

# Linux (allgemein): Software-RAID nachträglich einrichten

Problem:

Bei einigen Anbietern (hier speziell Hetzner genannt) werden zwar Systeme mit RAID angeboten aber mit bestimmten vorinstallierbaren Images (bei Hetzner z.B. das "Debian-4-Minimal") nicht genutzt.

Wer dies zu spät erkennt, findet hier nun eine Anleitung, wie er ein RAID-Array nachträglich einrichtet ohne die Daten der bisherigen Installation zu verlieren.

Vorbereitung:

Als erstes switchen wir dazu in den **Rescue-Modus**. Die ersten paar Schritte können z.B. bei Hetzner übersprungen werden da die Partitionen bereits auf beiden Platten identisch eingerichtet sind. Ich beschreibe es hier dennoch von Anfang an und einheitlich.

Desweiteren sind hier jeweils die Devices `/dev/sda` und `/dev/sdb` genannt. Wer keine SATA-/SCSI-Platten hat, wird wohl eher `/dev/hda` und `/dev/hdb` ansprechen müssen.

Aber Achtung mit der Nummerierung: die MD-Devices werden von 0 ab durchnummeriert. Wir erhalten hier also aus `/dev/sda1` die `/dev/md0` und aus `/dev/sda2` die `/dev/md1`

Wie gesagt gehe ich von Hetzner's Konfiguration aus. D.h. `/dev/sda1` ist die Swap-Partition und `/dev/sda2` die einzige Root-Partition. Wer es noch umpartitionieren will, sollte dies jetzt noch im Rescue-Modus per `parted` tun.

Installiert wird ein RAID-1. Denn für ein RAID-0 müsste man vollständig neu anfangen bzw. mit den Partitionen geschickt jonglieren.

Zweite Platte vorbereiten

```
#kopiere die Partitionstabelle von sda auf sdb
sfdisk -d /dev/sda | sfdisk /dev/sdb
fdisk /dev/sdb
#setze alle Partitionstypen auf "Linux raid autodetect" (=fd):
# [t] [1] fd [return]
# [t] [2] fd [return]
# [w]
```

# Linux (allgemein): Software-RAID nachträglich einrichten

Nun erstellen wir darauf bereits das Software-RAID wobei wir hier die erste Platte noch nicht mit einbinden, da sie sonst gelöscht werden würde:

```
mdadm -C /dev/md0 -l 1 -n 2 missing /dev/sdb1
mdadm -C /dev/md1 -l 1 -n 2 missing /dev/sdb2
```

```
#File-Systeme erstellen
mkswap /dev/md0
mkfs.ext3 /dev/md1
```

Daten kopieren

Nun mounten wir die erste Platte und das neu erstellte RAID:

```
mkdir /mnt/root
mount /dev/sda2 /mnt/root
mkdir /mnt/raid
mount /dev/md1 /mnt/raid
```

Wir ändern schon mal die nötigen Dateien um dem zu bootendem System das RAID an die Hand zu legen. Dabei werden in der `fstab` und `mtab` die Einträge zu `/dev/sda2` in `/dev/md1` verändert:

```
#Die MD-Devices bekannt machen:
mdadm --examine --scan >> /mnt/root/etc/mdadm/mdadm.conf
#fstab backup und bearbeiten
cp -p /mnt/root/etc/fstab /mnt/root/etc/fstab.orig
vim /mnt/root/etc/fstab
#mtab backup und bearbeiten
cp -p /mnt/root/etc/mtab /mnt/root/etc/mtab.orig
vim /mnt/root/etc/mtab
```

Nun folgt das Kopieren des vorhandenen Systems in das RAID mit anschließendem `umount`.

```
cp -dpRx /mnt/root/. /mnt/raid

#umount
umount /mnt/raid
umount /mnt/root
```

# Linux (allgemein): Software-RAID nachträglich einrichten

RAID vervollständigen

Nun binden wir `/dev/hda` in das RAID mit ein. Hierbei werden die Daten darauf gelöscht und von `/dev/md*` neu beschrieben.

```
mdadm --add /dev/md0 /dev/sda1
mdadm --add /dev/md1 /dev/sda2
```

```
#beobachten bis fertig (kann 2-3 Stunden dauern):
watch -n 5 cat /proc/mdstat
```

Boot-Loader aktualisieren

Um den Bootloader zu aktualisieren richten wir uns eine Chroot-Umgebung mit dem RAID-Array als Root ein:

```
mount /dev/md1 /mnt/raid
mount --bind /dev /mnt/raid/dev
mount --bind /proc /mnt/raid/proc
cd /mnt/raid
chroot /mnt/raid/
```

## Lilo:

Eine Debian std. Installation beinhaltet derzeit LILO.

Dazu brauchen wir lediglich einen Eintrag in der `/etc/lilo.conf` zu ändern:

```
root=/dev/md1
```

Ein anschließender Aufruf von `/sbin/lilo -v` schreibt den MBR.

## Grub:

Wer bereits GRUB installiert hat, braucht etwas mehr:

In `/boot/grub/menu.lst` muß die `kopt` angepasst werden:

# Linux (allgemein): Software-RAID nachträglich einrichten

```
# kopt=root=/dev/md1 ro
```

Anschließend kommen die üblichen Programme zum Zuge:

```
#kopt übernehmen:
```

```
update-grub
```

```
#den RAID-Treiber inkl. Device-Einstellungen in die initiale RamDisk mit aufnehmen
```

```
update-initramfs -u
```

```
#MBR sicherheitshalber auf beide Platten schreiben:
```

```
grub
```

Eingabe in der Grub-Shell:

```
root (hd0,1)
```

```
setup (hd0)
```

```
root (hd1,1)
```

```
setup (hd1)
```

```
quit
```

Fertig

Zu guter Letzt verlassen wir den Chroot und rebooten die Maschine:

```
exit
```

```
umount /mnt/raid
```

```
reboot
```

Probleme

Wenn er nicht bootet, heißt dies, das wir irgendwas am Boot-Loader vermässelt haben. Nicht weiter schlimm, da man jederzeit im Rescue-Modus dies wieder beheben kann: Einfach wieder in die Chroot und evtl. nachbessern.

GRUB hat mir dabei einige Probleme bereitet. Ich denke es war der o.g. manuelle Aufruf zum schreiben des MBR.

# Linux (allgemein): Software-RAID nachträglich einrichten

Falls nicht, habe ich zusätzlich in der `/boot/grub/menu.lst` folgende Änderungen hinzugefügt:

(`(hd0,1)` ist in dem Fall die `/dev/md1`)

```
title          Xen 3.0.3-1-amd64 / Debian GNU/Linux, kernel 2.6.18-6-xen-amd64
root           (hd0,1)
kernel         (hd0,1)/boot/xen-3.0.3-1-amd64.gz
module         (hd0,1)/boot/vmlinuz-2.6.18-6-xen-amd64 root=/dev/md1 ro console=tty0
module         (hd0,1)/boot/initrd.img-2.6.18-6-xen-amd64
```

Anmerkungen:

Diese Anleitung ist bereits zweimal von mir erfolgreich auf unterschiedlichen Hetzner-Servern ausgeführt worden. Einer mit `lilo` und einer mit `grub` (weil XEN).

Dennoch übernehme ich keine Garantie.

*Eindeutige ID: #1328*

*huschi*

*2013-04-26 12:51*