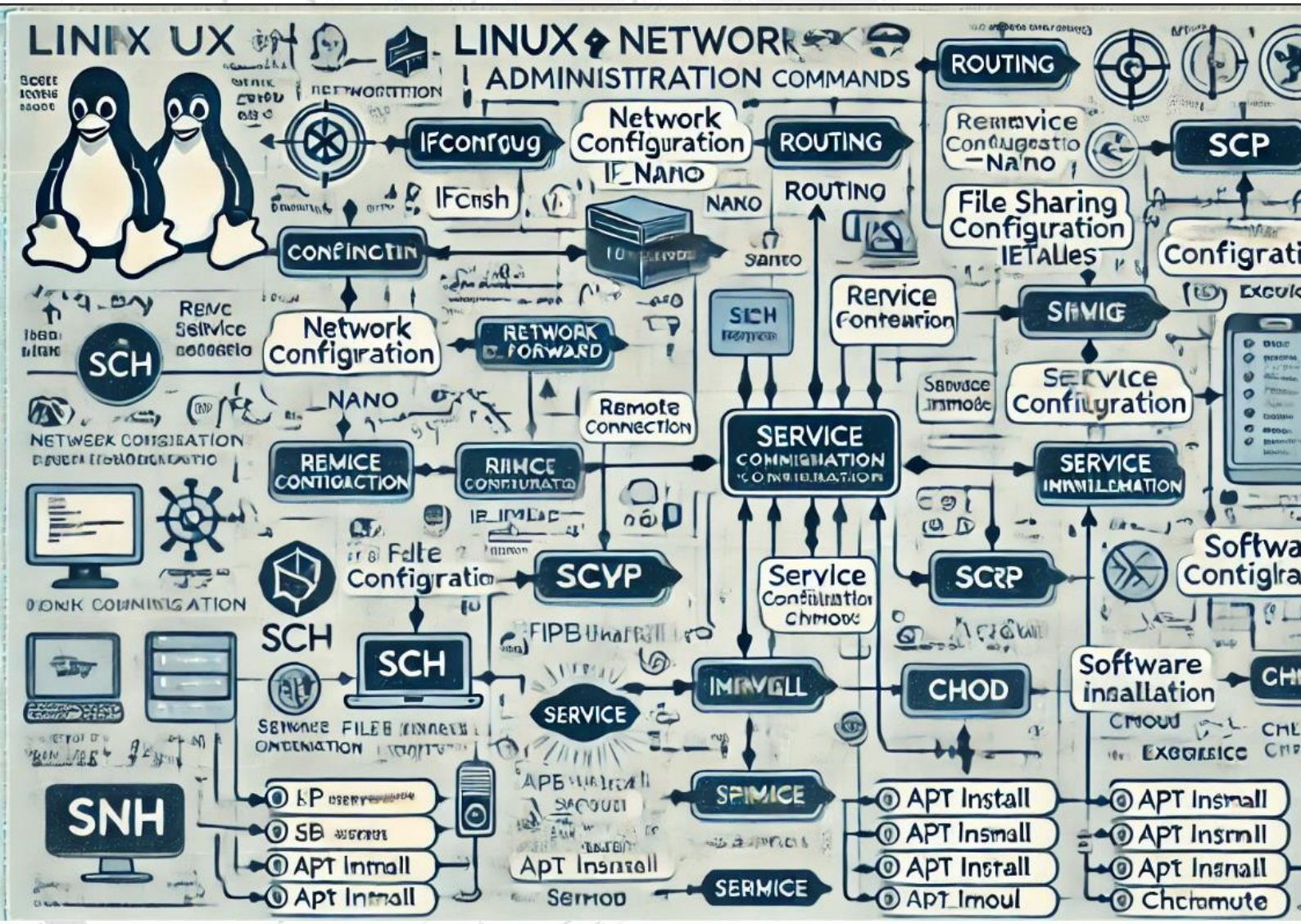
Unidad de Trabajo 10.- Administración de la red (Linux III)

CFGS Desarrollo de aplicaciones web

Sistemas informáticos

Eduardo de Lorenzo Ramos

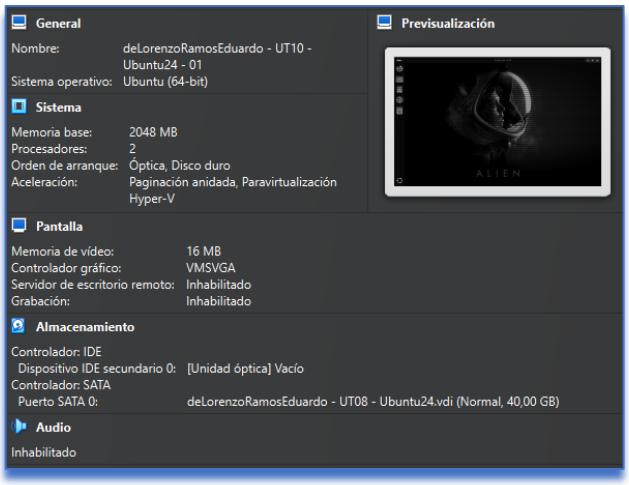


Contenido

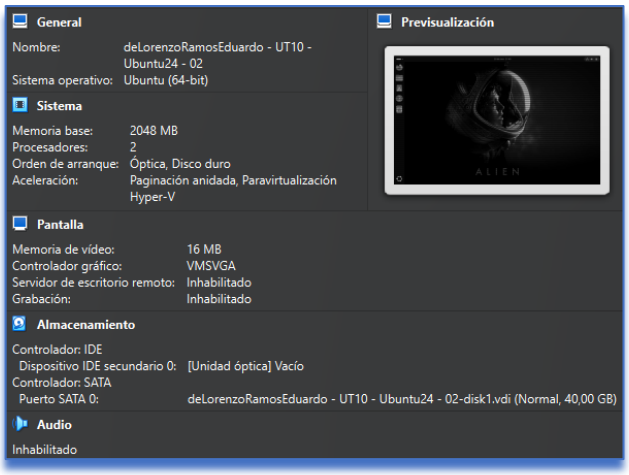
Prerrequisitos 3
Desarrollo de la práctica 4
Actividad 10.1 Configuración de los equipos en red 4
10.1.1. Configurar la red en la máquina con SO Ubuntu 24.04 - 1 4
10.1.2. Configurar la red en la máquina con SO Ubuntu 24.04 - 2 6
10.1.3. Configurar la red en la máquina con SO Windows 10 Pro 8
Actividad 10.2 Instalar Samba 10
10.2.1. Instalación del servidor Samba 10
10.2.2. Configuración del servidor Samba 11
10.2.3. Creación y concesión de permisos 14
10.2.4. Acceso desde Ubuntu al equipo servidor 15
10.2.5. Acceso desde Windows al equipo servidor 17
Actividad 10.3 Servicio SSH 17
10.3.1. Instalar servicio ssh 17
10.3.2. Conectar desde clienteLinux y ejecutar algunos comandos 18
10.3.3. Copiar el archivo /etc/hostname del servidor a la máquina cliente con el nombre hostnameServidor 19
10.3.4. Copiar el directorio /home/juana del cliente Linux al servidor como el usuario luis 20
Actividad 10.4 Realizar un mapa conceptual 22
Bibliografía/Webgrafía 22

Prerrequisitos

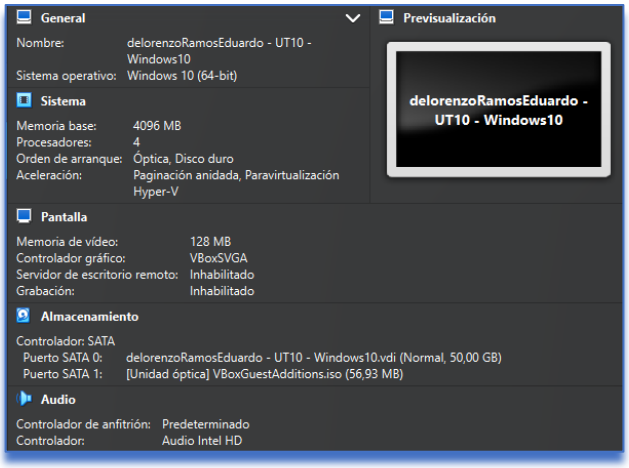
Para la elaboración de la práctica hemos preparado 3 máquinas virtuales con la siguiente configuración.



La máquina virtual configurada en VirtualBox, llamada deLorenzoRamosEduardo - UT10 - Ubuntu24 - 01, utiliza el sistema operativo Ubuntu (64 bits) con 2 GB de memoria RAM y 2 núcleos de CPU asignados, con aceleración habilitada mediante Paginación Anidada y Paravirtualización Hyper-V. En cuanto a gráficos, dispone de 16 MB de memoria de video y utiliza el controlador VMSVGA, adecuado para sistemas Linux, El almacenamiento un disco virtual SATA (.vdi) de 40 GB.



La máquina virtual configurada en VirtualBox, llamada deLorenzoRamosEduardo - UT10 - Ubuntu24 - 02, es una clonación de la anterior, por lo que todos sus parámetros de configuración son idénticos.



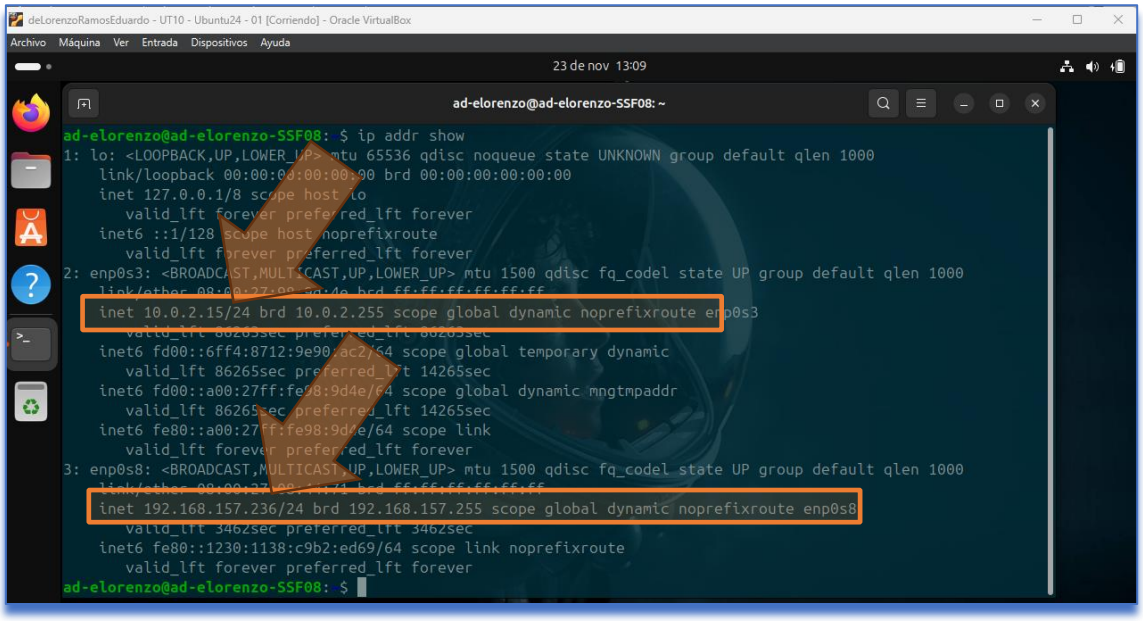
La máquina virtual deLorenzoRamosEduardo - UT10 - Windows10, configurada en VirtualBox, utiliza el sistema operativo Windows 10 (64 bits) con 4 GB de memoria RAM y 4 núcleos de CPU, priorizando el arranque desde dispositivos ópticos y disco duro, con aceleración habilitada mediante Paginación Anidada y Paravirtualización Hyper-V. El almacenamiento se organiza mediante una controladora SATA que incluye un disco virtual principal (Windows10.vdi) de 50 GB en formato.

Desarrollo de la práctica

Actividad 10.1 Configuración de los equipos en red

10.1.1. Configurar la red en la máquina con SO Ubuntu 24.04 - 1

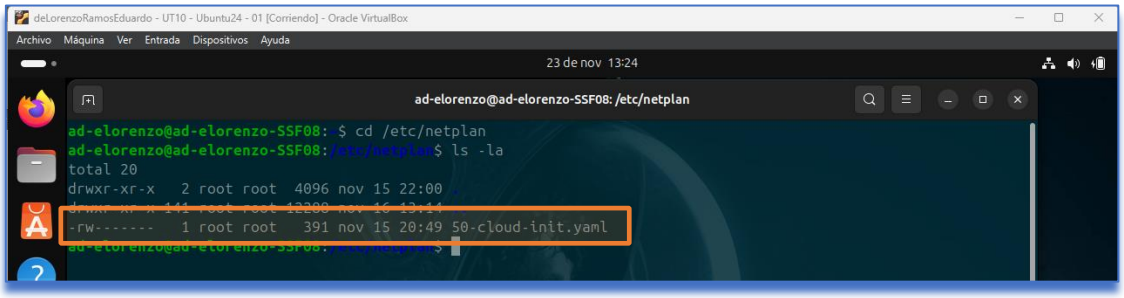
10.1.1.1. Accedemos a la sesión de nuestro usuario en el sistema (creado durante la instalación) y desde el escritorio abrimos un terminal para comprobar el estado de la configuración de los dos puertos de red habilitados en la máquina. Usamos el comando **ip addr show** para que nos muestre la información.



