Francisco Joaquín Murcia Gómez

48734281H

Grado en ingeniería informática UA

ADMINISTRACION DE SISTEMAS OPERATIVOS Y DE REDES DE COMPUTADORES

Hito 2



índice

[Licencias 3](#_Toc56718538)

[CentOS 4](#_Toc56718539)

[Particionado 4](#_Toc56718540)

[Administración remota 6](#_Toc56718541)

[SSH/SFTP/SCP 6](#_Toc56718542)

[RDP 9](#_Toc56718543)

[VNC 10](#_Toc56718544)

[Networking 11](#_Toc56718545)

[DHCP 11](#_Toc56718546)

[Servidor OwnCloud 12](#_Toc56718547)

[Servidor GIT 15](#_Toc56718548)

[File Server 15](#_Toc56718549)

[NFS 15](#_Toc56718550)

[SAMBA 16](#_Toc56718551)

[FreeNAS 17](#_Toc56718552)

[Impresión remota 18](#_Toc56718553)

[CUPS 18](#_Toc56718554)

[Base de datos y web servicies 19](#_Toc56718555)

[Base de datos 19](#_Toc56718556)

[FreeBSD 22](#_Toc56718557)

[Particionado 22](#_Toc56718558)

[Administración remota 24](#_Toc56718559)

[SSH/SFTP/SCP 24](#_Toc56718560)

[Networking 26](#_Toc56718561)

[DHCP 26](#_Toc56718562)

[Git server 28](#_Toc56718563)

[OwnCloud server 28](#_Toc56718564)

[Servicios de impresión 29](#_Toc56718565)

[Cups 29](#_Toc56718566)

# Licencias

Una licencia es un contrato mediante el cual una persona recibe de otra el derecho de uso, de copia, de distribución, de estudio y de modificación.

En el caso del software, serian todos los permisos que los desarrolladores otorgan a los usuarios, en estos se especifica:

* Los límites en la responsabilidad por fallos.
* El plazo de cesión de los derechos.
* Cesión del programa a terceros
* Reinstalación del programa en equipos distintos al original

En los sistemas operativos a analizar, Windows Server usa EULA, CentOS usa GPL GNU y FreeBSD, BSD

EULA:

* De pago, con precios entre 6€ y 800€
* Prohíbe la copia y modificación.
* Las actualizaciones del sistema pueden modificar la licencia
* Se conceden permisos a la distribuidora para realizar cambios en nuestro sistema
* Se conceden permisos a la distribuidora para utilizar nuestra información
* Podría dejar de funcionar si se realizan cambios de hardware

GNU GPL:

* Gratuito
* Permite la copia, modificación y redistribución del software.
* Puede ser vendido y se puede cobrar por los servicios sobre el software
* Se debe proveer con el código fuente
* Si hay un cambio en la licencia, los términos generales de la licencia existente se mantienen.

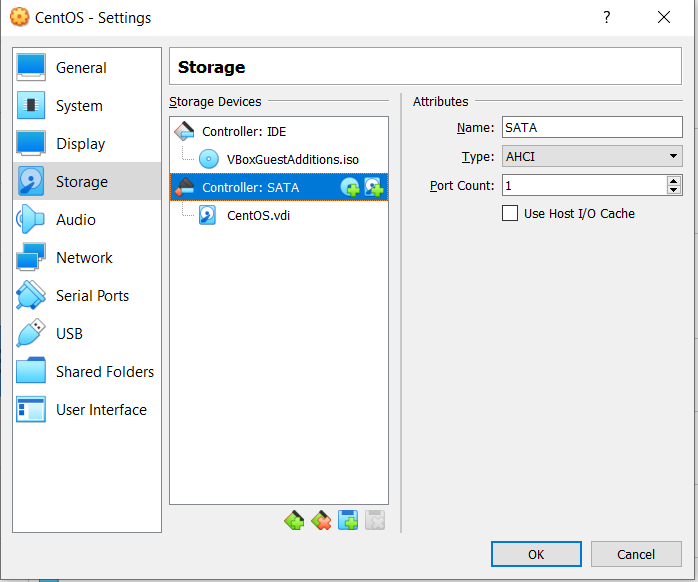
BSD

* Gratuito
* Permite la copia, modificación y redistribución del software
* BSD permite el cobro por la distribución
* El desarrollador no está en la obligación de incluir el código fuente

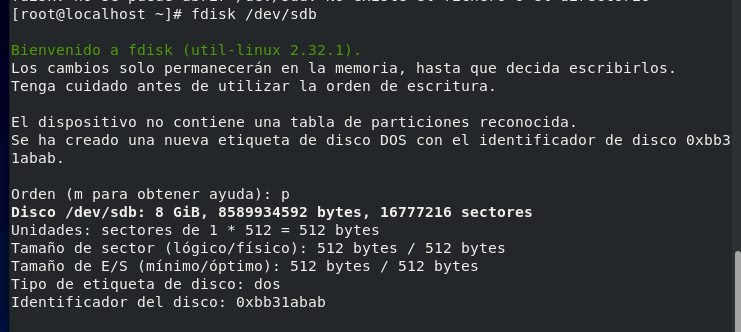
# CentOS

## Particionado

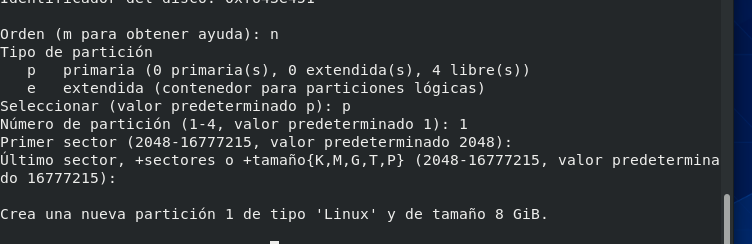
El apartado configuración de la máquina virtual, en el apartado de almacenamiento, clicamos en añadir un disco en el controlador SATA,



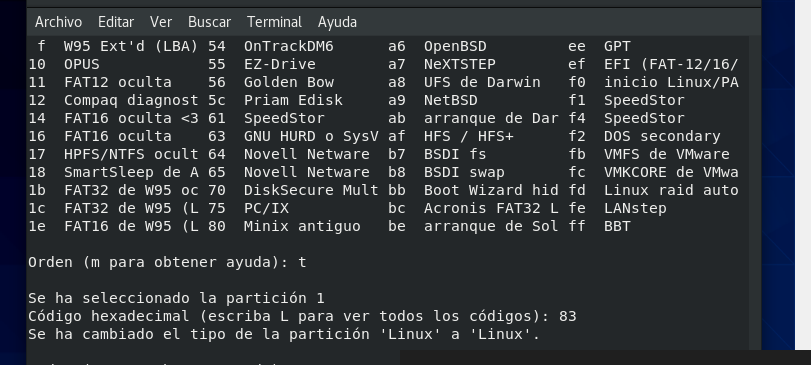
Una vez creado el disco para habilitarlo introducimos fdisk /dev/sdb, con el comando p nos muestra los discos listos para instalar.



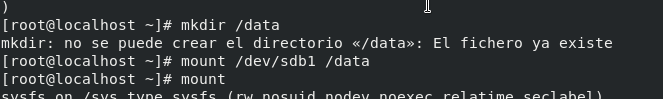
Con el comando n introducimos el nuevo disco, después nos pedirá una serie de requisitos si la queremos primaria introducimos p, el número de partición 1 y los sectores lo dejamos que el sistema elija.



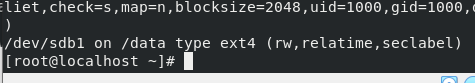
Para definir el tipo de partición tecleamos t, e introducimos el código del tipo de partición seleccionado, para saber los tipos, introducimos l, en el caso de Linux, es 83, así que lo introducimos.



Para finalizar de montar el disco tenemos que crear la carpeta data, para ello, usaremos mkdir /data y para montarla en el disco, mount /dev/sdb1 /data



Para conformar ejecutamos mount y al final del todo ha de aparecer la carpeta data en el disco.



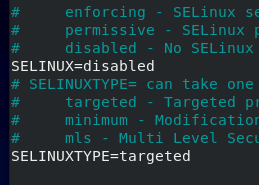
## Administración remota

### SSH/SFTP/SCP

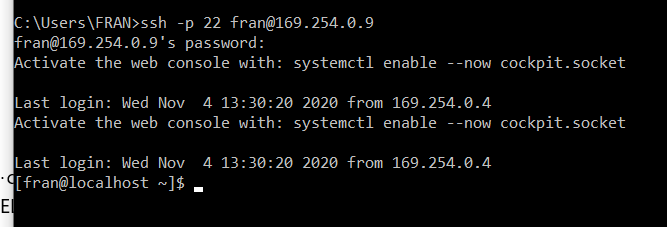
SSH viene instalado en CentOS, para configurarlo hay que acceder al fichero /etc/ssh/sshd\_config, en el aparece toda la configuración de SSH, en el añadimos lo siguiente:

* Añadir: AllowUsers <usuarios permitidos>
* Añadir PubkeyAuthentication yes
* Modificar: UsePAM yes
* Modificar: PermitRootLogin no

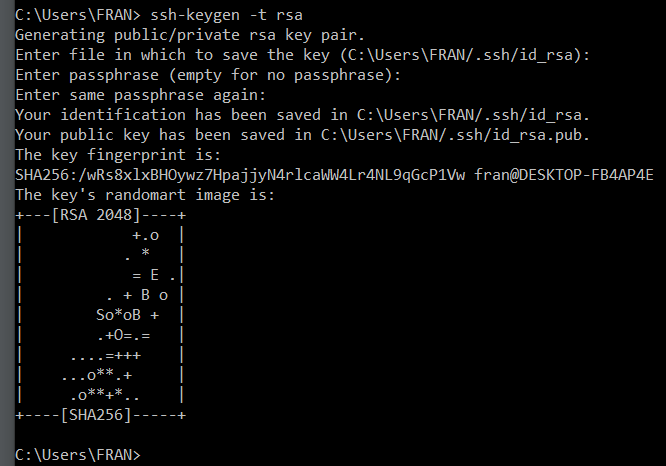
Una vez hecho eso para poder conectarse hay que cambiar la configuración del kernel modificándolo en el fichero /etc/sysconfig/selinux, SELINUX estará en enable, pues lo ponemos en diablead.



