

LVM Logical Volume Manager

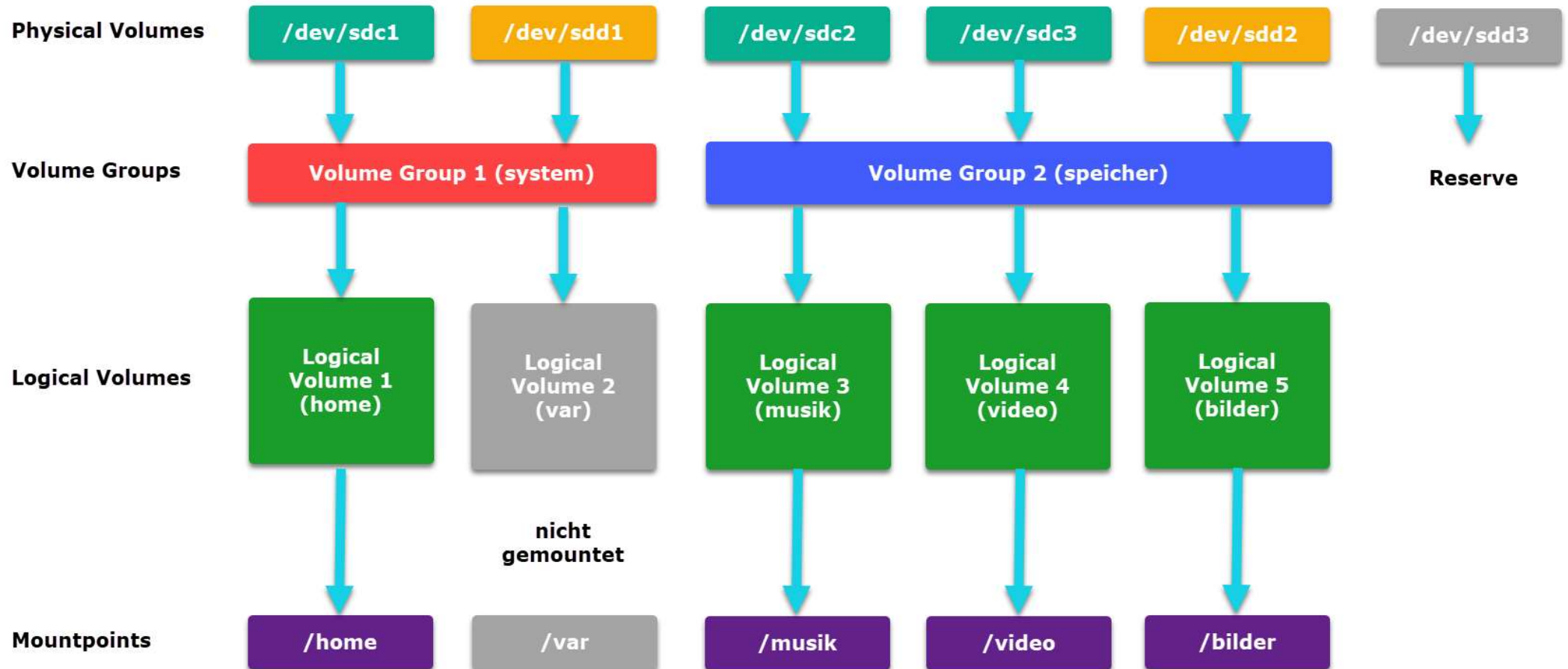
- Um die Speicherverwaltung flexibel zu gestalten, sollten Storages in Linux mittels LVM eingerichtet werden
- Um die nötige Sicherheit zu erreichen, werden diese dann mittels geeigneter RAID-Systeme abgesichert
- Die klassische Partitionierung ist dagegen vollkommen unflexibel und lässt sich nur schwer anpassen, zudem die vorhandenen Anpassungsmöglichkeiten auch noch unsicher sind und nicht selten mit Datenverlust einher gehen
- Letztlich bestimmt dabei die maximale Plattengröße das Maximum der Partition

LVM Projektfestplatten für den Server deb-s1

- Ein willkommener Nebeneffekt beim LVM ist, dass man Geschwindigkeitsgewinne erzielt, wenn die verwendeten Festplatten an verschiedenen Controller-Schnittstellen angeschlossen sind
- Ein ähnliches Prinzip existiert auch unter Windows und wird dort „Dynamische Speicherverwaltung“, genannt
 - Für das Praxisprojekt werden zwei zusätzliche virtuelle Festplatten mit jeweils einer Kapazität von 6 GB in der Hyper-V erstellt.
 - Diese Platten werden dann jeweils in Partitionen zu je ca. 2 GB aufgeteilt.
 - Im produktiven Umfeld werden die Festplatten- und Partitionsgrößen heute im TB-Bereich liegen
 - Bedingt durch die vorhandenen Festplatten ergeben sich die Laufwerke `/dev/sdc` und `/dev/sdd`
 - Die Partitionen heißen dann `/dev/sdc1`, `/dev/sdc2` `/dev/sdc3` sowie `/dev/sdd1`, `/dev/sdd2` und `/dev/sdd3`

LVM-Projekt

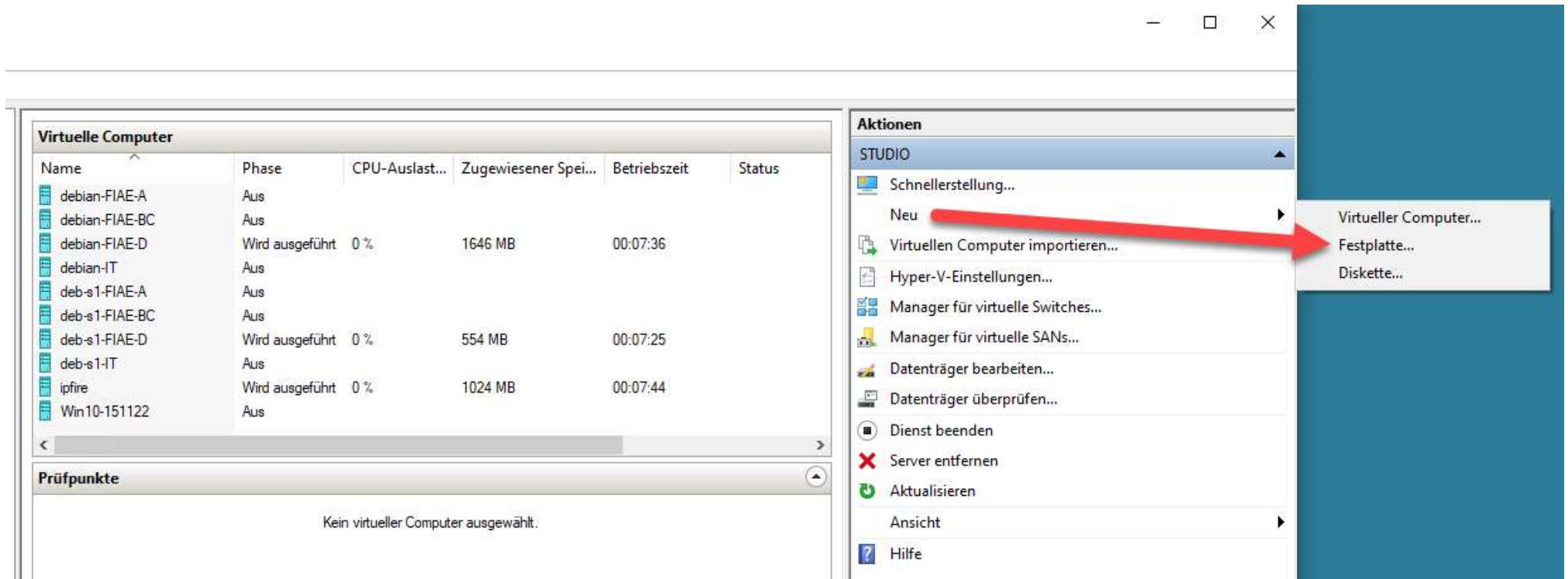
(bitte diese Folie separat ausdrucken, um sie als Vorlage zu nutzen)



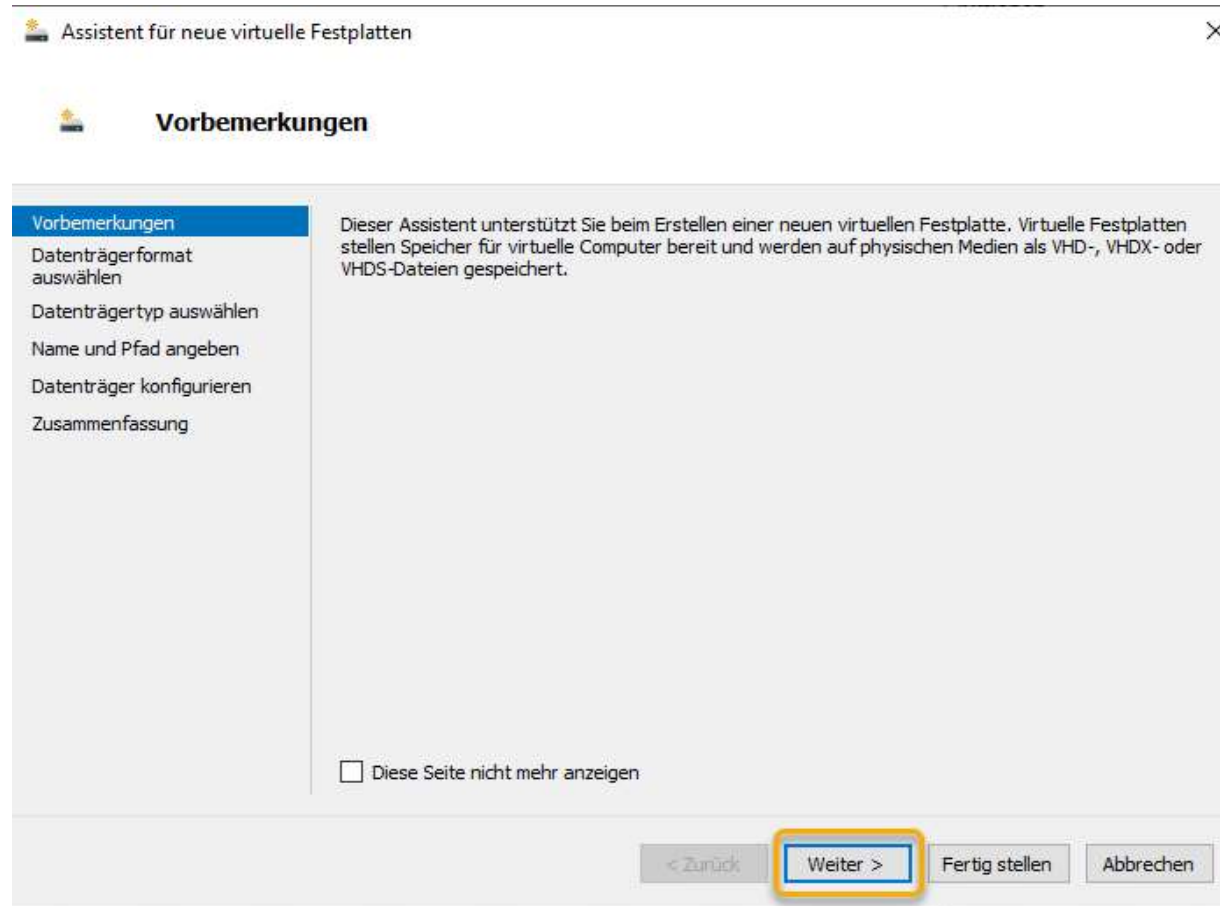
Virtuelle Festplatten erstellen

- Da wir bereits virtuelle Festplatten in der Hyper-V erstellt haben nutze ich in den folgenden Folien nur Bildschirmfotos ohne zusätzliche Erläuterungstexte
- An markanten Stellen finden Sie zusätzliche Elemente, die die Reihenfolge der Bearbeitungsschritte verdeutlichen
- Die Bildschirmfotos sollten selbsterklärend sein

LVM Projektfestplatten erstellen



LVM Projektfestplatten erstellen



LVM Projektfestplatten erstellen

Assistent für neue virtuelle Festplatten

Datenträgerformat auswählen

Vorbemerkungen

- Datenträgerformat auswählen**
- Datenträgertyp auswählen
- Name und Pfad angeben
- Datenträger konfigurieren
- Zusammenfassung

Welches Format soll für die virtuelle Festplatte verwendet werden?

☐ VHD
Unterstützt virtuelle Festplatten mit einer Größe von bis zu 2.040 GB.

☒ VHDX
Dieses Format unterstützt virtuelle Datenträger mit einer Größe von bis zu 64 TB und ist unempfindlich gegen stromausfallbedingte Konsistenzfehler, wird allerdings von Betriebssystemen vor Windows 8 nicht unterstützt.

☐ VHD-Satz
Dieses Format ist ausschließlich für freigegebene virtuelle Festplatten reserviert. Es ermöglicht die Sicherung von Gruppen virtueller Computer mithilfe freigegebener virtueller Festplatten. Das Format wird von Betriebssystemen vor Windows 10 nicht unterstützt.

< Zurück Weiter > Fertig stellen Abbrechen

LVM Projektfestplatten erstellen

Assistent für neue virtuelle Festplatten

Datenträgertyp auswählen

Vorbemerkungen
Datenträgerformat auswählen
Datenträgertyp auswählen
Name und Pfad angeben
Datenträger konfigurieren
Zusammenfassung

Welche Art von virtueller Festplatte möchten Sie erstellen?

☒ Feste Größe
Dieser Datenträgertyp zeichnet sich durch eine höhere Leistung aus und wird für Server empfohlen, auf denen Anwendungen mit hoher Datenträgeraktivität ausgeführt werden. Die Größe der erstellten VHD-Datei entspricht zunächst der Größe der virtuellen Festplatte und bleibt gleich, auch wenn Daten gelöscht oder hinzugefügt werden.

☐ Dynamisch erweiterbar
Dieser Datenträgertyp zeichnet sich durch eine bessere Ausnutzung des physischen Speicherplatzes aus und wird für Server ohne datenträgerintensive Anwendungen empfohlen. Die erstellte VHD-Datei ist zunächst klein und wird geändert, wenn Daten hinzugefügt werden.

☐ Differenzierung
Dieser Datenträgertyp wird über eine hierarchische Beziehung einem anderen Datenträger zugeordnet, der intakt bleiben soll. Da sich Änderungen an Daten oder am Betriebssystem nicht auf den übergeordneten Datenträger auswirken, können Änderungen problemlos wieder rückgängig gemacht werden. Alle untergeordneten Datenträger müssen das gleiche Format für virtuelle Datenträger besitzen wie der übergeordnete Datenträger (entweder VHD oder VHDX).

< Zurück Weiter > Fertig stellen Abbrechen

LVM Projektfestplatten erstellen

Assistent für neue virtuelle Festplatten

Name und Pfad angeben

Vorbemerkungen
Datenträgerformat auswählen
Datenträgertyp auswählen
Name und Pfad angeben
Datenträger konfigurieren
Zusammenfassung

Geben Sie Name und Speicherort für die zweite virtuelle Festplatte an.

Name: LVM1

Pfad: D:\ Durchsuchen...

Nur in meinen Einstellungen!
Sie können Ihren Pfad belassen

Die zweite virtuelle Festplatte erhält den Namen LVM2

< Zurück Weiter > Fertig stellen Abbrechen

LVM Projektfestplatten erstellen

Assistent für neue virtuelle Festplatten

Datenträger konfigurieren

Vorbemerkungen
Datenträgerformat auswählen
Datenträgertyp auswählen
Name und Pfad angeben
Datenträger konfigurieren
Zusammenfassung

Erstellen Sie entweder eine leere virtuelle Festplatte oder kopieren Sie den Inhalt einer vorhandenen physischen Festplatte.

☒ Neue virtuelle Festplatte ohne Inhalt erstellen
Größe: GB (Maximale Größe: 64 TB)

