

# Servicio DHCP en Ubuntu

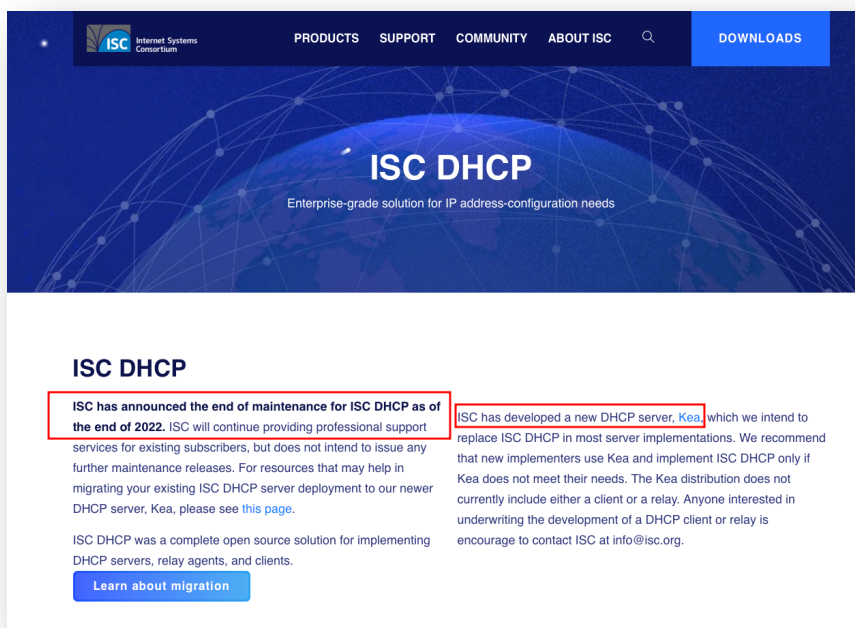
Este servicio se utiliza para configurar automáticamente los parámetros del protocolo IP en los equipos de una red. El mecanismo consiste en que los clientes DHCP en los hosts soliciten información TCP/IP. El servidor DHCP escucha la solicitud y asigna una dirección IP. Esta asignación puede ser:

- Manual, cuando el administrador de la red asigna manualmente una dirección IP.
- Automática, cuando la asignación se produce en el primer contacto y se mantiene permanentemente.
- Dinámica, cuando la asignación se realiza por un periodo de tiempo y al finalizar se devuelve.

En cualquiera de los casos, el mecanismo es el mismo. El cliente DHCP realiza una solicitud de configuración al puerto 67 de una dirección especial de broadcast, la 255.255.255.255.

En Ubuntu se usaba ISC DHCP como ejecutable por defecto para ofrecer servicio de DHCP pero desde finales de 2022 ya no tiene soporte desde su web oficial.

En la misma web oficial nos indican que el nuevo desarrollo y sucesor será Kea. Las principales diferencias es que ISC-DHCP era un software con arquitectura monolítica donde un único binario hacía todo y su configuración se guardaba en texto plano normalmente en `"/etc/dhcp/dhcpd.conf"`. En cambio Kea tiene una arquitectura modular y hay un binario para cada componente (DHCPv4, DHCPv6, bases de datos) además, los archivos de configuración se basan en archivos JSON o bases de datos. Otra novedad es mediante su API REST podemos gestionar configuraciones en tiempo real.



The screenshot shows the ISC DHCP website with a dark blue header and a network-themed background. The main heading is "ISC DHCP" with the subtitle "Enterprise-grade solution for IP address-configuration needs". Below this, a white box contains the announcement text. Two red boxes highlight key information: the end of maintenance for ISC DHCP and the introduction of Kea as a replacement.

**ISC DHCP**

ISC has announced the end of maintenance for ISC DHCP as of the end of 2022. ISC will continue providing professional support services for existing subscribers, but does not intend to issue any further maintenance releases. For resources that may help in migrating your existing ISC DHCP server deployment to our newer DHCP server, Kea, please see [this page](#).

ISC DHCP was a complete open source solution for implementing DHCP servers, relay agents, and clients.

