RED HAT 8/9 CERTIFIED SYSTEM ADMINISTRATOR (RHCSA)

Jorge G. Floriano

Red Hat 8/9 Certified System Administrator (RHCSA)

Conocer y usar las herramientas principales

- Acceder a un intérprete de comandos de shell y escribir comandos con la sintaxis correcta.

Esto se refiere a saber acceder a un terminal desde el escritorio, o mediante una **tty** mediante la combinación de teclas **Ctrl+Alt+F[1-7]**.

En cuanto a la parte de escribir una sintaxis correcta es el utilizar los comandos con los parámetros correspondientes de cada comando y saber utilizar las ayudas y el man de cada comando.

- Utilizar la redirección de entrada-salida (>, >>, |, 2>, etc.)

La entrada y salida estándar es la capacidad del intérprete de comandos o **Shell** para controlar y dirigir la entrada de datos de los programas, la salida de información útil y la salida de información de errores. Cuando un programa se ejecuta, automáticamente se le proporcionan 3 descriptores de archivo:

- **STDIN:** Entrada estándar o descriptor de archivo 0. El descriptor de archivo STDIN está asociado a la entrada de texto. Por defecto está asociado al teclado.

- **STDOUT:** Salida estándar o descriptor de archivo 1. El descriptor de archivo ST DOUT se corresponde con la salida de texto normal de los programas. Por defecto está asociado al terminal de pantalla.

- **STDERR:** Error estándar. El descriptor de archivo STDERR también es una salida de texto, pero se usa exclusivamente para mostrar los errores generados por los programas. Por defecto también está asociado al terminal de pantalla.
-

Operador	descripción
cmd < file	El contenido de file se utilizará como entrada de la orden cmd
cmd > file cmd 1> file	La salida estándar de la orden cmd se guardará en file. Si el archivo existe se sobrescribirá
cmd >> file cmd 1>> file	La salida estándar de la orden cmd se guardará en file. La información se añadirá al final del archivo.
cmd 2> file	Los errores de la orden cmd se guardarán en file. Si el archivo file existe se sobrescribirá.
cmd 2>> file	Los errores de la orden cmd se guardarán en file. La información se añadirá al final del archivo.
cmd > file 2>&1	La salida estándar y la de errores de la orden cmd se guardarán en file. Si el archivo file existe se sobrescribirá.
cmd > file 2>&1	La salida estándar y la de errores de la orden cmd se guardarán en file. La información se añadirá al final del archivo.
cmd1|cmd2	Redirecciona la salida del comando cmd1 hacia la entrada del comando cmd2

- Utilizar expresiones regulares y grep para analizar textos

Las expresiones regulares (ER) son una forma de describir cadenas de caracteres. Se usan en operaciones de apareamiento o comparación. Las expresiones regulares permiten realizar búsquedas o sustituciones de gran complejidad.

Una expresión regular es un patrón que describe un conjunto de cadenas de caracteres. Por ejemplo, el patrón `aba*.txt` describe el conjunto de cadenas de caracteres que comienzan con `aba`, contienen cualquier otro grupo de caracteres, luego un punto, y finalmente la cadena `txt`. El símbolo ***** se interpreta como ***"0, 1 o más caracteres ".***

Las expresiones regulares se construyen como las expresiones aritméticas, usando operadores para combinar expresiones más pequeñas. Analizaremos esos operadores y las reglas de construcción de expresiones regulares, atendiendo siempre al conjunto de cadenas que representa cada patrón.

Metacarácteres

La construcción de expresiones regulares depende de la asignación de significado especial a algunos caracteres. En el patrón `aba*.txt` el carácter * no vale por sí mismo, como el carácter asterisco, sino que indica un "conjunto de caracteres cualesquiera". Asimismo, el carácter "**?**" no se interpreta como el signo de interrogación, sino que representa "un carácter cualquiera y uno solo". Estos caracteres a los que se asigna significado especial se denominan ***"meta-caracteres"***.

El conjunto de meta-caracteres para expresiones regulares es el siguiente:

```
\ ^ $ . [ ] { } | ( ) * + ?
```

Estos caracteres, en una expresión regular, son interpretados en su significado especial y no como los caracteres que normalmente representan. Una búsqueda que implique alguno de estos caracteres obligará a *"**escaparlo**"* de la interpretación mediante. En una expresión regular, el caracter ***?*** representa ***"un caracter cualquiera"***; si escribimos ***\?, "estamos representando el caracter ? tal cual, sin significado adicional"***.

Expresiones Regulares Básicas.

Una expresión regular determina un conjunto de cadenas de caracteres. Un miembro de este conjunto de cadenas se dice que aparea, equipara o satisface la expresión regular.

Expresiones regulares de un sólo carácter.

Las expresiones regulares se componen de expresiones regulares elementales que aparean con un único carácter:

Exp.Reg.	concuerda con
c	ER que concuerda con el carácter ordinario c
.	(punto) ER que concuerda con un carácter cualquiera excepto nueva línea
[abc]	ER de un carácter que concuerda con a, b o c
[^abc]	ER de un carácter que no sea a, b o c
[0-9][a-z][A-Z]	ERs de un carácter que aparean con cualquier carácter en el intervalo indicado. El signo - indica un intervalo de caracteres consecutivos.
\e	ER que concuerda con alguno de estos caracteres (en lugar de la e): . * [\ cuando no están dentro de [ ] ^ al principio de la ER, o al principio dentro de [ ] $ al final de una ER / usado para delimitar una ER

Los paréntesis rectos `[]` delimitan listas de caracteres individuales. Muchos meta-caracteres pierden su significado si están dentro de listas: los caracteres especiales `. * [ \` valen por sí dentro de `[]`. Para incluir un carácter] en una lista, colocarlo al principio; para incluir un `^` colocarlo en cualquier lugar menos al principio; para incluir un `-` colocarlo al final. Dentro de los conjuntos de caracteres individuales, se reconocen las siguientes categorías:

[:alnum:]	alfanuméricos
[:alpha:]	alfabéticos
[:cntrl:]	de control
[:digit:]	dígitos
[:graph:]	gráficos
[:lower:]	minúsculas
[:print:]	imprimibles
[:punct:]	de puntuación
[:space:]	espacios
[:upper:]	mayúsculas
[:xdigit:]	dígitos hexadecimales

Por ejemplo, `[[:alnum:]]` significa `[0-9A-Za-z]`, pero esta última expresión depende de la secuencia de codificación ASCII, en cambio la primera es portable, no pierde su significado bajo distintas codificaciones. En los nombres de categorías, los paréntesis rectos forman parte del nombre de la categoría, no pueden ser omitidos.

Construcción de Expresiones Regulares.

Una Expresión Regular se construye con uno o más operadores que indican, cada uno, el carácter a buscar. Los operadores más comunes y aceptados son los siguientes:

Operador	Significado
c	un carácter no especial concuerda consigo mismo
\c	elimina significado especial de un carácter c; el \ escapa el significado especial
^	indica ubicado al comienzo de la línea (cadena nula al principio de línea)
$	indica ubicado al final de la línea (cadena nula al final de línea)
.	(punto) un carácter individual cualquiera
[...]	uno cualquiera de los caracteres ...; acepta intervalos del tipo a-z, 0-9, A-Z (lista)
[^...]	un carácter distinto de ... ; acepta intervalos del tipo a-z, 0-9, A-Z
r*	0, 1 o más ocurrencias de la ER r (repetición)
r1r2	la ER r1 seguida de la ER r2 (concatenación)

Ejemplos de Expresiones Regulares Básicas.

Las expresiones regulares se aprenden mejor con los ejemplos y el uso.

Exp.Reg.	concuerda con
a.b	axb aab abb aSb a#b ...
a..b	axxb aaab abbb a4$b ...
[abc]	a b c (cadenas de un caracter)
[aA]	a A (cadenas de un caracter)
[aA][bB]	ab Ab aB AB (cadenas de dos caracteres)
[0123456789]	0 1 2 3 4 5 6 7 8 9
[0-9]	0 1 2 3 4 5 6 7 8 9
[A-Za-z]	A B C ... Z a b c ... z
[0-9][0-9][0-9]	000 001 .. 009 010 .. 019 100 .. 999
[0-9]*	cadena_vacía 0 1 9 00 99 123 456 999 9999 ...
[0-9][0-9]*	0 1 9 00 99 123 456 999 9999 99999 99999999 ...
^.*$	cualquier línea completa

Expresiones Regulares Extendidas.

Algunos comandos, como `egrep` o `grep -E`, aceptan Expresiones Regulares Extendidas, que comprenden las Expresiones Regulares Básicas más algunos operadores que permiten construcciones más complejas. Los operadores incorporados son los siguientes:

Operador	Significado
r+	1 o más ocurrencias de la ER r
r?	0 o una ocurrencia de la ER r, y no más
r{n}	n ocurrencias de la ER r
r{n,}	n o más ocurrencias de la ER r
r{,m}	0 o a lo sumo m ocurrencias de la ER r
r{n,m}	n o más ocurrencias de la ER r, pero a lo sumo m
r1|r2	la ER r1 o la ER r2 (alternativa)
(r)	ER anidada

La repetición tiene precedencia sobre la concatenación; la concatenación tiene precedencia sobre la alternativa. Una expresión puede encerrarse entre paréntesis para ser evaluada primero.

Ejemplos de Expresiones Regulares Extendidas:

Exp.Reg.Ext.	concuerda con
[0-9]+	0 1 9 00 99 123 456 999 9999 99999 99999999 ..
[0-9]?	cadena vacía 0 1 2 .. 9
^a|b	a b
(ab)*	cadena vacía ab abab ababab ...
^[0-9]?b	b 0b 1b 2b .. 9b
([0-9]+ab)*	cadena vacía 1234ab 9ab9ab9ab 9876543210ab ...

- Acceder a los sistemas remotos con SSH

SSH es un protocolo que nos permite la administración remota de nuestros servidores Linux, como un reemplazo a telnet, ssh utiliza un cifrado para garantizar la comunicación entre el cliente y servidor, en este documento veremos cómo realizar esta tarea y las principales funciones del servidor ssh

Para poder conectarse a un sistema mediante el protocolo ssh en necesario saber el usuario y el host al que nos vamos a conectar, la sintaxis del comando ssh para una conexión simple es la siguiente:

```
ssh [opciones] usuario@host [comando]
```

Ejemplos de ssh

```
[franklin@cliente-ssh ~]$ ssh server1
[franklin@cliente-ssh ~]$ ssh -p 50022 server1
[franklin@cliente-ssh ~]$ ssh 10.9.33.24 -l hans
[franklin@cliente-ssh ~]$ ssh 10.9.33.24 -p 22 -l frankli -E login.log
```

Regularmente el usuario root no debería estar habilitado para una conexión ssh por temas de seguridad, así que para temas de ejemplo usaremos al usuario franklin en el host server1 y haremos la conexión desde el host cliente-ssh de la siguiente forma:

```
[franklin@cliente-ssh ~]$ ssh franklin@server1Explicando la salida anterior:
```

Al ejecutar el comando con ssh si es la primera vez que nos conectamos a este host nos pedirá confirmación **(yes)** para poder agregar el key fingerprint a nuestro archivo de host conocidos ubicado en ***~/.ssh/known_hosts,*** esto es parte del cifrado que se llevará al momento de la conexión.

- Iniciar sesión y cambiar de usuario en destinos multiusuario

En este capítulo vamos a ver cómo cambiar dentro del shell entre diferentes usuarios y esto se logra mediante el comando ***Switch User (su)***, para esto nos conectamos a un servidor y una vez logado haremos diferentes cambios de usuarios. Para ellos es necesario tener el password del usuario a cambiar. *(root no necesita el password al momento de Loguearnos con otro usuario no nos pedirá password)*.

```
[hans@server1 ~]$ su – maria
Password:
[maria@server1~]$ id
uid=1003(maria)          gid=1005(maria)          grupos=1005(maria),1001(sistemas)
contexto=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

Es importante mencionar el carácter ***– (guión)*** después del comando **su**, esto nos ayuda a cambiar las variables de entorno de acuerdo al usuario al que queremos cambiar

- Archivar, comprimir, desempaquetar y descomprimir archivos utilizando tar, star, gzip y bzip2

Comando tar

Permite el empaquetado de varios archivos o directorios dentro de un archivo *.tar, todo esto sin compresión solo sirve para llevar a cabo un empaquetado de varios archivos y directorios, su sintaxis.

```
tar [Operación] [Fichero]
```

Opciones:

```
-c crear archivo o empaquetar archivo
-f permite asignar el nombre al archivo TAR
-j permite comprimir o descomprimir un archivo utilizando el formato bzip2
-r permite agregar archivos a un TAR existente
-t muestra los archivos dentro de un TAR
-u actualiza un archivo TAR existente
-v modo verboso que permite la visualización de la operación efectuada
-z permite comprimir o descomprimir un archivo utilizando el formato gzip
-x desempaquetar y descomprimir
```

Ver el contenido de un tar:

```
tar -tvf media.tar
```

Desempaquetar el archivo.tar:

```
tar -xf media.tar
```

Agregar más archivos a un tar ya existente

```
tar rf media.tar crontab
tar -tf media.tar
```

gzip y gunzip

Comando con algoritmos de compresión LZW (Lempel-Ziv-Welch) que permiten compresión sobre los archivos. Opciones:

```
-d para descomprimir, aplica solo a gzip
-f forzar la compresión o descompresión si el archivo aún posee varios vínculos o
si el archivo correspondiente ya existe
-r comprime o descomprime recursivamente
-v detalle del proceso modo verboso
```

Ejemplos:

Comprimiendo archivo imagen.img

```
gzip imagen.img
```

Nótese que existe una gran diferencia de pesos después de la compresión un aprox. de 61% de tamaño menor

Descomprimiendo archivo imagen.img.gz

```
gunzip imagen.img.gz
```

bzip2 y bunzip2

Al igual que gzip y gunzip son comandos de compresión y descompresión solo que usan el algoritmo de ordenamiento por bloques llamado Burrows–Wheeler, bzip2 tiende a tardar más en el proceso de compresión que gzip, pero bzip2 tiende a comprimir un poco más.

Opciones:

```
-d para descomprimir, aplica solo a gzip
-f forzar la compresión o descompresión si el archivo aún posee varios vínculos o
si el archivo correspondiente ya existe
-r comprime o descomprime recursivamente
-v detalle del proceso modo verboso
-z para comprimir; esta aplica solo para bzip2
```

Descomprimiendo archivo imagen.img.bz

```
bzip2 imagen.img.bz2
```

Empaquetado y Compresión

En las líneas anteriores se pudo apreciar que gzip y bzip2 hicieron compresión de archivos únicamente para llevar a cabo la compresión de directorios, es necesario realizar un empaquetado con tar, visto al principio de este documento.

Ejemplos:

Empaquetado y compresión con formato bzip2 y gzip

```
tar zcvf media.tar.gz media/ gzip
tar jcvf media.tar.bz media/ bzip2
```

Desempaquetar un archivo TAR

```
tar xvf media.tar
```

Desempaquetar y descomprimir TAR.GZ

```
tar xzvf media.tar.gz
```

Descomprimir y desempaquetar el archivo /root/ejemplo.tar.gz en /tmp

```
tar xzvf /root/ejemplo.tar.gz -C /tmp/
```

Empaquetando y comprimiendo respetando los contextos ***SELinux*** de los archivos

```
tar --selinux -czvf selinux.tar.gz /etc/*.conf
```

Desempaquetando y descomprimiendo respetando los contextos SELinux de los archivos

```
tar --selinux -xzvf selinux.tar.gz
```

Comando star

En necesario instalar el paquete mediante

```
yum install -y star
```

Para crear un star:

```
star -xattr -H=exustar -c -f=etc_dir.star /etc
```

Y para descomprimirlo:

```
star -x -f=etc_dir.star
```

- Crear y editar archivos de texto

Touch

Este comando nos permite la creación de archivos vacíos, además de cambiar la fecha de modificación o creación del archivo, incluso podemos crear un archivo con una fecha diferente a la del momento. Sintaxis:

```
touch [opción] nombre_archivo
```

Opciones:

```
-a      cambia la fecha de último acceso
-m      cambia la fecha de la última modificación
-t      fecha y hora de creación en formato YYMMDDhhmm
```

File

Linux no necesita de extensiones para poder trabajar y determinar el tipo de diferentes archivos, sin embargo, si se necesita saber qué tipo es un archivo utilizamos el comando file

```
bonzo@starwing:~$ file y.variables
y.variables: ASCII text
```

Vi, vim

Vi, el editor más utilizado en ambientes Linux y Unix, ya que viene por defecto en cualquier distro de estos sistemas operativos, Para entrar a vi ejecutamos el comando **vi** o **vim** *(normalmente vim no viene por defecto, bastará con instalarlo con **yum -y install vim**)*.

Navegación

h	Izquierda ←
j	Abajo ↓
k	Arriba ↑
l	Derecha →
G	Final de archivo
gg	Principio de archivo
w	mover palabra por palabra hacia adelante
b	mover palabra por palabra hacia atrás

Inserción de Texto

a	Insertar después del cursos
A	Insertar al final de la línea
i	Insertar antes del cursor
I	Insertar al principio de la Línea
o	Abrir línea debajo de la actual
O	Abrir línea arriba de la actual
ESC	Salir de modo inserción

Borrar, Copiar y Pegar

dd	Cortar o Borrar
8dd	Cortar o Borrar 8 líneas
p	Pegar de pues del cursos
yy	copia líneas
7yy	Copia 7 líneas
x	Borra un carácter
R	Remplazar
u	Undo(deshacer)
crtl+R	Repeat (Rehacer)

- Crear, borrar, copiar y mover archivos y directorios

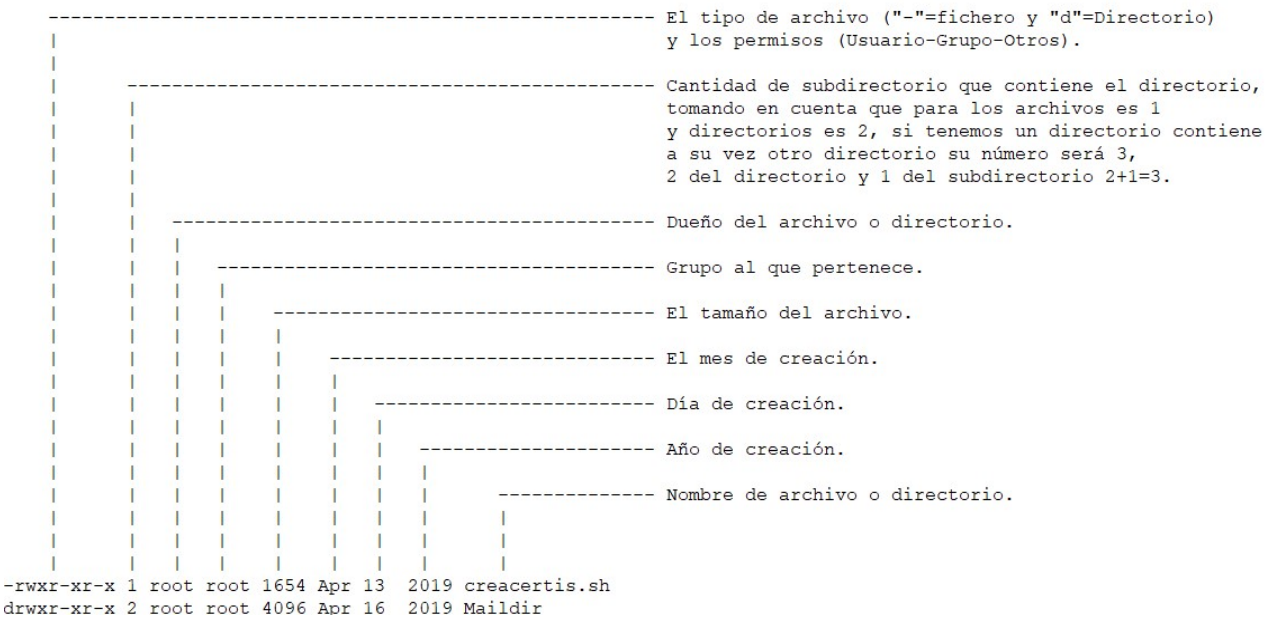
En este capítulo trataremos cómo trabajar con archivos y directorios, en complemento con el capítulo anterior ***"Crear y editar archivos de texto"***, los siguientes comandos son los más utilizados en el sistema operativo Linux.

Comando ls

Permite listar el contenido de los directorios de archivos, con sus opciones es capaz de mostrar las propiedades de los subdirectorios y archivos como tamaños fechas de creación permisos y también visualizar archivos ocultos su sintaxis.

```
ls  [opciones] directorio
```

```
Opciones:
-a      lista archivos ocultos.
-i      muestra el número de inodes de los archivos.
-l      muestra el listado con detalle.
-r      ordenamiento invertido.
-R      muestra jerarquías de los subdirectorios.
-t      ordena por la fecha de modificación.
-h      muestra en el tamaño de archivos más entendible para los humanos, es
        decir con unidades de medidas de almacenamiento.
```



Comando mkdir

Permite la creación de Directorios su sintaxis es:

```
mkdir [opciones] nombre_de_directorio
```

```
Opciones:

-m      Crear directorios con permisos establecidos
-p      Permite crear rutas absolutas de directorios
-v      Modo verbose
-Z      Establecer el contexto de seguridad SELinux de cada directorio creado al
        tipo predeterminado.
–context[=CTX]   Similar a -Z
–help          muestra esta ayuda y finaliza
–versión       informa de la versión y finaliza
```

Es posible crear varios directorios en una sola línea de comando:

```
root@starwing:~# mkdir {carpeta1,carpeta2}

root@starwing:~# ls -lrtd carpeta*
drwxr-xr-x 2 root root 4096 Oct 20 12:01 carpeta2
drwxr-xr-x 2 root root 4096 Oct 20 12:01 carpeta1
root@starwing:~#
```

Crea el directorio directorio_dos dentro de este crea directorio_2_1 y dentro de este crea directorio2_2, básicamente es crear jerarquías completas de directorios

```
root@starwing:~# mkdir -pv directorio_dos/directorio_2_1/directorio_2_2
mkdir: created directory 'directorio_dos'
mkdir: created directory 'directorio_dos/directorio_2_1'
mkdir: created directory 'directorio_dos/directorio_2_1/directorio_2_2'
```

Crear una estructura más compleja con mkdir

```
root@starwing:~# mkdir -p abuelo/{papa/hijo/{nieto,nieta},mama,tio}
root@starwing:~# tree abuelo
abuelo
├── mama
├── papa
│   └── hijo
│       ├── nieta
│       └── nieto
└── tio
6 directories, 0 files
```

Comando rmdir

El comando que permite borrar directorios solo si están vacíos su sintaxis es:

```
rmdir [opciones] nombre_del_directorio
```

Comando rm

Es importante que sepa que este comando debe ser manejado con mucho cuidado ya que una vez que sea borrado un archivo o directorio no hay forma de recuperar el mismo.

```
rm [opciones] nombre_archivo_diretorio
```

```
Opciones:
-f      ignora archivos no-existentes y no efectúa preguntas
-i      pregunta antes de remover cada archivo
-r –R   Remueve directorios y su contenido recursivamente
-v      Muestra el detalle de lo que se ejecuta
```

Comando mv

Comando que permite mover a diferentes ubicaciones y renombrar los archivos o directorios especificados, es muy importante tener claro las rutas absulutas y relativas, el mal uso del comando nos podría traer posibles problemas su sintaxis.

```
mv [opciones] origen destino
```

```
Opciones:
-f      no pregunta antes de sobrescribir
-i      Pregunta antes de sobrescribir
-v      muestra lo que se está efectuando
```

Comando cp

Herramienta para copiar archivos o directorios tenemos:

```
cp [opciones] ruta_origen ruta_destino.
```

```
Opciones:
-d      Preserva los vínculos.
-p      Preserva los modos como los propietarios y las estampas de tiempo
-r,-R   Copia el directorio de manera recursiva
-u      copia solamente cuando el fichero ORIGEN es más moderno que el fichero de
        destino, o cuando falta el fichero de destino.
```

- Crear enlaces físicos y simbólicos

Sencillamente definimos un ***enlace simbólico*** o ***"Soft Link"*** como una ruta de acceso a un archivo, por consiguiente, si el archivo original es borrado los enlaces simbólicos que apuntaban a este se pierden y ya no tienen razón de ser, a diferencia de los ***enlaces duros*** o ***"Hard Link",*** estos enlaces solo se permiten enlazar sobre archivos, comparten el mismo inodo, se considera una copia del archivo origen.

```
Enlaces duros
root@starwing:~# ln origen destino
```

```
Enlaces simbólicos
root@starwing:~# ln -s origen destino
```

Enlaces simbólicos:

1. Es posible crear enlaces simbólicos de archivos y directorios.
2. Tienen diferente inodo que el fichero origen enlazado.
3. Las modificaciones realizadas en el origen o en el enlace se aprecian en ambos.
4. Si el archivo original es borrado el enlace simbólico se rompe, perdiendo toda información, quedando este inservible, aún así el fichero enlace sigue existiendo.

Enlaces Duros:

1. No es posible crear enlaces duros de un directorio.
2. Ambos ficheros comparten el mismo inodo.
3. Modificaciones realizadas en el origen o en el enlace se aprecian en ambos ficheros.
4. Si el archivo original es borrado en enlace duro se mantiene, conservando los datos, ya que es una copia del original.

Enumerar, configurar y cambiar permisos ugo/rwx estándares

Cada uno de los elementos del sistema de ficheros de Linux posee permisos de acceso de acuerdo a tres tipos de usuarios:

- **U**: Su dueño, normalmemnte el creador, representado por la letra u ***(USER)***.
- **G**: Su grupo representado por la letra g ***(GROUP).***
- **O**: El resto de los usuarios que no son el dueño ni pertenecen al grupo. Se representa con o ***(OTHER)***.

Nota*: Para representar a todos los tipos de usuarios se utiliza la letra a (all).*

Para cada uno de estos tres grupos de usuarios existen tres tipos de permisos fundamentales:

- **r:** El usuario que tenga este permiso podrá si es un directorio, listar los recursos almacenados en él, y si es cualquier otro tipo de fichero podrá leer su contenido.

- **w:** Todo usuario que posea este permiso para un fichero podrá modificarlo. Si se posee para un directorio se podrán crear y borrar ficheros en su interior.

- **x:** Este permiso para el caso de los ficheros permitirá ejecutarlos desde la línea de comandos y para los directorios, el usuario que lo posea tendrá acceso para realizar el resto de las funciones permitidas mediante los otros permisos (lectura y/o escritura).

Para poder realizar operaciones sobre cualquier **directorio** (leer o escribir) será necesario siempre, tener otorgado además el permiso de ejecución. Para acceder a un recurso de cualquier forma (ejecución, lectura o escritura) se deben tener permisos de ejecución para todos los directorios que contienen al recurso directa e indirectamente.

Los tres tipos de permisos mencionados poseen una representación numérica basada en el sistema octal que parte de representar como ``1'' los bits de los permisos otorgados y ``0'' para los negados. Luego se transforma la representación binaria así obtenida en octal. Los permisos siempre van formando tríos, de la forma rwx.

r = 100 (4 en octal) (r--)
w = 010 (2 en octal) (-w-)
x = 001 (1 en octal) (--x)

Permiso	R	W	X
Valor	4	2	1

De esta forma se obtienen para cada tipo de permiso los siguientes valores:

La combinación de los tres tipos de permisos para un tipo de usuario oscila desde cero (ningún permiso) hasta siete (todos los permisos).

S y s: es un permiso que de no administrarse correctamente puede provocar problemas de seguridad. Para su representación a través de caracteres se utiliza el lugar del permiso de ejecución y de ahí la diferencia entre s y S: si es s (minúscula) significa que incluye además el permiso de ejecución (x y s) a diferencia de S que incluye solo el permiso (s) y no el x. Este permiso se puede asociar al dueño o al grupo del recurso. Si se asocia a un fichero significa que cuando este se ejecute por un usuario que tenga permisos para ello adquirirá los permisos de su dueño o grupo según a que esté asociado el permiso. Un ejemplo de fichero con este permiso es el comando passwd, el cual adquiere los permisos de root al ser ejecutado por los usuarios (sin argumentos) para poder modificar el fichero /etc/shadow que es donde se guardan las contraseñas de los usuarios. Para el caso de un directorio este permiso sólo tiene validez para el grupo del mismo permitiendo a los ficheros y a los subdirectorios que se creen en él heredar el grupo, los subdirectorios heredarán también el permiso s. Un ejemplo de directorio con este permiso es aquel donde se guardan los documentos de un sitio FTP anónimo. Este permiso se conoce como setuid bit o setgid bit, para el usuario y el grupo respectivamente.

T y t cuando está asociado a un directorio junto al permiso de escritura para un grupo de usuarios, indica que estos usuarios pueden escribir nuevos ficheros en el directorio, pero estos sólo podrán ser borrados por sus dueños o por root. Para un fichero el permiso expresa que el texto de este se almacena en memoria swap para ser accedido con mayor rapidez. Este permiso sólo se asocia al resto de los usuarios y para su representación se emplea el bit correspondiente al permiso de ejecución: si es t (minúscula) significa que incluye además el permiso de ejecución y T (mayúscula) no lo incluye. Ejemplo de un directorio con este permiso es /tmp donde todos los usuarios pueden escribir, pero sólo los dueños pueden borrar sus ficheros, además de root. Este permiso se conoce también como sticky bit.

Para representar los permisos t y s en el sistema se utilizan tres bits adicionales: el primero para s en el dueño, el segundo para s en el grupo y el tercero para t. Estos se colocan al inicio de la cadena numérica de nueve bits vista anteriormente. En la cadena de caracteres se mezclan con el permiso de ejecución y de ahí la necesidad de emplear las mayúsculas y minúsculas.

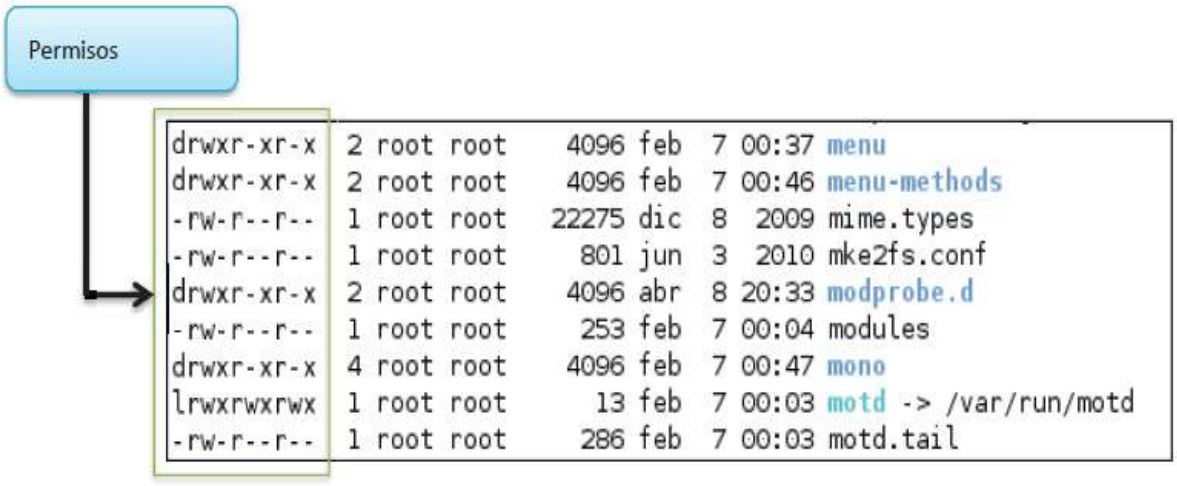
Ejemplos:

```
rws rwS r-- = 110 111 110 100 (6764 en octal)
rwx rws -wT = 011 111 111 010 (3772 en octal)
```

Estos permisos s y t son una causa frecuente de problemas, y son inseguros de por si, por lo que es muy recomendable no usarlos a menos que sea estrictamente necesario. De hecho, la mayoría de las distribuciones actuales ya no utilizan estos permisos.

Después de toda esta introducción a los permisos en Linux veamos cómo estos se muestran, modifican y se les asigna un valor por defecto.

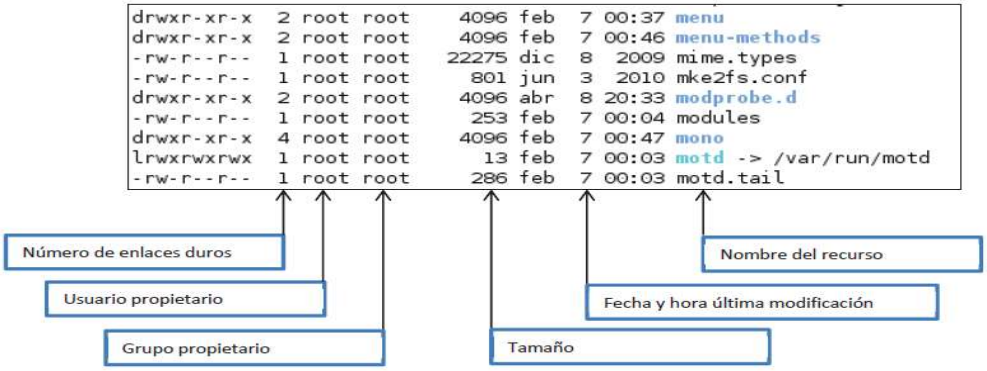
Posiblemente el comando más empleado en Linux es aquel que muestra el contenido de un directorio, llamado ls. Este comando con la opción -l permite observar los permisos que tienen asociados los recursos listados, además de otras características. Los permisos se muestran a través de una cadena de 10 caracteres:



1. El primer carácter indica el tipo de recurso, y puede ser:
    a. **d** : directorio.
    b. **l** : enlace.
    c. **b** : dispositivo de bloque.
    d. **c** : dispositivo de caracteres.
    e. **s** : socket.
    f. **p** : tubería (pipe).
    g. **-** : fichero regular.

2. Los caracteres 2,3 y 4 indican los permisos para el usuario dueño del recurso.
3. Los caracteres 5,6 y 7 indican los permisos para el grupo dueño del recurso.
4. Los caracteres 8,9 y 10 indican los permisos para el resto de usuarios, es decir, los usuarios que no son.

El resto de la información que nos proporciona un "ls -lia" es la siguiente:



Comando chgrp

Change Group o bien Cambiar Grupo es un comando que nos permite cambiar la propiedad de un archivo o directorio entorno al grupo al que pertenecen, su sintaxis.

