**How to extend root filesystem using LVM on Linux**

Welcome to our guide on how to extend root filesystem using [LVM on Linux](https://computingforgeeks.com/category/linux-tutorials/). This will cover both ext4 and XFS filesystem root partition extending. 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 file systems 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 root filesystem using LVM in Linux. You can skip some steps which don’t apply to use.

If you’re not using LVM, check our guide below which covers extending Ext2/3/4 and XFS file systems.

**Step 1: Confirm Disk Partitions in Distribution.**

Before we can do any extension, let’s just confirm our disk layout / partitioning scheme.

$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sr0 11:0 1 1024M 0 rom

vda 252:0 0 30G 0 disk

├─vda1 252:1 0 1G 0 part /boot

└─vda2 252:2 0 29G 0 part

├─rhel-root 253:0 0 26.9G 0 lvm /

└─rhel-swap 253:1 0 2.1G 0 lvm [SWAP]

As noted, we have a root filesystem on */dev/vda2* physical volume.

$ sudo pvs

PV VG Fmt Attr PSize PFree

/dev/vda2 rhel lvm2 a-- <29.00g 0

**Step 2: Extend your OS root disk**

As shown in step 1, my root filesystem is on a *30GB* disk. I’ll grow it to *40GB* by extending the virtual disk (VM disk device).

I use KVM virtualization technology, so this guide works for me: [How to extend/increase KVM Virtual Machine (VM) disk size﻿](https://computingforgeeks.com/how-to-extend-increase-kvm-virtual-machine-disk-size/)

$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sr0 11:0 1 1024M 0 rom

vda 252:0 0  **40G** 0 disk

├─vda1 252:1 0 1G 0 part /boot

└─vda2 252:2 0 29G 0 part

├─rhel-root 253:0 0 26.9G 0 lvm /

└─rhel-swap 253:1 0 2.1G 0 lvm [SWAP]

If you’re on a different Virtualization platform, refer to its documentation for how to extend OS disk.

Once the OS block device is resized, ssh to your Linux machine and extend LVM to use newly added disk capacity. The command below will expand the last partition (**Partition 2**), as shown by *252:2,*on the disk (*/dev/vda*) to the maximum size the disk provides.

**Install cloud utils package on the system**

For those new to **growpart**, it 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.

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

$ 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/sd

Now use *growpart* to extend your partition. In this example we’re extending partition **2**in disk **/dev/vda.**Replace *2*and */dev/vda*with your correct values.

$ sudo growpart /dev/vda 2

CHANGED: partition=2 start=2099200 old: size=18872320 end=20971520 new: size=60815327,end=62914527

Confirm if the change was successful.

$ lsblk

NAME MAJ:MIN RM SIZE RO TYPE MOUNTPOINT

sr0 11:0 1 1024M 0 rom

vda 252:0 0 40G 0 disk

├─vda1 252:1 0 1G 0 part /boot

└─vda2 252:2 0 **39G** 0 part

├─rhel-root 253:0 0 26.9G 0 lvm /

└─rhel-swap 253:1 0 2.1G 0 lvm [SWAP]

**Step 3: Resize root logical volume to occupy all space**

Resize physical volume.

$ sudo pvresize /dev/vda2

Physical volume "/dev/vda2" changed

1 physical volume(s) resized or updated / 0 physical volume(s) not resized

$ sudo pvs

PV VG Fmt Attr PSize PFree

/dev/vda2 rhel lvm2 a-- <39.00g 10.00g

Check the size of the volume group configured.

$ sudo vgs

VG #PV #LV #SN Attr VSize VFree

rhel 1 2 0 wz--n- <39.00g 10.00g

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

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

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

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

$ df -hT | grep mapper

/dev/mapper/rhel-root xfs **27G** 1.9G 26G 8% /

$ sudo lvextend -r -l +100%FREE /dev/mapper/rhel-root

Size of logical volume rhel/root changed from <26.93 GiB (6893 extents) to <36.93 GiB (9453 extents).

Logical volume rhel/root successfully resized.

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

-L, --size [+]LogicalVolumeSize[bBsSkKmMgGtTpPeE]

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

**Step 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 /

meta-data=/dev/mapper/rhel-root isize=512 agcount=4, agsize=1764608 blks

= sectsz=512 attr=2, projid32bit=1

= crc=1 finobt=1, sparse=1, rmapbt=0

= reflink=1

data = bsize=4096 blocks=7058432, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0, ftype=1

log =internal log bsize=4096 blocks=3446, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

data blocks changed from 7058432 to 9679872

***Recommended Learning Courses for Linux System Administration:***

**Conclusion**

You have learned how to extend root filesystem backed by nfs and ext4 with this how to extend root filesystem using LVM guide. I hope this was helpful and would like to thank you for reading.

**Tags:**

* Resize root filesystem on Linux
* Resize / on Linux with LVM
* Resize LVM root file system on Linux
* Easy way to resize an LVM root filesystem on Linux
* Extend root filesystem on Linux using LVM
* Extend LVM root filesystem on Linux
* Extend LVM / on Linux

Similar Articles: