Configuración básica de un servidor DNS

DNS (*Domain Name System*) es un protocolo de resolución de nombres que permite hacer la relación entre un nombre de host y una dirección IP. Fue creado en 1983 por **Paul Mockapetris** y **Jon Postel**, y definido originalmente en las RFC 882 y 883 (que fueron después reemplazadas por las RFC 1034 y 1035). Se ha convertido en el estándar de facto para la resolución de nombres en Internet.

1. Principios de DNS

DNS es un sistema de resolución de nombres estructurado en arborescencia. Esta arborescencia virtual está organizada en dominios y en subdominios. Cada dominio o subdominio constituye un nudo de la arborescencia y puede contener nombres asociados a direcciones IP. Existe también una rama de la arborescencia que permite resolver direcciones IP en nombres de host o de servicios (resolución inversa).

Los dominios y subdominios los gestionan servidores de nombres DNS que cooperan entre ellos para dar a los clientes DNS la relación entre nombres y direcciones IP. Estas últimas pueden ser direcciones IPv4 o IPv6.

La arborescencia DNS puede ser privada o pública. La arborescencia DNS más grande es la de Internet.

a. Clientes y servidores DNS

Un **cliente DNS** (*resolver*) interroga a un servidor DNS para conocer la o las direcciones IP que corresponden a un nombre (resolución de nombre), o al contrario, para conocer el o los nombres que corresponden a una dirección IP (resolución de dirección o resolución inversa). El servidor DNS responde si tiene elementos suficientes, en caso contrario interrogará a otros servidores DNS y responderá al final a su cliente, ofreciéndole los

elementos solicitados o un código de respuesta de fallo de la resolución.

Un nombre puede ser el nombre de máquina (nombre de host) o un nombre de servicio (por ejemplo, servidor de nombres, servidor de mensajería …). Un nombre puede corresponder a distintas direcciones IP, una dirección IP puede corresponder a distintos nombres.

Para poder gestionar un gran número de nombres, en evolución perpetua, de manera fiable y descentralizada, el sistema DNS está estructurado en un arborescencia que combina diferentes conjuntos independientes, gestionados por servidores DNS responsables de una parte de la arborescencia y cooperando entre ellos.

Esta organización ofrece muchas ventajas:

- Gestión dinámica y descentralizada: la actualización de la información se hace únicamente en el servidor DNS responsable de la parte correspondiente a la actualización.
- Tolerancia a fallos: los servidores DNS responsables de una zona pueden replicar sus datos hacia uno o varios servidores DNS, asegurando la tolerancia a fallos (excepto para las actualizaciones) y el reparto de la carga.
- Autonomía de gestión: el conjunto de los nombres/direcciones que se tienen que gestionar está distribuido entre entidades independientes entre ellas, pero que cooperan. No es necesaria una organización centralizada y única que administre el conjunto de los servidores. Si una parte de la arborescencia presenta un problema o está mal configurada, no afectará a las otras partes que se encuentran en el mismo nivel de la arborescencia.

b. Nombres de dominios completos (FQDN)

La raíz de la arborescencia se representa con el carácter **punto**. Directamente bajo la raíz se encuentran los dominios de primer nivel, TLD (*Top Level Domains*). Dentro de cada uno de ellos, encontramos múltiples dominios y subdominios. Para especificar el nivel de un dominio en la arborescencia, se utiliza la sintaxis siguiente:

```
NombreDominioN.NombreDominioN-1.NombreDominioN-2.[...].NombreDominio1[.]
```

Al contrario de la convención para las arborescencias de sistemas de archivos, partimos del nivel más bajo y subimos hacia la raíz. Esta última puede ser o no explícitamente especificada por el carácter punto.

Un dominio (o subdominio) corresponde a conjuntos de nombres, particularmente, nombres de máquinas, llamados nombres de host.

Un nombre de host corresponde a una o distintas direcciones IP, en versión 4 y/o 6. Para identificar de manera única un nombre de dominio, especificamos su FQDN (*Fully Qualified Domain Name*), subiendo toda la arborescencia del sistema DNS al que pertenece, según la sintaxis vista anteriormente. Según los casos, el carácter punto de la raíz es obligatorio o no.

Ejemplo

FQDN de un nombre de dominio de un servidor HTTP:

www.eni.com.

2. Los tipos de servidores DNS

Los servidores DNS aseguran dos funciones: gestionar los nombres de dominios DNS (nombres de hosts y nombres de servicios) y la relación con las direcciones IP correspondientes, y responder a las solicitudes de resolución de nombre o de dirección emitidas por los clientes DNS. Para ello, distinguimos diferentes tipos de servidores DNS, sabiendo que un mismo servidor puede desempeñar distintos roles.

a. Servidor de nombres DNS primario o secundario

Los nombres DNS están gestionados por servidores DNS. Cada conjunto de nombres, que puede constituir un dominio completo o un subdominio, se encuentra bajo la responsabilidad de un servidor DNS **maestro** (*master server*), llamado también servidor **primario** o **principal** (*primary server*), siendo un **servidor autoritativo** (*authoritative server*). Este servidor es el único que puede tomar en cuenta las actualizaciones de los nombres de dominio de los que se ocupa. Puede duplicar, en solo lectura, su base de nombres

hacia uno o distintos servidores **no maestros** (*non-master server*), llamados también **servidores secundarios** (*secondary server*). Estos también serán autoritativos en el dominio o el subdominio, pero no pueden gestionar directamente las actualizaciones de los nombres de dominio.

