

DHCP

Tu calle, 123

12345 Tu ciudad (Provincia)

<https://github.com/vcarceler/cirdan-dhcp/blob/master/dhcpd.conf>

DHCP

16 de Noviembre del 2021

DHCP 16.11.21 TEORIA

DHCP Dynamic Host Control Protocol, permite que un servidor especifique parámetros de red de manera automática a las máquinas que lo solicitan cuando se conectan a la red.

Parámetros básicos son la dirección de IP y la máscara de red. Si se quiere comunicar con otros hosts de otra red,

necesita la puerta de enlace y alomejor el servidor DNS.

Como resumen de DHCP primero emite un:

1. Discover: Pregunta una petición de UDP Broadcast.
2. Offer: El servidor ofrece que IP tiene disponibles, realiza un ping a ver si alguien lo tiene ya.
3. Request: El cliente tras la negociación del servidor, este le concede uso de una IP y otros parámetros de red al cliente por un tiempo determinado.

4. Acknowledge: El host recibe un acuse de recibo.

Es importante saber:

- El cliente no necesita tener la interfaz de red configurada para hacer un DHCP Discover.
- El servidor DHCP debe estar en la misma subred que el cliente.
- Cuando el servidor DHCP recibe la petición DHCP Discover, obtiene la MAC del cliente. Depende de cómo esté configurada, se le asigna una IP al cliente.
- El servidor DHCP realiza la entrega de la dirección IP por un tiempo determinado.

ISC-DHCP-SERVER

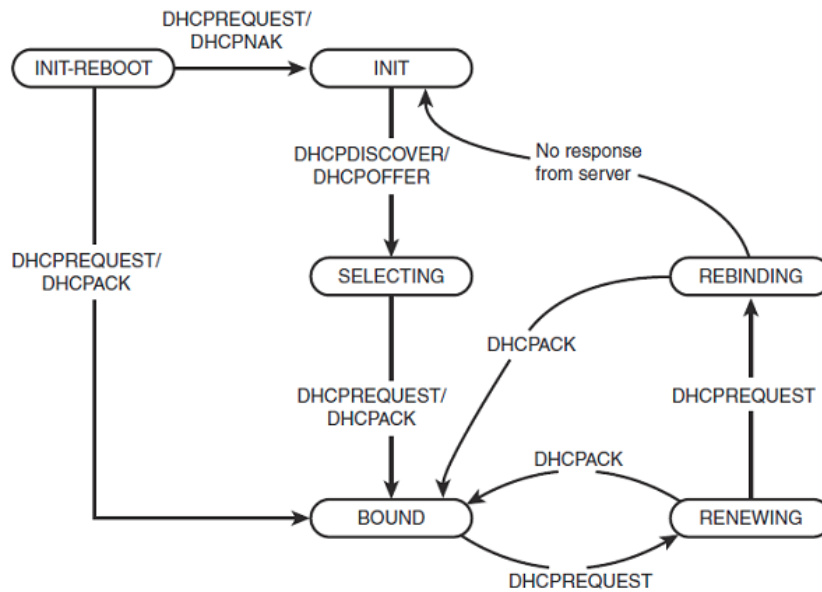
Configurar un servidor DHCP para que asigne configuraciones de red, se pueden hacer de tres maneras diferentes:

- **Asignación manual:** Se asigna una configuración en función de la **MAC del cliente**. El administrador debe escribir la tabla de correspondencias del servidor. Los clientes con MAC desconocida no se atienden.
- **Asignación dinámica:** El servidor DHCP dispone de **uno o varios rangos (pools)** de IP (Al menos uno por cada segmento de red por el que se atienden peticiones de clientes). Cuando el servidor recibe una **solicitud**, éste selecciona una de las pools adecuadas y se le asigna con un tiempo de alquiler (**lease time**). El tiempo se podrá prorrogar o en caso contrario, el cliente debe dejar la IP y comenzar una nueva solicitud.
- **Asignación automática:** La especificación del DHCP, el cual el servidor DHCP asigna una IP como en la **asignación dinámica, pero de forma permanente**.

