INSTALACIÓN DE UN SERVIDOR DNS EN UBUNTU

En Ubuntu existen varios software que permiten instalar un servidor DNS en un equipo cualquiera. Los 2 más conocidos son:

a) **Bind:** cuyo paquete de instalación se llama "bind9".
b) **Dnsmasq**: cuyo paquete de instalación se llama "dnsmasq"

Empezaremos viendo un servidor DNS usando Bind y luego lo haremos con Dnsmasq para que veais 2 softwares diferentes (e incompatibles) que hacen lo mismo. Los 2 usan el puerto 53, por tanto, o instalais uno u otro. Pero NUNCA ambos.

1.- Instalación y Configuración de un Servidor DNS Bind en Ubuntu

1.1- Instalación del servidor DNS con Bind

La aplicación que hace de servidor DNS en Linux se denomina **bind9**. Para instalarla, podemos hacerlo con apt-get desde una consola de root:

```
// Instalación del servidor DNS
# apt-get  install  bind9
```

De esta forma instalaríamos los programas necesarios para disponer de un servidor DNS. Tan solo será necesario configurarlo y ponerlo en marcha.

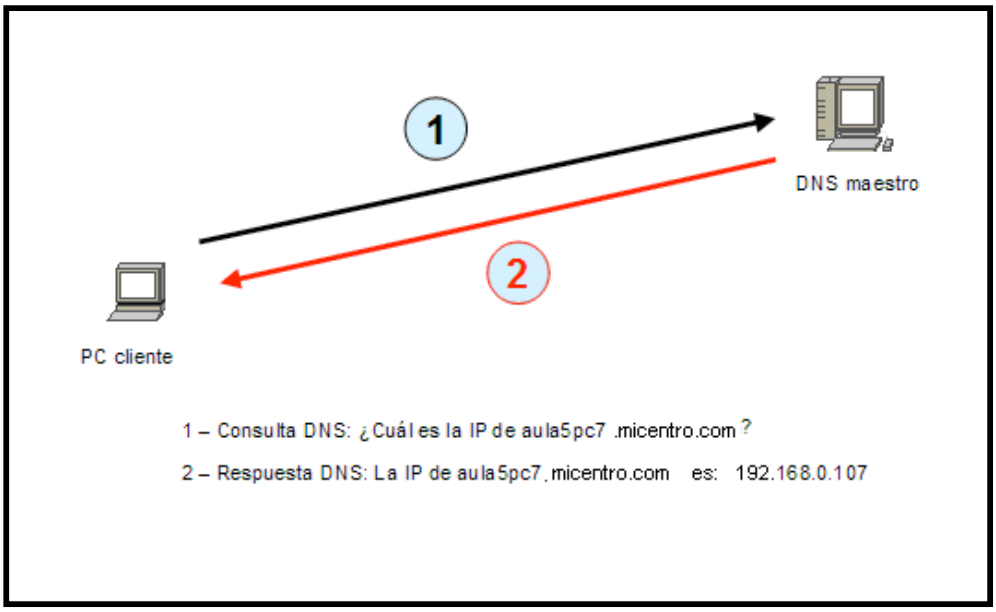
1.2- Configuración del servidor DNS con Bind

El servidor DNS admite tres modos de funcionamiento :

