Dibujo de una persona

Descripción generada automáticamente con confianza baja

***Enrutamiento de 2 maquinas virtuales***



****

**Achraf El mahjouby Sourour**

# ***Introducción***

Vamos a realizar una práctica, en la cual tenemos dos maquinas virtuales una que hace de servidor y otra es un clientes hemos escogido la versión de Linux Ubuntu en las 2.

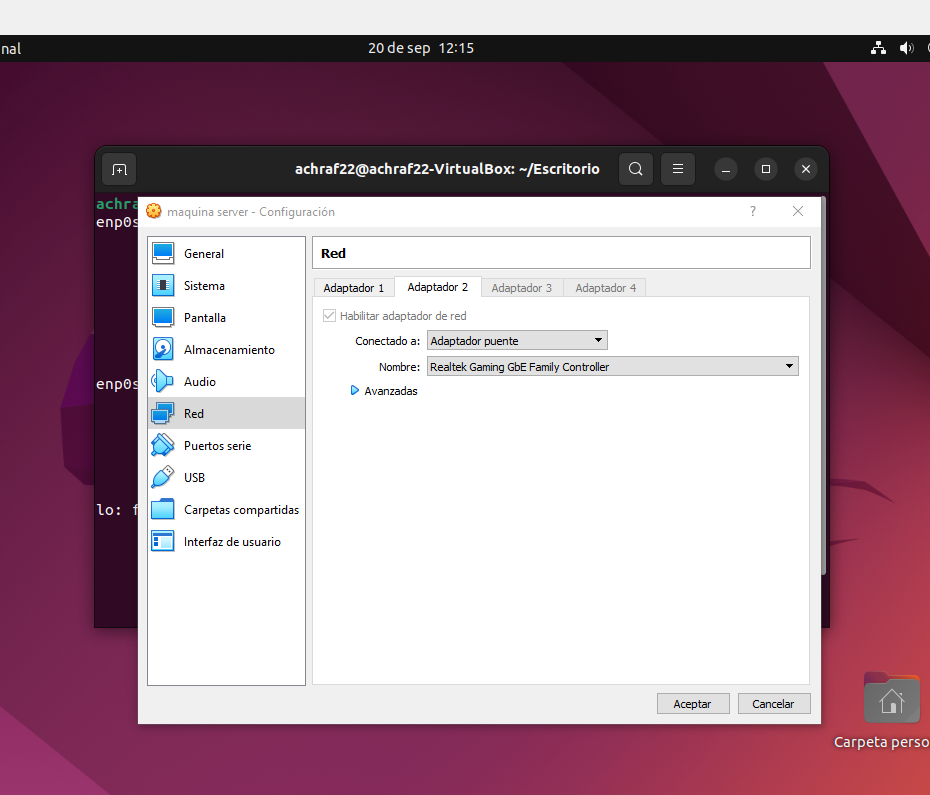
Objetivo de la práctica

El objetivo de la practica es conseguir que la maquina cliente se conecte con la maquina servidor, es decir que se pueda reenviar paquetes y y además que la maquina cliente tenga acceso a la red local mediante el servidor.

