

# Logical Volumes und RAID

mit Linux-Bordmitteln  
Logical Volume Manager (LVM) und mdadm

Linux-Cafe - 2016-08-01

Bernd Strößenreuther  
<[linux-cafe@stroessenreuther.net](mailto:linux-cafe@stroessenreuther.net)>

# Lizenz

Sie dürfen dieses Dokument verwenden unter den Bedingungen der Creative Commons Lizenz:

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>

Alle Grafiken und Icons – soweit nicht anders angegeben – von OpenClipArt.org "released to the public domain".

# Agenda

- Teil 1: Logical Volume Manager (LVM)
  - Einstieg: Warum will ich Logical Volumes?
  - Etwas Theorie
  - Übung: Alles selber ausprobieren
- Teil 2: Mehr Ausfallsicherheit mit RAID
  - Warum will ich RAID? Warum will ich Soft-RAIDs?
  - Etwas Theorie
  - Übung: Alles selber ausprobieren
- Beispiel-Setup für einen kleinen Server

# Abgrenzung

- Nur klassisches Volume- und RAID-Management
- Kein BTRFS, kein ZFS

# Begriffsklärung

- LVM: Logical Volume Manager
  - Die Größe meiner Platten und die Größe meiner gewünschten Filesysteme passen ggf. nicht zusammen.
  - Mit LVM kann eine Abstraktionsebene dazwischen eingezogen werden
- RAID: Redundant Array of Independent Disks  
oder auch: Redundant Array of Inexpensive Disks
  - Ausfallsicherheit erhöhen
- mdadm: multiple disk administration
  - Erzeugt Soft-RAIDs unter Linux

# Teil 1: LVM

# Filesysteme

- Platten-Devices:
  - (komplette) Platte (/dev/sda)
  - Partition (/dev/sda1)
- Ausgangssituation: Jedes Device ist mit einem Filesystem formatiert
- Jedes Device ist unter einem bestimmten Mountpoint ins System eingehängt (/home, /var, ...)
- Der freie Platz auf jedem Device steht dem jeweiligen Filesystem exklusiv zur Verfügung

# Filesystem-Layouts: Client-PC

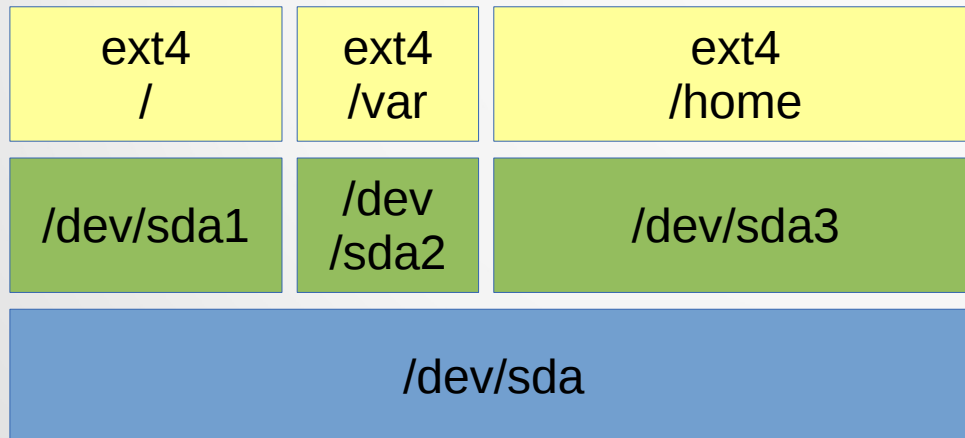
- Einsteiger-Setup:
  - Eine Festplatte oder SSD
  - Oft alles in einer Partition
  - Keine Vorteile durch LVM
- Mehrere Filesysteme
  - Im Zweifelsfall:  
Freier Platz immer an der falschen Stelle
- Mehrere Platten-Devices verbaut:
  - Ohne LVM: Zwangsweise mehrere Filesysteme, mindestens ein Filesystem pro Platte