[Learn about migration](#)

ISC has developed a new DHCP server, Kea, which we intend to replace ISC DHCP in most server implementations. We recommend that new implementers use Kea and implement ISC DHCP only if Kea does not meet their needs. The Kea distribution does not currently include either a client or a relay. Anyone interested in underwriting the development of a DHCP client or relay is encouraged to contact ISC at [info@isc.org](mailto:info@isc.org).

## 1. Instalación

Todas las distribuciones actuales de Linux incluyen en sus repositorios el software Kea. Así que empezaremos refrescando los repositorios para que se carguen con las últimas versiones de todos los programas.

```
sudo apt update
```

La instalación de Kea se puede hacer descargando y compilando los binarios más recientes desde la propia web de ISC (Internet Systems Consortium). Es habitual que en los repositorios de la propia distribución Linux se incluya una versión ligeramente anterior.

Por simplificar el proceso en esta guía instalaremos directamente desde los repositorios usando el siguiente comando:

```
sudo apt install kea-dhcp4-server
```

```
administrador@equipo:~$ sudo apt install kea-dhcp4-server
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  kea-common liblog4cplus-2.0.5 libmysqlclient21 libpq5 mysql-common
Paquetes sugeridos:
  kea-doc
Se instalarán los siguientes paquetes NUEVOS:
  kea-common kea-dhcp4-server liblog4cplus-2.0.5 libmysqlclient21 libpq5 mysql-common
0 actualizados, 6 nuevos se instalarán, 0 para eliminar y 266 no actualizados.
Se necesita descargar 4.783 kB de archivos.
Se utilizarán 19,6 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n]
```

Durante la instalación:

- se han descargado e instalado los ejecutables (keactrl y kea-dhcp4) en `"/usr/sbin/kea-dhcp4"`
- se ha generado un archivo de configuración en `"/etc/kea/kea-dhcp4.conf"`
- se ha creado un servicio llamado `"kea-dhcp4-server.service"` encargado de lanzar el servicio DHCP en cada arranque.
- se ha creado el usuario y grupo `"_kea"` para gestionar el servicio DHCP



```
ss -lun
```

Este comando devuelve todos los puertos udp (-u) que están a la escucha (-l) y, además nos muestra el número del puerto usado (-n).

```
usuario@nodo1:~$ ss -lun
```

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process
UNCONN	0	0	0.0.0.0:58402	0.0.0.0:*	
UNCONN	0	0	127.0.0.54:53	0.0.0.0:*	
UNCONN	0	0	127.0.0.53%lo:53	0.0.0.0:*	
UNCONN	0	0	10.0.6.100:67	0.0.0.0:*	
UNCONN	0	0	0.0.0.0:5353	0.0.0.0:*	
UNCONN	0	0	:::5353	:::*	
UNCONN	0	0	:::37605	:::*	

```
usuario@nodo1:~$
```

En la salida del comando anterior podemos confirmar que tenemos a nuestro equipo escuchando en el puerto 67 udp.

Ya tenemos nuestro servidor DHCP funcionando, solo queda probarlo ejecutando un "sudo netplan try" en cualquier otro equipo de la red que tenga su configuración IP en automático.

## Estructura del archivo de configuración

El archivo donde se guarda la configuración de kea-dhcp4 está en "/etc/kea/kea-dhcp4.conf".

Al abrirlos veremos muchas líneas comentadas que empiezan por "///". Estas líneas se tratan como comentarios y no se procesan. El resto de las líneas conforman un documento JSON que empieza en la llave "{" de la línea 24. A partir de ahí tenemos parejas de "clave: valor" donde la clave siempre es una cadena entre comillas y los valores pueden ser números, cadenas, booleanos (en minúscula), arrays u otros documentos anidados.