b. Servidor DNS de caché o transitario (forwarder)

Algunos servidores DNS no gestionan directamente los nombres. Su rol es tomar en cuenta las solicitudes de resolución de nombres emitidas por los clientes DNS, interrogando a otros servidores DNS. Cuando hayan obtenido una respuesta, la almacenan temporalmente en su caché de nombres y la devuelven al cliente.

c. Servidor de caché

Su rol ha sido descrito anteriormente. Cuantas más solicitudes diferentes haya gestionado, más eficaz será su labor, porque si la respuesta a la solicitud del cliente se encuentra en su caché y no ha consumido su tiempo de validez, podrá responder directamente sin tener que interrogar a otro servidor DNS.

d. Servidor transitario (forwarder)

Este servidor DNS no trata él mismo las solicitudes que recibe de los clientes, sino que las transfiere a otro servidor DNS. Este último trata la solicitud y responde al servidor de DNS que le ha transferido la solicitud; el servidor transitario puede, entonces, responder a su cliente.

Esta técnica permite a un servidor DNS interno dentro de una organización tomar en cuenta la resolución de los nombres exteriores de la organización, transmitiendo las solicitudes a un servidor DNS externo. Los clientes internos de la organización no tendrán directamente acceso a servidores DNS externos.

3. Gestión de la arborescencia DNS por los servidores DNS

Los servidores DNS reflejan la estructura de la arborescencia del sistema DNS, gracias a las nociones de autoridad y de delegación de autoridad. También se apoyan en los servidores de la raíz de la arborescencia, que desempeñan un rol específico en la resolución de nombres.

a. Autoridad y delegación de autoridad

Para cada dominio, un servidor DNS tiene que ejercer la función de autoridad, es decir tener la responsabilidad de gestionar los nombres que pertenezcan a ese dominio. Sin embargo, un servidor DNS puede dar una delegación de autoridad a un subconjunto (o subdominio) del dominio del que es responsable. Esto permite distribuir la gestión de los nombres, garantizando la coherencia de la arborescencia: un servidor DNS que tiene autoridad sobre un dominio debe conocer todos los nombres del dominio y todos los servidores DNS de los subdominios y de su dominio, servidores a los que ha delegado la autoridad sobre esos subdominios.

La mayoría de las veces, los servidores que tienen autoridad se encuentran duplicados, uno de ellos sería el servidor principal, y el otro el servidor secundario, para asegurar de esta manera la tolerancia a fallos.

b. Los servidores DNS de la raíz (root servers)

Los servidores DNS situados en la raíz de la arborescencia desempeñan un papel particular en la resolución de nombres. En efecto, cuando un servidor DNS recibe una solicitud de resolución para un nombre de dominio sobre el que no tiene autoridad, interrogará a uno de los servidores DNS de la raíz para conocer el servidor del nombre del dominio de primer nivel solicitado. Para ello, debe estar configurado con los nombres y las direcciones IP de sus diferentes servidores de nombres de la raíz.

Ejemplo

Los nombres y direcciones IPv4 de los servidores de la raíz del sistema DNS de Internet están declarados en un archivo que se encuentra por defecto con BIND:

```
.                       518400  IN      NS      a.root-servers.net.
.                       518400  IN      NS      b.root-servers.net.
```

```
.                   518400  IN      NS      c.root-servers.net.
.                   518400  IN      NS      d.root-servers.net.
.                   518400  IN      NS      e.root-servers.net.
.                   518400  IN      NS      f.root-servers.net.
.                   518400  IN      NS      g.root-servers.net.
.                   518400  IN      NS      h.root-servers.net.
.                   518400  IN      NS      i.root-servers.net.
.                   518400  IN      NS      j.root-servers.net.
.                   518400  IN      NS      k.root-servers.net.
.                   518400  IN      NS      l.root-servers.net.
.                   518400  IN      NS      m.root-servers.net.
a.root-servers.net. 518400  IN      A       198.41.0.4
b.root-servers.net. 518400  IN      A       199.9.14.201
c.root-servers.net. 518400  IN      A       192.33.4.12
d.root-servers.net. 518400  IN      A       199.7.91.13
e.root-servers.net. 518400  IN      A       192.203.230.10
f.root-servers.net. 518400  IN      A       192.5.5.241
g.root-servers.net. 518400  IN      A       192.112.36.4
h.root-servers.net. 518400  IN      A       198.97.190.53
i.root-servers.net. 518400  IN      A       192.36.148.17
j.root-servers.net. 518400  IN      A       192.58.128.30
k.root-servers.net. 518400  IN      A       193.0.14.129
l.root-servers.net. 518400  IN      A       199.7.83.42
m.root-servers.net. 518400  IN      A       202.12.27.33
```

c. Los dominios de primer nivel (Top Level Domains)

Los dominios de primer nivel (TLD, *Top Level Domain*) se sitúan justo bajo la raíz del sistema DNS. Al principio no eran muy numerosos y estaban organizados por dominios funcionales: `.com`, `.org`, `.net`, `.int`, `.edu`, `.gov` y `.mil`

Después se añadieron dominios geográficos (por país: `fr`, `us`, `uk`, `de`, etc.), y más tarde se implantaron un gran número de otros identificadores de dominios.

