



Universidad de Valladolid

CREACIÓN DE UN DIRECTORIO ACTIVO CON WINDOWS SERVER 2022

**Ingeniería informática
Tecnologías de la información y la comunicación**

Álvaro López Primo

Índice

1. Introducción

En este apartado responderemos a algunas preguntas antes de empezar con el proceso de instalación de windows server 2022.

1.1 ¿Cuáles son las principales diferencias entre Windows server 2022 y un Windows 11?

Para responder a esta pregunta, lo primero que tenemos que tener claro es que el **kernel** de ambos productos de software de Microsoft es el mismo, pero las tareas para las que están especializadas no son las mismas.

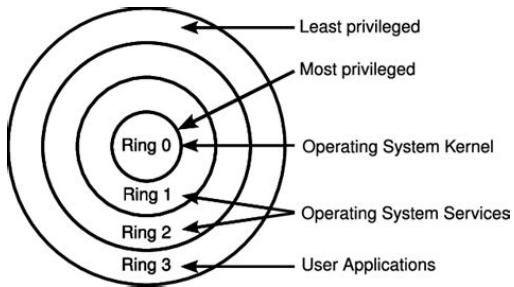


Figura 1: Distribución de los anillos dentro de un sistema operativo

Para entender esto bien, tenemos que recordar el concepto de los anillos dentro de un sistema operativo, los cuales definen la estructura de mismo, agrupando los niveles de mayor a menor privilegio.

El kernel, se encuentra en el anillo cero, es decir aquél más cercano a la CPU, por ende el que más "poder" tiene (en última instancia será el encargado de realizar las operaciones sobre el hardware del dispositivo, mediante interrupciones o diferentes métodos, en función de las necesidades de cada proceso.)

Que comparten el kernel implica que comparten el mismo *ntoskrnl.exe* (en windows este ejecutable es el manejador del kernel), la misma abstracción de hardware (planificación para el *multithreading*, *manejo de interrupciones...*)

Ahora bien, si el kernel es el mismo ¿por qué existen dos versiones diferentes de Windows? Esta pregunta se responde de manera bastante fácil, puesto que el motivo es el propósito para el que han sido diseñados estos productos de software por parte de Microsoft. El *Windows 11* es un software pensado para clientes, es decir dispositivos o hosts individuales, los cuales presentan un hardware comedido para un uso ofimático o para tareas poco pesadas (computacionalmente hablando), mientras que *Windows Server* está pensado como orquestador, es decir, para ofrecer un servicio a diferentes clientes, los cuales realizarán tareas sobre la máquina que hostea este sistema operativo, está pensado para gestionar los recursos de la máquina donde ha sido instalado así como para ofrecer servicio a múltiples hosts clientes. Para mejorar la legibilidad de esta comparativa, he decidido crear una tabla donde se muestran las principales diferencias:

Ámbito	Windows 11 (cliente)	Windows Server 2022
Licensing model	Retail/OEM por dispositivo; sin límite de conexiones	Servidor + CAL (usuario o dispositivo); valida conexiones
Roles de red	Ninguno; límite 20 conexiones SMB	AD DS, DNS, DHCP, RRAS, NLB, S2D, Storage Replica...
Seguridad por defecto	Secured-core PC optativo; HVCI manual	Secured-core Server habilitado; VBS + DMA Guard
Hyper-V	\leq 2 CPU / 2 TB RAM; sin réplica ni Live Migration	48 TB RAM por VM; réplica, nested, SR-IOV, SET
Almacenamiento kernel	NTFS + ReFS deshabilitado arranque; sin dedup	ReFS v3 block-cloning; dedup; S2D; Storage Replica
Protocolo de ficheros	SMB 3.1.1 básico	SMB 3.1.1 + SMB over QUIC (UDP-TLS 1.3)
Quantum planificador	Corto, prioriza latencia interactiva	Largo, prioriza throughput; timer 100 Hz
Interfaz por defecto	Shell moderno, Store, Widgets	Desktop Experience opcional; recomendado Server Core
Ciclo de patch	1 actualización funcional/año (H2)	Canal LTSC; 10 años soporte; parches acumulativos
Integración cloud	M365, OneDrive, Teams consumer	Azure Arc, JEA, Windows Admin Center, HCI

Cuadro 1: Comparativa entre Windows 11 y Windows Server 2022

2. Instalación de *Windows server* sobre la máquina virtual

En el guión de la práctica se comenta que hay diversas opciones para la instalación del software, las máquinas virtuales propias de la asignatura (mediante matrix) o una versión *offline* que se basa en la instalación del software en una partición o empleando los recursos de mi host. Finalmente me he decantado por la máquina virtual puesto que en mi pc no tengo el espacio suficiente como para poder hacer una partición dedicada para la práctica, por lo que a partir de ahora, todas las capturas de *Windows Server* serán también capturas de la máquina virtual de la asignatura.

2.1 Formateo de la máquina virtual

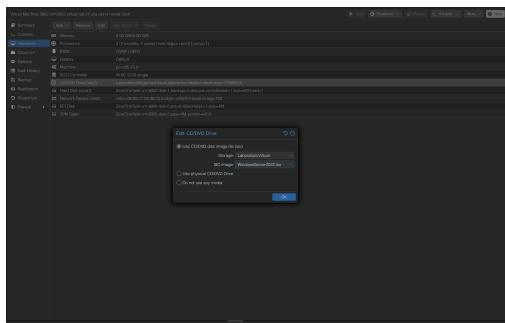


Figura 2: Seleccionaremos la ISO de Windows Server.

Anteriormente la máquina virtual de la asignatura ha sido empleada para realizar los guiones de prueba sobre un Ubuntu, por lo que el primer paso es formatear la máquina e instalar *Windows Server*. Como se indica en el guión, lo que haremos será entrar en la BIOS de la máquina virtual y en el *Boot Menu* elegir la opción de *UEFI QEMU DVDROM*, con esto lo que haremos será iniciar el instalador de *Windows Server* desde el CD y podremos comenzar con el proceso de instalación.

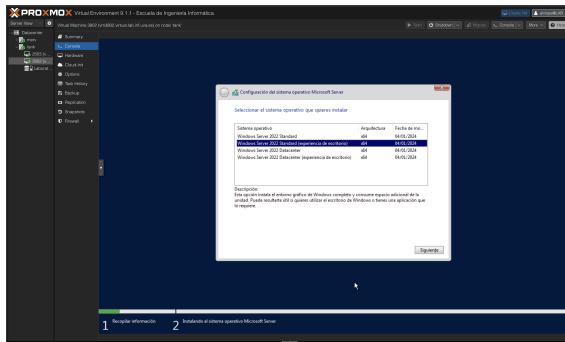


Figura 3: Seleccionamos la versión de *Windows Server* que se nos indica en el enunciado

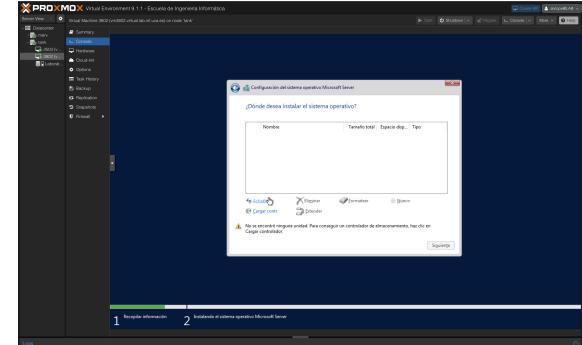


Figura 4: Problema con la instalación, no reconoce que haya discos duros en la máquina virtual.

Una vez que hemos decidido qué versión de *WS* vamos a instalar, y que deseamos borrar todo el disco duro sin mantener particiones o datos de las prácticas previas, lo que nos encontramos es que *no reconoce ninguno disco*, esto no es posible puesto que previamente hemos trabajado con esta máquina virtual y teníamos 60GB de espacio en la misma, por lo que me decanto con un problema de *drivers* es decir los "traductores" que permiten al OS emplear el hardware del pc.