Reiniciamos y ya en el host, introducimos ssh <usuario>@<ip servidor> nos pedirá la contraseña del usuario y ya estaríamos conectados al servidor



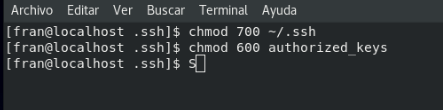
Para generar la clave publica, usaremos el comando ssh-keygen -t rsa en el host



En la Windows powershell introducimso la siguiente sentencia para asignarla al servidor:

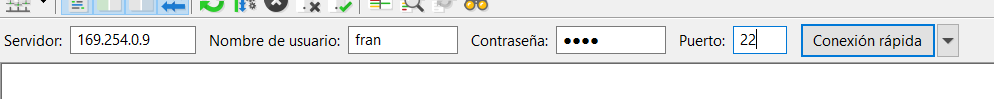
cat ~/.ssh/id\_rsa.pub | ssh <usuario>@<ip servidor> "mkdir ~/.ssh; cat >> ~/.ssh/authorized\_keys"

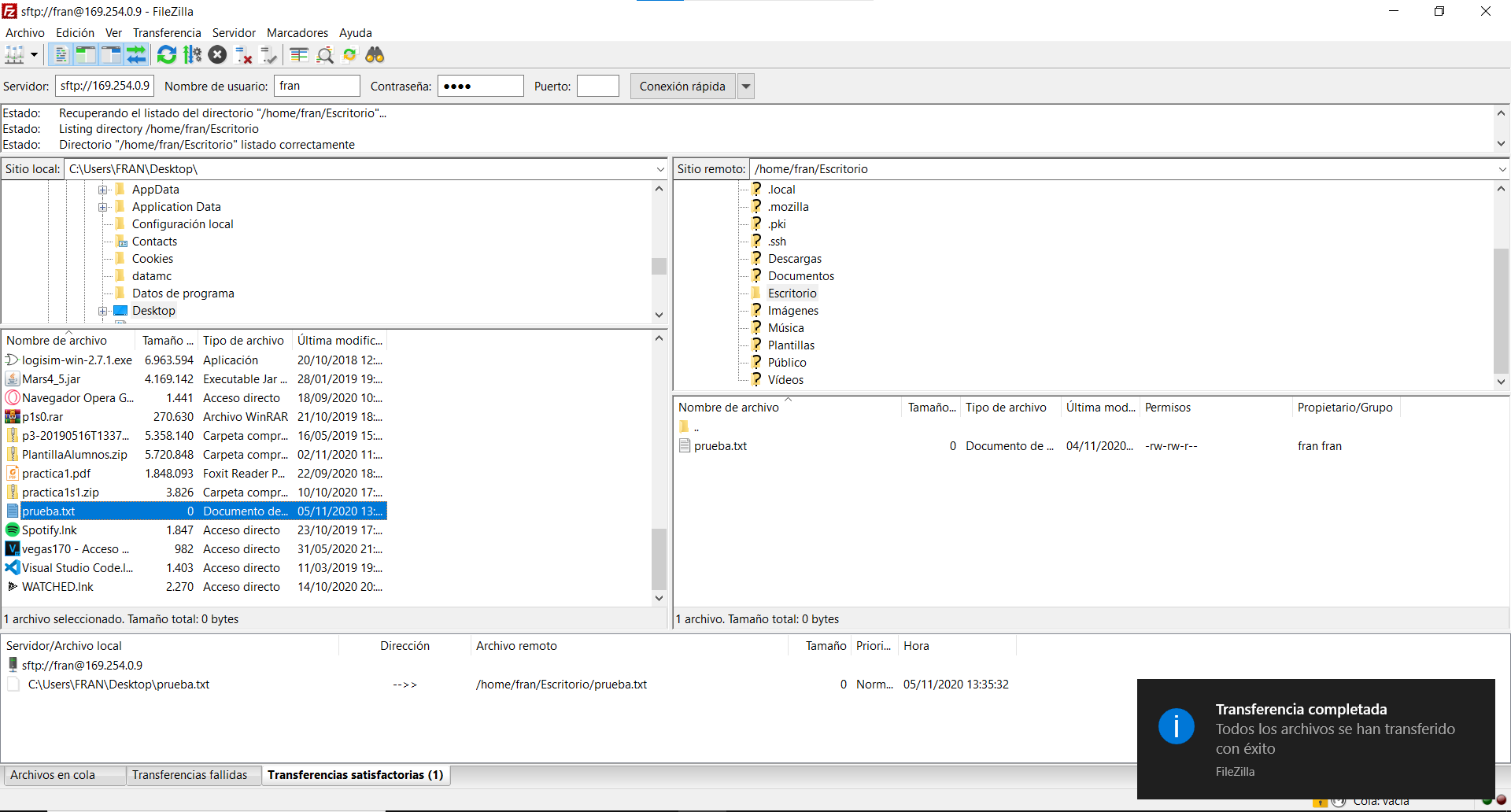
después en el servidor, signamos los premisos adecuados a la carpeta .ssh y al fichero que contiene las claves



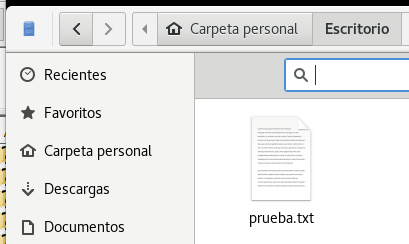
Para realizar la conexión SFTP, se ha utilizado el programa de gestión de archivos filezilla.

Nos conectamos a nuestro servidor con la ip usuario y contraseña



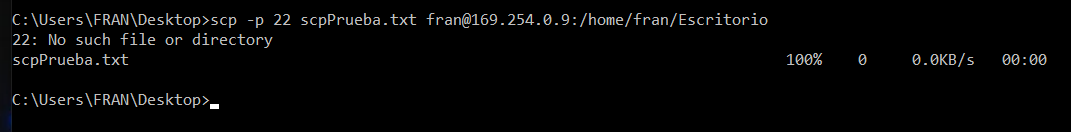
Y para pasar archivos del host al servidor o viceversa solo hay que pinchar y arrastrar, por ejempo, he pasado un archivo prueba al servidor

Si nos vamos al escritorio que es a donde lo he pasado, vemos que el archivo aparece.

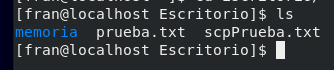


Para comprobar el servicio SCP he enviado un documento de prueba

scp -p <puerto> <archivo> <usuario>@<ip>:<ruta destino>



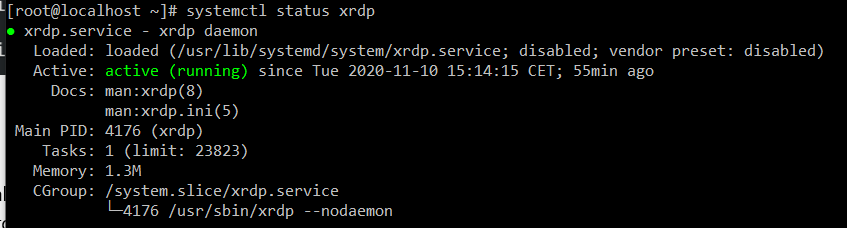
Si hacemos un ls en escritorio, observamos que el archivo se ha enviado



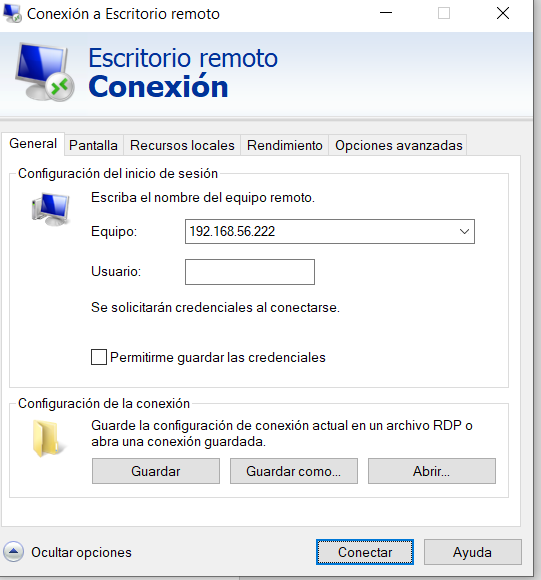
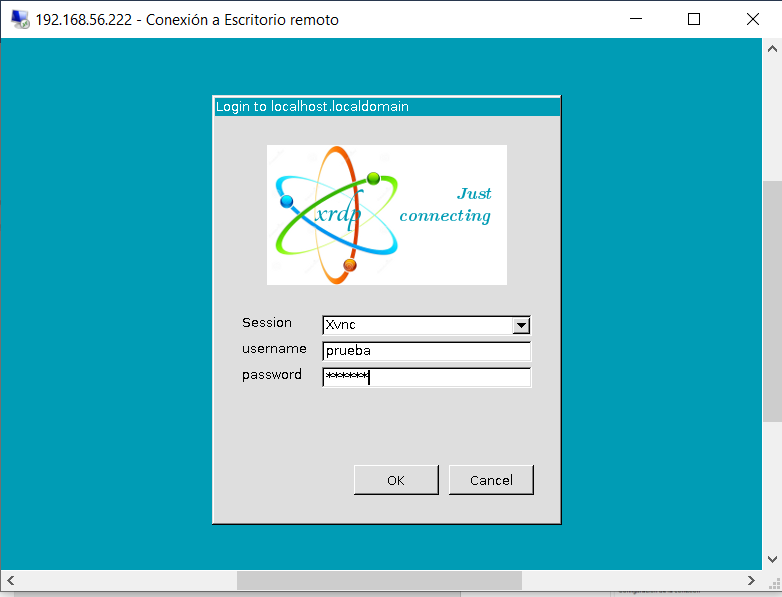
### RDP