Todos estos dominios están gestionados por servidores DNS, los cuales deben estar declarados como servidores de la raíz y tienen que conocer los servidores DNS de los dominios de nivel inmediatamente inferior, a los que han dado una delegación de autoridad en el subdominio.

d. El dominio de resolución inversa in-addr.arpa.

Esta rama de la arborescencia tiene como objetivo asegurar la resolución de dirección o la resolución inversa. Para ello, las direcciones están estructurados bajo la forma de una arborescencia, en el interior de un dominio llamado `in-addr.arpa` ., un nombre está estructurado en tantos elementos como bytes tenga la dirección, según la sintaxis: `z.y.x.w.in-addr.arpa` .

donde `w`, `x`, `y` y `z` corresponden a los cuatro bytes de una dirección IPv4, en orden inverso.

Ejemplo

Para buscar el nombre DNS de una máquina cuya dirección IP es `20.1.0.250` , escribiremos `250.0.1.20.in-addr.arpa` .

Esta arborescencia está gestionada por los servidores DNS, en paralelo con la arborescencia de los nombres de dominio.

4. Mecanismo de la resolución de nombres

La resolución de nombres es el mecanismo esencial que permite a un cliente DNS obtener la lista de las direcciones IP asociadas un nombre DNS. Si hacemos abstracción de los mecanismos de caché y de los servidores DNS transitorios, el principio de base es el siguiente:

Un cliente DNS tiene que estar configurado para conocer la dirección IP de al menos un servidor de DNS directamente accesible.

Si una aplicación necesita resolver un nombre DNS, solicitará al cliente DNS del sistema (el *resolver*, generalmente una función de la biblioteca de sockets del sistema) que se ocupe de ello.

- El cliente DNS envía una solicitud de resolución de nombre DNS, usando el protocolo de transporte UDP, hacia el puerto bien conocido 53 del primer servidor DNS de su archivo de configuración (`/etc/resolv.conf`). El cliente queda en espera de una respuesta, durante un cierto tiempo. Si no ha recibido ninguna respuesta después de este tiempo, puede interrogar al servidor DNS siguiente, si

tiene otro más declarado en el archivo de configuración.

- El servidor DNS trata la solicitud:

  Si tiene autoridad en el dominio solicitado, consulta la base de nombres, bien sea para encontrar directamente el nombre solicitado, bien para determinar el servidor DNS al que ha delegado la autoridad para ese subdominio. Responde directamente al cliente o transmite la solicitud al servidor DNS del subdominio, que le responderá con la lista de las direcciones que corresponden al FQDN (o un mensaje de tipo `NOT FOUND`, si ese nombre no está referenciado).

  Si no tiene autoridad, el servidor DNS efectuará una búsqueda **recursiva**: interroga a uno de los servidores de la raíz DNS, que le responderá con la lista de servidores de nombres del dominio de primer nivel correspondiente al FQDN buscado.

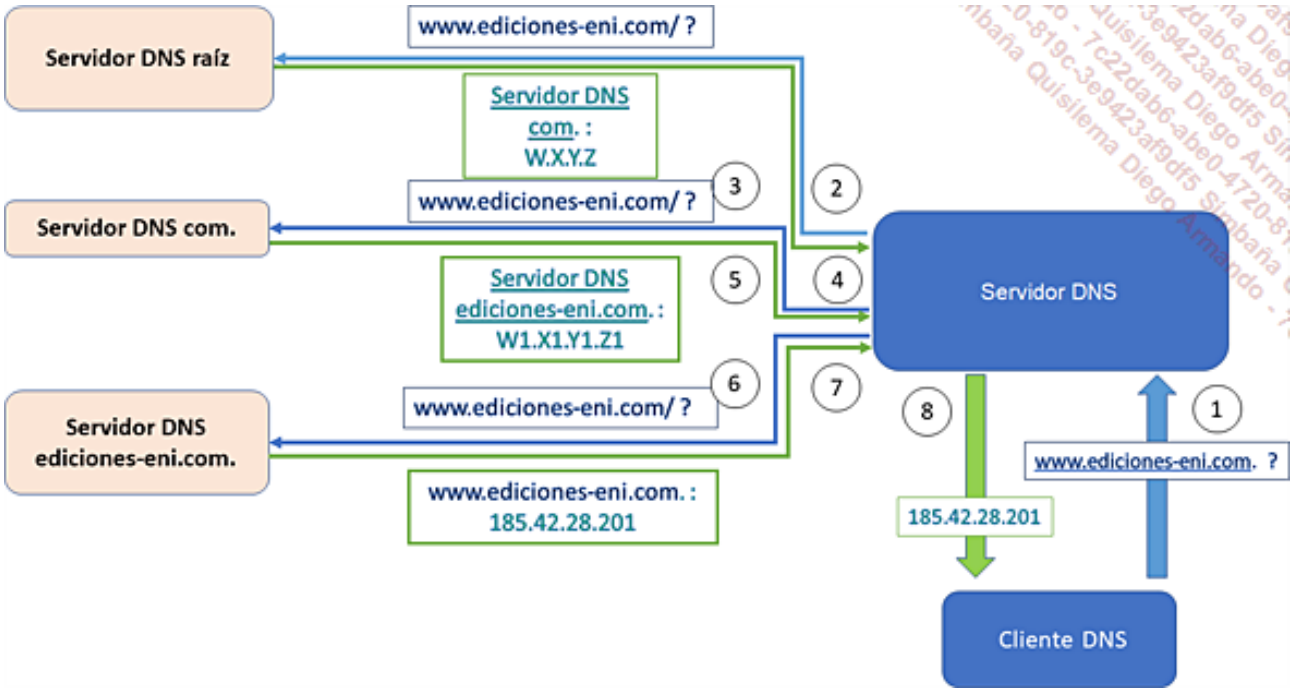
  A continuación, el servidor va a reiterar su solicitud, nivel por nivel, hasta encontrar el servidor DNS que tenga autoridad en el subdominio buscado, que le devolverá la lista de las direcciones correspondientes al FQDN (o un mensaje de tipo `NXDOMAIN`, si el nombre no está referenciado).

