Instalar el servicio DHCP
- Configurar el servicio DHCP para suministrar la red interna
- Instalar el servicio DNS
- Configurar el servicio DNS para suministrar el servicio a la red interna
- Configurar la tabla de enrutado para que las peticiones se reenvíen de la red interna a la externa y tener acceso a internet
- Utilizando un cliente con una sola interfaz en red interna:
  - Comprobar que se asigna la IP dentro de la subred suministrada
  - Comprobar que se puede hacer ping a ips externas
  - Comprobar que se puede hacer ping y navegar a urls externas
  - Comprobar que se puede acceder a un servicio interno (Por ejemplo, un servidor web Apache)

## **Instalación y configuración de Linux Server con dos adaptadores de red** **(Adaptador puente y red interna):**

Primero procedemos a instalar el servidor ubuntu, y como es bastante simple voy a ir dejando capturas seguidas de una breve descripción. En esta primera captura seleccionamos el lenguaje:

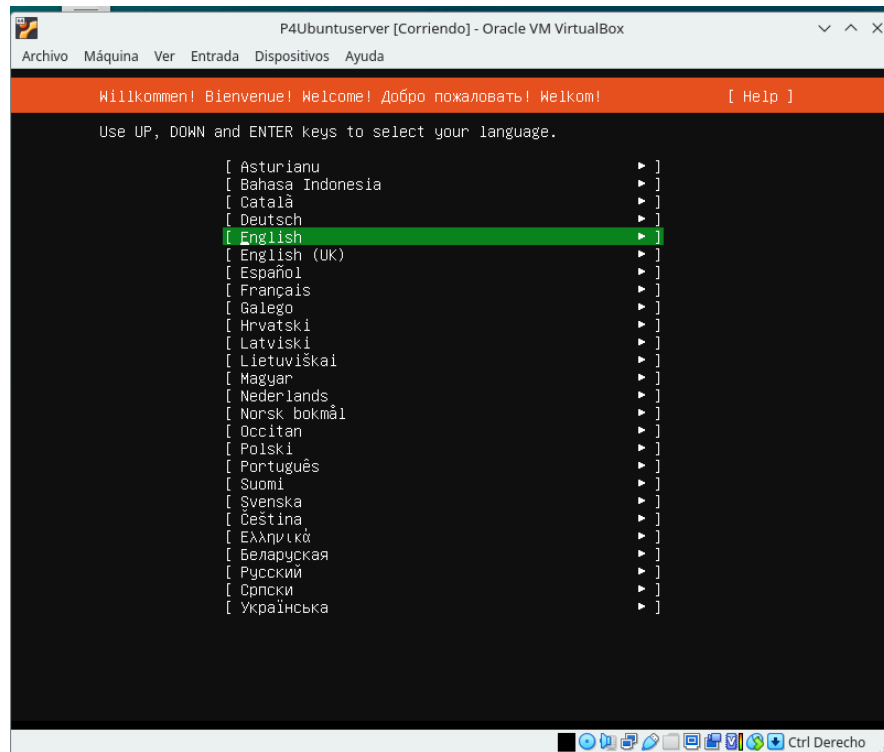


Imagen ilustrativa

Luego con el internet deshabilitado procedemos a instalar el server al completo:

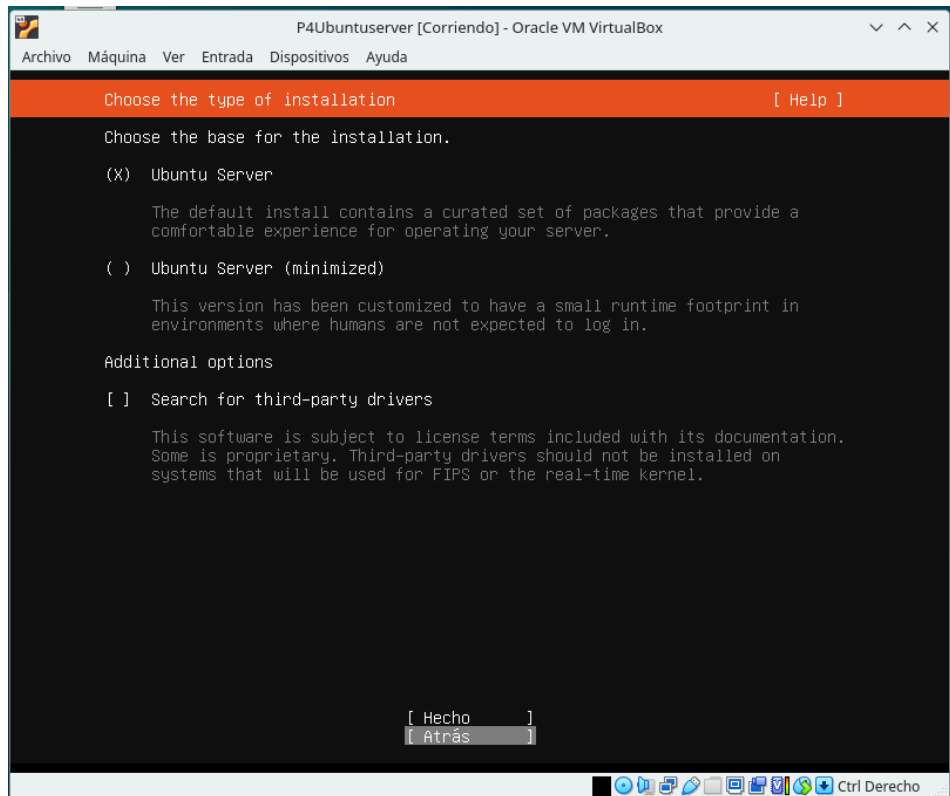


Imagen ilustrativa

Una vez instalado le damos a Hecho:

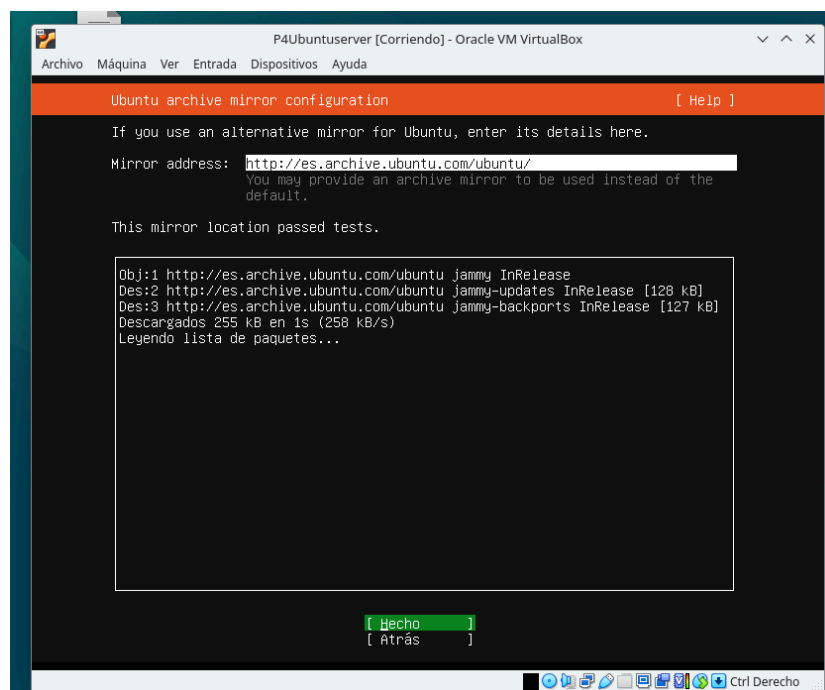


Imagen ilustrativa

Luego le damos a que use todo el disco:

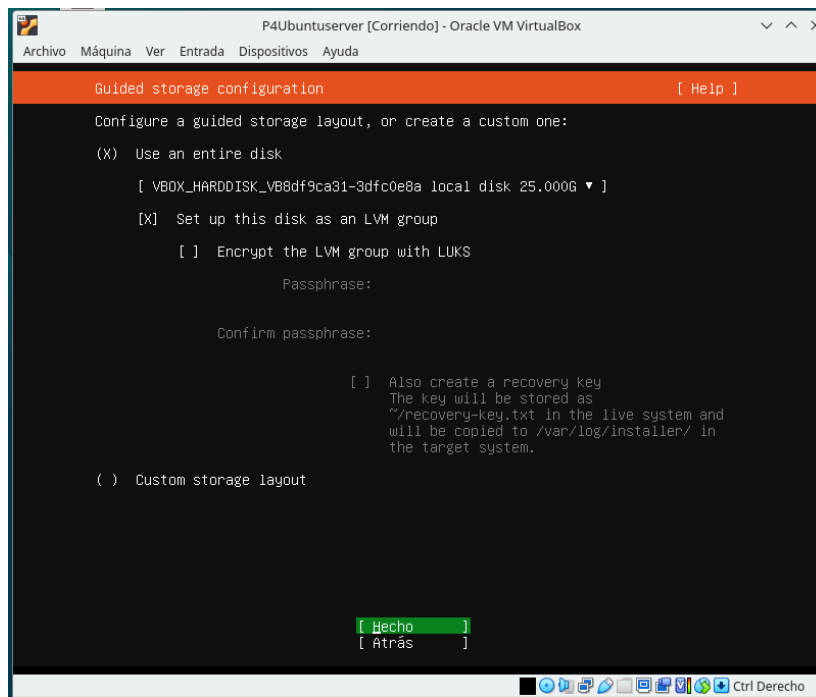


Imagen ilustrativa

Comprobamos la configuración de instalación y le damos a Hecho :

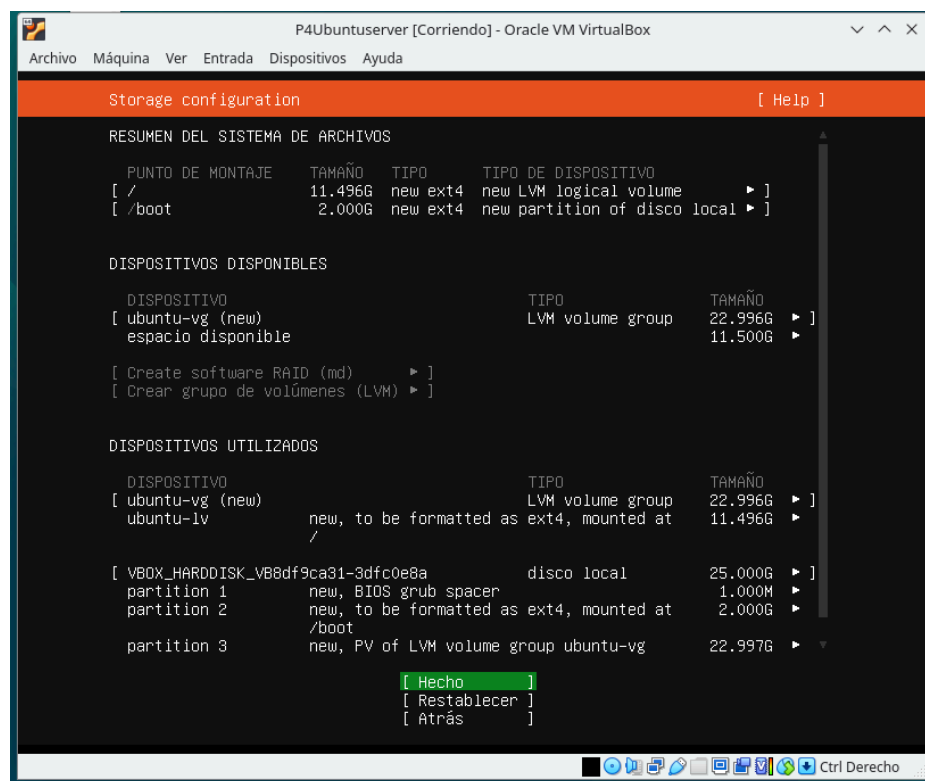


Imagen ilustrativa

Posteriormente procedemos a poner el nombre del server y una contraseña:

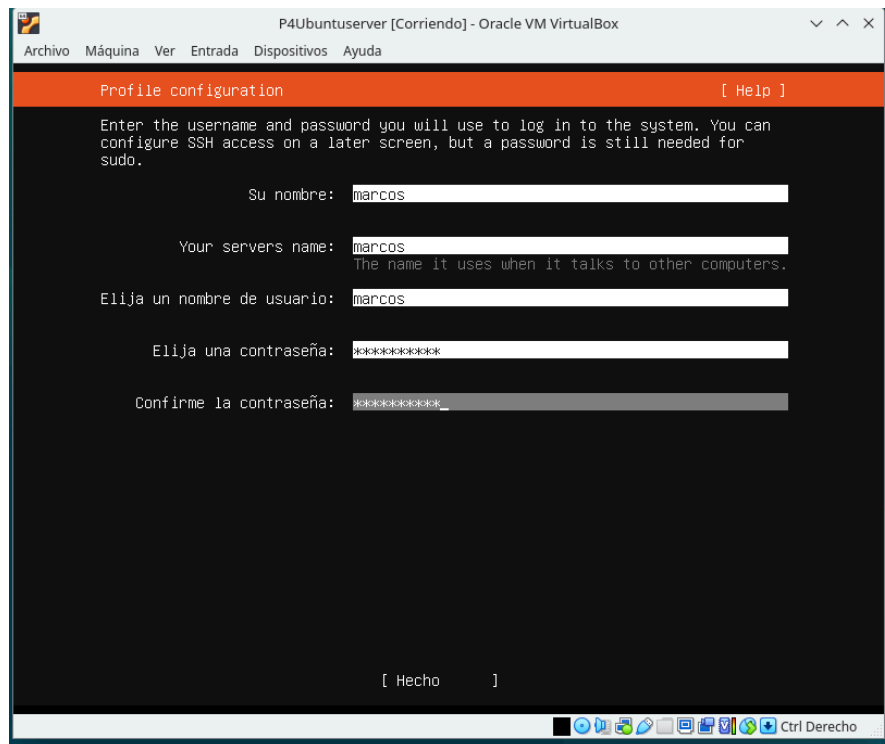


Imagen ilustrativa

Luego saltamos la opción de ubuntu pro:

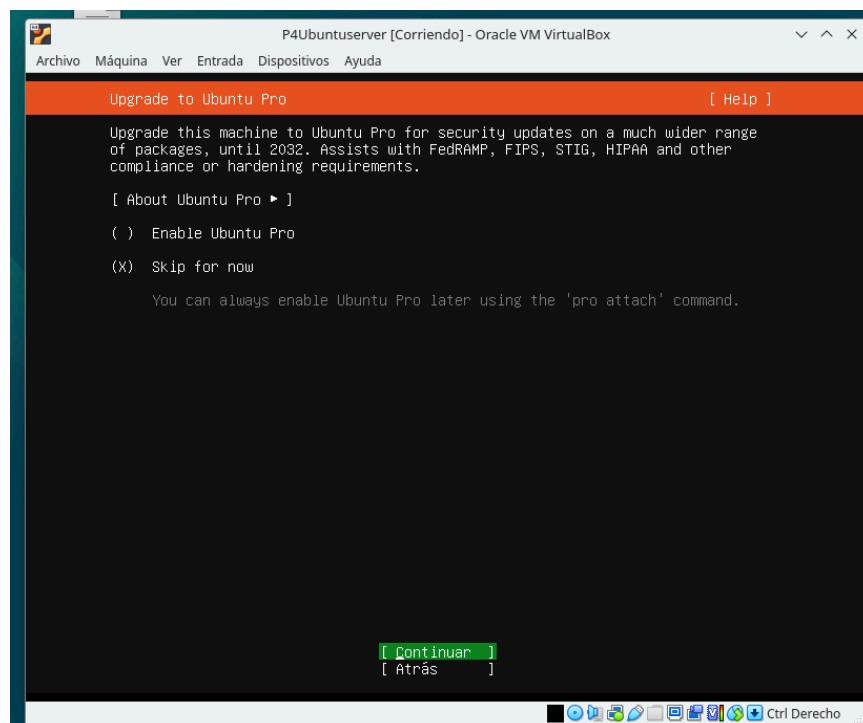


Imagen ilustrativa

Continuamos sin instalar el servidor ssh:

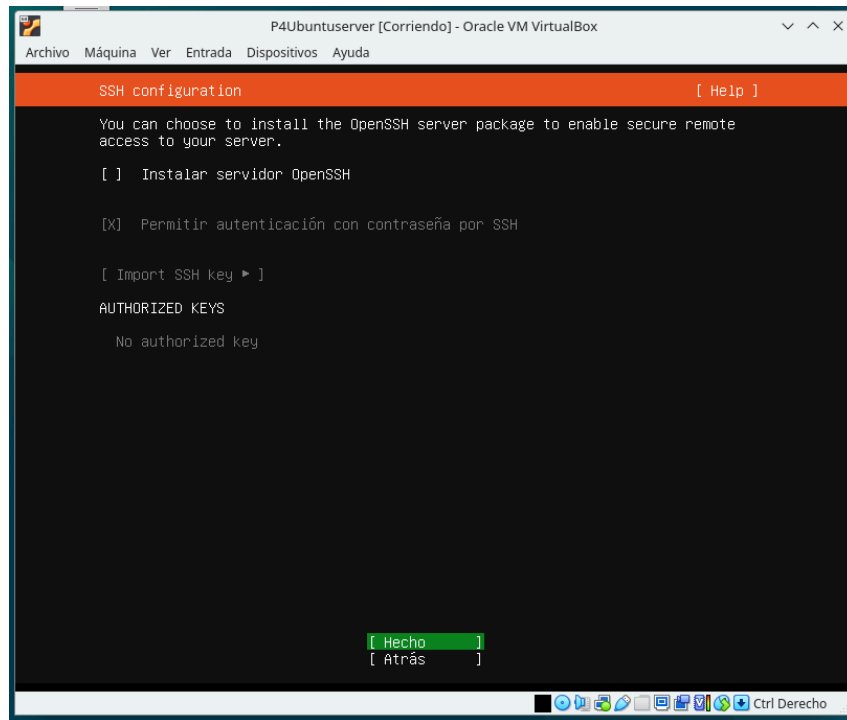


Imagen ilustrativa

Posteriormente procedemos a continuar sin instalar nada a mayores:

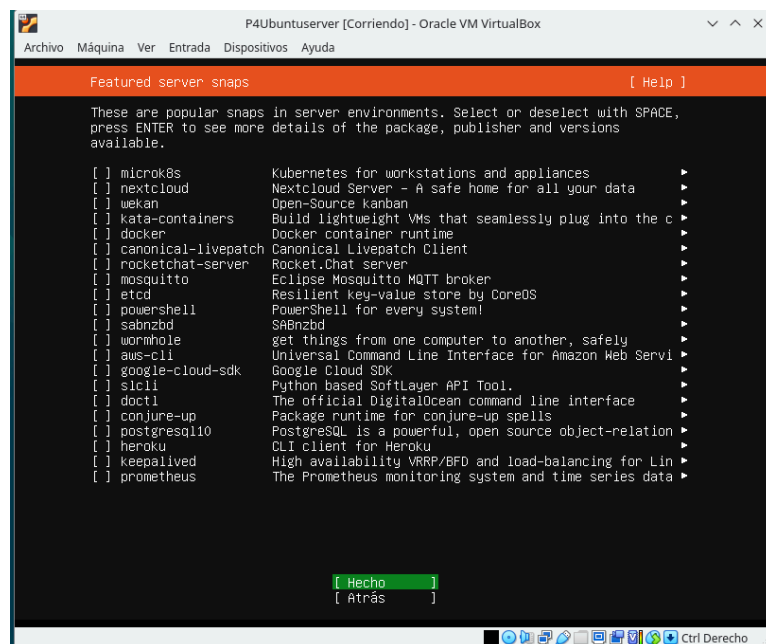


Imagen ilustrativa

Y una vez instalado todo le damos a reiniciar ahora para completar la instalación:

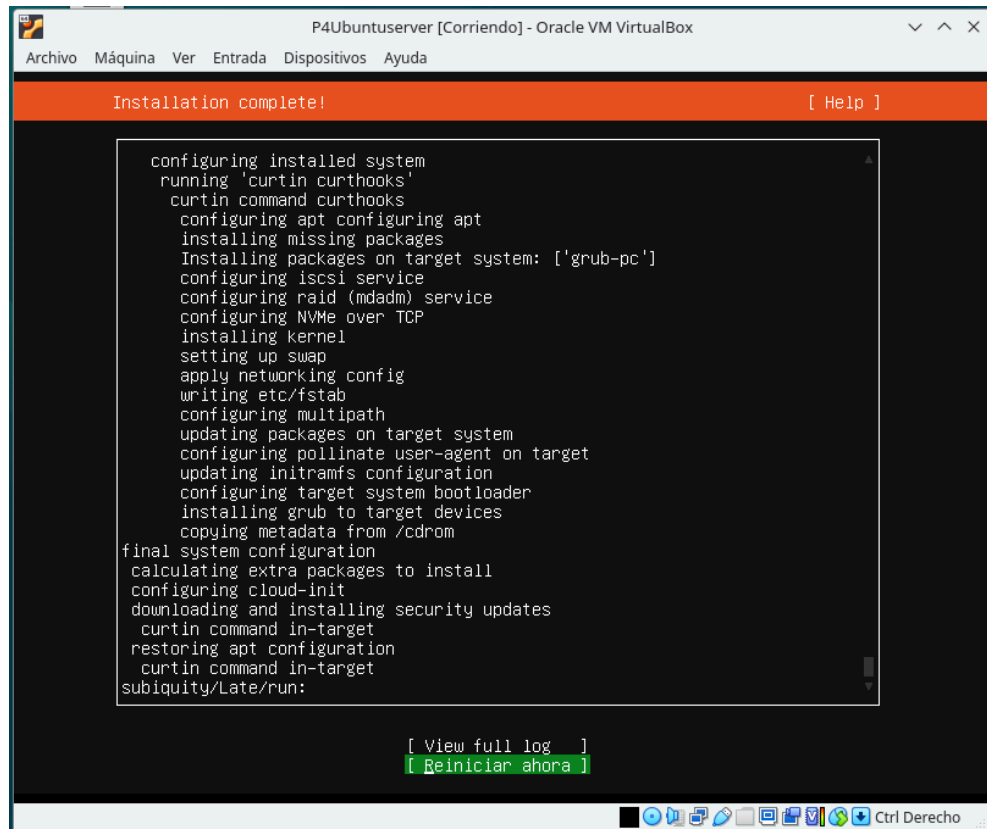


Imagen ilustrativa

### **- Instalar el servicio DHCP**

Para instalar el servicio simplemente debemos poner el comando “sudo apt install isc-dhcp-server” aunque puede que haga falta hacer un “sudo apt update” anteriormente.

```
marcos@marcos:~$ sudo apt install isc-dhcp-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
isc-dhcp-server ya está en su versión más reciente (4.4.1-2.3ubuntu1)
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 19 no se
```

Imagen ilustrativa

### **- Configurar el servicio DHCP para suministrar la red interna**

Para poder suministrar una red interna, primero debemos añadir un adaptador de red para esa red interna, para ello primero, con la máquina apagada debemos ponemos ir a las opciones de esta máquina y poner red interna para luego permitir todo en el modo promiscuo.



