

Texto de documentación sobre el sistema DNS y la configuración del servidor DNS Bind9 bajo sistemas GNU/Linux.

Esta documentación incluye, básicamente, ejemplos de comandos y configuraciones y explicaciones de esos ejemplos.

© 2011 Manuel Méndez - Manuko
Documento bajo Licencia de Documentación Libre GNU/FDL
Última revisión: 20110212-01:33

Documentación DNS y Bind

Internet se rige por un sistema de direccionamiento y uno de paquetería y entrega de información, llamados TCP/IP. Sin embargo, con direccionamiento IP estamos limitados a un servicio por puerto TCP, de modo que no podemos, por ejemplo, tener más de una web por máquina.

Por eso, y porque las direcciones IP, queramos o no, son números, y quedan muy feos para aprenderse, utilizamos DNS (Domain Name Service), un servicio que nos permite crear zonas de dominio con un nombre específico y comunicarnos entre distintas zonas dentro de una global controlada por 13 servidores de primer nivel (*root servers*).

Además, DNS nos permite redirigir una IP pública a diversos servicios virtuales funcionando a través del mismo puerto, mediante un filtro por nombre completo de dominio (FQDN o Fully Qualified Domain Name), creando subdominios.

Por ejemplo, un nombre de dominio puede ser instigado.net, y podemos tener varios subdominios, 1.instigado.net, 2.instigado.net, 3.instigado.net, etcétera, y utilizar o bien enlaces dentro del servidor DNS, por CNAME a las distintas carpetas dentro de instigado.net (instigado.net/1/; instigado.net/2/; etcétera), o bien redirigir todas a la misma IP directamente, y utilizar servidores virtuales dentro del servidor web o del servidor de correo para manejar las direcciones dentro del mismo equipo. De este modo, tenemos varias direcciones completas para distintos servicios dentro de la misma IP.

El caso que tratamos hoy es el de montar un servidor DNS en nuestro propio ordenador, utilizando Bind9 sobre GNU/Linux (en principio, la documentación sale de CentOS pero será revisada para responder a cualquier tipo de servidor GNU/Linux basado en Red Hat o Debian).

Índice

- [Configurar IP fija de forma permanente](#)
 - [Averiguar puerta de enlace](#)
 - [Red Hat/CentOS](#)
 - [Debian/Ubuntu](#)
- [Archivos de configuración DNS en Linux](#)
- [Estructura esquemática de una consulta DNS](#)
- [Herramientas de consulta DNS](#)
 - [Consultas con Dig](#)
 - [Consultas con host](#)
- [Tipos de registros DNS](#)
- [Instalar un servidor Bind9](#)
 - [En Red Hat/CentOS](#)
 - [En Debian/Ubuntu](#)
- [Ejemplo de archivo de zona](#)

Configurar IP fija de forma permanente.

Averiguar puerta de enlace

Si resulta que no conocemos la red y está configurada con DHCP, necesitaremos saber nuestra puerta de enlace. Para ello, tenemos dos opciones: *netstat -rn* o *ip route...*

```
[root@localhost ~]# netstat -rn
Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.1.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
0.0.0.0 192.168.1.1 0.0.0.0 UG 0 0 0 eth0
[root@localhost ~]# ip route
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.118
169.254.0.0/16 dev eth0 scope link
default via 192.168.1.1 dev eth0
```

[Volver al índice](#)

Red Hat/CentOS

Para configurar nuestra IP fija en CentOS, y que se mantenga siempre la misma al iniciar el equipo, utilizaremos cualquiera de las aplicaciones que ofrece System Config de Red Hat:

- *system-config-network-cmd*: Versión de comandos. Nos permite sacar toda la configuración de red a un fichero, modificar ese fichero con un editor de texto, e importarlo posteriormente, así:

```
[root@localhost linuxero]# system-config-network-cmd > network
[root@localhost linuxero]# vim network // Lo editamos y lo configuramos como queremos
[root@localhost linuxero]# system-config-network-cmd -i -c -f network
```

- *system-config-network-gui*: Versión gráfica, para hacerlo directamente con el ratón en una ventana.
- *system-config-network-tui*: Versión de menú de texto, nos presenta una ventana ncurses para poder configurar fácilmente los dispositivos de red en modo texto.

