

SAN con iSCSI y sistemas de ficheros OCFS2

Tabla de contenidos

- Instalación del entorno de prácticas
- Comprobación de recursos
- DISCOS
 - RAID5
 - Preparación de los dispositivos
 - Creación del RAID5
 - LVM
 - Creación del volumen físico
 - Creación del grupo de volúmenes
 - Creación de los volúmenes lógicos
 - iSCSI
 - Creación de targets
 - Creación de LUNs
 - Control de accesos
- CLIENTES
 - iSCSI
 - Descubriendo targets
 - Conexión con los volúmenes lógicos
 - Formateo de los volúmenes lógicos privados
 - Formateo del volumen lógico compartido

- Montaje de los volúmenes lógicos
- Bibliografía

Instalación del entorno de prácticas

Iniciamos el autoinstalador para Linux

```
curl -o- http://ccia.esei.uvigo.es/docencia/CDA/1819/practicas//ejercicio-iscsi.sh | bash -
```

Nos mandará poner un identificador único. Después de esto, se nos abrirá nuestro nuevo entorno de pruebas.

Para poder loguearnos en los sistema deberemos introducir el usuario **root** junto a la contraseña **purple**.

Si no se nos inicia el entorno gráfico, deberemos ejecutar el siguiente comando:

```
startx
```

Una vez tengamos nuestro entorno preparado, procederemos a ejecutar la aplicación **LXTerminal** en cada máquina.

Comprobación de recursos

Comprobamos los discos disponibles con cualquiera de los siguientes comandos en la máquina **DISCOS** (usaremos esta máquina hasta que se indique lo contrario):

```
lsblk  
fdisk -l  
parted -l
```

Personalmente prefiero **lsblk**. Una vez ejecutado ésta es su salida:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdc	8:32	0	100M	0	disk	
sdd	8:48	0	100M	0	disk	
sde	8:64	0	100M	0	disk	
sdf	8:80	0	100M	0	disk	

Como podemos comprobar, el sistema está montado sobre el disco **sda** y la partición SWAP sobre el **sdb**, pero disponemos a mayores cuatro discos para poder trabajar a gusto.

RAID5

Procederemos con la creación de un RAID5 utilizando los dispositivos `/dev/sdc1`, `/dev/sdd1`, `/dev/sde1` y `/dev/sdf1`.

Preparación de los dispositivos

Primeramente, crearemos una partición primaria en los dispositivos `/dev/sdc`, `/dev/sdd`, `/dev/sde` y `/dev/sdf` asignándole todo el espacio disponible usando el comando `parted`.

```
parted /dev/sdc
(parted) mklabel msdos
(parted) mkpart primary 1M 100%
(parted) set 1 raid on
(parted) quit
# Repetir el proceso usando /dev/sdd, /dev/sde y /dev/sdf
```

Comprobamos el resultado ejecutando `lsblk` de nuevo.

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdc	8:32	0	100M	0	disk	

└sdc1	8:33	0	99M	0	part
sdd	8:48	0	100M	0	disk
└sdd1	8:49	0	99M	0	part
sde	8:64	0	100M	0	disk
└sde1	8:65	0	99M	0	part
sdf	8:80	0	100M	0	disk
└sdf1	8:81	0	99M	0	part

Creación del RAID5

Una vez creados los dispositivos necesarios para nuestro RAID, usaremos `mdadm` para construir y gestionar nuestro RAID5. En concreto ejecutaremos este comando:

```
mdadm --create --verbose /dev/md/md_RAID5 --level=raid5 --  
raid-devices=4 \  
/dev/sdc1 /dev/sdd1 /dev/sde1 /dev/sdf1
```

El cual es bastante intuitivo y no necesita demasiada explicación:

```
mdadm [--create: Crear RAID] \  
[--verbose: Muestra logs por pantalla] \  
[/dev/md/md_RAID5: Crear el RAID en dicha ruta] \  
[-level=raid5: Tipo de RAID] \  
[--raid-devices=3: Utilizar 4 dispositivos para montarlo]  
\  