En el guión de la práctica ya se mencionaba esta posibilidad por lo que realizaremos los pasos que ahí se muestran

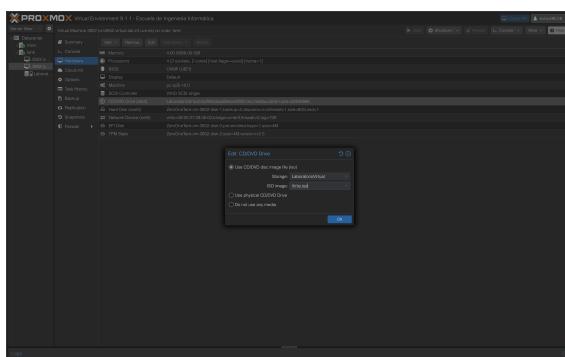


Figura 5: Cargamos el disco (en el mismo menú en el que hemos seleccionado la imagen del *WS*)

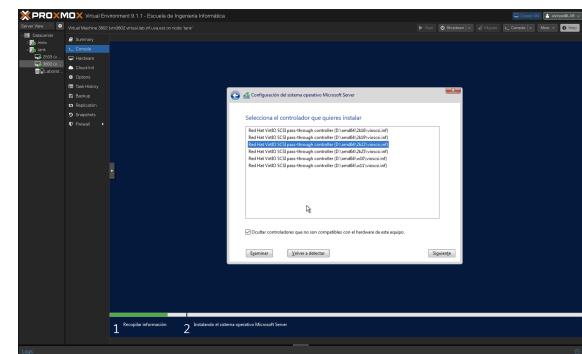


Figura 6: Una vez que ya tenemos esto, volvemos a la parte de Console dentro de promox y buscamos los drivers para los discos duros

En las opciones de driver para el disco duro, tenemos diferentes versiones, lo que debemos hacer es seleccionar la que sale seleccionada puesto que se diferencian en el año en el que han sido compilados. En nuestro caso como estamos instalando la versión de 2022, para que el driver de los menores problemas de compatibilidad posibles seleccionamos la *2k22*

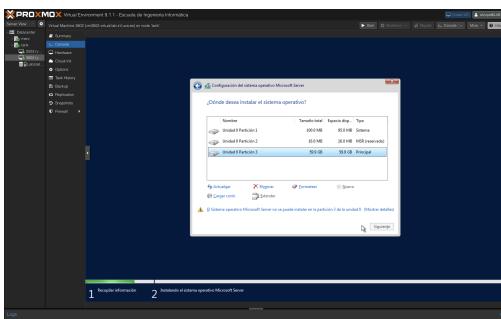


Figura 7: Nuevo esquema de particiones

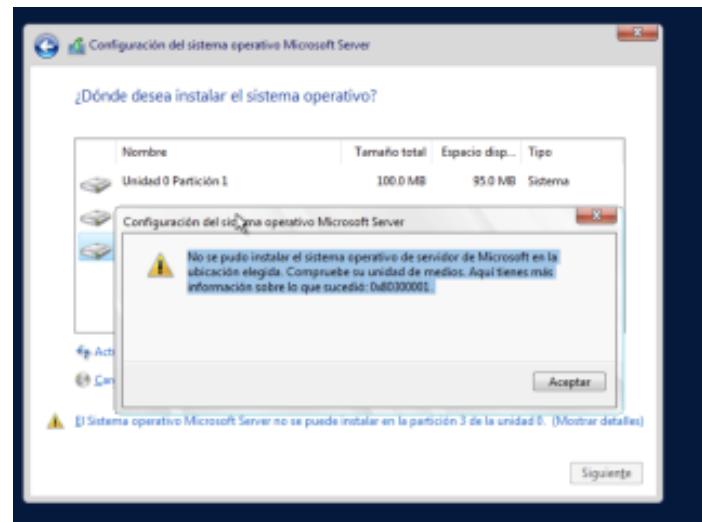


Figura 8: Error 0x80300001 = “Windows no ve el disco que acabas de seleccionar”, reconoce el disco porque está cargado en RAM pero realmente no puedo isntalar el WS, por lo que toca volver a ir al apartado de hardware y seleccionar la ISO de *Windows Server* para que ya reconozca el disco y se pueda proceder con la instalación

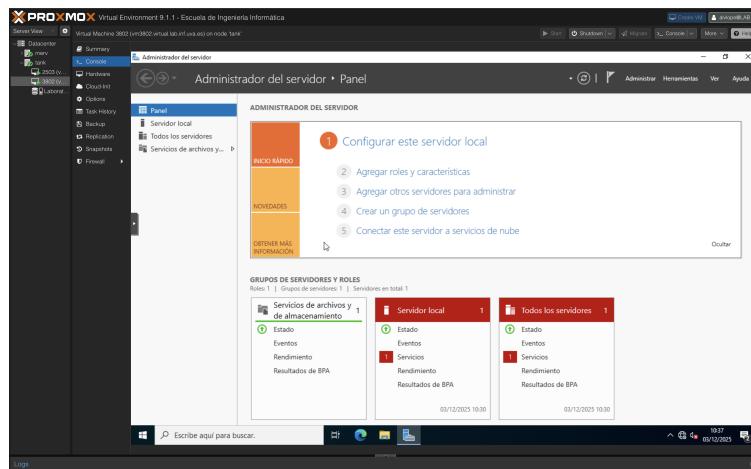


Figura 9: La contraseña de la cuenta de administrador (la cuál va a tener como *password*: *wmLG9N7n*)

2.2 Instalación de drivers

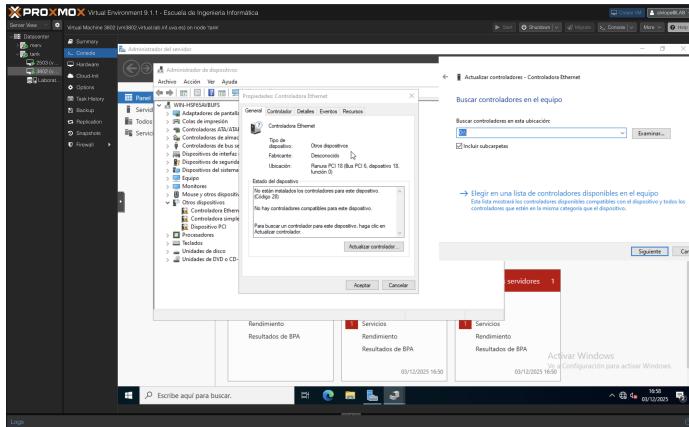


Figura 10: Seleccionaremos los drivers de red.

Los dispositivos que nos fallaban se encargan de:

- *Controladora Ethernet (Ethernet Controller)*: Necesario para la comunicación con la red.
- *Controladora simple de comunicaciones PCI (Simple Communications Controller PCI)*: Relacionado con las tareas de mantenimiento y de administración remota.
- *Dispositivo PCI (PCI Device)*: Dedicado a la comunicación entre el dispositivo conectado por PCI (una tarjeta de sonido por ejemplo).

2.3 Administrador del servidor

Una vez que tenemos todo instalado (de momento), vamos a investigar qué es el panel de *Administrador del servidor*.

2.4 Servidor Local

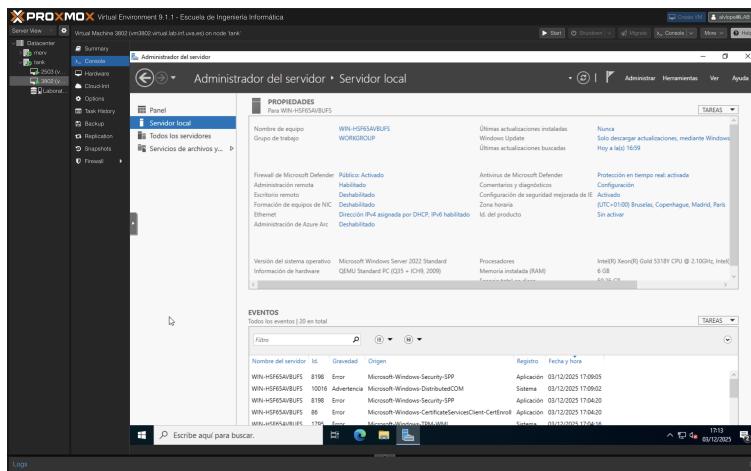


Figura 11: En este menú vemos toda la información relacionada con nuestra máquina virtual.

En esta imagen podemos mirar que tenemos un procesor típico de servidores, un firewall activado y la función de escritorio remoto la tenemos desactivada.

Una vez que ya hemos finalizado la instalación, me he dado cuenta de que faltan funcionalidades como puede ser la de red. Al igual que hicimos para que el instalador reconociese los discos, volvemos a cargar *Virtio.ISO* en el apartado de *hardware*. Posteriormente vamos al **Administrador de dispositivos** y actualizamos los controladores de todos aquellos dispositivos que aparezcan con una exclamación (drivers de red, dispositivos de comunicación simples PCI y dispositivo PCI). El proceso es el mismo

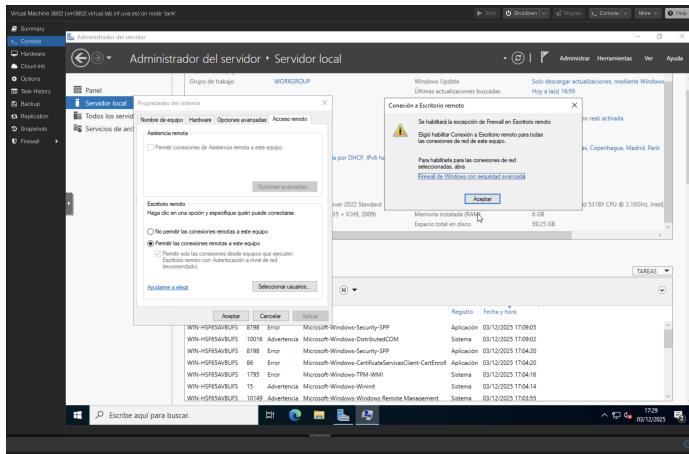


Figura 12: Habilitamos el servicio de escritorio remoto

2.4.1 Cambio del nombre del equipo y del grupo de trabajo

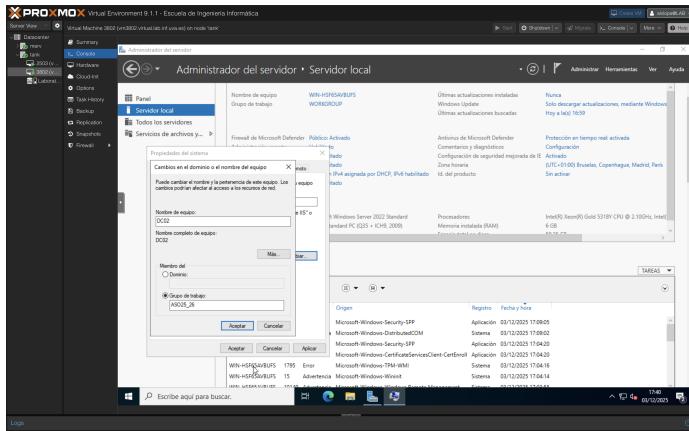


Figura 13: Cambiamos el nombre de la máquina y el del grupo de trabajo

2.4.2 Información sobre la red

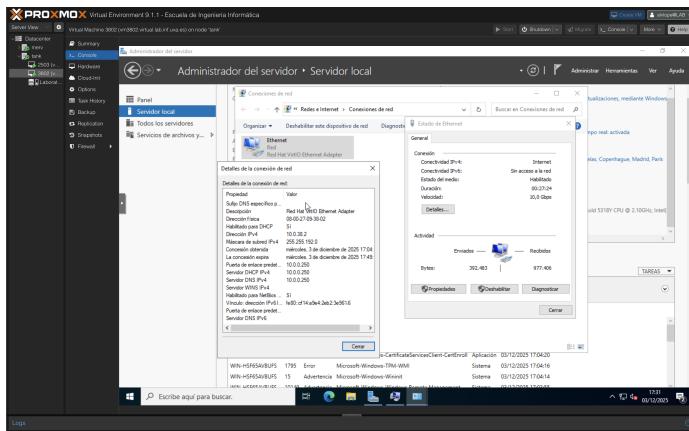


Figura 14: Información de la dirección IPv4 de nuestra máquina

Esta es la versión propietaria de *Microsoft* llamada **RDP** (*Remote Desktop Protocol*) de tal manera que permite a nuestra máquina establecer una conexión remota con otro pc que se encuentre dentro de una red, permitiéndote el acceso total al mismo, lógicamente esto está desactivado por motivos de seguridad, pero como nosotros vamos a administrar ese servicio **procedemos a habilitar el escritorio remoto**. con el consecuente aviso de que se va a abrir una excepción en el firewall para que se pueda emplear este servicio.

¿Qué significa que hagamos estos cambios? El nombre del Equipo es como se nos va a poder identificar dentro de la red, mientras que el *miembro de* indica a qué grupo de trabajo va a pertenecer nuestro servidor. En mi caso, como no existe ninguno, lo que hace el OS internamente es crear un grupo de trabajo vacío y se une a él.

Siguiendo el guión tenemos que mirar la información de la dirección IP de nuestra máquina, si nos fijamos en la captura, sólo tenemos configurada un dirección IPv4, la cual es la **10.0.38.2**, pero ojo, esta es una dirección IP privada es decir, identifica a la máquina virtual pero dentro de la subred (LAN) de la UVa(más probablemente de los servidores de la escuela de ingeniería informática), no de cara a internet, de tal manera que para poner conectarnos a internet con esta IP el cortafuegos deberá traducir esta IP privada a una pública que sea accesible desde Internet, para lo cuál utiliza la máscara de subred, para identificar la red y el dispositivo dentro de la misma.

2.4.3 Cambio de puerto del protocolo RDP

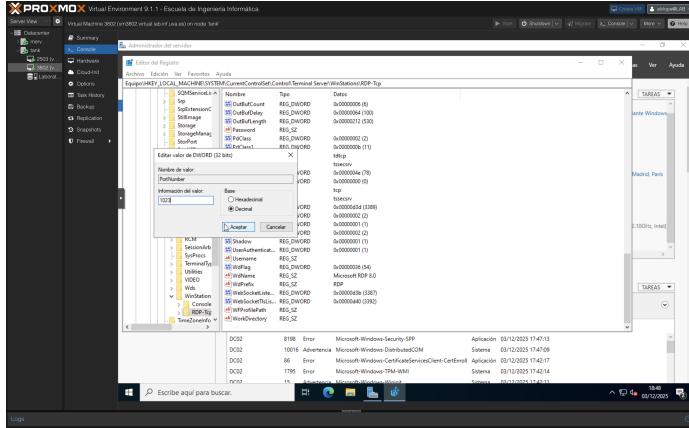


Figura 15: Cambio de puerto del *RDP*

Como se nos indica en el guión, el *RDP*, emplea *tcp* el puerto 3389, lo cual *WKP* (*Well Known Port*) (puertos cuyo uso está estandarizado para ciertos servicios) puesto que se sale del rango de los 1025 (0-1024) primeros puertos, pero sí es conocido en los sistemas de *Microsoft*.

El que no sea uno de los *WKP* es un arma de doble filo, porque permite a un usuario sin permisos de administrador (el equivalente a *sudo* en el OS de Linux) conectarse de manera remota a nuestra máquina, lo que hace que si se sufre un ataque nuestro *host* que expuesto de manera sencilla.

Lo que sí que es una ventaja es que el OS no tiene que emplear una multiplexión de puertos para redirigir las peticiones en caso de que otro servicio requiera el uso de dicho puerto.

En mi caso, prefiero mover el servicio a un *WKP* para forzar que el usuario tenga permisos de administrador o de *sudo* para realizar la conexión remota

2.5 Configuración del firewall

Si recordamos, en la el servidor local (11) tenemos habilitado un firewall, vamos a proceder a configurarlo, para permitir conexiones.

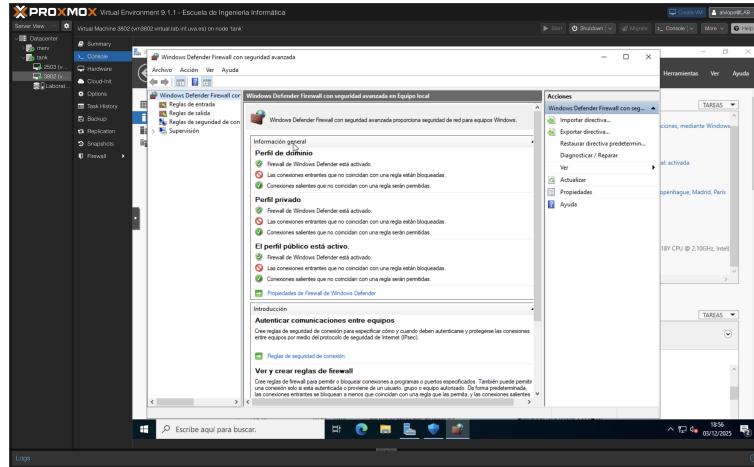


Figura 16: Entramos en la opción de firewall y nos vamos a opciones avanzadas

2.5.1 Puerto 22→ tcp/22

Este puerto es el estándar para permitir conexiones mediante el protocolo *SSH (Secure Shell)*, el cual permite establecer conexiones remotas entre *hosts*, traducido sería decir que nos permite conectarnos de manera remota a otro dispositivo.