[Volver al índice](#)

Debian/Ubuntu

Para configurar nuestra IP fija en un servidor Debian o Ubuntu Server, deberemos modificar el fichero */etc/network/interfaces*, borrando las líneas referentes a la tarjeta de red que queramos poner con IP fija - en este caso eth0 -, poniendo unas líneas de IP fija como estas:

```
iface eth0 inet static
address 192.168.0.130
netmask 255.255.255.0
gateway 192.168.0.1
auto eth0
```

De ese modo, cuando arranque la máquina tendremos la IP 192.168.0.130, y la puerta de enlace será 192.168.0.1. Para que se fije la IP sin necesidad de reiniciar, podemos ejecutar la recarga del servicio de red:

```
root@ekip:/etc/bind# service networking restart
```

En instalaciones Desktop de Ubuntu, e instalaciones Debian con NetworkManager, tendremos que utilizar las aplicaciones gráficas de NetworkManager para configurar la IP fija, o bien desinstalar NetworkManager y gestionar la red con el servicio networking.

Archivos de configuración del cliente DNS en Linux.

Comunmente el archivo en el que fijamos los servidores de nombres que deberá utilizar nuestro ordenador para resolver nombres de dominio es `/etc/resolv.conf`, con una sintaxis similar a esta:

```
[root@localhost linuxero]# cat /etc/resolv.conf
nameserver 80.58.61.250
nameserver 80.58.61.254
search ejemplo.com
```

Como vemos, tenemos dos `nameserver`, que son los servidores DNS a los que nuestro ordenador consultará para resolver nombres. También tenemos un `search` de nuestro propio dominio, un dominio propio para búsquedas en otros servidores si no se encuentra un nombre corto perteneciente a un `fqdn`, es decir, si buscamos al `host 1` dentro de nuestra red, y no lo encontramos, se solicitará a los servidores DNS la resolución de `1.ejemplo.com`.

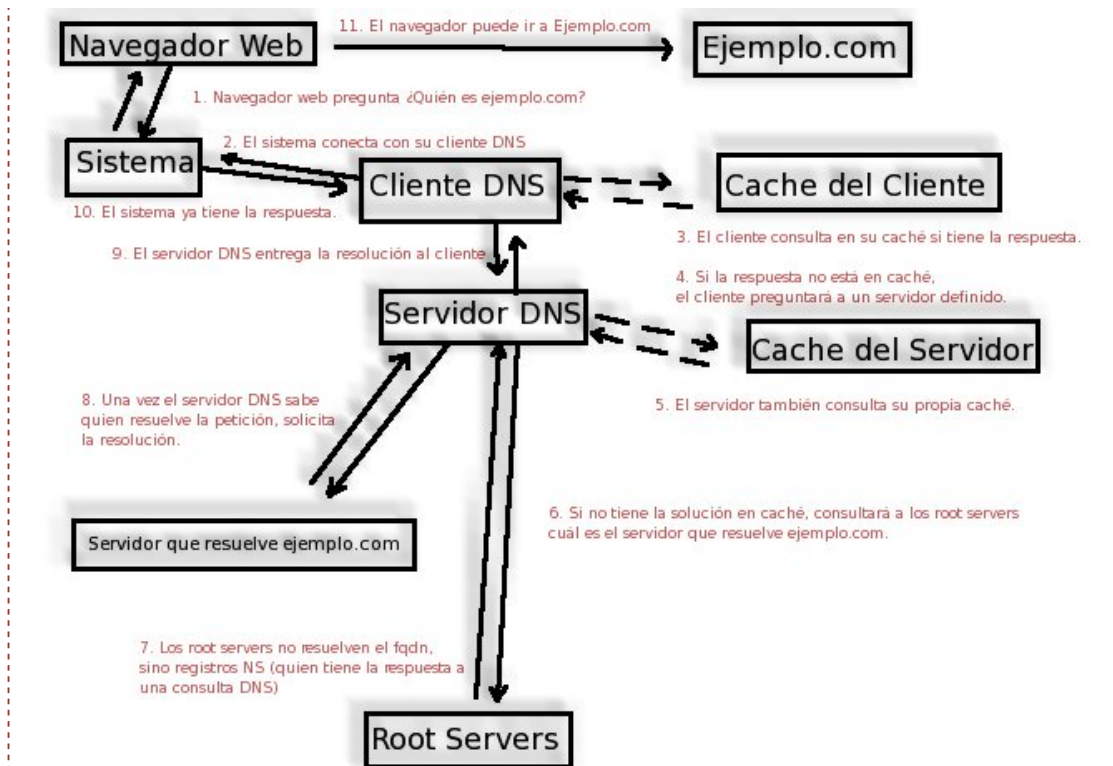
El siguiente ejemplo es el más común cuando tenemos un servidor DNS Bind9 funcionando en nuestra propia máquina:

```
[root@localhost linuxero]# cat /etc/resolv.conf
search ejemplo.com
nameserver 127.0.0.1
```

