RAID, LVM, LUKS: Play with disks.

Mario Polino

Corsi Avanzati Linux - 2014

POuL

have a lot of fun...

RAID

RAID What?


Raid
Mosche &
Zanzare
Efficace contro le vespe
Azione Rapida

RAID What?

RAID

Redundant Array of Inexpensive(Independent) Disks

ITA un sistema informatico che usa un gruppo di dischi rigidi per condividere o replicare le informazioni

ENG schemes that can divide and replicate data among **multiple physical drives**

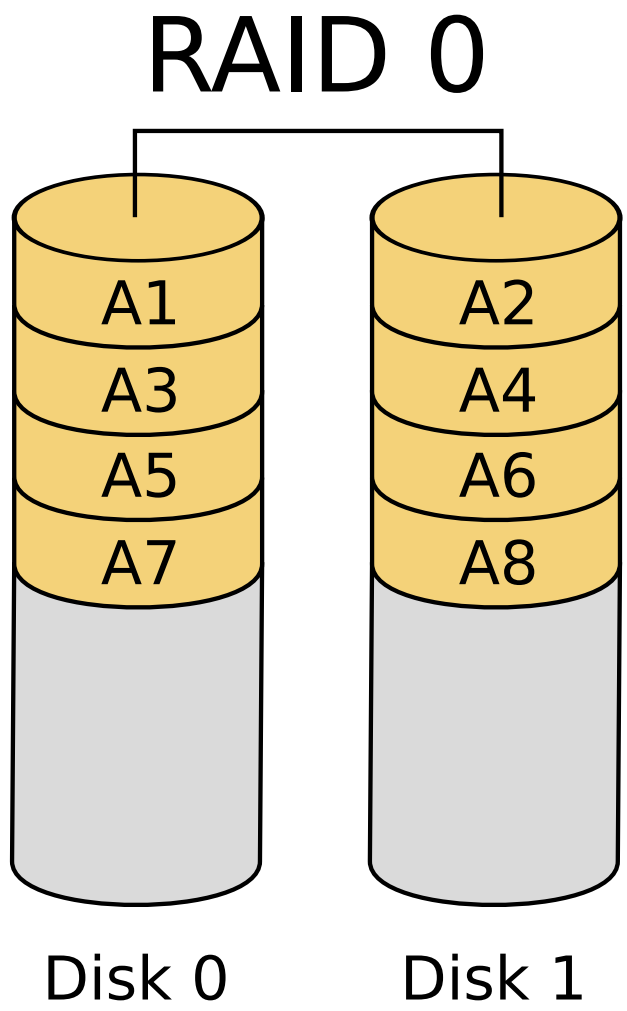
RAID 0: Striping

Pros

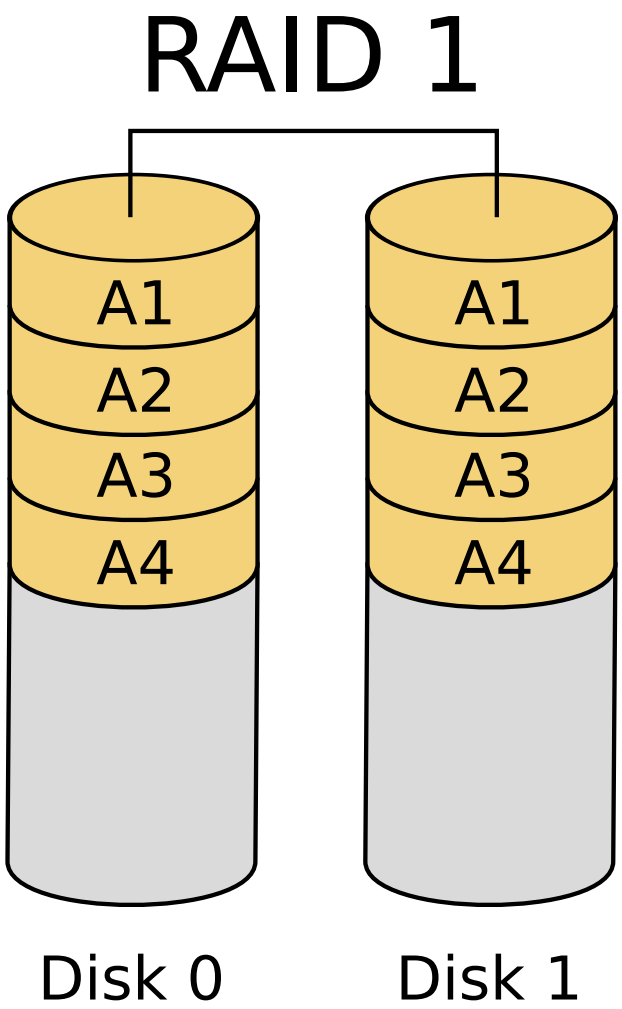
- Alte prestazioni in scrittura e lettura.

