

PRÀCTICA 3:

Arquitectures d'emmagatzament

Nom 1: David Morillo Massagué

NIU 1: 1666540

Nom 2: Adrià Muro Gómez

NIU 2: 1665191

Usuari utilitzat a la pràctica: gixpd-ged-2

Taula de continguts

Introducció.....	3
Objectius.....	3
Desenvolupament.....	4
1. Preparació de l'entorn a OpenNebula per a RAID.....	4
2. RAID 5.....	6
Crear particions i formatejar.....	6
Crear RAID 5.....	8
3. RAID 1.....	12
Crear particions i formatejar.....	12
Crear RAID 1.....	12
4. Configuració de GlusterFS a MVA i MVB:.....	14
5. Configuració de LVM:.....	16
6. Resultats:.....	20
6.1 Rendiment RAID (1 i 5).....	20
6.2 Fiabilitat RAID (1 i 5).....	20
6.3 Rendiment GlusterFS.....	21
6.4 Fiabilitat GlusterFS.....	22
6.5 Rendiment LVM.....	23
6.6 Fiabilitat LVM.....	23
Conclusions.....	26
Fiabilitat.....	26
Extensibilitat.....	26
Rendiment.....	26

Introducció

En aquesta pràctica, hem configurat i provat diverses tecnologies de gestió i optimització d'emmagatzematge de dades, incloent **RAID**, **ClusterFS** i **LVM**. Aquest procés inclou la configuració i instal·lació de cada tecnologia, així com la implementació de diferents nivells de RAID, la creació d'un sistema de fitxers distribuït amb ClusterFS, i la gestió flexible de volums lògics amb LVM.

Objectius

L'objectiu principal d'aquesta pràctica és adquirir experiència i familiaritzar-nos amb la configuració i gestió d'eines d'emmagatzematge de dades, que permeten augmentar la flexibilitat, la redundància i l'escalabilitat. Els objectius específics són:

- Configurar diferents nivells de RAID per millorar la redundància i el rendiment: Implementar configuracions RAID 1 i 5 per comprendre de primera mà els avantatges i limitacions de cada nivell en la gestió de dades.
- Desplegar un sistema de fitxers distribuït amb ClusterFS: Crear un sistema de fitxers amb ClusterFS que permeti l'accés i gestió de fitxers des de múltiples nodes, millorant la disponibilitat de dades en entorns distribuïts.
- Utilitzar LVM per a una gestió flexible de l'emmagatzematge: Configurar LVM per gestionar volums lògics que es poden redimensionar dinàmicament, facilitant així la gestió i l'escalabilitat de l'espai d'emmagatzematge.
- Aplicar bones pràctiques en la configuració d'emmagatzematge: Seguir recomanacions de configuració per garantir la seguretat, eficiència i mantenibilitat de les dades en entorns d'emmagatzematge empresarial.

Desenvolupament

1. Preparació de l'entorn a OpenNebula per a RAID

Abans de crear el sistema RAID a la màquina virtual, comprovem l'estat dels discos actual de la màquina per a evitar futures confusions:

```

root@adminip:~# lsblk
NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
loop0 7:0 0 4K 1 loop /snap/bare/5
loop1 7:1 0 74.2M 1 loop /snap/core22/1621
loop2 7:2 0 73.9M 1 loop /snap/core22/1663
loop3 7:3 0 269.8M 1 loop /snap/firefox/4698
loop4 7:4 0 273.6M 1 loop /snap/firefox/5187
loop5 7:5 0 505.1M 1 loop /snap/gnome-42-2204/176
loop6 7:6 0 91.7M 1 loop /snap/gtk-common-themes/1535
loop7 7:7 0 38.8M 1 loop /snap/snapd/21759
sda 8:0 0 40G 0 disk
├─sda1 8:1 0 1M 0 part
└─sda2 8:2 0 40G 0 part /
sr0 11:0 1 364K 0 rom
root@adminip:~#

```

Veiem que només tenim un sol disc útil en aquest moments, el disc principal amb el sistema operatiu (sda). Per a prevenir errors i tenir un entorn segur on experimentar, per al **RAID 5** connectarem tres discos més a la màquina virtual, **sdb**, **sdc** i **sdd**, i pel **RAID 1**, dos més: **sde** i **sdf**.

A la pàgina d'OpenNebula creem els discos necessaris, que en el nostre cas, amb 40 MB per disc és suficient per a comprovar que la configuració funciona:

Open Nebula Dashboard VMs Templates Services gixpd-ged-2 OpenNebula

Attach new disk

☐ Image ☒ Volatile disk

Size: MB

Disk type: Filesystem:

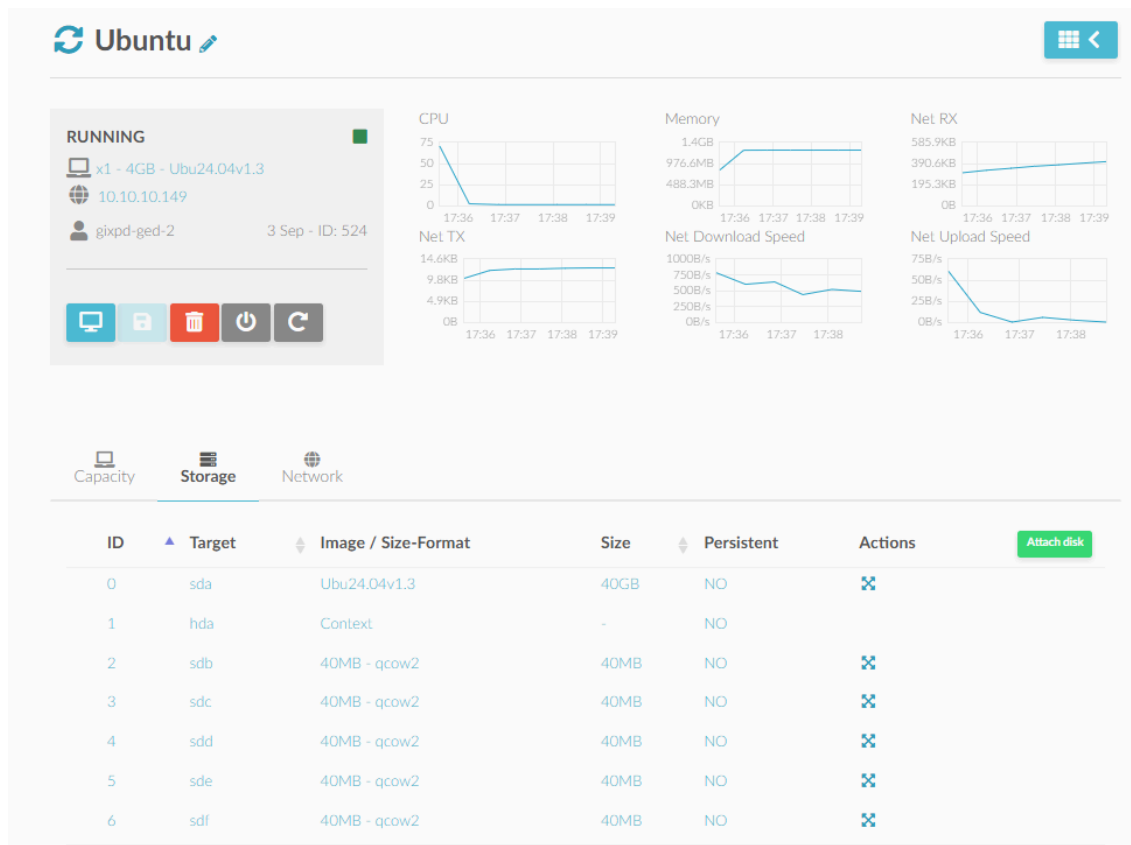
Format:

Advanced options

Target device: Read-only:

BUS: Bus adapter controller:

Disk provisioning type:



Veiem com ara tenim discos connectats, cosa que podem comprovar dins la terminal executant *lsblk*:

```
root@adminp:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0    0    4K  1 loop /snap/bare/5
loop1       7:1    0  74.2M  1 loop /snap/core22/1621
loop2       7:2    0  73.9M  1 loop /snap/core22/1663
loop3       7:3    0 269.8M  1 loop /snap/firefox/4698
loop4       7:4    0 273.6M  1 loop /snap/firefox/5187
loop5       7:5    0 505.1M  1 loop /snap/gnome-42-2204/176
loop6       7:6    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop7       7:7    0  38.8M  1 loop /snap/snapd/21759
sda         8:0    0   40G  0 disk
├─sda1      8:1    0    1M  0 part
└─sda2      8:2    0   40G  0 part /
sdb         8:16   0   40M  0 disk
sdc         8:32   0   40M  0 disk
sdd         8:48   0   40M  0 disk
sde         8:64   0   40M  0 disk
sdf         8:80   0   40M  0 disk
sr0        11:0    1 364K  0 rom
```

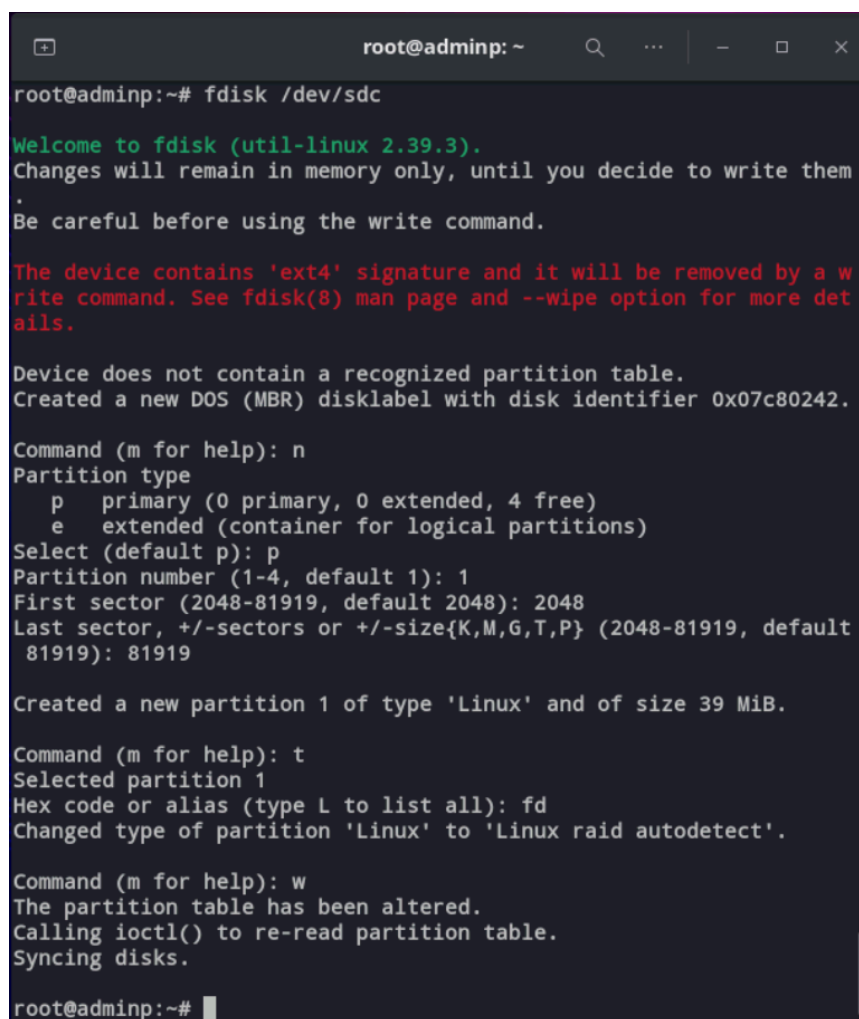
Amb aquesta configuració ja tenim els discos llestos per a crear els sistemes RAID.

