

CONFIGURACIÓN DE RED EN LINUX

CONTENIDO

Comandos básicos de red de Linux.....	1
Información básica de red	1
Comando <code>ip</code>	1
Otros comandos	1
Redirigir paquetes	1
NAT	1
NAT de origen.....	1
Port forwarding	2
Cómo añadir y guardar reglas de cortafuegos	2
Cómo configurar una red usando Netplan	3
Sintaxis básica de YAML	3
Comentarios.....	3
Tipos básicos	3
Diccionarios	3
listas	3
Indentación	4
Archivos de configuración de Netplan	5
Configuración estática de red	5
Aplicar cambios	6
Ayuda de Netplan.....	6
Generación automática del archivo de configuración.....	7
Crear una interfaz puente a una interfaz física	7
Cómo configurar una red usando <code>interfaces</code>	9
Interfaz de <i>loopback</i>	9
Enrutamiento estático	9
Servidores DNS.....	10
Activar y desactivar interfaces	10
Aplicar cambios	11
Ayuda de <code>interfaces</code>	11

COMANDOS BÁSICOS DE RED DE LINUX

INFORMACIÓN BÁSICA DE RED

COMANDO IP

En Linux existe el comando `ip`, una navaja suiza de configuraciones de red. Sirve para modificar configuraciones de red (de forma temporal, ya que no se guarda en ningún sitio) y para ver el estado actual de las configuraciones.

A continuación se muestran comandos útiles de `ip` en su versión larga y en su versión abreviada.

Para ver la información a nivel de enlace sobre las interfaces de red (o sobre una de ellas):

```
$ ip link show [<interfaz>]
$ ip l
```

Para ver la información a nivel de red (más completa) sobre las interfaces de red (o sobre una de ellas):

```
$ ip address show [<interfaz>]
$ ip a
```

Para ver la tabla de enrutamiento del equipo:

```
$ ip route list
$ ip r
```

Para ver la tabla ARP del equipo (los vecinos de la red):

```
$ ip neighbour show
$ ip n
```

OTROS COMANDOS

Para enviar un paquete a una IP o un nombre de dominio:

```
$ ping <host>
```

Para ver la configuración DNS actual (solo si se tiene `resolveconf` instalado):

```
$ resolvectl
```

REDIRIGIR PAQUETES

Para permitir a un equipo redirigir (enrutar) los paquetes entrantes de una red a otra:

```
# echo 'net.ipv4.ip_forward=1' > /etc/sysctl.d/99-local.conf
# sysctl -p /etc/sysctl.d/99-local.conf
```

NAT

NAT DE ORIGEN

Si se desea que los equipos de una red tengan acceso al exterior (y acceso a internet), será necesario que el router frontera de la red realice NAT de origen (*Source NAT*) de los paquetes salientes mediante la siguiente regla de cortafuegos:

```
# iptables -t nat -A POSTROUTING -o <int_ext> -j MASQUERADE
```

Donde <int_ext> es la interfaz de la red externa.

PORT FORWARDING

Si lo que se quiere es hacer una redirección de puertos (es decir, exponer un servicio de la red interna a la red externa), se puede añadir una regla como la siguiente:

```
# iptables -t nat -A PREROUTING -i <int_ext> -p <proto> --dport  
<puerto_dest> -j DNAT --to <IP_int>
```

Donde <int_ext> es la interfaz de la red externa, <proto> es el protocolo de transporte de la petición (tcp o udp), <puerto_dest> es el puerto destino del paquete (por ejemplo, tcp y 80 para HTTP) y <IP_int> es la dirección IP del servidor destino de la red interna.

CÓMO AÑADIR Y GUARDAR REGLAS DE CORTAFUEGOS

DEBIAN/UBUNTU

En distribuciones de la familia Debian, es necesario tener instalados (apt install) los paquetes iptables e iptables-persistent (marca <Yes> a las dos preguntas que hace el instalador).

Teniendo todo instalado, solo hay que crear una regla NAT de cortafuegos y guardarla para evitar que se borre al apagar:

```
# apt install iptables iptables-persistent  
#  
# #-- Inserta las reglas iptables que desees --  
#  
# netfilter-persistent save
```

ALPINE

En la distribución Alpine, es necesario tener instalado (apk add) el paquete iptables y cargarlo al inicio (rc-update add).

Una vez hecho lo anterior, solo hay que crear una regla NAT de cortafuegos y guardarla para evitar que se borre al apagar:

```
# apk add iptables  
# rc-update add iptables  
#  
# #-- Inserta las reglas iptables que desees --  
#  
# /etc/init.d/iptables save
```

CÓMO CONFIGURAR UNA RED USANDO NETPLAN

Los sistemas Ubuntu Server delegan la gestión de la red en la utilidad Netplan, por lo que el/los archivos de configuración de red se encuentran en el directorio `/etc/netplan/`. Lo más común es que haya uno, y si hay varios, se aplicarán según sus nombres en orden lexicográfico.

