

*Telefonica*

---



**HOSTED IP**

Linux



# Linux

Trabajando con la consola



# Linux

## Trabajar con la consola

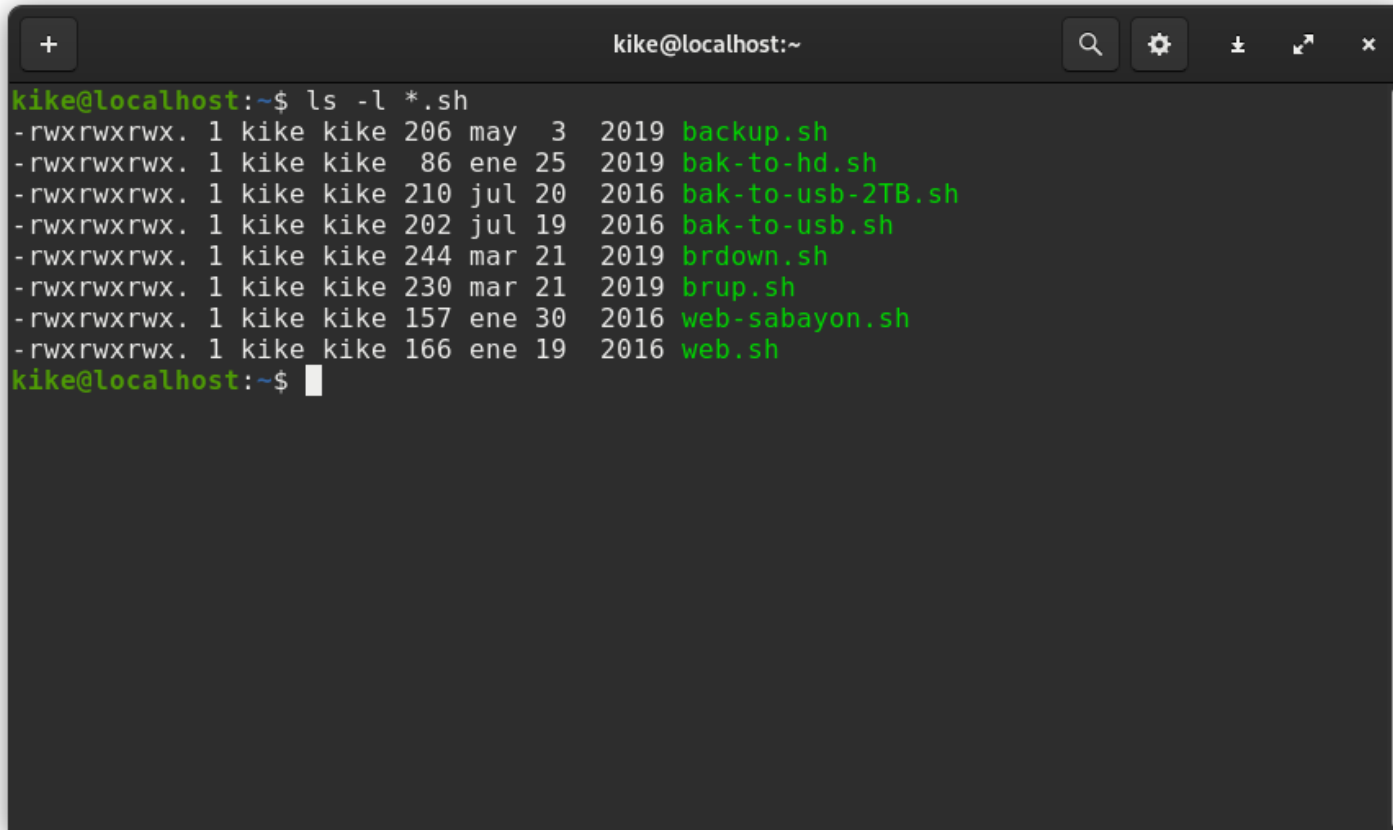
El interprete de comandos de Linux, es un potente lenguaje que nos permite escribir scripts para automatizar procesos