```
chgrp name_grupo archivo_directorio.
```

```
Opciones
-c, –changes            como `verbose' pero informa sólo de los cambios
-f, –silent, –quiet     suprime la mayoría de los mensajes de error
-v, –verbose            muestra un mensaje por cada fichero procesado
-R, –recursive          opera sobre ficheros y directorios recursivamente
```

Ejemplo:

```
root@starwing:/home# ls -lrtd pedro
drwxr-xr-x 3 pedro directivo 4096 Sep  5 21:13 pedro

root@starwing:/home# chgrp jefes pedro
root@starwing:/home# ls -lrtd papa*
drwxr-xr-x 3 pedro jefes 4096 Sep  5 21:13 pedro
```

Comando chown

Change Owner o bien Cambiar dueño Permite cambiar el usuario propietario del archivo o directorio, su sintaxis:

```
chown usuario archivo_directorio.
root@starwing:/home# ls -lrtd papa0001
drwxr-xr-x 3 papa0001 papa0002 4096 Sep  5 21:13 papa0001

root@starwing:/home# chown bonzo:root papa0001

root@starwing:/home# ls -lrtd papa0001
drwxr-xr-x 3 bonzo root 4096 Sep  5 21:13 papa0001
```

Comando chmod:

Las formas de expresar los nuevos permisos son diversas, se pueden usar números o caracteres para indicar los permisos.

```
chmod [opciones] <permisos> <ficheros>
```

Podremos comprender mejor cómo funciona la orden mirando directamente algunos ejemplos:

chmod u+x clase.txt	Añade el permiso de ejecución (+x) al usuario dueño (u) del fichero clase.txt.
chmod g=rx program.sh	Asigna exactamente los permisos de lectura y ejecución (rx) al grupo (g) sobre el fichero program.sh.
chmod go-w profile	Elimina el permiso de escritura (-w) en el grupo y en otros (go) del fichero o directorio profile.
chmod a+r,o-x *.ts	Añade el permiso de lectura (+r) para todos los usuarios (a) y elimina el de ejecución (-x) para otros (o) en todos los ficheros terminados en .ts.
chmod +t tmp/	Añade el permiso especial t al directorio tmp.
chmod 755 /home/pepe/doc/	Asigna los permisos con representación octal 755 (rwx r-x r-x) al fichero /home/pepe/doc.
chmod -R o+r apps/	Añade el permiso de lectura a otros en el directorio apps y además lo hace de forma recursiva, añadiendo dicho permiso también en todos los ficheros y directorios contenidos en apps.
chmod 4511 /usr/bin/passwd	Asigna los permisos con representación octal 4511 (r-s--x--x)
chmod 644 *	r w – r - - r - - Lectura y escritura para el usuario, lectura para el grupo y lectura para otros.

La siguiente tabla contiene los posibles valores que se pueden asignar con la nomenclatura octal:

Cadena de permisos	Código octal	Significado
rwx-rwx-rwx	777	Permisos de lectura, escritura y ejecución para todos los usuarios
rwxr-xr-x	755	Permisos de lectura y ejecución para todos los usuarios, el propietario también tiene permisos de escritura.
rwxr-x—	750	Permisos de lectura y ejecución para el propietario y el grupo, permisos de escritura para el propietario y sin acceso al fichero para todos los demás.
rwx——	700	Permisos de lectura, escritura y ejecución para el propietario, los demás no tendrán acceso.
rw-rw-rw-	666	Permisos de lectura y escritura para todos los usuarios, nadie tiene permiso de ejecución.
rw-rw-r–	664	Permisos de lectura y escritura para el propietario y el grupo y de solo lectura para el resto.
rw-rw—-	660	Permisos de lectura y escritura para el propietario y el grupo, y no hay permisos para el resto de usuarios.
rw-r–r–	644	Permisos de lectura y escritura para el propietario, permisos de solo lectura para el resto.
rw-r—–	640	Permisos de lectura y escritura para el propietario y de solo lectura para el grupo, no hay permisos para los demás.
rw——-	600	Permisos de lectura y escritura para el propietario y nadie más tiene permisos.
r——–	400	Permiso de lectura para el propietario y nadie más tiene permisos.

Umask

Es un parámetro de configuración por defecto definido en el archivo ***/etc/profile*** y hace referencia a los permisos que tendrán los directorios o ficheros que se creen en el sistema Linux. La mayoría de los sistemas Linux utilizan una umask por defecto de para los usuarios 002 o 022 para root.

Para saber los valores actuales para esta configuración utilizará el comando umask sin parámetros y si se le añade el parámetro -S se obtendrá expresada simbólicamente en lugar de en forma octal.

```
root@starwing:/home# umask
0022
```

```
root@starwing:/home# umask -S
u=rwx,g=rx,o=rx
```

Para calcular los permisos finales conociendo la máscara, se hace la siguiente operación por parte del sistema:

Tipo de ficheros	Operación	Desarrollo	Resultado
Ficheros normales	666 - máscara	(666 – 022 = 644)	644
Directorios y Ficheros ejecutables	777 - máscara	(777 – 022 = 755)	755

Ejemplos:

$ umask	Sin argumentos muestra la máscara actual en formato numérico.
$ umask -S	muestra el complemento de la máscara en formato de caracteres
$ umask 037	asigna la máscara 037
$ umask –S u=rwx,g=rwx,o=rx	Directorios y archivos ejecutables se crearán con los permisos 775 y los archivos comunes con los permisos 664.
$ umask 077	Los nuevos directorios tendrán el permiso 700 y los nuevos ficheros tendrán el permiso 600.

Como vemos, si usamos el formato numérico en umask, tendremos que restar la máscara al total de permisos para saber que permisos se asignarán por parte del sistema. Sin embargo, si usamos el formato simbólico (-S) de umask, asignaremos directamente los permisos que el sistema asignará.

Para dejar este umask fijo, conviene agregarlo al fichero ***/home/user/.bashrc*** de nuestro directorio de inicio.

Valores de muestra de umask y sus efectos:

Umask	Ficheros Creados	Directorios Creados
0	666 (rw-rw-rw-)	777 (rwxrwxrwx)
2	664 (rw-rw-r–)	775 (rwxrwxr-x)
22	644 (rw-r–r–)	755 (rwxr-xr-x)
27	640 (rw-r——)	750 (rwxr-x—)
77	600 (rw——-)	700 (rwx——)
277	400 (r———)	500 (r-x——)

- Localizar, leer y utilizar la documentación del sistema, lo que incluye man, info y archivos en /usr/share/doc

El sistema operativo Linux viene por defecto con un sin fin de documentación, manuales que nos dan apoyo a entender el funcionamiento de comandos, archivos de configuración, tareas de administración, llamada al sistema, entre otros. Estos documentos nos apoyaran a un mejor entendimiento y funcionalidad del sistema, es por eso la importancia de destacar el tema y aprender a usar estas características.

```
man
```

Son los manuales que vienen con el sistema operativo, muy extenso como para convertirse en múltiples libros impresos, cada manual contiene información específica de archivos y para un mejor entendimiento este es dividido en secciones por tema.

Una vez tengamos acceso al sistema es bueno usar mandb para poder tener todos los manuales de sistema disponibles:

```
mandb
```

Comandos de ayuda o búsqueda de manuales

```
whatis
apropos
```

GNU info / pinfo

Como parte del proyecto GNU se crea info, y de esta manera darle una definición de documentación general, sin bien man resulta muy útil como información formal pero no tan útil como documentación general, aquí la información es estructurada en nodos mediante hipervínculos, lo cual permite realizar análisis más minuciosos de comandos y conceptos complejos, resulta en ocasiones que info contiene información más detallada que man, para acceder a la información usamos los comandos info o bien pinfo, la diferencia entre estos radica que pinfo se diseñó para coincidir con las teclas de navegación de modo texto lynx, además del uso de colores entre otras funcionales, me atrevo a decir que es la versión mejorada de info

```
info
pinfo
```

En las nuevas instalaciones de programas, estos crean archivos de documentación para poder apoyarnos en el manejo de sus características esta información es almacenada en el directorio /usr/share/doc/nombre_del_paquete y es una extensión de información de los manuales de man y pinfo.

En las mejores prácticas primero se consulta el man posterior si no encontramos la soluciona a nuestras dudas usamos pinfo y al final usaremos la documentación ubicada en /usr/share/doc/nombre_del_paquete

Existen paquetes que en necesario instalar otro paquete (rpm) para poder tener información, esto lo podemos saber mediante el siguiente comando

```
yum list *-doc*
```

Crear scripts de shell sencillos

- Ejecutar el código condicionalmente (uso de if, test, [], etc.)

Estructura básica del IF:

```
if [[ -z "$string" ]]; then
  echo "String is empty"
elif [[ -n "$string" ]]; then
  echo "String is not empty"
fi
```

Condicionales para strings, números y expresiones regulares	
[[-z STRING]]	Cadena vacía
[[-n STRING]]	Cadena no vacia
[[STRING == STRING]]	Cadena igual
[[STRING != STRING]]	Cadena no igual
[[NUM -eq NUM]]	Igual
[[NUM -ne NUM]]	No es igual
[[NUM -lt NUM]]	Menos que
[[NUM -le NUM]]	Menos que o igual
[[NUM -gt NUM]]	Mayor que
[[NUM -ge NUM]]	Mayor que o igual
[[STRING =~ STRING]]	Regexp
((NUM < NUM))	Condiciones numericas
[[-o noclobber]]	Sí OPTIONNAME está habilitado
[[! EXPR]]	Not
[[X && Y]]	And
[[X || Y]]	Or

Condicionales para comprobar ficheros	
[[-e FILE]]	Existe
[[-r FILE]]	Legible
[[-h FILE]]	Symlink
[[-d FILE]]	Directorio
[[-w FILE]]	Writable
[[-s FILE]]	Ocupa > 0 bytes
[[-f FILE]]	Es un fichero
[[-x FILE]]	Ejecutable
[[FILE1 -nt FILE2]]	1 es mas reciente que 2
[[FILE1 -ot FILE2]]	2 es menos reciente que 1
[[FILE1 -ef FILE2]]	Archivos iguales

Estructura case:

```
case $edad in
  30)    echo "¡Correcto!";;
  *)     echo "¡Incorrecto!";;
esac
```

- Utilizar estructuras de bucle (for, etc.) para procesar la entrada de línea de comandos y archivos

For:

```
for i in /etc/rc.*; do
  echo $i
done
```

```
for ((i = 0 ; i < 100 ; i++)); do
  echo $i
done
```

```
for i in {1..5}; do
    echo "Welcome $i"
done
```

```
for i in {5..50..5}; do
    echo "Welcome $i"
done
```

```
for ((variable_1 = 1; variable_1 < 5; variable_1++)); do
    echo "Iteración superior: $variable_1:"
    for ((variable_2 = 1; variable_2 <= 3; variable_2++)); do
        echo "ciclo interno: $variable_2"
    done
done
```

Reading lines:

```
cat file.txt | while read line; do
  echo $line
done
```

```
while read line; do
  echo $line
done < file.txt
```

While:

```
while true; do echo “buenos dias”; done
```

- Procesar entradas de scripts ($ 1, $ 2, etc.)

En Bash, hay algunas variables especiales y que están definidas por defecto, y que se refieren al script, al que ha ejecutado el script, o a la máquina en la que se ha ejecutado el script. Así, algunas de ellas son las siguientes,

- **$0** representa el nombre del script.
- **$1 – $9** los primeros nueve argumentos que se pasan a un script en Bash.
- **$#** el número de argumentos que se pasan a un script.
- **$@** todos los argumentos que se han pasado al script.
- **$?** la salida del último proceso que se ha ejecutado.
- **$$** el ID del proceso del script.
- **$USER** el nombre del usuario que ha ejecutado el script.
- **$HOSTNAME** se refiere al hostname de la máquina en la que se está ejecutando el script.
- **$SECONDS** se refiere al tiempo transcurrido desde que se inició el script, contabilizado en segundos.
- **$RANDOM** devuelve un número aleatorio cada vez que se lee esta variable.
- **$LINENO** indica el número de líneas que tiene nuestro script.

Operar sistemas en funcionamiento

- Iniciar, reiniciar y apagar un sistema con normalidad

```
systemctl [ start | poweroff | reboot ]
```

- Iniciar sistemas manualmente en destinos diferentes

Seleccionar un ***"Target"***, los target´s son los diferentes perfiles que tiene el sistema, por ello los ***"destinos"*** serán dónde vamos a trabajar ***(Graphical.target , multi-user.target)***

Como ver el target que estas usando:

```
systemctl get-default
systemctl set-default
```

Cambiar de target, al reiniciar él servidor se mantendrá el ***target*** seleccionado.

```
systemctl set-default multi-user.target
```

En el examen las preguntas suelen indicar con que Target debe quedar el sistema, hay que tenerlo en cuenta.

Caso práctico:

Poner por defecto el target ***multi-user.target***, y una vez cambiado, pasa al modo ***graphical.target*** , pero manteniendo el ***multi-user.target*** por defecto.

Como usuario root:

```
systemctl set-default multi-user.target
systemctl reboot
```

- Interrumpir el proceso de arranque para obtener acceso a un sistema

Entrar en el GRUB:

Para ello arrancar el sistema y cuando salga para seleccionar opciones de arranque:

pulsar E para editar >> ir a la linea >> Linux16 pulsar “Ctrl+E " para ir al final de la línea

editar detrás de .../swap systemd.unit=multiuser.target "Es un cambio temporal al reiniciar no se guardará"

Presionar ctrl+x para iniciar con estos parámetros.

Ahora se accede en modo comandos y podemos comprobar "systemctl get-default" y verificar el Target actual.

Restablecer contraseña root olvidada o desconocida:

RHEL8

1º Entrar en el GRUB:

Para ello arrancar el sistema y cuando salga para seleccionar opciones de arranque:

pulsar E para editar >> ir a la línea >> Linux16 pulsar ***“Ctrl+E”*** para ir al final de la línea

editar eliminando ***"rhgb quiet"*** y añadir ***" rd.break "*** . Presionar ***“Ctrl+x”*** para iniciar sistema.

2º Una vez arrancado:

mount ver la última linea ***"/vg00-root"*** tiene permisos " ro " cambiarlo

```
mount -o rw,remount /sysroot/
```

mount ahora estará como rw

Para entrar en un entorno chroot dentro de un entorno seguro

```
chroot /sysroot
passwd root
```

Tiene que dar el mensaje ***“update succesfully”.***

3º seelinux aplicar políticas de seguridad

```
touch /.autorelabel
exit
```

Puede tardar bastante dependiendo del sistema de archivos que tengamos.

RHEL9

1º Entrar en el GRUB:

Para ello arrancar el sistema y cuando salga para seleccionar opciones de arranque:

pulsar E para editar >> ir a la línea >> Linux16 pulsar ***"Ctrl+E"*** para ir al final de la línea

```
rw init=/bin/bash
```

2º Cambiar password y aplicar policitas de seelinux

```
passwd
touch /.autorelabel
exec /sbin/init
```

- Identificar los procesos con un uso intensivo de la unidad central de procesamiento (CPU) y de la memoria, y eliminarlos.

Es importante saber cómo realizar tareas específicas con los procesos que se están ejecutando, ya sea en primer o segundo plano, si bien resulta imposible plasmar en un documento todo lo que pudiese ver con el tema de la administración de procesos, trataré de abarcar los temas más relevantes.

Conceptos básicos:

- **Procesos:** Conjunto de código compilado y que se está ejecutando en memoria, el sistema operativo otorga identificadores numéricos a los procesos llamados **ProcessID (PID)** cada proceso tiene un padre llamado **Parent Process ID (PPID)**, recordemos que el proceso con PID=1 es init o bien systemd en las versiones más actuales de Linux.

- **Daemon:** Procesos que se requiere que nunca caduquen como los servicios de red que se instalan en servidores DHCP, FTP, CORREO, etc.

Trabajando con Procesos

Visualizar comandos en ejecución:

PS (process status o estatus de proceso) nos permite visualizar todos los procesos que se estén ejecutando en el sistema, permite ver su PID, usuario propietario, tiempo de ejecución entre otros datos. Lazado como tal el comando lanza la lista de los procesos que al usuario actual pertenecen y los asociados a la terminal.

Opciones:

```
-a Muestra los procesos que le pertenecen a otros usuarios y que están asociados
a una terminal.
-l Formato detallado que incluye la prioridad, el PPID entre otra información.
-u Formato de usuario incluye hora de inicio de los procesos y nombres de
usuarios,
-x Incluye a los procesos que no están asociados a ninguna terminal. La opción –x
se utiliza para ver un proceso demonio y otros que no son incluidos en terminal
-C Muestra instancias del comando.
-U Muestra todos los procesos que pertenecen a un usuario.TOP
```

TOP es la unión de ***ps+uptime+free*** con la diferencia que muestra los procesos en tiempo real con actualización constante, por defecto top muestra los procesos en orden descendente de acuerdo al consumo de procesados. Si no se especifica otra opción, top seactualiza cada 3 segundos.

Opciones:

```
-b ejecuta top en modo batch.
-d especifica tiempo de refresh.
-i ignora los procesos en espera.
-n muestra la cantidad de refresh antes de salir.
-s ejecuta top en modo seguro.
```

JOBS Muestra una lista de los comandos que se encuentran en segundo plano ya ejecutándose o que fueron suspendidos. Es importante que considere la diferencia entre el jobID y el PID ya que el comando jobs muestra el jobID que está relacionado de la sesión actual de la shell mientras que el PID es el mismo en todo el sistema.

*Otros comandos que pueden ayudar **iostat, netstat, vmstat, sar.***

FREE es un comando con el que obtenemos la memoria total y la memoria utilizada por el sistema y la información acerca de la memoria de intercambio ***(swap).***

```
bonzo@starwing:~$ free -g
               total        used        free      shared  buff/cache   available
Mem:               7           2           0           0           4           4
Swap:              1           0           1
```

NICE es un comando que nos permite dar prioridad a la ejecución de ciertos procesos para que se le otorge mayores recursos de CPU o memoria, el comando **nice** permite iniciar procesos con una prioridad seleccionada, el rango de prioridad es de ***-20 que es más alta***, hasta la ***más baja que es la 19***.

```
nice -15 sleep 3000&
```

RENICE Una vez otorgada una la prioridad por nice al iniciar un proceso, podemos modificar la prioridad de procesos ya iniciados, sus sintaxis:

```
renice valor_proridad opciones objetivos
renice 10 -p 1903
1903 (process ID) old priority -15, new priority 10
```

Ejecutar procesos en segundo plano:

Con el carácter ***&*** colocado al final de un comando permite mandar ese proceso a segundo plano y nos devuelve el jobID, el PID y el PROMT para seguir trabajando.

El comando ***nohup***, permite que los procesos se continúen ejecutándose en segundo plano aún si el propietario de la sesión donde el proceso fue lanzado cerrará su sesión en el sistema, ya que por lo general cuando se cierra sesión todos los procesos iniciados por el usuario recibe

la señal singup y son terminados, así que nohup da el privilegio de ser inmune a esta señal al momento de logout del usuario y el proceso seguirá ejecutándose.

Con el comando ***bg*** puede retomar procesos que fueron detenidos y seguir la ejecución en segundo plano, cuando un usuario ejecuta un comando o proceso en un terminal este puede ejecutarse en primer o segundo plano, y decimos que se ejecuta en primer plano cuando el proceso o comando toma el control de la terminal en la que estamos trabajando, como ejemplo tenemos el comando top, en segundo plano tenemos, por ejemplo: crond, syslog y sendmail. Con el comando ***fg*** traerá el proceso en ejecución en segundo plano a primer plano.

Detención de procesos.

Comando ***KILL***, En ocasiones se requiere terminar un proceso que entró en conflicto con otros o por cuestiones de recursos de otros procesos, por cualquiera que sea la situación, el comando **kill** nos permite matar ***(terminar)*** los procesos enviando una señal de al proceso causando su terminación, su sintaxis es:

```
kill señal PID
```

Las señales más utilizadas son:

```
1) SIGHUP terminará el proceso.
9) SIGKILL forzará la terminación de los procesos sin ejecutar rutinas de finalización. La señal por default no especifica al comando kill.
15) SIGTERM termina el proceso, pero ejecuta la rutina de finalización como el cierre de documentos.
```

Para ver las señales posibles utilizar:

```
kill -l
```

Tambien se puede utilizar los comandos ***killall*** o ***pkil*** para fonalizar procesos por su nombre:

```
killall opciones nom_proceso
```

- Ajustar la programación de los procesos[1]

En Red Hat Enterprise Linux, la unidad más pequeña de ejecución de procesos se llama hilo. El programador del sistema determina qué procesador ejecuta un hilo y durante cuánto tiempo lo hace.

- **SCHED_FIFO**, también llamada programación de prioridad estática, es una política en tiempo real que define una prioridad fija para cada hilo. Se recomienda no ejecutar esta política durante un periodo de tiempo prolongado para tareas sensibles al tiempo.

  Cuando SCHED_FIFO está en uso, el programador escanea la lista de todos los hilos de SCHED_FIFO en orden de prioridad y programa el hilo de mayor prioridad que esté listo para ejecutarse. El nivel de prioridad de un subproceso de SCHED_FIFO puede ser cualquier número entero entre 1 y 99, donde 99 se trata como la prioridad más alta.

- **La SCHED_RR** es una variante round-robin de la SCHED_FIFO. Esta política es útil cuando varios hilos deben ejecutarse con el mismo nivel de prioridad.

  Al igual que SCHED_FIFO, SCHED_RR es una política en tiempo real que define una prioridad fija para cada hilo. El programador escanea la lista de todos los hilos SCHED_RR en orden de prioridad y programa el hilo de mayor prioridad que esté listo para ejecutarse. Sin embargo, a diferencia de SCHED_FIFO, los hilos que tienen la misma prioridad se programan en un estilo round-robin dentro de una determinada franja de tiempo.

- **La SCHED_OTHER** es la política de programación por defecto en Red Hat Enterprise Linux 8. Esta política utiliza el Programador Completamente Justo (CFS) para permitir un acceso justo al procesador a todos los hilos programados con esta política. Esta política es más útil cuando hay un gran número de hilos o cuando el rendimiento de los datos es una prioridad, ya que permite una programación más eficiente de los hilos en el tiempo. Cuando esta política está en uso, el planificador crea una lista de prioridad dinámica basada en parte en el valor de amabilidad de cada proceso.

[1] https://access.redhat.com/documentation/es-es/red_hat_enterprise_linux/8/html/monitoring_and_managing_system_status_and_performance/changing-the-priority-of-service-during-the-boot-process_tuning-scheduling-policy

Establecer las políticas del programador[2]

Ejemplo de cómo cambiar la política de un proceso a SCHED_FIFO con una prioridad de 50.

```
root@nodo1 ~]# ps -ef|grep -i httpd
root       48846       1  0 12:10 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     49564   48846  0 12:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     49567   48846  0 12:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     49568   48846  0 12:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     49569   48846  0 12:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
apache     49570   48846  0 12:24 ?        00:00:00 /usr/sbin/httpd -DFOREGROUND
root       51743   47424  0 13:06 pts/0    00:00:00 grep --color=auto -i httpd

[root@nodo1 ~]# chrt -m
SCHED_OTHER min/max priority    : 0/0
SCHED_FIFO min/max priority     : 1/99
SCHED_RR min/max priority       : 1/99
SCHED_BATCH min/max priority    : 0/0
SCHED_IDLE min/max priority     : 0/0
SCHED_DEADLINE min/max priority : 0/0

[root@nodo1 ~]# chrt -p 48846
pid 48846's current scheduling policy: SCHED_OTHER
pid 48846's current scheduling priority: 0
[root@nodo1 ~]#

[root@nodo1 ~]# chrt -f -p 50 49567
[root@nodo1 ~]#

[root@nodo1 ~]# chrt -p 49567
pid 49567's current scheduling policy: SCHED_FIFO
pid 49567's current scheduling priority: 50
[root@nodo1 ~]#
```

[2] https://www.youtube.com/watch?v=JPBQsTtHaIE

- Gestionar los perfiles de ajuste

Los administradores del sistema pueden optimizar el rendimiento de un sistema ajustando varias configuraciones de dispositivos en función de una variedad de cargas de trabajo de casos de uso. El demonio tuned aplica ajustes de sintonización tanto de forma estática como dinámica, utilizando perfiles de tuned que reflejan requisitos particulares de carga de trabajo.

El demonio tuned aplica ajustes de forma estática y dinámica[3]:

- **Ajuste estático:** El demonio tuned aplica la configuración del sistema cuando se inicia el servicio o al seleccionar un nuevo perfil de tuned. El ajuste estático configura parámetros de kernel predefinidos en perfiles que se aplican en tiempo de ejecución. Con el ajuste estático, los parámetros del kernel se establecen para las expectativas generales de rendimiento y no se ajustan a medida que cambian los niveles de actividad.

- **Ajuste dinámico:** El ajuste dinámico ajusta continuamente el ajuste para adaptarse a la carga de trabajo actual, comenzando con la configuración inicial declarada en el perfil de ajuste elegido.

Tuned se basa en dos tipos de perfiles que mejoran el rendimiento y se dividen en dos categorías:

- Ahorro de energía.

- Mejora de rendimiento. Estos a su vez podemos subdividirlos en:
  - Baja latencia para almacenamiento y red
  - Alto rendimiento para almacenamiento y red
  - Rendimiento de la máquina virtual
  - Rendimiento del host de virtualización.

