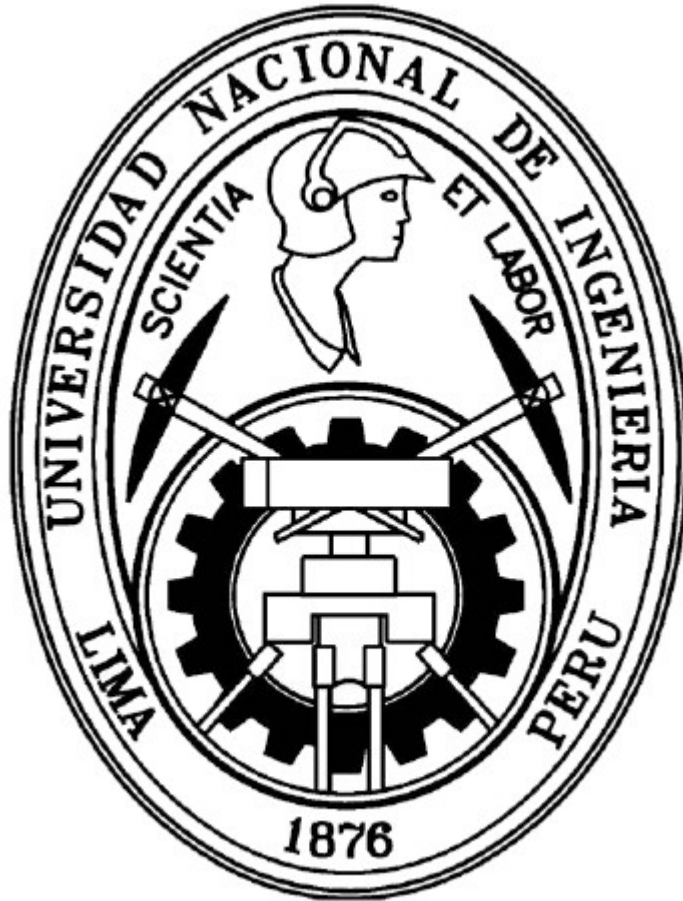


Laboratorio 5.2



Apellidos: Moreno Vera

Nombres: Felipe Adrian

Código: 20120354I

Asignatura: Administración de Redes (CC481)

2016 - I

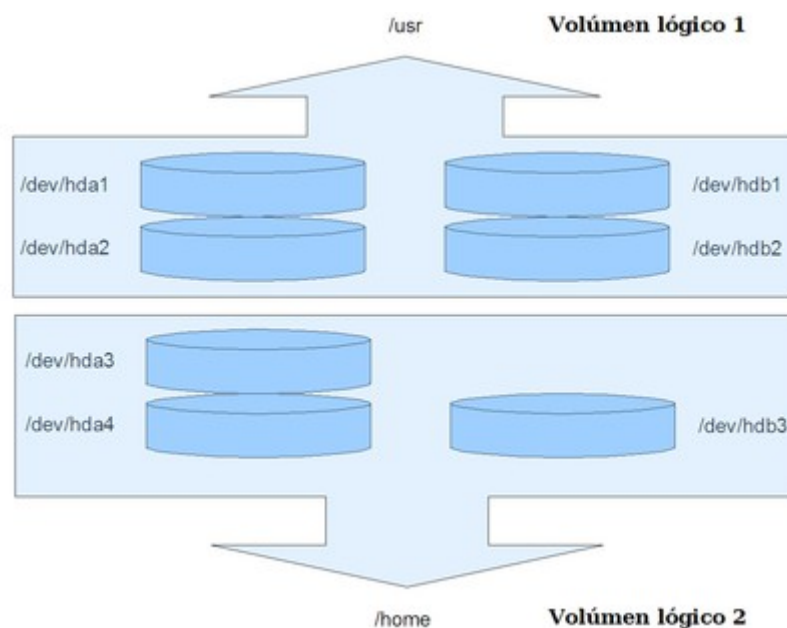
Indice

Actividad 1	(4)
Actividad 2	(4)
Actividad 3	(4)
Actividad 4	(5)
Actividad 5	(6)
Actividad 6	(8)
Actividad 7	(12)
Actividad 8	(13)
Actividad 9	(16)

Preámbulo . . .

Partición primaria: Son las divisiones crudas o primarias del disco, solo puede haber 4 de éstas o 3 primarias y una extendida. Depende de una tabla de particiones. Un disco físico completamente formateado consiste de una partición primaria que ocupa todo el espacio del disco y posee un sistema de archivos. A este tipo de particiones, prácticamente cualquier sistema operativo puede detectarlas y asignarles una unidad, siempre y cuando el sistema operativo reconozca su formato (sistema de archivos).