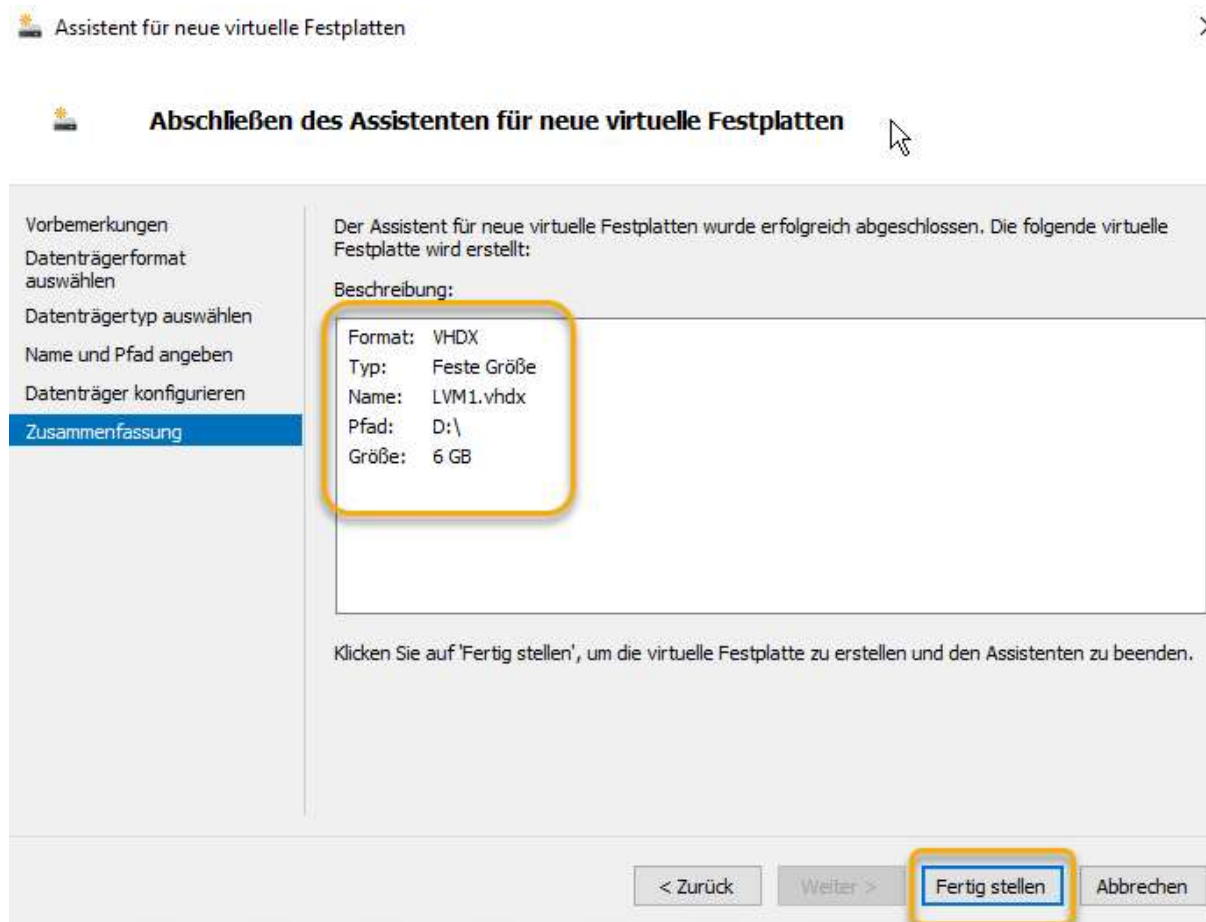
☐ Inhalt der angegebenen physischen Festplatte kopieren:

Physische Festplatte	Größe
\\.\PHYSICALDRIVE0	232 GB
\\.\PHYSICALDRIVE1	476 GB

☐ Inhalt der angegebenen virtuellen Festplatte kopieren
Pfad:

< Zurück **Weiter >** Fertig stellen Abbrechen

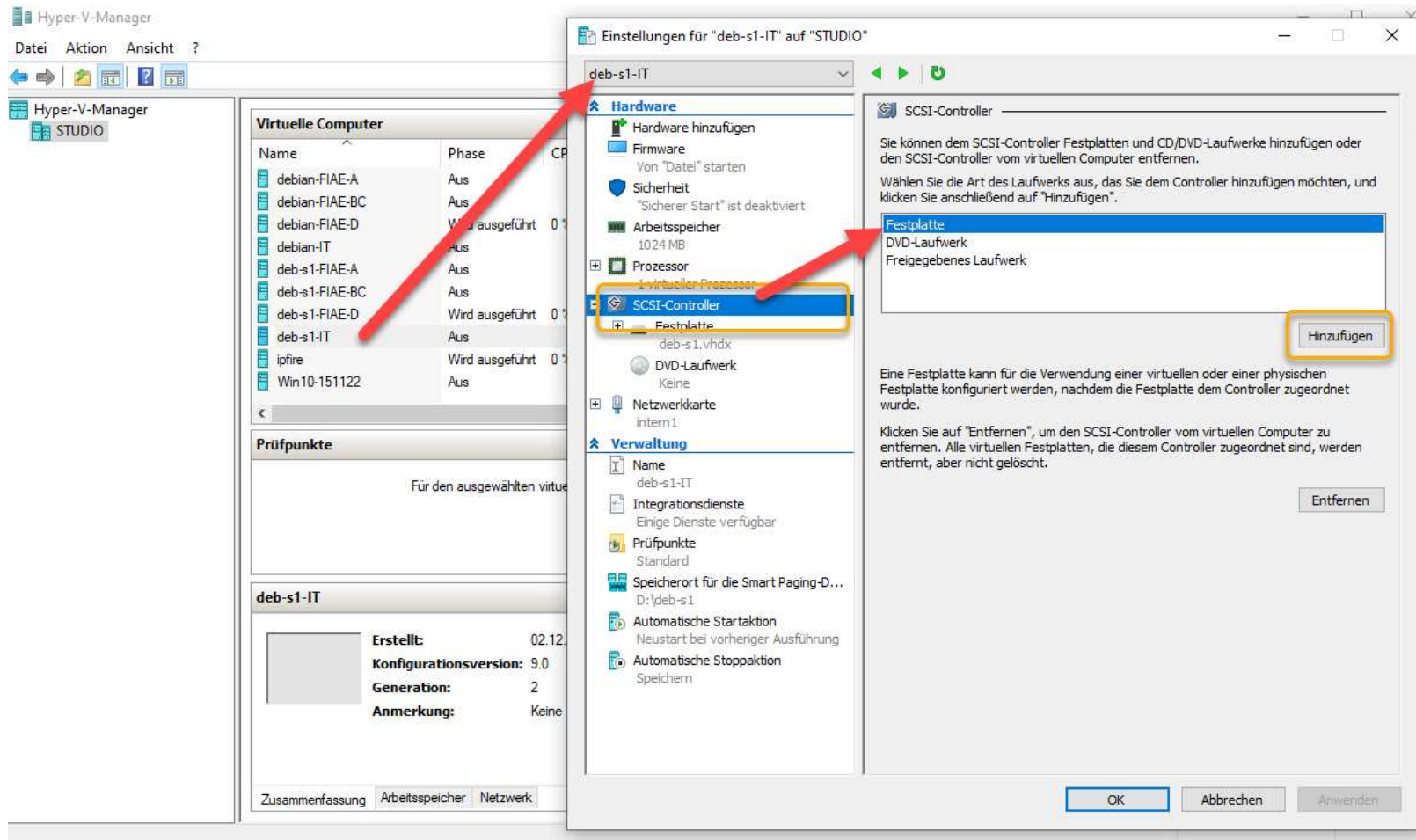
LVM Projektfestplatten erstellen



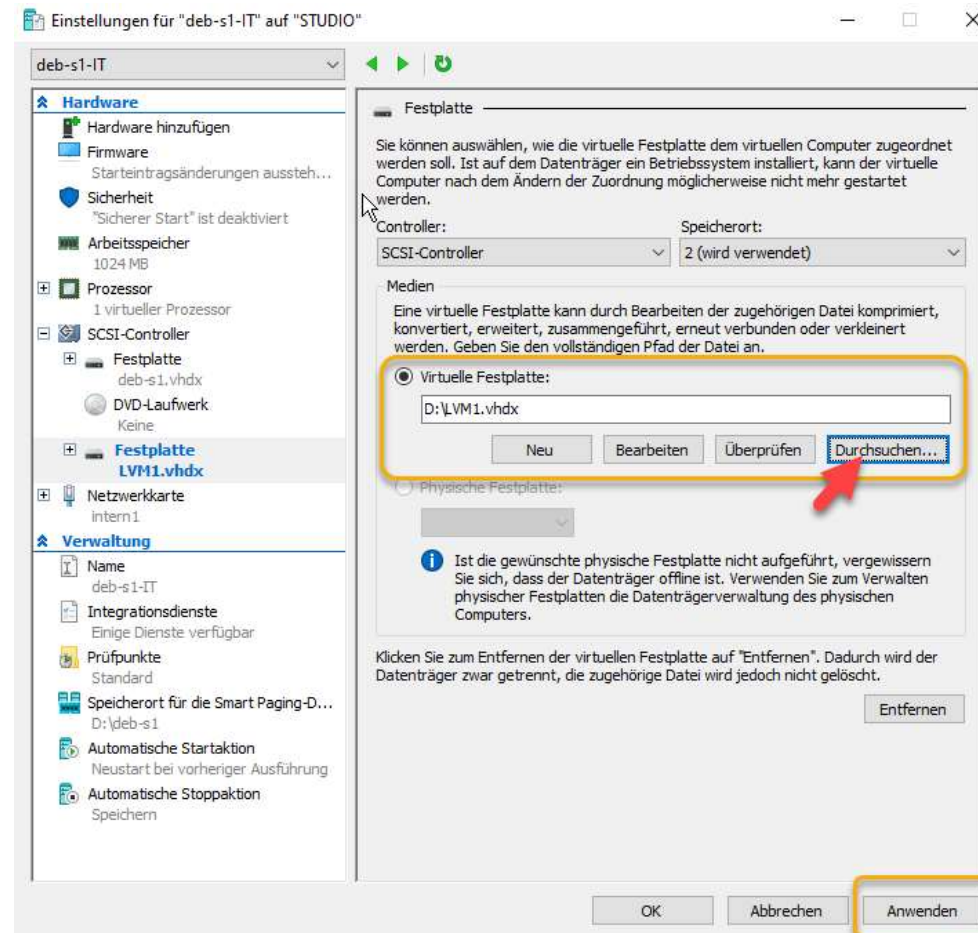
LVM Projektfestplatten erstellen

- Nachdem Sie diese virtuelle Festplatte erstellt haben, erstellen Sie eine zweite virtuelle Festplatte mit dem Namen **LVM2** nach dem gleichen Schema
- Die beiden Festplatten werden dann anschließend in Ihre VM deb-s1 eingebunden, was in den nächsten Folien beschrieben wird