2. RAID 5

Com hem dit al pas anterior, per al RAID 5 utilitzarem els discos sdb, sdc i sdd. El que fem en aquest pas és configurar els discos (crear particions i formatejar), i configurar el sistema RAID 5. Per a fer això hem seguit el següents passos:

Crear particions i formatejar

```
# Crear particions en cada disc (exemple per a sdb, repetir per a sdc i sdd)
fdisk /dev/sdb
# Dins de fdisk, executa les següents opcions:
# => n (nova partició)
# => p (partició primària)
# => t (canviar tipus de partició)
# => L (listar tipus)
# => fd (seleccionar Linux RAID auto)
# => w (escriure i sortir)
```



```
root@adminp: ~
root@adminp:~# fdisk /dev/sdc

Welcome to fdisk (util-linux 2.39.3).
Changes will remain in memory only, until you decide to write them
.
Be careful before using the write command.

The device contains 'ext4' signature and it will be removed by a w
rite command. See fdisk(8) man page and --wipe option for more det
ails.

Device does not contain a recognized partition table.
Created a new DOS (MBR) disklabel with disk identifier 0x07c80242.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-81919, default 2048): 2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-81919, default
 81919): 81919

Created a new partition 1 of type 'Linux' and of size 39 MiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): fd
Changed type of partition 'Linux' to 'Linux raid autodetect'.

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

root@adminp:~#
```

```

root@adminp:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
loop0       7:0    0    4K  1 loop /snap/bare/5
loop1       7:1    0  74.2M  1 loop /snap/core22/1621
loop2       7:2    0  73.9M  1 loop /snap/core22/1663
loop3       7:3    0 269.8M  1 loop /snap/firefox/4698
loop4       7:4    0 273.6M  1 loop /snap/firefox/5187
loop5       7:5    0 505.1M  1 loop /snap/gnome-42-2204/176
loop6       7:6    0  91.7M  1 loop /snap/gtk-common-themes/1535
loop7       7:7    0  38.8M  1 loop /snap/snapd/21759
sda         8:0    0   40G  0 disk
├─sda1      8:1    0    1M  0 part
└─sda2      8:2    0   40G  0 part /
sdb         8:16   0   40M  0 disk
└─sdb1      8:17   0   39M  0 part
sdc         8:32   0   40M  0 disk
└─sdc1      8:33   0   39M  0 part
sdd         8:48   0   40M  0 disk
└─sdd1      8:49   0   39M  0 part
sr0        11:0    1  364K  0 rom

```

Podem veure com s'han creat les particions correctament

```

Model: QEMU QEMU HARDDISK (scsi)
Disk /dev/sdb: 41.9MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
  1     1049kB 41.9MB 40.9MB primary raid

Model: QEMU QEMU HARDDISK (scsi)
Disk /dev/sdc: 41.9MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
  1     1049kB 41.9MB 40.9MB primary raid

Model: QEMU QEMU HARDDISK (scsi)
Disk /dev/sdd: 41.9MB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Disk Flags:

Number Start End Size Type File system Flags
  1     1049kB 41.9MB 40.9MB primary raid

```

Podem veure els 3 discos, ara amb la flag "raid" al fer *parted -l*

```

# Formatejar cada partició creada
mkfs.ext4 /dev/sdb1
mkfs.ext4 /dev/sdc1
mkfs.ext4 /dev/sdd1

```

```

root@adminip:~# mkfs.ext4 /dev/sdb1
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 9984 4k blocks and 9984 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

root@adminip:~# mkfs.ext4 /dev/sdc1
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 9984 4k blocks and 9984 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

root@adminip:~# mkfs.ext4 /dev/sdd1
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 9984 4k blocks and 9984 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

```

```

root@adminip:~# lsblk -f
NAME        FSTYPE FSVER
loop0
loop1
loop2
loop3
loop4
loop5
loop6
loop7
sda
├─sda1
└─sda2 ext4      1.0
sdb
├─sdb1 ext4      1.0
sdc
├─sdc1 ext4      1.0
sdd
├─sdd1 ext4      1.0
sr0         iso9660 Joliet Extension
root@adminip:~#

```

Passos completats del format de la partició de cada disc, a ext4

Crear RAID 5

Per a aquest cas, farem servir mdadm, amb els arguments següents:

- **level=5**: Defineix un RAID 5, que distribueix dades i paritat entre discos, permetent recuperar dades si un disc falla
- **raid-devices=3**: Indica que el RAID inclou tres discos, el mínim necessari per a RAID 5

```

# Configurar RAID 5 amb els discs
mdadm --create /dev/md0 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1
/dev/sdd1

# Verificar l'estat de RAID
cat /proc/mdstat

```

```

root@adminip:~# cat /proc/mdstat
Personalities : [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
      75776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>

```

Verificació de l'estat del RAID5, veiem que hi ha actives les particions **sdb1**, **sdc1** i **sdd1**

```

# Formatejar disc raid amb ext4
sudo mkfs.ext4 /dev/md0

# Crear punt de muntatge per a RAID
mkdir /mnt/md0

# Montar RAID

```



```
mount /dev/md0 /mnt/md0
```

```
root@adminp:~# sudo mkfs.ext4 /dev/md0
mke2fs 1.47.0 (5-Feb-2023)
Creating filesystem with 18944 4k blocks and 18944 inodes

Allocating group tables: done
Writing inode tables: done
Creating journal (1024 blocks): done
Writing superblocks and filesystem accounting information: done

root@adminp:~# mkdir /mnt/md0
root@adminp:~# mount /dev/md0 /mnt/md0
root@adminp:~#
```

Disc de RAID5 muntat com `/mnt/md0`

```
# Verificar que el RAID es munta correctament
df -h
```

```
root@adminp:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           392M  1.3M  391M   1% /run
/dev/sda2       40G   16G   23G  42% /
tmpfs           2.0G    0  2.0G   0% /dev/shm
tmpfs           5.0M    0  5.0M   0% /run/lock
tmpfs           2.0G    0  2.0G   0% /run/qemu
tmpfs           392M  172K  392M   1% /run/user/1000
/dev/md0        66M   24K   61M   1% /mnt/md0
root@adminp:~#
```

