**[How to extend/resize Logical Volume and Volume Group in Linux](http://www.golinuxhub.com/2014/03/how-to-extendresize-lvm-and-volume.html)**

POSTED BY DEEPAK PRASAD FRIDAY, MARCH 21, 2014 [10 COMMENTS](http://www.golinuxhub.com/2014/03/how-to-extendresize-lvm-and-volume.html#comment-form)

In the below article I will show you the steps to add a new disk which can be added into existing volume group to extend a logical volume.  
  
But if you want to reduce (shrink) an existing logical volume and use the extra space to extend another logical volume, please follow below link  
[How to reduce LVM size in Linux step by step (online without reboot)](http://www.golinuxhub.com/2017/07/how-to-reduce-lvm-size-in-linux-step-by.html)  
  
As of now you can see my root partition is on LVM with 5.8Gb size. I want to extend my root filesystem with +1GB. Now here there can be two scenarios where  
  
# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup-root  
                      5.8G  4.1G  1.4G  76% /  
tmpfs                 504M  260K  504M   1% /dev/shm  
/dev/sda1             194M   26M  158M  15% /boot  
1. You want to extend size using extra partition  
2. You want to extend size with space in existing Volume Group  
  
I will show you steps required to perform the same using both the scenarios

**Scenario 1**

**Extend LVM using additional partition**

# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes

255 heads, 63 sectors/track, 1305 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: 0x000cf049

   Device Boot      Start         End      Blocks   Id  System

/dev/sda1   \*           1          26      204800   83  Linux

Partition 1 does not end on cylinder boundary.

/dev/sda2              26        1306    10279936   8e  Linux LVM

Disk **/dev/sdb**: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x00000000

Disk /dev/mapper/VolGroup-root: 6278 MB, 6278873088 bytes

255 heads, 63 sectors/track, 763 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: 0x00000000

Disk /dev/mapper/VolGroup-swap: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x00000000

So as you see I have added extra partition hard disk **/dev/sdb** with **2GB** space. Let us use it to extend our root file system.  
  
Steps required to perform the same  
1. Create partition for /dev/sdb with 1 GB storage  
2. Create Physical Volume  
3. Extend existing Volume Group  
4. Extend LVM  
5. Verify the changes

**Create partition with /dev/sdb**

# fdisk /dev/sdb

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0x55693454.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

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): **1**

First cylinder (1-261, default 1): **1**

Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261): **+1G**

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

Disk /dev/sdb: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x55693454

   Device Boot      Start         End      Blocks   Id  System

/dev/sdb1               1         132     1060258+  **83**  Linux

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

Selected partition **1**

Hex code (type L to list codes): **L**

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris

 1  FAT12           39  Plan 9          82  Linux swap / So c1  DRDOS/sec (FAT-

 2  XENIX root      3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-

 3  XENIX usr       40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/sec (FAT-

 4  FAT16 <32M      41  PPC PReP Boot   85  Linux extended  c7  Syrinx

 5  Extended        42  SFS             86  NTFS volume set da  Non-FS data

 6  FAT16           4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .

 7  HPFS/NTFS       4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility

 8  AIX             4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt

 9  AIX bootable    50  OnTrack DM      93  Amoeba          e1  DOS access

 a  OS/2 Boot Manag 51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O

 b  W95 FAT32       52  CP/M            9f  BSD/OS          e4  SpeedStor

 c  W95 FAT32 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs

 e  W95 FAT16 (LBA) 54  OnTrackDM6      a5  FreeBSD         ee  GPT

 f  W95 Ext'd (LBA) 55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/

10  OPUS            56  Golden Bow      a7  NeXTSTEP        f0  Linux/PA-RISC b

11  Hidden FAT12    5c  Priam Edisk     a8  Darwin UFS      f1  SpeedStor

12  Compaq diagnost 61  SpeedStor       a9  NetBSD          f4  SpeedStor

14  Hidden FAT16 <3 63  GNU HURD or Sys ab  Darwin boot     f2  DOS secondary

16  Hidden FAT16    64  Novell Netware  af  HFS / HFS+      fb  VMware VMFS

17  Hidden HPFS/NTF 65  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE

18  AST SmartSleep  70  DiskSecure Mult b8  BSDI swap       fd  Linux raid auto

1b  Hidden W95 FAT3 75  PC/IX           bb  Boot Wizard hid fe  LANstep

1c  Hidden W95 FAT3 80  Old Minix       be  Solaris boot    ff  BBT

1e  Hidden W95 FAT1

Hex code (type L to list codes): **8e**

Changed system type of partition 1 to 8e (Linux LVM)