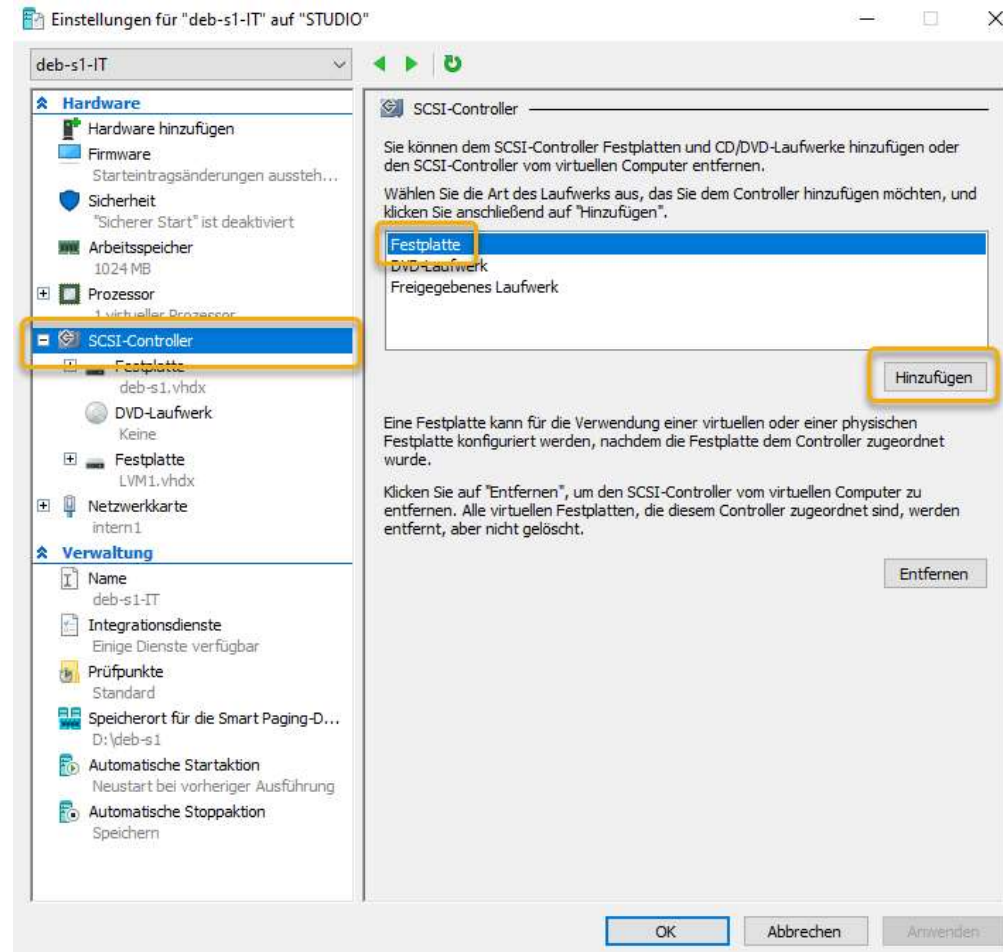
LVM Projektfestplatten in deb-s1 einbinden



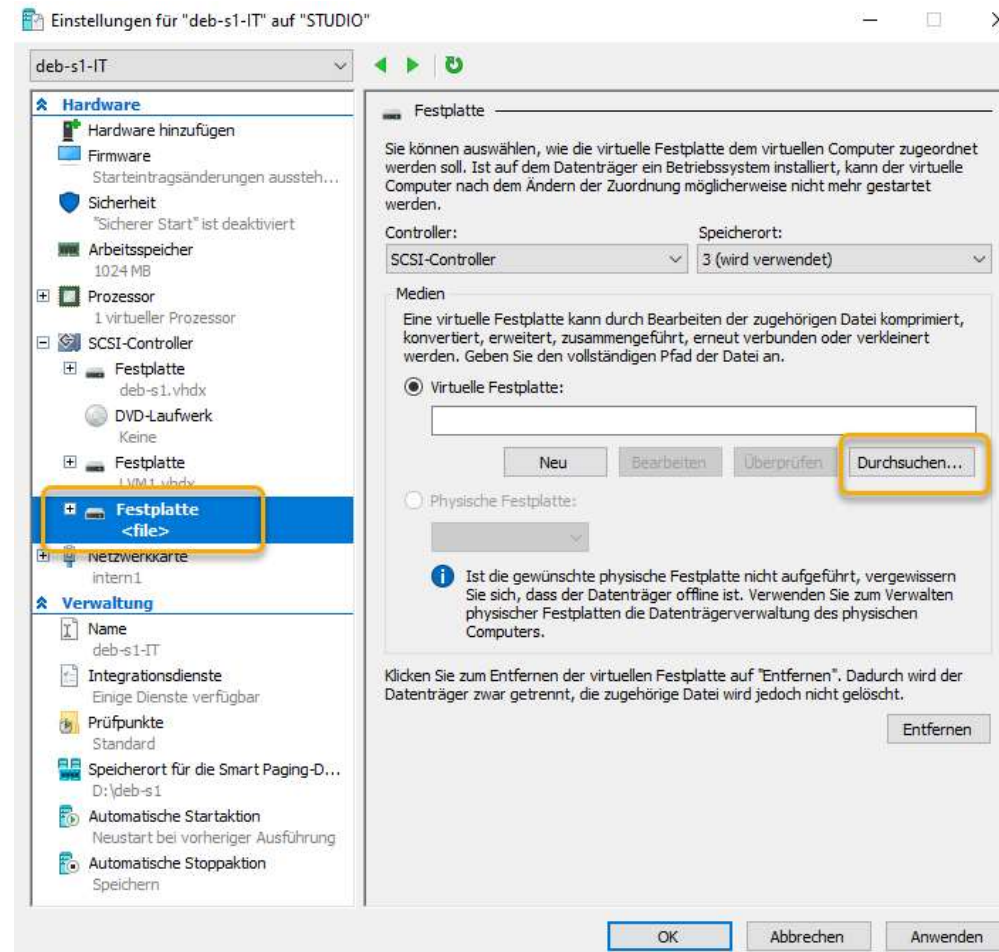
LVM Projektfestplatten in deb-s1 einbinden



