**S.I.G.P.D.**

Administración de sistemas operativos

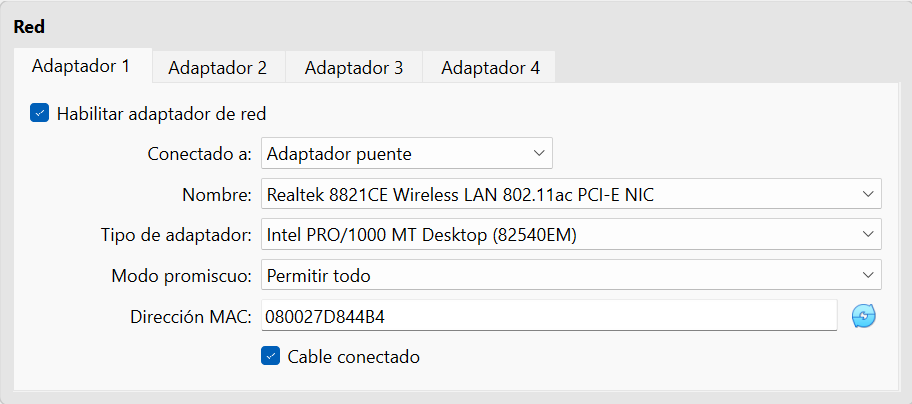
VifraSoft

| **Rol** | **Apellido** | **Nombre** | **C.I** | **Email** |
| --- | --- | --- | --- | --- |
| Coordinador | Reyes | Franco | 5.676.219-7 | franco07sierra@gmail.com |
| Sub-Coordinador | Bittencourt | Luis | 5.710.007-1 | SantiagoBittencourt17@gmail.com |
| Integrante 1 | Larrosa | Maria | 5.633.663-5 | victolarrosa13@gmail.com |

**Docente: Martinez, Santiago**

Configuración de red del servidor

El equipo de VifraSoft configuró la red de del servidor linux desde la máquina virtual que lo aloja. El objetivo de esta configuración es poder tener una comunicación más eficiente entre dispositivos para poder efectuar el correcto funcionamiento de la aplicación, a continuación se mostrará todo lo que el equipo hizo para tener una correcta y eficiente comunicación entre el dispositivo anfitrión y además tener la posibilidad de administrar el servidor linux fácilmente desde el navegador



Primeramente dejamos la configuración de red de la máquina virtual como se muestra en la imagen.Está configurada para que se pueda conectar

a internet y asi poder descargar los servicios deseados como ssh, nmtui y cockpit y también para que pueda haber una conexión entre la pc anfitrión y la VM, la dirección MAC siempre sera unica, no habrán dos iguales, y es sumamente necesario que la casilla de cable conectado esté seleccionada con el tic.

Una vez la configuración esté lista deberemos iniciar la maquina virtual Linux

Los servicios que utilizaremos serán: ssh, nmtui y cockpit

Los comandos para instalarlos son los siguientes:

ssh:

**dnf install openssh-server** para instalar ssh

**systemctl enable sshd** para activarlo

**systemctl start sshd** para iniciarlo

**firewall-cmd --add-service=ssh** para permitir las comunicaciones

nmtui:

**dnf install NetworkManager-tui** para instalar nmtui

cockpit:

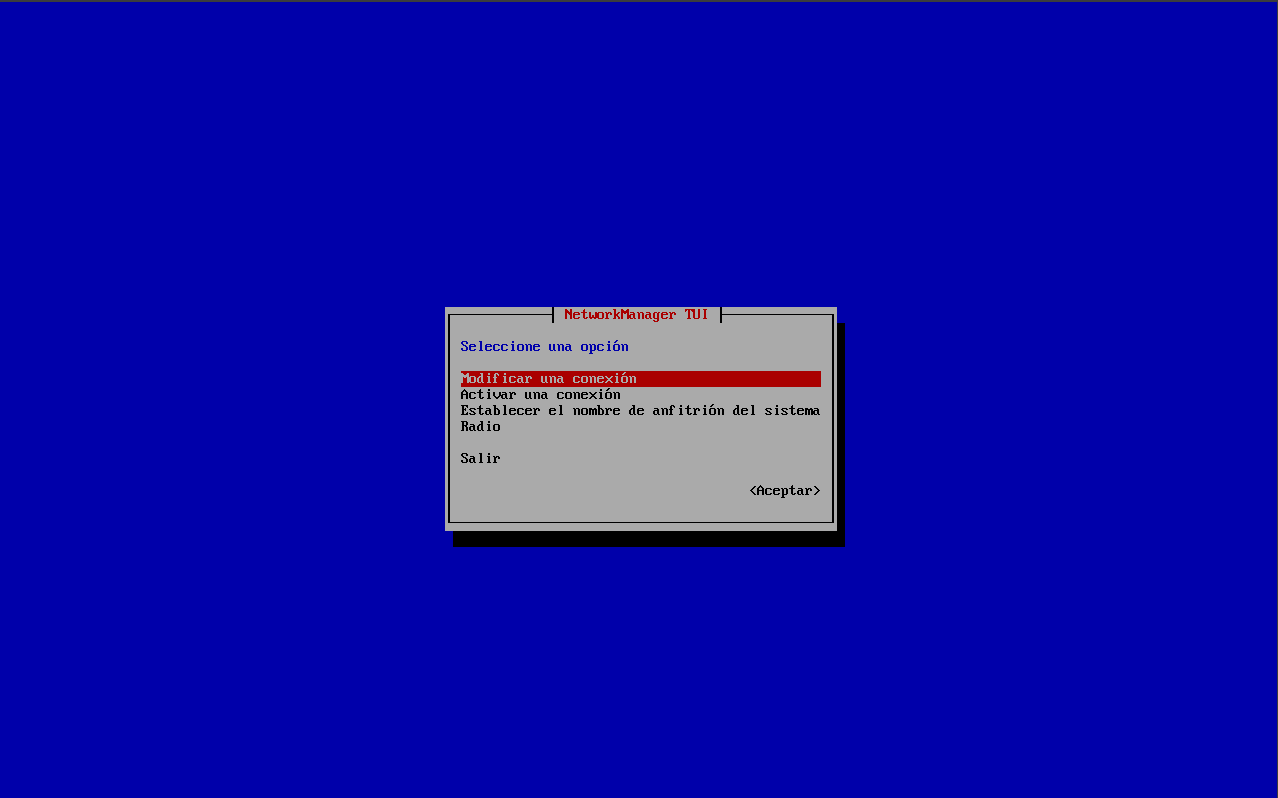
**dnf install cockpit** para instalar cockpit

**systemctl enable cockpit.socket** para activar el cockpit

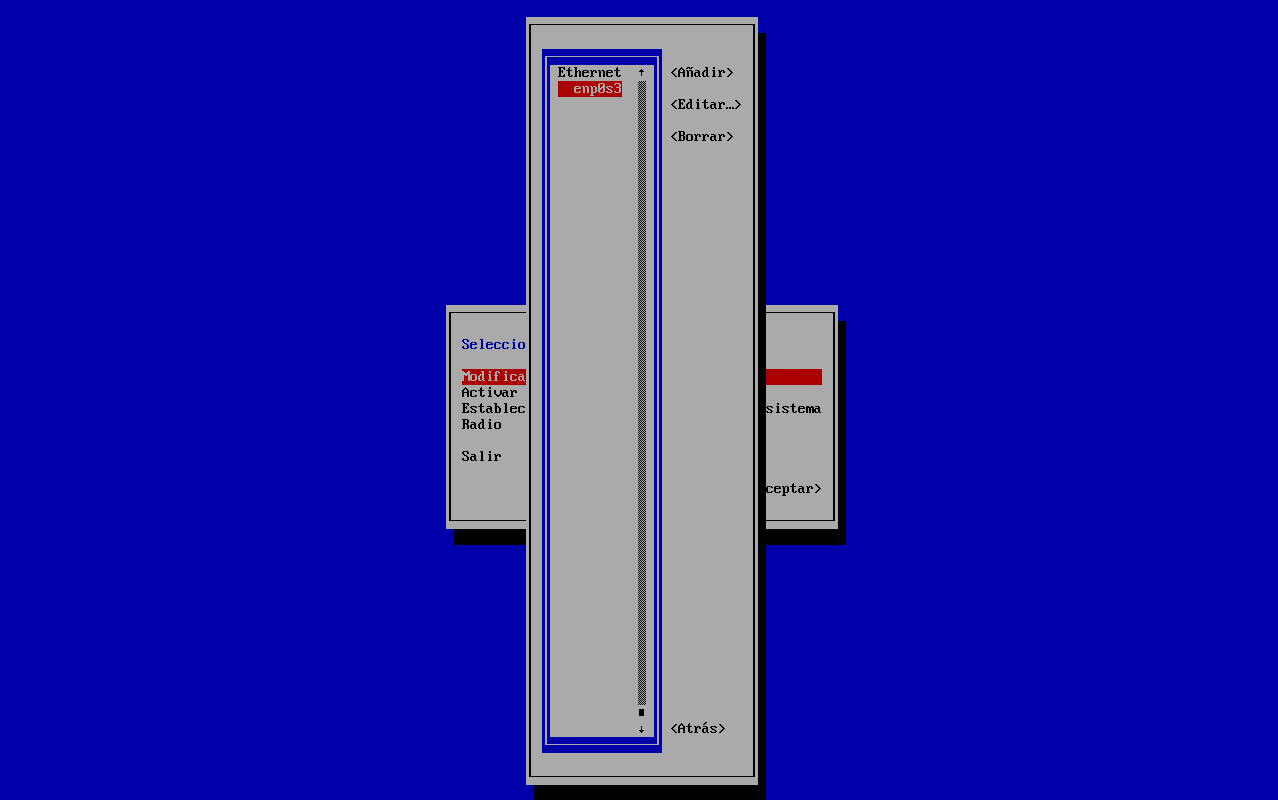
Una vez teniendo en cuenta que tenemos todos los servicios instalados en la máquina virtual linux pasaremos a la configuración y uso de cada uno

