

# Unidad 1

---



## Servicio de configuración automática

Servicios de red e internet



# Índice



## 1.1. Funcionamiento del servicio

- 11.1. Mecanismos automatizados de configuración
- 11.2. Formato del mensaje
- 11.3. Procedimiento y pautas en la solicitud de configuración de red

## 1.2. Asignación y tipos

## 1.3. Parámetros y declaraciones de configuración

- 13.1. Instalación de servidores de configuración de parámetros de red
- 13.2. Preparación del servicio para asignar configuraciones básicas de red
- 13.3. Configuración de asignaciones estáticas y dinámicas
- 13.4. Integración de opciones adicionales de configuración

## 1.4. Comandos utilizados para el funcionamiento del servicio



## Introducción

Durante el curso anterior aprendimos cómo funcionan los sistemas y su nivel de administración local, pero esto no solo se queda en un nivel básico local, sino que hay más opciones de configuración.

A lo largo de esta asignatura veremos los distintos servicios de red que se deben de implementar en un sistema en red para el correcto funcionamiento.

El primero de estos servicios es DHCP, que nos permitirá asignar a los equipos direcciones IP y demás configuraciones de red sin necesidad de ir configurándolos uno a uno.

A lo largo de la unidad veremos todos los mecanismos necesarios para la implementación del servicio y como se trabaja con él, sus opciones y sus posibilidades

## Al finalizar esta unidad

- + Sabremos cuales son los mecanismos automatizados de configuración de los parámetros de red y las ventajas que estos proporcionan.
- + Podremos preparar el servicio para asignar la configuración básica a los equipos de una red local.
- + Conoceremos como configurar asignaciones estáticas y dinámicas.
- + Seremos capaces de integrar en el servicio opciones adicionales de configuración.
- + Conoceremos los procedimientos y pautas que intervienen en una solicitud de configuración de los parámetros de red.
- + Seremos capaces de instalar servidores de configuración de los parámetros de red.

# 1.1.

## Funcionamiento del servicio

Cuando hablamos de servicio de configuración automática, nos queremos referir a DHCP

*DHCP* son las siglas de *Dynamic Host Configuration Protocol* (Protocolo de configuración dinámica de host) es un protocolo de red que trabaja en el nivel de aplicación y se encarga de asignar de manera automática las direcciones IP relativas a cada host, además de otros parámetros de red con la intención que de que el administrador no tenga que realizar tal tarea además de evitar colisiones de direcciones IP.

Entre sus numerosas ventajas destacan las siguientes:

- > Ahorro de tiempo por parte del administrador.
- > Mejor servicio al no depender de intervención externa.
- > Cambios de red por parte de equipos sin necesidad de excesiva configuración.
- > Elimina el uso de IP fijas en su mayoría, lo que es mejor para conexiones eventuales.
- > Se ahorran direcciones IP.

De manera general, este servicio es muy usado en la intranet de cualquier red, porque, por ejemplo, si nosotros nos ponemos una conexión internet en casa, el *router* lleva implementado el DHCP y cada vez que nos conectemos estaremos cogiendo una IP que se nos ha asignado.

Vamos a ver como trabajaría un servidor DHCP de manera resumida:

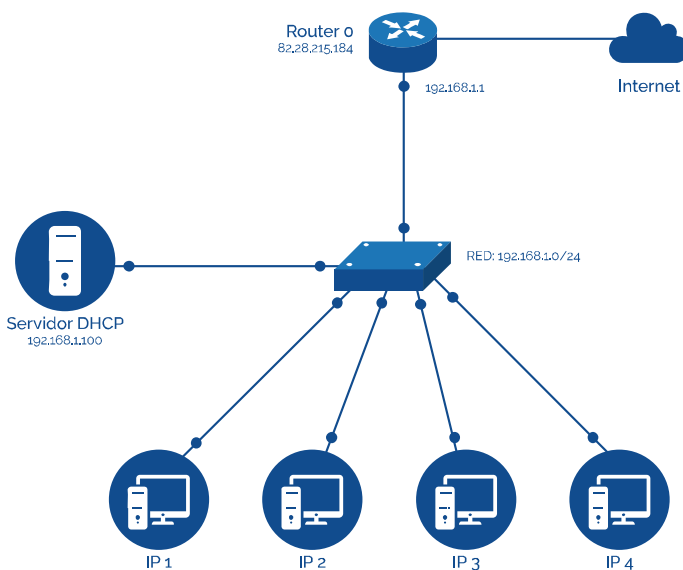


Imagen 1. Configuración de IP automática mediante DHCP

El diagrama anterior nos muestra como funcionaría el servidor DHCP, y sería del siguiente modo:

1. En primera instancia el *router* tendría una IP pública con la que se conectaría a internet y una IP privada dentro de la red, esta IP privada hará como default *gateway* de salida.
2. Después, tenemos un dispositivo, *switch*, *hub*... lo que sea, que hará de conmutador para establecer la red, también tendrá su propia IP.
3. Ahora, si el servidor de DHCP no lo tenemos en el *router*, sino a parte, se creará con una IP asignada fija y se establecerá el servicio.
4. En última instancia tenemos a los terminales finales que recibirán las IPs del servidor, del siguiente modo:
  - » Generalmente las primeras 10 se reservarán, de modo que se empezaría a asignar a partir de la propia 10, porque recordemos que la dirección 192.168.1.0 se cuenta.
  - » De ese modo, quedarían así si no pasa nada:
    - + IP 1: 192.168.1.10
    - + IP 2: 192.168.1.11
    - + IP3: 192.168.1.12
    - + IP4: 192.168.1.13

Para ejecutar todo esto será necesario que haya un administrador que mediante cierto *software* defina este tipo de direcciones, las IP reservadas, los rangos, y todo lo demás,

Las herramientas más usadas para esto son:

- > El rol de Servidor DHCP en Windows Server.
- > El paquete *isc-dhcp-server* en sistemas Linux.



