

ACCESO REMOTO

Raúl Arribas Redondo

ASO

Contenido

1	Apartado 1. Implantación de servicios de Escritorio Remoto y RemoteApps.....	2
1.1	Levanta los Servicios de Escritorio Remoto en una máquina Windows Server.....	2
1.2	Crear un grupo llamado estudiantes y dos usuarios, estudiante1 y estudiante2, que pertenezcan al grupo anterior.	4
1.3	Instala la aplicación <i>Notepad++</i> (u otra similar) en el servidor y compártela vía RemoteApp.	5
1.4	Haz que solamente el grupo estudiantes tenga acceso remoto a dicha aplicación....	10
2	Dadas dos máquinas virtuales de Windows Server y GNU/Linux, establece la siguiente configuración para conectarlas mediante un túnel SSH:	11
2.1	Nombre de equipo: wtunombre (para Windows), Inxtunombre (para GNU/Linux)...	11
	Direcciones IP: 10.10.10.11 (para Windows), 10.10.10.10 (para GNU/Linux).	11
2.2	Instala en el equipo GNU/Linux un servidor SSH y otro MySQL (mysql-server junto con sus dependencias).	12
2.3	Crea un túnel SSH con las siguientes características:.....	13
2.4	Una vez establecida la conexión, accede a la consola de administración de MySQL.	18
3	Apartado 3. Comparativa final y conclusión:.....	19
3.1	¿Cuáles son las ventajas y desventajas de utilizar el Escritorio Remoto en comparación con la conexión SSH?	19
3.2	¿Qué medidas de seguridad adicionales podrías implementar para proteger las conexiones remotas en ambos sistemas operativos?	21

1 Apartado 1. Implantación de servicios de Escritorio Remoto y RemoteApps

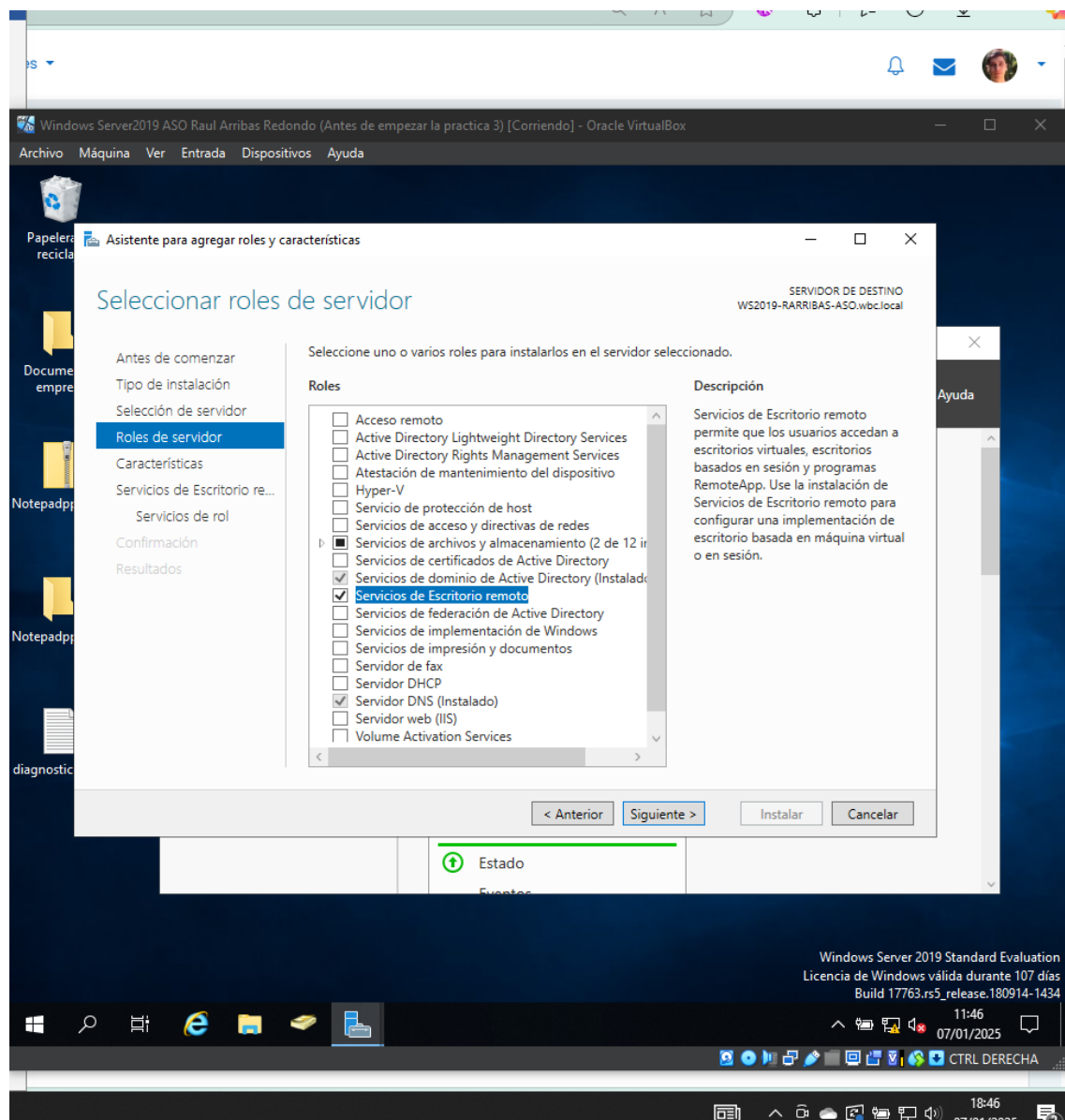
1.1 Levanta los Servicios de Escritorio Remoto en una máquina Windows Server. Iniciamos sesión en nuestra máquina Windows Server 2019.

Abrimos Administrador del Servidor → "Agregar Roles y Características".

Una vez estemos en el asistente, seleccionamos:

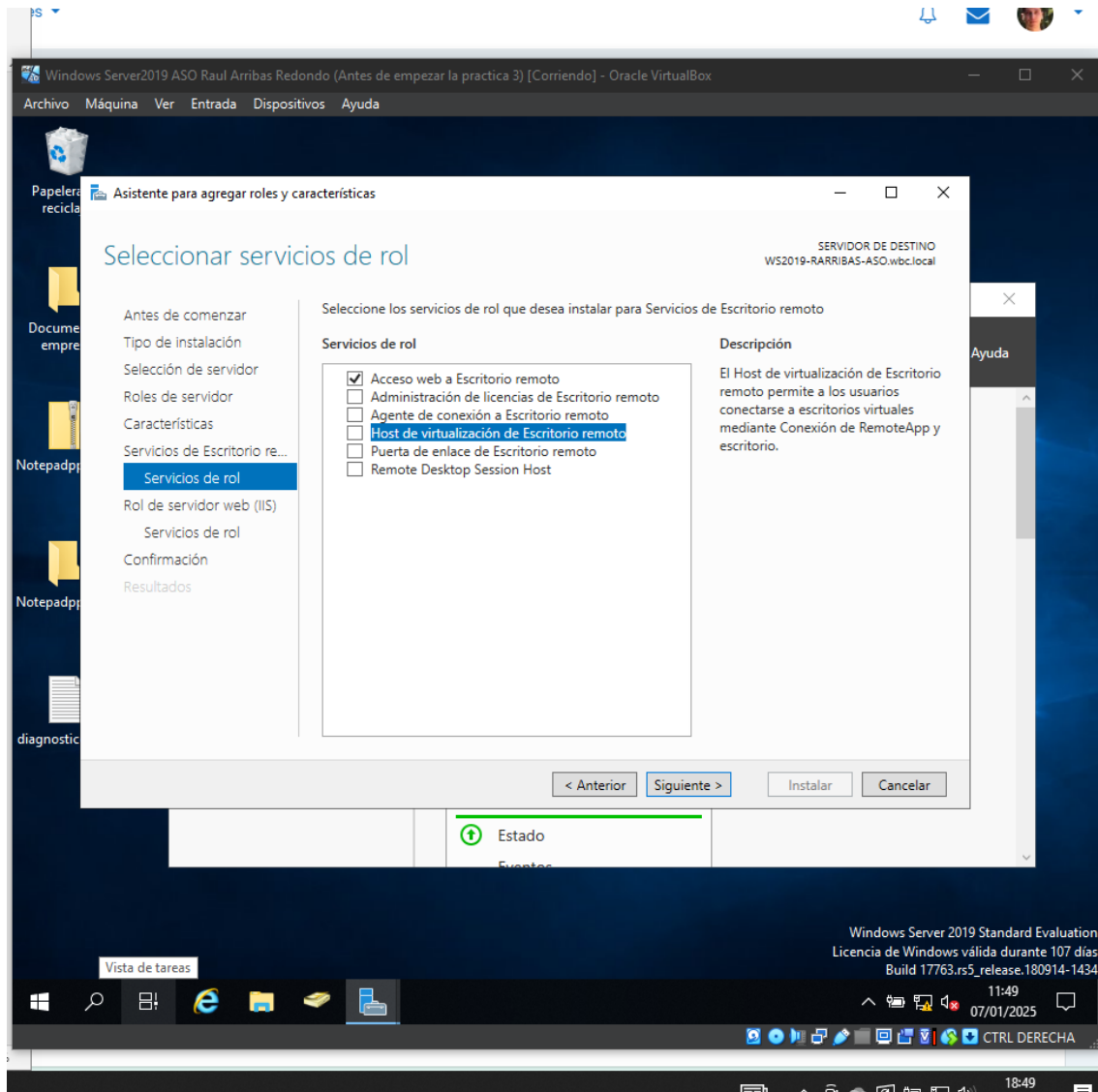
Tipo de instalación: "Basada en características o en roles".

Seleccionamos el rol "Servicios de Escritorio Remoto".



