



**Universidad
de Huelva**



Escuela Técnica Superior de
Ingeniería Universidad de Huelva

Memoria de Prácticas

ADMINISTRACIÓN DE SERVIDORES

Grado en Ingeniería Informática

Alejandro Gordillo Pedraza
12/12/2023

Contenido

ADMINISTRACIÓN DE SERVIDORES.....1

Grado en Ingeniería Informática1

Resumen.....3

Cápítulo 14

1. Ejercicios4

2. Conclusión..... 11

Resumen

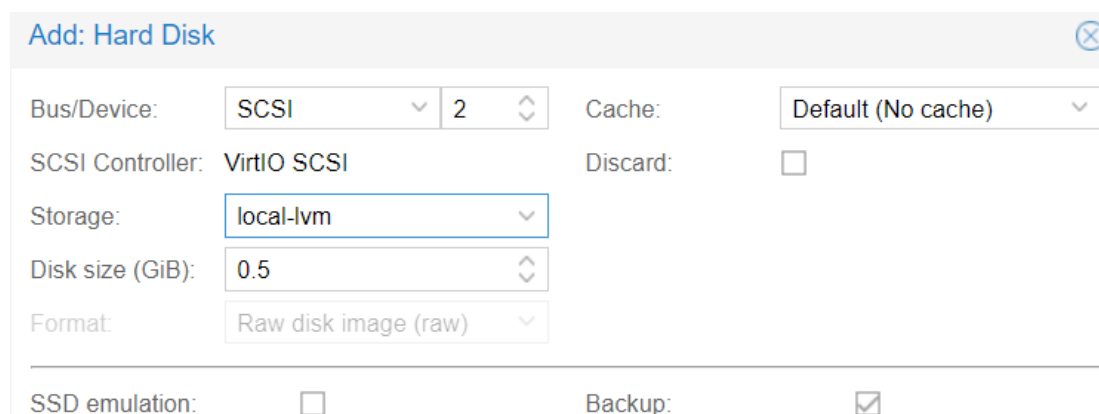
En esta práctica aprenderemos más acerca de Sistemas de Ficheros y dispositivos, para ellos necesitaremos añadir un disco duro nuevo a nuestra maquina virtual, crearle unas particiones y unas serie de modificaciones que le realizaremos a lo largo de la practica.

C  pulo 1

1. Ejercicios

Asocie a cualquiera de las m  quinas virtuales creadas un nuevo disco duro de 500 Mb.

Primero tenemos que elegir la m  quina virtual, yo en este caso voy a elegir la que tiene CentOS 7 ya que est   m  s actualizado. Una vez seleccionada, nos vamos a hardware y le damos a a  adir un nuevo disco duro. Como tiene que ser de 500Mb y solo deja poner en gigas, ponemos 0.5.



Add: Hard Disk

Bus/Device: SCSI 2 Cache: Default (No cache)

SCSI Controller: VirtIO SCSI Discard: ☐

Storage: local-lvm

Disk size (GiB): 0.5

Format: Raw disk image (raw)

SSD emulation: ☐ Backup: ☒

Cree dos particiones de 256 Mb cada una.

Para saber cu  l es el nuevo disco duro, nos vamos a `/sys/block` y ah   aparecen `sda` y `sdb`, si hacemos:

```
[root@localhost ~]# cat /sys/block/sdb/device/model
DEMU HARDDISK
```

`cat /sys/block/sdb/device/model`

Podemos ver qu   tipo es, en este caso es nuestro disco duro ya que es un hard disk y porque al ser el segundo que hemos a  adido su letra es una b. Una vez hecho esto, creamos las dos particiones con `fdisk` poniendo:

`fdisk /dev/sdb`

Y ponemos n para crear las particiones

```

troot@localhost /l# fdisk /dev/sdb
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 0x7b4fb2fb.

Orden (m para obtener ayuda): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): primary
Número de partición (1-4, default 1): 1
Primer sector (2048-1048575, valor predeterminado 2048): 2048
Last sector, +sectors or +size(K,M,G) (2048-1048575, valor predeterminado 1048575): +256M
Partition 1 of type Linux and of size 256 MiB is set

Orden (m para obtener ayuda): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): primary
Número de partición (2-4, default 2): 2
Primer sector (526336-1048575, valor predeterminado 526336): 526336
Last sector, +sectors or +size(K,M,G) (526336-1048575, valor predeterminado 1048575):
1048575
Partition 2 of type Linux and of size 255 MiB is set

```

Como podemos ver, ambas particiones se han creado con la capacidad que nos pedían. Para guardar los cambios, ponemos w.

```

Orden (m para obtener ayuda): w
¡Se ha modificado la tabla de particiones!

Llamando a ioctl() para volver a leer la tabla de particiones.
Se están sincronizando los discos.

```

En la primera partición, cree un sistema de fichero tipo ext4. En la segunda partición, cree un sistema de ficheros tipo btrfs.