```
[root@nodo1 ~]# tuned-adm list
Available profiles:
- balanced                    - General non-specialized tuned profile
- desktop                     - Optimize for the desktop use-case
- hpc-compute                 - Optimize for HPC compute workloads
- latency-performance           - Optimize for deterministic performance at the cost of
increased power consumption
- network-latency               - Optimize for deterministic performance at the cost of
increased power consumption, focused on low latency network performance
- network-throughput             - Optimize for streaming network throughput, generally
only necessary on older CPUs or 40G+ networks
- powersave                   - Optimize for low power consumption
- throughput-performance         - Broadly applicable tuning that provides excellent
performance across a variety of common server workloads
- virtual-guest               - Optimize for running inside a virtual guest
- virtual-host                - Optimize for running KVM guests
```

[3] https://www.redhat.com/sysadmin/linux-tuned-tuning-profiles

Tuning Profiles Distributed with CentOS/RHEL 8

TUNED PROFILE	PURPOSE
balanced	Ideal for systems that require a compromise between power saving and performance.
desktop	Derived from the balanced profile. Provides faster response of interactive applications.
throughput-performance	Tunes the system for maximum throughput.
latency-performance	Ideal for server systems that require low latency at the expense of power consumption.
network-latency	Derived from the latency-performance profile. It enables additional network tuning parameters to provide low network latency.
network-throughput	Derived from the throughput-performance profile. Additional network tuning parameters are applied for maximum network throughput.
powersave	Tunes the system for maximum power saving.
oracle	Optimized for Oracle database loads based on the throughput-performance profile.
virtual-guest	Tunes the system for maximum performance if it runs on a virtual machine.
virtual-host	Tunes the system for maximum performance if it acts as a host for virtual machines.

[4]

Cambiar profile del servidor con tuned:

```
[root@nodo1 ~]# tuned -amd --help

[root@nodo1 ~]# tuned-adm active
Current active profile: virtual-guest
[root@nodo1 ~]#

[root@nodo1 ~]# tuned-adm profile throughput-performance
[root@nodo1 ~]# tuned-adm active
Current active profile: throughput-performance

[root@nodo1 ~]# tuned-adm recommend
virtual-guest
[root@nodo1 ~]# tuned-adm profile virtual-guest

[root@nodo1 ~]# tuned-adm active
Current active profile: virtual-guest
[root@nodo1 ~]#
```

[4] https://www.thegeekdiary.com/beginners-guide-to-tuning-profiles-in-centos-rhel/

- Localizar e interpretar los diarios y los archivos de registro del sistema

Todos los eventos que suceden dentro del sistema operativo llevados a cabo tanto por el kernel o los procesos que estén corriendo deben ser registrados en archivos conocidos como logs, estos son muy útiles para la revisión de y corrección de errores o bien por temas de auditoría, de forma estándar estos archivos logs son guardados en el directorio ***/var/log***. En este directorio existen importantes archivos de log del sistema y de servicios:

- **boot.log**: Mensajes relacionados con el arranque del sistema (boot)
- **secure:** Eventos relacionados con autenticación y seguridad
- **maillog:** Eventos relacionados con el servicio de correo (postfix)
- **cron:** Mensajes relacionados con tareas programadas (crond)
- **messages**: El fichero de log más conocido y utilizado por los administradores ya que la mayoría de los eventos de syslog se registran aquí ***(rsyslog.service)***

ryslog

Syslog es el protocolo utilizado comúnmente por el sistema ***(kernel)***, procesos y programas para poder realizar los registros de eventos en los archivos determinados, un servicio o programa que utiliza este protocolo y que lo hemos utilizado en versiones pasadas de Red Hat Enterprise Linux es rsyslod.

Los eventos son clasificados por tipo de mensaje y prioridad (gravedad):

Prioridad	Codigo	Gravedad
emerg	0	El sistema no se puede usar
alert	1	Requiere acción inmediata
crit	2	Evento o condición crítica
err	3	Condición de error no crítica
warning	4	Advertencia
notice	5	Evento normal pero importante
Info	6	Condición informativa
debug	7	Mensaje a nivel depuración (debug)

El archivo de configuración del servicio rsyslog se encuentra en **/etc/rsyslog.conf**, es es en este donde se configura el tipo de mensaje y prioridad de los eventos mediante este archivo en el apartado **####RULES####** o en archivos *.conf dentro del directorio **/etc/rsyslog.d/**.

logrotate

Existe una utilidad en el sistema llamada logrotate que nos permite rotar los logs, para evitar que se llene el sistema de eventos en los archivos logs del directorio ***/var/log***, y consiste en renombrar el archivo de log regularmente con un sufijo o extensión en formato de fecha para saber el dia en que roto y creando un archivo vacío, notificando al servicio que debe escribir en él:

```
boot.log-20180911 roto el dia 11 de septiembre del 2018
messages-20181021 roto el dia 21 de octubre del 2018
```

La configuración de logrotate se encuentra en el fichero en **/etc/logrotate.conf** y este contiene lineas de configuracion como las de a continuación

```
#rota cada semana
weekly
##mantiene los últimos 4 archivos de log
rotate 4
## cuando se rota se cree uno nuevo
create
## que agregue el sufijo de fecha al archivo rotado
dateext
##se toman en cuenta los archivos *.conf dentro del directorio logrotate.d/
include /etc/logrotate.d
```

- Conservar los diarios del sistema

systemd-journald

Este servicio es integrado en las versiones RHEL 7.x y bien CentOS 7.x, es posible que sea una versión mejorada de rsyslog y su principal función o característica es tener todos los registros y eventos del sistema en una sola ubicación y esto nos permite realizar filtrados por usuario, servicios, horas etc. a diferencia de rsyslog que utiliza archivos separados de log.

Journal utiliza una base de datos ubicada en ***/run/log/journal***, y dado que todo lo que está en el directorio ***/run/*** al reiniciar se pierde esto nos da una idea que journal no es persistente al reinicio.

Trabajando con journalctl

Para llevar a cabo el análisis de logs o bien el monitoreo de eventos nos apoyamos del programa journalctl que con sus diferentes parámetros podemos obtener resultados bastantes granulares y complejos.

Ejecutando individualmente el siguiente comando obtendremos los eventos del sistema comenzando por la entrada más antigua, el programa resalta en negritas los mensajes de texto con severidad de aviso o advertencia y los mensajes de error o superiores lo resalta en rojo.

```
journalctl
```

Esto suele ser un poco complicado ya que obtendremos un sin fin de líneas y los mas sano es acotar estos resultados, mostrar la ultimas 8 lineas del registro

```
journalctl -n 8
```

Filtrar resultados por prioridad

```
journalctl -p err
journalctl -p 5
journalctl -p info
```

Similar al comando **tail -f o tailf, journalctl** nos proporciona el parámetro follow para tema de monitoreo en tiempo real

```
journalctl -f
```

Si se requiere buscar eventos por rango de tiempo utilizamos las opciones –since y –until con formato de fecha y hora **YYYY-MM-DD hh:mm:ss**

```
journalctl --since "2018-12-16 22:52:00" --until "2018-12-16 22:52:29"
```

Existe una manera de ver las líneas de un evento con un agregado de campos para obtener más información

```
journalctl -n 1 -o verbose
```

Eventos por servicio

```
journalctl -f _SYSTEMD_UNIT="sshd.service"
```

Hacer journal persistente

Como se había mencionado anteriormente journal no se mantiene después de un reinicio del sistema, pero si nuestra necesidad es que se mantenga de forma permanente en el file system llevaremos a cabo los siguientes pasos:

Crear el directorio ***/var/log/journal***

```
mdkir /var/log/journal
```

Asignar permisos y propietarios correspondientes

```
chown root:systemd-journal /var/log/journal
chmod 2755 /var/log/journal
```

#opcional determinar el peso (valor en MB) de los los para ser rotados:

```
vim /etc/systemd/journald.conf
SystemMaxUse=500
```

Reiniciar servicio para coger los nuevos parametros

```
systemctl restart systemd-journald
```

Consideraciones

- El journal tiene un mecanismo de rotación de registro mensualmente.
- El journal no podrá tener más del 10 % del sistema de archivos en el que está ubicado ni dejar menos del 15 % del sistema de archivos libre.
- Estos valores pueden ajustarse en el archivo de configuración ***/etc/systemd/journald.conf***
- Un avez realizado algún cambio en el archivo de configuración es importante reiniciar el ***servicio systemctl restart systemd-journald***

- Iniciar, detener y verificar el estado de los servicios de red

```
systemctl [ status | start | stop ] service
```

- Transferir archivos entre diferentes sistemas de forma segura

SFTP/FTP

SFTP funciona en un modelo cliente-servidor. Es un subsistema de SSH y admite todos los mecanismos de autenticación SSH.

Para abrir una conexión SFTP a un sistema remoto, use el comando sftp seguido del nombre de usuario del servidor remoto y la dirección IP o el nombre de dominio:

```
sftp remote_username@server_ip_or_hostname
```

SCP

SCP es un comando que, básicamente, permite realizar la transferencia de archivos. Podemos realizar la transferencia desde un ordenador local a un servidor remoto, desde el servidor remoto descargarnos un archivo en el ordenador local, o incluso enviar archivos entre servidores remotos.

- Transferir archivos hacia el servidor remoto:

```
scp archivo-en-local.tar user@1.2.3.4:/directorio/destino/servidor/remoto
```

- Transferir archivos desde el servidor remoto

```
scp user@1.2.3.4:/ruta/servidor/remoto/archivo.tgz archivo-en-local.tgz
```

RSYNC

Rsync es una herramienta de sincronización muy flexible y habilitada para la red. Debido a su presencia universal en sistemas Linux y sistemas similares a Unix, y su popularidad como herramienta para las secuencias de comandos del sistema, se incluye en la mayoría de las distribuciones de Linux de manera predeterminada.

```
rsync -avxHAX --numeric-ids --progress --exclude '/mnt/*' --exclude '/boot/*' /
/mnt/root
```

```
nohup rsync -avh --log-file=/tmp/DATA_SYNC_A.txt
/var/opt/documentum/ecaseprd/data/* /var/opt/documentum/ecaseprd/aristemp/ &
```

Configurar el almacenamiento local

- Enumerar, crear y eliminar particiones en discos MBR y GPT

Ejemplo de uso de fdisk para crear una partición en el espacio disponible:

```
[root@JBoss0 bonzo]# fdisk /dev/sda1
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x71056486.

Orden (m para obtener ayuda): m
Orden  Acción
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   g   create a new empty GPT partition table
   G   create an IRIX (SGI) partition table
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Opciones disponibles:

- m obtenemos la ayuda del fdisk.
- p print para obtener datos sobre las particiones.
- n new para crear una nueva partición.
- w write para grabar los cambios indicados.
- d delete borrar partición.

Ejercicio práctico: En un disco que añadamos crear 3 particiones primarias de 250MB, una extendida de 250MB, y el resto lógicas.

Para verificar que existen las particiones.

```
root@starwing:~# partprobe -s
/dev/sda: msdos partitions 1 2 3 4 <5 6>
```

- Crear y eliminar volúmenes físicos, Asignar volúmenes físicos a los grupos de volúmenes y Crear y eliminar volúmenes lógicos

Introducción LVM:

De forma simplificada podríamos decir que LVM es una capa de abstracción entre un dispositivo de almacenamiento (por ejemplo, un disco) y un sistema de ficheros. La ventaja que tienen más evidente es la flexibilidad frente al particionado tradicional, ya que permite distribuir el almacenamiento de una forma mucho más flexible.

Para entender el LVM debemos tener muy claros únicamente tres conceptos:

- **Volumen físico/Physical Volume (PV).**: Es una fuente de almacenamiento, es decir un dispositivo que nos proporciona espacio. Un PV no hay que formatearlo, simplemente se le entregará al LVM «en crudo» y desde ese momento será gestionado por el LVM, no volveremos a tocarlo.

- **Grupo de volúmenes/Volume Group (VG).**: Para poder usar el espacio de un PV, éste debe pertenecer a un VG. Un VG es un «disco» compuesto de UNO o más PVs y que crece simplemente añadiendo más PVs. A diferencia de un disco real, un VG puede crecer con el tiempo, sólo hay que «darle» un PV más.

- **Volumen Lógico/Logical Volume (LV).**: son «el producto final» del LVM. Son estos dispositivos los que usaremos para crear sistemas de ficheros, swap, etc… A diferencia de «sus primas» las particiones tradicionales, los LVs pueden crecer (mientras haya espacio en el VG)

Un LV puede crecer siempre que haya espacio libre en el VG al que pertenece. El LVM se encarga de que lo que haya sobre el LV (frecuentemente un sistema de ficheros) vea todo el espacio continuo.

Podemos cambiar el espacio asignado de un PV a un LV a otro PV (que tenga espacio suficiente libre). Me explico, yo puedo crear un LV de 10G en un PV que sea un disco. Si posteriormente meto en el VG un PV que sea un RAID, podría mover los 10G que estaba usando del disco al RAID, en caliente y de forma transparente al sistema de ficheros y las aplicaciones que lo usan.

Para poder disfrutar al máximo de la flexibilidad del LVM es importante tener la mayor cantidad de espacio libre en el VG (volveré a este tema más tarde), pero tarde o temprano nos quedaremos sin espacio en un VG (porque usemos todo el espacio de sus PVs).

Practica: Asignar PV a VG; crear, extender, LV; y crear el sistema de archivos ***(RHCSA apartado3)***

Partimos de la base de que ya tenemos los discos escaneados:

```
[root@serveripa]# ls /dev/sd*
/dev/sda   /dev/sda1 /dev/sdb /dev/sdc
```

Ahora creamos un Volumen físico al cual asignamos un disco:

```
[root@JBoss0 bonzo]# pvcreate /dev/sdc
  Physical volume "/dev/sdc" successfully created
```

Añadir Volumen físico a un Volumen de Grupo:

```
vgextend myvg1 /dev/sdc
```

Después seria extender el Volumen Logico:

```
lvextend -L+500M /dev/mapper/myvg1-home
```

Y por último hacer el resize del FS, esto puede ser de diferentes formas dependiendo del sistema de ficheros de cada LV **resize2fs** *<device> (ext2 / ext3 / ext4)* / **xfs*_growfs*** *<device> (xfs)*.

```
resize2fs /dev/myvg1/home
xfs_growfs /dev/rhel/var
```

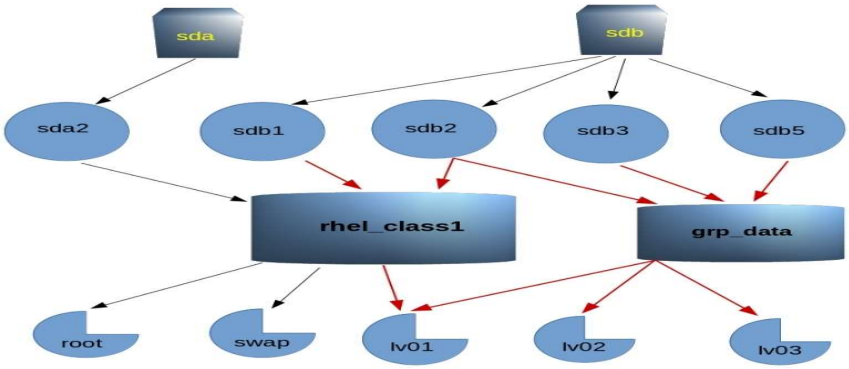
Ejemplo práctico FS de tipo xfs:

```
root@serveripa myvg1]# df -kh /var
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/rhel-var  2.0G  141M  1.9G   7% /var
```

```
[root@serveripa myvg1]# lvextend -L+500M /dev/mapper/rhel-var
  Size of logical volume rhel/var changed from 2.00 GiB (512 extents) to 2.49 GiB (637
extents).  Logical volume rhel/var successfully resized.
```

```
[root@serveripa myvg1]# xfs_growfs /dev/rhel/var
meta-data=/dev/mapper/rhel-var   isize=512    agcount=4, agsize=131072 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0 spinodes=0
data     =                       bsize=4096   blocks=524288, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal               bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
data blocks changed from 524288 to 652288
```

```
[root@serveripa myvg1]# df -kh /var
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/rhel-var  2.5G  141M  2.4G   6% /var
```



Cambiar particiones a LVM

La idea de este apartado es la de cambiar las particiones a tipo LVM, para ello, desde fdisk podemos gestionarlo de la siguiente manera:

```
fdisk /dev/vdb
```

Podemos listar las particiones:

```
Orden (m para obtener ayuda): p
Disk /dev/sda2: 20.4 GB, 20400046080 bytes, 39843840 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Identificador del disco: 0xa60a6959

  Disposit. Inicio    Comienzo      Fin      Bloques  Id  Sistema
  /dev/sdb5           314576896  344573951   14998528  83   Linux
  /dev/sdb6           344576000  374573055   14998528  83   Linux
  /dev/sdb7           374575104  390574079   7999488   82   Linux swap / Solaris
  /dev/sdb8           390576128  398573567   3998720   83   Linux
```

Para poder cambiar el tipo de partición se hace con el comando ***"t",*** pero debemos saber el código de tipo de partición que queremos usar y para saberlo se usa el comando ***"L"***:

```
Orden (m para obtener ayuda): L

 0  Vacía           24  NEC DOS         81  Minix / old Lin bf  Solaris
 1  FAT12           27  Hidden NTFS Win 82  Linux swap / So c1  DRDOS/sec (FAT-
 2  XENIX root      39  Plan 9          83  Linux           c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  84  Unidad C: ocult c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     85  Linux extendida c7  Syrinx
 5  Extendida       41  PPC PReP Boot   86  Conjunto de vol da  Datos sin SF
 6  FAT16           42  SFS             87  Conjunto de vol db  CP/M / CTOS / .
 7  HPFS/NTFS/exFAT 4d  QNX4.x          88  Linux plaintext de  Utilidad Dell
 8  AIX             4e  QNX4.x segunda  8e  Linux LVM       df  BootIt
```

Como queremos LVM seria seleccionar el **8e** y guardar los cambios:

```
Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!
Llamando a ioctl() para volver a leer la tabla de particiones.
```

Verificar las particiones:

```
root@starwing:~# partprobe -s
/dev/sda: msdos partitions 1 2 3 4 <5 6>
```

Gestion de Volumenes físicos (PV).

Este apartado anterior no es necesario ejecutarlo para crear un PVs.

Para ver los volúmenes físicos podemos usar el comando ***pvs*** y si queremos más información detallada sobre el PV se usa ***pvdisplay***:

```
 [root@JBoss0 bonzo]# pvs
  PV         VG     Fmt  Attr PSize   PFree
  /dev/sda2  centos lvm2 a--  <19,00g    0
```

```
[root@JBoss0 bonzo]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda2
  VG Name               centos
  PV Size               <19,00 GiB / not usable 3,00 MiB
  Allocatable           yes (but full)
  PE Size               4,00 MiB
  Total PE              4863
  Free PE               0
  Allocated PE          4863
  PV UUID               e2dQlV-R0uN-Z3Wd-sL5D-8bjR-KI0Q-BYEZCH
```

Crear un volumen lógico:

```
[root@JBoss0 bonzo]# pvcreate /dev/sda2
  Physical volume "/dev/sda2" successfully created
```

Eliminar un volumen lógico:

```
[root@JBoss0 bonzo]# pvremove /dev/sda2
  Labels on physical volume "/dev/sda2" successfully wiped.
```

Gestión de volúmenes de grupo (VG).

Crear un grupo de volúmenes:

```
vgcreate Nombre_Grupo_Vol NOMBRE_VOLUMENES_ASIGNADOS

vgcreate myvg1 /dev/vdb1 /dev/vdb2
```

Eliminar un grupo de volúmenes:

```
vgremove myvg1
```

Para ver los volúmenes de grupo existentes podemos usar el comando ***vgs*** y si queremos más información detallada sobre el VG se usa ***vgdisplay***:

```
[root@JBoss0 bonzo]# vgs
  VG     #PV #LV #SN Attr   VSize   VFree
  centos   1   2   0 wz--n- <19,00g    0
```

```
[root@JBoss0 bonzo]# vgdisplay
  --- Volume group ---
  VG Name               centos
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               <19,00 GiB
  PE Size               4,00 MiB
  Total PE              4863
  Alloc PE / Size       4863 / <19,00 GiB
  Free  PE / Size       0 / 0
  VG UUID               vIUogQ-9YZs-shm0-DlJs-fpKK-wS6o-7UtsB4
```

Como agregar un volumen a un grupo existente:

```
vgextend grupoVol /dev/vdb1
```

Gestión de volúmenes lógicos (LV).

Para ver los volúmenes lógicos podemos usar el comando ***lvs*** y si queremos más información detallada sobre el LV se usa ***lvdisplay***:

```
[root@serveripa bonzo]# lvs
  LV    VG      Attr        LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync
Convert
  home myvg1  -wi-ao----  2.98g
  var  rhel   -wi-ao----  2.49g
  root rhel00 -wi-ao---- 12.99g
```

```
 [root@serveripa bonzo]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/rhel00/root
  LV Name                root
  VG Name                rhel00
  LV UUID                LrjVhq-qX56-AAUw-xCYR-tUoz-Olpc-8NlQr9
  LV Write Access        read/write
  LV Creation host, time localhost.localdomain, 2019-08-08 12:18:13 -0700
  LV Status              available
  # open                 1
  LV Size                12.99 GiB
  Current LE             3325
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:0
```

Crear volúmenes lógicos:

```
lvcreate Nom_Vol_Grupo -n lv01 -L 300M
```

Extender el Volumen Lógico:

```
lvextend -L+500M /dev/mapper/myvg1-home
lvextend [Espacio_asignado] [LV a ampliar(es un dir diferente a FS(df)] [PV a
usar]
lvextend -L+120M /dev/rhel/var /dev/sdc
```

Y por último hacer el resize del FS, esto puede ser de diferentes formas dependiendo del sistema de ficheros de cada LV **resize2fs** *<device> (ext2 / ext3 / ext4)* / **xfs*_growfs*** *<device> (xfs)*.

```
resize2fs /dev/myvg1/home
xfs_growfs /dev/rhel/var
```

- Configurar los sistemas para montar los sistemas de archivos durante el arranque con un ID único universal (UUID) o una etiqueta

Trabajaremos con Volúmenes lógicos, por tanto, creamos unos cuantos de ellos:

```
lvcreate myvg1 -n lv01 -L 300M
lvcreate myvg1 -n lv02 -L 300M
lvcreate myvg1 -n lv03 -L 300M
```

Comandos para crear un el sistema de ficheros *(filesystem)* según su Tipo.

```
[root@rhel7test]#mkfs.
mkfs.btrfs     mkfs.ext2      mkfs.ext4      mkfs.minix     mkfs.vfat
mkfs.cramfs    mkfs.ext3      mkfs.fat       mkfs.msdos     mkfs.xfs
```

Asignar sistema de archivos a un volumen lógico creado:

a) Creamos un sistema de ficheros con ext3 en el LV lv01

```
mkfs.ext3 /dev/myvg1/lv01
```

b) Creamos un sistema de ficheros ext3 en el LV lv02 con una ***label*** my-ext4.

```
mkfs.ext4 -L my-ext4 /dev/myvg1/lv02
```

Añadir un ***label*** a un FS:

```
blkid|grep myvg1
tune2fs -L my-lb03 /dev/myvg1/lv01
blkid|grep myvg1/lv01
```

Montar sistemas de archivos con UUID universal o con label de forma permanente:

c) Crear un directorio donde montar los FS (LV):

```
mkdir /mnt/{lv01,lv02,lv03}
ls -ls /mnt/
```

d) Montar el FS mediante su ***UUID***, obtener el UUID y añadir una línea a /etc/fstab:

```
blkid|grep myvg1

echo "UUID="794b4ca6-05e1-434d-a256-8ea2b37f977c " /mnt/lv01 ext3 default 0 0" >>
/etc/fstab
```

e) Montar el FS mediante su ***label***, obtener el ***label*** y añadir una línea a /etc/fstab

```
blkid|grep myvg1

echo "LABEL="my-ext4" /mnt/lv02 ext4 default 0 0" >> /etc/fstab
```

f) Montar el FS mediante su ***"path regular"***, añadir una línea a /etc/fstab

```
blkid|grep myvg1
echo "/dev/mapper/myvg1-lv03" /mnt/lv03 ext3 default 0 0" >> /etc/fstab
```

Verificar que está bien documentado en fstab

```
cat /etc/fstab
```

Por último, montar todos los FS con el siguiente comando:

```
mount -a
```

Podemos verificar el sistema de archivos con df:

```
df -hT
```

- Agregar particiones y volúmenes lógicos nuevos, y cambiar a un sistema de forma no destructiva

Trabajaremos con la swap, pese a que la misma este en un LVM no se debe extender como cualquier otro LV. Cuando se agrega más swap se debe crear una nueva partición.

Ver cuantas particiones tenemos de swap:

```
[root@serveripa bonzo]# swapon
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition   2G   0B   -1
```

```
[root@serveripa bonzo]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dm-1                                       partition       2093052 0       -1
```

Ver espacio libre en particiones:

```
[root@serveripa bonzo]# parted /dev/sda unit MB print freeclear
Model: VMware, VMware Virtual S (scsi)
Disk /dev/sda: 21475MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number  Start    End      Size     Type      File system  Flags
        0.03MB   1.05MB   1.02MB             Free Space
 1      1.05MB   1075MB   1074MB   primary   xfs          boot
 2      1075MB   15024MB  13949MB  primary                lvm
 3      15024MB  17176MB  2152MB   primary                lvm
 4      17176MB  21475MB  4299MB   extended
 5      17177MB  19327MB  2151MB   logical                lvm
 6      19328MB  21475MB  2146MB   logical                lvm
```

Agregar más swap al sistema:

Agregar swap mediante una particion normal.

Agregar 500 MB a la swap, para ello crearemos una nueva particion en el /dev/sda:

```
[root@serveripa bonzo]# fdisk /dev/sdd
Welcome to fdisk (util-linux 2.23.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x521279e7.
```

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-1048575, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-1048575, default 1048575): +500MB
Partition 1 of type Linux and of size 477 MiB is set

Command (m for help): p

Disk /dev/sdd: 536 MB, 536870912 bytes, 1048576 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x521279e7

   Device Boot      Start         End      Blocks   Id  System
/dev/sdd1            2048      978943      488448   83  Linux

Hex code (type L to list all codes): 82
Changed type of partition 'Linux' to 'Linux swap / Solaris'

Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
Syncing disks.
```

Verificar la tabla de particiones:

```
[root@serveripa bonzo]# partprobe -s
/dev/sda: msdos partitions 1 2 3 4 <5 6>
/dev/sdd: msdos partitions 1
```

```
[root@serveripa bonzo]# mkswap /dev/sdd1
Setting up swapspace version 1, size = 488444 KiB
no label, UUID=fa499a0d-205d-4607-b91b-fbd1b8b85fb1

[root@serveripa bonzo]# echo "UUID=fa499a0d-205d-4607-b91b-fbd1b8b85fb1 swap swap
defaults 0 0">> /etc/fstab
```

```
[root@serveripa bonzo]# swapon
NAME      TYPE      SIZE USED PRIO
/dev/dm-1 partition   2G   0B   -1
```

Montar la swap creada:

```
[root@serveripa bonzo]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dm-1                                       partition       2093052 0       -1

[root@serveripa bonzo]# swapon -a

[root@serveripa bonzo]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dm-1                                       partition       2093052 0       -1
/dev/sdd1                                       partition       488444  0       -2
```

Agregar swap mediante un LV.

Se crea un nuevo LV desde un VG que tenga espacio libre:

```
[root@serveripa bonzo]# vgs
  VG     #PV #LV #SN Attr   VSize  VFree
  myvg1    2   1   0 wz--n-  4.00g 1.02g
  rhel     2   1   0 wz--n-  3.00g    0
  rhel00   1   1   0 wz--n- 12.99g    0
  vgswap   1   1   0 wz--n-  2.00g    0
```

```
 [root@serveripa bonzo]# lvcreate myvg1 -n swap01 -L 200M
  Logical volume "swap01" created.
```

```
[root@serveripa bonzo]# mkswap /dev/myvg1/swap01
Setting up swapspace version 1, size = 204796 KiB
no label, UUID=1c314e0d-ba1a-4e1f-a05d-27d1227aa093

[root@serveripa bonzo]# echo "UUID=1c314e0d-ba1a-4e1f-a05d-27d1227aa093 swap swap
defaults
```

```
[root@serveripa bonzo]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dm-1                                       partition       2093052 0       -1
```

```
[root@serveripa bonzo]# swapon -a
```

```
[root@serveripa bonzo]# swapon -s
Filename                                Type            Size    Used    Priority
/dev/dm-1                                       partition       2093052 0       -1
/dev/dm-4                                       partition       204796  0       -2
```

- Virtual Data Optimizer (VDO).

Virtual Data Optimizer (VDO) proporciona reducción de datos en línea para Linux en forma de deduplicación, compresión y aprovisionamiento ligero. Cuando configura un volumen VDO, especifica un dispositivo de bloque en el que construir su volumen VDO y la cantidad de almacenamiento lógico que planea presentar.

El comando ***"vdo"*** ya no están disponibles en ***RHEL9***, quedó obsoleto en la versión RHEL8.5. La tecnología VDO ahora está integrada en LVM, que permite crear LV deduplicados y/o comprimidos.

Práctica VDO y LVM

```
[root@rhel9 ~]# dnf install vdo
Updating Subscription Management repositories.
Last metadata expiration check: 2:19:10 ago on Sun 06 Nov 2022 16:51:31 CET.
Package vdo-8.1.1.360-1.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@rhel9 ~]#
```

```
[root@rhel9 ~]# vgcreate vg_vdo /dev/sdc
  Physical volume "/dev/sdc" successfully created.
  Volume group "vg_vdo" successfully created

[root@rhel9 ~]# lvcreate --type vdo -n vdo-lv -L 5G  -V 10G vg_vdo
    The VDO volume can address 2 GB in 1 data slab.
    It can grow to address at most 16 TB of physical storage in 8192 slabs.
    If a larger maximum size might be needed, use bigger slabs.
  Logical volume "vdo-lv" created.
```

```
[root@rhel9 ~]#  mkfs.ext4 -E nodiscard /dev/vg_vdo/vdo-lv
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 2621440 4k blocks and 655360 inodes
Filesystem UUID: b7d1aadf-2044-4680-8cb1-1618e7d234f6
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
[root@rhel9 ~]# lvs -a vg_vdo
  LV              VG      Attr       LSize   Pool    Origin Data%  Meta%  Move Log
Cpy%Sync Convert
```

```
  vdo-lv          vg_vdo vwi-a-v--- 10.00g vpool0        0.01
  vpool0          vg_vdo dwi-------  5.00g               60.04
  [vpool0_vdata] vg_vdo Dwi-ao----  5.00g
[root@rhel9 ~]#
```

```
root@rhel9 ~]# mkdir /vdo

[root@rhel9 ~]# mount /dev/vg_vdo/vdo-lv /vdo

[root@rhel9 ~]# df -h /vdo
Filesystem                  Size  Used Avail Use% Mounted on
/dev/mapper/vg_vdo-vdo--lv  9.8G   24K  9.3G   1% /vdo
```

Crear y configurar sistemas de archivos

- Crear, montar, desmontar y utilizar los sistemas de archivos vfat, ext4 y xfs

Crear el sistema de ficheros (filesystem)

```
mkfs.btrfs     mkfs.ext2      mkfs.ext4      mkfs.minix     mkfs.vfat
mkfs.cramfs    mkfs.ext3      mkfs.fat       mkfs.msdos     mkfs.xfs

[root@rhel7test]# mkfs.[type] <device>
[root@rhel7test]# mkfs.ext4 /dev/sdb1
```

Montar el FS.

a) Montar FS temporalmente:

```
[root@rhel7test]# mount -t ext4 /dev/sdb1 /mnt/sdb1
```

b) Montar FS de forma permanente:

```
echo "/dev/sdb1 /mnt/sdb1 ext4 defaults 0 0" >> /etc/fstab
```

a. Montarlo todos los FS establecidos en /etc/fstab:

```
[root@rhel7test]# mount -a
```

Desmontar el FS:

```
[root@rhel7test]# umount /mnt/sdb1
```

- Montar y desmontar los sistemas de archivos de red utilizando NFS y CIFS

Montar y desmontar sistemas de archivos de red NFS.

Instalar el servidor NFS y configuración básica del servidor:

```
# yum install nfs-kernel-server
```

- Crear un directorio que queramos compartir con la máquina por ejemplo el ***/srv/nfs***:

```
# mkdir -p /srv/nfs
```

- Modificamos el fichero ***/etc/exports*** añadiendo las carpetas que vamos a compartir y con quien compartimos:

```
/srv/nfs 192.168.122.2(rw,sync,no_root_squash,no_subtree_check)
```

- Reiniciar el servicio NFS:

```
# systemctl restart nfs-kernel-server
```

Montar NFS compartido en sistema de archivos locales:

- Ir al servidor donde queremos montar el NFS e instalar los paquetes si no los tiene:

```
# yum install nfs-util nfs-common
```

- Saber que está exportando un servidor:

```
# showmount –e servidoripa.example.com
```

- Crear directorio para montar el NFS compartido:

```
# mkdir /temp-nfs
```

- Crear un montaje permanente por fstab:

```
# echo "servidoripa.example.com:/srv/nfs /temp-nfs nfs _netdev 0 0" >> /etc/fstab
```

- Montar los filesystems del sistema de archivos:

```
# mount -a
```

Otros ejemplos de montaje no permanente:

```
# mount -t nfs -o vers=3 servidoripa.example.com:/srv/nfs /temp-nfs
# mount -t nfs servidoripa.example.com:/srv/nfs /temp-nfs
```

Montar y desmontar sistemas de archivos de red SAMBA (CIFS) (ESTO NO ENTRA EN RHEL8.

CIFS es el sistema de archivos que utiliza SAMBA para trabajar.

