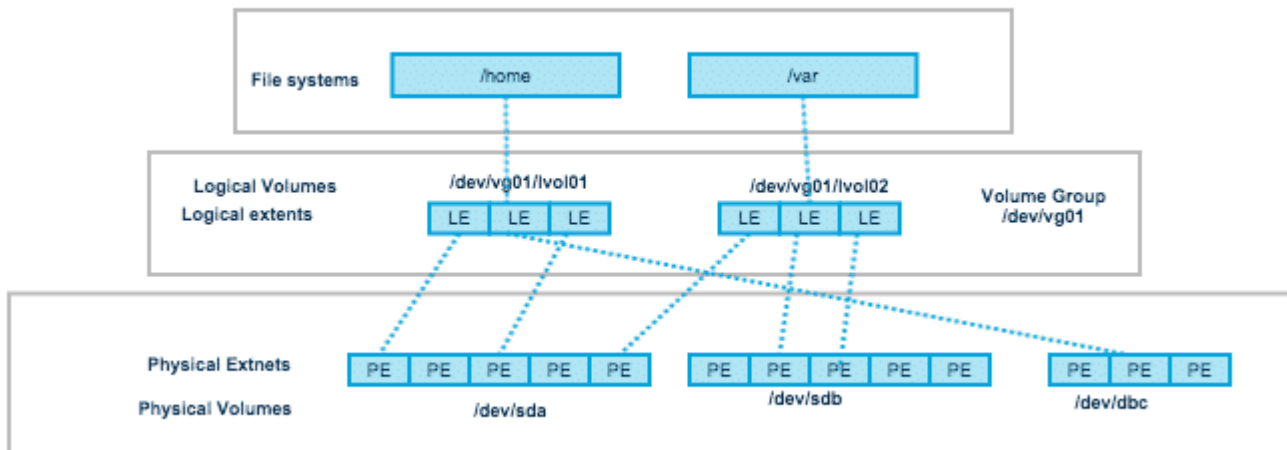


LVM - Logical Volume Manager

LVM steht für **Logical Volume Manager** und ist ein Verfahren, um Speicherplatz über mehrere Festplatten dynamisch zu verwalten. Es werden sozusagen, zusätzliche *logische Schichten zwischen der Hardware, den Partitionen und dem Dateisystemen* erstellt respektive eingefügt.



LVM bietet die Möglichkeit, mehrere Partitionen logisch zusammenzufassen (**Volume Group**) und in diesem logischen Verbund neue "logische Partitionen" (**Logical Volume**) anzulegen. Der Volume Group können dynamisch weitere Geräte hinzugefügt werden, wodurch sich der verfügbare Speicherplatz erhöht. Freier Speicherplatz kann ebenfalls dynamisch den Logical Volumes zugewiesen werden. Dadurch erreicht man eine höhere Flexibilität als beim klassischen Partitionierungsansatz. Dies ist vor allem für Server interessant, bei denen dynamisch auf das Wachsen der Datenbestände reagiert werden muss.

Vorteile:

- Volumes können im laufenden Betrieb vergrößert, angelegt und verkleinert werden
- Probleme durch falsche Partitionierung können vermieden werden
- Höhere Flexibilität
- Die Kapazität mehrerer Festplatten kann zu einem Volume zusammengefasst werden
- Verschiedene Versionen von Daten sind durch sogenannte "Snapshots" möglich
- Die Datensicherung ("backup") wird durch diese "Snapshots" vereinfacht
- LVM hat einen hohen Datendurchsatz ("performance") und beeinträchtigt das System nur unwesentlich

Nachteile:

- Liegen die Logical Volumes auf mehreren Festplatten, so wird das **Risiko** des **Single-Point-of-Failure** grösser, da beim Ausfall einer Platte alle Daten verloren sind, ähnlich wie bei **RAID0**
- Um auf Logical Volumes zugreifen zu können, benötigen sog. Rescue CDs auch LVM-Unterstützung
- Es ist eine zusätzliche `/boot`-Partition ausserhalb des LVM-Verbunds erforderlich (gilt für GRUB, nicht für GRUB 2!)