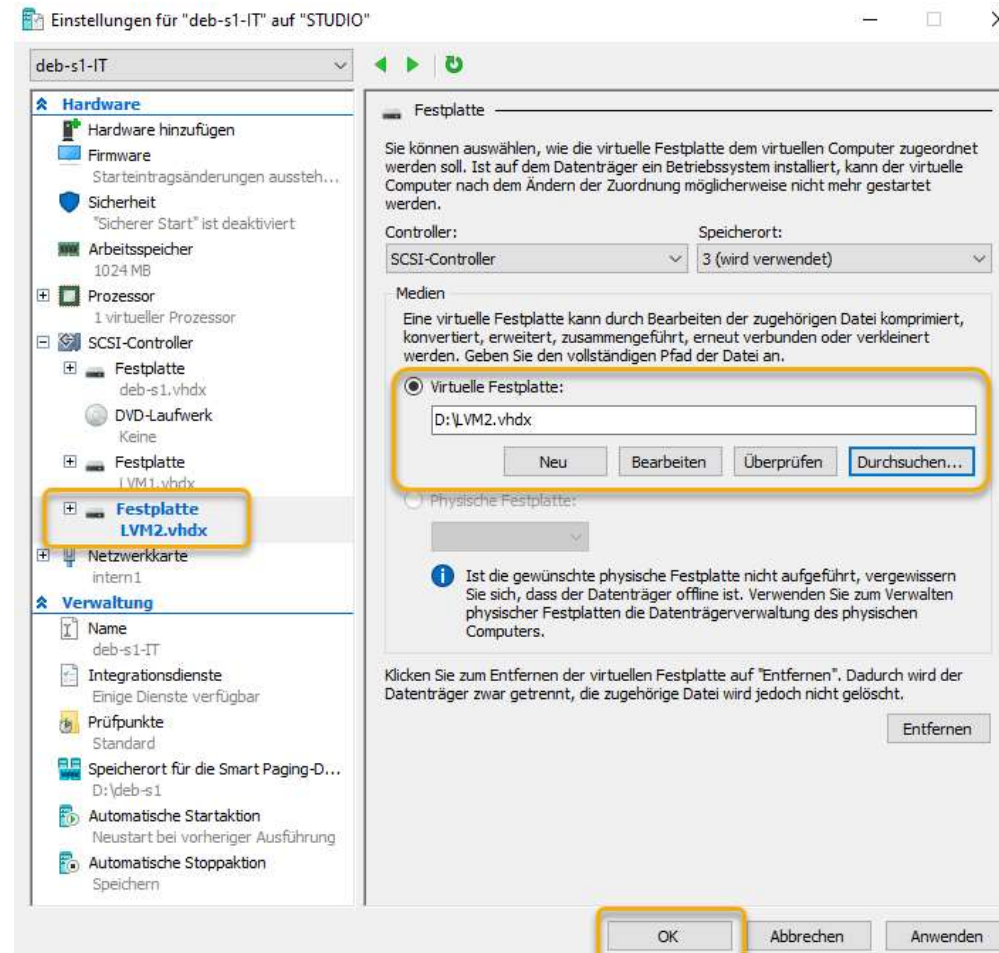
LVM Projektfestplatten in deb-s1 einbinden



LVM Projektfestplatten in deb-s1 einbinden



LVM Projektfestplatten in deb-s1 einbinden



Beide VM starten

- Nachdem beide Festplatten in der Hyper-V erstellt und die VM deb-s1 eingebunden sind, werden beide VM („debian“ und „deb-s1“, einschließlich „ipfire“) wie gewohnt gestartet.
- Danach verbinden Sie sich durch Eingabe von s1 in der VM „debian“ mit dem Server „deb-s1“ und melden sich als root an (su -)
- Die Festplatten benötigen eine Partitionstabelle (Partition Table), die Einrichtung der jeweils 3 Partitionen und auf jeder Partition ein Dateisystem (ext4)
- Um diese Aufgaben auszuführen stehen verschiedene Programme zur Verfügung
- Ich nutze wiederum „fdisk“, welches ohne Installation genutzt werden kann

Kontrolle der Festplatten

- Zunächst ist es wichtig, ob die beiden neuen Festplatten im System erkannt werden
- Um dies zu kontrollieren kann man das Kommando `lsblk` nutzen

```
root@deb-s1:~# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda   8:0    0   10G  0 disk
├─sda1 8:1    0   512M  0 part /boot/efi
├─sda2 8:2    0   8,5G  0 part /
└─sda3 8:3    0   976M  0 part [SWAP]
sdb   8:16   0   10G  0 disk
└─sdb1 8:17   0   10G  0 part /backup
sdc   8:32   0    6G  0 disk
sdd   8:48   0    6G  0 disk
sr0   11:0    1 1024M  0 rom
root@deb-s1:~#
```

Bitte beachten Sie, dass bei Ihnen die Festplatte sdb in mehrere Partitionen aufgeteilt ist. In diesem Testsystem habe ich darauf verzichtet, was aber keinen Einfluss auf die Festplatten sdc und sdd hat.

Partitionierung der Festplatte sdc

- Durch Eingabe von `fdisk /dev/sdc` wird das Programm `fdisk` geöffnet
- Mit der Taste `m` zeigt sich das Menü `fdisk`

```
root@deb-s1:~# fdisk /dev/sdc
```

```
Welcome to fdisk (util-linux 2.36.1).
```

```
Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.
```

```
Device does not contain a recognized partition table.
```

```
Created a new DOS disklabel with disk identifier 0x3dd4b652.
```

```
Command (m for help):
```

Partitionstabelle anlegen

- Zunächst benötigt die Festplatte eine Partitionstabelle, um darin Partitionen erstellen zu können
- Hier soll das heute moderne System GPT zum Einsatz kommen, womit es möglich ist, insgesamt 128 Partitionen zu erstellen:
- Dazu drückt man die Taste <g>

Create a new label

```
g   create a new empty GPT partition table
G   create a new empty SGI (IRIX) partition table
o   create a new empty DOS partition table
s   create a new empty Sun partition table
```


3 neue Partitionen erstellen

```
Command (m for help): n
Partition number (1-128, default 1): <Enter>
First sector (2048-12582878, default 2048): <Enter>
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-12582878, default 12582878): +2G
```

Created a new partition 1 of type 'Linux filesystem' and of size 2 GiB.

```
Command (m for help): n
Partition number (2-128, default 2): <Enter>
First sector (4196352-12582878, default 4196352): <Enter>
Last sector, +/-sectors or +/-size{K,M,G,T,P} (4196352-12582878, default 12582878): +2G
```

Created a new partition 2 of type 'Linux filesystem' and of size 2 GiB.

```
Command (m for help): n
Partition number (3-128, default 3): <Enter>
First sector (8390656-12582878, default 8390656): <Enter>
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8390656-12582878, default 12582878): <Enter>
```

Created a new partition 3 of type 'Linux filesystem' and of size 2 GiB.

```
Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

Partitionierung der Festplatte sdd

- Durch Eingabe von `fdisk /dev/sdd` verwenden Sie `fdisk` in gleicher Weise wie bei `sdc`
- Erstellen Sie zuerst die GPT-Partitionstabelle mit `<g>`
- Danach erstellen Sie ebenfalls die 3 Partitionen nach dem Schema der vorigen Folie
- `lsblk` zeigt die Partitionen

```
root@deb-s1:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda          8:0    0   10G  0 disk
├─sda1       8:1    0   512M  0 part /boot/efi
├─sda2       8:2    0    8,5G  0 part /
└─sda3       8:3    0   976M  0 part [SWAP]
sdb          8:16   0    10G  0 disk
└─sdb1       8:17   0    10G  0 part /backup
sdc          8:32   0     6G  0 disk
├─sdc1       8:33   0     2G  0 part
├─sdc2       8:34   0     2G  0 part
└─sdc3       8:35   0     2G  0 part
sdd          8:48   0     6G  0 disk
├─sdd1       8:49   0     2G  0 part
├─sdd2       8:50   0     2G  0 part
└─sdd3       8:51   0     2G  0 part
sr0         11:0    1 1024M  0 rom
```

root@deb-s1:~# █

Fleißarbeit – Filesystem der Partitionen

- Die insgesamt 6 Partitionen benötigen jetzt noch ein ext4-Filesystem
- Geben Sie der Reihe nach ein (am besten mit Autoausfüllen):
 - `mkfs.ext4 /dev/sdc1`
 - `mkfs.ext4 /dev/sdc2`
 - `mkfs.ext4 /dev/sdc3`
 - `mkfs.ext4 /dev/sdd1`
 - `mkfs.ext4 /dev/sdd2`
 - `mkfs.ext4 /dev/sdd3`

Fehlende Mount-Points erstellen

- Die zukünftigen Mount-Points musik, bilder und video existieren noch nicht im Wurzelverzeichnis
- Diese legen Sie mit mkdir gleichzeitig an und kontrollieren mit ls /

```
root@deb-s1:~# mkdir /musik /bilder /video
```

```
root@deb-s1:~# ls /
```

```
backup  bin    dev    home      initrd.img.old  lib32  libx32  media  musik  proc  run  srv  tmp  var  
bilder  boot  etc    initrd.img  lib          lib64  lost+found  mnt    opt    root  sbin  sys  usr  video
```

```
root@deb-s1:~# █
```

LVM installieren

- Nun ist es an der Zeit LVM zu installieren
- Das Paket trägt die Bezeichnung lvm2

```
root@deb-s1:~# apt install lvm2
```

```
Paketlisten werden gelesen... Fertig
```

```
Abhängigkeitsbaum wird aufgebaut... Fertig
```

```
Statusinformationen werden eingelesen... Fertig
```

```
Die folgenden zusätzlichen Pakete werden installiert:
```

```
  dmeventd libaio1 libdevmapper-event1.02.1 liblvm2cmd2.03 thin-provisioning-tools
```

