

# UNIVERSIDAD CAPITÁN GENERAL GERARDO BARRIOS.



## **“Configuraciones Iniciales en RHEL8”**

*Configuración de Clústeres y servidores*

### **Requerimientos e indicaciones:**

- Redhat Enterprise Linux Enterprise instalado versión 8.
- Tener una cuenta registrada en <https://developers.redhat.com/> configurado su usuario y contraseña.

ne

### **Generalidades:**

Será importante contar con la imagen iso (DVD) sorry por los que descargaron la versión 7.9, esta no nos servirá de mucho por las actualizaciones de los repositorios, disculpas cuando le dije la 7.9.

Enlace de descargar: <https://developers.redhat.com/products/rhel/download>

### **¿Porqué RHEL8 y no Debian, CentOS, Ubuntu, bla, bla, bla...?:**

RHEL es una distribución con un modelo de suscripciones muy robusto y confiable que mantiene entornos de producción de manera muy estable y que está alineado a proyectos modernos como Ansible, Docker, Docker Compose, Kubernetes, Quarkus, JavaEE, SpringBoot, Istio, Knative, Apache Kafka, entre otros por lo que estas herramientas corren de forma muy estable en esta plataforma. NOS INTERESA ESTABILIDAD no gastar mucho tiempo en setear entornos más bien una plataforma robusta que nos permita trabajar de lleno con las herramientas.

Redhat tiene además un modelo de certificaciones que también puede ser aprovechado por aquellos que quieran validar sus conocimientos posteriormente, estas certificaciones son muy reconocidas y solicitadas.

	VM ↔ HOST	VM1 ↔ VM2	VM → INTERNET	VM ← INTERNET
Solo anfitrión	SI	SI	NO	NO
Interna	NO	SI	NO	NO
Adaptador puente	SI	SI	SI	SI
NAT	NO	NO	SI	Reenvío de puertos
Red NAT	NO	SI	SI	Reenvío de puertos

## PARTE I: Setting inicial.

### 1. Conociendo las limitantes iniciales:

Vamos a validar nuestras suscripciones para ello emitimos el comando:

```
subscription-manager list
```

De momento no deberíamos tener ninguna suscripción valida, ya que no la hemos agregado, es importante tener ya nuestra cuenta antes mencionada.

Al emitir el siguiente comando:

```
yum update
```

**Nos debería devolver error ya que no hay ningún repositorio del cual actualizar nuestro entorno.**

### 2. Agregando nuestra suscripción:

Emitimos el siguiente comando:

```
subscription-manager register  
subscription-manager attach --auto
```

Esto nos solicitará usuario y contraseña que deben ser validos estos datos deben ser los que se obtienen en <https://developers.redhat.com/> de manera que ya contaremos con la suscripción y podremos agregar los repositorios posteriormente.

### 3. Agregando repositorios:

Una vez agregada la suscripción agregamos los repositorios iniciales de redhat de los cuales podremos descargar software y actualizaciones de seguridad.

Los repositorios varían de versiones por lo que se tiene la siguiente tabla tomada de la página oficial de redhat:

	Base Repos	Optional/Development Repos
RHEL 6	rhel-6-server-rpms	rhel-6-server-optional-rpms
RHEL 7	rhel-7-server-rpms	rhel-7-server-optional-rpms
RHEL 8	rhel-8-for-x86_64-baseos-rpms rhel-8-for-x86_64-appstream-rpms	codeready-builder-for-rhel-8-x86_64-rpms

*Illustration 1: <https://access.redhat.com/solutions/265523>*

```
subscription-manager repos --enable=rhel-8-for-x86_64-baseos-rpms
```

Y luego agregamos el repositorio opcional:

```
subscription-manager repos --enable=codeready-builder-for-rhel-8-x86_64-rpms
```

Esto nos permitirá instalar el software y actualizaciones de seguridad de Redhat.