Cons

- Impossibile montare dischi hot-spare.
- Affidabilità minore di un disco singolo.
- Non è fault tolerant.



RAID 1: Mirroring



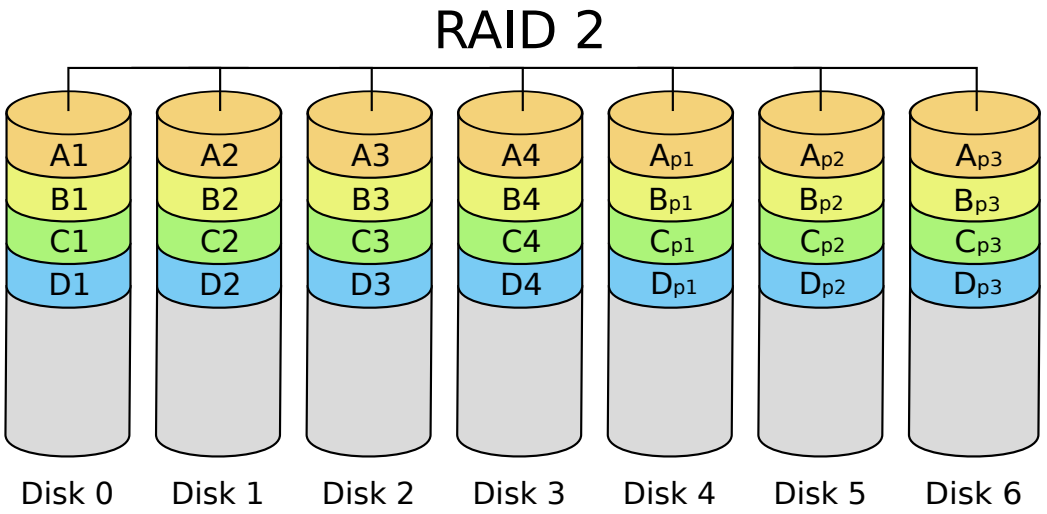
Pros

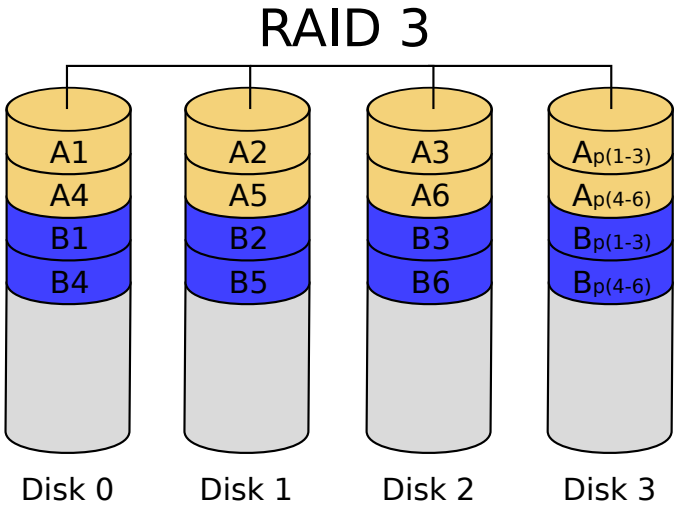
- Affidabilità proporzionale al numero di dischi.
- Fault tolerance.
- Velocità di lettura dipende dal disco più veloce