```
Die folgenden NEUEN Pakete werden installiert:
```

```
  dmeventd libaio1 libdevmapper-event1.02.1 liblvm2cmd2.03 lvm2 thin-provisioning-tools
```

```
0 aktualisiert, 6 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
```

```
Es müssen 2.403 kB an Archiven heruntergeladen werden.
```

```
Nach dieser Operation werden 8.760 kB Plattenplatz zusätzlich benutzt.
```

```
Möchten Sie fortfahren? [Y/n] █
```

Physical Volumes

- Festplatten oder Partitionen, die unter LVM genutzt werden sollen müssen als Physical Volumes eingetragen werden
- Dies geschieht mit dem Kommando **pvcreate**
 - **pvcreate /dev/sdc1**
 - **pvcreate /dev/sdc2**
 - **pvcreate /dev/sdc3**
 - **pvcreate /dev/sdd1**
 - **pvcreate /dev/sdd2**
 - **pvcreate /dev/sdd3**

Warnungen wegen der ext4-Signatur bestätigen Sie bitte mit y (yes)
- Mit **pvs** erhalten Sie die Übersicht der Physical Volumes

```
root@deb-s1:~# pvs
PV          VG Fmt  Attr PSize  PFree
/dev/sdc1   lvm2 ---  2,00g  2,00g
/dev/sdc2   lvm2 ---  2,00g  2,00g
/dev/sdc3   lvm2 ---  <2,00g <2,00g
/dev/sdd1   lvm2 ---  2,00g  2,00g
/dev/sdd2   lvm2 ---  2,00g  2,00g
/dev/sdd3   lvm2 ---  <2,00g <2,00g
root@deb-s1:~#
```

Volume Groups

- Aus den Partitionen lassen Gruppen bilden – Volume Groups
- Sie können diese beliebig benennen und zusammenstellen
- Hier sollen folgende Volume Groups gebildet werden:
 - Volume Group 1 mit dem Namen „system“ aus sdc1 und sdd1
 - Volume Group 2 mit dem Namen „speicher“ aus sdc2, sdc3 und sdd2
- Die Partition sdd3 soll vorerst nicht einbezogen werden. Sie steht als Reserve zur Verfügung.
- Volume Groups werden mit dem Kommando vgcreate erstellt

Volume Groups

- Wie beschrieben werden die Volume Groups erstellt

```
root@deb-s1:~# vgcreate system /dev/sdc1 /dev/sdd1
Volume group "system" successfully created
root@deb-s1:~# vgcreate speicher /dev/sdc2 /dev/sdc3 /dev/sdd2
Volume group "speicher" successfully created
root@deb-s1:~# █
```

- Mit dem Kommando vgs bekommt man die Übersicht angezeigt

```
root@deb-s1:~# vgs
VG          #PV #LV #SN Attr   VSize  VFree
speicher    3   0   0 wz--n-   <5,99g <5,99g
system      2   0   0 wz--n-    3,99g  3,99g
root@deb-s1:~# █
```

Logical Volumes

- Aus den Volume Groups lassen sich nun Logical Volumes bilden
- Diese werden vom System wie Partitionen behandelt, wobei keine Geschwindigkeitsnachteile auftreten
- Logical Volumes erstellt man mit dem Kommando `lvcreate`
- Allgemeine Form:
 - `lvcreate -L <Größe> -n <gewünschter Name des Logical Volumes> <Volume Group>`
- Es ist keine schlechte Idee den Namen des Logical Volumes so zu wählen, dass er mit dem Mount-Point übereinstimmt, also z.B. „bilder“ für den Mount-Point `/bilder`
- Mit dem Kommando `lvs` erhält man die Übersicht über die Logical Volumes

Logical Volumes

- Erstellen Sie nach dem genannten Schema die Logical Volumes

```
root@deb-s1:~# lvcreate -L 3G -n home system
Logical volume "home" created.
```

```
root@deb-s1:~# lvcreate -L 2G -n musik speicher
Logical volume "musik" created.
```

```
root@deb-s1:~# lvcreate -L 2G -n video speicher
Logical volume "video" created.
```

```
root@deb-s1:~# lvcreate -L 1G -n bilder speicher
Logical volume "bilder" created.
```

```
root@deb-s1:~# █
```

```
root@deb-s1:~# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
bilder	speicher	-wi-a-----	1,00g									
musik	speicher	-wi-a-----	2,00g									
video	speicher	-wi-a-----	2,00g									
home	system	-wi-a-----	3,00g									

```
root@deb-s1:~# █
```

Logical Volumes mit Mount-Points verbinden

- Nun können die Mount-Points mit den Logical Volumes verbunden werden.
- Wie üblich sind die derzeitigen Inhalte der Mount-Points danach nicht mehr sichtbar
 - Da /musik, /bilder und /video bisher keine Inhalte besitzen spielt es keine Rolle
 - Nur das Verzeichnis /home besitzt schon Inhalte, die vor dem Mounten gesichert und danach wieder eingespielt werden müssen
 - (In diesem Umfeld werden die Inhalte in /home nur mit cp kopiert, da das Mounten hier nur temporär erfolgt. In produktiven Umgebungen werden die Mount-Vorgänge in /etc/fstab eingetragen, so dass die Sicherung mittels mv aus dem /home-Verzeichnis entfernt werden)

/musik, /bilder und /video mounten

- Wie schon gesagt verhalten sich die erstellten Logical Volumes wie Partitionen
- Deshalb benötigen diese auch ein Dateisystem, wie üblich ext4
 - # mkfs.ext4 /dev/system/home
 - # mkfs.ext4 /dev/speicher/musik
 - # mkfs.ext4 /dev/speicher/bilder
 - # mkfs.ext4 /dev/speicher/video
- Jetzt kann gemountet werden

```
root@deb-s1:~# mount /dev/speicher/musik /musik  
root@deb-s1:~# mount /dev/speicher/bilder /bilder  
root@deb-s1:~# mount /dev/speicher/video /video
```

Mount-Kontrolle

- Mit dem Kommando `mount` (ohne weitere Optionen) werden alle vorhandenen Mount-Points angezeigt
- Am Ende der Liste stehen die gerade erstellten Verbindungen:

```
/dev/mapper/speicher-musik on /musik type ext4 (rw,relatime)  
/dev/mapper/speicher-bilder on /bilder type ext4 (rw,relatime)  
/dev/mapper/speicher-video on /video type ext4 (rw,relatime)  
root@deb-s1:~# █
```
- Zu sehen ist, dass eine Abstraktionsschicht „mapper“ vorhanden ist.
- Das mounten kann auch darüber ausgeführt werden

Mounten von /home

- Zunächst der Inhalt des /home-Verzeichnisses in das Wurzelverzeichnis / gesichert z.B. mit tar

```
root@deb-s1:~# tar -cf /homeverzeichnis.tar /home
tar: Entferne führende „/“ von Elementnamen
root@deb-s1:~# █
```

- Anschließend kann gemountet werden

```
root@deb-s1:~# mount /dev/system/home /home
root@deb-s1:~# █
```

- Kontrollieren Sie den Erfolg des Mountens mit dem Kommando mount (siehe vorige Folie)

Inhalte in /home zurückspielen

- Die Sicherung /homeverzeichnis.tar kann jetzt in das gemountete Verzeichnis /home zurückgespielt werden

- Wechseln Sie dazu ins Wurzelverzeichnis / und extrahieren Sie die .tar-Datei

```
root@deb-s1:~# cd /  
root@deb-s1:/# tar -xf homeverzeichnis.tar  
root@deb-s1:/# ls /home  
helmut  huber  korn  lost+found  meier  schmidt  
root@deb-s1:/# █
```

- Bei Ihnen gibt es natürlich kein Verzeichnis helmut, aber Ihr Homeverzeichnis sollte vorhanden sein
- Damit sind jetzt alle Mountvorgänge mit LVM erfolgreich ausgeführt

Zu wenig Speicherplatz

Volume Group „speicher“ und Logical Volume „video“ vergrößern

Derzeitiger Zustand (1)

- Wir stellen in diesem Szenario fest, dass das Logical Volume „video“ zu klein ist für unsere vielen Filmdateien
- Mittels „lvs“ können wir die derzeitige Größe des Logical Volumes „video“ sehen und auch, dass sich in der Volume Group „speicher“ befindet

```
root@deb-s1:/# lvs
  LV      VG      Attr      LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  bilder  speicher -wi-ao---- 1,00g
  musik   speicher -wi-ao---- 2,00g
  video   speicher -wi-ao---- 2,00g
  home    system  -wi-ao---- 3,00g
root@deb-s1:/#
```

Derzeitiger Zustand (2)