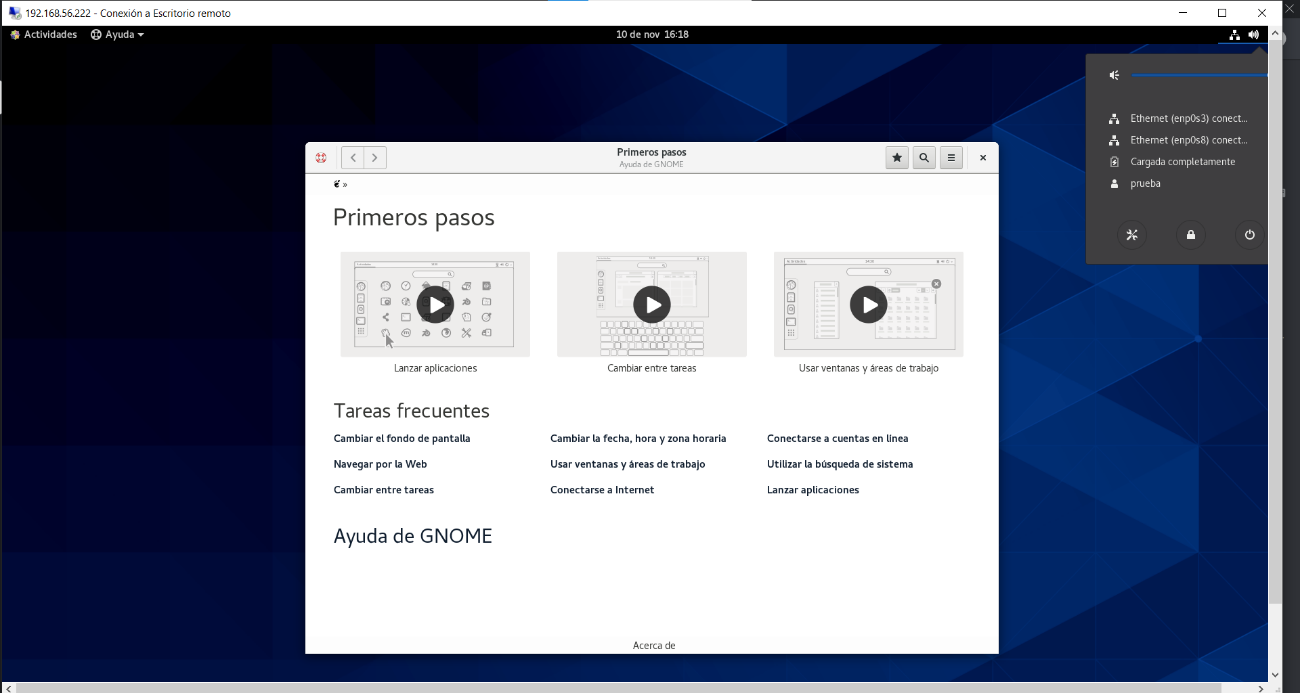
Para activar el servicio RDP necesitamos instalar el xrdp, para ello previamente se intalara epel-release para que deje instalar xrdp

En segundo paso, activamos el servicio con el comando systemctl start xrdp y lo comprobamos con systemctl status xrdp.



Con la aplicación para escritorios remotos de Windows colocamos la ip del servidor y nos conectamos

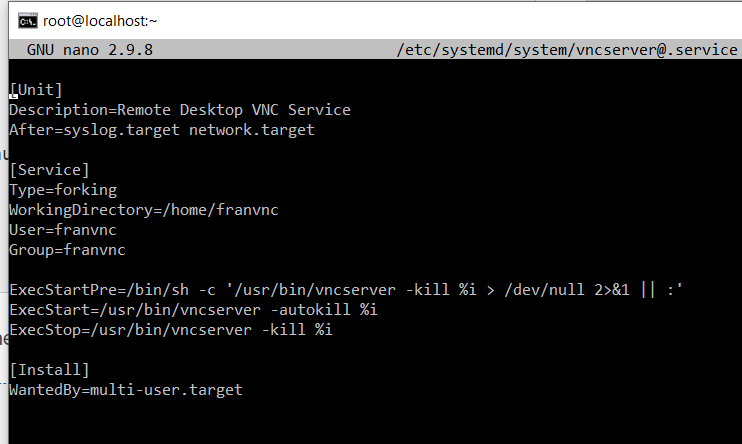




### VNC

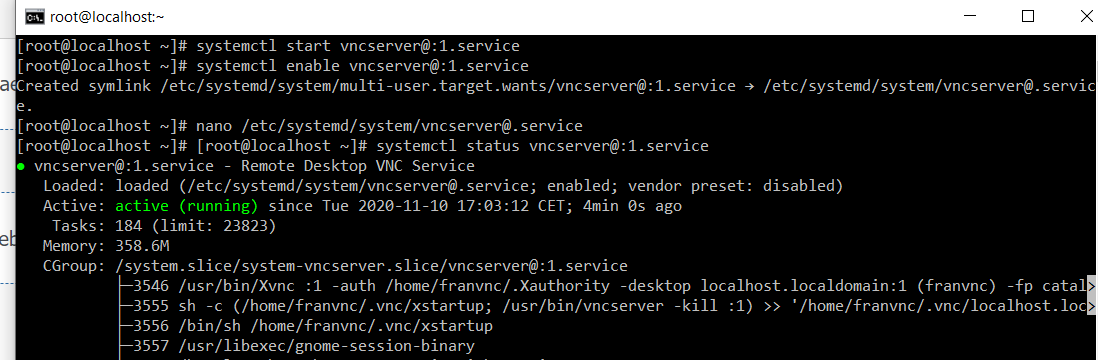
Para activar el servicio de vnc hay que instalat TigerVNC en primer lugar “dnf install tigervnc-server tigervnc-server-module -y”

Después hemos de crear el servicio de vnc, para ello modificamos el siguiente archivo [/etc/systemd/system/vncserver@.service](mailto:/etc/systemd/system/vncserver@.service) y añadiremos lo siguiente:



Después recargamos el fichero con systemctl daemon-reload, después activamos el servicio con systemctl start vncserver@:1.service y por ultimo systemctl enable vncserver@:1.service.

Comprobamos el estado con systemctl status vncserver@:1.service



Nos conectamos a la dirección 192.168.56.222 en el puerto 5901 en VNCViwer



## Networking

### DHCP

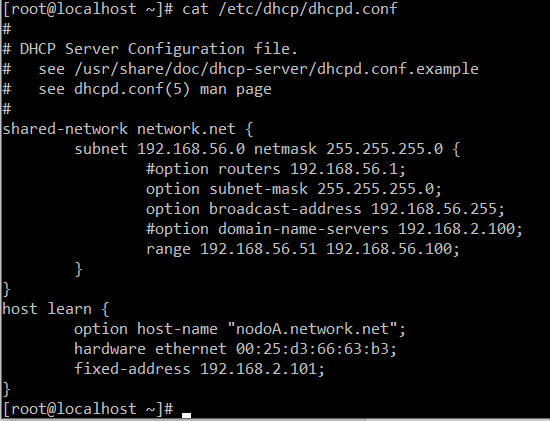
En primer lugar, tenemos que colocar la ip de la interfaz host-only en estática, para ello modificaremos el fichero /etc/sysconfig/network-scripts/ifcfg-enp0s8

* Modificar BOOTPROTO=none
* Añadir IPADDR=192.168.56.222
* Añadir NETMASK=255.255.255.0

Después reiniciamos la interfaz enp0s8 con “nmcli con reload” y después con “nmcli con up enp0s8”

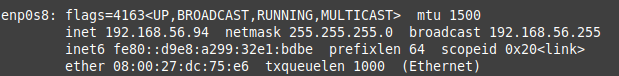
Desactivamos el servidor dhcp de VirtualBox nos descargamos “dhcp-server”

Modificamos el archivo /etc/dhcp/dhcpd.conf:



Finalmente activamos dhcp systemctl start dhcpd.