Veiem que aquest disc de RAID 5 té realment 61 MB disponibles

```
root@adminp:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
loop0       7:0    0    4K  1 loop  /snap/bare/5
loop1       7:1    0  74.2M  1 loop  /snap/core22/1621
loop2       7:2    0  73.9M  1 loop  /snap/core22/1663
loop3       7:3    0 269.8M  1 loop  /snap/firefox/4698
loop4       7:4    0 273.6M  1 loop  /snap/firefox/5187
loop5       7:5    0 505.1M  1 loop  /snap/gnome-42-2204/176
loop6       7:6    0  91.7M  1 loop  /snap/gtk-common-themes/1535
loop7       7:7    0  38.8M  1 loop  /snap/snapd/21759
sda         8:0    0   40G  0 disk
├─sda1      8:1    0    1M  0 part
└─sda2      8:2    0   40G  0 part  /
sdb         8:16   0   40M  0 disk
├─sdb1      8:17   0   39M  0 part
└─md0       9:0    0   74M  0 raid5
sdc         8:32   0   40M  0 disk
├─sdc1      8:33   0   39M  0 part
└─md0       9:0    0   74M  0 raid5
sdd         8:48   0   40M  0 disk
├─sdd1      8:49   0   39M  0 part
└─md0       9:0    0   74M  0 raid5
sr0        11:0    1  364K  0 rom
root@adminp:~#
```

Estat dels discos després d'aquesta última configuració, veiem que els 3 discos estan configurats en RAID5 i formen part de `md0`

Teòricament, amb la configuració que hem fet, veiem que l'espai que hauriem de tenir disponible seria de 80 MB. En el nostre cas, és una mica menys, degut a arxius que existeixen en cada disc de forma natural.

Input - enter your RAID parameters here

Number of disks

Single disk size, TB

RAID type

Results

Capacity
80 TB

Speed gain
2x read speed, no write speed gain

Fault tolerance
1-drive failure

Encara que l'espai disponible no sigui igual al teòric, per a fer una prova, creem un arxiu de 60 MB, una mida major a qualsevol del disc que formen aquest RAID:

```

root@adminp:~# du -h /mnt/md0/
16K    /mnt/md0/lost+found
20K    /mnt/md0/
root@adminp:~# fallocate -l 60M /mnt/md0/prova_60MB.bin
root@adminp:~# ls -lh /mnt/md0/
total 61M
drwx----- 2 root root 16K Oct 31 20:34 lost+found
-rw-r--r-- 1 root root 60M Oct 31 20:41 prova_60MB.bin
root@adminp:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           392M  1.3M  391M   1% /run
/dev/sda2       40G   16G   23G  42% /
tmpfs           2.0G    0  2.0G   0% /dev/shm
tmpfs           5.0M    0  5.0M   0% /run/lock
tmpfs           2.0G    0  2.0G   0% /run/qemu
tmpfs           392M  172K  392M   1% /run/user/1000
/dev/md0        66M   61M  128K 100% /mnt/md0

```

Veiem com efectivament, podem crear aquest arxiu sense problemes, en el disc de RAID.

Al reiniciar la màquina observem això:

```

root@adminp:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE  MOUNTPOINTS
loop0       7:0    0    4K  1 loop  /snap/bare/5
loop1       7:1    0  74.2M  1 loop  /snap/core22/1621
loop2       7:2    0  73.9M  1 loop  /snap/core22/1663
loop3       7:3    0 269.8M  1 loop  /snap/firefox/4698
loop4       7:4    0 273.6M  1 loop  /snap/firefox/5187
loop5       7:5    0 505.1M  1 loop  /snap/gnome-42-2204/176
loop6       7:6    0  91.7M  1 loop  /snap/gtk-common-themes/1535
loop7       7:7    0  38.8M  1 loop  /snap/snapd/21759
sda         8:0    0   40G  0 disk
├─sda1      8:1    0    1M  0 part
├─sda2      8:2    0   40G  0 part  /
sdb         8:16   0   40M  0 disk
├─sdb1      8:17   0   39M  0 part
├─md127     9:127  0   74M  0 raid5 /mnt/raid
sdc         8:32    0   40M  0 disk
├─sdc1      8:33    0   39M  0 part
├─md127     9:127  0   74M  0 raid5 /mnt/raid
sdd         8:48    0   40M  0 disk
├─sdd1      8:49    0   39M  0 part
├─md127     9:127  0   74M  0 raid5 /mnt/raid
sr0        11:0    1  364K  0 rom
root@adminp:~# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4] [raid0] [raid1] [raid10]
md127 : active raid5 sdd1[3] sdb1[0] sdc1[1]
       75776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>

```

Ara el RAID es troba a **md127**, en comptes de **md0**

Per a automatitzar el muntatge del disc correcte per cada cop que reiniciem la màquina, apliquem les següents comandes:

```

root@adminp:~# blkid /dev/md127
/dev/md127: UUID="ee1964b4-7373-49a7-93cc-4bcb626041b5" BLOCK_SIZE="4096" TYPE="ext4"

```

```

root@adminp: ~
GNU nano 7.2 /etc/fstab
/etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/sda2 during curtin installation
/dev/disk/by-uuid/8b341eed-f978-43cf-928c-1950797f945a / ext4 defaults 0 1
/swap.img none swap sw 0 0
UUID=ee1964b4-7373-49a7-93cc-4bcb626041b5 /mnt/raid ext4 defaults 0 0

```

A **/etc/fstab**, afegim aquesta última línia, amb el ID del dispositiu de RAID, per a muntar-lo a **/mnt/raid** automàticament al arrancar la màquina.

```

adminp@adminp:~$ ls /mnt/raid/
lost+found
adminp@adminp:~$

```

Comprovem que a l'arrancar, ja podem fer servir el disc amb normalitat

3. RAID 1

Per a aquesta configuració, els passos que hem seguit són gairebé els mateixos que per a RAID 5, i no s'entrarà en detall de cada pas sinó que s'explicarà els passos que difereixin dels prèviament explicats, i es mostraran igualment els resultats importants dels passos.

