

Instalación y configuración de un servidor BIND

Hamza Akdi

31/10/2025

Indice :

Indice :	2
PARTE 1: SERVIDOR COMO CACHE.....	3
PARTE 2: SERVIDOR DE ZONA.....	10
PARTE 3: DISTRIBUCIÓN DE CARGA UTILIZANDO DNS	15
PARTE FINAL.....	17
Comentarios	17

PARTE 1: SERVIDOR COMO CACHE

Instala el servidor de DNS bind en tu máquina Linux:

En primer lugar, actualizaremos los repositorios con el siguiente comando:

```
|root@debian:~# sudo apt-get update
```

Una vez actualizados los repositorios, comenzaremos con la instalación de **BIND9**.

```
root@debian:~# sudo apt-get install bind9
  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:
    bind9-doc resolvconf ufw
  Se instalarán los siguientes paquetes NUEVOS:
    bind9 bind9-utils
```

(Instalación de BIND9)

Además, instalaremos los paquetes sugeridos y sus dependencias.

```
root@debian:~# sudo apt install bind9-doc dnsutils resolvconf ufw python-ply-doc
Nota, seleccionando «bind9-dnsutils» en lugar de «dnsutils»
bind9-dnsutils ya está en su versión más reciente (1:9.20.15-1~deb13u1).
Installing:
  bind9-doc python-ply-doc resolvconf ufw

Installing dependencies:
  iptables libip4tc2 libip6tc2

Paquetes sugeridos:
  firewalld rsyslog
```

(Instalación de paquetes y dependencias)

El servicio se arranca automáticamente, lo cual podemos comprobar con el siguiente comando:

```
root@hamza:~# sudo service bind9 status
● named.service - BIND Domain Name Server
  Loaded: loaded (/usr/lib/systemd/system/named.service; enabled; preset: en>
  Active: active (running) since Thu 2025-10-30 17:38:23 CET; 11min ago
  Invocation: 94d56ff98a8d4d7eb88e9b6b17784335
    Docs: man:named(8)
   Main PID: 3200 (named)
     Status: "running"
        Tasks: 14 (limit: 5793)
      Memory: 56.4M (peak: 58.4M)
        CPU: 87ms
      CGroup: /system.slice/named.service
              └─3200 /usr/sbin/named -f -u bind
```

(Estado del servicio BIND)

Fichero de configuración que permite especificar los servidores DNS, configuraremos el primer nameserver con la **IP 127.0.0.1**, para redirigir todas las solicitudes de nombres de dominio a la propia máquina local (localhost).



Comprueba si se está ejecutando el proceso 'named' en tu máquina:

Finalizada la instalación, podremos comprobar si el servicio **DNS** se ha iniciado y el proceso “**named**”. Para ello ejecutaremos el siguiente comando:

```
root@debian:~# ps -ef | grep named
bind      5512      1  0 10:35 ?        00:00:00 /usr/sbin/named -f -u bind
root      6516    4303  0 10:45 pts/0    00:00:00 grep named
```

(Comprobación de inicio DNS)

Observa qué puertos utiliza el servidor de DNS (por ejemplo, con la orden netstat -ltun)

Observaremos también los puertos que utiliza el servidor DNS con el siguiente comando, pero antes debemos actualizar y descargar las herramientas network del sistema.

```
root@debian:~# sudo apt install net-tools
```

(Comando instalación Network)

```
root@debian:~# netstat -ltun | grep :53
```

tcp	0	0	10.0.2.15:53	0.0.0.0:*	LISTEN
tcp	0	0	10.0.2.15:53	0.0.0.0:*	LISTEN
tcp	0	0	10.0.2.15:53	0.0.0.0:*	LISTEN
tcp	0	0	10.0.2.15:53	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN
tcp	0	0	127.0.0.1:53	0.0.0.0:*	LISTEN

(Comando para ver los puertos DNS)

Realiza una captura del directorio /etc/bind. ¿Qué contiene el fichero named.conf?

