

Extending Disk Size and Logical Volume For Linux Based Systems

Version V1, 2024-01-12

Table of Contents

- 1. Version History1
- 2. Main Subject.....1
- 3. Main Steps2
- 4. Disk Expansion Procedure3

Chapter 1. Version History

Date	Version	Author	Description
2024-01-12	0.1	Bilel Bouneb	Disk expansion documentation

Chapter 2. Main Subject

This document has been meticulously crafted to provide a systematic and comprehensive guide for the step-by-step addition of a new hard drive partition to a Linux-based system. The detailed instructions aim to facilitate a seamless integration process, ensuring precision and clarity in each operational step for enhanced system management and storage expansion.

Chapter 3. Main Steps

In our strategic approach, we will systematically execute a series of ordered steps to ensure the seamless integration of a new disk into our system. The initial phase involves partitioning the newly added disk to allocate specific storage areas. Following this, we will create a new physical device to establish a connection between the disk and our system, ensuring its recognition. Subsequently, we aim to extend the existing volume group, allowing for a more comprehensive utilization of available storage resources. Building upon this foundation, we will proceed to extend the logical volume, effectively expanding the storage capacity at a higher level of abstraction. The final step in our well-organized process entails the extension of the filesystem, ensuring that the additional storage space is accessible and fully integrated into the existing file structure. By meticulously following these ordered steps, we guarantee a systematic and efficient integration of the new disk into our system architecture, optimizing storage capabilities for enhanced performance and flexibility.

Mainly we will proceed with these ordered steps in our action:

- 1) Partition the new disk**
- 2) Create New Physical Device**
- 3) Extend the existing volume group**
- 4) Extend the logical volume**
- 5) Extend the filesystem**

Chapter 4. Disk Expansion Procedure

1. Gain Root Privileges:

We should grant elevated privileges, to enter a shell with superuser permissions.

```
sudo -s
```

2. Check Disk Space:

```
df -h
```

This command will provide a human-readable summary of disk space usage, displaying information about available and used space on mounted file systems.

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/ubuntu--vg-ubuntu--lv	80G	75G	5G	94%	/
tmpfs	502M	0	502M	0%	/lib/init/rw
udev	497M	136K	496M	1%	/dev
tmpfs	502M	0	502M	0%	/dev/shm
/dev/sda1	228M	16M	201M	8%	/boot

3. Partition the New Disk:

```
fdisk /dev/sdb
```

This command will create a new partition as show below:

```
Welcome to fdisk (util-linux 2.37.2).
Changes will remain in memory only, until you decide to write them. Be careful
before using the write command.

Device does not contain a recognized partition table. Created a new DOS
disklabel with disk identifier 0x9667db8d.

Command (m for help): n Partition
type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions) Select
(default p): p
Partition number (1-4, default 1): 1
First sector (2048-209715199, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-209715199, default 209715199):
```

Created a new partition 1 of type 'Linux' and of size 100 GiB.

Command (m for help): w
The partition table has been altered. Calling ioctl() to
re-read partition table. Syncing disks.

4. Create a Physical Volume:

We should initialize a physical volume on the specified disk partition (/dev/sdb1), preparing it for use in a volume group within a logical volume management (LVM) setup

```
pvcreate /dev/sdb1
```

The output shows the creation of new physical volume:

```
Physical volume "/dev/sdb1" successfully created
```

5. Display Physical Volumes:

We should check some information about physical volumes, their size, allocation, and other relevant attributes in a logical volume management (LVM) configuration.

```
pvdisplay
```

This command will display the currently available physical volumes as below:

```
--- Physical volume --- PV
Name                       /dev/sda1
VG Name                    ubuntu-vg
PV Size                    <78.00 GiB / not usable 0 yes (but
Allocatable PE            full)
Size Total PE              4.00 MiB
Free PE                    19967
Allocated PE               0
PV UUID                    19967
                           f6X9Lu-BhG1-9SJN-u0VI-9iw1-R5LX-d3u975

--- Physical volume --- PV
Name                       /dev/sdb1
VG Name                    ubuntu-vg
PV Size                    <100.00 GiB / not usable 3.00 MiB yes
Allocatable PE            (but full)
Size Total PE              4.00 MiB
                           25599
```

