

Depuración de la red

Este punto retoma un cierto número de comandos y de herramientas que ya se han visto en puntos anteriores en este mismo capítulo. Nos vamos a concentrar en los diferentes aspectos de la configuración de las capas de red que pueden provocar problemas de comunicación en la red, así como en los diferentes registros (*logs*) útiles para el diagnóstico de los problemas que se puedan presentar:

- ✓ Gestión de la red con `NetworkManager`.
- ✓ Archivos de configuración de red durante el inicio del sistema.
- ✓ Archivos de configuración de la resolución de nombres.
- ✓ Archivos de configuración relativos al control de accesos a los servicios en red.
- ✓ Archivos de registro del sistema.
- ✓ Problemas de enrutamiento remoto.

1. El servicio NetworkManager

`NetworkManager` es un servicio implementado desde hace unos años en Linux, cuyo objetivo es simplificar y automatizar la configuración de las interfaces de red.

Puede detectar interfaces de red disponibles, del tipo que sean (Ethernet, Wi-Fi), activarlas y proporcionar automáticamente una configuración de redes completa y adaptada al entorno de red actual, conjuntamente con un servidor DHCP.

Este servicio es particularmente interesante para los ordenadores portátiles y para un uso orientado a puesto de trabajo (cliente de red).

`NetworkManager` está compuesto por un proceso, cuyo inicio y paro están gestionados por `systemd`, y por comandos que permiten interactuar con él, en modo gráfico (`control-center`, `nm-connection-editor`) o en línea de comandos (`nmcli`, `nmtui`).

En un sistema servidor, que necesita generalmente una configuración de red estable y permanente, el uso del servicio `NetworkManager` para configurar dinámicamente las

interfaces de red tiene poco interés y puede provocar problemas.



Con las versiones recientes de las distribuciones de tipo Red Hat, este servicio es el que inicia la red, usando para ello archivos de configuración tradicionales (en `/etc/sysconfig/network-scripts`). Los scripts de inicio de red tradicionales ya no están presentes por defecto. Sin embargo, se pueden implementar instalando el paquete `network-scripts`.

Es necesario conocer los diferentes métodos de configuración y arranque de la red, para la certificación LPIC-2, usando `systemd` o `init System V`, y usando también los scripts tradicionales (`network scripts`) o `NetworkManager`. Sin embargo, el uso de este último para la configuración dinámica de las interfaces de red no entra dentro del programa de la versión actual de la certificación.

Ejemplo

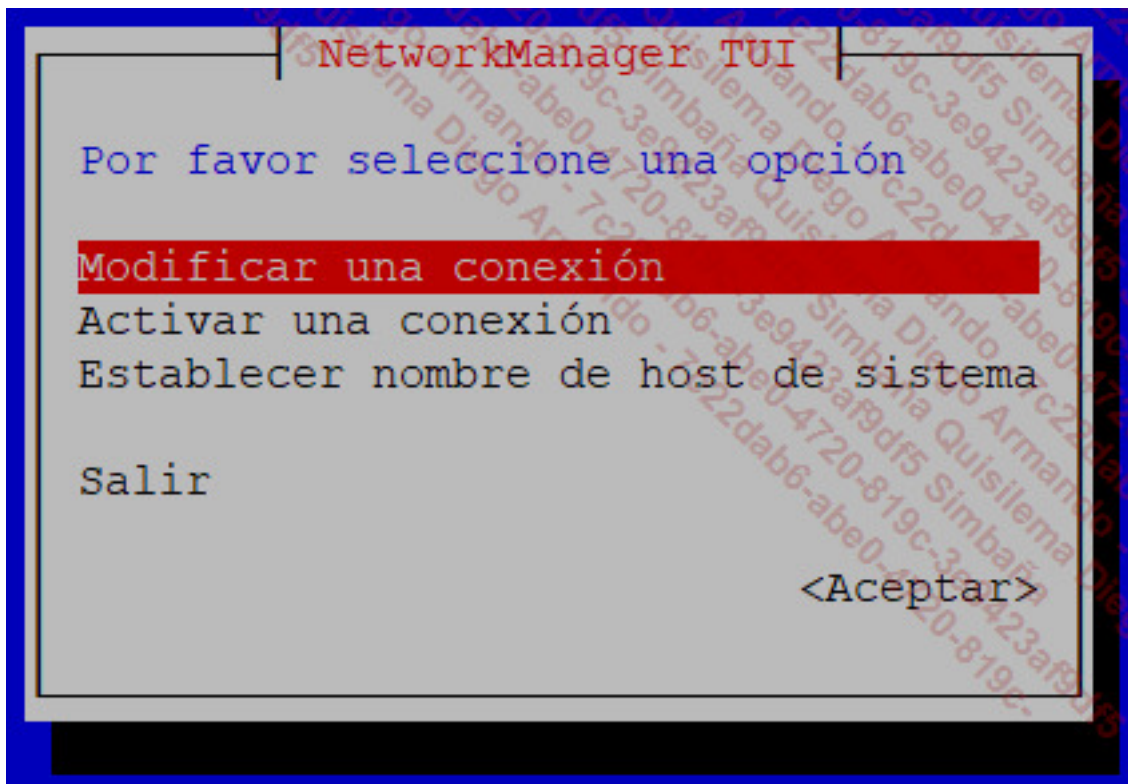
Un sistema Linux corriendo una distribución Debian 10 presenta una interfaz Wi-Fi, configurada a través de la interfaz gráfica de `NetworkManager` y otra interfaz Ethernet no configurada, conectada en una red sin servidor DHCP.

