Qué

Se han de instalar y configurar múltiples servidores DNS de forma colaborativa con toda la clase. Tendremos el dominio 2daw.iesmariaenriquez.es, y tres subdominios: fila1.2daw.iesmariaenriquez.es, fila2.2daw.iesmariaenriquez.es y fila3.2daw.iesmariaenriquez.es, que corresponderán a las diferentes filas de la clase.
Además, los dominios y zonas han de mantener una jerarquía.
En cada fila el sistema operativo de los servidores DNS será diferente, y se usarán servidores exclusivos de esos sistemas operativos. En el subdominio fila1 los servidores serán compatibles únicamente con SO *Unix-like*, en fila2 serán multiplataforma y en fila3 se usarán servidores sólo para Windows.
Finalmente, todos los servidores DNS han de poder comunicarse entre sí, y debe haber al menos uno autoritativo, uno pasivo y otro activo.

Para qué

El objetivo es crear un *Domain Name System* distribuido entre múltiples ordenadores de la clase con dominios y subdominios, y diferentes servidores DNS de diferentes tipos para simular como funciona el servicio DNS en situaciones reales y entender las dificultades logísticas que tiene gestionar un servicio de este tipo de forma distribuida.

Cómo

Modificar permanentemente el resolv.conf

En este punto se explica cómo hacer que una máquina con Ubuntu resuelva usando unos servidores específicos.

El problema reside en que, por defecto, el resolv.conf usa el servidor de nombres `127.0.0.53`, que es básicamente systemd-resolved. Además, como el aula tenemos ya un servidor DHCP, éste nos da 3 servidores de nombres por defecto, que systemd-resolved tratará de usar internamente para resolver nombres.

La solución a esto es seguir estos pasos:

1. Modificar el fichero `/etc/systemd/resolved.conf`: Se descomentan las lineas `DNS` y `FallbackDNS` y se añade en DNS el servidor o servidores que queremos que resuelva, y en FallbackDNS un servidor DNS de internet, como el 1.1.1.1 de Cloudflare o el 8.8.8.8 de Google.
2. Sobreescribir el enlace de /etc/resolv.conf: resolv.conf, por defecto, tiene un enlace a un archivo que se reestablece a su estado original periódicamente y tras cada reinicio del servicio systemd-resolved. Se ejecuta `ln -sf /run/systemd/resolve/resolv.conf /etc/resolv.conf` para sobreescribirlo por un archivo que solo tiene definidos como DNS los que

se especificaron en el paso anterior.
3. Reiniciar los servicios network-manager y systemd-resolved, usando el comando `systemctl restart` con permisos de administrador, seguido del nombre del servicio.
4. Comprobar los contenidos de /etc/resolv.conf. Si no son los deseados, se pueden modificar para añadir o eliminar los `nameserver` necesarios.

Primera fila

En la primera fila el sistema operativo es Ubuntu 20.04. Nos hemos decidido por **bind9**, que dejó de ser multiplataforma hace unos años, pues es de los pocos servidores DNS exclusivos para sistemas operativos tipo Unix que soporta todas las características que necesitamos para cumplir con lo que requiere la práctica, sin entrar en materia que no hemos visto y es bastante complicada, como DNSSEC.

Se configurarán dos servidores DNS en dos máquinas diferentes, 2daw.iesmariaenriquez.es, y el subdominio fila1.2daw.iesmariaenriquez.es.

Antes de instalarlo, hay que tener en cuenta que la versión de bind9 que está en los repositorios sí es multiplataforma. Por tanto, para instalar la versión multiplataforma, se tiene que añadir un repositorio con paquetes más actualizados antes de instalarlos:

```
sudo add-apt-repository ppa:isc/bind-dev
sudo apt update
sudo apt install bind9
```