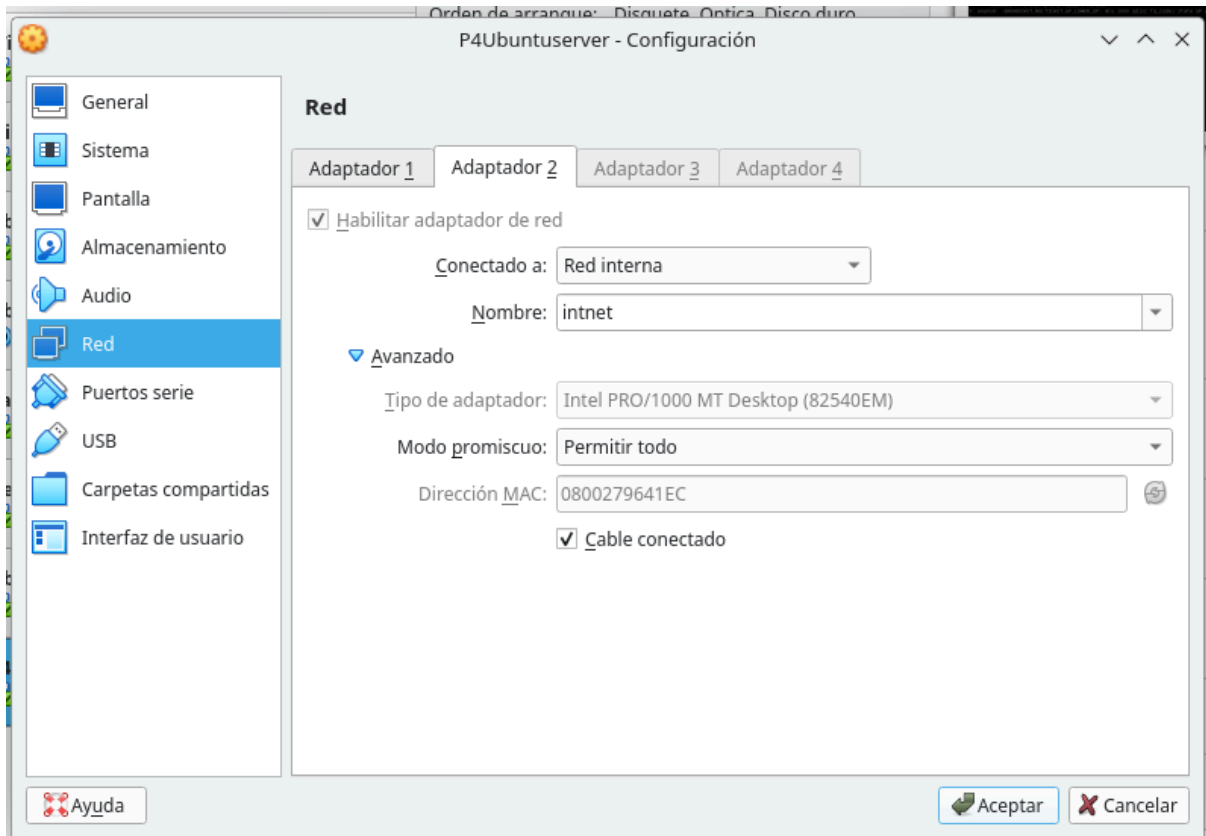
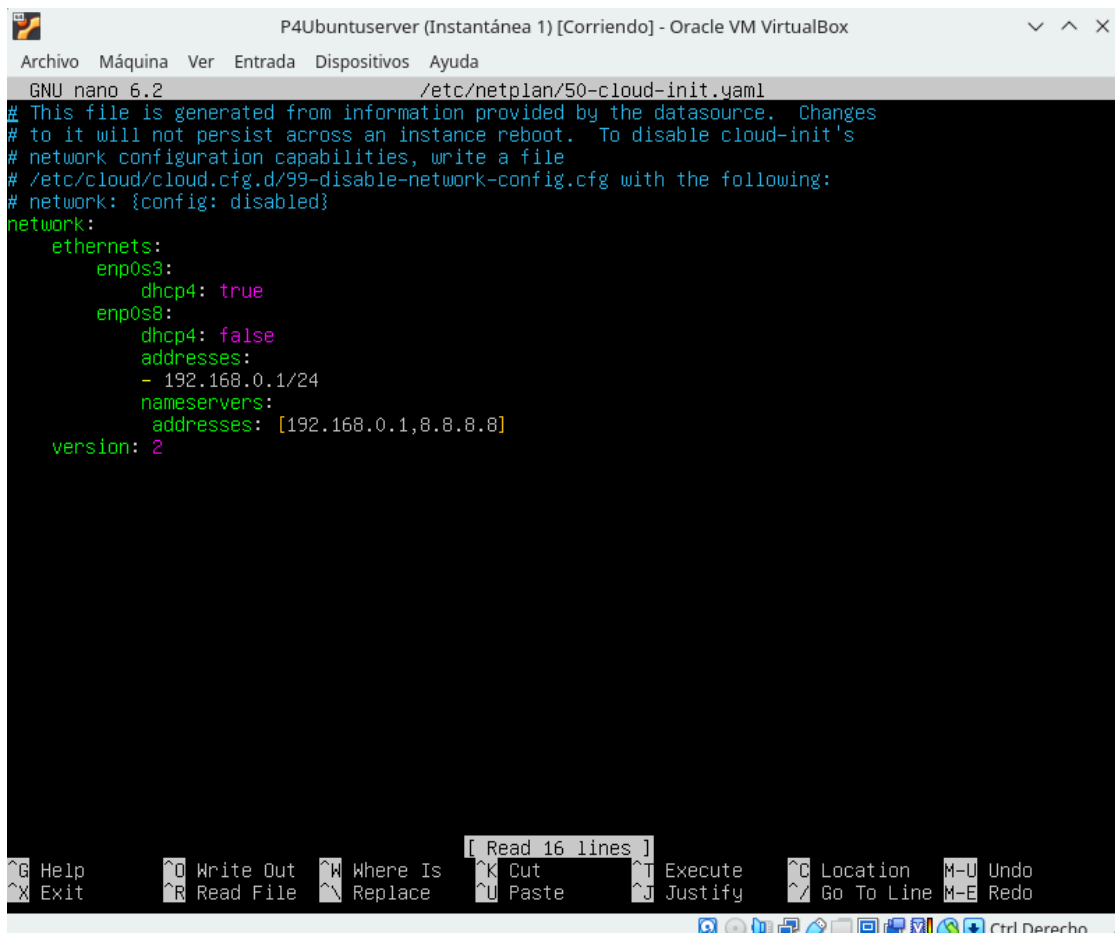


Imagen ilustrativa

Luego debemos modificar el fichero de netplan para configurar estas dos interfaces de red, tanto la que tiene conexión con el exterior como la de red interna. Para ello debemos hacer un “sudo nano /etc/netplan/50-cloud-init.yaml” (si no tienes este fichero tienes que hacer un “sudo netplan generate”) y configurar el adaptador de red enp0s3 y enp0s8 de la siguiente manera (sabiendo que enp0s3 es la interfaz que tiene acceso a internet y la enp0s8 la de la red interna):



The image shows a screenshot of a virtual machine window titled "P4UbuntuServer (Instantánea 1) [Corriendo] - Oracle VM VirtualBox". Inside the VM, the GNU nano 6.2 text editor is open, editing the file `/etc/netplan/50-cloud-init.yaml`. The file contains the following YAML configuration:

```
# 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:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses:
        - 192.168.0.1/24
      nameservers:
        addresses: [192.168.0.1, 8.8.8.8]
  version: 2
```

The nano editor's status bar at the bottom shows various keyboard shortcuts like `Ctrl+O` for Write Out, `Ctrl+R` for Read File, `Ctrl+W` for Where Is, `Ctrl+K` for Cut, `Ctrl+U` for Paste, `Ctrl+T` for Execute, `Ctrl+L` for Location, `Ctrl+G` for Go To Line, `Ctrl+M-U` for Undo, and `Ctrl+M-E` for Redo. A tooltip for `Read 16 lines` is also visible.

Imagen ilustrativa

Una vez configurado este fichero debemos hacer un “sudo netplan apply” para que se nos apliquen los cambios. (Una vez configurado este fichero .yaml yo recomiendo hacer el comando “sudo cp /etc/netplan/50-cloud-init.yaml /etc/netplan/50-cloud-init.yaml.bckp” ya que así hacemos una copia de seguridad porque justamente este sistema ubuntu tiene un bug que hace que cuando reinicias la maquina no se guarda el netplan que acabamos de configurar.) Y para poder ver que realmente se han realizado estos cambios podemos hacer un “ip a” para ver como nos suministra ambas interfaces:

```

marcos@marcos:~$ sudo netplan apply
^[[WARNING:root:Cannot call Open vSwitch: ovsdb-server.service is not running.
marcos@marcos:~$ 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:17:60:20 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
        valid_lft 86397sec preferred_lft 86397sec
    inet6 fe80::a00:27ff:fe17:6020/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:96:41:ec brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.1/24 brd 192.168.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe96:41ec/64 scope link
        valid_lft forever preferred_lft forever
marcos@marcos:~$ _

```

Imagen ilustrativa

Para indicarle cual interfaz va a ser el servidor dhcp (enp0s8) utilizaremos el comando “sudo nano /etc/default/isc-dhcp-server” y en la parte de ipv4 ponemos la interfaz que usaremos que será la enp0s8 (ya que no vamos a utilizar la ipv6).

```

P4Ubuntuuser [Corriendo] - Oracle VM VirtualBox
GNU nano 6.2 /etc/default/isc-dhcp-server *
# 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="enp0s8"
INTERFACESv6=""

```

Imagen ilustrativa

Luego para configurar este servicio de dhcp para que sea el servicio dhcp principal primero debemos hacer “sudo nano /etc/dhcp/dhcpd.conf” y borramos almohadilla en “#authoritative”.

```

GNU nano 6.2 /etc/dhcp/dhcpd.conf *
# dhcpd.conf
#
# Sample configuration file for ISC dhcpd
#
# Attention: If /etc/ltsp/dhcpd.conf exists, that will be used as
# configuration file instead of this file.
#
# option definitions common to all supported networks...
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.org;

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

# The ddns-updates-style parameter controls whether or not the server will
# attempt to do a DNS update when a lease is confirmed. We default to the
# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)
#ddns-update-style none;

# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;

# Use this to send dhcp log messages to a different log file (you also
# have to hack syslog.conf to complete the redirection).
#log-facility local7;

# No service will be given on this subnet, but declaring it helps the
# DHCP server to understand the network topology.

#subnet 10.152.187.0 netmask 255.255.255.0 {

```

Imagen ilustrativa

Ya que estamos aquí, bajamos en este mismo documento para configurar la subred así como el nombre de dominio del servidor como se puede ver a continuación:

```

GNU nano 6.2 /etc/dhcp/dhcpd.conf
#subnet 10.254.239.32 netmask 255.255.255.224 {
#   range dynamic-bootp 10.254.239.40 10.254.239.60;
#   option broadcast-address 10.254.239.31;
#   option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.000 {
  range 192.168.0.2 192.168.0.254;
  option domain-name-servers ns1.marcos.dc.org;
  option domain-name "marcos.dc.org";
  option subnet-mask 255.255.255.0;
  option routers 192.168.0.1;
  option broadcast-address 192.168.0.255;
  default-lease-time 600;
  max-lease-time 7200;
}

# Hosts which require special configuration options can be listed in
# host statements.  If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

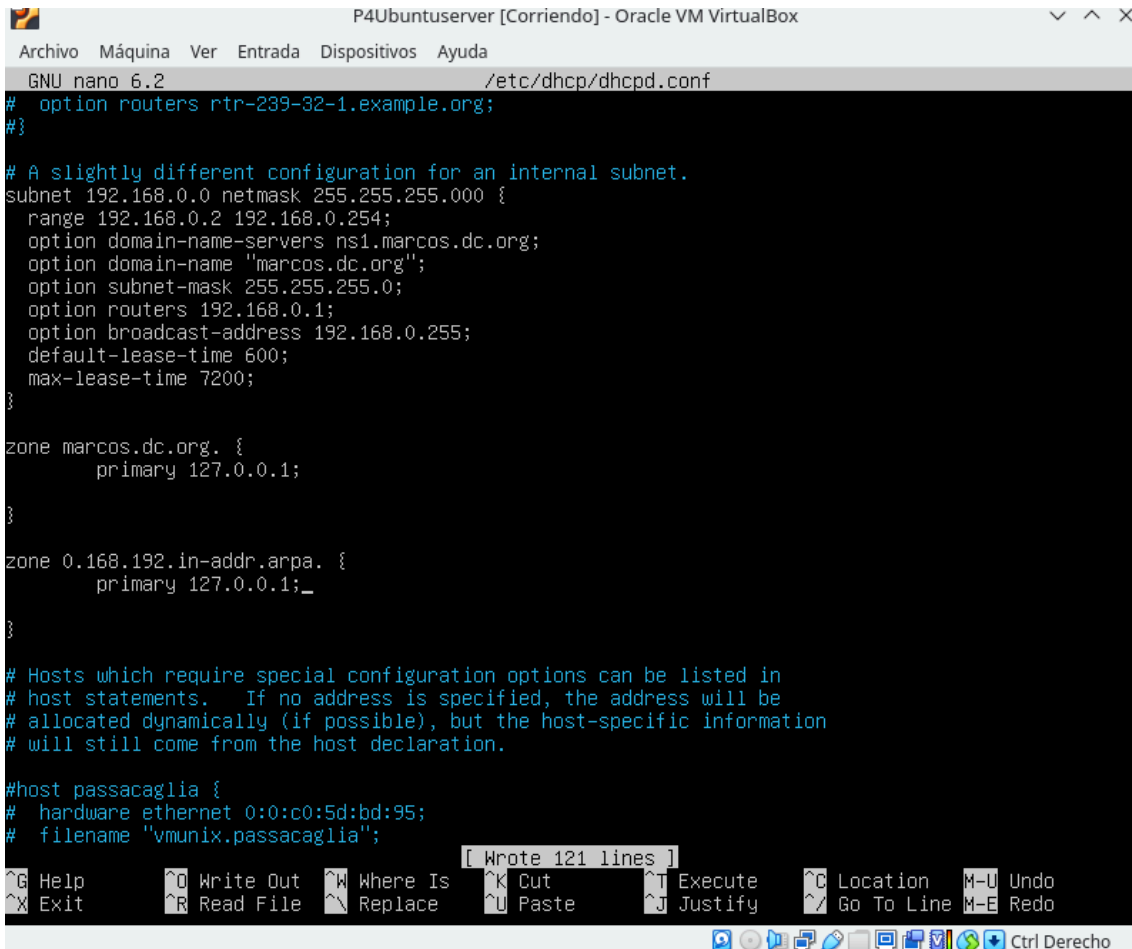
#host passacaglia {
#   hardware ethernet 0:0:c0:5d:bd:95;
#   filename "vmunix.passacaglia";
#   server-name "toccata.example.com";
#}

# Fixed IP addresses can also be specified for hosts.  These addresses
# should not also be listed as being available for dynamic assignment.
# Hosts for which fixed IP addresses have been specified can boot using
# BOOTP or DHCP.  Hosts for which no fixed address is specified can only

```

Imagen ilustrativa

Una vez configurado esta subred, debemos escribir las siguientes zonas, que básicamente es la zona del nombre de dominio principal y la negativa de este mismo:



```

GNU nano 6.2 /etc/dhcp/dhcpd.conf
# option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.000 {
  range 192.168.0.2 192.168.0.254;
  option domain-name-servers ns1.marcos.dc.org;
  option domain-name "marcos.dc.org";
  option subnet-mask 255.255.255.0;
  option routers 192.168.0.1;
  option broadcast-address 192.168.0.255;
  default-lease-time 600;
  max-lease-time 7200;
}

zone marcos.dc.org. {
  primary 127.0.0.1;
}

zone 0.168.192.in-addr.arpa. {
  primary 127.0.0.1;_
}

# Hosts which require special configuration options can be listed in
# host statements.  If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.

#host passacaglia {
#  hardware ethernet 0:0:c0:5d:bd:95;
#  filename "vmunix.passacaglia";

```

[ Wrote 121 lines ]

^G Help    ^O Write Out    ^W Where Is    ^K Cut    ^T Execute    ^C Location    M-U Undo  
 ^X Exit    ^R Read File    ^\_ Replace    ^U Paste    ^J Justify    ^\_ Go To Line    M-E Redo

Ctrl Derecho

Imagen ilustrativa

Cuando acabemos de configurar y guardar este fichero, debemos salir y poner el comando “sudo ip link set enp0s8 up”. Aunque esto solamente inicia la interfaz de enp0s8 por lo que se puede hacer anteriormente. Solo hay que tener en cuenta que para instalar cosas en el servidor o realizar acciones que necesiten de conexión a internet debemos hacer “sudo ip link set enp0s3 up” o “sudo ip link set enp0s8 down” para que se active el la interfaz que tiene conexión a internet (enp0s3) y se desconecte la de la red interna (enp0s8).