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

Disk /dev/sdb: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x55693454

   Device Boot      Start         End      Blocks   Id  System

/dev/sdb1               1         132     1060258+  8e  Linux LVM

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

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

# partprobe /dev/sdb1Now at this stage our partition is created as **/dev/sdb1** with LVM type.

**Create Physical Volume**

# pvcreate /dev/sdb1  
  Physical volume "/dev/sdb1" successfully created

**Extend Volume Group**

Before extending the size of our existing Volume group let us verify the size of the same  
# vgdisplay  
  --- Volume group ---  
  VG Name               VolGroup  
  System ID  
  Format                lvm2  
  Metadata Areas        1  
  Metadata Sequence No  5  
  VG Access             read/write  
  VG Status             resizable  
  MAX LV                0  
  Cur LV                2  
  Open LV               2  
  Max PV                0  
  Cur PV                1  
  Act PV                1  
  VG Size               9.80 GiB  
  PE Size               4.00 MiB  
  Total PE              2509  
  Alloc PE / Size       2009 / 7.85 GiB  
  Free  PE / Size       500 / 1.95 GiB  
  VG UUID               uH5AP5-b24E-92h7-nL8b-7Bio-fXe3-pstWIW  
So our Volume Group contains **9.80 GB**  
  
Next extend the Volume Group using the below command  
# vgextend VolGroup /dev/sdb1  
  Volume group "VolGroup" successfully extended  
Verify the change  
# vgdisplay  
  --- Volume group ---  
  VG Name               VolGroup  
  System ID  
  Format                lvm2  
  Metadata Areas        2  
  Metadata Sequence No  6  
  VG Access             read/write  
  VG Status             resizable  
  MAX LV                0  
  Cur LV                2  
  Open LV               2  
  Max PV                0  
  Cur PV                2  
  Act PV                2  
  VG Size               10.81 GiB  
  PE Size               4.00 MiB  
  Total PE              2767  
  Alloc PE / Size       2009 / 7.85 GiB  
  Free  PE / Size       758 / 2.96 GiB  
  VG UUID               uH5AP5-b24E-92h7-nL8b-7Bio-fXe3-pstWIW  
As you can verify the change i.e. extra 1 GB has been added to our Volume Group which we can use to extend out root filesystem

**Extend LVM**

Before extending the root fs let us verify the existing size  
# lvdisplay /dev/VolGroup/root  
  --- Logical volume ---  
  LV Path                /dev/VolGroup/root  
  LV Name                root  
  VG Name                VolGroup  
  LV UUID                Qn8TnI-TLNm-rl4Y-ORnd-zU3p-2Kj1-ALSLAg  
  LV Write Access        read/write  
  LV Creation host, time ,  
  LV Status              available  
  # open                 1  
  LV Size                5.85 GiB  
  Current LE             1497  
  Segments               1  
  Allocation             inherit  
  Read ahead sectors     auto  
  - currently set to     256  
  Block device           253:0  
So our existing LVM size for root is **5.85GB**  
  
Let us extend the size with **+1GB**  
# lvextend -L +1G /dev/VolGroup/root  
  Extending logical volume root to 6.85 GiB  
  Logical volume root successfully resized  
Verify the changes  
# lvdisplay /dev/VolGroup/root  
  --- Logical volume ---  
  LV Path                /dev/VolGroup/root  
  LV Name                root  
  VG Name                VolGroup  
  LV UUID                Qn8TnI-TLNm-rl4Y-ORnd-zU3p-2Kj1-ALSLAg  
  LV Write Access        read/write  
  LV Creation host, time ,  
  LV Status              available  
  # open                 1  
  LV Size                6.85 GiB  
  Current LE             1753  
  Segments               2  
  Allocation             inherit  
  Read ahead sectors     auto  
  - currently set to     256  
  Block device           253:0  
Let us see if the changes are reflected using df  
# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup-root  
                      **5.8G**  4.1G  1.4G  76% /  
tmpfs                 504M  260K  504M   1% /dev/shm  
/dev/sda1             194M   26M  158M  15% /boot  
But it still shows old size for root. The reason is because we need to update the partition table with the new value of LVM which can be done using the below command  
# resize2fs /dev/VolGroup/root  
resize2fs 1.41.12 (17-May-2010)  
Filesystem at /dev/VolGroup/root is mounted on /; on-line resizing required  
old desc\_blocks = 1, new\_desc\_blocks = 1  
Performing an on-line resize of /dev/VolGroup/root to 1795072 (4k) blocks.  
The filesystem on /dev/VolGroup/root is now 1795072 blocks long.  
  