### 1.1.1. Mecanismos automatizados de configuración

---

Cada vez que queremos que DHCP nos configure las direcciones de nuestros equipos, lo que realizamos es un diálogo con los servidores encargados de esta tarea en nuestro segmento de red hasta que nos otorgan los parámetros de la red necesarios para poder realizar nuestra configuración. Todos los mecanismos con los que trabaja DHCP se encuentran compuestos por los mensajes que clientes y servidores se envían mutuamente para llegar a la configuración acertada. El estándar regulador de DHCP es la *RFC 2131*.

Los mensajes mencionados anteriormente son:

- > **DHCPDISCOVER:** ese mensaje lo envía el equipo que solicita a IP como un mensaje de difusión para saber dónde se encuentran los servidores DHCP.
- > **DHCPOFFER:** la respuesta al mensaje anterior, la manda el servidor de DHCP al cliente que ha lanzado el *DHCPDISCOVER* en mensaje *unicast* con su oferta de parámetros de red. El cliente recibirá por lo general varios mensajes de este tipo, eligiendo solo una y descartando las otras.
- > **DHCPREQUEST:** en este mensaje *broadcast* el cliente informa de que oferta de configuración de red acepta y que todas las demás son declinadas.
- > **DHCPACK:** un mensaje de nuevo *unicast* mandado por el cliente al servidor indicando la confirmación de la configuración ofertada anteriormente.
- > **DHCPNACK:** otro mensaje *unicast* pero en este caso del cliente a los servidores no seleccionados indicando su declinación de la oferta.
- > **DHCPDECLINE:** mensaje *unicast* del cliente hacia el servidor donde se comenta que la configuración de la red está en uso.
- > **DHCPRELEASE:** en este mensaje que vuela a ser *unicast* mandado por el cliente hasta el servidor, se renuncia a la configuración asignada y se cancela el tiempo de concesión que se había otorgado.
- > **DHCPINFORM:** es un mensaje de nuevo del cliente al servidor en el que se le pide información adicional del primero al segundo.





### 1.1.2. Formato del mensaje

En el siguiente cuadro, podemos ver cuál es el formato de cada uno de los mensajes de DHCP y el tamaño de sus bytes:

| Campos de un mensaje DHCP |                |                 |               |
|---------------------------|----------------|-----------------|---------------|
| Op (1 byte)               | Htype (1 byte) | Hlen (1 byte)   | Hops (1 byte) |
| Xid (4 bytes)             |                |                 |               |
| Secs (2 bytes)            |                | Flags (2 bytes) |               |
| Ciaddr (4 bytes)          |                |                 |               |
| Yiaddr (4 bytes)          |                |                 |               |
| Siaddr (4 bytes)          |                |                 |               |
| Giaddr (4 bytes)          |                |                 |               |
| Chaddr (16 bytes)         |                |                 |               |
| Sname (64 bytes)          |                |                 |               |
| File (128 bytes)          |                |                 |               |
| Options (variable)        |                |                 |               |

Estos campos significan lo siguiente:

- > **Op:** es un código que nos indica la operación a la que pertenece el paquete, por ejemplo, 2 es BOOTREPLY.
- > **Htype:** se indica que dispositivo hardware se usa para hacer circular el paquete.
- > **Hlen:** nos indica el tamaño de la dirección hardware.
- > **Hops:** este campo, el cliente lo asigna a 0 mediante agentes de retransmisión (Relay Agent).
- > **Xid:** es el identificador de transacción. Se trata de un número aleatorio que escoge el cliente y se usa entre el cliente y el servidor para la asociación de mensajes y respuestas de forma única.
- > **Secs:** en este campo, que asigna el cliente, aparecen los segundos que han pasado desde el comienzo del proceso de adquisición de la dirección, o de renovación en otros casos.
- > **Flags:** indican los estados.
- > **Ciaddr:** la dirección de capa 3 del cliente. Si se ha solicitado concesión, el contenido del campo es 0.0.0.0, si no, la dirección IP que tenga en ese preciso momento.
- > **Yiaddr:** si el paquete es DHCP OFFER, aparece la dirección ofertada, si no, la del host.
- > **Siaddr:** la dirección IP del servidor siguiente, que lo responde el servidor DHCP en cuestión si estamos en mensajes de tipo DHCP OFFER o DHCP ACK.
- > **Giaddr:** dirección IP del agente de retransmisión.
- > **Chaddr:** dirección de la capa 2 del cliente.
- > **Sname:** nombre principal del servidor DHCP.
- > **File:** nombre del fichero de arranque.
- > **Options:** este campo es de parámetros opcionales.

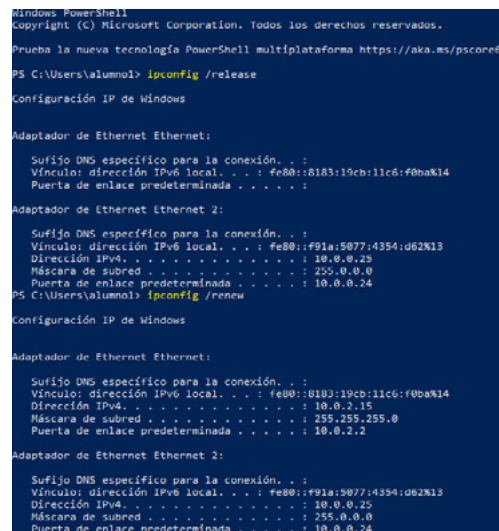
### 1.1.3. Procedimiento y pautas en la solicitud de configuración de red

El funcionamiento interno entre cliente y servidor de la configuración de la red mediante DHCP es el siguiente:

1. Lo primero que hace el cliente es enviar un *DCHPDISCOVER* en *broadcast* con la intención de solicitar la configuración a todos los *DHCP* alcanzables.
2. Los servidores que lo reciban mandan un *DHCPOFFER* en forma de *unicast*.
3. Se reciben las respuestas y se procesan en distintos tiempos.
4. Si no hay inconveniente alguno, se selecciona la primera por parte del cliente.
5. El cliente envía un mensaje *DHCPREQUEST* en *broadcast* para indicar cual ha sido la oferta seleccionada.
6. En el último paso, el servidor envía al cliente un mensaje de tipo *DHCPACK* confirmando la concesión.

Si queremos realizar con Windows el proceso de desprendernos de la dirección y solicitar una nueva, usamos los comandos:

`ipconfig /release`  
`ipconfig /renew`



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\alumno1> ipconfig /release

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufixo DNS específico para la conexión. . . : 
    Vínculo: dirección IPv6 local. . . : fe80::8183:19cb:11c6:f0ba%14
    Puerta de enlace predeterminada . . . . . : 

Adaptador de Ethernet Ethernet 2:

    Sufixo DNS específico para la conexión. . . : 
    Vínculo: dirección IPv6 local. . . : fe80::f91a:5077:4354:d62%13
    Dirección IPv4. . . . . : 10.0.0.25
    Máscara de subred . . . . . : 255.0.0.0
    Puerta de enlace predeterminada . . . . . : 10.0.0.24

PS C:\Users\alumno1> ipconfig /renew

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufixo DNS específico para la conexión. . . : 
    Vínculo: dirección IPv6 local. . . : fe80::8183:19cb:11c6:f0ba%14
    Dirección IPv4. . . . . : 10.0.2.15
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 10.0.2.2

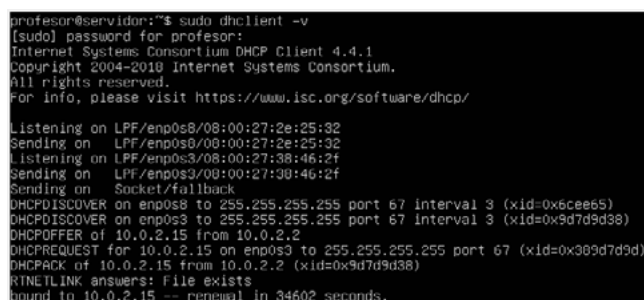
Adaptador de Ethernet Ethernet 2:

    Sufixo DNS específico para la conexión. . . : 
    Vínculo: dirección IPv6 local. . . : fe80::f91a:5077:4354:d62%13
    Dirección IPv4. . . . . : 10.0.0.25
    Máscara de subred . . . . . : 255.0.0.0
    Puerta de enlace predeterminada . . . . . : 10.0.0.24
```

Imagen 2. `ipconfig /release` y `/renew`

Para Linux, el comando idea sería escribir en una terminal:

`sudo dhclient -v`



```
profesor@servidor:~$ sudo dhclient -v
[sudo] password for profesor:
Internet Systems Consortium DHCP Client 4.4.1
Copyright 2004-2010 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Listening on LPF/enp0s8/08:00:27:2e:25:32
Sending on LPF/enp0s8/08:00:27:2e:25:32
Listening on LPF/enp0s3/08:00:27:38:46:2f
Sending on LPF/enp0s3/08:00:27:38:46:2f
Sending on Socket/fallback
DHCPDISCOVER on enp0s8 to 255.255.255.255 port 67 interval 3 (xid=0x6cee65)
DHCPDISCOVER on enp0s3 to 255.255.255.255 port 67 interval 3 (xid=0x9d7d9d38)
DHCPOFFER of 10.0.2.15 from 10.0.2.2
DHCPREQUEST for 10.0.2.15 on enp0s3 to 255.255.255.255 port 67 (xid=0x309d7d9d)
DHCPACK of 10.0.2.15 from 10.0.2.2 (xid=0x9d7d9d38)
RTNETLINK answers: File exists
bound to 10.0.2.15 -- renewal in 34602 seconds.
```

Imagen 3. `sudo dhclient -v`

# 1.2.

## Asignación y tipos

La *asignación* es la "apropiación" por parte de un equipo, de una dirección IP. Esta apropiación queda definida como un par dirección MAC-dirección IP y queda registrada en el equipo, y lógicamente en el servidor de DHCP, para saber que esa dirección ya se encuentra en uso. Se le puede llamar también *concesión*.

Como vimos en algunos temas del primer curso de diversas asignaturas, tenemos tres métodos de configuración de la dirección IP de un *host*, pero esto también se puede hacer mediante DHCP:

- > **Asignación manual o estática:** el servidor de DHCP se encarga de asignar de manera única cada dirección IP con la MAC que corresponda en la red. Esta asignación, tiene la particularidad de que nunca va a cambiar a no ser que sea el propio administrador del servicio el que cambie dichos valores. Este método suele ser muy usado en otros servidores con servicios fijos, o sujetos a monitorización.
- > **Asignación dinámica:** es la más común de las asignaciones de DHCP y la explicada anteriormente. El servidor elige una dirección IP entre las disponibles, llamado *rango de direcciones* o *pool de direcciones*.
- > **Asignación automática:** el servidor DHCP proporciona la configuración de la red a los *hosts* solicitantes de manera indefinida.

El tiempo que permanece activa la concesión de una IP para un cliente se denomina *tiempo de concesión* y una vez finalizado, el cliente deberá de intentar su renovación, lanzando un nuevo *DHCPDISCOVER* con muchos menos pasos, como podemos ver en la siguiente imagen:

Podemos ver que lo primero que se hace es una petición en *unicast* únicamente al servidor con el que ya teníamos la concesión, y si está todo correcto, el servidor nos devuelve un *DCHPACK* confirmando la renovación.

Para Windows, si ejecutamos el comando `ipconfig /all`, podemos ver lo siguiente:

```
Dirección física. . . . . : 08-00-27-F8-19-F6
DHCP habilitado. . . . . : sí
Configuración automática habilitada. . . . . : sí
Vínculo: dirección IPv6 local. . . . . : fe80::8183:19c6:11c6:f0ba%14(Preferido)
Dirección IPv4. . . . . : 10.0.2.15(Preferido)
Máscara de subred. . . . . : 255.255.255.0
Concesión obtenida. . . . . : lunes, 20 de junio de 2022 8:08:00
La concesión expira. . . . . : martes, 21 de junio de 2022 8:07:57
Puerta de enlace predeterminada. . . . . : 10.0.2.2
Servidor DHCP. . . . . : 10.0.2.2
IAID DHCPv6. . . . . : 101187623
DUID de cliente DHCPv6. . . . . : 00-01-00-01-29-80-22-18-08-00-27-F8-19-F6
```

Imagen 3. Información de DHCP en Windows

De la imagen de arriba sacamos que se indica lo siguiente (destacamos lo más importante):

- > **DHCP habilitado:** nos indica si la configuración es DHCP o no.
- > **Concesión obtenida:** nos dice en qué momento se realizó la concesión.
- > **La concesión expira:** aparece la fecha de fin de concesión.
- > **Servidor DHCP:** nos dice la IP de nuestro servidor de DHCP.

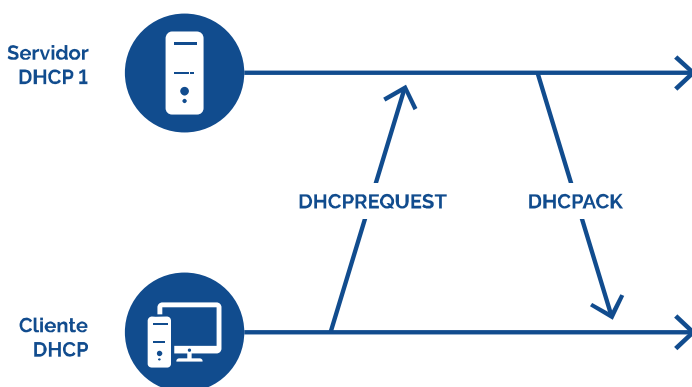


Imagen 4. Renovación de la concesión





# 1.3.

## Parámetros y declaraciones de configuración

Vamos a ver en este momento como se haría la configuración de un servidor de DHCP, esto lo haremos en Linux, con el sistema operativo *Ubuntu Server 21.10* (octubre 2021).

### 1.3.1. Instalación de servidores de configuración de parámetros de red

Lo primero que debemos de hacer, es, una vez *logueados* en nuestro servidor, pasar a ser *root* con el comando `sudo -i`. Nos pedirá la contraseña y ya tendremos los servicios de administración. Ahora, vamos a actualizar la lista de repositorios con el comando:

`apt update`

Cuando ya tenemos los repositorios actualizados, actualizamos los paquetes a sus versiones más recientes con el comando:

`apt upgrade`

Si durante este comando se nos pide una autorización, decimos que sí, y si se nos pide la reinstalación de algún servicio, lo dejamos por defecto con los seleccionados.

Una vez realizado esto, vamos a instalar el *software* de configuración de DHCP servidor (*Dynamic Host Configuration Protocol Daemon*). Con este *software* vamos a determinar las configuraciones de red que queremos que tengan los *hosts* que nos soliciten concesión. El paquete que debemos de instalar va a ser *isc-dhcp-server*, y su comando de instalación es:

`apt install isc-dhcp-server`