Para comprobar que nuestro servicio de dhcp funciona correctamente primero debemos hacer un “sudo service isc-dhcp-server restart” para reiniciar el servicio y luego un “sudo service isc-dhcp-server status” para ver el estado:

```

marcos@marcos:~$ sudo ip link set enp0s8 up
marcos@marcos:~$ sudo service isc-dhcp-server restart
marcos@marcos:~$ sudo service isc-dhcp-server status
● 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 Tue 2024-11-19 16:59:18 UTC; 2s ago
     Docs: man:dhcpd(8)
    Main PID: 1643 (dhcpd)
      Tasks: 4 (limit: 4564)
    Memory: 4.5M
       CPU: 4ms
    CGroup: /system.slice/isc-dhcp-server.service
           └─1643 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dh

nov 19 16:59:18 marcos sh[1643]: PID file: /run/dhcp-server/dhcpd.pid
nov 19 16:59:18 marcos dhcpd[1643]: Wrote 0 leases to leases file.
nov 19 16:59:18 marcos sh[1643]: Wrote 0 leases to leases file.
nov 19 16:59:18 marcos dhcpd[1643]: Listening on LPF/enp0s8/08:00:27:96:41:ec/192.168.0.0/24
nov 19 16:59:18 marcos sh[1643]: Listening on LPF/enp0s8/08:00:27:96:41:ec/192.168.0.0/24
nov 19 16:59:18 marcos dhcpd[1643]: Sending on LPF/enp0s8/08:00:27:96:41:ec/192.168.0.0/24
nov 19 16:59:18 marcos sh[1643]: Sending on LPF/enp0s8/08:00:27:96:41:ec/192.168.0.0/24
nov 19 16:59:18 marcos dhcpd[1643]: Sending on Socket/fallback/fallback-net
nov 19 16:59:18 marcos sh[1643]: Sending on Socket/fallback/fallback-net
nov 19 16:59:18 marcos dhcpd[1643]: Server starting service.
lines 1-21/21 (END)

```

Imagen ilustrativa

### **- Instalar el servicio DNS**

Antes de nada, debemos desactivar la interfaz de red interna que acabamos de iniciar. Se puede hacer por el comando que dije anteriormente o mediante clic derecho en la parte inferior de la máquina virtual como se puede ver a continuación:

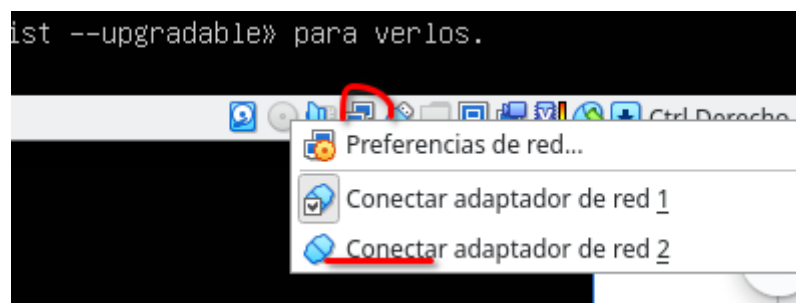
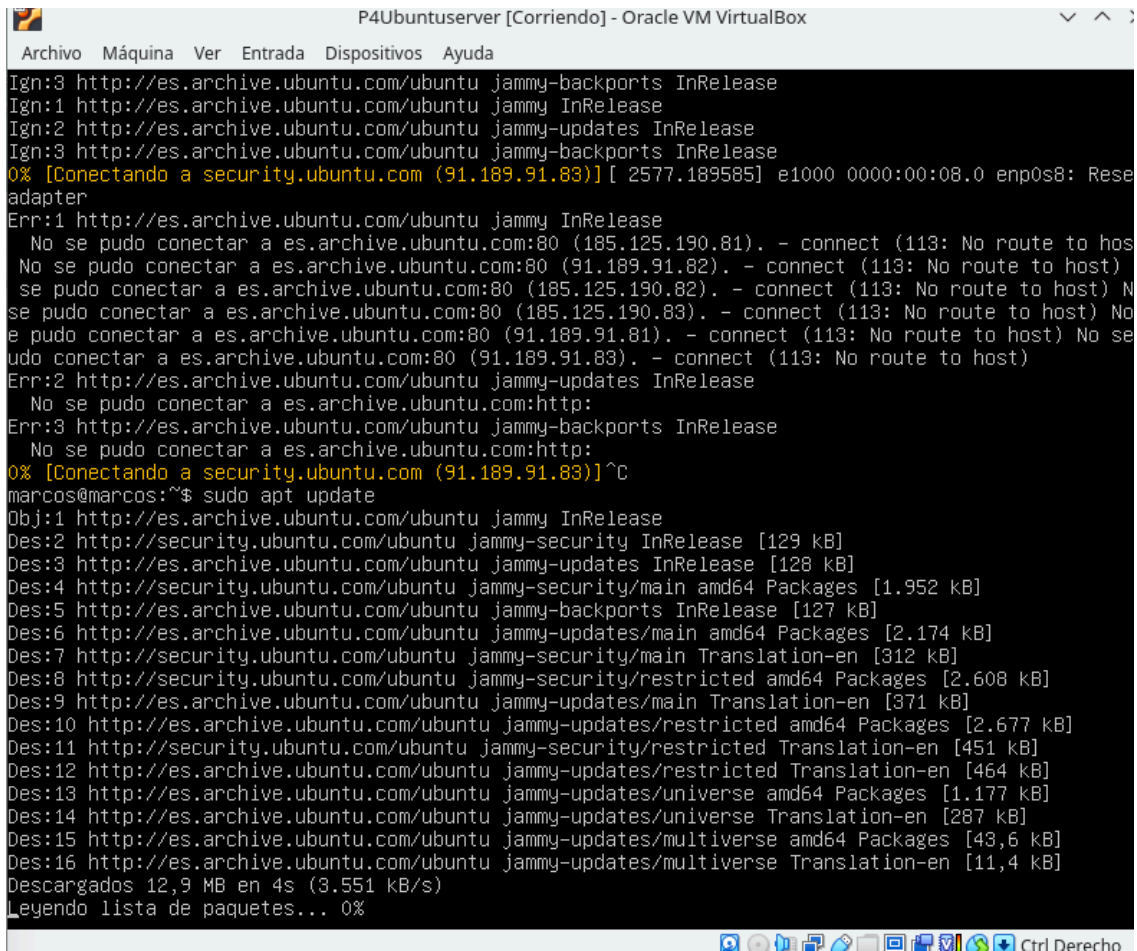


Imagen ilustrativa

Luego ya podemos hacer el “sudo apt update” si es necesario. Y como podemos ver en la siguiente imagen al principio intentaba hacer el update con la interfaz interna activada y luego una vez desactivada ya instalaba:



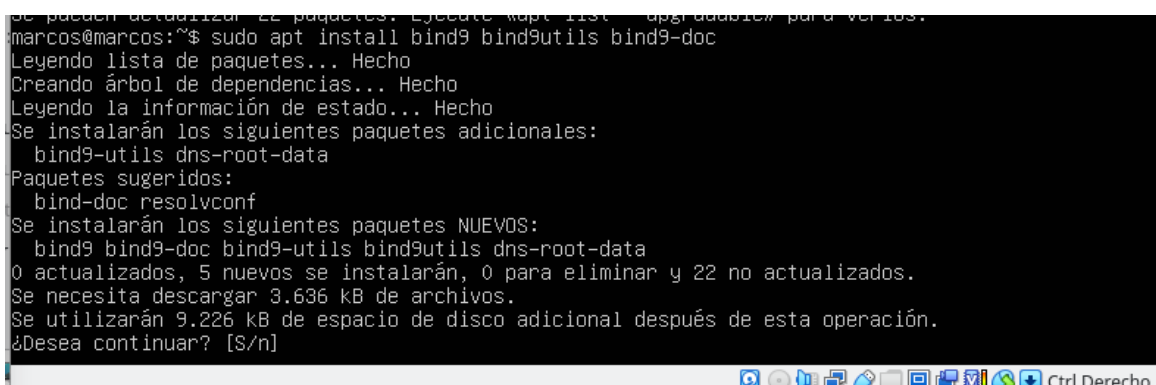
```

P4UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
Ign:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
0% [Conectando a security.ubuntu.com (91.189.91.83)] [ 2577.189585] e1000 0000:00:08:0 enp0s8: Rese
adapter
Err:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
  No se pudo conectar a es.archive.ubuntu.com:80 (185.125.190.81). - connect (113: No route to hos
  No se pudo conectar a es.archive.ubuntu.com:80 (91.189.91.82). - connect (113: No route to host)
  No se pudo conectar a es.archive.ubuntu.com:80 (185.125.190.82). - connect (113: No route to host) N
  No se pudo conectar a es.archive.ubuntu.com:80 (185.125.190.83). - connect (113: No route to host) No
  No se pudo conectar a es.archive.ubuntu.com:80 (91.189.91.81). - connect (113: No route to host) No se
  No se pudo conectar a es.archive.ubuntu.com:80 (91.189.91.83). - connect (113: No route to host)
Err:2 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease
  No se pudo conectar a es.archive.ubuntu.com:http:
Err:3 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease
  No se pudo conectar a es.archive.ubuntu.com:http:
0% [Conectando a security.ubuntu.com (91.189.91.83)] ^C
marcos@marcos:~$ sudo apt update
Obj:1 http://es.archive.ubuntu.com/ubuntu jammy InRelease
Des:2 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Des:3 http://es.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Des:4 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1.952 kB]
Des:5 http://es.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
Des:6 http://es.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [2.174 kB]
Des:7 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [312 kB]
Des:8 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2.608 kB]
Des:9 http://es.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [371 kB]
Des:10 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2.677 kB]
Des:11 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [451 kB]
Des:12 http://es.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [464 kB]
Des:13 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1.177 kB]
Des:14 http://es.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [287 kB]
Des:15 http://es.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43,6 kB]
Des:16 http://es.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [11,4 kB]
Descargados 12,9 MB en 4s (3.551 kB/s)
Leyendo lista de paquetes... 0%

```

Imagen ilustrativa

Una vez actualizados todos los repositorios instalamos el DNS de bind9 con el comando “sudo apt install bind9 bind9utils bind9-doc”



```

Se pueden actualizar 22 paquetes. Ejecute 'apt list --upgradable' para verlos.
marcos@marcos:~$ sudo apt install bind9 bind9utils bind9-doc
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  bind9-utils dns-root-data
Paquetes sugeridos:
  bind-doc resolvconf
Se instalarán los siguientes paquetes NUEVOS:
  bind9 bind9-doc bind9-utils bind9utils dns-root-data
0 actualizados, 5 nuevos se instalarán, 0 para eliminar y 22 no actualizados.
Se necesita descargar 3.636 kB de archivos.
Se utilizarán 9.226 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]

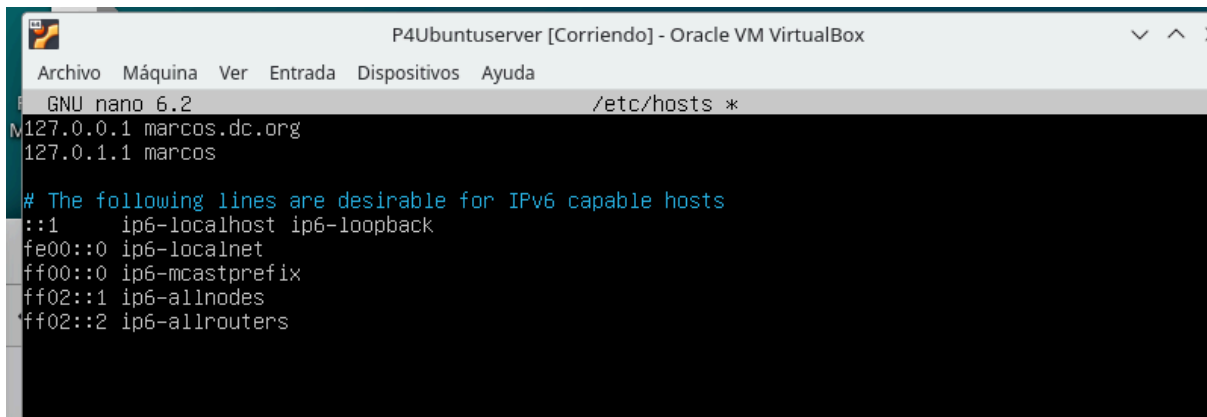
```