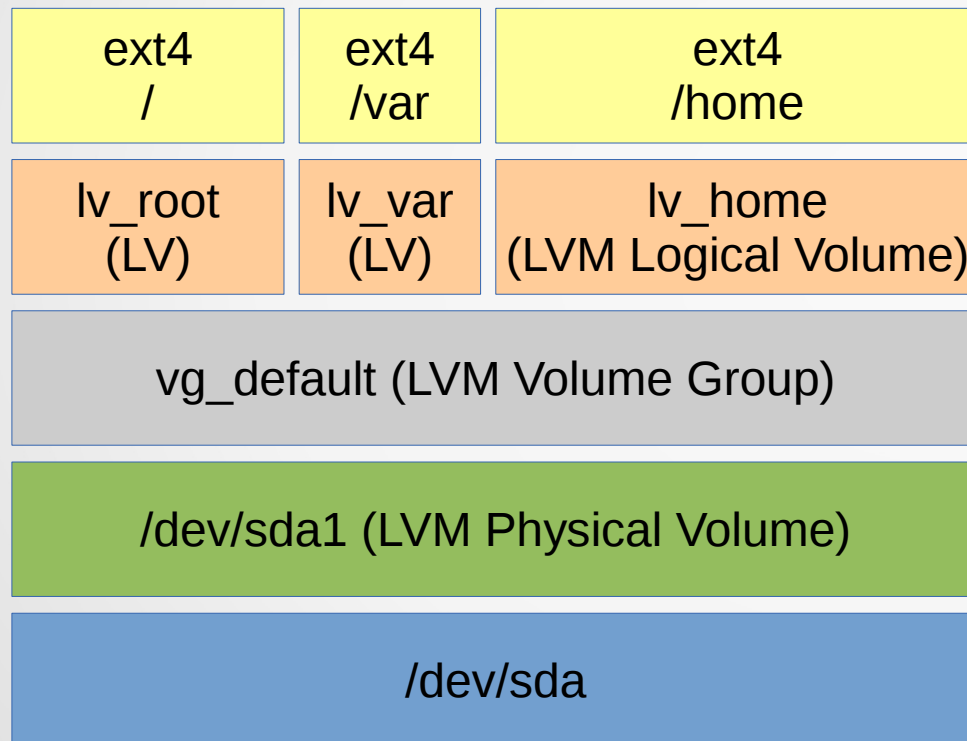
# Filesystem-Layouts: „enhanced“

- Was passiert, wenn das root-Filesystem vollläuft?
  - Im schlimmsten Fall: System bleibt stehen
- Risiko deutlich minimierbar durch sinnvolles Filesystem-Layout:  
mindestens /var, /tmp und /home als eigene Filesysteme
- Ggf. Compliance-Anforderung:  
Audit-Logs in separatem Filesystem