A continuación, realizaremos una captura del directorio **/etc/bind**. En el que podemos observar los ficheros en la ruta.

```
root@debian:/etc/bind# ls
```

```
named.conf  named.conf.local  named.conf.options  named.conf.root-hints  rndc.key
```

(Captura del directorio /etc/bind)

Una vez en la ruta, accederemos al fichero named.conf.

```
root@debian:/etc/bind# nano named.conf
```

```
GNU nano 8.4                               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.root-hints";
```

(Fichero de configuración named.conf)

- **include "/etc/bind/named.conf.options"**: Dónde configuras el modo caché y los reenviadores.
- **include "/etc/bind/named.conf.local"**: Donde declaras tus propios dominios.
- **include "/etc/bind/named.conf.root-hints"**: Esto es lo que permite que el servidor resuelva consultas recursivas desde el inicio de la jerarquía global de DNS.

Para poder responder consultas, bind debe tener las direcciones de los servidores raíz: investiga dónde se encuentran tales direcciones en tu Linux.

Las direcciones de los servidores raíz se encuentran en el fichero conocido como "**root hints**". Este archivo contiene la ubicación de la zona raíz, que es el indicado abajo:

```
GNU nano 8.4                               named.conf.root-hints *
/ prime the server with knowledge of the root servers
zone "." {
    type hint;
    file "/usr/share/dns/root.hints";
:                                         
```

(Fichero de ruta servidores root.hints)

Una vez tenemos la ruta de acceso al fichero pondremos el siguiente comando para acceder:

```
root@debian:~# nano /usr/share/dns/root.hints
```

Este es el directorio de direcciones de partida para el sistema de nombres de dominio (DNS) general. Es el punto de partida que permite que el servidor DNS pueda resolver *cualquier* dominio en Internet.

```
GNU nano 8.4                               /usr/share/dns/root.hints
;      This file holds the information on root name servers needed to
;      initialize cache of Internet domain name servers
;      (e.g. reference this file in the "cache . <file>" configuration
;      file of BIND domain name servers).
;
;      This file is made available by InterNIC
;      under anonymous FTP as
;          file                  /domain/named.cache
;          on server            FTP.INTERNIC.NET
;      -OR-
;          file                  /domain/named.cache
;          on server            RS.INTERNIC.NET
;
;      last update:    April 18, 2024
;      related version of root zone:   2024041801
;
; FORMERLY NS.INTERNIC.NET
;
.           3600000      NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000      A      198.41.0.4
A.ROOT-SERVERS.NET. 3600000      AAAA   2001:503:ba3e::2:30
.
```

(Fichero raíz de direcciones DNS)

Comprueba con nslookup que se resuelven consultas recursivas ya (sin configurar reenviadores, es decir, tal como hemos instalado bind).

El objetivo de este paso es forzar el servidor BIND a usar sus propias ‘pistas raíz’ para resolver una consulta externa de principio a fin, demostrando la funcionalidad recursiva.

Primero, debemos configurar el servicio nslookup, que es una herramienta de línea de comandos utilizada para consultar servidores de nombres de dominio (DNS) para obtener información sobre dominios o direcciones IP.

La herramienta cliente nos permite enviar peticiones de resolución de nombres a un servidor específico y ver si este responde correctamente.

```
root@debian:~# nslookup www.google.com 127.0.0.1
Server:  127.0.0.1
Address: 127.0.0.1#53

Non-authoritative answer:
Name: www.google.com
Address: 142.250.200.68
Name: www.google.com
Address: 2a00:1450:4003:80d::2004
```

(Peticiones de resolución de nombres a un servidor)

Consulta con dig un nombre de dominio que no esté en la caché de tu servidor.

El objetivo de esta consulta fue probar la capacidad de nuestro servidor BIND para realizar una resolución recursiva completa. Al especificar `@127.0.0.1`, forzamos al cliente a interrogar a nuestro servidor local, asegurando que cualquier respuesta proviene de él y no de un DNS externo.

```
root@debian:~# dig @127.0.0.1 www.marca.com

; <>> DiG 9.20.15-1~deb13u1-Debian <>> @127.0.0.1 www.marca.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 38001
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 64fb08345c762eee0100000069036b05c6af380d4ec0d708 (good)
;; QUESTION SECTION:
;www.marca.com.           IN      A

;; Query time: 108 msec
;; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
;; WHEN: Thu Oct 30 14:41:25 CET 2025
;; MSG SIZE  rcvd: 70
```

Como se observa en la captura el ‘**Query time**’, el tiempo de respuesta obtenido fue de 108 milisegundos. Es relativamente alto, es una prueba de que la consulta se realizó con la **caché fría**.

