

A thick dark blue vertical bar runs down the left side of the page. A blue arrow-shaped banner points to the right from this bar, containing the text 'DESPLIEGUE DE APLICACIONES WEB'. In the bottom left corner, there are several thin, curved, light blue lines that sweep upwards and to the right.

DESPLIEGUE DE APLICACIONES WEB

CONFIGURACIÓN DE UN SERVIDOR DNS

AUTOR: Alberto Martínez Pérez

CURSO: 2º CFGS Desarrollo de Aplicaciones Web (DAW)

MÓDULO: Despliegue de Aplicaciones Web

FECHA DE ENTREGA: 21 de febrero de 2024

ÍNDICE

INTRODUCCIÓN	3
INSTALACIÓN DEL SERVIDOR DNS	3
CONFIGURACIÓN DEL SERVIDOR MAESTRO.....	6
A. CONFIGURACIÓN DE named.conf.options	8
B. CONFIGURACIÓN DE named.conf.local	11
C. CREACIÓN DEL ARCHIVO DE ZONA	12
D. AÑADIR UN REGISTRO DE RECURSO CNAME.....	15
COMPROBACIÓN DE ERRORES (JOURNALCTL)	16
COMPROBACIÓN DEL FUNCIONAMIENTO DEL DNS DESDE UN CLIENTE.....	17
USO DEL SERVIDOR DNS PARA RESOLVER SOLICITUDES A UN SERVIDOR WEB	18
CREACIÓN DE UN DNS DELEGADO QUE GESTIONE UNA DELEGACIÓN DE ZONA	22
CREACIÓN DE UN SERVIDOR DNS ESCLAVO	25

INTRODUCCIÓN

En esta práctica vamos a realizar una instalación y configuración de un servidor DNS sobre máquinas virtuales con el SO Debian 12. Vamos a crear un servidor DNS primario o maestro y un servidor DNS secundario o esclavo y realizaremos solicitudes de resolución de dominio sobre ellos.

El servidor de DNS que vamos a utilizar es Bind (*Berkeley Internet Name Domain*) ya que es el estándar de facto para servidores de este tipo, concretamente usaremos la versión Bind9

INSTALACIÓN DEL SERVIDOR DNS

Para llevar a cabo la instalación de Bind9 debemos hacer primero una actualización de los repositorios de Linux de nuestro sistema, para ello utilizaremos los comandos:

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
albertom-servidor@debian-deaw:~$ sudo apt-get update
Obj:1 http://security.debian.org/debian-security bookworm-security InRelease
Obj:2 http://packages.microsoft.com/repos/code stable InRelease
Obj:3 http://deb.debian.org/debian bookworm InRelease
Obj:4 http://deb.debian.org/debian bookworm-updates InRelease
Leyendo lista de paquetes... Hecho
albertom-servidor@debian-deaw:~$ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los siguientes paquetes se han retenido:
  linux-headers-amd64 linux-image-amd64
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
```

Una vez hecho esto para instalar bind9 usamos el comando:

```
sudo apt-get install bind9 bind9utils bind9-doc
```

```
albertom-servidor@debian-deaw:~$ sudo apt-get 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
Paquetes sugeridos:
  bind-doc resolvconf ufw
Se instalarán los siguientes paquetes NUEVOS:
  bind9 bind9-doc bind9-utils bind9utils
0 actualizados, 4 nuevos se instalarán, 0 para eliminar y 2 no actualizados.
Se necesita descargar 4.601 kB de archivos.
Se utilizarán 9.931 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
```

Esto habrá instalado el servidor DNS Bind y, además habrá creado el usuario y grupo necesarios para el uso del servidor, ambos de nombre bind:

```
Añadiendo el grupo `bind' (GID 123) ...
Hecho.
Añadiendo el usuario del sistema `bind' (UID 115) ...
Añadiendo un nuevo usuario `bind' (UID 115) con grupo `bind' ...
```

Podemos comprobar el estado del servicio haciendo uso de systemctl sobre el servicio de nombre named.service:

```
sudo systemctl status named.service
```

```
albertom-servidor@debian-deaw:~$ sudo systemctl status named.service
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-01-26 16:32:32 CET; 2min 32s ago
     Docs: man:named(8)
  Main PID: 5002 (named)
    Status: "running"
     Tasks: 6 (limit: 2284)
    Memory: 16.2M
       CPU: 61ms
    CGroup: /system.slice/named.service
            └─5002 /usr/sbin/named -f -u bind
```

Además, podemos ver que ahora tenemos el puerto :53 disponible para escucha, este puerto es el de por defecto para DNS. Para ello usamos el comando netstat:

```
sudo netstat -ant | grep :53
```

```
albertom-servidor@debian-deaw:~$ sudo netstat -ant | grep :53
tcp        0      0 127.0.0.1:53          0.0.0.0:*              LISTEN
tcp        0      0 192.168.1.127:53      0.0.0.0:*              LISTEN
tcp        0      0 192.168.1.127:53      0.0.0.0:*              LISTEN
tcp6       0      0 fe80::a00:27ff:fe81::53 :::*                   LISTEN
tcp6       0      0 fe80::a00:27ff:fe81::53 :::*                   LISTEN
tcp6       0      0 :::1:53               :::*                   LISTEN
```

Todos los archivos de configuración y de zona que utilizará Bind se encuentran en un directorio que se ha creado en la instalación y que se encuentra en /etc/bind:

```
albertom-servidor@debian-deaw:~$ sudo ls -la /etc/bind
total 64
drwxr-sr-x  2 root bind  4096 ene 26 16:32 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind  458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind  498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind  165 sep 21 19:33 named.conf.local
-rw-r--r--  1 root bind  846 sep 21 19:33 named.conf.options
-rw-r----- 1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root 1317 sep 21 19:33 zones.rfc1918
```

Como sólo vamos a utilizar IPv4, podemos informar de esto a Bind en su archivo general de configuración que se llama named y se encuentra en /etc/default

```
albertom-servidor@debian-deaw:~$ sudo ls -la /etc/default
total 100
drwxr-xr-x  3 root root  4096 ene 26 16:32 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root   830 ene 10  2023 anacron
-rw-r--r--  1 root root   219 abr 19  2023 avahi-daemon
-rw-r--r--  1 root root   846 nov 16  2022 bluetooth
-rw-r--r--  1 root root   285 nov  1 10:58 console-setup
-rw-r--r--  1 root root   955 jul 17  2022 cron
-rw-r--r--  1 root root   297 sep 16 12:03 dbus
-rw-r--r--  1 root root  1492 nov  1 11:14 grub
drwxr-xr-x  2 root root  4096 nov  1 11:14 grub.d
-rw-r--r--  1 root root    81 feb 13  2023 hwclock
-rw-r--r--  1 root root  1752 mar 20  2023 im-config
-rw-r--r--  1 root root  1029 ago  8 22:21 intel-microcode
-rw-r--r--  1 root root   150 nov  1 10:58 keyboard
-rw-r--r--  1 root root    54 nov  1 10:57 locale
-rw-r--r--  1 root root    86 sep 21 19:33 named
-rw-r--r--  1 root root  1032 sep 14  2021 networking
-rw-r--r--  1 root root  1756 jul  6  2023 nss
-rw-r--r--  1 root root   291 jun 21  2019 numlockx
-rw-r--r--  1 root root    13 mar 28  2020 ofono
-rw-r--r--  1 root root   110 nov  1 11:13 saned
-rw-r--r--  1 root root   133 sep 24 00:11 ssh
-rw-r--r--  1 root root  1117 nov 11  2022 useradd
```

Modificamos el contenido del archivo cambiando la siguiente línea:

