Instalar y configurar un servidor DNS con Ubuntu Server paso a paso gracias a Slice of Linux

Un servidor DNS (*Domain Name System*) es un sistema que nos permite usar nombres de dominio en lugar de direcciones IP. Su principal ventaja es que para nosotros es mucho más fácil recordar un nombre que una dirección IP.

El servidor DNS más utilizado es Bind y, aunque teníamos un poco abandonado a Ubuntu Server en Slice of Linux, hoy vamos a ver cómo instalarlo y configurarlo sobre él paso a paso. El contenido de este tutorial es genérico pero las pruebas y capturas de pantalla se han hecho sobre Ubuntu 10.04 Server.

Los valores que debemos tener claros antes de comenzar son los siguientes:

Dirección IP del servidor: **192.168.2.1**

Nombre del servidor: **servidor**

Dominio que vamos a crear: ***alberto.edu***

Estos valores deberemos sustituirlos por los que necesitemos en cada caso.

Los pasos para instalar y configurar Bind en Ubuntu Server son los siguientes:

1. Actualizamos la información de los repositorios con el siguiente comando:

```
sudo aptitude update
```

2. Instalamos el servidor DNS Bind9:

```
sudo aptitude install bind9
```

3. Hacemos una copia de seguridad del archivo que vamos a modificar:

```
sudo cp /etc/bind/named.conf.local{,.original}
```

Este comando nos puede ahorrar mucho tiempo y está descrito en el artículo “hacer copias de seguridad de archivos rápidamente“.

4. Editamos el archivo */etc/bind/named.conf.local* con el siguiente comando:

```
sudo nano /etc/bind/named.conf.local
```

y añadimos el siguiente contenido:

```
zone "alberto.edu" {
  type master;
  file "db.alberto.edu";
};

zone "2.168.192.in-addr.arpa" {
  type master;
  file "db.192.168.2";
};
```

Esto se puede ver en la siguiente captura de pantalla:

Editamos el archivo /etc/bind/named.conf.local

Para guardar el archivo debemos pulsar la combinación de teclas **Control+O** y para salir **Control+X**.

5. Para comprobar la sintaxis de los archivos de configuración ejecutamos el siguiente comando:

```
named-checkconf
```

Si no aparece nada, la sintaxis de los archivos de configuración es correcta. ¡Ojo! Eso no significa que no haya ningún error, sólo que no hay errores de sintaxis.

Ejecución de named-checkconf sin errores

Si hubiésemos cometido un error de sintaxis, nos aparecería indicado junto a la línea en la que ocurre. En el siguiente ejemplo puede verse que en el archivo */etc/bind/named.conf.local* en la línea 15 hay un error:

Ejecución de named-checkconf con un error

6. Creamos el archivo */var/cache/bind/db.alberto.edu*:

```
sudo nano /var/cache/bind/db.alberto.edu
```

e incluimos el siguiente contenido:

```
$ORIGIN alberto.edu.
$TTL 86400 ; 1 dia
@     IN    SOA    servidor      postmaster (
      1    ; serie
      6H   ; refresco (6 horas)
      1H   ; reintentos (1 hora)
      2W   ; expira (2 semanas)
      3H   ; mínimo (3 horas)
)
       NS    servidor
servidor A     192.168.2.1
```

El contenido del archivo es bastante especial y no lo vamos a comentar pero para más información se puede leer el RFC 1912 y el RFC 2308. Aquí deberíamos añadir todos los equipos de nuestra red que quisiéramos mantener identificados pero como es más fácil hacerlo con DDNS (*Dynamic DNS*) ya lo veremos en otra ocasión.