Aspectos a tener en cuenta

- Hay 3 tipos de asignaciones IP: **Asignación estática (MAC)**, **Asignación dinámica (1 o varias Pools)** y **Asignación automática (Permanente)**.
- Antes lo dirigía BOOTP (Bootstrap Protocol).
- El cliente realiza un broadcast de UDP de **DHCP Discover** (incluyendo su MAC para saber si está en una filtro por MAC o no), el servidor DHCP, contesta con un **DHCP Offer** para saber que tiene una IP libre con un tiempo de alquiler y ciertas condiciones apropiada para el cliente, el cliente negocia la IP con el servidor, le vuelve a mandar un **DHCP Request** para quedarse con esa IP. El servidor DHCP, **manda un DHCP Acknowledge**, un acuse de recibo.
- Pueden haber varios servidores DHCP con next-server IP por si alguno falla, ya que es un servicio crítico.

-
- De otro modo:



- - El cliente no tiene una IP válida, estado **INIT**.
 - Se mueve al estado **SELECTING** cuando pregunta por una IP. (Pasar por DHCP Discover y DHCP Offer).
 - Cuando está en el último paso se pasa al estado **BOUND**, cuando ya tiene todo configurado. (Pasa por DHCP Request y DHCP ACK).
 - Cuando se reinicia el cliente, se pasa al estado **INIT-REBOOT** y **vuelve a hacer todo el proceso si la ip es válida**.
 - Antes de que expire el tiempo de alquiler, entra en estado **RENEWING**. Envía un mensaje unicast al servidor al que tuvo su **IP**. Si no recibe respuesta de renovación, se pasa al estado **REBINDING**, difunde un **Broadcast** para extender su contrato de alquiler en cualquier **servidor disponible**.
-

-

```
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

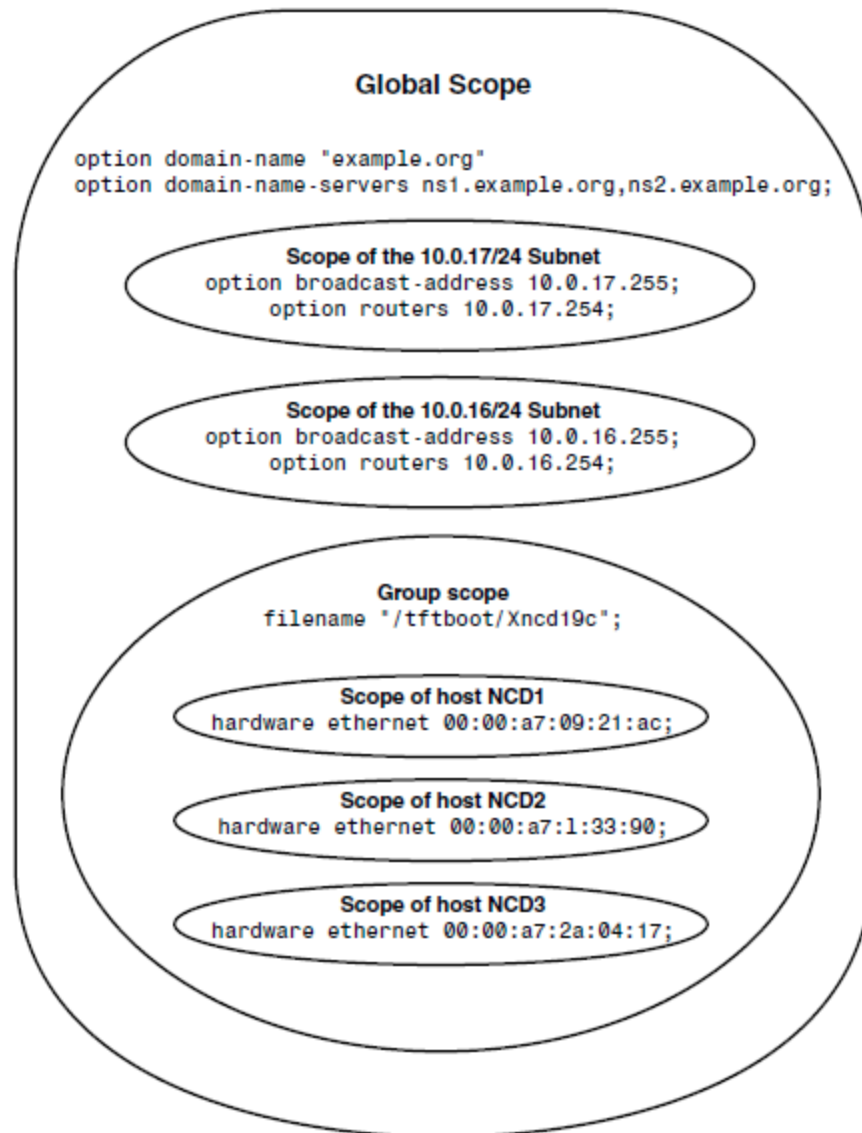
subnet 10.0.17.0 netmask 255.255.255.0 {
    option broadcast-address 10.0.17.255;
    option routers 10.0.17.254;
    range 10.0.17.10 10.0.17.253;
}

subnet 10.0.16.0 netmask 255.255.255.0 {
    option broadcast-address 10.0.16.255;
    option routers 10.0.16.254;
    range 10.0.16.10 10.0.16.253;
}

# Terminal NCD X

group {
    filename "/tftpboot/Xncd19c";
    host ncd1 { hardware ethernet 00:00:a7:09:21:ac; }
    host ncd2 { hardware ethernet 00:00:a7:1c:33:90; }
    host ncd3 { hardware ethernet 00:00:a7:2a:04:17; }
}
```