- Servidor DNS maestro
- Servidor DNS esclavo
- Servidor caché DNS

Servidor DNS maestro

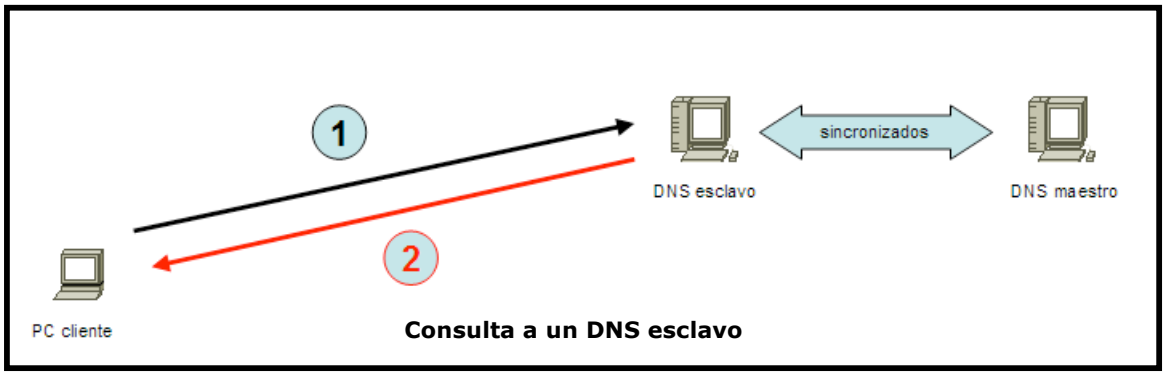
En este modo de funcionamiento, nuestro servidor se comporta como un auténtico servidor DNS para nuestra red local. En este caso es necesario tener zonas de búsqueda directa e inversa (esta última si se quiere que resuelve a la inversa). Atenderá directamente a las peticiones de resolución de direcciones pertenecientes a la red local y reenviará a servidores DNS externos las peticiones del resto de direcciones de Internet.



Consulta a un DNS maestro

Servidor DNS esclavo

Un servidor esclavo actuará como un servidor espejo de un servidor DNS maestro. Permanecerá sincronizado con el maestro. Se utilizan para repartir las peticiones entre varios servidores aunque las modificaciones solo se realicen en el maestro. En redes locales salvo por razones de disponibilidad, es raro que exista la necesidad de tener dos servidores DNS ya que con uno será suficiente.



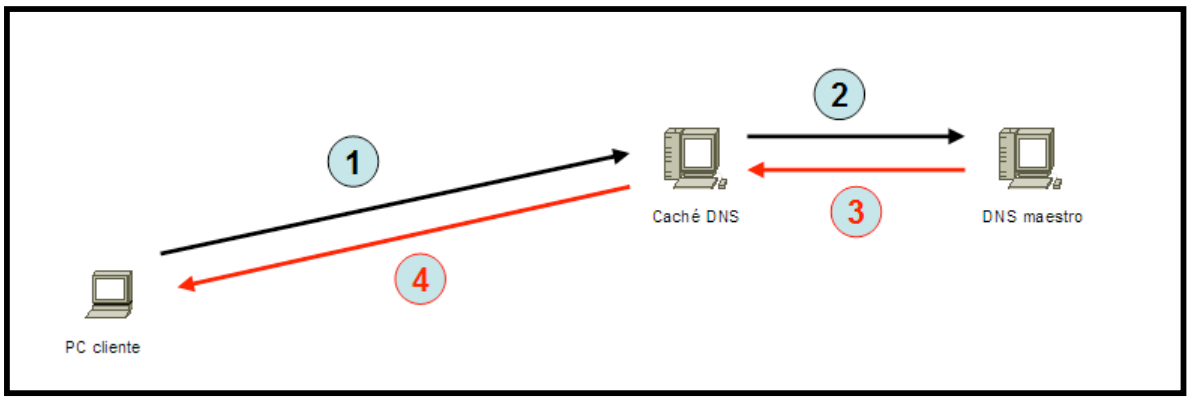
Consulta a un DNS esclavo

Servidor caché DNS

En este modo de funcionamiento, nuestro servidor se comporta como si fuera un auténtico servidor DNS para nuestra red local **aunque realmente no sea un servidor DNS propiamente dicho.** Cuando recibe una petición de DNS por parte de un cliente de nuestra red (**como él no tiene zonas de búsqueda, es decir, no sabe resolver**), la trasladará a un **DNS maestro** que puede estar en nuestra red o fuera, **almacenará en una memoria caché la respuesta y a la vez la comunicará a quien hizo la petición**. **Si un segundo cliente vuelve a realizar la misma petición, como nuestro servidor tiene la respuesta almacenada en su memoria caché, responderá inmediatamente sin tener que cursar la petición a ningún servidor DNS de Internet**. (Por eso agiliza las consultas a Internet).

Disponer de un servidor caché DNS en nuestra red local aumenta la velocidad de la conexión a Internet pues cuando navegamos por diferentes lugares, continuamente se están realizando peticiones DNS. Si nuestro caché DNS almacena la gran mayoría de peticiones que se realizan desde la red local, las respuestas de los clientes se satisfarán prácticamente de forma instantánea proporcionando al usuario una sensación de velocidad en la conexión.