```
GNU nano 7.2 /etc/default/named *
#
# run resolvconf?
RESOLVCONF=no

# startup options for the server
OPTIONS="-u bind -4"
```

Por su parte el archivo de configuración principal de Bind recibe el nombre de named.conf y se encuentra en el directorio /etc/bind:

```
albertom-servidor@debian-deaw:~$ sudo ls -la /etc/bind
total 64
drwxr-sr-x  2 root bind  4096 ene 26 16:32 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   165 sep 21 19:33 named.conf.local
-rw-r--r--  1 root bind   846 sep 21 19:33 named.conf.options
-rw-r----- 1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```

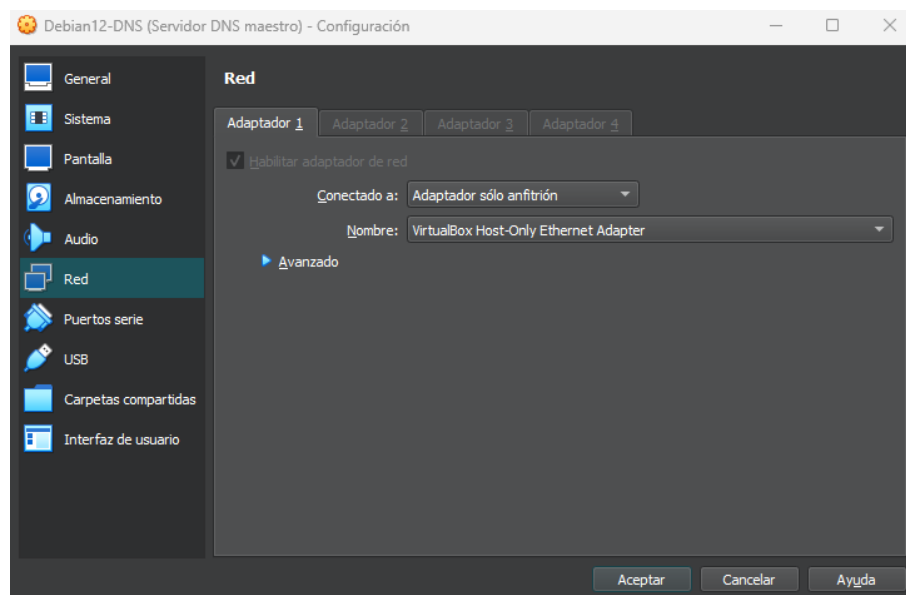
El contenido de este fichero se puede ver en la siguiente imagen y su función es simplemente agrupar los archivos de configuración principal, es decir, los que vamos a utilizar en los siguientes pasos de la práctica.

```
GNU nano 7.2 /etc/bind/named.conf
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

CONFIGURACIÓN DEL SERVIDOR MAESTRO

Lo primero que debemos hacer es establecer la conexión de la máquina virtual de nuestro servidor DNS en “adaptador sólo anfitrión”:



El siguiente paso será conocer nuestra IP asignada con el comando ip a:

```
albertom-servidor@debian-deaw:~$ 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: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen
1000
    link/ether 08:00:27:81:c3:45 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.108/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe81:c345/64 scope link
        valid_lft forever preferred_lft forever
```

Para hacer que esta IP sea fija y nunca cambie debemos editar el archivo `/etc/network/interfaces`:

```
GNU nano 7.2 /etc/network/interfaces *
# 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

allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.56.108
    netmask 255.255.255.0
    gateway 192.168.56.1
    dns-nameservers 8.8.8.8
```

Lo que significa cada una de las líneas es:

- **allow-hotplug enp0s3**: indica que la interfaz de red enp0s3 se gestionará automáticamente cuando se detecte que el cable de red está enchufado o cuando la interfaz de red se activa de alguna otra manera. La palabra clave **allow-hotplug**: permite que la interfaz se active o desactive automáticamente en función de si está conectada o desconectada.
- **iface enp0s3 inet static**: define la configuración para la interfaz de red enp0s3 y especifica que se utilizará una configuración de red estática (inet static). En una configuración estática, la dirección IP, la máscara de red, la puerta de enlace y otros parámetros se configuran manualmente.
- **address**: especifica la dirección IP estática que se asignará a la interfaz enp0s3.
- **netmask**: establece la máscara de red para la interfaz enp0s3.
- **gateway**: especifica la puerta de enlace predeterminada para la interfaz enp0s3. La puerta de enlace es la dirección IP del router o dispositivo de red que se utiliza para enrutar el tráfico fuera de la red local.
- **dns-nameservers**: establece la dirección IP del servidor DNS que se utilizará para resolver nombres de dominio.

A continuación, restablecemos el servicio de conexión:

```
albertom-servidor@debian-deaw:~$ sudo systemctl restart networking.service
albertom-servidor@debian-deaw:~$ sudo systemctl status networking.service
● networking.service - Raise network interfaces
   Loaded: loaded (/lib/systemd/system/networking.service; enabled; preset: enabled)
   Active: active (exited) since Fri 2024-01-26 17:13:03 CET; 5s ago
     Docs: man:interfaces(5)
  Process: 5334 ExecStart=/sbin/ifup -a --read-environment (code=exited, status=0/SUCCESS)
  Process: 5348 ExecStart=/bin/sh -c if [ -f /run/network/restart-hotplug ]; then /sbin/ifu
 Main PID: 5348 (code=exited, status=1/FAILURE)
    Tasks: 0 (limit: 2284)
   Memory: 880.0K
      CPU: 127ms
   CGroup: /system.slice/networking.service
```

Podría darse el caso de que después de este reinicio obtuviéramos dos IPs diferentes como se puede ver en la siguiente imagen:

```
albertom-servidor@debian-deaw:~$ 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: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f0:a0:92 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.108/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.56.109/24 brd 192.168.56.255 scope global secondary enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0:a092/64 scope link
        valid_lft forever preferred_lft forever
```

Para eliminar la IP que no necesitamos utilizamos el siguiente comando:

sudo ip a del IP_a_eliminar dev interfaz_de_red

```
albertom-servidor@debian-deaw:~$ sudo ip a del 192.168.56.109/24 dev enp0s3
```

De esta forma se eliminará la IP del listado de IPs, si volvemos a realizar un ip a no aparecerá esta IP:

```
albertom-servidor@debian-deaw:~$ 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: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:f0:a0:92 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.108/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe0:a092/64 scope link
        valid_lft forever preferred_lft forever
```

A. CONFIGURACIÓN DE named.conf.options

El primer fichero que vamos a configurar es el fichero named.conf.options cuya función es establecer opciones globales que afectan el comportamiento general del servidor Bind. Se encuentra en el directorio /etc/bind:

```
albertom-servidor@debian-deaw:~$ sudo ls -la /etc/bind
total 64
drwxr-xr-x  2 root bind  4096 ene 26 16:53 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   165 sep 21 19:33 named.conf.local
-rw-r--r--  1 root bind   846 sep 21 19:33 named.conf.options
-rw-r-----  1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```