- Este ejemplo lo podemos ver de esta forma:



-
- Si el cliente que **solicita IP** pertenece al segmento de la subred **10.0.16.0/24** obtendrá las opciones del ámbito específico de la subred más las del ámbito global. Pero si el mismo cliente tuviera la **MAC 00:00:a7:09:21:ac**, entonces también pertenecería al ámbito creado por la declaración **host NCD1**, por lo que a las opciones anteriores, se les unirían las opciones del anterior ámbito, más las del ámbito **Group scope** y las **globales** que ya estaban.

- Cuando un **cliente** pertenece a **varios ámbitos** y una opción o parámetro aparece **varias veces**, el **último** que se aplica en el **orden de evaluación** es el que **prevalece**, y este orden es de mayor a menor especificidad de las declaraciones que crean los ámbitos.
- La especificidad de una **declaración** es menor cuanto menos clientes pueden asociarse con ella y a la inversa. El orden de las declaraciones de mayor a menor especificidad es el siguiente:

1. **shared-network**
2. **pool**
3. **subnet (Varias subnets forman un shared)**
4. **class**
5. **subclass**
6. **host**

- Es importante recordar que cuando un **cliente** pertenece a un **ámbito específico**, también se tienen en cuenta todos los ámbitos que contienen a éste hasta llegar al **ámbito global**.
- Para el último ejemplo el orden de aplicación de los ámbitos fue el siguiente:

1. **Global**

2. **subnet**
3. **group**
4. **host**

- Los ámbitos que no vinculan a ningún cliente, como por ejemplo el global o el creado por la declaración **group**, no tienen una especificidad concreta, por lo que se aplican como consecuencia de **contener** a otros ámbitos más específicos.