Después del inicio del sistema, se muestra con el comando `nmcli` la configuración de las tarjetas de interfaz tomadas en cuenta automáticamente por `NetworkManager`:

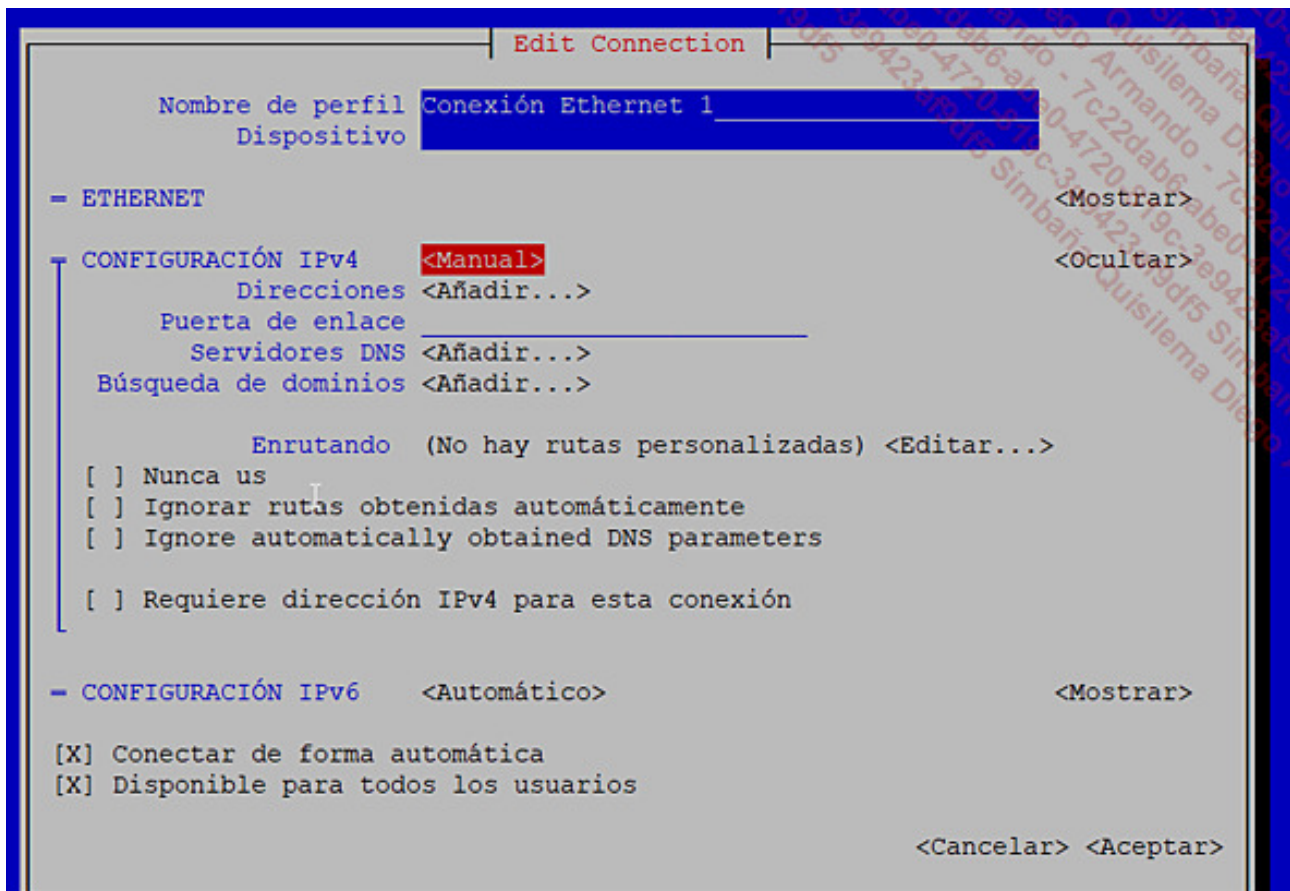
nmcli device

DEVICE	TYPE	STATE	CONNECTION
wlp3s0	wifi	conectado	wifiLAB
enp0s10	ethernet	desconectado	--
lo	loopback	sin gestión	--

Para gestionar la configuración de las tarjetas a través de `NetworkManager`, se puede usar el comando interactivo en modo semi gráfico `nmtui`:



Ahora podemos configurar en modo manual (no DHCP) la interfaz Ethernet:



2. Configuración del arranque de la red

La inicialización de la red durante el arranque del sistema operativo puede ser gestionada de manera diferente según la distribución utilizada, según las versiones e incluso según el propio criterio del administrador del sistema.

Sin embargo, podemos considerar dos maneras de configurar e inicializar la red, en función del tipo de distribución: Debian y derivadas o Red Hat y derivadas.

Para cada tipo, también habrá que distinguir entre el método «clásico» de arranque de la red usando scripts y el nuevo enfoque de `NetworkManager`.

El arranque de la red permite inicializar las diferentes tarjetas de interfaz de red, determinar la tabla de enrutamiento del núcleo y configurar la resolución de nombres (nombre de host local, métodos de resolución, servidores de nombres). Una vez que la inicialización haya sido efectuada, los servicios de redes de alto nivel podrán iniciarse.



La versión actual de la certificación LPIC-2 implica dominar los principios de configuración de los métodos clásicos usando `systemd` y tener en cuenta la hipotética acción de `NetworkManager`.

3. Configuración de redes de tipo Debian

Con respecto a las distribuciones Debian y derivadas (Ubuntu y otras), la configuración de las tarjetas de interfaces de red, durante el arranque del sistema, está definida esencialmente en el contenido de un solo archivo: `/etc/network/interfaces`, opcionalmente completado por la inclusión de otros archivos que se encuentran en el directorio `/etc/network/interfaces.d/`.

a. El archivo `/etc/network/interfaces`

Este archivo contiene los parámetros de las tarjetas de interfaz de red, incluyendo las que tendrán que inicializarse automáticamente durante el inicio de la red.