#### **4. Instalar actualizaciones de seguridad de RHEL:**

```
yum update --security
```

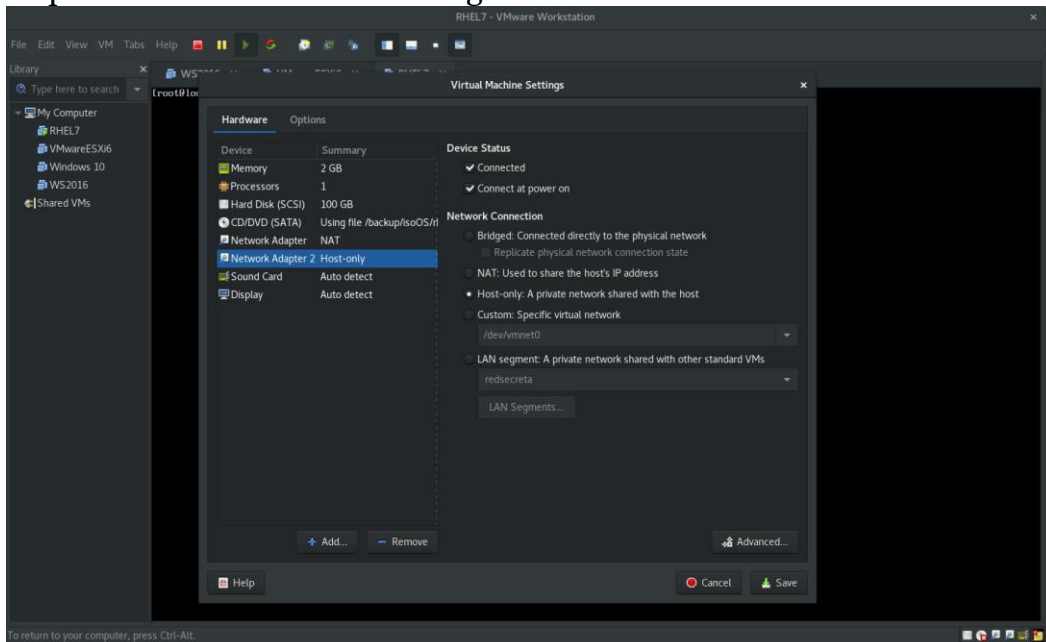
#### **5. Actualizar todas las aplicaciones:**

```
yum update
```

## PARTE II: Red solo anfitrión

### 1. Agregamos la tarjeta de red:

Vamos a VM → Settings → Agregar → Tarjeta de red. Nos deberá aparecer esta pantalla en la que seleccionaremos los settings como se muestra:



### 2. Obteniendo el nombre del adaptador en Redhat

Debemos obtener el nombre simbólico del adaptador:

```
ip addr
```

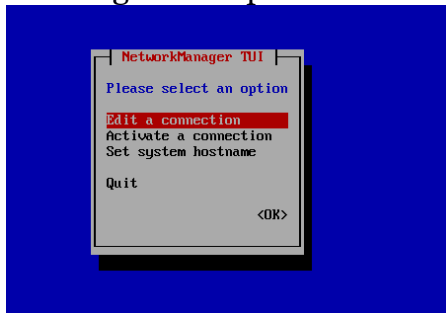
Esto nos deberá devolver las ip, de las cuales la última listada será nuestra red solo anfitrión:

En mi caso **ens36**

### 3. Setting de IP

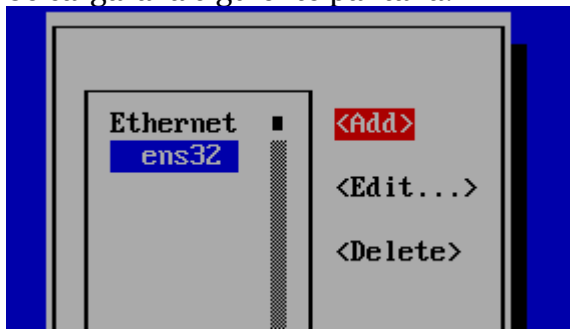
```
nmtui
```

Nos cargará esta pantalla:



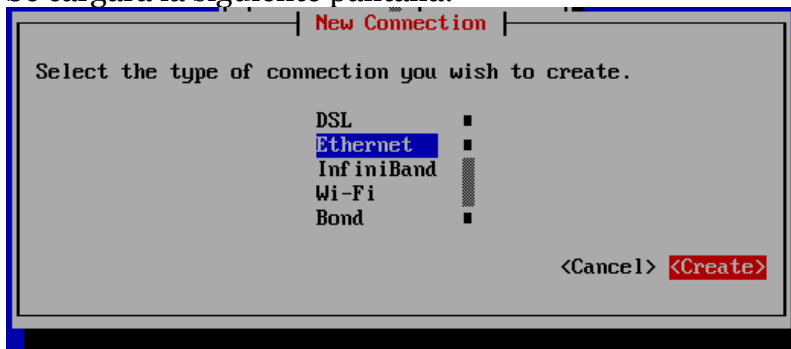
Damos enter:

Se cargará la siguiente pantalla:



Presionando un tab llegamos a Add puesto que no nos aparece el adaptador **ens36** que acabamos de agregar.

Se cargará la siguiente pantalla:



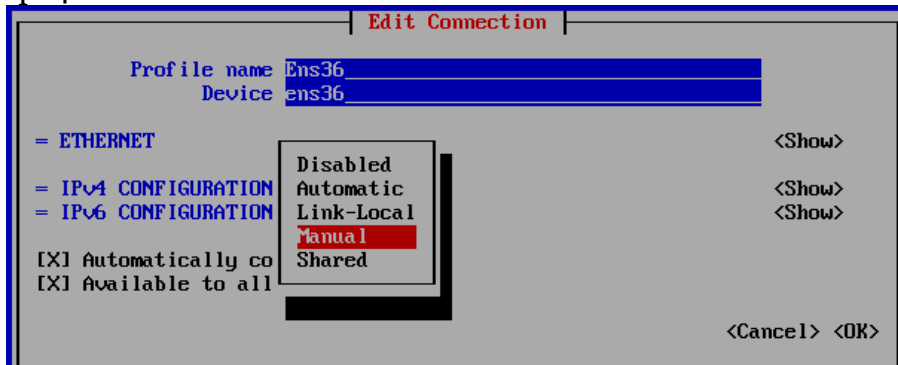
Seleccionamos **Ethernet** y con **tab** llegamos hasta la opción **Create** y presionamos **ENTER**



Setteamos nuestro nombre simbolico:

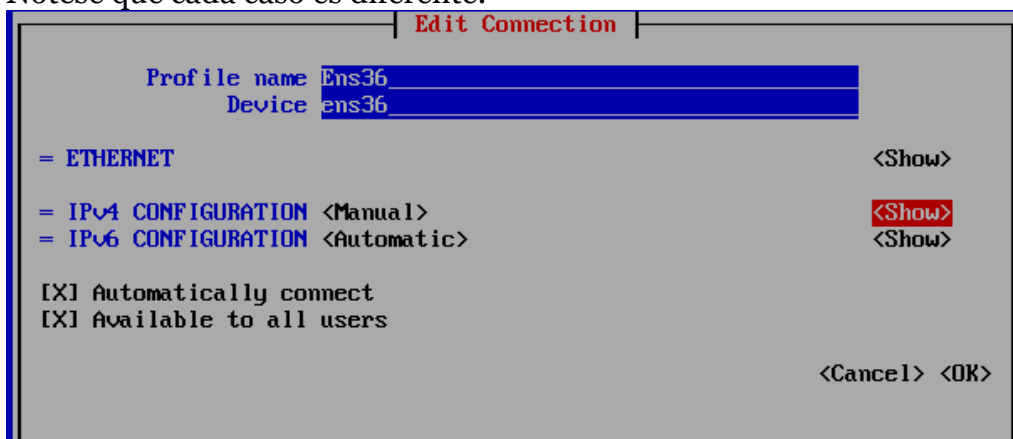
- Profile name: **Ens36**
- Device: **ens36**

