

VM disk Expansion LVM with XFS/EXT4

CREATE SNAPSHOT

Expand the original disk or add a second vdisk from **Infrastructure Client**.

~~If you want to perform a grow w/o reboot you have to add a second disk.~~

If you are on a kernel newer or equal to 2.6.32 then you can issue the command

```
echo "1" > /sys/block/sdX/device/rescan
```

to get the new size w/o reboot.

In case you cant see the "disk" do this:

```
echo "- - -" > /sys/class/scsi_host/host0/scan
```

Do this for all host0,1,2 etc.

or

```
echo "1" > /sys/class/scsi_device/2\:0\:0\:0/device/rescan
```

whereby after scsi_device you need to find the correct scsi device

EXPAND ORIGINAL DISK

Create an additional partition on the free space, say you increased the existing disc from 8GB to 15GB. The partition type is 8e.

```
cfdisk /dev/sda
```

for example if the new partition is /dev/sda3

create logical disk /dev/sda3 of type 8e (Linux LVM)

make disk visible to linux

```
partprobe  
pvcreate /dev/sda3
```

now you have to extend the volgroup. use `lvdisplay` to see which is the name of the group, in this example it's **base**

```
lvdisplay  
--- Logical volume ---  
LV Name                /dev/base/root  
VG Name                 base  
LV UUID                 8cL9Qd-ksIn-1Ve2-94ym-gTrW-8jet-91tnah  
LV Write Access         read/write
```

LV Status	available
# open	1
LV Size	5.00 GB
Current LE	1280
Segments	1
Allocation	inherit
Read ahead sectors	0
Block device	254:0

--- Logical volume ---

LV Name	/dev/base/tmp
VG Name	base
LV UUID	mEnEXY-Ut0f-P439-MDpg-BLT3-n8hI-Q6KIfm
LV Write Access	read/write
LV Status	available
# open	1
LV Size	1.00 GB
Current LE	256
Segments	1
Allocation	inherit
Read ahead sectors	0
Block device	254:1

--- Logical volume ---

LV Name	/dev/base/swap
VG Name	base
LV UUID	fm2A23-FPb3-itQa-Fvf2-QQmj-giwI-j8FZAf
LV Write Access	read/write
LV Status	available
# open	2
LV Size	2.00 GB
Current LE	512
Segments	1
Allocation	inherit
Read ahead sectors	0
Block device	254:2

--- Logical volume ---

LV Name	/dev/base/data
VG Name	base
LV UUID	GZKUbb-hZn2-igXN-3dxj-TNz9-1C15-I8u8MR
LV Write Access	read/write
LV Status	available
# open	1
LV Size	1.52 GB
Current LE	389
Segments	1
Allocation	inherit
Read ahead sectors	0

```
Block device      254:3
```

and we assume the new partition is /dev/sda3

```
vgextend base /dev/sda3
```

check with pvscan if the extend was successful

```
pvscan
```

Now we extend the “data” partition to 11.5 GB. See man lvextend for other options
11.5 G is the NEW total size of the disk we want to extend!

Extend the LV to use all free space

```
lvextend -l +100%FREE /dev/base/data
```

Extend with 20G

```
lvextend -L +20G /dev/base/data
```

now we have to grow the filesystem /data

```
xfs_growfs /data
```

FOR EXT3/4

Extend filesystem to use all free space

```
vgextend base /dev/sda3
```

```
lvextend -l +100%FREE /dev/base/root
```

Check that the filesystem is ok, **but only if the FS is unmounted**

```
fsck.ext4 -f /dev/base
```

Now resize the filesystem

```
resize2fs -p /dev/base/root # for ext2 on rpm systems  
resize2fs -p /dev/base/root # for ext4 on rpm systems
```

For swap

```
swapoff /dev/base/swap  
lvextend -L 3.9G /dev/base/swap
```

```
mkswap /dev/base/swap
swapon /dev/base/swap
free
```

Check if filesystems are ok, and only then release the snapshot

Create LVM

Find all LVM VG

```
vgscan
```

Create Physical Volumes

Der Befehl `pvcreate` legt den VGDA Block auf dem PV an. Er muss für jedes PV ausgeführt werden, bevor es von LVM verwendet werden kann:

```
pvcreate /dev/hda3
```

Volume Groups anlegen

Jetzt legen wir mit '`vgcreate <VolumeGroup> <Partition(en)>`' unsere Volume Group an und nennen sie "vg01":

```
vgcreate vg01 /dev/hda3 /dev/hdb2
```

Logische Volumes anlegen.

Hier legen wir jetzt alle unsere logischen Volumes an. Als Namen verwenden wir, etwas fantasielos, `lv01`, `lv02` und `lv03`. Dabei bedienen wir uns mit dem Plattenplatz auf Volume Group `vg01`, der einzigen Volume Group in diesem Beispiel. Es ist offensichtlich, dass wir hier nicht mehr Plattenplatz "verteilen" können, als wir bei `vgcreate` in die Volume Group "hineingesteckt" haben, und zwar in Form der beiden Partitionen `hda3` und `hdb2`. (`lvcreate -L<Grösse> -n <LogicalVolumeName> <Volume Group>`).

```
lvcreate -l +100%FREE -n lv01 vg01
```

oder wir könne die Größen angeben! (1500M)

```
lvcreate -L 1500M -n lv01 vg01
```

Filesysteme anlegen

Ab jetzt können die logischen Partitionen, genau so wie gewöhnliche Partitionen, über Ihre Device Files angesprochen werden. Gewöhnliche Partitionen werden mit `/dev/sd[a-z]*` oder `/dev/hd[a-z]*` bezeichnet; Logische Volumes werden mit `/dev/VolumeGroupName/LogicalVolumeName` angesprochen. Mit `mke2fs <LogicalVolumeName>` legen wir die ext2 Filesysteme an:

```
mkfs.ext4 /dev/vg01/lvol1
```

fstab anpassen

Damit die neuen Filesysteme nun bei jedem Systemstart automatisch gemountet werden, müssen wir sie in die Datei /etc/fstab eintragen (siehe 'man fstab'). In unserem Szenario sehen die zusätzlichen Einträge wie folgt aus:

```
/dev/vg01/lvol1 /usr ext4 defaults 1 2
```

<http://www.linuxhaven.de/dlhp/HOWTO-test/DE-LVM-HOWTO-2.html>

<http://www.howtogeek.com/howto/40702/how-to-manage-and-use-lvm-logical-volume-management-in-ubuntu/>

From:

<https://wiki.inf.unibz.it/> - **Engineering-Tech Wiki**

Permanent link:

https://wiki.inf.unibz.it/doku.php?id=tech:documentation:procedures:vm:vm_disk_expansion_lvm_xfs&rev=1394803556

Last update: **2019/01/16 10:03**