- Finalmente, el servidor DNS dará la respuesta al cliente.

Cuando un servidor DNS ha obtenido una respuesta a su solicitud, la guardará en su caché, para poder utilizarla más tarde si recibe la misma solicitud. La duración de validez de una respuesta en el caché está configurada por un parámetro asociado al registro de la relación del nombre de do-minio/dirección, llamada TTL (*Time To Live*).

a. Ejemplo

Solicitud de resolución del nombre DNS www.ediciones-eni.com:



1. El cliente DNS interroga a su servidor DNS.
2. El servidor DNS envía la solicitud a un servidor DNS de la raíz.
3. El servidor DNS de la raíz contesta con la dirección de un servidor DNS del dominio `com`.
4. El servidor DNS envía la solicitud al servidor DNS del dominio `com`.
5. El servidor DNS del dominio `com`. contesta con la dirección de un servidor DNS del dominio `ediciones-eni.com` .
6. El servidor DNS envía la solicitud al servidor DNS del dominio `ediciones-eni.com` .
7. El servidor DNS del dominio `ediciones-eni.com` contesta con la dirección asociada al nombre DNS solicitado.
8. El servidor DNS responde a la solicitud del cliente.

5. Zonas DNS

La información acerca de los objetos gestionados por un servidor DNS están almacenadas en un archivo llamado archivo de **zona**. Cada zona constituye un conjunto de elementos bajo la autoridad de un servidor DNS principal, que puede opcionalmente transmitir una copia de este archivo, en solo lectura, a uno o a varios servidores secundarios

En el archivo de zona están referenciados los nombres de los hosts y los nombres de servicios asociados a direcciones IP. También encontramos la información referente a los

servidores de nombres de la zona y, si fuera necesario, los servidores de nombres que tienen una delegación en los subdominios de esta zona.

Los archivos específicos de zona permiten gestionar la resolución inversa, de direcciones IP hacia nombres DNS.

Los archivos de zona están constituidos por registros de diferentes tipos, según la naturaleza de la información almacenada.

a. Registro de zona (tipo SOA)

Este registro define la zona y sus atributos, entre los que podemos encontrar:

- Nombre de la zona.
- FQDN del servidor primario.
- Correo electrónico del administrador.
- Número de serie (para gestionar las actualizaciones de los servidores secundarios).

b. Registros de recursos

Los elementos relativos a los nombres de host o de servicio y las direcciones IP correspondientes se llaman registros de recursos (RR, *Resource Record*). Pueden ser de distintos tipos. Los tipos principales están descritos a continuación.

Registro de dirección (tipo A o AAAA)

Estos registros contienen la información relativa a los nombres de host y su dirección IP. El tipo `A` corresponde a las direcciones IPv4, el tipo `AAAA` a las direcciones IPv6.

Registro de puntero (tipo PTR)

Este tipo de registro mantiene el enlace entre una dirección IP y un nombre DNS. Lo encontramos en los archivos de zona de búsqueda inversa.

Registro de alias (tipo CNAME)

Este tipo de registro (*Canonical name*) permite hacer una correspondencia entre un

nombre (*alias*) y un nombre FQDN.

Registro de servidor de nombres (tipo NS)

Este tipo de registro (*Name server*) permite declarar los servidores de nombres de un dominio o de un subdominio.

c. Zona de resolución inversa in-addr.arpa.

Esta zona gestiona la información que permite efectuar la resolución inversa de las direcciones de los nombres de dominio de la zona. Corresponde a un subdominio del dominio `in-addr.arpa.` , para la red/subred de las direcciones IP de la zona de nombres.

Un archivo de zona inversa contiene esencialmente registros de recursos de tipo PTR, que realizan el vínculo entre una dirección IP en notación inversa `in-addr.arpa` . y un nombre DNS.

6. Los servidores DNS en Linux

Varios programas implementan un servidor DNS en Linux, el más usado es BIND (*Berkeley Internet Name Domain*).

a. BIND

Concebido por la Universidad de Berkeley para Unix BSD en los años 80, BIND es uno de los primeros servidores DNS. Está mantenido y desarrollado por el ISC (*Internet Systems Consortium*), un consorcio público sin ánimo de lucro.

Su versión actual es la número 9, reescrita completamente al principio de los años 2000 para implementar una arquitectura más fácil de proteger. Es el servidor de nombres DNS más utilizado hoy día, particularmente en Linux.