Free PE	0
Allocated PE	25599
PV UUID	B29dWM-WuFx-z2Na-NgPN-Wa2x-hRTJ-

6. Extend the Volume Group:

We will extend the volume group 'ubuntu-vg' (make sure to put your own VG name) by incorporating the physical volume /dev/sdb1 (make sure to put your own PV name), thereby increasing the available storage within the logical volume management (LVM) configuration.

```
vgextend ubuntu-vg /dev/sdb1
```

This command will extend the volume group, as shown below:

```
Volume group "ubuntu-vg" successfully extended
```

7. Display Volume Group Information:

We need to display some information related to a volume group, presenting details such as size, free space, physical volumes, and other attributes within a logical volume management (LVM) setup.

```
vgdisplay
```

This command will display all the information for the volume group as shown below:

```
--- Volume group ---
VG Name                ubuntu-vg
System ID
Format                 lvm2
Metadata Areas         2
Metadata Sequence No   5
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 1
Open LV                 1
Max PV                 0
Cur PV                 2
Act PV                 2
VG Size                 80 GiB
PE Size                 4.00 MiB
Total PE                45566
Alloc PE / Size        45566 / 177.99 GiB
Free  PE / Size         0 / 0
```

VG UUID

1d1xd2-dfng-z4Ur-VdEM-DFcX-sNsU-6A3BCH

8. Display Logical Volume Information:

For this step we need to have a detailed overview of logical volumes, presenting essential information such as size, allocation, and status within a logical volume management (LVM) configuration.

```
lvdisplay
```

This command will display the logical volume information as shown below:

```
--- Logical volume ---
LV Path                /dev/ubuntu-vg/ubuntu-lv
LV Name                ubuntu-lv
VG Name                ubuntu-vg
LV UUID                tdlUbO-0Fz8-MjVy-uKBt-JKRH-1BwQ-Q1iS8G
LV Write Access        read/write
LV Creation host, time ubuntu-server, 2022-10-18 10:35:44 +0000 LV Status
                        available
# open                 1
LV Size                80 GiB
Current LE             45566
Segments               2
Allocation             inherit
Read ahead sectors     auto
- currently set to    256
Block device           253:0
```

9. Extend the Logical Volume:

For this step, it will dynamically increase the size of a logical volume, allowing for the expansion of storage space within a logical volume management (LVM) setup.

Make sure to change (/dev/ubuntu-vg/ubuntu-lv) with your own logical volume path!

```
lvextend -l+100%FREE /dev/ubuntu-vg/ubuntu-lv
```

This command will increase the size of a logical volume by utilizing 100% of the available free space as shown below:

```
Logical volume ubuntu-vg/ubuntu-lv successfully resized.
```

10. Resize the Filesystem:

Finally, we adjust the size of the file system to match the altered size of the underlying logical

volume, ensuring efficient utilization of storage within a Linux environment.

```
resize2fs /dev/mapper/ubuntu--vg-ubuntu--lv
```

This command will successfully adjust the file system size, expanding or shrinking it to match the altered partition or logical volume dimensions as shown below:

```
The filesystem on /dev/mapper/ubuntu--vg-ubuntu--lv is now 46659584 (4k) blocks long.
```

11. Final check:

Done. To validate, let's run the same command we ran at the beginning of this article.

First, we run "exit" command to exit the super user:

```
exit
```

We will see a confirmation as shown below:

```
logout
```

Then we check the summary of disk space usage, the new information about available and used space on our mounted file systems.

```
df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/ubuntu--vg-ubuntu--lv	180G	75G	105G	42%	/
tmpfs	502M	0	502M	0%	/lib/init/rw
udev	497M	136K	496M	1%	/dev
tmpfs	502M	0	502M	0%	/dev/shm
/dev/sda1	228M	16M	201M	8%	/boot

As we can see, the new hard disk has been mounted and ready.