Partición extendida: También conocida como partición secundaria es otro tipo de partición que actúa como una partición primaria; sirve para contener múltiples unidades lógicas en su interior. Fue ideada para romper la limitación de 4 particiones primarias en un solo disco físico. Solo puede existir una partición de este tipo por disco, y solo sirve para contener particiones lógicas. Por lo tanto, es el único tipo de partición que no soporta un sistema de archivos directamente.



En la imagen de arriba podemos observar cómo trabaja LVM. Los volúmenes lógicos agrupan particiones físicas de disco, y éstos a su vez, aunque no está representado en la figura, se engloban en un grupo lógico. De esta forma, /home se compone de hda3, hda4 y hdb3, y a su vez, /usr engloba a hda1, hda2, hdb1 y hdb2.

Actividad 1

1. Recuerde y explique brevemente que era un volumen lógico, partición lógica y partición física.

Volumen lógico : Son agrupaciones de particiones lógicas que se encuentran en particiones físicas, como por ejemplo, en la partición física sda, tenemos las particiones lógicas(primaria o extendida) sda0, sda1, sda2

Partición lógica : Ocupa una porción de la partición extendida o la totalidad de la misma, la cual se ha formateado con un tipo específico de sistema de archivos (FAT32, NTFS, ext2,...) y se le ha asignado una unidad, así el sistema operativo reconoce las particiones lógicas o su sistema de archivos. Puede haber un máximo de 23 particiones lógicas en una partición extendida.

Partición física : Es La partición cruda de disco y toma las letras según su partición.

Actividad 2

1. Busque más información y explique brevemente cada concepto descrito.

Volumen Físico : son particiones del disco duro con sistema de ficheros LVM.

Volumen Lógico : Es una agrupación de volúmenes físicos, es el equivalente a una partición en un sistema tradicional de disco duro.

Grupo de volumen : es una agrupación de volúmenes físicos y volúmenes lógicos.

Actividad 3

1. Crea un PV en cada una de las particiones anteriores con el comando pvcreate.

```
[root@localhost fapCentOSuser]# pvcreate /dev/sdd1
Physical volume "/dev/sdd1" successfully created
[root@localhost fapCentOSuser]# pvcreate /dev/sdd2
Physical volume "/dev/sdd2" successfully created
[root@localhost fapCentOSuser]# pvcreate /dev/sdd3
Physical volume "/dev/sdd3" successfully created
```

2. Busque información de este comando y las posibles opciones que nos ofrece.

--force : fuerza la creación del VF sin ninguna confirmación.

--uuid XXXXXX : asigna un UUID único al VF.

--setphysicalvolumensize tamaño : sobre-escribe el tamaño automático detectado en el VF.

Actividad 4

1. Listar los PV en el sistema con el comando `pvs`, y consultar las propiedades de alguno de ellos con el comando `pvdisplay`.

`pvs`:

```
[root@localhost fapCentOSuser]# pvs
  PV          VG      Fmt  Attr  PSize   PFree
  /dev/sdd1           lvm2  ---   203,92m 203,92m
  /dev/sdd2           lvm2  ---   203,95m 203,95m
  /dev/sdd3           lvm2  ---   101,98m 101,98m
```

`pvdisplay`:

```
[root@localhost fapCentOSuser]# pvdisplay
"/dev/sdd1" is a new physical volume of "203,92 MiB"
--- NEW Physical volume ---
PV Name           /dev/sdd1
VG Name
PV Size           203,92 MiB
Allocatable       NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE      0
PV UUID          02JLAL-tBk5-qMhW-Yzb9-RnAa-P1hQ-uGb0ts

"/dev/sdd2" is a new physical volume of "203,95 MiB"
--- NEW Physical volume ---
PV Name           /dev/sdd2
VG Name
PV Size           203,95 MiB
Allocatable       NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE      0
PV UUID          HxNUff-8WhU-hPu2-rR1z-kRMx-PP8r-EEgE61

"/dev/sdd3" is a new physical volume of "101,98 MiB"
--- NEW Physical volume ---
PV Name           /dev/sdd3
VG Name
PV Size           101,98 MiB
Allocatable       NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE      0
PV UUID          PB22xA-Q47c-R1yd-JQuX-2bpH-zlpo-5lhn52
```

2. Explique cada fila lo que nos muestra dicho comando

PV Name : es el nombre de la particion fisica, esta se pone por defecto como la direccion del fichero de la particion.

VG Name : el nombre del grupo de volumen

PV Size : tamaño de la particion

Allocatable : Volumen Fisico si es asignable o no.