```
root@servidor:~# apt install isc-dhcp-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
isc-dhcp-server ya está en su versión más reciente (4.4.1-2.3ubuntu1).
El paquete indicado a continuación se instaló de forma automática y ya no es necesario.
libfwupdplugin1
```

Imagen 4. Instalación del paquete

De nuevo, si nos pide confirmación o reinstalación, decimos que sí.

#### PARA TENER EN CUENTA...

Si no hemos entrado en la sesión de root de primeras, todos los comandos deberán de ir acompañados de `sudo`.



### 1.3.2. Preparación del servicio para asignar configuraciones básicas de red

Para poder trabajar con el servicio de dhcp, necesitamos usar estos dos ficheros:

- > **/etc/default/isc-dhcp-server**: en este archivo guardamos el valor de las interfaces del servidor físicas que van a recibir todas las peticiones de DHCP de los clientes.
- > **/etc/dhcp/dhcpd.conf**: en este fichero guardaremos los parámetros de configuración que se va a establecer en los clientes.

Los mensajes que se envía desde el cliente hasta el servidor entrarán en este último por el puerto 67, y en caso contrario, por el puerto 68. Ambos paquetes son UDP.

Lo primero que debemos de hacer es comprobar la red, para ver cuál es nuestra interfaz de escucha, con el comando **ip a**.

Esta interfaz debe de ser la de la red interna, lógicamente, y tener una IP asignada. De igual modo, la configuración de la red en Ubuntu Server la veremos en la asignatura *Administración de Sistemas Operativos*, y se basa en *netplan*.

```
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:2e:25:32 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.24/24 brd 10.0.0.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe2e:2532/64 scope link
        valid_lft forever preferred_lft forever
```

Una vez que sabemos que interfaz vamos a usar a modo de escucha, nos dirigimos al fichero **/etc/default/isc-dhcp-server**.

```
root@servidor:~# cd /etc/default
root@servidor:/etc/default# ls
amd64-microcode  cron      grub.d      isc-dhcp-server  mtd-news      rsync
apparmor         cryptdisks  grub.ucf-dist  keyboard        networkd-dispatcher  ssh
console-setup    dbus       intel-microcode  locale          open-iscsi       ufw
crda             grub       irqbalance      mdadm           pollinate        useradd
root@servidor:/etc/default#
```

Abrimos el fichero y vemos que todo lo que aparece con '#' se encuentra en color azul, esto nos indica que son comentarios, es decir, que no tienen participación en la configuración.

En la línea final, vemos que tenemos una sección **INTERFACESv4** e **INTERFACESv6**, para indicar que interfaz va a ser la que va a escuchar para peticiones de direcciones IPv4 e Ipv6 respectivamente.

Como podemos ver en la imagen siguiente, indicamos la interfaz deseada.

```
GNU nano 5.6.1      isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf).
#DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

# Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
#DHCPDv4_PID=/var/run/dhcpd.pid
#DHCPDv6_PID=/var/run/dhcpd6.pid

# Additional options to start dhcpd with.
# Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
# Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
INTERFACESv6=""
```



El siguiente de los ficheros que vamos a ver, se encuentra en `/etc/dhcp` y es `dhcpd.conf`.

```
root@servidor:/etc/default# cd /etc/dhcp/
root@servidor:/etc/dhcp# ls
ddns-keys  dhclient.conf      dhclient-exit-hooks.d  dhcpd.conf
debug      dhclient-enter-hooks.d  dhcpd6.conf
```

Una vez que entramos en el fichero, tenemos dos tipos diferentes de sentencias en dicho fichero:

- > **Parámetros:** son las sentencias que nos permiten asignar un valor a cierta opción para que tenga información de configuración, y tienen la siguiente sintaxis:

**Nombre\_parámetro valor/valores**

Algunos parámetros se pueden escribir también sin especificar el valor, o bien porque entonces es por defecto o bien porque solo se necesitan nombrar para aportar la información.

- > **Declaraciones:** son conjuntos de parámetros que se agrupan entre '}'. En las declaraciones también podemos especificar otras declaraciones, y su sintaxis es del siguiente modo:

```
Declaración {
    Parámetros;
    [Declaración{}];
}
```

Tenemos cinco tipos distintos de declaraciones:

- » **Subnet.** Este tipo de opciones nos indican la red bajo la cual se van a realizar las configuraciones de los clientes. Mínimo debe de aparecer el parámetro *range*, que es el que indica el rango de direcciones que se pueden coger de esa red para asignarlas a los clientes. Su sintaxis la podemos encontrar en la siguiente imagen, que además es un comentario del fichero `dhcpd.conf`.

```
#subnet 10.254.239.0 netmask 255.255.255.224 {
# range 10.254.239.10 10.254.239.20;
# option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
#}
```

Imagen 5. Subnet

- » **Shared-network.** Se usa para cuando en un mismo segmento de red, tenemos distintas subredes. Cuando esto ocurre, debemos de tener en el adaptador físico del servidor, una IP para cada una de las subredes a las que les vamos a dar configuración. Esto quiere decir, que, en esta declaración, tendremos varias declaraciones *subnet* dentro, una por cada subred, lógicamente, y la sintaxis quedaría como la de la imagen de más abajo.
- » **Pool.** El llamado *pool* de direcciones es una lista de permisos que deben de cumplir todos los miembros que se encuentren dentro de estos grupos, aquí definidos (o mediante *class*), su sintaxis la vemos en la siguiente imagen.