```

```
[/dev/sdc1 /dev/sdd1 /dev/sde1 /dev/sdf1: Dispositivos disponibles para ser usados]
```

Se nos notificará del éxito de la operación con el siguiente mensaje:

```
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: size set to 100352K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md/md_RAID5 started.
```

Si ejecutamos de nuevo `lsblk` comprobaremos la creación del RAID

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdc	8:32	0	100M	0	disk	
└─sdc1	8:33	0	99M	0	part	
└─md127	9:127	0	294M	0	raid5	
sdd	8:48	0	100M	0	disk	
└─sdd1	8:49	0	99M	0	part	
└─md127	9:127	0	294M	0	raid5	
sde	8:64	0	100M	0	disk	
└─sde1	8:65	0	99M	0	part	

└md127	9:127	0	294M	0	raid5
sdf	8:80	0	100M	0	disk
└sdf1	8:81	0	99M	0	part
└md127	9:127	0	294M	0	raid5

Si necesitamos información extra, podremos obtenerla de la siguiente manera:

```
mdadm --detail /dev/md/md_RAID5
```

Salida:

/dev/md/md_RAID5:
Version : 1.2
Creation Time : Sat Oct 20 19:47:01 2018
Raid Level : raid5
Array Size : 301056 (294.00 MiB 308.28 MB)
Used Dev Size : 100352 (98.00 MiB 102.76 MB)
Raid Devices : 4
Total Devices : 4
Persistence : Superblock is persistent
Update Time : Sat Oct 20 19:47:03 2018
State : clean
Active Devices : 4
Working Devices : 4
Failed Devices : 0

Spare Devices : 0

Layout : left-symmetric

Chunk Size : 512K

Name : discos.cda.net:md_RAID5 (local to host discos.cda.net)

UUID : 714941af:9ae82212:a280cbd9:473bb324

Events : 18

Number	Major	Minor	RaidDevice	State	
0	8	33	0	active sync	/dev /sdc1
1	8	49	1	active sync	/dev /sdd1
2	8	65	2	active sync	/dev /sde1
4	8	81	3	active sync	/dev /sdf1

LVM

Procederemos con la creación de un grupo de volúmenes LVM utilizando el RAID5 que constará de tres volúmenes lógicos llamados UNO , DOS , COMPARTIDO .

Creación del volumen físico

Lo primero será definir nuestro volumen físico en el RAID5 para crear posteriormente los volúmenes lógicos:

```
pvccreate /dev/md/md_RAID5
```

Se nos mostrará el mensaje **Physical volume "/dev/md/md_RAID5" successfully created.** si tuvimos éxito.

Creación del grupo de volúmenes

Para construir un grupo de volúmenes ejecutamos el siguiente comando.

```
vgcreate homelab /dev/md/md_RAID5
```

La salida **Volume group "homelab" successfully created** nos indica que la operación se ha efectuado correctamente.

Creación de los volúmenes lógicos

Es hora de gestionar nuestro espacio. Usaremos el comando **lvcreate** para generar volúmenes lógicos en nuestro grupo de volúmenes lógicos **homelab**. Cada uno de ellos tendrá 50MB y su correspondiente nombre como parámetros.

```
lvcreate homelab -L 50MB -n UNO
lvcreate homelab -L 50MB -n DOS
lvcreate homelab -L 50MB -n COMPARTIDO
```

Sabremos que hemos tenido éxito al ver la siguiente salida:

```
Rounding up size to full physical extent 52,00 MiB
  Logical volume "UNO" created.
Rounding up size to full physical extent 52,00 MiB
  Logical volume "DOS" created.
Rounding up size to full physical extent 52,00 MiB
  Logical volume "COMPARTIDO" created.
```