Cons

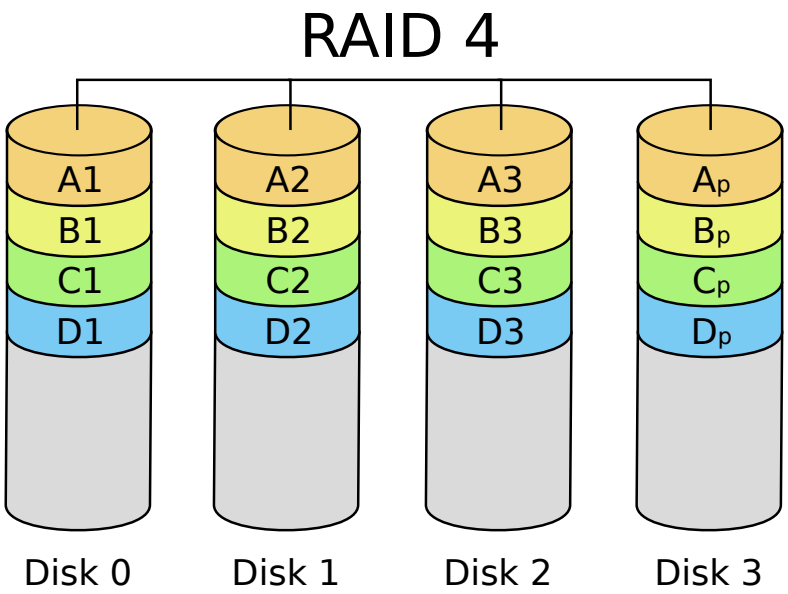
- Alto costo.
- Velocità di scrittura dipende dal disco più lento.

RAID 2,3: Never Implemented





RAID 4: Block Level Striping with Dedicated Parity Disk



Pros

- Read veloci grazie al parallelismo della struttura.
- Fault tolerance.
- Possibilità di inserire dischi hot-spare.

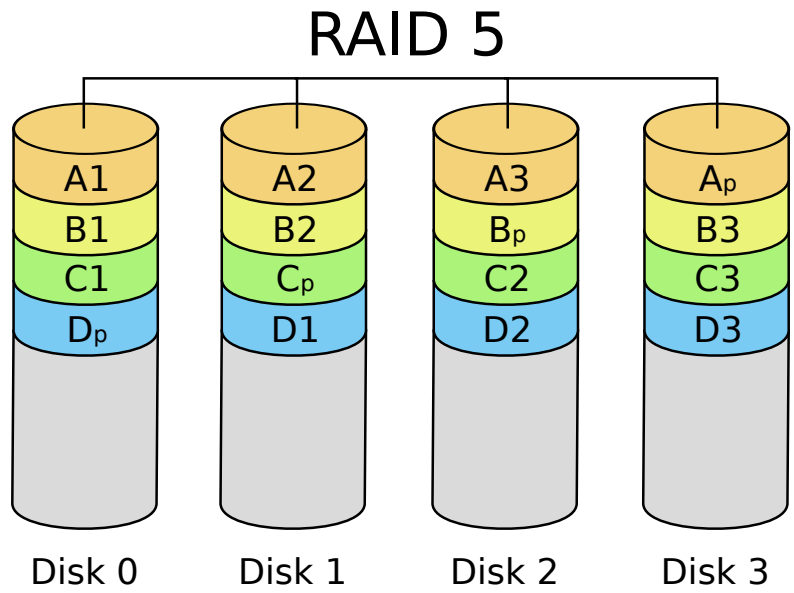
Cons

- Il disco utilizzato per la parità è il collo di bottiglia.
- Scrittura lenta a causa del calcolo della parità

RAID 5: Block Level Striping with Distributed Parity

Pros

- Read veloci grazie al parallelismo della struttura.
- Fault tolerance.
- Possibilità di inserire dischi hot-spare.



