

Zusatz-Lektion: LVM auf einen RAID1 installieren



Voraussetzungen

- Ausgangspunkt dieser Zusatzlektion ist eine separat auf deb-s1 eingebundene Festplatte mit einer Größe von 10GB
- Falls die Festplatte schon für andere Übungen eingesetzt wurde, müssen automatische Mountpoints in der Datei `/etc/fstab` entfernt oder kommentiert werden
- Falls sie schon Partitionen und ein Dateisystem besitzt und sich wichtige Daten in den Partitionen befinden, müssen diese gesichert werden, da die Festplatte komplett neu eingerichtet wird

Die unpartitionierte Festplatte /dev/sdb

- Die Festplatte trägt die Bezeichnung sdb und befindet sich wie üblich im Verzeichnis /dev also /dev/sdb
- Mit lsblk wird die Festplatte im System angezeigt
- Da ein RAID 1 erstellt werden soll benötigen wir 2 Partitionen
- In der Praxis werden natürlich 2 Festplatten gleicher Größe benutzt, die jeweils eine Partition erhalten

```
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
sr0        11:0    1 1024M  0 rom
root@deb-s1:~# █
```

Tastenmenü in fdisk

- Die Partitionierung erfolgt wiederum mit dem Kommando `fdisk /dev/sdb`
- In `fdisk` drückt man die Taste `m`, um die Übersicht der Tastenkürzel für die nötigen Aktionen zu bekommen

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

```
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 0xbe8aac6e.
```

```
Command (m for help): █
```

Tastenmenü in fdisk

Generic

- d delete a partition
- F list free unpartitioned space
- l list known partition types
- n add a new partition
- p print the partition table
- t change a partition type
- v verify the partition table
- i print information about a partition

Misc

- m print this menu
- u change display/entry units
- x extra functionality (experts only)

Script

- I load disk layout from sfdisk script file
- O dump disk layout to sfdisk script file

Save & Exit

- w write table to disk and exit
- q quit without saving changes

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

Partitionstabelle GPT

- Um Partitionen erstellen zu können benötigt jede Festplatte (auch SSD) eine Partitionstabelle (partition table)
- Heute nutzt man dafür die GPT-Partitionstabelle, die mit der Taste g erstellt wird

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

```
Created a new GPT disklabel (GUID: D47B9909-4910-0249-A082-AAB463657983).
```

```
Command (m for help): █
```

- Ihre GUID wird einen anderen Inhalt haben

Partitionen erstellen

- Eine neue Partition erstellt man mit der Taste n
- Die Größe der Partition wird mit +5000M angegeben

```
Command (m for help): n
Partition number (1-128, default 1): Enter
First sector (2048-20971486, default 2048): Enter
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971486, default 20971486): +5000M
```

Created a new partition 1 of type 'Linux filesystem' and of size 4,9 GiB.

```
Command (m for help): n
Partition number (2-128, default 2): Enter
First sector (10242048-20971486, default 10242048): Enter
Last sector, +/-sectors or +/-size{K,M,G,T,P} (10242048-20971486, default 20971486): +5000M
```

Created a new partition 2 of type 'Linux filesystem' and of size 4,9 GiB.

```
Command (m for help): █
```

Linux Filesystem

- Die Partitionen werden vom Typ „Linux Filesystem“ erstellt
- Da ein RAID-System eingerichtet werden soll, sollte der Typ auf „Linux RAID“ umgestellt werden
- Mit der Taste l kann man sich die Liste der möglichen Typen anzeigen lassen. Für „Linux RAID“ existiert die Typ-Nummer 29

27 Linux reserved	8DA63339-0007-60C0-C436-085AC8230908
28 Linux home	933AC7E1-2EB4-4F13-B844-0E14E2AEF915
29 Linux RAID	A19D880F-05FC-4D3B-A006-743F0F84911E
30 Linux LVM	E6D6D379-F507-44C2-A23C-238F2A3DF928
31 Linux variable data	4D21B016-B534-45C2-A9FB-5C16E091FD2D
32 Linux temporary data	75665557-3865-4A6A-B302-16555B5630B1

- Mit der Taste q kann man die Auflistung verlassen

Partitionstyp

- Um den Typ 29 für jede Partition einzugeben, benutzt man die Taste t
- Die Partitionen gibt man jeweils mit ihrer Partitionsnummer an, um danach jeweils den Typ 29 einzutragen

```
Command (m for help): t
Partition number (1,2, default 2): 1
Partition type or alias (type L to list all): 29
```

Changed type of partition 'Linux filesystem' to 'Linux RAID'.

```
Command (m for help): t
Partition number (1,2, default 2): 2
Partition type or alias (type L to list all): 29
```

Changed type of partition 'Linux filesystem' to 'Linux RAID'.

```
Command (m for help): █
```