[Volver al índice](#)

Estructura esquemática de una consulta DNS

Las consultas son tal y como se describe en la siguiente imagen:



La lista que puede leerse de acciones dentro de la imagen es esta:

- 1. Navegador web pregunta: ¿Quién es ejemplo.com?
- 2. El sistema conecta con el cliente DNS interno.
- 3. El cliente consulta en su caché si tiene la respuesta.
- 4. Si la respuesta no está en caché, el cliente preguntará a un servidor definido.
- 5. El servidor también consulta su propia caché.
- 6. Si no tiene la solución en caché, consultará a los root servers cuál es el servidor que resuelve ejemplo.com
- 7. Los root servers no resuelven el FQDN (Full Qualified Domain Name) sino solo registros NS según zona (quién tiene la respuesta a la consulta).
- 8. Una vez nuestro servidor DNS sabe quién resuelve la petición, solicita la resolución.
- 9. El servidor DNS entrega la respuesta al cliente.
- 10. El sistema ya tiene respuesta.
- 11. El navegador ya puede ir a ejemplo.com

[Volver al índice](#)

Herramientas de consulta DNS

Tenemos principalmente tres herramientas para consultar registros DNS relacionados con un nombre de dominio, como el servidor en el que se aloja la zona (registro NS), la redirección DNS para correo electrónico (registro MX), o cualquier otro registro como los de definición de servidor (registros A).

Estas tres herramientas son host, dig y nslookup, si bien vamos a olvidarnos de esta última, pues es actualmente obsoleta. Así pues, vamos a hacer consultas con dig y host...

Consultas con dig

```
[root@localhost ~]# dig google.com

;<<> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.e15_5.3 <<> google.com
;; global options: printcmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 25200
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 4, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                IN      A

;; ANSWER SECTION:
google.com.                48      IN      A       209.85.146.147
google.com.                48      IN      A       209.85.146.99
google.com.                48      IN      A       209.85.146.103
google.com.                48      IN      A       209.85.146.104
google.com.                48      IN      A       209.85.146.105
google.com.                48      IN      A       209.85.146.106

;; AUTHORITY SECTION:
google.com.                169910  IN      NS      ns2.google.com.
google.com.                169910  IN      NS      ns3.google.com.
google.com.                169910  IN      NS      ns4.google.com.
google.com.                169910  IN      NS      ns1.google.com.

;; Query time: 174 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Feb 11 19:09:58 2011
;; MSG SIZE rcvd: 196
```

Como vemos, dig nos escupe por pantalla toda la información de una zona DNS definida según el nombre de dominio que hemos especificado. Con -t podemos especificar qué registros queremos ver, por ejemplo:

Registros NS:

```
[root@localhost ~]# dig -t ns google.com

;<<> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.e15_5.3 <<> -t ns google.com
;; global options: printcmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 49885
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;google.com.                IN      NS

;; ANSWER SECTION:
google.com.                278549  IN      NS      ns4.google.com.
google.com.                278549  IN      NS      ns1.google.com.
google.com.                278549  IN      NS      ns2.google.com.
google.com.                278549  IN      NS      ns3.google.com.

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Feb 11 19:13:38 2011
;; MSG SIZE rcvd: 100
```

Registros MX

```
[root@localhost ~]# dig -t mx google.com

;<<> DiG 9.3.6-P1-RedHat-9.3.6-4.P1.e15_5.3 <<> -t mx google.com
;; global options: printcmd
;; Got answer:
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 60649
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 4, ADDITIONAL: 0

;; QUESTION SECTION:
```

```

;google.com.                IN      MX
;; ANSWER SECTION:
google.com.                 882    IN      MX      400 google.com.s9b2.psmtp.com.
google.com.                 882    IN      MX      100 google.com.s9a1.psmtp.com.
google.com.                 882    IN      MX      200 google.com.s9a2.psmtp.com.
google.com.                 882    IN      MX      300 google.com.s9b1.psmtp.com.

;; AUTHORITY SECTION:
google.com.                 278411 IN      NS      ns2.google.com.
google.com.                 278411 IN      NS      ns3.google.com.
google.com.                 278411 IN      NS      ns4.google.com.
google.com.                 278411 IN      NS      ns1.google.com.

;; Query time: 434 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Fri Feb 11 19:15:56 2011
;; MSG SIZE rcvd: 234

```

[Volver al índice](#)

Consultas con host

host es una herramienta un poco más útil que dig, ya que los resultados son más explícitos y tan completos como el usuario quiera, permitiendo ver mejor las diferencias entre zonas, incluso pudiendo copiar una zona con su texto plano mediante los registros AXFR - dig tiene una presentación muy parecida, pero *sui generis* -.

Ejemplos usando host:

Consultar servidores DNS de Google.com:

```

[root@localhost ~]# host -t ns google.com
google.com name server ns1.google.com.
google.com name server ns2.google.com.
google.com name server ns3.google.com.
google.com name server ns4.google.com.

```

Consultar servidores de correo:

```

[root@localhost ~]# host -t mx google.com
google.com mail is handled by 300 google.com.s9b1.psmtp.com.
google.com mail is handled by 400 google.com.s9b2.psmtp.com.
google.com mail is handled by 100 google.com.s9a1.psmtp.com.
google.com mail is handled by 200 google.com.s9a2.psmtp.com.

```

Consultar una zona completa de dominio para copiarla:

```

[root@localhost postfix]# host -t AXFR ejemplo.org localhost
Trying "ejemplo.org"
Using domain server:
Name: localhost
Address: 127.0.0.1#53
Aliases:

;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 17947
;; flags: qr aa ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;ejemplo.org.                IN      AXFR

;; ANSWER SECTION:
ejemplo.org.                 604800 IN      SOA     dns1.ejemplo.org. root.localhost.
20111012701 604800 86400 2419200 604800

```

```

ejemplo.org.      604800  IN      NS       dns1.ejemplo.org.
ejemplo.org.      604800  IN      MX       10 mail.ejemplo.org.
ejemplo.org.      604800  IN      A        192.168.1.123
*.ejemplo.org.    604800  IN      CNAME    www.ejemplo.org.
dns1.ejemplo.org. 604800  IN      A        192.168.1.123
mail.ejemplo.org. 604800  IN      A        192.168.1.123
www.ejemplo.org.  604800  IN      A        192.168.1.123
ejemplo.org.      604800  IN      SOA      dns1.ejemplo.org. root.localhost.
2011012701 604800 86400 2419200 604800

```

```
Received 238 bytes from 127.0.0.1#53 in 3 ms
```

[Volver al índice](#)

Tipos de registros DNS

Directamente desde Wikipedia...

