

# ADMINISTRACIÓN DE LA RED DE RED HAT ENTERPRISE LINUX

RH124

Capítulo 11

The background is a solid red color. On the right side, there is an abstract geometric design consisting of several thin white lines that intersect at small white dots. One dot is located near the top right, another near the bottom right, and a third near the bottom center. These lines and dots form a network-like structure. At the bottom of the image, there is a horizontal bar that is divided into two sections: a light gray section on the left and a white section on the right.

Descripción general	
<b>Meta</b>	Configurar la red IPv4 básica en los sistemas Red Hat Enterprise Linux.
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Explicar los conceptos fundamentales de la red de computadora.</li> <li>• Realizar una prueba y revisar la configuración de red actual con las utilidades básicas.</li> <li>• Administrar la configuración de la red y los dispositivos con <code>nmcli</code> y NetworkManager.</li> <li>• Modificar la configuración de la red mediante la edición de los archivos de configuración.</li> <li>• Configurar y probar el nombre del host del sistema y la resolución de nombre.</li> </ul>
<b>Secciones</b>	<ul style="list-style-type: none"> <li>• Conceptos de red (y práctica)</li> <li>• Validación de la configuración de red (y práctica)</li> <li>• Configuración de red con <code>nmcli</code> (y práctica)</li> <li>• Edición de archivos de configuración de red (y práctica)</li> <li>• Configuración de nombres de host y resolución de nombre (y práctica)</li> </ul>
<b>Trabajo de laboratorio</b>	<ul style="list-style-type: none"> <li>• Administración de la red de Red Hat Enterprise Linux</li> </ul>

# Conceptos de red

RH124



# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder explicar conceptos fundamentales de redes de computadoras.

# Redes IPv4

Los estándares de TCP/IP siguen un modelo de red de cuatro capas que se especifica en RFC1122.

- **Aplicación**

Cada aplicación tiene especificaciones para que los clientes y los servidores puedan comunicarse en las plataformas. Entre los protocolos comunes se incluyen SSH (inicio de sesión remoto), HTTPS (web segura), NFS o CIFS (uso compartido de archivo) y SMTP (envío de correo electrónico).

- **Transporte**

Los protocolos de transporte son TCP y UDP. TCP es una comunicación confiable orientada a la conexión, mientras que UDP es un protocolo *de datagramas* sin conexión. Los protocolos de aplicaciones utilizan puertos TCP o UDP. En el archivo `/etc/services`, encontrará una lista de puertos conocidos y registrados.

Cuando un paquete se envía por la red, la combinación del puerto de servicio y la dirección IP forma un socket. Cada paquete tiene un socket de origen y un socket de destino. Esta información puede utilizarse al realizar tareas de monitoreo y filtrado.

- **Internet**

Internet, o capa de red, transporta datos desde el host de origen hasta el host de destino. Cada host tiene una dirección IP y un prefijo que se utiliza para determinar direcciones de red. Los enrutadores se utilizan para conectar redes.

ICMP es un protocolo de control en esta capa. En lugar de puertos, tiene tipos. La utilidad **ping** es un ejemplo de paquetes ICMP para probar la conectividad. **ping** envía un paquete ICMP ECHO\_REQUEST. Un **ping** exitoso recibe una confirmación ICMP ECHO\_REPLY. Un **ping** no exitoso puede recibir mensajes de error ICMP, como "no se puede alcanzar el destino", o directamente puede no recibir ninguna respuesta.

- **Enlace**

La capa de enlace, o acceso a medios, proporciona la conexión a medios físicos. Los tipos de redes más comunes son Ethernet (802.3) cableada y WLAN (802.11) inalámbrica. Cada dispositivo físico tiene una dirección de hardware (MAC) que se utiliza para identificar el destino de paquetes en el segmento de red local.

**IP Address:**

172.17.5.3 = 10101100.00010001.00000101.00000011

Prefix: /16

**Netmask:**

255.255.0.0 = 11111111.11111111.00000000.00000000

10101100.00010001.00000101.00000011

Network

Host

**IP Address:**

192.168.5.3 = 11000000.10101000.00000101.00000011

Prefix: /24

**Netmask:**

255.255.255.0 = 11111111.11111111.11111111.00000000

11000000.10101000.00000101.00000011

Network

Host

*Figura 11.1: Máscaras de red y direcciones IPv4*

## Direcciones IPv4

Una dirección IPv4 es un número de 32 bits, el cual generalmente se expresa en decimales en cuatro *octetos*, cuyo valor oscila entre 0 y 255, separados por puntos. La dirección se divide en dos partes: la *parte de la red* y la *parte del host*. Todos los hosts en la misma subred, que pueden comunicarse entre sí directamente sin un enrutador, cuentan con la misma parte de red; la parte de red identifica la subred. Dos hosts en la misma subred no pueden tener la misma parte de host; la parte del host identifica un host en particular en una subred.