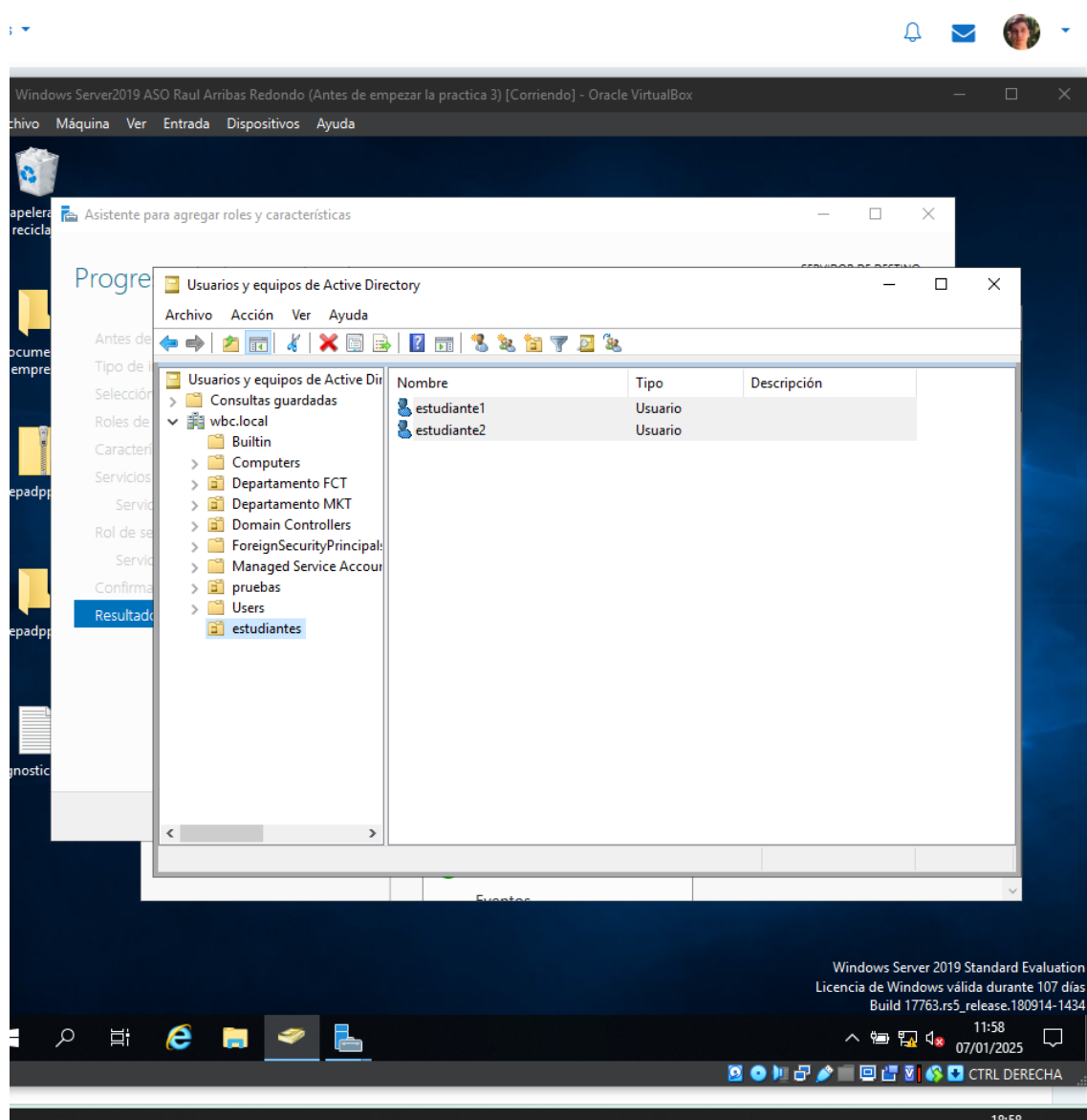
Marca las características: "Acceso web de Escritorio Remoto" (opcional si necesitas acceso vía navegador).

Acceso remoto



Acceso remoto

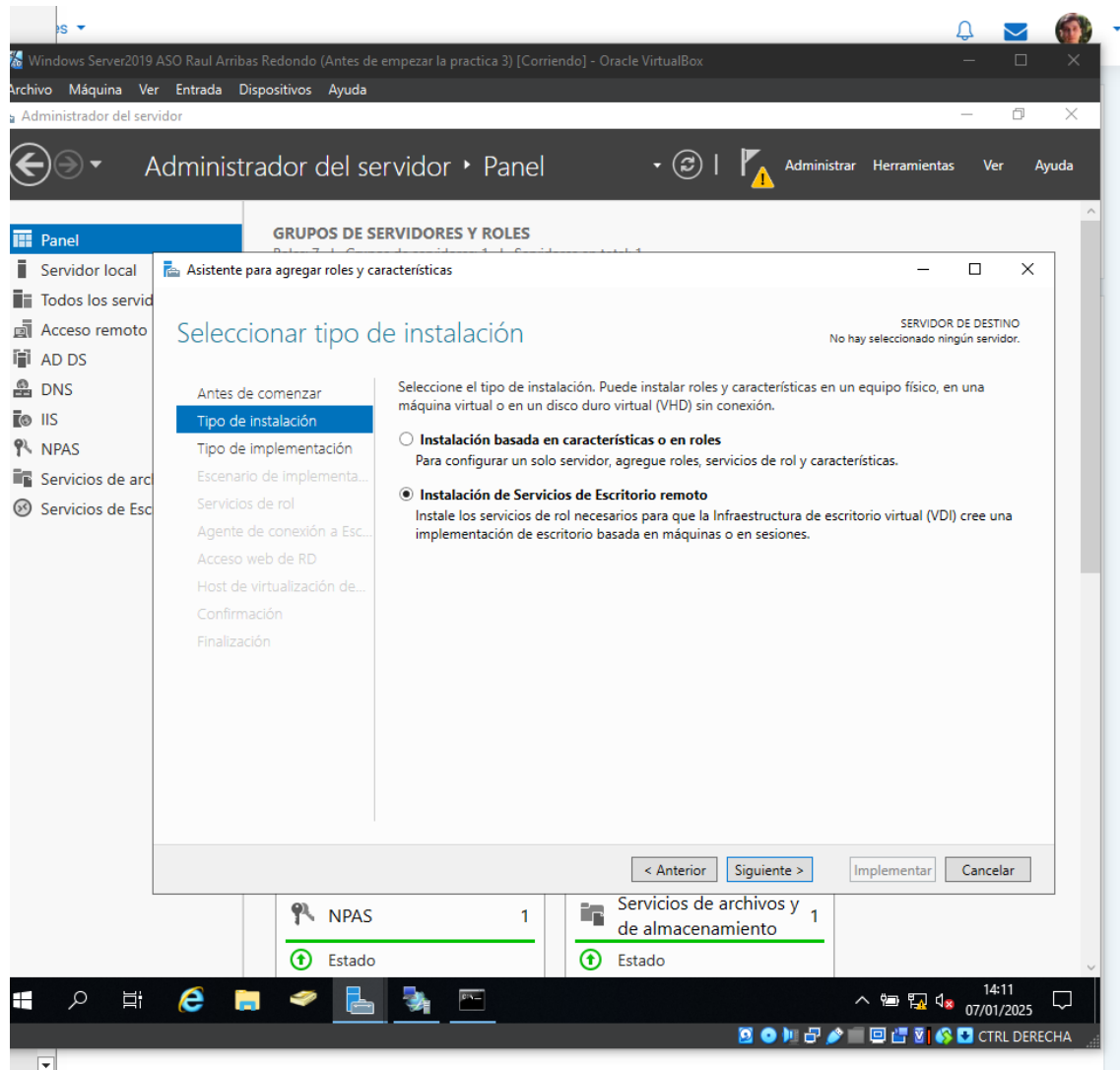
- 1.2 Crear un grupo llamado estudiantes y dos usuarios, estudiante1 y estudiante2, que pertenezcan al grupo anterior.



1.3 Instala la aplicación *Notepad++* (u otra similar) en el servidor y compártela vía RemoteApp.

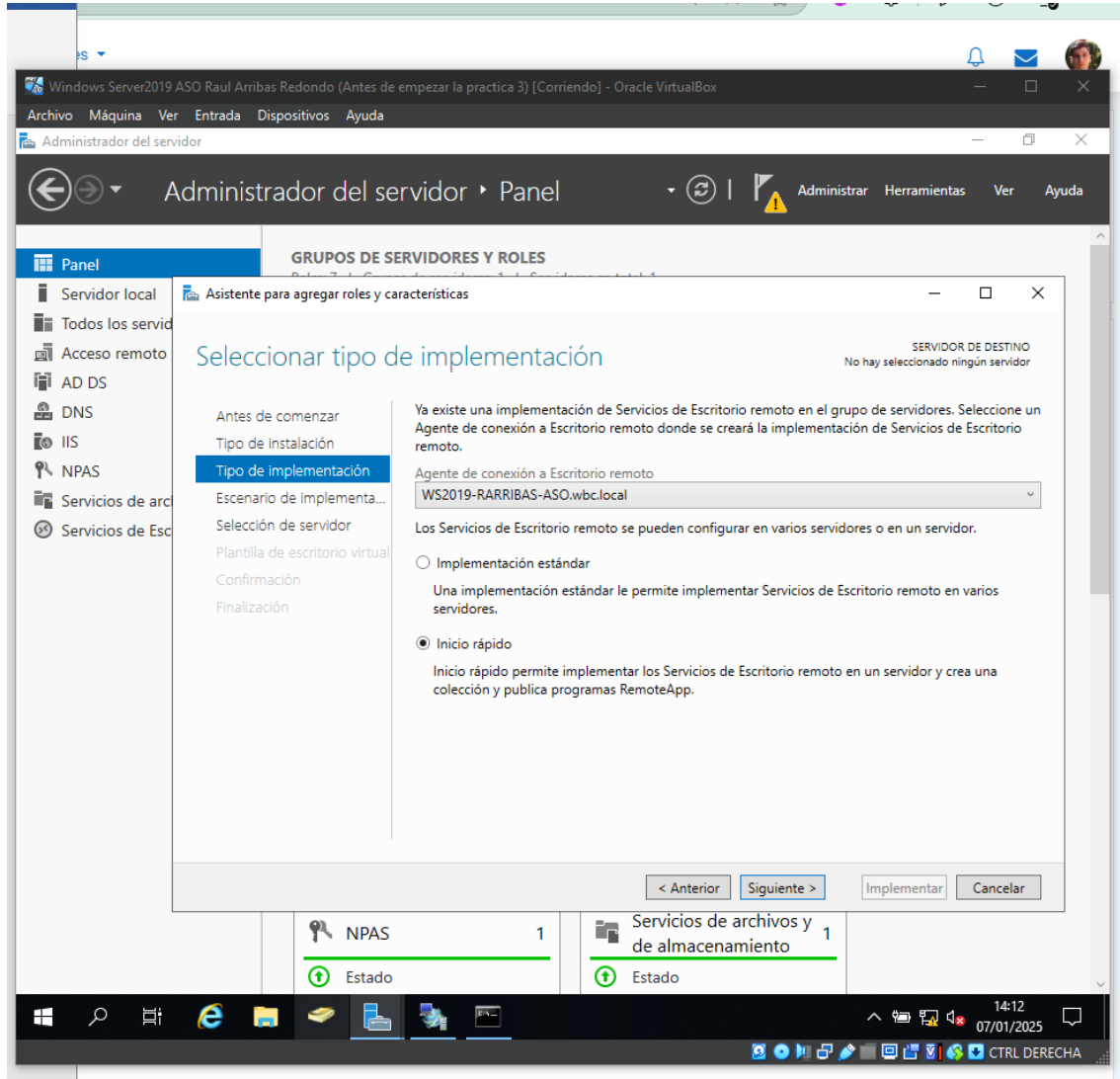
Nos descargamos Notepad en el servidor.

Dentro de agregar roles y características, hacemos click en el tipo de instalación de Servicios de escritorio remoto.

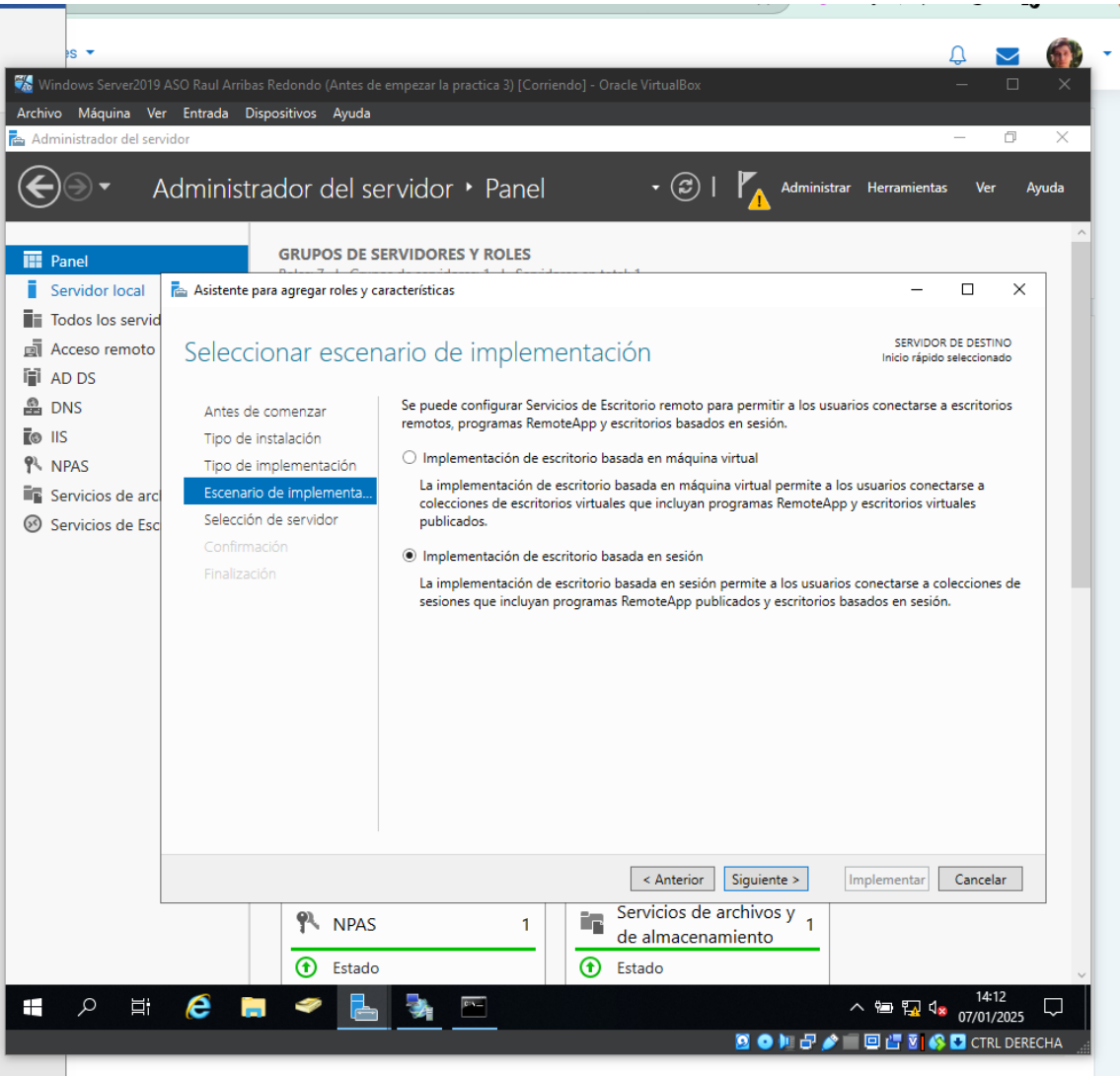


Acceso remoto

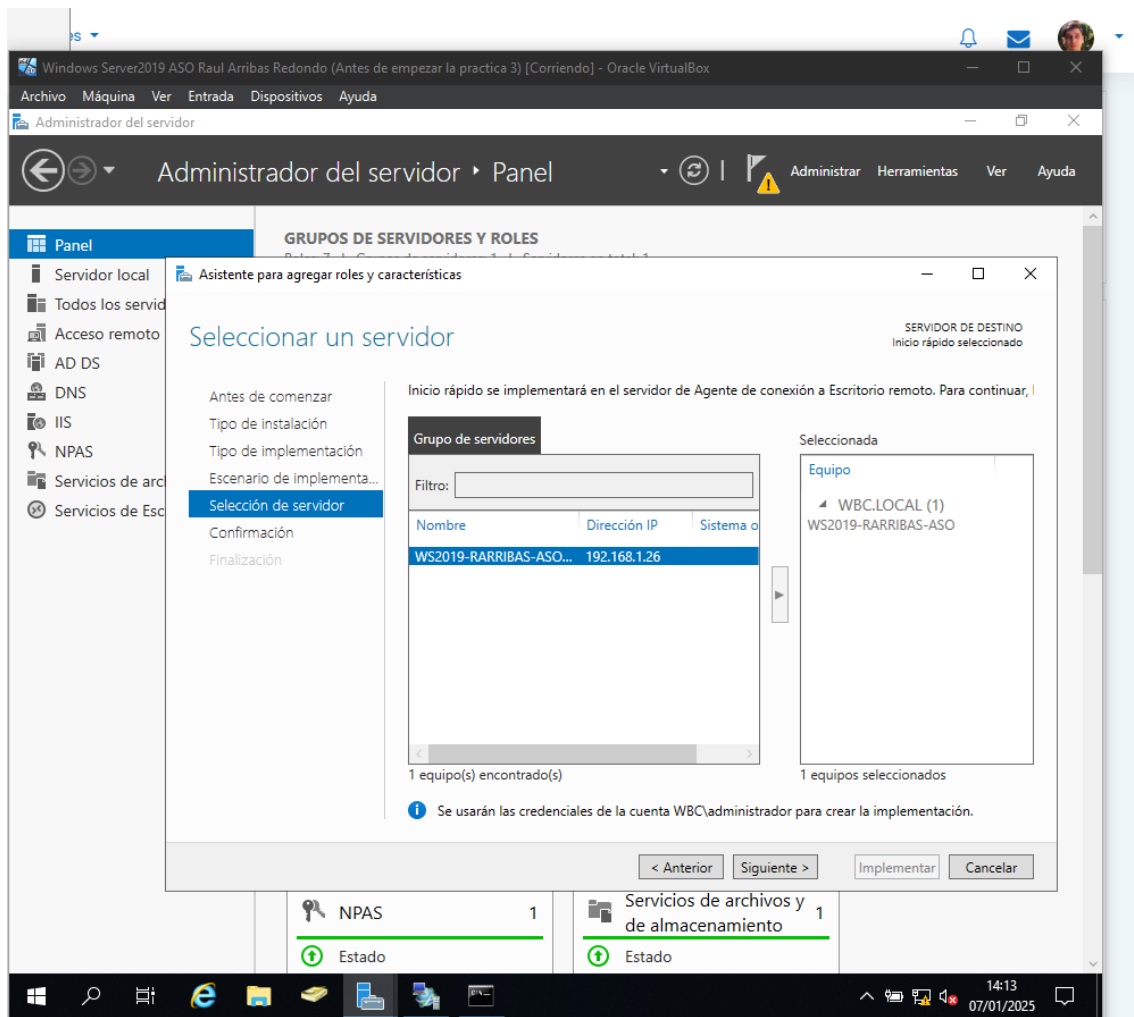
Clicamos en inicio rápido:



Implementación de escritorio basada en sesión:

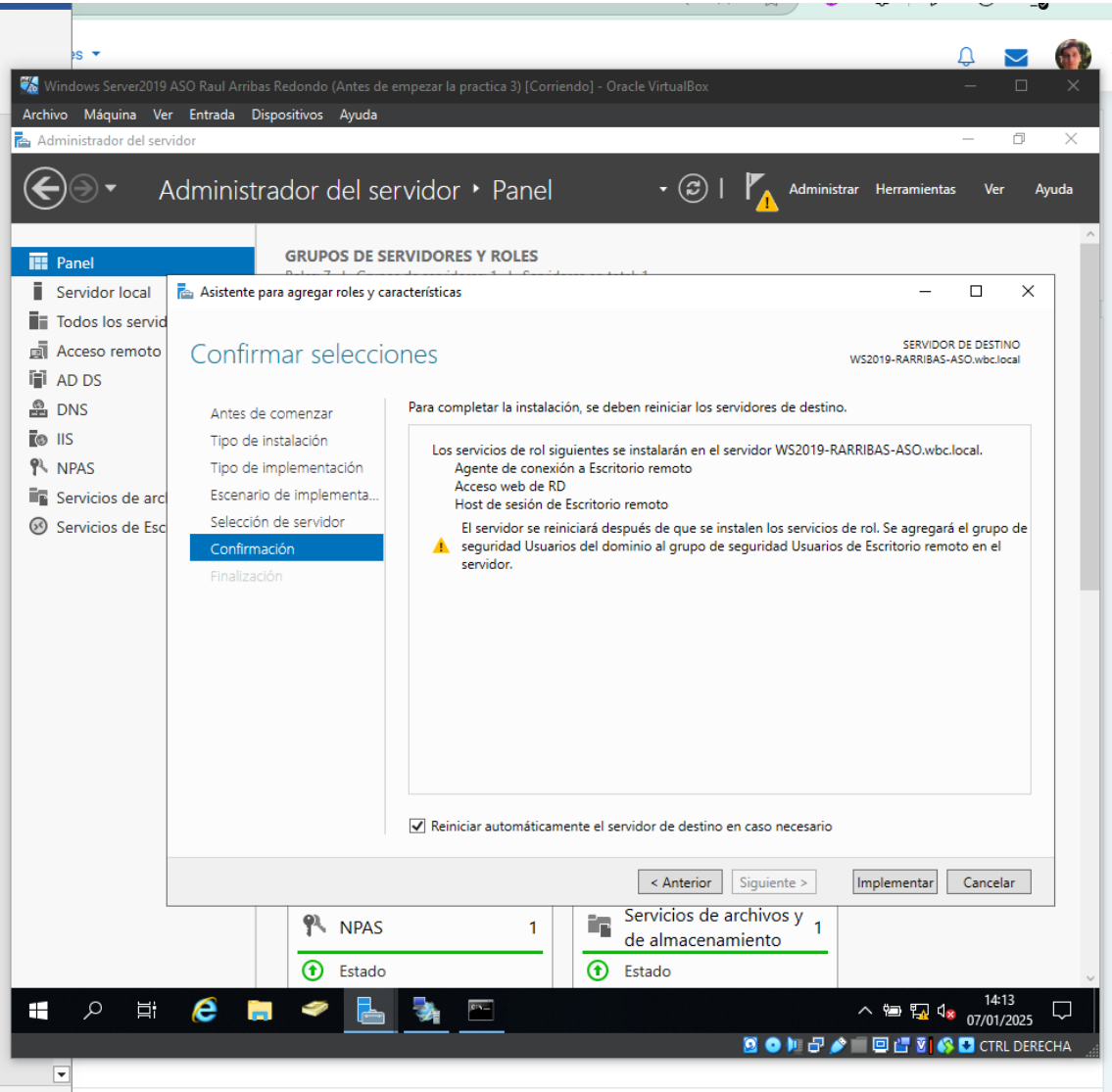


Seleccionamos el servidor donde lo queremos instalar:



Acceso remoto

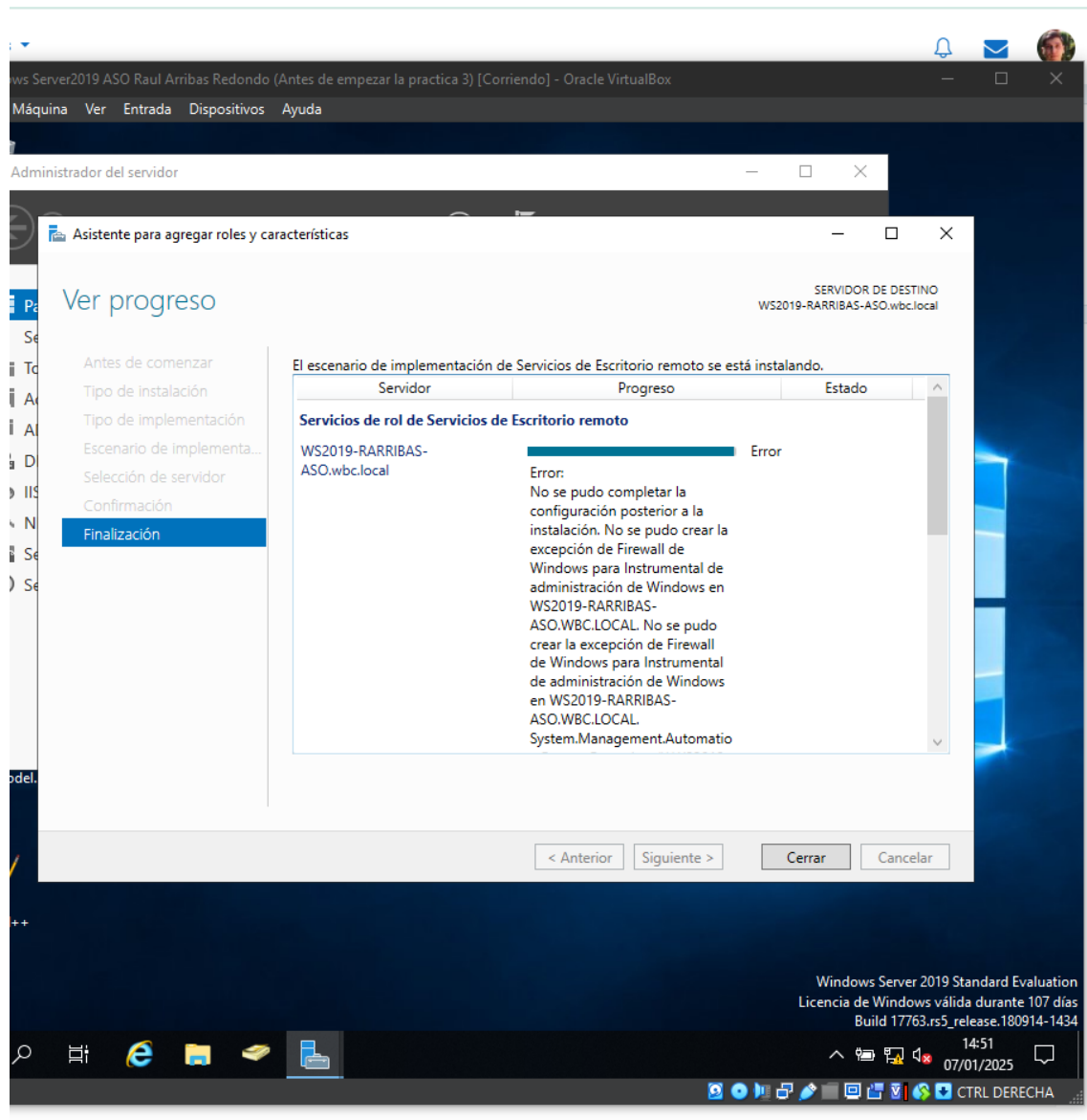
Seleccionamos implementar:



Acceso remoto

He estado un buen rato intentando esto pero me sale este error, no he conseguido solucionarlo:

He intentado permitir los puertos desde el firewall, desactivarlo..



1.4 Haz que solamente el grupo estudiantes tenga acceso remoto a dicha aplicación.

Como consecuencia del anterior fallo esta no la puedo realizar.

Basicamente sería irse a la aplicación de Remoteapp y configurar permisos para que el grupo de estudiantes tenga permisos.

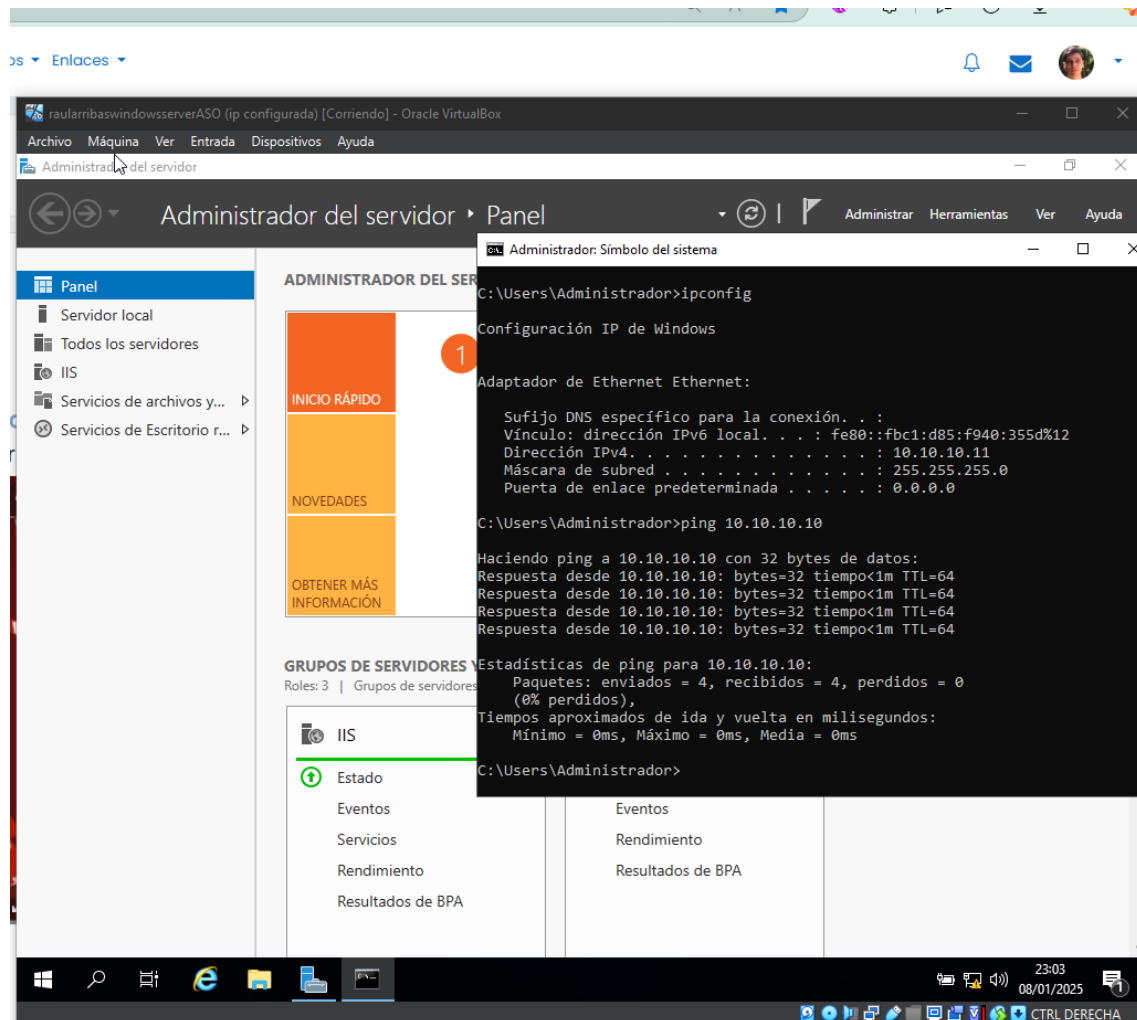
2 Dadas dos máquinas virtuales de Windows Server y GNU/Linux, establece la siguiente configuración para conectarlas mediante un túnel SSH:

2.1 Nombre de equipo: wtunombre (para Windows), lnextunombre (para GNU/Linux)

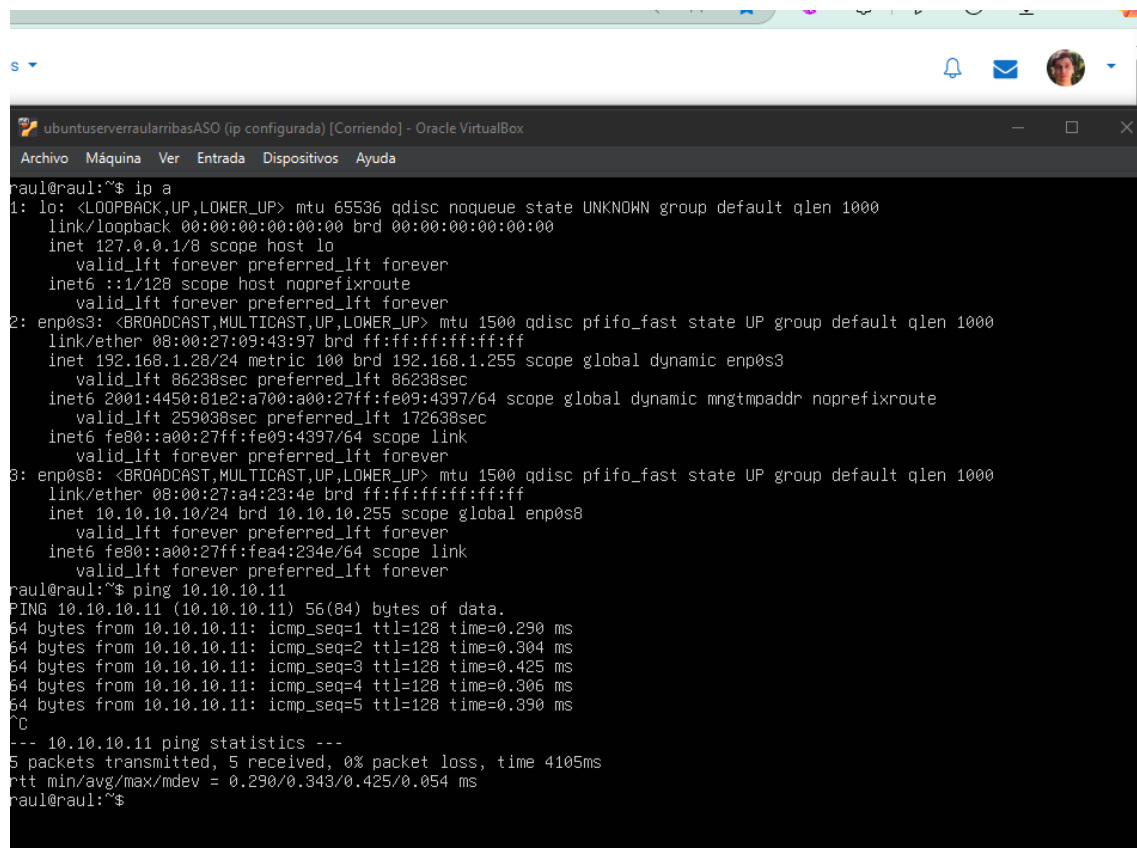
Direcciones IP: 10.10.10.11 (para Windows), 10.10.10.10 (para GNU/Linux).

