

Logical Volume Manager (LVM)

Johan Swenker

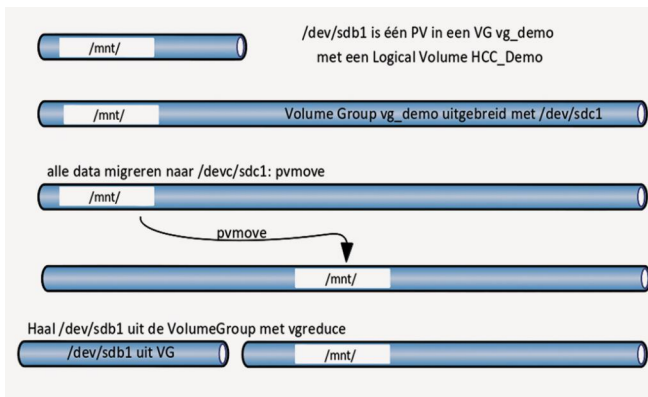
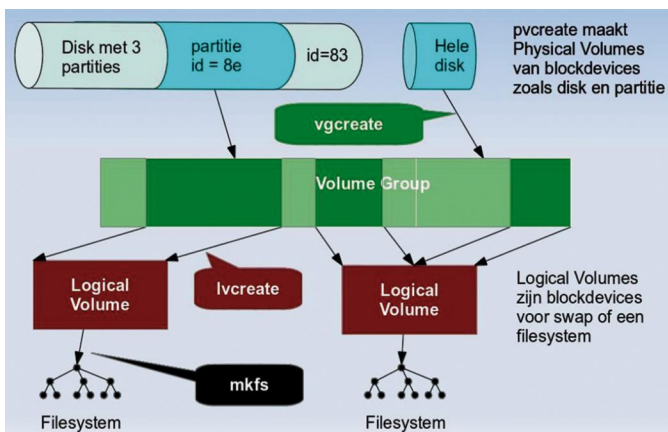
Tijdens de *CompUfair* in september 2016 gaf ik een presentatie over LVM, maar helaas ging de demonstratie een beetje de mist in. Ik was vergeten dat je met **Ctrl+Shift+Plus** de letters groot en leesbaar op het scherm kunt krijgen, net als bij een browser.

Vandaar dit artikel, zodat je thuis zelf de demo kunt uitvoeren.

Het concept van een logical volume manager bestaat al tientallen jaren. Ook in MS Windows bestaat iets vergelijkbaars, daar heet het 'Storage Space'.

De LVM zorgt ervoor dat je flexibel met de partities van een harde schijf kunt omgaan. In eerste instantie had ik op mijn NAS aparte partities voor foto's, muziek, en de belangrijke bestanden uit 'Mijn Documenten'. Dat ging goed tot de hoeveelheid muziek uit de partitie groeide, en zelfs uit de disk. Als ik hier vooraf rekening mee had gehouden, dan had ik de hele disk via LVM moeten beheren: LVM kan namelijk diskruimte van een nieuwe disk 'virtueel' toevoegen aan een bestaande partitie op de oude disk!

LVM beheert zogenaamde 'Volumes', en als je het op zijn Engels uitspreekt, dan is het een nieuw begrip, waarvan ik de betekenis nog moet uitleggen. Een volume heeft heel veel weg van een partitie. Het is een stukje disk waarop je een filesystem (zoals /var of in Windows-termen D:) kunt zetten. In LVM worden er twee soorten volumes onderscheiden: Physical Volume (PV) en Logical Volume (LV). Een Physical Volume is een brokje disk, en dat maak je met een parti-tiemanager zoals fdisk. Een Logical Volume wordt gemaakt door LVM en dat kun je gebruiken alsof het een partitie is.



Tussen het Physical Volume en het Logical Volume zit nog een Volume Group (VG). Een Volume Group is een collectie Physical Volumes; LVM maakt de Logical Volumes aan op basis van een Volume Group. Het is de Volume Group die aan LVM de flexibiliteit geeft. Dankzij die Volume Groups kun je met LVM zaken realiseren zoals RAID, en kun je ook transparant een nieuwe disk toevoegen aan de verzameling van disks in je systeem.

