

Comandos de Red en Linux - Ejercicios

Comandos de Red en Linux - Ejercicios

1. `ip a`
2. `ifconfig`
3. `ping`
4. `nslookup`
5. `netstat`
6. `curl`
7. `hostname`
8. `whois`
9. `ip route`
 - 🔗 Ejemplos de uso
10. `ip link`
 - 🔗 Ejemplos de uso

Cuestiones sobre Comandos de Red en Linux

1. `ip a`

El comando `ip a` (o `ip address`) muestra la configuración de las interfaces de red y las direcciones IP asociadas. Es una alternativa moderna al comando `ifconfig`. Proporciona detalles sobre el estado de las interfaces, direcciones IP asignadas, máscaras de red, etc.

Sintaxis:

```
ip a
```

2. `ifconfig`

El comando `ifconfig` muestra y configura las interfaces de red. Es más antiguo y está siendo reemplazado por `ip`, pero sigue siendo útil en algunos sistemas. Permite ver las direcciones IP asignadas y configurar las interfaces.

Sintaxis:

```
ifconfig
```

3. `ping`

El comando `ping` verifica la conectividad de red entre dos dispositivos mediante el envío de paquetes ICMP. Permite medir la latencia y la pérdida de paquetes entre el host de origen y el destino.

Sintaxis:

```
ping [dirección IP o dominio]
```

4. nslookup

El comando `nslookup` se usa para realizar consultas DNS, lo que permite resolver nombres de dominio a direcciones IP o viceversa. Es útil para verificar la configuración DNS y la disponibilidad de nombres de dominio.

Sintaxis:

```
nslookup [nombre de dominio o dirección IP]
```

5. netstat

El comando `netstat` proporciona estadísticas detalladas sobre las conexiones de red, incluyendo las conexiones activas, las tablas de enrutamiento y las estadísticas de las interfaces. Es útil para diagnosticar problemas de red y monitorear el tráfico de red.

Sintaxis:

```
netstat
```

6. curl

`curl` es una herramienta de línea de comandos para transferir datos usando varios protocolos, como HTTP, FTP, y más. Se utiliza frecuentemente para descargar archivos, realizar pruebas de APIs, y verificar la conectividad de servicios web.

Sintaxis:

```
curl [URL]
```

7. hostname

El comando `hostname` muestra o configura el nombre del host del sistema. Es útil para verificar el nombre de la máquina o cambiarlo temporalmente sin reiniciar el sistema.

Sintaxis:

```
hostname
```

8. whois

El comando `whois` realiza consultas sobre la información de registro de un dominio, incluyendo datos del propietario, fechas de creación y expiración, y los servidores DNS asociados. Es útil para obtener información detallada de dominios en internet.

Sintaxis:

```
whois [nombre de dominio]
```

9. ip route

Sirve para **ver, añadir, modificar y borrar rutas** en la tabla de enrutamiento del kernel

Se usa principalmente para:

- **Mostrar** rutas configuradas.
- **Definir** rutas hacia redes u hosts.
- **Configurar** el gateway (ruta por defecto).
- **Eliminar** rutas.

🔍 Ejemplos de uso

1. Mostrar la tabla de enrutamiento

```
ip route show
```

Ejemplo de salida:

```
default via 192.168.1.1 dev eth0 proto dhcp metric 100
192.168.1.0/24 dev eth0 proto kernel scope link src 192.168.1.50 metric 100
```

👉 Aquí vemos que la **ruta por defecto** (`default`) va a través de `192.168.1.1` en `eth0`. Y que la red local `192.168.1.0/24` está directamente conectada.

2. Añadir una ruta a una red

```
sudo ip route add 10.10.10.0/24 via 192.168.1.1 dev eth0
```

👉 Para llegar a la red `10.10.10.0/24`, se usará la puerta de enlace `192.168.1.1` a través de `eth0`.

3. Añadir una ruta a un host específico

```
sudo ip route add 172.16.0.50 via 192.168.1.1
```

👉 Sólo el host `172.16.0.50` pasará por el gateway `192.168.1.1`.

4. Añadir la ruta por defecto (gateway)

```
sudo ip route add default via 192.168.1.1 dev eth0
```

👉 Configura el **gateway predeterminado** para todo el tráfico no especificado.

5. Eliminar una ruta

```
sudo ip route del 10.10.10.0/24
```

👉 Elimina la ruta hacia la red `10.10.10.0/24`.

6. Reemplazar una ruta

```
sudo ip route replace default via 192.168.1.254 dev eth0
```

👉 Si existe una ruta por defecto la sobrescribe; si no, la añade.

7. Rutas avanzadas (por tabla de enrutamiento)

```
ip route show table all
```

👉 Permite ver todas las tablas de enrutamiento, no sólo la principal (útil en **policy routing**).

10. ip link

Sirve para **mostrar y gestionar interfaces de red** en Linux (`eth0`, `wlan0`, `lo`, etc.).

Con él puedes:

- Ver información detallada de las interfaces.
- Activar o desactivar una interfaz (UP/DOWN).
- Cambiar nombre a una interfaz.
- Cambiar dirección MAC.
- Ajustar parámetros avanzados de red.