Group = Contenedor

-
- **Authoritative:** Es importante en un servidor DHCP, podrá enviar mensajes DHCP ACK a aquellos clientes que hagan un DHCP Request.
 - **ddns-update-style:** Sirve si hará actualizaciones de DNS.
 - **subnet:** Define subredes de un segmento de red.
 - **shared-network:** Se le asignan más de una subred.
 - **range:** Se usa para especificar el rango de IP que tiene una subred disponible. Puede haber más de un range, para ello se usa **POOL**.

```
subnet 10.0.0.0 netmask 255.255.255.0 {  
    range 10.0.0.10 10.0.0.254;  
}
```

- **pool:** Se usa para crear un o varios rangos de direcciones IP con una lista de permisos que deberán **cumplir completamente**.

```
pool {  
    [ lista de permisos ]
```



```
[ declaración range ]  
  
[ estamentos ]  
  
}  
  
pool {  
  
    deny all clients;  
  
    range 10.0.0.10 10.0.0.99;  
  
}
```

- **option domain-name-servers:** Se emplea para enviar al cliente los servidores DNS que utilizará. Se especifican por orden de preferencia.

```
option domain-name-servers ns.midns.com;

option domain-name-servers 8.8.8.8, 8.8.4.4;

subnet 10.0.0.0 netmask 255.255.255.0 {
    range 10.0.0.10 10.0.0.254;
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

- **option routers** se usa para enviar al cliente su puerta de enlace.
- **option broadcast-address**: Se usa para enviar al cliente la dirección de Broadcast de la subred. No es necesario especificarla.
- **option subnet-mask**: Especificar la máscara de la subred (Solo si tenemos máscaras diferentes).
- **host**: Para asociar un cliente DHCP a una MAC de cliente. Tiene 3 objetivos.
 - Asignar una IP estática a un cliente.
 - Todo cliente que posea una declaración de **host** se define como **conocido (known)**. Esto se utilizará con estamentos condicionales para asignar información en función si el cliente es conocido o desconocido (**unknown**).
 - Crear un ámbito en el que se puedan ejecutar estamentos para un cliente determinado.
- **hardware**: Se usa para usar **ethernet**, es decir especificar la MAC.

```
host Aula224PC01 {
    hardware ethernet 99:aa:bb:0c:ad:e2;
}
```

- **option dhcp-client identifier**: Se usa para especificar al cliente cuando éste no se identifica con la declaración **hardware**.

```
host PC14 {
    option dhcp-client-identifier "\0mi-equipa";
}
```

Es como

```
host PC14 {
    option dhcp-client-identifier 1:8:0:2b:4c:72:17;
}
```

- **fixed-address:** Se usa para especificar la IP que se asignará al cliente, se que se vincule con una declaración host.

```
host PC14 {
    hardware ethernet 00:20:78:11:F9:14;
    fixed-address 192.168.11.252;
}
```

- **Min-lease-time:** El tiempo de alquiler mínimo. En segundos
- **default-lease-time:** El tiempo de alquiler por defecto. En segundos
- **max-lease-time:** El tiempo de alquiler máximo. En segundos.
- **La declaración group se usa para crear contenedores. Por ejemplo para agrupar declaraciones host.**
-

Permisos

- **Known-clients:** Tiene una declaración *host*. Se pretende permitir o prevenir la asignación de una IP de un pool a los clientes que tengan una declaración **host**.

[allow | deny] known-clients;

Ejemplo:

```
pool {  
    allow known-clients;  
    range 10.0.0.10 10.0.0.99;  
}
```

- **unknown-clients:** Con este permiso se pretende permitir o prevenir la asignación de una IP de un *pool* a los clientes que no tengan una declaración *host*. La sintaxis es la siguiente:

[allow | deny] unknown-clients;

Ejemplo:

```
pool {  
    deny unknown-clients;  
    range 10.0.0.10 10.0.0.99;  
}
```

- **members of:** Si pertenecen a una **clase**. Permitir o denegar la asignación de una IP de un pool a los clientes que pertenecen a una clase.

-

[allow | deny] members of "clase";

Ejemplo:

```
pool {  
    allow members of "primero";  
    range 10.0.0.10 10.0.0.99;  
}
```

- **all clients:** Permitir o denegar la asignación de IP de un POOL a todos los clientes.