```
#shared-network 224-29 {
# subnet 10.17.224.0 netmask 255.255.255.0 {
#   option routers rtr-224.example.org;
# }
# subnet 10.0.29.0 netmask 255.255.255.0 {
#   option routers rtr-29.example.org;
# }
# pool {
#   allow members of "foo";
#   range 10.17.224.10 10.17.224.250;
# }
# pool {
#   deny members of "foo";
#   range 10.0.29.10 10.0.29.230;
# }
```

Imagen 6. Shared network & pool



- » **Host.** Se usa para la reserva estática de direcciones a algunos equipos específicos, de modo que debemos de indicar la dirección física del dispositivo, y la dirección que queremos que tenga siempre, además de parámetros opcionales, como en todas las declaraciones. Su sintaxis es:

```
#host fantasia {
#   hardware ethernet 08:00:07:26:c0:a5;
#   fixed-address fantasia.example.com;
#}
```

Imagen 7. Host

- » **Class.** Es una agrupación de declaraciones para que se tengan unos parámetros específicos, y estas son sobre las que se aplican los *pool*. EN la imagen siguiente vemos su sintaxis, que es la más simple:

```
#class "foo" {
#   match if substring (option vendor-class-identifier, 0, 4) = "SUNW";
#}
```

Imagen 8. Class

Por último, es importante conocer los comandos con respecto al servicio de DHCP pues se van a ir usando conforme vayamos realizando distintas configuraciones. Estos comandos son:

- > Para iniciar el servicio *isc-dhcp-server*:  
`systemctl start isc-dhcp-server`
- > Para parar el servicio:  
`systemctl stop isc-dhcp-server`
- > Para reiniciar el servicio:  
`systemctl restart isc-dhcp-server`
- > Para comprobar el estado del servicio y ver si está o no activo:  
`systemctl status isc-dhcp-server`

```
root@servidor:/etc/dhcp# systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2022-06-20 09:06:40 UTC; 31s ago
     Docs: man:dhcpd(8)
    Main PID: 19999 (dhcpd)
      Tasks: 4 (limit: 2209)
    Memory: 4.4M
       CPU: 6ms
    OGroup: /system.slice/isc-dhcp-server.service
           └─19999 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhcpd.pid -cf /etc/dhcpd.conf

Jun 20 09:06:40 servidor sh[19999]: Listening on LPF/enp0s8/08:00:27:2e:25:32/10.0.0.0/24
Jun 20 09:06:40 servidor dhcpd[19999]: Sending on  LPF/enp0s8/08:00:27:2e:25:32/10.0.0.0/24
Jun 20 09:06:40 servidor sh[19999]: Sending on  LPF/enp0s8/08:00:27:2e:25:32/10.0.0.0/24
Jun 20 09:06:40 servidor dhcpd[19999]: Sending on  Socket/fallback/fallback-net
Jun 20 09:06:40 servidor sh[19999]: Sending on  Socket/fallback/fallback-net
Jun 20 09:06:40 servidor dhcpd[19999]: Server starting service.
Jun 20 09:06:41 servidor dhcpd[19999]: DHCPDISCOVER from 08:00:27:2e:25:32 via enp0s8
Jun 20 09:06:42 servidor dhcpd[19999]: DHCPOFFER on 10.0.0.26 to 08:00:27:2e:25:32 (servidor) via enp0s8
Jun 20 09:06:42 servidor dhcpd[19999]: DHCPREQUEST for 10.0.0.26 (10.0.0.24) from 08:00:27:2e:25:32 via enp0s8
Jun 20 09:06:42 servidor dhcpd[19999]: DHCPACK on 10.0.0.26 to 08:00:27:2e:25:32 (servidor) via enp0s8
lines 1-21/21 (END)
```

Imagen 9. Estado del servicio *isc-dhcp-server*, activo

### PARA TENER EN CUENTA...

Todos los comandos anteriores necesitan permisos de administrador para poder ser ejecutados.



### 1.3.3. Configuración de asignaciones estáticas y dinámicas

Vamos a ver ahora, ejemplos de cómo debe de configurarse el servicio de DHCP para direcciones estáticas, lo menos usado, y dinámicas, que es lo más usado.

#### Estática

En este punto vamos a explicar como se debe de configurar el servicio de DHCP para que se asigne una dirección estática o dirección de *reserva*, que es, como hemos dicho antes, la que se asocia con una dirección MACM específica y que mantiene dicha dirección IP indefinidamente.

Para realizar esta asignación, nos vamos al fichero `/etc/dhcp/dhcpd.conf`. En el apartado anterior, vimos que existen las declaraciones tipo *host* para justo estos casos, y eso va a ser lo que vamos a hacer, insertar una de este tipo. Estas declaraciones llevan los siguientes dos parámetros fijos:

- > **Hardware Ethernet MAC:** MAC es la dirección física del cliente que va a tener la dirección fija.
- > **Fixed-address IP:** aquí indicamos la dirección IUP que queremos que se asigne de manera estática al cliente en cuestión.

De modo que, si en nuestro fichero pusiéramos la configuración específica para configurar un *host* con la dirección IP 10.0.0.51:

```
host debian2 {  
    hardware ethernet 08:00:27:af:86:ad;  
    fixed-address 10.0.0.51;  
}
```

Imagen 10. Configuración estática 1

Hay que reiniciar ahora el servicio de DHCP mediante el comando,

**systemctl restart isc-dhcp-server**

Y ver su estado, si está activo, quiere decir que todo está correcto.

Acto seguido podemos comprobar que dicho *host*, que tiene la MAC antes mencionada, tiene la dirección IP elegida asignada.

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER UP> mtu 1500 qdisc fq_codel state UP gr  
oup default qlen 1000  
    link/ether 08:00:27:af:86:ad brd ff:ff:ff:ff:ff:ff  
    inet 10.0.0.51/24 brd 10.0.0.255 scope global dynamic enp0s3  
        valid_lft 3591sec preferred_lft 3591sec  
    inet6 fe80::a00:27ff:feaf:86ad/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever
```

Imagen 11. Configuración estática 2





## Dinámica

Si, por otra parte, queremos configurar nuestro servidor DHCP para que dé servicio de manera automática y dinámica, tendremos que usar la declaración *subnet*. Dentro de esta declaración, podemos usar muchos parámetros, pero los principales son:

- > **Subnet dirección\_red netmask máscara\_red.** Es la manera de abrir la declaración y nos va a indicar que se abre una nueva subred que declarar.
- > **Range IP\_inicio IP\_fin.** Este es el único de los parámetros obligatorios, e indica el rango de direcciones IP que el servidor puede asignar a los ordenadores que pidan concesión de direcciones IPv4.
- > **Option routers gateway\_IP.** Indicamos la IP del que queremos que sea el *router* de los equipos configurados.
- > **Option domain-name-servers IPdns, IPdns.2..., IPdns3.** Es el parámetro que indica los servidores DNS a los que va a consultar el cliente en caso de necesidad.
- > **Default-lease-time tiempo.** Tiempo en segundos que va a durar la concesión en caso de que el cliente no pida un tiempo específico.
- > **Max-lease-time tiempo.** Si el cliente, una vez terminado el tiempo de concesión, pidiera un tiempo adicional, se usaría este valor como máximo.

En el siguiente ejemplo, vamos a configurar un rango de direcciones para la red 10.0.0.0/24.

Lo primero que hacemos es configurar el fichero *dhcpd.conf* con la declaración *subnet* como podemos ver en la siguiente imagen.

```
subnet 10.0.0.0 netmask 255.255.255.0 {  
    range 10.0.0.26 10.0.0.50;  
    option routers 10.0.0.24;  
    option domain-name-servers 8.8.8.8, 8.8.4.4;  
    default-lease-time 3600;  
    max-lease-time 7200;  
}
```

Imagen 12. Configuración dinámica 1

Podemos ver que el rango de direcciones va desde la 26 hasta la 50, que el *gateway* es nuestro servidor, que los DNS son los de Google y que el tiempo por defecto de concesión son 3600 segundos (una hora) y el máximo es de 7200 segundos (dos horas).

Como antes, debemos de reiniciar el servicio y comprobar su estado.

Una vez hecho lo anterior, si nos vamos a cualquier cliente de la misma red que tenga el DHCP activado, podemos comprobar que efectivamente se le ha asignado una dirección IP dentro del rango que se ha especificado.



```
root@debian:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:56:c3:1f brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.27/24 brd 10.0.0.255 scope global dynamic noprefixroute enp0s3
        valid_lft 3489sec preferred_lft 3489sec
    inet6 fe80::a00:27ff:fe56:c31f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Imagen 13. Configuración dinámica 2

Otra comprobación que podemos hacer es consultar el fichero de concesiones. Este fichero se encuentra en la ruta `/var/lib/dhcp` y se llama `dhcpd.leases` (podemos comprobar que hay otro para clientes y otro para IPv6):

```
root@servidor:/var/lib# cd dhcp/
root@servidor:/var/lib/dhcp# ls
dhclient.leases dhcpd6.leases dhcpd6.leases~ dhcpd.leases dhcpd.leases~
root@servidor:/var/lib/dhcp#
```

Imagen 14. Configuración dinámica 3

Si abrimos el fichero, y buscamos por la dirección que hemos visto se ha asignado, podemos ver:

- > La IP de concesión.
- > El inicio de la concesión.
- > El fin de la concesión.
- > El estado actual.
- > La dirección MAC y el nombre del cliente

```
lease 10.0.0.27 {
    starts 1 2022/06/20 09:08:53;
    ends 1 2022/06/20 10:08:53;
    cltt 1 2022/06/20 09:08:53;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 08:00:27:56:c3:1f;
    uid "\001\010\000'\0303\037";
    client-hostname "debian";
}
```

Imagen 15. Configuración dinámica 4

Dentro del sistema, también podríamos consultar el fichero `/var/log/syslog` que se usa para consultar los logs del sistema, y significa que aquí también aparecerá el registro del servidor DHCP.



### 1.3.4. Integración de opciones adicionales de configuración

Ya hemos visto la que sería una configuración básica y más que suficiente para un sistema normal, pero a continuación vamos a ver las otras tres declaraciones opcionales, que, aunque no obligatorias, si son interesantes de saber.

#### Declaración de subredes con *shared-network*

La declaración *shared-network* se usa cuando existen varias subredes en una misma subred física. Tenemos entonces que declarar mínimo dos subredes dentro de esta declaración.

En el ejemplo que vamos a mostrar a continuación, indicamos los DNS y el *gateway* para todos los equipos de la red en general y luego partimos la red que teníamos en un principio en dos redes distintas indicando sus rangos de asignación de IP.

Nos fijamos en la declaración en la imagen siguiente:

```
shared-network mi_red {
    option domain-name-servers 8.8.8.8, 8.8.4.4;
    option routers 10.0.0.24;
    subnet 10.0.0.0 netmask 255.255.255.128 {
        range 10.0.0.26 10.0.0.50;
    }
    subnet 10.0.0.128 netmask 255.255.255.128 {
        range 10.0.0.150 10.0.0.200;
    }
}
```

Imagen 16. Declaración de *shared-network*

Ahora, como siempre, reiniciamos el servicio y comprobamos que se encuentra activo.

Una vez hecho esto, para comprobar que se han ido asignado las redes, podemos recurrir al archivo *dhcpd.leases* en */var/lib/dhcp* y vemos algo como lo siguiente:

```
lease 10.0.0.27 {
    starts 2 2022/06/21 07:44:32;
    ends 2 2022/06/21 07:54:32;
    cltt 2 2022/06/21 07:44:32;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 08:00:27:56:c3:1f;
    uid "\001\010\000\0303\037";
    client-hostname "debian";
}
lease 10.0.0.20 {
    starts 2 2022/06/21 07:45:25;
    ends 2 2022/06/21 07:55:25;
    cltt 2 2022/06/21 07:45:25;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 08:00:27:af:86:ad;
    uid "\001\010\000\0257\0206\0255";
    client-hostname "debian";
}
lease 10.0.0.29 {
    starts 2 2022/06/21 07:46:39;
    ends 2 2022/06/21 07:56:39;
    cltt 2 2022/06/21 07:46:39;
    binding state active;
    next binding state free;
    rewind binding state free;
    hardware ethernet 08:00:27:af:86:ad;
    client-hostname "debian";
}
```

Imagen 17. Visión de las concesiones

Podemos ver como se han ido haciendo distintas concesiones a un mismo dispositivo.



## Declaración de *pool* con listas de permisos

Los *pool* son agrupaciones de direcciones que se definen por alguna característica en concreto, por ejemplo, podríamos determinar un rango de direcciones únicamente aplicadas a clientes que conocemos. Estas declaraciones debemos de agruparlas dentro de las declaraciones de *shared-network* o de *subnet*. Realmente, solo se pueden usar tres estados para definir un *pool*, y son:

- > **Known-clients:** son los clientes que ya han recibido la dirección del servidor e intentan renovar su concesión.
- > **Unknown-clients:** son los clientes que no han obtenido concesión por parte del servidor y que han intentado conseguir una nueva.
- > **Member of "class":** si se ha definido previamente una clase con alguna característica, como puede ser que tenga una MAC con un final específico, se usan solo dichos equipos.

En el siguiente ejemplo, definimos una clase que indica que una MAC empieza por un número determinado y luego añadimos esa línea a nuestro *pool* de direcciones.

```
class "debian1" {  
    match if substring (hardware,1,3)="08:00:27";  
}  
  
subnet 10.0.0.0 netmask 255.255.255.0 {  
    pool {  
        range 10.0.0.26 10.0.0.50;  
        option routers 10.0.0.24;  
        option domain-name-servers 8.8.8.8, 8.8.4.4;  
        default-lease-time 3600;  
        max-lease-time 7200;  
        allow members of "debian1";  
    }  
}
```

Imagen 18. Class & Pool

Podemos observar de la imagen anterior que los parámetros son los mismo que en el anterior ejemplo de *subnet* que definimos, pero insertándolos en el *pool* y definiendo que solo se añadan a los miembros pertenecientes a dicha clase.

Si no se declara nada más, los que no estén dentro de esas situaciones, no obtendrán IP por parte del servidor.



# 1.4.

## Comandos utilizados para el funcionamiento del servicio

Vamos a mostrar ahora en forma de cuadro, los comandos más útiles para el servicio de DHCP en Ubuntu Server con *isc-dhcp-server*.

| Comandos systemctl               |  |
|----------------------------------|--|
| Acción                           | Comando  |
| Comprobar el estado del servicio | <code>systemctl status isc-dhcp-server</code>  |
| Iniciar el servicio              | <code>systemctl start isc-dhcp-server</code>   |
| Detener el servicio              | <code>systemctl stop isc-dhcp-server</code>    |
| Reiniciar el servicio            | <code>systemctl restart isc-dhcp-server</code> |

A parte, hay una serie de ficheros que sería interesante saber que existen para comprobar en caso de fallo del servicio:

| Ficheros de configuración de DHCP         |   |
|---|---|
| Nombre del fichero                        | Función   |
| <code>/etc/netplan/fichero.yaml</code>    | Fichero de configuración de red en sistemas Ubuntu. En este se definen todos los parámetros correspondientes, asignación estática o dinámica, dns, etc. |
| <code>/etc/dhcp/dhcpd.conf</code>         | Fichero principal de configuración del servicio.  |
| <code>/etc/init.d/isc-dhcp-server</code>  | Script de inicio del servicio.  |
| <code>/etc/default/isc-dhcp-server</code> | Fichero de identificación de interfaces para DHCP.  |
| <code>/var/lib/dhcp/dhcpd.leases</code>   | Fichero de registro de concesiones del servicio.  |
| <code>/usr/bin/dhcpd</code>               | Ejecutable del servicio. Este se corresponde con el <i>daemon</i> que gestiona peticiones y respuestas de los clientes.                                 |





 [www.universae.com](http://www.universae.com)