SINTAXIS BÁSICA DE YAML

Los archivos de configuración (`*.yaml`) de Netplan usan el lenguaje YAML, un lenguaje declarativo que define objetos y valores de la forma:

COMENTARIOS

Cualquier texto que empiece en un `#` es un comentario. Los comentarios son ignorados por el procesador YAML.

Los comentarios se extienden entre el primer `#` y el final de la línea. Ejemplo:

```
# Esto es un comentario
foo: bar # Otro comentario
```

TIPOS BÁSICOS

Un valor de tipo básico puede ser:

- Un número entero (`70`, `0xFFD37A`) o de coma flotante (`2.65`, `1e+08`).
- Una cadena de texto (Esto es una cadena, "Esto es otra cadena\n").
- Valor nulo (`~`, `null`).
- Un booleano, pudiendo tener el valor verdadero (`True`, `Yes`, `On`) o falso (`False`, `No`, `Off`).

DICCIONARIOS

Los diccionarios son estructuras de datos que consisten en un conjunto de pares atributo-valor de cualquier tipo, incluso pueden ser otros objetos.

Un diccionario se puede definir en una sola línea:

```
{atributo1: valor1, atributo2: valor2, atributo3: valor3}
```

O en varias líneas:

```
atributo1: valor1
atributo2: valor2
atributo3: valor3
```

LISTAS

Una lista es una secuencia de valores que aparecen en un cierto orden.

Una lista se puede definir en una línea:

```
[valor1, valor2, valor3]
```

O en varias líneas:

```
- valor1
- valor2
- valor3
```

Esta última sintaxis es útil para definir una lista con objetos, definiendo sus atributos sobre la marcha:

```
nombre:
- atributo1: valor1
  atributo2: valor2
- atributo1: valor3
  atributo3: valor4
  otroAtributo: valor5
```

En el ejemplo de arriba, `nombre` es una lista que contiene 2 elementos: un objeto con 2 atributos (`atributo1`, `atributo2`) y un objeto con 3 atributos (`atributo1`, `atributo3`, `otroAtributo`).

Los elementos de una lista pueden ser de cualquier tipo, incluso objetos o listas. Además, una lista puede tener cualquier combinación de tipos de elementos:

```
bolsa:
- apellido: García
  edad: 77
  sangre:
    letra: A
    signo: "+"
- edad: 0
- La pared es roja
- 5
```

INDENTACIÓN

YAML usa la indentación para definir objetos y listas en varias líneas, por lo que al definir los elementos de una lista (en su versión extendida) y al definir los atributos de un objeto es necesario que todos sus elementos tengan la misma indentación. Así:

```
objeto1:
  sub1:
    attr1: 10
    attr2: hola
  sub2: miau
```

Este código YAML define un objeto (`objeto1`) que tiene dos atributos (`sub1` y `sub2`), siendo el primero un objeto que tiene otros dos atributos (`attr1` y `attr2`).

Cambiando así la indentación:

```
objeto1:
sub1:
  attr1: 10
  attr2: hola
sub2: miau
```

Este cambio provocará un fallo de sintaxis, ya que `attr2` está más indentado que `attr1`, y `attr1` es un número, por lo que no puede tener atributos. Sin embargo, cambiado así la indentación:

```
objeto1:
  sub1:
    attr1: 10
    attr2: hola
  sub2: miau
```

Ahora `attr2` está a la altura de los atributos `sub1` y `sub2`, por lo que es un atributo más de `objeto1`, no de `sub1` como antes.

ARCHIVOS DE CONFIGURACIÓN DE NETPLAN

El contenido del archivo de configuración por defecto de Netplan que encontraremos generalmente es:

```
network:
  version: 2
  ethernets:
    <interfaz1>:
      dhcp4: true
```

Donde `ethernets` es un objeto de `network` que define las interfaces de red de los adaptadores físicos de red cableada del sistema (cada atributo de `ethernets` es una interfaz de red Ethernet).

Podemos definir tantas interfaces de red como queramos, pero solo funcionarán si sus nombres corresponden con el nombre de la interfaz identificado por el sistema.

La configuración más básica de una interfaz es si usará una configuración IPv4 estática (`dhcp4: false`) o por DHCP (`dhcp4: true`). En este último caso, no necesitaremos definir más configuraciones de la interfaz, ya que deberán llegar por DHCP.

En esta guía trabajaremos exclusivamente con IPv4.

