# How to Create Disk Storage with Logical Volume Management (LVM) in Linux – PART 1

Ravi Saive | Last Updated: August 7, 2023 | Read Time: 10 mins | Storage | 24 Comments

Logical Volume Management (LVM) is a powerful tool that greatly simplifies disk space management. With LVM, the task of allocating additional space to a file system becomes effortless.

Whenever a file system requires more space, it can easily draw from the available free space in its corresponding volume group. As a result, file systems can be dynamically resized to accommodate our specific needs.

Furthermore, LVM offers robust data protection in the event of disk failures. If a disk begins to show signs of failure, we can seamlessly integrate a replacement disk into the volume group as a new physical volume.

By doing so, the logical volumes can be efficiently migrated to the fresh disk, ensuring continuity and safeguarding against any potential data loss.

In essence, LVM empowers administrators to handle disk space allocation and management with unparalleled flexibility and resilience. Its dynamic resizing capabilities and built-in redundancy features make it an indispensable asset for optimizing storage utilization and enhancing data integrity in any system configuration.

Introducing the comprehensive series "**Preparation for Setting Up LVM (Logical Volume Management)**", thoughtfully crafted across Parts 1 to 6, with an in-depth exploration of the following essential topics:

1. *How to Setup Disk Storage with Logical Volume Management*
2. *How to Extend or Reduce LVM (Logical Volume Management) in Linux*
3. *How to Take Snapshot of Logical Volume and Restore in LVM*
4. *How to Setup Thin Provisioning Volumes in Logical Volume Management (LVM)*
5. *How to Manage Multiple LVM Disks using Striping I/O*
6. *How to Migrate LVM Partitions to New Logical Volume (Drive)*

Before proceeding with the LVM setup, consider the following prerequisites:

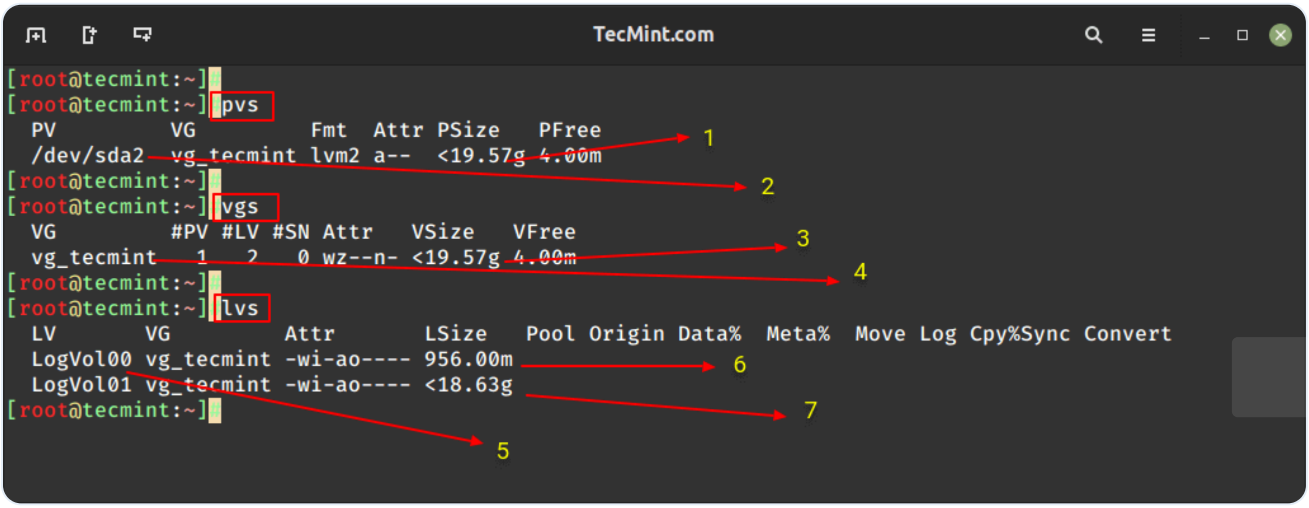
## My Server Setup – Requirements

- OS – RHEL 9 with LVM Installation
- IP – 192.168.0.200
- Disks – 3 disks with 20GB each.