Por defecto el Firewall no permite conexiones a ese puerto.

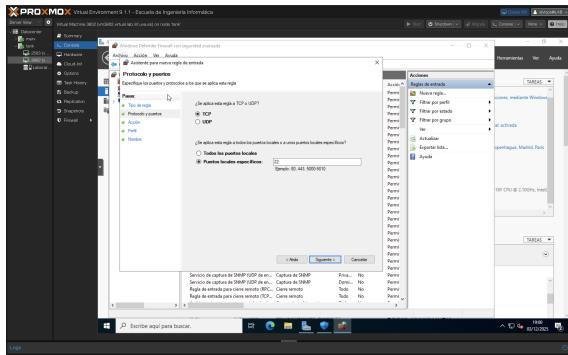


Figura 17: Primer paso para establecer la regla, decir que vamos a aplicarla sobre un puerto

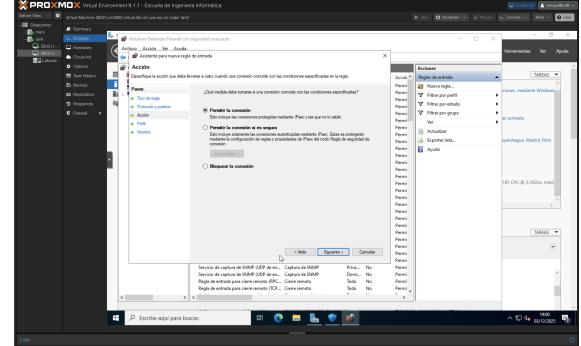


Figura 18: Especificamos el protocolo y el puerto, en nuestro caso tcp (puesto que a ser seguro necesita establecer el *handshake* y confirmación de recepción) y el puerto 22

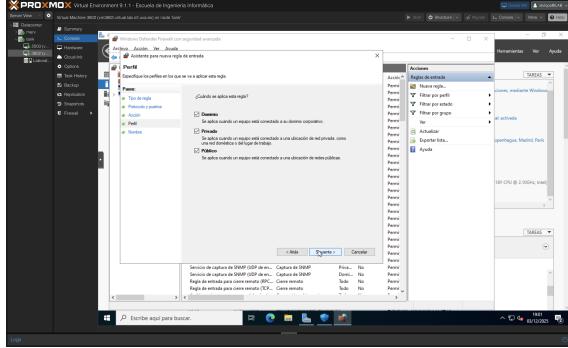


Figura 19: Establecemos el dominio de la norma, para la máquina virtual, en cualquier caso

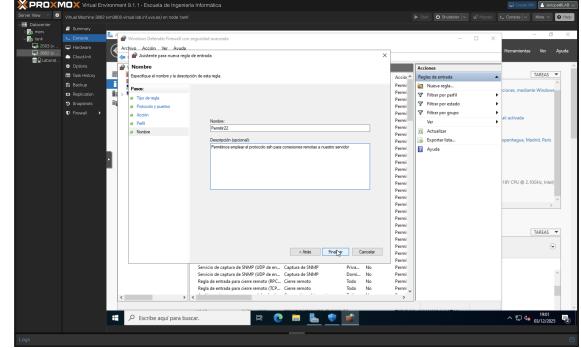


Figura 20: Le añadimos un nombre y descripción para poder localizarla una vez que se ha añadido

2.6 Prueba de conexión remota empleando la aplicación Escritorio Remoto

3. Configuración del *HIPERVISOR HYPER-V*

3.1 ¿Qué son los Roles en un *Windows Server*

En el contexto de un servidore son el **conjunto primario de servicios de software y configuraciones**, un ejemplo de esto es el *Active Directory* o un *servidor DHCP*. De tal manera que hace que el servidor se encargue de una tarea específica.

3.2 ¿Qué son las características en *Windows Server*

Son programas o herramientas que o bien complementan las funciones de un *Role* o que proporcionan herramientas las cuales son útiles de manera independiente. En el contexto de un *WS* tenemos como ejemplo el *.NET Framework* que es esencial para algunas herramientas de diagnóstico o gestión.

Ojo, que estas características son para los administradores del sistema no tienen por qué ser para los usuarios del mismo.

3.2.1 Agregar roles y características a nuestro servidor

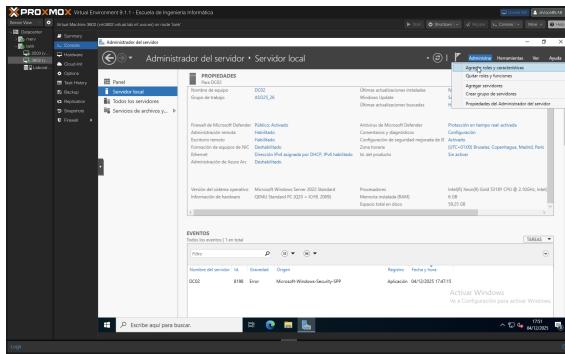


Figura 21: Menú para seleccionar los *roles* y *características*

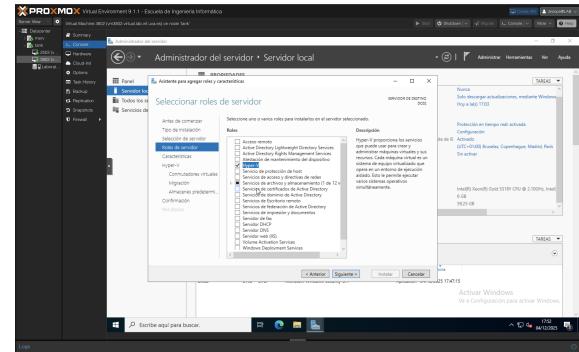


Figura 22: Instalamos las características que se nos indica en el guión tras seleccionar las instalaciones por roles o características

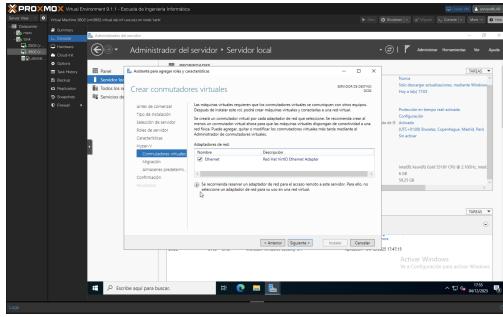


Figura 23: Seleccionaremos el comutador como indica el guión.

Hyper-V es una tecnología de virtualización de Windows, que permite alojar y ejecutar otras *Virtual Machines*(VM)

El **comutador de red** funciona como un *switch* permitiendo que las máquinas virtuales se comuniquen entre sí y que las mismas tengan conexión a internet (empleando nuestra tarjeta de red para ello).

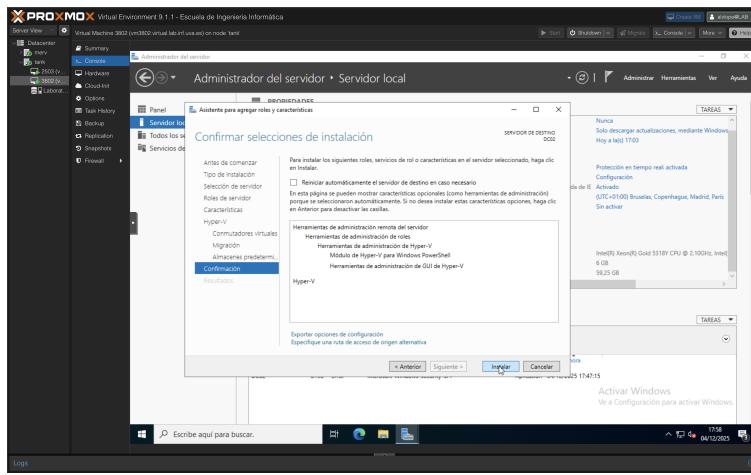


Figura 24: Un resumen de lo que vamos a instalar en nuestro WS, tras la instalación toca reiniciar la máquina para que se apliquen estos cambios.