7. Comprobamos la zona que acabamos de crear (*alberto.edu*):

```
named-checkzone alberto.edu
/var/cache/bind/db.alberto.edu
```

En esta ocasión siempre nos aparecerá una salida, ya sea para indicarnos que todo está bien (OK) o algún error.

```
sliceoflinux@servidor:~$ named-checkzone sliceoflinux.lan /var/cache/bind/db.sli
ceoflinux.lan
zone sliceoflinux.lan/IN: loaded serial 1
OK
```

Ejecución de named-checkzone sin errores

8. A continuación creamos el archivo */var/cache/bind/db.192.168.2* para la zona inversa:

```
sudo nano /var/cache/bind/db.192.168.2
```

e incluimos el siguiente contenido:

```
$ORIGIN 2.168.192.in-addr.arpa.
$TTL 86400    ; 1 dia
@     IN    SOA    servidor      postmaster (
      1     ; serie
      6H    ; refresco (6 horas)
      1H    ; reintentos (1 hora)
      2W    ; expire (2 semanas)
      3H    ; mínimo (3 horas)
)
     NS    servidor.alberto.edu.
      1    PTR    servidor.alberto.edu.
```

El número 1 se corresponde con el último dígito de la dirección IP del servidor (192.168.2.1).

9. Comprobamos la zona inversa recién creada:

```
named-checkzone 2.168.192.in-addr.arpa
/var/cache/bind/db.192.168.2
```

Al igual que antes obtendremos un mensaje para indicarnos tanto si la zona es correcta como si no lo es.

```
sliceoflinux@servidor:~$ named-checkzone 2.168.192.in-addr.arpa /var/cache/bind/
db.192.168.2
zone 2.168.192.in-addr.arpa/IN: loaded serial 1
OK
```

Ejecución de named-checkzone sin errores

10. Reiniciamos el servicio:

```
sudo service bind9 restart
```

Si todo va bien, veremos que está OK.

```
sliceoflinux@servidor:~$ sudo service bind9 restart
 * Stopping domain name service... bind9                          [ OK ]
 * Starting domain name service... bind9                          [ OK ]
```

Reiniciamos el servicio

11. Revisamos el log para comprobar que todo ha ido bien. Aunque se puede hacer con el comando tail, yo prefiero less porque me permite ver todo el contenido del mismo:

```
less /var/log/syslog
```

El resultado se puede ver en la siguiente captura:

```
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 255.255.255.255.IN-A
DDR.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 0.0.0.0.0.0.0.0.0.0.
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 1.0.0.0.0.0.0.0.0.0.
0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: D.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 8.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: 9.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: A.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: automatic empty zone: B.E.F.IP6.ARPA
Apr 20 16:30:19 servidor named[3403]: command channel listening on 127.0.0.1#953
Apr 20 16:30:19 servidor named[3403]: command channel listening on ::1#953
Apr 20 16:30:19 servidor named[3403]: zone 0.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone 127.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone 2.168.192.in-addr.arpa/IN: loaded ser
ial 1
Apr 20 16:30:19 servidor named[3403]: zone 255.in-addr.arpa/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone sliceoflinux.lan/IN: loaded serial 1
Apr 20 16:30:19 servidor named[3403]: zone localhost/IN: loaded serial 2
Apr 20 16:30:19 servidor named[3403]: running
Apr 20 16:30:19 servidor named[3403]: zone 2.168.192.in-addr.arpa/IN: sending no
tifies (serial 1)
```

Comprobamos que no hay errores en syslog

Para salir deberemos pulsar la tecla **q**.

12. Editamos el archivo */etc/resolv.conf* para que nuestro servidor resuelva las peticiones DNS:

```
sudo nano /etc/resolv.conf
```

Cambiando el primero de los servidores DNS por la IP del nuestro:

```
nameserver 192.168.2.1
nameserver 8.8.8.8
```

Probamos nuestro servidor de nombres:

```
dig alberto.edu
```

La respuesta será muy parecida a la siguiente:

Ejecución de dig *alberto.edu*

Probamos la resolución inversa:

```
dig -x 192.168.2.1
```

Esta sería la salida esperada del comando anterior:

Ejecución de dig -x 192.168.2.1

Por último, para poder sacarle partido al servidor que hemos creado nos faltaría incluir este servidor DNS en la configuración de los clientes. Para esto puedes nuestro artículo Configurar el servidor DNS de Ubuntu Karmic desde la interfaz gráfica.

Este manual esta sacado íntegramente de sliceoflinux , tienen un montón de manuales en español y explicados para su fácil entendimiento.