## Check LVM Disk Storage in Linux

**1.** To gain insight into our LVM setup, we can utilize the following commands to reveal the distinct components: Physical Volume (PV), Volume Group (VG), and Logical Volume (LV).

```
# pvs
# vgs
# lvs
```



```
[root@tecmint:~]#
[root@tecmint:~]#pvs
  PV         VG         Fmt  Attr PSize   PFree
  /dev/sda2  vg_tecmint lvm2 a--  <19.57g 4.00m
[root@tecmint:~]#
[root@tecmint:~]#vgs
  VG         #PV #LV #SN Attr   VSize   VFree
  vg_tecmint   1   2   0 wz--n- <19.57g 4.00m
[root@tecmint:~]#
[root@tecmint:~]#lvs
  LV       VG         Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  LogVol00 vg_tecmint -wi-ao---- 956.00m
  LogVol01 vg_tecmint -wi-ao---- <18.63g
[root@tecmint:~]#
```

List LVM Setup in Linux

Here, is the description of each parameter shown in the above screenshot.

- Physical Disk Size (PV Size)
- The disk used was Virtual Disk **sda**.
- Volume Group Size (VG Size)
- Volume Group name (**vg_tecmint**)
- Logical Volume name (**LogVol00**, **LogVol01**)
- **LogVol00** Assigned for a swap with 956MB Size
- **LogVol01** Assigned for/with 18.63GB

So, from here we come to know that there is not enough free space in the VDA disk.

## Create a New Volume Group in LVM

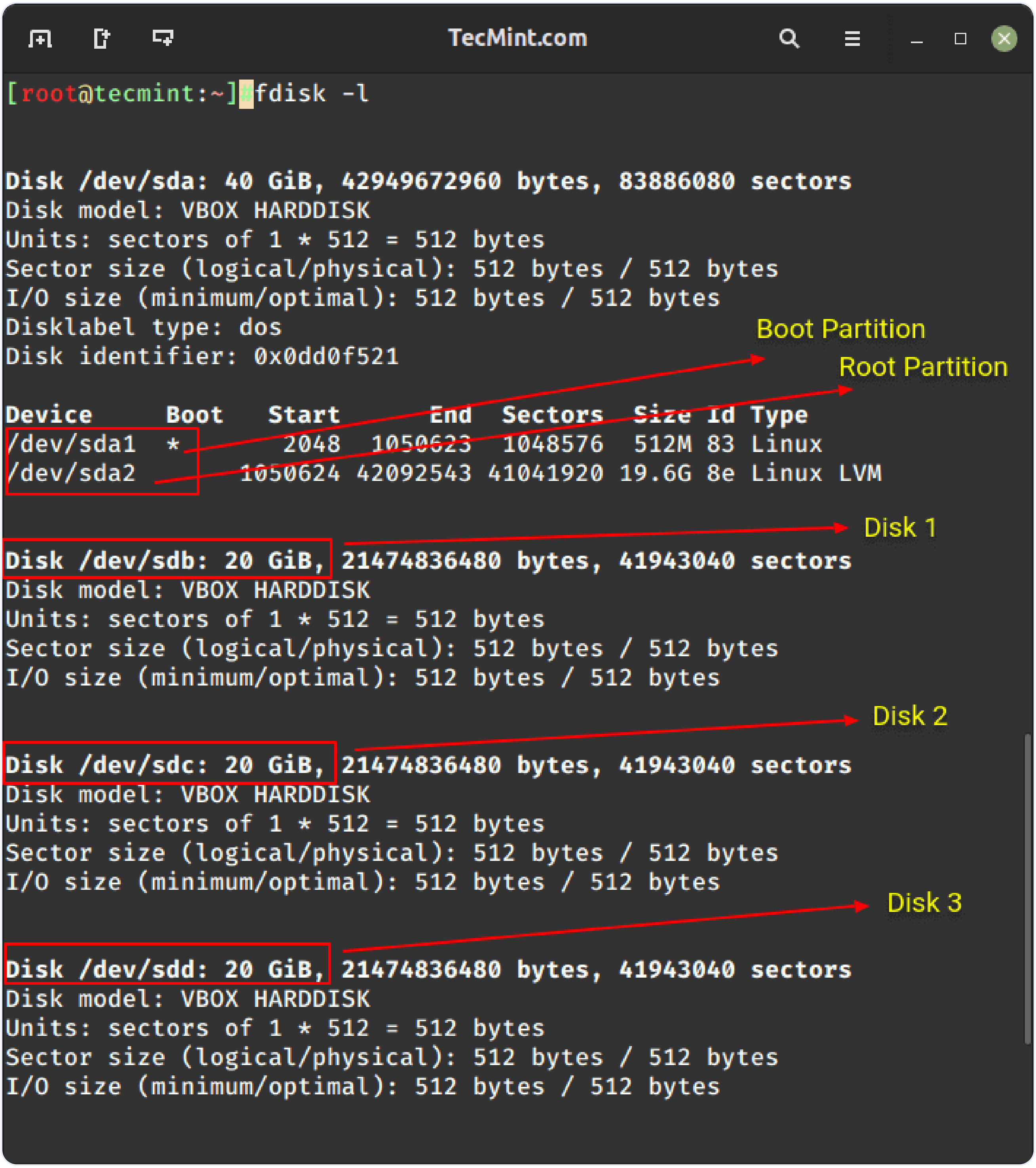
**2.** To create a new **Volume Group**, we need to add an additional 3 hard disks to this server. However, it is not compulsory to use 3 drives; just 1 is enough to create a new **VG** and **LV** (**Logical Volume**) inside that **VG**.