# Ausgangssituation

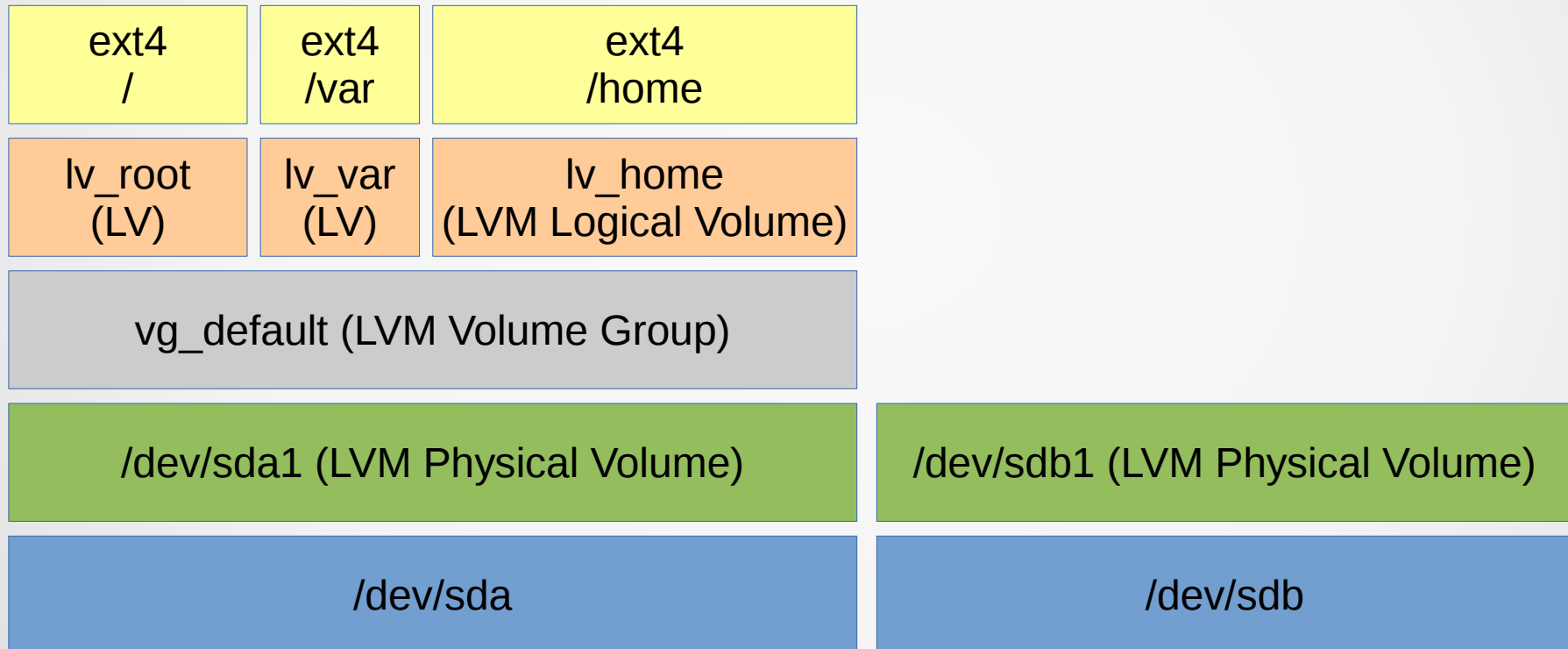


# Mit LVM

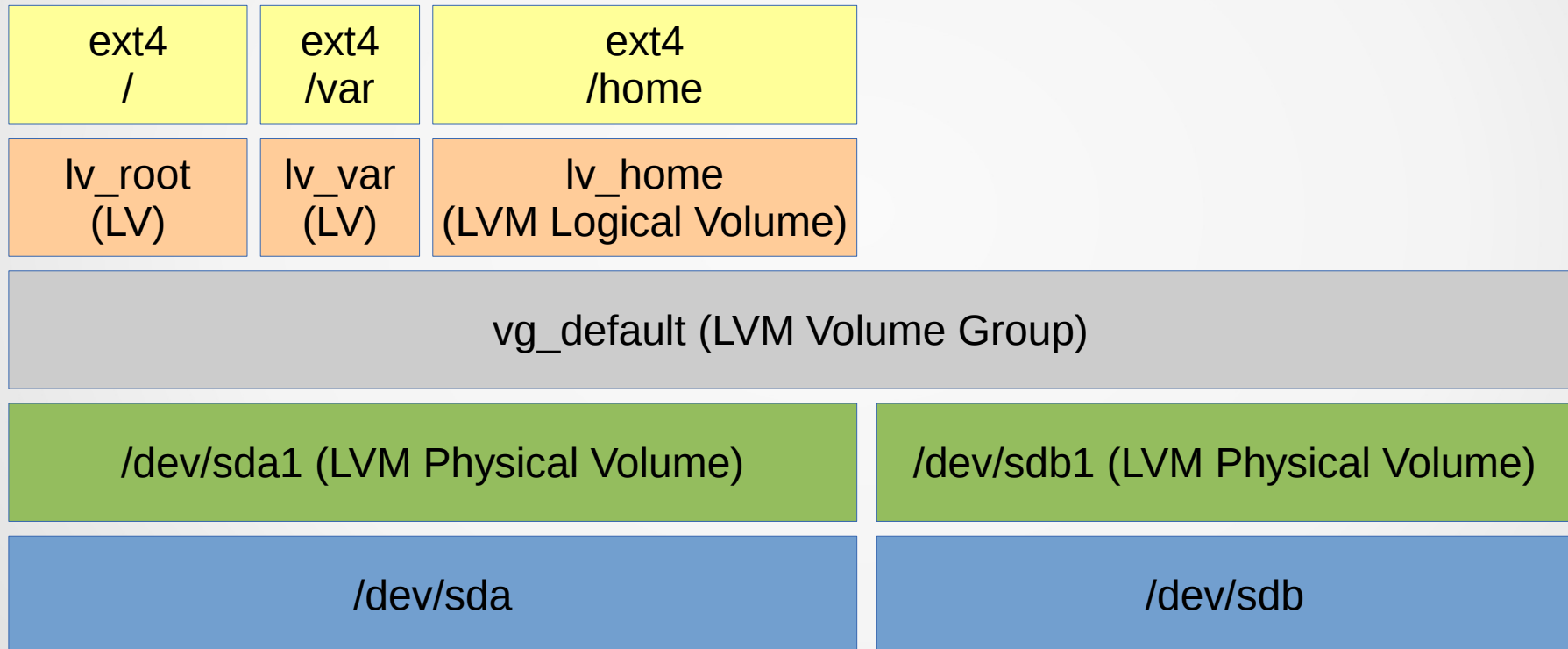


Vergrößern und  
verkleinern von  
Logical Volumes  
möglich

# Zusätzliche Festplatte

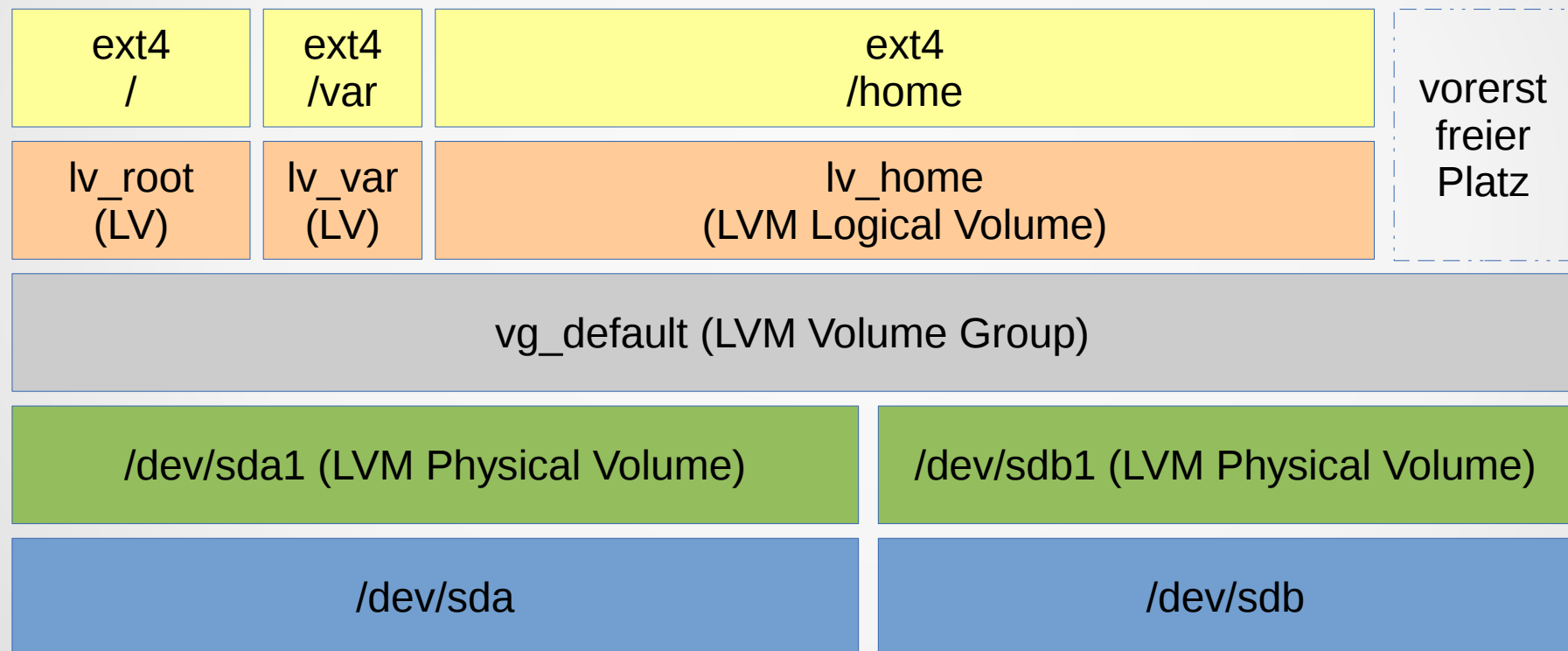


# Zusätzliche Festplatte

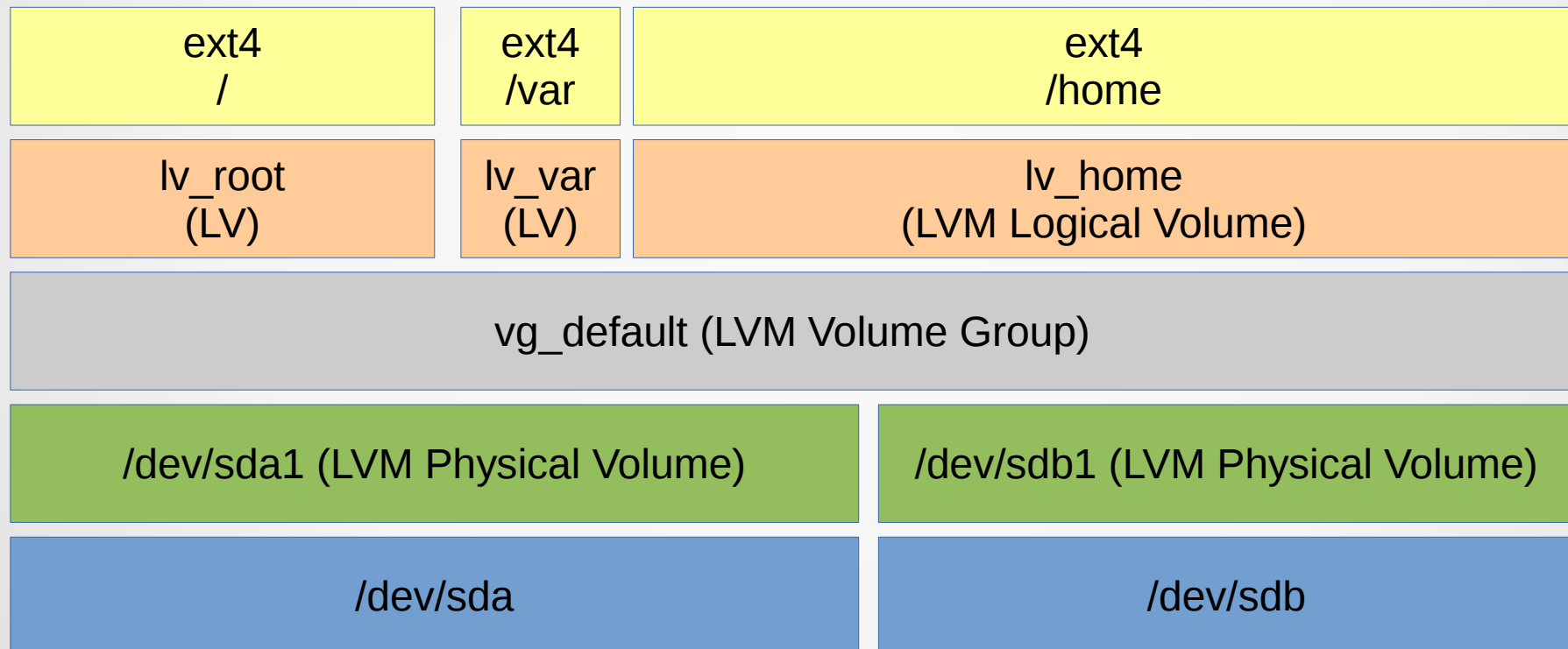


# Zusätzliche Festplatte

Ein LV kann größer sein als ein PV



# Zusätzliche Festplatte



# Doku

- LVM HowTo vom Linux Documentation Project:  
<http://tldp.org/HOWTO/LVM-HOWTO/>



# Übung: LVM

Wir stellen unserer Linux-VM noch eine zweite Festplatte zur Verfügung

```
fdisk -l
```

zeigt jetzt eine weitere Platte ohne Partitionen an,  
vermutlich /dev/sdb

Darauf legen wir eine Partition vom Typ 8e (Linux LVM) an  
(über die komplette Größe)

# Übung: Partition anlegen

```
fdisk /dev/sdb
```

```
Command (m for help): p           # Partitionen anzeigen
Command (m for help): n           # neue Partition erstellen
Select (default p): p             # primary
Partition number (1-4, default 1): 1 # Partition 1
First sector (2048-16777215, default 2048):· # default (Return)
Last sector, +sectors or +size{K,M,G,T,P}
(2048-16777215, default 16777215):· # default (Return)
Command (m for help): t           # Typ setzen
Hex code (type L to list all codes): L # alle Typen auflisten
Hex code (type L to list all codes): 8e # Linux LVM
Command (m for help): p           # nochmal anzeigen
Command (m for help): w           # schreiben
```