NMTUI:

Para poder utilizar el servicio de nmtui debemos ingresar este mismo en la terminal de linux y nos abrirá un menú tal que así:

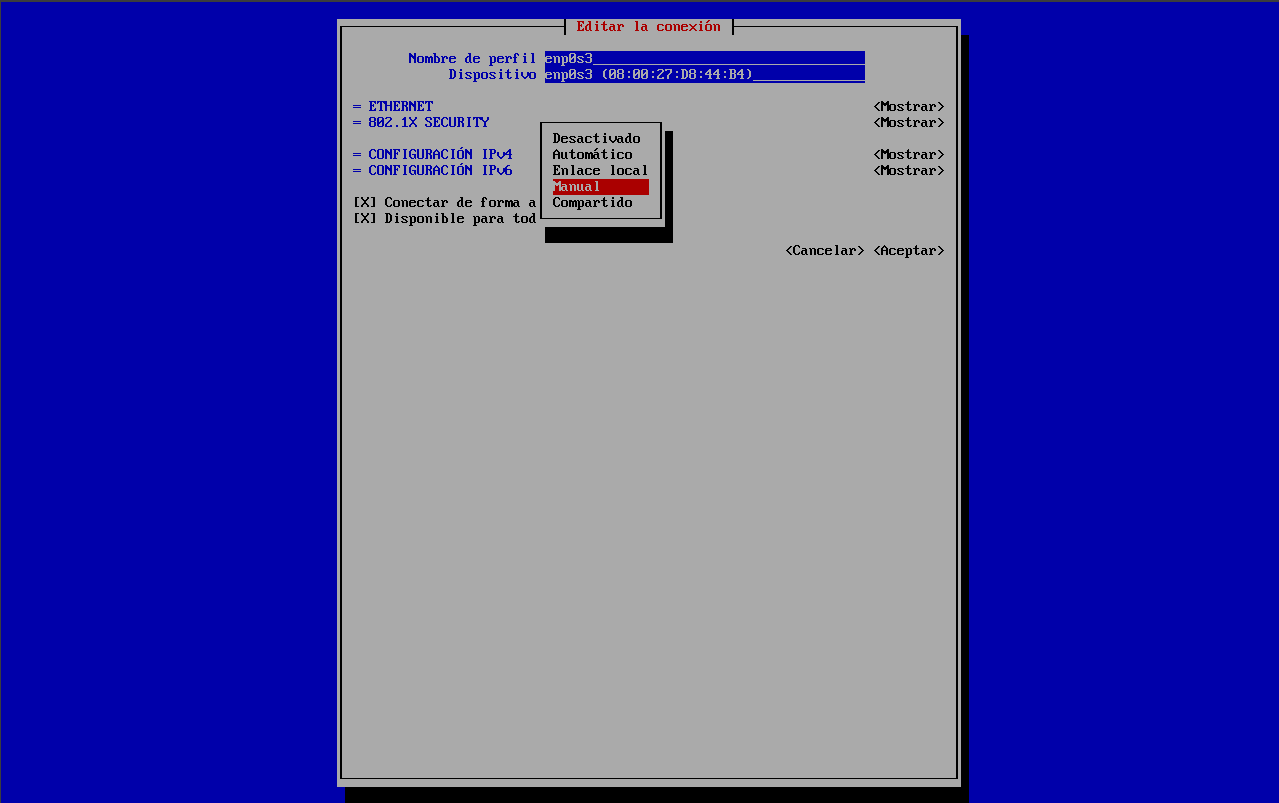


Debemos presionar **enter** en el mensaje “Modificar una conexión” como en la imagen.

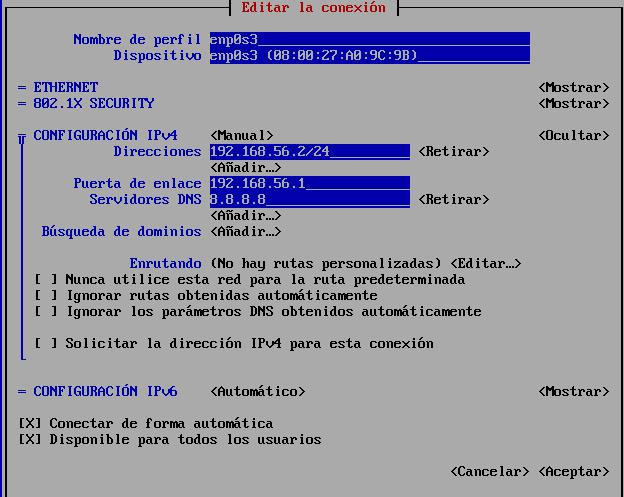


cuando se despliegue este menú lo que debemos hacer es seleccionar la única opción como se muestra en la imagen