I am adding the following 3 disks here for demonstration purposes and to provide more feature command explanations.

```
sdb, sdc, sdd
```

**3.** To list all the disks and their partitions, such as the disk name, size, partition type, start and end sectors, and more use the fdisk utility as shown.

```
# fdisk -l
```



```
[root@tecmint:~]#fdisk -l


Disk /dev/sda: 40 GiB, 42949672960 bytes, 83886080 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x0dd0f521

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048  1050623  1048576  512M 83 Linux
/dev/sda2       1050624 42092543 41041920 19.6G 8e Linux LVM


Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdc: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes


Disk /dev/sdd: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

List Disk Partitions in Linux

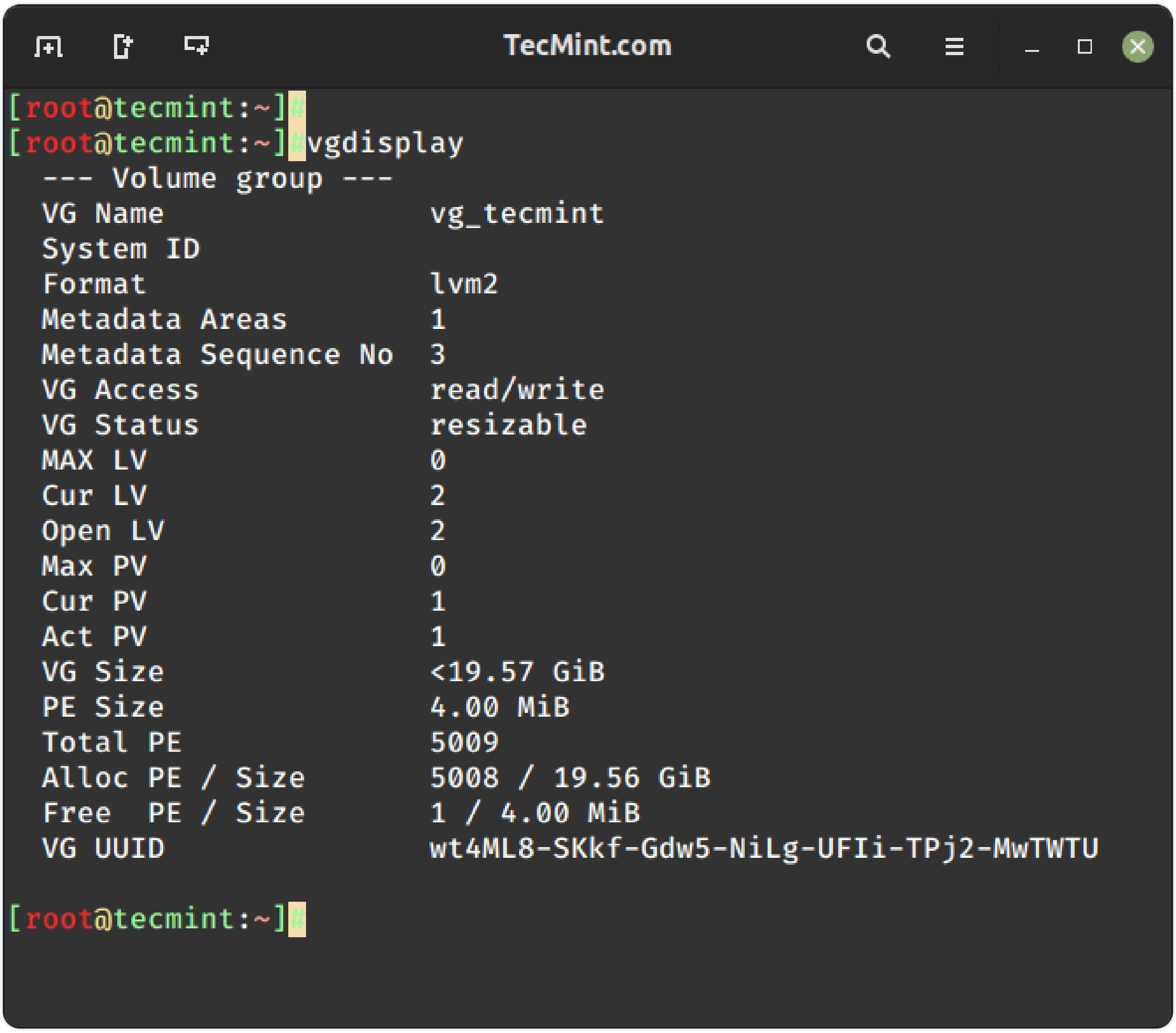
Here, is the description of each disk shown in the above screenshot.

- The default disk used for the Operating System is RHEL 9.
- Partitions defined on the default disk are as follows: (sda1 = boot), (sda2 = /).
- Additionally, added disks are mentioned as Disk1, Disk2, and Disk3.

Each and every disk is 20 GB in size.

**4.** Now run the **vgdisplay** command to view the detailed information about all the Volume Groups present on the system, including their name, size, free space, physical volume (PV) information, and more.

```
# vgdisplay
```



```
[root@tecmint:~]#
[root@tecmint:~]#vgdisplay
  --- Volume group ---
  VG Name               vg_tecmint
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  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               <19.57 GiB
  PE Size               4.00 MiB
  Total PE              5009
  Alloc PE / Size       5008 / 19.56 GiB
  Free  PE / Size       1 / 4.00 MiB
  VG UUID               wt4ML8-SKkf-Gdw5-NiLg-UFIi-TPj2-MwTWTU