3.2.2 Importar VM disponibles

Tras permitir la *detección de redes* (es simplemente abrir el menú y darle a la opción) permitiendo ser detectado por otros dispositivos de la red y poder acceder a tus recursos compartidos, aunque esto sea un

submenú dentro de la carpeta de red, en verdad lo que estamos haciendo es configurar el *firewall* y a la vez expone a nuestra máquina a ser descubierta por una exploración básica de la red.

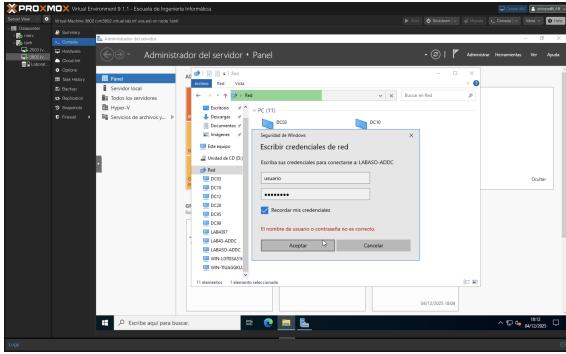


Figura 25: Introducimos las credenciales de acceso para acceder a esta máquina Usuario: usuario y contraseña: Passw0rd

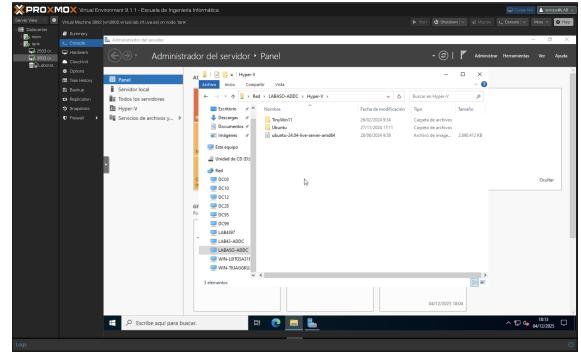


Figura 26: Los directorios para las máquinas virtuales, como se indica en el guión copiamos *Tiny-Win11* a nuestro equipo

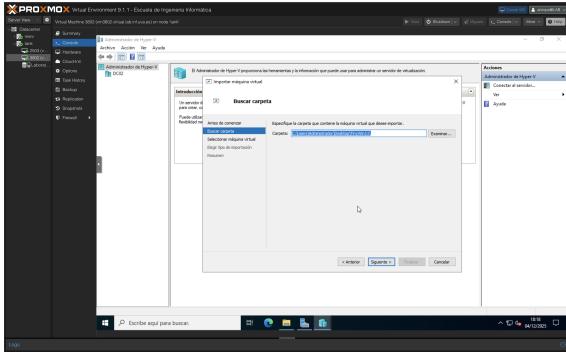


Figura 27: Vamos al menú de administrar el Hyper-V click derecho sobre nuestro grupo e importar máquina virtual

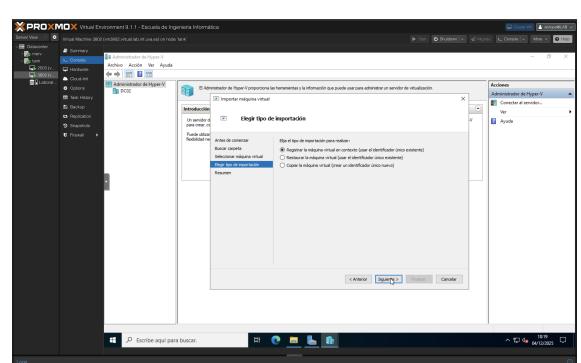


Figura 28: Las opciones que nos da para importar la VM

Si nos damos cuenta, aparecen varias opciones para importar la máquina virtual, voy a explicarlas poco a poco:

- Registrar la máquina virtual (usar el identificador único existente): Reutiliza la *VM* exactamente como fue exportada, es decir, la copiamos tal cual.
- Restaurar la máquina virtual (usar el identificador único existente): Reemplaza una *VM* existente por una nueva versión, sobreescribiendo la máquina antigua si es que comparten el mismo *ID*
- Copiar la máquina virtual (crear un identificador único nuevo): Crea una nueva copia de la *VM* (creando un nuevo identificador único), de tal manera que la clonamos para usarla como dos instancias diferentes.

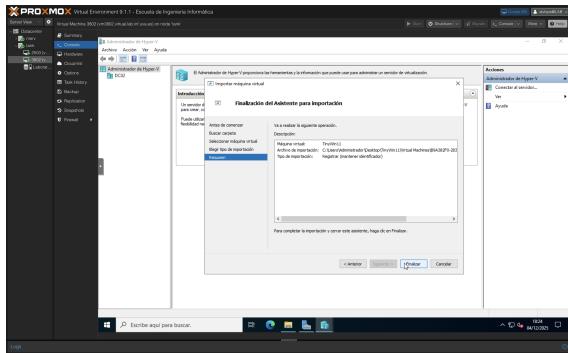


Figura 29: Se ha finalizado la instalación

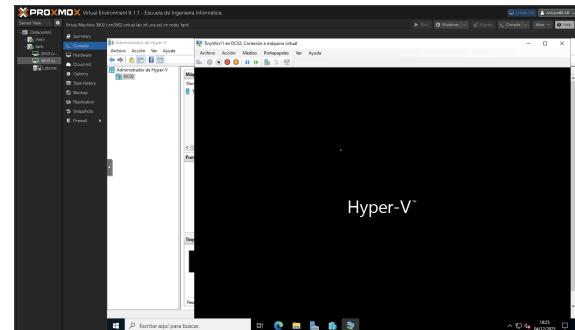


Figura 30: Damos a conectar la máquina virtual desde el menú de nuestra máquina en el administrador de *hyper-v*

3.2.3 Instalación de la VM con Ubuntu

Vamos a repetir el proceso, copiamos la imagen desde la máquina de red y volvemos al menú de *hyper-v*

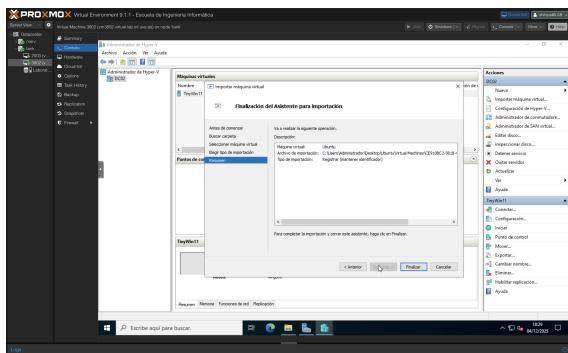


Figura 31: Comenzamos el proceso de importar la máquina virtual de Ubuntu

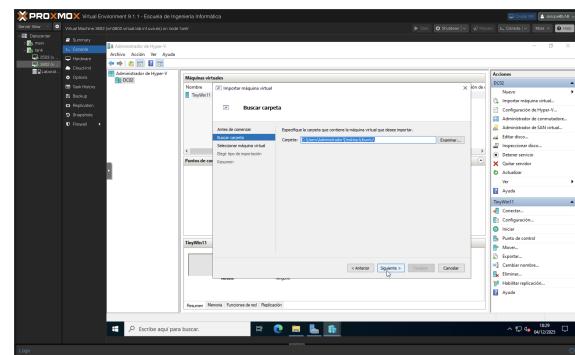


Figura 32: Damos a conectar la máquina virtual desde el menú de nuestra máquina en el administrador de *hyper-v*