Encendemos una maquina de prueba, en micaso un linuxMint y vemos que la ip esta en el rango adecuado



### Servidor OwnCloud

Para hacer un servidor owncloud, necesitamos instalarnos previamente LAMP (Linux – Apache – MySQL – PHP):

yum install httpd httpd-tools

yum install mariadb-server mariadb -y

yum install php

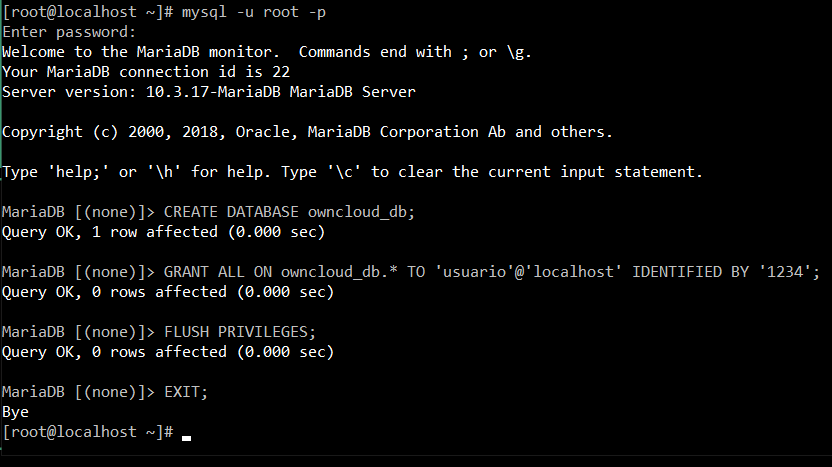
dnf install php-curl php-gd php-intl php-json php-ldap php-mbstring php-mysqlnd php-xml php-zip php-opcache

Deespues nos instalamos owncloud y extraemos su carpeta /var/www:

wget https://download.owncloud.org/community/owncloud-10.3.2.tar.bz2

tar -jxf owncloud-10.3.2.tar.bz2 -C /var/www/

Creamos la base de datos, en ella le clocamos usuario y contraseña para la base de datos



CREATE DATABASE owncloud\_db;

GRANT ALL ON owncloud\_db.\* TO 'usuario'@'localhost' IDENTIFIED BY 'contraseña';

FLUSH PRIVILEGES;

EXIT;

Modificamos el archivo de configuración /etc/httpd/conf.d/owncloud.conf y colocamos lo siguiente:

Alias /owncloud "/var/www/owncloud/"

<Directory /var/www/owncloud/>

Options +FollowSymlinks

AllowOverride All

<IfModule mod\_dav.c>

Dav off

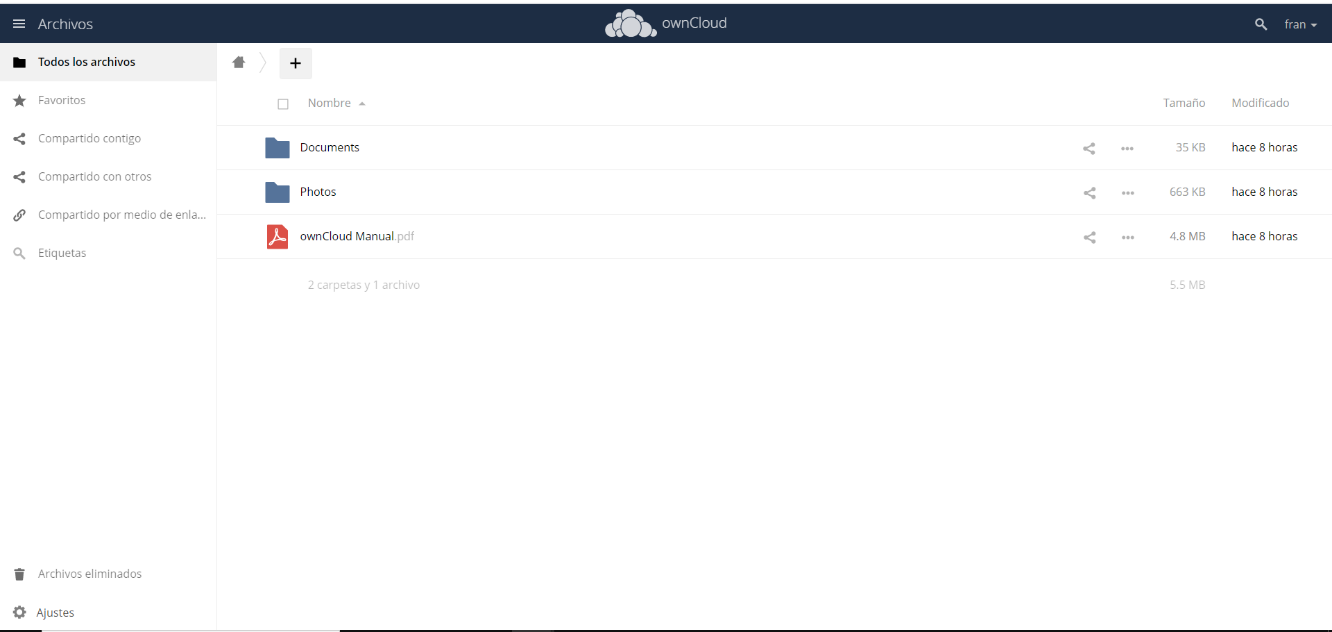
</IfModule>

SetEnv HOME /var/www/owncloud

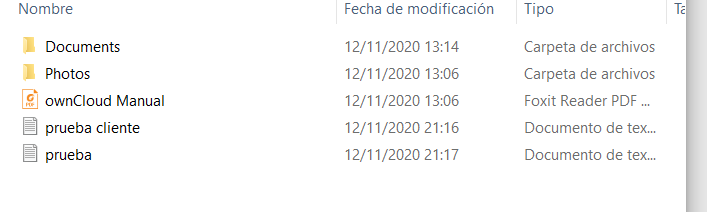
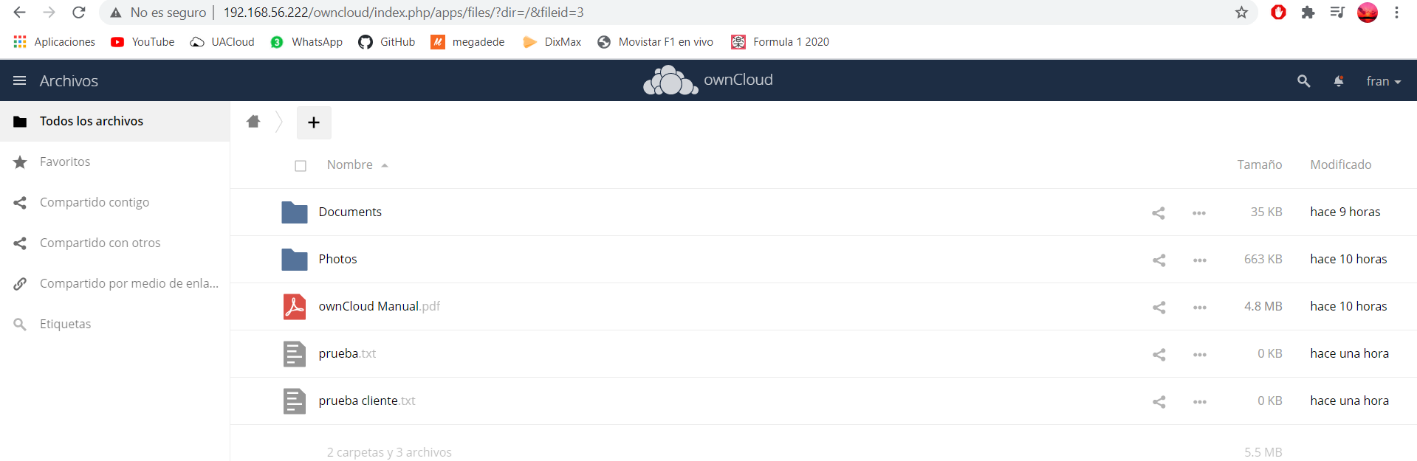
SetEnv HTTP\_HOME /var/www/owncloud

</Directory>

Por abrimos un navegador y colocamos http://ip-servidor/owncloud

Aquí podremos crearnos un usuario administrador y en la pestaña almacenamiento y base de datos, en la opción MySQL tendremos que colocar el usuario contraseña de la base de datos y la base de datos.

Y ya tendríamos nuestro servicio owncloud.

En Windows nos descragamos el cliente y nos logueamos colocamos una carpeta en la carpeta sincronizada y en la web aparecerá

### Servidor GIT

Para instalar un servidor git hemos de instalarnos previamente:

git, yum install git

A continuación, nos creamos un repositorio en una carpeta:

mkdir ~/repo

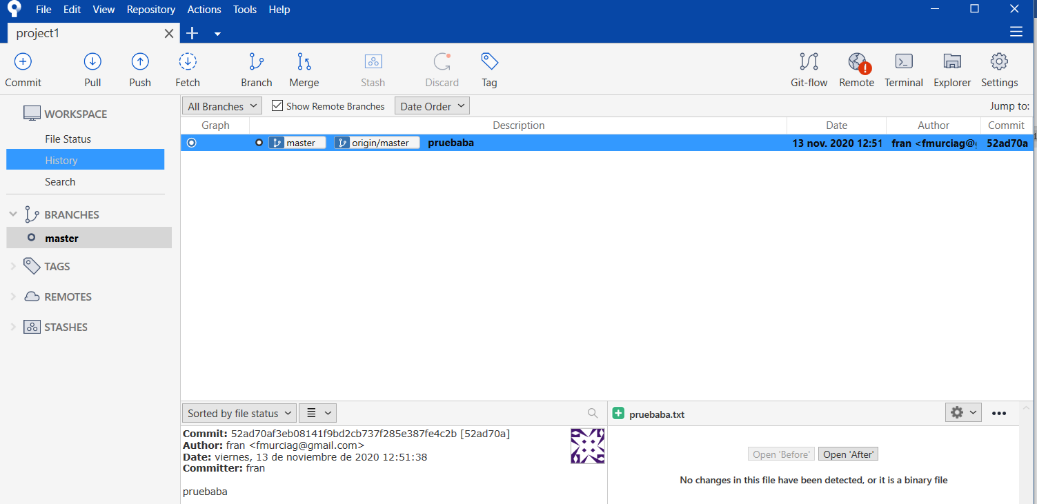
cd ~/repo

git init --bare --shared project1

A continuación, habilitamos post-update hook de la siguiente manera:

cd ~/repo/project1/hooks/

cp post-update.sample post-update

Una vez hecho todo esto, nos descargamos un cliente, en mi caso SourceTree, colocamos la dirección del repositorio en el instalador y ya podríamos trabajar con el:

## File Server

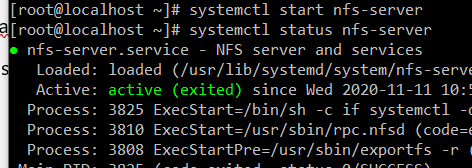
### NFS

En primer lugar, crearemos una carpeta para compartirla mkdir /data/nfs\_shared, y nos instalamos el servicio yum -y install nfs-utils,.