Para evitar conflictos, `NetworkManager` no gestiona ninguna tarjeta de interfaz de red que esté declarada en este archivo.

Los parámetros pueden ser aprendidos dinámicamente por DHCP cuando la interfaz de red se inicializa o estar fijados estáticamente por directivas dentro del archivo.

Los parámetros IPv6 pueden ser gestionados dinámicamente por los métodos propios a IPv6 (descubrimiento de la red, enlace local, etc.), por DHCP o estar fijados estáticamente en el archivo.

Una interfaz de red puede estar asociada a distintos conjuntos de parámetros, en el caso

en que deba gestionar distintas direcciones IP.

El archivo está constituido por conjuntos, llamados secciones (*stanzas*), cuyo formato general es el siguiente:

```
# Comentarios
NombreStanza NombreInterf [Params]
    [ OpciónId Params ]
...
    [ OpciónId Params ]
```

Donde:

NombreStanza : identificador del tipo de configuración para la interfaz (*iface*, *auto*, *allow-** ...).

NombreInterf : identificador de la tarjeta de interfaz de red.

OpciónId : identificador de la opción configurada (*address*, *netmask*, *gateway*, etc.).

Las principales secciones (*stanzas*) son:

<code>source /etc/network /interfaces.d/*</code>	Inclusión de los archivos del directorio especificado.
<code>auto NombreInterf</code>	Interfaz que deberá activarse durante el arranque del sistema.
<code>allow-hotplug NombreInterf</code>	Interfaz que deberá arrancarse en cuanto sea detectada.
<code>iface NombreInterf Params</code>	Definición de una instancia de la interfaz.

La sección *iface* permite definir los diferentes parámetros de una instancia de interfaz.

Los parámetros en la línea `iface` definen el protocolo (`inet` para IPv4 e `inet6` para IPv6) y el método de configuración (`dhcp` o `static`). Las líneas, opcionales, se encuentran justo después de la sección que define los parámetros de configuración (dirección estática, pasarela por defecto, etc.).



Si una interfaz utiliza IPv4 e IPv6, hay que declararla en las dos secciones, una con un parámetro `inet` y la otra con un parámetro `inet6`.

Ejemplos

Configuración estática de una interfaz de red en IPv4:

```
iface enp0s10 inet static
address 192.168.1.2/24
gateway 192.168.1.1
```

Configuración dinámica de una interfaz de red en IPv4:

```
iface enp0s10 inet dhcp
```

Los principales parámetros opcionales de una sección `iface` son:

<code>address</code> DirecciónCIDR	Dirección (preferentemente en formato CIDR), IPv4 o IPv6.
<code>netmask</code> Máscara	Máscara de subred (en vía de obsolescencia).
<code>gateway</code> Dirección	Pasarela por defecto.

Este archivo se usa para la configuración inicial de la red. En caso de modificación de su contenido, hay que reiniciar la red o forzar al servicio que vuelva a cargar la nueva configuración.

Ejemplo

Archivo de configuración de un sistema Debian 10, con una interfaz de red Ethernet `enp0s10` configurada para ser activada durante el arranque del sistema, con una dirección IPv4 estática. La interfaz Wi-Fi `wlp3s0` es gestionada por `NetworkManager`.

Archivo `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
source /etc/network/interfaces.d/*
# The loopback network interface
auto lo
iface lo inet loopback
# Interfaz Ethernet IPv4 statique
auto enp0s10
iface enp0s10 inet static
    address 10.1.0.2/16
    gateway 10.1.0.1
```

Después del arranque del sistema, mostramos el estado de las interfaces de red:

```
ip -br a
lo          UNKNOWN    127.0.0.1/8::1/128
enp0s10     UP          10.1.0.2/16 fe80::21b:24ff:fe6a:7814/64
wlp3s0      UP          192.168.0.11/24
2a01:e35:2439:1510:ee52:b40b:da60:3006/64 fe80::8373:d1d2:94e0:c8d8/64
```

La interfaz de red Ethernet, `enp0s10`, ha sido configurada correctamente con su dirección estática. La interfaz Wi-Fi, `wlp3s0`, ha sido activada y configurada por `NetworkManager`, como lo muestra el comando `nmcli`:

nmcli

wlp3s0: conectado to wifilab

"Broadcom and subsidiaries BCM4311"

wifi (wl), 00:1A:73:7E:14:7B, hw, mtu 1500

ip4 default, ip6 default

inet4 192.168.0.11/24

route4 0.0.0.0/0

route4 192.168.0.0/24

inet6 2a01:e35:2439:1510:ee52:b40b:da60:3006/64

inet6 fe80::8373:d1d2:94e0:c8d8/64

route6 fe80::/64

route6 2a01:e35:2439:1510::/64

route6::/0

route6 ff00::/8

enp0s10: sin gestión

"NVIDIA MCP67"

ethernet (forcedeth), 00:1B:24:6A:78:14, hw, mtu 1500

lo: sin gestión

"lo"

loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536

[...]

b. Interfaz con distintas direcciones IP de la misma versión

Para configurar una interfaz de red con diferentes direcciones IP de la misma versión, se pueden usar dos métodos.

- El primero consiste en definir distintas instancias de una tarjeta de interfaz de red, poniendo como sufijo su nombre con `:n`, donde `n` es el número de índice de la instancia. Cada instancia de una tarjeta tiene su propia sección con sus parámetros. Cada instancia será vista como una interfaz lógica distinta.
- El segundo consiste en declarar varias secciones para la misma tarjeta de interfaz de red, con los parámetros propios de cada dirección IP. La tarjeta de interfaz de red será vista como una sola instancia con distintas direcciones.

Ejemplos

Configuración estática multidirección IPv4, con distintas instancias de interfaz:

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.1/24
    gateway 192.168.1.254
```

```
auto eth0:0
allow-hotplug eth0:0
iface eth0:0 inet static
    address 192.168.2.1/24
    gateway 192.168.2.254
```