PE Size : numero de volúmenes logicos dentro de este volumen fisico.

Total PE : tamaño de areas fisicas en kb.

Free PE : numero total de areas fisicas libres.

Allocated PE : numeros de areas fisicas asignadas.

Actividad 5

1. Creación vgcreate. Crear un VG de nombre servidores, que incluya dos de las particiones anteriores. Estudie las diferentes opciones que nos da este comando.

```
[root@localhost fapCentOSuser]# vgcreate servidores /dev/sdd1 /dev/sdd2
Volume group "servidores" successfully created
```

2. Consulta. Comprobar las características del volumen creado con vgdisplay y listar los VGs del sistema con el comando vgs. Estudie y explique las diferentes salidas que nos ofrece el comando. Fíjese en VG Size que es de 400MB y Metadata Areas que nos indica que es 2.

Este comando vgdisplay muestra varias características del Grupo de Volumen, como su nombre, tamaño, formato, acceso, estado, etc.

```
[root@localhost fapCentOSuser]# vgdisplay
--- Volume group ---
VG Name                servidores
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                 0
Cur PV                 2
Act PV                  2
VG Size                 400,00 MiB
PE Size                 4,00 MiB
Total PE                100
Alloc PE / Size         0 / 0
Free PE / Size          100 / 400,00 MiB
VG UUID                u7B3HI-iGdy-R6Ia-Xk2c-Dmsg-xskG-mn0TEq
```

Vgs:

```
[root@localhost fapCentOSuser]# vgs
VG          #PV #LV #SN Attr   VSize   VFree
servidores   2   0   0 wz--n- 400,00m 400,00m
```

Este comando a diferencia del anterior, solo nos muestra el estado del Grupo de Volumen como la cantidad de volúmenes físicos, atributos, tamaño total y tamaño libre.

3. Reconocimiento. Se puede explorar las unidades del sistema para descubrir los VGs disponibles con el comando `vgscan`. Probar su uso. ¿Cuántas unidades le salen? A quién pertenecen las otras unidades.

Nos muestra solo el Grupo de Volumen que hemos creado.

```
[root@localhost fapCentOSuser]# vgscan
Reading all physical volumes. This may take a while...
Found volume group "servidores" using metadata type lvm2
```

4. Para extender un VG existente se usa `vgextend`. Añadir el otro de los PV creados en el apartado anterior al VG. Comprobar el resultado con los comandos `vgdisplay`, `pvddisplay`. El VG debe tener asignados ahora 3 de los 4 PV.

`vgextend`:

```
[root@localhost fapCentOSuser]# vgextend servidores /dev/sdd3
Volume group "servidores" successfully extended
```

`pvdDisplay`: vemos que antes en la etiqueta allotacable era NO, ahora aparece YES.

```
[root@localhost fapCentOSuser]# pvddisplay
--- Physical volume ---
PV Name           /dev/sdd1
VG Name           servidores
PV Size           203,92 MiB / not usable 3,92 MiB
Allocatable       yes
PE Size           4,00 MiB
Total PE          50
Free PE           50
Allocated PE      0
PV UUID           02JLAL-tBk5-qMhW-Yzb9-RnAa-P1hQ-uGb0ts

--- Physical volume ---
PV Name           /dev/sdd2
VG Name           servidores
PV Size           203,95 MiB / not usable 3,95 MiB
Allocatable       yes
PE Size           4,00 MiB
Total PE          50
Free PE           50
Allocated PE      0
PV UUID           HxNUff-8WhU-hPu2-rR1z-kRMx-PP8r-EEgE61

--- Physical volume ---
PV Name           /dev/sdd3
VG Name           servidores
PV Size           101,98 MiB / not usable 1,98 MiB
Allocatable       yes
PE Size           4,00 MiB
Total PE          25
Free PE           25
Allocated PE      0
PV UUID           PB22xA-Q47c-R1yd-JQuX-2bpH-zlpo-5lhn52
```

Vgdisplay: Antes tenía 2 metadata datas, ahora posee 3.

```
[root@localhost fapCentOSuser]# vgdisplay
--- Volume group ---
VG Name                servidores
System ID
Format                 lvm2
Metadata Areas         3
Metadata Sequence No   2
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                0
Open LV               0
Max PV                 0
Cur PV                3
Act PV                 3
VG Size                500,00 MiB
PE Size                4,00 MiB
Total PE               125
Alloc PE / Size        0 / 0
Free PE / Size         125 / 500,00 MiB
VG UUID                u7B3HI-iGdy-R6Ia-Xk2c-Dmsg-xskG-mn0TEq
```

5. Analice la salida que nos da comparándola con la que nos dió cuando fuimos creando las PV's.

Con el comando pvdisplay se observa que ahora tienen nombre de grupo de volumen, son asignables, y tienen áreas físicas determinadas.