Nos aparecerá el siguiente menú



Utilizando las flechas del teclado nos dirigiremos hacia configuración ipv4 para poner la opcion de configuracion manual como se muestra en la imagen, luego de esto deberás presionar el mensaje de “<Mostrar>” del lado derecho para ingresar las configuraciones deseadas



Para cada configuración aparecerá el mensaje de “<Añadir>” y solo configuraremos “Direcciones”, “Puerta de enlace” y “DNS”, la configuración de la imagen son únicamente ejemplos y cada uno debe ser único por servidor y a la vez coincidir en el rango de la ip de la pc anfitriona para poder tener una comunicación correcta entre ellas, el DNS recomendamos siempre dejarlo en “8.8.8.8” y la puerta de enlace siempre sera [XXX.XXX.XXX](http://xxx.xxx.xxx).1 según la ip

El motivo por el cual se instala nmtui es para tener un panel para modificar la o las conexiones de nuestra máquina virtual, en pocas palabras se utiliza para evitar hacer las configuraciones de conexión evitando usar comandos en consola lo cual genera una mayor comodidad y eficiencia.

SSH:

SSH, sigla de Secure Shell, es un protocolo de red que permite establecer una conexión segura y cifrada entre dos computadoras. En sistemas Linux, se utiliza principalmente para acceder y controlar remotamente otro equipo a través de la línea de comandos.

El uso de SSH es fundamental en la administración de servidores, especialmente aquellos que no cuentan con interfaz gráfica, como muchas versiones de Linux Server (por ejemplo, Fedora Server).

En la página 3 del documento se muestran los comandos de instalación de SSH pero de igual forma los reiteramos.

**dnf install openssh-server** para instalar ssh

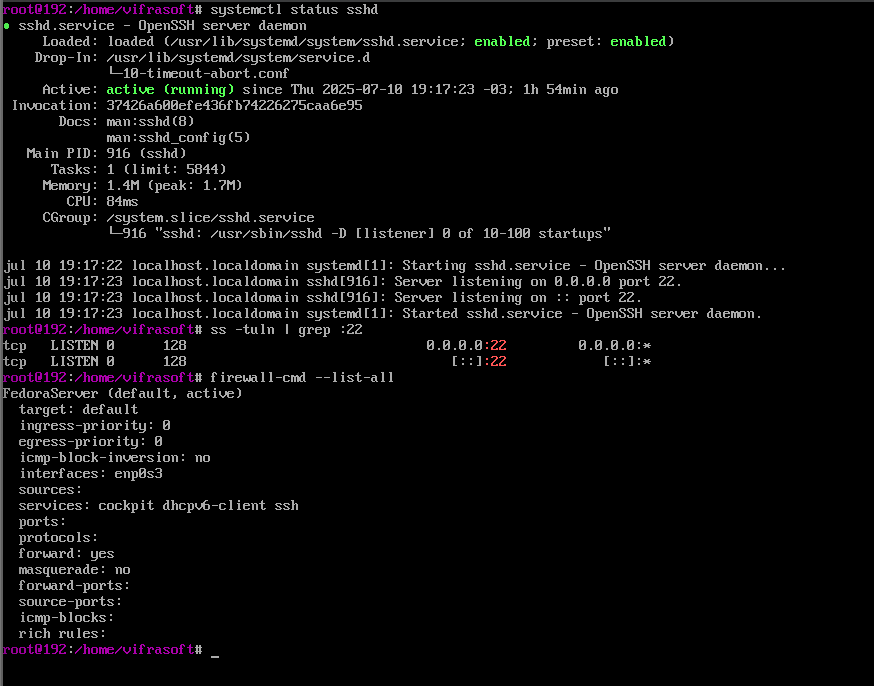
**systemctl enable sshd** para activarlo

**systemctl start sshd** para iniciarlo

**firewall-cmd --add-service=ssh** para permitir las comunicaciones

SSH también permite otras funciones avanzadas, como la transferencia segura de archivos (mediante scp o sftp), la creación de túneles de red, y la autenticación mediante llaves criptográficas, lo que mejora aún más la seguridad.

Una vez instalado todo correctamente e inicializado el servicio veremos el estado del servicio



En la imagen se muestra el comando **systemctl status sshd** la información que otorga este comando es en qué estado se encuentra el servicio que como muestra está activado por el comando **systemctl enable sshd.**

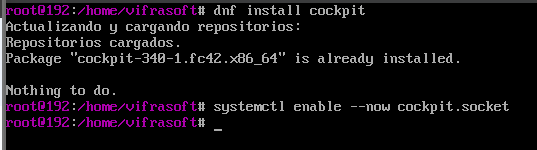
Después el conjunto de comandos **ss -tuln | grep :22** que es para verificar que el puerto definido por defecto para establecer las conexiones

Por último usamos el comando **firewall-cmd --list-all** ya que este nos muestra la configuración actual del firewall en linux lo cual nos permite hacer diagnósticos de errores de conexión o seguridad y además ver que está permitido o bloqueado dentro del mismo, por ejemplo en la imagen muestra los servicios permitidos.