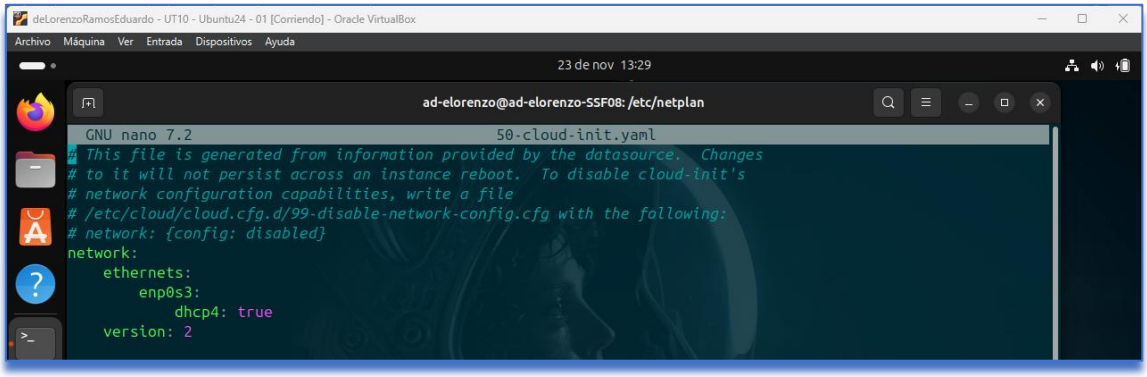
Las dos tarjetas de red (enp0s3, en modo nat, y enp0s8, en modo adaptador puente) responden correctamente, recibiendo ipv4 del host y del router respectivamente.

Configurar IP de la segunda tarjeta de red

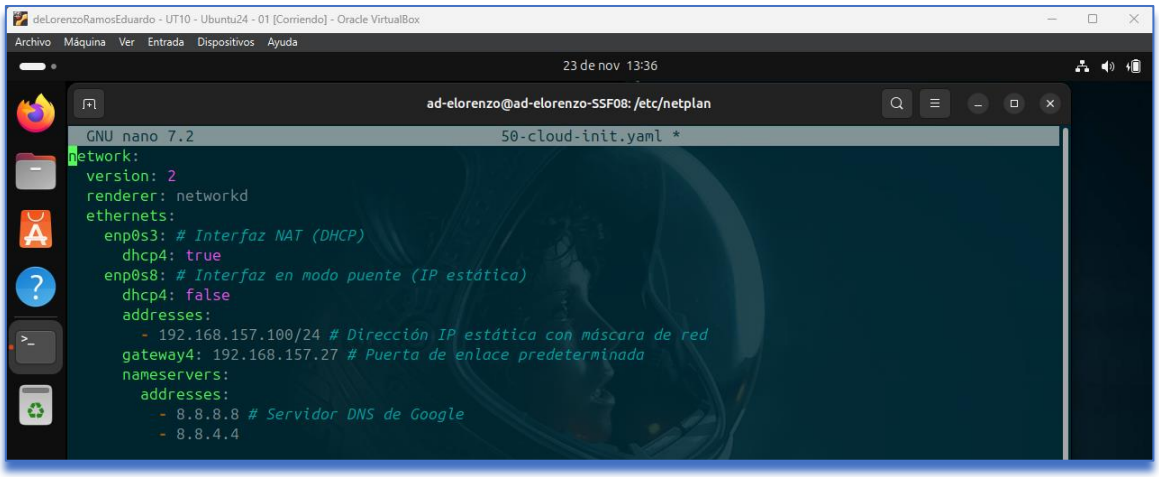
10.1.1.2. Como Ubuntu utiliza **Netplan** para gestionar la configuración de red el primer paso será localizar el archivo, por lo que introduciremos en el terminal el comando **cd /etc/netplan/**. Lo listamos con el comando **ls -la** y encontramos el archivo que debemos modificar para la configuración: **50-cloud.init.yaml**.



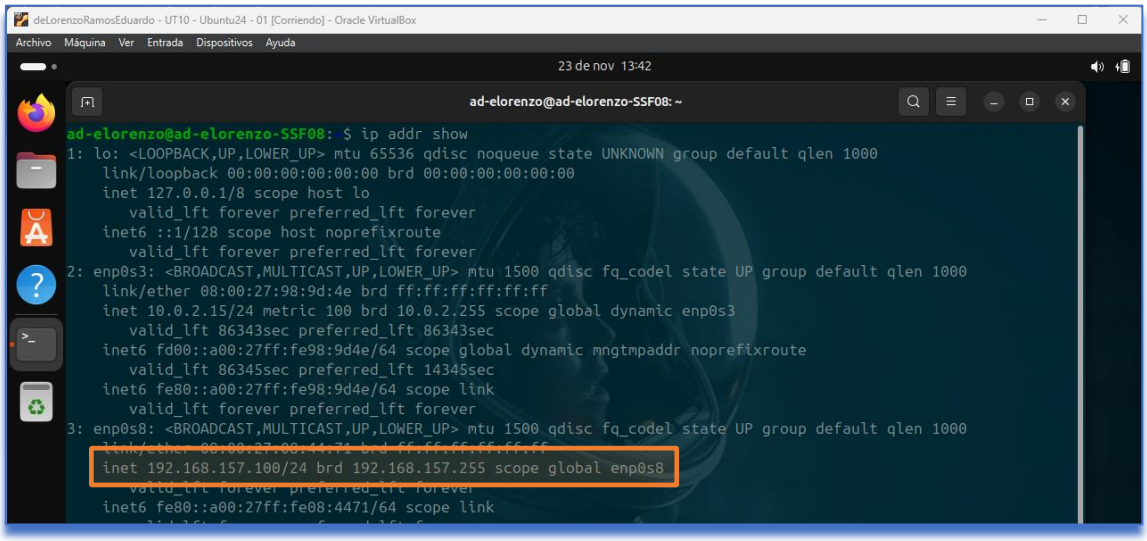
10.1.1.3. Para modificarlo usamos el comando **sudo nano 50-cloud.init.yaml**. Esta acción nos muestra el archivo de configuración para que podamos editarlo.



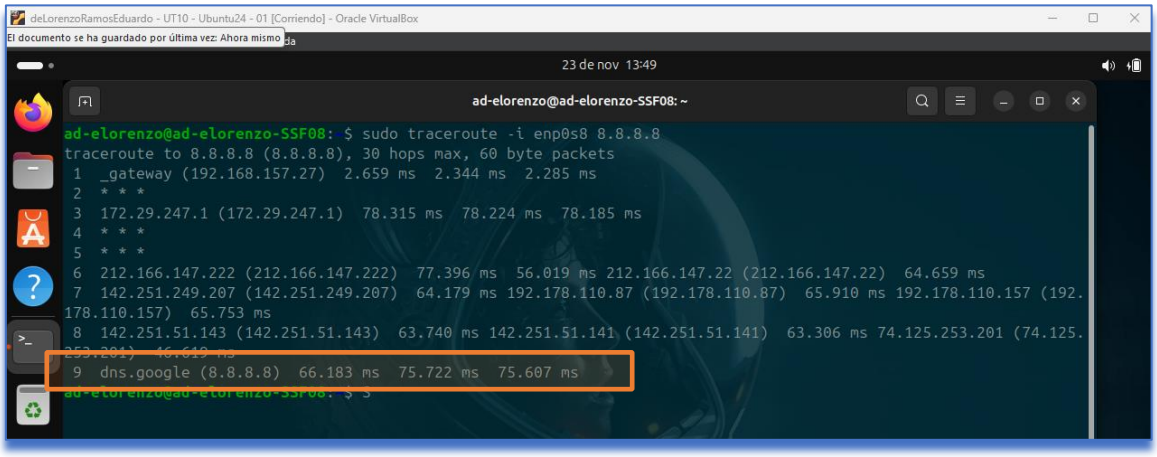
10.1.1.4. Realizamos la configuración del archivo. Mantenemos por DHCP la asignada por NAT a enp0s3, pero asignamos una IPv4 estática a enp0s8. Guardamos, salimos y reiniciamos el sistema para que se inicie con los cambios realizados.



10.1.1.5. Comprobamos ahora con **ip addr show** que los cambios se ha hecho efectivos.



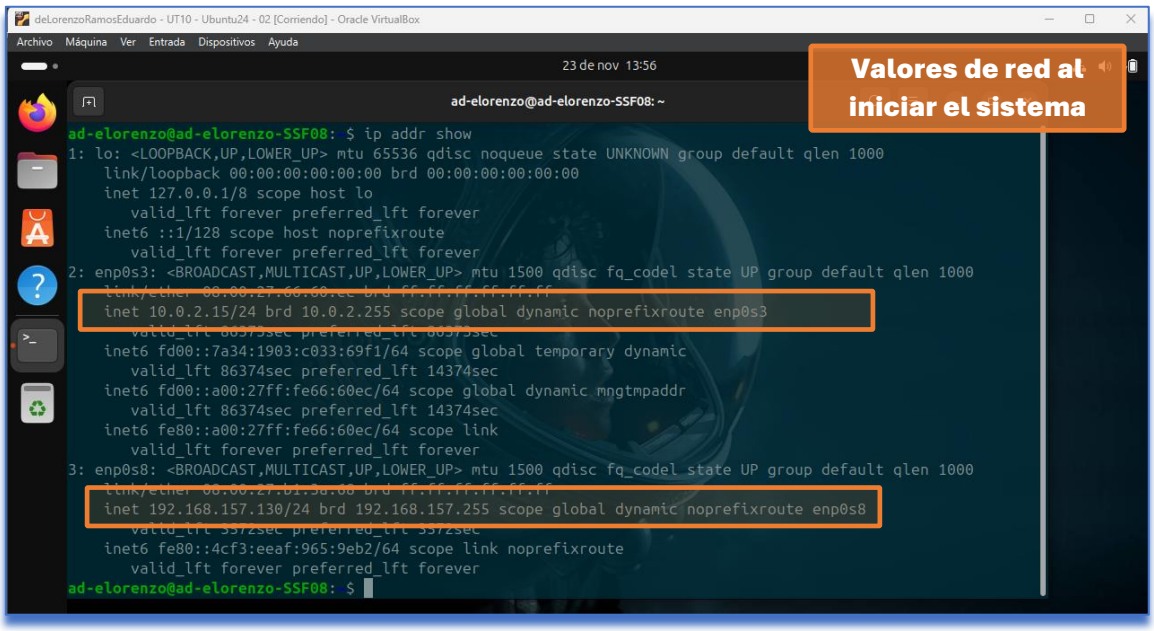
10.1.1.6. Ahora vamos a verificar que la configuración responde correctamente. Para ello vamos a forzar que la prueba de conectividad salga por la enp0s8 que hemos fijado como estática y no por la NAT. Para ello usamos el comando **sudo traceroute -i enp0s8**.

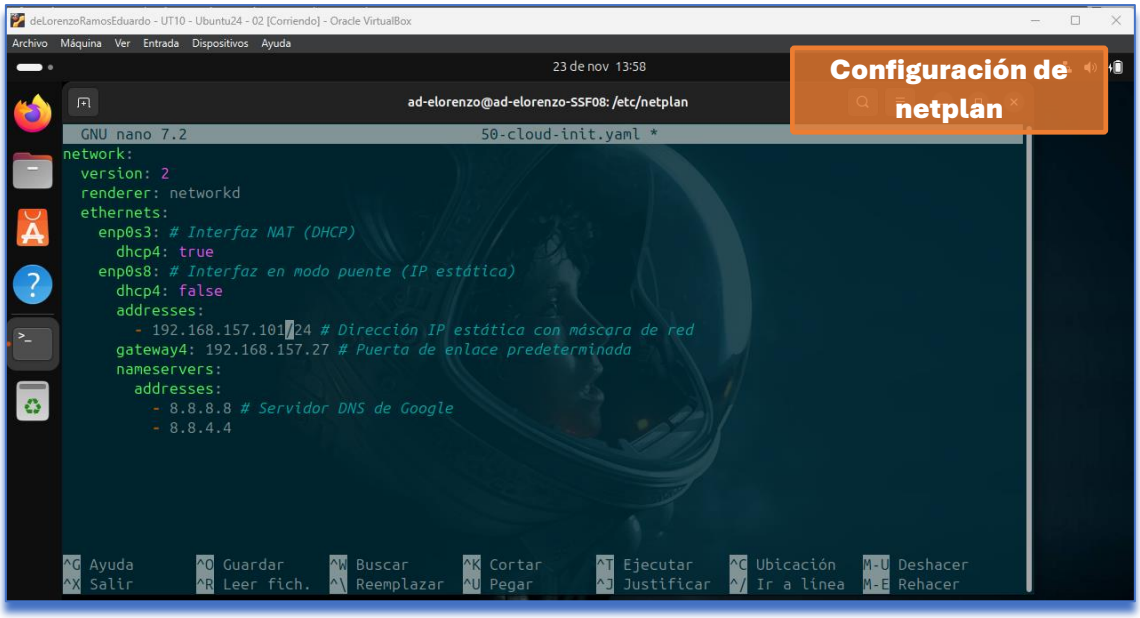


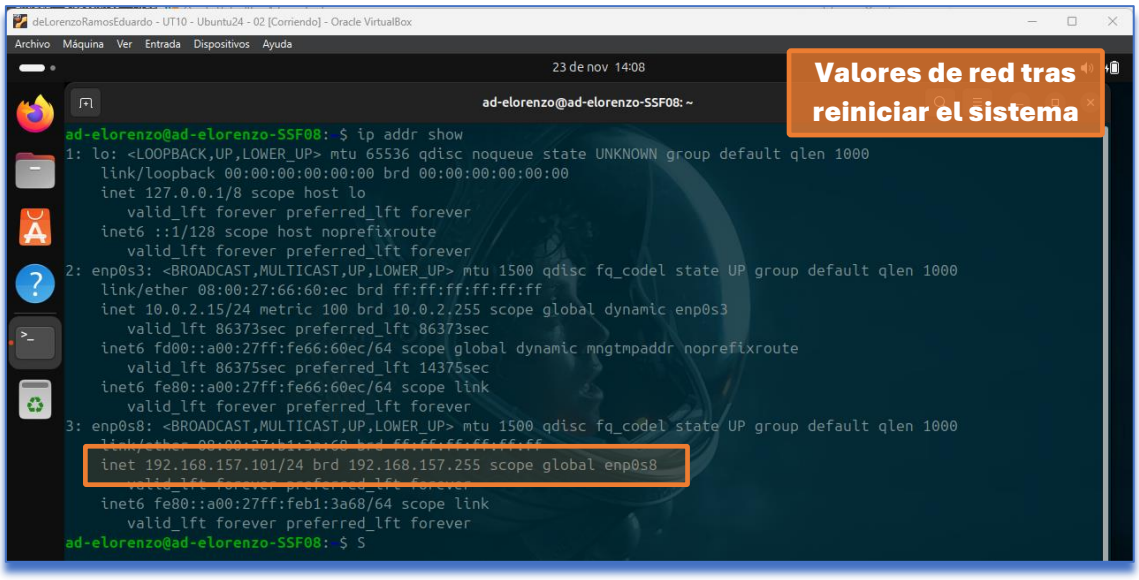
Vemos que alcanza la puerta de enlace y desde ahí recorre los saltos hasta llegar a la IP indicada en la prueba. Por lo que funciona correctamente.