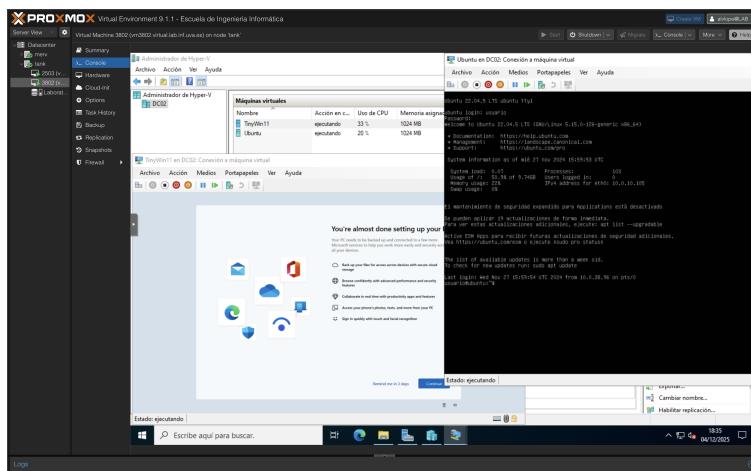


Figura 33: Ambas VM corriendo en el el WS

3.3 Ips de la máquinas virtuales en el servidor

```
last login: Mon Nov 27 15:59:54 UTC 2024 from 10.0.88.96 on pts/0
usuario@ubuntu:~$ ip addr
1: lo</loopback,00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 brd 127.0.0.1 scope host lo
        valid_lft forever preferred_lft forever
2: eth0<eth0:00:0c:29:11:15 brd 00:00:00:00:00:00 state UP group default qlen 1000
    link layer brd 00:0c:29:11:15:ff
    inet 10.0.10.14/18 brd 10.0.10.255 metric 100
        brd 10.0.10.14
        valid_lft 1690sec preferred_lft 1690sec
    link layer brd 00:0c:29:11:15:ff
    inet 10.0.10.14/18 brd 10.0.10.255 metric 100
        brd 10.0.10.14
        valid_lft forever preferred_lft forever
usuario@ubuntu:~$
```

Figura 34: Ejecución del comando `ip addr` para sacar la dirección ip de la VM

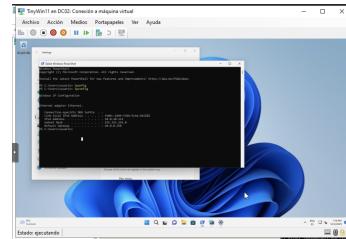


Figura 35: Dirección IP de la máquina con Windows

3.4 Comunicación entre VMs dentro del servidor

3.4.1 Configuración previa en la máquina virtual Windows 11

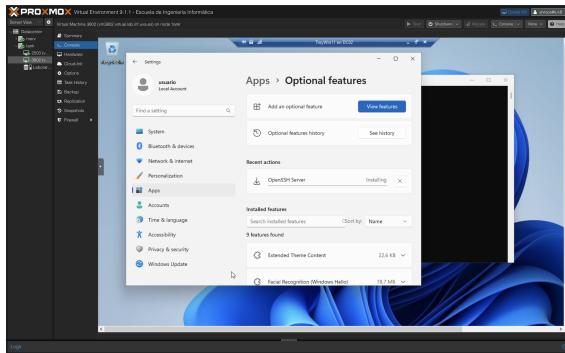


Figura 36: Instalacion del servicio openssh para windows

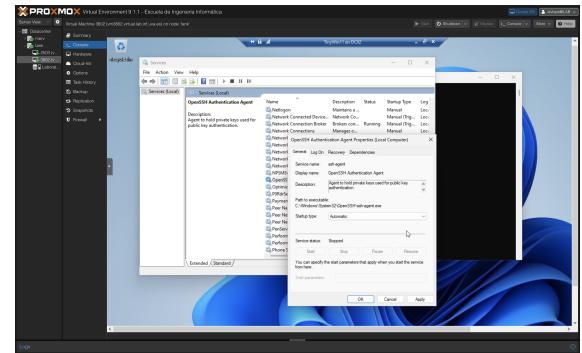


Figura 37: Establecemos que el servicio se inicie de manera automática con la máquina.

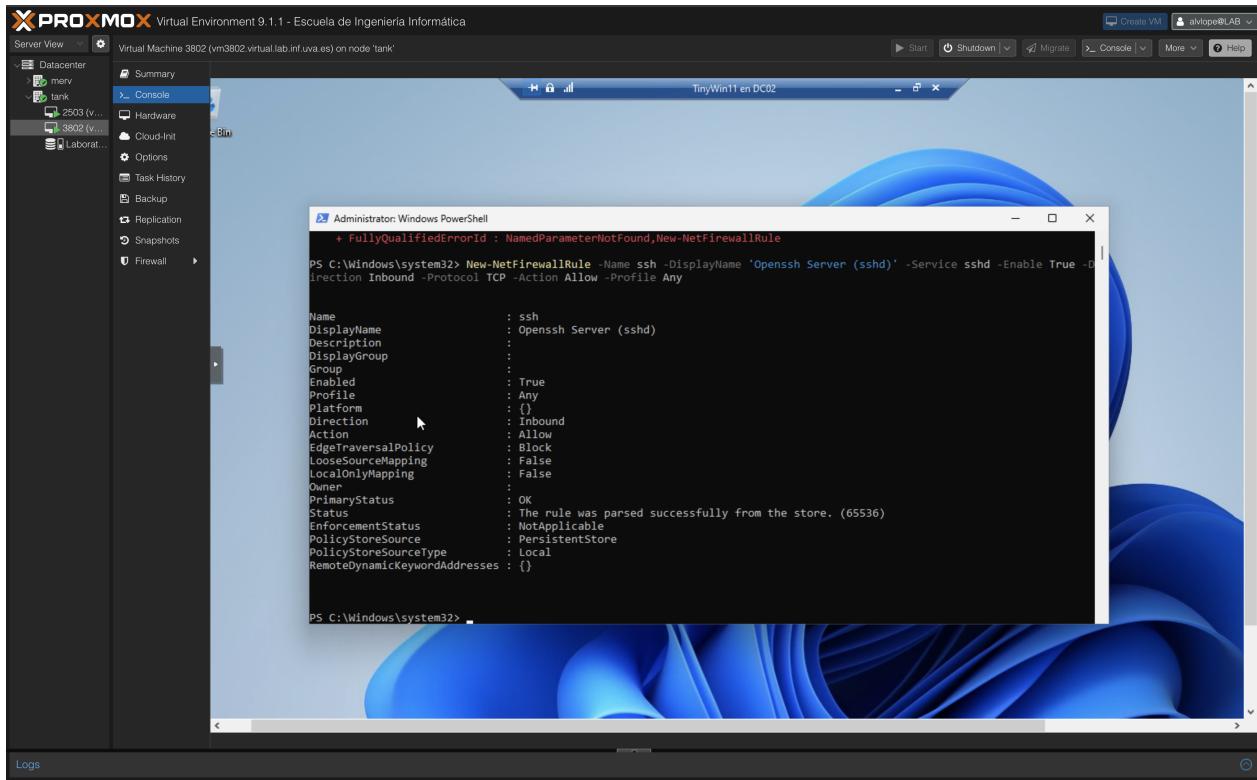


Figura 38: Aplicamos esta regla en el firewall para que permita el tráfico por el puerto 22.

3.4.2 *sshs* entre vms

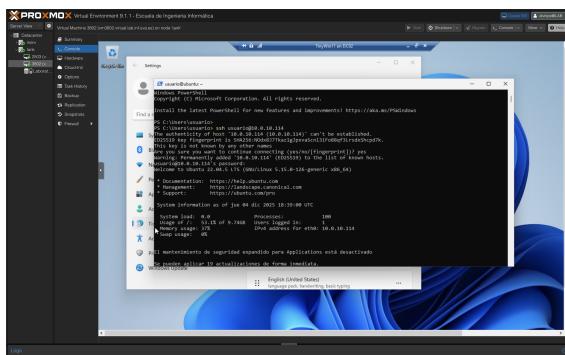


Figura 39: ssh de la máquina Ubuntu a la de Windows11

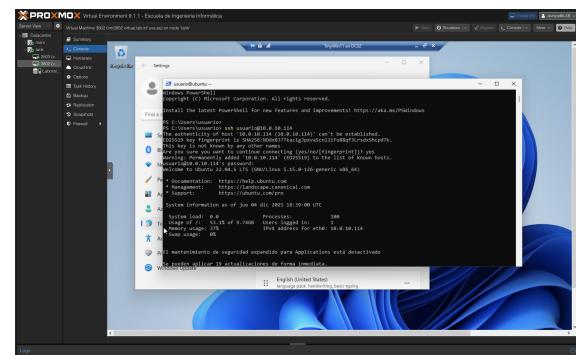


Figura 40: ssh de la máquina Windows 11 a la Ubuntu

No he conseguido hacer un ping de la máquina ubuntu a la windows, por ninguno de los puertos y no me devuelve conectividad

3.5 Comunicacion desde el servidor a las dos VMs

En la máquina con ubuntu es tan sencillo como hacer un ssh desde el *PowerShell* con permisos de administrador, sin embargo como no he conseguido hacer funcionar el servicio *OpenSSH* he empleado la conexión al escritorio remoto de windows (habilitando previamente esta característica en la máquina virtual.)