Actividad 6

1. Creación lvcreate. Crear un LV nombre vol1 (-n) en el VG servidores y el tamaño (-L) debe ser 400M. # lvcreate {-L/--size tamaño} {-n/--name nombre_del_lv} nombre_del_vg

```
[root@localhost fapCentOSuser]# lvcreate -n vol1 -L 400M servidores
Logical volume "vol1" created.
[root@localhost fapCentOSuser]#
```

2. Consulta. Comprobar las características del volumen lógico con lvdisplay (usa como argumento el VG) y listar todos los volúmenes disponibles con lvs.

```
[root@localhost fapCentOSuser]# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync
Convert
vol1 servidores -wi-a----- 400,00m
```



```
[root@localhost fapCentOSuser]# lvsdisplay servidores
--- Logical volume ---
LV Path                /dev/servidores/vol1
LV Name                 vol1
VG Name                 servidores
LV UUID                 KfSWV2-dZ3a-h2d4-qTPI-yY1e-412n-0ZEW5j
LV Write Access         read/write
LV Creation host, time localhost.localdomain, 2016-01-29 12:57:21 -0500
LV Status                available
# open                  0
LV Size                 400,00 MiB
Current LE              100
Segments                2
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:0
```

3. Crear un sistema de ficheros (mkfs) en el LV vol1. El dispositivo (/dev) que utiliza se puede obtener de la salida del comando lvsdisplay (fijarse en la estructura creada para los dispositivos).

Creemos un sistema de ficheros tipo ext4

```
[root@localhost fapCentOSuser]# mkfs -t ext4 /dev/servidores/vol1
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
102400 nodos-i, 409600 bloques
20480 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67633152
50 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2048 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Escribiendo las tablas de nodos-i: hecho
Creating journal (8192 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho
o

Este sistema de ficheros se revisará automáticamente cada 28 montajes o
180 días, lo que suceda primero.  Utilice tune2fs -c o -i para cambiarlo.
```

4. Montar el nuevo sistema de ficheros y comprobar el tamaño disponible.

mkfs.[extension] [volumen_logico]

Creamos el directorio \$ mkdir /root/mntLabo5_2

```
[root@localhost fapCentOSuser]# mkdir /root/mntLabo5_2
[root@localhost fapCentOSuser]# ls /root/
anaconda-ks.cfg  depen.sh      install.log.syslog  prueba.sh
dependencias.sh  install.log   mntLabo5_2         prueba.sh.save
```

Ahora montamos el sistema de ficheros.

```
[root@localhost fapCentOSuser]# mount -t ext4 /dev/servidores/vol1 /root/mntLabo5_2/
[root@localhost fapCentOSuser]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/sda3              9067288 1966688    6633352  23% /
tmpfs                  515236      0     515236   0% /dev/shm
/dev/sda1              499656    50856    422588   11% /boot
/dev/mapper/servidores-vol1
                      388480     2318    365682   1% /root/mntLabo5_2
```

Comprobando:

```
[root@localhost fapCentOSuser]# fdisk -l /dev/mapper/servidores-vol1

Disco /dev/mapper/servidores-vol1: 419 MB, 419430400 bytes
255 heads, 63 sectors/track, 50 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

5. Verificamos que hemos realizado correctamente nuestro volumen. Para ello montamos el nuevo sistema de ficheros. (Comando: mount -t).

```
[root@localhost fapCentOSuser]# mount -t ext4 /dev/servidores/vol1 /root/mntLabo5_2/
mount: /dev/mapper/servidores-vol1 ya está montado o /root/mntLabo5_2/ está ocupado
mount: según mtab, /dev/mapper/servidores-vol1 ya está montado en /root/mntLabo5_2
```

6. Añadir más capacidad a un volumen:

1. Crear un volumen vol2 con el resto del VG servidores, usar para ello la opción (-l 100%FREE). (Comando: lvcreate).

```
[root@localhost fapCentOSuser]# lvcreate -n vol2 -l 100%FREE servidores
Logical volume "vol2" created.
```

2. Hacer un sistema de ficheros ext4 en servidor/vol2. Montarlo y comprobar que se ha creado correctamente.