Este lapso representa el proceso completo que BIND tuvo que realizar: iniciar la recursión, consultar a los servidores raíz, recibir la referencia al TLD, y contactar finalmente al servidor.

Consulta con *dig* de nuevo el mismo nombre de dominio y apunta la diferencia de tiempos en este caso con el anterior.

Haremos la misma consulta de nuevo en el mismo nombre de dominio y claramente el motivo es el funcionamiento de BIND como caché. Para resolver la primera consulta BIND hizo una consulta recursiva, para resolver la segunda, utilizó la caché.

```
root@hamza:~# dig @127.0.0.1 www.marca.com

<>> DiG 9.20.15-1-deb13u1-Debian <>> @127.0.0.1 www.marca.com
(1 server found)
; global options: +cmd
; Got answer:
; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17439
; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

; OPT PSEUDOSECTION:
EDNS: version: 0, flags:; udp: 1232
COOKIE: a3b34d56818f956d01000000690398e41bf5b4150c05a3c8 (good)
; QUESTION SECTION:
www.marca.com.           IN      A

; ANSWER SECTION:
www.marca.com.      295     IN      CNAME   unidadeditorial.map.fastly.net.
unidadeditorial.map.fastly.net. 55 IN      A      199.232.33.50

; Query time: 0 msec
; SERVER: 127.0.0.1#53(127.0.0.1) (UDP)
; WHEN: Thu Oct 30 17:57:08 CET 2025
; MSG STZF rcvd: 130
```

(Segunda consulta dig al mismo nombre de dominio)

Vamos a hacer que nuestro servidor pueda reenviar consultas a otro servidor caché. Copia la parte del fichero de configuración donde has definido el reenviador (*forwarders*). Puede ser un servidor como 1.1.1.1, 8.8.8.8 o el de un operador.

Esta configuración indica a nuestro servidor BIND que, en lugar de realizar la resolución de nombres desde los servidores raíz, debe enviar esas consultas a los servidores DNS externos que especifiquemos.

Primero, iremos a la siguiente ruta y editaremos el archivo **named.conf.options**. Indicaremos como reenviadores (*forwarders*) las direcciones de Cloudflare y Google (1.1.1.1 - 8.8.8.8).

```
GNU nano 8.4                               /etc/bind/named.conf.options *
options {
    directory "/var/cache/bind";
    forwarders {
        1.1.1.1;
        8.8.8.8;
    };
};
```

(Configuración del reenviador)

Una vez configurado el archivo aplicaremos los cambios realizados reiniciando el servicio con el siguiente comando:

```
root@hamza:~# sudo systemctl restart bind9
```

Para comprobar el funcionamiento del servidor, volveremos a utilizar la herramienta nslookup:

```
root@hamza:~# nslookup www.wikipedia.com 127.0.0.1
Server:      127.0.0.1
Address:     127.0.0.1#53

Non-authoritative answer:
www.wikipedia.com      canonical name = ncredir-lb.wikimedia.org.
Name:    ncredir-lb.wikimedia.org
Address: 185.15.58.226
Name:    ncredir-lb.wikimedia.org
Address: 2a02:ec80:600:ed1a::3
```

(Comprobación de funcionamiento del servicio con nslookup)

PARTE 2: SERVIDOR DE ZONA

En este apartado debes diseñar un fichero de zona de resolución directa que contenga al menos los siguientes registros: SOA, NS, A (al menos 4), CNAME (2) y MX.

Este fichero contiene el listado de todos los recursos que componen el dominio (SOA, NS, A, MX, CNAME).