Übersicht

- Mit der Taste p lassen sich die gewählten Einstellungen übersichtlich anzeigen

Command (m for help): **p**

Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors

Disk model: Virtual Disk

Units: sectors of 1 * 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 4096 bytes

I/O size (minimum/optimal): 4096 bytes / 4096 bytes

Disklabel type: gpt

Disk identifier: D47B9909-4910-0249-A082-AAB463657983

Device	Start	End	Sectors	Size	Type
/dev/sdb1	2048	10242047	10240000	4,9G	Linux RAID
/dev/sdb2	10242048	20482047	10240000	4,9G	Linux RAID

Command (m for help): █

Festplatte einrichten

- Bisher sind die Einstellungen für die Festplatte nur angezeigt worden. Auf der Festplatte selbst wurde noch nichts geändert
- Wenn die Einstellungen der vorigen Folie in Ordnung sind, kann die Festplatte mit den Einstellungen eingerichtet werden
- Dazu drückt man die Taste w, wodurch man fdisk auch gleichzeitig verlässt

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

```
root@deb-s1:~# █
```

Abschließende Kontrolle

- Wiederum mit dem Kommando `lsblk` kann man kontrollieren, ob die beiden Partitionen `sdb1` und `sdb2` eingerichtet wurden

```
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   4,9G  0 part
└─sdb2      8:18    0   4,9G  0 part
sr0        11:0    1 1024M  0 rom
root@deb-s1:~#
```

Software-RAID mdadm

- Falls das Paket mdadm für vorherige Übungen noch nicht installiert wurde, kann dies mit `# apt install mdadm` aus den Paketquellen ausgeführt werden
- Auf dem Testsystem wurde mdadm schon vorher installiert

```
root@deb-s1:~# apt install mdadm
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Statusinformationen werden eingelesen... Fertig
mdadm ist schon die neueste Version (4.1-11).
0 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
root@deb-s1:~# █
```


RAID 1

- Ein RAID 1 benötigt 2 Festplatten (hier 2 Partitionen) deren Daten gespiegelt werden
- Die Netto-Kapazität der beiden verbundenen Festplatten beträgt also 50% der Gesamtkapazität
- Eine der beiden Festplatte kann ausfallen, ohne dass Daten verloren gehen
- Anders als beim Hardware-RAID gibt es keine optischen Signale, die den Ausfall anzeigen (in SAN- bzw. NAS-Systemen)
- Ein Software-RAID lässt sich aber mittels Befehlseingabe kontrollieren

RAID 1 einrichten

- Mittels mdadm und einer Reihe von Parametern lässt sich das RAID 1 einfach einrichten
- Das RAID-System erhält die Bezeichnung md0 und befindet sich wie alle Speichergeräte in Verzeichnis /dev

RAID erstellen

RAID 1

Anzahl der Geräte

die Geräte

```
root@deb-s1:~# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdb2
```

```
mdadm: Note: this array has metadata at the start and
may not be suitable as a boot device. If you plan to
store '/boot' on this device please ensure that
your boot-loader understands md/v1.x metadata, or use
--metadata=0.90
```

```
Continue creating array? yes
```

```
mdadm: Defaulting to version 1.2 metadata
```

```
mdadm: array /dev/md0 started.
```

```
root@deb-s1:~# █
```

Dateisystem einspielen

- In Debian und vielen anderen Distributionen wird als Standarddateisystem ext4 eingesetzt. Daher macht es auch Sinn unser RAID-1-System damit zu formatieren
- Hierbei kommt das Kommando mkfs (Make a Filesystem) zur Anwendung

```
root@deb-s1:~# mkfs.ext4 /dev/md0
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 1278720 4k blocks and 320000 inodes
Filesystem UUID: a848efcf-1382-4de5-9d30-fc591ee57ce2
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@deb-s1:~# █
```

Statusabfrage

- Zur Kontrolle, ob das RAID-System funktioniert fragt man dessen Status ab
- Dies geschieht, indem man sich im Verzeichnis /proc den Inhalt der Datei mdstat ansieht
- Dies ist auch die Überwachungsfunktion für unser Software-RAID

```
root@deb-s1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1] sdb1[0]
      5114880 blocks super 1.2 [2/2] [UU]

unused devices: <none>
root@deb-s1:~#
```

2 aktive Partitionen

LVM

- Das LVM (Logical Volume Management) ist ein sehr flexibles System zur Verwaltung und Verwendung von Datenträgern. Dabei können z.B. mehrere Datenträger zusammengefasst werden, die dann nach außen wie ein großer Datenträger wirken
- Nachteil eines LVM ist, wenn einer der Datenträger defekt ist, sind alle Daten nicht mehr erreichbar.
- Daher werden LVM-Systeme gerne auf einem RAID-System eingerichtet