Como medida de buena práctica siempre es conveniente realizar un backup del fichero de configuración para poder volver rápidamente a un estado anterior a los cambios, para ello usamos el comando:

```
sudo cp ./named.conf.options ./named.conf.options.backup
```

```
albertom-servidor@debian-deaw:/etc/bind$ sudo cp ./named.conf.options ./named.conf.options.backup
```

Esto habrá creado una copia del fichero original con la terminación backup:

```
albertom-servidor@debian-deaw:/etc/bind$ sudo ls -la
total 68
drwxr-sr-x  2 root bind  4096 ene 26 16:58 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   165 sep 21 19:33 named.conf.local
-rw-r--r--  1 root bind   846 sep 21 19:33 named.conf.options
-rw-r--r--  1 root bind   846 ene 26 16:58 named.conf.options.backup
-rw-r----- 1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```

Vamos a editar este fichero de configuración para que:

- Por motivos de seguridad sólo puedan hacer consultas recursivas al servidor aquellos host que nosotros decidamos o que consideremos confiables. En nuestro caso serán todos aquellos hosts de la red 192.168.56.0/24.

Esto debe ir escrito en un bloque acl antes del bloque options:

```
acl confiables {
    192.168.56.0/24;
};
```

- Que sólo se permitan las consultas recursivas a los hosts que hemos decidido en la lista de acceso anterior. Esta será la orden: allow-recursion {confiables;}.
- No permitir transferencia de zonas a nadie. Esta será la orden: allow-transfer {none;}.
- Configurar el servidor para que escuche consultas DNS en el puerto 53 y en la IP de su interfaz de la red privada (la IP de la máquina virtual). Esta será la orden: listen-on port 53{192.168.56.108;}.
- Permitir las consultas recursivas. Esta será la orden: recursion yes.

Las cuatro reglas anteriores deberán aparecer por dentro del bloque options:

```
allow-recursion {confiables;;}
allow-transfer {none;;}
listen-on port 53{192.168.56.108;;}
recursion yes;
```

Además, debemos borrar las dos líneas que hacen referencia al dnssec-validation y a la escucha de IPv6.

```
dnssec-validation auto;
```

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

Por lo cual el aspecto final de nuestro fichero de configuración será el siguiente:

```
GNU nano 7.2 /etc/bind/named.conf.options *
acl confiables {
    192.168.56.0/24;
};
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;
    // };
    allow-recursion {confiables;;}
    allow-transfer {none;;}
    listen-on port 53{192.168.56.108;;}
    recursion yes;

    //=====
    // 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
    //=====
};
```

Para comprobar si la configuración es correcta utilizamos el comando:

```
sudo named-checkconf
```

```
albertom-servidor@debian-deaw:/etc/bind$ sudo named-checkconf
albertom-servidor@debian-deaw:/etc/bind$
```

Si el comando no nos responde significará que no existen errores en la configuración, en el caso de que hubiera algún error recibiríamos un mensaje como el siguiente:

```
albertom-servidor@debian-deaw:/etc/bind$ sudo named-checkconf
/etc/bind/named.conf.options:4: missing ';' before 'options'
```

Por último, debemos reiniciar el servicio y comprobar su estado con los comandos correspondientes de systemctl:

```
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl restart named.service
albertom-servidor@debian-deaw:/etc/bind$ systemctl status named.service
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-01-26 17:26:30 CET; 2s ago
     Docs: man:named(8)
  Main PID: 5492 (named)
    Status: "running"
     Tasks: 4 (limit: 2284)
  Memory: 32.9M
     CPU: 124ms
   CGroup: /system.slice/named.service
           └─5492 /usr/sbin/named -f -u _bind -4
```

Como hemos podido ver este archivo en el bloque options nos informaba del directorio que funcionará como caché, es decir, por defecto el servidor ya tiene una configuración para ser DNS caché. El directorio donde se cachearán las zonas es /var/cache/bind:

```
albertom-servidor@debian-deaw:~$ sudo ls -la /var/cache/bind
total 16
drwxrwxr-x  2 root bind 4096 ene 26 17:27 .
drwxr-xr-x 21 root root 4096 ene 26 16:32 ..
-rw-r--r--  1 bind bind  821 ene 26 17:27 managed-keys.bind
-rw-r--r--  1 bind bind 3061 ene 26 17:26 managed-keys.bind.jnl
```

Además, ahora si hacemos uso de la herramienta netstat podremos comprobar que sólo se produce escucha de por DNS a través de los IPv4:

sudo netstat -an | grep :53

```
albertom-servidor@debian-deaw:~$ sudo netstat -an | grep :53
tcp        0      0 192.168.56.108:53 0.0.0.0:*        LISTEN
tcp        0      0 192.168.56.108:53 0.0.0.0:*        LISTEN
tcp        0      0 127.0.0.1:53      0.0.0.0:*        LISTEN
tcp6       0      0 :::1:53           :::*             LISTEN
udp        0      0 192.168.56.108:53 0.0.0.0:*
udp        0      0 192.168.56.108:53 0.0.0.0:*
udp        0      0 127.0.0.1:53      0.0.0.0:*
udp        0      0 0.0.0.0:5353      0.0.0.0:*
udp6       0      0 :::1:53           :::*
udp6       0      0 :::5353           :::*
```

B. CONFIGURACIÓN DE **named.conf.local**

Ahora vamos a configurar el fichero named.conf.local, este fichero se encarga de configurar la información específica del dominio y las zonas de nombres locales en un servidor DNS.

Al igual que antes es una buena práctica el realizar un backup del fichero:

sudo cp ./named.conf.local ./named.conf.local.backup

```
albertom-servidor@debian-deaw:/etc/bind$ sudo cp ./named.conf.local ./named.conf.local.backup
```

```
albertom-servidor@debian-deaw:/etc/bind$ sudo ls -la
total 72
drwxr-sr-x  2 root bind  4096 ene 26 17:36 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   165 sep 21 19:33 named.conf.local
-rw-r--r--  1 root bind   165 ene 26 17:36 named.conf.local.backup
-rw-r--r--  1 root bind   994 ene 26 17:24 named.conf.options
-rw-r--r--  1 root bind   846 ene 26 16:58 named.conf.options.backup
-rw-r----- 1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```

Vamos a modificar el contenido del fichero generando una zona de nombre deaw.es. Además, indicaremos que el DNS es maestro de la zona y dónde se encuentra el archivo de zona que se creará en el siguiente paso de la práctica:

```
GNU nano 7.2                                ./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 "deaw.es" {
    type master;
    file "/etc/bind/db.deaw.es";
};
```

Una vez hecho esto guardamos los cambios y hacemos un chequeo de la configuración, así como el reinicio del servicio:

```
albertom-servidor@debian-deaw:/etc/bind$ sudo named-checkconf
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl restart named.service
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl status named.service
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-01-26 17:56:14 CET; 5s ago
     Docs: man:named(8)
  Main PID: 5692 (named)
    Status: "running"
   Tasks: 4 (limit: 2284)
  Memory: 29.4M
     CPU: 48ms
   CGroup: /system.slice/named.service
           └─5692 /usr/sbin/named -f -u bind -4
```

C. CREACIÓN DEL ARCHIVO DE ZONA

En este paso vamos a crear el archivo de zona al que hemos hecho referencia en el paso previo. Este fichero nos servirá, por ejemplo, para la resolución directa de nombre.