# ***Pasos a seguir***

# ***Paso 1***

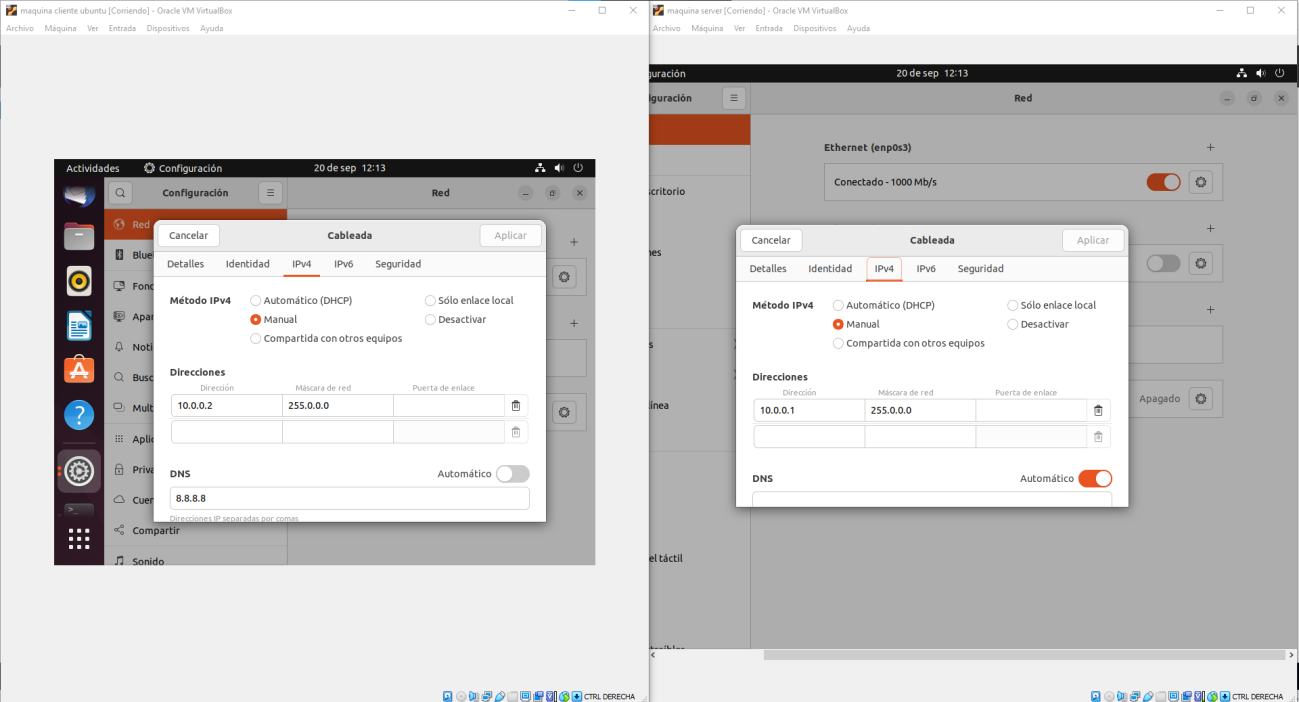
Lo primero que hemos hecho es crearnos las dos maquinas virtuales una cliente y una servidora, en el adaptador de red de la maquina clienta le pones el adaptar en modo de red interna y en la maquina cliente le ponemos dos adaptadores el 1º que sea red interna para que sea posible reenvío de paquetes, entre ellas 2, es decir que se puedan “ver” dicho de una forma más vulgar y el otro adaptador en modo “bridge” o puente que es el que hará como bien dice su nombre de puenteo de conexión entre la maquina clienta y la red local.



