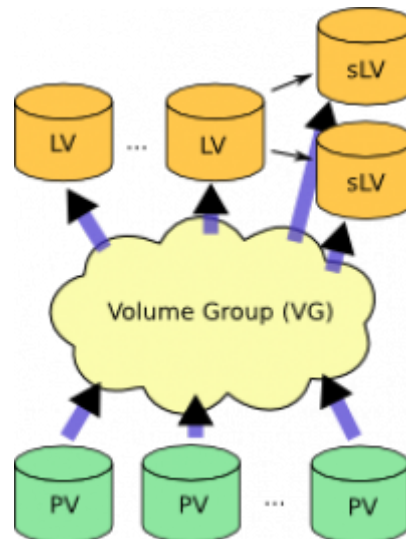


Aufbau eines logical volumes (LV)

Ein LV besteht aus mehreren Ebenen. Die unterste Ebene bilden die sogenannten **physical volumes** (PV).

Mehrere PVs kann man nun zusammenfassen zu einer logischen Einheit, der sogenannten **volume group** (VG).

Auf der VG kann man nun **logical volumes** (LV) erzeugen, die eine beliebige Größe haben können. In der Summe können die LVs die Größe der VG natürlich nicht überschreiten.



LVM einrichten

Pakete installieren (Debian):

```
apt install lvm2 parted
```

LVM-Partition anlegen:

```
parted /dev/sda print
parted /dev/sda set 2 lvm on
parted /dev/sda set 3 lvm on
```

Alte Metadaten löschen:

```
wipefs -af /dev/sda2
wipefs -af /dev/sda3
```

Partitionen anzeigen:

```
fdisk -l | grep -i lvm
```

in der Ausgabe sollten 2 LVM-Partitionen erscheinen:

```
/dev/sda2  30212096  59572223  29360128    14G Linux LVM
/dev/sda3  59572224  90114190  30541967   14,6G Linux LVM
```

LV anlegen

erstes Physical Volume anlegen:

```
pvcreate /dev/sda2
```

Bestehende Physical Volumes anzeigen:

```
pvs
```

Volume Group mit Namen `vg_system` anlegen: ¹⁾

```
vgcreate vg_system /dev/sda2
```

Volume Group aktivieren: (optional)

```
vgchange -a y vg_system
```

Bestehende Volume Groups anzeigen:

```
vgs
```

Logical Volume mit Namen `lv_test` anlegen:

```
lvcreate --size 5G --name /dev/vg_system/lv_test
```

Bestehende Logical Volumes anzeigen:

```
lvs
```

Formatieren (hier mit ext4-Filesystem):

```
mkfs.ext4 /dev/vg_system/lv_test
```

Testen

Mount-Point erstellen:

```
mkdir /mnt/lv_test
```

logical volume an Mount-Point in Dateisystem einhängen:

```
mount /dev/vg_system/lv_test /mnt/lv_test
```

Testweise Daten aus /usr/bin-Verzeichnis kopieren:

```
cp -a /usr/bin/. /mnt/lv_test
```

LV vergrößern / verkleinern

Der Vorteil eines LV ist, daß es zur Laufzeit vergrößert werden kann. Ein LV kann auch verkleinert werden, dass geht aber nur bei ext4 und auch da nur nach umount.

Logical Volume & Dateisystem vergrößern

inkl. Dateisystem

(geht nicht bei btrfs)

```
lvresize --size +1g --resizefs /dev/vg_system/lv_test
```

Volume und Dateisystem in zwei Schritten

```
lvextend -L +2g /dev/vg_system/lv_test
```

Dateisystem vergrößern:

ext3/4

```
resize2fs /dev/vg_system/lv_test
```

xfs

```
xfs_growfs /mnt/lv_test
```

btrfs

(Volume muss nach /mnt/lv_test gemountet sein)

```
btrfs filesystem resize max /mnt/lv_test
```

Snapshot erstellen

2)

```
lvcreate --snapshot --size 100M --name lv_snapshot1 /dev/vg_system/lv_test
mkdir /mnt/snapshot1
mount -o ro /dev/vg_system/lv_snapshot1 /mnt/snapshot1
```

3)

Snapshot entfernen

wie jedes lv

```
umount /mnt/snapshot1
lvremove /dev/vg_system/lv_snapshot1
```

Physical Volume entfernen

zweites Physical Volume anlegen:

```
pvcreeate /dev/sda3
vgextend vg_system /dev/sda3
```

```
pvmove /dev/sda2
vgreduce vg_system /dev/sda2
pvremove /dev/sda2
```

Logical Volume spiegeln (RAID 1)

zweites Physical Volume anlegen:

```
pvcreeate /dev/sda2
vgextend vg_system /dev/sda2
```

Logical Volume zu RAID 1 konvertieren:

```
lvconvert --type raid1 --name lv_test vg_system
journalctl -kf
```

(warten bis „recovery done“ erscheint)

4)

1)

- (Minus) in VG und LV Namen besser vermeiden

2)

Falls das Kernel Modul dm-snapshot noch nicht geladen ist:

```
modprobe dm-snapshot
```

3)

bei XFS

```
mount -o ro,nouuid /dev/vg_system/lv_snapshot /mnt/snapshot1
```

oder (falls es ohne norecovery nicht geht)

```
mount -o ro,nouuid,norecovery /dev/vg_system/lv_snapshot /mnt/snapshot1
```

4)

Unter CentOS 8 testen:

```
lvconvert --raidintegrity y /dev/vg_system/lv_test  
blkdiscard /dev/sda6
```

From:

<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:

<https://wiki.lab.linuxhotel.de/doku.php/lpi1:lvm>

Last update: **2022/05/05 11:48**