Es un modo de funcionamiento de sencilla configuración ya que prácticamente lo único que hay que configurar son las direcciones IP de un DNS primario y de un DNS secundario. Muchos routers ADSL ofrecen ya este servicio de caché, tan solo hay que activarlo y configurar una o dos IPs de servidores DNS en Internet. En los PCs de nuestra red local podríamos poner como DNS primario la IP de nuestro router y como DNS secundario una IP de un DNS de Internet.



Consulta a un cache DNS. En caso de fallo, se redirecciona hacia un DNS maestro

Archivos de configuracion del DNS

El archivo de configuración principal del DNS es el archivo **/etc/bind/named.conf**, pero éste hace referencia a otros cuantos archivos como por ejemplo:

Archivo	Descripción
named.conf	*Archivo principal de configuración que define zonas y opciones del servidor*
named.conf.options	*Opciones genéricas*
named.conf.local	*Especificación particular de este servidor DNS. Donde pondremos la configuración de nuestro servidor.*
db.localhost	*Fichero de definición de la zona localhost que es el interfaz loopback (para que mi equipo responda al nombre localhost)*
db.127	
db.root	*Es el archivo de definición de la zona de resolución inversa correspondiente a mi IP 127.*.*.* (es la zona inversa de "localhost").*
otros	*DNSs de nivel superior:referencia los servidores de la raíz del árbol de nombres.*
	db.0, db.255, db.empty, rndc.conf, rndc.key, zones.rfc1918

1.2.1- Configuración del servidor como caché DNS

Para configurar nuestro servidor DNS como caché sólo necesitamos modificar 2 ficheros de los citados anteriormente:

/etc/bind/named.conf.options

En este fichero, se deben especificar las IPs de 2 servidores DNS (que normalmente serán las de nuestro proveedor ISP) donde redirigir las peticiones DNS que reciba nuestro servidor y que serán quienes realmente resuelvan los nombres.

Esto se configura en la **sección forwarders:**

```
// Añadir IPs de los DNS de nuestro proveedor en /etc/bind/named.conf.options

forwarders {
   195.235.113.3; 62.37.228.20;
};
```

OJO: no olvideis ningún ;

/etc/resolv.conf

En este fichero, se debe especificar la IP de nuestro propio equipo para indicar que él va a ser nuestro servidor DNS (el equipo Ubuntu que estamos configurando). Dicha IP debe ir precedida de la palabra **nameserver**. Además se puede añadir el nombre de nuestro dominio precedido de la palabra **domain** y, una orden que indica en qué dominio debe buscar el servidor y que comienza con **search.**

```
// Indicamos que nosotros mismos somos servidores DNS
// y por defecto buscamos en nuestro dominio
// Editar /etc/resolv.conf del servidor DNS
domain      XXmicentro.com
nameserver  127.0.0.1
search      XXmicentro.com
```

donde XX será vuestro nº de dominio: 01, 02,…

Tan solo nos faltará **poner en marcha** nuestro servidor de nombres ejecutando en el servidor el **script de inicio (o demonio)** correspondiente:

```
// Arranque del servidor DNS
# /etc/init.d/bind9 restart
```

Configuración de los clientes

El resto de PCs de la red, serán los clientes. Estos clientes pueden ser equipos con Windows, con Ubuntu… Esto ya debería estar controlado.

Configuramos el DNS en el cliente: Para ello, **debemos de configurar en la propiedades del TCP/IP, el DNS** anotando ahí la **IP de nuestro servidor Ubuntu.**

Desde nuestro cliente probamos si el servidor DNS **resuelve nombre externos** (de Internet). Vamos al **Internet Explorer** y ponemos: www.google.es. Si nos aparece la página de Google es que todo va bien.

Otra forma de probar si funciona es usar: **ping** o **nslookup desde la consola MS-DOS:**

Probamos resolución directa:

ping ***www.google.es***

O también:

nslookup ***www.google.es***

Probamos resolución inversa:

ping ***ip_de_Internet*** donde *ip_de_Internet es la IP de cualquier máquina de Internet como las de google.es,…*

O también:

nslookup ***ip_de_Internet***

b) Configuración de un cliente Ubuntu:

Configuramos el **DNS en el cliente**: debemos indicarle la **IP del servidor DNS** en el **fichero /etc/resolv.conf de este cliente**:

Por ejemplo si la IP de nuestro servidor DNS Bind es 192.168.2.202, pondremos en cada /etc/resolv.conf (el de cada cliente):

```
// Editar /etc/resolv.conf del resto de PCs de la red
nameserver 192.168.2.202
```

Ahora desde nuestro cliente Ubuntu probamos si el servidor DNS **resuelve nombre externos** (de Internet). Vamos al **Navegador de Internet** y ponemos: www.google.es. Si nos aparece la página de Google es que todo va bien.

Otra forma de probar si funciona es usar: **dig , host, ping** o **nslookup desde la consola MS-DOS:**

Probamos resolución directa:

dig *www.google.es*

O también:

nslookup *www.google.es*

O también:

host *www.google.es*

COMANDO dig

El comando dig hace una consulta DNS a un servidor. Difiere del ping, host o nslookup en que además de hacer la consulta, muestra el QUERY TIME.
El QUERY TIME es el tiempo que tarda un servidor DNS en respondernos. Por eso, esta orden (dig) es muy importante para comprobar si de verdad hemos agilizado el acceso a Internet o no, al instalar un servidor DNS caché en nuestra red. Para realizar esta comprobación debemos hacer los siguiente:

1º) una vez instalado nuestro servidor DNS caché, desde cualquier cliente con Ubuntu (Windows no incluye dig) ponemos en la consola:

dig nombre_dns_de_internet

Por ejemplo:

dig www.google.es

Acto seguido observamos el Query Time con que nos responde.

2º) Volvemos a ejecutar el mismo comando otra vez y observamos el nuevo Query Time. Veremos que el tiempo de consulta se ha reducido mucho. Esto es debido a que esta segunda consulta no la ha hecho al DNS , sino que la ha sacado de la caché donde la había almacenado en la primera consulta.

Probamos resolución inversa:

ping ***ip_de_Internet*** donde *ip_de_Internet es la IP de cualquier máquina de Internet como las de google.es,…*

O también:

nslookup ***ip_de_Internet***

1.2.2- Configuración del servidor como DNS maestro

Como deseamos tener un dominio privado que es **XXmicentro.com** (XX es vuestro nº de dominio: 01, 02,…) y que todos los equipos de nuestra red pertenezcan a ese dominio. Es mejor instalar un DNS maestro que nos permita resolver nombres externos (como lo hacía el DNS caché) y también NOMBRES INTERNOS como **equipo1.XXmicentro.com.**

Nuestro servidor DNS maestro será capaz de resolver peticiones internas de nombres de dominio, tanto de forma directa como de forma inversa. Por eso debemos crear en él zona de búsqueda directa e inversa.

Para configurar nuestro servidor DNS como maestro necesitamos modificar los siguientes ficheros:

/etc/bind/named.conf.local

En este fichero añadiremos las **zonas de nuestro dominio 'XXmicentro.com'** y su **zona 'reverse'**, que no es más que la forma de decirle al servidor cómo localizar el nombre de dominio si le preguntamos por una IP. Escribimos el código siguiente y guardamos el archivo. Así pues el fichero de configuración "**named.conf.local**" debe quedar como:

```
// Añadir en /etc/bind/named.conf.local
//Archivo de zona para búsquedas directas

zone "XXmicentro.com" {
        type master;
        file "/etc/bind/db.XXmicentro.com";
};

// Archivo de zona para búsquedas inversas
zone "2.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
};
```

XX es vuestro nº de dominio: 01, 02,…

Aquí debeis poner los 3 primeros octetos de vuestra dirección IP de red, pero en orden inverso

Se pone 192 porque se suele poner el 1º octeto de la IP de red.

Explicación del código añadido para crear las zonas directa e inversa:

zone "XXmicentro.com" { nombre de vuestro dominio entre " "

type master; indica que el servidor para esa zona es maestro. Si fuese esclavo se pondría **type slave;**

file "/etc/bind/db.XXmicentro.com"; es el fichero que contendrá los registros de recursos con la información de la zona. Ese fichero no existe y debemos crearlo después.

