

# Festplattenverwaltung - fdisk,lvm,dd

## Partitionstabelle kopieren

```
dd if=/dev/sdc of=/dev/sdd bs=512 count=1
```

Parameter erklärt:

if -> input device

of -> output device

bs -> blocksize

count -> Anzahl der Blöcke, die kopiert werden sollen

## Festplatte vergrößern

### .. mit LVM

Das Programm "fdisk" für die Festplatte starten:

```
/# fdisk /dev/sda

Welcome to fdisk (util-linux 2.33.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): p
Disk /dev/sda: 25 GiB, 26843545600 bytes, 52428800 sectors
Disk model: QEMU HARDDISK
```

```
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xb5d10c66
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1	*	2048	1953791	1951744	953M	83	Linux
/dev/sda2		1953792	52428799	50475008	24,1G	5	Extended
/dev/sda5		1955840	52428799	50472960	24,1G	8e	Linux LVM

Dann sieht man alle aktuellen Partitionen.

Ich habe mich beim Einrichten für folgendes Setup entschieden:

```
-- /dev/sda
|- /dev/sda1 -> /boot
|- /dev/sda2 -> LVM partition
|- /dev/sda5 -> volumegroup (vg01)
  |-vg01 -> logical volume swap (lvswap)
  |-vg01 -> logical volume system (lvsys)
```

Zuerst muss man die Festplatte an sich vergrößern.

Da ich die Virtualisierungssoftware "Proxmox" einsetze ist das total einfach.

Entweder per CLI `qm resize VMID FestplattenID +5G` oder man nutzt die grafische Weboberfläche und fügt über den Punkt "Hardware" -> entsprechende Festplatte -> Knopf oben in der Leiste "resize disk" einfach die Größe hinzu.

Danach kann man sich die neue Größe mit `fdisk -l` anzeigen lassen/kontrollieren.

Nun muss man die bestehenden Partitionen mittels `fdisk /dev/sdX` (X steht für euren Festplatten Identifier Buchstabe) löschen und 1 primäre über die gesamte neue Größe und eine logische über die gesamte neue Größe erstellen.

Am Ende noch die Änderungen speichern mit dem Befehl "w" für "write" und dann noch mittels "partprobe" die neuen Festplatteninformationen einlesen lassen.

Dann kann man mit "vgdisplay" überprüfen ob es geklappt hat und es geht weiter mit dem "lvextend".

Da ich die gesamte Größe haben will kann ich einfach sagen er soll alles nehmen was frei ist:

```
lvextend -l +100%FREE /dev/vg01/lvsys
```

Und mit "lvsdisplay" sehe ich nun meine neue Größe:

```
/# lvsdisplay
--- Logical volume ---
LV Path                /dev/vg01/lvswap
LV Name                lvswap
VG Name                vg01
LV UUID                04chsG-gbxM-RrZZ-1wLv-Nj48-tPgE-0GmDIv
LV Write Access        read/write
LV Creation host, time debian, 2020-02-17 17:03:48 +0100
LV Status              available
# open                 2
LV Size                <1,86 GiB
Current LE             476
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    256
Block device           254:1

--- Logical volume ---
LV Path                /dev/vg01/lvsys
LV Name                lvsys
VG Name                vg01
LV UUID                nUBuD- WQ2c-wiMW-fXyS-tvoa-ELZB-rcUA6H
LV Write Access        read/write
LV Creation host, time debian, 2020-02-17 17:05:38 +0100
LV Status              available
# open                 1
LV Size                <22,21 GiB
Current LE             5685
Segments               1
Allocation             inherit
Read ahead sectors     auto
- currently set to    256
Block device           254:0
```