```
[root@localhost fapCentOSuser]# mkfs -t ext4 /dev/servidores/vol2
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
25688 nodos-i, 102400 bloques
5120 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67371008
13 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
1976 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729

Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 26 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

```
[root@localhost fapCentOSuser]# mkdir /root/mnt1Labo5_2/
[root@localhost fapCentOSuser]# ls /root
anaconda-ks.cfg  depen.sh      install.log.syslog  mntLabo5_2  prueba.sh.save
dependencias.sh  install.log  mnt1Labo5_2        prueba.sh
```

```
[root@localhost fapCentOSuser]# mount -t ext4 /dev/servidores/vol2 /root/mnt1Labo5_2/
[root@localhost fapCentOSuser]# fdisk -l /dev/mapper/servidores-vol2

Disco /dev/mapper/servidores-vol2: 104 MB, 104857600 bytes
255 heads, 63 sectors/track, 12 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

3. Añadir el PV adicional que tenemos a servidor con el comando vgextend y comprobar que el resultado es exitoso.

```
[root@localhost fapCentOSuser]# vgextend servidores /dev/sdc1
Physical volume "/dev/sdc1" successfully created
Volume group "servidores" successfully extended
```

4. Desmontar vol2 (umount) y añadir el espacio adicional en servidor al LV vol2 (lvextend -l +100%FREE)

```
[root@localhost fapCentOSuser]# umount /dev/servidores/vol2
[root@localhost fapCentOSuser]# lvextend -l +100%FREE /dev/servidores/vol2
Size of logical volume servidores/vol2 changed from 100,00 MiB (25 extents) to 296,00 MiB (74 extents).
Logical volume vol2 successfully resized
```

5. Extender el sistema de ficheros con el comando resize2fs. Montar el nuevo sistema de ficheros y comprobar el resultado con pvs.

```
[root@localhost fapCentOSuser]# resize2fs -f /dev/servidores/vol2
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/servidores/vol2 to 303104 (1k) blocks.
El sistema de ficheros en /dev/servidores/vol2 tiene ahora 303104 bloques.

[root@localhost fapCentOSuser]# mount -t ext4 /dev/servidores/vol2 /root/mnt1Labo5_2/
```

```
[root@localhost fapCentOSuser]# fdisk -l /dev/mapper/servidores-vol2

Disco /dev/mapper/servidores-vol2: 310 MB, 310378496 bytes
255 heads, 63 sectors/track, 37 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

[root@localhost fapCentOSuser]# pvs
  PV          VG          Fmt Attr PSize  PFree
  /dev/sdc1   servidores lvm2 a--  196,00m    0
  /dev/sdd1   servidores lvm2 a--  200,00m    0
  /dev/sdd2   servidores lvm2 a--  200,00m    0
  /dev/sdd3   servidores lvm2 a--  100,00m    0
```

Actividad 7 (Actividad 1 de Sistemas de Ficheros Encriptados)

1. Busque información detallada sobre cryptsetup y LUKS

cryptsetup : Configura volúmenes cifrados por dm-crypt, hace la tarea mas limpia que usando las herramientas de dm_crypt.

LUKS : Es una especificación de cifrado de disco diseñado originalmente para linux. Intenta hacer una estandarización de disco por las múltiples especificaciones que existe en el mercado.

Actividad 8 (Actividad 2 de Sistemas de Ficheros Encriptados)

1. El primer paso es preparar la partición que queremos guardar encriptada. Usar una de las particiones libres (3) que tenemos disponibles.

cryptsetup luksFormat <partición>

```
[root@localhost fapCentOSuser]# cryptsetup luksFormat /dev/sdc2

WARNING!
=====
This will overwrite data on /dev/sdc2 irrevocably.

Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase:
Verify passphrase:
```

2. Podemos comprobar el estado de la partición (algoritmo de encriptación, identificador del volumen...) con `cryptsetup luksDump`.

```
[root@localhost fapCentOSuser]# cryptsetup luksDump /dev/sdc2
LUKS header information for /dev/sdc2

Version:          1
Cipher name:      aes
Cipher mode:      cbc-essiv:sha256
Hash spec:        sha1
Payload offset:   4096
MK bits:          256
MK digest:        7a 25 b1 86 10 3b 50 65 dc a1 ce ea e9 8b ea a1 d7 64 95 9f
MK salt:          ee 95 54 d7 59 49 86 a5 64 50 77 16 e3 7a 1e 6b
                  3c ce 9f 78 82 96 c3 ec 08 62 00 44 63 a4 95 f8
MK iterations:    195250
UUID:             775c8189-8a2f-45fa-9096-3f4c8cb8d55b