Crear particions i formatejar

Per a aquest pas, seguirem el mateix procés que amb RAID 5, però fent servir els discos **sde** i **sdf**. Aquests són els resultats:

```
sde          8:64    0    40M    0 disk
└─sde1       8:65    0    39M    0 part
sdf          8:80    0    40M    0 disk
└─sdf1       8:81    0    39M    0 part
sr0         11:0     1    364K    0 rom
root@adminp:~#
```

Particions creades i discos formatejats

Crear RAID 1

Per aquest pas, existeixen dos principals diferències en la comanda de la creació del RAID:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sde1 /dev/sdf1
```

En comptes de configurar el *level* de RAID a 5, el configurem a 1. Al ser RAID 1, necessitem un mínim de 2 discos, i per tant farem servir 2 *raid-devices* (sde1 i sdf1)

El resultat de seguir també la resta de passos de la configuració anterior ens dona com a resultat:

```
root@adminp:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
tmpfs           392M  1.3M  391M   1% /run
/dev/sda2       40G   16G   23G  42% /
tmpfs           2.0G    0  2.0G   0% /dev/shm
tmpfs           5.0M    0  5.0M   0% /run/lock
tmpfs           2.0G    0  2.0G   0% /run/qemu
/dev/md127       66M   24K   61M   1% /mnt/raid
tmpfs           392M  172K  392M   1% /run/user/1000
/dev/md0        32M   24K   29M   1% /mnt/md0
root@adminp:~#
```

```
sde          8:64    0    40M    0 disk
└─sde1       8:65    0    39M    0 part
   └─md0      9:0     0    38M    0 raid1 /mnt/raid1
sdf          8:80    0    40M    0 disk
└─sdf1       8:81    0    39M    0 part
   └─md0      9:0     0    38M    0 raid1 /mnt/raid1
sr0         11:0     1   364K    0 rom
root@adminp:~#
```

Veiem que el RAID 1 s'ha muntat a /mnt/raid1, i que en aquest cas l'espai que tenim és de 29 MB (en la teoria 40 MB).

4. Configuració de GlusterFS a MVA i MVB:

Com son comandes repetitives vam decidir crear dos arxius bash amb les instruccions MA.sh, MB.sh.

- **A MVA:**
 - Instal·la *glusterfs-server* i *glusterfs-client*.
 - Crea les particions en cada disc, formateja-les i crea els directoris de muntatge.
 - Afegeix els punts de muntatge a */etc/fstab* per muntatge automàtic.
 - Inicia i habilita el servei GlusterFS.
 - Afegeix MVB com a node en el clúster de GlusterFS.

```
# Instal·lar GlusterFS
apt update
apt install -y glusterfs-server glusterfs-client

# Crear i formatar particions en cada disc
fdisk /dev/sda
fdisk /dev/sdb
fdisk /dev/sdc
mkfs.ext4 /dev/sda1
mkfs.ext4 /dev/sdb1
mkfs.ext4 /dev/sdc1

# Crear directoris de muntatge
mkdir -p /data/node1 /data/node2 /data/node3

# Afegir els punts de muntatge a /etc/fstab
echo "/dev/sda1 /data/node1 ext4 defaults 0 1" >> /etc/fstab
echo "/dev/sdb1 /data/node2 ext4 defaults 0 1" >> /etc/fstab
echo "/dev/sdc1 /data/node3 ext4 defaults 0 1" >> /etc/fstab
mount -a

# Iniciar i habilitar GlusterFS
systemctl start glusterd
systemctl enable glusterd

# Afegir MVB com a node en el clúster de GlusterFS
gluster peer probe mvb
```

- **A MVB:**

- Instal·la *glusterfs-server* i *glusterfs-client*.
- Crea particions i directoris de muntatge similars a MVA.
- Afegeix els punts de muntatge a */etc/fstab* per muntatge automàtic.
- Estableix la connexió amb MVA.
- Crea i munta el volum replicat.

```
apt update
apt install -y glusterfs-server glusterfs-client

# Crear particions i directoris de muntatge similars a MVA
fdisk /dev/sda
fdisk /dev/sdb
fdisk /dev/sdc
mkfs.ext4 /dev/sda1
mkfs.ext4 /dev/sdb1
mkfs.ext4 /dev/sdc1
mkdir -p /data/node4 /data/node5 /data/node6

# Afegir punts de muntatge a /etc/fstab
echo "/dev/sda1 /data/node4 ext4 defaults 0 1" >> /etc/fstab
echo "/dev/sdb1 /data/node5 ext4 defaults 0 1" >> /etc/fstab
echo "/dev/sdc1 /data/node6 ext4 defaults 0 1" >> /etc/fstab
mount -a

# Establir connexió amb MVA
gluster peer probe mva

# Crear i muntar el volum replicat
gluster volume create tv1 replica 2 mva:/data/node1/brick0
mva:/data/node4/brick0
gluster volume start tv1
```

5. Configuració de LVM:

Comencem per assegurar-nos que el sistema operatiu està actualitzat, per garantir l'estabilitat i disponibilitat de les eines necessàries amb *apt update*.

Un cop actualitzat el sistema, instal·lem el paquet *lvm2* que ens proporcionarà les eines per gestionar LVM.

```
root@ma:~# apt install lvm2
```

Visualitzem els discos disponibles amb *lsblk* per identificar els que farem servir per a la configuració d'LVM. En el nostre cas *sda*, *sdb*, *sdc*.

```
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda         8:0    0   40M  0 disk
sdb         8:16    0   40M  0 disk
sdc         8:32    0   40M  0 disk
```

A continuació, utilitzem *fdisk* per crear una nova partició en cada disc (*/dev/sda*, */dev/sdb*, */dev/sdc*). Durant aquest procés, seguim aquests passos:

- Triem l'opció *n* per crear una nova partició.
- Seleccionem *p* per indicar que volem una partició primària.
- Definim els límits per utilitzar tota la capacitat del disc.
- Escrivim els canvis amb *w* per guardar i sortir.

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-81919, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-81919, default 81919):

Created a new partition 1 of type 'Linux' and of size 39 MiB.

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

Un cop particionats els discos, procedim a crear els volums físics (Physical Volumes, PV) utilitzant les particions creades. Això ho farem amb la següent comanda: *pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1*.