};

zone "2.168.192.in-addr.arpa" { los 3 primeros octetos de vuestra IP de red y se añade ***.in-addr.arpa***

type master; indica que el servidor para esa zona es maestro. Si fuese esclavo se pondría **type slave;**

file "/etc/bind/db.192"; es el fichero que contendrá los registros de recursos con la información de la zona. Ese fichero no existe y debemos crearlo después.

};

Explicación de por qué la zona inversa tiene ese nombre terminado en .in-addr.arpa:

Si suponemos que nuestra red tiene la IP 192.168.2.0, nuestra zona inversa podría llamarse de varias formas posibles que serían:

a) zone "2.168.192.in-addr.arpa"
b) zone "168.192.in-addr.arpa"
c) zone "192.in-addr.arpa"

Cualquiera de estos nombres sería válido siempre y cuando después, dentro del fichero "db.192" se ponga el contenido adecuado.

¿Qué se pondría dentro del fichero "db.192"?:

Dentro de este fichero se ponen los registros de recursos que hacen referencia a los PCs de la red. En función de cómo se hubiese llamado la zona habrá que poner una IP u otra para los PCs. Así:

a) Si la zona la llamamos zone "2.168.192.in-addr.arpa", en "db.192" la IP de los PCs será sólo el último octeto. Por ejemplo si tenemos 2 PCs con IPs *192.168.2.2* y nombre *PC2* y *192.168.2.5* y nombre *PC5*, pondremos:

```
2       IN      PTR   pc2.XXmicentro.com.
5       IN      PTR   pc5.XXmicentro.com.
```

donde XX es el nº de vuestro dominio.

b) Si la zona la llamamos zone "168.192.in-addr.arpa", en "db.192" la IP de los PCs será los 2 últimos octetos en orden inverso. Por ejemplo si tenemos 2 PCs con IPs *192.168.2.2* y nombre *PC2* y *192.168.2.5* y nombre *PC5*, pondremos:

```
2.2     IN      PTR   pc2.XXmicentro.com.
5.2     IN      PTR   pc5.XXmicentro.com.
```

donde XX es el nº de vuestro dominio.

c) Si la zona la llamamos zone "192.in-addr.arpa", en "db.192" la IP de los PCs será los 3 últimos octetos en orden inverso. Por ejemplo si tenemos 2 PCs con IPs *192.168.2.2* y nombre *PC2* y *192.168.2.5* y nombre *PC5*, pondremos:

```
2.2.168     IN      PTR   pc2.XXmicentro.com.
5.2.168     IN      PTR   pc5.XXmicentro.com.
```

donde XX es el nº de vuestro dominio.

/etc/bind/db.XXmicentro.com

Este fichero NO EXISTE. Debemos crearlo nuevo y es el que contendrá los registros de recursos de nuestra zona de búsqueda directa.

Para crearlo, lo más fácil es copiar el contenido del fichero "/etc/bind/db.local" que sí existe y luego modificarlo. Para ello, desde la consola ejecutamos el comando:

```
cp   /etc/bind/db.localhost   /etc/bind/db.XXmicentro.com
```

Con este comando tendremos en nuestro fichero **/etc/bind/db.XXmicentro.com** una copia del contenido de **/etc/bind/db.localhost. Ahora tendremos que modificarlo para adaptarlo a nuestra zona de forma que nos podría quedar algo así:**

*Supongamos que en nuestra red local tenemos un aula llamada **info** con 10 PCs con IPs que van desde la 192.168.2.203 hasta 210 y cuyos nombres van desde infopc3 hasta infopc10, luego un servidor web (pc11) y un servidor de correo electrónico que además es servidor DNS (pc2). Supongamos también que nuestro servidor DNS se llama sandra-PC (su hostname).*

El archivo de configuración DNS de nuestro dominio podría ser así:

```
// Archivo /etc/bind/db.XXmicentro.com
;
; BIND data file for XXmicentro.com
;
@       IN      SOA     XXmicentro.com. root.XXmicentro.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Default TTL

@               IN       NS       sandra-PC.XXmicentro.com.
@               IN       A        192.168.2.202
@               IN       MX       10       mail.XXmicentro.com.

sandra-PC       IN       A        192.168.2.202
inforpc3        IN       A        192.168.2.203
inforpc4        IN       A        192.168.2.204
inforpc5        IN       A        192.168.2.205
inforpc6        IN       A        192.168.2.206
inforpc7        IN       A        192.168.2.207
inforpc8        IN       A        192.168.2.208
inforpc9        IN       A        192.168.2.209
inforpc10       IN       A        192.168.2.210
www             IN       A        192.168.2.211
mail            IN       A        192.168.2.202
```