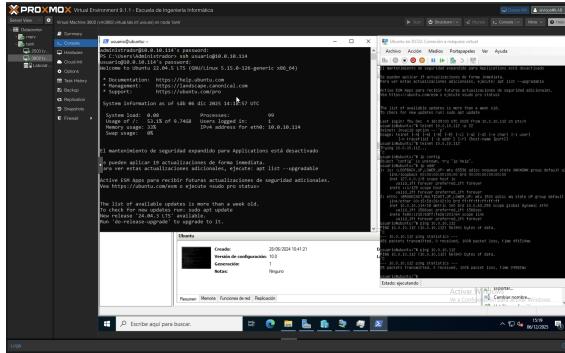


Figura 41: ssh desde el WS a la VM con Ubuntu

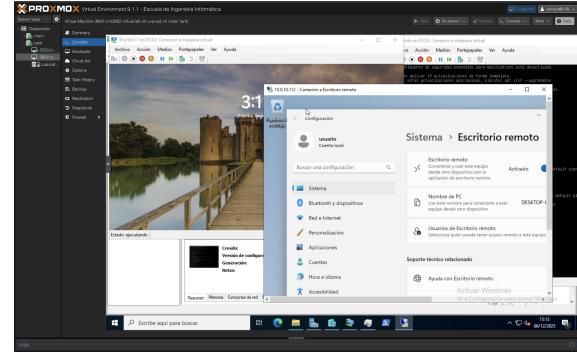


Figura 42: Conexión del escritorio remoto desde nuestro WS a la máquina virtual de Windows

3.6 Creación de una red interna para las máquinas virtuales anidadas

3.6.1 Creación de un comutador virtual

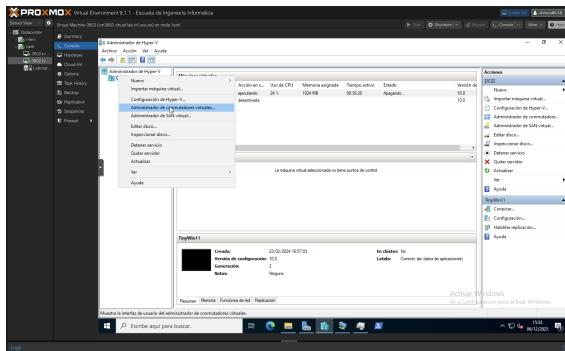


Figura 43: Tenemos que entender que los comunicadores, son un equivalente a las interfaces de red

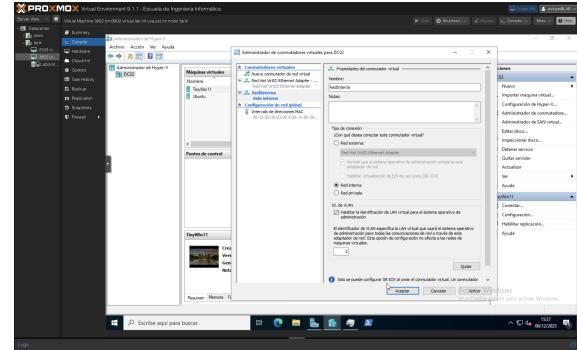


Figura 44: Creamos un comunicador que sólo va a funcionar como una red interna para nuestras VMs

3.6.2 Propiedades de la red

Nombre:	vEthernet (RedInterna)
Descripción:	Hyper-V Virtual Ethernet Adapter #2
Dirección física (MAC):	00:15:5d:26:02:02
Estado:	Operativo
Unidad de transmisión máxima:	1500
Dirección IPv4:	169.254.151.57/16
Dirección IPv6:	fe80::49a:1c59ff:fe23:a88%31/64
Servidores DNS:	fec0:0:ffff:1%1 (sin cifrar) fec0:0:ffff:2%1 (sin cifrar) fec0:0:ffff:3%1 (sin cifrar)
Conectividad (IPv4/IPv6):	Conectado a red desconocida, no hay tráfico

Figura 45: La direcciónIP del comunicador que hemos creado

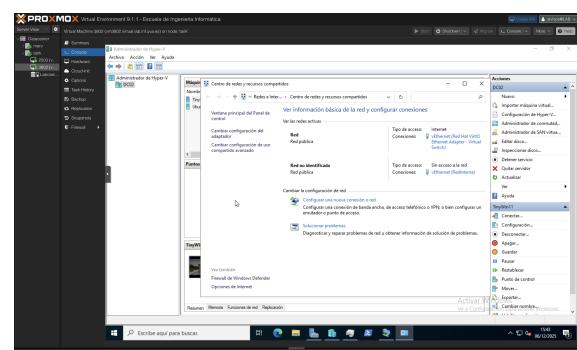


Figura 46: Si nos vamos al apartado de recursos compartidos, vemos que se ha creado una nueva conexión de red.

3.6.3 Cambio del adaptador de red

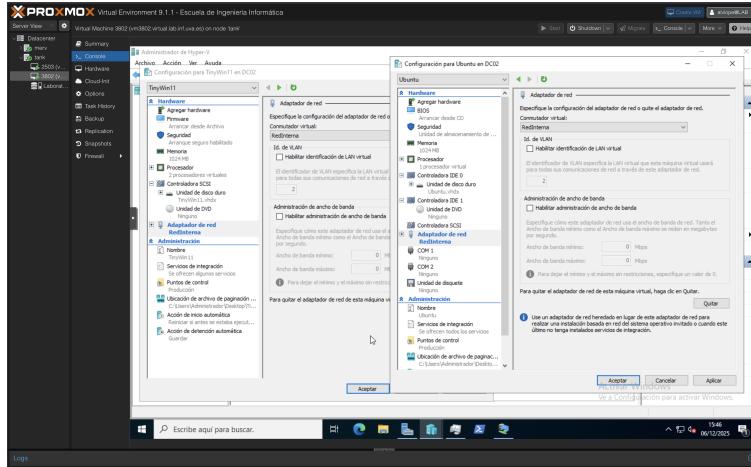


Figura 47: Establecemos el nuevo adaptador como el predefinido para las dos máquinas virtuales

3.6.4 Configuración del adaptador de red en las VMs

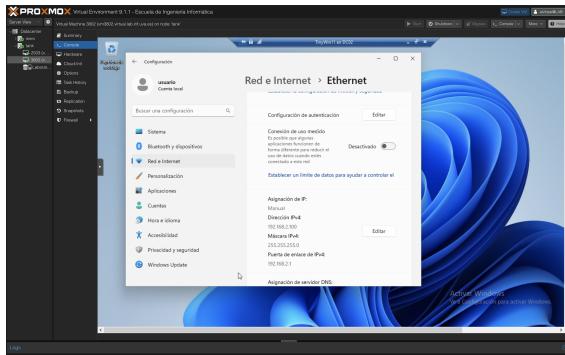


Figura 48: Configuramos la ip estática de la máquina a mano, para que ya no consiga la ip del servidor DHCP sino de lo que le acabamos de escribir

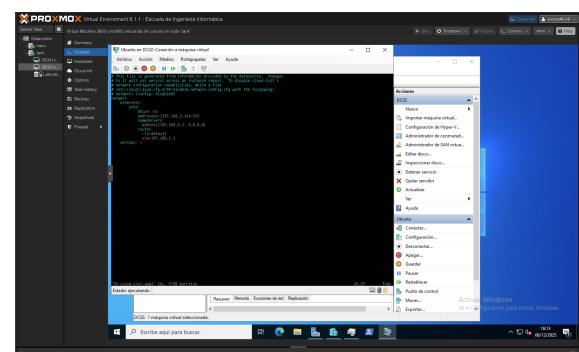


Figura 49: editamos el fichero para establecer los mismos ajustes a mano que en el windows, tras esto debemos aplicar el comando `netplay apply`

3.7 Problema con el *netplay apply*

```
pi@raspberrypi: ~ $ cat /etc/netplan/50-cloud-init.yaml
#cloud-init: &lt;comment>generated by cloud-init</comment>
#cloud-init: &lt;comment>version: 0.7.2</comment>
#cloud-init: &lt;comment>last modified: 2015-05-20 11:45:18.000000+0000</comment>
#cloud-init: &lt;comment>md5sum: 5a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a</comment>
#cloud-init: &lt;comment>source: /etc/cloud/templates/50-cloud-init.yaml</comment>
#cloud-init: &lt;comment>original source: /usr/share/cloud-init/templates/50-cloud-init.yaml</comment>

ethernets:
  eth0:
    dhcp4: no
    addresses:[192.168.2.1, 8.8.8.8]
    routes:
      - to: default
        via: 192.168.2.1
version: 2

pi@raspberrypi: ~ $ sudo netplan apply
pi@raspberrypi: ~ $
```

"50-cloud-init.yaml": 16, 5748 escritos
pi@raspberrypi: ~ \$ sudo netplan apply
pi@raspberrypi: ~ \$ etcd-mirror:50-cloud-init.yaml:1113: Invalid YAML: inconsistent indentation:
 nameservers:
pi@raspberrypi: ~ \$

Figura 50: Al ejecutar el comando me sale ese error.