```
root@ma:~# pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1
Physical volume "/dev/sda1" successfully created.
Physical volume "/dev/sdb1" successfully created.
Physical volume "/dev/sdc1" successfully created.
```

Confirmem la creació dels volums físics amb la comanda *pvs*

```
root@ma:~# pvs
PV          VG Fmt  Attr PSize  PFree
/dev/sda1   lvm2 ---  39.00m 39.00m
/dev/sdb1   lvm2 ---  39.00m 39.00m
/dev/sdc1   lvm2 ---  39.00m 39.00m
```

Agrupem els volums físics en un únic grup de volums, *grp1*, amb *vgcreate grp1 /dev/sda1 /dev/sdb1 /dev/sdc1*. Visualitzem el resultat amb *vgdisplay*.

```
root@ma:~# vgdisplay
--- Volume group ---
VG Name          grp1
System ID
Format           lvm2
Metadata Areas   3
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          3
Act PV           3
VG Size          108.00 MiB
PE Size          4.00 MiB
Total PE         27
Alloc PE / Size  0 / 0
Free PE / Size   27 / 108.00 MiB
VG UUID          UPJj5S-5dl7-VBn3-GprM-ZEYy-QBwa-Hik6UR
```

A partir del grup de volums *grp1*, creem un volum lògic anomenat *vol01* amb una mida de 50 MB: *lvcreate -L 50 -n vol01 grp1*

```
root@ma:~# lvcreate -L 50 -n vol01 grp1
Rounding up size to full physical extent 52.00 MiB
Logical volume "vol01" created.
```

Repetim el mateix procés per crear un segon volum lògic, *vol02*, amb una mida de 50 MB: *lvcreate -L 50 -n vol02 grp1*

```
root@ma:~# lvcreate -L 50 -n vol02 grp1
Rounding up size to full physical extent 52.00 MiB
Logical volume "vol02" created.
```

Formatem els volums lògics creats amb ext4 per fer-los accessibles. Per formatar vol01 i vol02, utilitzem: `mkfs.ext4 /dev/grp1/vol01` `mkfs.ext4 /dev/grp1/vol02`, respectivament.

```
root@ma:~# mkfs.ext4 /dev/grp1/vol01
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 53248 1k blocks and 13328 inodes
Filesystem UUID: 78ac65f7-32d6-4465-95c2-15bee1d04bb0
Superblock backups stored on blocks:
    8193, 24577, 40961

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

```
root@ma:~# mkfs.ext4 /dev/grp1/vol02
mke2fs 1.47.0 (5-Feb-2023)
Discarding device blocks: done
Creating filesystem with 53248 1k blocks and 13328 inodes
Filesystem UUID: 73cb7d07-8e21-4550-8877-1fe923e001d8
Superblock backups stored on blocks:
    8193, 24577, 40961

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

Definim punts de muntatge per a cada volum lògic. En el nostre cas creem les carpetes `/mnt/vol01` i `/mnt/vol02`.

```
root@ma:~# mkdir /mnt/vol01
root@ma:~# mkdir /mnt/vol02
```

Montem cada volum lògic al seu punt de muntatge editant l'arxiu `/etc/fstab` i utilitzant la comanda `mount -a` per muntar les particions definides al fitxer.

Arxiu: `/etc/fstab`

```
#LVM Vol 1
/dev/grp1/vol01 /mnt/vol01 ext4 defaults 0 2

#LVM Vol 2
/dev/grp1/vol02 /mnt/vol02 ext4 defaults 0 2
```

```
root@ma:~# mount -a
```

Un cop creat el volum lògic vol01 dins del grup de volums grp1, procedim a ampliar aquest volum amb més espai. Per començar, utilitzem la comanda `df -h` per veure l'espai actual de les particions muntades. Aquesta comanda ens mostrarà l'espai utilitzat i disponible en el sistema de fitxers.

```

root@ma:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                     1.9G         0   1.9G   0% /dev
tmpfs                    392M      908K   391M   1% /run
/dev/vda1                 16G       3.4G    12G  23% /
tmpfs                    2.0G         0   2.0G   0% /dev/shm
tmpfs                    5.0M         0   5.0M   0% /run/lock
/dev/vda15               124M       12M   113M  10% /boot/efi
localhost:/nfs_tmp        16G       3.4G    12G  23% /mnt
tmpfs                    392M       52K   392M   1% /run/user/1000
/dev/sr0                  364K      364K     0 100% /media/adminp/CONTEXT
/dev/mapper/grp1-vol01    44M       14K    41M   1% /mnt/vol01
/dev/mapper/grp1-vol02    44M       14K    41M   1% /mnt/vol02

```

Afegim espai lògic amb `lvextend -L +2 /dev/grp1/vol01` i `lvextend -L +10 /dev/grp1/vol01`, però `df -h` no es veu modificat. Per què es modifiqui executem `resize2fs /dev/grp1/vol01`, comanda que ajusta el sistema de fitxers ext4 perquè utilitzi tot l'espai extra afegit al volum lògic.

```

root@ma:~# resize2fs /dev/grp1/vol01
resize2fs 1.47.0 (5-Feb-2023)
Filesystem at /dev/grp1/vol01 is mounted on /mnt/vol01; on-line resizing required
old_desc_blocks = 1, new_desc_blocks = 1
The filesystem on /dev/grp1/vol01 is now 57344 (1k) blocks long.

```

Un cop el sistema de fitxers ha estat redimensionat, tornem a executar `df -h` per verificar que l'espai disponible ha augmentat. Ara sí, el sistema de fitxers reconeix l'espai afegit i es mostra a la sortida de la comanda.

