Tutorial Post-Instalación Esencial para Linux

Fecha Actual: Jueves, 10 de abril de 2025

Ubicación: Málaga, Andalucía, España

¡Felicidades por instalar Linux! Ahora que tienes el sistema base, hay algunos pasos importantes para asegurarte de que esté actualizado, configurado a tu gusto y sea seguro. Esta guía te ayudará con las tareas iniciales más comunes. Casi todos los comandos necesitarán privilegios de administrador, así que prepárate para usar sudo.

1. Actualizar el Sistema Completamente

¿Por qué? Lo primero y más crucial es actualizar todo el software. La imagen de instalación que usaste puede tener semanas o meses, y desde entonces se habrán publicado parches de seguridad importantes, correcciones de errores y nuevas versiones de software.

¿Cómo? Abre una terminal y ejecuta los comandos según tu familia de distribución:

- **Para Debian, Ubuntu, Kali Linux y derivados (usan** apt**):**

Bash

1. Actualiza la lista de paquetes disponibles en los repositorios

sudo apt update

2. Descarga e instala las actualizaciones de todos los paquetes instalados

'dist-upgrade' maneja cambios de dependencias de forma más inteligente que 'upgrade'

sudo apt dist-upgrade -y

(Opcional) Limpia paquetes descargados y dependencias innecesarias

sudo apt autoclean

sudo apt autoremove -y

- **Para Fedora, CentOS Stream, RHEL y derivados (usan** dnf**, o** yum **en versiones más antiguas):**

Bash

DNF actualiza la lista y los paquetes en un solo paso

sudo dnf update -y

O si usas una versión más antigua con yum:

sudo yum update -y

(Opcional - DNF suele manejar esto bien, pero por si acaso) Limpiar caché

sudo dnf clean all

Nota: Una actualización grande, especialmente del kernel, podría requerir un **reinicio** para aplicar todos los cambios. El sistema te lo indicará si es necesario.

Bash

sudo reboot

2. Cambiar el Nombre del Equipo (Hostname)

¿Por qué? Durante la instalación, se asigna un nombre al equipo (hostname). Puede que quieras cambiarlo a algo más descriptivo o personal, especialmente si aceptaste el nombre por defecto. Este nombre identifica a tu máquina en la red.

¿Cómo? La forma moderna y recomendada en la mayoría de las distribuciones que usan systemd es con hostnamectl.

1. **Ver el nombre actual:**

Bash

hostnamectl

2. **Cambiar el nombre:** Reemplaza <tu_nuevo_nombre> con el nombre que desees (generalmente en minúsculas, sin espacios ni caracteres especiales, excepto quizás guiones).

Bash

sudo hostnamectl set-hostname <tu_nuevo_nombre>

3. **Verificar el cambio:**

Bash

hostnamectl

Deberías ver el nuevo "Static hostname".

4. **Importante: Actualizar** /etc/hosts **(a veces necesario):** El archivo /etc/hosts mapea direcciones IP a nombres de host localmente. Es buena idea asegurarse de que la entrada para 127.0.1.1 (o a veces 127.0.0.1 si no existe la anterior) apunte a tu nuevo hostname.

- Edita el archivo: sudo nano /etc/hosts
- Busca una línea como 127.0.1.1 nombre_antiguo y cámbiala a 127.0.1.1 <tu_nuevo_nombre>. Si no existe, puedes añadirla (aunque hostnamectl a menudo lo hace).
- Guarda (Ctrl+O, Enter) y sal (Ctrl+X).

El cambio de nombre debería ser efectivo inmediatamente o tras reiniciar la sesión o el sistema.

3. Configurar la Red

Normalmente, tu conexión de red se configura automáticamente usando **DHCP** (el router le asigna una IP a tu PC). Pero a veces necesitas una **IP estática** (manual), especialmente para servidores o configuraciones específicas.

3.a. Configuración Automática (DHCP - Por defecto)

- **Verificar:** Puedes ver la configuración IP actual asignada por DHCP con:

Bash

ip a

Busca tu interfaz de red activa (ej: eth0, enpXsY, wlan0) y mira la línea 'inet'

- **Herramientas Comunes:** La mayoría de las distros de escritorio usan **NetworkManager**. Puedes gestionarlo gráficamente (icono de red en la barra de tareas) o con herramientas de terminal:
  - nmtui: Interfaz de texto sencilla para gestionar conexiones. Ejecútala como sudo nmtui para editar. [Captura de pantalla opcional: Interfaz de nmtui]
  - nmcli: Herramienta de línea de comandos potente. nmcli device status lista interfaces, nmcli connection show lista conexiones.