- A; Address (Dirección) - Este registro se usa para traducir nombres de hosts a direcciones IPv4.
- AAAA; Address (Dirección) - Este registro se usa para traducir nombres de hosts a direcciones IPv6.
- CNAME; Canonical Name (Nombre Canónico) - Se usa para crear nombres de hosts adicionales, o alias, para los hosts de un dominio. Es usado cuando se están corriendo múltiples servicios (como ftp y web server) en un servidor con una sola dirección ip. Cada servicio tiene su propia entrada de DNS (como ftp.ejemplo.com. y www.ejemplo.com.).
- NS; Name Server (Servidor de Nombres) - Define la asociación que existe entre un nombre de dominio y los servidores de nombres que almacenan la información de dicho dominio. Cada dominio se puede asociar a una cantidad cualquiera de servidores de nombres.
- MX; Mail Exchange (Registro de Intercambio de Correo) - Asocia un nombre de dominio a una lista de servidores de intercambio de correo para ese dominio.
- PTR; Pointer (Indicador) - También conocido como 'registro inverso', funciona a la inversa del registro A, traduciendo IPs en nombres de dominio.
- SOA; Start of authority (Autoridad de la zona) - Proporciona información sobre la zona.
- HINFO; Host INFORMATION (Información del sistema informático) - Descripción del host, permite que la gente conozca el tipo de máquina y sistema operativo al que corresponde un dominio.
- TXT; TeXT (Información textual) - Permite a los dominios identificarse de modos arbitrarios.
- LOC; LOCALización - Permite indicar las coordenadas del dominio.
- WKS; Generalización del registro MX para indicar los servicios que ofrece el dominio. Obsoleto en favor de SRV.
- SRV; SeRVicios - Permite indicar los servicios que ofrece el dominio. RFC 2782
- SPF; Sender Policy Framework - Ayuda a combatir el Spam. En este registro se especifica cual o cuales hosts están autorizados a enviar correo desde el dominio dado. El servidor que recibe consulta el SPF para comparar la IP desde la cual le llega, con los datos de este registro.

[Volver al índice](#)

Instalar un servidor Bind9

En RedHat/CentOS

Para poner a funcionar Bind9 en Red Hat/CentOS, utilizaremos yum para instalar el paquete bind, y configuraremos el archivo named.conf para modificar las directivas y añadir todos los archivos de zona que tengamos.

Así que, eso, instalamos con yum...

```
[root@localhost ~]# yum install bind
```

Y si todo va bien, ya está instalado bind. Solo nos falta arrancarlo y activar que se inicie el servicio automáticamente al reiniciar la máquina...

!!! - En Red Hat y CentOS, el servicio de bind se llama named.

Para eso, ejecutaremos chkconfig para comprobar si Bind se ejecuta en los niveles de ejecución 2345, o directamente los añadiremos.

También podemos utilizar, de forma más sencilla, el comando system-config-services, que nos permitirá activar el servicio en una ventana gráfica, con el ratón.

Todo lo demás será configurar el servidor para que resuelva correctamente. Para eso, necesitamos modificar el archivo `/etc/named.conf`. Podemos empezar por echarle un vistazo al archivo de ejemplo localizado en `/usr/share/doc/bind-9.3.6/sample/etc/named.conf`. Incluso podemos copiarlo y utilizarlo como base para nuestra configuración...

```
[root@localhost ~]# cp /usr/share/doc/bind-9.3.6/sample/etc/named.conf /etc/named.conf
```

