

HOW TO EXTEND FILESYSTEM ON LINUX (ROOT AND OTHER)

Table of Contents

- growpart command :
 - How to Install cloud utils package on the system
 - How to extend root filesystem using LVM on Linux
 - 1. Check the condition of the disk and its structure
 - 2: Extend your desired disk size (root partition or other)
 - Step 3: Resize root logical volume to occupy all space
 - 4: Update changes on the filesystem (If you didn't use -r option in step 3)

In this tutorial, we will teach you how to resize your hard disk partitions in Linux by introducing a very useful tool. You can easily resize all partitions with this utility. Whether this is a root partition or any other partition. No need to enter a rescue mode or reboot. this will also cover both ext4 and XFS filesystem root partition extending. This tool is called **growpart** and we are going to do a lot of things in this tutorial using this tool.

growpart command :

growpart is a [Linux](#) command-line tool used to extend a partition in a partition table to fill available space. This command is provided by cloud utils package.

How to Install cloud utils package on the system

On Ubuntu / Debian system, run the commands below to install *growpart* tool.

```
sudo apt install cloud-guest-utils
```

For CentOS server, run:

```
sudo yum -y install cloud-utils-growpart
```

Help page can be viewed by passing **-h** argument

```

[root@mahdi ~]# growpart -h
growpart disk partition
rewrite partition table so that partition takes up all the space it can
options:
-h | --help          print Usage and exit
  | --fudge F        if part could be resized, but change would be
  |                  less than 'F' bytes, do not resize (default: 1048576)
-N | --dry-run       only report what would be done, show new 'sfdisk -d'
-v | --verbose       increase verbosity / debug
-u | --update R      update the the kernel partition table info after growing
  |                  this requires kernel support and 'partx --update'
  |                  R is one of:
  |                  - 'auto' : [default] update partition if possible
  |                  - 'force' : try despite sanity checks (fail on failure)
  |                  - 'off'  : do not attempt
  |                  - 'on'   : fail if sanity checks indicate no support

Example:
- growpart /dev/sda 1
  Resize partition 1 on /dev/sda
[root@mahdi ~]# █

```

How to extend filesystem on Linux

Ok , Now we can go to extend our root partitions (or other) !

In the first step, we start the tutorial according to the LVM structure on the root partition:

How to extend root filesystem using LVM on Linux

To demonstrate a complete LVM lifecycle, we will perform the following actions:

- Create an LVM physical volume, volume group, and logical volume.
- Create an XFS and ext4 filesystem on the logical volumes
- Extend LVM logical volumes (root and non-root filesystem)

LVM allows you to create, resize or delete partitions on a running system without requiring any reboot. So check the steps below to extend the root filesystem using LVM in Linux. You can skip some steps which don't apply to use.

1. Check the condition of the disk and its structure

Before we can do any extension, let's just check our disk layout/partitioning scheme. (command : **lsblk**)

```
[root@mahdi ~]# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   20G  0 disk
├─sda1                8:1    0    1G  0 part /boot
└─sda2                8:2    0   19G  0 part
   ├─centos-root      253:0   0   17G  0 lvm  /
   └─centos-swap      253:1   0    2G  0 lvm  [SWAP]
sr0                  11:0    1 1024M  0 rom
```

As you see, we have a root filesystem on `/dev/sda2` physical volume.

```
[root@mahdi ~]# pvs
PV          VG      Fmt Attr PSize  PFree
/dev/sda2  centos lvm2 a-- <19.00g  0
```

2: Extend your desired disk size (root partition or other)

As shown in step 1, my root filesystem is on a `20GB` disk. I'll grow it to `50GB` by extending the virtual disk (VM disk device).

If you did not reboot your server after resizing the partition, rescan your SCSI devices as such.

First, check the name(s) of your SCSI devices.

```
$ ls /sys/class/scsi_device/
0:0:0:0 1:0:0:0 2:0:0:0
```

Then rescan the scsi bus. Below you can replace the `'0:0:0:0'` with the actual scsi bus name found with the previous command. Each colon is prefixed with a slash, which is what makes it look weird.

```
~$ echo 1 > /sys/class/scsi_device/0\:\0\:\0\:\0/device/rescan
```

That will rescan the current scsi bus and the disk size that has changed will show up.

```
[root@mahdi ~]# ls /sys/class/scsi_device/
0:0:0:0  3:0:0:0
[root@mahdi ~]# echo 1 > /sys/class/scsi_device/0\:0\:0\:0/device/rescan
[root@mahdi ~]# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0    1G  0 part /boot
├─sda2                8:2    0   19G  0 part
│   ├─centos-root    253:0   0   17G  0 lvm  /
│   └─centos-swap    253:1   0    2G  0 lvm  [SWAP]
sr0                  11:0    1 1024M  0 rom
```