El interprete más utilizado es BASH

# Linux

## Trabajar con la consola

Se pueden utilizar varias consolas a la vez



```
kike@localhost:~$ ls -l *.sh
-rwxrwxrwx. 1 kike kike 206 may  3  2019 backup.sh
-rwxrwxrwx. 1 kike kike  86 ene 25  2019 bak-to-hd.sh
-rwxrwxrwx. 1 kike kike 210 jul 20  2016 bak-to-usb-2TB.sh
-rwxrwxrwx. 1 kike kike 202 jul 19  2016 bak-to-usb.sh
-rwxrwxrwx. 1 kike kike 244 mar 21  2019 brdown.sh
-rwxrwxrwx. 1 kike kike 230 mar 21  2019 brup.sh
-rwxrwxrwx. 1 kike kike 157 ene 30  2016 web-sabayon.sh
-rwxrwxrwx. 1 kike kike 166 ene 19  2016 web.sh
kike@localhost:~$
```

A terminal window titled 'kike@localhost:~' showing the output of the command 'ls -l \*.sh'. The output lists eight shell scripts with their permissions, owner, group, size, date, and filename. The permissions for all files are '-rwxrwxrwx'. The files are: backup.sh (206 bytes, May 3, 2019), bak-to-hd.sh (86 bytes, Jan 25, 2019), bak-to-usb-2TB.sh (210 bytes, Jul 20, 2016), bak-to-usb.sh (202 bytes, Jul 19, 2016), brdown.sh (244 bytes, Mar 21, 2019), brup.sh (230 bytes, Mar 21, 2019), web-sabayon.sh (157 bytes, Jan 30, 2016), and web.sh (166 bytes, Jan 19, 2016). The prompt 'kike@localhost:~\$' is shown at the bottom.

# Linux

## Trabajar con la consola

Una característica importante, es la capacidad de referirse a más de un fichero, utilizando caracteres especiales

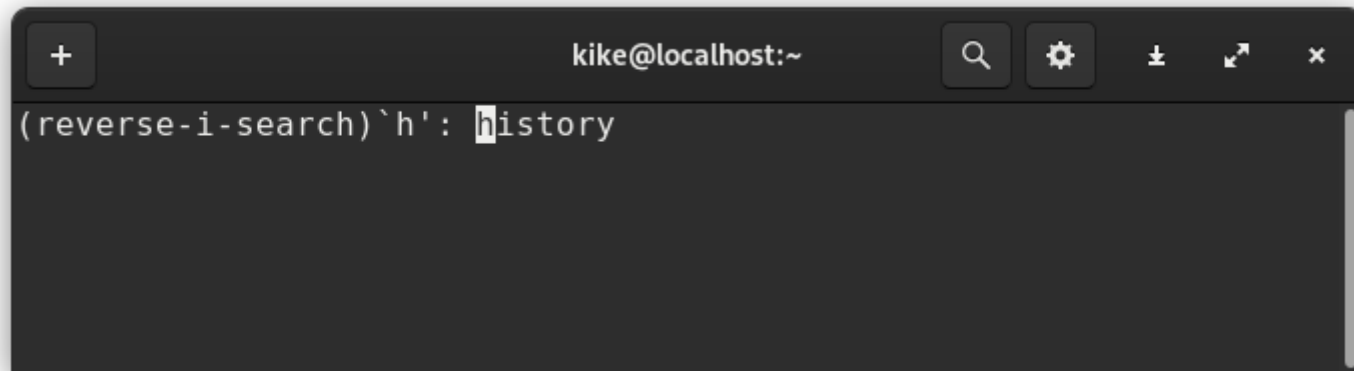
- El comodín \*, hace referencia a cualquier carácter o cadena de caracteres
- El comodín ?, hace referencia a un solo carácter

# Linux

## Trabajar con la consola

La consola, posee un histórico de comandos, se accede con las teclas del cursor

- Con las teclas del cursor
- Con el comando “history”
- Con “Ctrl+R”, podemos buscar lo último introducido que coincida con lo que vamos tecleando

A terminal window with a dark background and light text. The title bar shows a plus icon, the text 'kike@localhost:~', and icons for search, settings, and window control. The terminal content shows '(reverse-i-search)`h': h' followed by the word 'history' with a cursor at the end.

```
+ kike@localhost:~ (reverse-i-search)`h': h' history
```

# Linux

## Trabajar con la consola

### Trabajando con directorios

- Como sé en que directorio me encuentro **pwd**
- Para cambiar de directorio **cd otro\_directorio**
- Listar archivos y directorios **ls -l**
- Crear un directorio **mkdir nom\_directorio**
- Eliminar directorio vacío **rmdir nom\_directorio**