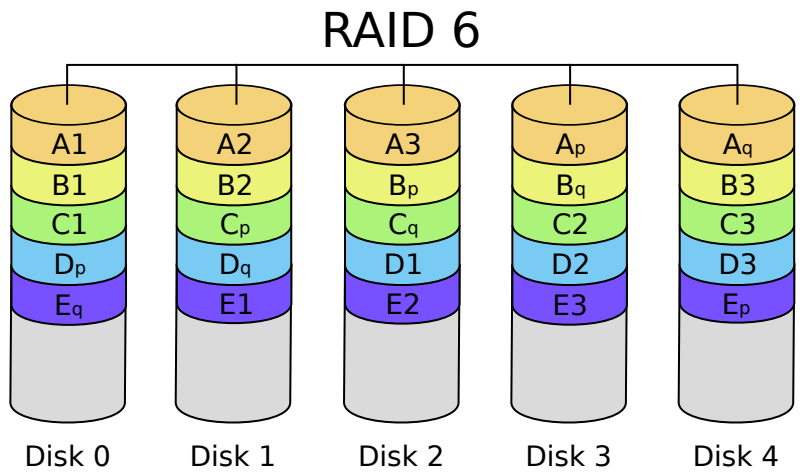
Cons

- Scrittura lenta a causa del calcolo della parità

RAID 6: Block Level Striping with Double Distributed Parity

Pros

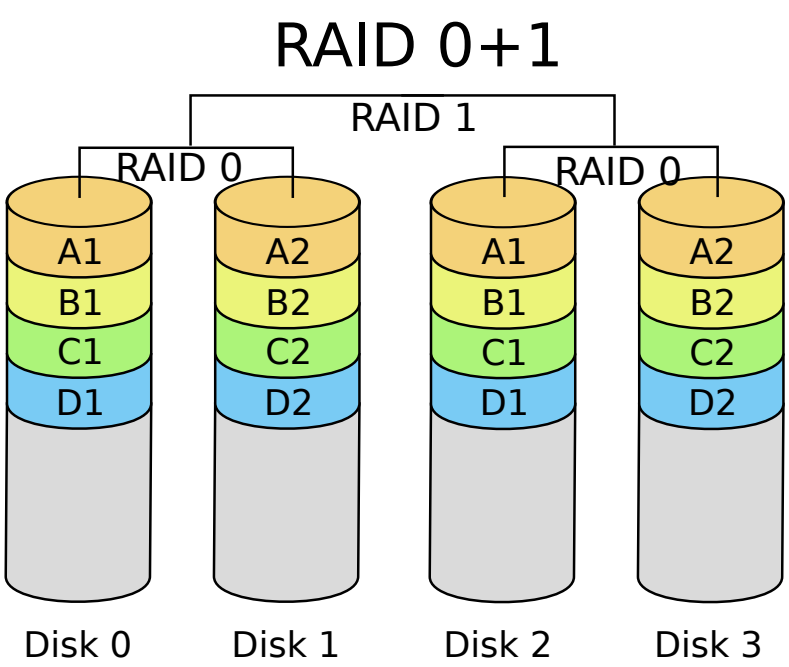
- Read veloci grazie al parallelismo della struttura.
- Fault tolerance.
- Possibilità di inserire dischi hot-spare.

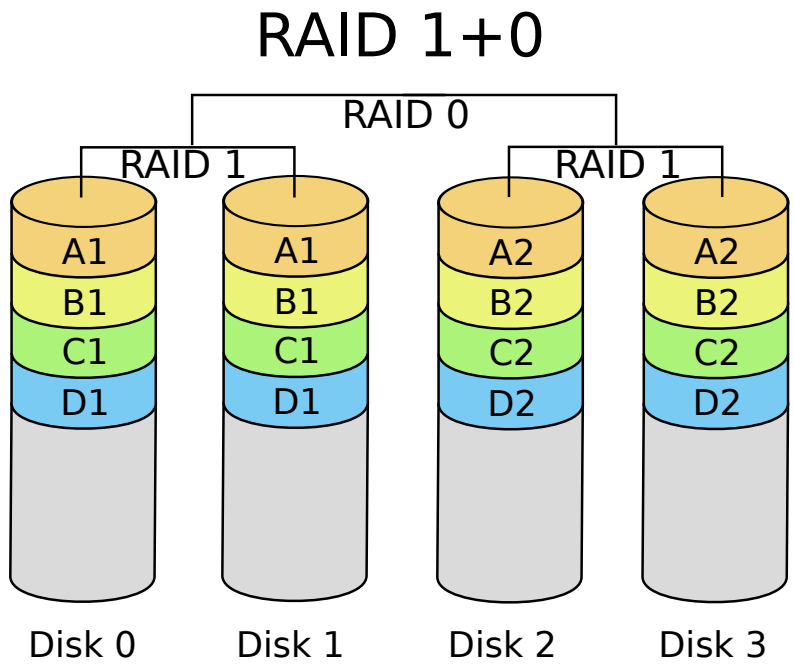


Cons

- Scrittura *'molto'* lenta a causa del calcolo della parità

RAID 0+1 vs 1+0





- 1+0 è una scelta migliore.