```
systemctl restart smb nmb
```

Los principales paquetes que necesitamos para trabajar con samba como cliente son, samba-client samba-common, podemos ver los paquetes relacionados con samba que hay en el repositorio de esta manera:

```
# yum whatprovides  */samba
```

Para instalar samba completo, como servidor, solo hace falta el siguiente comando:

```
# yum install samba cifs-utils
```

Pero si vamos a instalar solo un cliente con el siguiente repositorio es suficiente:

```
# yum install samba-client cifs-utils
```

Si estamos en un Active Directory hay que asegurarse de que tenemos los siguientes paquetes instalados.

```
# yum install samba-winbind
# yum install samba-winbind-clients
# yum install samba-winbind-krb5-locator
```

Y estos serían todos los paquetes que debería tener el servidor

```
samba-libs  samba-winbind-krb5-locator  samba-winbind-modules  samba-vfs-glusterfs
samba-winbind samba-client samba-common samba-winbind-clients samba
```

Como conectar a un directorio compartido:

```
# smbclient //servidoripa.example.com/[NAME_DIR_SHARED] -U USUARIO

# smbclient -L servidoripa.example.com

# smbclient //servidoripa.example.com/public -U sambauser1
root´s password: PASSWORD
       smb: \> ?
       smb: \> help
       smb: \> exit
```

Como montar un directorio compartido por samba en nuestro sistema de archivos:

- Crear directorio local:

```
# mkdir /temporalmente
```

- Montaje de forma manual no permanente, pedirá password para establecer conexión:

```
#      mount      -t      cifs      -o      username=sambauser1,password=password
//servidoripa.example.com/data /temporalmente
```

- Desmontar para hacerlo correctamente:

```
# umount /temporal
```

Montar un directorio compartido por samba de forma permanente:

- Para que no nos pida la contraseña cada vez que el sistema monte los filesystem por reinicio del servidor, podemos crear un archivo que contenga las credenciales:

```
# /root/creed <<EOF
> username=sambauser1
> password=password
> EOF
```

Cambiamos permisos a dicho fichero para que solo lo pueda leer el root:

```
# chmod 400 /root/creed
```

Añadimos la línea siguienre al fstab y realizamos un montaje de las unidades:

```
# echo "//servidoripa.example.com/data      /permanente                    cifs
_netdev,credentials=/root/cred 0 0" >> /etc/fstab

# mount –a
```

- Configurar autofs.

El `autofs` es una alternativa a `/etc/`, consta de los siguientes componentes: puede montar y desmontar sistemas de archivos automáticamente (bajo demanda), ahorrando así recursos del sistema. Puede utilizarse para montar sistemas de archivos como NFS, AFS, SMBFS, CIFS y sistemas de archivos locales.

Para que funcione tiene que estar el servicio instalado y habilitado:

```
dnf install autofs
systemctl enable --now autofs
```

El archivo de mapa maestro

El servicio autofs utiliza /etc/auto.master como su archivo de configuración principal por defecto y el /etc/autofs.conf para configurar el servicio.

```
 [root@nodo1 ~]# ls -lrt /etc/autofs.conf /etc/auto.master
-rw-r--r--. 1 root root   795 Oct 30  2020 /etc/auto.master
-rw-r--r--. 1 root root 15137 Oct 30  2020 /etc/autofs.conf

[root@nodo1 ~]# grep -i auto.master /etc/autofs.conf
#master_map_name = auto.master
```

El archivo de mapa maestro enumera los puntos de montaje controlados por `autofs`, y sus correspondientes archivos de configuración o fuentes de red conocidos como mapas de automontaje. El formato del mapa maestro es el siguiente:

```
mount-point  map-name  options
```

Las variables utilizadas en este formato son:

- ***mount-point:*** El punto de montaje `autofs`; por ejemplo, `/mnt/data/`
- ***map-file:*** El archivo fuente del mapa, que contiene una lista de puntos de montaje y la ubicación del sistema de archivos desde la que se deben montarse
- ***options****:* Si se suministran, se aplican a todas las entradas del mapa dado, si no tienen ellas mismas opciones especificadas.

Archivos de mapas

Los archivos de mapa configuran las propiedades de los puntos de montaje individuales bajo demanda. El contador automático crea los directorios si no existen. Si los directorios existen antes de que se inicie el contador automático, éste no los eliminará cuando salga. Si se especifica un tiempo de espera, el directorio se desmonta automáticamente si no se accede a

él durante el periodo de tiempo de espera. El formato general de los mapas es similar al del mapa maestro:

```
mount-point  options  location
```

- ***mount-point:*** Se refiere al punto de montaje de `autofs`.
- ***options:*** son las opciones de montaje para las entradas del mapa que no especifican sus propias opciones. Este campo es opcional
- ***location:*** Se refiere a la ubicación del sistema de archivos, como una ruta del sistema de archivos local.

Ejemplo Un archivo de mapas

```
[root@nodo1~]#cat /etc/auto.misc
payroll  -fstype=nfs4  personnel:/dev/disk/by-uuid/52b94495-e106-4f29-b868-fe6f6c2789b1
sales    -fstype=xfs   :/dev/disk/by-uuid/5564ed00-6aac-4406-bfb4-c59bf5de48b5
```

La primera columna del archivo de mapa indica el punto de montaje `autofs`: `sales` y `payroll` del servidor llamado `personnel`. La segunda columna indica las opciones para el montaje de `autofs`. La tercera columna indica el origen del montaje.

Siguiendo la configuración dada, los puntos de montaje `autofs` serán `/home/payroll` y `/home/sales`. La opción `-fstype=` suele omitirse y, por lo general, no es necesaria para su correcto funcionamiento.

Práctica 1: Configuración de punto de montaje autofs con un Filesystem local:

```
[root@rhel9 ~]#  cat /etc/auto.master|grep -v ^#
/misc   /etc/auto.misc
/net    -hosts
+dir:/etc/auto.master.d
+auto.master
```

```
[root@rhel9 ~]# cat /etc/auto.master.d/local.autofs
/-      /etc/auto.local
```

```
[root@rhel9 ~]# cat /etc/auto.local
/local_fs -fstype=ext2  :/dev/disk/by-uuid/5f5ce1ff-5d15-4ecb-b07b-3f681d24a8d
```

```
[root@rhel9 ~]# systemctl restart autofs.service
```

```
[root@rhel9 ~]# df -h /local_fs
Filesystem                          Size  Used Avail Use% Mounted on
/dev/mapper/vg_local-lv_local_autofs  5.0G   24K  4.7G   1% /local_fs
```

Práctica 2: Configuración de punto de montaje autofs con un NFS:

```
[root@rhel9-client ~]# cat /etc/auto.master|grep -v ^#
/misc   /etc/auto.misc
/net    -hosts
+dir:/etc/auto.master.d
+auto.master
```

```
[root@rhel9-client ~]# cat /etc/auto.master.d/nfs.autofs
/- /etc/auto.shared
```

```
[root@rhel9-client ~]# cat /etc/auto.shared
/autofs_shared  -rw,sync,fstype=nfs4    192.168.0.188:/autofs_local
```

```
[root@rhel9-client ~]# systemctl restart autofs.service
```

```
[root@rhel9-client ~]# df -h /autofs_shared
Filesystem                  Size  Used Avail Use% Mounted on
192.168.0.188:/autofs_local  5.0G     0  4.7G   0% /autofs_shared
```

Práctica 3: Automatización de los directorios de usuario del servidor NFS con el servicio autofs

Este procedimiento describe cómo configurar el servicio autofs para que monte automáticamente los directorios personales de los usuarios. Está practica suele salir en el examen.

- Necesitamos configurar un NFS:

```
[root@rhel9 ~]# ip a show enp0s3|grep -i "inet "
    inet 192.168.0.188/24 brd 192.168.0.255 scope global noprefixroute enp0s3
```

```
[root@rhel9 ~]# cat /etc/exports
/autofs_local *(rw,sync,no_root_squash,no_subtree_check)
```

```
[root@rhel9 ~]# showmount -e 192.168.0.188
Export list for 192.168.0.188:
/autofs_local *
```

- Los **UIDs** y **GIDs** tienen que ser iguales para los usuarios en ambos servidores, sino la ejecución de esta práctica no será posible.

```
[root @rhel9 ~]$ id bonzo
uid=1000(bonzo) gid=1000(bonzo) groups=1000(bonzo)

[root@rhel9-client ~]# id bonzo
uid=1000(bonzo) gid=1000(bonzo) groups=1000(bonzo)
```

- Verificado lo anterior copiamos todos los datos de los home de cada usuario, en el share que compartimos:

```
[root@rhel9 ~]# rsync -avh /home/* /autofs_local/
sending incremental file list
bonzo/
bonzo/.bash_history
bonzo/.bash_logout
bonzo/.bash_profile
bonzo/.bashrc
pepe/
pepe/.bash_history
pepe/.bash_logout
pepe/.bash_profile
pepe/.bashrc

sent 2.83K bytes  received 176 bytes  6.00K bytes/sec
total size is 2.19K  speedup is 0.73
```

- Establecer la configuración del autofs:

```
[root@rhel9-client ~]# cat /etc/auto.master|grep -v ^#
/misc   /etc/auto.misc
/net    -hosts
+dir:/etc/auto.master.d
+auto.master
```

```
[root@rhel9-client ~]# cat /etc/auto.master.d/nfs_home.autofs
/home   /etc/auto.home
```

```
[root@rhel9-client ~]# cat  /etc/auto.home
*       -rw,sync,fstype=nfs4    192.168.0.188:/autofs_local/&
```

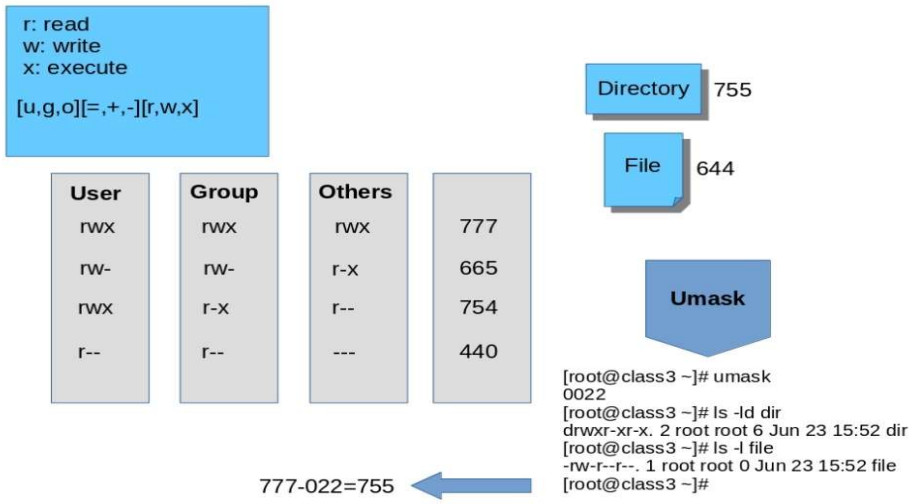
- Comprobaciones:

```
[root@rhel9-client ~]# su - bonzo
[bonzo@rhel9-client ~]$ pwd
/home/bonzo
[bonzo@rhel9-client ~]$ df -h .
Filesystem                          Size  Used Avail Use% Mounted on
192.168.0.188:/autofs_local/bonzo  5.0G     0  4.7G   0% /home/bonzo
```

```
[root@rhel9-client ~]# su - pepe
[pepe@rhel9-client ~]$ pwd
/home/pepe
[pepe@rhel9-client ~]$ df -h .
Filesystem                          Size  Used Avail Use% Mounted on
192.168.0.188:/autofs_local/pepe  5.0G     0  4.7G   0% /home/pepe
```

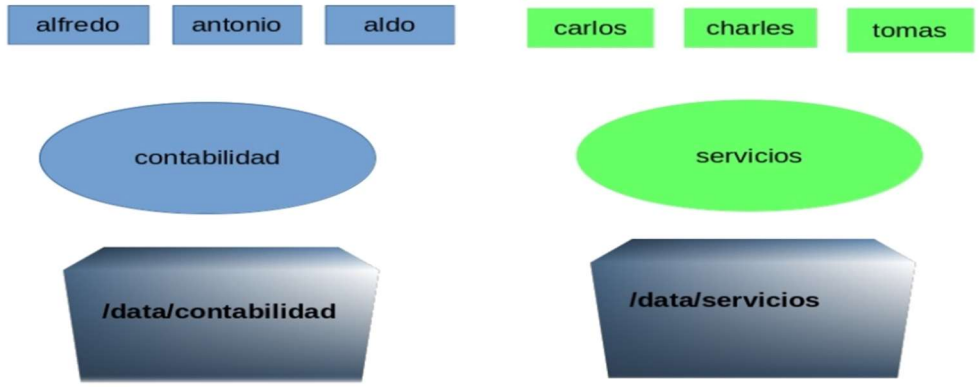
- Crear y configurar directorios con GID definido para la colaboración

Con el GID establecemos que los miembros que pertenezcan a un grupo van a poder compartir archivos y directorios entre ellos. Podemos establecer restricciones para que solo el usuario que creo el directorio o el archivo pueda eliminarlo, o que todo lo que se cree en este directorio pertenezca a este grupo. A continuación, un esquema de cómo funcionan los permisos básicos en Linux y como calcularlos:



Para entender este apartado vamos a crear la siguiente estructura de usuarios, grupos y directorios de colaboración:

```
#!/bin/bash
# Script de preparacion para la clase 6 del entrenamiento: ULTIMATE RHCSA TRAINING

# Creando usuarios y estableciendo password
for i in alfredo antonio aldo carlos charles tomas; do useradd $i; done
for i in alfredo antonio aldo carlos charles tomas; do echo "password" | passwd $i --
stdin; done

# Creando grupos
groupadd contabilidad
groupadd servicios

# Asignando usuarios a grupos
for i in alfredo antonio aldo; do usermod -aG contabilidad $i; done
for i in carlos charles tomas; do usermod -aG servicios $i; done

# Verificando
getent group contabilidad
getent group servicios

# Creando direcotrio de trabajo
mkdir -p /data/{contabilidad,servicios}
```



Para asignar un grupo a un directorio podemos usar los comandos chgrp o chmod:

```
[root@serveripa data]# ls -lrt /data
total 0
drwxr-xr-x. 2 root root 6 Aug 11 01:37 servicios
drwxr-xr-x. 2 root root 6 Aug 11 01:37 contabilidad
```

```
[root@serveripa data]# chgrp contabilidad contabilidad/
[root@serveripa data]# ls -ld servicios
drwxr-xr-x. 2 root contabilidad 6 Aug 11 01:37 contabilidad
```

```
[root@serveripa data]# chwon :servicios servicios/
[root@serveripa data]# ls -ld servicios
drwxr-xr-x. 2 root servicios    6 Aug 11 01:37 servicios
```

Para cambiar permisos de un directorio o fichero se hace con el comando chmod, se puede hacer de varias formas y por ello es interesante ver la tabla al principio del capítulo, a continuación, varios ejemplos de asignación de permisos:

```
[root@serveripa data]# chmod 770 contabilidad/
[root@serveripa data]# ls -ld contabilidad
drwxrwx---. 2 root contabilidad 6 Aug 11 01:37 contabilidad

[root@serveripa data]# chown 777 contabilidad/
[root@serveripa data]# ls -ld contabilidad
drwx-w--wx. 2  777 contabilidad 6 Aug 11 01:37 contabilidad

[root@serveripa data]# chmod g=rw contabilidad/
[root@serveripa data]# ls -ld contabilidad/
drwxrw----. 2 root contabilidad 6 Aug 11 01:37 contabilidad/

[root@serveripa data]# chmod g=--- contabilidad/
[root@serveripa data]# ls -ld contabilidad/
drwx------. 2 root contabilidad 6 Aug 11 01:37 contabilidad/
```

```
[root@serveripa data]# chmod g+w contabilidad/
[root@serveripa data]# ls -ld contabilidad/
drwx-w----. 2 root contabilidad 6 Aug 11 01:37 contabilidad/

[root@serveripa data]# chmod g+w,o+wx contabilidad/
[root@serveripa data]# ls -ld contabilidad/
drwx-w--wx. 2 root contabilidad 6 Aug 11 01:37 contabilidad/
```

Permisos especiales del SUID, GUID, Sticky Bit:

```
        rwx rwx rwx
        421 421 421
        4   2   1
        2770
```

- **SUID**: significa que el que lo ejecute va a tener los mismos permisos que el que creó el archivo. Esto puede llegar a ser muy util en algunas situaciones, pero hay que utilizarlo con cuidado, dado que puede generar grandes problemas de seguridad. Esto se usa en el archivo binario /usr/bin/passwd. passwd necesita modificar el archivo /etc/passwd y /etc/shadow que guarda la información de las cuentas de usuario y los hashes de contraseña respectivamente, pero estos archivos solo pueden ser modificados por root.

```
[root@serveripa data]#echo "Di Hola" >> contabilidad/escris-root.sh
[root@serveripa data]# cd ./contabilidad
[root@serveripa data]# chmod 4775 ./contabilidad/escris-root.sh
[root@serveripa data]# chmod -x ./contabilidad/escris-root.sh

[root@serveripa contabilidad]# su aldo
[aldo@serveripa contabilidad]$ cat escris-root.sh
Di Hola
```

- **GUID**: sería la “Herencia de grupo”, que sirve para que todos los archivos que se creen en el directorio de colaboracion pertenezcan al grupo padre. Esto se consigue de la siguiente forma:

```
[root@serveripa data]# chmod 2770 contabilidad/
[root@serveripa data]# ls -ld contabilidad/
drwxrws---. 2 777 contabilidad 6 Aug 11 01:37 contabilidad/
```

- **El Sticky bit** se utiliza para permitir que cualquiera pueda escribir y modificar sobre un archivo o directorio, pero que solo su propietario o root pueda eliminarlo. El **sticky bit** se aplica al tercer grupo de permisos, al grupo otros. Y como en los otros dos, sustituye al permiso de ejecución `x` por una `t` o `T`. El significado de una letra mayúscula o minúscula es el mismo, si es una `T` significa que está implementado el `sticky bit` pero que debajo de él no hay permisos de ejecución `x`.

```
[root@serveripa data]# chmod 3770 contabilidad/
[root@serveripa data]# ls -ld contabilidad/
drwxrws--T. 2  777 contabilidad 6 Aug 11 01:37 contabilidad

[root@serveripa data]# chmod +t servicios/
[root@serveripa data]# ls -ld servicios
drwxr-xr-t. 2 root servicios    6 Aug 11 01:37 servicios
```

Implementar, configurar y mantener sistemas

- Programar tareas con at y cron

At

Los ficheros de configuración para permitir o restringir realizar tareas mediante ***at*** son ***/etc/at.allow*** y ***/etc/at.deny***. Por defecto el ***/etc/at.allow*** no existe, ya que si este esta, solo los usuarios dentro de ese archivo podrán programar tareas, de tal manera que por defecto solo esta ***/etc/at.deny*** y donde se indican los usuarios que no pueden crear tareas.

Para planificar una tarea lo primero es identificar la hora del sistema:

```
[root@serveripa tmp]# timedatectl
      Local time: Sun 2019-08-11 07:59:52 PDT
  Universal time: Sun 2019-08-11 14:59:52 UTC
        RTC time: Sun 2019-08-11 14:59:52
       Time zone: America/Los_Angeles (PDT, -0700)
     NTP enabled: yes
NTP synchronized: no
 RTC in local TZ: no
      DST active: yes
 Last DST change: DST began at
                  Sun 2019-03-10 01:59:59 PST
                  Sun 2019-03-10 03:00:00 PDT
 Next DST change: DST ends (the clock jumps one hour backwards) at
                  Sun 2019-11-03 01:59:59 PDT
                  Sun 2019-11-03 01:00:00 PST
```

Ahora programaremos la siguiente tarea a las 8:03 se cierra el comentario con ctrl+d :

```
[root@serveripa tmp]# at 8:03
at> touch /root/at-new
at> echo "hola esto es con at" >> /root/at-new
at> <EOT>
job 1 at Sun Aug 11 08:03:00 2019
```

Para ver las tareas pendientes con ***at*** se usa ***atq***:

```
[root@serveripa tmp]# atq
1       Sun Aug 11 08:03:00 2019 a root
```

Para ver el contenido de esa tarea se hace con ***at –c $Num_Tarea***:

```
[root@serveripa tmp]# at -c 1|tail -10
OLDPWD=/var/tmp/ojetecalor; export OLDPWD
cd /var/tmp || {
        echo 'Execution directory inaccessible' >&2
        exit 1
```

```
}
${SHELL:-/bin/sh} << 'marcinDELIMITER7cd6c506'
touch /root/at-new
echo "hola esto es con at" >> /root/at-new

marcinDELIMITER7cd6c506
```

Una vez llegada la hora se ejecuta la tarea (comprobación):

```
[root@serveripa tmp]# ls -lrt /root/at-new
-rw-r--r--. 1 root root 20 Aug 11 08:03 /root/at-new

[root@serveripa tmp]# cat /root/at-new
hola esto es con at
```

Eliminar una tarea se hace con ***atrm $Num_Tarea***:

```
[root@serveripa tmp]# atrm 2
```

Cron

crond; servicio empleado en la programación de tareas mediante 'cron' y 'anacron'. Los ficheros de configuración para permitir o restringir realizar tareas mediante ***cron*** son ***/etc/cron.allow*** y ***/etc/cron.deny***. Por defecto el ***/etc/cron.allow*** no existe, ya que si este esta, solo los usuarios dentro de ese archivo podrán programar tareas, de tal manera que por defecto solo esta ***/etc/cron.deny*** y donde se indican los usuarios que no pueden crear tareas.

/etc/crontab contiene tanto la configuración global de ***'cron'***, incluyendo las variables ***'SHELL'*** *(por defecto, '/bin/bash')*, ***'PATH'*** *('/sbin:/bin:/usr/sbin:/usr/bin')* y ***'MAILTO'*** *('root')*, así como las definiciones de las tareas programadas *(jobs)* para el usuario ***'root'***.

```
[root@serveripa data]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
#  |    |    |    |    .----  day  of  week  (0  -  6)  (Sunday=0  or  7)  OR
sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

/etc/cron.hourly/; /etc/cron.daily/; / /etc/cron.weekly/; /etc/cron.monthly/; los *scripts* presentes en estos directorios serán oportunamente ejecutados por '**crond**' cada hora, día, semana y mes, respectivamente.

/etc/anacrontab; contiene tanto la configuración global de ***'anacron'***, incluyendo las variables ***'SHELL'*** *(por defecto, '/bin/sh')*, ***'PATH'*** *('/sbin:/bin:/usr/sbin:/usr/bin')* y ***'MAILTO'*** *('root')*, así como las definiciones de las tareas programadas *(jobs)* para el usuario **'root'**.

Las principales diferencias con respecto a 'cron' son:

- Si el sistema se encuentra apagado a la hora programada, 'anacron' pospondrá la ejecución de la tarea hasta que éste vuelva a estar disponible.
- Una tarea de 'anacron' tan sólo puede ejecutarse, a lo sumo, una vez al día.

La definición de una tarea programada se realiza por medio de una entrada en este fichero de configuración y consta de 4

```
# .------------------------------ frecuencia, en días
# | .---------------------------- demora en la ejecución, en minutos
# | | .-------------------------  identificador de la tarea, empleado para los logs
# | | |
# * * * <command>
```

Como editar el cron:

```
crontab -l [ -u <username> ]; para listar las tareas programadas del usuario indicado.
```

```
crontab -e [ -u <username> ]; para editar las tareas programadas del usuario indicado.
```

```
crontab -r [ -u <username> ]; para eliminar la tabla de tareas programadas asociada al usuario indicado.
```

Herramienta para calcular los tiempos de ejecución te las tareas de crontab *https://crontab.guru/.*

- Iniciar y detener los servicios, además de configurarlos para que se inicien automáticamente durante el arranque

```
systemctl status <service>; para obtener información acerca del estado de un servicio.
```

```
systemctl start <service>; para iniciar un servicio.
```

```
systemctl stop <service>; para detener un servicio.
```

```
systemctl restart <service>; para reiniciar un servicio.
```

```
systemctl is-active <service>; para determinar si un servicio ha sido iniciado ('active') o no ('unknown').
```

```
systemctl enable <service>; para activar el inicio automático de un servicio al arranque del sistema.
```

```
systemctl disable <service>; para desactivar el inicio automático de un servicio.
```

```
systemctl is-enabled <service>; para determinar si un servicio se iniciará automáticamente al arranque del sistema ('enabled') o no ('disabled').
```

```
systemctl mask <service>; para desactivar completamente un servicio ('ln -s '/dev/null' '/etc/systemd/system/<service>.service'').
```

```
systemctl unmask <service>; para reactivar nuevamente un servicio.
```

- Configurar los sistemas para que se inicien automáticamente en un destino específico

Los targets en systemd actúan como puntos de sincronización durante el inicio de su sistema. Los archivos de unidades de destino, que terminan con la extensión de archivo ".target", representan los destinos de systemd. El propósito de las unidades de destino es agrupar varias unidades systemd a través de una cadena de dependencias.

- **graphical.target** para iniciar una sesión gráfica.
- **multi-user.target** inicia otros servicios esenciales del sistema como NetworkManager o D-Busy activa otra unidad de destino denominada basic.target.
-

Visualizar las unit targets:

- Ver el target activo por defecto:

```
[root@rhel9-client ~]$  systemctl get-default
multi-user.target

[bonzo@rhel9-client ~]$ ls -l /usr/lib/systemd/system/default.target
lrwxrwxrwx. 1 root root 16 Jul 20 18:48 /usr/lib/systemd/system/default.target ->
graphical.target
```

- Enumerar todas las unit targets cargadas independientemente de su estado:

```
systemctl list-units --type target --all
```

- Enumerar todas las unit de targets actualmente cargadas:

```
[bonzo@rhel9-client ~]$ systemctl list-units --type target
  UNIT                   LOAD   ACTIVE SUB    DESCRIPTION
  basic.target           loaded active active Basic System
  cryptsetup.target      loaded active active Local Encrypted Volumes
  getty.target           loaded active active Login Prompts
  integritysetup.target  loaded active active Local Integrity Protected Volumes
  local-fs-pre.target    loaded active active Preparation for Local File Systems
  multi-user.target      loaded active active Multi-User System
  network-online.target  loaded active active Network is Online
  network-pre.target     loaded active active Preparation for Network
  network.target         loaded active active Network
  rpcbind.target         loaded active active RPC Port Mapper

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.
26 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemc
```

Modificar target por defecto:

Tabla de targets para comando set-default:

basic	cubre el arranque básico.
rescue	atrae el sistema base y genera un shell de rescate.
multi-user	configurar un sistema multiusuario
graphical	configurar una pantalla de inicio de sesión gráfica.
Emergency	inicia un shell de emergencia en la consola principal.
sysinit	extrae los servicios necesarios para la inicialización del sistema.