Con el comando mkfs podemos crear un sistema de ficheros del tipo que le digamos, de este modo:

```
mkfs -t ext4 /dev/sdb1
```

Siendo sdb1 la primera partición y sdb2 la segunda.

Monte los sistemas de ficheros creados en /mnt/ext4 y /mnt/btrfs respectivamente.

Nos vamos a /mnt y creamos los directorios donde los vamos a montar, una vez que los tengamos creados usamos el comando:

```
mount /dev/sdb1 /mnt/ext4
```

```
[root@localhost /]# mount /dev/sdb1 /mnt/ext4
[root@localhost /]# mount /dev/sdb2 /mnt/btrfs
```

Usando el comando du y df, calcule el espacio libre y ocupado en los sistemas de ficheros creados. ¿Observa alguna diferencia entre los valores devueltos por df y du?

Dentro del directorio mnt ponemos df y du.

```
[root@localhost mnt]# df
S.ficheros          bloques de 1K   Usados Disponibles Uso% Montado en
devtmpfs             237280         0      237280    0% /dev
tmpfs                249216         0      249216    0% /dev/shm
tmpfs                249216      4576      244640    2% /run
tmpfs                249216         0      249216    0% /sys/fs/cgroup
/dev/mapper/centos-root 31433732 10764212  20669520   35% /
/dev/sda1            1038336      211064    827272    21% /boot
tmpfs                49844         0      49844     0% /run/user/0
/dev/sdb1            245671       2062     226406    1% /mnt/ext4
/dev/sdb2            261120      16704     178176    9% /mnt/btrfs

[root@localhost mnt]# du
12      ./ext4/lost+found
13      ./ext4
0       ./btrfs/snapshot
0       ./btrfs/home
16      ./btrfs
29      .
```

Como podemos ver, la primera partición usa 2062 bloques de un Kb, por lo que está ocupando un poco más de dos Mb y tiene disponibles más de 226 megas.

En cambio, la segunda partición está usando 17,7 Mb y tiene libre 178,2 Mb.

Sí hay diferencias entre df y du: En la primera partición pone en df que hay 2062 bloques usados, pero en du pone que hay en disco doce megas y esto se debe a que df solo cuenta los archivos abiertos en ese momento y du suma los archivos relacionados en cada carpeta.

Modifique las etiquetas de los sistemas de ficheros montados, las nuevas etiquetas serán, respectivamente, EXT4 Y BTRFS.

Usando la función e2label podemos modificar etiquetas, ponemos:

```
e2label /dev/sdb1/ EXT4
```

```
[root@localhost mnt]# e2label /dev/sdb1 EXT4
```

Aunque para btrfs esta función no sirve, por lo que usaremos:

```
btrfs filesystem label /mnt BTRFS
```

```
[root@localhost mnt]# btrfs filesystem label /mnt BTRFS
```

En el sistema de ficheros de tipo ext4 modifique el porcentaje reservado para el root y compruebe si se ve reflejado en los valores devueltos por df y du.

Para modificar el porcentaje utilizamos el comando

```
tune2fs -m 2 /dev/sdb1
```

```
[root@localhost /]# tune2fs -m 2 /dev/sdb1
tune2fs 1.42.9 (28-Dec-2013)
Setting reserved blocks percentage to 2% (5242 blocks)
```

```
[root@localhost /]# df
```

S.ficheros	bloques de 1K	Usados	Disponibles	Uso%	Montado en
devtmpfs	237280	0	237280	0%	/dev
tmpfs	249216	0	249216	0%	/dev/shm
tmpfs	249216	4592	244624	2%	/run
tmpfs	249216	0	249216	0%	/sys/fs/cgroup
/dev/mapper/centos-root	31433732	10766228	20667504	35%	/
/dev/sdb2	261120	16704	178176	9%	/mnt/btrfs
/dev/sdb1	245671	2062	234271	1%	/mnt/ext4
/dev/sda1	1038336	211064	827272	21%	/boot
tmpfs	49844	0	49844	0%	/run/user/0

En la partición btrfs cree dos subvolúmenes. El primero se denominará home y el segundo snapshot.

Primero, dentro de /mnt/btrfs tenemos que crear dos directorios, esto lo hacemos con mkdir, poniendo:

```
mkdir /btrfs/home
```

```
mkdir /btrfs/snapshot
```

Una vez hecho esto utilizamos la función subvolume create y ponemos dónde queremos crearlo.

```
btrfs subvolume create /mnt/btrfs/home
```

```
[root@localhost /]# cd /mnt
[root@localhost mnt]# btrfs subvolume create /mnt/btrfs/snapshot
Create subvolume '/mnt/btrfs/snapshot'
```

```
[root@localhost mnt]# btrfs subvolume create /mnt/btrfs/home
Create subvolume '/mnt/btrfs/home'
```

Modifique el fichero `/etc/fstab` para que los sistemas de ficheros creados se monten al arrancar. El sistema de ficheros ext4 se montará en `/mnt/ext4` mientras que el subvolumen home del sistema de ficheros btrfs se debe montar en home.