In case you notice above initially the command throws out an error as we were attempting to resize on a mounted partition but since our file system is ext4, we get an additional feature of online resizing where we do not need to unmount the file system for extending its size  
  
**NOTE:** We can resize the file system online if following conditions are met

* 2.6.x kernel sereis
* Must be formatted with ext3 fs or above

**Re-verify the change now**

# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup-root  
                      **6.8G**  4.2G  2.3G  65% /  
tmpfs                 504M  260K  504M   1% /dev/shm  
/dev/sda1             194M   26M  158M  15% /boot  
So everything worked as expected.

### [How to reduce LVM size in Linux step by step (online without reboot)](http://www.golinuxhub.com/2017/07/how-to-reduce-lvm-size-in-linux-step-by.html)

POSTED BY DEEPAK PRASAD SATURDAY, JULY 01, 2017 [NO COMMENTS](http://www.golinuxhub.com/2017/07/how-to-reduce-lvm-size-in-linux-step-by.html#comment-form)

#### Do you have a node where there is a lot of free space available on some logical volume which is unused while there is a different partition which you are using is going out of space?

In such cases you need not add extra disk to extend your volume group and your logical volume, why not we take some free space from this large lvm and extend our partition, which can be performed on the same session without reboot.

**IMPORTANT NOTE**: The article shows the steps to reduce logical volume online but it is recommended to do these steps in runlevel one as there your partitions will not be used by any process, in any other run level there is a high risk that partition will be in used so you won't be able to perform these LVM operations.

Reducing your LVM may risk in loosing the data stored in the partition so perform these steps at your own risk.

In this article I will show you the steps to perform a LVM shrinking and using the shrinked space to extend another partition.  
  
  
If you are looking for steps to extend logical volume by adding extra disk to your node I would recommend you to follow below tutorial  
[How to extend/resize Logical Volume and Volume Group in Linux](http://www.golinuxhub.com/2014/03/how-to-extendresize-lvm-and-volume.html)  
  
These steps are validated on Red Hat Enterprise Linux 7  
  
  
Below is my sample setup where I have almost 90 GB free in /opt/sdf/backup partition while my /tmp partition has very less size, so lets take 10 GB from /opt/sdf/backup and add it to /tmp  
# df -h  
Filesystem                    Size  Used Avail Use% Mounted on  
/dev/mapper/system-root       2.0G  1.4G  453M  76% /  
devtmpfs                      3.8G     0  3.8G   0% /dev  
tmpfs                         3.8G     0  3.8G   0% /dev/shm  
tmpfs                         3.8G  8.9M  3.8G   1% /run  
tmpfs                         3.8G     0  3.8G   0% /sys/fs/cgroup  
/dev/sda1                     120M   92M   20M  83% /boot  
/dev/mapper/system-tmp        2.0G  6.8M  1.8G   1% /tmp  
/dev/mapper/system-opt        2.0G  220M  1.6G  12% /opt  
/dev/mapper/system-sdf        5.8G   66M  5.5G   2% /opt/sdf  
tmpfs                         1.0M  4.0K 1020K   1% /opt/sdf/queues  
/dev/mapper/system-var        2.0G   63M  1.8G   4% /var  
/dev/mapper/system-sdfbackup   97G   63M   92G   1% /opt/sdf/backup  
Make sure the partition you are planning to shrink is not being used by any process, this can be validated using below command  
# lsof /opt/sdf/backup  
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF NODE NAME  
bash    18978 root  cwd    DIR  253,2     4096    2 /opt/sdf/backup  
If you get any output like above it means that the partition is in use by some process so you cannot continue with LVM shrinking, make sure all the processes are closed using this partition before starting  
  
You must get a blank output from this command as below  
# lsof /opt/sdf/backup  
Next un-mount the partition  
# umount /opt/sdf/backup  
Do a filesystem check to make sure your disk is healthy and will survive the LVM operations, obviously we do not want our disk to get corrupted in between leaving us no other option and risking our valuable data.  
# e2fsck -f /dev/mapper/system-sdfbackup  
e2fsck 1.42.9 (28-Dec-2013)  
Pass 1: Checking inodes, blocks, and sizes  
Pass 2: Checking directory structure  
Pass 3: Checking directory connectivity  
Pass 4: Checking reference counts  
Pass 5: Checking group summary information  
/dev/mapper/system-sdfbackup: 60/6414336 files (0.0% non-contiguous), 451402/25649152 blocks  
If you get any warning or error in above step, it is strongly recommend to validate your disk health status and do not perform lvm reduction.  
  