Para cambiar el default target se hace mediante el comando *"systemctl set-default $unit.target",* para que se haga permanente hay que reiniciar el servidor y se mantendrá el ***target*** seleccionado:

```
[root@rhel9-client ~]# systemctl get-default
multi-user.target

[root@rhel9-client ~]# systemctl set-default graphical.target
Removed /etc/systemd/system/default.target.
Created           symlink           /etc/systemd/system/default.target           →
/usr/lib/systemd/system/graphical.target.

[root@rhel9-client ~]#  systemctl get-default
graphical.target

[root@rhel9-client ~]#  reboot
```

Cambiar el target actual para la sesión en ejecución

Cuando se establece una unidad de destino predeterminada, el destino actual permanece sin cambios hasta el próximo reinicio. Si se desea cambiar la unidad de destino en la sesión actual sin reiniciar, se debe ejecutar el comando "systemctl isolate".

```
systemctl isolate graphical.target
```

- Configurar los clientes de servicios de tiempo

Ntpd / Chrony

```
Rpq –qa|grep chorny
Yum isntall chrony
/etc/ntpd.conf
/etc/chrony.conf
```

VIM Líneas

server servidoripa.example.com iburst

RED DEBE ESTAR CORRECTAMENTE

```
systemctl status chronyd
systemctl status ntpd
systemctl is-enable servicio
systemctl stop chronyd
systemctl start chronyd
chronyc tracking
chronyc sources –v
timedatectl
```

deshabilitar ntp synchronized

```
timedatectl set-ntp false
```

- chrony; paquete requerido para el uso del servicio 'chronyd'.
- chronyd; servicio empleado en la sincronización de la hora local del sistema.
- timedatectl; para mostrar la configuración actual tanto de la hora local del sistema así como de la zona horaria establecida.
- timedatectl list-timezones; para mostrar la lista de las distintas zonas horarias disponibles.
- timedatectl set-timezone <timezone>; para asignar la zona horaria indicada.
- cat /etc/chrony.conf | grep ^server; para determinar la lista de servidores empleados en la sincronización de la hora local del sistema.
- chronyc tracking; para mostrar información acerca de la hora de referencia utilizada.
- chronyc sources -v / chronyc sourcestats -v; para mostrar información acerca del proceso de sincronización.

ntpdate <server>; para sincronizar de manera inmediata la hora local del sistema con el servidor indicado.

- Instalar y actualizar paquetes de software desde Red Hat Network, desde un repositorio remoto o desde el sistema de archivos local

Como actualizar desde un repositorio local, se creó un repositorio local que se encuentra montado en ***/mnt/iso*** y creamos un repositorio local para instalar un sistema de archivos local.

```
cat /etc/yum.repos.d/base.repo
```

```
[InstallMedia]
name=Red Hat Enterprise Linux 7.3
mediaid=1476915898.899142
metadata_expire=-1
baseurl=file:///mnt/iso
enabled=1
gpgcheck=0
cost=500
```

Para activar la suscripción mediante interfaz gráfico:

Aplicactions -> System tools -> Red Hat Subscrition Manager -> Register (meter datos)

Para activar la suscripción por comandos:

Con los datos de tu cuenta RedHat y el siguiente comando se activa la suscripción:

```
subscription-manager register
```

Para listar subscriciones activas/disponibles:

```
subscription-manager list --available
```

Buscar el Pool ID: XXXX, para poder activarlo en el siguiente paso:

```
subscription-manager attach --pool=XXXX
```

Para desactivar todos los repositorios:

```
subscription-manager repos --disable=*
```

Habilitando un repositorio necesario para el entrenamiento

```
subscription-manager repos -enable=rhel-9-server-rpms
```

Ver repositorios disponibles.

```
yum repolist
```

Configurar un repositorio para compartirlo con clientes mediante FTP

Ahora vamos a ver cómo trabajar con este repositorio para compartirlo vía FTP para que la máquina cliente pueda actualizarse. Vamos a descargar este repositorio a nuestra máquina local y así tenerlo disponible y no tener que estar suscribiendo otras máquinas.

Para ello editar el repositorio ***/etc/yum.repos.d/base.repo*** para deshabilitarlo:

```
vim /etc/yum.repos.d/base.repo

enabled=0
```

Limpiar y actualizar los repositorios

```
yum clean all
yum repolist
```

Ahora procederemos a descargar el repositorio para tenerlo disponible para los clientes. Creando un directorio updates para compartirlo con los clientes.

```
mkdir /var/ftp/pub/repos/updates
```

Comando para **descargar** los repositorios en local (No instala los paquetes, los DESCARGA).

```
reposync -l -n -p /var/ftp/pub/repos/updates/ --downloadcomps --download-metadata
```

Verificar que están en el directorio los elementos del repositorio sobre todo si esta ***rhel-9-server-rmps.***

```
cd /var/ftp/pub/repos/updates
ls
rhel-9-server-rmps
```

Generar la BBDD SQlite del repositorio, también creará un directorio en ***/var/ftp/pub/repos/updates/repodata***

```
createrepo /var/ftp/pub/repos/updates/rhel-9-server-rmps/ -g comps.xml
```

Ahora hacer una copia del ***base.repo*** y lo llamarlo ***updates.repo***:

```
cp /var/ftp/pub/repos/base.repo /var/ftp/pub/repos/update.repo
```

Y modificar el fichero ***updates.repo*** de la siguiente manera:

```
[updates]
name = Updates Repository for RHEL 9
baseurl = ftp://servidoripa.example.com/pub/repos/updates/rhel-9-server-rmps
```

```
enabled = 0
gpgcheck = 0
```

Configuración del repositorio en el cliente:

Modificando el fichero */etc/yum.repos.d/os.repo*:

```
[os]
name = Repositorio Base RHEL 9
baseurl = ftp://servidoripa.example.com/pub/repos/updates/rhel-9-server-rmps
enabled = 1
gpgcheck = 0
```

```
yum clean all
yum repolist
```

Borrar fichero de configuración de repositorio.

```
rm /etc/yum.repos.d/os.repo
yum clean all
```

Utilizar un archivo remoto

Copiamos la URL ftp://servidoripa.example.com/pub/repos/base.repo del FTP donde se encuentra la configuración del repositorio que tenemos el en servidor y lo añadimos de la siguiente manera al cliente:

```
yum-config-manager --add-repo ftp://servidoripa.example.com/pub/repos/base.repo
```

```
cat /etc/yum.repos.d/base.repo
```

Ahora hay que habilitarlo y se puede de hacer de dos maneras:

1. Opción: Editar el fichero ***/etc/yum.repos.d/base.repo*** cambiando el parámetro ***"enabled = 0"*** por ***"enabled = 1"***.

2. Opción: Utiliar el siguiente comando:

```
yum-config-manager --enable base
cat /etc/yum.repos.d/base.repo
```

Por último actualizar las listas del repositorio:

```
yum repolist
```

Instalar paquetes

```
yum groups list
yum groups install "NOMBRE DE GRUPO"
yum history
yum history info
yum history info NUMERO
yum remove PAQUETE

yum history undo NUMERO [elimina los paquetes instalados en esa "transaccion"]

yum history info samba

rpm -qf /etc/samba/smb.conf
rpm -qlc "TEXTO SACADO DEL COMANDO DE ARRIBA"

yum provides sepolicy
yum whatprovides */sepolicy
```

DNF

Es una **versión de yum mejorada** (se trata de un ***fork*** de éste). Una de las ventajas del nuevo sistema es la resolución de dependencias, que funciona en la mayoría de ocasiones. Si nos viene a la cabeza la pregunta, ¿qué diferencias existen en su uso? Pues la respuesta es bien sencilla, ninguna. La gran modificación es el sistema de módulos y su manejo. Pero en definitiva los parámetros que funcionaron yum funcionaran don dnf.

- Modificar el cargador de arranque del sistema

Ver los kernel disponibles

```
grep ^menuentry /boot/grub2/grub.cfg
```

```
[root@servidoripa ~]# grep ^menuentry /boot/grub2/grub.cfg
       menuentry 'Red Hat Enterprise Linux Server (3.10.0-514.el7.x86_64) 7.3 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-3.10.0-514.el7.x86_64-advanced-5ccb6e72-d195-444d-91f8-d7ad269cf5a7' {
       menuentry 'Red Hat Enterprise Linux Server (0-rescue-c6c883618253486183c0643f0a786ec1) 7.3 (Maipo)' --class red --class gnu-linux --class gnu --class os --unrestricted $menuentry_id_option 'gnulinux-0-rescue-c6c883618253486183c0643f0a786ec1-advanced-5ccb6e72-d195-444d-91f8-d7ad269cf5a7' {
```

Existen dos por defecto kernel 0 y 1, pero se pueden instalar más e irán aumentando el número consecutivamente. Revisar ***/boot/grub2/grub.cfg*** y modificar ***/etc/default/grub.***

Se debe modificar el ***/etc/default/grub*** , en "GRUB_DEFAULT", se puede seleccionar 0 ó 1 para cambiar en el arranque.

```
[root@servidoripa ~]# cat /etc/default/grub
GRUB_TIMEOUT=5
GRUB_DISTRIBUTOR="$(sed 's, release .*$,,g' /etc/system-release)"
GRUB_DEFAULT=saved
GRUB_DISABLE_SUBMENU=true
GRUB_TERMINAL_OUTPUT="console"
GRUB_CMDLINE_LINUX="crashkernel=auto                    rd.lvm.lv=rhel_servidoripa/root rd.lvm.lv=rhel_servidoripa/swap rhgb quiet"
GRUB_DISABLE_RECOVERY="true"
```

Aplicar configuración nueva y reiniciar el sistema.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
systemctl reboot
```

Gestionar las conexiones de red básicas

- Configurar la resolución de nombre de host

Cambiar hostname tiene que ser FQND:

```
hostnamectl
hostnamectl set-hostname blablabla.example.comandos
systemctl restart systemd-hostnamed
logout
```

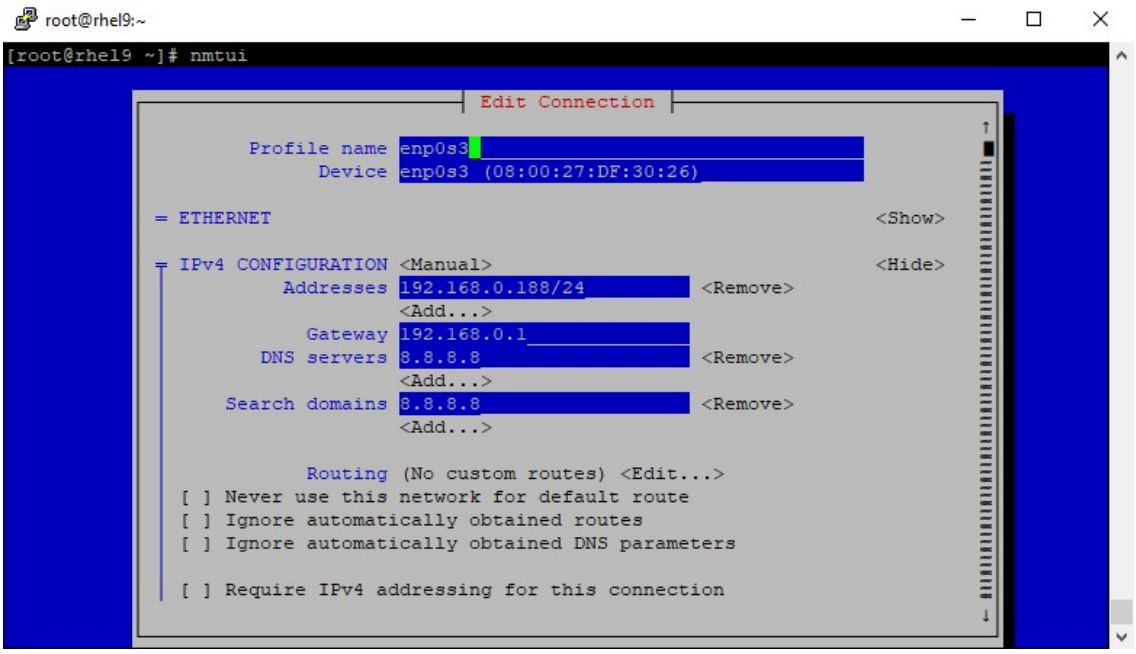
Abrir una nueva sesión.

- Configurar los servicios de red para que se inicien automáticamente durante el arranque

```
systemctl enable Networkmanager
```

- Configurar las direcciones IPv4 e IPv6

Modo – GRAFICAMENTE mediante terminal mntui.



Modo - TERMINAL mediante ficheros de interfaz.

- Para ver configuración de interfaces:

```
[root@rhel9 ~]# ip a show enp0s3
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 08:00:27:df:30:26 brd ff:ff:ff:ff:ff:ff
    inet 192.168.0.188/24 brd 192.168.0.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fedf:3026/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

- Ver tabla de ruteo:

```
[root@rhel9 ~]# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.0.1     0.0.0.0         UG    100    0        0 enp0s3
192.168.0.0     0.0.0.0         255.255.255.0   U     100    0        0 enp0s3
```

- Ver ruta por defecto (Gateway):

```
[root@rhel9 ~]# ip r s
default via 192.168.0.1 dev enp0s3 proto static metric 100
192.168.0.0/24 dev enp0s3 proto kernel scope link src 192.168.0.188 metric 100
```

El fichero ***/etc/resolv.conf*** se encarga de los nombres DNS que tenemos configurados, en principio no se debe modificar a mano, sino con la modificación del servicio indicado en el apartado anterior.

El fichero de configuración ***/etc/sysconfig/network-scripts/ifcfg-[interface]*** contiene la configuración para el interfaz seleccionado y aquí es donde debe modificarse a mano.

Valores para configuración estática:

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETWORK=10.0.1.0
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

Valores para configuración DHCP:

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

Modo - TERMINAL mediante ficheros en RHEL9:

Anteriormente, ***NetworkManager*** almacenaba nuevas configuraciones de red en ***/etc/sysconfig/network-scripts/*** en formato ifcfg. A partir de RHEL 9.0, RHEL almacena nuevas configuraciones de red en ***/etc/NetworkManager/system-connections/*** en un formato de archivo clave. Las conexiones para las que se almacenan las configuraciones en ***/etc/sysconfig/network-scripts/*** en el formato anterior aún funcionan sin interrupciones. Las modificaciones en los perfiles existentes continúan actualizando los archivos más antiguos.

Ejemplo de cómo cambiar la configuración:

```
[root@rhel9-client ~]# ip a show enp0s3|grep -i "inet "
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
    inet 192.168.0.190/24 brd 192.168.0.255 scope global noprefixroute enp0s3
```

```
[root@rhel9-client~]#              cat              /etc/NetworkManager/system-
connections/enp0s3.nmconnection
[connection]
id=enp0s3
uuid=16d6e1dd-523d-3cdb-9cca-01aaed0923d8
type=ethernet
autoconnect-priority=-999
interface-name=enp0s3
timestamp=1667733574

[ethernet]

[ipv4]
address1=192.168.0.189/24,192.168.0.1
dns=8.8.8.8;
dns-search=8.8.8.8;
method=manual

[ipv6]
addr-gen-mode=eui64
method=auto
```

```
[root@rhel9-client ~]# nmcli connection load /etc/NetworkManager/system-
connections/enp0s3.nmconnection
[root@rhel9-client ~]# nmcli connection up /etc/NetworkManager/system-
connections/enp0s3.nmconnection
```

```
[root@rhel9-client ~]# ip a show enp0s3|grep -i "inet "
    inet 192.168.0.189/24 brd 192.168.0.255 scope global noprefixroute enp0s3
```

Modo - TERMINAL mediante comandos nmcli.

Ver nombres de los interfaces conectados

```
nmcli device s ---
```

Borrar un interfaz para poder configurarlo:

```
nmcli delete eth0
```

Configuración manual de interfaces:

```
nmcli connection add type ethernet ifname eth0 con-name eth0 ipv4 method manual ipv4.addresses 192.168.4.123/24 ipv4.gateway 192.168.4.1 ipv4.dns 192.168.4.230 connection.autoconnect yes
```

Modificación de configuración de interfaz:

```
nmcli connection modify eth0 ipv4.addresses 192.168.4.123/24 ipv4.gateway 192.168.4.1 ipv4.dns 192.168.4.230 connection.autoconnect yes ipv4.medthod manual
```

Reiniciar el interfaz seleccionado:

```
nmcli connection down eth0; nmcli connection up echo
```

Verificar la configuración del interfaz

```
ip a
```

Configurar interfaz a DHCP

```
nmcli delete eth0

nmcli connection modify eth0 connection.autoconnect yes ipv4.medthod auto
```

```
nmcli connection down eth0
nmcli connection up echo
```

```
ip a
```

Gestionar usuarios y grupos

- Crear, borrar y modificar cuentas de usuario locales

Crear usuario:

```
useradd -c "Usuario de pRueba"  -u 1234 -G "GRUPO" -s /sbin/nologin user1
```

Ver los usuarios del sistema:

```
getent passwd
```

Cambiar el login de un usuario, pero mantener el viejo:

```
usermod -l NUEVOusuario user1
usermod -m -d /home/test1 test1
```

Borrar usuario:

```
userdel test1
userdel -r bob
```

Al crear un usuario los datos por defecto que se requieren para el usuario se encuentran en el fichero ***/etc/login.defs***:

```
cat /etc/login.defs
```

En el fichero ***/etc/default/useradd*** también incluye datos de donde tiene el home, el skel etc.…

```
cat /etc/default/useradd
```

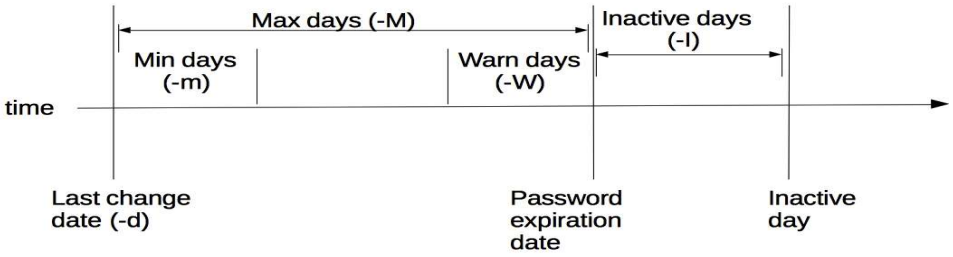
Son archivos ocultos que se generan para cada usuario, como bashrc, profile etc ….

```
root@starwing:~# ls -lrta /etc/skel
total 28
-rw-r--r--   1 root root   675 May 15  2017 .profile
-rw-r--r--   1 root root  3526 May 15  2017 .bashrc
-rw-r--r--   1 root root   220 May 15  2017 .bash_logout
drwxr-xr-x   2 root root  4096 Jul 12  2018 .
drwxr-xr-x 111 root root 12288 Nov 18 03:47 ..
```

- Cambiar contraseñas y ajustar la duración de las contraseñas para las cuentas de usuario locales

Crear un usuario

```
useradd user1
```



Ver perfil de un usuario:

```
root@starwing:~# chage -l user1
                Last password change                                    : Mar 13,
2019
                Password expires                                        : never
                Password inactive                                       : never
                Account expires                                         : never
                Minimum number of days between password change          : 0
                Maximum number of days between password change          : 99999
                Number of days of warning before password expires       : 7
```

Ayuda chage

```
root@starwing:~# chage --help
        Usage: chage [options] LOGIN

        Options:
        -d, --lastday LAST_DAY            set date of last password change to
LAST_DAY
        -E, --expiredate EXPIRE_DATE  set account expiration date to EXPIRE_DATE
        -h, --help                    display this help message and exit
        -I, --inactive INACTIVE           set password inactive after expiration
        to INACTIVE
        -l, --list                    show account aging information
        -m, --mindays MIN_DAYS        set minimum number of days before password
                                      change to MIN_DAYS
        -M, --maxdays MAX_DAYS        set maximim number of days before password
                                      change to MAX_DAYS
        -R, --root CHROOT_DIR         directory to chroot into
        -W, --warndays WARN_DAYS      set expiration warning days to WARN_DAYS
```

Establecer que una contraseña caduque q los 60 días:

```
root@starwing:~# date -d "60 days"
Sun May 12 23:44:44 CEST 2019
```

```
root@starwing:~# date -d "60 days" "+%Y-%-m-%d"
2019-5-12
root@starwing:~#chage -E 2019-5-12 user1
```

```
root@starwing:~# chage -l user1
        Last password change                                    : Mar 13, 2019
        Password expires                                        : never
        Password inactive                                       : never
        Account expires                                         : never
        Minimum number of days between password change          : 0
        Maximum number of days between password change          : 99999
        Number of days of warning before password expires       : 7
```

```
root@starwing:~# chage -E 2019-5-12 user1

root@starwing:~# chage -l user1
        Last password change                                    : Mar 13, 2019
        Password expires                                        : never
        Password inactive                                       : never
        Account expires                                         : May 12, 2019
        Minimum number of days between password change          : 0
        Maximum number of days between password change          : 99999
        Number of days of warning before password expires       : 7
```

```
root@starwing:~# chage -M 35 user1

root@starwing:~# chage -l user1
        Last password change                                    : Mar 13, 2019
        Password expires                                        : Apr 17, 2019
        Password inactive                                       : never
        Account expires                                         : May 12, 2019
        Minimum number of days between password change          : 0
        Maximum number of days between password change          : 35
        Number of days of warning before password expires       : 7
```

Forzar cambio de contraseña en el próximo inicio de sesión:

```
chage -d 0 user1
```

- Crear, borrar y modificar los grupos locales y la pertenencia a grupos

Crear un grupo

```
groupadd nombre1
```

Crear un grupo con nombre2 cuyo Group ID (GID) sea 1234. El parámetro **-G** es para el Grupo Principal y **-g** para Grupo secundario o GID:

```
groupadd nombre2 -g 1234
```

El fichero /etc/group contiene a los grupos del sistema, podemos visualizarlos con:

```
tail /etc/group
```

Editar nombre de grupo

groupmod -n grupo02 nombre2

Eliminar grupo

```
groupdel -R grupo02
```

Pertenencia a Grupos

Como agregar usuarios a un grupo, para ello crear el grupo contabilidad y crear un par de usuarios:

1. Crear grupo contabilidad

```
groupadd contabilidad
```

2. Crear usuarios:

```
for i in alfredo antonio aldo;do useradd $i;done
```

3. Verificar datos de un usuario:

```
id aldo
```

4. Agregar usuarios al grupo contabilidad:

```
for i in alfredo antonio aldo;do usermod -aG contabilidad $i;done
```

Crear un usuario que pertenezca a un grupo específico:

```
useradd jack -G contabilidad
```

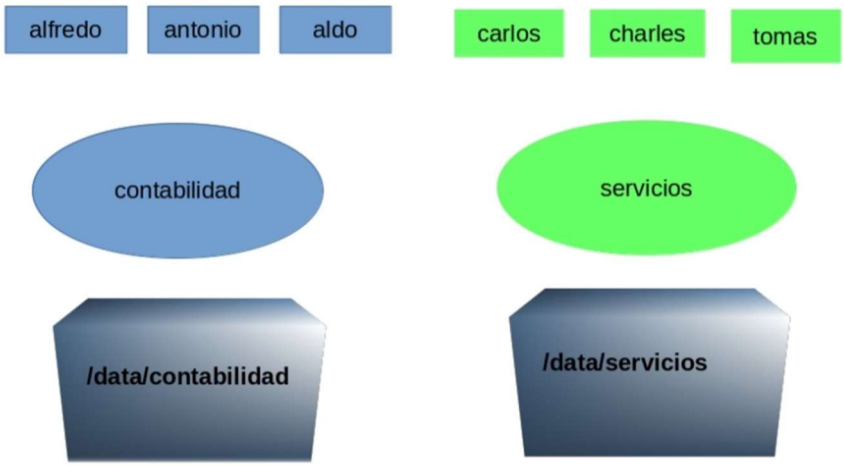
Comandos para visualizar los usuarios que pertenecen a un grupo:

```
groupmens -gcontabilidad -last

getent group contabilidad
```

Eliminar un usuario de un grupo, se puede hacer de dos maneras, pero la segunda opción es la más elegante:

1. Editar el fichero ***/etc/group*** y eliminar el usuario del grupo seleccionado.
2. Utilizar el siguiente comando:

```
gpasswd -d $USER $GROUP
gpasswd -d jack contabilidad
```



Nota: *a la hora de hacer el examen tener en cuenta sí el grupo que te piden es primario –G o secundario –g.*

- Configurar el acceso de superusuario

Para proporcionar privilegios a los usuarios o grupos para ejecutar comandos como root u otro usuario, tenemos el fichero /etc/sudoers. Donde podemos otorgar dichos privilegios.

Visudo

Permite la edición del /etc/sudoers y verifica su correcto estado:

```
[root@rhel9 ~]# visudo
visudo: /etc/sudoers.tmp unchanged
```

```
[root@rhel9 ~]# visudo -c
/etc/sudoers: parsed OK
```

```
[root@rhel9 ~]# grep -i include /etc/sudoers
#includedir /etc/sudoers.d
```

Dar privilegios a usuario para convertirse en root sin conocer la password:

```
[root@rhel9 ~]# echo "bonzo ALL=/usr/bin/su - " > /etc/sudoers.d/bonzo
```

```
[root@rhel9 ~]# sudo -l -U bonzo
Matching Defaults entries for bonzo on rhel9:
    !visiblepw, always_set_home, match_group_by_gid, always_query_group_plugin,
env_reset, env_keep="COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR
    LS_COLORS", env_keep+="MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE",
env_keep+="LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT
    LC_MESSAGES", env_keep+="LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE",
env_keep+="LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET
    XAUTHORITY", secure_path=/sbin\:/bin\:/usr/sbin\:/usr/bin

User bonzo may run the following commands on rhel9:
    (root) /usr/bin/su -
```

Ejemplos:

Permisos a grupos:

```
%work_team ALL=ALL
%work_team ALL=/usr/bin/systemctl restart autofs, /usr/bin/systemctl restart nfs-client
```

Permisos a usuarios:

```
bonzo ALL=/usr/bin/su -
```

Gestionar la seguridad

- Establecer los ajustes de firewall con firewall-cmd o firewalld

```
[root@nodo1 ~]# firewall-cmd --get-default-zone
public
[root@nodo1 ~]#
```

```
[root@nodo1 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
[root@nodo1 ~]# firewall-cmd --list-all --zone=dmz
dmz
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
[root@nodo1 ~]# firewall-cmd --list-all-zones
block
  target: %%REJECT%%
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
```

```
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


dmz
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


drop
  target: DROP
  icmp-block-inversion: no
  interfaces:
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


external
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


internal
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client mdns samba-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


trusted
  target: ACCEPT
  icmp-block-inversion: no
  interfaces:
```

```
  sources:
  services:
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:


work
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

```
[root@nodo1 ~]# ls -lrt /etc/firewalld/
total 8
drwxr-x---. 2 root root   46 Apr 15  2021 zones
drwxr-x---. 2 root root    6 Apr 15  2021 services
-rw-r--r--. 1 root root  272 Apr 15  2021 lockdown-whitelist.xml
drwxr-x---. 2 root root    6 Apr 15  2021 ipsets
drwxr-x---. 2 root root    6 Apr 15  2021 icmptypes
drwxr-x---. 2 root root    6 Apr 15  2021 helpers
-rw-r--r--. 1 root root 2706 Apr 15  2021 firewalld.conf
```

```
[root@nodo1 ~]# ls -lrt /etc/firewalld/zones/
total 8
-rw-r--r--. 1 root root 315 Feb 17 17:40 public.xml.old
-rw-r--r--. 1 root root 315 Feb 17 17:40 public.xml
[root@nodo1 ~]#
```

- Modificar una zona por defecto:

```
[root@nodo1 ~]# firewall-cmd --set-default-zone=work
success
[root@nodo1 ~]# firewall-cmd --get-default-zone
work
[root@nodo1 ~]# firewall-cmd --list-all
work (active)
```

```
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- Deshabilitar ssh para la zona por defecto

```
[root@nodo1 ~]# firewall-cmd --permanent --remove-service=ssh
success
[root@nodo1 ~]# firewall-cmd --list-all
work (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

- Cambiar zona y habilitar de nuevo el ssh

```
[root@nodo1 ~]# firewall-cmd --set-default-zone=public
Success


[root@nodo1 ~]# firewall-cmd --get-default-zone
public
[root@nodo1 ~]#
```

- Reiniciar

```
[root@nodo1 ~]# firewall-cmd --reload
success
[root@nodo1 ~]#
```

- Ver zonas

```
[root@nodo1 ~]# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
[root@nodo1 ~]#
```

- Abrir un Puerto de forma permanente

```
[root@nodo1 ~]# firewall-cmd --add-port=8080/tcp --permanent
Success

[root@nodo1 ~]#  firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@nodo1 ~]# firewall-cmd --reload
Success