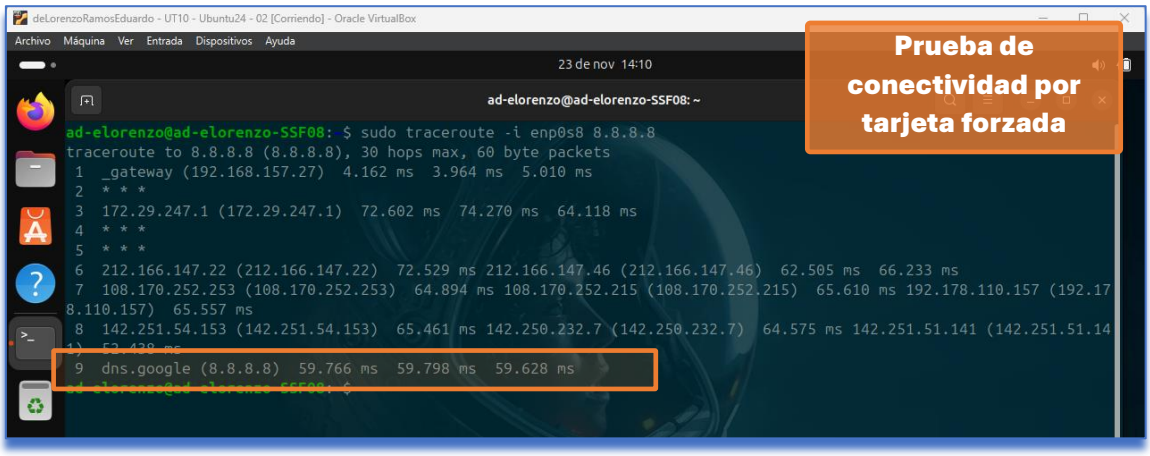
10.1.2. Configurar la red en la máquina con SO Ubuntu 24.04 - 2

10.1.2.1. Repetimos la misma secuencia de acciones para la configuración en la segunda MV de Ubuntu.



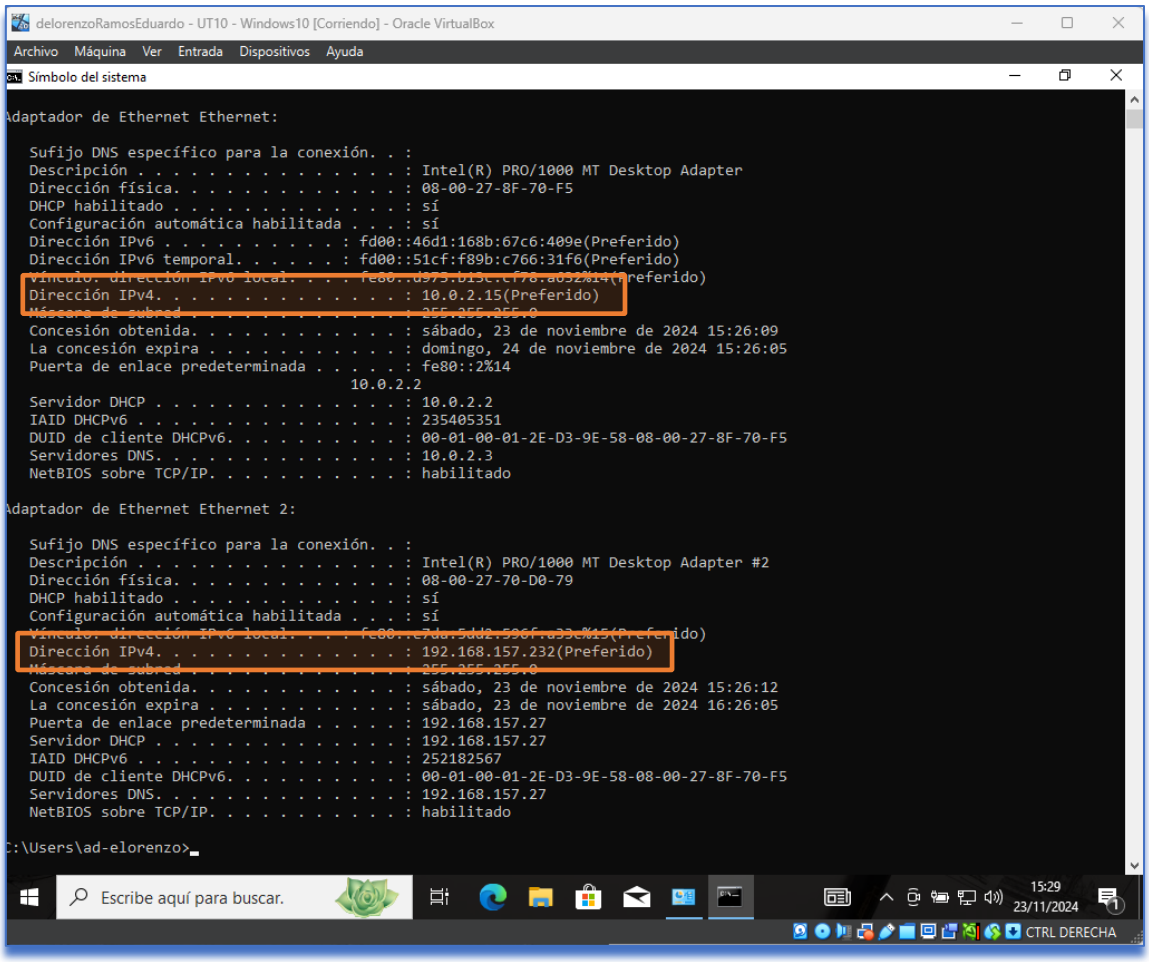

deLorenzoRamosEduardo - UT10 - Ubuntu24 - 02 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
23 de nov 13:58
ad-elorenzo@ad-elorenzo-SSF08: /etc/netplan
Configuración de netplan
GNU nano 7.2 50-cloud-init.yaml *
network:
version: 2
renderer: networkd
ethernets:
enp0s3: # Interfaz NAT (DHCP)
dhcp4: true
enp0s8: # Interfaz en modo puente (IP estática)
dhcp4: false
addresses:
- 192.168.157.101/24 # Dirección IP estática con máscara de red
gateway4: 192.168.157.27 # Puerta de enlace predeterminada
nameservers:
addresses:
- 8.8.8.8 # Servidor DNS de Google
- 8.8.4.4
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^/ Ir a línea M-E Rehacer


deLorenzoRamosEduardo - UT10 - Ubuntu24 - 02 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
23 de nov 14:08
ad-elorenzo@ad-elorenzo-SSF08: ~
Valores de red tras reiniciar el sistema
ad-elorenzo@ad-elorenzo-SSF08: $ ip addr 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 08:00:27:66:60:ec brd ff:ff:ff:ff:ff:ff
inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp0s3
valid_lft 86373sec preferred_lft 86373sec
inet6 fd00::a00:27ff:fe66:60ec/64 scope global dynamic mngtmpaddr noprefixroute
valid_lft 86375sec preferred_lft 14375sec
inet6 fe80::a00:27ff:fe66:60ec/64 scope link
valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
inet 192.168.157.101/24 brd 192.168.157.255 scope global enp0s8
inet6 fe80::a00:27ff:feb1:3a68/64 scope link
valid_lft forever preferred_lft forever
ad-elorenzo@ad-elorenzo-SSF08: $ S

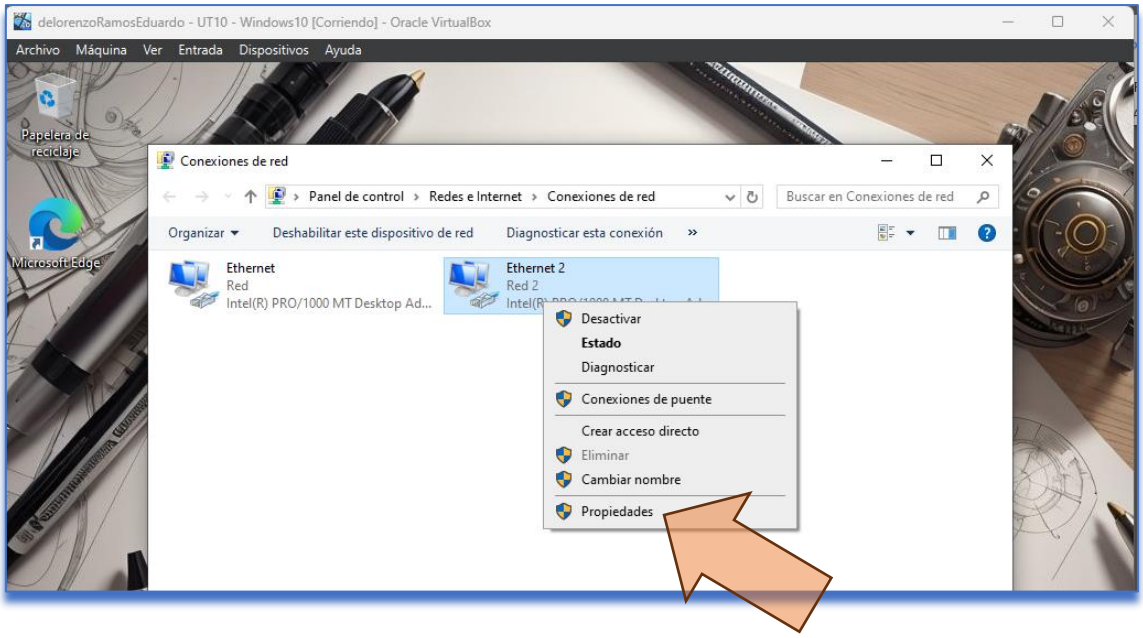

deLorenzoRamosEduardo - UT10 - Ubuntu24 - 02 [Corriendo] - Oracle VirtualBox
Archivo Máquina Ver Entrada Dispositivos Ayuda
23 de nov 14:10
ad-elorenzo@ad-elorenzo-SSF08: ~
Prueba de conectividad por tarjeta forzada
ad-elorenzo@ad-elorenzo-SSF08: $ sudo traceroute -i enp0s8 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 60 byte packets
1 _gateway (192.168.157.27) 4.162 ms 3.964 ms 5.010 ms
2 * * *
3 172.29.247.1 (172.29.247.1) 72.602 ms 74.270 ms 64.118 ms
4 * * *
5 * * *
6 212.166.147.22 (212.166.147.22) 72.529 ms 212.166.147.46 (212.166.147.46) 62.505 ms 66.233 ms
7 108.170.252.253 (108.170.252.253) 64.894 ms 108.170.252.215 (108.170.252.215) 65.610 ms 192.178.110.157 (192.178.110.157) 65.557 ms
8 142.251.54.153 (142.251.54.153) 65.461 ms 142.250.232.7 (142.250.232.7) 64.575 ms 142.251.51.141 (142.251.51.141) 53.438 ms
9 dns.google (8.8.8.8) 59.766 ms 59.798 ms 59.628 ms