```
pool {  
    deny all clients;  
    range 10.0.0.10 10.0.0.99;  
}  
  
pool {  
    allow all clients;  
    range 10.0.0.100 10.0.0.199;  
}
```

Aspectos a tener en cuenta

Una *clase* es un agrupamiento de clientes que cumplen las mismas condiciones basadas en la información que estos envían, es un agrupamiento más general que el que se hace con la declaración *host*. Las clases se crean con la declaración **class** y un cliente pertenecerá a una clase si concuerda con todas las reglas de dicha clase o bien concuerda con una de las subclases de la clase.

La sintaxis de la declaración *class* es la siguiente:

```
class "nombre-clase" {  
  
    [ estamentos ]  
  
    [ declaraciones ]  
  
}
```

El nombre de la clase es lo que se utiliza con el permiso *member of*.

match if

El estamento **match if** se utiliza para especificar una expresión booleana (condición) que el cliente debe cumplir para ser asignado a la clase. Su sintaxis es la siguiente:

```
match if expresión-booleana;
```

La expresión booleana normalmente dependerá de algún valor enviado por el cliente al servidor.

En el ejemplo siguiente un cliente pertenecerá a la clase *ventas* si éste ha enviado en la opción *departamento* el valor de *ventas*.

```
class "ventas" {  
    match if option departamento = "ventas";  
}
```

match

El estamento ***match*** se utiliza para asociar un cliente DHCP a una clase mediante la asociación previa a una subclase. Su sintaxis es la siguiente:

match expresión-de-datos;

La expresión de datos debe ser una expresión que devuelva un determinado dato a partir de la información que llega desde el cliente y que se utilizará para compararlo mediante coincidencia exacta con los valores ligados a las subclases.

En el siguiente ejemplo se utiliza la opción *dhcp-client-identifier* para asignar los clientes a la clase *aula224*:

```
class "aula224" {  
    match option dhcp-client-identifier;  
}
```

Para completar la asociación del cliente con la clase, se deben crear subclases donde se especifique el valor exacto con el que tiene que coincidir el valor de la opción *dhcp-client-identifier* del cliente.

Ejemplo: dhcpd.conf

```
ddns-update-style none;
subnet 192.168.0.0 netmask 255.255.255.0 {
    # default gateway
    option routers 192.168.0.1;
    option subnet-mask 255.255.255.0;

    # DNS
    option domain-name "domain.org";
    option domain-name-servers 192.168.0.10;

    # Pool
    allow unknown-clients;
    range 192.168.13.100 192.168.13.254;

    default-lease-time 21600;
    max-lease-time 43200;

    # IP fija
    host ns {
        hardware ethernet 12:34:56:78:AB:CD;
        fixed-address 192.168.0.10;
    }
}
```

Ejercicio práctico

DHCP.conf

```
option domain-name "informatica.edt";  
option domain-name-servers 8.8.8.8;  
default-lease-time 600;  
max-lease-time 7200;  
ddns-update-style none;  
authoritative;  
  
allow booting; #Se inicia automáticamente el servicio DHCP  
  
subnet 10.0.0.0 netmask 255.255.255.0 {  
    pool {  
        option router 10.0.0.1;  
        range 10.0.0.100 10.0.0.200;  
        next-server 10.0.0.1; #Nuestra IP, del servidor  
        filename "pxelinux.0";  
    }  
}
```