Estos ficheros tienen una estructura específica, por ejemplo:

```
albertom-servidor@debian-deaw:/etc/bind$ cat ./db.local
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2      ; Serial
                        604800  ; Refresh
                        86400   ; Retry
                        2419200 ; Expire
                        604800 ) ; Negative Cache TTL
;
@         IN      NS       localhost.
@         IN      A        127.0.0.1
@         IN      AAAA     ::1
```

Por lo que, para mantener esta estructura, vamos a hacer una copia del fichero con el nuevo nombre:

```
sudo cp ./db.local ./db.deaw.es
```

```
albertom-servidor@debian-deaw:/etc/bind$ sudo cp ./db.local ./db.deaw.es
albertom-servidor@debian-deaw:/etc/bind$ ls -la
total 76
drwxr-xr-x  2 root bind  4096 ene 26 17:44 .
drwxr-xr-x 136 root root 12288 ene 26 16:32 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root bind   270 ene 26 17:44 db.deaw.es
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   229 ene 26 17:39 named.conf.local
-rw-r--r--  1 root bind   165 ene 26 17:36 named.conf.local.backup
-rw-r--r--  1 root bind   994 ene 26 17:24 named.conf.options
-rw-r--r--  1 root bind   846 ene 26 16:58 named.conf.options.backup
-rw-r--r--  1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```

Y modificamos el contenido para que quede así:

```
GNU nano 7.2 /etc/bind/db.deaw.es *
$TTL      604800
@         IN      SOA      debian-deaw.deaw.es. admin.deaw.es. (
                        2024012601 ; Serial
                        3600      ; Refresh
                        1800      ; Retry
                        604800    ; Expire
                        86400)    ; Minimum TTL

deaw.es.   IN      NS       debian-deaw.deaw.es.
debian-deaw IN      A        192.168.56.108
```

De esta forma nos queda:

- **TTL:** Este valor especifica la duración máxima de tiempo que un registro DNS puede permanecer en caché antes de que se considere obsoleto. En este caso, 604800 segundos (7 días).
- **SOA:** Es el encabezado principal de una zona DNS y contiene información importante sobre la zona, como el dominio primario, la dirección de correo electrónico del administrador, y parámetros de tiempo (como TTL, Refresh, Retry, etc.). En la primera línea debe

aparecer el dominio primario (debian-deaw.deaw.es) y la dirección de correo del administrador (admin.deaw.es.).

- **Serial:** Número de versión del archivo de zona. Debe incrementarse cada vez que se realicen cambios.
- **Refresh:** Intervalo de tiempo en segundos antes de que un servidor secundario intente actualizar la zona desde el servidor primario.
- **Retry:** Intervalo de tiempo en segundos que un servidor secundario debe esperar antes de intentar nuevamente después de un fallo de actualización.
- **Expire:** Tiempo máximo en segundos que un servidor secundario puede esperar antes de considerar que los datos de la zona están obsoletos.
- **Minimum TTL:** El TTL predeterminado para los registros en la zona.
- **NS:** Se utiliza para especificar los servidores de nombres (name servers) autoritativos para el dominio.
- **A:** Se utiliza para asociar un nombre de dominio con una dirección IPv4.

Una vez finalizada la creación se realiza al igual que antes un chequeo de la configuración y un reinicio del servicio named.

```
albertom-servidor@debian-deaw:/etc/bind$ sudo named-checkconf
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl restart named.service
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl status named.service
• named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-01-26 17:56:14 CET; 5s ago
     Docs: man:named(8)
  Main PID: 5692 (named)
    Status: "running"
     Tasks: 4 (limit: 2284)
  Memory: 29.4M
     CPU: 48ms
   CGroup: /system.slice/named.service
           └─5692 /usr/sbin/named -f -u bind -4
```

Vamos a comprobar el funcionamiento correcto del DNS haciendo uso del comando nslookup para comprobar que el DNS es capaz de realizar una solicitud de resolución de nombres sobre el dominio debian-deaw.deaw.es:

```
albertom-servidor@debian-deaw:~$ nslookup
> server 192.168.56.108
Default server: 192.168.56.108
Address: 192.168.56.108#53
> debian-deaw.deaw.es
Server:      192.168.56.108
Address:     192.168.56.108#53
Name:   debian-deaw.deaw.es
Address: 192.168.56.108
> exit
```

Obtenemos la resolución del nombre.

Asignamos el servidor DNS que queremos que responda.

Preguntamos el dominio.

D. AÑADIR UN REGISTRO DE RECURSO CNAME

Vamos a añadir ahora a nuestro fichero de zona un registro de recurso de tipo CNAME, es decir, un alias para que también se pueda entrar a nuestro servidor desde él.

Para ello debemos actualizar el fichero db.deaw.es y añadir una nueva línea:

```
GNU nano 7.2                               ./db.deaw.es *
```

\$TTL	604800		
@	IN	SOA	debian-deaw.deaw.es. admin.deaw.es. (
			2024012601 ; Serial
			3600 ; Refresh
			1800 ; Retry
			604800 ; Expire
			86400) ; Minimum TTL
deaw.es.	IN	NS	debian-deaw.deaw.es.
debian-deaw	IN	A	192.168.56.108
pokemon	IN	CNAME	debian-deaw

Reiniciamos el servicio y comprobamos el estado:

```
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl restart named.service
albertom-servidor@debian-deaw:/etc/bind$ sudo systemctl status named.service
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-01-29 15:32:22 CET; 19s ago
     Docs: man:named(8)
  Main PID: 2961 (named)
    Status: "running"
   Tasks: 4 (limit: 2284)
  Memory: 34.9M
     CPU: 182ms
  CGroup: /system.slice/named.service
          └─2961 /usr/sbin/named -f -u bind -4
```

Una vez hecho esto vamos a comprobar que funciona de forma correcta, para ello vamos a volver a utilizar el comando nslookup pero en este caso preguntaremos por el nombre de dominio pokemon.deaw.es:

```
albertom-servidor@debian-deaw:/etc/bind$ nslookup
> server 192.168.56.108
Default server: 192.168.56.108
Address: 192.168.56.108#53
> pokemon.deaw.es
Server:          192.168.56.108
Address:         192.168.56.108#53

pokemon.deaw.es canonical name = debian-deaw.deaw.es.
Name:   debian-deaw.deaw.es
Address: 192.168.56.108
```


Para evitar tener que escribir continuamente la IP del servidor, vamos a modificar el archivo `/etc/resolv.conf` para añadir la IP de nuestra máquina como nameserver.