Con esto se instala la versión de desarrollo de bind9, la 19.9, en vez de la 16.1 que tienen los repositorios de Ubuntu 20.04.
Ahora se comprueba que el servicio este en marcha con `systemctl status named`. Si no lo está, puede que sea culpa del servicio systemd-resolved ocupando el puerto 53 en localhost. Se puede comprobar ejecutando `sudo lsof -i:53`.
Este problema sucede cuando un servicio DNS trata de escuchar por 0.0.0.0, pues intenta escuchar por localhost y ya está ocupado, por lo que se arregla especificando por que direcciones IP puede escuchar añadiendo `listen-on {<lista_ips>};` en /etc/bind/named.conf.options. De todos modos, en el servicio de DNS de bind9 debería poder iniciarse sin problemas sin hacer nada de esto.
Una vez tenemos bind9 instalado y ejecutándose, hacemos una copia de los ficheros de configuración antes de modificarlos, en caso de que hagamos algún cambio que no sepamos cómo revertir:

- Archivo de configuración de las opciones de bin9:
  `sudo cp /etc/bind/named.conf.options{,.bck}`
- Archivo de configuración donde se definen las zonas:
  `sudo cp /etc/bind/named.conf.local{,.bck}`

- Archivo de configuración principal, agrega las opciones de los dos anteriores: `sudo cp /etc/bind/named.conf{,.bck}`

Tras eso, se han hecho dos cambios en el fichero /etc/bind/named.conf.local de ambas máquinas. - Se ha decomentado la línea siguiente línea del fichero de configuración que importa un fichero adicional, /etc/bind/named.conf.local, para cumplir con el RFC 1918:

```
include "/etc/bind/zones.rfc1918";
```

Este archivo contiene zonas para la resolución inversa de dominios asociados a IPs dentro del espacio privado de direcciones, Que un conjunto de direcciones que las organismos, empresas e individuos pueden usar sin coordinarse con IANA, pero que no pueden usarse como direcciones IP del Internet público, según el punto 3 del RFC 1918.

El archivo named.local.options sólo incluye el directorio que usa bind9 para guardar los archivos relacionados con el servidor DNS, y se ha dejado su valor por defecto, /var/cache/bind.

Zona 2daw.iesmariaenriquez.es

Esta zona define el servidor autoritativo del dominio 2daw.iesmariaenriquez.es, el `10.0.109.15`.
Primero, se ha modificado el fichero de configuración /etc/bind/named.conf.local, para añadir la zona 2daw.iesmariaenriquez.es:

```
zone "2daw.iesmariaenriquez.es" {
    type master;
    file "/etc/bind/2daw.iesmariaenriquez.es.zone";
};
```

Esto es lo que se ha indicado en este fichero de configuración:

- `zone` especifica el nombre de la zona. No es necesario que sea igual que el nombre del dominio, pero simplifica la creación de registros DNS.
- `type` indica el tipo de zona que puede ser un dominio. En este caso se ha establecido `master` (que es equivalente a `primary`), pero hay otros tantos, como `forward`, `hint`, o `slave` (equivalente a `secondary`).
  Este tipo indica que este los registros DNS de la zona no le llegan a este servidor de otro servidor, sino que es el origen.
  Referencia: https://bind9.readthedocs.io/en/v9_16_6/reference.html#zone-types
- `file` va seguido de un string que indica la ruta absoluta al fichero donde están los registros DNS de la zona.

Lo siguiente es crear un registro DNS para la zona recién creada antes de reiniciar el servicio:

```
$TTL 120
```

```
@ IN SOA 2daw.iesmariaenriquez.es. root (
        16660187 ; serial
        30 ; refresh
        120 ; retry
        60 ; expire
        120 ; minimum
        )

@   IN  NS  2daw.iesmariaenriquez.es.
@   IN  A   10.0.109.15
root    IN  CNAME   2daw.iesmariaenriquez.es.
cache   IN  A   10.0.109.13
@   IN  MX  10 2daw.iesmariaenriquez.es.

; Registros A
pc01    IN  A   10.0.109.111
pc02    IN  A   10.0.109.112
pc03    IN  A   10.0.109.113

; Delegación de zonas (No ha funcionado)
dns1    IN  A   10.0.109.14
fila1   IN  NS  dns1.2daw.iesmariaenriquez.es.
dns2    IN  A   10.0.109.9
fila2   IN  NS  dns2.2daw.iesmariaenriquez.es.
dns3    IN  A   10.0.109.9
fila3   IN  NS  dns3.2daw.iesmariaenriquez.es.
```