- **SOA (Start of Authority):** Define la autoridad primaria del servidor para la zona, incluyendo parámetros de tiempo y el número de serie que debe incrementarse cada vez que se modifica el archivo.
- **NS (Name Server):** Identifica a ns1.hamza.net (192.168.1.22) como el servidor de nombres oficial para el dominio.
- **A (Address):** Mapea un nombre de host a una dirección IPv4.
- **MX (Mail Exchanger):** Indica qué servidor es el responsable de recibir el correo electrónico de tu dominio (mail.hamza.net).
- **CNAME (Canonical Name):** Crea alias o sobrenombres que apuntan a otros hosts dentro de tu dominio.

El objetivo es convertir nuestro servidor DNS en una **autoridad** para el dominio hamza.net, creando y configurando los archivos necesarios que veremos a continuación:

Abriremos el archivo: **sudo nano /etc/bind/named.conf.local**

```
GNU nano 8.4                               named.conf.local *
// Do any local configuration here
//
// Declaración de la zona (hamza.net)
zone "hamza.net" {
    type master;
    file "/etc/bind/db.hamza.net";
    allow-update {none; };
};
```

(Archivo declaración de zona)

Realiza una captura del fichero donde específicas que tu servidor trabaja como maestro de la zona hamza.net.

Realiza una captura del fichero de zona que has creado y explícalo.

Creación del fichero de zona (hamza.net), reemplazaremos el contenido del nuevo archivo de zona por el siguiente.

```
GNU nano 8.4                               /etc/bind/db.hamza.net *
$TTL 604800
@ IN SOA ns1.hamza.net. admin.hamza.net. (
    2025110103
    604800
    86400
    2419200
    604800 )

@ IN NS ns1.hamza.net.
ns1 IN A 192.168.1.22
@ IN MX 10 mail.hamza.net.
mail IN A 192.168.1.55

www IN A 192.168.1.23
servidor1 IN A 192.168.106.61
servidor2 IN A 192.168.106.62
db IN A 192.168.106.63
vpn IN A 192.168.106.64

ftp IN CNAME servidor1
docs IN CNAME db
```

(Configuración del archivo de resolución directa)

Verificar y Reiniciar

Verificamos la correcta configuración del archivo, en la que será correcto si aparece en la parte inferior el mensaje “OK”.

```
root@hamza:/home/hamza# sudo named-checkzone hamza.net /etc/bind/db.hamza.net
zone hamza.net/IN: loaded serial 202510103
OK
```

Por último, reiniciamos el servicio **BIND**.

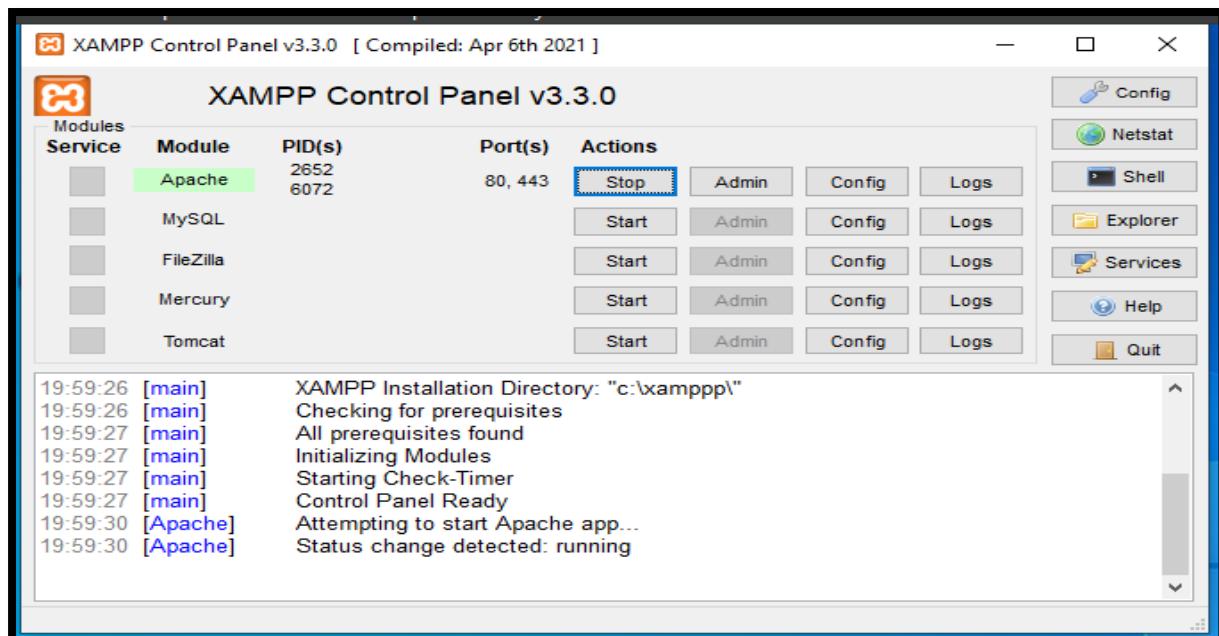
```
[root@hamza:/home/hamza# sudo systemctl restart bind9
```