Por si acaso podemos usar la ya más que conocida herramienta `lsblk` para realizar la comprobación:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPO
INT						
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdC	8:32	0	100M	0	disk	
└─sdC1	8:33	0	99M	0	part	
└─md127	9:127	0	294M	0	raid5	
└─└─homelab-UNO	253:0	0	52M	0	lvm	

└─homelab-DOS	253:1	0	52M	0	lvm
└─homelab-COMPARTIDO	253:2	0	52M	0	lvm
sdd	8:48	0	100M	0	disk
└─sdd1	8:49	0	99M	0	part
└─md127	9:127	0	294M	0	raid5
└─homelab-UN0	253:0	0	52M	0	lvm
└─homelab-DOS	253:1	0	52M	0	lvm
└─homelab-COMPARTIDO	253:2	0	52M	0	lvm
sde	8:64	0	100M	0	disk
└─sde1	8:65	0	99M	0	part
└─md127	9:127	0	294M	0	raid5
└─homelab-UN0	253:0	0	52M	0	lvm
└─homelab-DOS	253:1	0	52M	0	lvm
└─homelab-COMPARTIDO	253:2	0	52M	0	lvm
sdf	8:80	0	100M	0	disk
└─sdf1	8:81	0	99M	0	part
└─md127	9:127	0	294M	0	raid5
└─homelab-UN0	253:0	0	52M	0	lvm
└─homelab-DOS	253:1	0	52M	0	lvm
└─homelab-COMPARTIDO	253:2	0	52M	0	lvm

iSCSI

Es el momento de gestionar nuestros volúmenes lógicos para crear un sistema de archivos iSCSI.

Creación de targets

Crearemos un target usando un iqn y un tid para identificarlo:

```
tgtadm --lld iscsi --mode target --op new --tid=1 --target  
name iqn.2018-09.net.cda.discos:homelab.UNO
```

El comando se compone de los siguientes argumentos:

```
tgtadm [--lld iscsi: Driver/Tipo de almacenamiento] \  
[--mode target: Especificación de objetivo (logicalunit, t  
arget...)] \  
[--op new: Operación deseada - Crear] \  
[--tid=1: Identificador (0 reservado)] \  
[--targetname iqn.2018-09.net.cda.discos:homelab.UNO: Nomb  
re siguiendo el estándar]
```

De esta misma forma, crearemos otros dos nuevos targets con los tid 2 y 3:

```
tgtadm --lld iscsi --mode target --op new --tid=2 \  
--targetname iqn.2018-09.net.cda.discos:homelab.DOS  
  
tgtadm --lld iscsi --mode target --op new --tid=3 \  
--targetname iqn.2018-09.net.cda.discos:homelab.COMPARTIDO
```

Podemos confirmar que todo ha salido bien ejecutando la operación

`show` del comando `tgtadm`:

```
tgtadm --lld iscsi --mode target --op show
```

Nos dará una salida similar a esta:

```
Target 1: iqn.2018-09.net.cda.discos:homelab.UNO
```

```
System information:
```

```
Driver: iscsi
```

```
State: ready
```

```
I_T nexus information:
```

```
LUN information:
```

```
LUN: 0
```

```
Type: controller
```

```
SCSI ID: IET 00010000
```

```
SCSI SN: beaf10
```

```
Size: 0 MB, Block size: 1
```

```
Online: Yes
```

```
Removable media: No
```

```
Prevent removal: No
```

```
Readonly: No
```

```
SWP: No
```

```
Thin-provisioning: No
```

```
Backing store type: null
```

```
Backing store path: None
```

```
Backing store flags:
```

Account information:

ACL information:

Target 2: iqn.2018-09.net.cda.discos:homelab.DOS

System information:

Driver: iscsi

State: ready

I_T nexus information:

LUN information:

LUN: 0

Type: controller

SCSI ID: IET 00020000

SCSI SN: beaf20

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

Account information:

ACL information:

Target 3: iqn.2018-09.net.cda.discos:homelab.COMPARTIDO

System information:

