Capítulo 3
Configuración del sistema

1. Configuración de la red

La configuración de la red en sistemas Linux es una habilidad fundamental para administradores de sistemas, ingenieros de redes y cualquier persona interesada en la administración de las tecnologías informáticas. Consiste en ajustar un conjunto de parámetros y servicios para que un ordenador Linux se pueda comunicar a través de una red informática. En el centro de esta configuración, se encuentran varios archivos de configuración, herramientas de línea de comandos e interfaces gráficas, que permiten controlar el comportamiento en red de la máquina.

Linux ofrece una gran flexibilidad a la hora de configurar la red, gracias a su arquitectura abierta y a la variedad de herramientas disponibles. Entre ellas se encuentran `ifconfig` e `ip`, que se utilizan para configurar interfaces de red y, las más recientes, `netplan` o `NetworkManager`, que ofrecen un enfoque más automatizado y fácil de usar para la administración de redes. Cada distribución de Linux puede ofrecer sus propias herramientas o preferencias para la gestión de redes, lo que subraya la importancia de consultar la documentación específica de cada distribución.

La configuración de la red en Linux también puede implicar la configuración de servicios de red más avanzados, como el enrutamiento IP, el cortafuegos con `iptables` o `nftables` y la seguridad de las comunicaciones con herramientas como OpenSSL para configurar VPN o conexiones seguras. Comprender estos servicios y su configuración es crucial para garantizar la seguridad y la eficiencia de la red. Veremos estos conceptos en detalle en un capítulo posterior.

Dominar la configuración de red en Linux es esencial para una gestión sólida y segura de las infraestructuras informáticas. Ya sea para servidores, estaciones de trabajo o sistemas integrados, Linux ofrece las herramientas y la flexibilidad necesarias para satisfacer casi cualquier requisito de red. Con la constante evolución de la tecnología de redes, mantenerse al día de las últimas prácticas y herramientas disponibles en Linux es esencial para cualquier profesional del sector.

La gestión y diagnóstico de redes informáticas en Linux implica un conjunto de potentes herramientas diseñadas para ayudar a los administradores a configurar, supervisar y solucionar problemas de red. Cada una de estas herramientas desempeña un papel crucial en el arsenal del administrador de red, permitiéndole gestionar eficazmente las interfaces de red, supervisar las conexiones y el flujo de datos y diagnosticar una gran variedad de problemas de red.

`ifconfig` e `ip` son esenciales para configurar interfaces de red. `ifconfig`, aunque se considera obsoleto en las distribuciones modernas, se utilizaba tradicionalmente para configurar interfaces de red, definir direcciones IP y gestionar los estados de las interfaces. Su sucesor, `ip`, ofrece una funcionalidad más robusta y detallada, permitiendo una gestión precisa de direcciones IP, rutas y políticas de enrutamiento, así como una monitorización detallada de las interfaces de red. Estas herramientas ofrecen una potente interfaz de comandos para ajustar la red a las necesidades de los administradores.

Por otro lado, `netstat` y `ss` proporcionan una visión detallada de varias estadísticas de red, como conexiones activas, tablas de enrutamiento, puertos de escucha y más. Aunque `netstat` ha sido durante mucho tiempo la referencia para obtener información de red, `ss` ha emergido como una herramienta más rápida y capaz, ofreciendo una mejor visión de los sockets y los detalles de conexión. Para el diagnóstico y la resolución de problemas, `ping` y `traceroute` tienen un valor incalculable, ya que permiten a los usuarios probar la conectividad de la red y rastrear las rutas que siguen los paquetes a través de la red, proporcionando pistas vitales para resolver problemas de red.

Comprender y dominar la configuración de interfaces de red a través de la línea de comandos, también es crucial en un contexto en el que las redes son cada vez más complejas y en el que los requisitos de seguridad y rendimiento son elevados. Esto permite a los administradores gestionar la red de forma eficaz, con capacidad para automatizar tareas, programar configuraciones y realizar ajustes de precisión, garantizando que la red satisface las necesidades de la organización, al tiempo que se mantiene ágil y segura ante los constantes cambios tecnológicos.

Vamos a examinar más de cerca algunas de estas herramientas.