10.1.3. Configurar la red en la máquina con SO Windows 10 Pro

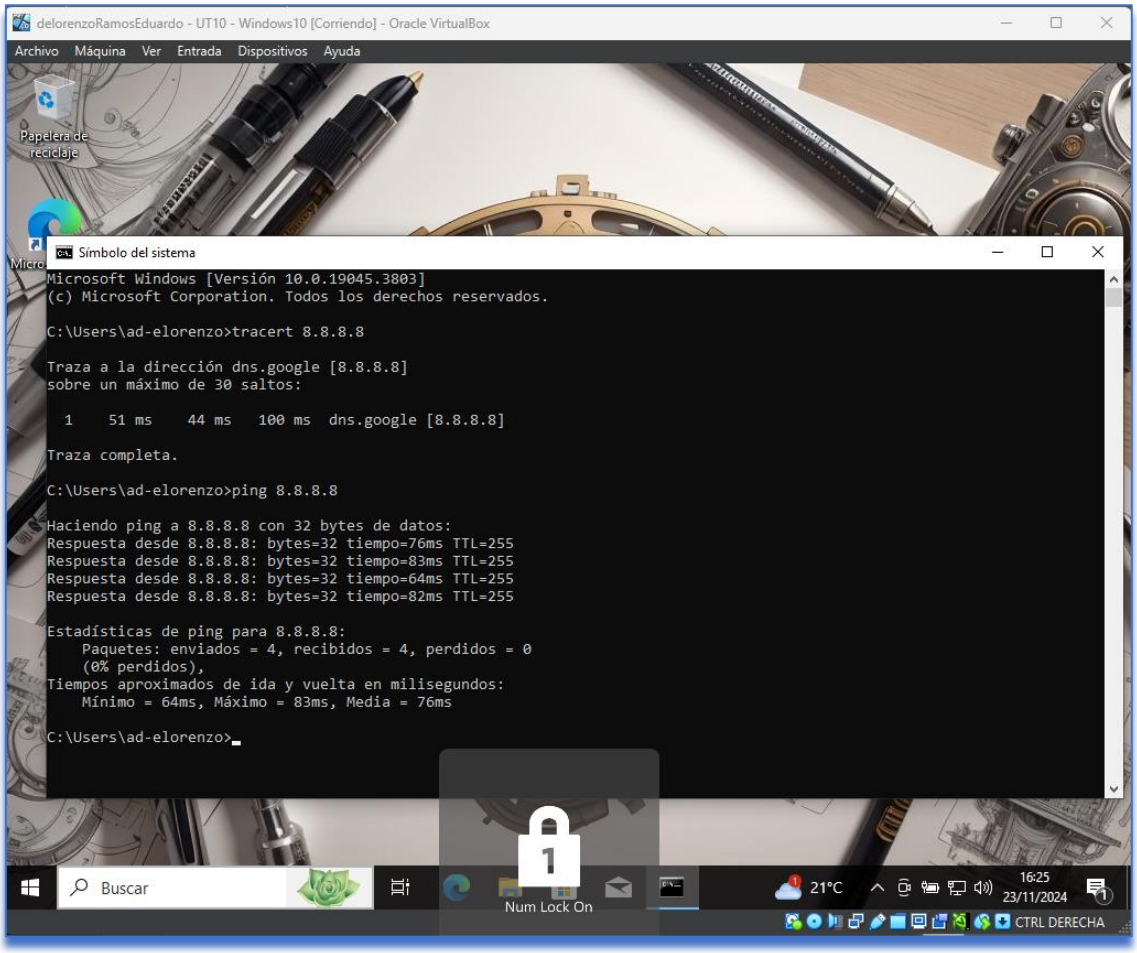
10.1.3.1. En Windows abrimos el CMD e introducimos el comando ipconfig /all para comprobar el estado de las tarjetas habilitadas para la máquina. Comprobamos que las dos están operativas y recibiendo asignación de IP, una por la NAT del host y la otra por el DHCP del router.



10.1.3.2. Para modificar la IP de la tarjeta que está configurada en modo puente (ethernet 2) accedemos al panel de conexiones de red. Sobre su icono desplegamos el menú contextual y elegimos la opción de **Propiedades.**



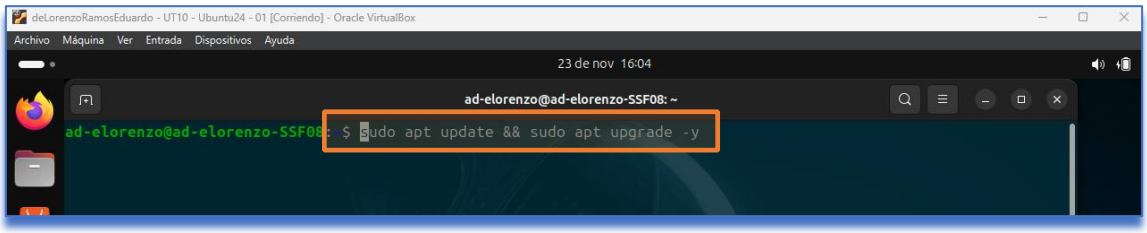
10.1.3.3. Seleccionamos Protocolo de Internet versión 4 (TCP/IPv4) y pulsamos en el botón de **Propiedades.** Tras esta acción se abrirá una nueva ventana en la que introduciremos manualmente la configuración de la tarjeta. Modificamos también las métricas para que sea la tarjeta prioritaria para el sistema. Hacemos pruebas de conectividad y comprobamos que responde correctamente.



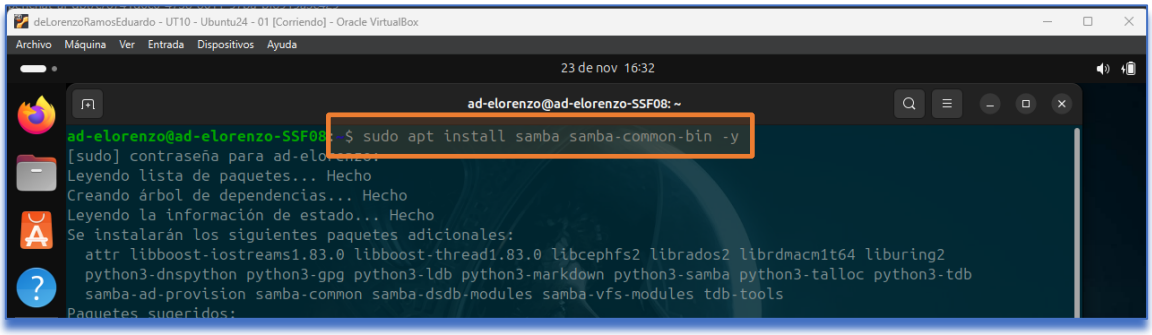
Actividad 10.2 Instalar Samba

10.2.1. Instalación del servidor Samba

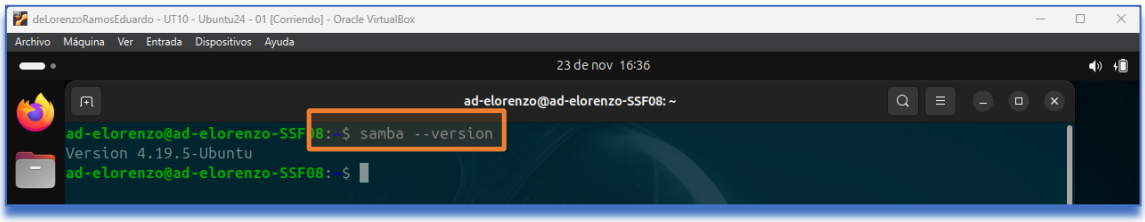
10.2.1.1. En primer lugar, corroboramos que el equipo está actualizado correctamente. Para ello vamos a usar la cadena de comandos **sudo apt update && sudo apt upgrade -y**



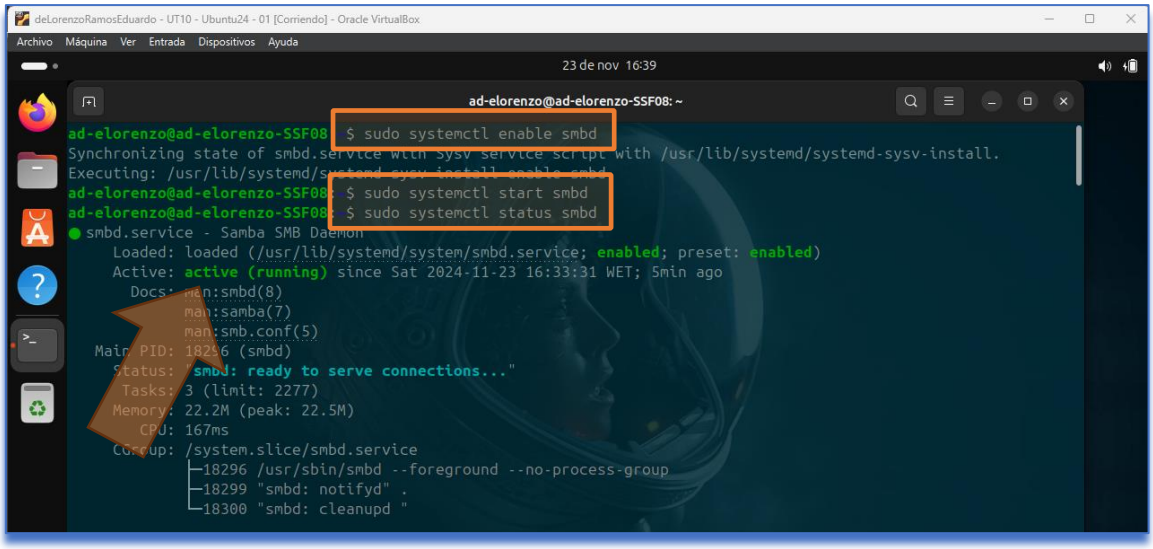
10.2.1.2. Ahora instalaremos el servidor samba y sus herramientas asociadas con el comando **sudo apt install samba samba-common-bin -y**. Esto instalará el servicio Samba y utilidades como smbpasswd para la gestión de usuarios



10.2.1.3. Tras finalizar la instalación comprobamos la versión que está corriendo en la máquina con el comando **samba –versión**, indicándonos el sistema que es la 4.19.5.

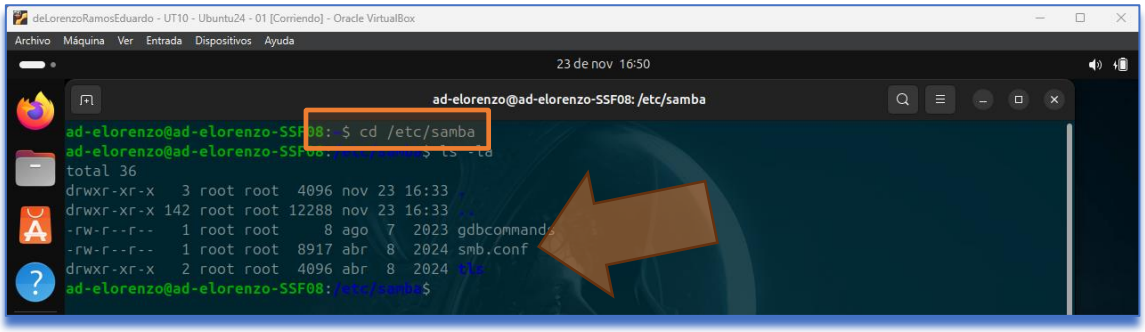


10.2.1.4. Ahora activamos el samba para que se inicie con el sistema con el comando **sudo systemctl enabled samba**. Lo iniciamos con **sudo systemctl start samba** y finalmente comprobamos su estado en el sistema con **sudo systemctl status samba**.

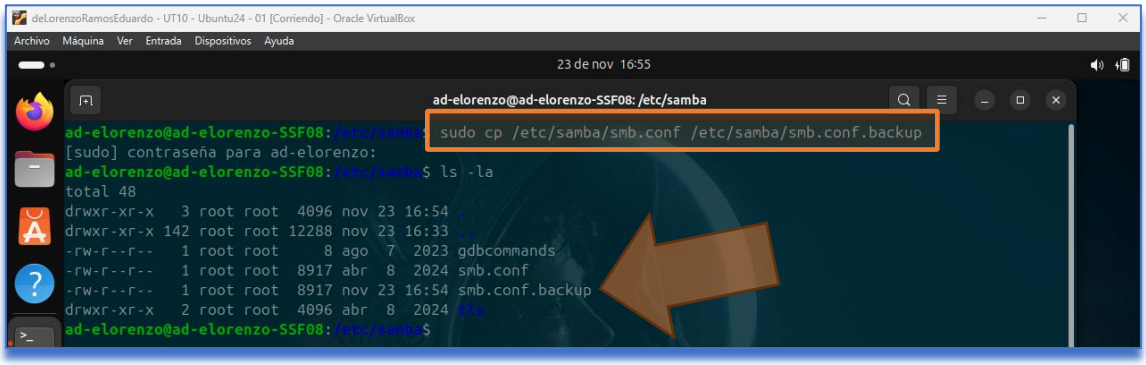


10.2.2. Configuración del servidor Samba

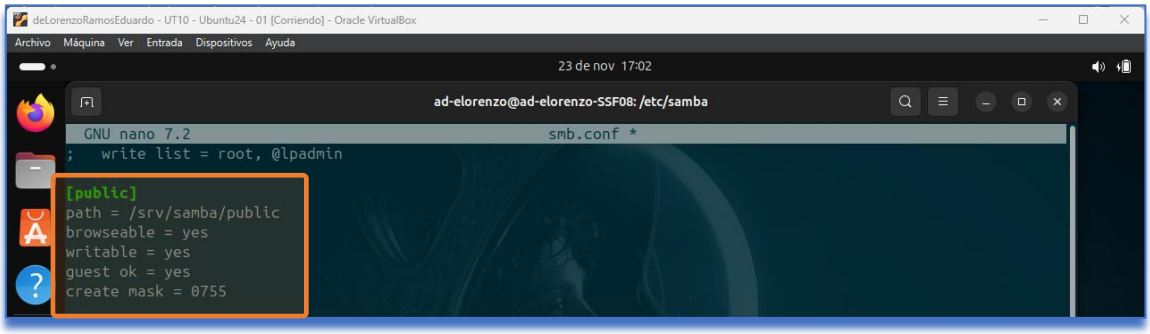
10.2.2.1. Para configurar el Samba lo primero será desplazarnos al archivo que se encuentra en la ruta **/etc/samba/smb.conf**



10.2.2.2. Antes de realizar modificaciones es recomendable hacer una copia de seguridad por si se produce alguna incidencia durante la configuración. Para ello usaremos el comando **sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.backup.** Listamos con **ls -la** y comprobamos que se ha creado correctamente.