Tenemos que conocer bien el servidor BIND para la certificación.

b. Otros servidores DNS

Otros tipos de servidores DNS se usan en entornos DNS, entre ellos:

- Dnsmasq: este programa ofrece un peculiar doble servidor DHCP y DNS transitario (*forwarder*). Lo encontramos implementado a menudo en equipos de tipo router de Internet, porque permite distribuir las direcciones IP privadas para las máquinas del abonado (a través del servidor DHCP) y dar acceso a los servidores DNS de Internet a través del servidor DNS.
- Djbdns: desarrollado desde 2001 por Daniel J. Bernstein (el creador del servidor de mensajería Gmail), este programa tiene como objetivo resolver múltiples problemas de seguridad encontrados en BIND. Está aconsejado por algunas distribuciones Linux. Se ha efectuado un fork bajo el nombre de dbndns, en el marco del proyecto Debian.
- PowerDNS: programa propietario creado en 1993, se ha convertido en open source en 2003. Provee servicios DNS repartidos entre múltiples servidores con equilibrado de carga (*load balancing*).

7. Configuración básica de un servidor primario DNS BIND

El servidor BIND es ejecutado por el daemon `named`. Normalmente se inicia durante el arranque del sistema y su archivo de configuración por defecto es `/etc/named.conf` (o `/etc/bind/named.conf`). Su configuración es más o menos compleja en función de su o de sus roles: servidor primario, secundario, servidor de caché y/o servidor transitario. En esta parte, vamos a ver la configuración básica de un servidor DNS primario simple.

El firewall de los sistemas Linux está, a menudo, configurado por defecto para bloquear las conexiones entrantes en los puertos usados por DNS, 53 UDP y 53 TCP. Hay que autorizar los mensajes entrantes hacia esos puertos para que el servidor DNS pueda recibir las solicitudes de los clientes o de otros servidores DNS.

a. El archivo named.conf

El archivo de configuración `named.conf` está generalmente completado por archivos incluidos a través de la directiva `include`.

Este archivo define las opciones del servidor BIND y su rol, así como la ubicación de los diferentes archivos de zona gestionados por ese servidor (ya que se trata de un servidor principal), y el archivo de declaración de los servidores DNS de la raíz de la arborescencia.

Un servidor primario gestiona al menos una zona para los nombres de dominio o de subdominio en los que ejerce la autoridad, así como la zona de búsqueda inversa correspondiente.

Estos archivos de zona están declarados en el archivo `named.conf` o en uno de los archivos incluidos, con su ubicación y tipo.

Un archivo de configuración `named.conf` contiene un conjunto de directivas, las más frecuentes están descritas a continuación.

La directiva `include`

Esta directiva permite incluir un archivo en el archivo de configuración principal.

La directiva `options`

La directiva `options` define un conjunto de opciones según la sintaxis siguiente:

```
options { <opción>; [<opción>; ...] };
```

Entre las numerosas opciones existentes, las principales son:

`allow-query {lista};`	Lista de los clientes autorizados a interrogar al servidor sobre sus zonas de autoridad (por defecto: todos).
`allow-query-cache {lista};`	Lista de los clientes autorizados a interrogar al servidor sobre su caché (por defecto: todos).
`blackhole {lista};`	Lista de clientes prohibidos (por defecto: ninguno).
`directory Path;`	Directorio de los archivos de datos (por defecto: `/var/named/`).
`forwarders {lista};`	Lista de los servidores DNS hacia los que hay que transferir las solicitudes que queden fuera de las zonas de autoridad.
`forward Tipo;`	Tipo de *forwarding* (`first`, `only`) en el caso de un servidor DNS transitario.
`listen-on {lista};`	Direcciones IPv4 de las interfaces en las que el servidor espera las solicitudes (por defecto: todas).
`listen-on-v6 {lista};`	Direcciones IPv6 de las interfaces en las que el servidor espera las solicitudes (por defecto: todas).
`pid-file Path;`	Camino de acceso hacia el archivo que contiene el PID del daemon `named`.
`recursion yes|no;`	Especifica si el servidor gestiona las solicitudes recursivas (por defecto: `yes`).
`statistics-file NombreArchivo;`	Camino de acceso hacia el archivo de estadísticas (por defecto: `/var/named/named.stats`).

La directiva `zone`

Esta directiva especifica el nombre de dominio o de subdominio que corresponde a la zona (el punto final es facultativo), sus características y la ubicación del archivo de zona correspondiente.

Los principales atributos son:

`file Nombre;`	Nombre del archivo de zona (en el directorio de los archivos del servidor).
`type Tipo;`	Tipo de zona: `hint` (zona especial para los servidores raíz), `master` (servidor principal) y `slave` (servidor secundario).
`masters {Dirección};`	Dirección del servidor principal para esta zona (en el caso de que haya un servidor secundario).
`allow-query {lista};`	Lista de los clientes autorizados a efectuar solicitudes sobre esta zona (por defecto: todos).
`allow-transfer {lista};`	Lista de los servidores secundarios autorizados a solicitar la transferencia del archivo de zona (por defecto: todos).
`allow-update {lista};`	Lista de los hosts autorizados a solicitar una actualización dinámica en el archivo de zona (por defecto: ninguno).