```
subnet 10.0.1.0 netmask 255.255.255.0 {  
    pool {  
        option router 10.0.1.1;  
        range 10.0.1.100 10.0.1.200;  
        next-server 10.0.1.1; #Nuestra IP, del servidor  
        filename "pxelinux.0";  
    }  
}  
  
subnet 10.0.2.0 netmask 255.255.255.0 {  
    pool {  
        option router 10.0.2.1;  
        range 10.0.2.100 10.0.2.200;  
        next-server 10.0.2.1; #Nuestra IP, del servidor  
        filename "pxelinux.0";  
    }  
}  
  
group {
```

```
option domain-name-servers 1.1.1.1;

host maq1 {
    hardware ethernet 22:22:22:22:22:01;
    default-lease-time 800;
}

host maq2 {
    hardware ethernet 22:22:22:22:22:02;
}

host maq3 {
    hardware ethernet 22:22:22:22:22:03;
}
```

Ejemplos en clase

DIRECTIVA GROUP

```
-----  
  
group {  
    option domain-name-servers 192.168.10.8;  
  
    subnet 192.168.10.0 netmask 255.255.255.128 {  
        pool {  
            range 192.168.10.11 192.168.10.40;  
            allow known-clients;  
        }  
    }  
}  
  
group {  
    option domain-name-servers 192.168.10.9;  
  
    subnet 192.168.10.128 netmask 255.255.255.128 {  
        pool {  
            range 192.168.10.129 192.168.10.140;  
            deny unknown-clients;  
        }  
    }  
}  
  
-----  
  
subnet 192.168.10.0 netmask 255.255.255.0 {  
    pool {  
        range 192.168.10.10 192.168.10.30;  
        allow known-clients;  
    }  
}
```

```
pool {
    range 192.168.10.40 192.168.10.50;
    allow unknown-clients;
}

group {
    option domain-name-servers 192.168.10.9;
    host maq1 {
        hardware ethernet 52:54:00:a9:e0:f8;
    }

    host maq2 {
        hardware ethernet 52:54:00:70:8f:55;
    }
}

# Esta configuración hace los que estén en el GROUP, les dará una ip entre 10 a 30 y los
# desconocidos a partir del 40.

# Se graba mediante la MAC del usuario
```

DIRECTIVA CLASS

```
class "profes" {  
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:52:54:0";  
}
```

```
subnet 192.168.10.0 netmask 255.255.255.0 {  
    pool {  
        range 192.168.10.40 192.168.10.50;  
        allow members of "profes";  
    }  
}
```

```
host maq1 {  
    hardware ethernet 52:54:00:a9:e0:f8;  
}  
host maq2 {  
    hardware ethernet 52:54:00:70:8f:55;  
}
```

```
# El primero es ethernet "1:0:60:60";
```

Última

08.11.21 - DHCP GROUP / CLASS

Classes ASIX

DHCP --> 3 interfaces 10.0.0.1 / 24 || CAMBIAR INTERFACES ALIAS auto enp7s0:0 (ALIAS)

ASIX - 10.0.0.1 / 24

SMX - 10.0.1.1 / 24

DAM - 10.0.2.1 / 24

```
class "profes" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:52:54:0";
}
```

SUBSTRING COGE LOS 4 PRIMEROS BYTES

BINARY TO ASCII --> LOS CONVIERTE A ASCII, LOS CONVIERTE EN HEXADECIMAL. DE 8 EN 8.

Se unen con : a Hexadecimal. Se unen con :

07 --> Lo convierte en 7 solo 1 numero

El primer byte, indica el tipo interfaz. El 1 es Ethernet = Dirección de red Ethernet.

El primero es Ethernet.

Ethernet1 y su MAC. "1:52:54:0";

#

1 : F0 : 7 : 31

#

#


```
class "asix" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:aa:aa:aa";
}
```

```
class "smx" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:ee:ee:ee";
}
```

```
class "dam" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:cc:cc:cc";
}
```

Cambiar la MAC

```
class "asix" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:aa:aa:aa";
}
```

```
class "smx" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:bb:bb:bb";
}
```

```
class "dam" {
    match if binary-to-ascii (16,8,":",substring(hardware, 0, 4)) = "1:cc:cc:cc";
}
```