In het plaatje in de vorige kolom kunnen we zien hoe dit in zijn werk gaat:

- Bovenin zien we links een disk met drie partities. Een van die partities heeft het kenmerk 8e. Dit kenmerk geeft aan dat de partitie bedoeld is voor LVM. Het bekendste kenmerk is waarschijnlijk 5 voor een extended partitie en 7 voor een NTFS-partitie.
- Bovenin rechts is aangegeven dat LVM ook een hele (ongepartitioneerde) disk kan beheren.
- Het commando `vgcreate` plaatst de Physical Volumes in een Volume Group.
- De groene balk in het midden geeft zo'n Volume Group aan, waarbij lichtgroen vrije ruimte is en donkergroen in gebruik is voor een Logical Volume.
- Het commando `lvcreate` maakt Logical Volumes uit een Volume Group.
- Het linkse donkerrode blok geeft een Logical Volume aan dat uit een aangesloten deel van één disk komt.
- Het rechtse donkerrode blok geeft een Logical Volume aan dat uit twee verschillende disks is samengesteld.
- Helemaal onderaan is aangegeven dat je met `mkfs` een filesystem kunt aanmaken op een Logical Volume.

Het bovenstaande schetst al dat je een nieuwe partitie (of hele disk) als Physical Volume kunt toevoegen aan een reeds bestaande Volume Group. Daarna heb je dus veel meer ruimte in die Volume Group. Die ruimte kun je gebruiken voor een nieuw Logical Volume, of om een bestaand Logical Volume uit te breiden.

Je kunt die nieuwe ruimte ook gebruiken om een bestaand Physical Volume vrij te maken: eerst buiten gebruik stellen, dan echt vrijmaken en uit de Volume Group halen, en daarna kun je eventueel de disk waar dat oude Physical Volume op stond uit de pc halen.

Demonstratie / workshop

Om de demonstratie na te doen heb je twee USB-memorysticks nodig, waarvan de inhoud niet meer belangrijk is. Verder heb je een Linux-systeem nodig. De beschrijvingen zijn voor de `ubuntu-16.04.1-desktop-amd64.iso`-DVD die ik gedownload heb van <http://releases.ubuntu.com/16.04.1/>, maar het werkt net zo goed op bijvoorbeeld een Raspberry Pi.

Boot de pc van de live-CD of USB-stick; stop de beide USB-sticks in de pc; start een terminal op, zodat we kunnen gaan typen; rechter muisklik; open terminal.

Alle acties die we gaan uitvoeren, moeten we als 'root' uitvoeren. Let op, root kan de harde schijf van je pc wissen, ook als je met een live-CD gestart bent. We moeten dus eerst zorgvuldig uitzoeken hoe binnen Linux de beide USB-sticks genoemd worden.

Het commando `df -h | grep media` laat zien hoe de beide USB-sticks heten.

```
/dev/sdb1 981M 64K 980M 1% /media/
          ubuntu/D193-2027
/dev/sdc1 2.0G 4.0K 2.0G 1% /media/
          ubuntu/08E2-D382
```

Bij mij is `/dev/sdb1` de stick van 1 GB en `/dev/sdc1` de stick van 2 GB. Nu je weet hoe de sticks heten, kun je (relatief) veilig root worden: `sudo bash`

We gaan eerst van de 1 GB-stick een Physical Volume maken. Maar let op, controleer eerst of jouw stick ook `/dev/sdb` is; eventueel moet je `/dev/sdc`, `/dev/sdd`, of iets anders gebruiken.

```
umount /dev/sdb1
fdisk /dev/sdb
```

- gebruik `m` om een lijst van commando's te zien
- gebruik `p` om de huidige partitie tabel te zien een normale stick heeft één FAT-partitie, als je iets anders ziet ga dan niet verder voordat je zeker weet dat je niet op je harde schijf bezig bent
- gebruik `t` om het partitie type te veranderen van 'b' naar '8e'
- gebruik nog eens `p` om te zien dat het partitie type veranderd is
- gebruik `w` om het nieuwe partitie type weg te schrijven

```
pvcreate --metadatascopies 2 /dev/sdb1
```

De volgende drie commando's laten het effect van de `pvcreate` zien:

```
pvscan -n
pvdisplay /dev/sdb1
pvs /dev/sdb1
```

We gaan nu een Volume Group maken, met als basis het Physical Volume dat we net gemaakt hebben:

```
vgcreate vg_demo /dev/sdb1
```

Het commando `vgdisplay` laat zien dat we nu een Volume Group hebben met als naam `vg_demo`.