1.1 ifconfig

La herramienta `ifconfig` (*interface configuration*), se utiliza tradicionalmente para configurar, gestionar y mostrar parámetros de interfaz de red. Aunque `ip` se considera obsoleta, sigue estando disponible en muchos sistemas por motivos de compatibilidad. Estos son los comandos principales:

- **Mostrar todas las interfaces de red**: simplemente ejecutando `ifconfig` sin ningún argumento, puede mostrar todas las interfaces de red activas, incluidas sus direcciones IP, el estado de la interfaz, la máscara de subred, etc.

  ```
  ifconfig
  ```

- **Configurar una dirección IP**: permite configurar manualmente una dirección IP y una máscara de subred, en una interfaz de red específica. Por ejemplo:

  ```
  ifconfig eth0 192.168.1.100 netmask 255.255.255.0
  ```

- **Activar/desactivar una interfaz**: activa o desactiva la interfaz de red especificada, lo que puede resultar útil para gestionar la conectividad o resolver problemas de red.
  - Activar la interfaz `eth0`: `ifconfig eth0 up`
  - Desactivar la interfaz `eth0`: `ifconfig eth0 down`
- **Configurar una dirección MAC**: cambia la dirección MAC (*Media Access Control*) de la interfaz de red, que puede ser necesaria por motivos de seguridad o de pruebas. Por ejemplo: `ifconfig eth0 hw ether 02:01:02:03:04:05` cambia la dirección MAC de la interfaz `eth0` a `02:01:02:03:04:05`.

– **Mostrar una interfaz específica**: muestra detalles como la dirección IP, la máscara de subred y otra información de configuración de red para una interfaz específica.

`ifconfig eth0` muestra detalles de la interfaz `eth0`, incluyendo su dirección IP, máscara de subred y estado (*active* o *inactive*).

```
root@debian-networking:~# ifconfig
Dev: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.60  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::6b7e:cd1b:a907:38d0  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:01:af:ad  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7464  bytes 2428287 (2.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.18  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 08:00:27:48:c6:c8  txqueuelen 1000  (Ethernet)
        RX packets 441973  bytes 402288423 (383.6 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68247  bytes 7430328 (7.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.44  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 08:00:27:0f:ef:10  txqueuelen 1000  (Ethernet)
        RX packets 489928  bytes 414977765 (395.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 55970  bytes 5108867 (4.8 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s9: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.48  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 2a01:e0a:15e:b5b0:e8d7:5b40:6d84:586f  prefixlen 64  scopeid 0x0<global>
        inet6 2a01:e0a:15e:b5b0:1b28:8a2d:a43a:f12b  prefixlen 64  scopeid 0x0<global>
```

```
root@debian-networking:~# ifconfig enp0s3 192.168.1.19 netmask 255.255.255.0
root@debian-networking:~# ifconfig enp0s3 down
root@debian-networking:~# ifconfig enp0s3 up
root@debian-networking:~# ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.19  netmask 255.255.255.0  broadcast 192.168.1.255
        ether 08:00:27:48:c6:c8  txqueuelen 1000  (Ethernet)
        RX packets 442526  bytes 402364174 (383.7 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 68311  bytes 7436112 (7.0 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

root@debian-networking:~#
```

1.2 ip

`ip` es una herramienta más reciente y potente que `ifconfig`, que ofrece una amplia gama de funciones para gestionar interfaces de red, rutas, direcciones IP y mucho más.

- **Mostrar interfaces**: este comando se utiliza para mostrar detalles de todas las interfaces de red del sistema, incluidas las direcciones IP, el estado de la interfaz y otra información de red relevante.

  ```
  ip addr show
  ```

- **Añadir o eliminar direcciones IP**: permite modificar la configuración IP de una interfaz, añadiendo o eliminando direcciones IP.
  - Añadir: asigna una nueva dirección IP a una interfaz de red específica.

    ```
    ip add add 192.168.1.10/24 dev eth0
    ```

  - Borrar: elimina una dirección IP de una interfaz de red.

    ```
    ip addr del 192.168.1.10/24 dev eth0
    ```

- **Gestionar el estado de las interfaces**: activa o desactiva las interfaces de red, lo que resulta esencial para la gestión de la conectividad y el control de las interfaces.
  - Activar la interfaz `eth0`: `ip link set eth0 up`
  - Desactivar la interfaz `eth0`: `ip link set eth0 down`
- **Modificar dirección MAC**: cambia la dirección MAC de la interfaz de red, lo que puede ser necesario por motivos de seguridad, confidencialidad o pruebas.