Key Slot 0: ENABLED
    Iterations:    781322
    Salt:          78 ab 92 c2 18 81 3d b4 4e 16 ae d8 a6 f6 e2 f2
                  3a d5 0e 05 d6 ae 7d a4 dd 36 5b bd 17 41 12 53
    Key material offset: 8
    AF stripes:    4000
Key Slot 1: DISABLED
Key Slot 2: DISABLED
Key Slot 3: DISABLED
Key Slot 4: DISABLED
Key Slot 5: DISABLED
Key Slot 6: DISABLED
Key Slot 7: DISABLED
```

3. Busque información de todo lo que nos muestra dicho comando, poniendo énfasis cuál es el algoritmo utilizado, las características y ventajas y desventajas respecto a otros.

Muestra el nombre y el modo del cifrado, la función hash y al final muestra los espacios de las claves usadas y disponibles.

Vemos en Cipher name; aes, y eso es **Advanced Encryption Standard (AES)**, también conocido como **Rijndael**, es un esquema de cifrado por bloques.

AES tiene un tamaño de bloque fijo de 128 bits y tamaños de llave de 128, 192 o 256 bits, mientras que Rijndael puede ser especificado por una clave que sea múltiplo de 32 bits, con un mínimo de 128 bits y un máximo de 256 bits.

4. Para poder acceder al sistema de ficheros debe habilitarse introduciendo la clave. El comando que usaremos es cryptsetup luksOpen. Este comando necesita el dispositivo encriptado y un nombre que se asignará al dispositivo que usaremos para montar el sistema de ficheros. Una vez disponible el dispositivo estará en /dev/mapper/<nombre>.

```
[root@localhost fapCentOSuser]# cryptsetup luksOpen /dev/sdc2 Meow
Enter passphrase for /dev/sdc2:
[root@localhost fapCentOSuser]# fdisk -l /dev/mapper/Meow

Disco /dev/mapper/Meow: 207 MB, 207618048 bytes
255 heads, 63 sectors/track, 25 cylinders
Units = cilindros of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x640935be
```

5. Explique el directorio dev/mapper y que contiene en él.

Contiene los volúmenes lógicos creados en el laboratorio pasado y la unidad cifrada.

```
[root@localhost fapCentOSuser]# ls /dev/mapper/
control Meow servidores-vol1 servidores-vol2
```


6. Dar formato y montar el sistema de ficheros encriptado.

```
[root@localhost fapCentOSuser]# mkfs -t ext4 /dev/mapper/Meow
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
50800 nodos-i, 202752 bloques
10137 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67371008
25 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2032 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729

Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 25 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

```
[root@localhost fapCentOSuser]# mount -t ext4 /dev/mapper/Meow /root/mntLabo5_2/
[root@localhost fapCentOSuser]# ls /root/mntLabo5_2/
lost+found
```

7. Una vez que terminamos, desmontar el sistema de ficheros y cerrarlo con cryptsetup luksClose.

```
[root@localhost fapCentOSuser]# umount /dev/mapper/Meow
[root@localhost fapCentOSuser]# cryptsetup luksClose Meow
```

Actividad 9 (Actividad 3 de Sistemas de Ficheros Encriptados)

1. Los sistemas de ficheros que se montan en el arranque están definidos en el fichero `/etc/fstab`. Consultar e identificar las secciones principales de este fichero (dispositivo, punto de montaje, tipo de sistema de ficheros, opciones, dump - backup, fsck - 0, 1, 2).

```
[root@localhost fapCentOSuser]# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Thu Jan 21 11:58:41 2016
#
# 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
#
UUID=a1c313e5-ebd0-49c1-af5e-cd31592bd057 / ext4 defaults
1 1
UUID=dd5ccf2a-a23d-481f-aa85-8de4c6aa3642 /boot ext4 defaults
1 2
UUID=085f1919-0348-4ff7-aea2-9fbe4212d342 swap swap defaults
0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

La primera columna muestra los dispositivos por su UUID note que los 4 últimos son virtuales, es decir solo existen para el kernel

la segunda fila son los puntos de montaje de los dispositivos

la tercera fila describe el tipo del sistema de ficheros

la cuarta fila describe las opciones de montaje asociados con el sistema de ficheros

el quinto campo es usado por estos sistemas de ficheros para determinar cual sistema de fichero necesita ser respaldado o no

la sexta y ultima columna es usado por fsck para determinar el orden en cual las revisiones del sistema de ficheros son hechos en el tiempo de reinicio del sistema.

2. Añadir los LVs que hemos creado en fstab. Se puede hacer por nombre (`/dev/mapper/nombre_vg-nombre_lv`) o por identificador de sistema de ficheros UUID (obtenerlo con `dumpe2fs`, o mediante el comando `blkid`).

Obteniendo los UUID:

```
[root@localhost fapCentOSuser]# blkid /dev/mapper/servidores-vol1
/dev/mapper/servidores-vol1: UUID="daec377c-68af-4729-8c1d-1c4c81e0732b" TYPE="ext4"
[root@localhost fapCentOSuser]# blkid /dev/mapper/servidores-vol2
/dev/mapper/servidores-vol2: UUID="4579f37d-07fa-4ad8-99eb-97b446d97102" TYPE="ext4"
```

luego añadimos a `/etc/fstab`.