Imagen ilustrativa



### **- Configurar el servicio DNS para suministrar el servicio a la red interna**

Para empezar a configurar este servicio de DNS debemos acceder al fichero de hosts con “sudo nano /etc/hosts” para así cambiar la primera línea que pone localhost por tu servidor para que así se resuelva por el dns del mismo (que es nombre que le dimos en el documento /etc/dhcp/dhcpd.conf en el apartado de option domain-name-servers):



```

GNU nano 6.2 /etc/hosts *
127.0.0.1 marcos.dc.org
127.0.1.1 marcos

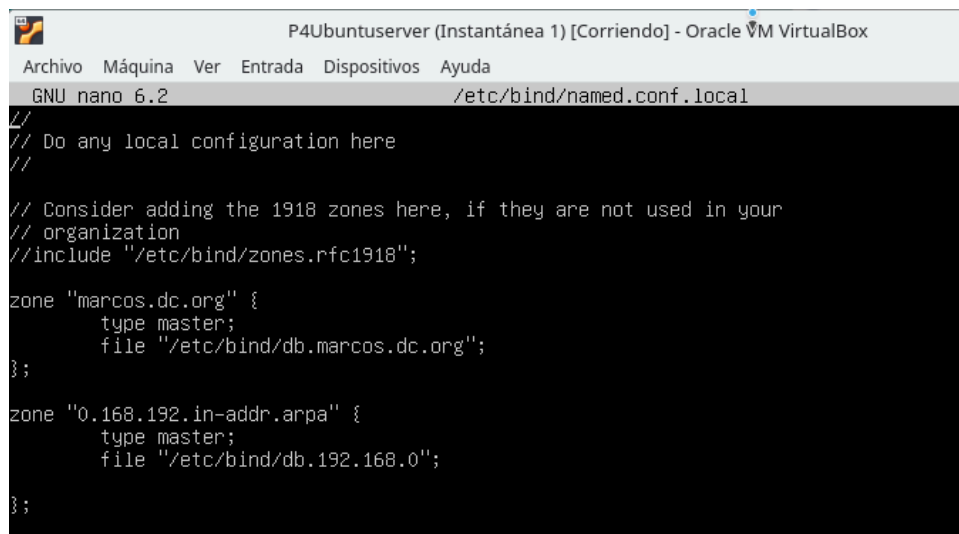
# The following lines are desirable for IPv6 capable hosts
::1 ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

Imagen ilustrativa

Un fichero que en esta práctica no tocaremos pero que se puede modificar para incluir únicamente las redes que podemos escuchar sería el fichero /etc/bind/named.conf (para modificarlo: “sudo nano /etc/bind/named.conf”).

Luego nos metemos en el fichero /etc/bind/named.conf.local (para acceder: “sudo nano /etc/bind/named.conf.local”) y ponemos la zona de nuestro servidor junto con su zona negativa que también configuramos en el servicio dhcp.



```

GNU nano 6.2 /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 "marcos.dc.org" {
    type master;
    file "/etc/bind/db.marcos.dc.org";
};

zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.192.168.0";
};

```

Imagen ilustrativa

Luego copiamos el archivo de base de datos de la zona local para luego modificarla con nuestro dominio del servidor:

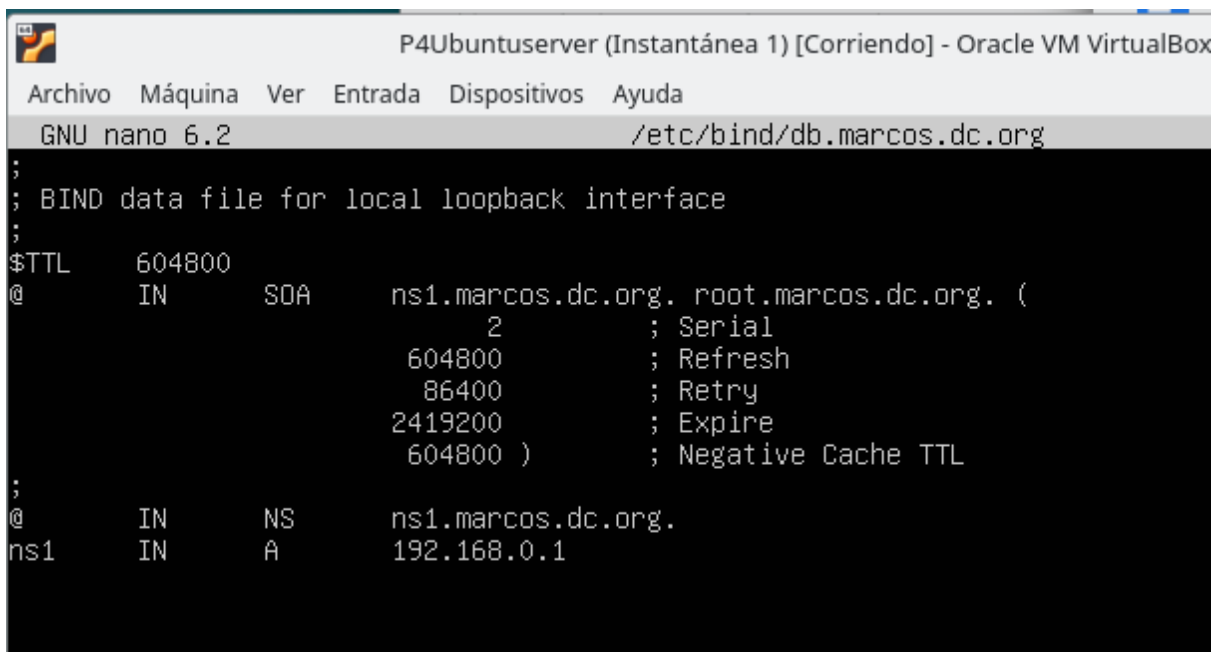
```
marcos@marcos:~$ sudo cp /etc/bind/db.local /etc/bind/db.marcos.dc.org
marcos@marcos:~$ _
```

Imagen ilustrativa

Posteriormente hacemos un sudo nano a este fichero que acabamos de copiar (etc/bind/db.marcos.dc.org)

Y aquí modificamos todos los parámetros poniendo los de nuestro servidor. Cambiamos la parte de la A que es la ipv4 y eliminamos la AAAA que es la ipv6 ya que no vamos a recibir peticiones ipv6.

Una cosa importante es poner puntos al final de la dirección de nuestro servidor ya que sino da error. Una vez acabada la configuración quedaría tal que así:



```
P4Ubtuserver (Instantánea 1) [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 6.2 /etc/bind/db.marcos.dc.org
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      ns1.marcos.dc.org. root.marcos.dc.org. (
; Serial
; 604800      ; Refresh
; 86400       ; Retry
; 2419200    ; Expire
; 604800 )    ; Negative Cache TTL
;
@         IN      NS       ns1.marcos.dc.org.
ns1       IN      A        192.168.0.1
```

Imagen ilustrativa

A continuación volvemos a copiar el fichero pero esta vez como es de la parte negativa del dominio hacemos el comando “sudo cp /etc/bind/db.127 /etc/bind/db.192.168”

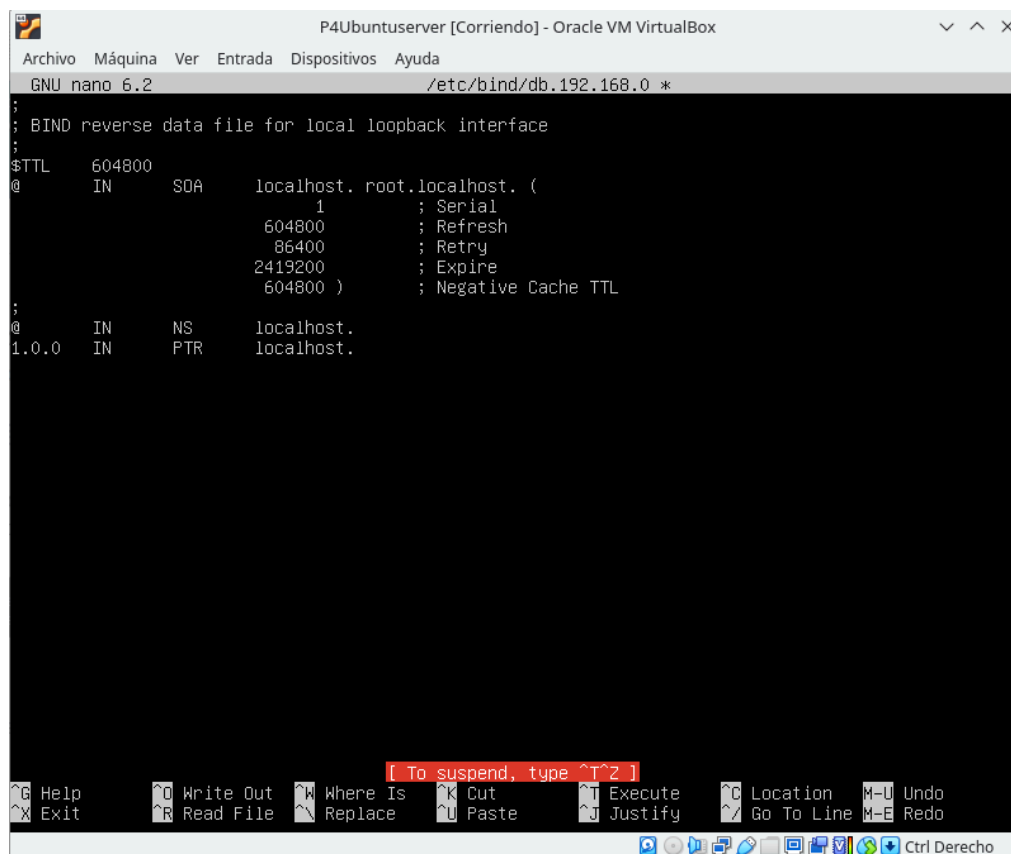
```

marcos@marcos:~$ sudo cp /etc/bind/db.127 /etc/bind/db.192.168.0
marcos@marcos:~$ _

```

Imagen ilustrativa

Luego accedemos a ese fichero que acabamos de modificar con “sudo nano /etc/bind/db.192.168.0” y cambiamos lo mismo, y para que se vea todo lo que modificó gráficamente, voy a dejar una imagen del archivo sin modificar y a continuación el ya modificado y listo para guardar (como se puede apreciar cambiamos todo lo que poner localhost por nuestro nombre de dominio de servidor):



```

P4UbuntuServer [Corriendo] - Oracle VM VirtualBox
GNU nano 6.2 /etc/bind/db.192.168.0 *
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@      IN      SOA      localhost. root.localhost. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@      IN      NS       localhost.
1.0.0  IN      PTR      localhost.

```

Imagen ilustrativa

```

P4Ubuntuserver [Corriendo] - Oracle VM VirtualBox
GNU nano 6.2 /etc/bind/db.192.168.0
;
; BIND reverse data file for local loopback interface
;
$TTL 604800
@ IN SOA ns1.marcos.castelao.org. root.marcos.dc.org. (
        1 ; Serial
        604800 ; Refresh
        86400 ; Retry
        2419200 ; Expire
        604800 ) ; Negative Cache TTL
;
@ IN NS ns1.marcos.dc.org.
1 IN PTR ns1.marcos.dc.org.

```

[ Wrote 13 lines ]

Help Write Out Where Is Cut Execute Location M-U Undo  
Exit Read File Replace Paste Justify Go To Line M-E Redo

Ctrl Derech

Imagen ilustrativa

Una vez configurado esto, si esta todo bien configurado, al hacer un “sudo named-checkconf” debería de no aparecer nada:

```

marcos@marcos:~$ sudo named-checkconf
marcos@marcos:~$ _

```

Imagen ilustrativa

Luego para comprobar el otro fichero de configuración hacemos un “sudo named-checkconf /etc/bind/named.conf” y debería de dar lo mismo:

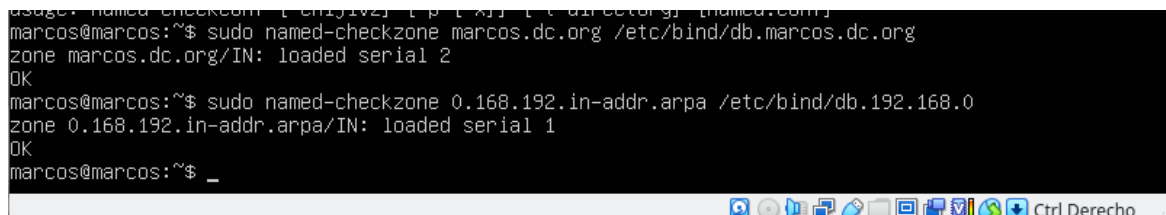
```

marcos@marcos:~$ sudo named-checkconf
marcos@marcos:~$ sudo named-checkconf /etc/bind/named.conf
marcos@marcos:~$

```

Imagen ilustrativa

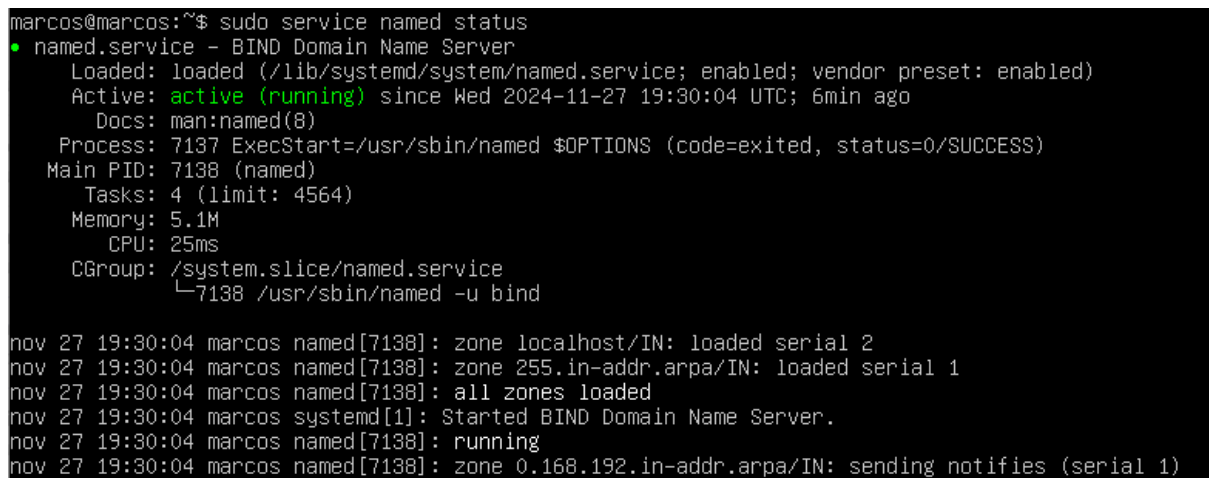
Luego para ver si las zonas están bien configuradas ponemos lo siguiente con “checkzone”, primero un “sudo named-checkzone marcos.dc.org /etc/bind/db.marcos.dc.org” para ver si la sintaxis de la zona del nombre de dominio del servidor está correcta y luego un “sudo named-checkzone 0.168.192.in-addr.arpa /etc/bin/db.192.168.0” para comprobar si la sintaxis de la zona negativa también está correcta (si no esta correcto, evidentemente saltara un error) :



```
usage: named-checkzone [-c file] [-t file] [-d directory] [named.conf]
marcos@marcos:~$ sudo named-checkzone marcos.dc.org /etc/bind/db.marcos.dc.org
zone marcos.dc.org/IN: loaded serial 2
OK
marcos@marcos:~$ sudo named-checkzone 0.168.192.in-addr.arpa /etc/bind/db.192.168.0
zone 0.168.192.in-addr.arpa/IN: loaded serial 1
OK
marcos@marcos:~$
```

Imagen ilustrativa

Luego reiniciamos el servicio con “sudo service named restart” y vemos el status su estado (está activo en mi caso):



```
marcos@marcos:~$ sudo service named status
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2024-11-27 19:30:04 UTC; 6min ago
     Docs: man:named(8)
  Process: 7137 ExecStart=/usr/sbin/named $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 7138 (named)
    Tasks: 4 (limit: 4564)
   Memory: 5.1M
        CPU: 25ms
   CGroup: /system.slice/named.service
           └─7138 /usr/sbin/named -u bind

nov 27 19:30:04 marcos named[7138]: zone localhost/IN: loaded serial 2
nov 27 19:30:04 marcos named[7138]: zone 255.in-addr.arpa/IN: loaded serial 1
nov 27 19:30:04 marcos named[7138]: all zones loaded
nov 27 19:30:04 marcos systemd[1]: Started BIND Domain Name Server.
nov 27 19:30:04 marcos named[7138]: running
nov 27 19:30:04 marcos named[7138]: zone 0.168.192.in-addr.arpa/IN: sending notifies (serial 1)
```

Imagen ilustrativa

Luego en el fichero “/etc/bind/named.conf.options” modificamos el contenido con “sudo nano /etc/bind/named.conf.options” y ponemos los forwarders que seran los servidores a los que enviamos las peticiones (8.8.8.8 google) si no somos capaces de responderlas por nosotros mismos y cambiamos el “dnssec-validation” en “no” para que luego nos deje conectarnos y no nos bloquee la conexión.

A continuación volveré a dejar una image del fichero sin modificar seguido del fichero modificado :

```

GNU nano 6.2 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    // forwarders {
    //     0.0.0.0;
    // };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation auto;

    listen-on-v6 { any; };
};

```

Read 24 lines

Imagen ilustrativa

```

GNU nano 6.2 /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    //=====
    // If BIND logs error messages about the root key being expired,
    // you will need to update your keys. See https://www.isc.org/bind-keys
    //=====
    dnssec-validation no;

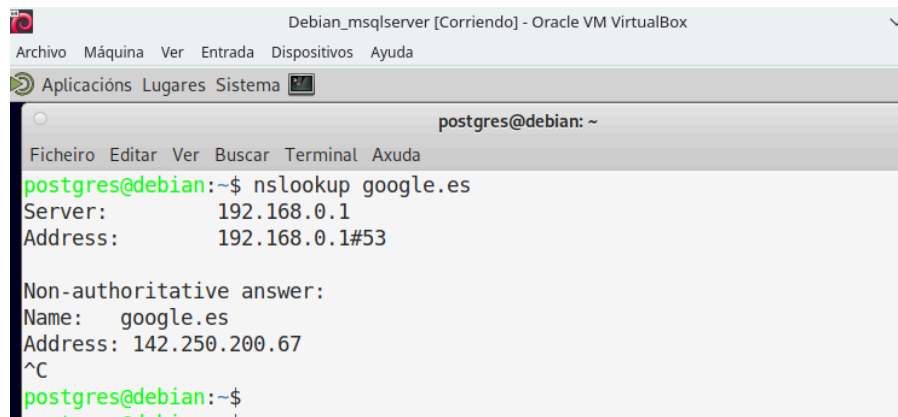
    listen-on-v6 { any; };
};

```

Wrote 25 lines

Imagen ilustrativa

Una vez configurado todo voy a comprobar que al menos tiene conexión el cliente para google (con el comando “nslookup google.es” y cabe destacar que el cliente debe estar en red interna). Ya que nslookup pregunta a nuestro servidor DNS que si tiene la ip de google que es la única que tiene junto con 8.8.4.4 (que también es google) por eso puede resolver la consulta. Y para resolver otras consultas necesitamos la tabla de enrutado que haremos a continuación:



```
Debian_msqlserver [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
Aplicacions  Lugares  Sistema
postgres@debian: ~
Ficheiro  Editar  Ver  Buscar  Terminal  Axuda
postgres@debian:~$ nslookup google.es
Server:      192.168.0.1
Address:     192.168.0.1#53

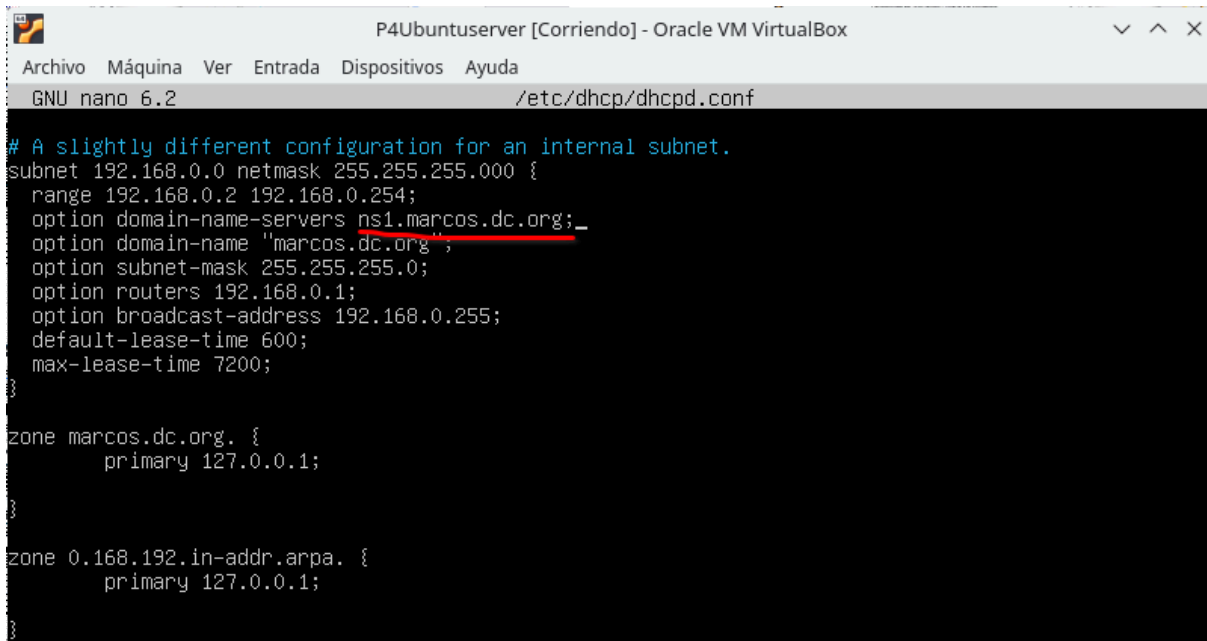
Non-authoritative answer:
Name:   google.es
Address: 142.250.200.67
^C
postgres@debian:~$
```

Imagen ilustrativa

**- Configurar la tabla de enrutado para que las peticiones se reenvíen de la red interna a la externa y tener acceso a internet**

Para que el cliente pueda resolver otras consultas externas necesitamos cambiar un ajuste del fichero /etc/dhcp/dhcpd.conf, cambiando la “option domain-name-servers” por la ip del servidor (192.168.0.1) y la ip del servidor de google (8.8.8.8). La razón de poner como segunda opción la ip de google es para que el cliente al acceder a internet primero resuelva por nuestro servidor dns y tenga a google de servidor backup y que la conexion sea mas rapido al no tener que estar preguntando nuestro servidor principal por todas las peticiones que no sepa responder (así las responde google).

A continuación dejaré las capturas de el fichero “/etc/dhcp/dhcpd.conf” sin modificar y luego el modificado:



```

P4Ubtuserver [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 6.2 /etc/dhcp/dhcpd.conf

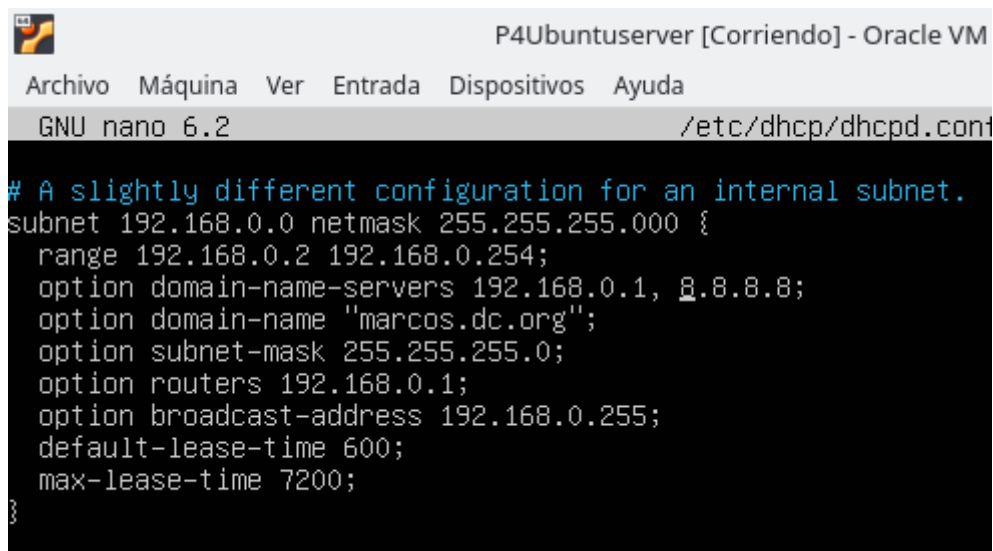
# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.000 {
    range 192.168.0.2 192.168.0.254;
    option domain-name-servers ns1.marcos.dc.org;_
    option domain-name "marcos.dc.org";
    option subnet-mask 255.255.255.0;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    default-lease-time 600;
    max-lease-time 7200;
}

zone marcos.dc.org. {
    primary 127.0.0.1;
}

zone 0.168.192.in-addr.arpa. {
    primary 127.0.0.1;
}

```

Imagen ilustrativa



```

P4Ubtuserver [Corriendo] - Oracle VM
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 6.2 /etc/dhcp/dhcpd.conf

# A slightly different configuration for an internal subnet.
subnet 192.168.0.0 netmask 255.255.255.000 {
    range 192.168.0.2 192.168.0.254;
    option domain-name-servers 192.168.0.1, 8.8.8.8;
    option domain-name "marcos.dc.org";
    option subnet-mask 255.255.255.0;
    option routers 192.168.0.1;
    option broadcast-address 192.168.0.255;
    default-lease-time 600;
    max-lease-time 7200;
}

```

Imagen ilustrativa

Una vez aplicado esto, debemos hacer un “sudo service restart” para reiniciar el servicio dhcp y que se apliquen los cambios.