Now use *growpart* to extend your partition. In this example, we're extending partition 2 in disk */dev/sda*. (Replace 2 and */dev/sda* with your correct values.)

```
# growpart /dev/sda 2
```

```
CHANGED: partition=2 start=2099200 old: size=39843840 end=41943040
new: size=102758367 end=104857567
```

Confirm if the change was successful.

```
[root@mahdi ~]# lsblk
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sda                  8:0    0   50G  0 disk
├─sda1                8:1    0    1G  0 part /boot
├─sda2                8:2    0   49G  0 part
│   ├─centos-root    253:0   0   17G  0 lvm  /
│   └─centos-swap    253:1   0    2G  0 lvm  [SWAP]
sr0                  11:0    1 1024M  0 rom
```

Step 3: Resize root logical volume to occupy all space

Resize physical volume. (command:*pvresize*)

```
# pvresize /dev/sda2
Physical volume "/dev/sda2" changed
1 physical volume(s) resized or updated / 0 physical volume(s) not resized
```

```
[root@mahdi ~]# pvs
PV          VG      Fmt  Attr PSize  PFree
 /dev/sda2  centos lvm2 a--  <49.00g 30.00g
[root@mahdi ~]#
```

Check the size of the volume group configured.

```
[root@mahdi ~]# vgs
VG      #PV #LV #SN Attr   VSize  VFree
 centos  1  2   0 wz--n- <49.00g 30.00g
[root@mahdi ~]#
```

Then resize logical volume used by the root file system using the extended volume group:

lvextend -r -l +100%FREE /dev/name-of-volume-group/root

Here's an example of my setup file system extension:

```
[root@mahdi ~]# lvextend -r -l +100%FREE /dev/mapper/centos-root
Size of logical volume centos/root changed from <17.00 GiB (4351 extents) to <47.00 GiB (12031 extents).
Logical volume centos/root successfully resized.
meta-data=/dev/mapper/centos-root isize=512    agcount=4, agsize=1113856 blks
         =                               sectsz=512    attr=2, projid32bit=1
         =                               crc=1      finobt=0 spinodes=0
data     =                               bsize=4096  blocks=4455424, imaxpct=25
         =                               sunit=0     swidth=0 blks
naming   =version 2                      bsize=4096  ascii-ci=0 ftype=1
log      =internal                       bsize=4096  blocks=2560, version=2
         =                               sectsz=512  sunit=0 blks, lazy-count=1
realtime =none                            extsz=4096  blocks=0, rtextents=0
data blocks changed from 4455424 to 12319744
[root@mahdi ~]#
```

This extends the logical volume to use all available capacity in the volume group. With the + sign the value is added to the actual size of the logical volume.

Command options used:

- **-l** – extend or set the logical volume size in units of logical extents
- **-r** – Resize underlying filesystem together with the logical volume

If you prefer setting the size to be extended manually, use command option:

-L, -size [+]LogicalVolumeSize**[bBsSkMgGtTpPeE]**

Where size suffix are:

- **M** for megabytes
- **G** for gigabytes
- **T** for terabytes
- **P** for petabytes
- **E** for exabytes

Without the + sign the value is taken as an absolute one.

Add 20 gigabytes to the current logical volume size
\$ **sudo lvextend -r -L +20G /dev/name-of-volume-group/root**

So we have now:

```

Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        899M   0 899M  0% /dev
tmpfs           910M   0 910M  0% /dev/shm
tmpfs           910M  58M 853M  7% /run
tmpfs           910M   0 910M  0% /sys/fs/cgroup
/dev/mapper/centos-root 67G  1.5G  66G  3% /
/dev/sda1       1014M  192M  823M  19% /boot
tmpfs           182M   0 182M  0% /run/user/0

```

and still 30G Free:

```

[root@mahdi ~]# vgs
VG      #PV #LV #SN Attr   VSize  VFree
centos  1  2  0 wz--n- <99.00g 30.00g
[root@mahdi ~]# pvs
PV      VG      Fmt  Attr PSize  PFree
/dev/sda2 centos lvm2 a--  <99.00g 30.00g

```

4: Update changes on the filesystem (If you didn't use `-r` option in step 3)

Your root filesystem will still show the old size.

```

$ df -hT | grep mapper
/dev/mapper/rhel-root xfs 27G 1.9G 26G 8% /

```

Let's make the filesystem report the actual size, including extended.

For ext4 filesystem

```

sudo resize2fs /dev/name-of-volume-group/root

```

For xfs filesystem

```

$ sudo xfs_growfs /

```