```
# /etc/fstab
# Created by anaconda on Thu Jan 21 11:58:41 2016
#
# 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
#
UUID=a1c313e5-ebd0-49c1-af5e-cd31592bd057 / ext4 defaults 1 1
UUID=dd5ccf2a-a23d-481f-aa85-8de4c6aa3642 /boot ext4 defaults 1 2
UUID=085f1919-0348-4ff7-aea2-9fbe4212d342 swap swap defaults 0 0
# /dev/mapper/servidores-vol1
UUID=daec377c-68af-4729-8c1d-1c4c81e0732b /root/mntLabo5_2 ext4 defaults 0 0
# /dev/mapper/servidores-vol2
UUID=4579f37d-07fa-4ad8-99eb-97b446d97102 /root/mnt1Labo5_2 ext4 defaults 0 0

tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
```

```
[root@localhost fapCentOSuser]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda3        9067288 1994856   6605184 24% /
tmpfs            515236      0    515236  0% /dev/shm
/dev/sda1        499656    50856   422588 11% /boot
```

3. Reiniciar el sistema y comprobar que se montan los nuevos sistemas de forma automática.

```
[root@localhost ~]# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/sda3        9067288 1971944   6628096 23% /
tmpfs            515236      0    515236  0% /dev/shm
/dev/sda1        499656    50856   422588 11% /boot
/dev/mapper/servidores-vol1
                 388480     2318   365682  1% /root/mntLabo5_2
/dev/mapper/servidores-vol2
                 289770     2054   272561  1% /root/mnt1Labo5_2
```

4. mount tiene muchas opciones, comprobar las opciones en mount las generales (rw, ro, auto, noatime, noexec, nosuid, remount) del sistema ext. Estas opciones se pueden especificar en fstab. o en la línea de comandos con la opción -o (ej. mount -o remount). Consultar estas opciones.

mount -o rw : permite leer y escribir en el sistema de ficheros

mount -o ro : solo esta permitido leer sobre el sistema de ficheros

mount -o auto : puede ser montado con la opción -a

mount -o noatime : no actualiza el tiempo de acceso de todos los ficheros en el sistema de ficheros

mount -o noexec : no permite la ejecución de binarios en el sistema de ficheros montado

mount -o nosuid : no permite que los identificadores de usuario o grupo tengan efecto

mount -o remount: intenta remontar un sistema de ficheros ya montado, generalmente usado para probar otras opciones como las descritas aquí.

5. Un sistema de ficheros en un archivo:

1. Crear un fichero con el tamaño deseado (ej. 100MB)

```
# dd if=/dev/zero of=./mi_fs seek=1 bs=100M count=1
```

```
[root@localhost ~]# dd if=/dev/zero of=./miFileSystem_Meow seek=1 bs=100M count=1
1+0 records in
1+0 records out
104857600 bytes (105 MB) copied, 0,173311 s, 605 MB/s
```

2. Crear un sistema de ficheros usando como dispositivo ./mi_fs

```
[root@localhost ~]# mkfs -t ext4 /root/miFileSystem_Meow
mke2fs 1.41.12 (17-May-2010)
/root/miFileSystem_Meow no es un dispositivo especial de bloques.
¿Continuar de todas formas? (s,n) s
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
51200 nodos-i, 204800 bloques
10240 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67371008
25 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
2048 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729

Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 23 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

3. Montarlo en un directorio con la opción -o loop.

```
[root@localhost fapCentOSuser]# mount -t ext4 -o loop /root/miFileSystem_Meow /root/pruebita_Labo5_2/
[root@localhost fapCentOSuser]# df
Filesystem            1K-blocks    Used Available Use% Mounted on
/dev/sda3              9067288 2082864   6517176  25% /
tmpfs                  515236      0    515236   0% /dev/shm
/dev/sda1              499656    50856   422588  11% /boot
/dev/mapper/servidores-vol1
                      388480     2318   365682   1% /root/mntLabo5_2
/dev/mapper/servidores-vol2
                      289770     2054   272561   1% /root/mnt1Labo5_2
/root/miFileSystem_Meow
                      194241     1550   182451   1% /root/pruebita_Labo5_2
```

6. Hacer un sistema de ficheros encriptado en un fichero. Es necesario mapear el archivo en un dispositivo con el comando losetup.