  `ip link set dev eth0 address aa:bb:cc:dd:ee:ff` cambia la dirección MAC de la interfaz `eth0` a `aa:bb:cc:dd:ee:ff`.

- **Mostrar rutas**: muestra la tabla de enrutamiento IP del sistema, mostrando cómo se enrutan los paquetes de una interfaz a otra o a redes específicas.

  `ip route show` muestra la tabla de enrutamiento actual, incluyendo rutas por defecto, rutas específicas de red y métricas asociadas.

– **Añadir o eliminar rutas**: modifica las rutas que utilizan los paquetes para llegar a su destino, un aspecto crucial de la configuración de la red.
  – Añadir: crea una nueva ruta en la tabla de enrutamiento.

    `ip route add 192.168.2.0/24 via 192.168.1.1` añade una ruta para llegar a la red `192.168.2.0/24`, a través de la puerta de enlace o pasarela `192.168.1.1`.
  – Borrar: elimina una ruta existente de la tabla de enrutamiento.

    `ip route del 192.168.2.0/24` borra la ruta a la red `192.168.2.0/24`.

```
root@debian-networking:~# ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:48:c6:c8 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.19/24 brd 192.168.1.255 scope global enp0s3
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:0f:ef:10 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.44/24 brd 192.168.1.255 scope global dynamic enp0s8
       valid_lft 28648sec preferred_lft 28648sec
4: enp0s9: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:01:af:ad brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.48/24 brd 192.168.1.255 scope global enp0s9
       valid_lft forever preferred_lft forever
    inet6 2a01:e0a:15e:b5b0:b7f3:325f:7a5a:2f74/64 scope global temporary deprecated dynamic
       valid_lft 86386sec preferred_lft 0sec
    inet6 2a01:e0a:15e:b5b0:4d4f:a34f:8d2f:4296/64 scope global temporary deprecated dynamic
       valid_lft 86386sec preferred_lft 0sec
    inet6 2a01:e0a:15e:b5b0:e8d7:5b40:6d84:586f/64 scope global temporary deprecated dynamic
       valid_lft 86386sec preferred_lft 0sec
    inet6 2a01:e0a:15e:b5b0:1b28:8a2d:a43a:f12b/64 scope global dynamic mngtmpaddr noprefixroute
       valid_lft 86386sec preferred_lft 86386sec
    inet6 fe80::946a:fd84:3f6a:c08e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
5: enp0s10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
```

```
root@debian-networking:~# ip addr add 192.168.119/24 dev enp0s3
root@debian-networking:~# ip link set enp0s3 up
root@debian-networking:~# ip route show
default via 192.168.1.254 dev enp0s8
default via 192.168.1.254 dev Dev proto static metric 400
10.8.0.0/24 dev tun0 proto kernel scope link src 10.8.0.1
192.168.1.0/24 dev enp0s8 proto kernel scope link src 192.168.1.44
192.168.1.0/24 dev enp0s10 proto kernel scope link src 192.168.1.28
192.168.1.0/24 dev enp0s9 proto kernel scope link src 192.168.1.48
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.19
192.168.1.0/24 dev Dev proto kernel scope link src 192.168.1.60 metric 400
192.168.119.0/24 dev enp0s3 proto kernel scope link src 192.168.119.0
root@debian-networking:~# ip route add 192.168.2.0/24 via 192.168.1.254
root@debian-networking:~# ip route show
default via 192.168.1.254 dev enp0s8
default via 192.168.1.254 dev Dev proto static metric 400
10.8.0.0/24 dev tun0 proto kernel scope link src 10.8.0.1
192.168.1.0/24 dev enp0s8 proto kernel scope link src 192.168.1.44
192.168.1.0/24 dev enp0s10 proto kernel scope link src 192.168.1.28
192.168.1.0/24 dev enp0s9 proto kernel scope link src 192.168.1.48
192.168.1.0/24 dev enp0s3 proto kernel scope link src 192.168.1.19
192.168.1.0/24 dev Dev proto kernel scope link src 192.168.1.60 metric 400
192.168.2.0/24 via 192.168.1.254 dev enp0s8
192.168.119.0/24 dev enp0s3 proto kernel scope link src 192.168.119.0
root@debian-networking:~#
```