# Linux

## Trabajar con la consola

### Trabajando con ficheros

- Ver el contenido de un fichero de texto **cat fichero.conf**
- Copiando ficheros **cp fich.conf /mnt/fich.conf**
- Moviendo ficheros **mv fich.conf ./basura/**
- Borrar un fichero **rm nom\_fichero**
- Cambiar el nombre de un fichero **mv fich.bak fich.conf**



# Linux

## Trabajar con la consola

### Trabajando con ficheros

- Mostrar el contenido de un fichero o la salida de un comando pantalla a pantalla
  - **more fichero.conf**
  - **ls -l | more**
- Enviar a la pantalla los argumentos pasados
  - **echo “Esto es una prueba”**
- Mostrar todas las líneas de un fichero que coincidan con un patrón
  - **grep patron fichero.conf**
  - **sipregister | grep 3120**

# Linux

## Trabajar con la consola

### Fontaneria

- Muchos comandos, toman sus datos de la entrada estándar (stdin, teclado) y envían su salida a la salida estándar (stdout, pantalla)
- El interprete de comandos, nos permite redireccionar la salida estándar a un fichero, utilizando el operador “>”
  - **echo “Esta cadena la envio al fichero” > fichero.txt**
- También se puede redireccionar la entrada con el operador “<”
- Un filtro, es un programa que lee datos de stdin, los procesa y los devuelve a stdout

# Linux

## Trabajar con la consola

El interprete de comandos, nos permite usar pipelines, con el operador “|”, que nos permite conectar dos programas, donde la salida del primero, es enviada a la entrada del segundo

- **cat /etc/passwd | grep kike**


La redirección “>” es destructiva, sin embargo “>>”, añade los datos enviados al contenido del fichero

- **sort fichero.txt >> fichero\_ordenado.txt**

# Linux

## Trabajar con la consola

Se pueden agrupar varios comandos en un guión, para ejecutarlos con una sola orden

A terminal window with a dark background and light text. The title bar shows the path 'kike@localhost:/run/media/kike/6E917A1C1BAFE39F/Portatil-tesa'. The terminal content shows a user running 'cat brup.sh', which displays a series of Linux commands for stopping networking, adding a bridge, configuring interfaces, and starting networking again.

```
kike@localhost:/run/media/kike/6E917A1C1BAFE39F/Portatil-tesa$ cat brup.sh
#!/bin/bash
systemctl stop networking.service
brctl addbr br0
ifconfig eth0 0.0.0.0 promisc up
ifconfig eth1 0.0.0.0 promisc up
brctl addif br0 eth0
brctl addif br0 eth1
ip link set br0 up
systemctl start networking.service
kike@localhost:/run/media/kike/6E917A1C1BAFE39F/Portatil-tesa$
```

# Linux

## Trabajar con la consola

El interprete de comandos, permite definir variables, se asignan con el operador “=” y se accede a ellas, anteponiendo “\$”

```
kike@localhost:~$ echo $HOME
/home/kike
kike@localhost:~$
```

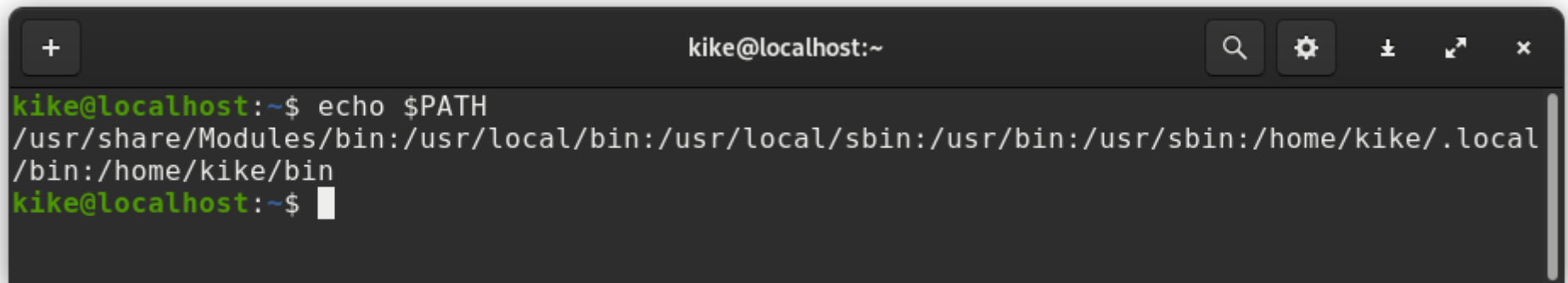
Las variables son internas al interprete, se pueden exportar al sistema con **export**

```
# Configuración entorno virtual para python 3
# Variable que determina donde ubicaremos nuestros entornos virtuales
export WORKON_HOME=$HOME/Proyectos/Python
# Variable para los scripts de automatización de tareas en virtualenvwrapper
export VIRTUALENVWRAPPER_HOOK_DIR=$WORKON_HOME/hooks
# Utilización de python3
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
# Script que los comandos para utilizar virtualenvwrapper
source /usr/bin/virtualenvwrapper-3.sh
```