[root@nodo1 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client ssh
  ports: 8080/tcp
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Práctica habilitar puertos 80 y 443 para acceso a httpd.

```
[root@nodo1 ~]# systemctl mask nftables
Created symlink from /etc/systemd/system/nftables.service to /dev/null.
[root@nodo1 ~]# systemctl status nftables
● nftables.service
   Loaded: masked (/dev/null; bad)
   Active: inactive (dead)
```

```
[root@nodo1 ~]# yum install httpd

[root@nodo1 ~]# echo "Prueba firewalld" >> /var/www/html/index.html
[root@nodo1 ~]# systemctl enable --now httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to
/usr/lib/systemd/system/httpd.service.
[root@nodo1 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset:
disabled)
   Active: active (running) since Sat 2022-06-18 21:19:03 CEST; 5s ago
     Docs: man:httpd(8)
           man:apachectl(8)
 Main PID: 35149 (httpd)
   Status: "Processing requests..."
   CGroup: /system.slice/httpd.service
           ├─35149 /usr/sbin/httpd -DFOREGROUND
           ├─35150 /usr/sbin/httpd -DFOREGROUND
           ├─35151 /usr/sbin/httpd -DFOREGROUND
           ├─35152 /usr/sbin/httpd -DFOREGROUND
           ├─35153 /usr/sbin/httpd -DFOREGROUND
           └─35154 /usr/sbin/httpd -DFOREGROUND

Jun 18 21:19:03 nodo1.escleiron.es systemd[1]: Starting The Apache HTTP Server...
Jun 18 21:19:03 nodo1.escleiron.es systemd[1]: Started The Apache HTTP Server.

[root@nodo1 ~]# curl -k https://localhost
curl: (7) Failed connect to localhost:443; Connection refused
[root@nodo1 ~]# curl -k http://localhost
curl: (7) Failed connect to localhost:80; Connection refused
[root@nodo1                                                                    ~]#
```

```
[root@nodo1 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
```

```
  services: dhcpv6-client ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@nodo1 ~]# firewall-cmd --add-service=http --permanent
success
[root@nodo1 ~]# firewall-cmd --add-service=https --permanent
Success

[root@nodo1 ~]# firewall-cmd --reload
Success

[root@nodo1 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens33
  sources:
  services: dhcpv6-client http https ssh
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:

[root@nodo1 ~]# curl -k https://localhost
Prueba firewalld
[root@nodo1 ~]# curl -k http://localhost
Prueba firewalld
```

Practica agregar un Puerto al servicio http

Ejecutar este script para poder hacer la práctica:

```
#!/bin/bash
echo ""
echo -e "\033[1mListo para comenzar el Laboratorio\033[0m"
echo -e "\033[1mEste script le rompera su sistema literalmente\033[0m"
echo -e "\033[1mSon unos pocos cambios para que su sistema quede listo\n\033[0m"

for i in {5..1};do echo -n "$i." && sleep 1; done
echo ""
```

```
### Apache ###
yum install -y httpd httpd-manual
systemctl start httpd
systemctl enable httpd
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
echo "Hola esta es una web de pruebas" >> /var/www/html/index.html
rm -rf /etc/httpd/conf.d/welcome.conf
rm -rf /etc/httpd/conf.d/autoindex.conf
systemctl restart httpd

echo "Listen 1009" >> /etc/httpd/conf.d/vhost02.conf
echo "<VirtualHost *:1009>" >> /etc/httpd/conf.d/vhost02.conf
echo "ServerAdmin webmaster@rhel8cliente1.labrhel.com" >> /etc/httpd/conf.d/vhost02.conf
echo "DocumentRoot "/var/www/html"" >> /etc/httpd/conf.d/vhost02.conf
echo "ServerName rhel8cliente1.labrhel.com" >> /etc/httpd/conf.d/vhost02.conf
echo "ServerAlias rhel8cliente1.labrhel.com" >> /etc/httpd/conf.d/vhost02.conf
echo       "ErrorLog       "/var/log/httpd/rhel8cliente1.labrhel.com-error_log""       >>
/etc/httpd/conf.d/vhost02.conf
echo   "CustomLog   "/var/log/httpd/rhel8cliente1.labrhel.com-access_log"   combined"   >>
/etc/httpd/conf.d/vhost02.conf
echo "</VirtualHost>" >> /etc/httpd/conf.d/vhost02.conf
echo "<Directory "/var/www/html">" >> /etc/httpd/conf.d/vhost02.conf
echo "AllowOverride None" >> /etc/httpd/conf.d/vhost02.conf
echo "Require all granted" >> /etc/httpd/conf.d/vhost02.conf
echo "</Directory>" >> /etc/httpd/conf.d/vhost02.conf

systemctl restart httpd
```

Troubleshooting

```
[root@nodo1 ~]# sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
--------------------------------------------------------------------------------

SELinux is preventing /usr/sbin/httpd from name_bind access on the tcp_socket port 1009.

*****  Plugin bind_ports (99.5 confidence) suggests   ************************

If you want to allow /usr/sbin/httpd to bind to network port 1009
Then you need to modify the port type.
Do
# semanage port -a -t PORT_TYPE -p tcp 1009
    where   PORT_TYPE   is   one   of   the   following:   http_cache_port_t,   http_port_t,
jboss_management_port_t, jboss_messaging_port_t, ntop_port_t, puppet_port_t.
```

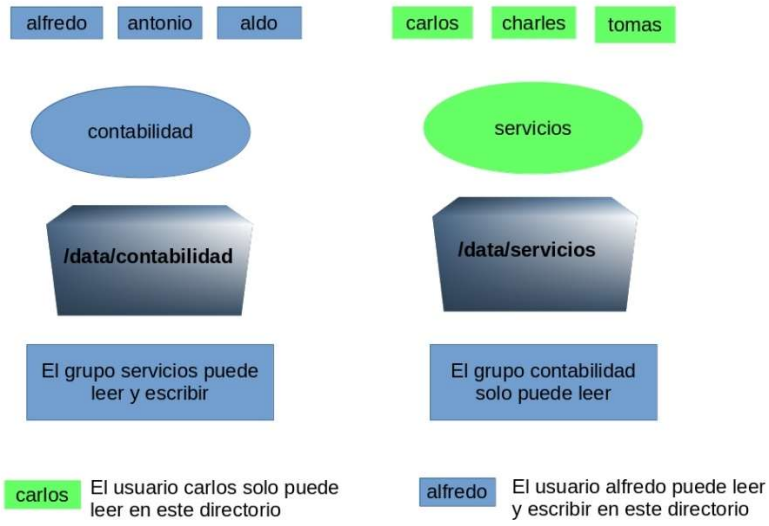
Detectamos que hay un problema porque el puerto asignado a http es el 1009 y este no esta asociado al SELinux.

```
[root@nodo1 ~]# systemctl status httpd|grep -i active
   Active: failed (Result: exit-code) since Sat 2022-06-18 21:50:50 CEST; 3min 6s ago

[root@nodo1 ~]# semanage  port -l|grep -i http
http_cache_port_t              tcp      8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989
[root@nodo1                                                                    ~]#

[root@nodo1 ~]# man -k semanage|grep -i port
semanage-export (8)  - SELinux Policy Management import tool
semanage-ibendport (8) - SELinux Policy Management ibendport mapping tool
semanage-import (8)  - SELinux Policy Management import tool
semanage-port (8)    - SELinux Policy Management port mapping tool
[root@nodo1                                                                    ~]#



[root@nodo1 ~]# man semanage-port|tail -20

EXAMPLE
       List all port definitions
       # semanage port -l
       Allow Apache to listen on tcp port 81
       # semanage port -a -t http_port_t -p tcp 81
       Allow sshd to listen on tcp port 8991
       # semanage port -a -t ssh_port_t -p tcp 8991


[root@nodo1 ~]# semanage port -a -t http_port_t -p tcp 1009

[root@nodo1 ~]# semanage  port -l|grep -i http|grep -i 1009
http_port_t                    tcp      1009, 80, 81, 443, 488, 8008, 8009, 8443, 9000
[root@nodo1 ~]#

[root@nodo1 ~]# curl -k localhost:1009
curl: (7) Failed connect to localhost:1009; Connection refused

[root@nodo1 ~]# systemctl restart httpd

[root@nodo1 ~]# curl -k localhost
Hola esta es una web de pruebas

[root@nodo2 ~]# curl -k 192.168.0.20:1009
```

```
curl: (7) Failed connect to 192.168.0.20:1009; No route to host

root@nodo1 ~]# firewall-cmd --add-port=1009/tcp --permanen
success
[root@nodo1 ~]# firewall-cmd --reload
success

[root@nodo2 ~]# curl -k 192.168.0.20:1009
Prueba firewalld
Hola esta es una web de pruebas
```

- Crear y utilizar las listas de control de accesos a archivos

Crear y administrar listar de control de acceso ACL.




```
r - leer
w- lectura
x- ejecucion en un directorio es que puede hacer CD y en un archivo es que sea
ejecutable.

Esto lo controla el UMASK =0022
directorio 755
fichero 644
setfacl -Rm g:servicios:rwX /data/contabidad
       X= permisos de ejecucion en directorio pero no ejecucion en archivo (cd ls)
       -R=recursivamente
       ls -als
```

Ejercicio 1:

El grupo servicios va a poder leer y escribir dentro de contabilidad, pero el usuario carlos solo puede leer dentro del directorio.

```
[root@serveripa data]# getfacl contabilidad/
# file: contabilidad/
# owner: 777
# group: contabilidad
# flags: -s-
user::rwx
group::rwx
other::---
```

```
[root@serveripa data]# ls -lrt
total 0
drwxrws--T. 2 root servicios     6 Aug 11 01:37 servicios
drwxrws---. 2  777 contabilidad 28 Aug 11 02:55 contabilidad
```

```
[root@serveripa data]# setfacl -Rm g:servicios:rwX /data/contabilidad/
[root@serveripa data]# setfacl -m d:g:servicios:rwx /data/contabilidad/
```

```
[root@serveripa data]# ls -lrt
total 0
drwxrws--T. 2 root servicios     6 Aug 11 01:37 servicios
drwxrws---+ 2  777 contabilidad 28 Aug 11 02:55 contabilidad
```

```
[root@serveripa data]#  getfacl contabilidad/
# file: contabilidad/
# owner: 777
# group: contabilidad
# flags: -s-
user::rwx
group::rwx
group:servicios:rwx
mask::rwx
other::---
default:user::rwx
default:group::rwx
default:group:servicios:rwx
default:mask::rwx
default:other::---
```

```
[root@serveripa data]# setfacl -Rm u:carlos:rX /data/contabilidad/
[root@serveripa data]# setfacl -m d:u:carlos:rx /data/contabilidad/
```

```
[root@serveripa data]# getfacl contabilidad/
# file: contabilidad/
# owner: 777
# group: contabilidad
# flags: -s-
user::rwx
user:carlos:r-x
group::rwx
group:servicios:rwx
mask::rwx
other::---
default:user::rwx
```

```
default:user:carlos:r-x
default:group::rwx
default:group:servicios:rwx
default:mask::rwx
default:other::---
```

Ejercicio 2:

El grupo contabilidad va a poder solamente leer dentro de servicios, pero el usuario alfredo puede leer y escribir dentro del directorio.

```
[root@serveripa data]# setfacl -Rm g:contabilidad:rwX /data/servicios/
[root@serveripa data]# setfacl -m d:g:contabilidad:rw /data/servicios/
```

```
[root@serveripa data]# setfacl -Rm u:alfredo:rwX /data/servicios
[root@serveripa data]# setfacl -m d:u:alfredo:rwx /data/servicios/
```

```
[root@serveripa data]# getfacl servicios/
# file: servicios/
# owner: root
# group: servicios
# flags: -st
user::rwx
user:alfredo:rwx
group::rwx
group:contabilidad:rwx
mask::rwx
other::---
default:user::rwx
default:user:alfredo:rwx
default:group::rwx
default:group:contabilidad:rwx
default:mask::rwx
default:other::---
```

Ejercicio 3: IMPORTANTISIMO

En este ejercicio vamos a dar los mismos permisos ACL del directorio servicios al directorio New_servicios:

1. Asignar los mismos permisos que tiene el directorio servicios:

```
[root@serveripa data]# chmod 3770 New_servicios
[root@serveripa data]# chown root:servicios New_servicios/
```

```
[root@serveripa data]# ls -lrt|grep -v contabilida
total 0
drwxrws--T+ 2 root servicios     6 Aug 11 01:37 servicios
drwxrws--T. 2 root servicios     6 Aug 11 04:00 New_servicios
```

2. Podemos verificar los permisos ACL antes de ejecutar ninguna modificación en el directorio servicios y New_Servicios

```
[root@serveripa data]# getfacl servicios
# file: servicios
# owner: root
# group: servicios
# flags: -st
user::rwx
user:alfredo:rwx
group::rwx
group:contabilidad:rwx
mask::rwx
other::---
default:user::rwx
default:user:alfredo:rwx
default:group::rwx
default:group:contabilidad:rwx
default:mask::rwx
default:other::---
```

```
[root@serveripa data]# getfacl New_servicios/
# file: New_servicios/
# owner: root
# group: servicios
# flags: -st
user::rwx
group::rwx
other::---
```

3. Aplicamos los cambios:

```
[root@serveripa data]# getfacl servicios |setfacl -R --set-file=- New_servicios
```

```
[root@serveripa data]# getfacl New_servicios/
# file: New_servicios/
# owner: root
# group: servicios
# flags: -st
user::rwx
user:alfredo:rwx
group::rwx
group:contabilidad:rwx
mask::rwx
other::---
default:user::rwx
default:user:alfredo:rwx
```

```
default:group::rwx
default:group:contabilidad:rwx
default:mask::rwx
default:other::---
```

Practicando ACL

Partiendo de la configuracion que ha realizado en el video anterior:

- Cree un directorio nuevo con el nombre que usted prefiera en **/var/tmp/**
- A continuacion, aplique las mismas reglas ACL que aplico en el directorio **/data/contabilidad** pero **NO** de la forma que lo realizo en el ejercicio anterior. En esta ocacion utilice las opciones de **backup** y **restore**. Como ayuda puede ver en "**man setfacl**"

En el man tener en cuenta *--restore=file* y *--set-file*. Seria lo mismo que en la practica 3, por otro lado, el restore solo es valido para el mismo directorio ya que si faltan ficheros que estén definidos en el ACL no se ejecutará.

- Hacer un backup de los ACL de un directorio:

```
[root@serveripa]# getfacl -R /data/contabilidad/ >> /root/contabilidadACL
```

- Hacer restauración de un backup solo funciona si en el nuevo directorio hay una copia de todos los ficheros o si es el mismo directorio sobre el que se hizo el backup:

```
 [root@serveripa ojetecalor]# setfacl --restore=/root/contabilidadACL
/var/tmp/ojetecalor/
setfacl: data/contabilidad/: No such file or directory
setfacl: data/contabilidad//escris-root.sh: No such file or directory
setfacl: data/contabilidad//contabilidad_acl: No such file or directory
setfacl: data/contabilidad/: No such file or directory
setfacl: data/contabilidad//escris-root.sh: No such file or directory
setfacl: data/contabilidad//contabilidad_acl: No such file or directory
setfacl: data/contabilidad//escris-root.sh: No such file or directory
setfacl: data/contabilidad//contabilidad_acl: No such file or directory
```

- Copiar permisos de un dir con ACL a otro:

```
[root@serveripa ojetecalor]# setfacl --set-file=/root/contabilidadACL
/var/tmp/ojetecalor/
```

```
[root@serveripa ojetecalor]# getfacl .
# file: .
# owner: root
# group: root
```

```
user::rw-
user:carlos:r-x                    #effective:r--
group::rwx                  #effective:rw-
group:servicios:rwx         #effective:rw-
mask::rw-
other::---
default:user::rwx
default:user:carlos:r-x
default:group::rwx
default:group:servicios:rwx
default:mask::rwx
default:other::---
```

- Configurar la autenticación basada en claves para SSH

Crear ssh-key de un usuario para poder conectase a otra maquina sin contraseña:

```
ORIGEN: nodo1
DESTINO: nodo2
```

Generamos la clave en el nodo1:

```
 [root@nodo1 ~]# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:mroyCznTMdeFWq2llwH70dI+1WDxr6RcluacNWhAGZI root@nodo1.escleiron.es
The key's randomart image is:
+---[RSA 2048]----+
|      .  ..o*.   |
|       = Eoo +   |
|      + B o.. o  |
|     + * * .. .o |
|  o o + S o  o*.o|
| o +   +   o.O +.|
|= .   o     o =  |
| +o  .           |
|  .+o.           |
+----[SHA256]-----+
[root@nodo1 ~]#
```

Verificar que la clave se creó correctamente:

```
[root@nodo1 ~]# ls -lrt ~/.ssh/id_rsa.pub
-rw-r--r--. 1 root root 405 Jun 18 22:04 /root/.ssh/id_rsa.pub
```

```
[root@nodo1 ~]#
```

Propagar la clave y confirmar el login sin clave:

```
[root@nodo1 ~]# ssh-copy-id root@nodo2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/root/.ssh/id_rsa.pub"
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any
that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now
it is to install the new keys
root@nodo2's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'root@nodo2'"
and check to make sure that only the key(s) you wanted were added.

[root@nodo1 ~]# ssh  root@nodo2
Last login: Sat Jun 18 15:31:13 2022 from 192.168.0.20
[root@nodo2 ~]# uptime
 16:07:27 up 36 min,  3 users,  load average: 0.00, 0.01, 0.04
[root@nodo2 ~]#



[root@nodo2 ~]# journalctl -f -u sshd
-- Logs begin at Sat 2022-06-18 15:30:32 EDT. --
Jun 18 15:30:36 nodo2.escleiron.es systemd[1]: Starting OpenSSH server daemon...
Jun 18 15:30:36 nodo2.escleiron.es sshd[1141]: Server listening on 0.0.0.0 port 22.
Jun 18 15:30:36 nodo2.escleiron.es sshd[1141]: Server listening on :: port 22.
Jun 18 15:30:36 nodo2.escleiron.es systemd[1]: Started OpenSSH server daemon.
Jun  18  15:31:13  nodo2.escleiron.es  sshd[8311]:  Accepted  password  for  root  from
192.168.0.20 port 42216 ssh2
Jun  18  16:04:17  nodo2.escleiron.es  sshd[8400]:  Connection  closed  by  192.168.0.20  port
42264 [preauth]
Jun  18  16:07:10  nodo2.escleiron.es  sshd[8403]:  Connection  closed  by  192.168.0.20  port
42266 [preauth]
Jun  18  16:07:10  nodo2.escleiron.es  sshd[8405]:  Connection  closed  by  192.168.0.20  port
42268 [preauth]
Jun  18  16:07:15  nodo2.escleiron.es  sshd[8407]:  Accepted  password  for  root  from
192.168.0.20 port 42270 ssh2
Jun  18  16:07:24  nodo2.escleiron.es  sshd[8418]:  Accepted  publickey  for  root  from
192.168.0.20 port 42272 ssh2: RSA SHA256:mroyCznTMdeFWq2llwH70dI+1WDxr6RcluacNWhAGZI
```

- Establecer el modo de enforcing y el modo permissive para SELinux

SELinux puede estar habilitado o deshabilitado. Cuando está habilitado, SELinux tiene dos modos: enforcing y permissive. Los comandos `getenforce` o `sestatus` muestran en qué modo se está ejecutando SELinux. El comando `getenforce` devuelve `Enforcing`, `Permissive`, o `Disabled`.

Cuando SELinux se ejecuta en modo permisivo, la política de SELinux no se aplica. El sistema permanece operativo y SELinux no deniega ninguna operación, sino que sólo registra los mensajes de CVA, que pueden utilizarse para la resolución de problemas, la depuración y la mejora de la política de SELinux. Cada CVA se registra sólo una vez en este caso.

```
$ sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      31
```

Requisitos previos

- Los paquetes selinux-policy-targeted, libselinux-utils, y policycoreutils están instalados en su sistema.
- Los parámetros del núcleo selinux=0 o enforcing=0 no se utilizan.

Procedimientos

Cambio sin reinicio:

```
[root@nodo1 www]# setenforce
usage:  setenforce [ Enforcing | Permissive | 1 | 0 ]
[root@nodo1 www]# getenforce
Enforcing
[root@nodo1 www]# setenforce 0
[root@nodo1 www]# getenforce
Permissive
[root@nodo1 www]# setenforce 1
[root@nodo1 www]# getenforce
Enforcing
[root@nodo1 www]#
```

Cambio con reinicio:

1. editar el archivo /etc/selinux/config :

```
# vi /etc/selinux/config
```

2. Seleccionamos el modo que queramos establecer de en el campo SELINUX, estableciendo enforcing, Permissive o Disabled:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#       enforcing - SELinux security policy is enforced.
#       permissive - SELinux prints warnings instead of enforcing.
#       disabled - No SELinux policy is loaded.
SELINUX=permissive
# SELINUXTYPE= can take one of these two values:
#       targeted - Targeted processes are protected,
#       mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

3. Reinicia el sistema para asignar el modo establecido:

```
# reboot
```

Comprobación de los cambios

Ejecutar getenforce para verificar el modo activado:

```
$ getenforce
Permissive
```

Tambien se puede añadir como parámetro del kernel enforcing=0 o seelinux=0 en /etc/grub.conf. Aunque Red Hat no recomienda usar seelinux=0 , sino utilizar modo permissive.

```
# cat /etc/grub.conf
        root (hd0,0)
        kernel /vmlinuz-2.6.32-279.el6.x86_64 root=/dev/md3 selinux=0
        initrd /initramfs-2.6.32-279.el6.x86_64.img
```

- Enumerar e identificar el contexto del proceso y el archivo de SELinux

Un contexto en SELinux se define como la información adicional sobre un proceso o archivo con el que este mecanismo de seguridad es capaz de tomar decisiones de control de acceso.

Esta información adicional contiene las siguientes cuatro entidades:

- **Usuario de SELinux:** define la identidad del usuario que accede, posee, modifica o elimina un proceso o archivo en sistemas operativos basados en Linux.

- **Rol:** Basado en esta entidad, a un usuario se le permite o deniega el acceso a un determinado objeto en SELinux. Los derechos de acceso asociados con el rol particular de un usuario se asignan automáticamente a ese usuario.

- **Tipo:** esta entidad se utiliza para definir tipos de archivos y dominios de procesos en SELinux. Al usar esta entidad, se otorga acceso si y solo si una regla en la política de control de acceso de SELinux está presente para ese tipo en particular, y también la regla está ahí para otorgar acceso y no viceversa al revés.

- **Nivel:** esta entidad representa la seguridad multinivel (MLS) y la seguridad multicategoría (MCS). Los niveles de seguridad se definen mediante términos como alto, bajo, etc.

En resumen, un contexto SELinux es una combinación de estos cuatro atributos. Con la ayuda de estos cuatro atributos, SELinux otorga o niega el acceso a un archivo o proceso a un usuario.

Métodos para enumerar contextos de SELinux

- Usando el comando "semanage": Para enumerar los contextos SELinux para todos los archivos y procesos en su sistema.

```
[root@nodo1 ~]# semanage fcontext -l|head -5
SELinux fcontext                                   type               Context
/.*             all files          system_u:object_r:default_t:s0
/[^/]+          regular file       system_u:object_r:etc_runtime_t:s0
/a?quota\.(user|group) regular file       system_u:object_r:quota_db_t:s0
[root@nodo1 ~]#
```

- Uso del comando "ls": Para obtener todos los contextos de archivos SELinux.

```
[root@nodo1 ~]# ls -LZ /web1
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index.html
[root@nodo1 ~]#
```

- Uso del comando "ps": Para obtener los contextos SELinux de los procesos en ejecución.

```
[root@nodo1 ~]# ps axZ|head -2
LABEL                             PID TTY      STAT   TIME COMMAND
```

```
system_u:system_r:init_t:s0                    1  ?               Ss          0:04
/usr/lib/systemd/systemd --switched-root --system --deserialize 22
[root@nodo1 ~]#
```

- Uso del comando "id": Para obtener el contexto SELinux asociado a un usuario.

```
[root@nodo1 ~]# id -Z
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
[root@nodo1 ~]#
```

- Restaurar los contextos de archivos predeterminados

Si SELinux está en modo enforcing, los ficheros deben pertenecer al contexto adecuado de cada servicio. De no ser así, esos ficheros no podrán ser utilizados por el servicio correspondiente.

- Para poder identificar el contexto de un archivo se utiliza el parámetro "Z" con el "ls":

```
[root@nodo1 html]# ls -lZ index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index.html
```

- Para ver los contextos disponibles ejecutar "semanage fcontext -l"

```
[root@nodo1 html]# semanage fcontext -l|grep nfsd_exec_t
/usr/sbin/rpc\.nfsd    regular file       system_u:object_r:nfsd_exec_t:s0
/usr/sbin/rpc\.mountd  regular file       system_u:object_r:nfsd_exec_t:s0
```

- Para modificar el contexto SElinux de un fichero se usa "chcon".

```
[root@nodo1 html]# chcon -t nfsd_exec_t index.html
[root@nodo1 html]# ls -lZ index.html
-rw-r--r--. root root unconfined_u:object_r:nfsd_exec_t:s0 index.html
```

Ejemplos prácticos

En este caso tenemos un servidor httpd, en el cual vamos a modificar el contexto del fichero index.html, observaremos que al cambiar el contexto el httpd ya no puede usar dicho fichero:

```
[root@nodo1 html]# pwd
/var/www/html
```

```
[root@nodo1 html]# curl http://192.168.0.20/
Hola esta es una web de pruebas
```

```
[root@nodo1 html]# ls -lZ index.html
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index.html
```

```
[root@nodo1 html]# semanage fcontext -l|grep nfsd_exec_t
/usr/sbin/rpc\.nfsd        regular file       system_u:object_r:nfsd_exec_t:s0
/usr/sbin/rpc\.mountd      regular file       system_u:object_r:nfsd_exec_t:s0
```

```
[root@nodo1 html]# chcon -t nfsd_exec_t index.html
[root@nodo1 html]# ls -lZ index.html
-rw-r--r--. root root unconfined_u:object_r:nfsd_exec_t:s0 index.html
```

```
[root@nodo1 html]# systemctl restart httpd.service
[root@nodo1 html]# curl http://192.168.0.20/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /
on this server.</p>
</body></html>
```

```
[root@nodo1 html]# cd /var/www/html
```

```
root@nodo1 www]# ls -ldZ html/ html/index.html
drwxr-xr-x. root root system_u:object_r:httpd_sys_content_t:s0 html/
-rw-r--r--. root root unconfined_u:object_r:nfsd_exec_t:s0 html/index.html
```

```
[root@nodo1 www]# restorecon -Rvvv html/
restorecon            reset            /var/www/html/index.html            context
unconfined_u:object_r:nfsd_exec_t:s0-
>unconfined_u:object_r:httpd_sys_content_t:s0
```

```
[root@nodo1 www]# ls -ldZ html/index.html
-rw-r--r--.      root      root      unconfined_u:object_r:httpd_sys_content_t:s0
html/index.html
```

```
[root@nodo1 www]# systemctl restart httpd.service
```

```
[root@nodo1 www]# curl http://192.168.0.20/
Hola esta es una web de pruebas
```

- Utilizar una configuración booleana para modificar los ajustes de SELinux del sistema

Los booleanos permiten cambiar partes de las políticas SELinux en caliente, funcionan como "interruptores" que permiten o no ciertas políticas para fomentar la seguridad.

Para listar todas las opciones booleanas de SELinux se usa el comando "semanage" o "getsebool":

```
[root@nodo1 www]# semanage boolean -l|head -5
SELinux boolean                State  Default Description

privoxy_connect_any            (on   ,   on)  Allow privoxy to connect any
smartmon_3ware                 (off  ,  off)  Allow smartmon to 3ware
mpd_enable_homedirs            (off  ,  off)  Allow mpd to enable homedirs
IOError: [Errno 32] Broken pipe
```

```
[root@nodo1 www]# getsebool -a|head
abrt_anon_write --> off
abrt_handle_event --> off
abrt_upload_watch_anon_write --> on
```

Para cambiar el estado de un booleano ejecutamos, "setsebool nombre_del_booleano on/off". Este cambio de estado no es persistente, para hacerlo persistente añadiremos el parametro "-P":

```
[root@nodo1 www]# setsebool httpd_enable_homedirs on

[root@nodo1 www]# semanage boolean -l|grep httpd|grep -i home
httpd_enable_homedirs          (on   ,  off)  Allow httpd to enable homedirs
```

```
[root@nodo1 www]# setsebool -P httpd_enable_homedirs on
```

```
[root@nodo1 www]# semanage boolean -l|grep httpd|grep -i home
httpd_enable_homedirs          (on   ,   on)  Allow httpd to enable homedirs
[root@nodo1 www]#
```

Ejemplo:

Vamos a habilitar que las páginas web se puedan alojar en los directorios $HOME de cada usuario. Para poder hacerlo hay que activar el booleano "httpd_enable_homedirs" y hacerlo permanente, para ello podemos seguir los siguientes pasos:

```
[root@nodo1 www]# getsebool -a|grep -i httpd|grep -i homedirs
httpd_enable_homedirs --> off
```

```
[root@nodo1 www]# semanage boolean -l|grep httpd|grep -i homedir
httpd_enable_homedirs          (off  ,  off)  Allow httpd to enable homedirs
```

```
[root@nodo1 httpd]# systemctl restart httpd.service
[root@nodo1 httpd]# curl http://192.168.0.20/~bonzo/
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /~bonzo/
on this server.</p>
</body></html>
```

```
[root@nodo1 www]# cat /etc/httpd/conf.d/userdir.conf |grep -i UserDir|grep -v "#"
<IfModule mod_userdir.c>
    UserDir disabled
[root@nodo1 www]#

[root@nodo1 www]# vi /etc/httpd/conf.d/userdir.conf
[root@nodo1 www]#

[root@nodo1 www]# cat /etc/httpd/conf.d/userdir.conf |grep -i UserDir|grep -v "#"
<IfModule mod_userdir.c>
    UserDir public_html
```

```
[bonzo@nodo1 ~]$ pwd
/home/bonzo
[root@nodo1 ~]# chmod 711 /home/bonzo/
[bonzo@nodo1 ~]$ mkdir public_html
[bonzo@nodo1 ~]$ echo "Esto es una web homedir" > public_html/index.html
[bonzo@nodo1 ~]$ cat public_html/index.html
Esto es una web homedir
```

```
[root@nodo1 www]# setsebool -P httpd_enable_homedirs on
[root@nodo1 www]# semanage boolean -l|grep httpd|grep -i home
httpd_enable_homedirs          (on   ,   on)  Allow httpd to enable homedirs
```

```
[root@nodo1 httpd]# systemctl restart httpd.service
[root@nodo1 httpd]# curl http://192.168.0.20/~bonzo/
Esto es una web homedir
[root@nodo1 httpd]#
```

- Diagnosticar y abordar los incumplimientos diarios de las políticas de SELinux

Para analizar cualquier problema con SELinux, lo mejor siempre será verificar si desactivando el modo Enforcing, el problema se soluciona. Si está premisa se cumple lo mejor es analizar logs y verificar los contextos asignados a los ficheros afectados.