[root@tecmint:~]#
```

List Volume Groups in Linux

Here, is the description of each parameter shown in the above screenshot.

- VG Name – A volume group name.
- Format – LVM architecture used lvm2.
- VG Access – The **Volume Group** is in read-and-write mode and ready to use.

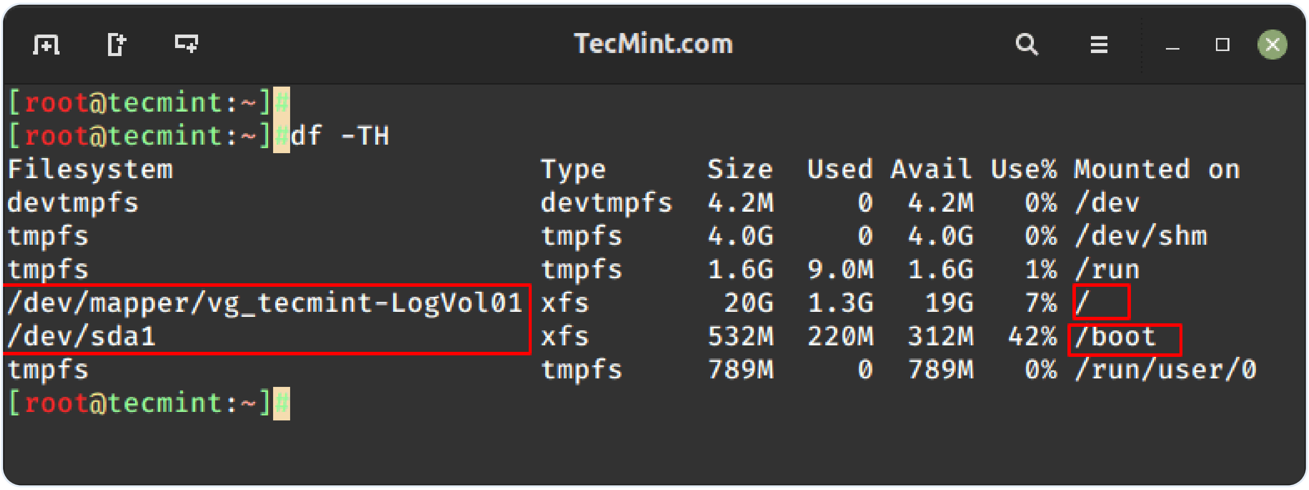
- VG Status – The Volume Group can be resized. We can expand it if we need to add more space.
- Cur LV – Currently, there are 2 Logical volumes in this Volume Group.
- CurPV and Act PV – Currently, the physical disk in use is 1 (vda), and it's active. So, we can use this Volume Group.
- PE Size – Physical Extents (PEs) and size for a disk can be defined using either PE or GB size. The default PE size of LVM is 4 MB. For example, if we need to create a 5 GB logical volume, we can use a sum of 1280 PEs. Do you understand what I'm saying?

Here's the explanation: 1 GB is equal to 1024 MB, so 1024 MB x 5 = 5120 PE = 5 GB. Now, divide 5120 by 4 = 1280. 4 is the default PE size.

- Total PE – This Volume Group has.
- Alloc PE – Total PE Used, full PE already Used, 5008 x 4PE = 20032.
- Free PE – Here it's already used so there was no free PE.

5. Now list the file system disk space information, here only **sda** is used with **/boot**, **/**, and **swap** on the sda physical disk using LVM. There is no space remaining on this disk.