Driver: iscsi

State: ready

I_T nexus information:

LUN information:

LUN: 0

Type: controller

SCSI ID: IET 00030000

SCSI SN: beaf30

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

Account information:

ACL information:

Creación de LUNs

Para la exposición de dispositivos usando LUN, el comando es muy similar al anterior, con la diferencia de que deberemos *bindear* Unidades Lógicas a Targets de la siguiente manera:

```
tgtadm --lld iscsi --mode logicalunit --op new --tid 1 --lun 1 --backing-store /dev/homelab/UN0
```

Del mismo modo asignaremos dispositivos de almacenamiento al resto de targets:

```
tgtadm --lld iscsi --mode logicalunit --op new --tid 2 --lun 2 --backing-store /dev/homelab/DOS
```

```
tgtadm --lld iscsi --mode logicalunit --op new --tid 3 --lun 3 --backing-store /dev/homelab/COMPARTIDO
```

Para confirmar que todo ha salido bien, ejecutamos la operación **show** de nuevo y nos fijamos en el **Backing storage path**:

```
Target 1: iqn.2018-09.net.cda.discos:homelab.UN0
```

```
System information:
```

```
Driver: iscsi
```

```
State: ready
```

```
I_T nexus information:
```

```
LUN information:
```

```
LUN: 0
```

```
Type: controller
```

```
SCSI ID: IET 00010000
```

```
SCSI SN: beaf10
```

```
Size: 0 MB, Block size: 1
```

```
Online: Yes
```

```
Removable media: No
```

```
Prevent removal: No
```


Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

LUN: 1

Type: disk

SCSI ID: IET 00010001

SCSI SN: beaf11

Size: 55 MB, Block size: 512

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: rdwr

Backing store path: /dev/homelab/UN0

Backing store flags:

Account information:

ACL information:

Target 2: iqn.2018-09.net.cda.discos:homelab.D0S

System information:

Driver: iscsi

State: ready

I_T nexus information:

LUN information:

LUN: 0

Type: controller

SCSI ID: IET 00020000

SCSI SN: beaf20

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

LUN: 2

Type: disk

SCSI ID: IET 00020002

SCSI SN: beaf22

Size: 55 MB, Block size: 512

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: rdwr

Backing store path: /dev/homelab/DOS

Backing store flags:

Account information:

ACL information:

Target 3: iqn.2018-09.net.cda.discos:homelab.COMPARTIDO

System information:

Driver: iscsi

State: ready

I_T nexus information:

LUN information:

LUN: 0

Type: controller

SCSI ID: IET 00030000

SCSI SN: beaf30

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

LUN: 3

Type: disk

SCSI ID: IET 00030003

SCSI SN: beaf33

Size: 55 MB, Block size: 512

Online: Yes

Removable media: No
Prevent removal: No
Readonly: No
SWP: No
Thin-provisioning: No
Backing store type : rdwr
Backing store path: /dev/home1ab/COMPARTIDO
Backing store flags:
Account information:
ACL information:

Control de accesos

Como nos interesa que cada volúmen lógico sea accedido según la siguiente tabla, debemos restringir el acceso mediante IP:

IP	Máquina
192.168.100.11	Discos
192.168.100.22	Cliente1
192.168.100.33	Cliente2

LV	Propietario(s)
UNO	Cliente1
DOS	Cliente2
COMPARTIDO	Ambos

Para ello, ejecutaremos la operación `bind` del comando `tgtadm` para conceder acceso a las direcciones IP especificadas:

```
# UNO
tgtadm --lld iscsi --mode target --op bind --tid 1 --initiator-address 192.168.100.22

# DOS
tgtadm --lld iscsi --mode target --op bind --tid 2 --initiator-address 192.168.100.33

# COMPARTIDO
tgtadm --lld iscsi --mode target --op bind --tid 3 --initiator-address 192.168.100.22
tgtadm --lld iscsi --mode target --op bind --tid 3 --initiator-address 192.168.100.33
```