Comandos importantes:

```
getenforce / setenforce
semanage fcontext -l
semanage fcontext -a -t $contexto "/$path(/.*)?"
sealert -a /var/log/audit/audit.log
chcon
ls -lrZ
```

Práctica de análisis:

Ejecutar este script para generar los problemas en el servidor, con enforcing enabled:

```
#!/bin/bash
echo ""
echo -e "\033[1mListo para comenzar el Laboratorio\033[0m"
echo -e "\033[1mEste script le rompera su sistema literalmente\033[0m"
echo -e "\033[1mSon unos pocos cambios para que su sistema quede listo\n\033[0m"

for i in {5..1};do echo -n "$i." && sleep 1; done
echo ""

### Apache ###
yum install -y httpd httpd-manual
systemctl start httpd
systemctl enable httpd
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
mkdir /web1
echo "Hola esta es una web de pruebas para la practica de seelinux" >>
/web1/index.html
rm -rf /etc/httpd/conf.d/welcome.conf
rm -rf /etc/httpd/conf.d/autoindex.conf
systemctl restart httpd

echo "<VirtualHost *:80>" >> /etc/httpd/conf.d/vhost01.conf
echo "ServerAdmin webmaster@rhel8cliente1.labrhel.com" >>
/etc/httpd/conf.d/vhost01.conf
echo "DocumentRoot "/web1"" >> /etc/httpd/conf.d/vhost01.conf
echo "ServerName rhel8cliente1.labrhel.com" >> /etc/httpd/conf.d/vhost01.conf
echo "ServerAlias rhel8cliente1.labrhel.com" >> /etc/httpd/conf.d/vhost01.conf
echo "ErrorLog "/var/log/httpd/rhel8cliente1.labrhel.com-error_log"" >>
/etc/httpd/conf.d/vhost01.conf
```

```
echo "CustomLog "/var/log/httpd/rhel8cliente1.labrhel.com-access_log" combined" >>
/etc/httpd/conf.d/vhost01.conf
echo "</VirtualHost>" >> /etc/httpd/conf.d/vhost01.conf
echo "<Directory "/web1">" >> /etc/httpd/conf.d/vhost01.conf
echo "AllowOverride None" >> /etc/httpd/conf.d/vhost01.conf
echo "Require all granted" >> /etc/httpd/conf.d/vhost01.conf
echo "</Directory>" >> /etc/httpd/conf.d/vhost01.conf

systemctl restart httpd
```

Análisis:

La web configurada en el servidor no funciona, el primer paso seria comprobar si cambiando el modo de SELinux Permissive para descartar que no sea este el problema:

```
[root@nodo1 ~]# getenforce
Enforcing

[root@nodo1 ~]# curl localhost
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /
on this server.</p>
</body></html>

[root@nodo1 ~]# setenforce 0
[root@nodo1 ~]# getenforce
Permissive

[root@nodo1 ~]# curl localhost
Hola esta es una web de para la practica de seelinux
[root@nodo1 ~]#
```

Sabiendo que el problema proviene de la configuración del SELinux, tocaría analizar los errores para poder arreglar el problema y mantener el modo Enforcing:

```
[root@nodo1 ~]# getenforce
Enforcing

[root@nodo1 ~]# sealert -a /var/log/audit/audit.log
  0% done
found 0 alerts in /var/log/audit/audit.log
[root@nodo1 ~]# curl localhost
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>403 Forbidden</title>
```

```
</head><body>
<h1>Forbidden</h1>
<p>You don't have permission to access /
on this server.</p>
</body></html>

root@nodo1 ~]# grep -i httpd /var/log/messages|tail -2
Oct 23 11:57:28 nodo1 setroubleshoot: SELinux is preventing httpd from getattr access on
the file /web1/index.html. For complete SELinux messages run: sealert -l 5bae00bf-2cb2-
4827-bc19-01096a671079
Oct 23 11:57:28 nodo1 python: SELinux is preventing httpd from getattr access on the
file /web1/index.html.#012#012*****   Plugin catchall_labels (83.8 confidence) suggests
******************

[root@nodo1 ~]# sealert -a /var/log/audit/audit.log
100% done
found 1 alerts in /var/log/audit/audit.log
--------------------------------------------------------------------------------

SELinux is preventing /usr/sbin/httpd from getattr access on the file /web1/index.html.

*****  Plugin catchall_labels (83.8 confidence) suggests   ******************

If you want to allow httpd to have getattr access on the index.html file
Then you need to change the label on /web1/index.html
Do
# semanage fcontext -a -t FILE_TYPE '/web1/index.html'
where FILE_TYPE is one of the following: XXXXXXXXXXXX
Then execute:
restorecon -v '/web1/index.html'


*****  Plugin catchall (17.1 confidence) suggests   **************************

If you believe that httpd should be allowed getattr access on the index.html file by
default.
Then you should report this as a bug.
You can generate a local policy module to allow this access.
Do
allow this access for now by executing:
# ausearch -c 'httpd' --raw | audit2allow -M my-httpd
# semodule -i my-httpd.pp


Additional Information:
Source Context                system_u:system_r:httpd_t:s0
Target Context                unconfined_u:object_r:default_t:s0
Target Objects                /web1/index.html [ file ]
Source                        httpd
Source Path                   /usr/sbin/httpd
Port                          <Unknown>
```

```
Host                          <Unknown>
Source RPM Packages           httpd-2.4.6-97.el7_9.5.x86_64
Target RPM Packages
Policy RPM                    selinux-policy-3.13.1-268.el7_9.2.noarch
Selinux Enabled               True
Policy Type                   targeted
Enforcing Mode                Enforcing
Host Name                     nodo1.escleiron.es
Platform                      Linux nodo1.escleiron.es
                              3.10.0-1160.53.1.el7.x86_64 #1 SMP Thu Dec 16
                              10:19:28 UTC 2021 x86_64 x86_64
Alert Count                   1
First Seen                    2022-10-23 11:57:12 CEST
Last Seen                     2022-10-23 11:57:12 CEST
Local ID                      58d7bba0-6417-4892-b351-29e33c51ac0c

Raw Audit Messages
type=AVC msg=audit(1666519032.319:1390): avc:  denied  { getattr } for  pid=46649
comm="httpd"          path="/web1/index.html"          dev="dm-0"          ino=16831185
scontext=system_u:system_r:httpd_t:s0        tcontext=unconfined_u:object_r:default_t:s0
tclass=file permissive=0


type=SYSCALL  msg=audit(1666519032.319:1390):  arch=x86_64  syscall=stat  success=no
exit=EACCES  a0=5573f123b360  a1=7ffca65100c0  a2=7ffca65100c0  a3=7fa85aaec772  items=0
ppid=46646  pid=46649  auid=4294967295  uid=48  gid=48  euid=48  suid=48  fsuid=48  egid=48
sgid=48    fsgid=48    tty=(none)    ses=4294967295    comm=httpd    exe=/usr/sbin/httpd
subj=system_u:system_r:httpd_t:s0 key=(null)

Hash: httpd,httpd_t,default_t,file,getattr
```

Sealert y el /var/log/messages nos da suficiente información como para identificar que hay un problema con el contexto del directorio de nuestra virtualhost “/web1” ya que no está incluido dentro del contexto de httpd para SELinux.

```
[root@nodo1 ~]# ls -ldZ /web1/
drwxr-xr-x. root root unconfined_u:object_r:default_t:s0 /web1/
[root@nodo1 ~]# ls -ldZ /web1/index.html
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 /web1/index.html
[root@nodo1                                                                    ~]#

[root@nodo1 ~]# semanage fcontext -l|grep web1
[root@nodo1 ~]#
```

Resolución:

Añadir el directorio del virtualhost al contexto adecuado:

```
[root@nodo1 ~]# semanage fcontext -l|grep web1
[root@nodo1 ~]# semanage fcontext -a -t httpd_sys_content_t "/web1(/.*)?"
[root@nodo1 ~]# semanage fcontext -l|grep web1
/web1(/.*)?      all files          system_u:object_r:httpd_sys_content_t:s0
[root@nodo1 ~]#
```

Restaurar los valores de contesto para el directorio del virtualhost:

```
[root@nodo1 ~]# ls -lrZ /web1/
-rw-r--r--. root root unconfined_u:object_r:default_t:s0 index.html

[root@nodo1 ~]# restorecon -Rvvv /web1
restorecon       reset       /web1       context       unconfined_u:object_r:default_t:s0-
       >unconfined_u:object_r:httpd_sys_content_t:s0
restorecon    reset    /web1/index.html    context    unconfined_u:object_r:default_t:s0-
       >unconfined_u:object_r:httpd_sys_content_t:s0

[root@nodo1 ~]#  ls -lrZ /web1/
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0 index.html
[root@nodo1 ~]#

[root@nodo1 ~]# getenforce
Enforcing
[root@nodo1 ~]# curl localhost
Hola esta es una web de para la practica de seelinux
[root@nodo1 ~]#
```

Gestionar contenedores

- Hallar imágenes en contenedores y extraerlas desde un registro remoto.

Lo primero es tener la maquina registrada, habilitar los repositorios en Red-Hat y el repositorio EPEL para instalar el paquete ***containers-tools, podman-docker y buildah***:

```
[root@rhel9 ~]#   subscription-manager register
Registering to: subscription.rhsm.redhat.com:443/subscription
Username: <username>
Password: <password>
```

```
[root@rhel9 ~]#  dnf install \
https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
Updating Subscription Management repositories.
Red Hat Enterprise Linux 9 for x86_64 - BaseOS (RPMs)                    9.0 MB/s | 6.9
MB     00:00
Red Hat Enterprise Linux 9 for x86_64 - AppStream (RPMs)                  18 MB/s |  15
MB     00:00
Last metadata expiration check: 0:00:01 ago on Sun 20 Nov 2022 23:51:54 CET.
epel-release-latest-9.noarch.rpm                                          42 kB/s |  18
kB     00:00
Package epel-release-9-4.el9.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
[root@rhel9 ~]# dnf search container-tools podman-docker buildah
... ... ... ... ... ...
Installed:
  container-tools-1-12.el9.noarch                                  podman-docker-2:4.2.0-
7.el9_1.noarch
  podman-remote-2:4.2.0-7.el9_1.x86_64              python3-libselinux-3.4-3.el9.x86_64
  python3-libsemanage-3.4-2.el9.x86_64                        python3-podman-3:4.2.1-
1.el9_1.noarch
  python3-pyxdg-0.27-3.el9.noarch                   python3-toml-0.10.2-6.el9.noarch
  skopeo-2:1.9.4-0.1.el9_1.x86_64                   toolbox-0.0.99.3-5.el9.x86_64
  udica-0.2.6-4.el9.noarch

Complete!
```

Para obtener imágenes de contenedores de un registro remoto (como el propio registro de contenedores de Red Hat) y añadirlas a su sistema local, utilice el comando `podman pull`:

```
# podman pull <registry>[:<port>]/[<namespace>/]<name>:<tag>
```

Utilice la opción **pull** para extraer una imagen de un registro remoto. Para extraer la imagen base de RHEL **ubi** y la imagen de registro **rsyslog** del registro de Red-Hat, escriba:

```
[root@rhel9 ~]#  podman login registry.redhat.io
Username: jagfloriano@gmail.com
Password:
Login Succeeded!
```

```
[root@rhel9 ~]# # podman pull registry.redhat.io/ubi9/ubi
```

```
[root@rhel9 ~]# podman pull registry.redhat.io/rhel9/rsyslog
Trying to pull registry.redhat.io/rhel9/rsyslog:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob 67a425a6e064 done
Copying blob d73280727573 done
Copying config a0901f8777 done
Writing manifest to image destination
Storing signatures
a0901f8777da385a006e344f796170e33e5e2eaf5f5a6f09ebaaf2919a6e374b
```

Para ver las imágenes resultantes del comando **podman** anterior, junto con cualquier otra imagen de su sistema, escriba **podman images**:

```
[root@rhel9 ~]# podman images
REPOSITORY                          TAG         IMAGE ID      CREATED                SIZE
registry.redhat.io/rhel9/rsyslog    latest      a0901f8777da  Less than a second ago  240 MB
registry.access.redhat.com/ubi9     latest      75f9d700cce5  Less than a second ago  219 MB
```

Para listar las imágenes en el almacenamiento local, introduzca:

```
[root@rhel9 ~]# podman images
REPOSITORY                          TAG         IMAGE ID      CREATED                SIZE
registry.access.redhat.com/ubi9     latest      b1e15923f8c9  Less than a second ago  219 MB
registry.redhat.io/rhel9/rsyslog    latest      a0901f8777da  Less than a second ago  240 MB
<none>                              <none>      75f9d700cce5  Less than a second ago  219 MB
```

- Examinar las imágenes en contenedores

Después de extraer una imagen a su sistema local y antes de ejecutarla, es una buena idea investigar esa imagen. Las razones para investigar una imagen antes de ejecutarla incluyen entender lo que hace la imagen y comprobar qué software hay dentro de la imagen.

El comando `podman inspect` muestra información básica sobre lo que hace una imagen.

```
[root@rhel9 ~]# podman  pull registry.access.redhat.com/ubi9
Trying to pull registry.access.redhat.com/ubi9:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob d74e20a2726b done
Copying config b1e15923f8 done
Writing manifest to image destination
Storing signatures
b1e15923f8c95d6a26ccf327169aafd0e404e564bbdd9c79985e47d71dc1a820
```

```
[root@rhel9 ~]# podman inspect registry.access.redhat.com/ubi9|head -15
[
     {
          "Id": "b1e15923f8c95d6a26ccf327169aafd0e404e564bbdd9c79985e47d71dc1a820",
          "Digest":
"sha256:7ad5a240ed08d5374ef1a0171582a11a90c0ddf6a9356700619973b7110ca903",
```

Mount a container: Usando el comando `podman`, monta un contenedor activo para investigar más a fondo su contenido:

```
[root@rhel9 ~]# podman run -d registry.redhat.io/rhel9/rsyslog
5d13c962d3639fd53d86872b3849d41e6695b476ccbbe30f7e86bae8150c532d
```

```
[root@rhel9 ~]# podman ps
CONTAINER ID  IMAGE                                    COMMAND           CREATED
STATUS            PORTS       NAMES
5d13c962d363  registry.redhat.io/rhel9/rsyslog:latest  /bin/rsyslog.sh  2 minutes ago
Up 2 minutes ago              determined_sammet
```

```
[root@rhel9 ~]# podman mount   5d13c962d363
/var/lib/containers/storage/overlay/b92997d488e9e146493a060865fcb4e12aa0b5135613b187f5b5
5befa832eb26/merged
```

```
[root@rhel9                                ~]#                                ls
/var/lib/containers/storage/overlay/b92997d488e9e146493a060865fcb4e12aa0b5135613b187f5b5
5befa832eb26/merged
afs  boot  etc   lib    lost+found  mnt  proc  run   srv  tmp  var
bin  dev   home  lib64  media       opt  root  sbin  sys  usr
```

Check the image's package list: Para comprobar los paquetes instalados en el contenedor, indique al comando `rpm` que examine los paquetes instalados en el punto de montaje del contenedor:

```
[root@rhel9 ~]# rpm -qa --root=/var/lib/containers/storage/overlay/b92997d488e9e146493a060865fcb4e12aa0b5135613b187f5b55befa832eb26/merged|head -5
libgcc-11.2.1-9.4.el9.x86_64
crypto-policies-20220223-1.git5203b41.el9_0.1.noarch
tzdata-2022e-1.el9_0.noarch
subscription-manager-rhsm-certificates-1.29.26.1-1.el9_0.x86_64
redhat-release-9.0-2.17.el9.x86_64
```

Inspección de imágenes remotas

Para inspeccionar una imagen de contenedor antes de llevarla a su sistema, puede utilizar el comando `skopeo inspect`. Con `skopeo inspect`, puede mostrar información sobre una imagen que reside en un registro de contenedores remoto. El siguiente comando inspecciona la imagen `ubi9-init` desde el registro de Red Hat.

```
[root@rhel9 ~]# skopeo inspect docker://registry.access.redhat.com/ubi9/ubi-init
{
    "Name": "registry.access.redhat.com/ubi9/ubi-init",
    "Digest":
"sha256:8f2d38f839f536937390ada9a11d1f5eb3b49a252c2aaccccf23b7a027457521",
    "RepoTags": [
        "9.0.0-16",
        "9.0.0-19",
        "9.1.0",
        "9.0.0",
        "9.0.0-16.1655192132",
        "9.0.0-26.1666626006",
        "9.0.0-26.1665072052",
        "9.0.0-16.1655192132-source",
        "9.0.0-26.1666626006-source",
        "9.0.0-26.1665072052-source",
        "9.0.0-28",
        "9.0.0-29",
        "9.0.0-23",
        "9.0.0-26",
        "9.0.0-26-source",
        "9.0.0-28-source",
        "9.1.0-5-source",
        "9.0.0-29-source",
        "9.1.0-5",
        "9.0.0-19-source",
        "9.0.0-16-source",
        "9.0.0-23-source",
        "latest"
    ],
```

- Gestionar los contenedores con comandos, como Podman y Skopeo:

Podman.

El comando `podman` (que significa Pod Manager) permite ejecutar contenedores como entidades independientes, sin necesidad de que intervengan Kubernetes, el tiempo de ejecución de Docker o cualquier otro tiempo de ejecución de contenedores. Es una herramienta que puede actuar como reemplazo del comando `Docker`

Las características de podman incluyen:

- **Based on the Docker interface**: Dado que la sintaxis de podman es un reflejo del comando docker, la transición a podman debería ser fácil para quienes estén familiarizados con docker.
- **Managing containers and images**: Tanto las imágenes de contenedores compatibles con Docker como con OCI pueden utilizarse con podman para:
    - Ejecutar, detener y reiniciar contenedores
    - Creación y gestión de imágenes de contenedores (push, commit, configuración, build, etc.)
- **Managing pods**: Además de ejecutar contenedores individuales, podman puede ejecutar un conjunto de contenedores agrupados en un pod. Un pod es la unidad de contenedores más pequeña que gestiona Kubernetes.
- **Working with no runtime**: podman no utiliza ningún entorno de ejecución para trabajar con contenedores.

Skopeo

Con el comando skopeo, puedes trabajar con imágenes de contenedores desde registros sin usar el demonio docker o el comando docker. Los registros pueden incluir el Registro Docker, sus propios registros locales, los registros de Red Hat Quay u OpenShift. Las actividades que puedes hacer con skopeo incluyen:

- **inspect**: La salida de un comando skopeo inspect es similar a la de un comando docker inspect: información de bajo nivel sobre la imagen del contenedor. Esa salida puede estar en formato json (por defecto) o en formato raw (usando la opción --raw).
- **copy**: Con skopeo copy puede copiar una imagen de contenedor de un registro a otro registro o a un directorio local.
- **layers**: El comando skopeo layers permite descargar las capas asociadas a las imágenes para que se almacenen como bolas de tar y archivos de manifiesto asociados en un directorio local.

Buildah

Es una herramienta open source basada en Linux que se utiliza para diseñar contenedores compatibles con la Open Container Initiative (OCI), por lo cual también se pueden utilizar con Docker y Kubernetes. Buildah le permite usar sus herramientas preferidas para diseñar imágenes de contenedores eficientes con mayor flexibilidad y seguridad, ya sea desde cero a partir de una imagen vacía, o bien con las de base actuales.

- Realizar una gestión básica de los contenedores, como ejecutar, iniciar, detener y registrar aquellos que se encuentran en funcionamiento

Podman

Para investigar dentro de un contenedor en ejecución, puede utilizar el comando `podman exec`. Con `podman exec`, puede ejecutar un comando (como `/bin/bash`) para entrar en un proceso de contenedor en ejecución para investigar ese contenedor.

La razón para usar `podman exec`, en lugar de simplemente lanzar el contenedor en un shell bash, es que puedes investigar el contenedor mientras está ejecutando su aplicación prevista. Al adjuntar al contenedor mientras está realizando su tarea prevista, se obtiene una mejor visión de lo que el contenedor realmente hace, sin necesariamente interrumpir la actividad del contenedor.

Aquí hay un ejemplo usando `podman exec` para mirar dentro de un `rsyslog` en funcionamiento, y luego mirar dentro de ese contenedor.

- `Launch a container`: Lanza un contenedor como la imagen del contenedor `rsyslog` descrita anteriormente. Escriba `podman ps` para asegurarse de que se está ejecutando:

```
[root@rhel9 ~]# podman run -d registry.redhat.io/rhel9/rsyslog
2a42e1fccb0c87d40df40d171f7922fc2b5924b4809fc44c0e5366f50852af1a
```

```
[root@rhel9 ~]# podman ps
CONTAINER ID  IMAGE                                       COMMAND           CREATED
STATUS            PORTS       NAMES
2a42e1fccb0c  registry.redhat.io/rhel9/rsyslog:latest  /bin/rsyslog.sh  6 seconds
ago  Up 6 seconds ago              gallant_mahavira
```

- Entre en el contenedor con `podman exec`: Utilice el ID o el nombre del contenedor para abrir un shell bash y acceder al contenedor en ejecución. A continuación, puede investigar los atributos del contenedor de la siguiente manera:

```
[root@rhel9 ~]# podman exec -it  2a42e1fccb0c /bin/bash
[root@2a42e1fccb0c /]# cat /etc/redhat-release
Red Hat Enterprise Linux release 9.0 (Plow)
```

```
[root@2a42e1fccb0c /]# ps -ef;date
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 23:02 ?        00:00:00 /usr/sbin/rsyslogd -n
root           4       0  0 23:03 pts/0    00:00:00 /bin/bash
root          42       4  0 23:05 pts/0    00:00:00 ps -ef
Sun Nov 20 23:05:07 UTC 2022
```

```
[root@2a42e1fccb0c /]# df -h
Filesystem      Size  Used Avail Use% Mounted on
overlay          26G  2.6G   24G  10% /
tmpfs            64M     0   64M   0% /dev
shm              63M     0   63M   0% /dev/shm
tmpfs           1.6G  9.0M  1.5G   1% /etc/hosts
devtmpfs        4.0M     0  4.0M   0% /proc/keys
[root@2a42e1fccb0c /]#
```

Los comandos que se acaban de ejecutar desde el shell bash (que se ejecuta dentro del contenedor) muestran varias cosas.

Para trabajar con los contenedores desde el sistema anfitrión, puede abrir un shell y probar algunos de los siguientes comandos. `podman ps`: La opción `ps` muestra todos los contenedores que se están ejecutando actualmente:

```
[root@rhel9 ~]# podman run -d registry.redhat.io/rhel9/rsyslog
2a42e1fccb0c87d40df40d171f7922fc2b5924b4809fc44c0e5366f50852af1a
```

```
[root@rhel9 ~]# podman ps
CONTAINER ID  IMAGE                                       COMMAND           CREATED
STATUS            PORTS       NAMES
2a42e1fccb0c  registry.redhat.io/rhel9/rsyslog:latest  /bin/rsyslog.sh  6 seconds
ago  Up 6 seconds ago              gallant_mahavira
```

Si hay contenedores que no se están ejecutando, pero no fueron eliminados (opción`--rm`), los contenedores están presentes y pueden ser reiniciados. El comando `podman ps -a` muestra todos los contenedores, en ejecución o detenidos.

```
[root@rhel9 ~]# podman ps -a
CONTAINER ID  IMAGE                                       COMMAND           CREATED
STATUS                  PORTS       NAMES
a128d4ffa24e  registry.access.redhat.com/ubi9:latest    top               2 weeks
ago    Created                              myubi
5d13c962d363   registry.redhat.io/rhel9/rsyslog:latest   /bin/rsyslog.sh   2 weeks
ago    Exited (0) 2 weeks ago              determined_sammet
2a42e1fccb0c  registry.redhat.io/rhel9/rsyslog:latest  /bin/rsyslog.sh  4 minutes
ago  Up 4 minutes ago                    gallant_mahavira
```

Para inspeccionar los metadatos de un contenedor existente, utilice el comando `podman inspect`. Puede mostrar todos los metadatos o sólo los seleccionados para el contenedor. Por ejemplo, para mostrar todos los metadatos de un contenedor seleccionado, escriba:

```
podman inspect CONTAINER-ID
```

Un contenedor que no necesita ejecutarse de forma interactiva a veces puede ser reiniciado después de ser detenido con sólo la opción `start` y el ID o nombre del contenedor.

```
[root@rhel9 ~]# podman run --name=myubi1 -it registry.access.redhat.com/ubi9/ubi
[root@2f4126a7f7df /]# exit
```

```
[root@rhel9 ~]# podman start myubi
myubi
```

Para iniciar un contenedor y poder trabajar con él desde el shell local, utiliza las opciones `-a` (adjuntar) y `-i` (interactivo). Una vez que se inicie el shell bash, ejecute los comandos que desee dentro del contenedor y escriba exit para matar el shell y detener el contenedor.

```
[root@rhel9 ~]# podman start -a -i  myubi
[root@3ce8eb2d0c7c /]# date
Sun Nov 20 23:26:52 UTC 2022
[root@3ce8eb2d0c7c /]# exit
exit
[root@rhel9 ~]#
```

Para detener un contenedor en ejecución que no está unido a una sesión de terminal, utilice la opción de parada y el ID o número del contenedor. Por ejemplo:

```
[root@rhel9 ~]# podman stop myubi
myubi
```

La opción `stop` envía una señal SIGTERM para terminar un contenedor en ejecución. Si el contenedor no se detiene después de un período de gracia (10 segundos por defecto), `podman` envía una señal SIGKILL. También puedes utilizar el comando `podman kill` para matar un contenedor (SIGKILL) o enviar una señal diferente a un contenedor. Aquí hay un ejemplo de enviar una señal SIGHUP a un contenedor (si es soportado por la aplicación, un SIGHUP hace que la aplicación vuelva a leer sus archivos de configuración):

```
[root@rhel9 ~]# podman ps
CONTAINER ID  IMAGE  COMMAND   CREATED     STATUS            PORTS       NAMES
3ce8eb2d0c7c  registry.access.redhat.com/ubi9/ubi:latest  /bin/bash    3 minutes
ago  Up 4 seconds ago              myubi
```

```
[root@rhel9 ~]# podman kill --signal="SIGHUP"  myubi
Myubi
```

```
[root@rhel9 ~]# podman ps
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[root@rhel9 ~]#
```

Skopeo

Cuando se inspecciona una imagen de contenedor desde un registro, es necesario identificar el formato del contenedor (como docker), la ubicación del registro (como docker.io o localhost) y el repositorio/imagen (como ubi9/ubi). El siguiente ejemplo inspecciona la imagen del contenedor mariadb desde el Registro Docker:

```
[root@rhel9 ~]# skopeo inspect docker://docker.io/library/mariadb|head -10
```

Este procedimiento demuestra cómo inspeccionar una imagen de contenedor remoto usando Skopeo. Ejecutar Skopeo en un contenedor significa que el sistema de archivos raíz del contenedor está aislado del sistema de archivos raíz del host. Para compartir o copiar archivos entre el host y el contenedor, debe montar archivos y directorios.

- Inicie sesión en registry.redhat.io registry

```
[root@rhel9 ~]# podman login registry.redhat.io
Username: XXXXXXXXXXX
Password:
Login Succeeded!
```

- Obtenga la imagen del contenedor registration.redhat.io/rhel9/skopeo:

```
[root@rhel9 ~]# podman pull registry.redhat.io/rhel9/skopeo
8be654bae76a5754721d29c0fa5cc30ab3232cf1706ff8c80af2d89e9e4a8fd1
```

- Inspeccionar imagen de contenedor remoto registration.access.redhat.com/ubi9/ubi usando Skopeo:

```
[root@rhel9 ~]# podman run --rm registry.redhat.io/rhel9/skopeo
```

```
[root@rhel9 ~]# skopeo inspect docker://registry.access.redhat.com/ubi9/ubi|head
-3
{
    "Name": "registry.access.redhat.com/ubi9/ubi",
    "Digest":
```

La opción --rm elimina la imagen registration.redhat.io/rhel9/skopeo después de que se cierra el contenedor.Trabajar con registros de contenedores requiere una autenticación para acceder y modificar datos. Skopeo admite varias formas de especificar credenciales.

Con este enfoque, puede especificar las credenciales en la línea de comandos mediante la opción --cred USERNAME[:PASSWORD].

```
podman run --rm registry.redhat.io/rhel9/skopeo inspect --creds $USER:$PASSWORD
docker://$IMAGE
```

Buildah

El procedimiento demuestra cómo ejecutar Buildah en un contenedor y crear un contenedor de trabajo basado en una imagen.

```
[root@rhel9 ~]# # podman run --rm --device /dev/fuse -it \
registry.redhat.io/rhel9/buildah /bin/bash
```

```
root@rhel9 ~]# buildah from registry.access.redhat.com/ubi9
ubi-working-container
```

```
[root@rhel9 ~]# buildah run --isolation=chroot ubi-working-container ls /
afs  boot  etc   lib    lost+found  mnt  proc  run   srv  tmp  var
bin  dev   home  lib64  media       opt  root  sbin  sys  usr
```

```
[root@rhel9 ~]# buildah images
REPOSITORY                          TAG      IMAGE ID       CREATED       SIZE
registry.redhat.io/rhel9/skopeo     latest   8be654bae76a   11 days ago   265 MB
registry.access.redhat.com/ubi9/ubi latest   b1e15923f8c9   11 days ago   219 MB
registry.access.redhat.com/ubi9     latest   b1e15923f8c9   11 days ago   219 MB
registry.redhat.io/rhel9/rsyslog    latest   a0901f8777da   13 days ago   240 MB
<none>                              <none>   75f9d700cce5   13 days ago   219 MB
```

```
[root@rhel9 ~]# buildah containers
CONTAINER ID  BUILDER  IMAGE ID     IMAGE NAME                       CONTAINER
NAME
8184479facec     *     b1e15923f8c9 registry.access.redhat.com/ub... ubi-working-
container
```

- Ejecutar un servicio dentro de un contenedor

Podman permite dentro de los contenedores su propio systemd, esto viene habilitado siempre por defecto, pero se puede modificar mediante **"--systemd=*true* | *false* | *always*".**

Según el man, El valor siempre aplica el modo systemd sin mirar el nombre del ejecutable. De lo contrario, si se establece en verdadero y el comando que está ejecutando dentro del contenedor es systemd, /usr/sbin/init, /sbin/init o /usr/local/sbin/init.