Next resize your LVM to the size you want the LVM to be, for eg I want my partition to become 80G after reduction so I am freeing up around 17 GB of size from existing volume  
# resize2fs /dev/mapper/system-sdfbackup 80G  
resize2fs 1.42.9 (28-Dec-2013)  
Resizing the filesystem on /dev/mapper/system-sdfbackup to 20971520 (4k) blocks.  
The filesystem on /dev/mapper/system-sdfbackup is now 20971520 blocks long.  
This passed successfully  
Next perform lvreduce with same size option as used above.  
# lvreduce -L 80G /dev/mapper/system-sdfbackup  
  WARNING: Reducing active logical volume to 80.00 GiB.  
  THIS MAY DESTROY YOUR DATA (filesystem etc.)  
Do you really want to reduce system/sdfbackup? [y/n]: y  
  Size of logical volume system/sdfbackup changed from 97.84 GiB (3131 extents) to 80.00 GiB (2560 extents).  
  Logical volume system/sdfbackup successfully resized.  
We are done here and ready to mount our partition  
# mount /dev/mapper/system-sdfbackup /opt/sdf/backup  
Validate the new size  
# df -h /dev/mapper/system-sdfbackup  
Filesystem                    Size  Used Avail Use% Mounted on  
/dev/mapper/system-sdfbackup   79G   59M   75G   1% /opt/sdf/backup  
Time to extend our /tmp partition with the available size now  
**NOTE:**Keep some free size buffer free space depending upon the size reduced as there might not be enough free blocks available.  
  
Here I will extend my **/tmp** partition with **10 GB**  
# lvextend -L +10G /dev/mapper/system-tmp  
  Size of logical volume system/tmp changed from 2.00 GiB (64 extents) to 12.00 GiB (384 extents).  
  Logical volume system/tmp successfully resized.  
Now resize your filesystem with the changes we did above  
# resize2fs /dev/mapper/system-tmp  
resize2fs 1.42.9 (28-Dec-2013)  
Filesystem at /dev/mapper/system-tmp is mounted on /tmp; on-line resizing required  
old\_desc\_blocks = 1, new\_desc\_blocks = 2  
The filesystem on /dev/mapper/system-tmp is now 3145728 blocks long.  
Check the output status of our last command  
# echo $?  
0  
The magic has been done and we can see our /tmp partition size is now 12 GB  
# df -h /dev/mapper/system-tmp  
Filesystem              Size  Used Avail Use% Mounted on  
/dev/mapper/system-tmp   12G   11M   12G   1% /tmp  
  
I hope the article was useful.

[**How to extend/resize Logical Volume and Volume Group in Linux**](http://www.golinuxhub.com/2014/03/how-to-extendresize-lvm-and-volume.html)

POSTED BY DEEPAK PRASAD FRIDAY, MARCH 21, 2014 [10 COMMENTS](http://www.golinuxhub.com/2014/03/how-to-extendresize-lvm-and-volume.html#comment-form)

In the below article I will show you the steps to add a new disk which can be added into existing volume group to extend a logical volume.  
  
But if you want to reduce (shrink) an existing logical volume and use the extra space to extend another logical volume, please follow below link  
[How to reduce LVM size in Linux step by step (online without reboot)](http://www.golinuxhub.com/2017/07/how-to-reduce-lvm-size-in-linux-step-by.html)  
  
As of now you can see my root partition is on LVM with 5.8Gb size. I want to extend my root filesystem with +1GB. Now here there can be two scenarios where  
  
# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup-root  
                      5.8G  4.1G  1.4G  76% /  
tmpfs                 504M  260K  504M   1% /dev/shm  
/dev/sda1             194M   26M  158M  15% /boot  
1. You want to extend size using extra partition  
2. You want to extend size with space in existing Volume Group  
  
I will show you steps required to perform the same using both the scenarios

**Scenario 1**

**Extend LVM using additional partition**

# fdisk -l

Disk /dev/sda: 10.7 GB, 10737418240 bytes

255 heads, 63 sectors/track, 1305 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: 0x000cf049

   Device Boot      Start         End      Blocks   Id  System

/dev/sda1   \*           1          26      204800   83  Linux

Partition 1 does not end on cylinder boundary.