Antes de justificar por que se han puesto estos valores, esta es la explicación de cada parámetro:

- **$TTL**: Esto es una directiva (también conocida como variable o macro) que especifica el *Time to Live* por defecto, que es el tiempo en segundos que que una consulta es considerada válida en la cache del servidor DNS. Si no se especificase, tendría que ponerse en cada registro que se crease en el archivo, antes del `IN`.
- **@**: Representa y es equivalente al nombre de la zona, seguido por un punto. Este también es el valor que tiene por defecto la directiva $ORIGIN, que añade su valor a cualquier host en los registros que no acabe con '.'. Por esto se ha llamado igual a la zona que al dominio, simplifica la configuración y la hace más legible.
- **SOA**: Las siglas de Start Of Authority. Identifica al servidor autoritario de una zona y es dónde se establece la configuración de ciertos valores:
    1. Serial: Es un identificador para este registro DNS. Debería ser diferente de todos los demás registros de zonas de este servidor, pues algunas funcionalidades del DNS lo requiere. Es práctica estándar

ponerle de valor una fecha en formato AAAA-MM-DD o un timestamp de Unix, y se suele cambiar cada vez que se modifica para dar a entender que se ha modificado el registro.
    2. Refresh: Tiempo en segundos que un servidor secundario ha de esperar antes de refrescar los valores de un registro.
    3. Retry: Tiempo en segundos que un servidor DNS secundario ha de esperar tras fallar en recuperar los datos del servidor primario.
    4. Expire: Es la cantidad máxima de tiempo, en segundos, que un servidor DNS secundario tiene permitido conservar los registros DNS.
    5. Minimum: El TTL mínimo.

- **NS**: Siglas de Name Server. Especifica qué host o hosts son los encargados de resolver los nombres de un dominio o subdominio.
- **A**: Registros de dirección IPv4. Indica la dirección IP de un host.
- **CNAME**: Acortación de *Canonical Name*. Es un alias para un host que ya tenga un registro A.
- **MX**: Acortación de *Mail*. Indica el servidor de correo asociado a un host. Dicho host ha de tener un registro A. Además, el registro MX va acompañado de un número que indica su prioridad. El valor que se suele usar como base es 10.

Teniendo lo anterior en cuenta, esto es lo que tiene este servidor:

La autoridad de este dominio es 2daw.iesmariaenriquez.es, que también es el servidor de nombres del dominio, que es el host con IP 10.0.109.15, es decir, la misma máquina que tiene las zonas.

Además, el servidor de correo del dominio 2daw.iesmariaenriquez.es es ese mismo ordenador, con una prioridad de 10, que de momento no es relevante porque no hay más servidores de correo para este dominio.

Tenemos una serie de hosts definidos en el servidos, algunos de los cuales son servidores de nombres para las zonas fila1, fila2, y fila3. Esto, aunque desconozco por qué, no funciona.

Cabe destacar que todos los con el tiempo se han puesto muy bajos para agilizar las cosas en caso de que se fuesen a usar, pero al final no se han hecho servidores secundarios así que la mayoría de estos valores no tienen efecto en el funcionamiento actual. La única excepción es el TTL mínimo, pues si tenemos un servidor caché, al que tanto esto como el TTL por defecto.

Por último, una vez está hecha toda la configuración, se reinicia el servicio con `sudo service named restart` y se compueba que funcione con `service named status` y `dig @127.0.0.1 pc01.2daw.iesmariaenriquez.es`.

Zona fila1.2daw.iesmariaenriquez.es

Esta zona define el servidor autoritativo del dominio 2daw.iesmariaenriquez.es, el `10.0.109.14`.
Al igual que con el anterior, se ha modificado el fichero de configuración /etc/bind/named.conf.local, para añadir la zona fila1.2daw.iesmariaenriquez.es:

```
zone "fila1.2daw.iesmariaenriquez.es" {
    type master;
    file "/etc/bind/fila1.2daw.iesmariaenriquez.es.zone";
};
```