1.3 netstat

`netstat` (*Network Statistics*) es una herramienta de línea de comandos que muestra conexiones de red, tablas de enrutamiento, estadísticas de interfaz, conexiones enmascaradas y miembros de multidifusión. Aunque hoy en día se utiliza menos, ya que ha sido sustituido en gran medida por `ss`, sus principales comandos son:

– `netstat -a`: muestra todas las conexiones y puertos de escucha.
– `netstat -t`: muestra las conexiones TCP.
– `netstat -u`: lista las conexiones UDP.
– `netstat -l`: sólo muestra los sockets en modo escucha.
– `netstat -r`: muestra la tabla de enrutamiento.
– `netstat -s`: muestra estadísticas por protocolo.
– `netstat -p`: muestra el programa asociado a cada socket.

Observación

Los argumentos se pueden agrupar, por ejemplo `netstat -atu` para mostrar todas las conexiones TCP y UDP.

```
root@debian-networking:~# netstat -atu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 localhost:ipp           0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:domain          0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:ssh             0.0.0.0:*               LISTEN
tcp6       0      0 [::]:domain             [::]:*                  LISTEN
tcp6       0      0 [::]:ssh                [::]:*                  LISTEN
tcp6       0      0 localhost:ipp           [::]:*                  LISTEN
udp        0      0 0.0.0.0:631             0.0.0.0:*
udp        0      0 0.0.0.0:38801           0.0.0.0:*
udp        0      0 0.0.0.0:domain          0.0.0.0:*
udp        0      0 0.0.0.0:bootpc          0.0.0.0:*
udp        0      0 0.0.0.0:openvpn         0.0.0.0:*
udp        0      0 0.0.0.0:mdns            0.0.0.0:*
udp6       0      0 [::]:46480              [::]:*
udp6       0      0 [::]:domain             [::]:*
udp6       0      0 [::]:mdns               [::]:*
root@debian-networking:~#
```

Los campos que aparecen en la salida del comando se explican a continuación:

– `Proto`: indica el protocolo de conexión, que puede ser TCP, UDP, TCP6 (TCP sobre IPv6) o UDP6 (UDP sobre IPv6).

– `Recv-Q`: representa la cola de recepción, que indica el número de bytes en espera de ser recibidos por una aplicación local.

– `Send-Q`: representa la cola de envío, mostrando el número de bytes en espera de ser enviados por una aplicación local.

– `Local Address`: muestra la dirección y el puerto en los que el ordenador local está escuchando o conectado. Puede ser una dirección IP específica de una de las interfaces de la máquina o «0.0.0.0» (o «::» para IPv6), lo que significa que la máquina está escuchando en todas las direcciones IP disponibles.

– `Foreign Address`: indica la dirección IP y el puerto de la máquina remota, con la que se establece la conexión local. Un asterisco (*) en esta columna significa que el sistema escucha conexiones entrantes desde cualquier dirección remota.

– `State`: muestra el estado de la conexión. Por ejemplo, `LISTEN` significa que la aplicación está esperando una conexión entrante, `ESTABLISHED` significa que hay una conexión activa.

Los puertos se indican después de las direcciones IP, separados por dos puntos. Por ejemplo, «localhost:ipp» indica que el servicio de impresión por Internet (IPP) escucha las conexiones entrantes en la dirección loopback (localhost), que suele ser `127.0.0.1` para IPv4 o `::1` para IPv6.

He aquí algunos detalles específicos sobre determinadas líneas de la salida:

– Las líneas con `domain` se refieren generalmente al puerto 53, utilizado por el servicio DNS.
– `ssh` se refiere al servicio Secure Shell que, por defecto, escucha en el puerto 22.
– `mdns` en el puerto 5353 lo utiliza Multicast DNS, parte del protocolo Zeroconf.
– `ipp` en el puerto 631 se refiere al protocolo de impresión de Internet, utilizado para los servicios de impresión en red.

1.4 ss

`ss` (*Socket Statistics*) es una herramienta moderna que sustituye a `netstat`, ofreciendo más información y siendo más rápida. Se utiliza para mostrar información detallada sobre sockets individuales. Sus principales comandos son:

– `ss -t`: muestra los sockets TCP.
– `ss -u`: lista los sockets UDP.
– `ss -l`: muestra los sockets en modo escucha.
– `ss -a`: muestra todos los sockets.
– `ss -p`: muestra el proceso que utiliza el socket.
– `ss -s`: muestra las estadísticas del socket.