Modificamos el fichero /etc/exports y aña dimos los siguiente:

/directorio\_a\_compartir 192.168.56.1/24(rw,no\_root\_squash)

Después activamos el servicio y comporbamos su estado



### SAMBA

Para realizar samba primero tenemos que descargarnos el servicio de samba:

dnf -y install samba samba-client samba-common

Una vez instalado, creamos un directorio que es el que será conpartido:

mkdir -p /data/samba\_sared

Modificamos el archivo de configuración de samba:

nano /etc/samba/smb.conf

* workgroup = workgroup
* [samba\_sared]

comment=carpeta compartida

path=/data/samba\_sared

browsable =yes

writable = yes

guest ok = yes

read only = no

Nos creamos una contraseña de samba con el comando smbpasswd -a <usuario> y encendemos el servicio systemctl start nmb smb

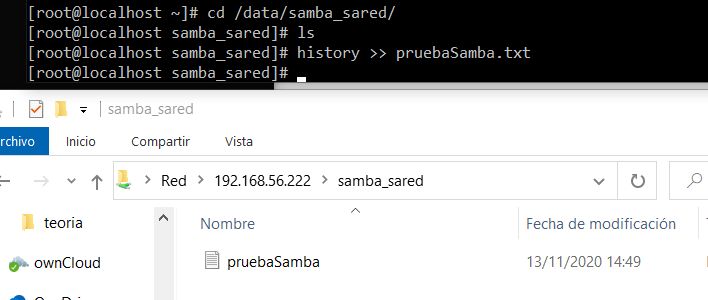
Damos permiso al firewall para que habilite samba:

firewall-cmd --add-service=samba --zone=public –permanent

En el host abrimos la ventana de ejecución win+R e introducimos:

\\<ipServidor>

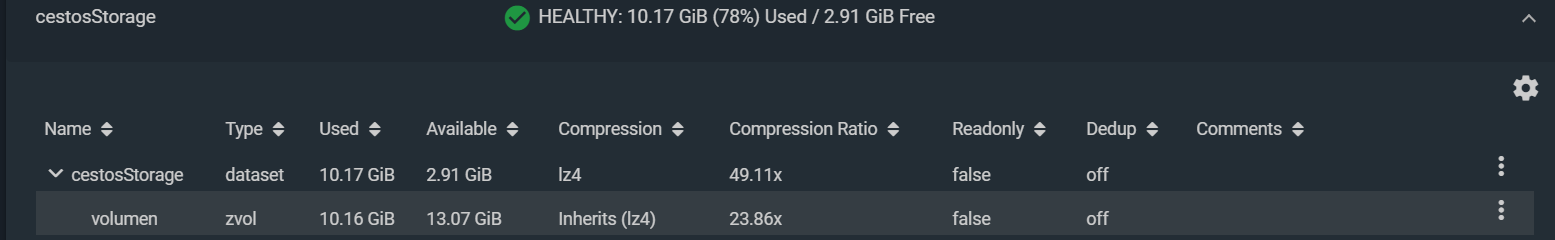
Nos abrirá la carpeta compartida y ya podríamos compartir archivos:



### FreeNAS

Nos descargamos la iso de freeNass y creamaos una maquina con esa iso, instalamos la maquina y nso metemos en a url que nos indica la maquina FreeNAS

En ella nos creamos una pool con estas caracteriaticas :

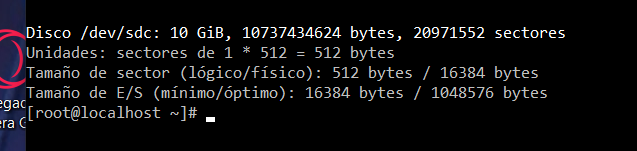


En la página, habilitamos el servicio ISCSI, y creamos un disco con dicha característica, luego nos vamos a la maquina CentOS y nos instalamos ISCSI dnf install iscsi-utils

Después ejecutamos los siguientes comandos:

Una vez hecho, montamos el disco nuevo con el proceso explicado en el apartado de [particiones](#_Particionado)

Como resultado tenemos el disco sdc instalado y montado:



## Impresión remota

### CUPS

Para instalar cups en centos necesitamos instalar una impresora, en nuestro caso la cups-pdf, para ello ejecutaremos los siguientes comandos:

yum update

yum -y install cups gcc gcc-c++ cups-devel tar wget

wget https://www.cups-pdf.de/src/cups-pdf\_3.0.1.tar.gz

tar -xvf cups-pdf\_3.0.1.tar.gz

cd cups-pdf-3.0.1/src/

gcc -O9 -s cups-pdf.c -o cups-pdf -lcups

chmod 700 cups-pdf

cp -p cups-pdf /usr/lib/cups/backend/

cd ../extra

cp cups-pdf.conf /etc/cups/

cp CUPS-PDF\_opt.ppd /usr/share/cups/model/

firewall-cmd –zone=public –add-port=631/tcp –permanent

firewall-cmd –reload

sed -i “s/Allow \@LOCAL/Allow all/g” /etc/cups/cupsd.conf

systemctl restart cups

systemctl enable cups

cupsctl –remote-admin

después modificamos el fichero /etc/cups/cupsd.conf y añadimos:

Port 631

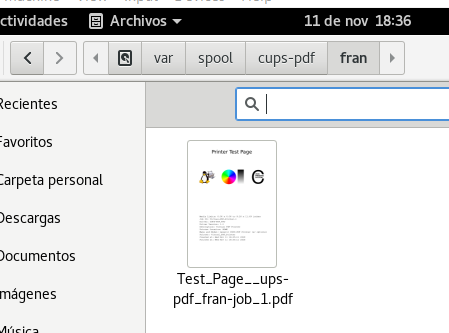
Listen localhost:631

Listen 192.168.56.222:631

Después entramos en <https://192.168.56.222:631/> y en administración, añadir impresoras

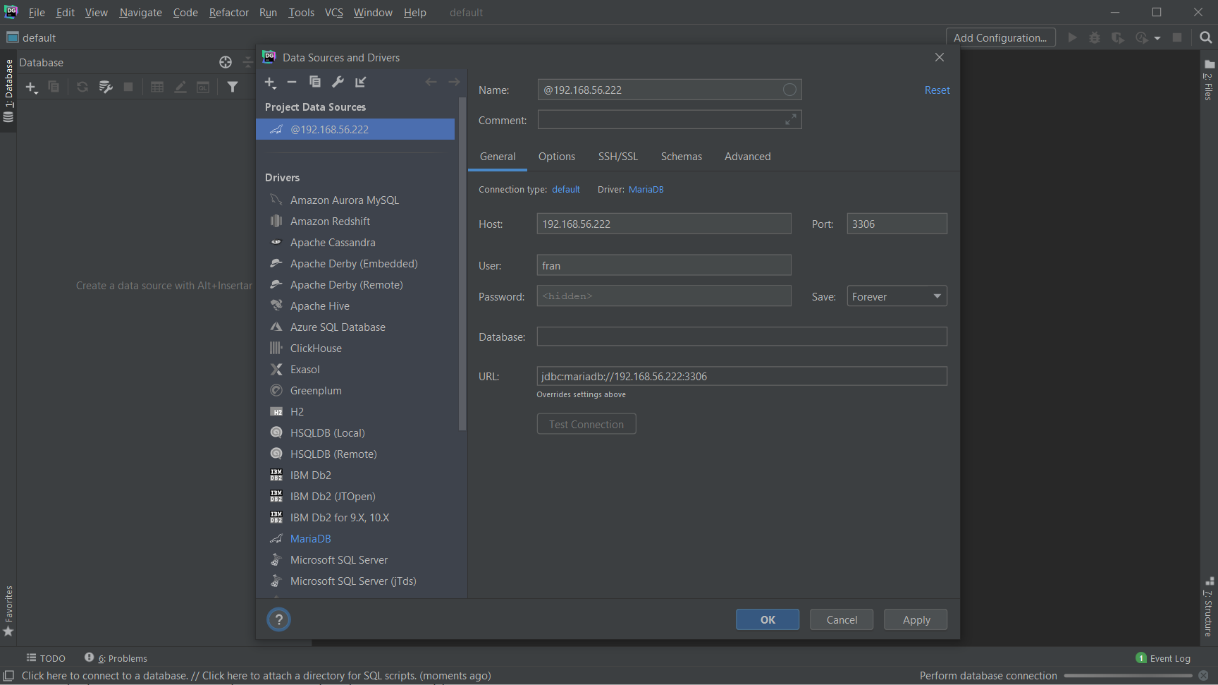


Añadimos la CUPS\_PDF y cuando imprimimos una pagina de prueba nos aparecerá en el directorio /var/spool/cups-pdf