En la Internet moderna, el tamaño de una subred IPv4 es variable. Para saber qué parte de una dirección IPv4 es la parte de red y cuál es la parte de host, un administrador debe conocer la *máscara de red* que se asignó a la subred. La máscara de red indica cuántos bits de la dirección IPv4 pertenecen a la subred. Cuantos más bits haya disponibles para la parte del host, más hosts habrá en la subred.

La dirección más baja posible en una subred (la parte de host son todos ceros en binario), algunas veces, se denomina *dirección de red*. La dirección más alta posible en una subred (la parte de host son todos los unos en binario) se utiliza para los mensajes de broadcast en IPv4 y se denomina *dirección de broadcast*.

Las máscaras de red se expresan de dos formas. La sintaxis más antigua de la máscara de red que utiliza 24 bits para la parte de red indicaría 255.255.255.0. Una sintaxis más nueva denominada notación CIDR, especificaría un *prefijo de red* de /24. Ambas formas transmiten la misma información; a saber, cuántos bits principales en la dirección IP contribuyen a la dirección de red.



**Cálculo de la dirección de red para 192.168.1.107/24**

Dirección de host	192.168.1.107	11000000.10101000.00000001.01101011
Prefijo de red	/24 (255.255.255.0)	11111111.11111111.11111111.00000000
Dirección de red	192.168.1.0	11000000.10101000.00000001.00000000
Dirección de broadcast	192.168.1.255	11000000.10101000.00000001.11111111

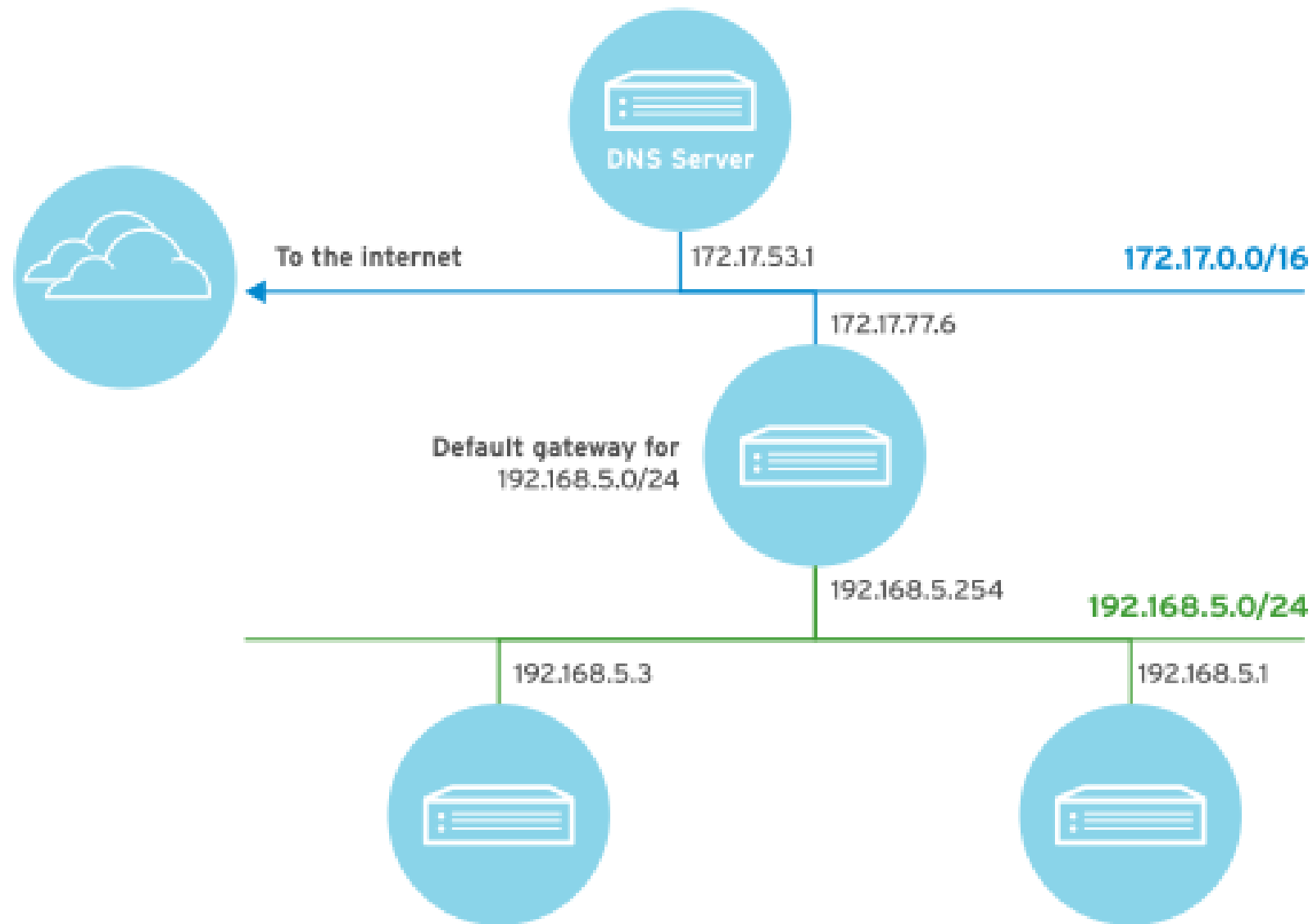
**Cálculo de la dirección de red para 10.1.1.18/8**

Dirección de host	10.1.1.18	00001010.00000001.00000001.00010010
Prefijo de red	/8 (255.0.0.0)	11111111.00000000.00000000.00000000
Dirección de red	10.0.0.0	00001010.00000000.00000000.00000000
Dirección de broadcast	10.255.255.255	00001010.11111111.11111111.11111111

**Cálculo de la dirección de red para 172.16.181.23/19**

Dirección de host	172.168.181.23	10101100.10101000.10110101.00010111
Prefijo de red	/19 (255.255.224.0)	11111111.11111111.11100000.00000000
Dirección de red	172.168.160.0	10101100.10101000.10100000.00000000
Dirección de broadcast	172.168.191.255	10101100.10101000.10111111.11111111

La dirección especial 127.0.0.1 siempre apunta al sistema local ("localhost") y la red 127.0.0.0/8 pertenece al sistema local, para que pueda comunicarse con ella misma usando protocolos de red.



*Figura 11.2: Conceptos de DNS y enrutamiento de red*

## Enrutamiento IPv4

Ya sea que se utilice IPv4 o IPv6, el tráfico de red debe desplazarse desde un host a otro y de una red a otra. Cada host tiene una *tabla de enrutamiento*, que le indica cómo enrutar el tráfico para redes particulares. Las entradas de la tabla de enrutamiento incluirán una red de destino, que se conecta para dirigir el tráfico, y la dirección IP de cualquier enrutador intermedio necesaria para transmitir el mensaje a su destino final. La entrada de la tabla de enrutamiento que coincide con el destino del tráfico de red se utiliza para enrutarla. Si dos entradas coinciden, se utiliza la que tiene el prefijo más extenso.

Si el tráfico de red no coincide con una ruta más específica, la tabla de enrutamiento normalmente tiene una entrada para una *ruta predeterminada* a toda la Internet IPv4, 0.0.0.0/0. Esta ruta predeterminada apunta a un *enrutador* en una subred a la que se puede acceder, es decir, en una subred que tiene una ruta más específica en la tabla de enrutamiento del host.

Si un enrutador recibe tráfico que no está dirigido a este, en lugar de ignorarlo como un host normal, *reenvía* el tráfico sobre la base de su propia tabla de enrutamiento. Esto puede enviar el tráfico directamente al host de destino (si el enrutador se encuentra en la subred de destino) o se lo puede reenviar a otro enrutador. Este proceso de reenvío continúa hasta que el tráfico alcanza su destino final.

### Ejemplo de tabla de enrutamiento

Destino	Interfaz	Enrutador (si es necesario)
192.0.2.0/24	wlo1	
192.168.5.0/24	enp3s0	

Destino	Interfaz	Enrutador (si es necesario)
0.0.0.0/0 (predeterminado)	enp3s0	192.168.5.254

En este ejemplo, el tráfico que se mueve para la dirección IP 192.0.2.102 desde este host se transmitirá directamente a ese destino a través de la interfaz inalámbrica **wlo1** porque su coincidencia más cercana es con la ruta 192.0.2.0/24. El tráfico de la dirección IP 192.168.5.3 se transmitirá directamente a ese destino a través de la interfaz Ethernet **enp3s0** porque su coincidencia más cercana es con la ruta 192.168.5.0/24.

El tráfico de la dirección IP 10.2.24.1 se transmitirá de una interfaz Ethernet **enp3s0** a un enrutador a través de 192.168.5.254, que reenviará el tráfico al destino final. La coincidencia más cercana de ese tráfico es con la ruta 0.0.0.0/0, ya que no hay una ruta más específica en la tabla de enrutamiento de este host. El enrutador usará su propia tabla de enrutamiento para determinar adónde reenviar el tráfico.

## Nombres y direcciones IP

El protocolo IP utiliza direcciones para comunicarse, pero los seres humanos preferirían trabajar con nombres en lugar de cadenas de números largos y difíciles de recordar. *DNS*, el sistema de nombres de dominio, es una red distribuida de servidores que asignan nombres de host a direcciones IP. Para que el servicio de nombres funcione, el host debe estar apuntado a un *servidor de nombres*. Este servidor de nombres no debe estar en la misma subred, simplemente necesita que el host tenga acceso a ella.

## Configuración de red estática o DHCP

Muchos sistemas están configurados para obtener valores de red automáticamente durante el proceso de arranque. De acuerdo con los archivos de configuración local, debe usarse DHCP y un servicio de cliente aparte consulta a la red por un servidor y obtiene un alquiler de valores de red.

Si no hay un servidor DHCP disponible, el sistema debe usar una configuración *estática* en la que los valores de red se lean desde un archivo de configuración local. El administrador de red o el equipo de arquitectura son los encargados de obtener los valores de red correctos para asegurarse de que no haya ningún conflicto con otros sistemas.

Como DHCP usa la dirección de hardware para hacer un seguimiento de las asignaciones, solo una dirección puede asignarse por interfaz con DHCP. Múltiples direcciones estáticas pueden asignarse a una sola interfaz. Esta práctica es común en sistemas que alojan servicios para varios clientes, como el alojamiento basado en la IP HTTP. Las interfaces Red Hat Enterprise Linux generalmente tienen una dirección IPv4 y una dirección de enlace local IPv6, aunque pueden tener asignadas más direcciones.

# Nombres de interfaces de red

Tradicionalmente, las interfaces de red en Linux se enumeran de la siguiente manera: **eth0**, **eth1**, **eth2**, etc. Sin embargo, el mecanismo que define estos nombres puede efectuar cambios en cuanto a qué interfaz recibe un determinado nombre a medida que se añaden o eliminan dispositivos. El comportamiento de asignación de nombre predeterminado en Red Hat Enterprise Linux 7 consiste en asignar nombres fijos según firmware, topología de dispositivo y tipo de dispositivo. Los nombres de interfaces tienen los siguientes caracteres:

- Las interfaces Ethernet comienzan con *en*, las interfaces WLAN comienzan con *wl* y las interfaces WWAN comienzan con *ww*.
- El o los siguientes caracteres representan el tipo de adaptador: *o* significa incorporado, *ss* significa ranura de conexión en caliente y *p* significa ubicación geográfica de PCI. Aunque no esté disponible de manera predeterminada, los administradores también utilizan una *x* para incorporar una dirección MAC.
- Por último, un número *N* se utiliza para representar un índice, una identificación o un puerto.
- Si el nombre fijo no puede determinarse, se usarán los nombres tradicionales, como *ethN*.

Por ejemplo, la primera interfaz de red integrada puede tener el nombre **eno1** y una interfaz de tarjeta PCI puede llamarse **enp2s0**. Los nombres nuevos facilitan la distinción de la relación entre un puerto y su nombre si el usuario conoce los dos, pero la desventaja es que el usuario no puede suponer que un sistema con una interfaz llame **eth0** a esa interfaz.



## nota

Los nombres de las interfaces de red pueden anularse. Si el administrador instaló y habilitó el paquete **biosdevname**, o si definió reglas de asignación de nombres a dispositivos **udev** personalizadas, los valores anularán el esquema de asignación de nombres predeterminado. Dependiendo del soporte para **biosdevname** en el BIOS del sistema, podrán utilizarse nombres como **em1**, **em2**, etc. para tarjetas de red incorporadas (correspondientes a sus nombres en el chasis). Las tarjetas PCI(e) se representan con **pYpX** (por ejemplo: **p4p1**), en el que **Y** es el número de ranura PCI y **X** es el número para el puerto de esa tarjeta específica.



## Referencias

Páginas del manual: **services(5)**, **ping(8)**, **biosdevname(1)** y **udev(7)**

Es posible encontrar información adicional en los capítulos sobre la configuración de redes y la asignación de nombres de dispositivos de red consistente en la *Guía de administración de la red de Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

<https://access.redhat.com/documentation/>

# Práctica: Conceptos de red



# Validación de la configuración de red

RH124

An abstract network diagram is visible in the background on the right side of the slide. It features a central dark red circle with a white dot at its center. Several thin white lines radiate from this central point, connecting to other white dots located at various positions. These dots are further connected by additional white lines, forming a network structure. The background is a solid dark red color.

# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder probar y revisar la configuración de red actual con las utilidades básicas.

## Visualización de las direcciones IP

El comando `/sbin/ip` se usa para mostrar la información del dispositivo y la dirección.

```
[student@desktopX ~]$ ip addr show eth0  
2: eth0: <BROADCAST,MULTICAST, ❶ UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000  
    ❷ link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff  
    ❸ inet 172.25.0.10/24 brd ❹ 172.25.0.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    ❺ inet6 fe80::5054:ff:fe00:b/64 scope link  
        valid_lft forever preferred_lft forever
```

- ❶ Una interfaz activa tiene el estado de **UP**.
- ❷ La línea del enlace especifica la dirección de hardware (MAC) del dispositivo.
- ❸ La línea `inet` muestra la dirección IPv4 y el prefijo.
- ❹ La dirección, el alcance y el nombre del dispositivo de transmisión también están en esta línea.
- ❺ La línea `inet6` muestra la información de IPv6.