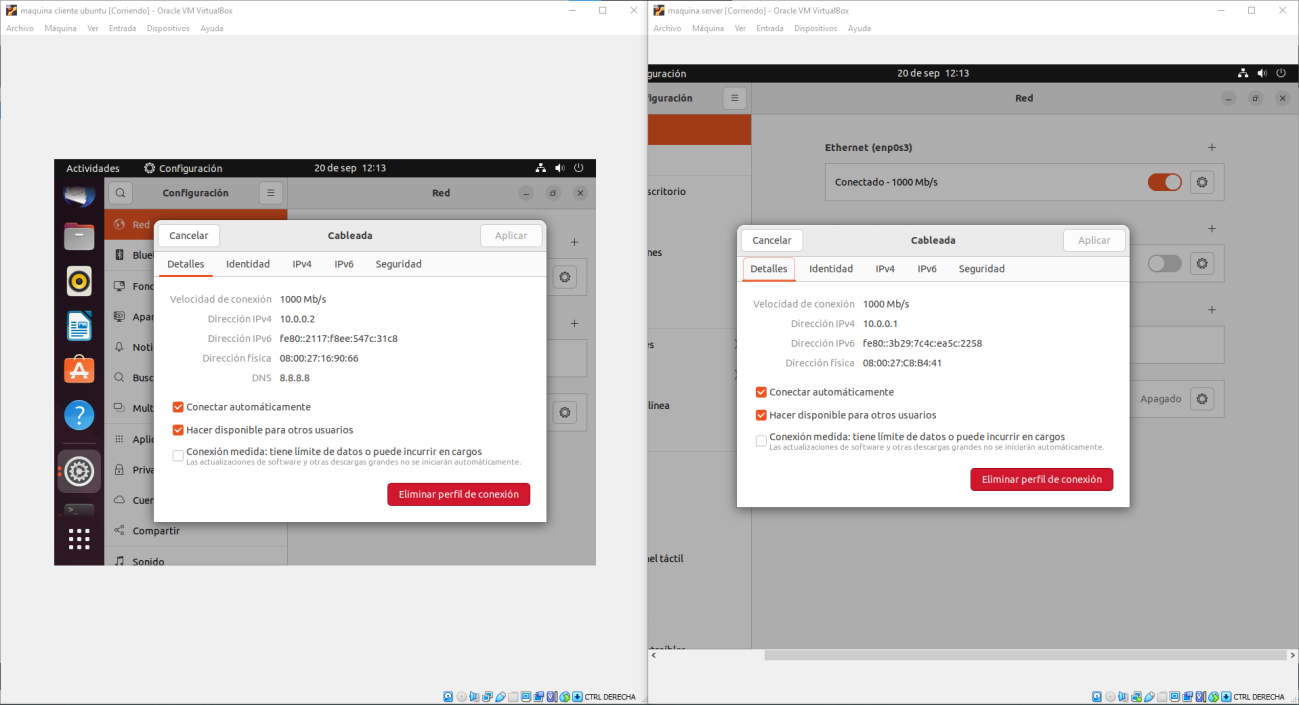
#### **Paso 2**

Después de eso iniciamos las dos maquinas a la vez, para ir manejándolas a la vez ahora lo que haríamos seria irnos a la configuración de internet de las maquinas, y asignarles ips de manera manual en el adaptador de red interna y en este caso hemos elegido la red 10.0.0.0 y con su correspondiente mascara de subred como se muestra en la siguiente imagen.

Además en la maquina cliente tenemos que asignarle la ip 8.8.8.8 es el servidor DNS público de google, para que no nos genere direcciones de DNS automáticamente y siempre nos los redirija al de google.



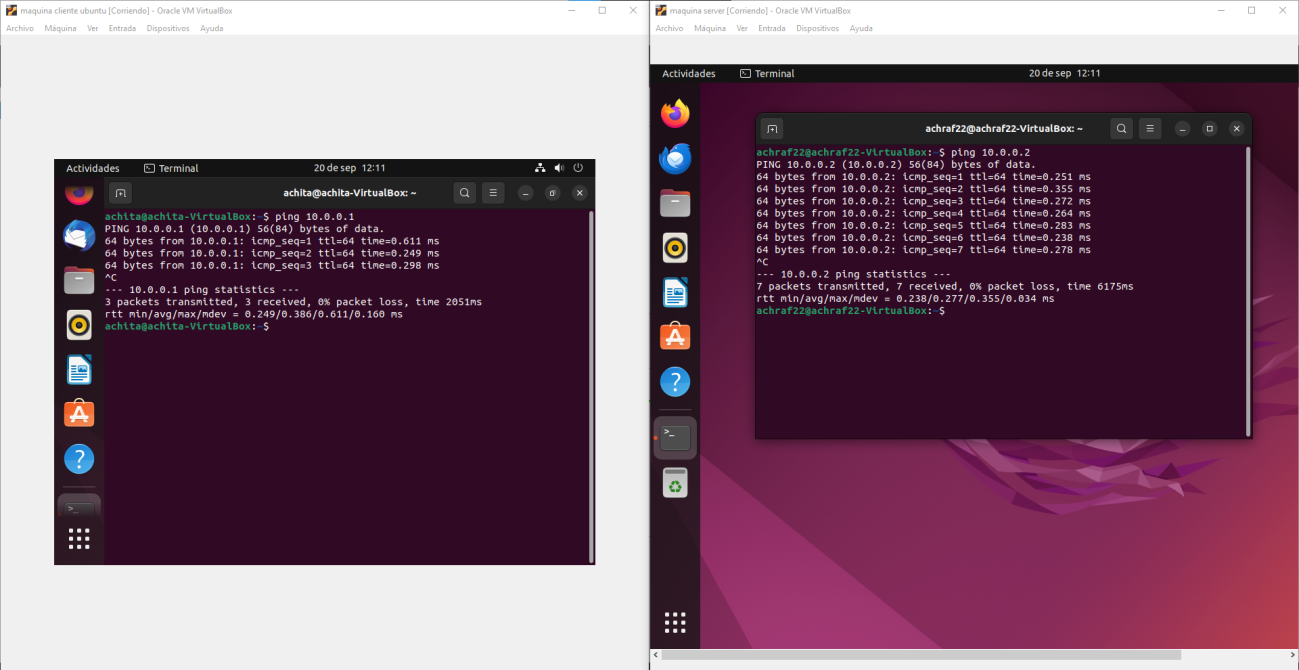
Como podemos observar en la siguiente imagen en la configuración de internet de las maquinas en el apartado de IPv4, ya estaría configurado manualmente con cada una de sus IPs.

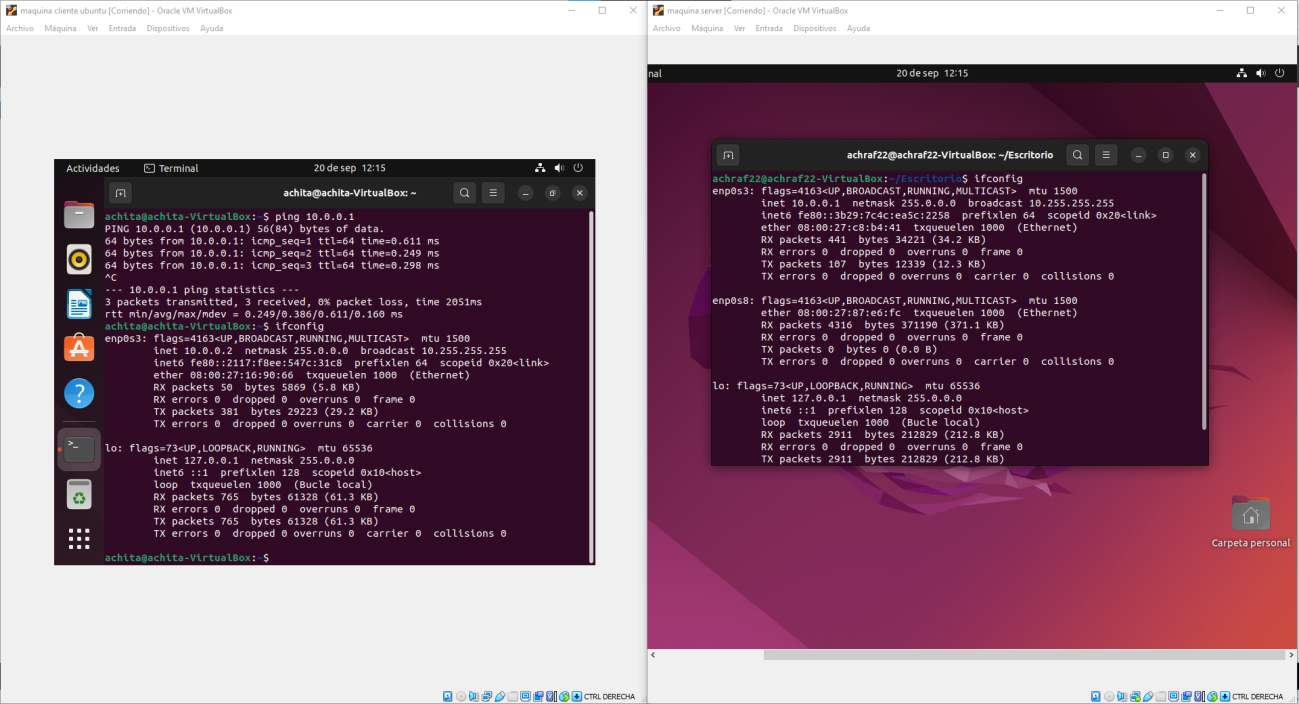


#### **Paso 3**

Ahora tocaría comprobar que todo correcto haciendo ping de una maquina otra para ver que están contestadas entre sí.