```
GNU nano 7.2 /etc/resolv.conf
nameserver 192.168.56.108
nameserver 192.168.1.1
```

De esta forma si hacemos un `nslookup` del dominio `debian-deaw.deaw.es` será el nameserver `192.168.56.108` el que nos responda y en caso de fallar respondería el siguiente, es decir el `192.168.1.1`:

```
albertom-servidor@debian-deaw:~$ nslookup debian-deaw.deaw.es
Server:          192.168.56.108
Address:         192.168.56.108#53

Name:   debian-deaw.deaw.es
Address: 192.168.56.108
```

COMPROBACIÓN DE ERRORES (JOURNALCTL)

Para ver qué información nos da el sistema cuando aparecen errores durante la configuración del DNS podemos revisar el contenido de `journalctl`, se trata de un comando que sigue la siguiente estructura:

`sudo journalctl [opciones] servicio [opciones]`

Por ejemplo:

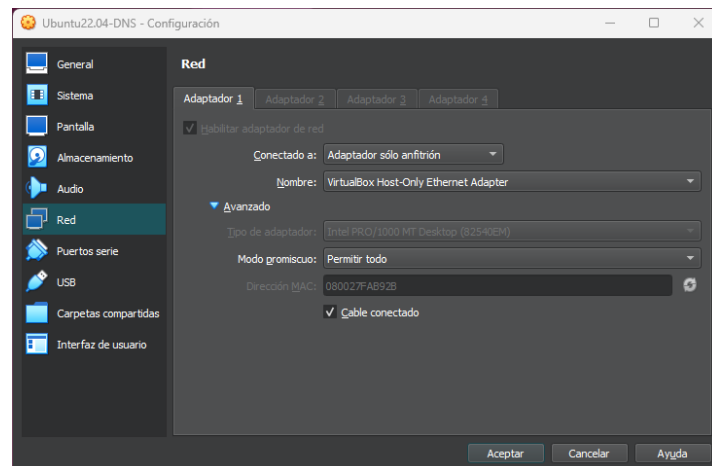
```
albertom-servidor@debian-deaw:~$ sudo journalctl -u named.service --since today
ene 30 19:06:06 debian-deaw systemd[1]: Starting named.service - BIND Domain Name S
ene 30 19:06:07 debian-deaw named[771]: starting BIND 9.18.19-1~deb12u1-Debian (Ext
ene 30 19:06:07 debian-deaw named[771]: running on Linux x86_64 6.1.0-16-amd64 #1 S
ene 30 19:06:07 debian-deaw named[771]: built with '--build=x86_64-linux-gnu' '--p
ene 30 19:06:07 debian-deaw named[771]: running as: named -f -u bind -4
ene 30 19:06:07 debian-deaw named[771]: compiled by GCC 12.2.0
ene 30 19:06:07 debian-deaw named[771]: compiled with OpenSSL version: OpenSSL 3.0.
ene 30 19:06:07 debian-deaw named[771]: linked to OpenSSL version: OpenSSL 3.0.11 1
ene 30 19:06:07 debian-deaw named[771]: compiled with libuv version: 1.44.2
ene 30 19:06:07 debian-deaw named[771]: linked to libuv version: 1.44.2
ene 30 19:06:07 debian-deaw named[771]: compiled with libxml2 version: 2.9.14
ene 30 19:06:07 debian-deaw named[771]: linked to libxml2 version: 20914
ene 30 19:06:07 debian-deaw named[771]: compiled with json-c version: 0.16
ene 30 19:06:07 debian-deaw named[771]: linked to json-c version: 0.16
ene 30 19:06:07 debian-deaw named[771]: compiled with zlib version: 1.2.13
```

Podemos filtrar resultados usando `grep`:

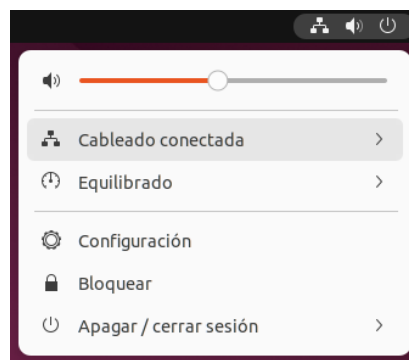
```
albertom-servidor@debian-deaw:~$ sudo journalctl -u named.service --since today | grep failure
ene 30 19:06:08 debian-deaw named[771]: managed-keys-zone: Unable to fetch DNSKEY set '.': failure
ene 30 19:06:08 debian-deaw named[771]: resolver priming query complete: failure
ene 30 19:06:08 debian-deaw named[771]: managed-keys-zone: Unable to fetch DNSKEY set '.': failure
ene 30 19:06:08 debian-deaw named[771]: managed-keys-zone: Unable to fetch DNSKEY set '.': failure
ene 30 19:06:08 debian-deaw named[771]: managed-keys-zone: Unable to fetch DNSKEY set '.': failure
```


COMPROBACIÓN DEL FUNCIONAMIENTO DEL DNS DESDE UN CLIENTE

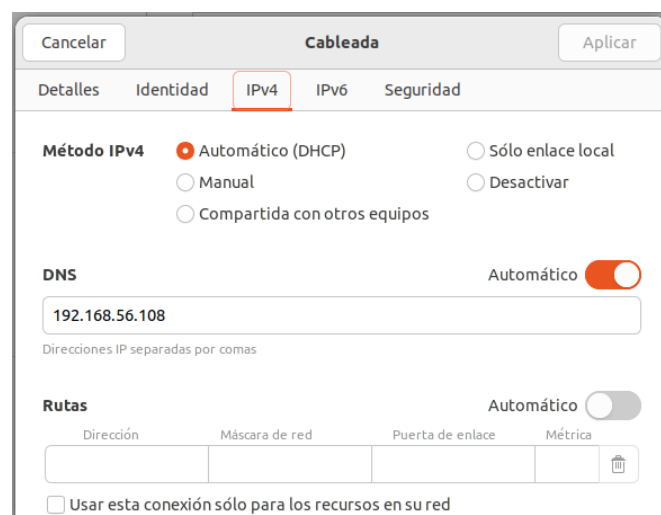
Para hacer la función de cliente vamos a usar una máquina virtual con el SO Ubuntu 22.04, lo primero que haremos es configurar la red de nuestra máquina para que también se encuentre en “adaptador sólo anfitrión”:



A continuación, debemos configurar la red cableada:



En el menú de configuración debemos establecer el DNS con la IP de nuestro servidor DNS.



Una vez hecho esto y aplicados los cambios, debemos reiniciar nuestra conexión, para ellos apagamos y encendemos el “cableado”.

También debemos modificar el archivo /etc/resolv.conf para que nuestro servidor DNS sea el que hemos configurado en pasos anteriores:

```
nameserver 192.168.56.108
nameserver 127.0.0.53
options edns0 trust-ad
search .
```

Ahora en una terminal debemos usar el comando nslookup y comprobar la resolución de nombre tanto de debian-deaw.deaw.es como de pokemon.deaw.es:

```
albertom-cliente@albertoMartinezPerez:~$ nslookup debian-deaw.deaw.es
Server:      192.168.56.108
Address:     192.168.56.108#53

Name:   debian-deaw.deaw.es
Address: 192.168.56.108

albertom-cliente@albertoMartinezPerez:~$ nslookup pokemon.deaw.es
Server:      192.168.56.108
Address:     192.168.56.108#53

pokemon.deaw.es canonical name = debian-deaw.deaw.es.
Name:   debian-deaw.deaw.es
Address: 192.168.56.108
```

USO DEL SERVIDOR DNS PARA RESOLVER SOLICITUDES A UN SERVIDOR WEB

Vamos a crear una nueva máquina virtual que contenga un servidor web, en nuestro caso vamos a instalar el servidor NGINX. Esta máquina virtual también tendrá una red de tipo adaptador sólo anfitrión.

Debemos tener el archivo /etc/hosts tal y como se ve en la siguiente imagen, es decir, sólo debe aparecer la IP del localhost y de la propia máquina:

```
GNU nano 7.2
127.0.0.1      localhost
127.0.1.1      nginx
```

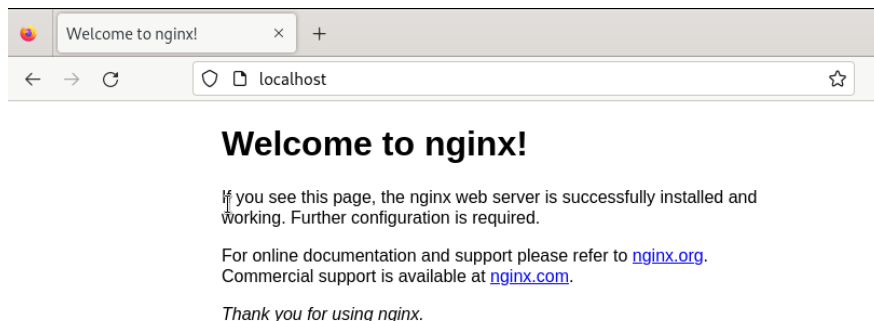
Así mismo debemos hacer que la IP sea fija modificando el fichero /etc/network/interfaces:

```
allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.56.111
    netmask 255.255.255.0
    gateway 192.168.56.1
    dns-nameservers 8.8.8.8
```

Reiniciamos el servicio networking y comprobamos que la IP es la que hemos especificado en este fichero:

```
albertom-servidor@nginx:~$ sudo systemctl restart networking.service
albertom-servidor@nginx:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN g
t 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: enp0s3: <BROADCAST,MULTICAST,DYNAMIC,UP,LOWER_UP> mtu 1500 qdisc f
te UP group default qlen 1000
    link/ether 08:00:27:41:84:31 brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.111/24 brd 192.168.56.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe41:8431/64 scope link
        valid_lft forever preferred_lft forever
```

Para esta práctica vamos a utilizar la página por defecto de NGINX para comprobar el funcionamiento del DNS:



Para ello debemos modificar el archivo default de la carpeta /etc/nginx/sites-available colocando como nameserver www.deaw.es.

```
GNU nano 7.2 /etc/nginx/sites-available/default *

root /var/www/html;

# Add index.php to the list if you are using PHP
index index.html index.htm index.nginx-debian.html;

server_name www.deaw.es;

location / {
    # First attempt to serve request as file, then
    # as directory, then fall back to displaying a 404.
    try_files $uri $uri/ =404;
}
```

Guardamos los cambios y reiniciamos el servicio NGINX para que los cambios surtan efecto:

```
albertom-servidor@nginx:~$ sudo systemctl restart nginx.service
albertom-servidor@nginx:~$ sudo systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-02-02 17:06:20 CET; 5s ago
     Docs: man:nginx(8)
  Process: 3204 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=e
 Process: 3206 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, st
 Main PID: 3207 (nginx)
    Tasks: 3 (limit: 2284)
   Memory: 2.5M
      CPU: 19ms
   CGroup: /system.slice/nginx.service
           └─3207 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─3208 "nginx: worker process"
               └─3209 "nginx: worker process"
```

Volvemos ahora a nuestra máquina virtual que funciona como servidor DNS y vamos a añadir `www` como un nuevo dominio en nuestro archivo de zona `db.deaw.es`, el cual será un RR de tipo A que quedará asociado con la IP de nuestra máquina virtual con rol de servidor web:

```
GNU nano 7.2 /etc/bind/db.deaw.es *
$TTL      604800
@          IN      SOA      debian-deaw.deaw.es. admin.deaw.es. (
                                2024012601      ; Serial
                                3600              ; Refresh
                                1800              ; Retry
                                604800            ; Expire
                                86400)           ; Minimum TTL

deaw.es.   IN      NS       debian-deaw.deaw.es.
debian-deaw IN      A        192.168.56.108
pokemon    IN      CNAME     debian-deaw
www         IN      A         192.168.56.111
```

Guardamos cambios y reiniciamos el servicio `named`:

```
albertom-servidor@debian-deaw:~$ sudo systemctl restart named.service
albertom-servidor@debian-deaw:~$ sudo systemctl status named.service
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Fri 2024-02-02 17:13:51 CET; 1s ago
     Docs: man:named(8)
    Main PID: 2572 (named)
      Status: "running"
        Tasks: 4 (limit: 2284)
      Memory: 32.9M
         CPU: 58ms
    CGroup: /system.slice/named.service
            └─2572 /usr/sbin/named -f -u bind -4
```

Para comprobar el funcionamiento primero vamos a utilizar al servidor DNS como cliente DNS, para ello como ya cambiamos el `nameserver` a la IP del servidor DNS (192.168.56.108) podemos hacer ping contra www.deaw.es:

```
albertom-servidor@debian-deaw:~$ ping www.deaw.es
PING www.deaw.es (192.168.56.111) 56(84) bytes of data:
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=1 ttl=64 time=0.631 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=2 ttl=64 time=4.97 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=3 ttl=64 time=0.857 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=4 ttl=64 time=3.14 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=5 ttl=64 time=0.747 ms
^C
--- www.deaw.es ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4010ms
rtt min/avg/max/mdev = 0.631/2.068/4.972/1.723 ms
```

También podemos preguntar el RR de tipo A utilizando `nslookup`:

```
albertom-servidor@debian-deaw:~$ nslookup www.deaw.es
Server:          192.168.56.108
Address:         192.168.56.108#53

Name:   www.deaw.es
Address: 192.168.56.111
```

Y si accedemos a través de un cliente web al dominio www.deaw.es veremos la página de entrada base de NGINX:



Si ahora abrimos nuestra máquina virtual cliente que creamos en el apartado anterior, podemos hacer las mismas pruebas.

Empezamos realizando la prueba de ping a la dirección www.deaw.es:

```
albertom-cliente@albertoMartinezPerez:~$ ping www.deaw.es
PING www.deaw.es (192.168.56.111) 56(84) bytes of data.
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=1 ttl=64 time=0.724 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=2 ttl=64 time=0.829 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=3 ttl=64 time=0.822 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=4 ttl=64 time=3.44 ms
64 bytes from 192.168.56.111 (192.168.56.111): icmp_seq=5 ttl=64 time=1.13 ms
^C
--- www.deaw.es ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4045ms
rtt min/avg/max/mdev = 0.724/1.389/3.437/1.033 ms
```

Ahora realizamos el nslookup de esta dirección de dominio:

```
albertom-cliente@albertoMartinezPerez:~$ nslookup www.deaw.es
Server:          192.168.56.108
Address:         192.168.56.108#53