Asignamos las IP a cada equipo creado, lo tenemos puesto en red interna.

Equipo Windows configuración de red:



Equipo Linux configuración de red:



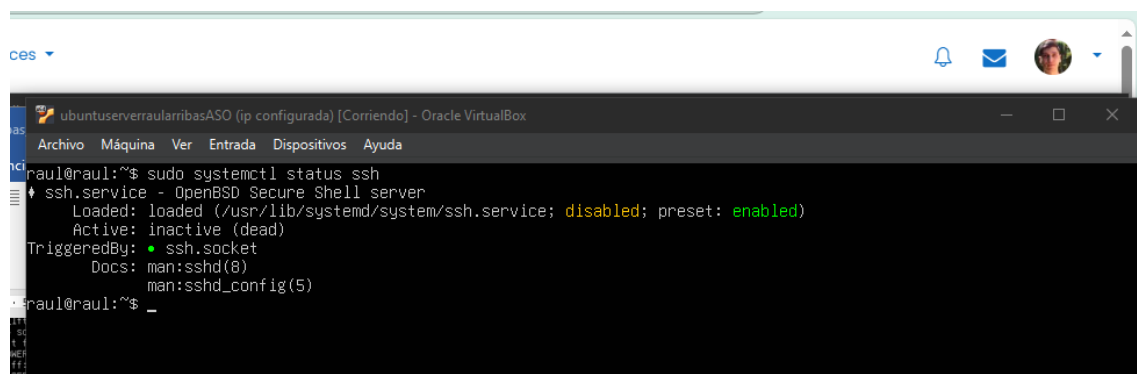
```
raul@raul:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:09:43:97 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.28/24 metric 100 brd 192.168.1.255 scope global dynamic enp0s3
        valid_lft 86238sec preferred_lft 86238sec
    inet6 2001:4450:81e2:a700:a00:27ff:fe09:4397/64 scope global dynamic mngtmpaddr noprefixroute
        valid_lft 259038sec preferred_lft 172638sec
    inet6 fe80::a00:27ff:fe09:4397/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:a4:23:4e brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.10/24 brd 10.10.10.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fea4:234e/64 scope link
        valid_lft forever preferred_lft forever
raul@raul:~$ ping 10.10.10.11
PING 10.10.10.11 (10.10.10.11) 56(84) bytes of data:
64 bytes from 10.10.10.11: icmp_seq=1 ttl=128 time=0.290 ms
64 bytes from 10.10.10.11: icmp_seq=2 ttl=128 time=0.304 ms
64 bytes from 10.10.10.11: icmp_seq=3 ttl=128 time=0.425 ms
64 bytes from 10.10.10.11: icmp_seq=4 ttl=128 time=0.306 ms
64 bytes from 10.10.10.11: icmp_seq=5 ttl=128 time=0.390 ms
^C
--- 10.10.10.11 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4105ms
rtt min/avg/max/mdev = 0.290/0.343/0.425/0.054 ms
raul@raul:~$
```

2.2 Instala en el equipo GNU/Linux un servidor SSH y otro MySQL (mysql-server junto con sus dependencias).

Instalar SSH

Nosotros hemos instalado el servicio SSH por defecto en la instalación de Ubuntu Server, para instalar SSH ejecutar el comando **sudo apt install openssh-server -y**

Para comprobar que lo tenemos instalado ejecutamos **sudo systemctl status ssh**:



```
raul@raul:~$ sudo systemctl status ssh
ssh.service - OpenBSD Secure Shell server
Loaded: loaded (/usr/lib/systemd/system/ssh.service; disabled; preset: enabled)
Active: inactive (dead)
TriggeredBy: ● ssh.socket
Docs: man:sshd(8)
     man:sshd_config(5)
raul@raul:~$
```

Acceso remoto

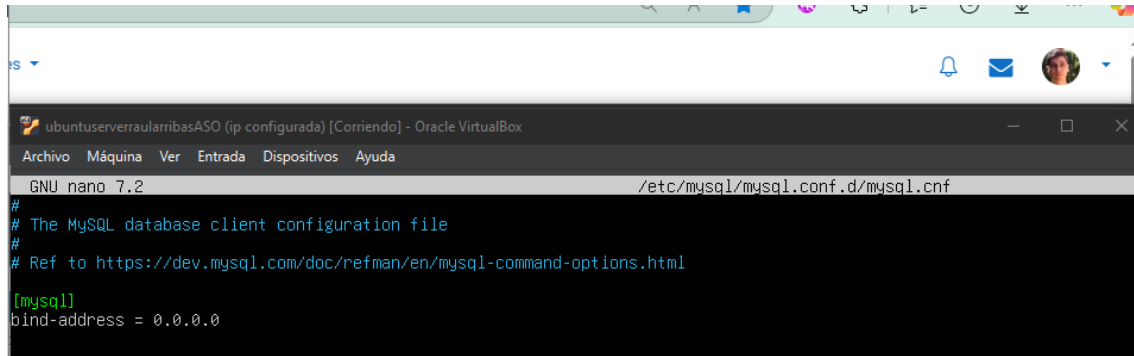
Instalar MySQL:

Para instalar MySQL server y sus dependencias tenemos que ejecutar el siguiente comando:

sudo apt install mysql-server -y

Configuramos MySQL para que acepte conexiones de todas las maquinas:

sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf



```
GNU nano 7.2 /etc/mysql/mysql.conf.d/mysqld.cnf
#
# The MySQL database client configuration file
#
# Ref to https://dev.mysql.com/doc/refman/en/mysql-command-options.html
[mysql]
bind-address = 0.0.0.0
```

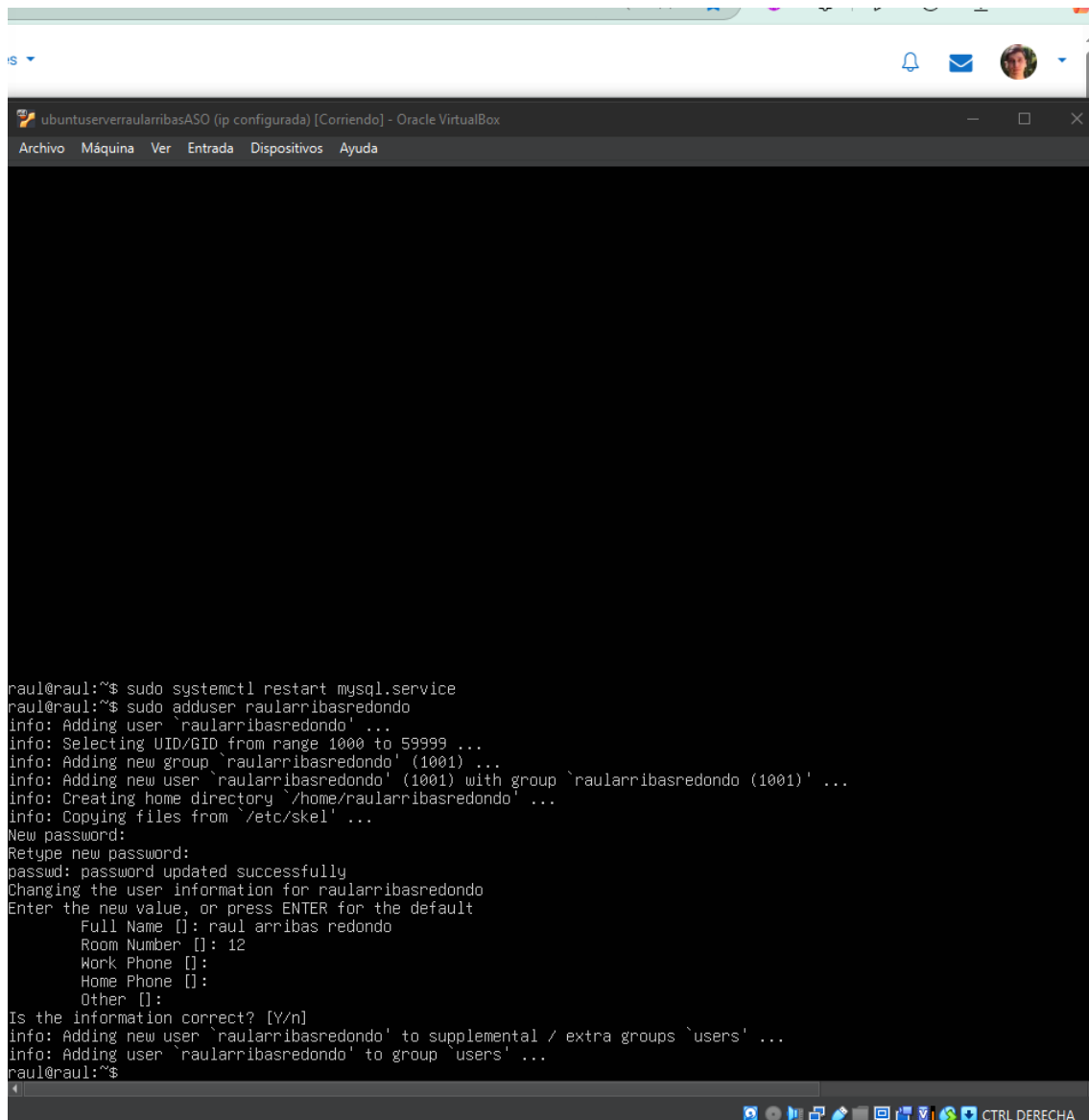
2.3 Crea un túnel SSH con las siguientes características:

- **Direcciones IP: 10.10.10.11 (para origen), 10.10.10.10 (para destino).**
- **Puertos: 3306 (para MySQL), 5252 (para el servidor SSH).**
- **El usuario que acceda deberá ser uno llamado tunombreadm, que habrás creado previamente en el equipo GNU/Linux y que tendrá autorización el acceso.**

Creamos usuario raularribasredondoaso en la maquina Linux

Sudo adduser raularribasredondo

Acceso remoto



The screenshot shows a remote terminal window titled 'ubuntuserverraularribasASO (ip configurada) [Corriendo] - Oracle VirtualBox'. The terminal output shows the following commands and their results:

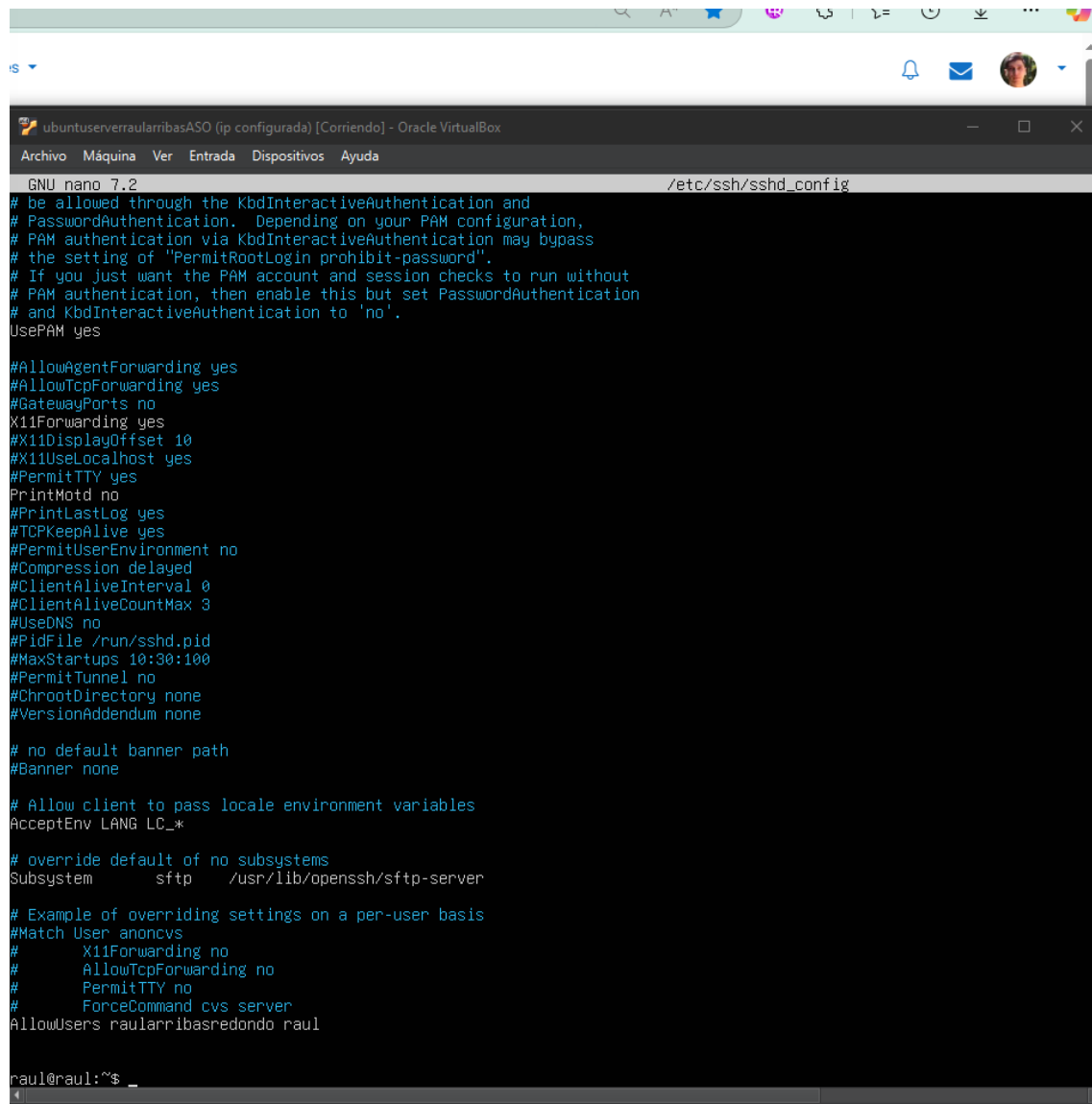
```
raul@raul:~$ sudo systemctl restart mysql.service
raul@raul:~$ sudo adduser raularribasredondo
info: Adding user `raularribasredondo' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `raularribasredondo' (1001) ...
info: Adding new user `raularribasredondo' (1001) with group `raularribasredondo (1001)' ...
info: Creating home directory `/home/raularribasredondo' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for raularribasredondo
Enter the new value, or press ENTER for the default
  Full Name []: raul arribas redondo
    Room Number []: 12
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n]
info: Adding new user `raularribasredondo' to supplemental / extra groups `users' ...
info: Adding user `raularribasredondo' to group `users' ...
raul@raul:~$
```

Acceso remoto

Configuramos para que el usuario que hemos creado pueda acceder por ssh:

sudo nano /etc/ssh/sshd_config

AllowUsers raularribasredondo raul



```
GNU nano 7.2 /etc/ssh/sshd_config
# be allowed through the KbdInteractiveAuthentication and
# PasswordAuthentication. Depending on your PAM configuration,
# PAM authentication via KbdInteractiveAuthentication may bypass
# the setting of "PermitRootLogin prohibit-password".
# If you just want the PAM account and session checks to run without
# PAM authentication, then enable this but set PasswordAuthentication
# and KbdInteractiveAuthentication to 'no'.
UsePAM yes

#AllowAgentForwarding yes
#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PermitTTY yes
PrintMotd no
#PrintLastLog yes
#TCPKeepAlive yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS no
#PidFile /run/ssh.pid
#MaxStartups 10:30:100
#PermitTunnel no
#ChrootDirectory none
#VersionAddendum none

# no default banner path
#Banner none

# Allow client to pass locale environment variables
AcceptEnv LANG LC_*

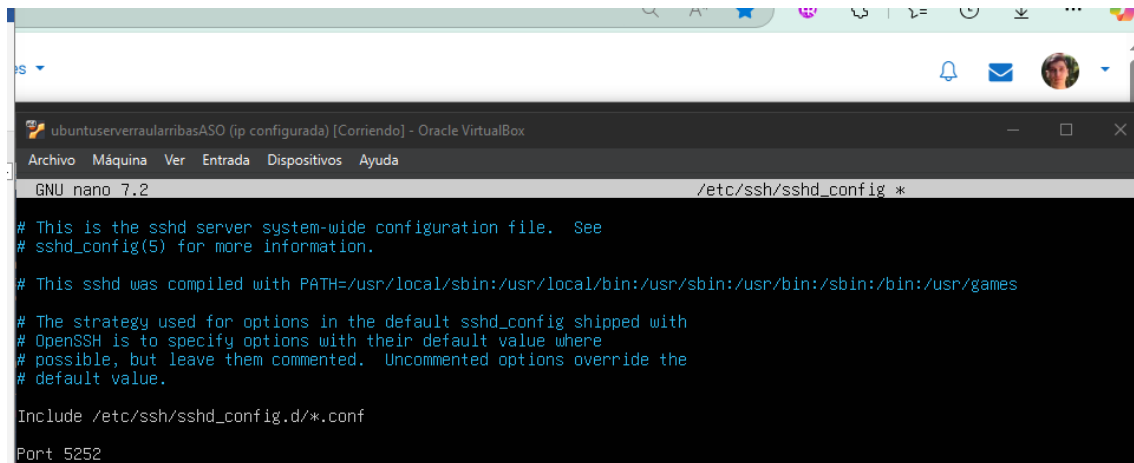
# override default of no subsystems
Subsystem sftp /usr/lib/openssh/sftp-server

# Example of overriding settings on a per-user basis
#Match User anoncvs
#    X11Forwarding no
#    AllowTcpForwarding no
#    PermitTTY no
#    ForceCommand cvs server
AllowUsers raularribasredondo raul

raul@raul:~$ _
```

Tambien cambiamos el Puerto al 5252, donde pone #Port 22, ponemos # Port 5252

Acceso remoto



The screenshot shows a terminal window titled 'ubuntuserverraularribasASO (ip configurada) [Corriendo] - Oracle VirtualBox'. The terminal is running the nano text editor on the file /etc/ssh/sshd_config. The visible content of the file includes comments about the configuration file, the compilation path, and the default port 5252.

```
GNU nano 7.2 /etc/ssh/sshd_config *

# This is the sshd server system-wide configuration file. See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games

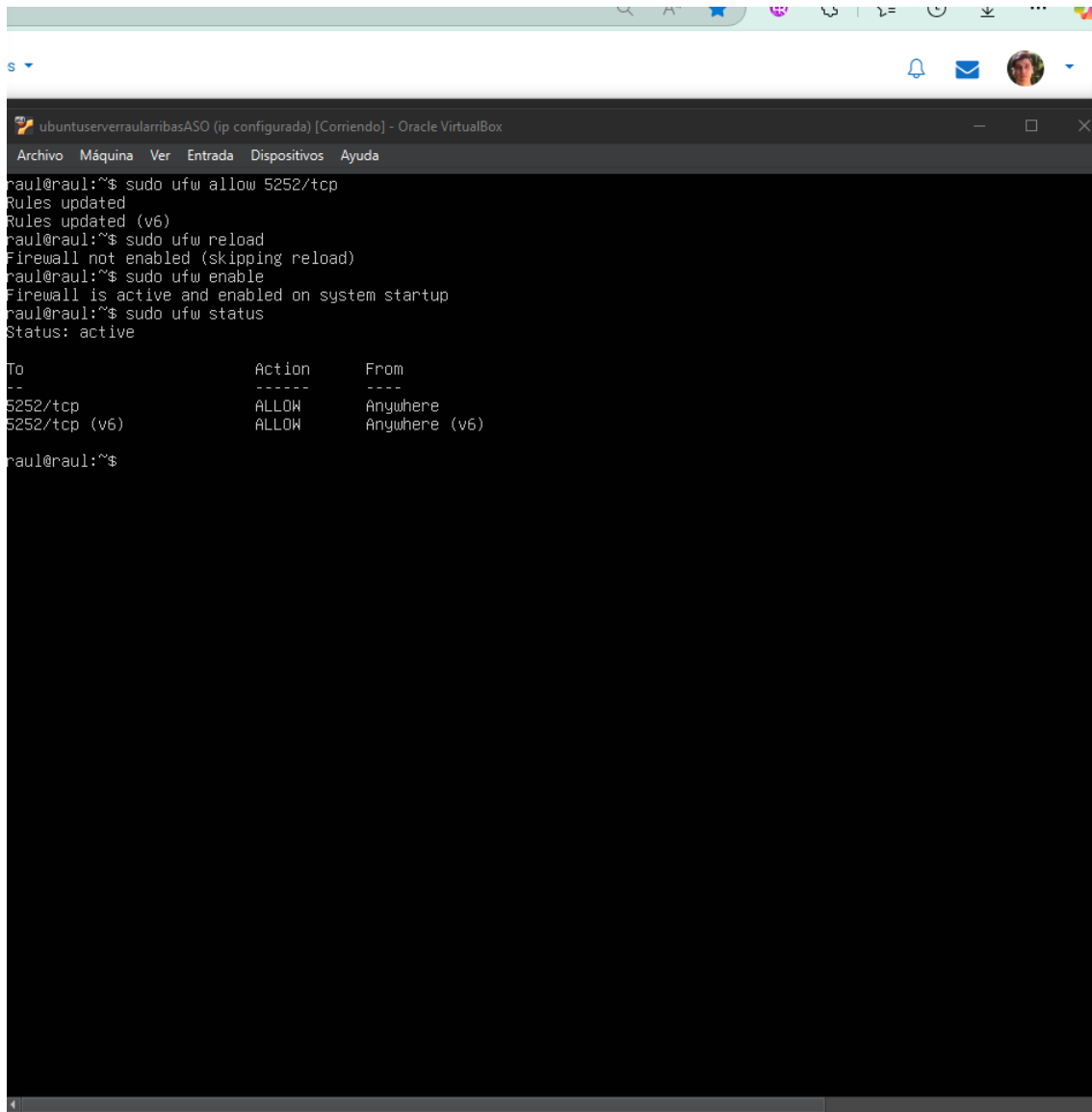
# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 5252
```