El comando `ip` también puede usarse para mostrar las estadísticas sobre el rendimiento de la red. Los paquetes recibidos (RX) y transmitidos (TX), los errores y los contadores dados de baja pueden usarse para identificar problemas de red provocados por congestión, poca memoria y saturación de ejecuciones.

```
[student@desktopX ~]$ ip -s link show eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
link/ether 52:54:00:00:00:0a brd ff:ff:ff:ff:ff:ff
  RX: bytes  packets  errors  dropped overrun mcast
    269850    2931      0       0        0        0
  TX: bytes  packets  errors  dropped carrier collsns
    300556    3250      0       0        0        0
```

# Solución de problemas de ruta

El comando `/sbin/ip` también se usa para mostrar la información de ruta.

```
[student@desktopX ~]$ ip route
default via 172.25.0.254 dev eth0  proto static  metric 1024
172.25.X.0/24 dev eth0  proto kernel  scope link  src 172.25.X.10
10.0.0.0/8 dev eth1  proto kernel  scope link  src 10.0.0.11
```

Todos los paquetes que estén destinados para la red 10.0.0.0/8 se enviarán directamente al destino mediante la `eth1` del dispositivo. Todos los paquetes que estén destinados para la red 172.25.X.0/24 se enviará directamente al destino mediante la `eth0` del dispositivo. Todos los demás paquetes se enviarán al enrutador predeterminado que está ubicado en 172.25.X.254, y también mediante la `eth0` del dispositivo.

El comando `ping` se usa para comprobar la conectividad. El comando continuará ejecutándose hasta que se presione **Ctrl+C**, a menos que se indiquen otras opciones para limitar la cantidad de paquetes enviados.

```
[student@desktopX ~]$ ping -c3 172.25.X.254
```

Para realizar el seguimiento de la ruta hacia un host remoto, use **traceroute** o **tracpath**. Ambos comandos pueden usarse para realizar el seguimiento de una ruta con paquetes de UDP; sin embargo, muchas redes bloquean el tráfico de UDP e ICMP. El comando **traceroute** tiene opciones para realizar el seguimiento de la ruta con paquetes UDP (predeterminado), ICMP (**-I**) o TCP (**-T**), pero es probable que no se instalen de manera predeterminada.

```
[student@desktopX ~]$ tracpath access.redhat.com
...
4: 71-32-28-145.rcmt.qwest.net 48.853ms asymm 5
5: dcp-brdr-04.inet.qwest.net 100.732ms asymm 7
6: 206.111.0.153.ptr.us.xo.net 96.245ms asymm 7
7: 207.88.14.162.ptr.us.xo.net 85.270ms asymm 8
8: ae1d0.cir1.atlanta6-ga.us.xo.net 64.160ms asymm 7
9: 216.156.108.98.ptr.us.xo.net 108.652ms
10: bu-ether13.atlmgmq46w-bcr00.tbone.rr.com 107.286ms asymm 12
...
```

Cada línea del resultado de **tracpath** representa un enrutador o *hop* por donde pasa el paquete entre el origen y el destino final. Se proporciona información adicional como disponible, que incluye la sincronización en ambos sentidos (RTT) y cualquier cambio en el tamaño de la unidad de transmisión máxima (MTU).

# Solución de problemas en puertos y servicios

Los servicios TCP usan sockets como terminales para la comunicación y se componen de una dirección IP, protocolo y número de puerto. En general, los servicios están atentos a los puertos estándar mientras que los clientes usan un puerto disponible en forma aleatoria. Los nombres más conocidos de puertos estándares están enumerados en el archivo **/etc/services**.

El comando **ss** se usa para mostrar las estadísticas del socket. El comando **ss** tiene por objeto reemplazar la herramienta anterior **netstat**, incluida en el paquete *net-tools*, que algunos administradores de sistemas pueden conocer más, pero que es probable que no siempre esté instalada.

```
[student@desktopX ~]$ ss -ta
State      Recv-Q Send-Q      Local Address:Port      Peer Address:Port
LISTEN     0      128             *:sunrpc                  *:
LISTEN     0      128             ① *:ssh                     *:
LISTEN     0      100             ② 127.0.0.1:smtp            *:
LISTEN     0      128             *:36889                   *:
ESTAB      0      0              ③ 172.25.X.10:ssh          172.25.254.254:59392
LISTEN     0      128             :::sunrpc                 :::
LISTEN     0      128             ④ :::ssh                    :::
LISTEN     0      100             ⑤ :::1:smtp                 :::
LISTEN     0      128             :::34946                   :::
```

- ① El puerto usado para SSH escucha todas las direcciones IPv4. El "\*" se usa para indicar "todos" cuando se hace referencia a los puertos o las direcciones IPv4.
- ② El puerto usado para SMTP presta atención a la interfaz de circuito de retorno de la IPv4 127.0.0.1.
- ③ La conexión SSH establecida está en la interfaz 172.25.X.10 y se origina de un sistema con una dirección de 172.25.254.254.
- ④ El puerto usado para SSH está atento a todas las direcciones IPv6. Se usa la sintaxis "::" para representar todas las interfaces de IPv6.
- ⑤ El puerto usado para SMTP presta atención a la interfaz de circuito de retorno de la IPv6 ::1.