Name:   www.deaw.es
Address: 192.168.56.111
```

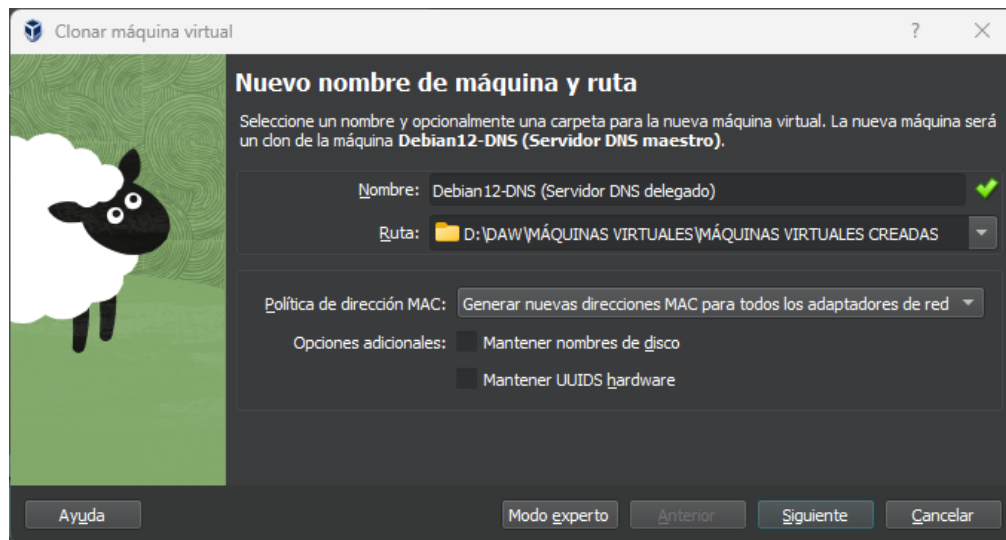
Por último, realizamos un acceso a la web a través de un cliente web:



CREACIÓN DE UN DNS DELEGADO QUE GESTIONE UNA DELEGACIÓN DE ZONA

Ahora vamos a crear un DNS delegado para gestionar una delegación de zona del dominio deaw.es.

Para ello lo primero que haremos es clonar nuestra máquina generando nuevas direcciones MAC para que tenga una IP propia.



Cuando se complete la clonación iniciamos el DNS que actuará como delegado y lo primero que haremos será configurar el archivo `/etc/network/interfaces` para que tenga la IP fija:

```
GNU nano 7.2 /etc/network/interfaces *
# This file describes the network interfaces available on
# and how to activate them. For more information, see the
# file /etc/network/interfaces.d/*

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

allow-hotplug enp0s3
iface enp0s3 inet static
    address 192.168.56.109
    netmask 255.255.255.0
    gateway 192.168.56.1
    dns-nameservers 8.8.8.8
```

También actualizamos el archivo `/etc/hostname` para cambiar el nombre de la máquina:

```
GNU nano 7.2 /etc/hostname *
delegado
```

Por último, actualizamos el archivo `/etc/hosts` para cambiar el nombre asociado al localhost (127.0.1.1):

```
GNU nano 7.2 /etc/hosts *
127.0.0.1    localhost
127.0.1.1    delegado

# The following lines are desirable for IPv6 capable hosts
::1         localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
```

Una vez hechos estos cambios, reiniciamos la máquina para que se tengan en cuenta la siguiente vez que se inicie.

Vamos a hacer que el subdominio “persianas” de `deaw.es` sea gestionado por este nuevo servidor DNS, para ello en el archivo de zona del servidor DNS maestro debemos añadir las siguientes líneas:

```
GNU nano 7.2 ./db.deaw.es *
$TTL      604800
@          IN      SOA      debian-deaw.deaw.es. admin.deaw.es. (
                                2024012601      ; Serial
                                3600              ; Refresh
                                1800              ; Retry
                                604800           ; Expire
                                86400)           ; Minimum TTL

deaw.es.   IN      NS       debian-deaw.deaw.es.
debian-deaw IN      A        192.168.56.108
pokemon    IN      CNAME     debian-deaw
www         IN      A        192.168.56.111

persianas.deaw.es. IN      NS       delegado.persianas.deaw.es.
delegado.persianas.deaw.es. IN      A        192.168.56.109
```

El registro de recurso A recibe el nombre de “glue record” ya que sirve de unión entre el subdominio (persianas) y el dominio (deaw.es).

Ahora volvemos al DNS delegado y vamos a renombrar el fichero `db.deaw.es` a `db.persianas.deaw.es`, para ello desde el directorio `/etc/bind/` usamos el comando:

```
sudo mv ./db.deaw.es ./db.persianas.deaw.es
```

```
albertom-servidor@delegado:/etc/bind$ sudo mv ./db.deaw.es ./db.persianas.deaw.es
albertom-servidor@delegado:/etc/bind$ ls -la
total 76
drwxr-sr-x  2 root bind  4096 ene 29 16:31 .
drwxr-xr-x 136 root root 12288 ene 29 16:18 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   289 ene 29 15:32 db.persianas.deaw.es
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   229 ene 26 17:39 named.conf.local
-rw-r--r--  1 root bind   165 ene 26 17:36 named.conf.local.backup
-rw-r--r--  1 root bind   994 ene 26 17:24 named.conf.options
-rw-r--r--  1 root bind   846 ene 26 16:58 named.conf.options.backup
-rw-r----- 1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```

Debemos modificar el contenido del archivo de zona, de forma que:

- En el registro SOA debemos actualizar el nombre del servidor DNS a delegado.persianas.deaw.es y el correo del administrador a admin.persianas.deaw.es.
- Debemos crear un registro NS que asocie el nombre de dominio persianas.deaw.es. a la máquina que ejerce el rol de DNS delegado.
- Creamos un registro A para resolver el nombre de dominio con la IP correspondiente a esta máquina.

```
GNU nano 7.2                                ./db.persianas.deaw.es
$TTL      604800
@         IN      SOA      delegado.persianas.deaw.es.  admin.persianas.deaw.es. (
                                2024012601      ; Serial
                                3600             ; Refresh
                                1800             ; Retry
                                604800           ; Expire
                                86400)          ; Minimum TTL

persianas.deaw.es.      IN      NS      delegado.persianas.deaw.es.
delegado.persianas.deaw.es.  IN    A      192.168.56.109
```

A continuación modificamos el archivo named.conf.local para que tenga en cuenta este fichero que acabamos de modificar:

```
GNU nano 7.2                                ./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 "persianas.deaw.es" {
    type master;
    file "/etc/bind/db.persianas.deaw.es";
};
```

Por último, vamos a modificar en este servidor el archivo /etc/resolv.conf para que sea el nameserver con la IP de este equipo el que resuelva las solicitudes de nombres:

```
GNU nano 7.2                                /etc/resolv.conf *
nameserver 192.168.56.109
nameserver 192.168.1.1
```

Comprobamos la resolución de nombre con un nslookup de delegado.persianas.deaw.es:

```
albertom-servidor@delegado:/etc/bind$ nslookup delegado.persianas.deaw.es
Server:                192.168.56.109
Address:               192.168.56.109#53

Name:   delegado.persianas.deaw.es
Address: 192.168.56.109
```


CREACIÓN DE UN SERVIDOR DNS ESCLAVO

En este apartado vamos a crear un DNS esclavo para visualizar la transferencia de zona, en este caso debemos seguir los mismos pasos que en el paso anterior y clonar la máquina virtual del DNS primario.

Así mismo debemos modificar:

- El archivo `/etc/network/interfaces` para asegurarnos el tener una IP fija.
- El fichero `/etc/hostname` como en pasos previos para que la máquina tenga un nombre único.
- El fichero `/etc/hosts` para que la IP 127.0.1.1 coincida con el nombre de la máquina.
- El fichero `/etc/resolv.conf` para que el nameserver sea el propio servidor.

A continuación, debemos eliminar el fichero de zona de este servidor DNS ya que lo recibiremos por transferencia de zona:

```
sudo rm /etc/bind/nombre_fichero_zona
```

```
albertom-servidor@esclavo:~$ sudo rm /etc/bind/db.deaw.es
```

Con esto listo podemos comenzar con las configuraciones necesarias para permitir la transferencia de zona.