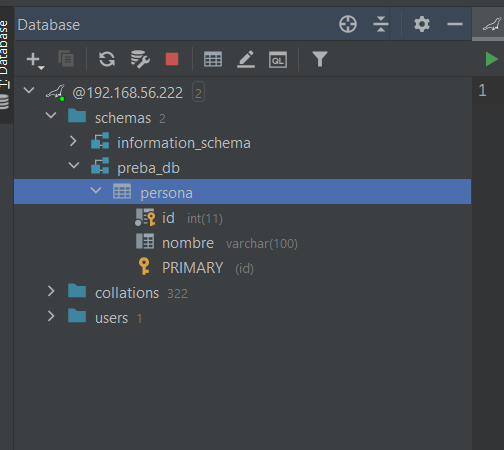


## Base de datos y web servicies

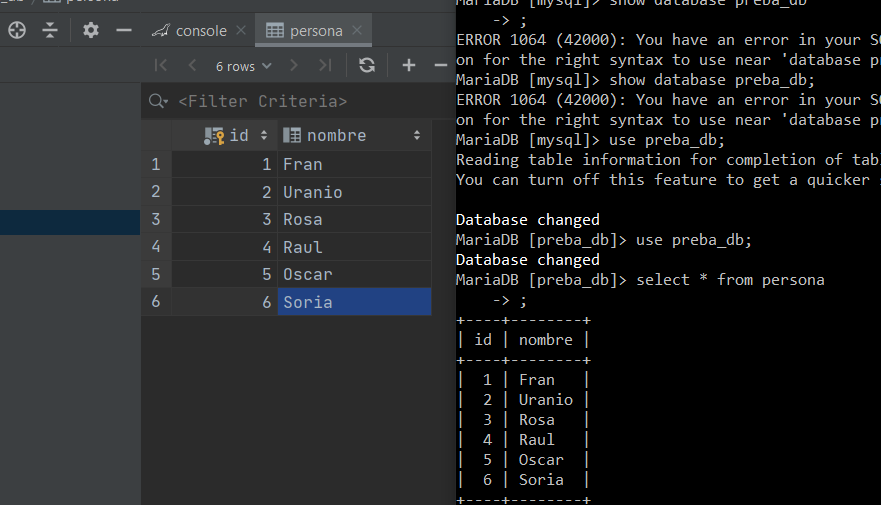
### Base de datos

Utilizando el proceso de instalación utilicado en OwnCloud, creamos una base de datos, luego nos descargamos el cliente de bases de datos DataGrep y colocamos la base de datos:

Nos aparecerán nuestras bases de datos a la izquierda:



Con la interfaz de DataGrep nos cramos unos usuarios:



En centos nos creamos una queri en php para que nos muestre la base de datos:

/var/www/html/index.php

<?php

$servername = "localhost";

$username = "fran";

$password = "fran";

$dbname = "preba\_db";

//Crear conexion

$conn = new mysqli($servername, $username, $password, $dbname);

//Comprobar conexion

if($conn->connect\_error) {

die("Connection failed: " . $conn->connect\_error);

}

$sql = "SELECT \* FROM persona";

$result = $conn->query($sql);

if($result->num\_rows > 0) {

//salida de cada fila

while($row = $result ->fetch\_assoc()) {

echo "id: " . $row["id"]. " - Nombre: " . $row["nombre"]. "<br>";

}

} else {

echo "0 results";

}

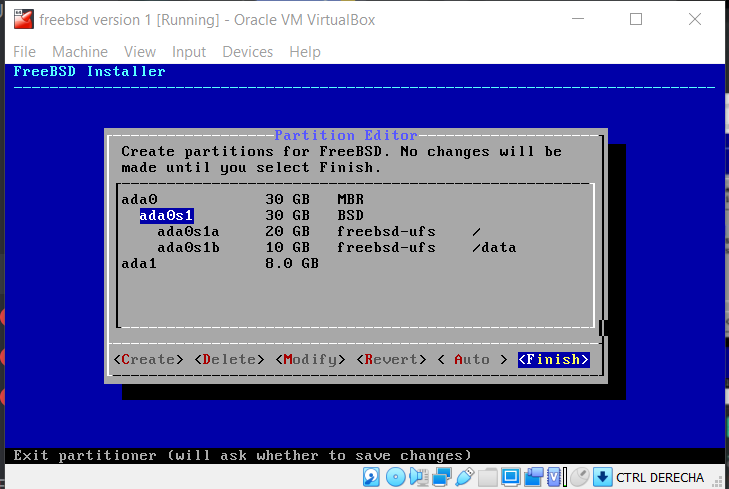
$conn->close();

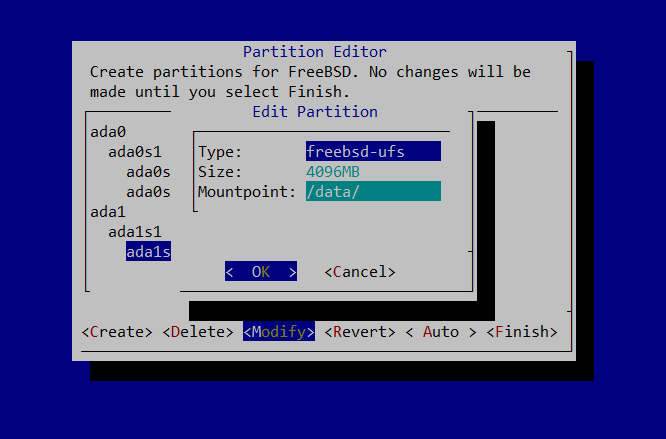
?>



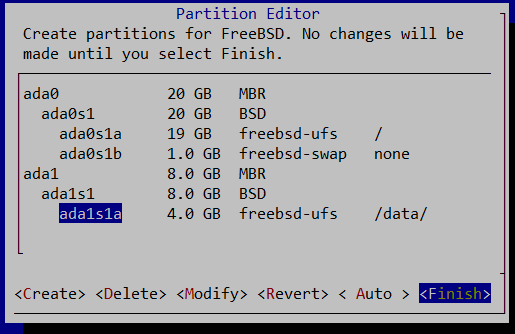
# FreeBSD

## Particionado

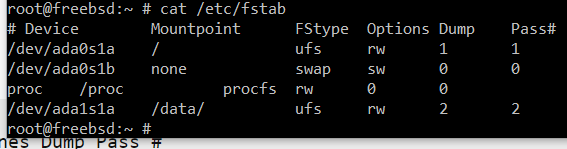
Ejecutamos el comando “sade”, y nos aparecerá una ventana, en ella podemos crear particiones en los dicos que queramos

Seecionamos el disco ada1 y lo particionamos:

Como resultado tenemos las siguientes particiones:



Comporbamos la partición con cat /etc/fstab

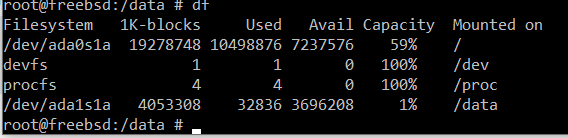


Luego tenemos que montarlo en la carpeta data:

mkdir /data/

mount -a

después ejecutamos df en la carpeta data y nos aparecerá la partición montada



## Administración remota

### SSH/SFTP/SCP

SSH viene instalado en BSD, solo hay que entrar en el archivo /etc/rc.conf y colocar sshd\_enable="YES”, para configurarlo hay que acceder al fichero /etc/ssh/sshd\_config, en el aparece toda la configuración de SSH, en el añadimos lo siguiente:

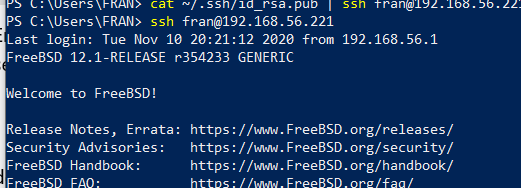
* Añadir: AllowUsers <usuarios permitidos>
* Modificar PubkeyAuthentication yes
* Modificar: UsePAM yes
* Modificar: PermitRootLogin no

Después se conecta a BSD colocando en la cmd del host ssh <usuario>@<ip servidor>.

Para la clave publica en este caso solo habría que enviarla puesto que ya la creamos para CentOS, la enviaremos esta vez sin el mkdir puesto que ya viene creada la carpeta en BSD:

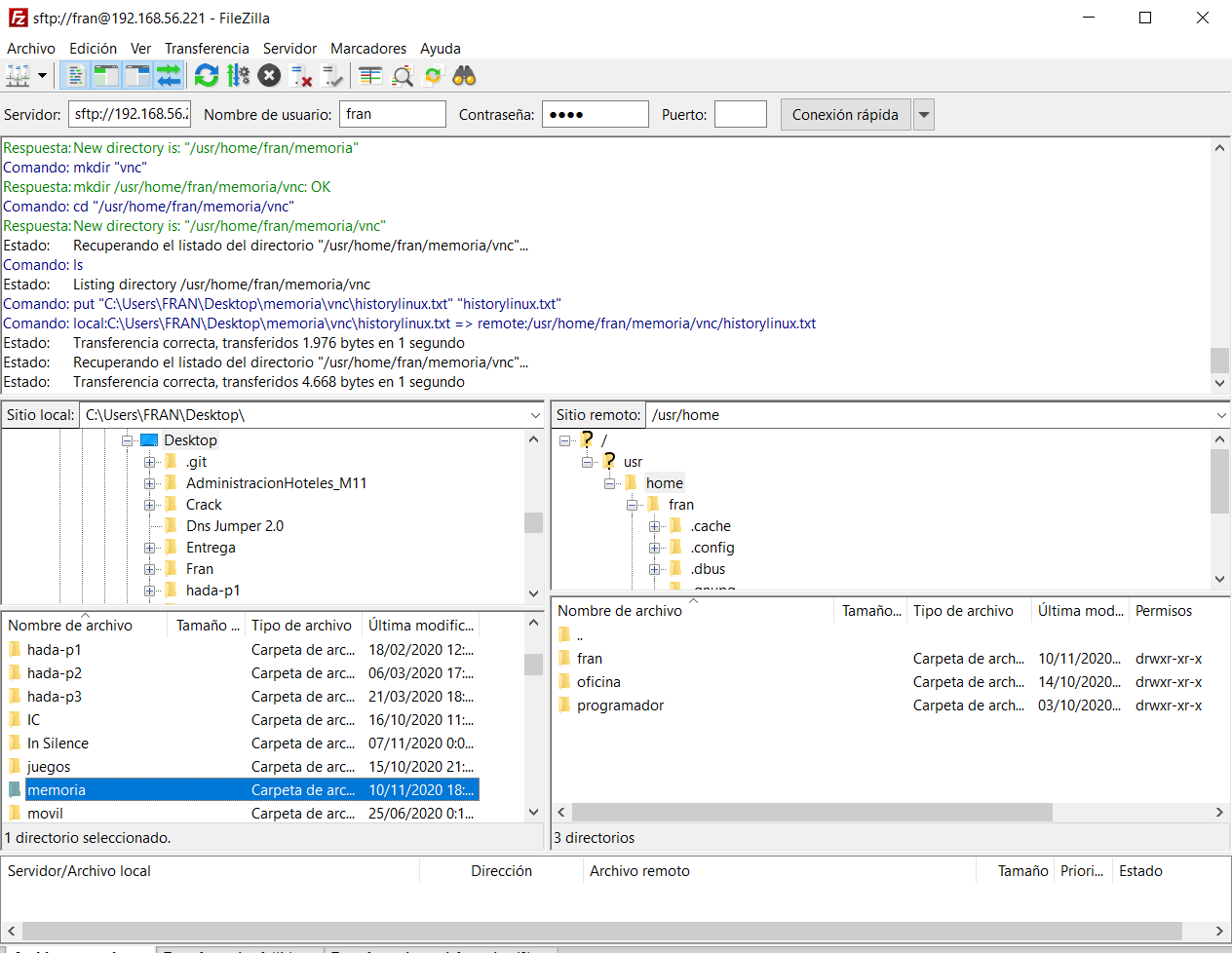
cat ~/.ssh/id\_rsa.pub | ssh <usuario>@<ip servidor> " cat >> ~/.ssh/authorized\_keys"