Posteriormente con tab nos posicionamos en:  
Ipv4 CONFIGURATION <automatic> damos enter:

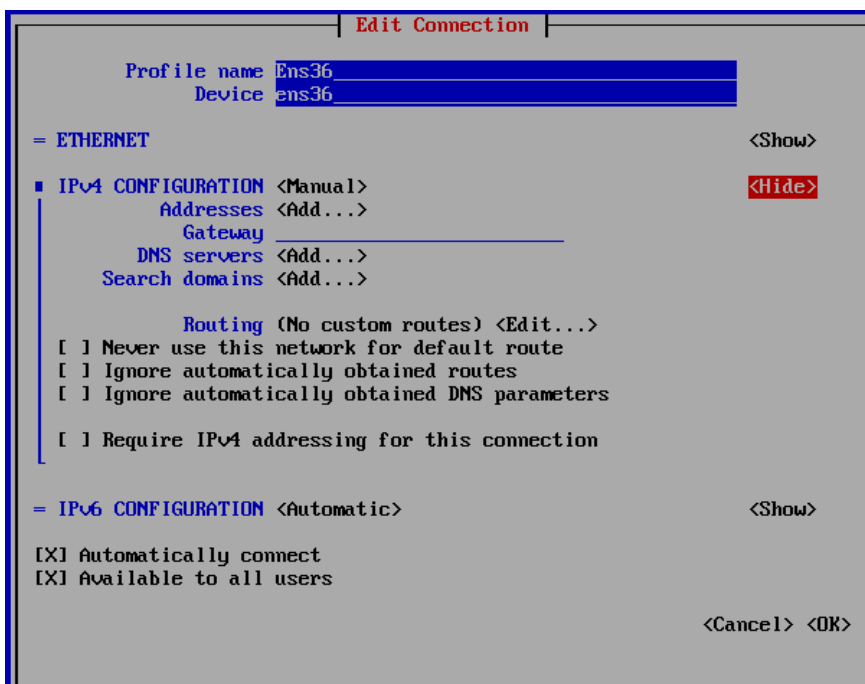


Seleccionamos manual y seteamos la IP de acuerdo a la IP de la tarjeta de red solo anfitrión de nuestra computadora en mi caso mi computadora tiene la IP: **192.168.23.1** por lo cual yo utilizaré la **192.168.23.2**.

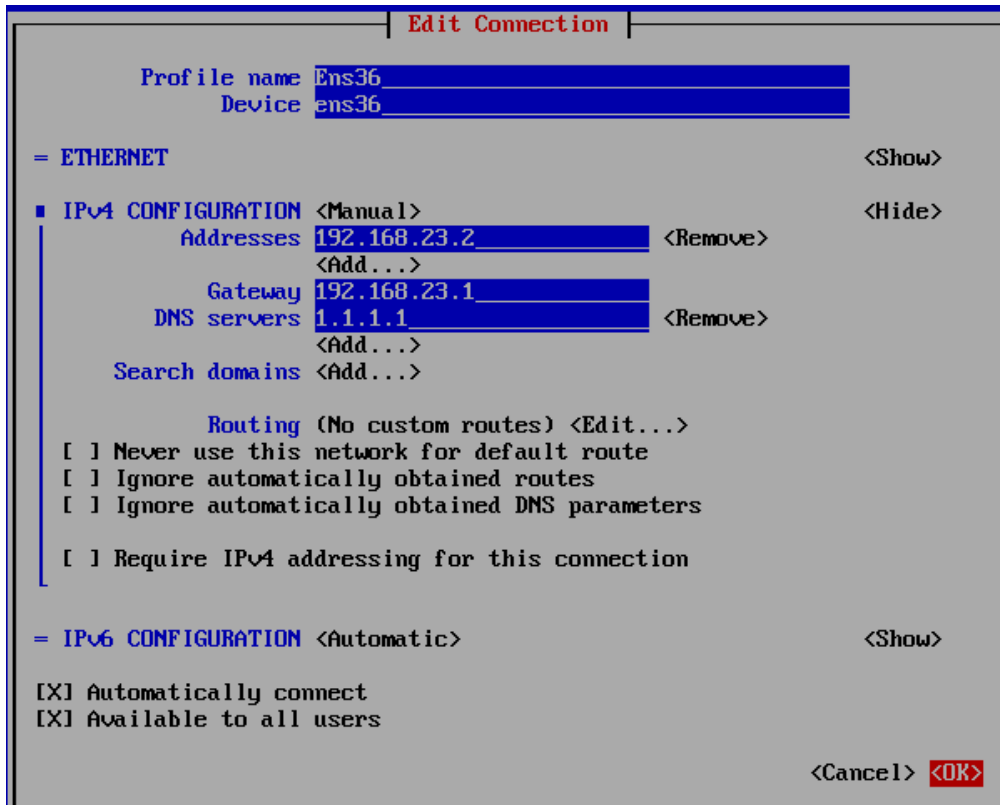
Notese que cada caso es diferente:



Una vez este manual damos tab hasta llegar a **Show** y luego nos mostrará una pantalla similar a la siguiente:



Deberemos navegar con tab agregando información hasta que nos quede de la siguiente manera:



Navegamos hasta **OK** y damos enter y las configuraciones mostradas serán aplicadas.

Finalmente salimos de la interfaza hasta volver a nuestra querida consola.

#### 4. Pruebas de configuración de IP:

```
ip addr
```

Debe mostrar ya la IP seteada de forma estatica.

Probar la conectividad:

```
Ping 192.168.23.1
```

Nos debe devolver el ping correctamente.

Pero lo más relevante de todo es el comando:

```
ip route
```

Notese la salida:

```
[root@localhost alex]# ip route
default via 172.16.86.2 dev ens32 proto dhcp metric 100
default via 192.168.23.1 dev ens36 proto static metric 101
172.16.86.0/24 dev ens32 proto kernel scope link src 172.16.86.136 metric 100
192.168.23.0/24 dev ens36 proto kernel scope link src 192.168.23.2 metric 101
[root@localhost alex]#
```

Estamos directamente conectados a la red 192.168.23.0/24 vía ens36.

Cabe destacar que esas configuraciones realizadas con la UI gráfica pueden ser agregadas directamente desde un archivo de configuración sin tanta pantalla y pantalla....

Ver el siguiente enlace:

<https://www.cyberciti.biz/faq/howto-setting-rhel7-centos-7-static-ip-configuration/> #Method 1 Debe hacerse una lectura comprensiva no emitir los comandos y rogar a que funcionen, al contrario debe revisarse bien que información es la que debo modificar para adaptar la solución a mi caso particular. (Esto es aplicable para la mayor parte de comandos y guías que uno se puede encontrar para el entorno de Linux).



## PARTE III: Instalación de Servidor samba:

### 1. Creando un nuevo espacio de sitio

```
yum install -y samba* net-tools
```

Verificamos puertos abiertos:

```
netstat -tln
```

Verificamos e estado de samba:

```
systemctl status smb
```

Habilitamos samba para que arranque al iniciar la maquina y arracamos también el servicio:

```
systemctl enable smb.service && systemctl start smb.service
```

Verificamos puertos abiertos:

```
netstat -tln
```

Verificamos e estado de samba:

```
systemctl status smb
```

### 2. Vamos a ver las configuraciones de nuestro firewall.

Redhat utiliza **firewalld** como Firewall por lo que tendremos que listar nuestras reglas en la zona publica:

```
firewall-cmd --zone=public --list-services
```

Como podremos ver en las redes publicas que corresponde a **ens32** y **ens36** solo tendremos acceso a para ssh y también para icmp.

### 3. Tenemos a smb:

```
firewall-cmd --zone=public --add-service=samba-client --permanent  
firewall-cmd --zone=public --add-service=samba --permanent  
firewall-cmd --reload
```

### 4. Listamos nuevamente las configuraciones.

```
firewall-cmd --zone=public --list-services
```

Veremos que smb-client será ya permitido.