/dev/sda2              26        1306    10279936   8e  Linux LVM

Disk **/dev/sdb**: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x00000000

Disk /dev/mapper/VolGroup-root: 6278 MB, 6278873088 bytes

255 heads, 63 sectors/track, 763 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: 0x00000000

Disk /dev/mapper/VolGroup-swap: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x00000000

So as you see I have added extra partition hard disk **/dev/sdb** with **2GB** space. Let us use it to extend our root file system.  
  
Steps required to perform the same  
1. Create partition for /dev/sdb with 1 GB storage  
2. Create Physical Volume  
3. Extend existing Volume Group  
4. Extend LVM  
5. Verify the changes

**Create partition with /dev/sdb**

# fdisk /dev/sdb

Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel

Building a new DOS disklabel with disk identifier 0x55693454.

Changes will remain in memory only, until you decide to write them.

After that, of course, the previous content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

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): **1**

First cylinder (1-261, default 1): **1**

Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261): **+1G**

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

Disk /dev/sdb: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x55693454

   Device Boot      Start         End      Blocks   Id  System

/dev/sdb1               1         132     1060258+  **83**  Linux

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

Selected partition **1**

Hex code (type L to list codes): **L**

 0  Empty           24  NEC DOS         81  Minix / old Lin bf  Solaris

 1  FAT12           39  Plan 9          82  Linux swap / So c1  DRDOS/sec (FAT-

 2  XENIX root      3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-

 3  XENIX usr       40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/sec (FAT-

 4  FAT16 <32M      41  PPC PReP Boot   85  Linux extended  c7  Syrinx

 5  Extended        42  SFS             86  NTFS volume set da  Non-FS data

 6  FAT16           4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .

 7  HPFS/NTFS       4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility

 8  AIX             4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt

 9  AIX bootable    50  OnTrack DM      93  Amoeba          e1  DOS access

 a  OS/2 Boot Manag 51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O

 b  W95 FAT32       52  CP/M            9f  BSD/OS          e4  SpeedStor

 c  W95 FAT32 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs

 e  W95 FAT16 (LBA) 54  OnTrackDM6      a5  FreeBSD         ee  GPT

 f  W95 Ext'd (LBA) 55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/

10  OPUS            56  Golden Bow      a7  NeXTSTEP        f0  Linux/PA-RISC b

11  Hidden FAT12    5c  Priam Edisk     a8  Darwin UFS      f1  SpeedStor

12  Compaq diagnost 61  SpeedStor       a9  NetBSD          f4  SpeedStor

14  Hidden FAT16 <3 63  GNU HURD or Sys ab  Darwin boot     f2  DOS secondary

16  Hidden FAT16    64  Novell Netware  af  HFS / HFS+      fb  VMware VMFS

17  Hidden HPFS/NTF 65  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE

18  AST SmartSleep  70  DiskSecure Mult b8  BSDI swap       fd  Linux raid auto

1b  Hidden W95 FAT3 75  PC/IX           bb  Boot Wizard hid fe  LANstep

1c  Hidden W95 FAT3 80  Old Minix       be  Solaris boot    ff  BBT

1e  Hidden W95 FAT1

Hex code (type L to list codes): **8e**

Changed system type of partition 1 to 8e (Linux LVM)

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

Disk /dev/sdb: 2147 MB, 2147483648 bytes

255 heads, 63 sectors/track, 261 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: 0x55693454

   Device Boot      Start         End      Blocks   Id  System

/dev/sdb1               1         132     1060258+  8e  Linux LVM

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

The partition table has been altered!

Calling ioctl() to re-read partition table.

Syncing disks.