mdadm - manage MD devices aka Linux Software RAID

mdadm [mode] <raiddevice> [options] <component-devices>

mdadm - Supporto

- RAID**0**
- RAID**1**
- RAID**4**
- RAID**5**
- RAID**6**
- RAID**10** (1 + 0)
- LINEAR (Come RAID0 con dischi di dimensione diversa)

mdadm - mode

- -A Assemble: Assembla i componenti di un array già creato in un block device attivo.
- -C Create: Crea un nuovo array di dischi e lo attiva.
- -F Follow/Monitor: Controlla lo stato dei dischi.
- -G Grow: Cambia il numero di dischi il layout, etc.

mdadm - Create

```
mdadm -Cv /dev/md0 –level=5 –raid-devices=3 /dev/sdb1
/dev/sdc1 /dev/sdd1 –spare-devices=1 /dev/sde1
```

mdadm - Add/Remove per un Disco

Add

```
mdadm –add /dev/md0 /dev/sdb1
```

Remove

```
mdadm /dev/md0 –fail /dev/sda1 –remove /dev/sda1
```

mdadm - Monitorare un Array

```
mdadm –detail /dev/md0
```

```
cat /proc/mdstat
```

```
Personalities : [raid1]
md0 : active raid1 sdb1[1] sda1[0]
104320 blocks [2/2] [UU]
md1 : active raid1 sdb3[1] sda3[0]
19542976 blocks [2/2] [UU]
md2 : active raid1 sdb4[1] sda4[0]
223504192 blocks [2/2] [UU]
```

mdadm - Stop/Delete per un Array

Stop

```
mdadm –stop /dev/md0
```

Delete

```
mdadm –remove /dev/md0
```

mdadm - /etc/mdadm.conf

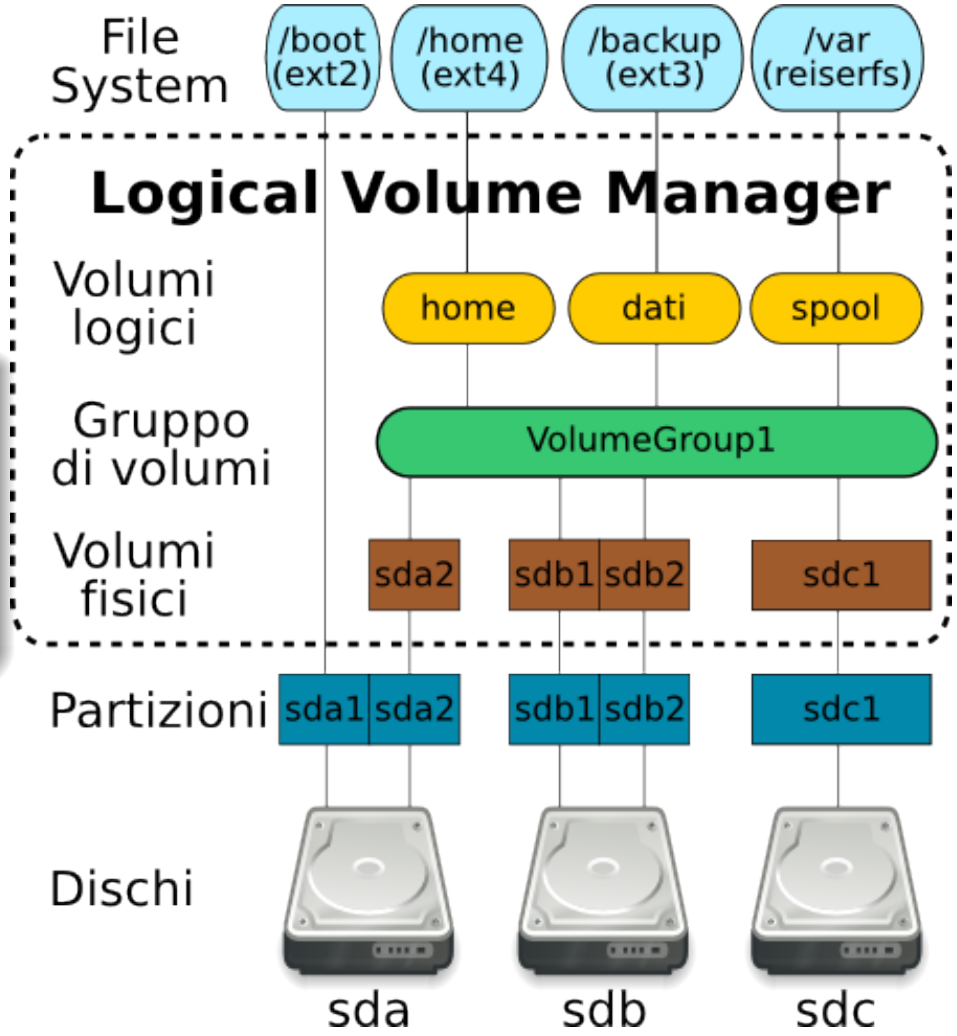
```
mdadm –examine –scan > /etc/mdadm.conf
```