- „vgs“ zeigt die derzeitige Größe (VSize) und den freien Platz (Vfree) der beiden Volume Groups an

```
root@deb-s1:/# vgs
VG          #PV #LV #SN Attr   VSize  VFree
speicher    3   3   0 wz--n- <5,99g 1012,00m
system      2   1   0 wz--n- 3,99g 1016,00m
root@deb-s1:/# █
```

- „pvs“ zeigt an, dass /dev/sdd3 noch nicht zugewiesen ist

```
root@deb-s1:/# pvs
PV          VG          Fmt  Attr PSize  PFree
/dev/sdc1   system    lvm2 a--  <2,00g    0
/dev/sdc2   speicher  lvm2 a--  <2,00g    0
/dev/sdc3   speicher  lvm2 a--  <2,00g 1012,00m
/dev/sdd1   system    lvm2 a--  <2,00g 1016,00m
/dev/sdd2   speicher  lvm2 a--  <2,00g    0
/dev/sdd3   speicher  lvm2 ---  <2,00g  <2,00g
root@deb-s1:/# █
```


Volume Group „speicher“ vergrößern

- Jetzt soll die noch nicht eingebundene Partition `/dev/sdd3` in die Volume Group „speicher“ eingebunden werden, um mehr Speicherplatz darin zu erhalten
 - Bedenken Sie, was wir hier mühsam im GiB-bereich ausführen, wird im „richtigen Leben“ im TiB-Bereich ausgeführt 😊
- Mit „`vgextend`“ erweitern wir die Volume Group „speicher“ und kontrollieren den Erfolg mit „`vgs`“ – 2 GiB größerer Speicherplatz

```
root@deb-s1:/# vgextend speicher /dev/sdd3
Volume group "speicher" successfully extended
root@deb-s1:/# vgs
VG          #PV #LV #SN Attr   VSize VFree
speicher    4   3   0 wz--n- 7,98g  2,98g
system      2   1   0 wz--n- 3,99g 1016,00m
root@deb-s1:/# █
```


Vorarbeit: Logical Volume „video“ vergrößern

- Bevor das Logical Volume „video“ vergrößert werden kann, muss es ausgehängt werden, wenn es gemountet ist

```
root@deb-s1:/# umount /video  
root@deb-s1:/# █
```

- Jetzt sollte das Dateisystem aus Konsistenzgründen gecheckt werden

```
root@deb-s1:/# fsck.ext4 /dev/speicher/video  
e2fsck 1.46.2 (28-Feb-2021)  
/dev/speicher/video: clean, 11/131072 files, 26156/524288 blocks  
root@deb-s1:/# █
```

- Mit „lvresize“ kann nun die gewünschte Größe für das Logical Volume „video“ erstellt werden
 - Die Gesamtgröße sollte in absoluter Form eingetragen werden, also z.B. mit 4g für 4 GiB

Logical Volume „video“ vergrößern

■ Allgemeine Form:

- # `lvresize -L <absolute Größe> /dev/Volume Group/Logical Volume`

```
root@deb-s1:/# lvresize -L 4g /dev/speicher/video
Size of logical volume speicher/video changed from 2,00 GiB (512 extents) to 4,00 GiB (1024 extents).
Logical volume speicher/video successfully resized.
root@deb-s1:/# █
```

■ Mit „lvs“ kann man die neue Größe des Logical Volumes „video“ kontrollieren

```
root@deb-s1:/# lvs
LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
bilder  speicher -wi-ao---- 1,00g
musik   speicher -wi-ao---- 2,00g
video   speicher -wi-a----- 4,00g
home    system  -wi-ao---- 3,00g
root@deb-s1:/# █
```

Dateisystem ext4 anpassen (1)

- Vergessen Sie nicht, dass das Dateisystem ext4 noch nichts von der Größenänderung mitbekommen hat. Das ist aber unbedingt notwendig
 - Hierfür verwendet man das Kommando „resize2fs“ in allgemeiner Form:
 - `# resize2fs /dev/Volume Group/Logical Volume <neue Größe>`
 - Bevor man das Kommando ausführen kann wird man meistens aufgefordert einen Systemcheck mit „e2fsck“ auszuführen
 - `# e2fsck -f /dev/Volume Group/Logical Volume`

```
root@deb-s1:/# e2fsck -f /dev/speicher/video
e2fsck 1.46.2 (28-Feb-2021)
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/speicher/video: 11/131072 files (0.0% non-contiguous), 26156/524288 blocks
root@deb-s1:/# █
```

Dateisystem ext4 anpassen (2)

- Nach bestandener Prüfung kann „resize2fs“ problemlos ausgeführt werden

```
root@deb-s1:/# resize2fs /dev/speicher/video 4g
resize2fs 1.46.2 (20-Feb-2021)
Resizing the filesystem on /dev/speicher/video to 1048576 (4k) blocks.
The filesystem on /dev/speicher/video is now 1048576 (4k) blocks long.
```

```
root@deb-s1:/# █
```

- Danach kann wieder gemountet werden. Der größere Speicherplatz steht jetzt zur Verfügung

```
root@deb-s1:/# mount /dev/speicher/video /video
root@deb-s1:/# █
```

- Kontrollieren Sie den Erfolg mit dem Kommando „mount“

snapshot

Die dynamische Momentaufnahme

Linux

snapshots

- **Snapshots bieten die Möglichkeit eine Momentaufnahme eines Logical Volumes zu erstellen, um es im Notfall wieder auf den Zustand des Snapshots zurück zusetzen.**
- **Diese Funktion steht auch in den meisten Virtualisierungslösungen zur Verfügung**
- **Im LVM stehen uns mit Snapshots weitaus mehr Möglichkeiten zur Verfügung**
 - **Aus einem Snapshot im LVM lassen sich einfach Backups der Logical Volumes erstellen und**
 - **die Snapshots selbst lassen sich dynamisch erweitern**
- **Als Beispiel möchte ich die Möglichkeiten anhand eines Snapshots des gemounteten Logical Volumes „musik“ erstellen**

Kontrollen und Vorarbeiten

- Zunächst ist es erst einmal notwendig zu kontrollieren, ob das Logical Volume „musik“ gemountet ist. Das einfache Kommando „mount“ gibt Auskunft darüber

```
/dev/mapper/speicher-musik on /musik type ext4 (rw,relatime)
/dev/mapper/speicher-bilder on /bilder type ext4 (rw,relatime)
/dev/mapper/system-home on /home type ext4 (rw,relatime)
/dev/mapper/speicher-video on /video type ext4 (rw,relatime)
root@deb-s1:/# █
```

- Nachdem das geklärt ist, benötigen wir einige Inhalte im gemounteten Logical Volume
 - Erstellen Sie mit touch einige leere Musik-Dateien mit dem Suffix .mp3 und zwei leere Verzeichnisse mit mkdir, so oder so ähnlich wie in der nächsten Folie angezeigt
 - Ihrer Kreativität ist keine Grenze gesetzt (außer der, der hier begrenzten Größe des Snapshots)

Inhalte erstellen

- So könnten die Musiktitel und Ordner aussehen, nachdem Sie mit `cd` in `/musik` gewechselt sind:

```
root@deb-s1:/# cd /musik
root@deb-s1:/musik# touch rock1.mp3
root@deb-s1:/musik# touch rock2.mp3
root@deb-s1:/musik# touch rock3.mp3
root@deb-s1:/musik# touch jazz1.mp3
root@deb-s1:/musik# touch jazz2.mp3
root@deb-s1:/musik# touch jazz3.mp3
root@deb-s1:/musik# mkdir tagmusik
root@deb-s1:/musik# mkdir nachtmusik
root@deb-s1:/musik# █
```

- Einige der Titel und ein Ordner werden, nach dem der Snapshot erstellt ist, wieder gelöscht, um das Wiederherstellen zu demonstrieren

Snapshot erstellen

- Jetzt kann der Snapshot mit dem bekannten „lvcreate“ erstellt werden
 - Hier in der Übungsumgebung steht dafür nur begrenzten Speicherplatz zur Verfügung, daher werde ich zuerst einen Fehlversuch mit zu großem Speicherbedarf demonstrieren, der aber Auskunft gibt, wieviel Speicher für den Snapshot verwendet werden kann.
 - Mit der Option -L wird die gewünschte absolute Größe 1g (1 GiB) angegeben
 - -s bestimmt die Erstellung eines snapshots
 - Hinter -n steht der gewünschte (sprechende) Snapshotname z.B. snapmusik
 - Das letzte Argument ist der Pfad zum Logical Volume hier: /dev/speicher/musik

```
root@deb-s1:/musik# lvcreate -L 1g -s -n snapmusik /dev/speicher/musik
Volume group "speicher" has insufficient free space (252 extents): 256 required.
root@deb-s1:/musik# █
```

