

vm208:~#

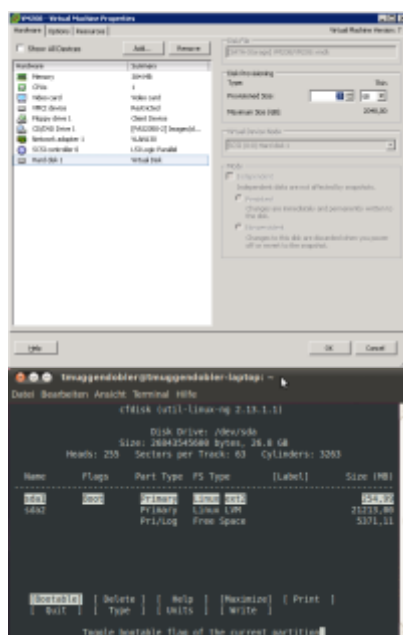
Es soll nun das LV (Logical Volume) „root“ der VG (Volume Group) „vm208“ (/dev/mapper/vm208-root) vergrößert werden.

Schritt-für-Schritt Anleitung

Der nachfolgende Abschnitt zeigt anhand einer Schritt-für-Schritt Anleitung, wie Sie das LVM nun vergrößern.

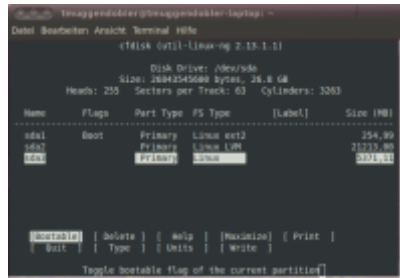
Vergrößern des Festplattenspeichers auf VMware Ebene

Durch Rechtsklick auf die VM im vSphere Client und durch Klick auf „Edit Settings“ erhält man den Dialog zum Konfigurieren der Einstellungen der VM.



Erstellen einer weiteren Partition

Um nun diesen zusätzlichen Festplattenspeicher nutzen zu können, muss zunächst eine weitere Partition (z. B. mittels dem Tool cfdisk) erstellt werden. Dazu hebt man den freien Speicher mittels Cursortasten hervor und wählt New > Primary > komplett verfügbaren Speicher (falls gewünscht). Abschließend sieht die Partitionierung wie in folgendem Screenshot aus.



Damit man die Vergrößerung fortsetzen kann, muss die Partitionstabelle neu eingelesen werden. Um einen weiteren Reboot zu vermeiden, kann man die Partitionstabelle mit dem Kommando „partprobe“ (Bestandteil des Paketes „parted“) neu einlesen.

Initialisieren der neuen Partition als PV (Physical Volume)

Damit diese zusätzliche Partition nun für LVM genutzt werden kann und einer VG zugewiesen werden kann, muss sie zunächst als PV initialisiert werden:

```
vm208:~# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sda2   vm208  lvm2 a-   19,75G  0
vm208:~# pvcreate /dev/sda3
Physical volume "/dev/sda3" successfully created
vm208:~# pvs
PV          VG      Fmt  Attr PSize  PFree
/dev/sda2   vm208  lvm2 a-   19,75G  0
/dev/sda3    vm208  lvm2 --    5,00G  5,00G
vm208:~#
```

Vergrößern der VG

Damit man wie gewünscht das LV „root“ vergrößern kann, muss zuvor noch die entsprechende VG vergrößert werden:

```
vm208:~# vgs
VG      #PV #LV #SN Attr   VSize  VFree
vm208   1  6  0 wz--n- 19,75G  0
vm208:~# vgextend vm208 /dev/sda3
Volume group "vm208" successfully extended
vm208:~# vgs
VG      #PV #LV #SN Attr   VSize  VFree
vm208   2  6  0 wz--n- 24,75G  5,00G
vm208:~#
```

Vergrößern des LV

Als nächstes kann nun das LV „root“ vergrößert werden:

```
vm208:~# lvs
LV      VG      Attr      LSize   Origin Snap%   Move Log Copy%   Convert
home    vm208  -wi-ao    10,54G
root    vm208  -wi-ao    332,00M
swap_1  vm208  -wi-ao     1,07G
tmp     vm208  -wi-ao    380,00M
usr     vm208  -wi-ao     4,66G
var     vm208  -wi-ao     2,79G
vm208:~# lvextend -L 1G /dev/mapper/vm208-root
Extending logical volume root to 1,00 GB
Logical volume root successfully resized
vm208:~# lvs
LV      VG      Attr      LSize   Origin Snap%   Move Log Copy%   Convert
home    vm208  -wi-ao    10,54G
root    vm208  -wi-ao     1,00G
swap_1  vm208  -wi-ao     1,07G
tmp     vm208  -wi-ao    380,00M
usr     vm208  -wi-ao     4,66G
var     vm208  -wi-ao     2,79G
vm208:~#
```

In diesem Beispiel vergrößern wir das LV root nur auf ca. 1GB. Den restlichen freie Speicher für die VG vm208 heben wir uns für eine spätere Verwendung für z. B. andere LV auf.