```

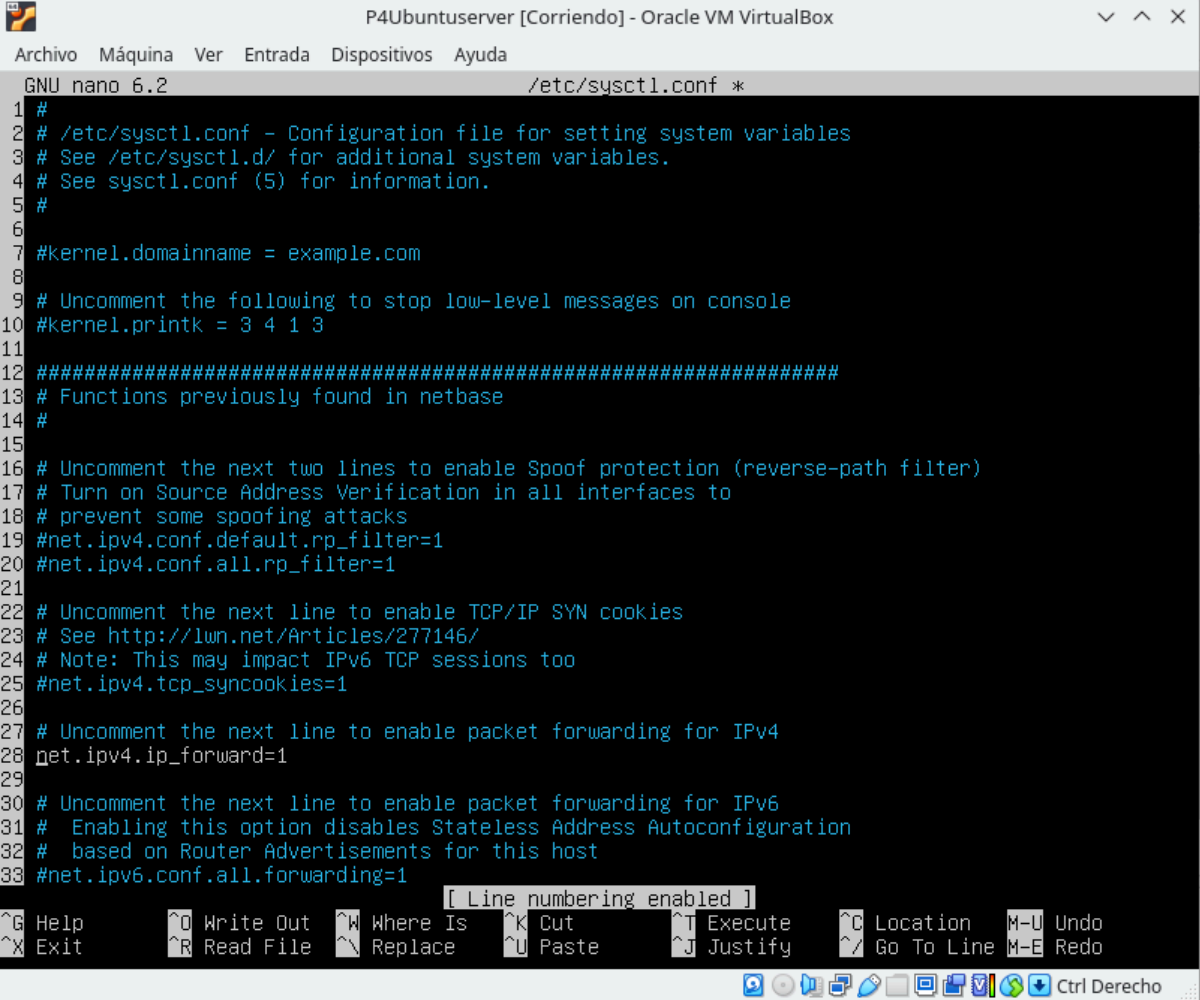
marcos@marcos:~$ sudo service isc-dhcp-server restart
marcos@marcos:~$ _

```

Imagen ilustrativa



Posteriormente hacemos “sudo nano /etc/sysctl.conf” y descomentamos la línea 28 para habilitar que google nos resuelva lo que nosotros no podremos:



```

GNU nano 6.2 /etc/sysctl.conf *
1 #
2 # /etc/sysctl.conf - Configuration file for setting system variables
3 # See /etc/sysctl.d/ for additional system variables.
4 # See sysctl.conf (5) for information.
5 #
6
7 #kernel.domainname = example.com
8
9 # Uncomment the following to stop low-level messages on console
10 #kernel.printk = 3 4 1 3
11
12 #####
13 # Functions previously found in netbase
14 #
15
16 # Uncomment the next two lines to enable Spoof protection (reverse-path filter)
17 # Turn on Source Address Verification in all interfaces to
18 # prevent some spoofing attacks
19 #net.ipv4.conf.default.rp_filter=1
20 #net.ipv4.conf.all.rp_filter=1
21
22 # Uncomment the next line to enable TCP/IP SYN cookies
23 # See http://lwn.net/Articles/277146/
24 # Note: This may impact IPv6 TCP sessions too
25 #net.ipv4.tcp_syncookies=1
26
27 # Uncomment the next line to enable packet forwarding for IPv4
28 net.ipv4.ip_forward=1
29
30 # Uncomment the next line to enable packet forwarding for IPv6
31 # Enabling this option disables Stateless Address Autoconfiguration
32 # based on Router Advertisements for this host
33 #net.ipv6.conf.all.forwarding=1

```

Imagen ilustrativa

Una vez modificado esto debemos hacer el comando “sudo sysctl -p” para que aplique los cambios realizados.

```

marcos@marcos:~$ sudo sysctl -p
net.ipv4.ip_forward = 1

```

Imagen ilustrativa

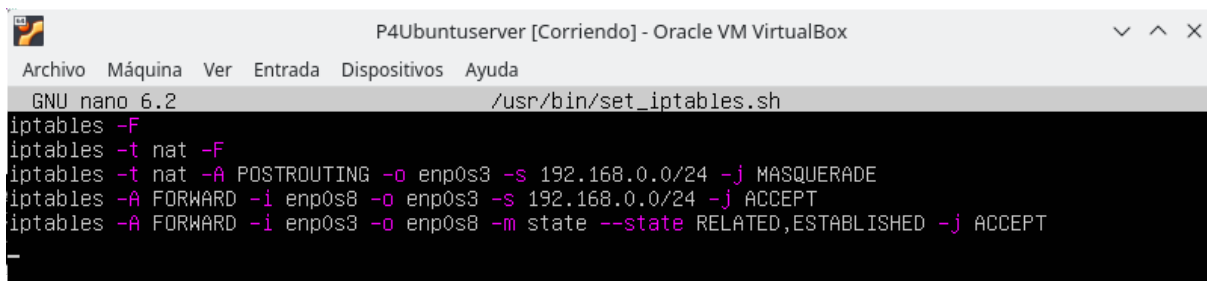
El siguiente paso se puede hacer con “nftables” pero yo elegí hacerlo con el sistema de “iptables”.

Por lo que debemos crear un fichero en /usr/bin llamado “set\_ipables.sh” para configurar el enrutador mediante la herramienta iptables. También debemos hacer un chmod para que podamos modificarlo y ejecutarlo ya que luego entraremos en este archivo y crearemos la configuración:

```
marcos@marcos:~$ sudo touch /usr/bin/set_ipables.sh
marcos@marcos:~$ sudo chmod 755 /usr/bin/set_ipables.sh
marcos@marcos:~$ sudo nano /usr/bin/set_ipables.sh
```

Imagen ilustrativa

Luego cómo se puede ver en la anterior captura hacemos un nano para meter las siguientes normas.



```
P4UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 6.2 /usr/bin/set_ipables.sh
iptables -F
iptables -t nat -F
iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.0.0/24 -j MASQUERADE
iptables -A FORWARD -i enp0s8 -o enp0s3 -s 192.168.0.0/24 -j ACCEPT
iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Imagen ilustrativa

A continuación dejo la misma captura pero con la explicación de cada línea del archivo para que así quede mas claro los parámetros usados:



```
GNU nano 6.2 set_ipables.sh
iptables -F
iptables -t nat -F #Eliminar las reglas en la tabla NAT

iptables -t nat -A POSTROUTING -o enp0s3 -s 192.168.0.0/24 -j MASQUERADE
#Esta regla se ejecutará tras el reenvio de paquetes
#hay que decir a que interfaz de red va a afectar
# -s para asignar la subred
#-J para hacer el nating

iptables -A FORWARD -i enp0s8 -o enp0s3 -s 192.168.0.0/24 -j ACCEPT
#-J: Para que la ip del cliente pueda acceder a internet

iptables -A FORWARD -i enp0s3 -o enp0s8 -m state --state RELATED,ESTABLISHED -j ACCEPT
#--state: aquellos estados de la red que vamos a aceptar
```

Imagen ilustrativa

Luego lanzamos este sh con el comando “sudo /usr/bin/set\_ipables.sh” y posteriormente hacemos el comando “sudo iptables -L” para comprobar que esta configuración esta correcta:

```

marcos@marcos:~$ sudo /usr/bin/set_iptables.sh
marcos@marcos:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination
ACCEPT     all  --  192.168.0.0/24          anywhere
ACCEPT     all  --  anywhere               anywhere             state RELATED,ESTABLISHED

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
marcos@marcos:~$

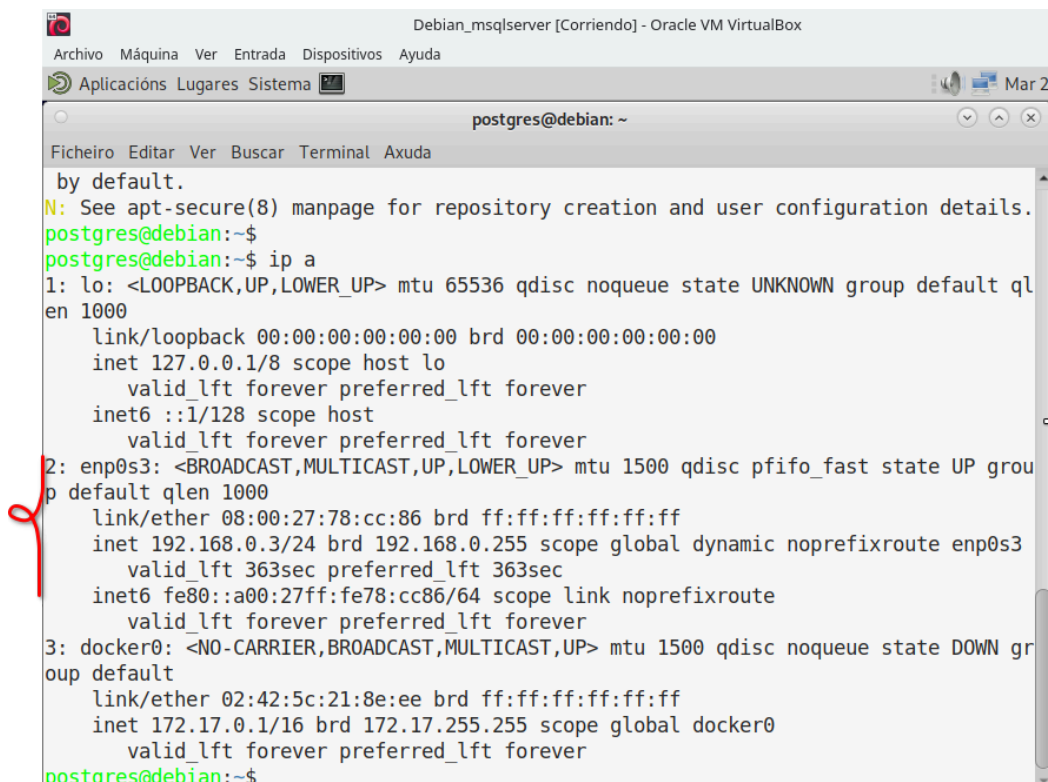
```

Imagen ilustrativa

### **- Utilizando un cliente con una sola interfaz en red interna:**

#### **- Comprobar que se asigna la IP dentro de la subred suministrada**

Para esto no metemos en el cliente en red interna y ponemos "ip a" y podemos comprobar que si se le asigna una ip:



```

Debian_msqlserver [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Aplicaciones Lugares Sistema Mar 2
postgres@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
by default.
N: See apt-secure(8) manpage for repository creation and user configuration details.
postgres@debian:~$
postgres@debian:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default ql
en 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: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP grou
p default qlen 1000
    link/ether 08:00:27:78:cc:86 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.3/24 brd 192.168.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 363sec preferred_lft 363sec
    inet6 fe80::a00:27ff:fe78:cc86/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN gr
oup default
    link/ether 02:42:5c:21:8e:ee brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
postgres@debian:~$