Ahora permitimos con ufw el puerto 5252:

sudo ufw allow 5252/tcp



The screenshot shows a terminal window with the following commands and output:

```
raul@raul:~$ sudo ufw allow 5252/tcp
Rules updated
Rules updated (v6)
raul@raul:~$ sudo ufw reload
Firewall not enabled (skipping reload)
raul@raul:~$ sudo ufw enable
Firewall is active and enabled on system startup
raul@raul:~$ sudo ufw status
Status: active

To Action From
--
5252/tcp ALLOW Anywhere
5252/tcp (v6) ALLOW Anywhere (v6)

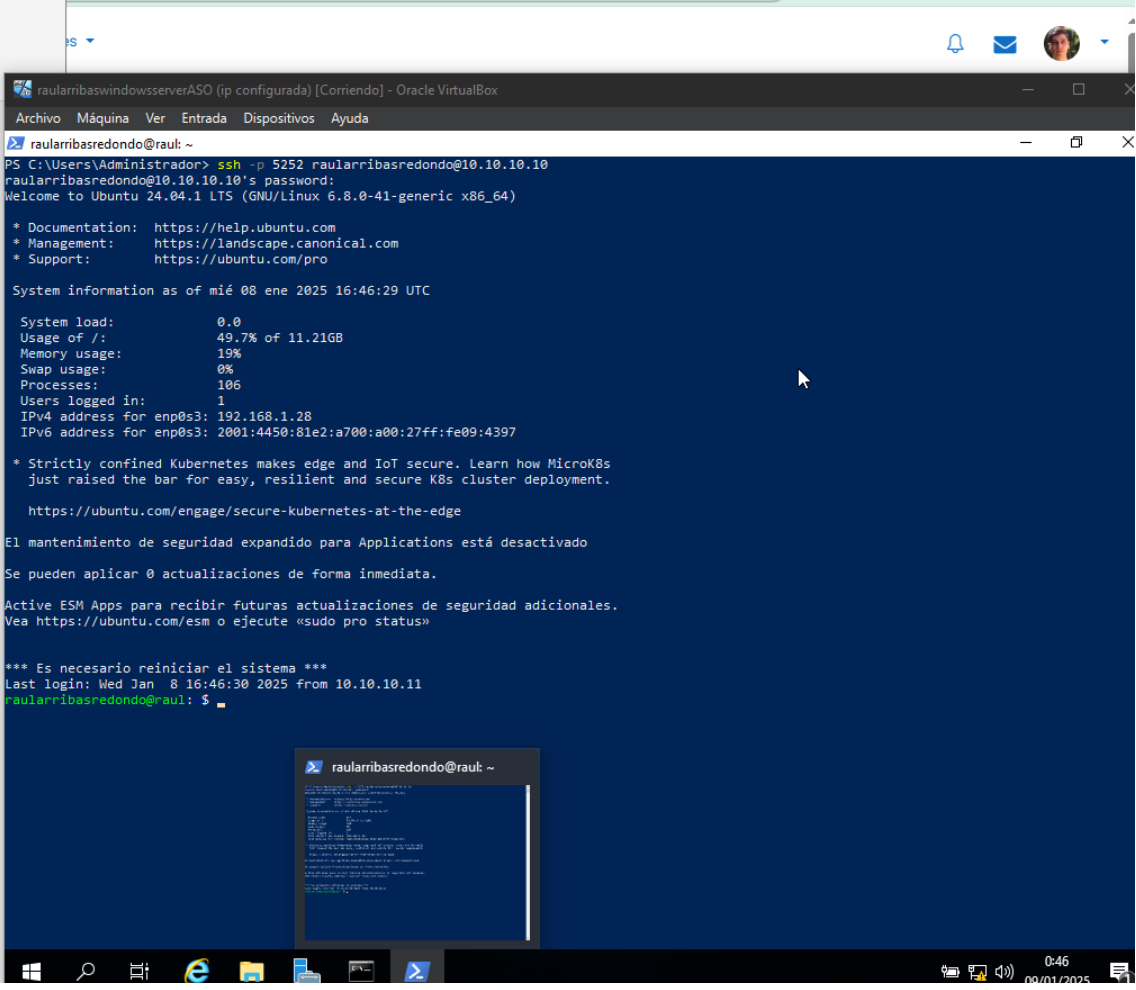
raul@raul:~$
```

Reiniciamos el servidor ssh:

Acceso remoto

Sudo systemctl restart ssh

Comprobamos que podemos acceder por ssh desde la maquina Windows Server previamente configurada con la ip 10.10.10.11:



```
PS C:\Users\Administrador> ssh -p 5252 raularribasredondo@10.10.10.10
raularribasredondo@10.10.10.10's password:
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-41-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of mié 08 ene 2025 16:46:29 UTC

System load:          0.0
Usage of /:            49.7% of 11.21GB
Memory usage:         19%
Swap usage:           0%
Processes:            106
Users logged in:      1
IPv4 address for enp0s3: 192.168.1.28
IPv6 address for enp0s3: 2001:4450:81e2:a700:a00:27ff:fe09:4397

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

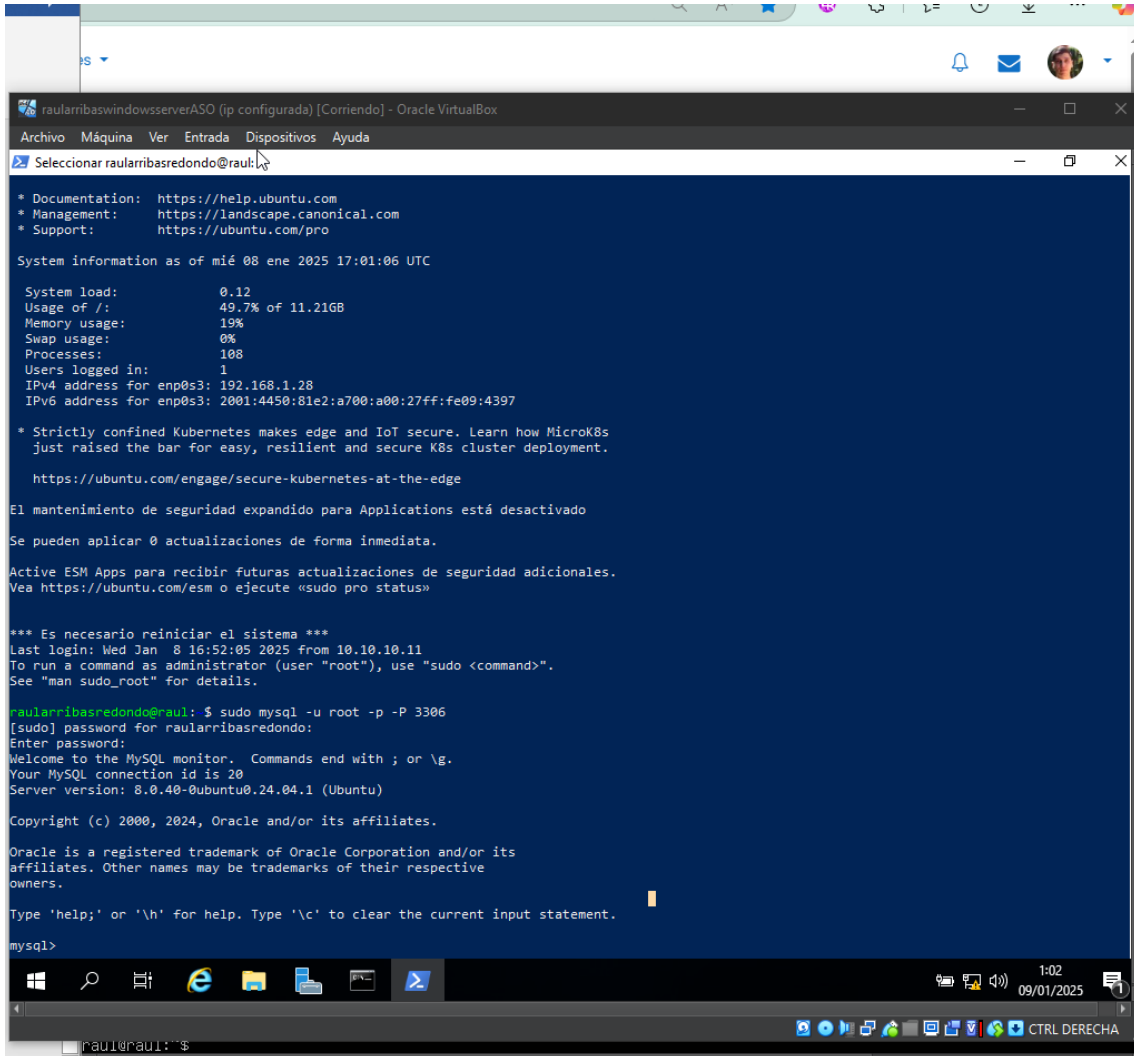
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 0 actualizaciones de forma inmediata.
Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

*** Es necesario reiniciar el sistema ***
Last login: Wed Jan  8 16:46:30 2025 from 10.10.10.11
raularribasredondo@raul: $
```

2.4 Una vez establecida la conexión, accede a la consola de administración de MySQL.

Accedemos desde el ssh a nuestro mysql.