Y se ha creado este archivo de registros DNS:

```
$TTL    86400

@   IN  SOA fila1.2daw.iesmariaenriquez.es. root (
                         8891 ; serial
                         3H       ; refresh
                         15       ; retry
                         1w       ; expire
                         3h       ; minimum
                        )

@           IN  NS  fila1.2daw.iesmariaenriquez.es.
@           IN  A   10.0.109.14
root            IN  CNAME   fila1.2daw.iesmariaenriquez.es.
pepe            IN  A   10.0.109.100
sandra          IN  A   10.0.109.16
manel           IN  A   10.0.109.7
paul            IN  A   10.0.109.12
raul            IN  A   10.0.109.10
alejandro       IN  A   10.0.109.15
@           IN  MX 10   fila1.2daw.iesmariaenriquez.es.
pc11            IN  A   10.0.109.120
pc12            IN  A   10.0.109.121
```

En este caso como, a diferencia del servidor de 2daw.iesmariaenriquez.es, este es el que debe resolver los hosts de la primera fila, tiene muchos más hosts definidos.

Otra diferencia con 2daw.iesmariaenriquez.es es que este servidor nunca estuvo pensado para tener un servidor secundario asociado, así que los valores de TTL, refresh, retry, expire y minimum TTL se han dejado por defecto. Tienen una nomenclatura diferente porque se han importado desde Yadifa, el servidor DNS que estabamos usando en la primera fila anteriormente, pero los registros DNS de yadifa son un subset de los de bind9, así que son compatibles.

Aparte de eso, no hay cambios significativos en la forma en la que se han definido los registros.

Segunda fila

En la segunda fila se ha usado también **bind9**, pero la versión que viene en los repositorios de Ubuntu 20.04, la 16.1 que aún es válida como multiplataforma. Se ha elegido porque es el *defacto standard* en lo que a servidores DNS se refiere. Tiene más documentación, soporte, y estabilidad qeu ningún otro. Al principio se optó por Unbound, pero no servía como autoritativo, así que se desechó la idea.

El proceso de instalación es aún más sencillo que en la primera fila:

```
sudo apt install bind9
```

El resto de pasos que tienen que ver con la instalación y la configuración inicial de la fila 1 se aplican también a esta versión de bind9.

La principal diferencia entre esta fila y la anterior, es que se ha configurado un servidor DNS para el subdominio fila2.2daw.iesmariaenriquez.es, y un servidor DNS en una máquina aparte que hace, exclusivamente, de caché de 2daw.iesmariaenriquez.es.

Zona fila2.2daw.iesmariaenriquez.es

Esta zona define el servidor autoritativo del dominio fila2.2daw.iesmariaenriquez.es, el `10.0.109.9`.

Primero, se ha modificado el fichero de configuración /etc/bind/named.conf.local, para añadir la zona fila2.2daw.iesmariaenriquez.es:

```
zone "fila2.2daw.iesmariaenriquez.es"{
    type master;
    Allow-query{127.0.0.1;10.0.109.0/24;};
    Allow-transfer{127.0.0.1;10.0.109.0/24;};
    file "/etc/bind/directofila2.conf";
};
```

Esta zona se ha configurado para que solo se le respondan peticiones a los ordenadores de clases, con el parámetro Allow-query. Además, como estaba pensada inicialmente como una zona primaria que tendría máquinas con esa zona como zona secundaria, se le ha permitido la transferencia de zonas a los ordenadores de clase con Allow-transfer, pero, para los propósitos de esta práctica, no tiene importancia.

Después se ha hecho el registro DNS:

```
;
; BIND reverse data file for local loopback interface
;
$TTL    604800
@   IN  SOA fila2.2daw.iesmariaenriquez.es. root (
        6672587     ; Serial
```

```
        604800      ; Refresh
        86400       ; Retry
        2419200     ; Expire
        604800 )    ; Negative Cache TTL
;
@   IN  A   10.0.109.9
@   IN  NS  fila2.2daw.iesmariaenriquez.es.
root    IN  CNAME   fila2.2daw.iesmariaenriquez.es.
pc21    IN  A   10.0.109.21
pc22    IN  A   10.0.109.11
mail    IN  A   10.0.109.9
@   IN  MX 10   mail.fila2.2daw.iesmariaenriquez.es.
```

Los tipos de registros son los mismos que en la última versión de bind9, así que no merece la pena explicarlos de nuevo.