Las primeras líneas son unos parámetros relacionados con la actualización del DNS (*número de serie y periodos de actuación*). Las tres siguientes líneas indican quién es el servidor primario (**NS = Name Server y A=IP del servidor**) y quien procesa el correo electrónico del dominio (**MX = Mail eXchange**). Las siguentes líneas especifican las IPs de los distintos PCs componentes del dominio (**A = Address**).

Si olvidamos algún punto y coma, dará errores y no funcionará correctamente. Para revisar los archivos disponemos de los comandos **named-checkconf** y **named-checkzone** que analizan que esté correcta la sintaxis de los mismos.

OJO → Determinar como funcionan los dos comandos anteriores.

Note que todos los nombres de máquinas excepto el del registro SOA no terminan en punto y, por tanto, serán completados con el nombre de la zona (XXmicentro.com declarada en /etc/bind/named.conf.local.

/etc/bind/db.192

Este fichero NO EXISTE. Debemos crearlo nuevo y es el que contendrá los registros de recursos de nuestra zona de búsqueda inversa.

Para crearlo, lo más fácil es copiar el contenido del fichero "/etc/bind/db.127" que sí existe y luego modificarlo. Para ello, desde la consola ejecutamos el comando:

cp /etc/bind/db.127 /etc/bind/db.192

Con este comando tendremos en nuestro fichero **/etc/bind/db.192** una copia del contenido de **/etc/bind/db.127. Ahora tendremos que modificarlo para adaptarlo a nuestra zona de forma que nos podría quedar algo así:**

Si suponemos, como en la zona de búsqueda directa, que en nuestra red local tenemos un aula llamada asi1 con 10 PCs con IPs que van desde la 192.168.2.203 hasta 210 y cuyos nombres van desde inforpc3 hasta inforpc10, luego un servidor web (pc11) y un servidor de correo electrónico que además es servidor DNS (pc2). Supongamos también que nuestro servidor DNS se llama sandra-PC (su hostname). Tendremos:

```
// Archivo /etc/bind/db.192
;
; BIND reverse data file for 192.168.2.0
;
@       IN      SOA     XXmicentro.com. root.XXmicentro.com. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Default TTL

@       IN       NS       sandra-PC.XXmicentro.com.
202     IN       PTR      sandra-PC.XXmicentro.com.

203     IN       PTR              inforpc3.XXmicentro.com.
204     IN       PTR              inforpc4.XXmicentro.com.
205     IN       PTR              inforpc5.XXmicentro.com.
206     IN       PTR              inforpc6.XXmicentro.com.
207     IN       PTR              inforpc7.XXmicentro.com.
208     IN       PTR              inforpc8.XXmicentro.com.
209     IN       PTR              inforpc9.XXmicentro.com.
210     IN       PTR              inforpc10.XXmicentro.com.
211     IN       PTR              www.XXmicentro.com.
202     IN       PTR              mail.XXmicentro.com.
```

/etc/bind/named.conf.options

Ahora es el turno de decirle a BIND **las DNS de nuestro proveedor de servicios (ISP),** para ello editaremos el fichero de opciones **(/etc/bind/named.conf.options**) poniendo dentro de fowarders{} las IPs de los servidores DNS de nuestro ISP. Así conseguiremos que nuestro servidor DNS también resuelva IPs de Internet.

En el caso de trabajar en la red correspondiente a informática, el servidor DNS será el mismo que tienen las máquinas reales configurado.