De nuevo, podemos comprobar que todo ha salido bien si ejecutamos la operación `show` del comando `tgtadm`. Esta vez deberemos fijarnos en el apartado `ACL Information`:

```
Target 1: iqn.2018-09.net.cda.discos:homelab.UNO
  System information:
    Driver: iscsi
    State: ready
  I_T nexus information:
  LUN information:
    LUN: 0
    Type: controller
```

SCSI ID: IET 00010000

SCSI SN: beaf10

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

LUN: 1

Type: disk

SCSI ID: IET 00010001

SCSI SN: beaf11

Size: 55 MB, Block size: 512

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: rdwr

Backing store path: /dev/homelab/UN0

Backing store flags:

Account information:

ACL information:

192.168.100.22

Target 2: iqn.2018-09.net.cda.discos:homelab.D0S

System information:

Driver: iscsi

State: ready

I_T nexus information:

LUN information:

LUN: 0

Type: controller

SCSI ID: IET 00020000

SCSI SN: beaf20

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null

Backing store path: None

Backing store flags:

LUN: 2

Type: disk

SCSI ID: IET 00020002

SCSI SN: beaf22

Size: 55 MB, Block size: 512

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: rdwr

Backing store path: /dev/homelab/DOS

Backing store flags:

Account information:

ACL information:

192.168.100.33

Target 3: iqn.2018-09.net.cda.discos:homelab.COMPARTIDO

System information:

Driver: iscsi

State: ready

I_T nexus information:

LUN information:

LUN: 0

Type: controller

SCSI ID: IET 00030000

SCSI SN: beaf30

Size: 0 MB, Block size: 1

Online: Yes

Removable media: No

Prevent removal: No

Readonly: No

SWP: No

Thin-provisioning: No

Backing store type: null


```
Backing store path: None
Backing store flags:
LUN: 3
Type: disk
SCSI ID: IET      00030003
SCSI SN: beaf33
Size: 55 MB, Block size: 512
Online: Yes
Removable media: No
Prevent removal: No
Readonly: No
SWP: No
Thin-provisioning: No
Backing store type: rdwr
Backing store path: /dev/homelab/COMPARTIDO
Backing store flags:
Account information:
ACL information:
192.168.100.22
192.168.100.33
```

CLIENTES

Descubriendo targets

Lo primero que deberemos hacer en las máquinas cliente será

inspeccionar los targets disponibles en el servidor. Para ello ejecutaremos el siguiente comando:

```
iscsiadm --mode discovery --type sendtargets --portal 192.168.100.11
```

Nos mostrará lo siguiente según el cliente:

```
# Cliente1
192.168.100.11:3260,1 iqn.2018-09.net.cda.discos:homelab.U
NO
192.168.100.11:3260,1 iqn.2018-09.net.cda.discos:homelab.C
OMPARTIDO
# Cliente2
192.168.100.11:3260,1 iqn.2018-09.net.cda.discos:homelab.D
OS
192.168.100.11:3260,1 iqn.2018-09.net.cda.discos:homelab.C
OMPARTIDO
```

En cualquier momento podremos consultar estos targets recién descubiertos con el comando `iscsiadm --mode node`.

También tendremos información extra en el directorio `/etc/iscsi/nodes/`, donde se encuentran los archivos de configuración de los nodos descubiertos.

Conexión con los volúmenes lógicos

Antes de realizar cualquier tipo de conexión, comprobamos los recursos de nuestros clientes con el comando `lsblk`:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]

Podemos observar que además de los ocupados por la raíz y la partición SWAP no disponemos de más dispositivos.