COCKPIT:

Cockpit es una herramienta de administración de servidores Linux que permite gestionar el sistema a través de una interfaz web moderna y fácil de usar. Está diseñada para facilitar tareas comunes sin necesidad de utilizar exclusivamente la línea de comandos, lo que la convierte en una opción ideal tanto para administradores experimentados como para principiantes.

A través de Cockpit es posible supervisar el estado del sistema en tiempo real, controlar el uso de CPU, memoria, red y disco, gestionar servicios, usuarios, paquetes, y configurar el firewall. También permite administrar discos y particiones, revisar los logs del sistema y acceder directamente a una terminal desde el navegador.

Su instalacion y activacion son muy sencillas y se mostrará en la siguiente imagen   


Cockpit se accede ingresando la dirección IP del servidor en un navegador, seguida del puerto 9090 (por ejemplo, <https://192.168.56.2:9090>).

COMUNICACIÓN SSH:

A continuación haremos que la PC anfitriona se conecte con la máquina virtual linux mediante el SSH con autenticación de llave pública y privada

Las llaves pública y privada son un par de archivos criptográficos que se utilizan para autenticar de forma segura el acceso a un servidor. En lugar de usar una contraseña que podría ser interceptada o adivinada, SSH usa este par de llaves para verificar tu identidad.

La **llave pública** es un archivo que puedes compartir sin preocupaciones. Piensa en ella como un candado abierto que entregas al servidor al que deseas conectarte. Este candado puede ser visto por cualquiera, pero solo puede ser abierto por la llave privada correspondiente.

La **llave privada** es el componente secreto y crucial de este sistema. Esta llave debe permanecer siempre en tu posesión y nunca, bajo ninguna circunstancia, debe ser compartida con nadie. Es la llave que abre el candado creado por la llave pública.

En conjunto, este par de llaves permite una autenticación robusta: la llave pública cifra un mensaje que solo puede ser descifrado por la llave privada. Al descifrar y responder correctamente, demuestras que tienes la llave privada sin necesidad de enviarla por la red, lo que la convierte en un método de acceso mucho más seguro que las contraseñas.

# Paso a paso:

Para empezar con la configuración de las comunicaciones SSH, sabiendo que el servicio ya está instalado descargamos un programa externo para hacer todo de forma más eficiente, El programa se llama PuTTY y la instalación es sencilla

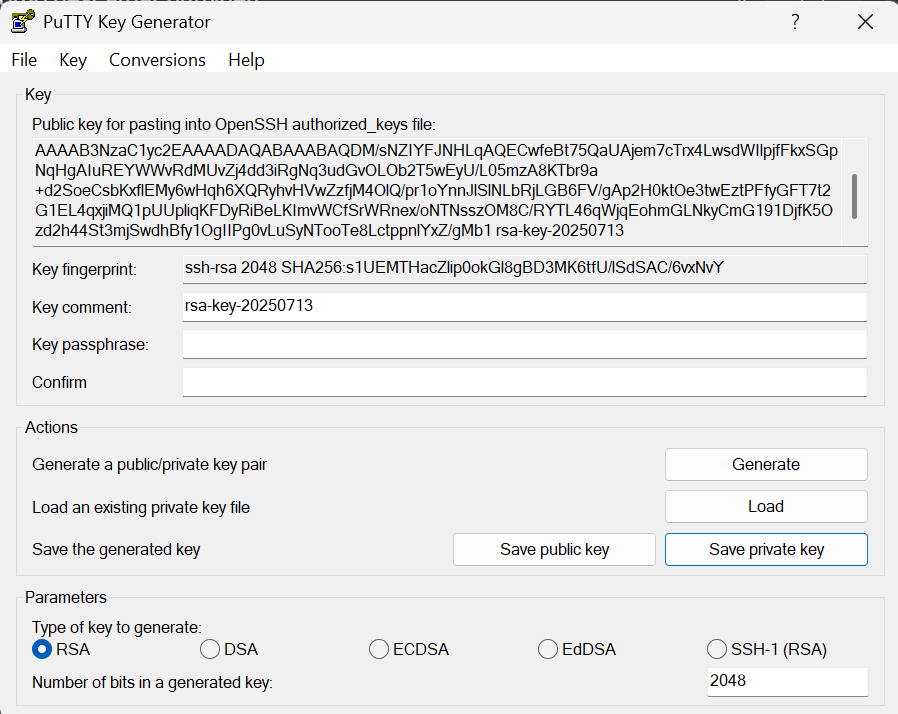
Enlace: [PuTTY](https://the.earth.li/~sgtatham/putty/latest/w64/putty-64bit-0.83-installer.msi)

Y además debemos descargar un complemento de PuTTY llamado puttygen que es la que nos hará crear y guardar las llaves tanto pública como privada.