# Übung: Volume Group anlegen

```
# /dev/sdb1 zu einem Physical Volume machen  
pvcreate /dev/sdb1  
  
# Volume Group anlegen  
vgcreate vg_default /dev/sdb1  
  
# Volume Group anzeigen  
vgdisplay
```

# Übung: Logical Volume anlegen

```
# logical Volume anlegen
lvcreate -L2G -nlv_opt vg_default
# anzeigen
lvdisplay
vgdisplay
# Filesystem erzeugen
mkfs -t ext4 /dev/vg_default/lv_opt
mount /dev/vg_default/lv_opt /opt
```

# Übung: Logical Volume benutzen

```
# alten Inhalt wegsichern / im alten FS löschen
tar cvzf /tmp/opt.tgz /opt/*
rm -Rf /opt/*

# LV device einhängen
mount /dev/vg_default/lv_opt /opt

# rücksichern
cd /

tar xvzf /tmp/opt.tgz && rm /tmp/opt.tgz
```

# Übung: Logical Volume vergrößern

```
df -h /opt/
```

```
# wir merken uns mal die Größe
```

```
lvextend -L+1G /dev/vg_default/lv_opt
```

```
df -h /opt/
```

```
# hat nicht geholfen?
```

```
lvdisplay
```

```
resize2fs /dev/vg_default/lv_opt
```

```
df -h /opt/
```

# Übung: Volume Group erweitern

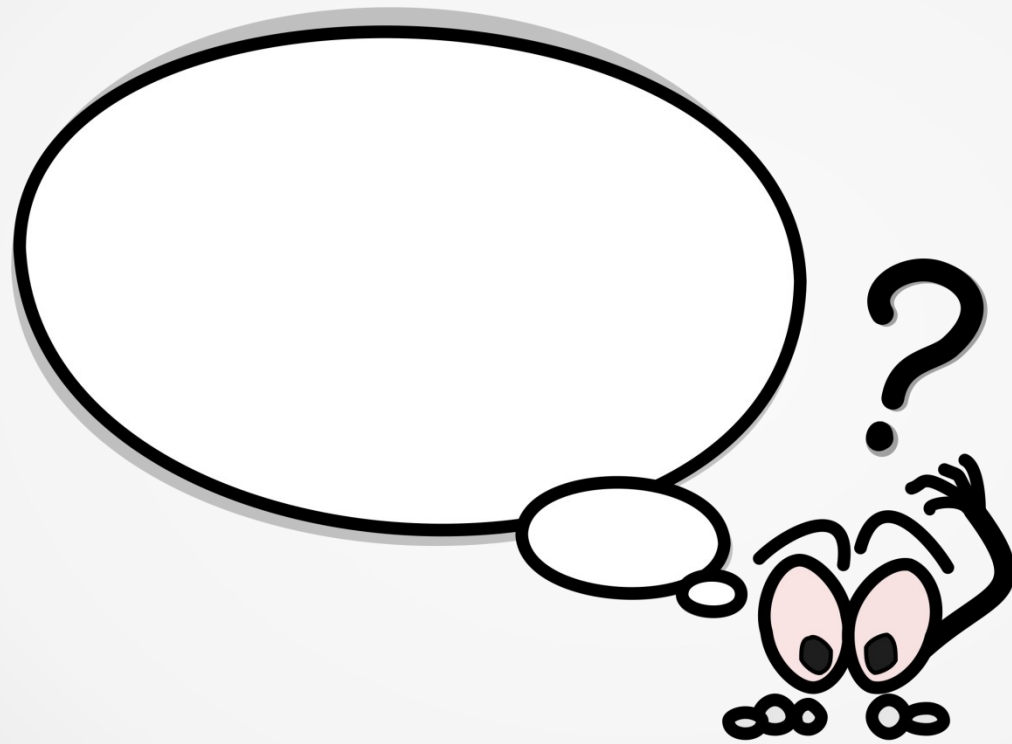
```
# Plattendevise aussuchen aus:  
fdisk -l  
  
# Physical Volume kann auch eine ganze Platte sein  
pvcreate /dev/sdc  
vgextend vg_default /dev/sdc  
  
# Ergebnis anschauen  
pvdisplay  
vgdisplay  
  
# Platz irgendwie sinnvoll nutzen  
lvextend -L+9G /dev/vg_default/lv_opt  
resize2fs /dev/vg_default/lv_opt
```

# Hint: ggf. Filesystemcheck nötig

```
root@debianix:~# resize2fs /dev/vg_default/lv_opt
resize2fs 1.42.12 (29-Aug-2014)
Please run 'e2fsck -f /dev/vg_default/lv_opt' first.
root@debianix:~# e2fsck -f /dev/vg_default/lv_opt
e2fsck 1.42.12 (29-Aug-2014)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vg_default/lv_opt: 15/196608 files (0.0% non-contiguous),
29501/786432 blocks
root@debianix:~# resize2fs /dev/vg_default/lv_opt
resize2fs 1.42.12 (29-Aug-2014)
Resizing the filesystem on /dev/vg_default/lv_opt to 3145728 (4k)
blocks.
The filesystem on /dev/vg_default/lv_opt is now 3145728 (4k) blocks
long.
```



Bis hier hin: Noch Fragen?