Ejecutar el contenedor en modo systemd provoca los siguientes cambios:

- Podman monta sistemas de archivos tmpfs en los siguientes directorios
  - */run*
  - */run/lock*
  - */tmp*
  - */sys/fs/cgroup/systemd*
  - */var/lib/journal*

- Podman establece la señal de parada predeterminada en **SIGRTMIN+3.**
-
- Podman establece la variable de entorno container_uuid en el contenedor en los primeros 32 caracteres de la identificación del contenedor.

Esto permite que systemd se ejecute en un contenedor confinado sin ninguna modificación.

Tenga en cuenta que en los sistemas SELinux, systemd intenta escribir en el sistema de archivos cgroup. Los contenedores que escriben en el sistema de archivos cgroup se niegan de forma predeterminada. El booleano container_manage_cgroup debe estar habilitado para que esto se permita en un sistema separado de SELinux.

```
setsebool -P container_manage_cgroup true
```

Ejemplo práctico:

- Crear un Dockerfile para ejecutar systemd en un contenedor usando Podman:

```
[root@rhel9 ~]# cat Dockerfile
FROM fedora
RUN dnf -y install httpd; dnf clean all; systemctl enable httpd
EXPOSE 80
CMD [ "/sbin/init" ]
[root@rhel9 ~]#
```

- Crear el contenedor:

```
[root@rhel9 ~]#  podman build -t systemd .
STEP 1/4: FROM fedora
STEP 2/4: RUN dnf -y install httpd; dnf clean all; systemctl enable httpd
--> Using cache b2ac7c528c60995d0fedf500a3693b4e3ad11a422f89c3467aaa56fcd25db64a
--> b2ac7c528c6
STEP 3/4: EXPOSE 80
--> Using cache ab86ff732b7ad864ea73091eebd8f3e5bdc2ed3d15a1212338ea9100d0893bd9
--> ab86ff732b7
STEP 4/4: CMD [ "/sbin/init" ]
--> Using cache 910a343896d1b83a5fdca48c285b9d7dd54944e5f65af36d5c53d837241cb894
COMMIT systemd
--> 910a343896d
Successfully tagged localhost/systemd:latest
910a343896d1b83a5fdca48c285b9d7dd54944e5f65af36d5c53d837241cb894
```

- Dígale a SELinux que está bien permitir que systemd manipule su configuración de Cgroups.

```
[root@rhel9 ~]#  setsebool -P container_manage_cgroup true
[root@rhel9 ~]#
```

- Arrancar el contenedor:

```
[root@rhel9 ~]# podman run -ti -p 80:80 systemd
systemd 251.7-611.fc37 running in system mode (+PAM +AUDIT +SELINUX -APPARMOR
+IMA +SMACK +SECCOMP -GCRYPT +GNUTLS +OPENSSL +ACL +BLKID +CURL +ELFUTILS +FIDO2
+IDN2 -IDN -IPTC +KMOD +LIBCRYPTSETUP +LIBFDISK +PCRE2 +PWQUALITY +P11KIT
+QRENCODE +TPM2 +BZIP2 +LZ4 +XZ +ZLIB +ZSTD +BPF_FRAMEWORK +XKBCOMMON +UTMP
+SYSVINIT default-hierarchy=unified)
Detected virtualization container-other.
Detected architecture x86-64.

Welcome to Fedora Linux 37 (Container Image)!
..............................
[  OK  ] Started console-getty.service - Console Getty.
[  OK  ] Reached target getty.target - Login Prompts.
[  OK  ] Started dbus-broker.service - D-Bus System Message Bus.
[  OK  ] Started systemd-logind.service - User Login Management.
[  OK  ] Started httpd.service - The Apache HTTP Server.
[  OK  ] Reached target multi-user.target - Multi-User System.
[  OK  ] Reached target graphical.target - Graphical Interface.
         Starting systemd-update-utmp-runlevel.service - Record Runlevel Change
in UTMP...
[  OK  ] Finished systemd-update-utmp-runlevel.service - Record Runlevel Change
in UTMP.
```

- Verificar que el servicio está funcionando

```
curl localhost
```

Portar contenedores a systemd usando Podman

Habilitar servicios systemd.

- Para habilitar un servicio al inicio del sistema, sin importar si el usuario inició sesión o no, Debe copiar los archivos de la unidad systemd en el directorio `/etc/systemd/system`:

```
cp container.service  /etc/systemd/system
systemctl enable <service>
```

- Para iniciar un servicio en el inicio de sesión del usuario y detenerlo en el cierre de sesión del usuario, Debe copiar los archivos de la unidad systemd en `$HOME/.config/systemd/user`:

```
cp container.service  $HOME/.config/systemd/user
su - $user
systemctl –user enable <service>
```

- Para permitir que los usuarios inicien un servicio al inicio del sistema y persistan después de los cierres de sesión, ingrese:

```
loginctl enable-linger <username>
```

Arranque automático de contenedores con systemd

Puedes controlar el estado del sistema systemd y del gestor de servicios utilizando el comando `systemctl`. Esta sección muestra el procedimiento general sobre cómo habilitar, iniciar y detener el servicio como usuario no root. Para instalar el servicio como usuario root, omita la opción `--user`.

- Recargar la configuración del gestor systemd:

```
systemctl --user daemon-reload
```

- Habilitar el servicio `container.service` e iniciarlo en el momento del arranque:

```
systemctl --user enable container.service
```

- Para iniciar el servicio inmediatamente:

```
systemctl --user start container.service
```

- Comprueba el estado del servicio:

```
systemctl --user status container.service
```

Puede comprobar si el servicio está activado mediante el comando `systemctl is-enabled container.service`.

Generar un archivo de unidad systemd usando Podman

Podman permite a systemd controlar y gestionar los procesos de los contenedores. Puede generar un archivo de unidad systemd para los contenedores y pods existentes utilizando el comando `podman generate systemd`, esto asegura que se obtiene la última versión de los archivos de unidades.

Por defecto, Podman genera un archivo de unidad para los contenedores o pods existentes. Puede generar archivos de unidad systemd más portables utilizando la opción `podman generate systemd --new`. La bandera `--new` indica a Podman que genere archivos de unidad que creen, inicien y eliminen contenedores.

A continuación un ejemplo práctico de como generar un fichero de unidad systemd para crear un servicio que arranque un contenedor con httpd24:

- Extraiga la imagen que desea utilizar en su sistema. Por ejemplo, para extraer la imagen `httpd-24`:

```
[root@rhel9 ~]# podman pull registry.access.redhat.com/ubi9/httpd-24
Trying to pull registry.access.redhat.com/ubi9/httpd-24:latest...
Getting image source signatures
Checking if image destination supports signatures
Copying blob d74e20a2726b skipped: already exists
Copying blob 47d859467738 done
Copying blob dac0192805be done
Copying config 4afe283d91 done
Writing manifest to image destination
Storing signatures
4afe283d911ab76387e4a7054decb8c90db0b6fc5b271d1907979c6163db5f21
```

- Enumera todas las imágenes disponibles en tu sistema:

```
[root@rhel9 ~]# podman images|grep -i httpd-2
registry.access.redhat.com/ubi9/httpd-24 latest  4afe283d911a 11 days ago  379 MB
```

- Cree el contenedor `httpd`:

```
[root@rhel9 ~]# podman create --name httpd -p 8080:8080
registry.access.redhat.com/ubi9/httpd-24
b6ff93c25b68177a0a9aa3a64d5186d80cd29157512a62df387686822f964025
```

- Para verificar que el contenedor ha sido creado, liste todos los contenedores:

```
[root@rhel9 ~]# podman ps -a
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
b6ff93c25b68    registry.access.redhat.com/ubi9/httpd-24:latest    /usr/bin/run-
http...  About a minute ago  Created                      0.0.0.0:8080->8080/tcp
httpd
```

- Generar un archivo de unidad systemd para el contenedor `busybox`:

```
[root@rhel9 ~]# podman generate systemd --new --files --name httpd
/root/container-httpd.service
```

- Muestra el contenido del archivo de la unidad systemd generado en `container-httpd.service`:

```
[root@rhel9 ~]# cat /root/container-httpd.service
# container-httpd.service
# autogenerated by Podman 4.2.0
# Mon Nov 21 01:41:15 CET 2022

[Unit]
Description=Podman container-httpd.service
Documentation=man:podman-generate-systemd(1)
Wants=network-online.target
After=network-online.target
RequiresMountsFor=%t/containers

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStartPre=/bin/rm -f %t/%n.ctr-id
ExecStart=/usr/bin/podman run \
        --cidfile=%t/%n.ctr-id \
        --cgroups=no-conmon \
        --rm \
        --sdnotify=conmon \
        -d \
        --replace \
        --name httpd \
        -p 8080:8080 registry.access.redhat.com/ubi9/httpd-24
ExecStop=/usr/bin/podman stop --ignore --cidfile=%t/%n.ctr-id
ExecStopPost=/usr/bin/podman rm -f --ignore --cidfile=%t/%n.ctr-id
Type=notify
NotifyAccess=all

[Install]
WantedBy=default.target
```

- Copie los archivos de la unidad en `/usr/lib/systemd/system` para instalarlos como usuario root:

```
[root@rhel9 ~]# cp -Z container-httpd.service /etc/systemd/system
[root@rhel9 ~]# ls -lrt /etc/systemd/system/container-httpd.service
-rw-r--r--. 1 root root 786 Nov 21 01:42 /etc/systemd/system/container-
httpd.service
```

- Habilitar y arrancar `container-httpd.service`:

```
[root@rhel9 ~]# systemctl --user daemon-reload
```

```
[root@rhel9 ~]# systemctl --user enable /etc/systemd/system/container-httpd.service
Created symlink /root/.config/systemd/user/container-httpd.service → /etc/systemd/system/container-httpd.service.
Created symlink /root/.config/systemd/user/default.target.wants/container-httpd.service → /etc/systemd/system/container-httpd.service.
[root@rhel9 ~]#
```

```
[root@rhel9 ~]# systemctl start container-httpd.service
```

- Verificar estado del servicio:

```
[root@rhel9 ~]# systemctl status container-httpd.service
● container-httpd.service - Podman container-httpd.service
     Loaded: loaded (/etc/systemd/system/container-httpd.service; disabled; vendor preset: disabled)
     Active: active (running) since Mon 2022-11-21 01:46:29 CET; 1s ago
       Docs: man:podman-generate-systemd(1)
    Process: 15530 ExecStartPre=/bin/rm -f /run/container-httpd.service.ctr-id (code=exited, status=0/SUCCESS)
   Main PID: 15624 (conmon)
      Tasks: 2 (limit: 48895)
     Memory: 1.0M
        CPU: 144ms
     CGroup: /system.slice/container-httpd.service
             └─15624 /usr/bin/conmon --api-version 1 -c d4c60b8acc0f9dc2157f6e0163404ef18f8d538a4a64940ca3c2255cd24e8662 -u d4c60b8acc0f9dc2157f6e016340
```

```
[root@rhel9 ~]# curl  localhost:8080|wc -l
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  5909  100  5909    0     0  5770k      0 --:--:-- --:--:-- --:--:-- 5770k
179
```

Crear un servicio systemd que arranque un contenedor cuando un usuario se logee.

- Crear un usuario y logarse con él:

```
[root@rhel9 ~]# useradd test
[root@rhel9 ~]# su - test
```

- Obtener imagen y verificarlo:

```
[test@rhel9 ~]$ podman pull docker.io/library/httpd
Trying to pull docker.io/library/httpd:latest...
Getting image source signatures
Copying blob b1c114085b25 done
Copying blob 4691bd33efec done
Copying blob a603fa5e3b41 done
Copying blob ff7b0b8c417a done
Copying blob 9df1012343c7 done
Copying config 8653efc8c7 done
Writing manifest to image destination
Storing signatures
8653efc8c72daee8c359a37a1dded6270ecd1aede2066cbecd5be7f21c916770
```

```
[test@rhel9 ~]$ podman images
REPOSITORY                TAG         IMAGE ID      CREATED     SIZE
docker.io/library/httpd   latest      8653efc8c72d  6 days ago  150 MB
```

- Verificamos que el contenedor está parado:

```
[test@rhel9 ~]$ podman ps
CONTAINER ID  IMAGE       COMMAND     CREATED     STATUS      PORTS       NAMES
[test@rhel9 ~]$
```

- Preparar el entorno para que el servicio se arranque tras logarse el usuario:

```
[test@rhel9 ~]$ mkdir  -p ~/.config/systemd/user
```

- Generar el fichero de unidad mediante podman:

```
test@rhel9~]$  podman  generate  systemd  New  >  ~/.config/systemd/user/New-
container.service
```

- Hacer un demon-reload y arrancar el nuevo servicio manunalmente.

```
[test@rhel9 ~]$ systemctl --user daemon-reload
[test@rhel9 ~]$  systemctl --user start New-container.service
```

- Verificar que el servicio arranca sin problemas:

```
[test@rhel9 ~]$ systemctl --user status New-container.service
● New-container.service - Podman container-
f66f0a9bbc5888d91b5abfcf463d490b27cf9255f1b9d1b8e506cd2c58354d06.service
     Loaded: loaded (/home/test/.config/systemd/user/New-container.service;
disabled; vendor preset: disabled)
     Active: active (running) since Mon 2022-11-21 20:11:02 CET; 3s ago
       Docs: man:podman-generate-systemd(1)
    Process: 2518 ExecStart=/usr/bin/podman start
f66f0a9bbc5888d91b5abfcf463d490b27cf9255f1b9d1b8e506cd2c58354d06 (code=exited,
status=0/SUCCESS)
   Main PID: 2557 (conmon)
      Tasks: 17 (limit: 48895)
     Memory: 12.3M
        CPU: 114ms
     CGroup: /user.slice/user-1002.slice/user@1002.service/app.slice/New-
container.service
             ├─2540 /usr/bin/slirp4netns --disable-host-loopback --mtu=65520 --
enable-sandbox --enable-seccomp --enable-ipv6 -c -e 3 -r 4 --netns-type=path
/run/user/1002/netns/net>
             ├─2542 rootlessport
             ├─2548 rootlessport-child
             └─2557 /usr/bin/conmon --api-version 1 -c
f66f0a9bbc5888d91b5abfcf463d490b27cf9255f1b9d1b8e506cd2c58354d06 -u
f66f0a9bbc5888d91b5abfcf463d490b27cf9255f1b9d1b8e506cd2c5>
```

```
[test@rhel9 ~]$  podman ps
CONTAINER ID  IMAGE                                COMMAND               CREATED
STATUS            PORTS                 NAMES
f66f0a9bbc58  docker.io/library/httpd:latest  httpd-foreground  5 minutes ago  Up 8
seconds ago  0.0.0.0:8085->80/tcp  New
```

- Parar el servicio para posteriormente habilitarlo de manera automática:

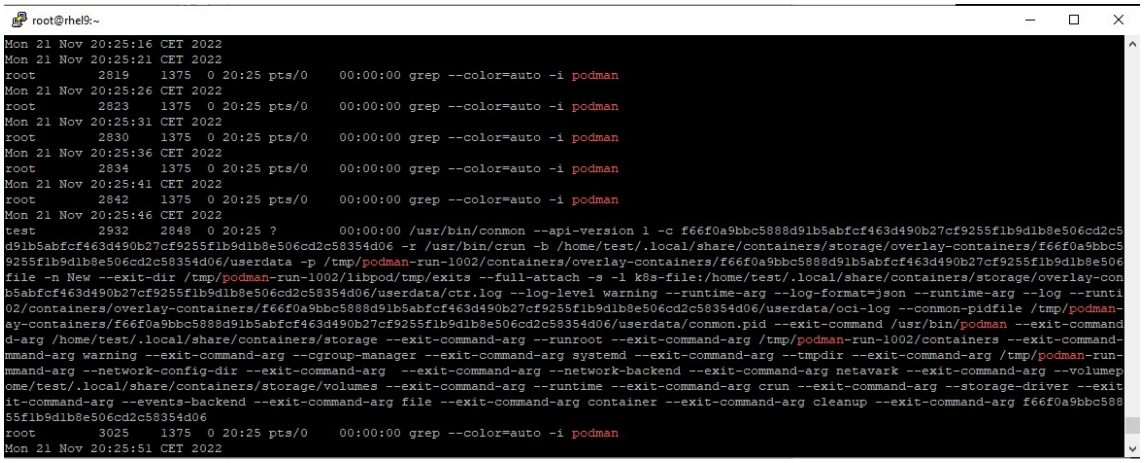
```
[test@rhel9 ~]$ systemctl --user stop New-container.service
```

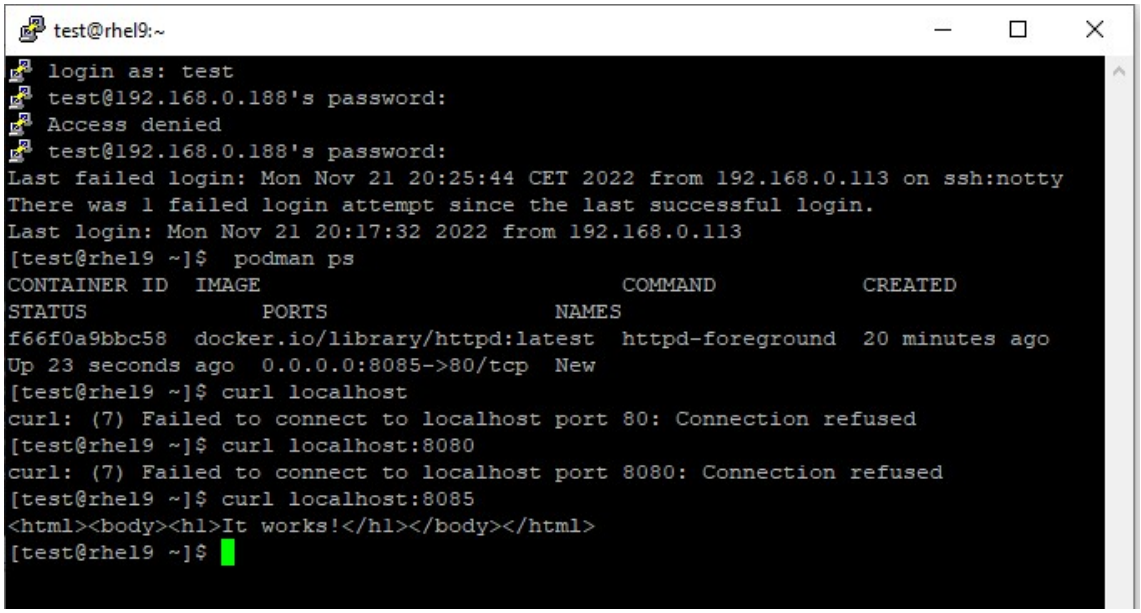
```
[test@rhel9 ~]$  podman ps
CONTAINER ID  IMAGE COMMAND  CREATED STATUS   PORTS  NAMES
[test@rhel9 ~]$
```

```
[test@rhel9 ~]$ systemctl --user enable New-container.service
Created symlink /home/test/.config/systemd/user/default.target.wants/New-
container.service → /home/test/.config/systemd/user/New-container.service.
[test@rhel9 ~]$ exit
```

Ahora siempre que el usuario test inicie sesión en el sistema, automáticamente se arrancará el servicio ***New-container*** hasta que la sesión termine.

Podemos verificarlo de la siguiente manera, desde un terminal tenemos un bucle que chequea todos los procesos que ejecuten podman, por lo que mientras no exista sesión del usuario test, el contenedor estará parado y una vez se inicia sesión aparecerá un proceso que estará vivo hasta finalizar la sesión.





Arranque automático de pods mediante systemd

Podríamos definir un pod como una combinación de contenedores que funcionan como un conjunto. Este apartado se ha ejecutado según la documentación de Red-Hat[5], pero no se ha podido demostrar sus resultados. Puede iniciar varios contenedores como servicios systemd. Tenga en cuenta que el comando `systemctl` sólo debe utilizarse en el pod y no debe iniciar o detener contenedores individualmente a través de `systemctl`, ya que son gestionados por el servicio del pod junto con el infra-contenedor interno.

- Cree un pod vacío, por ejemplo, llamado `systemd-pod`:

```
[bonzo@rhel9 ~]$  podman pod create --name systemd-pod
180a5b366733d7afa17d49814ecc367b3e32d3a2360d048e8a32c06c58b7bd77
```

- Enumerar todas las vainas:

```
[bonzo@rhel9 ~]$  podman ps -a
CONTAINER ID  IMAGE                                         COMMAND      CREATED
STATUS      PORTS       NAMES
ca13e4c1ce55  localhost/podman-pause:4.2.0-1666809014                6 seconds ago
Created                 180a5b366733-infra
```

- Cree dos contenedores en el pod vacío. Por ejemplo, para crear `container0` y `container1` en `systemd-pod`:

```
[bonzo@rhel9  ~]$  podman  create  --pod  systemd-pod  --name  container1
registry.access.redhat.com/ubi9/httpd-24 top
7e9b862f58120dd76b12c9019bb60cdf841863049082165429deeb5b1652a0a8
[bonzo@rhel9  ~]$  podman  create  --pod  systemd-pod  --name  container0
registry.access.redhat.com/ubi9/httpd-24 top
efb045d67de27d4658314c389bffe56982d7ef5fff48f7c5520f8a1c6a99e181
```

- Enumerar todos los pods y contenedores asociados a ellos:

```
[bonzo@rhel9 ~]$ podman ps -a --pod
CONTAINER ID  IMAGE     COMMAND            CREATED     STATUS       PORTS        NAMES
POD ID        PODNAME
ca13e4c1ce55  localhost/podman-pause:4.2.0-1666809014                          43
minutes ago  Up 32 minutes ago                      180a5b366733-infra  180a5b366733
systemd-pod
efb045d67de2  registry.access.redhat.com/ubi9/httpd-24:latest  top               41
minutes ago  Exited (1) 31 minutes ago             container0          180a5b366733
systemd-pod
7e9b862f5812  registry.access.redhat.com/ubi9/httpd-24:latest  top -p 8080:8080  41
minutes ago  Exited (1) 31 minutes ago             container1          180a5b366733
systemd-pod
```

[5] https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/9/html/building_running_and_managing_containers/assembly_porting-containers-to-systemd-using-podman_building-running-and-managing-containers

- Generar el archivo de unidad systemd para el nuevo pod:

```
[bonzo@rhel9 ~]$   podman generate systemd --files --name systemd-pod
/home/bonzo/pod-systemd-pod.service
/home/bonzo/container-container0.service
/home/bonzo/container-container1.service
```

Observe que se generan tres archivos de unidad systemd, uno para el pod `systemd-pod` y otro para el contenedor `container0 y container1`.

- Mostrar `pod-systemd-pod.service` archivo de la unidad:

```
[bonzo@rhel9 ~]$ cat pod-systemd-pod.service
# pod-systemd-pod.service
# autogenerated by Podman 4.2.0
# Mon Nov 21 18:53:55 CET 2022

[Unit]
Description=Podman pod-systemd-pod.service
Documentation=man:podman-generate-systemd(1)
Wants=network-online.target
After=network-online.target
RequiresMountsFor=
Requires=container-container0.service container-container1.service
Before=container-container0.service container-container1.service

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStart=/usr/bin/podman start 180a5b366733-infra
ExecStop=/usr/bin/podman stop -t 10 180a5b366733-infra
ExecStopPost=/usr/bin/podman stop -t 10 180a5b366733-infra
PIDFile=/tmp/podman-run-1000/containers/overlay-
containers/ca13e4c1ce55cd20391e151f001c9d2cf990ba4d2a322892a6308fc7554c59e9/userdata
/conmon.pid
Type=forking

[Install]
WantedBy=default.target
[bonzo@rhel9 ~]$
```

La línea `Requires` en la sección `[Unit]` define las dependencias de los archivos de unidad `container-container0.service` y `container-container1.service`. Ambos archivos de unidad se activarán.

Las líneas `ExecStart` y `ExecStop` de la sección `[Service]` inician y detienen el infra-contenedor, respectivamente.

- Mostrar `container-container0.service` archivo de la unidad:

```
[bonzo@rhel9 ~]$ cat container-container0.service
# container-container0.service
# autogenerated by Podman 4.2.0
[Unit]
Description=Podman container-container0.service
Documentation=man:podman-generate-systemd(1)
Wants=network-online.target
After=network-online.target
RequiresMountsFor=/tmp/podman-run-1000/containers
BindsTo=pod-systemd-pod.service
After=pod-systemd-pod.service

[Service]
Environment=PODMAN_SYSTEMD_UNIT=%n
Restart=on-failure
TimeoutStopSec=70
ExecStart=/usr/bin/podman start container0
ExecStop=/usr/bin/podman stop -t 10 container0
ExecStopPost=/usr/bin/podman stop -t 10 container0
PIDFile=/tmp/podman-run-1000/containers/overlay-
containers/efb045d67de27d4658314c389bffe56982d7ef5fff48f7c5520f8a1c6a99e181/userdata
/conmon.pid
Type=forking

[Install]
WantedBy=default.target
```

La línea `BindsTo` de la sección `[Unit]` define la dependencia del archivo de unidad `pod-systemd-pod.service`. Las líneas `ExecStart` y `ExecStop` de la sección `[Service]` inician y detienen el `container0` respectivamente.

- Copie todos los archivos generados en $HOME/.config/systemd/user para instalarlos como usuario no root:

```
[bonzo@rhel9 ~]# cp pod-systemd-pod.service container-container0.service
container-container1.service $HOME/.config/systemd/user
```

- Habilitar el servicio e iniciarlo al iniciar la sesión del usuario, el servicio se detendrá al cerrar la sesión del usuario:

```
[bonzo@rhel9 ~]$ systemctl enable --user pod-systemd-pod.service
Created symlink /home/bonzo/.config/systemd/user/default.target.wants/pod-
systemd-pod.service → /home/bonzo/.config/systemd/user/pod-systemd-pod.service.
```

- Comprueba si el servicio está activado:

```
[bonzo @rhel9 ~]# systemctl is-enabled --user pod-systemd-pod.service
enabled
```

- Asignar un almacenamiento permanente a un contenedor

Preparación de ubicaciones de almacenamiento permanente

El almacenamiento en contenedores es efímero, ya que su contenido no se conserva después de que se retira el contenedor.

Un contenedor en ejecución obtiene una nueva capa sobre su imagen de contenedor base, y esta capa es el almacenamiento del contenedor. Al principio, esta capa es el único almacenamiento de lectura/escritura disponible para el contenedor y se usa para crear archivos de trabajo, archivos temporales y archivos de registro. Esos archivos se consideran volátiles.

Una aplicación no deja de funcionar si se pierden. La capa de almacenamiento del contenedor es exclusiva del contenedor en ejecución, por lo que, si se crea otro contenedor a partir de la misma imagen base, obtiene otra capa de lectura/escritura. Esto asegura que los recursos de cada contenedor estén aislados de otros contenedores similares.

El almacenamiento de contenedores efímeros no es suficiente para las aplicaciones que necesitan conservar datos más allá de la vida útil del contenedor, como las bases de datos.

Para admitir tales aplicaciones, el administrador debe proporcionar un contenedor con almacenamiento persistente.

El directorio de trabajo es el directorio que contiene todos los archivos necesarios para construir la imagen. Crear un directorio de trabajo vacío es una buena práctica para evitar incorporar archivos innecesarios a la imagen. Por razones de seguridad, el directorio raíz, /, nunca debe usarse como directorio de trabajo para compilaciones de imágenes.

Preparación del directorio de host

Podman puede montar directorios de host dentro de un contenedor en ejecución.

La aplicación en contenedores ve estos directorios de host como parte del almacenamiento del contenedor, al igual que las aplicaciones normales ven un volumen de red remoto como si fuera parte del sistema de archivos del host. ***Pero el contenido de estos directorios de host no se recupera después de que se detiene el contenedor***, por lo que se pueden montar en nuevos contenedores cuando sea necesario.

A continuación, se describe una forma de configurar el directorio de host:

- Crear un directorio:

```
[user@host ~]$ mkdir -pv /empty/working/directory
```

- Proporcionar permisos, El usuario que ejecuta procesos en el contenedor debe ser capaz de escribir archivos en el directorio. El permiso debe definirse con el ID de usuario numérico (UID) del contenedor. `El comando podman unshare proporciona

una sesión para ejecutar comandos dentro del mismo espacio de nombres de usuario que el proceso` que se ejecuta dentro del contenedor.

```
[user@host ~]$ podman unshare chown -R UID:GID /working/directory
```

- Aplique el contexto `container_file_t` al directorio (y todos los subdirectorios) para permitir que los contenedores accedan a todo su contenido.

```
[user@host ~]$ sudo semanage fcontext -a -t container_file_t
'/working/directory(/.*)?'
```

- Aplique la política SELinux al contenedor que configuró en el primer paso al directorio recién creado:

```
[user@host ~]$ sudo restorecon -Rv /working/directory
```

- Ejecute el siguiente comando con podman unshare para verificar la ID de usuario numérica (UID) del contenedor.

```
[user@host ~]$ podman unshare ls -ld /working/directory
`P.S. The host directory must be configured before starting the container that
uses the directory.`
```

Montar el volumen.

Después de crear y configurar el directorio del host, el siguiente paso es montar este directorio en un contenedor.

Para vincular el montaje de un directorio de host a un contenedor, agregue la opción -v al comando podman run, especificando la ruta del directorio de host y la ruta de almacenamiento del contenedor, separadas por dos puntos (:).

```
[user@host ~]$ podman run -v /host/dir/path:/container/dir/path container/image
```

En este comando, si /container/dir/path ya existe dentro de la imagen contenedora container/image, el montaje /host/dir/path superpone, pero no elimina, el contenido de la imagen contenedora. Si se elimina el montaje, se puede volver a acceder al contenido original porque los datos se guardan en el host /host/dir/path.