y ya nos podríamos conectar:



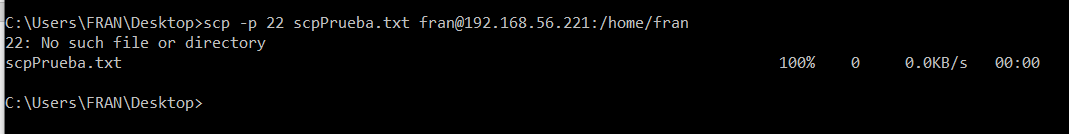
Para realizar la conexión SFTP, se ha utilizado el programa de gestión de archivos filezilla.

Nos conectamos a nuestro servidor con la ip usuario y contraseña

Y para pasar archivos del host al servidor o viceversa solo hay que pinchar y arrastrar, por ejempo, he pasado un archivo prueba al servidor

Para comprobar el servicio SCP he enviado un documento de prueba

scp -p <puerto> <archivo> <usuario>@<ip>:<ruta destino>

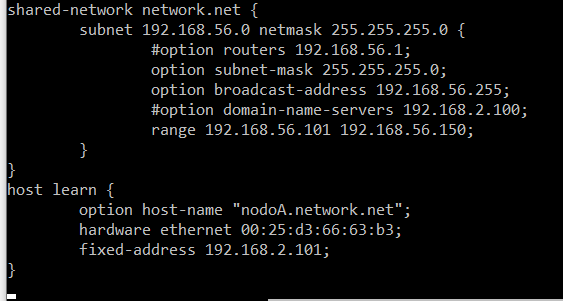


## Networking

### DHCP

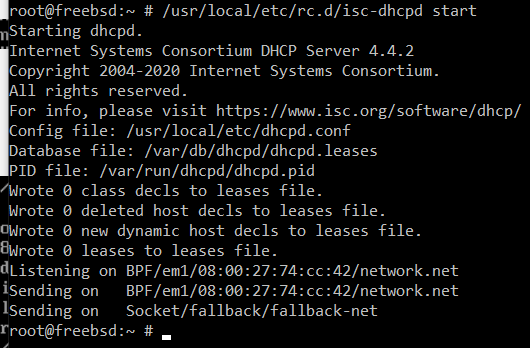
Para usar dhcp en FreeBSD, hemos de instalar el servicio: pkg install isc-dhcp44-server.

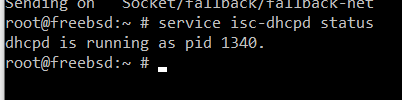
Modificamos el archivo /usr/local/etc/dhcpd.conf y añadimos lo siguiente:



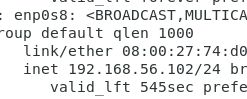
Modificamos el archivo -c /etc/rc.conf y añadimos dhcpd\_enable=”YES” para habilitar dhcp y dhcpd\_ifaces=”em1″ para asignad la interfaz host-only

Activamos el servicio /usr/local/etc/rc.d/isc-dhcpd start



Comprobamos su estado:

He arrancado una maquina debian para comprobar el servicio y vemos que funciona



### Git server

Para instalar un servidor git hemos de instalarnos previamente el gestor git:

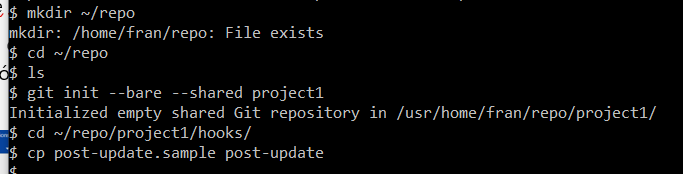
pkg install git

A continuación, nos creamos un repositorio en una carpeta:

mkdir ~/repo

cd ~/repo

git init --bare --shared project1

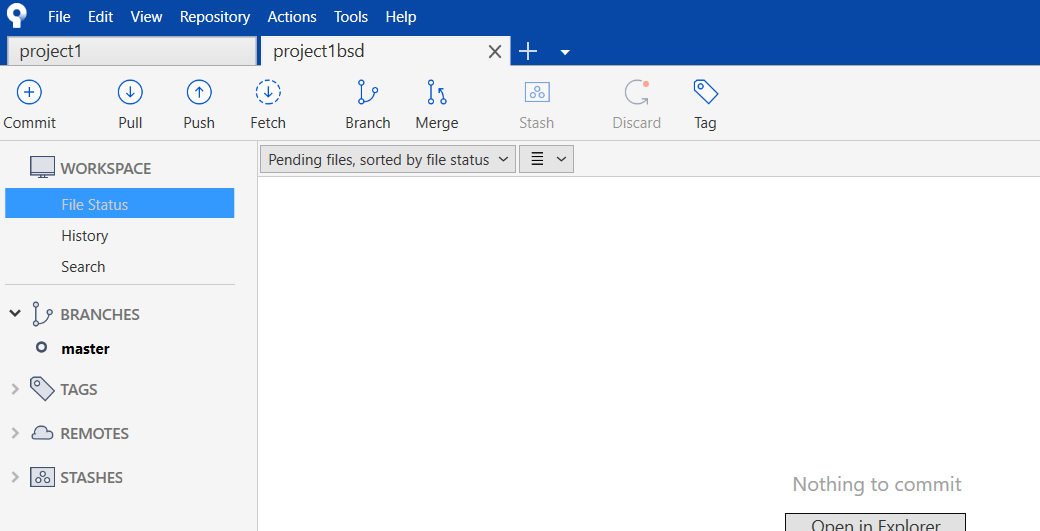


A continuación, habilitamos post-update hook de la siguiente manera:

cd ~/repo/project1/hooks/

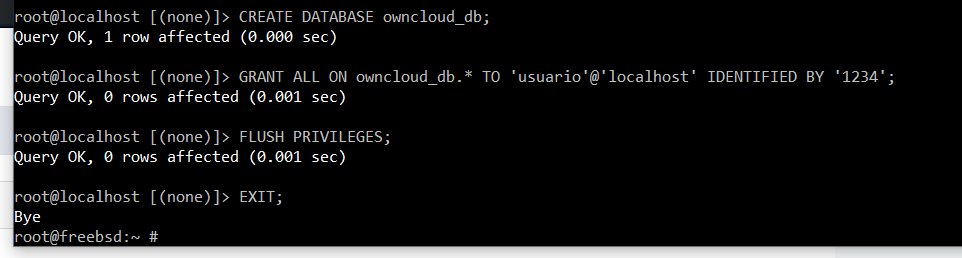
cp post-update.sample post-update

Una vez hecho todo esto, nos descargamos un cliente, en mi caso SourceTree, colocamos la dirección del repositorio en el instalador y ya podríamos trabajar con el:

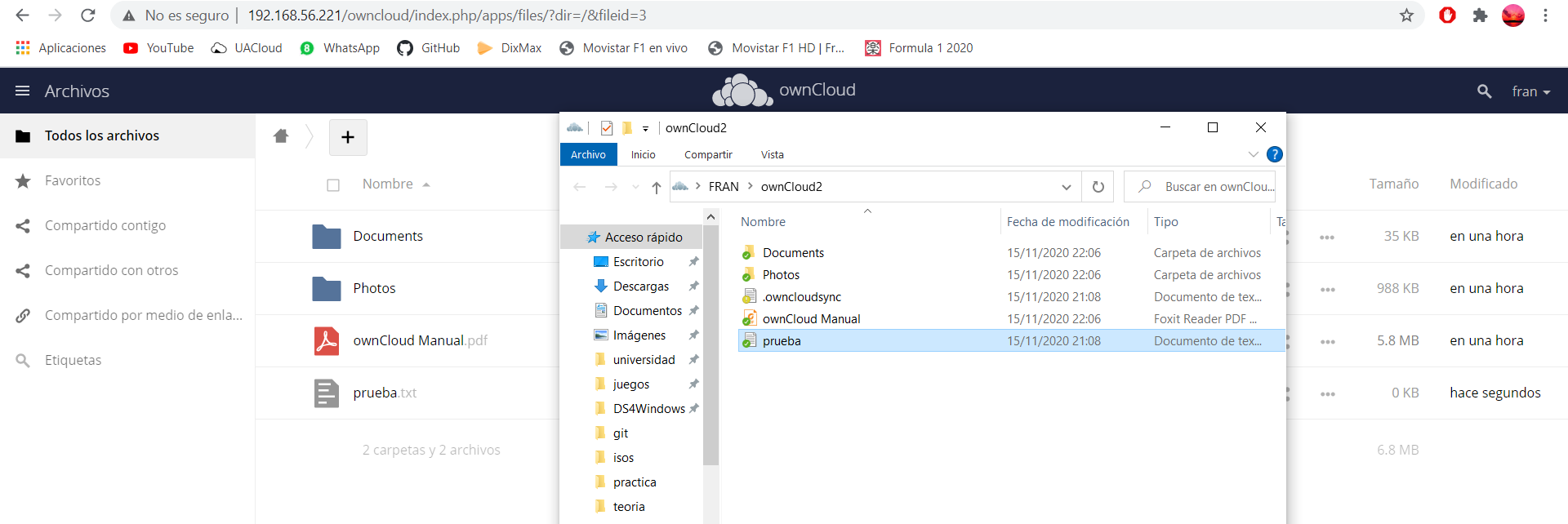


### OwnCloud server

Para implementar el servicio owncloud, primero tenemos que crear una pila FAMP, si seguimos los pasos de [esta web](https://comoinstalar.me/como-instalar-famp-en-freebsd-12/) instalaremos php, mariadb, apache y sus respectivas configuraciones