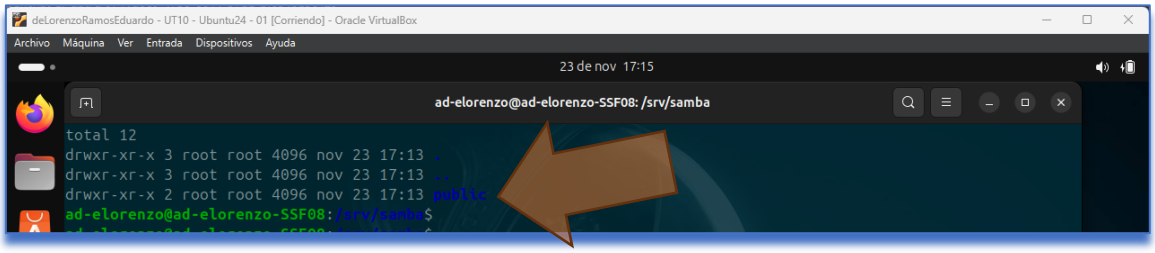
10.2.2.3. Procedemos a acceder al archivo .conf para su edición. Para ello usamos el comando **sudo nano /etc/samba/smb.conf**. Una vez abierto bajamos al final de su contenido e insertamos nuevos valores de configuración.



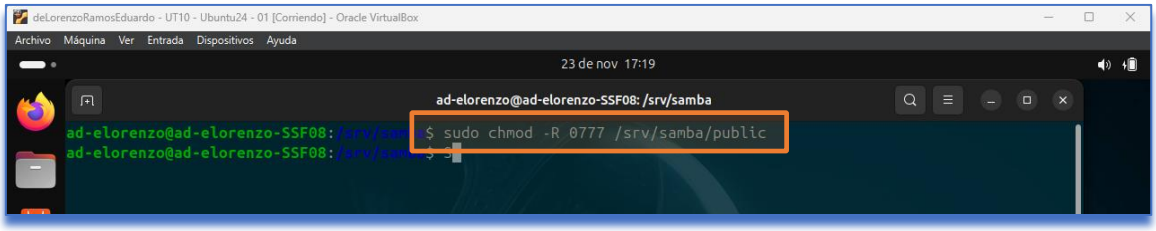
Estos son los valores escogidos.

path	Ruta al directorio compartido en el sistema Linux.
browseable	Permite que el recurso sea visible en la red
writable	Habilita escritura en el recurso compartido.
guest ok	Permite acceso sin autenticación para usuarios invitados.
create mask	Define los permisos predeterminados para los archivos creados

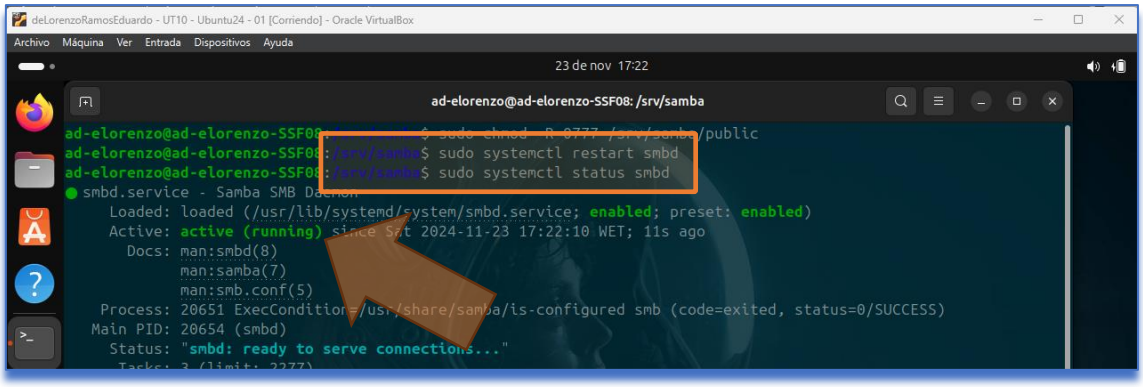
10.2.2.4. Ahora es necesario crear el directorio que hemos cargado como compartido con el comando **sudo mkdir -p /srv/samba/public.** Listamos y comprobamos que se ha creado correctamente.



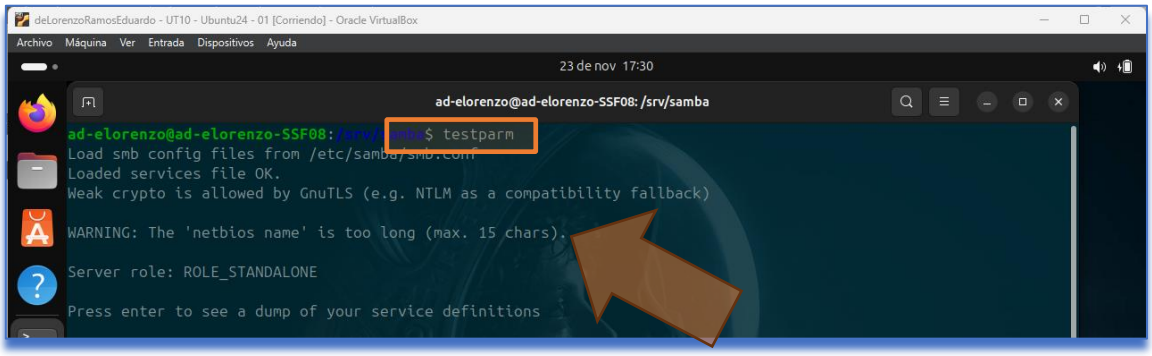
10.2.2.5. Modificamos los permisos del directorio que hemos creado para que los usuarios puedan trabajar en él sin restricciones con el comando **sudo chmod -R 0777 /srv/samba/public**



10.2.2.6. Reiniciamos el servicio para que para que aplique todos los cambios que hemos realizado. Para ello usaremos el comando **sudo systemctl restart smbd.** Luego, con **sudo systemctl status smbd** comprobamos que está corriendo con normalidad.

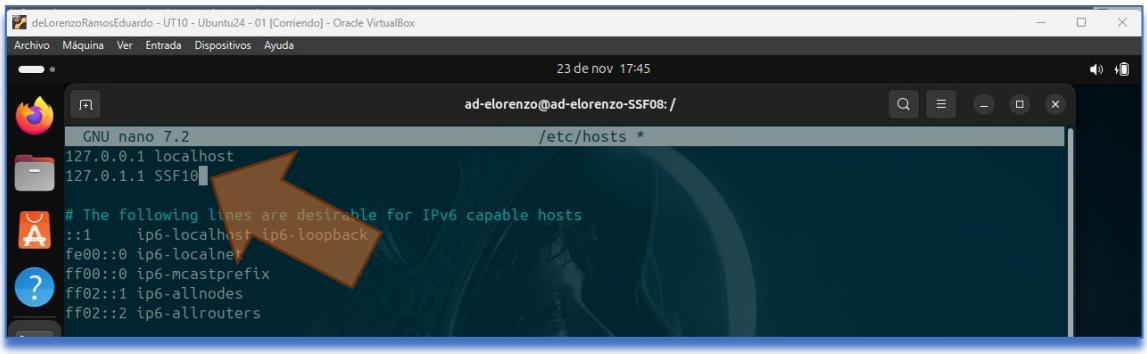


10.2.2.7. Verificamos la configuración con el comando **testparm** y vemos que nos devuelve un mensaje de aviso indicando que el netbios name es demasiado largo. Por lo que será necesario cambiarlo.

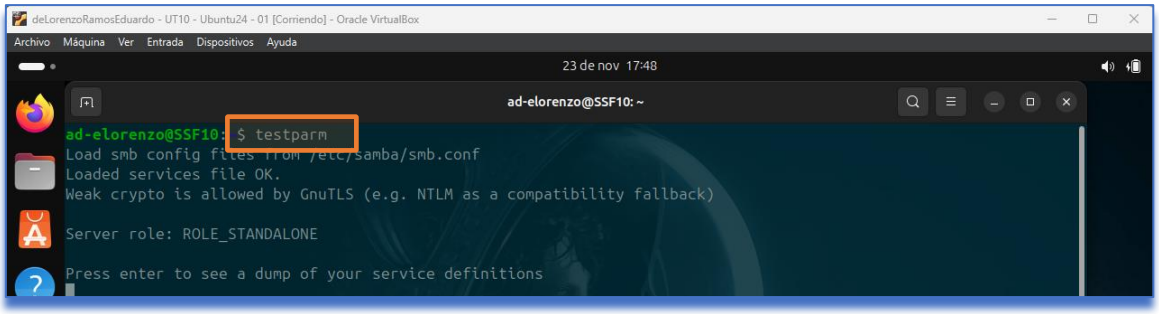


10.2.2.8. Comprobamos con el comando **hostnamectl** el que actualmente figura en el sistema.

10.2.2.9. Para cambiarlo usamos el comando **sudo nano /etc/hosts**, realizamos la modificación y reiniciamos la máquina para que haga efectivos los cambios.



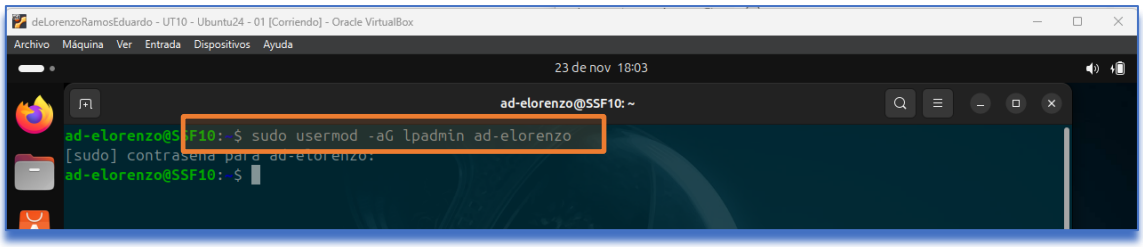
10.2.2.10. Tras ese paso probamos de nuevo con **testparm** y comprobamos que ya no muestra el mensaje de aviso.



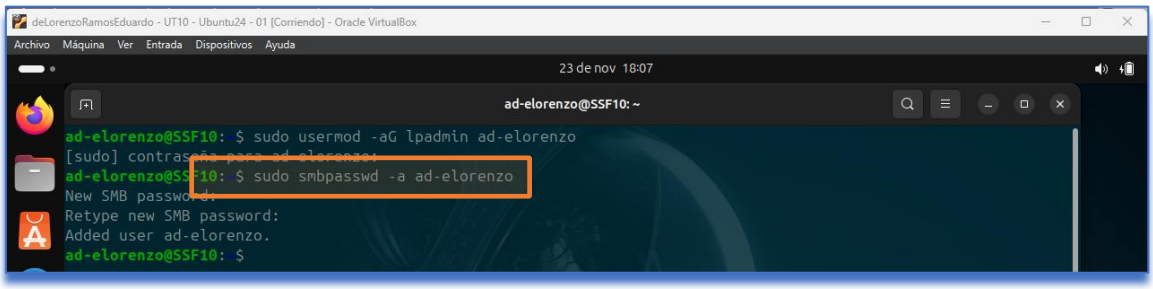
El término **ROLE_STANDALONE** en Samba indica que el servidor está configurado como un servidor autónomo, lo que significa que no forma parte de un dominio de Windows ni actúa como un controlador de dominio. Es decir, el servidor Samba funciona de forma independiente, manejando la autenticación y las solicitudes de acceso a los recursos compartidos localmente, sin depender de una infraestructura externa como un Active Directory.

10.2.3. Creación y concesión de permisos

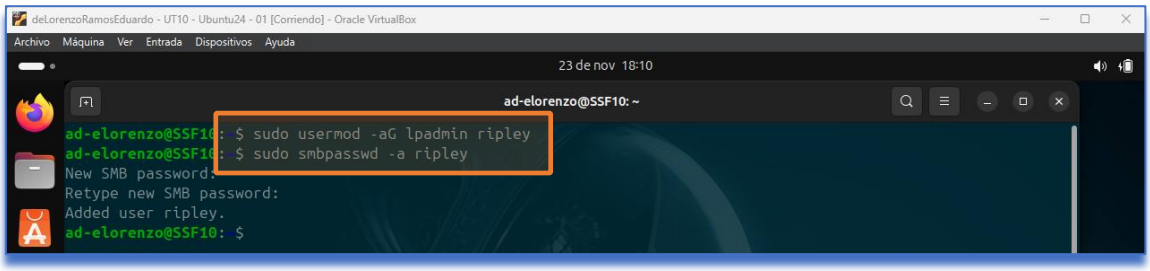
10.2.3.1. El primer paso será darle permiso de acceso al Samba al usuario que ya tenemos creado en el sistema **ad-elorenzo**. Para ello vamos a usar el comando **sudo usermod -aG lpadmin [nombreDelUsuario]**



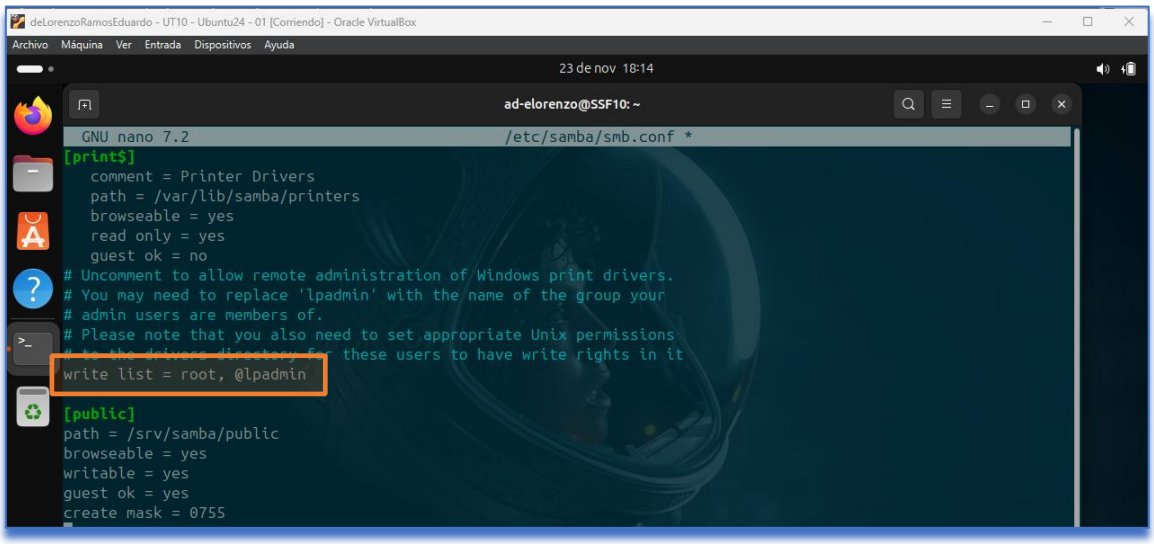
10.2.3.2. Le generamos también una contraseña específica para Samba al usuario con el comando **sudo smbpasswd -a ad-elorenzo.**