Asigna un alias www a una máquina Windows donde tengas un servidor Apache.

Realiza pruebas de conexión al servidor utilizando las consultas al servidor bind.

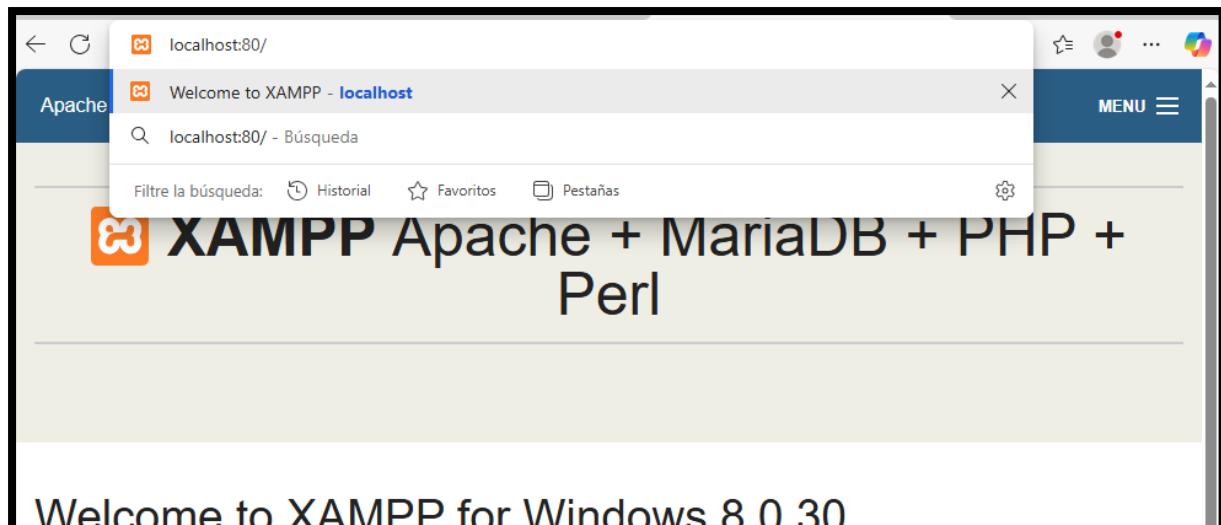
Para asignar un alias a nuestra máquina Cliente Windows, primero debemos instalar **XAMPP**.

Una vez instalado activaremos el **servidor Apache** y escuchando en el **puerto 80**.



(Activación servicio Apache desde XAMPP)

Una vez activado desde **XAMPP**, comprobaremos desde el navegador que está abierto el servidor de Apache.



(Búsqueda en navegador *localhost:80*)

A continuación, **nslookup** intentará resolver el nombre `www.hamza.net`. Al añadir la IP `192.168.1.22` al final, forzamos a que la consulta se dirija directamente a tu servidor **BIND**.

```
root@hamza:/home/hamza# nslookup www.hamza.net 192.168.1.22
Server:      192.168.1.22
Address:     192.168.1.22#53

Name:   www.hamza.net
Address: 192.168.1.23
```

(Resolución de alias directa al servidor BIND)

La imagen verifica que **BIND** utiliza el registro **www IN A 192.168.1.23** del fichero de zona, confirmando que la zona funciona correctamente y el DNS es válido.