Y comprobar que la ip esta en el adaptador correcto abriendo una terminal y poniendo el comando “ifconfig”.





#### **Paso 4**

Una vez que ya hemos logrado que estén conectadas entre si lo que tenemos que hacer ahora es que la maquina cliente pueda conectarse a internet, es decir a la red local.

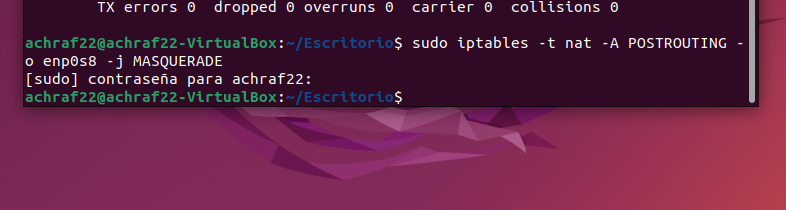
Para ello tenemos que habilitar el adaptador puente y la interfaz de Salida a la red local del servidor

Para eso abrimos una termina y ejecutamos el siguiente comando “sudo iptables -t nat -A POSTROUTING -o enp0s8 -j MASQUERADE” y lo a continuación lo desglosare para explicar cómo funciona:

1. Sabemos que cuando ejecutamos un comando con “sudo”, es con los privilegios de súper usuario para modificar las iptables.
2. “iptables” es la herramienta que gestiona las reglas de firewall en linux y permite manipulas esas tablas del firewall.
3. “-t nat” en este caso seleccionamos la tabla NAT que se usa para traducir direcciones IP y es la técnica que se emplea para que varios dispositivos en una red interna (la maquina servidor y la cliente) compartan una misma IP pública.
4. “-A POSTROUTING” en este caso con el –A estamos añadiendo una regla a la cadena POSTROUTING que se encarga de modificar los paquetes justo antes de que salgan de la interfaz de red.
5. “-o enp0s8” aquí se especifica la interfaz de salida ,significa que esa regla aplicara a los paquetes que salgan de allí.
6. “-j MASQUERADE” es un tipo de NAT que cambia o modifica las direcciones IP de los paquetes que salen por la interfaz de salida a la IP pública de esa interfaz.

# ***Conclusión***

Hemos conseguido que las maquinas que están en red interna con IPs privadas compartan una pública para conectarse a internet , ya que el servidor puede hacerlo por lo que las maquinas conectadas en red interna a el también atrás de él:

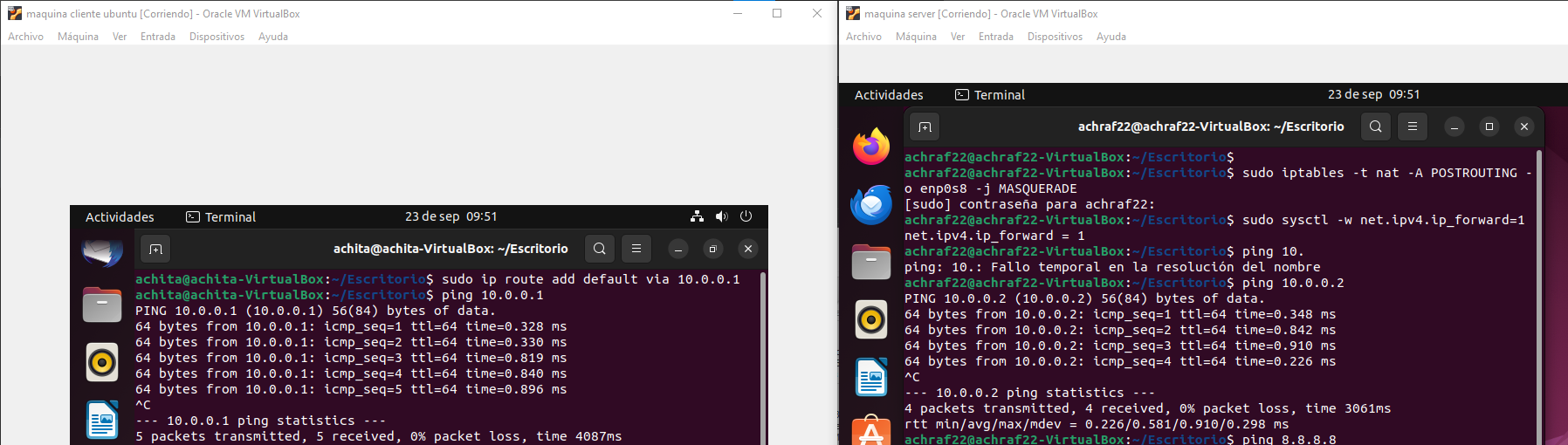


En la siguiente imagen mostramos un comando mas que se tiene que ejecutar en el servidor y otro en la maquina cliente para terminar de verificar el acceso a la red pública.

El comando “sudo sysctl -w net.ipv4.ip\_forward=1” sirve para habilitar reenvio de paquetes IPv4 en linux, es necesario aplicar este comando ya que la máquina servidor esta haciendo como router entre la red interna y la externa.

Y el comando “sudo ip route add default via 10.0.0.1”, con este comando lo que le decimos a la maquina cliente que agregue una ruta por defecto a la tabla de enrutamiento de la maquina.

La ruta por defecto indica al S.O que puerta de enlace usar para enviar el tráfico que no tiene una ruta especifica en la tabla de enrutamiento que en este caso es la IP del servidor la 10.0.0.1.



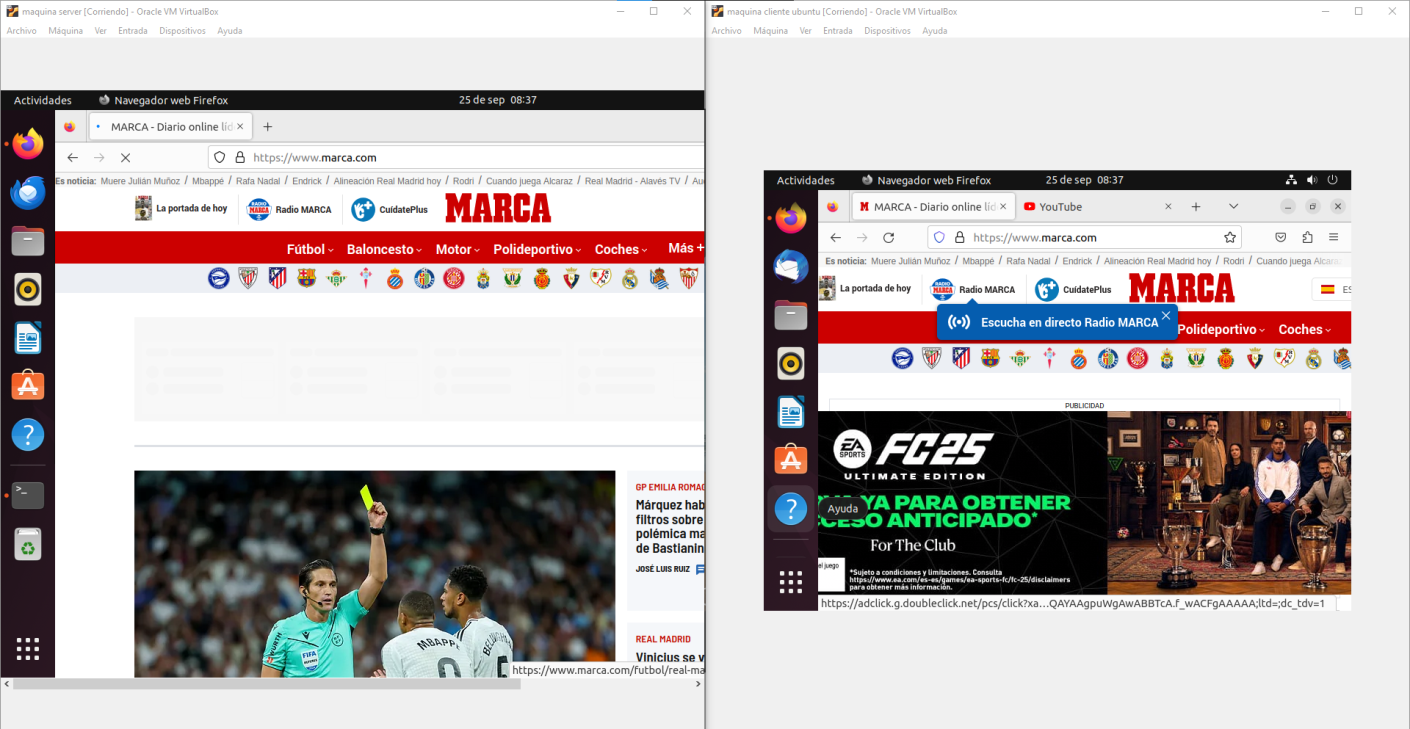
Y por último para acabar con la comprobación de que la maquina cliente ya puede acceder a la red local lo que tenemos que hacer es hacer un ping 8.8.8.8 y luego uno a “[www.google.com](http://www.google.com/)” para ver si los paquetes que se envían son recibido en el servidor y mandados a la puerta de enlace.