## Opciones para ss y netstat.

Opción	Descripción
-n	Muestra números en lugar de nombres para las interfaces y los puertos.
-t	Muestra los sockets TCP.
-u	Muestra los sockets UDP.
-l	Muestra solo los sockets a los que está atento.
-a	Muestra todos los sockets (a los que presta atención y los establecidos).
-p	Muestra el proceso de usar los sockets.



## Referencias

Páginas del manual: `ip-link(8)`, `ip-address(8)`, `ip-route(8)`, `ip(8)`, `ping(8)`, `tracepath(8)`, `traceroute(8)`, `ss(8)` y `netstat(8)`.

Es posible encontrar información adicional en el capítulo sobre configuración de la red en la *Guía de administración de la red de Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

| <https://access.redhat.com/documentation/>



# Práctica: Cómo examinar la configuración de red

# Configuración de red con nmcli

RH124



# Objetivos

Tras finalizar esta sección, los estudiantes deberían poder administrar valores y dispositivos de red con `nmcli` y NetworkManager.

## NetworkManager

NetworkManager es un demonio que monitorea y administra valores de red. Además del demonio, hay una miniaplicación del área de notificaciones de GNOME que proporciona información sobre el estado de la red. Las herramientas gráficas y de la línea de comandos se comunican con NetworkManager y guardan archivos de configuración en el directorio `/etc/sysconfig/network-scripts`.

Un *dispositivo* es una interfaz de red. Una *conexión* es una configuración utilizada para un dispositivo que está compuesto por un grupo de valores. Es posible que existan múltiples conexiones para un dispositivo, pero solo puede haber una activa por vez. Por ejemplo, un sistema normalmente está conectado con una red con valores proporcionados por DHCP. En ocasiones, el sistema debe estar conectado con una red de laboratorio o de centro de datos, que solo puede ser estática. En lugar de cambiar la configuración manualmente, cada configuración puede almacenarse como una conexión independiente.

# Visualización de información de red con nmcli

Para visualizar una lista con todas las conexiones, use **nmcli con show**. Para enumerar solo las conexiones activas, añada la opción **--active**.

```
[root@desktopX ~]# nmcli con show
```

NAME	UUID	TYPE	DEVICE
static-eth0	f3e8dd32-3c9d-48f6-9066-551e5b6e612d	802-3-ethernet	eth0
System eth0	5fb06bd0-0bb0-7fffb-45f1-d6edd65f3e03	802-3-ethernet	--
guest	f601ca8a-6647-4188-a431-dab48cc63bf4	802-11-wireless	wlp3s0

```
[root@desktopX ~]# nmcli con show --active
```

NAME	UUID	TYPE	DEVICE
static-eth0	f3e8dd32-3c9d-48f6-9066-551e5b6e612d	802-3-ethernet	eth0
guest	f601ca8a-6647-4188-a431-dab48cc63bf4	802-11-wireless	wlp3s0

Especifique una identificación de conexión (nombre) para ver información detallada sobre esa conexión. Los valores en minúscula representan la configuración de la conexión. Los nombres de propiedad y configuración se definen en la página del manual **nm-settings(5)**. Los valores en mayúscula son datos activos.

```
[root@desktopX ~]# nmcli con show "static-eth0"
...
ipv4.method:                manual
ipv4.dns:                    172.25.254.254, 8.8.8.8
ipv4.dns-search:
ipv4.addresses:              { ip = 172.25.X.10/24, gw = 172.25.X.254 }
ipv4.routes:
ipv4.ignore-auto-routes:    no
ipv4.ignore-auto-dns:       no
ipv4.dhcp-client-id:        --
ipv4.dhcp-send-hostname:    yes
ipv4.dhcp-hostname:         --
ipv4.never-default:         no
ipv4.may-fail:              yes
ipv6.method:                auto
...
```

El comando **nmcli** también puede usarse para visualizar el estado del dispositivo e información detallada sobre el mismo.

```
[root@desktopX ~]# nmcli dev status
DEVICE  TYPE      STATE      CONNECTION
eth0     ethernet  connected  static-eth0
wlp3s0   wifi      connected  guest
lo       loopback  unmanaged  --
[root@desktopX ~]# nmcli dev show eth0
GENERAL.DEVICE:                eth0
GENERAL.TYPE:                  ethernet
GENERAL.HWADDR:                52:54:00:00:00:0A
GENERAL.MTU:                   1500
GENERAL.STATE:                 100 (connected)
GENERAL.CONNECTION:            static-eth0
GENERAL.CON-PATH:              /org/freedesktop/NetworkManager/
ActiveConnection/1
WIRED-PROPERTIES.CARRIER:     on
IP4.ADDRESS[1]:                ip = 172.25.X.10/24, gw = 172.25.X.254
IP4.DNS[1]:                    172.25.254.254
IP6.ADDRESS[1]:                ip = fe80::5054:ff:fe00:b/64, gw = ::
```

# Creación de conexiones de red con nmcli

Cuando se crea una conexión de red nueva con **nmcli**, el orden de los argumentos importa. Los argumentos comunes aparecen primero y deben incluir el tipo y la interfaz. Luego, se deben determinar los argumentos específicos del tipo y, finalmente, definir la dirección IP, el prefijo y la información de la puerta de enlace. Múltiples direcciones IP pueden especificarse para un único dispositivo. Valores adicionales, como un servidor DNS, se definen como modificaciones una vez creada la conexión.