```
subnet 10.0.0.0 netmask 255.255.255.0 {
    pool {
        range 10.0.0.20 10.0.0.30;
        allow members of "asix";
    }
}
```

```
}  
  
subnet 10.0.1.0 netmask 255.255.255.0 {  
    pool {  
        range 10.0.1.20 10.0.1.30;  
        allow members of "smx";  
    }  
}  
  
subnet 10.0.2.0 netmask 255.255.255.0 {  
    pool {  
        range 10.0.2.20 10.0.2.30;  
        allow members of "dam";  
    }  
}
```

Cambiar la interface PARA TENER VARIAS INTERFÍCIES

source-directory /run/network/interfaces.d

```
auto enp1s0  
iface enp1s0 inet dhcp
```

```
auto enp7s0  
iface enp7s0 inet static  
    address 10.0.0.1/24
```

```
auto enp7s0:0  
iface enp7s0:0 inet static  
    address 10.0.1.1/24
```

```
auto enp7s0:1  
iface enp7s0:1 inet static  
    address 10.0.2.1/24
```

Ver resultado

```

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    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: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:f8:55:60 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.141/24 brd 192.168.100.255 scope global dynamic enp1s0
        valid_lft 3555sec preferred_lft 3555sec
    inet6 fe80::5054:ff:fef8:5560/64 scope link
        valid_lft forever preferred_lft forever
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 52:54:00:60:b8:96 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.1/24 brd 10.0.0.255 scope global enp7s0
        valid_lft forever preferred_lft forever
    inet 10.0.1.1/24 brd 10.0.1.255 scope global enp7s0:0
        valid_lft forever preferred_lft forever
    inet 10.0.2.1/24 brd 10.0.2.255 scope global enp7s0:1
        valid_lft forever preferred_lft forever
    inet6 fe80::5054:ff:fe60:b896/64 scope link
        valid_lft forever preferred_lft forever

```

Cambiar el /etc/default/isc-dhcp-server

INTERFACESv4="enp7s0 enp7s0:0 enp7s0:1"

Comprobar

```

root@SERVIDOR:/etc/default# nano isc-dhcp-server
root@SERVIDOR:/etc/default# systemctl restart networking.service
root@SERVIDOR:/etc/default# systemctl restart isc-dhcp-server.service
root@SERVIDOR:/etc/default# systemctl status isc-dhcp-server.service

```

```

● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Mon 2021-11-08 10:58:06 UTC; 5s ago
     Docs: man:systemd-sysv-generator(8)
    Process: 1481 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited,
status=0/SUCCESS)
     Tasks: 4 (limit: 1132)
    Memory: 6.4M
       CPU: 29ms
    CGroup: /system.slice/isc-dhcp-server.service
           └─1496 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf enp7s0 enp7s0:0 enp7s0:1

```

```

Nov 08 10:58:04 SERVIDOR systemd[1]: Starting LSB: DHCP server...
Nov 08 10:58:04 SERVIDOR isc-dhcp-server[1481]: Launching IPv4 server only.
Nov 08 10:58:04 SERVIDOR dhcpd[1496]: lease 192.168.10.40: no subnet.
Nov 08 10:58:04 SERVIDOR dhcpd[1496]: lease 192.168.10.41: no subnet.
Nov 08 10:58:04 SERVIDOR dhcpd[1496]: lease 192.168.10.42: no subnet.
Nov 08 10:58:04 SERVIDOR dhcpd[1496]: Wrote 0 class decls to leases file.
Nov 08 10:58:04 SERVIDOR dhcpd[1496]: Wrote 0 leases to leases file.
Nov 08 10:58:04 SERVIDOR dhcpd[1496]: Server starting service.
Nov 08 10:58:06 SERVIDOR isc-dhcp-server[1481]: Starting ISC DHCPv4 server: dhcpd.
Nov 08 10:58:06 SERVIDOR systemd[1]: Started LSB: DHCP server.
root@SERVIDOR:/etc/default#

```

Abrir las máquinas