```

Imagen ilustrativa

#### **- Comprobar que se puede hacer ping a ips externas**

Para realizar este paso primero hago dig a chatgpt.com para asi saber la ip y hacer un ping, como se puede ver a continuación:

```

Debian_msqlserver [Corriendo] - Oracle VM VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
Aplicaciónes Lugares Sistema
postgres@debian: ~
Ficheiro Editar Ver Buscar Terminal Axuda
; chatgpt.com.                IN      A

;; ANSWER SECTION:
chatgpt.com.                 300     IN      A       104.18.32.47
chatgpt.com.                 300     IN      A       172.64.155.209

;; Query time: 68 msec
;; SERVER: 192.168.0.1#53(192.168.0.1)
;; WHEN: Tue Nov 26 16:47:06 CET 2024
;; MSG SIZE rcvd: 100

postgres@debian:~$
postgres@debian:~$ ping 104.18.32.47
PING 104.18.32.47 (104.18.32.47) 56(84) bytes of data:
64 bytes from 104.18.32.47: icmp_seq=1 ttl=61 time=18.0 ms
64 bytes from 104.18.32.47: icmp_seq=2 ttl=61 time=17.9 ms
64 bytes from 104.18.32.47: icmp_seq=3 ttl=61 time=17.4 ms
64 bytes from 104.18.32.47: icmp_seq=4 ttl=61 time=18.3 ms
64 bytes from 104.18.32.47: icmp_seq=5 ttl=61 time=17.6 ms
^C
--- 104.18.32.47 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 17.373/17.812/18.290/0.316 ms
postgres@debian:~$

```

Imagen ilustrativa

## - Comprobar que se puede hacer ping y navegar a urls externas

Luego nos metemos a navegar por urls externas como la de esemtia:

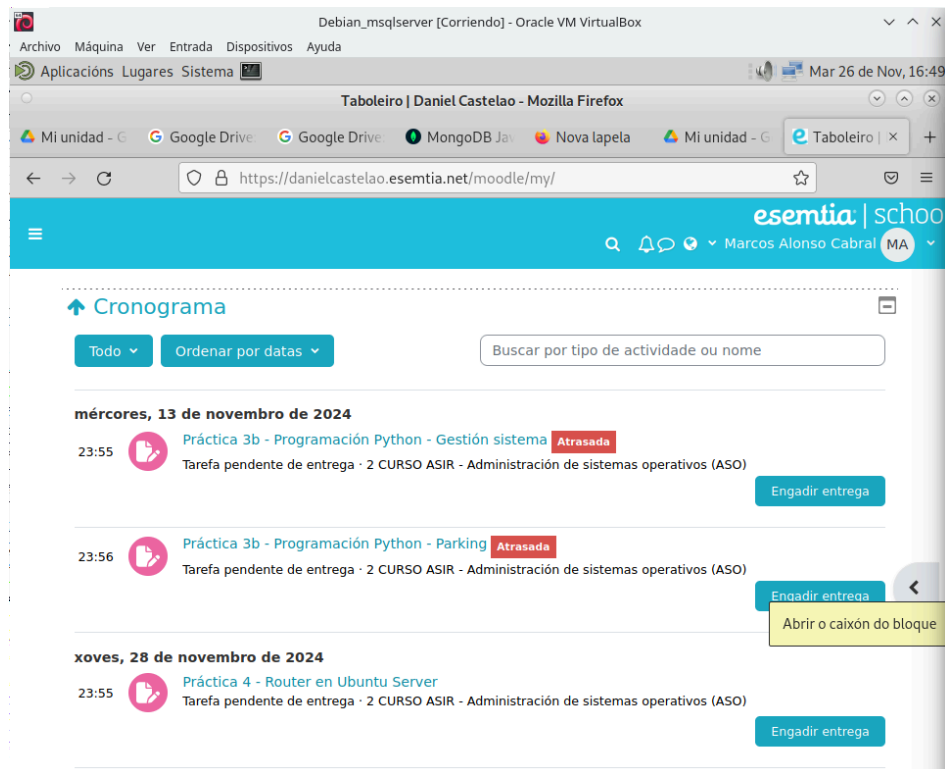


Imagen ilustrativa

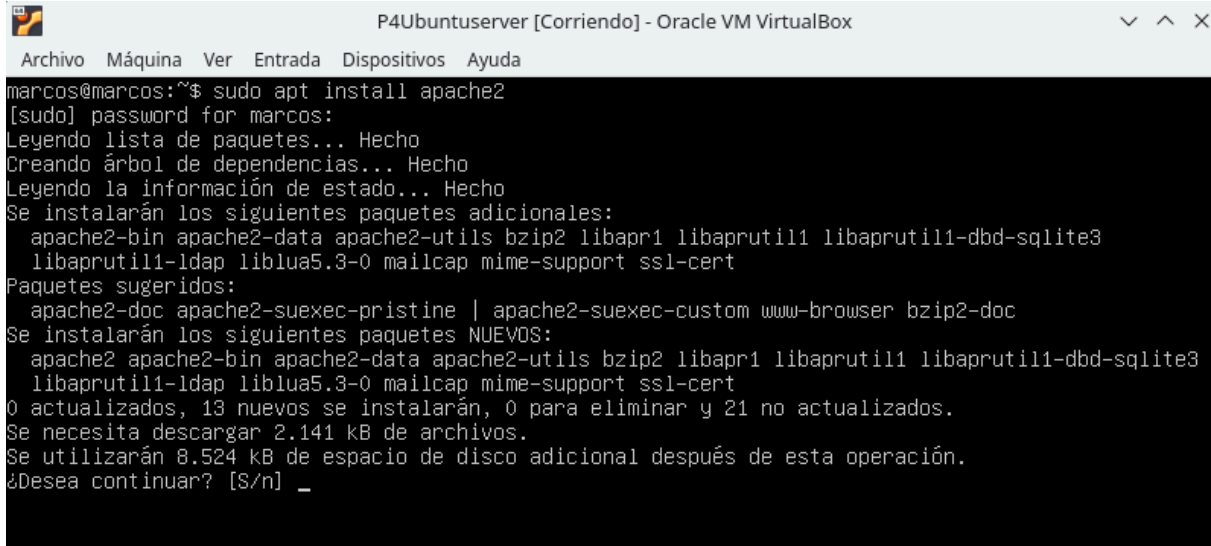
Y podemos hacer ping a urls externas como por ejemplo chatgpt.com como vimos anteriormente:

```
postgres@debian:~$ ping chatgpt.com
PING chatgpt.com (172.64.155.209) 56(84) bytes of data.
64 bytes from 172.64.155.209 (172.64.155.209): icmp_seq=1 ttl=61 time=13.7 ms
64 bytes from 172.64.155.209 (172.64.155.209): icmp_seq=2 ttl=61 time=14.8 ms
64 bytes from 172.64.155.209 (172.64.155.209): icmp_seq=3 ttl=61 time=14.4 ms
64 bytes from 172.64.155.209 (172.64.155.209): icmp_seq=4 ttl=61 time=13.9 ms
64 bytes from 172.64.155.209 (172.64.155.209): icmp_seq=5 ttl=61 time=14.2 ms
^C
--- chatgpt.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 13.679/14.189/14.789/0.387 ms
postgres@debian:~$
```

Imagen ilustrativa

### **- Comprobar que se puede acceder a un servicio interno (Por ejemplo, un servidor web Apache)**

Para instalar el servicio de apache2 en el servidor de la red interna debemos primero que todo hacer un “sudo apt update” y una vez actualizados los repositorios hacemos un “sudo apt install apache2” como podemos ver en la imagen:



```
P4UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
marcos@marcos:~$ sudo apt install apache2
[sudo] password for marcos:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Paquetes sugeridos:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
Se instalarán los siguientes paquetes NUEVOS:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 actualizados, 13 nuevos se instalarán, 0 para eliminar y 21 no actualizados.
Se necesita descargar 2.141 kB de archivos.
Se utilizarán 8.524 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] _
```

Imagen ilustrativa

Para verificar que el servidor que acabamos de instalar esta en ejecucion ponemos el comando “sudo systemctl status apache2”:

```

marcos@marcos:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2024-11-26 15:54:24 UTC; 23s ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 6107 (apache2)
    Tasks: 55 (limit: 4564)
   Memory: 5.0M
      CPU: 15ms
   CGroup: /system.slice/apache2.service
           └─6107 /usr/sbin/apache2 -k start
             └─6109 /usr/sbin/apache2 -k start
               └─6110 /usr/sbin/apache2 -k start

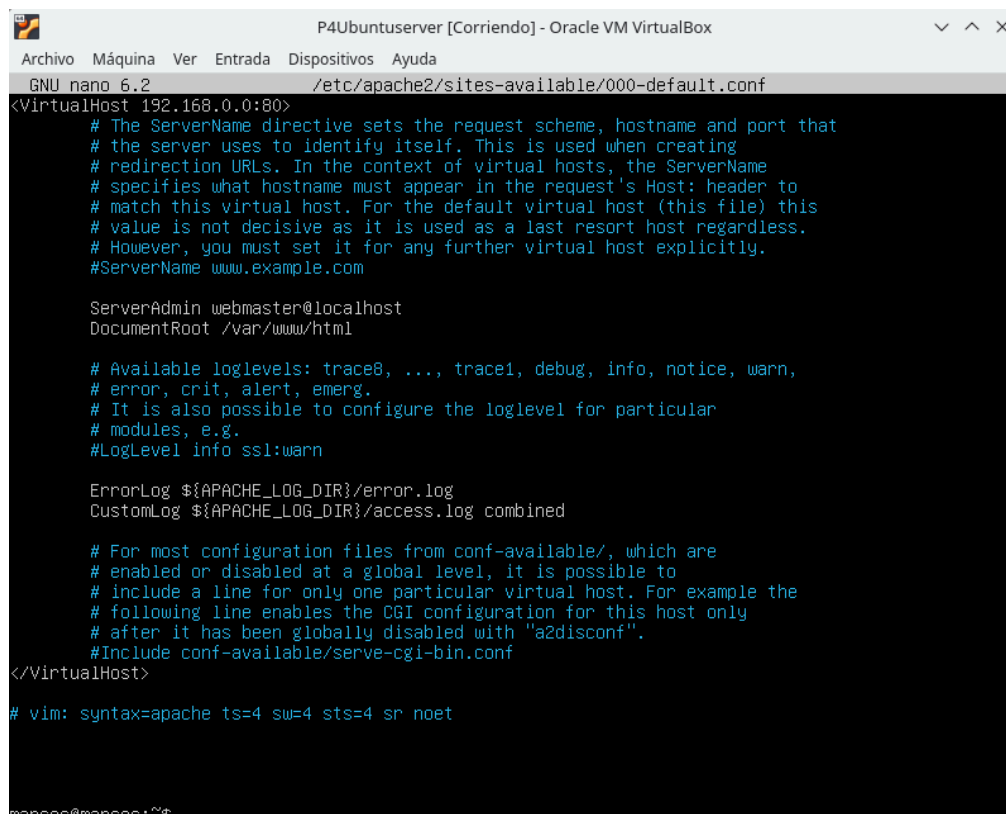
nov 26 15:54:24 marcos systemd[1]: Starting The Apache HTTP Server...
nov 26 15:54:24 marcos apachectl[6105]: AH00558: apache2: Could not reliably determine the server's s
nov 26 15:54:24 marcos systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)

```

Imagen ilustrativa

Si no está activo podemos iniciarlo poniendo el comando “sudo systemctl start apache2”, aunque en mi caso ya se inició al instalarlo.

Y por último para que solo se pueda acceder mediante la red interna, ponemos el comando “sudo nano /etc/apache2/sites-available/000-default.conf” y en esta configuración de apache cambiamos la primera línea y en vez de un “\*” ponemos la “ip de la red del servidor” que en mi caso es la 192.168.0.0:



```

P4UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 6.2 /etc/apache2/sites-available/000-default.conf
<VirtualHost 192.168.0.0:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
#ServerName www.example.com

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
marcos@marcos:~$

```

Imagen ilustrativa

Con esta configuración de apache, el servidor principal solo suministra este servicio a la red interna.

Una vez modificado y guardado este fichero, debemos poner el comando “sudo apache2ctl configtest” para comprobar que no tiene ningun error y luego hacemos un “sudo systemctl restart apache2” para reiniciar el servicio y que se aplique lo que acabamos de modificar:

```
marcos@marcos:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1.1. Set the 'ServerName' directive globally to suppress this message
[Tue Nov 26 16:30:30.327861 2024] [core:error] [pid 6552:tid 140221782157184] (EAI 2)Name or service not known: AH00549: Failed to resolve server name for 192.168.0.0 (check DNS) -- or specify an explicit ServerName
Syntax OK
marcos@marcos:~$ sudo systemctl restart apache2
```

Imagen ilustrativa

Luego entramos en el cliente y ponemos la ip del servidor que es la 192.168.0.1 en mi caso. Y como podemos ver en esta última captura, el sistema muestra la página de apache que debería mostrar por defecto :



Imagen ilustrativa