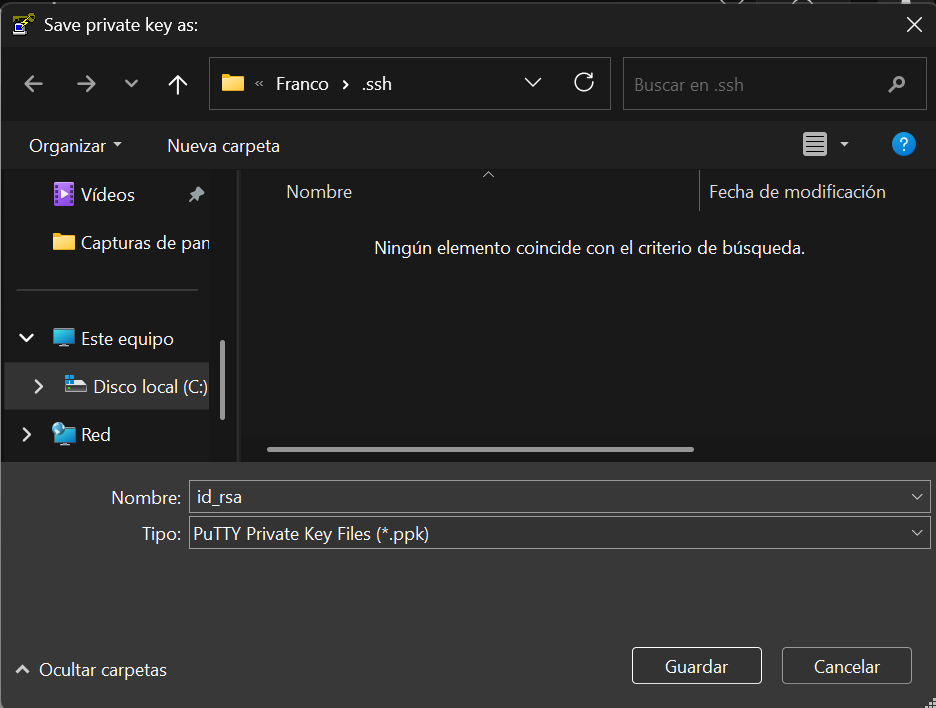
Enlace: [puttygen](https://the.earth.li/~sgtatham/putty/latest/wa32/puttygen.exe)

PuTTY se deberá ejecutar su instalador y puttygen al abrirlo funcionará automáticamente

Una vez instalado todo lo que tendremos que hacer es abrir el programa puttygen

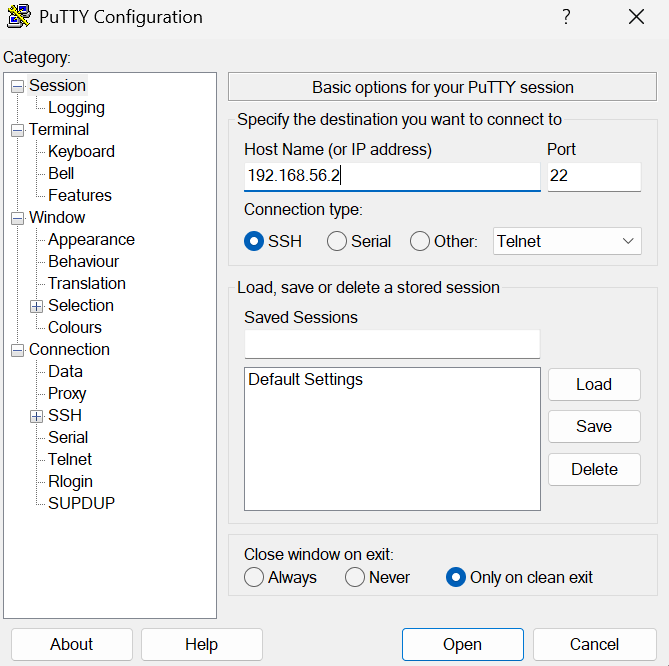


Aquí tendremos que presionar el botón generate y generará la llave pública que es todo el texto que se ve en la casilla y si vemos hay dos botones que dicen **save public key** y **save private key,** guardaremos las dos.