- Die Größe von 1 GiB steht nicht zur Verfügung, aber 256 MiB (256m) können verwendet werden

Snapshot in der möglichen Größe

- Die angezeigte Meldung (vorige Folie) gibt an, dass 1 GiB nicht zur Verfügung steht, aber 256 MiB genutzt werden könnten

```
root@deb-s1:/musik# lvcreate -L 256m -s -n snapmusik /dev/speicher/musik
Logical volume "snapmusik" created.
root@deb-s1:/musik#
```

- Der Snapshot benötigt also nicht die gleiche Größe, wie das Logical Volume, da er nur eine Momentaufnahme ist
- Die jetzt 256 MiB stehen dem Snapshot für die Aufnahme aller Veränderungen zur Verfügung
- Allerdings: Ist der Speicher verbraucht, wird der Snapshot ungültig

Kontrolle

- Mit „lvs“ wird auch der erstellte Snapshot angezeigt
- Die Größe „Data%“ zeigt an, wieviel %-Änderung von den 256 GiB schon verbraucht sind

```
root@deb-s1:/musik# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
bilder	speicher	-wi-ao----	1,00g									
musik	speicher	owi-aos---	2,00g									
snapmusik	speicher	swi-a-s---	256,00m		musik	0,01						
video	speicher	-wi-ao----	4,00g									
home	system	-wi-ao----	3,00g									

```
root@deb-s1:/musik# █
```

Snapshot mounten

■ Auch ein Snapshot kann gemountet werden

- Dazu muss noch ein geeigneter Mountpoint geschaffen werden.
- Um kein Chaos zu produzieren, nutzen wir das vorhandene Verzeichnis /mnt, welches vom Namen her schon dafür vorgesehen ist.
- Um das gesamte Verzeichnis nicht zu blockieren, erstellen wir für unseren Zweck ein Unterverzeichnis mit dem Namen „snapmusik“, also mkdir /mnt/snapmusik und mounten darin unseren Snapshot

```
root@deb-s1:~# mkdir /mnt/snapmusik
root@deb-s1:~# ls /mnt
snapmusik
root@deb-s1:~# mount /dev/speicher/snapmusik /mnt/snapmusik/
root@deb-s1:~# █
```

Identische Inhalte

- Da der Snapshot auf Basis des Verzeichnisses /musik erstellt wurde, sind die Inhalte des Ordners /musik und das Snapshots /mnt/snapmusik identisch

```
root@deb-s1:~# ls /musik/
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  tagmusik
root@deb-s1:~# ls /mnt/snapmusik/
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  tagmusik
root@deb-s1:~# █
```

Neue Datei in /musik erstellen

- Um das Zusammenspiel zwischen Original und Snapshot zu verstehen, erstellen wir als nächstes mit echo eine neue Datei im Ordner /musik
- Die Datei soll abweichend von den Musiktiteln „nachsnapshot.txt“ heißen und den Inhalt erhalten: „Diese Datei wurde nach dem Snapshot erstellt.“

```
root@deb-s1:~# echo "Diese Datei wurde nach dem Snapshot erstellt" > /musik/nachsnapshot.txt
root@deb-s1:~# ls /musik
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachsnapshot.txt  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  tagmusik
root@deb-s1:~# ls /mnt/snapmusik/
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  tagmusik
root@deb-s1:~#
```

- Im Snapshot ist die Datei natürlich nicht vorhanden, wie ls beweist

Inhalte aus /musik entfernen

- Als nächstes werden wir ein paar Musiktitel und eines der Verzeichnisse löschen
 - Ich lösche die Musikdateien „jazz3.mp3“ und „rock3.mp3“, sowie das Verzeichnis „nachtmusik“.
 - (Suchen Sie sich beliebige Titel zum Löschen aus, nur **nicht** die Datei „nachsnapshot.txt“)

```
root@deb-s1:~# rm /musik/*3.mp3
root@deb-s1:~# rmdir /musik/nachtmusik/
root@deb-s1:~# ls /musik
jazz1.mp3  jazz2.mp3  lost+found  nachsnapshot.txt  rock1.mp3  rock2.mp3  tagmusik
root@deb-s1:~# █
```

- Wie zu erwarten, sind die genannten Inhalte gelöscht, im Snapshot natürlich nicht

```
root@deb-s1:~# ls /mnt/snapmusik/
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  tagmusik
root@deb-s1:~# █
```

Der Snapshot reagiert auf die Änderungen

- Bedingt durch die gelöschten Inhalte in /musik werden die Änderungen prozentual im Snapshot angezeigt.

- Mit „lvs“ kann man sich den jeweiligen Stand anzeigen lassen

```
root@deb-s1:~# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
bilder	speicher	-wi-ao----	1,00g									
musik	speicher	owi-aos---	2,00g									
snapmusik	speicher	swi-aos---	256,00m		musik	0,12						
video	speicher	-wi-ao----	4,00g									
home	system	-wi-ao----	3,00g									

```
root@deb-s1:~#
```

- Der Wert ist in keiner Weise kritisch, aber schon sichtbar (direkt nach dem Anfertigen des Snapshots lag er bei 0,01%)

Den Snapshot ändern

■ Auch der Snapshot selbst kann bearbeitet werden

- Gezeigt werden soll dies, indem eine neue Textdatei mit dem Namen „snapdatei.txt“ und dem Inhalt „Diese Datei entstand auf dem Snapshot.“ wiederum mit dem Kommando „echo“ erstellt werden soll.
- Mit ls lässt sich auch deren Existenz im Snapshot nachweisen. Im Verzeichnis /musik ist sie natürlich nicht vorhanden

```
root@deb-s1:~# echo "Diese Datei entstand auf dem Snapshot." > /mnt/snapmusik/snapdatei.txt
root@deb-s1:~# ls /mnt/snapmusik/
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  snapdatei.txt  tagmusik
root@deb-s1:~# ls /musik/
jazz1.mp3  jazz2.mp3  lost+found  nachsnapshot.txt  rock1.mp3  rock2.mp3  tagmusik
root@deb-s1:~# █
```

Den Snapshot zurückspielen

- Nachdem nun beide Systeme große Unterschiede aufweisen soll der Inhalt des Snapshots in das Verzeichnis /musik zurück gespielt werden
- Was ist zu erwarten?
 - Die Datei „neuedatei.txt“ entstand nach dem Snapshot und sollte daher verloren gehen
 - Die gelöschten Musiktitel und das gelöschte Verzeichnis waren Bestandteil des Snapshots und sollten daher wieder hergestellt werden
 - Die Datei „snapdatei.txt“ entstand zusätzlich im Snapshot und sollte auch in /musik übertragen werden
- Um den Snapshot zurückzuspielen müssen beide Systeme ausgehängt werden

```
root@deb-s1:~# umount /musik/  
root@deb-s1:~# umount /mnt/snapmusik/  
root@deb-s1:~# █
```

Snapshot zurückspielen und mounten

- Das Zurückspielen geschieht mit dem Kommando „lvconvert“
 - Der Vorgang selbst dauert einige Sekunden
 - Gibt man anschließend „lvs“ ein, stellt man fest, dass der Snapshot verschwunden ist
 - Danach kann man das Logical Volume „musik“ wieder in /musik mounten

```
root@deb-s1:~# lvconvert --merge /dev/speicher/snapmusik
```

```
Merging of volume speicher/snapmusik started.
```

```
speicher/musik: Merged: 99,96%
```

```
speicher/musik: Merged: 100,00%
```

```
root@deb-s1:~# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync	Convert
bilder	speicher	-wi-ao----	1,00g									
musik	speicher	-wi-a-----	2,00g									
video	speicher	-wi-ao----	4,00g									
home	system	-wi-ao----	3,00g									

```
root@deb-s1:~# mount /dev/speicher/musik /musik/
```

Die Inhalte von /musik und Auswertung

- Wie üblich, können wir mit „ls“ den jetzigen Inhalt von /musik ansehen
- Die Inhalte entsprechen den Erwartungen:

```
root@deb-s1:~# ls /musik/  
jazz1.mp3  jazz2.mp3  jazz3.mp3  lost+found  nachtmusik  rock1.mp3  rock2.mp3  rock3.mp3  snapdatei.txt  tagmusik
```

- Die Mountpoints sind momentan natürlich nur temporär verbunden
- Zur ständigen Nutzung trägt man sie einfach in die Datei /etc/fstab ein
- Nach den Einträgen in die /etc/fstab müssen Sie keinen Neustart ausführen, da sie mit dem Kommando `mount -a` automatisch gemountet werden und der Erfolg lässt sich dann mit `mount` (ohne Parameter) einfach kontrollieren
- UUID's sind nicht notwendig, stehen aber zur Verfügung