# Teil 2: RAID

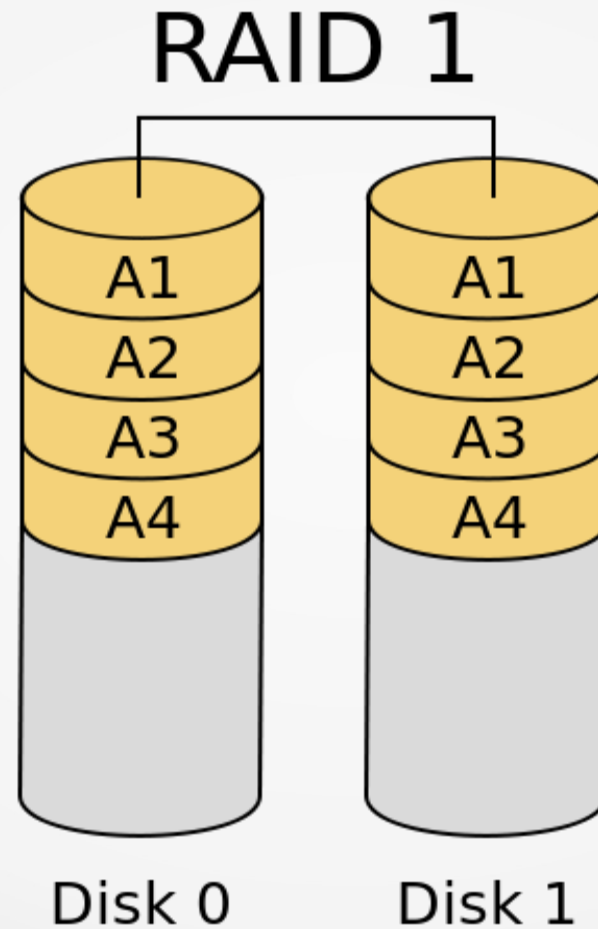
# Warum will ich RAID?

- PC mit nur einem Platten-Device: RAID weder sinnvoll noch möglich
- Ab dem zweiten Platten-Device: Über RAID nachdenken
- Unser Beispiel: LVM über zwei physikalische Platten
  - Risiko für Datenverlust durch Hardware-Defekt hat sich mal eben verdoppelt!!
- RAID: Daten Redundant speichern

# RAID-Level 1: Spiegelung

- Daten werden 1-zu-1 gespiegelt
- Overhead 50%:  
Nur die Hälfte meines physischen Plattenplatzes kann effektiv genutzt werden
- Ab 2 Platten möglich
- Die kleinste Platte bestimmt die Kapazität des RAID
- Solange noch eine Platte lebt, ist das RAID verfügbar
- Monitoring des RAID unbedingt erforderlich!!  
(Der Anwender bemerkt keine Einschränkungen, wenn eine Platte ausfällt.)