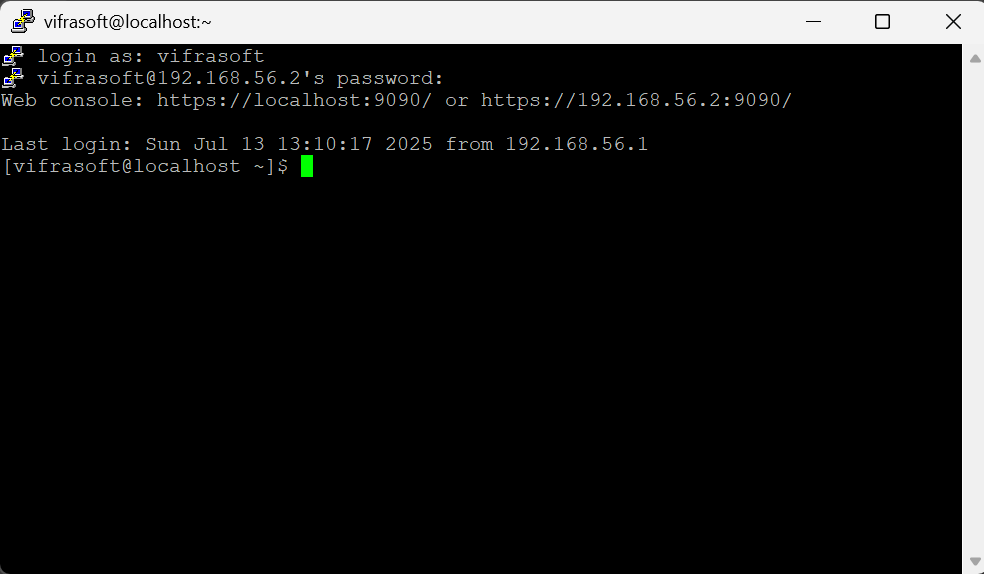
Si la carpeta ssh no existe se crea

El ejemplo de la imagen es con la llave privada pero tendremos que hacerlo de igual forma con la llave publica, solamente que se debera guardar como “id\_rsa.pub” en el tipo aparecera como All files

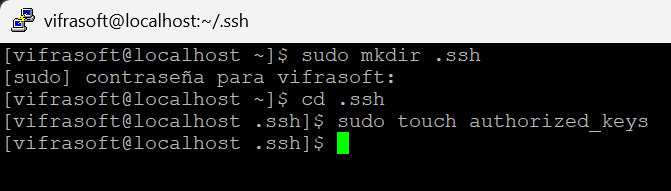
Lo siguiente es abrir el programa PuTTY



Nos aparecerá este panel en el cual dentro de la casilla Host Name debemos poner la ip de nuestro servidor como se muestra en la imagen y presionar enter o el botón open



En seguida de darle enter nos mostrara la siguiente terminal y debemos iniciar con el usuario y contraseña creado para el servidor, de la misma forma que iniciamos sesión cuando encendemos la VM



crearemos el el directorio .ssh dentro de la carpeta usuario y dentro del directorio .ssh crearemos el archivo **authorized\_keys** que va a contener la llave pública que creamos con puttygen

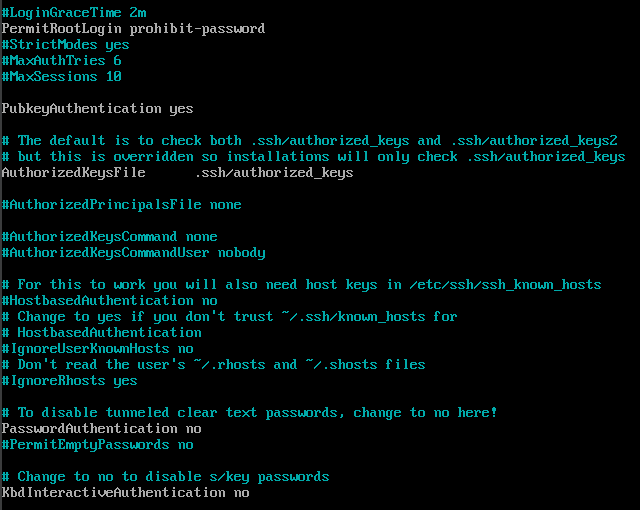
Lo siguiente que tendremos que hacer es usar el comando nano para pegar la llave pública, por ende debemos copiarla del programa puttygen y usando

**sudo nano authorized\_keys** posicionado en el directorio .ssh y con un simple ctrl+v o click derecho para copiar la llave pública, una vez copiada presionaremos ctrl+o para guardar el archivo modificado y ctrl+x para salir del editor

con todo esto hecho escribiremos en la terminal “exit” para salir de la misma

Después de salir de la terminal abriremos nuevamente el programa de PuTTY

De nuevo en el programa y ya con la llave pública dentro del servidor tendremos que configurar todo para que al momento de hacer la conexión se usen las llaves y no una contraseña, antes de configurar PuTTY nos dirigiremos a la VM para modificar un documento que son las configuraciones del ssh para eso mismo tendremos que estar posicionados en la base de archivos del linux y poner el siguiente comando “**sudo nano etc/ssh/sshd\_config**” y nos mostrará lo siguiente



si prestamos atención veremos que son comentarios ya que inician con “#” para habilitar las configuraciones tendremos que borrar el “#” en la imagen muestra los que están en blanco son los que habilitados y mismo en la imagen ya se ven como lo configuramos nosotros que sería así:

PermitRootLogin **prohibit-password**

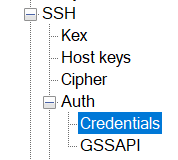
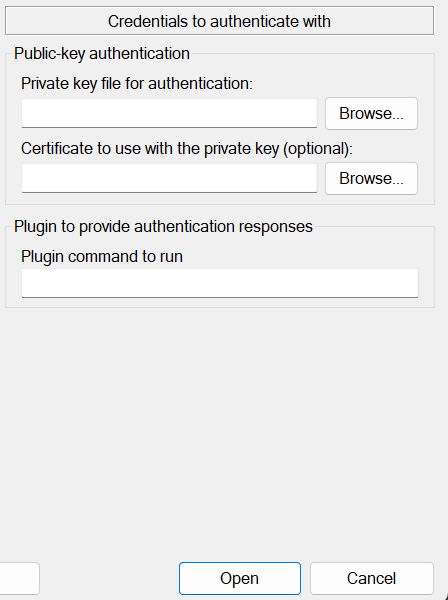
PubKeyAuthentication **yes**

PasswordAuthentication **no**

KdbInteractiveAuthentication **no**

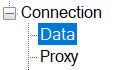
Una vez configurado todo haremos la combinación de (ctrl+o) y (ctrl+x) para guardar todo y poder ir al siguiente paso.

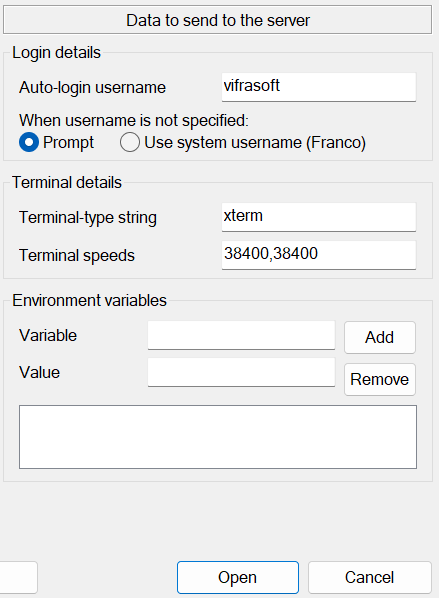
Lista la configuración del ssh el siguiente paso es cargar los datos de usuario y llave privada en en PuTTY para iniciar automáticamente con ssh por ende ejecutaremos el programa y haremos lo siguiente:

En el panel izquierdo del programa mostrará esta ruta SSH >> Auth >> Credentials y nos mostrará el panel de la imagen derecha y en la casilla de **Private key file for authentication** y con el botón browse elegiremos la llave privada que descargamos en la carpeta ssh de windows

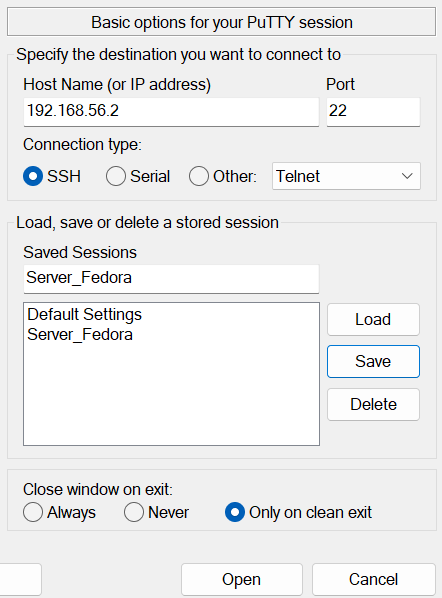
El siguiente paso va a ser dejar el nombre de usuario del servidor anotado para que al momento de hacer la conexion no lo pregunte y lo inicie automáticamente

 De igual forma que el paso anterior con el panel izquierdo nos dirigiremos a Connection >> Data



y en la casilla de Auto-login username pondremos el nombre del usuario

Una vez hecho todo esto volveremos al panel principal y guardaremos la configuración



en la casilla de Saved Sessions pondremos el nombre que queramos para guardar la configuración y le daremos al botón save y cada vez que queramos usarla la seleccionaremos y le daremos al botón de open

Si todo está correcto al momento de dar enter u Open entrara al servidor sin pedir contraseña ya que usará las llaves.

Conclusión:

Lo que se ha visto en esta documentación es la forma en cómo el equipo de VifraSoft configuró la red del servidor para dejar una IP estática y tener los puertos necesarios abiertos para tener una correcta comunicación con otros equipos, principalmente el host.

Además se vio la instalación, gestión y configuración de los paquetes necesarios para que el servidor pueda alojar y ejecutar la aplicación deseada, reiterando los paquetes: **SSH** para el control remoto, **cockpit** para la administración web y **nmtui** para la configuración de red.