# Linux

## Trabajar con la consola

El interprete de comandos, usa la variable PATH para localizar los ficheros ejecutables que tecleamos en la consola

A terminal window with a dark background. The title bar shows 'kike@localhost:~'. The terminal content shows the command 'echo \$PATH' being executed, resulting in the output: '/usr/share/Modules/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/home/kike/.local/bin:/home/kike/bin'. The prompt 'kike@localhost:~\$' is visible at the end of the line.

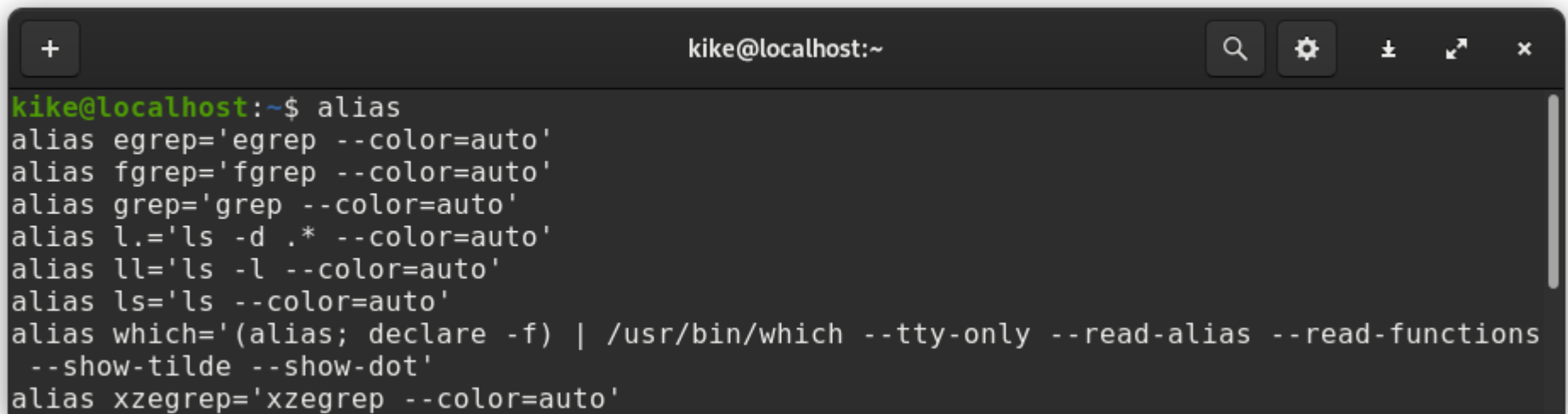
```
kike@localhost:~$ echo $PATH
/usr/share/Modules/bin:/usr/local/bin:/usr/local/sbin:/usr/bin:/usr/sbin:/home/kike/.local
/bin:/home/kike/bin
kike@localhost:~$
```

# Linux

## Trabajar con la consola

Los **alias**, permiten que una cadena, se sustituya por una palabra cuando se emplee esta, como una orden simple

El nombre del **alias** y el texto de reemplazo, pueden contener cualquier entrada válida para la shell

A terminal window with a dark background and light text. The title bar at the top shows 'kike@localhost:~' and standard window controls (search, settings, zoom, close). The terminal content shows the command 'alias' being executed, followed by a list of aliases: 'egrep', 'fgrep', 'grep', 'l.', 'll', 'ls', 'which', and 'xzegrep'. Each alias is defined as a shorthand for a specific command with options like '--color=auto' or '--show-tilde --show-dot'.

```
kike@localhost:~$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias which='(alias; declare -f) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --show-dot'
alias xzegrep='xzegrep --color=auto'
```

# Linux

## Gestión de procesos





# Linux

## Gestión de procesos

Un demonio, es un programa que se ejecuta como un proceso, en segundo plano, esperando que ocurran eventos y ofreciendo servicios. Un buen ejemplo, es un servidor web, o el proceso que gestiona la red

En linux, los demonios, son administrados por **systemd**. **Systemd**, proporciona un gestor de sistemas y servicios que se ejecuta como PID 1 e inicia el resto del sistema.