Una vez hecho, se reinicia el servicio de bind9, usando `sudo service named restart`.

Servidor caché de 2daw.iesmariaenriquez.es

Este servidor no sólo no es autoritativo de ninguna zona, si no que no tiene zonas definidas. La función de este servidor es hacer consultas periódicamente a 10.0.109.15, el servidor DNS de 2daw.iesmariaenriquez.es y sólo a él de entre los de clase, y guardar las respuestas en caché el tiempo que especifique su TTL.

Para conseguir esto, se tiene que modificar el fichero /etc/bind/named.conf.options para permitir recursión. Para eso, se añaden los parámetros `recursion yes;` y `allow-query {127.0.0.1/10.0.109.0/24}`. Lo último es porque `allow-query`, además de limitar qué máquinas pueden usar este servidor es una lista ACL que sirve como fallback de `allow-recursion`. Podría ponerse esta última, pero no es necesario y así la configuración es más simple.

Luego se tiene que hacer que este servidor DNS haga *forwarding*, usando el parámetro `forwarders {}`. Se han añadido 10.0.109.15, el servidor de 2daw.iesmariaenriquez.es porque es necesario, y 8.8.8.8, el servidor DNS de google, para aportarle un poco más de funcionalidad. También se le pone el parametro `forward only;`, porque no queremos que tenga sus propias zonas, ni ahora ni a futuro.

Por último se habilita la validación de dnssec para evitar problemas de compatibilidad con otros servidores que si lo tengan habilitado.

Este sería el fichero de configuración final:

```
options {
    directory "/var/cache/bind";

    recursion yes;
```

```
    allow-query {127.0.0.1;10.0.109.0/24;};

    forwarders{
    10.0.109.15;
    8.8.8.8;
    };
    forward only;

    dnssec-validation yes;

    auth-nxdomain no;

    listen-on-v6 { any; };
};
```

Ahora, se añade al resolv.conf como nameserver 10.0.109.15, y se reinicia el servicio de bind9 con `sudo service named restart`.

Fila 3

En la fila 3 sólo se ha instalado un servidor, el autoritativo del dominio fila3.2daw.iesmariaenriquez.es, con IP 10.0.109.231, pero esta vez el sistema operativo es Windows Server 2019, sin interfaz gráfica. El servidor DNS es **WINS**, porque es uno de los dos programas de este tipo que funcionan sólo en MS Windows, y es el que más información, compatibilidad y soporte tiene.

Instalación de WINS

Como es sin interfaz gráfica, se tiene que hacer en una máquina virtual y con línea de comandos. El proceso de instalación del sistema operativo está guiado, es sencillo y esta fuerá del ámbito de está práctica, así que no se va a cubrir.

El primer paso para instalar el servidor DNS WINS, es instalar el rol de Servicios de dominio de Active Directory. Para ello, se introdue el comando `Install-WindowsFeature -Name AD-Domain-Services -IncludeManagementTools`.

Después, se tiene que se tiene que indicar que este servidor es un controlador de dominio, con el comando `Import-Module ADDSDeployment`.

El siguiente paso es crear un bosque (Que es el concepto que usa MS en vez de zonas), con el comando `Install-ADDSForest`. Pedirá el nombre del dominio que queremos crear, en este caso, fila3.2daw.iesmariaenriquez.es. Al terminar, se reiniciará el servidor.

Con el comando `sconfig`, podemos comprobar que el dominio se ha establecido correctamente.

Instalación de RSAT

RSAT es un programa para gestionar ordenadores con SO Windows Server de forma remota a través de una interfaz gráfica, y es lo que hemos usado para modificar la configuración y añadir registros al servidor DNS WINS.

Para instalarlo se ha de acceder a un cliente cualquiera de clase con Windows 10, y buscamos la aplicación 'características opcionales', desde donde instalaremos 'control remoto del servidor' (el propio RSAT) y 'control DNS'.

Una vez instalado, se abre el programa y se agrega el servidor poniendo la IP del servidor DNS. Cuando encuentre al servidor, se introduce un usuario con permisos elevados y su contraseña, por ejemplo, Administrador.