Documenta capturas con `nslookup` y `dig` de accesos a cada tipo de registro que tienes en el fichero de zona (puedes hacer las consultas desde tu propia máquina Linux).

Ahora documentaremos con `nslookup` y `dig` accesos a cada tipo de registro en el fichero de zona de la siguiente manera:

- **Registro CNAME (Canonical Name):** Esta prueba `dig` muestra que `ftp.hamza.net` es un **alias (CNAME)** que apunta internamente a `servidor1.hamza.net`. Esto demuestra el uso de registros canónicos para asignar nombres secundarios a un mismo servidor (192.168.1.61), facilitando la gestión y creando múltiples puntos de acceso.

```

root@hamza:/home/hamza# dig db.hamza.net @192.168.1.22
; <>> DiG 9.20.15-1~deb13u1-Debian <>> db.hamza.net @192.168.1.22
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 8092
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 8749ef4ff20a64b40100000069065fd380be45377d3c641c (good)
;; QUESTION SECTION:
;db.hamza.net.           IN      A

;; ANSWER SECTION:
db.hamza.net.       604800  IN      A      192.168.106.63

;; Query time: 0 msec
;; SERVER: 192.168.1.22#53(192.168.1.22) (UDP)
;; WHEN: Sat Nov 01 20:30:27 CET 2025
;; MSG SIZE rcvd: 85

```

(Registro CNAME)

- Registro A (Address) y Alias: La consulta directa a **BIND** mediante nslookup confirma que el registro **A** para www.hamza.net se resuelve correctamente a **192.168.1.23**. Esto verifica el mapeo de nombres de *host* a direcciones IPv4 y es fundamental para la conectividad del servicio Apache."

```

root@hamza:/home/hamza# nslookup www.hamza.net 192.168.1.22
Server:      192.168.1.22
Address:     192.168.1.22#53

Name:   www.hamza.net
Address: 192.168.1.23

```

(Registro A (Address))

- Registro MX (Mail Exchange): La consulta **MX** (Mail Exchanger) confirma que el servidor de correo de la zona es **mail.hamza.net** y tiene una prioridad de **10**. Este registro es esencial para dirigir el tráfico de correo electrónico entrante a un *host* específico en la red.

```

root@hamza:/home/hamza# nslookup -type=MX hamza.net 192.168.1.22
Server:      192.168.1.22
Address:     192.168.1.22#53

hamza.net      mail exchanger = 10 mail.hamza.net.

root@hamza:/home/hamza# nslookup -type=MX hamza.net 192.168.1.22
Server:      192.168.1.22
Address:     192.168.1.22#53

hamza.net      mail exchanger = 10 mail.hamza.net.

```

(Registro MX (Mail Exchange))

- Registro NS (Name Server): El resultado de esta consulta NS (Name Server) muestra que el servidor autorizado para la zona hamza.net es ns1.hamza.net (192.168.1.22). Esto valida que el servidor está actuando correctamente como **autoridad principal** para la zona, tal como se configuró en el **named.conf.local**.

```
root@hamza:/home/hamza# dig hamza.net NS @192.168.1.22
; <>> DiG 9.20.15-1~deb13u1-Debian <>> hamza.net NS @192.168.1.22
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 19049
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 2
;
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: c9d074bb02dc68e6010000006906636acb12b77adb74b0da (good)
; QUESTION SECTION:
;hamza.net.           IN      NS
;
;; ANSWER SECTION:
hamza.net.        604800  IN      NS      ns1.hamza.net.
;
;; ADDITIONAL SECTION:
ns1.hamza.net.    604800  IN      A       192.168.1.22
;
;; Query time: 0 msec
;; SERVER: 192.168.1.22#53(192.168.1.22) (UDP)
;; WHEN: Sat Nov 01 20:45:46 CET 2025
;; MSG SIZE  rcvd: 100
```

(Registro NS (NameServer))

PARTE 3: DISTRIBUCIÓN DE CARGA UTILIZANDO DNS

¿Qué versión de bind tienes instalada?