Primero creamos el archivo que querramos.

```
dd if=/dev/zero of=./miFileSystem_Meow_encrypt seek=1 bs=100M count=1
```

```
[root@localhost fapCentOSuser]# dd if=/dev/zero of=./miFileSystem_Meow_encrypt seek=1 bs=100M count=1
1+0 records in
1+0 records out
104857600 bytes (105 MB) copied, 0,100897 s, 1,0 GB/s
[root@localhost fapCentOSuser]#
```

Bloqueamos el acceso al fichero: `chmod 600 /root/miFileSystem_Meow_encrypt`

Asocia un loopback con el fichero: `losetup /dev/loop0 /root/miFileSystem_Meow_encrypt`

Encriptando almacenamiento: `cryptsetup -y create miFileSystem_Meow_encrypt /dev/loop0`

verificando estado: `cryptsetup status miFileSystem_Meow_encrypt`

```
[root@localhost fapCentOSuser]# chmod 600 /root/miFileSystem_Meow_encrypt
[root@localhost fapCentOSuser]# losetup /dev/lo
log loop0 loop1 loop2 loop3 loop4 loop5 loop6 loop7
[root@localhost fapCentOSuser]# losetup /dev/loop0 /root/miFileSystem_Meow_encrypt
[root@localhost fapCentOSuser]# cryptsetup -y create miFileSystem_Meow_encrypt /dev/loop0
Enter passphrase:
Verify passphrase:
[root@localhost fapCentOSuser]# cryptsetup status miFileSystem_Meow_encrypt
/dev/mapper/miFileSystem_Meow_encrypt is inactive.
[root@localhost fapCentOSuser]#
```

Creamos un sistema de archivos

```
[root@localhost fapCentOSuser]# mkfs -t ext4 /dev/mapper/miFileSystem_Meow_encrypt
mke2fs 1.41.12 (17-May-2010)
Etiqueta del sistema de ficheros=
Tipo de SO: Linux
Tamaño del bloque=1024 (bitácora=0)
Tamaño del fragmento=1024 (bitácora=0)
Stride=0 blocks, Stripe width=0 blocks
25688 nodos-i, 102400 bloques
5120 bloques (5.00%) reservados para el superusuario
Primer bloque de datos=1
Número máximo de bloques del sistema de ficheros=67371008
13 bloque de grupos
8192 bloques por grupo, 8192 fragmentos por grupo
1976 nodos-i por grupo
Respaldo del superbloque guardado en los bloques:
      8193, 24577, 40961, 57345, 73729

Escribiendo las tablas de nodos-i: hecho
Creating journal (4096 blocks): hecho
Escribiendo superbloques y la información contable del sistema de ficheros: hecho

Este sistema de ficheros se revisará automáticamente cada 21 montajes o
180 días, lo que suceda primero. Utilice tune2fs -c o -i para cambiarlo.
```

Montando: mount /dev/mapper/miFileSystem_Meow_encrypt /root/mntEncryptLabo5_2
y verificando con df

```
[root@localhost fapCentOSuser]# mkdir /root/mntEncryptLabo5_2
[root@localhost fapCentOSuser]# mount /dev/mapper/miFileSystem_Meow_encrypt /root/mntEncryptLabo5_2/
[root@localhost fapCentOSuser]# df
```

Filesystem	1K-blocks	Used	Available	Use%	Mounted on
/dev/sda3	9067288	2193036	6407004	26%	/
tmpfs	515236	0	515236	0%	/dev/shm
/dev/sda1	499656	50856	422588	11%	/boot
/dev/mapper/servidores-vol1	388480	2318	365682	1%	/root/mntLabo5_2
/dev/mapper/servidores-vol2	289770	2054	272561	1%	/root/mnt1Labo5_2
/dev/mapper/miFileSystem_Meow_encrypt	95054	1550	88384	2%	/root/mntEncryptLabo5_2

Referencias:

https://es.wikipedia.org/wiki/Sistema_de_archivos
https://es.wikipedia.org/wiki/Partici%C3%B3n_de_disco
https://es.wikipedia.org/wiki/Tabla_de_particiones
https://es.wikipedia.org/wiki/Logical_Volume_Manager#Creaci.C3.B3n_del_LV
<http://www.saulo.net/pub/ddypart/a.htm>
<https://wiki.centos.org/es/TipsAndTricks/EncryptedFilesystem>