## PARTE IV: Configuración

### 1. Carpeta a compartir:

```
adduser --system shareuser  
mkdir /shared_folder/  
chown shareuser:shareuser /shared_folder/
```

### 2. Configuración principal:

Emitir en la terminal el siguiente comando:

```
vi /etc/samba/smb.conf
```

### 3. Configuraremos lo siguiente al final del archivo:

```
[shared_folder]  
comment = My Shared folder in samba server  
path = /shared_folder  
create mask = 0644  
directory mask = 0755  
force user = shareuser  
writeable = yes
```

### 4. Finalmente vamos a probar la integridad del archivo de configuración:

```
testparm
```

Deberá devolvernos una salida como la siguiente:

```
[root@localhost alex]# testparm  
Load smb config files from /etc/samba/smb.conf  
Loaded services file OK.  
Server role: ROLE_STANDALONE  
  
Press enter to see a dump of your service definitions
```

Nótese que la sintaxis como muestra la captura ha sido detectada como correcta.

**Es importante recalcar que el usuario **shareuser** no será un usuario de acceso, solo será un usuario del sistema que nos ayudará a acceder a la data ya que el directorio **/share\_user** es propiedad de dicho usuario por lo cual no tendremos problemas de permisos.**

## PARTE IV: Creación de usuarios en Samba.

### 1. Creamos nuestro primer usuario de samba

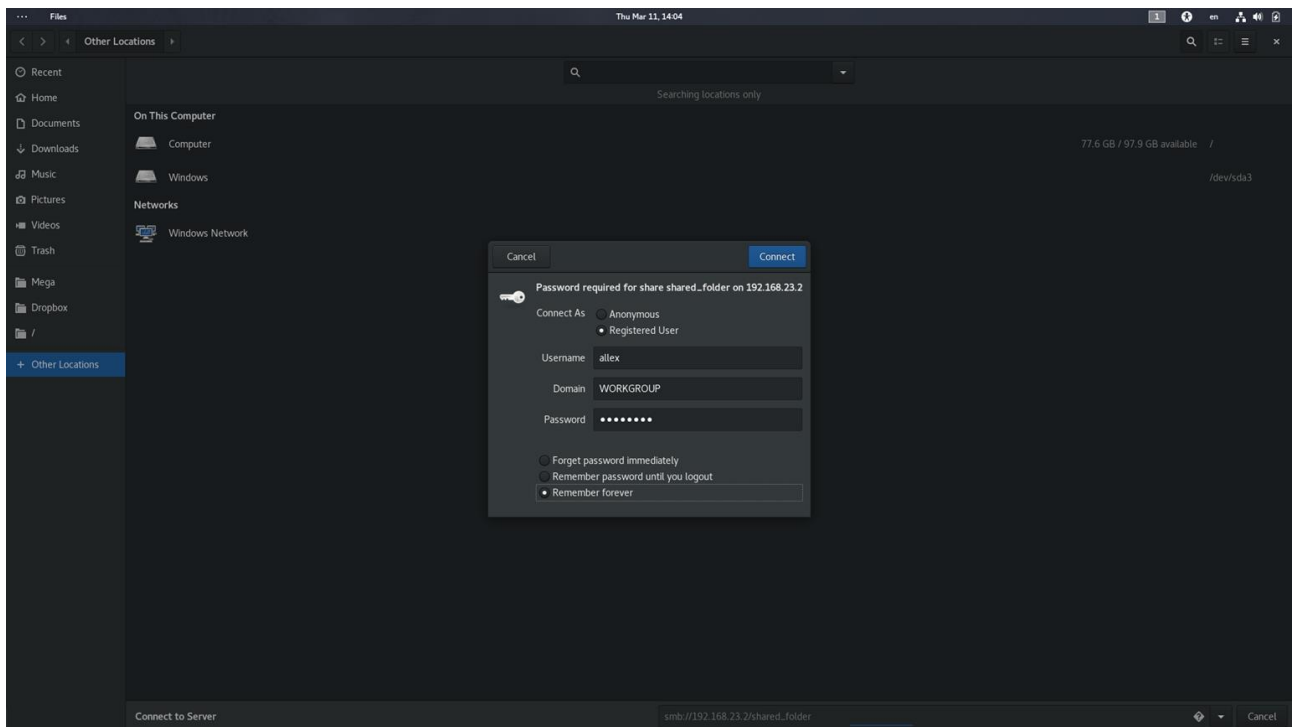
```
smbpasswd -a sackno
```

### 2. Verificando acceso al servidor samba

Verificación de acceso en la terminal:

```
smbclient -L localhost -U sackno
```

## Parte VI: Conexión desde Linux



**A pesar de todas las configuraciones notará que el usuario no tiene permisos de acceder completamente al sistema por lo cual es importante tener en cuenta lo siguiente al trabajar en Redhat.**

Vamos a ver en detalle que pasa:

```
ps -eZ | grep smb
```

**Nos devolverá algo similar a lo siguiente:**

```
system_u:system_r:smbd_t:s0 9653 ? 00:00:00 smbd
```

La parte del contexto **smbd\_t**, es el tipo de contexto que puede leer. Un tipo define un dominio de procesos y un tipo de archivos que se pueden leer y escribir. En este caso podemos decir que el **smbd process** esta corriendo en el dominio **smbd\_t**.

**¿Qué archivos puede leer smbd?**

Por defecto puede leer archivos etiquetados con el contexto **samba\_share\_t** de modo que los directorios que no esten etiquetados de esta manera no podrán ser leídos ni escritos por **smbd**.