Procedemos a conectarnos a cualquiera de los volúmenes individuales con el siguiente comando, según la máquina cliente:

```
# Cliente1
iscsiadm --m node --targetname iqn.2018-09.net.cda.disco
s:homelab.UNO --portal 192.168.100.11 --login

# Cliente2
iscsiadm --m node --targetname iqn.2018-09.net.cda.disco
s:homelab.DOS --portal 192.168.100.11 --login

# AMBOS
iscsiadm --m node --targetname iqn.2018-09.net.cda.disco
s:homelab.COMPARTIDO --portal 192.168.100.11 --login
```

Nos mostrará un mensaje similar a este si todo ha salido de la manera correcta:

```
# Cliente1
Login to [iface: default, target: iqn.2018-09.net.cda.disc
os:homelab.UNO, portal: 192.168.100.11,3260] successful.
# Cliente2
Login to [iface: default, target: iqn.2018-09.net.cda.disc
os:homelab.DOS, portal: 192.168.100.11,3260] successful.
# Ambos
Logging in to [iface: default, target: iqn.2018-09.net.cda
.discos:homelab.COMPARTIDO, portal: 192.168.100.11,3260] (
multiple)
Login to [iface: default, target: iqn.2018-09.net.cda.disc
os:homelab.COMPARTIDO, portal: 192.168.100.11,3260] succes
sful.
```

Si comprobamos los recursos disponibles con `lsblk` veremos que contamos con discos nuevos:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/
sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdc	8:32	0	52M	0	disk	# UNO/DOS
sdd	8:48	0	52M	0	disk	# COMPARTIDO

Formateo de los volúmenes lógicos privados

Para poder usar estos nuevos dispositivos, lo primero será formatear el disco privado (`sdc`) en `ext3` . Para ello usaremos el comando:

```
mkfs.ext3 /dev/sdc
```

La salida debería ser algo similar a esto:

```
mke2fs 1.43.4 (31-Jan-2017)
```

```
Se está creando un sistema de ficheros con 53248 bloques d  
e 1k y 13328
```

```
nodos-i
```

```
UUID del sistema de ficheros: 8c0221d0-1583-4f6c-94cf-f2ee  
a444a059
```

```
Respaldo del superbloque guardado en los bloques:
```

```
8193, 24577, 40961
```

```
Reservando las tablas de grupo: hecho
```

```
Escribiendo las tablas de nodos-i: hecho
```

```
Creando el fichero de transacciones (4096 bloques): hecho
```

```
Escribiendo superbloques y la información contable del sis
```

tema de ficheros:

hecho

Formateo del volumen lógico compartido

Una vez formateados los discos privados, procederemos a formatear el volumen lógico compartido. Para evitar errores de uso compartido, levantaremos un servicio OSCF2.

El primer paso es realizar los siguientes comandos en ambos clientes para crear el clúster:

```
o2cb add-cluster homelab
o2cb add-node homelab cliente1 --ip 192.168.100.22
o2cb add-node homelab cliente2 --ip 192.168.100.33
```

Posteriormente deberemos editar a mano los ficheros

`/etc/default/o2cb` de cada cliente para habilitar el clúster y asignarle el nombre correcto, de tal modo que nos quede así:

```
# O2CB_ENABLED: 'true' means to load the driver on boot.
O2CB_ENABLED=true

# O2CB_BOOTCLUSTER: If not empty, the name of a cluster to
start.
```

```
O2CB_BOOTCLUSTER=homelab
```

Reiniciar el servicio `o2cb` para que se realicen los cambios:

```
service o2cb restart
service o2cb status
```