CONFIGURACIÓN ESTÁTICA DE RED

En caso de usar una configuración estática (`dhcp4: false`), se podrán definir los siguientes datos:

La **dirección IP** (o direcciones IP) de la interfaz en formato CIDR. Aunque solo se defina una, debe ser una lista:

```
addresses: [<IP>/<máscara>, ...]
```

Una **tabla de enrutamiento** básica. Se pueden añadir todas las rutas que se deseen, pero solo una de las interfaces debe tener configurado un *default gateway*. Para configurar un *default gateway*:

```
routes:
- to: default
  via: <IP>
```

También se pueden definir otros enrutamientos estáticos diferentes, siempre indicando la dirección IP de la red a la que se quiere acceder + la máscara de dicha red (`to`) y la dirección IP de salto (`via`):

```
routes:
...
- to: 10.100.0.0/20
  via: 10.1.0.2
- to: 10.100.16.0/20
  via: 10.1.0.3
...
```

El/los servidores DNS. Aunque solo se defina uno, debe ser una lista:

```
nameservers:
  addresses: [<IP>, ...]
```

Ejemplo de configuración de red:

```
network:
  version: 2
  ethernet:
    ens33:
      dhcp4: false
      addresses: [192.168.5.3/24]
      routes:
        - to: 192.168.6.0/24
          via: 192.168.5.4
    ens37:
      dhcp4: true
```

En el ejemplo anterior, se ha configurado la interfaz `ens33` con una IP estática en la red `192.168.5.0/24` con una ruta a la red `192.168.6.0/24`, y la interfaz `ens37` con una IP dinámica mediante DHCP (por aquí se nos definirá un default gateway, por lo que ponerlo en la otra interfaz podría ser erróneo).

APLICAR CAMBIOS

Si creamos o modificamos un archivo de configuración Netplan, no bastará con guardar los cambios, sino que habrá que aplicarlos con el siguiente comando (requiere permisos de administrador):

```
# netplan apply
```

Este comando buscará errores de sintaxis que haya en el archivo, y no se aplicarán los cambios si encuentra alguno.

AYUDA DE NETPLAN

Ante cualquier duda sobre los archivos de configuración de Netplan, se puede consultar su página del manual:

```
$ man 5 netplan
```

GENERACIÓN AUTOMÁTICA DEL ARCHIVO DE CONFIGURACIÓN

Muchas imágenes de contenedores y máquinas virtuales tienen una variante **cloud** (es el caso de las imágenes que tengan `/cloud` en el nombre, como `alpine/3.20/cloud`, y todas las imágenes del repositorio `ubuntu:`). Estas variantes usan la utilidad `cloud-init` para instalar programas y generar configuraciones en la instancia de forma automática en su primer inicio (o tras cada reinicio).

En contenedores Ubuntu, uno de los archivos que `cloud-init` cambia automáticamente es el fichero de configuración de Netplan `/etc/netplan/50-cloud-init.yaml`. Para evitar que se borren nuestras configuraciones en cada reinicio, hay dos vías posibles:

- a. Ejecuta la siguiente línea en el shell de la instancia:

```
# echo 'network: {config: disabled}' >/etc/cloud/cloud.cfg.d/99-disable-network-config.cfg
```

Tras esto, el fichero `/etc/netplan/50-cloud-init.yaml` dejará de actualizarse automáticamente, por lo que podremos personalizarlo libremente.

- b. Usa otro archivo de configuración dentro de `/etc/netplan` que tenga un nombre mayor que el existente (por ejemplo, `99-custom.yaml`, ya que `'9' > '5'`), y escribe en él la configuración que desees. Así, Netplan leerá el nuevo archivo después del otro y sobrescribirá las configuraciones que entren en conflicto con el otro fichero.

CREAR UNA INTERFAZ PUENTE A UNA INTERFAZ FÍSICA

Supongamos que tenemos la siguiente configuración estática de la interfaz física `eth0` en Netplan:

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: false
      addresses: [172.16.1.247/16]
      routes:
        - to: default
          via: 172.16.0.1
      nameservers:
        addresses: [1.1.1.1, 1.0.0.1]
```

Por diversos motivos, podríamos querer crear una interfaz puente sobre `eth0`. Para ello, trasladamos la configuración de `eth0` (excepto el atributo `dhcp4`) al nuevo puente (`br0` por ejemplo), además de añadir algunos parámetros extra a `br0`:

```
network:
  version: 2
  ethernets:
    eth0: {}
  bridges:
    br0:
      interfaces: [eth0]
      dhcp4: false
```