```

root@ma:~# df -h
Filesystem                Size      Used Avail Use% Mounted on
udev                     1.9G         0   1.9G   0% /dev
tmpfs                    392M      908K   391M   1% /run
/dev/vda1                 16G       3.4G    12G  23% /
tmpfs                    2.0G         0   2.0G   0% /dev/shm
tmpfs                    5.0M         0   5.0M   0% /run/lock
/dev/vda15               124M       12M   113M  10% /boot/efi
localhost:/nfs_tmp        16G       3.4G    12G  23% /mnt
tmpfs                    392M       52K   392M   1% /run/user/1000
/dev/sr0                  364K      364K     0 100% /media/adminp/CONTEXT
/dev/mapper/grp1-vol01    48M       14K    45M   1% /mnt/vol01
/dev/mapper/grp1-vol02    44M       14K    41M   1% /mnt/vol02

```

6. Resultats:

6.1 Rendiment RAID (1 i 5)

Per a provar el rendiment d'aquestes configuracions, hem fet servir un arxiu de 42MB (més gran que qualsevol disc) i les instruccions *cp* per a escriptura, i *cat* per lectura, i hem mesurat els seus temps:

```
truncate --size 42M sample.txt
time cp sample.txt /mnt/raid
time cat /mnt/raid/sample.txt > /dev/null
```

	REAL	USER	SYS
ESCRITURA RAID 1	0.012s	0.002s	0.002s
LECTURA RAID 1	0.052s	0.001s	0.050s
ESCRITURA RAID 5	0.004s	0.001s	0.003s
LECTURA RAID 5	0.037s	0.000s	0.036s

6.2 Fiabilitat RAID (1 i 5)

Per a provar la fiabilitat de les configuracions RAID 1 i RAID 5 davant la fallada d'un disc, hem simulat la desconnexió d'un dels discos i hem comprovat que els arxius encara són accessibles.

Hem marcat la partició **/dev/sdb1** com a fallada per simular una fallada de disc amb la següent comanda:

```
root@adminp:~# mdadm --manage /dev/md126 --fail /dev/sdb1
mdadm: Value "adminp:0" cannot be set as devname. Reason: Not POSIX compatible. Value ignored.
mdadm: set /dev/sdb1 faulty in /dev/md126
root@adminp:~# cat /proc/mdstat
Personalities : [raid1] [raid6] [raid5] [raid4] [raid0] [raid10]
md126 : active raid5 sdb1[0](F) sdd1[3] sdc1[1]
      75776 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]

md127 : active (auto-read-only) raid1 sde1[0] sdf1[1]
      38912 blocks super 1.2 [2/2] [UU]

unused devices: <none>
root@adminp:~# ls /mnt/raid
lost+found sample.txt
root@adminp:~#
```

Comprovem com realment ha fallat la partició **sdb1** amb *mdadm -detail /dev/md126*:

```

root@adminp:~# mdadm --detail /dev/md126
mdadm: Value "adminp:0" cannot be set as devname. Reason: Not POS
/dev/md126:
    Version : 1.2
    Creation Time : Thu Oct 31 20:25:57 2024
    Raid Level : raid5
    Array Size : 75776 (74.00 MiB 77.59 MB)
    Used Dev Size : 37888 (37.00 MiB 38.80 MB)
    Raid Devices : 3
    Total Devices : 3
    Persistence : Superblock is persistent

    Update Time : Wed Nov 13 20:43:01 2024
    State : clean, degraded
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 1
    Spare Devices : 0

    Layout : left-symmetric
    Chunk Size : 512K

Consistency Policy : resync

    Name : adminp:0 (local to host adminp)
    UUID : f25eb84b:e6d6413e:820cb5d3:8a295005
    Events : 44

   Number   Major   Minor   RaidDevice State
    -       -       -       -       -
    0         0         0         0      removed
    1         8        33         1      active sync  /dev/sdc1
    3         8        49         2      active sync  /dev/sdd1

    0         8        17         -      faulty    /dev/sdb1

```

Finalment, hem comprovat l'accés als arxius al muntatge de RAID per assegurar-nos que seguien disponibles:

```

root@adminp:~# ls /mnt/raid
lost+found sample.txt

```

6.3 Rendiment GlusterFS

Per avaluar el rendiment del sistema GlusterFS creem un arxiu de 42MB i el laborem una taula amb els temps de lectura, escriptura utilitzant instruccions com *cp* / *cat* amb el mòdul *time*

```

truncate --size 42M sample.txt
time cp sample.txt /data-client # Per a GlusterFS
time cat sample.txt > /dev/null

```

	REAL	USER	SYS
ESCRITURA	0,035s	0,002s	0,0018s
LECTURA	0,045s	0,003s	0,022s

6.4 Fiabilitat GlusterFS

Per comprovar la fiabilitat de GlusterFS, realitzarem una sèrie de proves per verificar el comportament del sistema davant la fallada d'un node i la replicació de dades. Aquestes proves permetran assegurar que, fins i tot en condicions adverses, el sistema manté la integritat i disponibilitat dels fitxers.

Configurem el volum GlusterFS amb `replica=2` per garantir que cada fitxer estigui replicat entre dos nodes. Això augmenta la fiabilitat del sistema, ja que qualsevol fallada en un node no afectarà la disponibilitat dels fitxers.

```
gluster volume create tv1 replica 2 transport tcp mva:/data/node4/brick0
mva:/data/node5/brick0
gluster volume start tv1
```

Amb aquest pas, hem configurat la replicació, assegurant que els fitxers es mantindran disponibles fins i tot si un node deixa de funcionar.

Per comprovar la fiabilitat en cas de fallada, simulem la desconnexió del node B temporalment.

```
gluster peer detach mva
```

Tornem al client i verifiquem que els fitxers continuen sent accessibles malgrat la fallada del node B, gràcies a la rèplica.

```
ls -l /data-client
```

Comprovem que el contingut dels fitxers es manté intacte llegint-ne alguns exemples:

```
cat /data-client/prova1.txt
```

```
root@ma:~# cat data-client/prova1.txt
root@ma:~# █
```

Els fitxers han continuat disponibles a través del node A, i el contingut s'ha mantingut íntegre. Aquesta prova confirma que GlusterFS és fiable en cas de fallada d'un node, ja que els fitxers romanen accessibles sense interrupció.

6.5 Rendiment LVM

Per avaluar el rendiment del sistema LVM creem un arxiu de 42MB i elaborem una taula amb els temps de lectura, escriptura utilitzant instruccions com *cp* / *cat* amb el mòdul *time*.