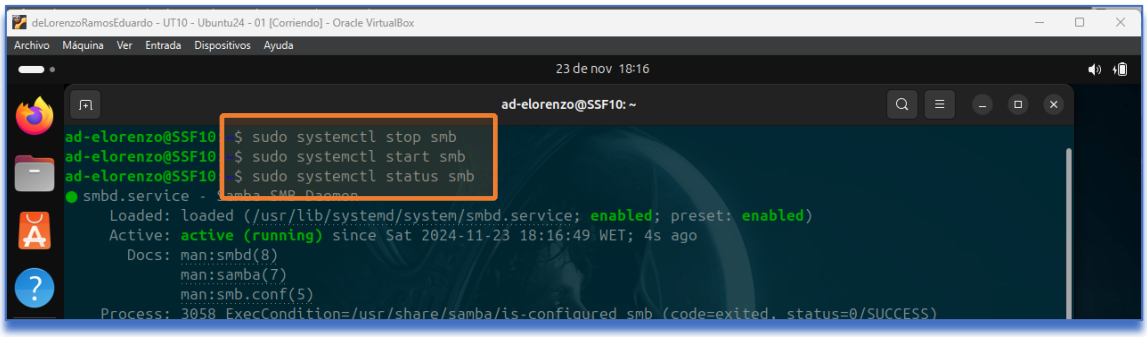
10.2.3.3. Repetimos el proceso para otro usuario, para hacer el acceso desde un user distinto al que está logado en el sistema desde el que intenta el acceso a la carpeta.



10.2.3.4. Accedemos de nuevo al archivo de configuración con **sudo nano /etc/samba/smb.conf** y activamos la línea de **write list = root, @lpadmin**.

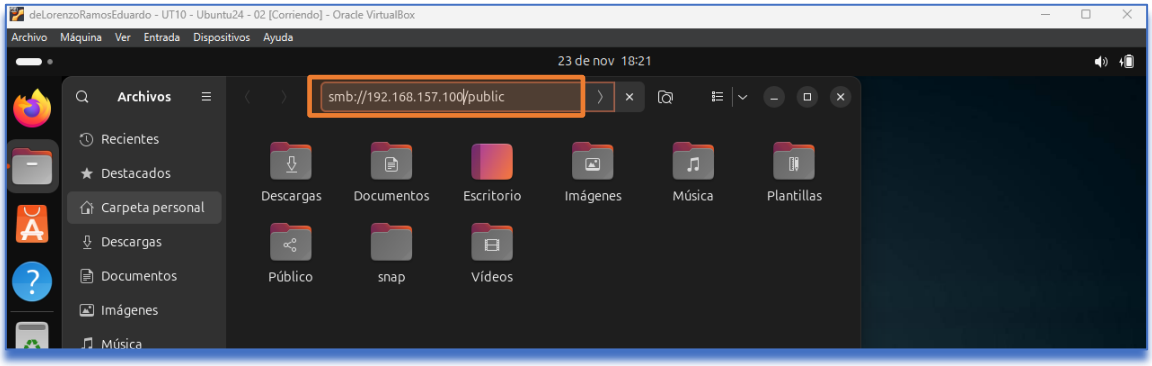


10.2.3.5. Tras grabar los cambios paramos el servicio con **sudo systemctl stop smb,** lo volvemos a arrancar con **sudo systemctl start smb**, y comprobamos que corre sin incidencias con **sudo systemctl status smb.**

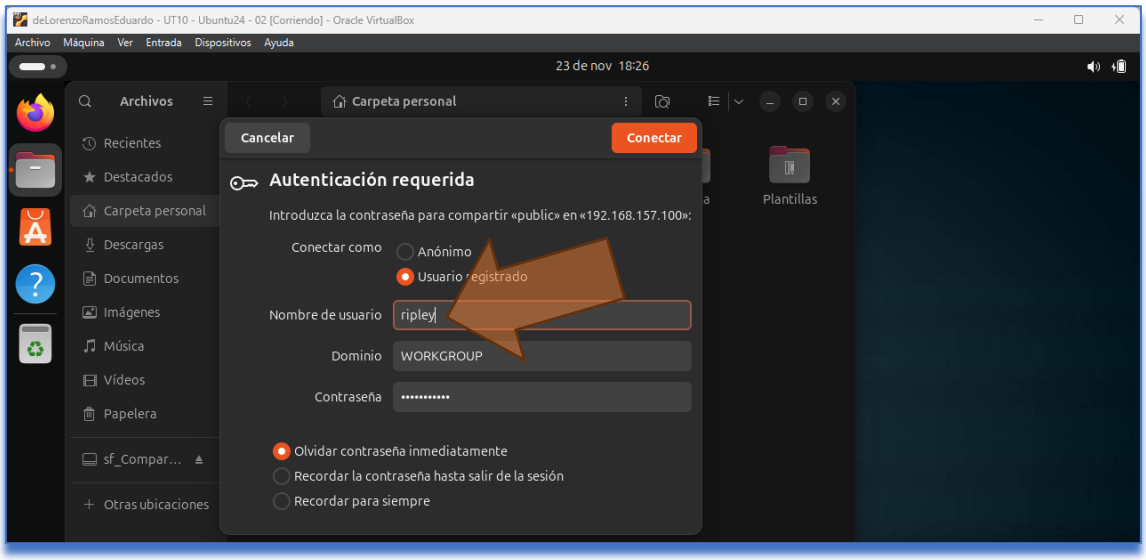


10.2.4. Acceso desde Ubuntu al equipo servidor

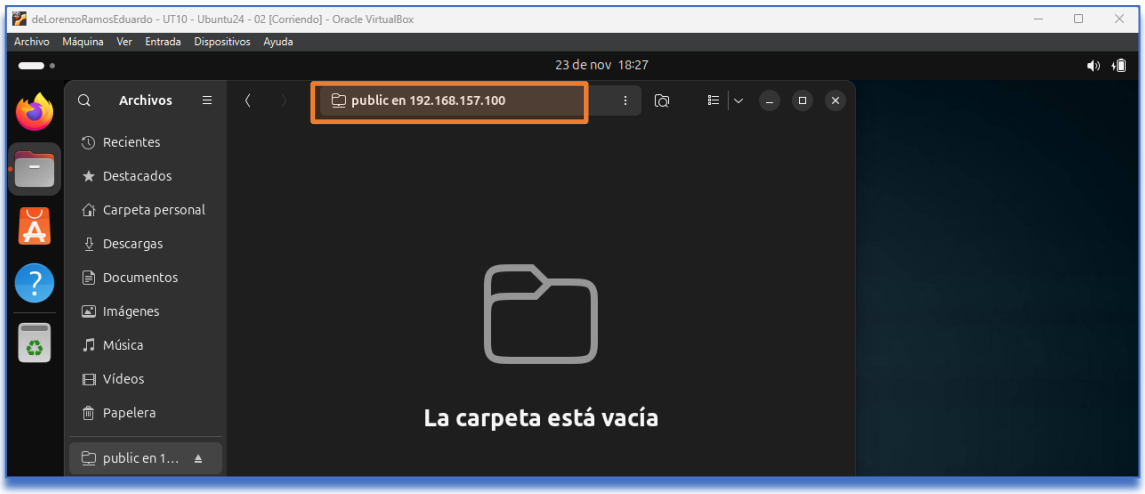
10.2.4.1. Accedemos a la segunda máquina de Ubuntu y abrimos un explorador de archivos. Introducimos **smb://<IP_del_servidor>/public** que, para el caso de la práctica va a ser, **smb://192.168.157.100/public**



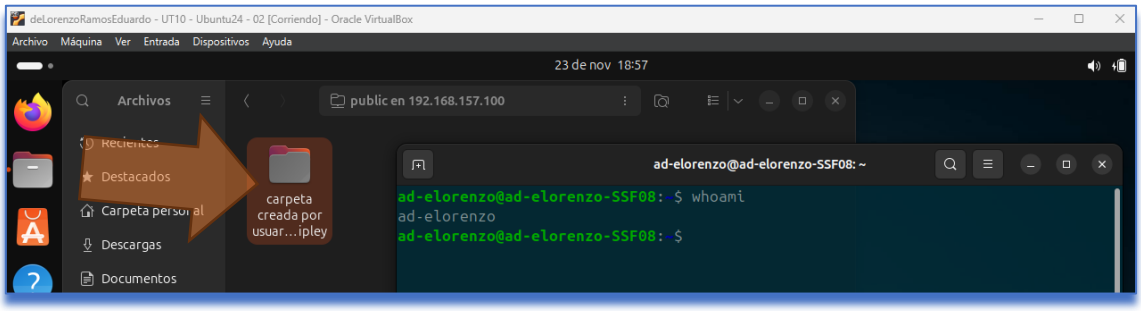
10.2.4.2. El sistema lanzará una ventana para que seleccionemos la tipología de acceso. Para comprobar el funcionamiento de los usuarios creados probamos con el de **ripley** ya que el del sistema es **ad-elorenzo.**



10.2.4.3. Accedemos sin problemas con el usuario tras el login**.**

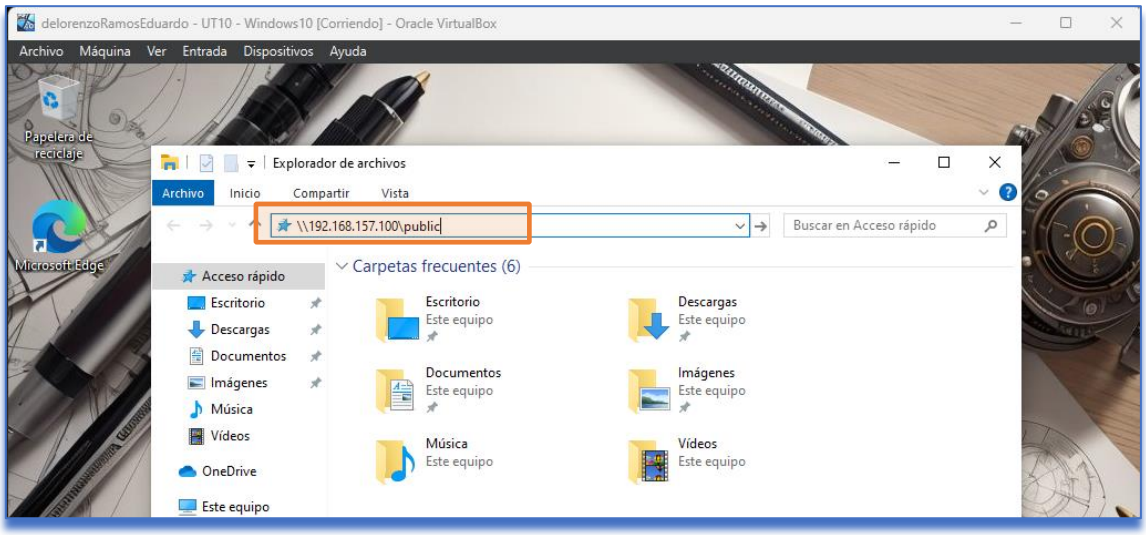


10.2.4.4. Creamos una carpeta con el usuario para comprobar el acceso desde el equipo con SO Windows.

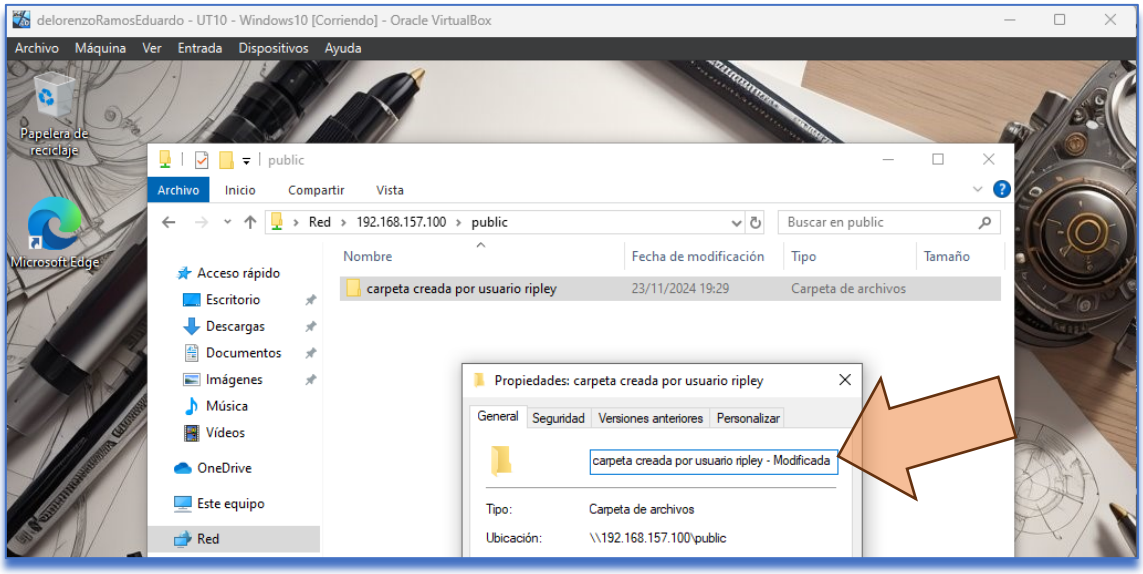


10.2.5. Acceso desde Windows al equipo servidor

10.2.5.1. Desde Windows abrimos en el explorador de archivos **\\<IP_del_servidor>\public** que para el caso de la práctica va a ser **\\192.168.157.100\public**



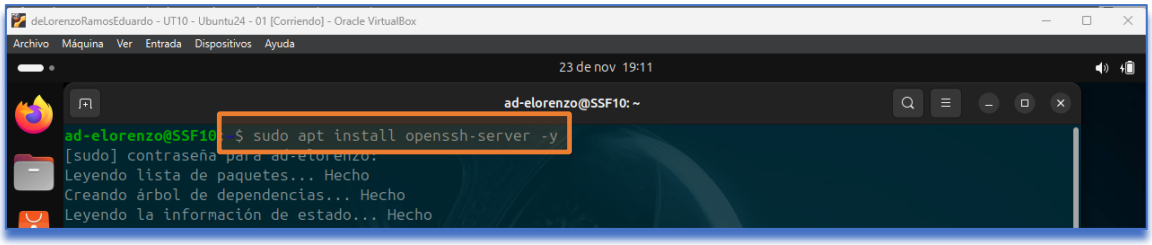
10.2.5.2. Accedemos a la carpeta pública en el servidor y vemos la carpeta que antes creamos con el usuario ripley. Podemos cambiarla y modificar su nombre.