LVM

Cosa è LVM?

Logical Volume Manager

Crea un livello di astrazione che permette di creare dischi logici che vengono poi archiviati in dischi fisici.



I Vantaggi di LVM

- **Capacità flessibile**: Un Filesystem può essere esteso su più dischi fisici.
- **Storage ridimensionabile**: Si può fare il redimensionamento dei volumi logici senza intaccare le partizioni.
- **Online data relocation**: Puoi riposizionare i dati mentra il sistama è attivo. Per esempio puoi decidere di svuotare completamente un disco prima di rimuoverlo.
- **Nomi significativi**: I volumi possono essere raggruppati e nominati a piacimento dell'utente.
- **Disk striping**: Simile al RAID0
- **Mirroring volumes**: Simile al RAID1
- **Volume Snapshot**: Si possono creare snapshot dei volumi (e quindi fare modifiche che non modificano i dati reali)

lvm - Gestire i volumi fisici

pvcreate /dev/sda2

inizializza la partizione */dev/sda2* per l'uso di LVM (Si può usare anche su interi dischi)

pvscan

pvscan scans all supported LVM block devices in the system for physical volumes.

pvdisplay

Mostra le informazioni su un volume fisico

lvm - Creare gruppi di volumi

vgcreate VolGroup00 /dev/sda2

Crea un nuovo gruppo di volumi di nome *VolGroup00* che contiene il disco */dev/sda2*

vgextend VolGroup00 /dev/sdb1

Aggiunge la partizione */dev/sdb1* al gruppo *VolGroup00*

vgscan

Legge tutti i dischi alla ricerca di volumi fisici che contengono metadati di LVM.

vgdisplay

Mostra informazioni sui Volume Group

lvm - Creare i volumi logici lineari

lvcreate -L 10G VolGroup00 -n lvolhome

Crea un volume logico di 10GB nome *lvolhome*
Il volume è accessibile da */dev/mapper/VolGroup00-lvolhome* o */dev/VolGroup00/lvolhome*
Si può usare l'opzione -*C y* per assicurarsi che il volume sia contiguo.

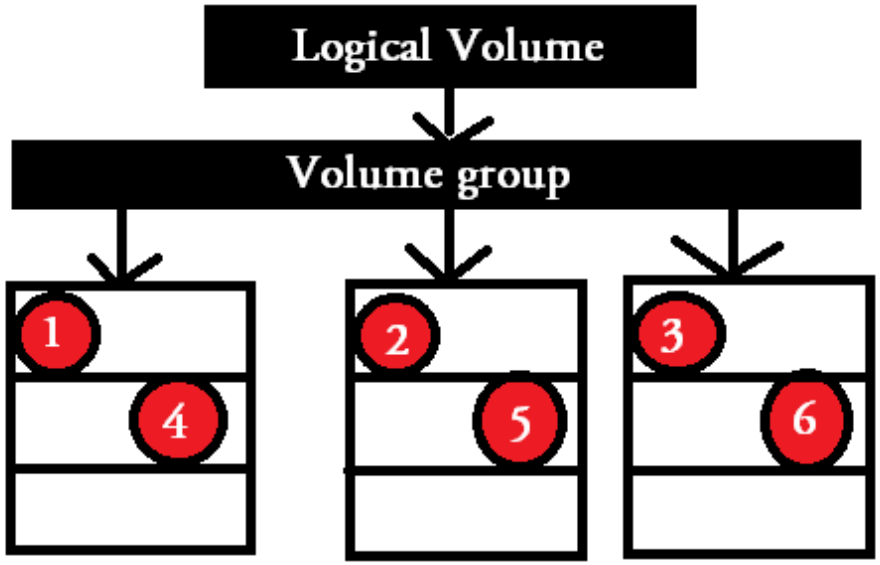
lvcreate -l +100%FREE VolGroup00 -n lvolmedia