Grundlagen zur LVM Mechanik

Vier Begriffe sind beim LVM essentiell und sollen zunächst verstanden werden:

Physical Extent: Das sog. Physical Extent (**PE**) ist die kleinste mögliche Speichergrösse in einem LVM. Das PE ist vergleichbar mit den Sektoren von normalen Festplatten bzw. Partitionen. Es ist nicht möglich, die Grösse der PEs nachträglich zu verändern. Dies ist aber auch nicht nötig, da die Anzahl der PEs seit LVM2 unbegrenzt ist, die Standardgrösse beträgt 4 MByte.

Physical Volume: Ein Physical Volume (**PV**) ist der eigentliche Datenspeicher eines LVMs. Ein PV kann eine Partition oder ein ganzes Laufwerk sein. Es ist auch möglich, sogenannte "Loop-Dateien" zu verwenden.

Volume Group: Eine Volume Group (**VG**) ist eine Art Container für mindestens ein PV.

Logical Volume: Ein Logical Volume (**LV** oder auch Volume) ist im LVM Kontext für den Anwender eine Art Partition innerhalb einer Volume Group. Es ist nicht möglich Logical Volumes ausserhalb einer VG zu erstellen. Das LV kann wie eine normale Partition verwendet werden, d.h. es können Dateisysteme darauf angelegt werden und es kann gemounted werden.

Skalierbarkeit: Im LVM können **alle Elemente** (PV, VG, LV), ausser der PE, im laufenden Betrieb vergrössert und verkleinert werden. Doch die Erweiterbarkeit des LVM allein hilft noch nicht weiter, auch das darin liegende Dateisystem muss verändert werden, um es tatsächlich an die neue Grösse anpassen zu können. Nicht jedes Dateisystem beherrscht das Vergrössern oder Verkleinern. Weitere Informationen sind unter Dateisystemgrössen verändern verfügbar.

Erstellen eines neuen LVM Konstruktes

Zu Beginn, muss man sich überhaupt erst klar werden, wo man sich sein neues LVM-Volumen aufbauen will. Um also erst einen Überblick über das aktuelle System zu erlangen, kann das Tool: **lsblk** verwendet werden:

```
# lsblk
```

```
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0    0  450G  0 disk
├─sda1                              8:1    0   200M  0 part /boot/efi
├─sda2                              8:2    0    1G    0 part /boot
└─sda3                              8:3    0 448.8G  0 part
   └─cl_server-root 253:0    0 442.9G  0 lvm  /
      └─cl_server-swap 253:1    0   5.9G  0 lvm  [SWAP]
sdb                                  8:0    0  200G  0 disk
sdc                                  8:0    0  200G  0 disk
```

Wie wir hier sehen, sind momentan die Disks **sdb** und **sdc** noch nicht in Gebrauch. Deshalb werden wir nun auf diesen zwei unser neues LV-Volumen schritt für schritt aufsetzen.

- Zu Beginn muss auf den Disken einmal das **Physical Volume (PE)** Datei-label erstellt werden. Dies geschieht mit folgendem command:

```
# pvcreate /dev/sdb /dev/sdc
```

```
Physical volume "/dev/sdb" successfully created  
Physical volume "/dev/sdc" successfully created
```

- Als nächstes, erstellen wir *aus den zwei PV's* unseren neue **Volume Group** mit dem Namen "vg_alpha":

```
# vgcreate vg_alpha /dev/sdb /dev/sdc
```

```
Volume group "vg_alpha" successfully created
```

Die soeben erstellte VG, verfügt nun über den gesamten Speicherplatz der beiden PVs (in PE). Das heisst, die Volume Group vg_alpha ist momentan rund **400GB** gross!