Si a este fichero le quitamos los comentarios, es tal que así:

```
options
{
    directory "/var/named";
    dump-file      "data/cache_dump.db";
    statistics-file "data/named_stats.txt";
    memstatistics-file "data/named_mem_stats.txt";
};

logging
{
    channel default_debug {
        file "data/named.run";
        severity dynamic;
    };
};

view "localhost_resolver"
{
    match-clients      { localhost; };
    match-destinations { localhost; };
    recursion yes;
    include "/etc/named.root.hints";
};
```

```
        include "/etc/named.rfc1912.zones";
    };
view "internal"
{
    match-clients      { localnets; };
    match-destinations { localnets; };
    recursion yes;

    include "/etc/named.root.hints";

    zone "my.internal.zone" {
        type master;
        file "my.internal.zone.db";
    };
    zone "my.slave.internal.zone" {
        type slave;
        file "slaves/my.slave.internal.zone.db";
        masters { /* put master nameserver IPs here */ 127.0.0.1; };
        // put slave zones in the slaves/ directory so named can update them
    };
    zone "my.ddns.internal.zone" {
        type master;
        allow-update { key ddns_key; };
        file "slaves/my.ddns.internal.zone.db";
    };
};

key ddns_key
{
    algorithm hmac-md5;
    secret "use /usr/sbin/dns-keygen to generate TSIG keys";
};

view "external"
{
    match-clients      { any; };
    match-destinations { any; };

    recursion no;

    zone "my.external.zone" {
        type master;
        file "my.external.zone.db";
    };
};
```

Como vemos, tiene una sección de opciones, otra de login, otra de claves de seguridad, y tres vistas: interna, local y externa. Básicamente, no sirve para nada porque está sin modificar. Más aún, no incluye ni lo necesario. El archivo de ejemplo ni siquiera respeta el estándar de la industria [RFC1035](#), que el que no se tiene en cuenta ningún tipo de autenticación para consultas DNS...

En primer lugar, nos hacen falta ficheros de zona para que nuestro servidor DNS sepa resolver al menos lo mínimo. Ofrezco un tarball con las zonas incluidas en la instalación de Ubuntu directamente [para descargar desde aquí](#) para quien no las tenga y quiera descargarlas desde aquí. En Red Hat y CentOS, hay que descomprimirlas directamente en `/var/named/`, o en el directorio de trabajo de Bind que hayamos definido en nuestro `/etc/named.conf`.

Ahora vamos a modificar el archivo para que sea seguro y con opciones propias, pero respetando lo principal: los estándares y la jerarquía de ficheros de Red Hat. Como vemos, el fichero por defecto en el que trabaja el servicio named es `/var/named`. Ahí situaremos los archivos de zonas, de las distintas zonas básicas (las de los root servers) y de las que queramos.

Vamos a ello...

```
options {
    directory "/var/named";

    forwarders {
        mi.ip.publica.aqui;
    };

    auth-nxdomain no;
    listen-on-v6 { any; };
    allow-transfer { 127.0.0.1; };
//    allow-query { 127.0.0.1; };
//    allow-recursion { 127.0.0.1; };
};

zone "." {
    type hint;
    file "db.root";
};

/*
zone "localhost" {
    type master;
    file "db.local";
};

zone "127.in-addr.arpa" {
    type master;
    file "db.127";
};

zone "0.in-addr.arpa" {
    type master;
    file "db.0";
};

zone "255.in-addr.arpa" {
    type master;
    file "db.255";
};
*/

zone "ejemplo.net" {
    type master;
    file "db.ejemplo.org";
};

zone "ejemplo.org" {
    type master;
    file "db.ejemplo.org";
};
```

[Volver al indice](#)

En Debian/Ubuntu

Para poner a funcionar Bind9 en Debian/Ubuntu, utilizaremos apt-get para instalar el paquete bind9, y configuraremos distintos archivos del directorio de configuración de bind.

Así que eso, instalamos con apt-get...

```
[root@localhost ~]# apt-get install bind9
```

Y si todo va bien, ya está instalado bind y se iniciará solo cuando se arranque el ordenador. Para la configuración, veremos que en Ubuntu se distribuye en distintos ficheros dentro de

/etc/bind/...

```
root@lampserver:~# ls /etc/bind
bind.keys  db.empty  named.conf.default-zones  zones.rfc1918
db.0      db.local  named.conf.local
db.127    db.root   named.conf.options
db.255    named.conf  rndc.key
```

Como vemos, Ubuntu distribuye en distintos ficheros toda la configuración de Bind9. El archivo *named.conf* simplemente incluye los archivos *named.conf.default-zones*, *named.conf.local* y *named.conf.options*, de modo que en cada uno de ellos tenemos cada una de las cosas: las opciones, las zonas locales, y las zonas por defecto. Si queremos añadir más zonas, es tan fácil como editar un nuevo archivo y enlazarlo en el *named.conf*...