El comando **systemctl**, es la interfaz de usuario para su administración y control del sistema

# Linux

## Gestión de procesos

Con **systemctl**, podemos gestionar el estado de un servicio

- Activar y parar un servicio
  - **systemctl start/stop nombre.service**
- Reiniciar o recargar un servicio
  - **systemctl restart/reload nombre.service**
- Ver el estado de un servicio un servicio
  - **systemctl status nombre.service**
- Reiniciar o recargar un servicio
  - **systemctl restart/reload nombre.service**

# Linux

## Gestión de procesos

### Estado de un proceso

```
kike@localhost:~$ systemctl status postgresql
● postgresql.service - PostgreSQL database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql.service; disabled; vendor preset: disabled)
   Active: inactive (dead)

kike@localhost:~$ systemctl status NetworkManager
● NetworkManager.service - Network Manager
   Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-02-10 17:51:39 CET; 3h 4min ago
     Docs: man:NetworkManager(8)
  Main PID: 1216 (NetworkManager)
    Tasks: 4 (limit: 18852)
   Memory: 13.8M
      CPU: 6.685s
   CGroup: /system.slice/NetworkManager.service
           └─1216 /usr/sbin/NetworkManager --no-daemon

feb 10 20:50:37 localhost.localdomain NetworkManager[1216]: <info> [1581364237.6472] device (wlp64s0): se>
feb 10 20:50:37 localhost.localdomain NetworkManager[1216]: <info> [1581364237.6618] device (wlp64s0): su>
feb 10 20:50:37 localhost.localdomain NetworkManager[1216]: <info> [1581364237.6618] device (p2p-dev-wlp6>
feb 10 20:50:37 localhost.localdomain NetworkManager[1216]: <info> [1581364237.6692] device (wlp64s0): su>
feb 10 20:50:37 localhost.localdomain NetworkManager[1216]: <info> [1581364237.6695] device (p2p-dev-wlp6>
feb 10 20:55:52 localhost.localdomain NetworkManager[1216]: <info> [1581364552.6412] device (wlp64s0): se>
feb 10 20:55:52 localhost.localdomain NetworkManager[1216]: <info> [1581364552.6511] device (wlp64s0): su>
feb 10 20:55:52 localhost.localdomain NetworkManager[1216]: <info> [1581364552.6512] device (p2p-dev-wlp6>
feb 10 20:55:52 localhost.localdomain NetworkManager[1216]: <info> [1581364552.6681] device (wlp64s0): su>
feb 10 20:55:52 localhost.localdomain NetworkManager[1216]: <info> [1581364552.6684] device (p2p-dev-wlp6>
kike@localhost:~$
```

# Linux

## Gestión de procesos

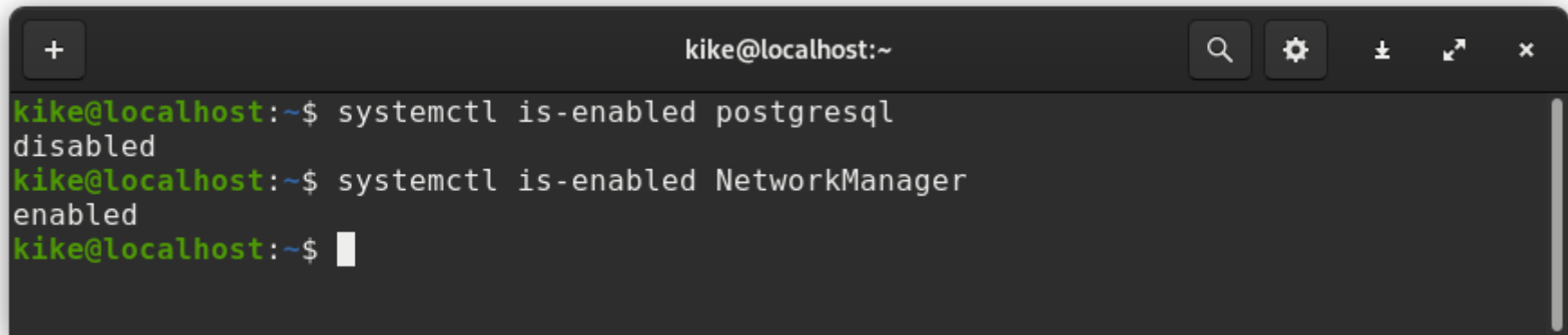
Con **systemctl**, podemos gestionar si el servicio arranca al iniciar la máquina o no