Configuración estática multidirección IPv4, con una sola instancia de interfaz:

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.1/24
    gateway 192.168.1.254
iface eth0 inet static
    address 192.168.2.1/24
    gateway 192.168.2.254
```

c. Arranque de la red usando el script networking

La inicialización de las tarjetas de red y el arranque de la red se pueden hacer combinando el script clásico, `networking`, y el script de arranque de `NetworkManager`, `network-manager`. Estos dos scripts pueden ser tomados en cuenta según el antiguo método `init System V`, o por `systemd`. Los dos se encuentran en el directorio de los scripts de gestión de servicios, `/etc/init.d`.

El script de arranque de `NetworkManager` no gestiona las interfaces ausentes del archivo de configuración `/etc/network/interfaces`. Se usa poco en los servidores, los cuales necesitan configuraciones estables para las diferentes interfaces de

red.

El script de arranque « clásico » de la red, `/etc/init.d/networking`, usa el archivo de configuración `/etc/network/interfaces`. Si este último se modifica después del arranque de la red, se puede recargar la configuración usando el comando siguiente:

```
systemctl restart networking
```

En las antiguas versiones sin `systemd`, se puede usar:

```
/etc/init.d/networking reload
```

o

```
service networking reload
```

Ejemplo

Modificamos la pasarela por defecto configurada en el archivo `/etc/network/interfaces`:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interfaz
auto lo
iface lo inet loopback
# Interfaz Ethernet IPv4 statique
auto enp0s10
iface enp0s10 inet static
    address 10.1.0.2/16
    gateway 10.1.0.3
```

Mostramos la configuración activa:

```
ip route show default
default via 10.1.0.1 dev enp0s10 onlink
default via 192.168.0.254 dev wlp3s0 proto dhcp metric 600
```

La modificación no ha sido tomada en cuenta.

Se reinicia la red:

```
systemctl restart networking
ip route show default
default via 10.1.0.3 dev enp0s10 onlink
default via 192.168.0.254 dev wlp3s0 proto dhcp metric 600
```

La modificación ha sido aplicada.

4. Configuración de red de tipo Red Hat

Para las distribuciones Red Hat y derivadas (CentOS, Fedora y otras), la configuración de las tarjetas de interfaz de red, durante el arranque del sistema, está definida ya sea a través de `NetworkManager`, o a través de archivos de nombre `ifcfg-NombreInterf`, uno para cada interfaz `NombreInterf` que se tendrá que inicializar, en el directorio `/etc/sysconfig/network-scripts`.

a. Los archivos `/etc/sysconfig/network-scripts/ifcfg*`

Estos archivos contienen los parámetros de las tarjetas de interfaz de red, incluyendo las que se inicializarán automáticamente durante al arranque de la red. Hay un archivo para cada tarjeta de interfaz de red.

Los parámetros pueden ser proporcionados dinámicamente por DHCP cuando la interfaz de red se inicializa o estar fijados por directivas dentro del archivo.

Los parámetros IPv6 pueden ser gestionados dinámicamente por los métodos propios a IPv6 (descubrimiento de la red, enlace local, etc.), por DHCP o estar fijados estáticamente en el archivo.

Si una interfaz de red tiene que estar asociada a distintos conjuntos de parámetros, en el caso de que tenga que gestionar distintas direcciones IP, habrá que definir tantos archivos como direcciones presente la interfaz, añadiendo un sufijo al nombre de la interfaz con su índice (:1, etc.).

El archivo está constituido por un conjunto de líneas, cada una define una variable de configuración. Estas variables serán utilizadas en los scripts de arranque o de paro de las interfaces (`ifup`, `ifdown` ...).

Ejemplos

Configuración estática de una interfaz de red en IPv4, archivo `ifcfg-enp1s0` :

```
DEVICE=enp1s0
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.1.0.3
PREFIX=24
GATEWAY=10.1.0.1
```

Configuración dinámica de una interfaz de red en IPv4, archivo `ifcfg-enp1s0` :

```
DEVICE=enp1s0
BOOTPROTO=dhcp
```

Las principales variables de configuración son:

<code>DEVICE=</code>	Nombre de la interfaz de red.
<code>BOOTPROTO=</code>	Protocolo de configuración (<code>dhcp</code> , <code>none</code>).
<code>ONBOOT=</code>	Activación en el arranque del sistema (<code>yes</code> , <code>no</code>).
<code>NETMASK=</code>	Máscara de subred en notación decimal punteada (obsoleto).
<code>PREFIX=</code>	Tamaño del identificador de red/subred en bits (reemplaza a la anterior).
<code>IPADDR=</code>	Dirección IPv4 en notación decimal punteada.
<code>GATEWAY=</code>	Pasarela por defecto en notación decimal punteada.
<code>IPV6INIT=</code>	Configuración IPv6 (<code>yes</code> , <code>no</code>).
<code>IPV6ADDR=</code>	Dirección IPv6 en notación CIDR.
<code>NM_CONTROLLED=</code>	Interfaz gestionada por <code>NetworkManager</code> (<code>yes</code> , <code>no</code>).