**Veamos las etiquetas que tiene /shared\_folder/**

```
ls -ld -Z /shared_folder/
```

Estará etiquetado como **default\_t** ya que pertenece al directorio raíz y no se le podrá asignar un contexto de servicio, en caso que un servicio desee acceder a dicho folder tendremos que actualizar su etiqueta a una que el dominio del servicio pueda acceder.

**Instalar paquete de utilidades para el trabajo con SELinux:**

```
yum install policycoreutils
```

**¿Como podemos etiquetar este directorio a **samba\_share\_t**?**

```
chcon -R -t samba_share_t /shared_folder
```

De modo que esto debería habilitar a samba para que el servicio smbd sea capaz de leer y escribir en el directorio.

## ¿Que otras opciones tenemos?

SELinux y Samba también poseen una funcionalidad llamada que permiten cambiar la forma en la que se delegan permisos a nivel de servicios y procesos, esto es conocido como **BOOLEANOS**.

Los booleanos son formas de configuración en tiempo de ejecución que SELinux soporta y que permiten hacer que los procesos sean más flexibles en cuanto a la forma en la que acceden a diferentes recursos del sistema.

### Para comprender que pasa veremos la siguiente tabla de booleanos para SAMBA:

<b>smbd_anon_write</b>	Permite escribir en un directorio publico en el que se puedan compartir archivos diversos.
<b>samba_create_home_dirs</b>	Samba crea directorios independientes para cada usuario.
<b>samba_domain_controller</b>	Actua como un controlador de dominos y permite ejecución de comandos como useradd, groupadd, and passwd.
<b>samba_enable_home_dirs</b>	Comparte el directorio home de los usuarios.
<b>samba_export_all_ro</b>	Permite lectura de todos los directorios compartidos.
<b>samba_export_all_rw</b>	Permite escritura de todos los directorios compartidos.
<b>samba_run_unconfined</b>	Corre sin restricciones.
<b>samba_share_nfs</b>	Permitirá compartir recursos NFS
<b>use_samba_home_dirs</b>	Permite utilizar un servidor remoto para directorios de usuarios.
<b>virt_use_samba</b>	Permite acceso remoto a la maquina para archivos CIFS

### Como cambiar el booleando de ejecución de SAMBA:

```
semanage boolean -l  
getsebool -a  
getsebool samba_export_all_rw  
setsebool samba_export_all_rw on
```

De manera que esta también sería una opción viable.

**Deberá conectarse desde Windows y Android:**

- Para Android deberá configurar una red Puente muy probablemente.

**FIN.**