# Schaubild RAID 1

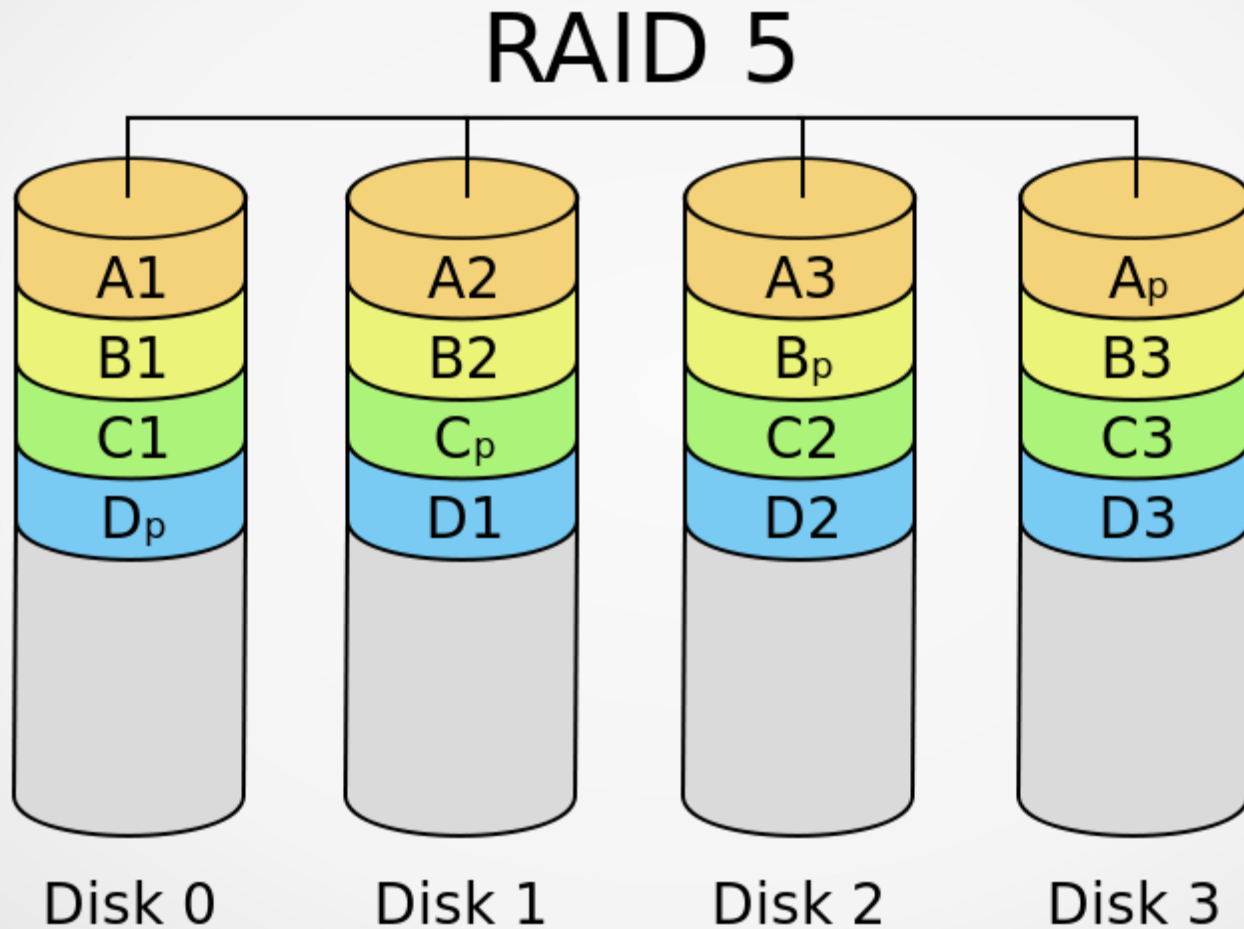


Quelle: [https://de.wikipedia.org/wiki/Datei:RAID\\_1.svg](https://de.wikipedia.org/wiki/Datei:RAID_1.svg), Lizenz: GFDL

# RAID-Level 5: Parity

- Die Kapazität einer Platte wird für Paritätsinformationen benutzt
- Der Ausfall genau einer Platte kann kompensiert werden
- Overhead: eine Platte
- Ab 3 Platten möglich
- Monitoring des RAID erforderlich

# Schaubild RAID 5



Quelle: [https://de.wikipedia.org/wiki/Datei:RAID\\_5.svg](https://de.wikipedia.org/wiki/Datei:RAID_5.svg) Lizenz: GFDL

# Warum will ich Soft-RAID? (1)

- Bei Enterprise-Server-Hardware bildet üblicherweise ein spezieller RAID-Controller die RAID-Logik in Hardware ab
  - Die CPU der Server wird nicht belastet
  - Ersatzteil-Verfügbarkeit über die gesamte Lebensdauer der Server-Hardware ist gewährleistet



# Warum will ich Soft-RAID? (2)

- Für Consumer-Hardware sind ebenfalls RAID-Controller im Angebot
  - Ersatzteil-Verfügbarkeit schon nach kurzer Zeit nicht mehr sichergestellt
  - Jeder Controller schreibt die Daten physikalisch nach seinem eigenen System
  - Controller wird zum Single Point of Failure
  - Wahrscheinlichkeit für Datenverlust erhöht sich!!  
(im schlimmsten Fall sogar höher als ganz ohne RAID!!)
- Lieber Soft-RAID nach offenen Standards
  - CPU-Overhead eher gering

# Übung: Voraussetzungen schaffen

```
# nach den LVM-Übungen erstmal aufräumen
```

```
umount /opt
```

```
vgremove vg_default
```

```
# ggf. mdadm installieren
```

```
aptitude install mdadm
```

# Übung: RAID 1

- Jede Platte bekommt genau eine Partition vom Type fd (→ fdisk)

```
# RAID 1 anlegen mit 2 Platten
```

```
mdadm --create /dev/md0 --level 1 ↻  
      --raid-devices 2  /dev/sdb1 /dev/sdc1
```

```
# überprüfen
```

```
fdisk -l
```

```
#nutzen
```

```
mkfs -t ext4 /dev/md0
```

```
mount /dev/md0 /mnt/
```

# Übung: RAID-Status anzeigen

```
cat /proc/mdstat
```

```
Personalities : [raid1]
```

```
md0 : active raid1 sdc1[1] sdb1[0]  
      8380416 blocks super 1.2 [2/2] [UU]
```

# Übung: RAID 5

```
# Aufräumen nach RAID 1 Übung
```

```
umount /mnt
```

```
mdadm --stop /dev/md0
```

```
# prüfen: ist das Device wirklich weg?
```

```
cat /proc/mdstat
```

```
# RAID 5 anlegen mit 3 Platten
```

```
mdadm --create /dev/md0 --level 5 ↻
```

```
    --raid-devices 3 /dev/sdb1 /dev/sdc1 /dev/sdd1
```

# Übung: nochmal RAID-Status

```
# dritte Platte wird noch synchronisiert
```

```
cat /proc/mdstat
```

```
Personalities : [raid1] [raid6] [raid5] [raid4]
```

```
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
```

```
16760832 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [UU_]
```

```
[=====>.....] recovery = 39.0% (3276304/8380416)
```

```
finish=1.0min speed=82016K/sec
```

```
# RAID vollständig aufgebaut
```

```
cat /proc/mdstat
```

```
Personalities : [raid1] [raid6] [raid5] [raid4]
```

```
md0 : active raid5 sdd1[3] sdc1[1] sdb1[0]
```

```
16760832 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
```

# Übung: LVM auf RAID

```
pvcreate /dev/md0
vgcreate vg_default /dev/md0
vgdisplay
lvcreate -L12G -nlv_opt vg_default
lvdisplay
mkfs -t ext4 /dev/vg_default/lv_opt
mount /dev/vg_default/lv_opt /opt
df -h
```

# Setup bei der Linux-Installation (1)

- Sowohl RAIDs als auch LVM können direkt bei der Installation von Linux im Setup-Wizard angelegt werden