Este archivo se usa para la configuración inicial de la red. En caso de modificación de su contenido, hay que reiniciar la red o forzar que el servicio cargue la nueva configuración.



Si el script `/etc/init.d/network` existe (CentOS 7, por defecto) y se quiere evitar que NetworkManager gestione una interfaz, hay que posicionar la variable `NM_CONTROLLED` a `no` en el archivo de configuración de esta interfaz. Sin embargo, si el script de arranque del sistema no existe (CentOS 8, por defecto), no hay que posicionar esta variable a `no`, si no la tarjeta no será activada en el arranque de la red.

Ejemplo

Archivo de configuración de un sistema CentOS 7 con una sola interfaz de red, de tipo Ethernet, `enp0s3`, configurada para ser activada durante el arranque del sistema, con una dirección IPv4 estática.

Archivo `/etc/sysconfig/network-scripts/ifcfg-enp0s3`:

```
NM_CONTROLLED="no"
TIPO="Ethernet"
BOOTPROTO="none"
NAME="enp0s3"
DEVICE="enp0s3"
ONBOOT="yes"
IPADDR=10.1.0.3
PREFIX=24
GATEWAY=10.1.0.1
```

Después de arrancar el sistema, mostramos las interfaces de red:

```
ip -br a
lo          UNKNOWN    127.0.0.1/8::1/128
enp0s3      UP          10.1.0.3/24
2a01:e35:2439:1510:a00:27ff:fe15:f6f9/64 fe80::a00:27ff:fe15:f6f9/64
```

La interfaz de red Ethernet, `enp0s3`, ha sido configurada correctamente con su dirección

IPv4 estática.

La pasarela por defecto también se ha configurado:

```
ip route show default
default via 10.1.0.1 dev enp0s3 proto static metric 100
```

NetworkManager no gestiona la interfaz, como lo muestra el comando `nmcli`:

```
nmcli
enp0s3: sin gestión
"Intel 82540EM"
ethernet (e1000), 08:00:27:15:F6:F9, hw, mtu 1500
lo: sin gestión
"lo"
loopback (unknown), 00:00:00:00:00:00, sw, mtu 65536
```

b. Interfaz con distintas direcciones IP de la misma versión

Para configurar una interfaz de red con distintas direcciones IP de la misma versión, hay que declarar distintas instancias de direcciones, la palabra clave del parámetro contiene como sufijo el número de índice de la dirección de red.

Ejemplo

Configuración estática multidirección IPv4, archivo `ifcfg-enp1s0`:

```
DEVICE=enp1s0
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.1.0.3
PREFIX=24
GATEWAY=10.1.0.1
IPADDR1=10.2.0.3
PREFIX1=24
GATEWAY1=10.2.0.254
```


c. Arranque de la red

La inicialización de las tarjetas de interfaz de red y el arranque de la red se pueden hacer combinando un script clásico, `network`, y el servicio `NetworkManager`, gestionado directamente por `systemd`.

El script `network` se puede gestionar según el antiguo método `init System V`, o con `systemd`. Se encuentra en el directorio de los scripts de gestión de los servicios, `/etc/init.d`.

Si el script `network` no existe en el directorio `/etc/init.d` y el sistema usa `systemd` para iniciar los servicios, este último lanzará directamente `NetworkManager`, lo que activará todas las interfaces de red que presenten un archivo de configuración en `/etc/sysconfig/network-scripts`, con la variable `ONBOOT` con el valor `yes`, excepto aquellas que presenten explícitamente una variable `NM_CONTROLLED` con el valor `no`. También activará las interfaces no definidas en este directorio, usando para ello sus propios archivos de configuración.



La version 8 de las distribuciones de tipo Red Hat ya no presenta el script por defecto `network`.

Ejemplo

En un sistema CentOS 8, añadimos una dirección IPv4 en el archivo de configuración de la interfaz de red `enp38s0`, `/etc/sysconfig/network-scripts/ifcfg-enp38s0`:

```
NAME=enp38s0
DEVICE=enp38s0
TYPE=Ethernet
ONBOOT=yes
BOOTPROTO=dhcp
# Dirección estática IPv4:
```

IPADDR1=10.1.0.1
PREFIX1=24

Mostramos el estado IPv4 de las tarjetas de de interfaz de red:

```
ip -br -4 a
lo          UNKNOWN    127.0.0.1/8
enp38s0     UP          192.168.0.3/24
```

La modificación no ha sido tomada en cuenta, la tarjeta solamente tiene una dirección IPv4, distribuida por DHCP.

Después de reiniciar el sistema:

```
ip -4 -br a
lo          UNKNOWN    127.0.0.1/8
enp38s0     UP          10.1.0.1/24 192.168.0.3/24
wlo1        UP          192.168.0.5/24
```

La modificación ha sido aplicada.

5. Control del enrutamiento

Si la comunicación dentro de la red/subred funciona correctamente pero el sistema no consigue comunicar fuera de su red/subred, es posible que se trate de un problema de enrutamiento. Diferentes comandos permiten obtener datos sobre la configuración del enrutamiento y el estado de los diferentes routers.

a. Los comandos `ip route` y `route`

Estos comandos, que ya han sido estudiados anteriormente para la configuración de una pasarela por defecto, permiten mostrar la tabla de enrutamiento gestionada por el núcleo. Esto permite comprobar los elementos de enrutamiento conocidos por el núcleo, para

acceder a las diferentes redes/subredes, así como las rutas por defecto de las diferentes interfaces.