El archivo de configuración original tiene mucha más información de la que necesitamos para nuestro escenario inicial. Vamos a guardar este archivo original con otro nombre para mantener una copia de seguridad por si algo falla, lo llamaremos "/etc/kea/kea-dhcp4.conf.backup"

```
sudo mv /etc/kea/kea-dhcp4.conf /etc/kea/kea-dhcp4.conf.backup
```

Ahora que ya tenemos una copia de seguridad creamos un nuevo documento vacío con el mismo nombre.

```
sudo touch /etc/kea/kea-dhcp4.conf
```

Y lo configuramos de la siguiente manera:

```
{
  "Dhcp4": {
    # Valores globales
    "valid-lifetime": 3600,
```



```

"renew-timer": 900,
"rebind-timer": 1800,

# Identificación de las interfaces del servidor
"interfaces-config": {
  "interfaces": ["enp0s8"]
},

# Especificación del almacenamiento de las cesiones
"lease-database": {
  "type": "memfile",
  "persist": true,
  "name": "/var/lib/kea/dhcp4.csv",
  "lfc-interval": 3600
},

# Listado de subredes en las que ser ceden direcciones IP
"subnet4": [
  {
    "subnet": "172.16.1.0/24",
    "id": 100,
    "pools": [ { "pool": "172.16.1.100 - 172.16.1.199" } ],
    "option-data": [
      {
        "name": "routers",
        "data": "172.16.1.254"
      },
      {
        "name": "domain-name-servers",
        "data": "172.16.0.10, 172.16.0.11"
      },
      {
        "name": "domain-search",
        "data": "linux.lan"
      }
    ]
  }
]
}

```

## Valores globales

Estos valores son todos opcionales. En caso de no indicarlos en el archivo de configuración se tomarán los valores por defecto.

El parámetro **"valid-lifetime"** indica durante cuántos segundos la cesión de una IP es válida. Por defecto son 4000 segundos, algo más de 1 hora.

El parámetro **"renew-timer"** define el temporizador en el que se espera renovar la cesión de la IP con el mismo servidor DHCP. No parece tener mucho sentido esperar a perder la IP cuando se agote el "valid-lifetime" por lo que unos segundos antes (los definidos aquí en renew-timer) empezará a hacer intentos de renovación contra el servidor DHCP. Mas o menos este tiempo se suele configurar como el 50% del valid-lifetime.

El parámetro **"rebind-timer"** define el tiempo antes de que el cliente intente conectar con otro servidor DHCP cualquiera reiniciando una petición al puerto 67 de la IP de broadcast 255.255.255.255. La idea es la misma que con renew-timer. Se suele configurar al 87% del valid-lifetime.

La idea detrás de todos estos temporizadores es intentar una nueva renovación con el mismo servidor DHCP a partir del 50% de tiempo de cesión consumido. Si llegamos al 87% de tiempo de cesión y aún no hemos obtenido respuesta, entonces empezamos a intentar una cesión con cualquier otro servidor DHCP.

## Identificación de las interfaces

En esta sección se especifican las interfaces de red en las que el servidor debe escuchar peticiones DHCP. El parámetro **interfaces** guarda la lista de interfaces de escucha en forma de array entre corchetes y con sus elementos separados por comas.

En el siguiente ejemplo vemos cómo sería la configuración en un servidor DHCP que tuviera dos interfaces en las que quisiera atender peticiones DHCP

```
{  
  "interfaces-config": {  
    "interfaces": [ "enp0s3", "enp0s8" ]  
  },  
}
```

## Almacenamiento de las cesiones

Aunque podemos usar una base de datos para almacenar esta información, en esta guía usaremos una base de datos en memoria que mantiene los datos persistentes mediante un archivo del tipo CSV guardado en el sistema de ficheros del servidor.

La sección de **"lease-database"** será suficiente con un campo **"type"** y de valor **"memfile"**. En función del tipo de almacenamiento usado tendremos distintas opciones para configurar. El campo **"persist"** es un booleano que indica si los datos se guardarán de forma persistente en un fichero o solamente se almacenarán en memoria. En el caso de que sea persistente y se guarde en un fichero podemos usar **"name"** para indicar la ruta del archivo donde se guardarán las cesiones.

## Subredes

Esta es la sección más importante de todo el archivo de configuración. Aquí se definen todas las subredes de las que es servidor recibirá peticiones de configuración en forma de array de documentos, uno por subred, separados por comas.

En este ejemplo vemos la configuración de 3 subredes. Todas ellas en su propio documento y dentro del array.

```
{  
  "subnet4": [
```

```
{
  "id": 1,
  "pools": [ { "pool": "192.0.2.1 - 192.0.2.200" } ],
  "subnet": "192.0.2.0/24"
},
{
  "id": 2,
  "pools": [ { "pool": "192.0.3.100 - 192.0.3.200" } ],
  "subnet": "192.0.3.0/24"
},
{
  "id": 3,
  "pools": [ { "pool": "192.0.4.1 - 192.0.4.254" } ],
  "subnet": "192.0.4.0/24"
}
],
}
```

Para cada una de las subredes debemos definir un identificador numérico "id" que se asociará a cada una de las cesiones de IP que hagamos a esa determinada subred. La siguiente manera de identificar una subred es a través de su dirección de red y máscara que encontraremos bajo el campo "subnet".

El campo más importante es el "pools" donde encontraremos un array de uno o más pools de IPs en forma de rango. Por ejemplo, el siguiente archivo de configuración define dos pools de direccion IP no consecutivas

```
"Dhcp4": {
  "subnet4": [
    {
      "subnet": "192.0.2.0/24",
      "pools": [
        { "pool": "192.0.2.10-192.0.2.20" },
        { "pool": "192.0.2.64-192.0.2.127" }
      ],
      ...
    }
  ],
  ...
}
```

Otro de los campos que podemos indicar es el de proveer a los clientes de opciones de configuración extra como la dirección IP de la puerta de enlace (routers), servidores DNS (name-servers), servidores de dominio (domain-name-servers) y muchos otros (time-servers, default-ip-ttl, static-routes, smtp-server, pop-server, www-server, domain-search,...)

```
{
  "subnet4": [
    {
      "subnet": "192.0.2.0/24",
      "option-data": [
        {
          "name": "domain-name-servers",
          "data": "10.2.5.76, 10.2.5.77"
        }
      ],
      ...
    }
  ],
  ...
}
```

En el ejemplo anterior estamos configurando una subred a la que además se envía una IP indicando dos direcciones IP de servidores DNS.

Muchas de las propiedades pueden aparecer en lugares diversos del archivo de configuración con los mismos significados pero con ámbitos distintos. Por ejemplo, en este código solo se enviará información de los servidores DNS si el equipo recibe una IP de ese pool concreto



```
"Dhcp4": {
  "subnet4": [
    {
      "pools": [
        {
          "pool": "192.0.2.1 - 192.0.2.200",
          "option-data": [
            {
              "name": "domain-name-servers",
              "data": "192.0.2.3"
            },
            ...
          ],
          ...
        },
        ...
      ],
      ...
    },
    ...
  ],
  ...
},
...
],
...
}
```

En cambio, en el siguiente código, todas las IPs asignadas llevarán la información de los servidores DNS porque el "option-data" está en la zona de variables globales

```
"Dhcp4": {
  "option-data": [
    {
      "name": "domain-name-servers",
      "data": "192.0.2.1, 192.0.2.2"
    },
    ...
  ],
  "subnet4": [
    {
      "subnet": "192.0.3.0/24",
      ...
    },
    ...
  ],
  ...
},
...
}
```

## Reserva de IPs por MAC en una subred

En muchas ocasiones vamos a querer asignar la misma IP a un determinado nodo. Para ello usaremos su dirección MAC de manera que estableceremos una asignación manual:

```
{
  "subnet4": [
    {
      "id": 1,
      "pools": [ { "pool": "192.0.2.1 - 192.0.2.200" } ],
      "subnet": "192.0.2.0/24",
      "interface": "eth0",
      "reservations": [
        {
          "hw-address": "1a:1b:1c:1d:1e:1f",
          "ip-address": "192.0.2.202"
        },
        {
          "hw-address": "2a:2b:2c:2d:2e:2f ",
          "ip-address": "192.0.2.100",
          "hostname": "alice-laptop"
        }
      ]
    }
  ],
  ...
}
```





En este fragmento del archivo de configuración podemos ver como se reservan dos direcciones y en una de ellas se informa del nombre de host. También es interesante ver como se indica la "interface" que escucha el servidor DHCP. Este último parámetro no es realmente necesario porque se fija en la interfaz por la que llega una petición y aplica el pool que corresponde con la subred de la IP fija que el propio servidor DHCP tiene en esa interfaz.