3.b. Configuración Manual (IP Estática)

Necesitarás saber qué IP, máscara de red (netmask), puerta de enlace (gateway) y servidores DNS quieres usar. Consulta la configuración de tu red o habla con tu administrador de red.

Método 1: Usando NetworkManager (Común en Fedora, Ubuntu Desktop, etc.)

- **Con** nmtui **(más fácil):**
  1. Ejecuta sudo nmtui.
  2. Selecciona "Edit a connection".

3. Elige tu interfaz de red (ej: eth0) y presiona Enter (<Edit>).
4. Busca la sección "IPv4 CONFIGURATION". Cámbiala de <Automatic> a <Manual>.
5. Selecciona <Show> al lado de IPv4 para desplegar los campos.
6. Rellena "Addresses" (ej: 192.168.1.100/24 - la /24 es la máscara 255.255.255.0), "Gateway" (ej: 192.168.1.1), y "DNS servers" (ej: 8.8.8.8, 1.1.1.1).
7. Navega hasta <OK> y presiona Enter.
8. Sal de nmtui (<Back>, <Quit>).
9. Reinicia la conexión para aplicar: sudo systemctl restart NetworkManager o desactiva/activa la conexión en nmtui.

- **Con** nmcli **(más avanzado):**
    1. Identifica el nombre de tu conexión: nmcli connection show (ej: Wired connection 1)
    2. Modifica la conexión (todo en una línea o varias):

Bash

sudo nmcli connection modify "Wired connection 1" ipv4.method manual ipv4.addresses 192.168.1.100/24 ipv4.gateway 192.168.1.1 ipv4.dns "8.8.8.8,1.1.1.1"

3. Aplica los cambios (reactivando la conexión):

Bash

sudo nmcli connection down "Wired connection 1"

sudo nmcli connection up "Wired connection 1"

Método 2: Usando Netplan (Ubuntu Server y algunas versiones de escritorio)

Ubuntu usa Netplan para gestionar la configuración de red a través de archivos YAML en /etc/netplan/.

1. Busca el archivo de configuración (suele empezar por 01- o 50-):

Bash

ls /etc/netplan/

Ejemplo de salida: 01-network-manager-all.yaml

2. Edita el archivo con sudo nano /etc/netplan/<nombre_archivo>.yaml.

3. Modifica la configuración para tu interfaz (ej: eth0). **¡La sintaxis YAML es sensible a la indentación (espacios)!**

YAML

```
network:
  version: 2
  renderer: networkd # o NetworkManager
  ethernets:
    eth0: # <--- ¡Cambia esto por tu interfaz real!
      dhcp4: no # Desactiva DHCP
      addresses:
        - 192.168.1.100/24 # IP y máscara
      gateway4: 192.168.1.1 # Puerta de enlace
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1] # Servidores DNS
```

4. Guarda el archivo (Ctrl+O, Enter) y sal (Ctrl+X).
5. Aplica la configuración:

Bash

```
sudo netplan apply
```

Verificación: Después de configurar manualmente, usa ip a para confirmar que la interfaz tiene la IP estática asignada y ping google.com para verificar la conectividad y DNS.

4. Configurar el Firewall (UFW)

¿Por qué? Un firewall controla qué tráfico de red puede entrar o salir de tu sistema. Es una capa esencial de seguridad. ufw (Uncomplicated Firewall) es una herramienta popular y fácil de usar, común en Ubuntu/Debian. Fedora/RHEL usan firewalld por defecto, pero como pediste ufw, nos centraremos en él (puedes instalarlo si no viene).

¿Cómo usar UFW?

1. **Instalar UFW (si no está):**
    - sudo apt update && sudo apt install ufw (Debian/Ubuntu)
    - sudo dnf install ufw (Fedora - aunque firewalld es lo normal aquí)

2. **Verificar Estado:**

Bash

sudo ufw status verbose

Probablemente dirá 'Status: inactive' al principio

3. **Establecer Políticas por Defecto (¡Importante!):** Denegar todo lo entrante y permitir todo lo saliente es un buen punto de partida seguro.