Ejemplo

Vista de la tabla de enrutamiento del núcleo (`-n` para mostrar las direcciones en lugar de los nombres) de un sistema que se encuentra en la red `192.168.0.0/24`:

```
route -n
Kernel IP routing table
Destination Gateway Genmask Flags Metric Ref Use Iface
0.0.0.0 192.168.0.254 0.0.0.0 UG 0 0 0 enp0s3
169.254.0.0 0.0.0.0 255.255.0.0 U 1002 0 0 enp0s3
192.168.0.0 0.0.0.0 255.255.255.0 U 0 0 0 enp0s3
```

La pasarela por defecto (destination `0.0.0.0`) es `192.168.0.254`.

Vista de la tabla de enrutamiento del núcleo (`-r` para mostrar las direcciones en lugar de los nombres) de otros sistema que también se encuentra en la red `192.168.0.0/24`, que no consigue comunicar con el exterior de la red:

```
ip route
default via 192.168.0.14 dev enp0s3
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.6
```

La pasarela por defecto es incorrecta, `192.168.0.14`. Podemos modificarla dinámicamente, luego habrá que corregir el archivo de configuración de la interfaz:

```
ip route change default via 192.168.0.254
ip r show
default via 192.168.0.254 dev enp0s3
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.6
```

b. traceroute

Este comando ofrece datos sobre las rutas recorridas para llegar hasta un host remoto. Puede permitir identificar un problema en un router remoto (rendimiento, bloqueo...).



Este comando debe utilizarse con moderación, porque solicita los routers intermediarios que no tienen por qué estar administrados dentro de la organización donde se encuentra el sistema local. Por otro lado, el comando usa el protocolo ICMP, y algunos routers pueden estar configurados para no contestar a esos mensajes, lo que puede hacer que el diagnóstico no sea correcto.

Sintaxis

`tracert [-Opciones] Dest`

Parámetros principales

<code>Dest</code>	Dirección o nombre de host destino.
<code>-n</code>	Sin resolución de nombres.
<code>-4 6</code>	Versión de IP.
<code>-i Interf</code>	Interfaz.
<code>-m max_salto</code>	Número máximo de routers que se podrán atravesar (por defecto 30).

Descripción

Sin ninguna opción, el comando intenta contactar con el host especificado interrogando

sucesivamente a cada router intermediario. Muestra las estadísticas de cada segmento que atraviesa hasta el destino o hasta el último router accesible.

Ejemplo

Vista de la información sobre las diferentes redes y routers hasta llegar al servidor HTTP de Google:

```
tracert www.google.com
tracert to www.google.com (172.217.19.228), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.254) 0.654 ms 0.605 ms 0.572 ms
 2 cha92-2_migr-82-67-145-254.fbx.proxad.net (82.67.145.254) 6.533 ms 6.544 ms 6.948 ms
 3 78.255.141.126 (78.255.141.126) 1.841 ms 2.011 ms 2.126 ms
 4 pes75-ncs540-1-be3.intf.nro.proxad.net (78.254.253.42) 1.832 ms 1.826 ms 1.820 ms
 5 p13-9k-3-be2002.intf.nro.proxad.net (78.254.242.54) 1.859 ms 1.842 ms 1.911 ms
 6 194.149.165.209 (194.149.165.209) 2.342 ms 1.419 ms *
 7 194.149.166.62 (194.149.166.62) 1.444 ms 1.382 ms 1.557 ms
 8 72.14.221.62 (72.14.221.62) 1.335 ms 1.509 ms 1.510 ms
 9 108.170.245.1 (108.170.245.1) 2.637 ms 2.628 ms 108.170.244.193 (108.170.244.193) 1.746 ms
10 216.239.59.211 (216.239.59.211) 1.751 ms 1.746 ms 1.252 ms
11 par21s11-in-f4.1e100.net (172.217.19.228) 1.218 ms 1.210 ms 1.373 ms
```

6. Configuración de la resolución de nombres de host

Cuando la capa de red IP ya ha sido inicializada para las distintas interfaces, las aplicaciones pueden usar las direcciones IP para comunicarse entre ellas. Sin embargo, usan generalmente nombres de host para identificar el sistema con el que desean hacer un intercambio de mensajes. Esto conlleva una fase suplementaria, anterior al intercambio de datos, que consiste en asociar una dirección IP a un nombre de host. Esta fase se llama resolución de nombres y puede utilizar distintos métodos, según la configuración del sistema.

a. Configuración de un nombre de host

El comando `hostname Nombre` permite asignar dinámicamente el nombre de host

Nombre al sistema local. Esta modificación es tomada en cuenta inmediatamente pero no se queda guardada, se perderá cuando el sistema se pare. Sin argumento, el comando muestra el nombre de red del sistema.



No se aconseja modificar dinámicamente el nombre de host de un sistema con servicios de red activos, esta acción puede perturbar el funcionamiento de estos.

Ejemplo

Vista del nombre de host:

```
hostname
srvpba
```

Para usar el nombre de host durante el arranque del sistema, hay que declararlo en un archivo de configuración. El procedimiento es diferente según la distribución y la versión.

- ▾ Distribución de tipo Debian o Red Hat (versión 7 y superiores): el nombre de host está declarado en el archivo `/etc/hostname`.
- ▾ Distribución de tipo Red Hat antes de la versión 7: el nombre de host está declarado en la variable `HOSTNAME`, ella misma se ha declarado en el archivo `/etc/sysconfig/network`.

b. Configuración de la resolución de nombres

Existen numerosos métodos de resolución de nombres de host. Estos métodos están configurados para ser utilizados por una biblioteca de resolución de nombres (resolver), implementando funciones de búsqueda del nombre de host o de dirección, dicha biblioteca es utilizada por la mayoría de las aplicaciones de red.