# partprobe /dev/sdb1Now at this stage our partition is created as **/dev/sdb1** with LVM type.

**Create Physical Volume**

# pvcreate /dev/sdb1  
  Physical volume "/dev/sdb1" successfully created

**Extend Volume Group**

Before extending the size of our existing Volume group let us verify the size of the same  
# vgdisplay  
  --- Volume group ---  
  VG Name               VolGroup  
  System ID  
  Format                lvm2  
  Metadata Areas        1  
  Metadata Sequence No  5  
  VG Access             read/write  
  VG Status             resizable  
  MAX LV                0  
  Cur LV                2  
  Open LV               2  
  Max PV                0  
  Cur PV                1  
  Act PV                1  
  VG Size               9.80 GiB  
  PE Size               4.00 MiB  
  Total PE              2509  
  Alloc PE / Size       2009 / 7.85 GiB  
  Free  PE / Size       500 / 1.95 GiB  
  VG UUID               uH5AP5-b24E-92h7-nL8b-7Bio-fXe3-pstWIW  
So our Volume Group contains **9.80 GB**  
  
Next extend the Volume Group using the below command  
# vgextend VolGroup /dev/sdb1  
  Volume group "VolGroup" successfully extended  
Verify the change  
# vgdisplay  
  --- Volume group ---  
  VG Name               VolGroup  
  System ID  
  Format                lvm2  
  Metadata Areas        2  
  Metadata Sequence No  6  
  VG Access             read/write  
  VG Status             resizable  
  MAX LV                0  
  Cur LV                2  
  Open LV               2  
  Max PV                0  
  Cur PV                2  
  Act PV                2  
  VG Size               10.81 GiB  
  PE Size               4.00 MiB  
  Total PE              2767  
  Alloc PE / Size       2009 / 7.85 GiB  
  Free  PE / Size       758 / 2.96 GiB  
  VG UUID               uH5AP5-b24E-92h7-nL8b-7Bio-fXe3-pstWIW  
As you can verify the change i.e. extra 1 GB has been added to our Volume Group which we can use to extend out root filesystem

**Extend LVM**

Before extending the root fs let us verify the existing size  
# lvdisplay /dev/VolGroup/root  
  --- Logical volume ---  
  LV Path                /dev/VolGroup/root  
  LV Name                root  
  VG Name                VolGroup  
  LV UUID                Qn8TnI-TLNm-rl4Y-ORnd-zU3p-2Kj1-ALSLAg  
  LV Write Access        read/write  
  LV Creation host, time ,  
  LV Status              available  
  # open                 1  
  LV Size                5.85 GiB  
  Current LE             1497  
  Segments               1  
  Allocation             inherit  
  Read ahead sectors     auto  
  - currently set to     256  
  Block device           253:0  
So our existing LVM size for root is **5.85GB**  
  
Let us extend the size with **+1GB**  
# lvextend -L +1G /dev/VolGroup/root  
  Extending logical volume root to 6.85 GiB  
  Logical volume root successfully resized  
Verify the changes  
# lvdisplay /dev/VolGroup/root  
  --- Logical volume ---  
  LV Path                /dev/VolGroup/root  
  LV Name                root  
  VG Name                VolGroup  
  LV UUID                Qn8TnI-TLNm-rl4Y-ORnd-zU3p-2Kj1-ALSLAg  
  LV Write Access        read/write  
  LV Creation host, time ,  
  LV Status              available  
  # open                 1  
  LV Size                6.85 GiB  
  Current LE             1753  
  Segments               2  
  Allocation             inherit  
  Read ahead sectors     auto  
  - currently set to     256  
  Block device           253:0  
Let us see if the changes are reflected using df  
# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup-root  
                      **5.8G**  4.1G  1.4G  76% /  
tmpfs                 504M  260K  504M   1% /dev/shm  
/dev/sda1             194M   26M  158M  15% /boot  
But it still shows old size for root. The reason is because we need to update the partition table with the new value of LVM which can be done using the below command  
# resize2fs /dev/VolGroup/root  
resize2fs 1.41.12 (17-May-2010)  
Filesystem at /dev/VolGroup/root is mounted on /; on-line resizing required  
old desc\_blocks = 1, new\_desc\_blocks = 1  
Performing an on-line resize of /dev/VolGroup/root to 1795072 (4k) blocks.  
The filesystem on /dev/VolGroup/root is now 1795072 blocks long.  
  
In case you notice above initially the command throws out an error as we were attempting to resize on a mounted partition but since our file system is ext4, we get an additional feature of online resizing where we do not need to unmount the file system for extending its size  
  
**NOTE:** We can resize the file system online if following conditions are met

* 2.6.x kernel sereis
* Must be formatted with ext3 fs or above

**Re-verify the change now**

# df -h  
Filesystem            Size  Used Avail Use% Mounted on  
/dev/mapper/VolGroup-root  
                      **6.8G**  4.2G  2.3G  65% /  
tmpfs                 504M  260K  504M   1% /dev/shm  
/dev/sda1             194M   26M  158M  15% /boot  
So everything worked as expected.