- Activar unidad para inicio en el arranque
  - **systemctl enable nombre.service**
- Ver si está activada en el arranque
  - **systemctl is-enable nombre.service**
- Desactivar arranque automático
  - **systemctl disable nombre.service**

# Linux

## Gestión de procesos

Ver si un proceso está habilitado en el arranque



```
kike@localhost:~$ systemctl is-enabled postgresql
disabled
kike@localhost:~$ systemctl is-enabled NetworkManager
enabled
kike@localhost:~$
```

# Linux

## Gestión de procesos

Con **systemctl**, podemos gestionar el arranque de la máquina

- Provocar un reinicio del sistema
  - **systemctl reboot**
- Apagado del sistema
  - **systemctl poweroff**
- Suspensión o hibernación del sistema
  - **systemctl suspend/hibernate nombre.service**

# Linux

## Configuración de red



# Linux

## Configuración de red

Todas las comunicaciones de red, se manejan entre interfaces de software configurados y dispositivos físicos conectados al sistema

- Un interfaz es un nombre lógico que tiene asociado un dispositivo físico
- A cada interfaz, hay que asignarle un conjunto de parámetros que definan su conectividad



# Linux

## Configuración de red

Las interfaces se nombran dependiendo de su localización física en la motherboard

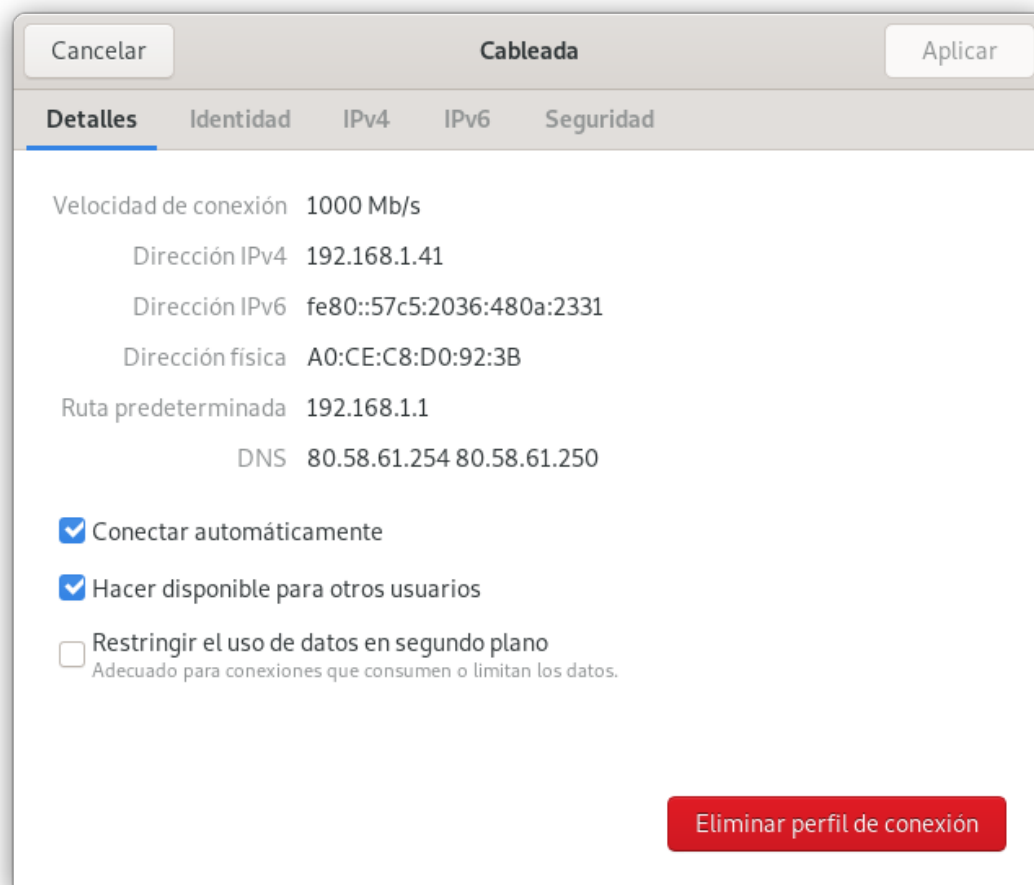
```
kike@localhost:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp61s0ulu3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether a0:ce:c8:d0:92:3b brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.41/24 brd 192.168.1.255 scope global dynamic noprefixroute enp61s0ulu3
        valid_lft 30247sec preferred_lft 30247sec
    inet6 fe80::57c5:2036:480a:2331/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
3: wlp64s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 46:e7:ea:77:81:31 brd ff:ff:ff:ff:ff:ff
4: virbr0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default qlen 1000
    link/ether 52:54:00:5f:a0:63 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.1/24 brd 192.168.122.255 scope global virbr0
        valid_lft forever preferred_lft forever
5: virbr0-nic: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel master virbr0 state DOWN group default qlen 1000
    link/ether 52:54:00:5f:a0:63 brd ff:ff:ff:ff:ff:ff
kike@localhost:~$
```