Añadir registros

El proceso de añadir registros es relativamente sencillo. Se entra en el Administrador de DNS haciendo clic derecho sobre el nombre de equipo del servidor, se entra en la carpeta de 'Zonas de búsqueda directa', desde ahí se hace clic derecho en la carpeta de fila3.2daw.iesmariaenriquez.es y aparecerán diferentes opciones para crear diferentes tipos de registro: A, AAAA, MX, NS, SOA, etc.

Al hacer clic sobre cualquiera de las opciones aparacerá una nueva ventana con un pequeño formulario pidiendo toda la información necesaria, y se repite este proceso hasta que se tengan todos los registros necesarios.

Al acabar, se le puede hacer clic derecho una vez más y seleccionar la opción de exportar la lista para ver el resultado en forma de texto de lo que se ha configurado de forma gráfica:

```
Nombre  Tipo    Datos   Marca de tiempo
(igual que la carpeta principal)    Inicio de autoridad (SOA)   [42],
win-7njvgt39ei8.fila3.2daw. iesmariaenriquez.es.,
hostmaster.fila3.2daw.iesmariaenriquez.es.  static
(igual que la carpeta principal)    Servidor de nombres (NS)
win-7njvgt39ei8.fila3.2daw. iesmariaenriquez.es.    static
(igual que la carpeta principal)    Host (A)    10.0.109.231    ?28/?10/?2022 17:00:00
_msdcs
_sites
_tcp
_udp
DomainDnsZones
ForestDnsZones
franco  Host (A)    10.0.109.26 static
pc31    Host (A)    10.0.109.50 static
pc32    Host (A)    10.0.109.51 static
win-7njvgt39ei8 Host (A)    10.0.109.231    static
```

Conclusión

En esta práctica no siento que haya aprendido nada, a diferencia de la última. Considero que se exáctamente lo mismo de DNS que sabía tras acabar SMX, porque sí, hemos estudiado teoría, pero saber el funcionamiento teórico del DNS no ayuda demasiado en muchos aspectos de su configuración si no sabes como configurar el servidor para haga lo que quieres que haga.

Y es justamente esa parte la que no hemos dado en clase. En su lugar, tenemos retringidos los servidores DNS que podemos usar de forma un tanto arbitraria, un objetivo general que al principio no sabíamos como llevar a cabo, y grupos demasiado grandes como para configurar unos pocos servidores en unos pocos días.

Porque sí, por supuesto que tiene sentido que entendamos como se configuren distintos tipos de servidores DNS, sobretodo entender la diferencia entre las abstracciones que hace todo el mundo y las que hace Windows a la hora de organizar dominios y hosts. Lo que no tiene sentido es dar palos de ciego con un servidor sólo para Linux que tras días de intertar hacerlo funcionar, descubrimos que no soporta una de las características clave para poder hacer lo que teníamos que hacer para completar el ejercicio, porque todos los servidores que son sólo para SO *Unix-like* son muy de nicho y están limitados en características, y los que realmente se usan regularmente en estos SO son multiplataforma.

Y sí, tiene sentido que este trabajo sea en grupos para similar la configuración de un *Domain Name System* distribuido, pero si tenemos el triple de personas que ordenadores a configurar, es casi imposible dividir tareas de forma efectiva, así que la gente que está más rezagada acaba sin poder hacer prácticamente nada, y por tanto no le sacan provecho a la práctica, no ganan experiencia, y acaban entendiendo poco más de lo que entendian al empezar.

Creo que si hubiesemos sido 3 grupos pequeños montando su propio *Domain Name System* distribuido, con dos servidores DNS seleccionados de antemano, para Linux y Windows (Sin la limitación de la exclusividad), habríamos sido más eficientes, todo el mundo habría tenido algo que hacer, y todo el mundo habría sacado algo de provecho. Además, se podría hacer que, si sale bien y hay tiempo dejar como tarea opcional conectar estos tres grupos al final, como tarea opcional.

No tengo ni idea de si es factible o no, o de si es realmente mejor o no, pero así es como me gustaría que hubiese sido la práctica. No creo que haga falta decir que esta práctica no me ha gustado.