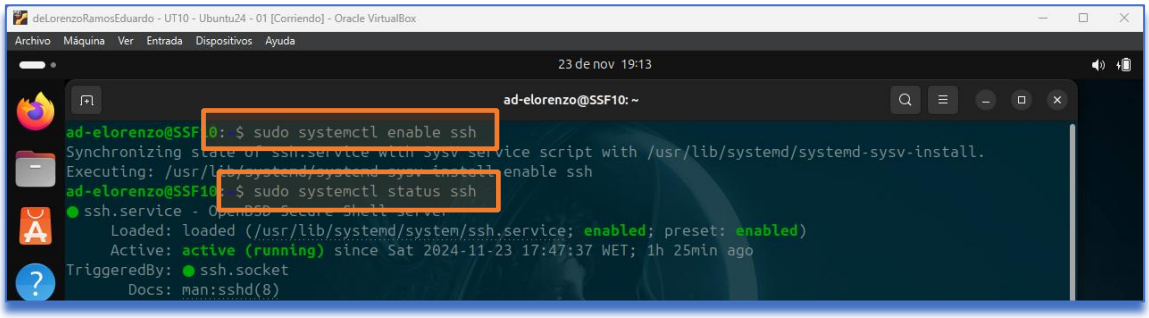
Actividad 10.3 Servicio SSH

10.3.1. Instalar servicio ssh

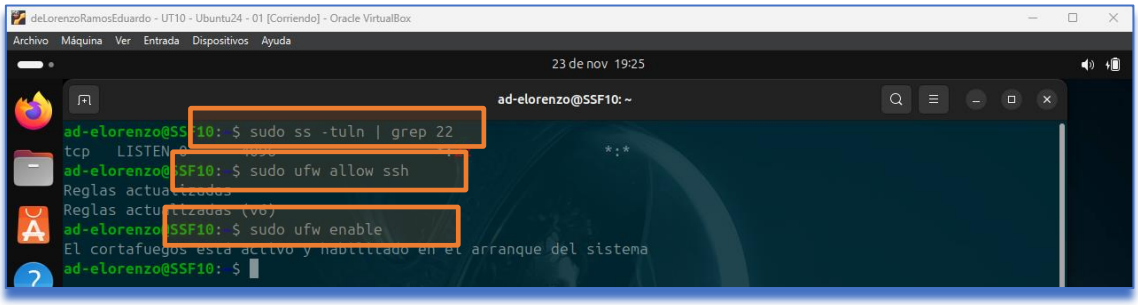
10.3.1.1. Desde el equipo Ubuntu-01 abrimos el terminal y usamos el comando **sudo apt install openssh-server -y** para instalar el servidor de ssh.



10.3.1.2. Ahora lo activamos para que se inicie con el arranque del sistema con el comando **sudo systemctl enable ssh** y comprobamos que está corriendo con normalidad con **sudo systemctl status ssh.**

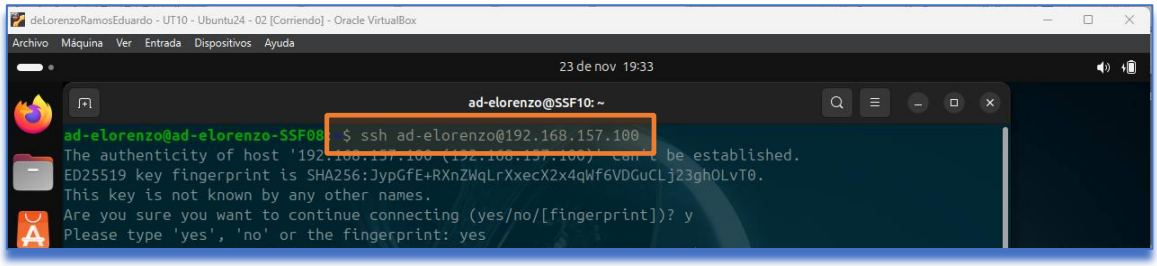


10.3.1.3. Ahora pasamos a comprobar que el puerto estándar del SSH esté levantado con el comando **sudo ss -tuln | grep 22** y también que tenga la excepción en el firewall con **sudo ufw allow ssh** para, finalmente, activarlo con **sudo ufw enable**

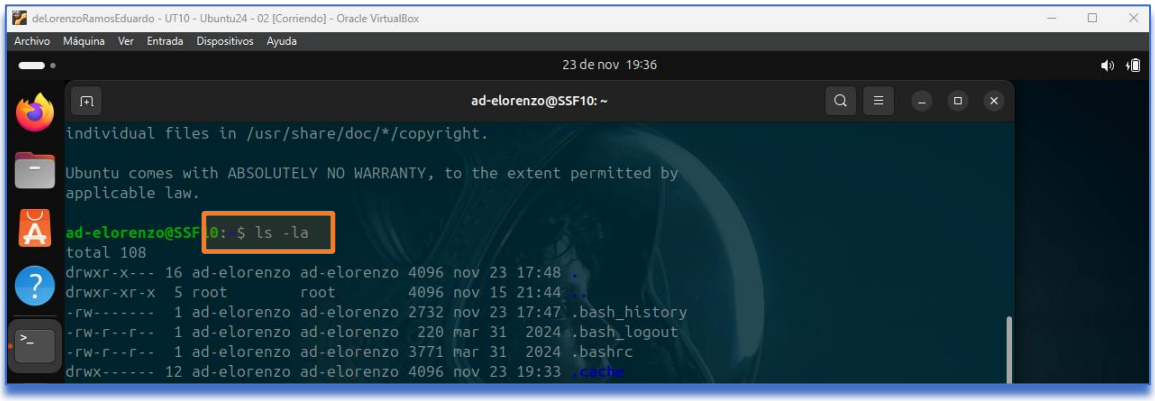


10.3.2. Conectar desde clienteLinux y ejecutar algunos comandos

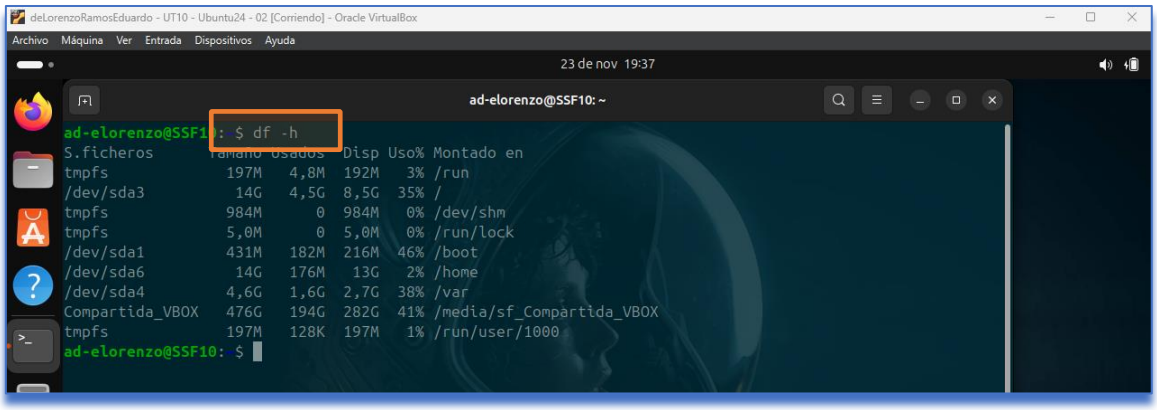
10.3.2.1. Probamos ahora la conexión por SSH desde la máquina 2 de Ubuntu. Para ello abrimos un terminal y usamos el comando **ssh usuario@IP_del_servidor** que, para el caso de la práctica, será **ssh ad-elorenzo@192.168.157.100.** Al no tener certificaciones que acrediten su autenticidad la aplicación nos avisa de ello al establecer la conexión, pero es normal.



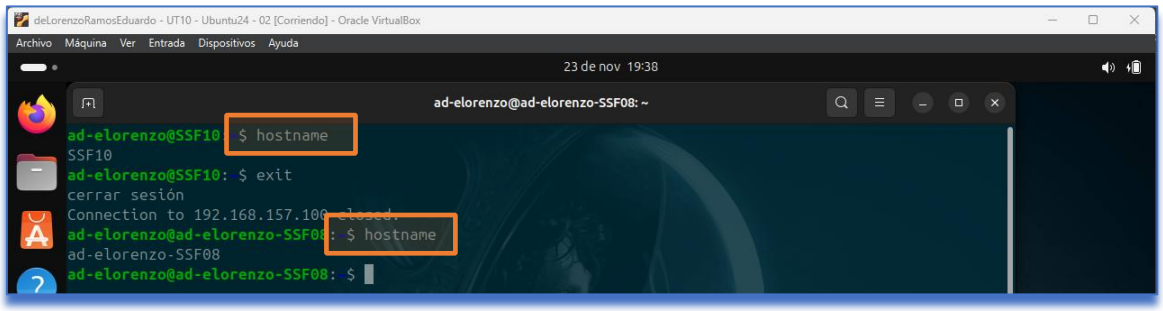
10.3.2.2. Conectados por ssh a la máquina en 192.168.157.100 usamos el comando **ls -la** y vemos que nos lista correctamente el directorio.



10.3.2.3. Usamos también el comando **df -h** para que nos verifique el espacio en el disco.

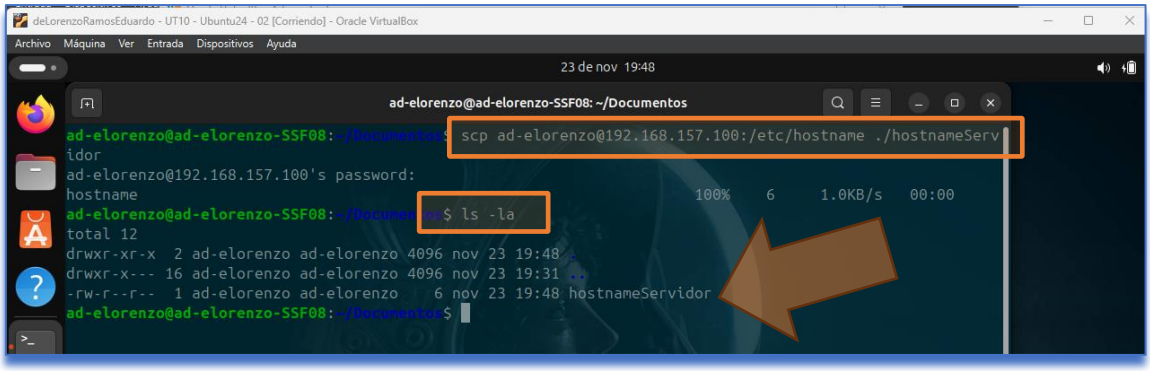


10.3.2.4. Probamos también con el comando **hostname**. Salimos de la sesión con el comando **exit** y lo volvemos a usar **hostname** en la máquina 2 para ver la diferencia entre ambas



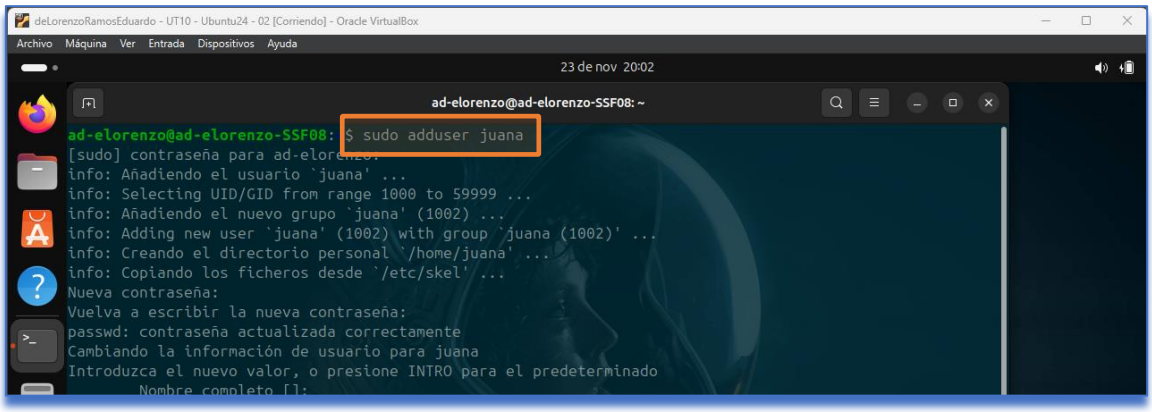
10.3.3. Copiar el archivo /etc/hostname del servidor a la máquina cliente con el nombre hostnameServidor

10.3.3.1. Para hacer la copia usamos el comando **scp usuario@IP_del_servidor:/etc/hostname ./hostnameServidor** que, para el caso de la práctica será **scp ad-elorenzo@192.168.157.100:/etc/hostname ./hostnameServidor**. Una vez nos informa el sistema de la finalización del proceso de copia listamos el directorio en el que nos encontramos en el cliente y vemos que el archivo figura correctamente.