🔍 Ejemplos de uso

1. Mostrar todas las interfaces de red

```
ip link show
```

Salida típica:

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000
    link/ether 08:00:27:4a:35:9c brd ff:ff:ff:ff:ff:ff
```

👉 Te muestra **ID de interfaz, nombre, estado, MTU, dirección MAC, etc.**

2. Mostrar sólo una interfaz concreta

```
ip link show eth0
```

👉 Filtra la información de la interfaz **eth0**.

3. Levantar una interfaz (activarla)

```
sudo ip link set eth0 up
```

👉 Activa la tarjeta de red **eth0**.

4. Bajar una interfaz (desactivarla)

```
sudo ip link set eth0 down
```

👉 Desactiva la interfaz **eth0** (útil para pruebas o evitar tráfico temporalmente).

5. Cambiar el nombre de una interfaz

```
sudo ip link set eth0 name lan0
```

👉 Renombra la interfaz de **eth0** a **lan0**.

6. Cambiar la dirección MAC

```
sudo ip link set dev eth0 address 02:1A:2B:3C:4D:5E
```

👉 Establece una nueva dirección **MAC** para la interfaz **eth0**.

(Ojo: *algunas tarjetas no permiten cambiar la MAC*).

⚡ Resumen rápido

- `ip link show` → ver interfaces.
- `ip link set dev x up/down` → activar/desactivar.
- `ip link set dev x name nuevo_nombre` → cambiar nombre.
- `ip link set dev x address MAC` → cambiar MAC.

Cuestiones sobre Comandos de Red en Linux

Responde a las siguientes cuestiones - Incluye un bloque de código con el comando y, si procede, una captura de pantalla con la salida de dicho comando

IMPORTANTE: Comprueba los nombres de tus interfaces, y sustituye `ethxx` por las tuyas

1. Muestra todas las interfaces de red activas y sus direcciones IP en el sistema.

```
$ ip link show  
$
```

```
jonatan@clienteUbuntu:~$ ip link show  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode DEFAULT group default qlen 1000  
    link/ether 08:00:27:61:2f:54 brd ff:ff:ff:ff:ff:ff
```

2. ¿Cómo mostrarías solo la información de la interfaz de red **eth0** usando `ip a`?

```
$ ip a show dev enp0s3
```

```
jonatan@clienteUbuntu:~$ ip a show dev enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:61:2f:54 brd ff:ff:ff:ff:ff:ff
        inet 172.17.0.3/24 brd 172.17.0.255 scope global dynamic noprefixroute enp0s3
            valid_lft 574sec preferred_lft 574sec
        inet6 fe80::a00:27ff:fe61:2f54/64 scope link
            valid_lft forever preferred_lft forever
```

3. Configura manualmente la dirección IP `192.168.1.100/24` en la interfaz `eth0` con `ifconfig`.

```
$ sudo ifconfig eth0 192.168.1.100 netmask 255.255.255.0 up
```

4. Envía 10 paquetes ICMP a la dirección IP `8.8.8.8` usando `ping`.

5.

6. Consulta la dirección IP de `www.example.com` usando `nslookup`.

7. Muestra las conexiones TCP activas en el sistema usando `netstat`.

8. Descarga el contenido de la página principal de `www.example.com` usando `curl` y guárdalo en un archivo llamado `example.html`.

9. Consulta el nombre del host actual del sistema.

10. Obtén la información de registro del dominio `example.com` usando `whois`.

11. Cambia temporalmente el nombre del host a `servidor01` usando `hostname`.

12. Envía un ping a la dirección `192.168.1.1` y muéstralos en modo detallado (verbose).

13. Muestra las estadísticas de la red, como la cantidad de paquetes transmitidos, usando `netstat`.

14. Realiza una consulta inversa para obtener el nombre de dominio asociado a la IP `8.8.8.8` con `nslookup`.

15. Configura temporalmente la máscara de subred `255.255.255.128` en la interfaz `eth1` usando `ifconfig`.

16. Muestra las rutas de enrutamiento actuales usando `netstat`.

17. Realiza una solicitud HTTP GET a la API de GitHub para obtener los repositorios de `usuario123` usando `curl`.

18. Envía un ping a la dirección `2001:4860:4860::8888` (IPv6 de Google) con `ping6` y limita los paquetes a 4.

19. Obtén las estadísticas de los sockets activos en el sistema con `netstat`.

20. Cambia temporalmente la dirección MAC de la interfaz `eth0` a `00:11:22:33:44:55` usando `ifconfig`.

21. Realiza una solicitud HTTP POST a `https://httpbin.org/post` enviando el usuario `admin` y la contraseña `12345` usando `curl`.

22. Consulta el nombre de dominio completo (FQDN) de tu sistema usando `hostname`.

23. Muestra solo las conexiones activas en la interfaz `eth0` usando `netstat`.

24. Muestra las conexiones activas con nombres de dominio en lugar de direcciones IP usando `netstat`.

25. Configura una nueva puerta de enlace predeterminada con la dirección 192.168.1.1 usando `ip route`.
26. ¿Qué comando usarías para ver todas las rutas configuradas en tu sistema?
27. ¿Cómo configuras que todo el tráfico destinado a la red 10.10.10.0/24 pase por el gateway 192.168.1.1 en la interfaz `eth0`?
28. ¿Cómo eliminas la ruta añadida en el ejercicio anterior?
29. Si la interfaz `eth0` está deshabilitada, ¿qué comando usarías para levantarla?
30. ¿Qué comando utilizas para asignar la dirección MAC 02:1A:2B:3C:4D:5E a la interfaz `eth0`?
31. ¿Cómo renombrarías la interfaz `eth0` para que pase a llamarse `1an0`?