```
addresses: [172.16.1.247/16]
routes:
  - to: default
    via: 172.16.0.1
nameservers:
  addresses: [1.1.1.1, 1.0.0.1]
parameters:
  forward-delay: 4
```

No se pueden crear puentes sobre interfaces inalámbricas.

CÓMO CONFIGURAR UNA RED USANDO INTERFACES

En distribuciones como Debian o Alpine, la configuración de red se realiza mediante el fichero `/etc/network/interfaces` (en otras distribuciones puede estar en otra ruta).

Para definir una interfaz de red, se ha de incluir la siguiente línea:

```
auto <interfaz>
```

Donde `auto` determina que la interfaz de red (`<interfaz>`) se activará automáticamente al inicio del sistema.

La siguiente línea definirá si se configura estática o dinámicamente:

```
iface <interfaz> inet dhcp|static
iface <interfaz> inet6 auto|dhcp|static
```

- La primera línea (`inet`) configura IPv4, la segunda (`inet6`) configura IPv6. Se puede poner una de las dos o ambas.
- `dhcp` para configuración dinámica, `static` para configuración estática

Si se configura de forma dinámica (`dhcp`), no es necesario añadir más líneas a dicha configuración:

```
auto <interfaz>
iface <interfaz> inet dhcp
```

Para una configuración estática (`static`), se ha de definir su dirección IP, su máscara (en punto-decimal), su *default gateway* y sus servidores DNS:

```
auto <interfaz>
iface <interfaz> inet static
    address <IP>
    netmask <máscara>
    gateway <IP_gateway>
```

Solo una de las interfaces debe tener configurado un *default gateway*.

También se puede definir la dirección de broadcast de la red:

```
broadcast <IP_broadcast>
```

INTERFAZ DE LOOPBACK

La interfaz de *loopback* (`lo`) debería estar definida siempre en el fichero `interfaces`:

```
auto lo
iface lo inet loopback
```

ENRUTAMIENTO ESTÁTICO

Si se quieren definir enrutamiento estático más allá del *default gateway*, se puede añadir una línea como la siguiente por cada ruta estática que se quiera configurar:

```
post-up ip route add <IP_red>/< mascara> via <IP_gateway>
```

`<máscara>` se debe escribir en formato decimal.

Para que no falle al recargar la configuración, se debe desactivar la ruta al desactivar la interfaz, por lo que por cada línea `up` se ha de añadir una línea casi idéntica `down` (cambiando `add` por `del`):

```
pre-down ip route del <IP_red>/<mascara> via <IP_gateway>
```

SERVIDORES DNS

Para definir servidores DNS de la red, existen dos métodos:

ARCHIVO `RESOLV.CONF`

Este método solo se debe usar si no se tiene instalado el programa `resolvconf`.

El archivo de configuración `/etc/resolv.conf` contiene las direcciones IP de los servidores de nombres configurados en el sistema. Para definirlos/modificarlos, se ha de editar el fichero y, por cada servidor que se quiera incluir, se añadirá una línea con su dirección IP:

```
nameserver <IP>
```

Por ejemplo, para definir los servidores DNS de Google `8.8.8.8` y `8.8.4.4`:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Se puede ver la ayuda de este fichero mediante:

```
$ man 5 resolv.conf
```

PROGRAMA `RESOLVCONF`

El programa `resolvconf` es el encargado de modificar automáticamente el fichero `resolv.conf` a partir de distintas configuraciones, por lo que para configurar servidores de nombres en una máquina con `resolvconf` instalado, no se debe modificar el fichero `resolv.conf`, sino incluir la siguiente línea en una interfaz estática en `interfaces` con uno o más servidores DNS:

```
dns-nameservers <IP> <IP> ...
```

ACTIVAR Y DESACTIVAR INTERFACES

Para activar una interfaz inactiva configurada en `interfaces` (requiere permisos de administrador):

```
# ifup <interfaz>
```

Para desactivar una interfaz activa (requiere permisos de administrador):

```
# ifdown <interfaz>
```

También se pueden activar/desactivar todas las interfaces marcadas como `auto` con `-a`:

```
# ifup -a
# ifdown -a
```

`ifup` buscará errores de sintaxis que haya en el archivo `interfaces`, y no se activará(n) la(s) interfaz(ces) si encuentra alguno.

APLICAR CAMBIOS

Si hacemos cambios en alguna interfaz del fichero `interfaces`, no bastará con guardar los cambios, sino que habrá que aplicarlos de alguna manera:

- Reiniciando la máquina (`reboot`).
- Desactivando (`ifdown`) y activando (`ifup`) las interfaces modificadas (o todas con `-a`).
- Reiniciando el servicio que gestiona la red:

```
# service networking restart
```

Esta acción enviará solicitudes DHCP para las interfaces con configuración dinámica.

AYUDA DE INTERFACES

Ante cualquier duda sobre el archivo de configuración `interfaces`, se puede consultar su página del manual:

```
$ man 5 interfaces
```