Hinweis: Bei dem `lvextend` Kommando wird nicht angegeben um wieviel das LV vergrößert werden soll, sondern auf welche Endgröße es vergrößert werden soll. Alternativ kann man auch ein `+` vor die Größe setzen, dann wird das LV um die angegebene Größe vergrößert.

Vergrößern des LV auf die maximale Größe

Um den sämtlichen nicht zugeordneten Speicherplatz für das logical Volume zu verwenden, ist folgendes Kommando erforderlich:

```
lvextend -l +100%FREE /dev/mapper/vm208-root
```

Vergrößern des Dateisystems

Um den zusätzlichen Speicherplatz auch nutzen zu können, muss abschließend noch das Dateisystem vergrößert werden. In unserem Beispiel wird `ext3` als Dateisystem verwendet, welches diese Vergrößerung (auch im eingehängten Zustand) problemlos unterstützt:

```
vm208:~# df -h
Dateisystem      Größe Benut  Verf Ben% Eingehängt auf
/dev/mapper/vm208-root
                  322M  175M   131M   58% /
tmpfs             187M     0   187M    0% /lib/init/rw
```

```

udev                10M   616K   9,4M   7% /dev
tmpfs                187M     0   187M   0% /dev/shm
/dev/sda1            228M   28M   189M  13% /boot
/dev/mapper/vm208-home 11G   155M   9,7G   2% /home
/dev/mapper/vm208-tmp 368M   11M   339M   3% /tmp
/dev/mapper/vm208-usr 4,6G   332M   4,1G   8% /usr
/dev/mapper/vm208-var 2,8G   281M   2,4G  11% /var

```

```
vm208:~# resize2fs -p /dev/mapper/vm208-root
```

```
resize2fs 1.41.3 (12-Oct-2008)
```

Das Dateisystem auf /dev/mapper/vm208-root ist auf / eingehängt; Online-Größenveränderung nötig

```
old desc_blocks = 2, new_desc_blocks = 4
```

Führe eine Online-Größenänderung von /dev/mapper/vm208-root auf 1048576 (1k) Blöcke durch.

Das Dateisystem auf /dev/mapper/vm208-root ist nun 1048576 Blöcke groß.

```
vm208:~# df -h
```

Dateisystem	Größe	Benut	Verf	Ben%	Eingehängt auf
/dev/mapper/vm208-root	993M	176M	766M	19%	/
tmpfs	187M	0	187M	0%	/lib/init/rw
udev	10M	616K	9,4M	7%	/dev
tmpfs	187M	0	187M	0%	/dev/shm
/dev/sda1	228M	28M	189M	13%	/boot
/dev/mapper/vm208-home	11G	155M	9,7G	2%	/home
/dev/mapper/vm208-tmp	368M	11M	339M	3%	/tmp
/dev/mapper/vm208-usr	4,6G	332M	4,1G	8%	/usr
/dev/mapper/vm208-var	2,8G	281M	2,4G	11%	/var

```
vm208:~#
```

Falls die Vergrößerung eines LVs mit `resize2fs -p /dev/mapper/vm208-root` nicht funktionieren sollte, kann alternativ folgender Befehl verwendet werden (Danke an Daniel Friedlmaier für den Hinweis):

```
lvextend --resizefs -l +100%FREE /dev/mapper/vm208-root
```

Bei einem XFS-Dateisystem können Sie mit dem Befehl `xfs_growfs` das Dateisystem vergrößern:

```
xfs_growfs /dev/mapper/<lv-name>
```

Last
update:
2023/02/11 07:00 it-wiki:linux:lvm_index:lvm-vergroessern https://blog.cooltux.net/doku.php?id=it-wiki:linux:lvm_index:lvm-vergroessern&rev=1676098844

From:
<https://blog.cooltux.net/> - **TuxNet DokuWiki**

Permanent link:
https://blog.cooltux.net/doku.php?id=it-wiki:linux:lvm_index:lvm-vergroessern&rev=1676098844

Last update: **2023/02/11 07:00**