Los métodos más habituales son:

- ✓ Búsqueda estática en el archivo `/etc/hosts`:

Es el método más antiguo. Cada sistema define en ese archivo de texto la relación entre la dirección IP y el nombre de host de los sistemas. Este método está muy limitado, y solamente se utiliza para declarar el nombre o los nombres de host de la máquina local.

El archivo `/etc/hosts` es un archivo de texto, cada línea define una dirección IP asociada a un nombre de host y a veces también a alias de ese mismo nombre, con la sintaxis:

```
DirecciónIP NombreHost [Alias1 ... AliasN]
```

Si la máquina forma parte de un dominio DNS, el segundo campo debe ser su nombre DNS completo (FQDN, *Fully Qualified Domain Name*).

- ✓ Servidores DNS:

Es el método más versátil y el más utilizado. Cuando una aplicación necesita resolver un nombre de host o una dirección IP, la biblioteca de resolución de nombres interroga a uno o distintos servidores DNS.

- ✓ Servidores de nombre no DNS:

Un sistema puede también usar servidores de nombres que no sean de DNS, por ejemplo un servidor de anuario LDAP un servidor de configuración centralizada NIS (*Network Information Service*).

Estos métodos pueden ser combinados para ser usados sucesivamente hasta satisfacer la resolución.

Los métodos que se utilizarán se encuentran configurados en el archivo `/etc/nsswitch.conf`.

En ese archivo de texto, la línea `hosts` define los métodos que se utilizarán, y el orden en el que se hará, por ejemplo:

```
hosts: files dns
```

La biblioteca de resolución de nombres buscará primero en el archivo local `/etc/hosts` (método `files`). Si no encuentra la resolución en ese archivo, utilizará un servidor DNS.

c. Configuración de un cliente DNS

El archivo de configuración de un cliente DNS es `/etc/resolv.conf`. Se trata de un archivo de texto donde se define el dominio DNS de la máquina local (directiva `domain`) y los servidores DNS que se utilizarán para la resolución de nombres (directiva `nameserver`), siguiendo este orden de interrogación.



Si la configuración DNS es suministrada por un servidor DHCP, el cliente DHCP actualizará el contenido de ese archivo.

Sintaxis

```
domain      DominioDNS
nameserver  Dirección1
[nameserver Dirección2]
```



También se pueden declarar en ese archivo los sufijos de búsqueda de los nombres para los nombres simples, con la palabra clave `search` seguida de una lista de prefijos DNS.

Ejemplo

Configuración de la resolución de nombres.

Comprobamos la configuración de los métodos de resolución de nombres en el archivo `/etc/nsswitch.conf`:

```
grep hosts /etc/nsswitch.conf
# hosts: files dns
```



```
# hosts: files dns # from user file
hosts:    files dns myhostname
```

El sistema usa primero su archivo `/etc/hosts`, y después DNS.

Configuración del cliente DNS:

```
cat /etc/resolv.conf
nameserver 192.168.0.254
```

El sistema interroga a un solo servidor de nombres.

Comprobamos el funcionamiento con el comando `host`:

```
host www.google.com
www.google.com has address 172.217.19.228
www.google.com has IPv6 address 2a00:1450:4007:80c::2004
```

La resolución de nombres DNS funciona.

Modificamos el archivo de configuración `/etc/resolv.conf`:

```
vi /etc/resolv.conf
nameserver 192.168.0.250
```

Comprobamos el funcionamiento:

```
host www.google.com
;; connection timed out; no servers could be reached
```

El servidor de DNS no responde.

7. Restricción de acceso a los servicios de red

El control de acceso a los servicios de red se puede hacer en distintos niveles. En la mayoría de las distribuciones hay módulos de firewall implementados, administrados la mayoría de las veces por el núcleo, para filtrar los mensajes entrantes y salientes según distintos criterios (dirección origen o destino, número de puerto de destino, etc.).

Los módulos de autenticación y de conexión PAM (*Pluggable Authentication Method*) realizan también un control, aplicación por aplicación, con respecto a las solicitudes de conexión, en función de distintos criterios (nombre de usuario, máquina origen, etc.).

Algunos servicios de red se basan en una capa de software intermediaria entre los sockets y el servidor de red destinatario, esta capa es proporcionada por una biblioteca llamada `tcp_wrappers`. Recibe las solicitudes de conexión a los puertos TCP o UDP y aplica reglas para determinar si debe transmitir o no la solicitud de conexión al servidor destinatario.



Incluso aunque este método de control de redes tienda a hacerse obsoleto y ya casi nunca se utiliza, hay que conocerlo para diagnosticar posibles problemas de conexión a servidores y también para la certificación LPIC-2.

a. Configuración de TCP Wrappers

La biblioteca `tcp_wrappers` permite controlar los accesos a diferentes servicios de red, según las direcciones o los nombres de host de los clientes. Para ello, se tiene que administrar una lista para cada servicio con los clientes autorizados y/o una lista para los clientes no autorizados:

- ~ `/etc/hosts.allow` define los clientes autorizados.
- ~ `/etc/hosts.deny` define los clientes no autorizados.

Distintos softwares utilizan esta biblioteca, y en particular el servicio `tcpd`. El rol de este

servicio es comprobar que el cliente está autorizado a comunicarse con el servicio de red solicitado. Aplica las reglas de autorización o de bloqueo configuradas. Si el cliente está autorizado, activa el servidor correspondiente, en caso contrario deniega la solicitud.

La sintaxis de una línea de configuración es la siguiente:

`ListaServicios: ListaClientes`

Donde:

`ListaServicios` Uno o distintos nombres de servicios de red.