Un ejemplo:

```
root@ekip:/etc/bind# echo "include "/etc/bind/allzones.conf";" >> /etc/bind/named.conf
root@ekip:/etc/bind# touch /etc/bind/allzones.conf
```

Ahora tenemos el archivo */etc/bind/allzones.conf* listo para incluir las zonas que queramos, por ejemplo...

```
zone "ejemplo.net" {
    type master;
    file "db.ejemplo.org";
};

zone "ejemplo.org" {
    type master;
    file "db.ejemplo.org";
};
```

Así, tanto *ejemplo.net* como *ejemplo.org* quedarán definidos por el fichero de zona *db.ejemplo.org*.

Pero... ¿Dónde hay que poner el archivo *db.ejemplo.org*? Pues eso está en las opciones:

```
root@ekip:/etc/bind# cat /etc/bind/named.conf.options
options {
    directory "/var/cache/bind";

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

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

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

    auth-nxdomain no;    # conform to RFC1035
    listen-on-v6 { any; };
};
```

Ahí lo tenemos, el directorio de trabajo de Bind en Ubuntu es */var/cache/bind*, y dentro de ese directorio tendremos que colocar nuestros

archivos de zona.

[Volver al índice](#)

Ejemplo de archivo de zona

En un archivo de zona, hacen falta al menos tres registros básicos: SOA, NS y A.

- SOA: Proporciona información sobre la zona y las opciones de configuración como cada cuanto expira y la última vez que se modificó (si está bien definida)
- NS: Un nombre de un servidor DNS en el que se aloja la zona.
- A: Concretamente, deberemos especificar como poco la dirección del servidor DNS al que hemos nombrado en el registro NS.

A partir de ahí, podemos añadir cualquier tipo de registro. Los registros siguen esta sintaxis:

```
name      IN      NS      address
```

Con un nombre que apunta a una dirección a través de un tipo de registro DNS (en este caso NS).

Un ejemplo de archivo perfectamente viable sería este:

```
$TTL      604800
@         IN      SOA      dns1 ejemplo.org. (
                2011012501      ; Serial
                604800      ; Refresh
                86400      ; Retry
                2419200      ; Expire
                604800 )      ; Negative Cache TTL
;
@         IN      NS      dns1
@         IN      NS      dns2
@         IN      MX      10  mail
@         IN      MX      40  mailbackup
www       IN      A        192.168.1.254
dns1      IN      A        192.168.1.254
dns2      IN      A        192.168.1.122
mail      IN      A        192.168.1.254
mailbackup IN    A        192.168.1.200
;
;@        IN      AAAA     ::1
intranet  IN      CNAME    www
```

Ahí definimos que el dominio ejemplo.org con una semana de refresco, veintiocho días de expiración, y los siguientes subdominios:

- dns1.ejemplo.org - Servidor DNS (registros A y NS)
- dns2.ejemplo.org - Servidor DNS (registros A y NS)
- mail.ejemplo.org - Servidor de correo (registros A y MX con prioridad máxima - 10 -)
- mailbackup.ejemplo.org - Servidor de backup de correo (registros A y MX con

prioridad menor - 40 -)

- o `www.ejemplo.org` - Redirección seguramente a un servicio web (registro A)
- o `intranet.ejemplo.com` - redirecciona directamente a `www` (registro CNAME)
- o Un registro IPv6 (AAAA) que redirecciona a la IPv4.

En este fichero, @ equivale al nombre del dominio tal y como se especifica en el archivo en que se crea la zona (*/etc/named.conf* o cualquier incluido). Si se modifican las zonas, habrá que utilizar el comando `rndc reload` para recargar los archivos de base de datos.

[Volver al índice](#)

© 2011 Manuel Méndez - Manuko
Documento bajo Licencia de Documentación Libre GNU/FDL
Última revisión: 20110212-01:33