The screenshot shows a terminal window titled "raularribasredondo@raul:~" with a menu bar (Archivo, Máquina, Ver, Entrada, Dispositivos, Ayuda). The terminal output includes system information as of January 8, 2025, at 17:01:06 UTC, such as system load (0.12), memory usage (49.7% of 11.21GB), and network addresses. It also displays security notices about ESM apps and a warning to restart the system. The user then runs the command "sudo mysql -u root -p -P 3306", enters a password, and is welcomed to the MySQL monitor. The prompt changes from "mysql>" to "mysql>". The terminal window is running on an Oracle VM VirtualBox with the guest OS being Ubuntu.

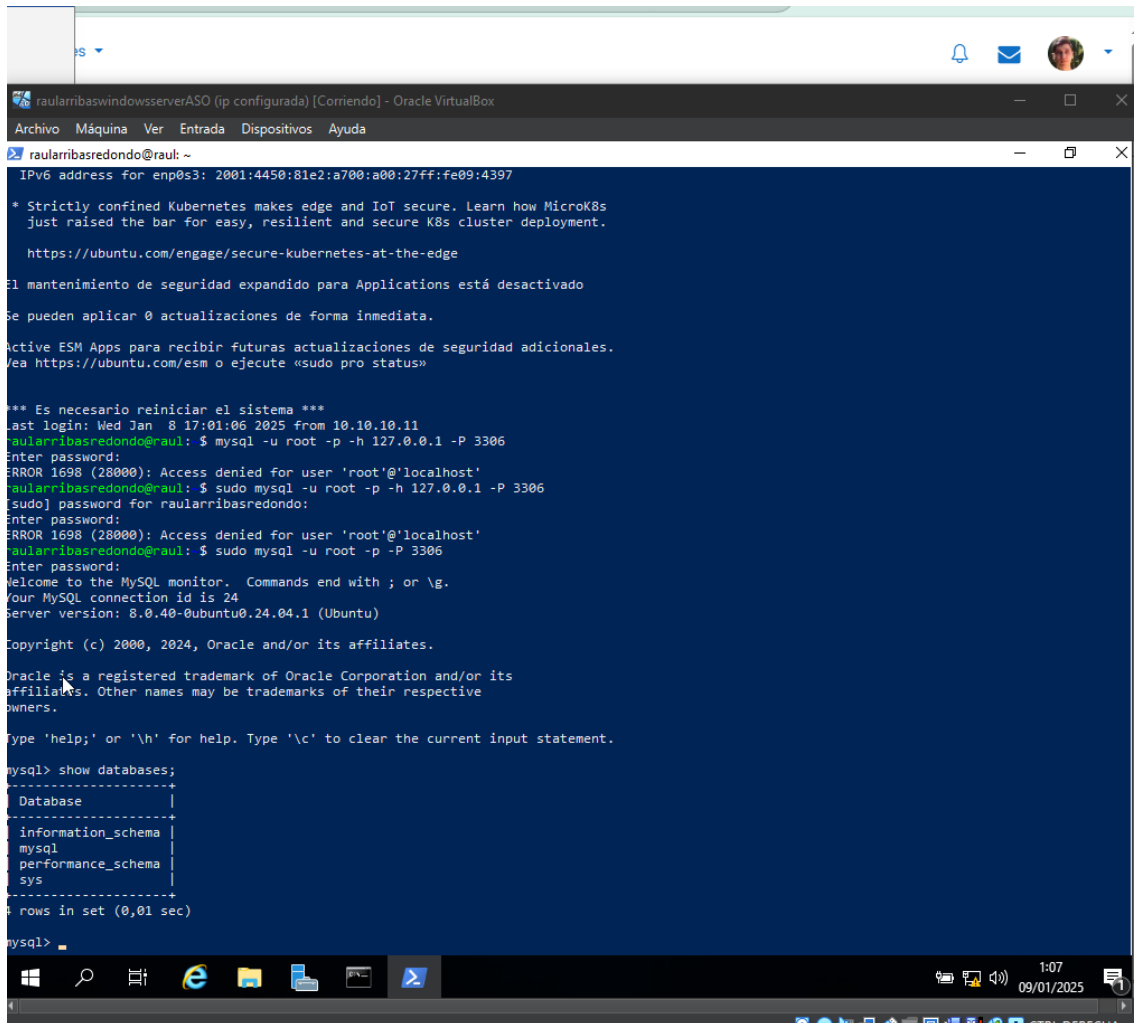
```
raularribasredondo@raul:~$ sudo mysql -u root -p -P 3306
[sudo] password for raularribasredondo:
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 20
Server version: 8.0.40-0ubuntu0.24.04.1 (Ubuntu)

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```



The screenshot shows a remote desktop window titled 'raularribaswindowsserverASO (ip configurada) [Corriendo] - Oracle VirtualBox'. Inside the window is a terminal window titled 'raularribasredondo@raul: ~'. The terminal output shows the user's IP address, a link to a Kubernetes article, and a message about security updates. The user then attempts to connect to a MySQL database using 'mysql -u root -p -h 127.0.0.1 -P 3306'. After entering the password, the user is prompted to enter the password again. The terminal then shows the MySQL monitor interface with the command 'show databases;' and the output listing the databases: information_schema, mysql, performance_schema, and sys.

Comprobamos que funciona bien la base de datos.

3 Apartado 3. Comparativa final y conclusión:

3.1 ¿Cuáles son las ventajas y desventajas de utilizar el Escritorio Remoto en comparación con la conexión SSH?

➤ Ventajas de Escritorio Remoto:

Interfaz Gráfica de Usuario (GUI):

- El Escritorio Remoto permite a los usuarios interactuar con el servidor utilizando una interfaz gráfica, lo que facilita la administración y el uso de aplicaciones que requieren una GUI, como Notepad++.
- Ideal para usuarios no técnicos o cuando se necesita acceso completo a un entorno de escritorio sin depender de la línea de comandos.

Acceso a aplicaciones específicas (RemoteApp):

- RemoteApp permite ejecutar aplicaciones específicas sin acceder completamente a un escritorio remoto. Esto es útil para aplicaciones que deben ejecutarse en el servidor, pero que no requieren que el usuario vea toda la interfaz del servidor.

Soporte para múltiples usuarios simultáneos:

- El Escritorio Remoto en servidores Windows permite que varios usuarios se conecten y trabajen simultáneamente en el mismo servidor, lo que es útil en entornos donde se necesita compartir recursos.

➤ Desventajas de Escritorio Remoto:

Consumo de recursos:

- El uso de Escritorio Remoto puede ser intensivo en recursos, especialmente en máquinas virtuales o servidores con hardware limitado, ya que cada sesión de usuario consume recursos del servidor (memoria, CPU, etc.).

Mayor carga en la red:

- El acceso remoto gráfico puede consumir un gran ancho de banda, lo que podría afectar el rendimiento si la conexión de red es inestable o limitada.

Mayor superficie de ataque:

- Dado que el Escritorio Remoto expone el servidor completo (o partes de él) al acceso remoto, podría ser más susceptible a ataques, como intentos de acceso no autorizado o explotación de vulnerabilidades de la RDP.

➤ Ventajas de SSH:

Ligereza y bajo consumo de recursos:

- SSH es más eficiente en cuanto a recursos, ya que solo proporciona acceso a la línea de comandos, sin necesidad de procesar ni renderizar una interfaz gráfica.

Conexiones seguras y cifradas:

- SSH utiliza encriptación fuerte para garantizar la seguridad de las comunicaciones, lo que hace que las conexiones sean seguras, incluso a través de redes no seguras.

Menor carga en la red:

- Debido a que solo se transmite texto y comandos, SSH consume mucho menos ancho de banda que una sesión de Escritorio Remoto, lo que lo hace más adecuado para conexiones lentas o limitadas.

➤ **Desventajas de SSH:**

Interfaz de línea de comandos:

- Requiere conocimientos técnicos para usar la línea de comandos, lo que puede resultar desafiante para usuarios no técnicos.

Sin soporte para aplicaciones gráficas:

- Aunque es posible redirigir aplicaciones gráficas mediante X11 (en Linux), SSH por sí mismo no soporta interfaces gráficas de manera nativa, lo que lo hace menos adecuado para ciertas aplicaciones que requieren GUI.

3.2 ¿Qué medidas de seguridad adicionales podrías implementar para proteger las conexiones remotas en ambos sistemas operativos?

Para Escritorio Remoto (RDP) en Windows:

Autenticación de Dos Factores (2FA):

Habilitar autenticación de dos factores para la conexión RDP puede agregar una capa extra de seguridad, requiriendo un código adicional además de la contraseña para acceder al servidor.

Configurar la política de contraseñas seguras:

Imponer contraseñas fuertes y políticas de caducidad de contraseñas para prevenir accesos no autorizados.

Usar VPN (Red Privada Virtual):

Configurar una VPN para asegurar que las conexiones RDP solo puedan realizarse dentro de una red privada, lo que reduce la exposición del servidor RDP a internet.

Filtrado de direcciones IP:

Restringir las direcciones IP que pueden conectarse a través de RDP, permitiendo solo ciertas direcciones IP de confianza.

Desactivar RDP cuando no se necesite:

Desactivar el servicio de Escritorio Remoto cuando no se esté utilizando para minimizar el riesgo de ataques.

Firewall y monitorización de seguridad:

Configurar reglas de firewall estrictas para limitar el acceso a RDP y activar la monitorización de eventos de seguridad para detectar intentos de acceso no autorizado.

Para SSH en Linux:

Cambiar el puerto por defecto (22):

Cambiar el puerto por defecto de SSH para dificultar los intentos de acceso por parte de atacantes automatizados.

Desactivar el acceso root a través de SSH:

En el archivo `/etc/ssh/sshd_config`, establecer `PermitRootLogin` no para evitar el acceso directo como root, lo que hace que los atacantes tengan que adivinar una contraseña de un usuario normal antes de escalar privilegios.

Uso de llaves SSH en lugar de contraseñas:

Desactivar la autenticación por contraseña y usar llaves SSH para autenticarse. Esto aumenta la seguridad, ya que las llaves SSH son mucho más difíciles de adivinar que las contraseñas.

Configuración de fail2ban:

Usar herramientas como fail2ban para bloquear temporalmente direcciones IP que intenten hacer múltiples intentos fallidos de inicio de sesión, protegiendo el servidor contra ataques de fuerza bruta.

Autenticación de Dos Factores (2FA):

Configurar un sistema de autenticación de dos factores para SSH, añadiendo una capa extra de protección.

Firewall:

Configurar un firewall para permitir el acceso SSH solo desde direcciones IP específicas, limitando la superficie de ataque.

Monitoreo de acceso SSH:

Monitorear regularmente los registros de acceso (/var/log/auth.log) para detectar cualquier actividad sospechosa.