## Ejemplos de creación de conexiones nuevas

Realice los siguientes pasos mientras el instructor habla sobre la sintaxis **nmcli**.

1. Defina una conexión nueva con el nombre "default" que establezca su conexión automática como conexión Ethernet en el dispositivo eth0 usando DHCP.

```
[root@desktopX ~]# nmcli con add con-name "default" type ethernet ifname eth0
```

2. Cree una conexión nueva con el nombre "static"; luego especifique la dirección IP y la puerta de enlace. No establezca la conexión automática.

```
[root@desktopX ~]# nmcli con add con-name "static" ifname eth0 autoconnect no type ethernet ip4 172.25.X.10/24 gw4 172.25.X.254
```

3. El sistema establecerá la conexión automática usando DHCP al inicio. Cambie a la conexión estática.

```
[root@desktopX ~]# nmcli con up "static"
```

# Resumen de los comandos nmcli

Comandos básicos de conexiones y dispositivos de **nmcli**:

## Comandos nmcli

Comando	Usar el
estado nmcli dev	Enumerar todos los dispositivos.
nmcli con show	Enumerar todas las conexiones.
nmcli con up "<ID>"	Activar una conexión.
nmcli con down "<ID>"	Desactivar una conexión. La conexión se reiniciará si la conexión automática está activada.
nmcli dev dis <DEV>	Desactivar una interfaz y deshabilitar temporalmente la conexión automática.

Comando	Usar el
<code>nmcli net off</code>	Deshabilitar todas las interfaces administradas.
<code>nmcli con add ...</code>	Añadir una conexión nueva.
<code>nmcli con mod "&lt;ID&gt;" ...</code>	Modificar una conexión.
<code>nmcli con del "&lt;ID&gt;"</code>	Eliminar una conexión.



## nota

El comando **nmcli** también tiene un modo de edición interactivo. Para una interfaz gráfica, utilice **nm-connection-editor**.



## Referencias

Páginas del manual: **nmcli(5)**, **nmcli-examples(5)** y **nm-settings(1)**

Es posible encontrar información adicional en la sección sobre el uso de la herramienta de línea de comandos NetworkManager nmcli en la *Guía de administración de la red de Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

<https://access.redhat.com/documentation/>



# Práctica: Cómo examinar la configuración de red

# Edición de archivos de configuración de red

## Objetivos

Tras finalizar esta sección, los estudiantes deberían poder modificar los parámetros de configuración de red mediante la edición de los archivos de configuración.

## Modificación de la configuración de red

También se puede configurar la red editando los archivos de configuración de interfaz. Los archivos de configuración de interfaz controlan las interfaces de software para dispositivos de red individuales. En general, estos archivos se denominan **/etc/sysconfig/network-scripts/ifcfg-`<name>`**, donde `<name>` se refiere al nombre del dispositivo o a la conexión que controla el archivo de configuración. A continuación, se detallan las variables estándares que se encuentran en el archivo usado para la configuración estática o dinámica.

## Opciones de configuración para el archivo ifcfg

<i>Estática</i>	<i>Dinámica</i>	<i>Cualquiera de las opciones</i>
<b>BOOTPROTO=none</b>	<b>BOOTPROTO=dhcp</b>	<b>DEVICE=eth0</b>
<b>IPADDR0=172.25.X.10</b>		<b>NAME="System eth0"</b>
<b>PREFIX0=24</b>		<b>ONBOOT=yes</b>
<b>GATEWAY0=172.25.X.254</b>		<b>UUID=f3e8dd32-3...</b>
<b>DEFROUTE=yes</b>		<b>USERCTL=yes</b>
<b>DNS1=172.25.254.254</b>		

En estos parámetros de configuración estáticos, las variables para la dirección IP, el prefijo y la puerta de enlace tienen un número al final. Esto permite que se asignen varios conjuntos de valores a la interfaz. La variable DNS también tiene un número que se usa para especificar el orden de la búsqueda cuando se especifican varios servidores.

Después de modificar los archivos de configuración, ejecute **nmcli con reload** para que NetworkManager lea los cambios de configuración. La interfaz todavía necesita reiniciarse para que se implementen los cambios.

```
[root@serverX ~]# nmcli con reload  
[root@serverX ~]# nmcli con down "System eth0"  
[root@serverX ~]# nmcli con up "System eth0"
```



## Referencias

Página del manual (1)**nmcli**

Es posible encontrar información adicional en el capítulo sobre configuración de la red en la *Guía de administración de la red de Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

<https://access.redhat.com/documentation/>

# Práctica: Edición de archivos de configuración de red

# Configuración de nombres de host y resolución de nombre

## Objetivos

Tras finalizar esta sección, los estudiantes deberían poder configurar y probar el nombre del host del sistema y la resolución de nombre.