La versión de **BIND** instalada en el servidor es la **9.20.15**. Este dato es relevante para asegurar que todas las funcionalidades, como el **Round Robin**, estén disponibles y se configuren según la sintaxis estándar de esta rama de BIND.

```
root@hamza:/home/hamza# sudo apt show bind9
Package: bind9                                         Comando para ver versión
Version: 1:9.20.15-1~deb13u1                         Versión Bind9
Priority: optional
Section: net
Maintainer: Debian DNS Team <team+dns@tracker.debian.org>
Installed-Size: 912 kB
Pre-Depends: init-system-helpers (>= 1.54~)
Depends: adduser, bind9-libs (= 1:9.20.15-1~deb13u1), bind9-utils (= 1:9.20.15->
Suggests: bind9-doc, bind9-dnsutils, resolvconf, ufw
```

(Versión de BIND)

Comprueba el funcionamiento de esta característica: crea tres entradas asociadas al mismo nombre de dominio con tres direcciones IP diferentes y haz pruebas con el nslookup. Muestra las consultas y los registros añadidos al fichero de zona de resolución directa.

Para implementar el Round Robin, añadimos múltiples registros A con el **mismo nombre de host** (web-rr) y diferentes **IPs**. Esto hace que BIND rote las respuestas en cada consulta.

Primero abriremos el fichero de zona en la ruta **/etc/bind/db.hamza.net** y añadiremos las siguientes líneas al final de los **registros A**.

```
GNU nano 8.4                                     /etc/bind/db.hamza.net *
$TTL    604800
@       IN      SOA     ns1.hamza.net. admin.hamza.net. (
                        2025110103
                        604800
                        86400
                        2419200
                        604800 )

@       IN      NS      ns1.hamza.net.
ns1    IN      A       192.168.1.22

@       IN      MX 10   mail.hamza.net.
mail   IN      A       192.168.1.55

www    IN      A       192.168.1.23
servidor1 IN      A       192.168.106.61
servidor2 IN      A       192.168.106.62
db      IN      A       192.168.106.63
vpn    IN      A       192.168.106.64

ftp    IN      CNAME   servidor1
docs   IN      CNAME   db

web-rr IN A 192.168.1.201
web-rr IN A 192.168.1.202
web-rr IN A 192.168.1.203
```



(Registros de Round Robin)

Una vez añadida la anterior configuración, guardaremos el archivo y verificaremos que la sintaxis y la configuración se ha hecho correctamente con el siguiente comando:

```
root@hamza:/home/hamza# sudo named-checkzone hamza.net /etc/bind/db.hamza.net
zone hamza.net/IN: loaded serial 2025110103
OK
```

Por último, reiniciaremos el servicio BIND.

```
root@hamza:/home/hamza# sudo systemctl restart bind9
```

Después de reiniciar **BIND**, mostraremos que las respuestas para **web-rr.hamza.net** rotan en cada consulta, lo que confirma el funcionamiento del Round Robin. Utilizaremos el siguiente comando de prueba, que tendrá que ser ejecutado **tres veces seguidas**, apuntando al servidor **BIND**. Con cada consulta. Esto demuestra que BIND está configurado para distribuir la carga de manera equitativa entre los hosts definidos.

```
root@hamza:/home/hamza# nslookup web-rr.hamza.net 192.168.1.22
Server:      192.168.1.22
Address:    192.168.1.22#53
Name:  web-rr.hamza.net
Address: 192.168.1.201
Name:  web-rr.hamza.net
Address: 192.168.1.203
Name:  web-rr.hamza.net
Address: 192.168.1.202
```

Este comando tres veces seguidas

(Confirmación funcionamiento Round Robin)

PARTE FINAL

Comentarios

Por favor, comenta en este apartado problemas que has tenido, qué partes no has podido realizar/probar, aspectos que te han llamado la atención y cualquier otro aspecto que consideres interesante.

Problemas que he tenido:

- Elegí el dominio hamza.net, presentando un riesgo de colisión con servidores DNS externos, pero se mitigó asegurando que BIND actuara como autoridad principal.