Crea un volume logico che occupa tutto il restante spazio disponibile

lvdisplay

Mostra informazioni sui Volumi logici

lvm - Creare i volumi logici stripped

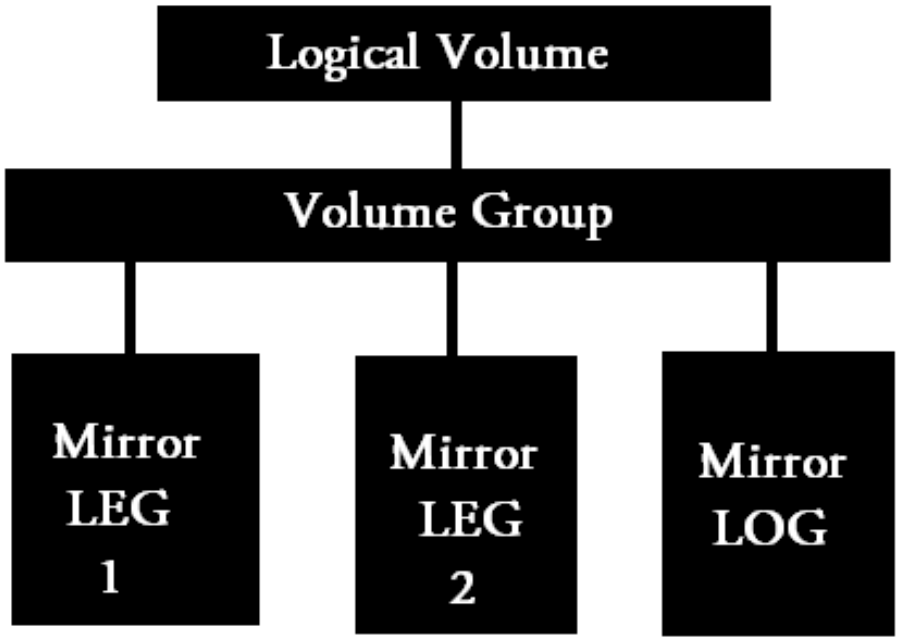


lvcreate -L 10G -**i2** -**I64** -n example VolGroup00

Crea un volume logico stile RAID 0 con blocchi di dimensione di 64 kb, dimensione 10G, usando due volumi fisici.

Hai bisogno di **almeno 2** volumi fisici

lvm - Creare i volumi logici mirrored



Hai bisogno di **almeno 3** volumi fisici, il terzo è usato per log relativi al mirroring

lvconvert -m1 /dev/vg00/lv0

Il mirroring può essere aggiunto o eliminato (-m0) anche da un volume già esistente

lvcreate -L 10G -**m1** -n mirrorexample VolGroup00

Crea un volume logico con un mirror (stile RAID 1) di dimensione 10G.

lvm - Snapshot

COW (copy-no-write)

Lo snapshot iniziale conterrà hard-link agli inode dei dati attuali. Fino a che i dati restano invariati, lo snapshot conterrà solo i link agli inode e non i dati stessi. Quando verrà modificato un file o una cartella ai quali lo snapshot punta, LVM automaticamente clonerà i dati, la vecchia copia referenziata dallo snapshot, e la nuova copia referenziata dall'attuale sistema.

lvcreate –size 100M –snapshot –name snap01 /dev/mapper/vg0-pv

Crea un nuovo volume logico basato su */dev/mapper/vg0-pv* con massi 100MB di modifiche apportabili.

LUKS

Cosa è LUKS?

LUKS

Linux Unified Key Setup

Linux Unified Key Setup

è lo standard per hard disk cifrati su linux.