- Nun können wir auch schon bereits unser erstes **LV-Volumen** in der VG vg_alpha erstellen:

```
# lvcreate -n hercules -L 100G vg_alpha
```

```
Logical volume "hercules" successfully created
```

Formatieren und einbinden des neuen LV-Volumen

Mit dem nachfolgenden Ablauf, wird garantiert, dass die neu erstellten Logical Volumen auch korrekt beim System eingebunden werden, sprich erst einmal überhaupt verwendet werden können.

1. **Formatieren** des neuen LV-Volumes "hercules" mit **XFS**:

```
# mkfs -t xfs /dev/vg_alpha/lv_hercules
```

2. **Erstellen eines mount-Ordners** um das Volumen zu mounten:

```
# mkdir /mnt/hercules
```

3. Passender Eintrag zum **automatischen mounten** beim Systemstart in die /etc/fstab Datei ergänzen:

```
# vim /etc/fstab
```

```
# /etc/fstab
# Created by anaconda on Thu Jun 29 10:01:07 2017
#
# 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
#
/dev/mapper/cl_server-root                /                    xfs
defaults                                0 0
UUID=e6e17315-a455-4d57-851f-10ed1f90e4a9 /boot                xfs
defaults                                0 0
UUID=3C13-8534                            /boot/efi            vfat
umask=0077,shortname=winnt 0 0
/dev/mapper/cl_server-swap                swap                 swap
defaults                                0 0

/dev/vg_alpha/lv_hercules                /mnt/hercules        xfs
defaults                                0 0
#
```

4. Nun können mit “mount -a” alle **nicht gemounteten filesystems**, welche in der Datei /etc/fstab festgelegt sind/wurden gemountet werden!

```
# mount -a
```

5. **Das neue LV-Volume, sollte nun mit df -h gemountet und ersichtlich sein!**

Administration der LVM Volumen

Verkleinern eines LV's

In diesem Beispiel, soll das LV-Volume /dev/vg_alpha/lv_hercules um 512MB verkleinert werden!

```
# lvreduce -L -512M /dev/vg_alpha/lv_hercules -r
```

ACHTUNG: Hier darf die Option -r NICHT vergessen gehen, da mit jenem Schalter, dass Filesystem noch vor dem effektiven Verkleinern automatisch vor-verkleinert wird; würde beim weglassen dieser Option das Filesystem im wahrsten Sinne des Wortes “abgehackt” und würde somit korrupt!

Vergrössern eines LV's

Hier, soll das LV-Volume /dev/vg_alpha/lv_hercules um 2GB vergrössert werden! Dies geht entweder in zwei separaten Befehlen, wo man zuerst das LV-Volume vergrössert und anschliessend erst das Filesystem:

```
# lvextend -L +2G /dev/vg_alpha/lv_hercules

# xfs_growfs /dev/vg_alpha/lv_hercules           //Nur für das xfs
filesystem geeignet!
# resize2fs /dev/vg_alpha/lv_hercules           //Nur für das
ext4/ext3/ext2 filesystem geeignet!
```

Oder beides, automatisch in einem Befehl kombiniert: (-r → --resizefs)

```
# lvextend -L +2G /dev/vg_alpha/lv_hercules -r
```

Tipp: Soll das LV-Volumen mit allem noch zur verfügbaren freien Speicherplatz in der VG erweitert werden, kann folgender Befehl verwendet werden:

```
# lvextend -l +100%FREE /dev/vg_alpha/lv_hercules -r
```

Entfernen einer defekten Disk aus einer VG

Fall einmal der Server einen **SMART-Error** auf einer der im LVM verwendeten Disk entdeckt; so sollte die Disk unbedingt nicht mehr länger produktiv verwendet werden! Um Datenverlust zu verhindern, sollte diese also möglichst schnell ersetzt werden.