3.7.1 Solución del problema en el fichero *yaml*

```
pi@raspberrypi: ~ $ cat /etc/netplan/50-cloud-init.yaml
#cloud-init: &lt;comment>generated by cloud-init</comment>
#cloud-init: &lt;comment>version: 0.7.2</comment>
#cloud-init: &lt;comment>last modified: 2015-05-20 11:45:18.000000+0000</comment>
#cloud-init: &lt;comment>md5sum: 5a2a2a2a2a2a2a2a2a2a2a2a2a2a2a2a</comment>
#cloud-init: &lt;comment>source: /etc/cloud/templates/50-cloud-init.yaml</comment>
#cloud-init: &lt;comment>original source: /usr/share/cloud-init/templates/50-cloud-init.yaml</comment>

ethernets:
  eth0:
    dhcp4: no
    addresses: [192.168.2.1, 8.8.8.8]
    routes:
      - to: default
        via: 192.168.2.1

pi@raspberrypi: ~ $
```

Figura 51: Corrijo los errores de indentación y sintaxis y ejecuto el comando para que se pueda aplicar el cambio.

```
pi@raspberrypi: ~ $ ip addr show eth0
1: eth0: <NO-CARRIER> brd 00:0c:29:1e:00:00 state DOWN group 1000
    link/ether 00:0c:29:1e:00:00 brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.10/24 brd 192.168.2.255 scope global eth0
        link-layer MAC address 00:0c:29:1e:00:00
        brd ff:ff:ff:ff:ff:ff
        valid_lft forever preferred_lft forever
    unicast ff:ff:ff:ff:ff:ff dev eth0 scope link
        link-layer MAC address ff:ff:ff:ff:ff:ff
        brd ff:ff:ff:ff:ff:ff
        valid_lft forever preferred_lft forever
pi@raspberrypi: ~ $
```

Figura 52: Ejecuto el comando *ip addr show eth0* para ver que ahora la ip de la máquina Ubuntu es la que deseo tener de manera estática.

3.8 Conexiones entre las máquinas con el cambio de ip

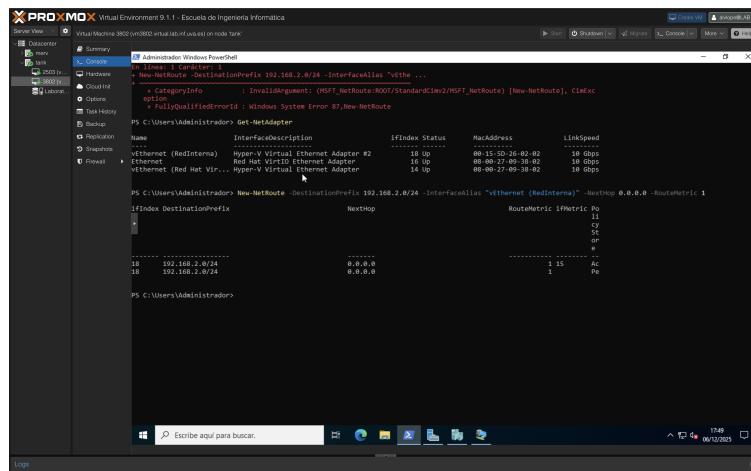


Figura 53: Creamos la regla que hace que todo lo que vaya hacia la red interna de nuestras máquinas virtuales pase por la interfaz que hemos creado en el *Hyper-V*

Adicionalmente me doy cuenta de que el adaptador no tiene una red definida, por lo que la defino con los siguientes comando en *PowerShell* como administrador

```
New-NetIPAddress -InterfaceAlias "vEthernet (RedInterna)" IPAddress 192.168.2.1 -PrefixLength 24
Set-NetConnectionProfile -InterfaceAlias "vEthernet (RedInterna)" NetworkCategory Private
```

3.8.1 Comunicación entre el WS y las VM

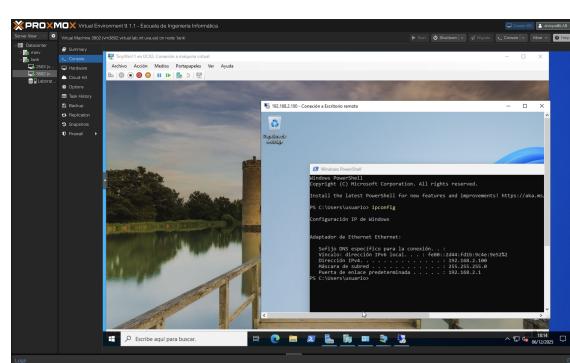


Figura 54: Establezco el *RDP* en la dirección 192.168.2.100

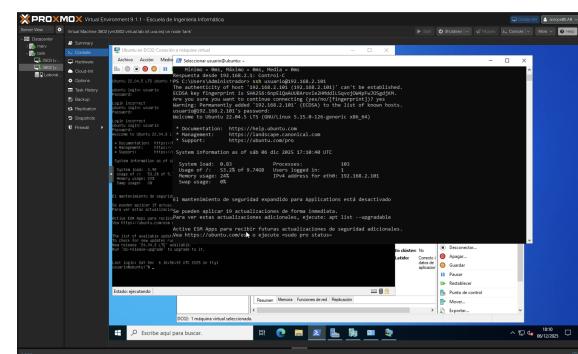
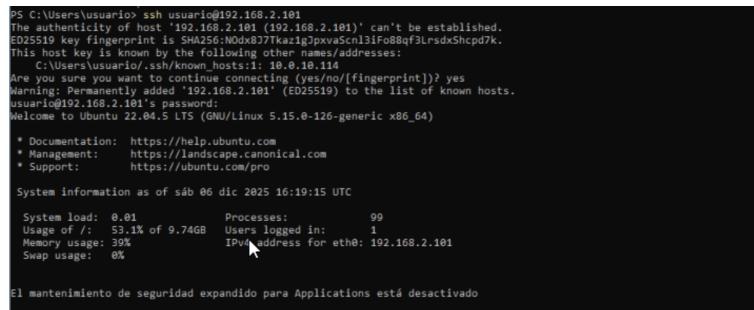


Figura 55: Ejecuto el comando *ip addr show eth0* para ver que ahora la ip de la máquina Ubuntu es la que deseo tener de manera estática la cual es 192.168.2.101.

3.8.2 Conexión de la máquina Ubuntu a la windows



```
PS C:\Users\usuario> ssh usuario@192.168.2.101
The authenticity of host '192.168.2.101 (192.168.2.101)' can't be established.
ED25519 key fingerprint is SHA256:Nowx837TkaZigpxw5cnl3lFo88qf3Lrsdx5hcdp7k.
This key is known.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.2.101' (ED25519) to the list of known hosts.
usuario@192.168.2.101's password:
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.0-126-generic x86_64)

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

System information as of sáb 06 dic 2025 16:19:15 UTC
System load: 0.01      Processes:          99
Usage of /: 53.1% of 9.74GB  Users logged in:   1
Memory usage: 39%
Swap usage:  0%
El mantenimiento de seguridad expandido para Applications está desactivado
```

Figura 56: Conexión ssh desde la máquina windows a la máquina con Ubuntu