I Vantaggi di LUKS

- Compatibilità via standardizzazione.
- Sicuro contro attacchi low entropy.
- Supporto per multi key.
- Effective passphrase revocation.
- Free

Wipe the Disk

cat /dev/zero > /dev/sdx

Scrive zero sul disco *sdx*.
Se il disco è stato usato in precedenza, è una buona idea fare un wipe prima di creare un LUKS container per rimuovere ogni traccia del vecchio filesystem e dei vecchi dati.

dd if=/dev/urandom of=/dev/sdx

Flash memory/SSD

Le memorie SSD e/o flash adottano un livello di astrazione per cui i dati non sono scritti esattamente nel settore indicato. Alcuni fanno addirittura la compressione dei dati, quindi scrivere solo zeri risulta in avere sovrascritto poco spazio. `https://www.usenix.org/legacy/events/fast11/tech/full_papers/Wei.pdf`

The Command

cryptsetup <OPTIONS> <action> <action-specific> <device> <dmname>

Mode

- **LUKS**: Default
- plain: Per usare dm-crypt in plain mode
- loopaes: legacy mode
- tcrypt: Modo compatibile con Truecrypt

Format/Open

cryptsetup -v –cipher aes-xts-plain64 –key-size 512 –hash sha1 –iter-time 1000 **luksFormat** <device>

–*chiper* per impostare il cifrario da usare (aes-xts-plain64)
–*key-size* dimensione della chiave (256)
–*hash* funzione di hash per PBKDF2 (sha1)
–*iter-time* numero di millisecondi spesi dentro PBKDF2 (1000)

cryptsetup **open –type luks** /dev/sdx chipDevice

Apre la partizione cifrata *sdx* creando un device a blocchi */dev/mapper/chipDevice* che si può utilizzare come un normale disco.

Keyfiles

dd if=/dev/urandom of=mykeyfile bs=512 count=4

Genera un file *mykeyfile* e lo riempie di dati random

cryptsetup -c <desired cipher> -s <key size> luksFormat /dev/<volume to encrypt> /path/to/mykeyfile

Crea un disco cifrato e usa *mykeyfile* come chiave di cifratura.

cryptsetup open –type luks –key-file /path/to/mykeyfile /dev/sdx chipDevice

Apre la partizione cifrata *sdx* usanto *mykeyfile* come file.

Si può utilizzare come chiave sia un file che un device a blocchi (una partizione, magari su un dispositivo rimovibile). Si può utilizzare l'opzione *–keyfile-offset* per indicare l'offset di inizio della chiave sul file/dispositivo e *–keyfile-size* per indicare la quantità di dati da considerare per la chiave.

Key management /1

Si possono usare fino a **8** chiavi differenti. I key slot vanno da 0 a 7

cryptsetup luksDump /dev/<device>

Informazioni sul device cifrato e sulle chiavi utilizzate

cryptsetup luksAddKey /dev/<device> [/path/to/<additionalkeyfile>] [-d /path/to/<keyfile>]

Aggiunge una nuova chiave nel primo slot libero. Per l'aggiunta di una nuova chiave bisogna essere in possesso di almeno una delle vecchie.
-d è usata per indicare il percorso del keyfile da utilizzare come chiave.
-S specifica lo slot nel quale si vuole aggiungere la chiave.

Key management /2

cryptsetup luksChangeKey /dev/<device> -S 6

Cambia la chiave dello slot 6. Bisogna essere a conoscenza della chiave che si sta cambiando.

Con entrambe le opzioni seguenti è possibile rimuovere anche l'ultima chiave rimasta e quindi rendere i dati persi per sempre.

cryptsetup luksRemoveKey /dev/<device>

Rimuove la chiave inserita.

cryptsetup luksKillSlot /dev/<device> 6

Rimuove la chiave nello slot 6 con una qualsiasi altra chiave.

Backup LUKS Header

L'header della partizione cifrata è di vitale importanza per poter accedere hai dati contenuti, quindi un danno accidentale a questa sezione può rendere inaccessibili tutti i dati.

cryptsetup luksHeaderBackup /dev/sdx –header-backup-file /path/to/header.img

Salva in *header.img* una copia di backup del header della partizione *sdx*

Restore LUKS Header

cryptsetup -v –header /path/to/header.img open /dev/sdx testHead

Apre il device *sdx* utilizzando l'header *header.img*. Non sovrascrive l'header originale del device. É utile per controllare che l'header sia corretto prima di fare il Restore.

cryptsetup luksHeaderRestore /dev/sdx –header-backup-file ./path/to/header.img

Sovrascrive l'header del dispositivo *sdx* con il backup contenuto in *header.img*

Grazie per l'attenzione.