Wichtig: Da die Daten in einer VG / LV willkürlich über alle Disken verteilt werden und somit auch alle Disken für den Betrieb benötigt werden; müssen bestehende Daten zuerst von der defekten Disk migriert werden!

1. Verschieben der Physical Extents (PE's) mit dem **Tool: pvmove**

```
# pvmove /dev/sdb
```

Mit diesem Befehl, werden als erstes, **sämtliche beschriebenen PE's** (Daten) auf der Disk /dev/sdb zu anderen **freien PE's** in der **selben Volume Group** verschoben!

2. Defekte Disk aus Volume Group entfernen:

```
# vgreduce vg_alpha /dev/sdb
```

Die Disk /dev/sdb ist nach fertigstellen dieses Commands nicht mehr Teil der Volume Group vg_alpha und kann somit aus dem Server ausgebaut / ersetzt werden!

Hinzufügen einer neuen Disk zu einer bestehenden VG

Um eine Weitere Disk zu einer bestehenden Volume Group hinzuzufügen, wird nach folgendem Muster vorgegangen:

1. Neue Disk in den Server Einbauen und durch lsblk überprüfen, ob diese auch erkannt wird.
2. Durch die Eingabe vom korrekten pvcreate Befehl, die neue Disk mit dem LVM-Dateisystem (PV's) → PE's vorbereiten:

```
# pvcreate /dev/sdb
```

Dies wird die neue Festplatte komplett leeren und mit dem PV Label überschreiben!

3. Nun kann die neue Disk zur bestehenden VG vg_alpha hinzugefügt werden:

```
# vgextend vg_alpha /dev/sdb
```

Dies erweitert nun unsere vg_alpha mit dem zusätzlichen Storage der neuen Disk!

4. Der neu hinzugewonnene Speicherplatz, kann nun an bestehende LV's verteilt werden oder auch genutzt werden um ein eigenes, neues LV zu erzeugen.

Löschen eines einzelnen LV's

Bevor ein **LV-Volumen** gelöscht wird, muss dies zuerst **zwingend ungemountet** werden! Anschliessend, kann es durch folgenden Befehl gelöst werden:

ACHTUNG, DIES KANN NICHT RÜCKGÄNGIG GEMACHT WERDEN!

```
# lvremove /dev/vg_alpha/hercules
```

Entfernen eines kompletten LVM Konstruktes

1. Zu Beginn, müssen alle Daten gesichert werden, welche **NICHT VERLOREN gehen sollen!**
2. Dann, kann als erster Schritt das abzubauenende Volume ungemounted werden:

```
# umount /mnt/hercules
```

3. Im nächsten Schritt, wird das logical volume gelöscht:

```
# lvremove /dev/vg_alpha/hercules
```

4. Nun wird die volume group entfernt:

```
# vgremove vg_alpha
```

Das VG's physische Volumen wird somit gelehrt und steht so bereits wieder für neue VG's zur Verfügung!

5. Wird die Festplatte/Festplatten anschliessend nicht mehr für neue VG's verwendet, empfiehlt es sich, auch dort die noch bestehenden PV's auf ALLEN Disk zu entfernen! Dies würde durch folgenden Befehl erledigt:

```
# pvremove /dev/vda2 /dev/vdb1
```

Somit sind nun auch die PV Metadaten der alten Disken gelöscht und sie können nun entfernt; sprich erneut formatiert werden!

6. **Zu guter letzt, sollte nicht vergessen werden, den noch bestehenden Eintrag aus der /etc/fstab zu entfernen!**
-

Redhat Dokumentation zum Thema

red_hat_enterprise_linux-7-logical_volume_manager_administration-en-us.pdf

Weitere Links

- https://wiki.ubuntuusers.de/Logical_Volume_Manager/
- https://www.thomas-krenn.com/de/wiki/LVM_Grundlagen
- <https://linuxconfig.org/linux-lvm-logical-volume-manager>

Last update: **2019/02/21 10:40**