Como se puede comprobar lo hemos hechos en las 2 maquinas y funciona.

# ***Comprobación final***

Como se puede observar en la siguiente imagen ya nos dejaba acceder a internet, es decir, nos ha dejado

acceder a la red mediante el adaptador puente de la maquina servidor que en este caso hacia de gateway.

******

# ***Últimos problemas***

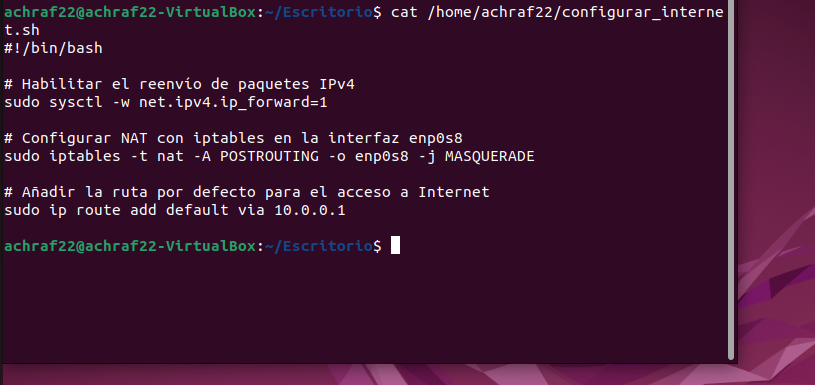
A la hora de finalizar la practica existía un problema, cuando apagábamos la maquinas se restauraba el sistema y aparecía como si no hubiésemos ejecutado ningún comando anterior porque ya no funcionaba el acceso de internet, ademas se apagaba por defecto la interfaz del adaptador puente del servidor.

Por lo que la solución al problema fue la creación de un script de inicio de sesión.

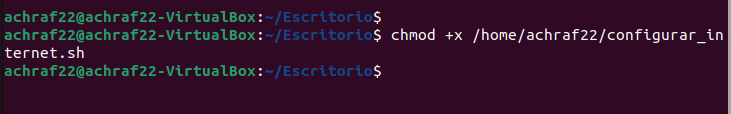
Investigando un poco encontré que el .bashrc es un archivo de configuración que interpreta comandos en sistemas operativos de Linux y su función principal era que esas funciones se cargases justo al iniciarse el sistema.

Por lo que comenzaremos a desarrollar un script, para ellos teníamos que crear un archivo.sh en mi caso lo llame Configurar\_internet.sh y para escribir dentro de el utilizábamos el comando nano.

Dentro de él escribíamos cada comando de cada maquina tanto la servidora como la cliente.



Ahora le dábamos permisos de ejecución con el comando chmod +x.



Por últimos guardábamos nuestro a script dentro del archivo de configuración .bashrc en la última linea de comando añadiéndole un & para que lo ejecute en segundo plano.

Cuando apagamos e iniciamos las maquinas lo único que nos faltaba es volver a encender el adaptador puente y ya volvía a funcionar todo correctamente.