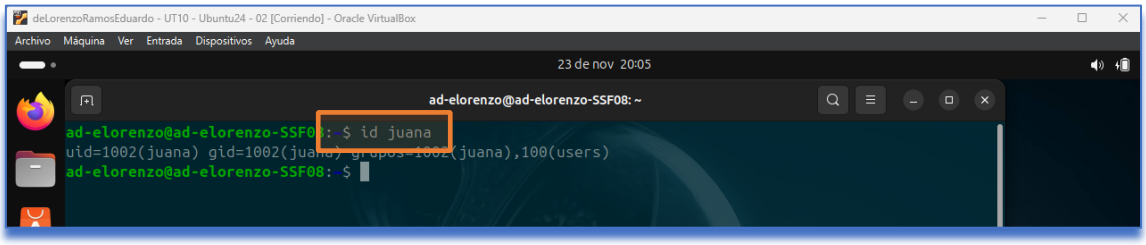


10.3.4. Copiar el directorio /home/juana del cliente Linux al servidor como el usuario luis

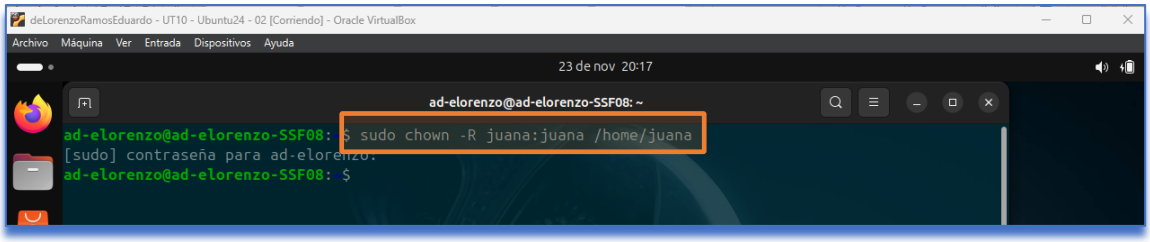
10.3.4.1. El primer paso es crear el directorio juana en el equipo cliente. Para eso usamos el comando **sudo adduser Juana** y le asignamos la contraseña.



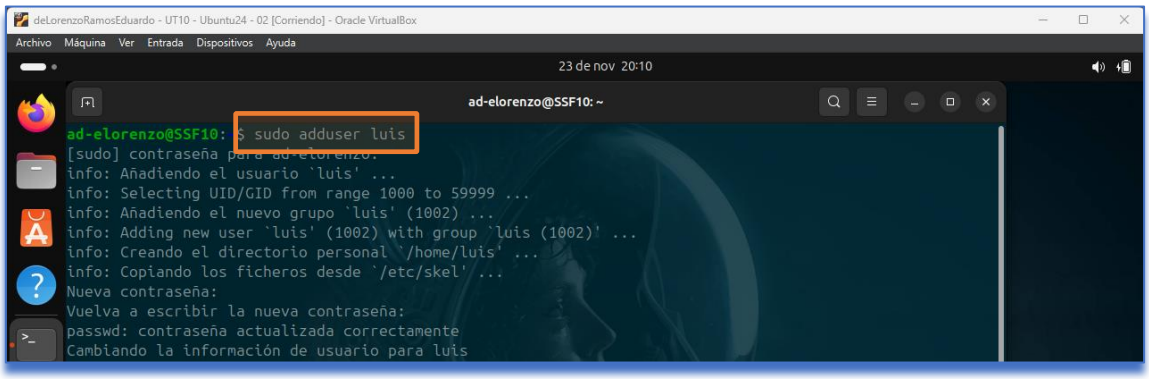
10.3.4.2. Comprobamos con el comando **id juana** que el usuario está correctamente creado en la máquina cliente



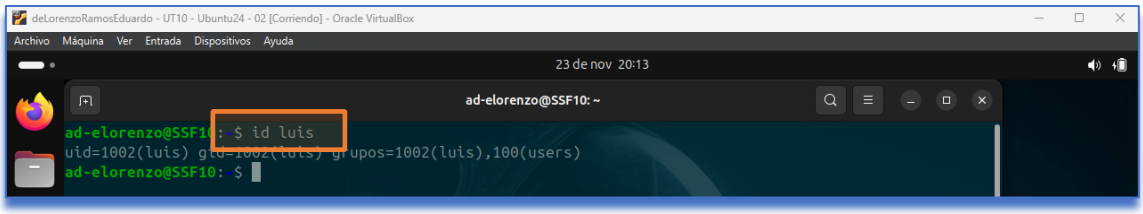
10.3.4.3. Actualizamos los permisos del nuevo usuario con el comando **sudo chown -R juana:juana /home/juana**



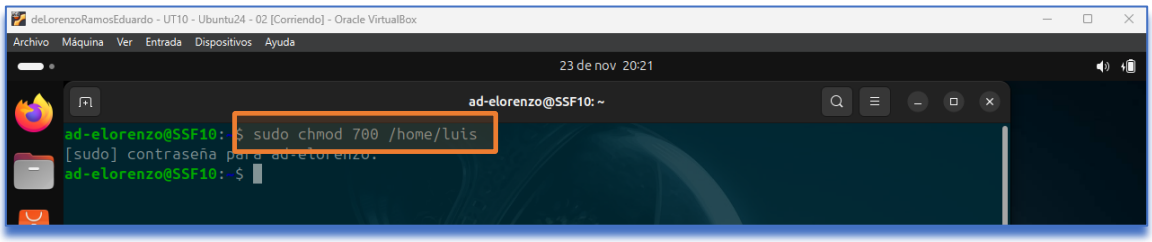
10.3.4.3. Accedemos por ssh al servidor y creamos al usuario Luis con **sudo adduser Luis** y le asignamos una contraseña.



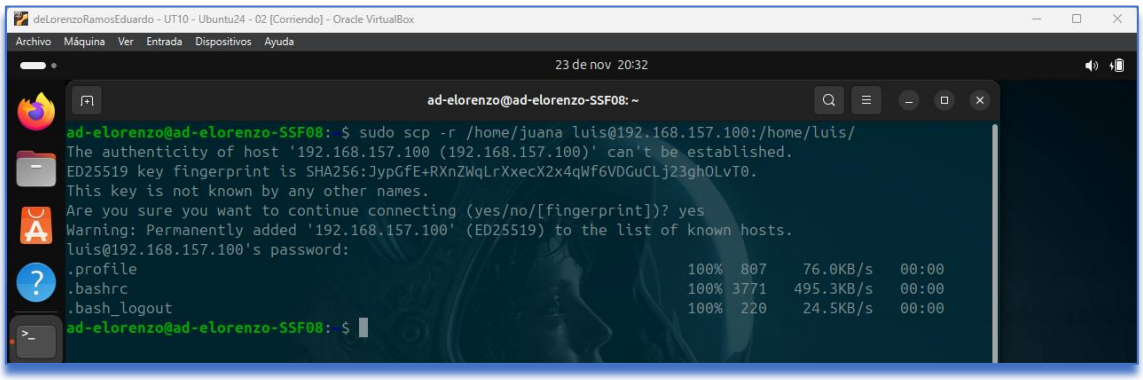
10.3.4.4. Comprobamos con el comando **id luis** que el usuario está correctamente creado en la máquina del servidor.



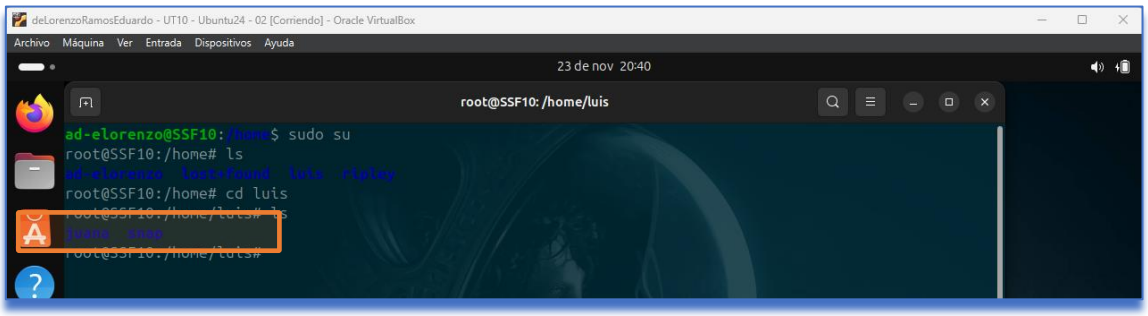
10.3.4.5. Actualizamos los permisos del usuario luis con el comando **sudo chmod 700 /home/Luis**



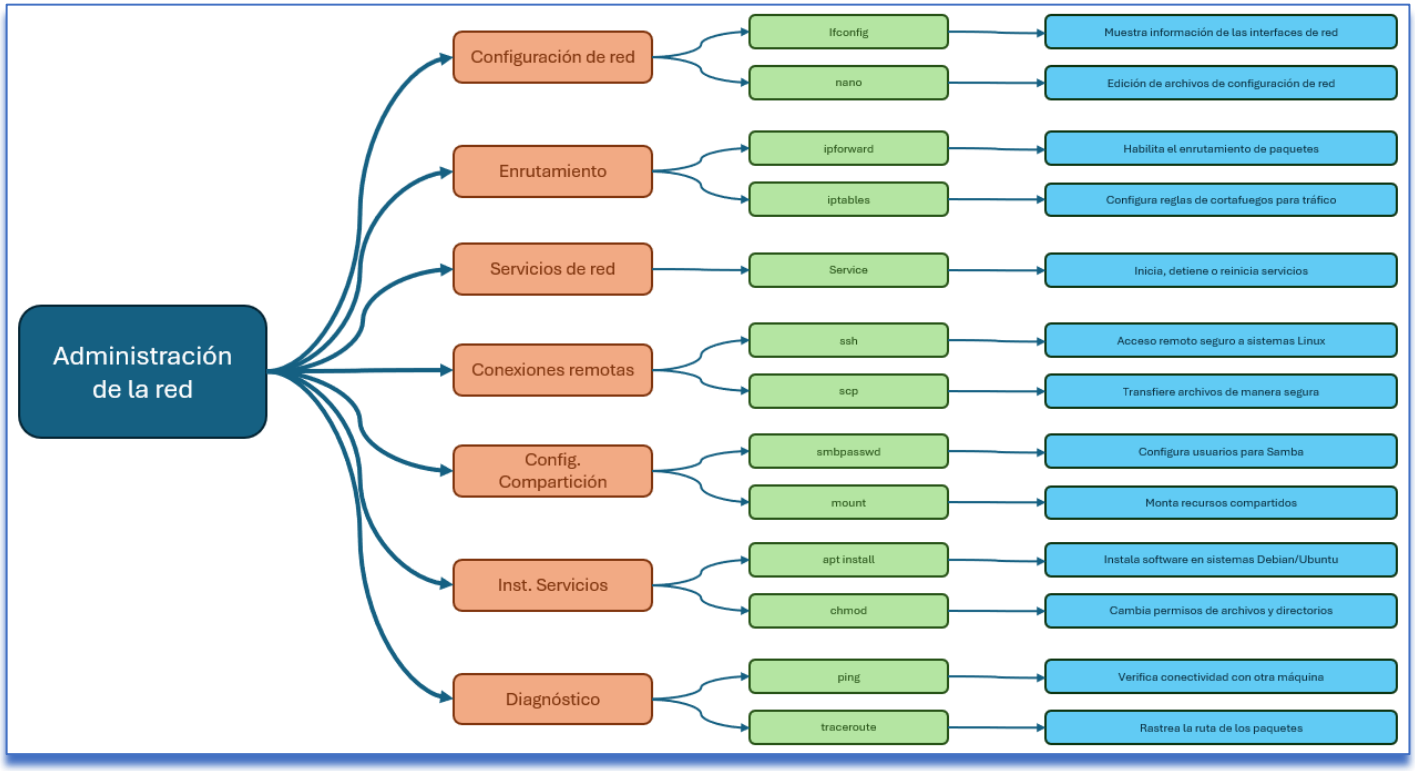
10.3.4.6. Ahora desde el cliente usamos el comando **sudo scp -r /home/juana luis@IP_servidor:/home/luis/** para la copia de las carpetas. Para el caso de la práctica será **sudo scp -r /home/juana ad-elorenzo@192.168.157.100:/home/luis/** Comprobamos que se realiza correctamente la copia de los archivos.



10.3.4.7. Nos conectamos por ssh desde el cliente y comprobamos, listando con el comando **ls**, que dentro del usuario luis se encuentra la carpeta de juana.



Actividad 10.4 Realizar un mapa conceptual



Bibliografía/Webgrafía