```
truncate --size 42M sample.txt
time cp sample.txt /mnt/lv_data
time cat sample.txt > /dev/null
```

	REAL	USER	SYS
ESCRITURA	0,016s	0,000s	0,006s
LECTURA	0,015s	0,000s	0,015s

6.6 Fiabilitat LVM

En aquesta secció, hem configurat un volum lògic (LV) amb LVM utilitzant el paràmetre *--type raid1*, el qual permet implementar una configuració RAID1 en el volum lògic, proporcionant una redundància de dades a través de la còpia exacta de les dades en dos discos físics. Aquesta configuració ajuda a garantir la fiabilitat del sistema d'emmagatzematge, ja que si un dels discs falla, les dades encara estan disponibles en el segon disc. A continuació es detalla el procés realitzat per provar la fiabilitat del sistema.

Per a configurar LVM amb RAID1, vam utilitzar la següent comanda:

```
lvcreate -L 25 -n vol01 --type raid1 grp1 /dev/sda /dev/sdb
```

Aquesta comanda crea un volum lògic anomenat vol01 de 25 MB de mida, utilitzant els discs /dev/sda i /dev/sdb amb la configuració RAID1. Això significa que les dades es replicaran de manera idèntica entre aquests dos discs, proporcionant una còpia de seguretat automàtica en cas de fallada d'un dels discs.

```
root@ma:~# lvcreate -L 25 -n vol01 --type raid1 grp1 /dev/sda /dev/sdb
Rounding up size to full physical extent 28.00 MiB
Logical volume "vol01" created.
root@ma:~# lvdisplay
--- Logical volume ---
LV Path                /dev/grp1/vol01
LV Name                 vol01
VG Name                 grp1
LV UUID                 7Y2Hgb-rQbv-RBdY-intw-LOTf-RrmM-TBiAuy
LV Write Access         read/write
LV Creation host, time  ma, 2024-11-09 18:58:37 +0100
LV Status                available
# open                  0
LV Size                 28.00 MiB
Current LE               7
Mirrored volumes        2
Segments                1
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            253:4
```

Per demostrar la fiabilitat de la configuració RAID1 amb LVM, realitzem la següent prova: desconnectar un dels discs físics (per exemple, /dev/sda) i comprovar si el volum lògic segueix disponible i accessible sense interrupcions.

```
root@ma:~# vgreduce grp1 /dev/sda
Physical volume "/dev/sda" still in use
```

Desconnectem el disc /dev/sda utilitzant la comanda següent: *vgreduce grp1 /dev/sda*
Un cop desconnectat el disc, comprovem si el volum lògic segueix funcionant amb la comanda *lvdisplay*.


```
root@ma:~# lvsdisplay
--- Logical volume ---
LV Path                /dev/grp1/vol01
LV Name                 vol01
VG Name                 grp1
LV UUID                 7Y2Hgb-rQbv-RBdY-intw-LOTf-RrmM-TBiAuy
LV Write Access         read/write
LV Creation host, time  ma, 2024-11-09 18:58:37 +0100
LV Status                available
# open                  0
LV Size                 28.00 MiB
Current LE              7
Mirrored volumes        2
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     256
Block device            253:4
```

Aquesta comanda mostra que el volum vol01 continua muntat i accessible, tot i que un dels discs ha estat desconnectat.

Conclusions

Fiabilitat

RAID1: Ofereix una alta fiabilitat a causa de la replicació mirall. En fallar un disc, el sistema segueix funcionant correctament gràcies a la redundància, i és possible afegir discos addicionals per augmentar la capacitat sense comprometre la fiabilitat.

RAID5: Equilibra capacitat i fiabilitat mitjançant paritat distribuïda, permetent recuperar dades en fallar un disc. Tanmateix, en fallar més d'un disc, el sistema no pot recuperar les dades, fet que compromet la seva fiabilitat en escenaris de múltiples fallades.

GlusterFS: La fiabilitat és configurable mitjançant el paràmetre de rèplica. Per exemple, amb `replica=2`, els blocs es repliquen a dos nodes, cosa que millora la disponibilitat. Tot i això, la fiabilitat depèn de la configuració de rèpliques i la infraestructura subjacent.

LVM: No ofereix una redundància nativa a nivell de sistema d'arxius; no obstant això, és possible aplicar configuracions RAID (com `--type raid1`) per aconseguir redundància, augmentant la fiabilitat en combinació amb altres tecnologies RAID.

Extensibilitat

RAID1: L'extensibilitat és limitada, ja que cada nou disc afegit implica duplicar la capacitat. No és ideal per escalar sense un cost significatiu.

RAID5: Permet una extensió més eficient en termes d'ús d'espai. Afegir nous discos a RAID5 augmenta la capacitat, tot mantenint la redundància de paritat.

GlusterFS: Excel·lent per a escalar tant horitzontalment com verticalment. És possible afegir nous nodes o rèpliques, fet que facilita la gestió de la capacitat sense reconfigurar la infraestructura.

LVM: Ofereix una gran flexibilitat per augmentar el volum amb discos addicionals i expandir els volums lògics sense afectar el sistema d'arxius existent.

Rendiment

RAID1: Millora el rendiment de lectura (pot llegir de qualsevol disc).

RAID5: Igual que RAID1, les lectures són ràpides.

GlusterFS: Les prestacions varien segons la configuració (distribuït, rèplica, o distribució-replicació). En configuració distribuïda, l'escalabilitat pot augmentar el rendiment de lectura i escriptura, però la rèplica pot impactar negativament en les escriptures.

LVM: El rendiment depèn de la configuració subjacents (tipus RAID en volums lògics). La gestió de múltiples discos facilita un rendiment òptim, però les operacions com l'extensió de volums poden afectar temporalment el rendiment.