## Cambio de nombre del host del sistema

El comando **hostname** muestra o modifica provisoriamente el nombre del host totalmente calificado del sistema.

```
[root@desktopX ~]# hostname  
desktopX.example.com
```

Puede especificarse un nombre del host estático en el archivo **/etc/hostname**. Se usa el comando **hostnamectl** para modificar este archivo y puede utilizarse para ver el estado del nombre del host totalmente calificado del sistema. Si este archivo no existe, el nombre del host se establece mediante una consulta de DNS invertida una vez que la interfaz tiene una dirección IP asignada.

```
[root@desktopX ~]# hostnamectl set-hostname desktopX.example.com
[root@desktopX ~]# hostnamectl status
  Static hostname: desktopX.example.com
        Icon name: computer
        Chassis: n/a
  Machine ID: 9f6fb63045a845d79e5e870b914c61c9
    Boot ID: aa6c3259825e4b8c92bd0f601089ddf7
  Virtualization: kvm
  Operating System: Red Hat Enterprise Linux Server 7.0 (Maipo)
    CPE OS Name: cpe:/o:redhat:enterprise_linux:7.0:GA:server
      Kernel: Linux 3.10.0-97.el7.x86_64
    Architecture: x86_64
[root@desktopX ~]# cat /etc/hostname
desktopX.example.com
```



## Importante

El nombre del host estático se guarda en **/etc/hostname**. Las versiones anteriores de Red Hat Enterprise Linux almacenaban el nombre del host como una variable en el archivo **/etc/sysconfig/network**.



# Configuración de la resolución de nombre

El *sistema de resolución de nombres interno* se utiliza para convertir nombres de host en direcciones IP o a la inversa. Los contenidos del archivo **/etc/hosts** se verifican en primer lugar.

```
[root@desktopX ~]# cat /etc/hosts
```

El comando **getent hosts hostname** puede usarse para probar la resolución de nombre del host con el archivo **/etc/hosts**.

Si no se encuentra una entrada en ese archivo, el sistema de resolución de nombres interno buscará la información en un servidor de nombres DNS. El archivo **/etc/resolv.conf** controla la forma en que se realiza esta consulta:

- **nameserver**: la dirección IP de un servidor de nombres que se consultará. Se pueden proporcionar hasta tres directivas de servidor de nombres para proporcionar copias de seguridad en caso de que una no funcione.
- **search**: una lista de nombres de dominio para probar con un nombre del host corto. Tanto este como el **domain** no deben configurarse en el mismo archivo; si esto ocurre, prevalece la última instancia. Vea **resolv.conf(5)** para obtener más detalles.

```
[root@desktopX ~]# cat /etc/resolv.conf
# Generated by NetworkManager
domain example.com
search example.com
nameserver 172.25.254.254
```



NetworkManager actualizará el archivo **/etc/resolv.conf** con los parámetros de configuración de DNS en los archivos de configuración de conexión. Use **nmcli** para modificar las conexiones.

```
[root@desktopX ~]# nmcli con mod ID ipv4.dns IP
[root@desktopX ~]# nmcli con down ID
[root@desktopX ~]# nmcli con up ID
[root@desktopX ~]# cat /etc/sysconfig/network-scripts/ifcfg-ID
...
DNS1=8.8.8.8
...
```

El comportamiento predeterminado de **nmcli con mod *ID* ipv4.dns *IP*** es reemplazar cualquier parámetro de configuración de DNS anterior con la nueva lista de IP provista. El símbolo +/- que está frente al argumento **ipv4.dns** agregará o eliminará una entrada individual.

```
[root@desktopX ~]# nmcli con mod ID +ipv4.dns IP
```

El comando de **host** *HOSTNAME* puede usarse para probar la conectividad del servidor DNS.

```
[root@desktopX ~]# host classroom.example.com
classroom.example.com has address 172.25.254.254
[root@desktopX ~]# host 172.25.254.254
254.254.25.172.in-addr.arpa domain name pointer classroom.example.com.
```



## Importante

Si se usa DHCP, **/etc/resolv.conf** se reescribe automáticamente a medida que se inician las interfaces, a menos que usted especifique **PEERDNS=no** en los archivos de configuración de interfaz correspondientes. El cambio puede realizarse con **nmcli**.

```
[root@desktopX ~]# nmcli con mod "System eth0" ipv4.ignore-auto-dns yes
```



## Referencias

Páginas del manual: **nmcli(5)**, **hostnamectl(1)**, **hosts(1)**, **getent(1)**, **host(1)** y **resolv.conf(5)**.

Es posible encontrar información adicional en el capítulo sobre configuración de nombres de host en la *Guía de administración de red de Red Hat Enterprise Linux* para Red Hat Enterprise Linux 7, que se puede encontrar en

<https://access.redhat.com/documentation/>

# Práctica: Configuración de nombres de hosts y resolución de nombres

# Ejercicio de laboratorio: Administración de la red de Red Hat Enterprise Linux