Ejemplo

Archivo `/etc/named.conf` (simplificado) en una distribución CentOS 8:

```
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
	listen-on port 53 { 127.0.0.1; };
	listen-on-v6 port 53 {::1; };
	directory 	"/var/named";
	dump-file 	"/var/named/data/cache_dump.db";
	statistics-file "/var/named/data/named_stats.txt";
	allow-query 	{ localhost; };
	/*
	 - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
	 - If you are building a RECURSIVE (caching) DNS server, you need to enable
	   recursion.
	 - If your recursive DNS server has a public IP address, you MUST enable access
	   control to limit queries to your legitimate users. Failing to do so will
	   cause your server to become part of large scale DNS amplification
	   attacks. Implementing BCP38 within your network would greatly
	   reduce such attack surface
	*/
	recursion yes;
	pid-file "/run/named/named.pid";
};
zone "." IN {
	type hint;
	file "named.ca";
};
include "/etc/named.rfc1912.zones";
```

El servidor tiene el rol de servidor de caché y no gestiona ninguna zona, excepto la zona "`.`" que contiene las direcciones y los nombres de los servidores DNS de la raíz, así como las zonas correspondientes a la red de loopback (`named.rfc1912.zones`). Escucha en el

puerto bien conocido 53, solamente para las solicitudes de la máquina local (dirección loopback `127.0.0.1` *). La opción* `recursion yes` *lo autoriza a efectuar solicitudes de resolución recursiva, interrogando sucesivamente servidores DNS hasta obtener una respuesta de un servidor DNS que tenga autoridad sobre el dominio del nombre buscado.*

b. Los archivos de zona por defecto

El paquete de software `BIND` proporciona generalmente varios archivos de zona.

Un archivo de zona corresponde a la raíz del sistema DNS (zona "`.`"), en el que están declarados los diferentes servidores de nombres de la raíz del sistema DNS Internet.

Si su servidor DNS es puramente interno en su organización (es decir que no tiene acceso a Internet), hay que modificar el contenido de esta zona para declarar los servidores DNS de la raíz de su arborescencia DNS privada.

Dos archivos de zona corresponden a los nombres de hosts de la red de loopback así como a sus direcciones (zona de búsqueda inversa): zone "`localhost` " y zone "`127.in-addr.arpa` ".

Los dos otros archivos de zona de búsqueda inversa corresponden a las direcciones de *broadcast* y al identificador de la red IP local: zone "`0.in-addr.arpa` " y zone "`255.in-addr.arpa` ".

c. Ejemplo

Archivo de configuración de un servidor primario BIND, ejerciendo autoridad sobre la zona del dominio `lpic2test.com.`

```
options {
    listen-on port 53 {127.0.0.1;192.168.0.60;};
```

```
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query {127.0.0.1;192.168.0.0/24;};
        recursion yes;
        pid-file "/run/named/named.pid";
};
zone "." IN {
        type hint;
        file "named.ca";
};
zone "lpic2test.com" IN {
        type master;
        file "lpic2.zone";
};
zone "0.168.192.in-addr.arpa" IN {
        type  master;
        file  "db.192.168.0";
};
include "/etc/named.rfc1912.zones";
```

8. Configuración de un servidor de caché o transitario

Algunos servidores DNS no gestionan ninguna parte de la arborescencia del sistema DNS al que pertenecen. Su único rol es el de responder a las solicitudes de sus clientes. Se trata de servidores de caché y/o servidores transitarios (*forwarder*).

a. Configuración de servidor de caché

Un servidor DNS de caché asegura la resolución de nombres, pero no gestiona ninguna zona.

Este efectúa búsquedas recursivas en los servidores DNS de la arborescencia, para responder a las solicitudes de sus clientes, y almacena las respuestas obtenidas, temporalmente, en su caché. De esta manera podrá responder directamente si una

solicitud es relativa a un nombre o a una dirección (resolución inversa) que ya se encuentra en su caché, con la condición de que la vida de la información no haya expirado (TTL).

La opcion `recursion yes` permite al servidor BIND desempeñar el rol de servidor de caché.

Ejemplo

Archivo de configuración de un servidor de caché BIND.

```
/* servidor DNS de caché */
options {
    listen-on port 53 {127.0.0.1;192.168.0.60;};
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query {127.0.0.1;192.168.0.0/24;};
    recursion yes;
    pid-file "/run/named/named.pid";
};
zone "." IN {
    type hint;
    file "named.ca";
};
include "/etc/named.rfc1912.zones";
```

Exceptuando la zona de los servidores DNS de la raíz y las zonas por defecto, no hay ninguna otra zona declarada.

b. Configuración un servidor transitario (forwarder)

Un servidor transitario (*forwarder*) no efectúa directamente búsquedas recursivas, si no que transfiere las solicitudes de resolución de sus clientes hacia uno o varios servidores DNS que se encargarán de efectuar las solicitudes de resolución recursiva.

Un servidor puede hacer ese rol para todas las solicitudes, o solamente para las solicitudes relativas a una parte de la arborescencia. Puede gestionar un dominio privado,

interno, y difundir hacia servidores DNS externos para los dominios externos (los de Internet, en particular).

El servidor transitario almacena las respuestas obtenidas temporalmente en su caché. De esta manera puede responder directamente si una solicitud es relativa a un nombre o a una dirección que se encuentra en su caché, siempre y cuando la duración de vida de la información no haya llegado a expiración (TTL).

La opción `forwarders` permite especificar el o los DNS a los que hay que transferir las solicitudes de los clientes DNS.