Como podemos comprobar, el servicio se ha levantado correctamente:

```
● o2cb.service - LSB: Load O2CB cluster services at system
   boot.
     Loaded: loaded (/etc/init.d/o2cb; generated; vendor pre
   set: enabled)
     Active: active (running) since Sun 2018-10-21 13:06:18
   CEST; 5s ago
     Docs: man:systemd-sysv-generator(8)
    Process: 1906 ExecStop=/etc/init.d/o2cb stop (code=exite
   d, status=0/SUCCESS)
    Process: 1931 ExecStart=/etc/init.d/o2cb start (code=exi
   ted, status=0/SUCCESS)
     Tasks: 1 (limit: 4915)
    CGroup: /system.slice/o2cb.service
           └─1981 o2hbmonitor

oct 21 13:06:18 cliente1.cda.net systemd[1]: Starting LSB:
   Load O2CB cluster ser
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Loading stack
```

```
plugin "o2cb": OK
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Loading files
system "ocfs2_dlmfs": 0
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Creating dire
ctory '/dlm': OK
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Mounting ocfs
2_dlmfs filesystem at
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Setting clust
er stack "o2cb": OK
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Registering 0
2CB cluster "homelab":
oct 21 13:06:18 cliente1.cda.net o2cb[1931]: Setting 02CB
cluster timeouts : OK
oct 21 13:06:18 cliente1.cda.net systemd[1]: Started LSB:
Load 02CB cluster serv
oct 21 13:06:18 cliente1.cda.net o2hbmonitor[1981]: Starti
ng
```

Va quedando menos...

Desde uno de los clientes, formatearemos el volumen lógico
COMPARTIDO. En mi caso usaré el **Cliente1**:

```
mkfs.ocfs2 /dev/sdd
```

El formateo se realizará con éxito si vemos esto:

mkfs.ocfs2 1.8.4

Cluster stack: classic o2cb

Label:

Features: sparse extended-slotmap backup-super unwritten i
nline-data strict-journal-super xattr indexed-dirs refcoun
t discontig-bg

Block size: 1024 (10 bits)

Cluster size: 4096 (12 bits)

Volume size: 54525952 (13312 clusters) (53248 blocks)

Cluster groups: 2 (tail covers 5632 clusters, rest cover 7
680 clusters)

Extent allocator size: 0 (0 groups)

Journal size: 4194304

Node slots: 2

Creating bitmaps: done

Initializing superblock: done

Writing system files: done

Writing superblock: done

Writing backup superblock: 0 block(s)

Formatting Journals: done

Growing extent allocator: done

Formatting slot map: done

Formatting quota files: done

Writing lost+found: done

mkfs.ocfs2 successful

Montaje de los volúmenes lógicos

Ahora sólo falta probar que todo funcione correctamente.

Para montar los discos privados usaremos el siguiente comando:

```
# Cliente1
mkdir /mnt/uno
mount -t ext3 /dev/sdc /mnt/uno

# Cliente2
mkdir /mnt/dos
mount -t ext3 /dev/sdc /mnt/dos
```

Para montar el disco compartido, por el contrario, escribiremos el siguiente comando:

```
mkdir /mnt/compartido
mount -t ocfs2 /dev/sdd /mnt/compartido
```

Para comprobar que todo está correcto, como de costumbre, utilizaremos `lsblk`:

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	16G	0	disk	
└─sda1	8:1	0	16G	0	part	/

sdb	8:16	0	1G	0	disk	
└─sdb1	8:17	0	1022M	0	part	[SWAP]
sdc	8:32	0	52M	0	disk	/mnt/{uno,dos}
sdd	8:48	0	52M	0	disk	/mnt/compartido

Ya sólo queda probar si el disco compartido funciona como debería.

Para ello creamos un archivo en dicho disco desde el **Cliente1**:

```
touch /mnt/compartido/"Esto se merece un 10"
```

¿Será accesible en tiempo real desde el **Cliente2**? ¿Se habrá sincronizado? Lo comprobamos con el comando `ls -l /mnt/compartido`:

```
total 1
-rw-r--r-- 1 root root 0 oct 21 13:17 Esto se merece un
10
drwxr-xr-x 2 root root 824 oct 21 13:10 lost+found
```

¡Funciona! ¡Ya tenemos nuestro sistema de archivos compartido!

Bibliografía

- <https://github.com/Student-Puma/HomeLab>
- [\[1\]](#)

<http://ccia.esei.uvigo.es/docencia/CDA/1819/practicas/ejercicio-iscsi/ejercicio-iscsi.html>