# Linux

## Configuración de red

**NetworkManager** es un programa que proporciona a los sistemas la detección y configuración automática de red

Para configurar la red, se utiliza un applet del entorno gráfico (depende del entorno gráfico)



# Linux

## Configuración de red

**NetworkManager**, corre con privilegios de root y por defecto, está configurado para iniciarse en el arranque

- **systemctl status NetworkManager**

# Linux

## Configuración de red

**NetworkManager**, también proporcionan un conjunto de comandos para su utilización en consola

- **nmcli**, son la serie de comandos para interactuar desde la consola
- Los comandos **ip**, se pueden utilizar en paralelo y serán reconocidos por **nmcli**, pero no serán persistentes cuando se reinicie el ordenador

# Linux

## Configuración de red

### Comandos ip

- Mostrar direcciones e información de red

```
# Mostrar información sobre todas las direcciones  
ip addr
```

```
# Mostrar información de un solo dispositivo  
ip addr show dev enp0s0
```

```
# Mostrar información de todas las interfaces  
ip link
```

```
# Mostrar información de un solo dispositivo  
ip link show dev enp0s0
```

```
# Mostrar estadísticas de interface  
ip -s link
```

# Linux

## Configuración de red

### Comandos ip

- Modificar direccionamiento y propiedades

```
# Configuramos la dirección 192.168.1.1 con máscara 255.255.255.0 al interface enp0s0
ip addr add 192.168.1.1/24 dev enp0s0

# Borramos la dirección antes creada
ip addr del 192.168.1.1/24 dev enp0s0

# Cambiamos el estado de el interface
ip link set enp0s0 up
ip link set enp0s0 down
ip link set enp0s0 mtu 9000

# Habilita el modo promiscuo
ip link set enp0s0 promisc on
```

# Linux

## Configuración de red

### Instalación de **vlan** en Linux

- Instalamos el software necesario y cargamos el módulo necesario en el kernel
- O instalamos de forma permanente en el kernel

```
#apt-get install vlan  
#modprobe 8021q  
  
#echo "8021q" >> /etc/modules
```

# Linux

## Configuración de red

Creamos un script para configurar la interfaz de red en una VLAN concreta, en este caso configuramos el script para la VLAN 90

```
#!/bin/bash
# Paramos el servicio NetworkManager
systemctl stop NetworkManager
# Añadimos la VLAN a la interfaz ethernet
ip link add link enp3s0 name enp3s0.90 type vlan id 90
# Configuramos direccionamiento y levantamos interfaz
ip addr add $1 dev enp3s0.90
ip link set up enp3s0.90

# Comprobamos con
cat /proc/net/vlan/config
```



# Linux

## Configuración de red

Instalación de un servidor DHCP en Debian. Instalamos el servidor

```
#apt-get install isc-dhcp-server
```

En el archivo **/etc/default/isc-dhcp-server**, configuramos el interfaz de red

```
INTERFACESv4="enp3s0"
```

# Linux

## Configuración de red

En el archivo **/etc/dhcp/dhcp.conf**, configuramos los rangos de red a servir

```
subnet 192.168.2.0 netmask 255.255.255.0 {  
    range 192.168.2.11 192.168.2.254;  
    option routers 192.168.2.1  
    option broadcast-address 192.168.2.255  
}
```

*Telefónica*

---