LVM Physical Volumes

- Zuerst müssen die physischen Datenträger (Physical Volumes) gekennzeichnet werden, die in einem LVM-System eingesetzt werden sollen.
- Dies geschieht mit dem Kommando `pvcreeate` gefolgt von der Datenträgerliste, hier nur `/dev/md0`

```
root@deb-s1:~# pvcreeate /dev/md0
WARNING: ext4 signature detected on /dev/md0 at offset 1080. Wipe it? [y/n]: y
Wiping ext4 signature on /dev/md0.
Physical volume "/dev/md0" successfully created.
root@deb-s1:~# pvs
  PV          VG Fmt  Attr PSize  PFree
  /dev/md0    lvm2 ---  <4,88g <4,88g
root@deb-s1:~# █
```

- Mit dem Kommando `pvs` fragt man den Status der bzw. des Physical Volumes ab

LVM Volume Groups

- Nun bildet man aus den verschiedensten Physical Volumes (PV) eine oder mehrere Volume Groups (VG) mit dem Kommando `vgcreate` gefolgt von einem passenden Namen für die VG
- In unserem Fall existiert nur ein PV, so dass auch nur eine VG eingerichtet wird

```
root@deb-s1:~# vgcreate speicher /dev/md0
Volume group "speicher" successfully created
root@deb-s1:~# vgs
VG          #PV #LV #SN Attr   VSize  VFree
speicher    1   0   0 wz--n- <4,88g <4,88g
root@deb-s1:~# █
```

- Mit dem Kommando `vgs` fragt man den Status der Volume Groups ab

LVM Logical Volumes

- Aus einer Volume Group (VG) können eine oder mehrere Logical Volumes (LV) gebildet werden. In unserem Fall soll nur ein LV erstellt werden, welches dann auf das Verzeichnis /backup gemountet wird
- Es ist sinnvoll den Namen eines LV entsprechend des Mount-Points zu benennen
- Unser LV erhält also den Namen „backup“, was durch -n angegeben wird
- Beim Erstellen eines LV wird auch die LV-Größe mit -L definiert. Die Größe darf maximal der VG-Größe entsprechen. Sie kann später verkleinert und durch Hinzufügen weiterer Datenträger auch vergrößert werden

LVM Logical Volumes

- Das LV wird erstellt:

```
root@deb-s1:~# lvcreate -L 4,8g -n backup speicher
Rounding up size to full physical extent 4,80 GiB
Logical volume "backup" created.
root@deb-s1:~# lvs
  LV      VG      Attr      LSize Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  backup  speicher -wi-a----- 4,80g
root@deb-s1:~# █
```

- Mit dem Kommando lvs kann man sich die Eigenschaften des LV anzeigen lassen
- Das LV muss natürlich wieder ein Dateisystem erhalten, da das ext4-System vom RAID entfernt wurde (wipe)

LVM Dateisystem für das Logical Volume

- Das LV benötigt zur Nutzung wieder ein ext4-Dateisystem
- Die Zuweisung geschieht über den Namen der VG und dem Namen des LV

```
root@deb-s1:~# mkfs.ext4 /dev/speicher/backup
mke2fs 1.46.2 (28-Feb-2021)
Discarding device blocks: done
Creating filesystem with 1258496 4k blocks and 315120 inodes
Filesystem UUID: 339beae8-d4d3-4efc-ad8a-a3f10e3135ad
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

root@deb-s1:~# █
```


Logical Volume mounten

- Das erstellte Logical Volume kann wie jede Partition gemountet werden. Es treten dabei keine Geschwindigkeitsverluste auf
- Natürlich ist auch ein Eintrag in die Datei `/etc/fstab` möglich, wenn ein permanentes Mounten gewünscht ist
- Zunächst jedoch das manuelle mounten mit anschließender Kontrolle des Inhalts:

```
root@deb-s1:~# mount /dev/speicher/backup /backup
root@deb-s1:~# ls /backup
lost+found
root@deb-s1:~# █
```

Der simulierte Fehler

- **Das funktionierende RAID-System:**

```
root@deb-s1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1] sdb1[0]
      5114880 blocks super 1.2 [2/2] [UU]
```

```
unused devices: <none>
```

```
root@deb-s1:~# █
```

- **Die Partition /dev/sdb1 als defekte Partition kennzeichnen:**

```
root@deb-s1:~# mdadm /dev/md0 --fail /dev/sdb1
mdadm: set /dev/sdb1 faulty in /dev/md0
root@deb-s1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1] sdb1[0](F)
      5114880 blocks super 1.2 [2/1] [_U]
```

```
unused devices: <none>
```

```
root@deb-s1:~# █
```

Sind die Daten noch vorhanden?