Bash

sudo ufw default deny incoming

sudo ufw default allow outgoing

4. **Permitir Conexiones Específicas:** Abre solo los puertos/servicios que necesites.
    - **Por nombre de servicio (recomendado si está definido):**

Bash

sudo ufw allow ssh   # Permite conexiones SSH (puerto 22)

sudo ufw allow http   # Permite conexiones HTTP (puerto 80)

sudo ufw allow https   # Permite conexiones HTTPS (puerto 443)

Ver lista de aplicaciones conocidas: sudo ufw app list

    - **Por número de puerto:**

Bash

sudo ufw allow 22/tcp # Equivalente a 'allow ssh'

sudo ufw allow 8080/tcp # Permitir un puerto específico para una app web

sudo ufw allow 10000:10100/udp # Permitir un rango de puertos UDP

    - **Permitir desde una IP específica:**

Bash

sudo ufw allow from 192.168.1.50 to any port 22 proto tcp

5. **Habilitar UFW:**

Bash

sudo ufw enable

Te advertirá que puede interrumpir conexiones existentes (como SSH si te conectas remotamente). Escribe 'y' si estás seguro.

6. **Ver Reglas Activas:**

Bash

sudo ufw status numbered

7. **Eliminar una Regla:**

Bash

sudo ufw delete <numero_de_regla> # Usa el número de 'status numbered'

o

sudo ufw delete allow ssh

8. **Deshabilitar UFW:**

Bash

sudo ufw disable

9. **Restablecer a valores por defecto:**

Bash

sudo ufw reset

¡Esto borrará todas las reglas y lo desactivará!

Nota sobre firewalld **(Fedora/RHEL):** Si estás en una de estas distros, firewalld es el firewall por defecto. Los comandos básicos son diferentes:

- sudo systemctl status firewalld (Ver estado)
- sudo firewall-cmd --list-all (Ver configuración actual y reglas)
- sudo firewall-cmd --add-service=ssh --permanent (Permitir SSH permanentemente)
- sudo firewall-cmd --add-port=8080/tcp --permanent (Permitir puerto permanentemente)
- sudo firewall-cmd --reload (Aplicar cambios permanentes)

5. Habilitar Servicios (Ejemplo: SSH)

¿Por qué? Los servicios (o daemons) son programas que se ejecutan en segundo plano para realizar tareas específicas (servidor web, servidor SSH, base de datos, etc.). Puede que necesites habilitar algunos que no vienen activados por defecto. Usaremos SSH como ejemplo común para acceso remoto seguro.

¿Cómo? La mayoría de las distribuciones modernas usan systemd para gestionar servicios. Los comandos systemctl son estándar.

1. **Instalar el Servicio (si no está):** Para SSH, el paquete suele ser openssh-server.
    - sudo apt update && sudo apt install openssh-server (Debian/Ubuntu/Kali)
    - sudo dnf install openssh-server (Fedora/CentOS/RHEL)
2. **Gestionar el Servicio (**sshd **para OpenSSH):**
    - **Iniciar el servicio ahora:**

Bash

sudo systemctl start sshd

- **Detener el servicio ahora:**

Bash

sudo systemctl stop sshd

- **Reiniciar el servicio (después de cambios de configuración):**

Bash

sudo systemctl restart sshd

- **Ver el estado del servicio:**

Bash

sudo systemctl status sshd

Busca 'Active: active (running)'

- **Habilitar el servicio para que arranque automáticamente al iniciar el sistema:**

Bash

sudo systemctl enable sshd

- **Deshabilitar el inicio automático:**

Bash

sudo systemctl disable sshd

- **Ver si está habilitado para el arranque:**

Bash

sudo systemctl is-enabled sshd

3. **¡No olvides el Firewall!** Si habilitaste un firewall (UFW o firewalld), asegúrate de haber permitido el tráfico para el servicio (ej: sudo ufw allow ssh o sudo firewall-cmd --add-service=ssh --permanent && sudo firewall-cmd --reload).

6. Conclusión

¡Listo! Has completado los pasos de configuración inicial más importantes para tu nueva instalación de Linux:

- **Actualizaste** el sistema para tener lo último en seguridad y software.
- Personalizaste el **nombre del equipo**.
- Verificaste y/o configuraste la **red** (automática o manual).
- Configuraste un **firewall** básico (ufw) para proteger tu sistema.
- Aprendiste a gestionar e **habilitar servicios** esenciales como SSH.

Estos pasos sientan una base sólida para empezar a usar tu sistema Linux de forma segura y eficiente. ¡Ahora puedes continuar instalando el software específico que necesites y explorando todo lo que Linux tiene para ofrecer!