Nu moeten we in die Volume Group een Logical Volume maken. Ik heb gekozen voor een Logical Volume van 300 MB:

```
lvcreate -v -n HCC_Demo -L 300M vg_demo
```

De volgende commando's laten weer zien wat we gedaan hebben: `lvs` en `vgdisplay -v`.

Onthoud het Logical Volume Path. Als je deze workshop letterlijk volgt, dan zal dat Path `/dev/vg_demo/HCC_Demo` zijn. Dit Path hebben we nodig voor de volgende stappen. Kijk ook even naar de regels met het woordje PE. LVM allociert diskruimte in blokken ter grootte van een Physical Extend (PE). Het commando `vgdisplay` geeft een paar parameters over die Physical Extends, zoals grootte en aantal.

Op dat Logical Volume gaan we een File System aanmaken. Daarmee maken we de structuren aan op de disk, zodat Linux er bestanden op kan plaatsen en terug kan halen.

```
mkfs.ext4 /dev/vg_demo/HCC_Demo
mount /dev/vg_demo/HCC_Demo /mnt
df -h
```

We hebben bijna 300 MB beschikbaar in de directory `/mnt` en laten we daar ook maar gebruik van gaan maken. Zoek een filmpje of een muziekje in ogg-formaat (om licentieredenen zitten de mp3- en mp4-codecs niet in de live-CD, maar ze zijn wel beschikbaar).

```
wget -O /mnt/BBB_480.ogg http://download.
blender.org/peach/bigbuckbunny_movies/
big_buck_bunny_480p_stereo.ogg
```

Dit filmpje duurt tien minuten; dat is lang genoeg om de rest van de demonstratie te vullen.

Nu gaan we met de andere USB-stick aan de gang. Ook van die andere USB-stick maken we een Physical Volume. Dat gaat op precies dezelfde manier als voor `/dev/sdb1`

```
umount /dev/sdc1
fdisk /dev/sdc1
          en de subcommando's t (8e) en w
pvcreate --metadatascopies 2 /dev/sdc1
```

De bestaande Volume Group `vg_demo` gaan we uitbreiden: `vgextend vg_demo /dev/sdc1`

Nu wordt het tijd om het filmpje op te starten. Maar voordat we dat gaan doen moeten we eerst de file system cache van Ubuntu leegmaken, anders zal Ubuntu het filmpje uit de cache halen en niet vanaf de USB-stick

```
sync; echo 3 > /proc/sys/vm/drop_caches
```

Dubbelklik nu op de bestandsnaam en schuif het venster over de zijkant van het scherm. Dan staat het niet in de weg en kun je toch horen en zien dat alles soepel doorloopt.

Eerst gaan we de eerste USB-stick buiten gebruik stellen:

```
pvchange -xn /dev/sdb1
```

Met het commando `vgdisplay -v` kun je zien dat de eerste USB-stick niet meer gebruikt zal worden voor nieuwe Logical Volumes.

De data staat nog op de eerste USB-stick. De data moet naar de andere stick, zonder dat het afspelen van de film onderbroken wordt.

```
pvmove -i 10 -n /dev/vg_demo/HCC_Demo
/dev/sdb1
```

Het commando `pvmove` doet er ongeveer 3 minuten over om de data te verplaatsen. Het resultaat is weer te verifiëren met `vgdisplay -v`: op de eerste disk zijn geen Physical Extends meer in gebruik.

Nu moet het Physical Volume `/dev/sdb1` nog uit de Volume Group gehaald worden:

```
vgreduce vg_demo /dev/sdb1
```

Nu kan de eerste USB-stick uit de pc gehaald worden, terwijl de film gewoon doorspeelt. Als je ook de tweede USB-stick uit de pc haalt, dan stopt het afspelen binnen een paar seconden (als je tenminste de buffercache gewist hebt)

O ja, met je pc is niets gebeurd. Als je die reboot, start die weer gewoon op zoals je gewend was. De beide USB-sticks zijn door workshop aangepast. Dat kun je herstellen met:

```
fdisk /dev/sdb
          en de subcommando's t (b) en w
mkfs.fat /dev/sdb1
```