En primer lugar vamos a modificar la máquina que tiene el rol de esclavo en esta práctica, debemos modificar el archivo `named.conf.local`:

```
zone "deaw.es" {  
    type slave;  
    file "/etc/bind/db.deaw.es";  
    masters{192.168.56.108};  
};
```

Añadimos una directiva `masters` que contiene las IPs que serán maestra de este servidor, además cambiamos el `type` de la zona de `master` a `slave` ya que este será el servidor esclavo.

En la máquina principal que tiene el rol de maestro, debemos modificar el fichero `named.conf.local`:

```
zone "deaw.es" {  
    type master;  
    file "/etc/bind/db.deaw.es";  
    allow-transfer{192.168.56.112};  
    also-notify{192.168.56.111};  
};
```

Añadimos dos nuevas directivas:

- `allow-transfer`: Que indica el conjunto de IPs que recibirán el fichero de zona. Es decir, que serán servidores secundarios o esclavos.
- `also-notify`: Indica el conjunto de IPs que serán notificados.

También debemos modificar el fichero de zona:

```
GNU nano 7.2
$TTL      604800
@         IN      SOA      debian-deaw.deaw.es. admin.deaw.es. (
                                2024021201
                                3600
                                1800
                                604800
                                86400 )

deaw.es.   IN      NS       debian-deaw.deaw.es.
debian-deaw IN      A       192.168.56.108
pokemon    IN      CNAME    debian-deaw
www         IN      A       192.168.56.111

deaw.es.   IN      NS       esclavo.deaw.es.
esclavo     IN      A       192.168.56.112
```

Añadimos dos nuevos RR, un NS que asocie el dominio deaw.es. al nombre de la máquina que ejercerá el rol de DNS secundario (en nuestro caso 'esclavo') y un A que asocie el dominio esclavo con la IP del DNS secundario.

Con esto finalizaría la configuración del DNS primario. Reiniciamos el servicio y comprobamos el estado:

```
albertom-servidor@debian-deaw:~$ sudo systemctl restart named.service
albertom-servidor@debian-deaw:~$ sudo systemctl status named.service
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-02-12 22:13:18 CET; 5s ago
     Docs: man:named(8)
    Main PID: 2818 (named)
    Status: "running"
      Tasks: 4 (limit: 2284)
     Memory: 31.0M
        CPU: 36ms
    CGroup: /system.slice/named.service
            └─2818 /usr/sbin/named -f -u bind -4

feb 12 22:13:18 debian-deaw named[2818]: managed-keys-zone: loaded serial 71
feb 12 22:13:18 debian-deaw named[2818]: zone localhost/IN: loaded serial 2
feb 12 22:13:18 debian-deaw named[2818]: zone 0.in-addr.arpa/IN: loaded serial 1
feb 12 22:13:18 debian-deaw named[2818]: zone 127.in-addr.arpa/IN: loaded serial 1
feb 12 22:13:18 debian-deaw named[2818]: zone deaw.es/IN: loaded serial 2024021201
feb 12 22:13:18 debian-deaw named[2818]: zone 255.in-addr.arpa/IN: loaded serial 1
feb 12 22:13:18 debian-deaw named[2818]: all zones loaded
feb 12 22:13:18 debian-deaw named[2818]: running
feb 12 22:13:18 debian-deaw named[2818]: zone deaw.es/IN: sending notifies (serial 2024021201)
feb 12 22:13:18 debian-deaw systemd[1]: Started named.service - BIND Domain Name Server.
```

Si todo ha funcionado de forma correcta hacemos lo mismo en nuestra máquina esclava:

```
albertom-servidor@esclavo:~$ sudo systemctl restart named.service
albertom-servidor@esclavo:~$ sudo systemctl status named.service
● named.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/named.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-02-12 22:24:29 CET; 1s ago
     Docs: man:named(8)
    Main PID: 3017 (named)
    Status: "running"
      Tasks: 4 (limit: 2284)
     Memory: 32.9M
        CPU: 37ms
    CGroup: /system.slice/named.service
            └─3017 /usr/sbin/named -f -u bind -4

feb 12 22:24:29 esclavo named[3017]: all zones loaded
feb 12 22:24:29 esclavo systemd[1]: Started named.service - BIND Domain Name Server.
feb 12 22:24:29 esclavo named[3017]: running
feb 12 22:24:29 esclavo named[3017]: zone deaw.es/IN: Transfer started.
feb 12 22:24:29 esclavo named[3017]: transfer of 'deaw.es/IN' from 192.168.56.108#53: connected using 192.168.56.108#53
feb 12 22:24:29 esclavo named[3017]: zone deaw.es/IN: transferred serial 2024021201
feb 12 22:24:29 esclavo named[3017]: transfer of 'deaw.es/IN' from 192.168.56.108#53: Transfer status: success
feb 12 22:24:29 esclavo named[3017]: transfer of 'deaw.es/IN' from 192.168.56.108#53: Transfer completed: 1 messages, 11 records,
feb 12 22:24:29 esclavo named[3017]: zone deaw.es/IN: sending notifies (serial 2024021201)
```

Esto ha generado una copia del fichero db.deaw.es es nuestro servidor esclavo:

```
albertom-servidor@esclavo:~$ ls -la /etc/bind/
total 76
drwxr-sr-x  2 bind bind  4096 feb 12 22:24 .
drwxr-xr-x 137 root root 12288 feb 12 22:00 ..
-rw-r--r--  1 root root  2403 sep 21 19:33 bind.keys
-rw-r--r--  1 root root   255 sep 21 19:33 db.0
-rw-r--r--  1 root root   271 sep 21 19:33 db.127
-rw-r--r--  1 root root   237 sep 21 19:33 db.255
-rw-r--r--  1 root bind   718 feb 12 22:24 db.deaw.es
-rw-r--r--  1 root root   353 sep 21 19:33 db.empty
-rw-r--r--  1 root root   270 sep 21 19:33 db.local
-rw-r--r--  1 root bind   458 sep 21 19:33 named.conf
-rw-r--r--  1 root bind   498 sep 21 19:33 named.conf.default-zones
-rw-r--r--  1 root bind   255 feb 12 22:12 named.conf.local
-rw-r--r--  1 root bind   165 ene 26 17:36 named.conf.local.backup
-rw-r--r--  1 root bind   944 feb 12 16:06 named.conf.options
-rw-r--r--  1 root bind   846 ene 26 16:58 named.conf.options.backup
-rw-r-----  1 bind bind   100 ene 26 16:32 rndc.key
-rw-r--r--  1 root root  1317 sep 21 19:33 zones.rfc1918
```

El contenido de este fichero es exactamente el mismo que en el servidor maestro:

```
albertom-servidor@esclavo: ~
/etc/bind/db.deaw.es

GNU nano 7.2
@TTL      604800      IN      SOA      debian-deaw.deaw.es. admin.deaw.es. (
                                2024021201
                                3600
                                1800
                                604800
                                86400 )

deaw.es.      IN      NS       debian-deaw.deaw.es.
debian-deaw  IN      A        192.168.56.108
pokemon      IN      CNAME     debian-deaw
www          IN      A        192.168.56.111

deaw.es.      IN      NS       esclavo.deaw.es.
esclavo      IN      A        192.168.56.112
```