

VMWare 환경에서 CentOS 7 Root Partition Resizing

- Created by [Fredrik Mikker](#), last modified by [Martin Sharples](#) on Jul 25, 2017

[Go to start of metadata](#)

Version

This article was written for version 6 of OP5 Appliance System, it could work on both lower and higher version if nothing else is stated.

Articles in community space is not covered by OP5 Support

When running virtual machines there might be a need to add space your virtual machine when it's running low, in this example we will add 10GB to the root filesystem of a virtual instance of OP5 Appliance System 6.5.

Prerequisites

To follow this article the following will be needed:

- OP5 Appliance System 6 or CentOS 6
- As always when working with storage, **make sure you have a recent backup of the virtual machine to be safe if anything should go wrong.**
- No snapshots can be present on the VM when resizing the disk
- Basic knowledge about Linux and LVM2

1. Shut down your virtual machine
2. Change the setting "Provisioned Size" in VMware to the total size that you want to have available in the operating system of your virtual server:

[blocked URL](#)

3. Start the virtual machine and log in as root to check that size on your disk has changed:

```
# fdisk -l /dev/sda
Disk /dev/sda: 107.4 GB, 107374182400 bytes
255 heads, 63 sectors/track, 13054 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x000076f1
   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *          1           17        131072   83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2           17         8355     66976768   8e  Linux LVM
```

In this example, the disk /dev/sda has been extended to 107.4 GB and there are only two primary partitions currently in use, this is good because there are room for two more primary partitions.

4. What you need to do now is to create the new partition sda3 that disk size can be taken from when extending the logical volume later on. This will be done with fdisk
The default values here should be sane, just press enter after the command input.

```
# fdisk /dev/sda
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to
switch off the mode (command 'c') and change display units to
sectors (command 'u').
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 3
First cylinder (8355-13054, default 8355):
Using default value 8355
Last cylinder, +cylinders or +size{K,M,G} (8355-13054, default 13054):
Using default value 13054
```

5. In the previous step, a new partition was created from the space that was added from VMware, and now we need to set the correct partition type before writing the partition table and saving the changes:

```

Command (m for help): t
Partition number (1-4): 3
Hex code (type L to list codes): 8e
Changed system type of partition 3 to 8e (Linux LVM)
Command (m for help): w
The partition table has been altered!
Calling ioctl() to re-read partition table.
WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.

```

The warning message above can be addressed in two ways: 1) Reboot the system, 2) Re-scan the device and add the new partition.

We will do the latter:

```

# partx -v -a /dev/sda
device /dev/sda: start 0 size 209715200
gpt: 0 slices
dos: 4 slices
# 1:      2048-   264191 (   262144 sectors,    134 MB)
# 2:      264192-134217727 (133953536 sectors, 68584 MB)
# 3: 134217728-209712509 ( 75494782 sectors, 38653 MB)
# 4:           0-     -1 (     0 sectors,     0 MB)
BLKPG: Device or resource busy
error adding partition 1
BLKPG: Device or resource busy
error adding partition 2
added partition 3

```

The partx application will give some errors for the existing partitions, but that's nothing to worry about. The important part is that partition 3 was successfully added.

If partition 3 isn't added you need to reboot the machine to make the changes take effect.

6. Now we need to create a new physical volume that can be used for adding space to the volume group

```

# pvcreate /dev/sda3
Physical volume "/dev/sda3" successfully created

```

7. Check what name your volume group your root filesystem uses:

```

# vgdisplay
--- Volume group ---
VG Name                1
System ID
Format                 lvm2
[..]

```

In this case the Volume group is called **1**

8. Now, lets extend the volume group with the space we added with vgextend:

```

# vgextend 1 /dev/sda3
Volume group "1" successfully extended

```

9. With the utility pvscan you can see what physical volumes are available, and how much free space there is within them. The free space can be used to extend your root partition, or other volumes that are short on space.

```

# pvscan
PV /dev/sda2 VG 1                lvm2 [63,87 GiB / 29,87 GiB free]
PV /dev/sda3 VG 1                lvm2 [36,00 GiB / 36,00 GiB free]
Total: 2 [99,87 GiB] / in use: 2 [99,87 GiB] / in no VG: 0 [0 ]

```

10. Now in this step we will at 10G to the root partition. First find out what logical volume that the root partition belongs to:

```

# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/1-sys 7,8G  1,4G  6,0G  20% /
tmpfs           246M    0  246M   0% /dev/shm
/dev/sda1       120M   27M   88M  24% /boot
/dev/mapper/1-opt 7,8G  181M  7,2G   3% /opt
/dev/mapper/1-tmp 7,8G   19M  7,4G   1% /tmp
/dev/mapper/1-var 7,8G  475M  6,9G   7% /var

```

In this case, the root partition originates from the logical volume "sys", and the total size is 7.8 GB.

11. The following command will add 10 GB to the Logical Volume "sys" in the Volume Group "1" and the disk area will be taken from the Physical Volume **sda3**.

```
# lvextend -L+10G /dev/mapper/1-sys /dev/sda3
Size of logical volume 1/sys changed from 8,00 GiB (2048 extents) to 18,00 GiB (4608 extents).
Logical volume sys successfully resized.
```

12. Run the following command to expand the ext4 filesystem inside of the Logical Volume:

```
# xfs_growfs -d /dev/mapper/cl-root
meta-data=/dev/mapper/cl-root isize=512 agcount=4, agsize=2293504 blks
        =           sectsz=512 attr=2, projid32bit=1
        =           crc=1 finobt=0 spinodes=0
data      =           bsize=4096 blocks=9174016, imapct=25
        =           sunit=0 swidth=0 blks
naming    =version 2          bsize=4096 ascii-ci=0 ftype=1
log       =internal        bsize=4096 blocks=4479, version=2
        =           sectsz=512 sunit=0 blks, lazy-count=1
realtime  =none            extsz=4096 blocks=0, rtextents=0
data blocks changed from 9174016 to 28865536
```

And voli! Now your root partition is extended with 10 GB:

```
# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/1-sys 18G  1,5G   16G   9% /
tmpfs           246M    0  246M   0% /dev/shm
/dev/sda1       120M   27M   88M  24% /boot
/dev/mapper/1-opt 7,8G  181M   7,2G   3% /opt
/dev/mapper/1-tmp 7,8G   19M   7,4G   1% /tmp
/dev/mapper/1-var 7,8G  475M   6,9G   7% /var
```