Si trabajamos nuestros servidores DNS de nuestro ISP, pondríamos, por ejemplo 195.235.113.3 y 62.37.228.20, nos quedará el fichero:

```
// Archivo /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 might need to uncomment the query-source
    // directive below.  Previous versions of BIND always asked
    // questions using port 53, but BIND 8.1 and later use an
    // unprivileged port by default.

    // query-source address * port 53;

    // 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 {
         // Aquí las DNS de tu proveedor ISP
         195.235.113.3;
         62.37.228.20;
     };

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

/etc/resolv.conf

Una vez configurado nuestro servidor DNS, debemos **indicar a nuestro servidor Linux que el servidor DNS es él mismo**, lo cual se especifica en el archivo **/etc/resolv.conf**. Así tendremos:

```
// Indicamos que nosotros mismos somos servidores DNS
// y por defecto buscamos en nuestro dominio

domain XXmicentro.com
nameserver 127.0.0.1
search XXmicentro.com
```

En este punto nuestra configuración de BIND estaría completa, excepto por el detalle de que hay que reiniciar el servidor para que cargue las nuevas zonas definidas, para ello volvemos al terminal y escribimos como root:

/etc/init.d/bind9 restart

Ya sólo falta probar nuestro servidor con la orden ping, host, dig o nslookup y configurar nuestros clientes y probar desde ellos si resuelve nuestro servidor.

1.2.3- Configuración del servidor como DNS esclavo

Si deseamos configurar nuestro servidor DNS para que actúe como esclavo de un servidor DNS maestro, la configuración es mucho más sencilla que en el caso anterior ya que únicamente será necesario indicar en el DNS esclavo quién es el servidor DNS maestro, y en el DNS maestro, la IP del DNS esclavo.

Ejemplo, supongamos que el nombre del **DNS maestro** es sandra-PC.XXmicentro.com (IP 192.168.2.202) y que el nombre del **DNS esclavo** es dns2.XXmicentro.com. En el archivo '**db.XXmicentro.com'** de zona de búsqueda directa del **maestro** añadiremos la línea del segundo dns justo debajo de donde está la del primero:

```
// Añadir línea en /etc/bind/db.XXmicentro.com del maestro

....
            IN      NS      sandra-PC.XXmicentro.com.
            IN      NS      dns2.XXmicentro.com.    //    Nueva    línea
....
```

de esta forma indicaremos que existen más servidores DNS para dicha zona.

Lo mismo haremos en el archivo '**db.192'** de la zona inversa del **maestro**:

```
// Añadir línea en /etc/bind/db.192 del maestro

....
            IN      NS      sandra-PC.XXmicentro.com.
            IN      NS      dns2.XXmicentro.com.    //    Nueva    línea
....
```

En el archivo **/etc/bind/named.conf.local** del servidor **DNS esclavo debemos indicar que se trata de un servidor esclavo y** también debemos indicar **quién es el maestro**:

```
// Añadir en /etc/bind/named.conf.local del esclavo
zone "XXmicentro.com" {
        type slave;
        file "/etc/bind/db.XXmicentro.com";
        masters { 192.168.2.202; };
};

zone "2.168.192.in-addr.arpa" {
        type slave;
        file "/etc/bind/db.192";
        masters { 192.168.2.202; };
};
```

En el archivo **/etc/bind/named.conf.local** del servidor **DNS maestro** podemos utilizar **also-notify para mantener los DNS sincronizados**. Con also-notify pasamos los cambios de zonas del maestro al esclavo:

```
// Archivo /etc/bind/named.conf.local del maestro
zone "XXmicentro.com" {
        type master;
        file "/etc/bind/db.XXmicentro.com";
        also-notify {ip_del_esclavo;}
};

zone "2.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192";
        also-notify {ip_del_esclavo;}
};
```

De ésta forma dispondremos en la red de un servidor DNS esclavo que podrá satisfacer las peticiones DNS al igual que lo haría el maestro. Es interesante si el número de peticiones es muy elevado y se requiere distribuir la carga entre los dos servidores, o si deseamos disponer de servicio DNS de alta disponibilidad de forma que aunque el servidor maestro deje de funcionar, el servidor esclavo podrá seguir ofreciendo el servicio.

NOTA: Cada vez que hagamos un cambio en los archivos **/etc/bind/db.XXmicentro.com** y **/etc/bind/db.192** del maestro, debemos acordarnos de actualizar el **parámetro serial** (*incrementar en una unidad*) para que los dns dependientes del maestro sepan que ha cambiado y actualicen su información para mantenerse perfectamente sincronizados.