Después para configurar owncloud, seguiremos los pasos de [esta web](https://comoinstalar.me/como-instalar-owncloud-en-freebsd-12/) donde instalaremos owncloud lo vincularemos a la base de datos y configuraremos php y apache para activar el servicio



## Servicios de impresión

### Cups

Configuramos el fichero /etc/cups/cupsd.conf y añadimos:

Port 631

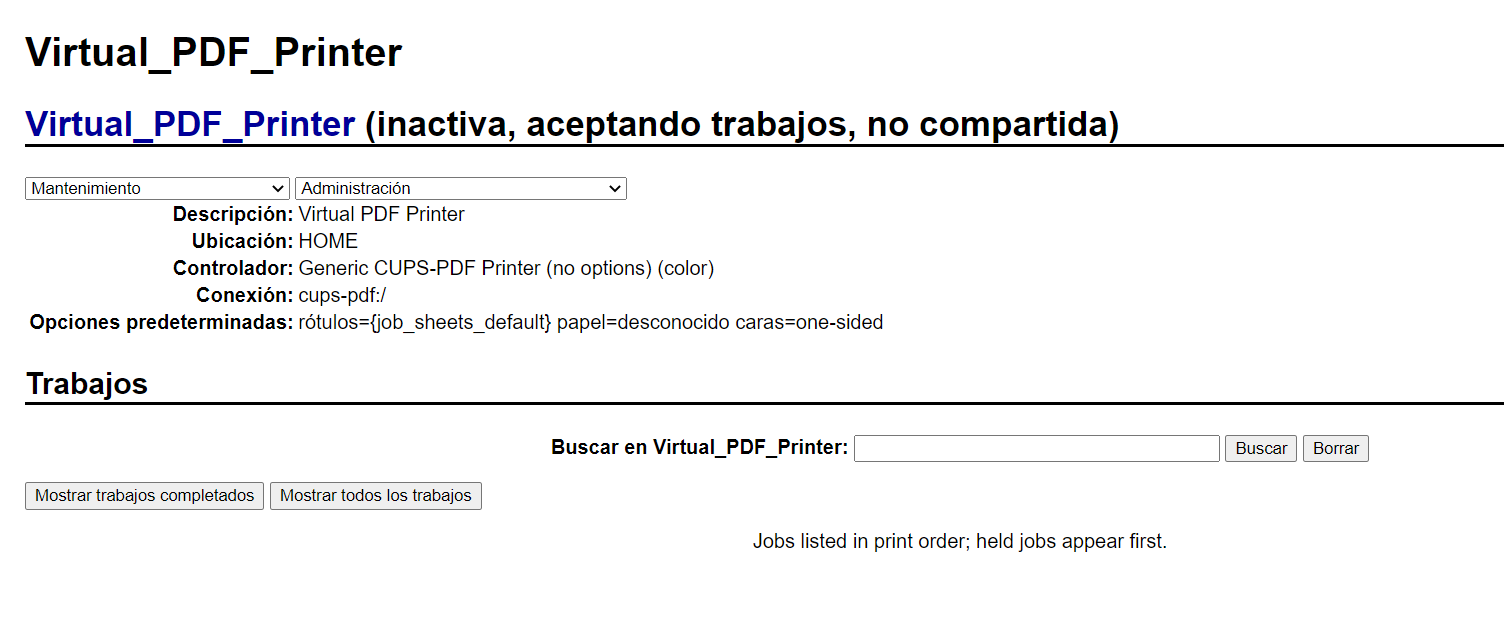
Listen localhost:631

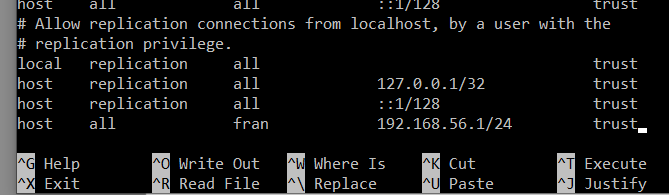
Listen 192.168.56.221:631

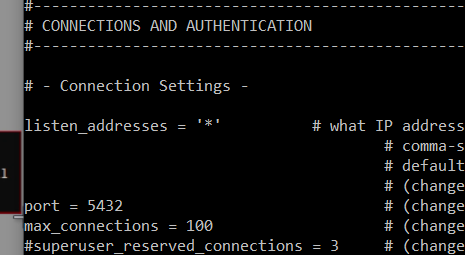
Nos instalamos Cups PDF con pkg install cups-pdf

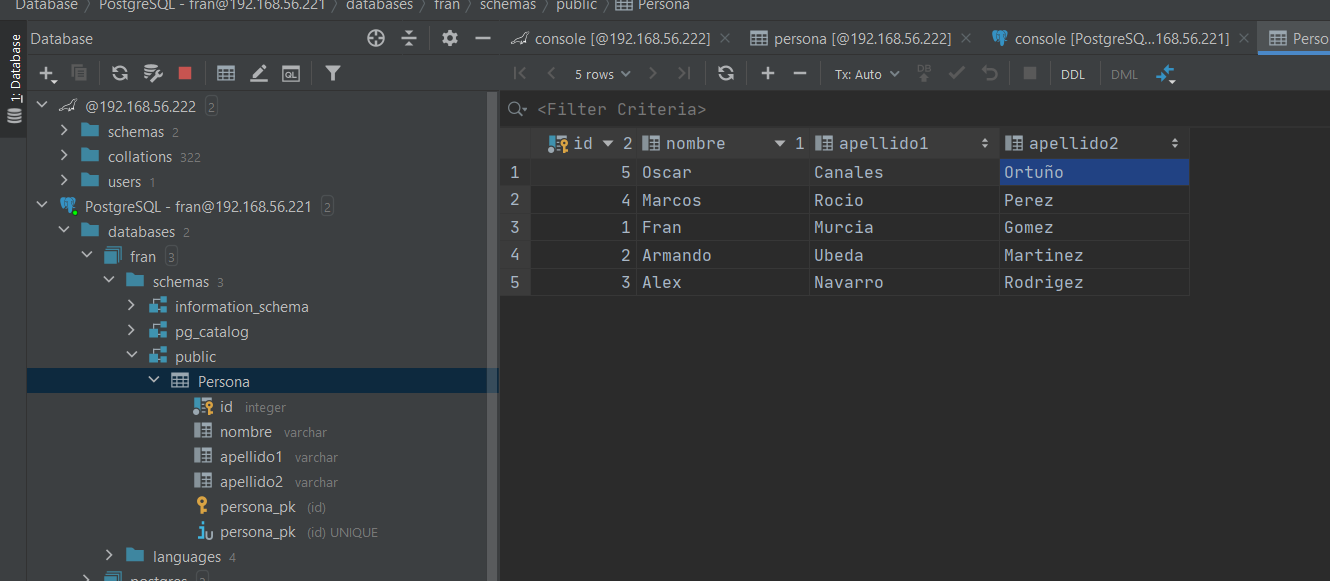
Configuramos el fichero /etc/cups/cups-pdf.conf y añadimos:

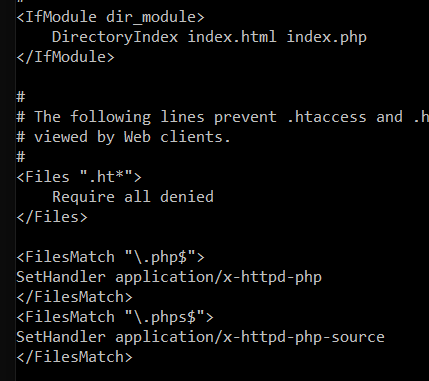
Out${HOME}

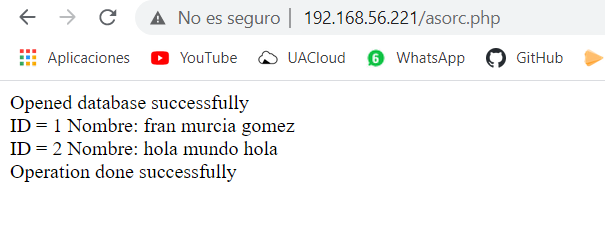
Depues entramos en https://192.168.56.22:631/ y en administración, añadir impresoras y añadimos una impresora genérica de tipo cups-pdf

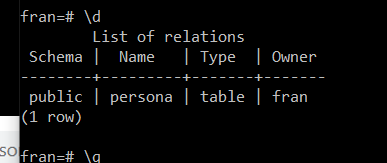












# Windows

### DHCP