- Auf der gespiegelten Partition /dev/sdb2 müssten die Daten noch existieren:

```
root@deb-s1:~# ls /backup
liste.txt  lost+found
root@deb-s1:~# █
```

- Die defekte Partition entfernen

```
root@deb-s1:~# mdadm --remove /dev/md0 /dev/sdb1
mdadm: hot removed /dev/sdb1 from /dev/md0
root@deb-s1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb2[1]
      5114880 blocks super 1.2 [2/1] [_U]

unused devices: <none>
root@deb-s1:~# █
```

Eine neue Partition /dev/sdb1 hinzufügen

- Nachdem die fehlerhafte Partition entfernt wurde, kann jetzt eine neue Partition dafür eingesetzt werden (entspricht dem Tausch einer Festplatte)

```
root@deb-s1:~# mdadm --add /dev/md0 /dev/sdb1
mdadm: added /dev/sdb1
root@deb-s1:~# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4] [raid10]
md0 : active raid1 sdb1[2] sdb2[1]
      5114880 blocks super 1.2 [2/1] [_U]
      [=====>.....]   recovery = 31.3% (1601792/5114880) finish=0.2min speed=200224K/sec

unused devices: <none>
root@deb-s1:~# █
```

Die Daten von /dev/sdb2 werden auf die neue Partition /dev/sdb1 gespiegelt

- Nach der Beendigung des Recovery-Vorgangs sind die Daten auf beiden Partitionen wieder synchron verteilt. Das RAID-1-System ist wieder hergestellt

Dauerhaftes mounten

- Normalerweise wird es gewünscht, dass ein LVM-System beim Systemstart automatisch gemountet wird
- Dazu muss man die UUID vom `/dev/mapper/speicher-backup` herausfinden.
- Leider ist auf meinem Testsystem ein Hyper-V-Effekt aufgetreten, der die Bezeichnung `/dev/md0` in `/dev/md127` umgewandelt hat, was aber für die weiteren Tätigkeiten bedeutungslos ist.

UUID herausbinden

- Mit dem Kommando `blkid` findet man die UUID's aller im System befindlichen blockorientierten Speichermedien heraus:

```
/dev/md127: UUID="6307IX-UbIL-vCHB-qIIL-ZinL-5FL4-0VcPNC" TYPE="LVM2 member"
```

```
/dev/mapper/speicher-backup: UUID="339beae8-d4d3-4efc-ad8a-a3f10e3135ad" BLOCK_SIZE="4096" TYPE="ext4"
```

- Unter anderem wird auch die UUID von `/dev/mapper/speicher-backup` angezeigt
- Diese lässt sich durch Markieren mit der Maus und anschließendem Rechtsklick, durch Klick auf „Kopieren“ im Kontextmenü in die Zwischenablage speichern
- Die UUID vom `/dev/md127` nicht benutzen!

Datei /etc/fstab bearbeiten

- Die Datei fstab im /etc-Verzeichnis nimmt alle Speichergeräte auf, die beim Systemstart automatisch gemountet werden sollen
 - Die Bearbeitung erfolgt mit nano /etc/fstab
 - Für den neuen Eintrag der UUID von /dev/mapper/speicher-backup bewegt man sich mit dem Cursor unterhalb der letzten Zeile (Dort kann man, wenn gewünscht einen Kommentareintrag unterbringen, der den Mount-Eintrag beschreibt)
 - Darunter wird zunächst, wiederum mit Rechtsklick und „Einfügen“ die UUID von /dev/mapper/speicher-backup eingefügt und folgendermaßen ergänzt:

```
# LVM in Raid1 (dev/mapper/speicher-backup) dauerhaft mounten
UUID=339beae8-d4d3-4efc-ad8a-a3f10e3135ad      /backup ext4      defaults      0      0
```

- Die Anführungszeichen vor und nach der UUID werden entfernt, hinter der UUID werden jeweils mit Tabstopp getrennt der Mountpoint (/backup), das Dateisystem (ext4), die Optionen (defaults), sowie der dump- und pasw-Wert (jeweils 0) eingetragen

Datei /etc/fstab testen

- Zum Testen der Einstellungen in der Datei /etc/fstab ist kein Neustart notwendig – er wird auch nicht empfohlen einen Neustart auszuführen

- Mit dem Kommando `mount -a` werden alle Einträge in der Datei /etc/fstab gemountet. Sollten dabei Fehler auftreten, können diese direkt bearbeitet werden

```
root@deb-s1:~# mount -a
root@deb-s1:~# █
```

- Bei einem fehlerhaften Neustart ist der Korrekturaufwand viel höher und zeitraubender
- Mit dem anschließenden Kommando `mount` (ohne Option) kann man sich vom Erfolg (in der letzten Zeile überzeugen:

```
root@deb-s1:~# mount
/dev/mapper/speicher-backup on /backup type ext4 (rw,relatime)
root@deb-s1:~# █
```

- Ab sofort wird unser LVM im RAID1 automatisch beim Systemstart gemountet