Para modificar el fichero ponemos:

```
vi /etc/fstab
```

Como uno es un subvolumen se hace de forma distinta. Para ambas, primero ponemos la partición que queremos montar, a continuación, ponemos dónde lo queremos montar y ponemos qué tipo de sistema de ficheros es. Luego, para ext4 ponemos que se monte automáticamente. Por último, para el subvolumen ponemos defaults.

```
#
# /etc/fstab
# Created by anaconda on Fri Nov 11 10:27:18 2022
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=517f3c3c-127a-40b8-8e7e-c2160c4815f6 /boot xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
/dev/sdb1 /mnt/ext4 ext4 auto,x-systemd.automount 0 0
/dev/sdb2 /mnt/btrfs/home btrfs defaults,subvol=home 0 0
```

Luego hacemos reboot.

Pruebe a hacer una instantánea del subvolumen home. Modifique los datos en home y restaure la instantánea antes creada.

Para crear un subvolumen se hace con la utilidad que hemos usado antes, la de subvolume, tenemos que poner:

```
btrfs subvolume snapshot /mnt/btrfs/home /mnt/btrfs/home_snapshot
```

```
[root@localhost mnt]# btrfs subvolume snapshot /mnt/btrfs/home /mnt/btrfs/home_snapshot
Create a snapshot of '/mnt/btrfs/home' in '/mnt/btrfs/home_snapshot'
```


Ahora creamos ficheros en home y restauramos la instantánea.

```
[root@localhost home]# cat > texto1.txt
hola
[root@localhost home]# cat > texto2.txt
Hoy es viernes 9 de diciembre
[root@localhost home]# cat > texto3.txt
Práctica 4 de adm de servidores
[root@localhost home]# ls
texto1.txt  texto2.txt  texto3.txt
```

```
[root@localhost /]# rsync -avz --delete /mnt/btrfs/home_snapshot /mnt/btrfs/home
sending incremental file list
home_snapshot/

sent 67 bytes  received 20 bytes  174.00 bytes/sec
total size is 0  speedup is 0.00
[root@localhost /]# cd mnt
[root@localhost mnt]# cd btrfs
[root@localhost btrfs]# cd home
[root@localhost home]# ls
home_snapshot  texto1.txt  texto2.txt  texto3.txt
```

dm-crypt es una utilidad que permite cifrar de forma transparente dispositivo de bloques. Haciendo uso de esta herramienta, cree una nueva partición que se montará dentro del directorio /privado y cuyos contenidos deben ir cifrados.

Primero creamos la partición con fdisk y el disco donde la vayamos a crear.

Luego instalamos cryptsetup y cryptmount y cargamos los módulos aed, md-crypt y sha256.

```
yum install cryptsetup
```

```
yum install cryptmount
```

```
modprobe aed
```

```
modprobe sha256
```

```
modprobe md-crypt
```

Una vez hecho esto, inicializamos la partición usando cryptsetup y le ponemos una contraseña, esta contraseña debe ser obligatoriamente robusta.

```
cryptsetup -s 512 luksFormat /dev/sdb2
```

```
[root@localhost ~]# cryptsetup -s 512 luksFormat /dev/sdb2

WARNING!
=====
Esto sobrescribirá los datos en /dev/sdb2 de forma irrevocable.

Are you sure? (Type uppercase yes): YES
Introduzca la frase contraseña de /dev/sdb2:
Verifique la frase contraseña:
```

A continuación, abrimos la partición y le damos un nombre como identificador.

```
cryptsetup luksOpen /dev/sdb2 micifrado
```

Hecho esto, se habrá creado el dispositivo /dev/mapper/micifrado y ahora tendríamos que crear el sistema de ficheros de este dispositivo.

```
mkfs.btrfs /dev/mapper/micifrado
```

```
[root@localhost ~]# mkfs.btrfs /dev/mapper/micifrado
btrfs-progs v4.9.1
See http://btrfs.wiki.kernel.org for more information.

Label:                (null)
UUID:                 e482f555-ae4b-41f3-9c6c-0f49bd8824de
Node size:            16384
Sector size:          4096
Filesystem size:      253.00MiB
Block group profiles:
  Data:               single          8.00MiB
  Metadata:           DUP             32.00MiB
  System:             DUP             8.00MiB
SSD detected:         no
Incompat features:    extref, skinny-metadata
Number of devices:    1
Devices:
  ID     SIZE  PATH
  1     253.00MiB /dev/mapper/micifrado
```

Por último, en el directorio raíz crearíamos el directorio /privado y montaríamos ahí el dispositivo.

```
mkdir /privado
```

```
mount /dev/mapper/micifrado /privado
```

2. Conclusión

En esencia lo que he podido sacar de esta práctica ha sido que me ha parecido un tema muy interesante, porque siempre viene bien saber un poco mas acerca de las modificaciones o particiones que le podemos hacer a nuestras unidades de almacenamiento. Esto es algo de lo que ya sabia un poco pero no a nivel de comandos, lo cual me ha gustado bastante aprender.