`ListaClientes` Uno o distintos conjuntos de direcciones o de nombres de host.

Para definir la lista de los clientes, se pueden utilizar palabras clave predefinidas (`ALL`, `NONE`, `LOCAL` ...) y los caracteres comodín (`*`, `?`). Cada conjunto está separado por un espacio o por una coma.

Para determinar si un cliente está autorizado o no a acceder a un servicio, se aplicará la regla siguiente:

Si el archivo `/etc/hosts.allow` existe
 Si el cliente está autorizado por el servicio, solicitud aceptada.
 Si `no`, solicitud denegada.
 Si `no`
 Si el archivo `/etc/hosts.deny` existe
 Si el cliente es bloqueado por el servicio, solicitud denegada.
 Si `no`, solicitud aceptada.
 Si `no`, solicitud aceptada.



El control de acceso de la biblioteca `tcp_wrappers` puede ser utilizado, directamente o a través de `tcpd`, por los superservidores como `inetd` o el más reciente `xinetd`, para controlar el acceso a los servicios de redes administrados por ellos mismos.

Ejemplo

Archivo `/etc/hosts.allow` :

```
# Todas las direcciones que comienzan por 10.1 o 10.2
# están autorizadas a acceder al servidor ftp local
ftpd: 10.1. , 10.2.
# Solamente la dirección de red 10.1.0.1 puede conectarse al servidor sshd local:
sshd: 10.1.0.1
# La dirección de red de loopback interna puede acceder a todos los servidores
locales
ALL: 127.0.0.1
```

8. Archivos de registro

Incluso aunque los servicios de red tienen generalmente sus propios archivos de registro de eventos (*logs*), también se pueden encontrar registros relativos a las actividades de red en el archivo del registro general del sistema, así como en el gestionado por `systemd`.

a. Archivos de registro del sistema

Los archivos del registro del sistema están gestionados por un servicio de registro del sistema, el viejo `syslogd` o el más reciente `rsyslogd`. Su archivo de configuración determina en qué archivos guardará los registros, para qué tipo de evento y en qué formato.

En las versiones recientes de las distribuciones de tipo Debian y Red Hat, el servicio utilizado es `rsyslogd`. Su archivo de configuración es `/etc/rsyslog.conf`.

En este archivo encontramos la ubicación del archivo del registro general del sistema. Por defecto, se trata del archivo `/var/log/messages`, como se puede ver a continuación:

Extracto del archivo `/etc/rsyslog.conf`:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none /var/log/messages
```

Las distribuciones de tipo Debian usan también un archivo de registro específico para cada servicio, por defecto `/var/log/daemon.log`:

Extracto del archivo `/etc/rsyslog.conf` de una distribución Debian:

```
daemon.*    -/var/log/daemon.log
```

Estos archivos se archivan regularmente en su directorio, se les pone un sufijo con la fecha de archivo.

En caso de fallo de la red, pueden aportar elementos de información importantes para el diagnóstico del problema.

Ejemplo

```
grep -i error /var/log/messages
Apr  9 09:12:51 beta iscsiadm[6655]: iscsiadm: initiator reported error
(8 - connection timed out)
Apr  9 09:12:51 beta iscsiadm[6655]: iscsiadm: initiator reported error
(8 - connection timed out)
Apr  9 09:12:57 beta iscsiadm[6655]: iscsiadm: initiator reported error
(8 - connection timed out)
```

En este sistema, hay un problema de conexión de un periférico iSCSI.

b. El registro de systemd

`systemd` implementa su propio mecanismo de registro de eventos en su archivo de registro. Este archivo no es un archivo de texto, tiene formato binario y no se puede leer directamente. Hay que usar el comando `journalctl` para poder consultarlo.

Sintaxis

```
journalctl [ -Opciones ] [ Criterios ]
```

Parámetros principales

<code>-Opciones</code>	Opciones.
<code>Criterios</code>	Criterios de selección de eventos.

Descripción

Sin ninguna opción ni criterio, el comando muestra todos los eventos guardados en el registro, con el formato por defecto, y marcando un espacio entre las páginas. Las opciones permiten especificar los campos que se mostrarán, el orden de la vista, etc. Los criterios de forma CAMPO=VALOR, permiten seleccionar los eventos según diferentes criterios combinables.

Ejemplo

Vista de los eventos registrados por `systemd` para el arranque de la red por el script `network`, en una distribución CentOS 7:

```
journalctl _SYSTEMD_UNIT=network.service
--Logs begin at vie. 2020-04-10 12:22:17 CEST, end
at vie. 2020-04-10 16:01:01 CEST. --
abril 10 15:02:32 centos7-1 network[1957]: Activación de la interfaz loopback: [ OK ]
abril 10 15:02:32 centos7-1 network[1957]: Activación de la interfaz enp0s3:
abril 10 15:02:34 centos7-1 dhclient[2103]: DHCPDISCOVER on enp0s3 to 255.255.255.255
```

```
port 67 interval 5 (xid=0xbebab44)
abril 10 15:02:34 centos7-1 dhclient[2103]: DHCPREQUEST on enp0s3 to 255.255.255.255
port 67 (xid=0xbebab44)
abril 10 15:02:34 centos7-1 dhclient[2103]: DHCPOFFER from 192.168.0.254
abril 10 15:02:36 centos7-1 dhclient[2103]: DHCPACK from 192.168.0.254 (xid=0xbebab44)
abril 10 15:02:38 centos7-1 network[1957]: Definición de la información IP para enp0s3.
hecho.
abril 10 15:02:39 centos7-1 network[1957]: [ OK ]
```

Podemos ver cómo el cliente DHCP solicita y obtiene una dirección de un servidor DHCP.