Ejemplo

```
/* Servidor DNS transitario */
options {
    listen-on port 53 {127.0.0.1;192.168.0.60;};
    directory      "/var/named";
    dump-file      "/var/named/data/cache_dump.db";
    statistics-file "/var/named/data/named_stats.txt";
    memstatistics-file "/var/named/data/named_mem_stats.txt";
    allow-query {127.0.0.1;192.168.0.0/24;};
    pid-file "/run/named/named.pid";
    forwarders {192.168.0.254;};
    recursion yes;
};
include "/etc/named.rfc1912.zones";
```

Exceptuando las zonas por defecto, no hay ninguna otra zona declarada. El servidor DNS que efectuará las solicitudes a los diferentes servidores DNS de la arborescencia DNS tiene la dirección IP `192.168.0.254` .

9. Monitorización de un servidor DNS BIND

El daemon `named` es arrancado por un script de inicio `init system V` o por `systemd` . Si su configuración cambia, habría que reiniciarlo o pedirle que vuelva a cargar su configuración. Para ello, podemos usar el script de inicio o `systemd` . También

podemos enviar la señal `HUP` (señal `1`) al daemon a través del comando `kill`.

El comando `rndc`, proporcionado con el paquete `BIND`, también permite interactuar con el daemon.

a. Recarga de la configuración del servidor

Si el daemon `named` ha sido lanzado con un script de inicio, le podemos pedir que vuelva a cargar su configuración gracias al argumento `reload` del script.

Ejemplo

Con una distribución Debian 10:

```
/etc/init.d/bind9 reload
[ ok ] Reloading bind9 configuration (via systemctl): bind9.service.
```

Si el daemon `named` ha sido lanzado con `systemd`, podemos pedirle que vuelva a cargar su configuración gracias al argumento `reload` del comando `systemctl`.

Ejemplo

Con una distribución Debian 10:

```
systemctl reload bind9
```

Con una distribución CentOS 8:

```
systemctl reload named
```

Otro método consistiría en enviar una señal 1 (`HUP`) al proceso que ejecuta `named`, lo que provocará que recargue su configuración.

Ejemplo

El PID del proceso que ejecuta el servidor BIND está almacenado en el archivo definido por

la opción `pid-file` del archivo `named.conf` :

```
grep pid-file /etc/named.conf
    pid-file "/run/named/named.pid";
cat /run/named/named.pid
14565
kill -HUP 14565
```

b. Control del archivo de configuración: named-checkconf

El comando `named-checkconf` permite comprobar la sintaxis del contenido del archivo de configuración `named.conf` .

El comando devuelve `0` si la sintaxis del contenido del archivo es correcta, en el caso contrario muestra `1`, con un mensaje de error.

Ejemplo

```
named-checkconf
echo $?
0
```

El archivo de configuración es correcto desde el punto de vista de la sintaxis.

Quitamos un `;` al final de una línea de directiva:

```
options {
    listen-on port 53 { 127.0.0.1; }
    listen-on-v6 port 53 {::1; };
named-checkconf
/etc/named.conf:12: missing ';' before 'listen-on-v6'
echo $?
1
```

El comando muestra un error y devuelve `1`.

c. El comando rndc

El comando `rndc` permite interactuar con el daemon `named`.

Sintaxis

```
rndc comando [parámetros]
```

Principales subcomandos

`-s Servidor`	Interactúa con el daemon `named` del servidor especificado.
`reload`	Recarga los archivos de configuración y la información de zona.
`reload Zona ArchivoZona`	Recarga el archivo de la zona indicada.
`retransfer Zona`	Fuerza la transferencia de la zona indicada desde el servidor principal.
`reconfig`	Carga solamente los archivos de las nuevas zonas.
`flush`	Vacía la caché del servidor.
`status`	Muestra el estado del servidor.
`dumpdb`	Copia el contenido de la caché en el archivo de dump configurado por la opción `dump-file` en `named.conf`.

Ejemplo

```
rndc status
version: BIND 9.11.4-P2-RedHat-9.11.4-26.P2.el8 (Extended Support Version)
<id:7107deb>
running on centos8: Linux x86_64 4.18.0-147.5.1.el8_1.x86_64 #1 SMP Wed
Feb 5 02:00:39 UTC 2020
boot time: Mon, 04 May 2020 13:07:21 GMT
last configured: Tue, 05 May 2020 15:21:38 GMT
configuration file: /etc/named.conf
CPUs found: 2
worker threads: 2
UDP listeners per interface: 1
number of zones: 104 (97 automatic)
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/900/1000
tcp clients: 2/150
server is up and running
```

Recargamos la configuración:

```
rndc reload
server reload successful
rndc status
version: BIND 9.11.4-P2-RedHat-9.11.4-26.P2.el8 (Extended Support Version)
<id:7107deb>
running on centos8: Linux x86_64 4.18.0-147.5.1.el8_1.x86_64 #1 SMP Wed
Feb 5 02:00:39 UTC 2020
boot time: Mon, 04 May 2020 13:07:21 GMT
last configured: Tue, 05 May 2020 15:26:32 GMT
configuration file: /etc/named.conf
CPUs found: 2
worker threads: 2
UDP listeners per interface: 1
```

```
number of zones: 7 (0 automatic)
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/900/1000
tcp clients: 2/150
server is up and running
```

Observando la hora de la última configuración, constatamos que el servidor `named` ha cargado la configuración correctamente.