## !!! Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk  
Guided - use entire disk and set up LVM  
Guided - use entire disk and set up encrypted LVM  
Manual

<Go Back>



# Setup bei der Linux-Installation (2)

## [!!] Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

Guided partitioning

Configure software RAID

Configure the Logical Volume Manager

Configure encrypted volumes

Configure iSCSI volumes

LVM VG debianix-vg, LV root - 20.3 GB Linux device-mapper (linear)

#1 20.3 GB f ext4 /

LVM VG debianix-vg, LV swap\_1 - 914.4 MB Linux device-mapper (linear)

#1 914.4 MB f swap swap

SCSI1 (0,0,0) (sda) - 21.5 GB ATA VBOX HARDDISK

#1 primary 254.8 MB f ext2 /boot

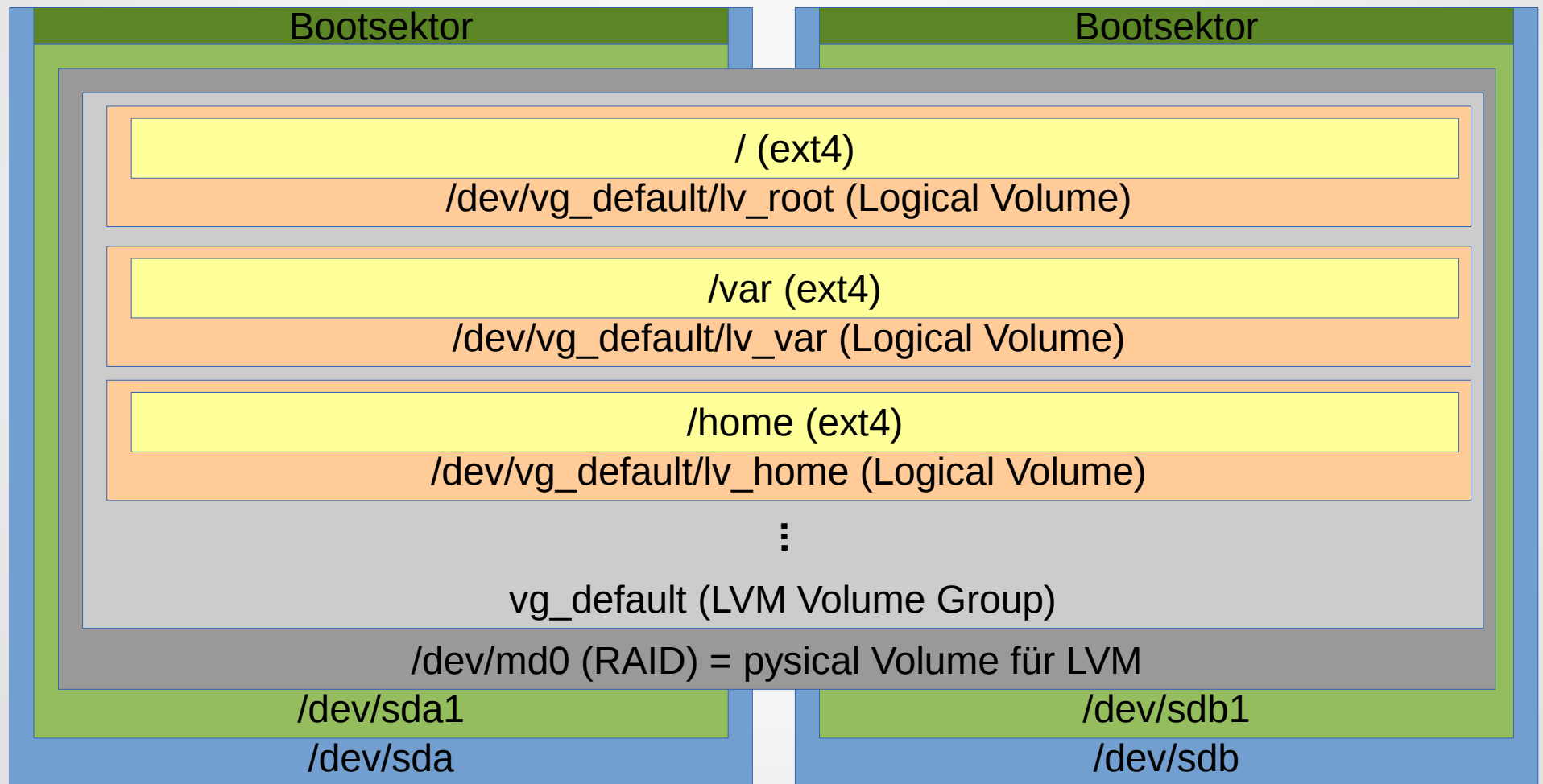
#5 logical 21.2 GB K lvm

Undo changes to partitions

**Finish partitioning and write changes to disk**

<Go Back>

# Beispiel für einen kleinen Server



# Tip: Bootsektor

- Bootsektor immer auch (manuell) auf die zweite Platte installieren:  
`grub-install /dev/sdb`
- Vorteil: System bootet auch bei Ausfall der ersten Platte noch
- Am einfachsten:
  - Script für Upgrade schreiben
  - darin den Befehl immer mit aufrufen

# Literatur

- <https://wiki.ubuntuusers.de/Software-RAID/>
- Unbedingt nutzen:  
LVM HowTo vom Linux Documentation Project:  
<http://tldp.org/HOWTO/LVM-HOWTO/>

# Fragen / Diskussion