El comando se usa para administrar servidores DNS remotos, con la la opción `-S Servidor`. Sin embargo, por razones de seguridad, es necesario que este uso sea configurado en los dos servidores, con un intercambio de claves TSIG. Esta configuración puede hacerse con el comando `rndc-confgen`.

10. Prueba de un servidor DNS BIND

Para comprobar el funcionamiento correcto de un servidor DNS, podemos usar diferentes comandos. En primer lugar, vamos a usar los comandos `host` y `dig` para hacer comprobaciones básicas.

a. El comando host

El comando `host` permite interrogar a un servidor DNS para resolver un nombre o una dirección.

Sintaxis

```
host [Opciones] Nombre|Dirección [ DirServidor ]
```

Parámetros principales

`-v`	Visualización detallada de la solicitud y de la respuesta.
`-4`	Usar IPv4.
`-6`	Usar IPv6.
`-t Tipo`	Tipo de registro que se buscará (`A`, `AAAA`, `MX`, `NS`, `PTR`…).
`Nombre|Dirección`	Nombre o dirección que se tiene que resolver.
`DirecciónServidor`	Dirección del servidor DNS, si no estuviera especificada, se usará: `/etc/resolv.conf` .

Ejemplo

```
host www.google.com
www.google.com has address 216.58.204.132
www.google.com has IPv6 address 2a00:1450:4007:812::2004
host 216.58.204.132
132.204.58.216.in-addr.arpa domain name pointer par21s05-in-f4.1e100.net.
132.204.58.216.in-addr.arpa domain name pointer par21s05-in-f132.1e100.net.
```

b. El comando dig

El comando `dig` (*Domain Information Groper*) permite comprobar de manera detallada el funcionamiento de un servidor DNS. Dispone de muchas opciones para gestionar

diferentes aspectos de la resolución de nombres.

En esta parte del capítulo, presentamos un uso básico del comando.

En las distribuciones de tipo Debian, el comando forma parte del paquete `dnsutils`.

Sintaxis

```
dig   [ Opciones ] [@DirServidor] Nombre|Dirección
```

Parámetros principales

`-t Tipo`	Tipo de registro que se buscará (`A`, `AAAA`, `MX`, `NS`, `PTR`…).
`-x`	Búsqueda inversa para la dirección IP específica como argumento.
`-4`	Usar IPv4.
`-6`	Usar IPv6.
`Nombre|Dirección`	Nombre o dirección que se tiene que resolver.
`@DirServidor`	Dirección del servidor DNS, si no estuviera especificada, se usará: `/etc/resolv.conf`.

Ejemplo

dig www.google.com

```
; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el8 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 44428
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;www.google.com.                IN      A

;; ANSWER SECTION:
www.google.com.         298     IN      A       216.58.201.228

;; Query time: 1 msec
;; SERVER: 192.168.0.254#53(192.168.0.254)
;; WHEN: mar. mayo 05 17:10:50 BST 2020
;; MSG SIZE  rcvd: 59
```

dig -t PTR 229.201.58.216.in-addr.arpa.

```
; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el8 <<>> -t PTR
229.201.58.216.in-addr.arpa.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18296
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1472
;; QUESTION SECTION:
;229.201.58.216.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
229.201.58.216.in-addr.arpa. 40032 IN   PTR     fra02s18-in-f5.1e100.net.
229.201.58.216.in-addr.arpa. 40032 IN   PTR     par10s33-in-f5.1e100.net.

;; Query time: 2 msec
```

```
;; SERVER: 192.168.0.254#53(192.168.0.254)
;; WHEN: mar. mayo 05 17:15:29 BST 2020
;; MSG SIZE  rcvd: 123
```

Búsqueda inversa directamente a través de una dirección:

```
dig -x 216.58.201.228
; <<>> DiG 9.11.4-P2-RedHat-9.11.4-26.P2.el8 <<>> -x 216.58.201.228
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 50451
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 9

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 9080499845c2cd258fb102e95eb905fa80af16b731e9ac0d (good)
;; QUESTION SECTION:
;228.201.58.216.in-addr.arpa.   IN     PTR

;; ANSWER SECTION:
228.201.58.216.in-addr.arpa. 86400 IN   PTR    par10s33-in-f4.1e100.net.
228.201.58.216.in-addr.arpa. 86400 IN   PTR    fra02s18-in-f4.1e100.net.

;; AUTHORITY SECTION:
201.58.216.in-addr.arpa. 85877  IN     NS     ns4.google.com.
201.58.216.in-addr.arpa. 85877  IN     NS     ns1.google.com.
201.58.216.in-addr.arpa. 85877  IN     NS     ns2.google.com.
201.58.216.in-addr.arpa. 85877  IN     NS     ns3.google.com.

;; ADDITIONAL SECTION:
ns4.google.com.      170927  IN     A      216.239.38.10
ns2.google.com.      170927  IN     A      216.239.34.10
ns1.google.com.      170927  IN     A      216.239.32.10
ns3.google.com.      170927  IN     A      216.239.36.10
ns4.google.com.      170927  IN     AAAA   2001:4860:4802:38::a
ns2.google.com.      170927  IN     AAAA   2001:4860:4802:34::a
ns1.google.com.      170927  IN     AAAA   2001:4860:4802:32::a
ns3.google.com.      170927  IN     AAAA   2001:4860:4802:36::a
```

```
;; Query time: 7 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: lun. mayo 11 09:59:54 CEST 2020
;; MSG SIZE  rcvd: 409
```