```
# df -TH
```



```
[root@tecmint:~]#
[root@tecmint:~]#df -TH
Filesystem                       Type      Size  Used Avail Use% Mounted on
devtmpfs                         devtmpfs  4.2M     0  4.2M   0% /dev
tmpfs                            tmpfs     4.0G     0  4.0G   0% /dev/shm
tmpfs                            tmpfs     1.6G  9.0M  1.6G   1% /run
/dev/mapper/vg_tecmint-LogVol01  xfs        20G  1.3G   19G   7% /
/dev/sda1                        xfs       532M  220M  312M  42% /boot
tmpfs                            tmpfs     789M     0  789M   0% /run/user/0
[root@tecmint:~]#
```

List File System Disk Space

The above image shows the mount point we are using, and the **19GB** is fully used for the root, so there is no free space available.

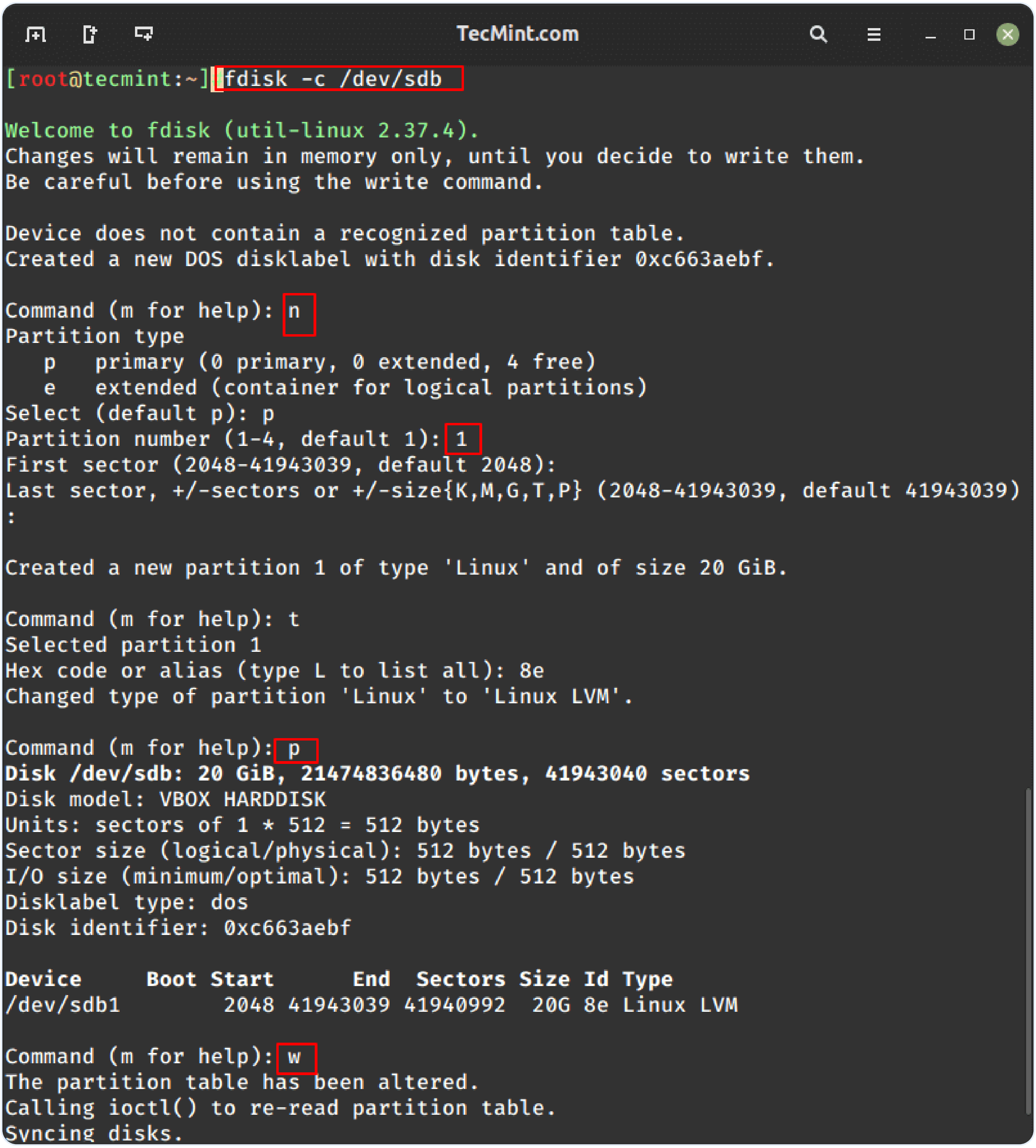
## Create a Disk Partition

**6.** So, let's create a new physical volume (PV) and volume group (VG) named **tecmint_add_vg**, and create logical volumes (LVs) within it. Here, we can create 4 logical volumes with the names **tecmint_documents**, **tecmint_manager**, and **tecmint_public**.

We can extend the **Volume Group** of the currently used **VG** to get more space. However, in this case, we are going to create a new **Volume Group** and experiment with it. Later, we can see how to extend the file systems of the **Volume Group** that is currently in use.

Before using a new disk, we need to partition the disk using the **fdisk** command as shown.

```
# fdisk -c /dev/sdb
```



```
[root@tecmint:~]# fdisk -c /dev/sdb

Welcome to fdisk (util-linux 2.37.4).
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 0xc663aebf.

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 (2048-41943039, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-41943039, default 41943039)
:

Created a new partition 1 of type 'Linux' and of size 20 GiB.

Command (m for help): t
Selected partition 1
Hex code or alias (type L to list all): 8e
Changed type of partition 'Linux' to 'Linux LVM'.

Command (m for help): p
Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc663aebf

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 41943039 41940992  20G 8e Linux LVM

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

Create /dev/sdb Disk Partition

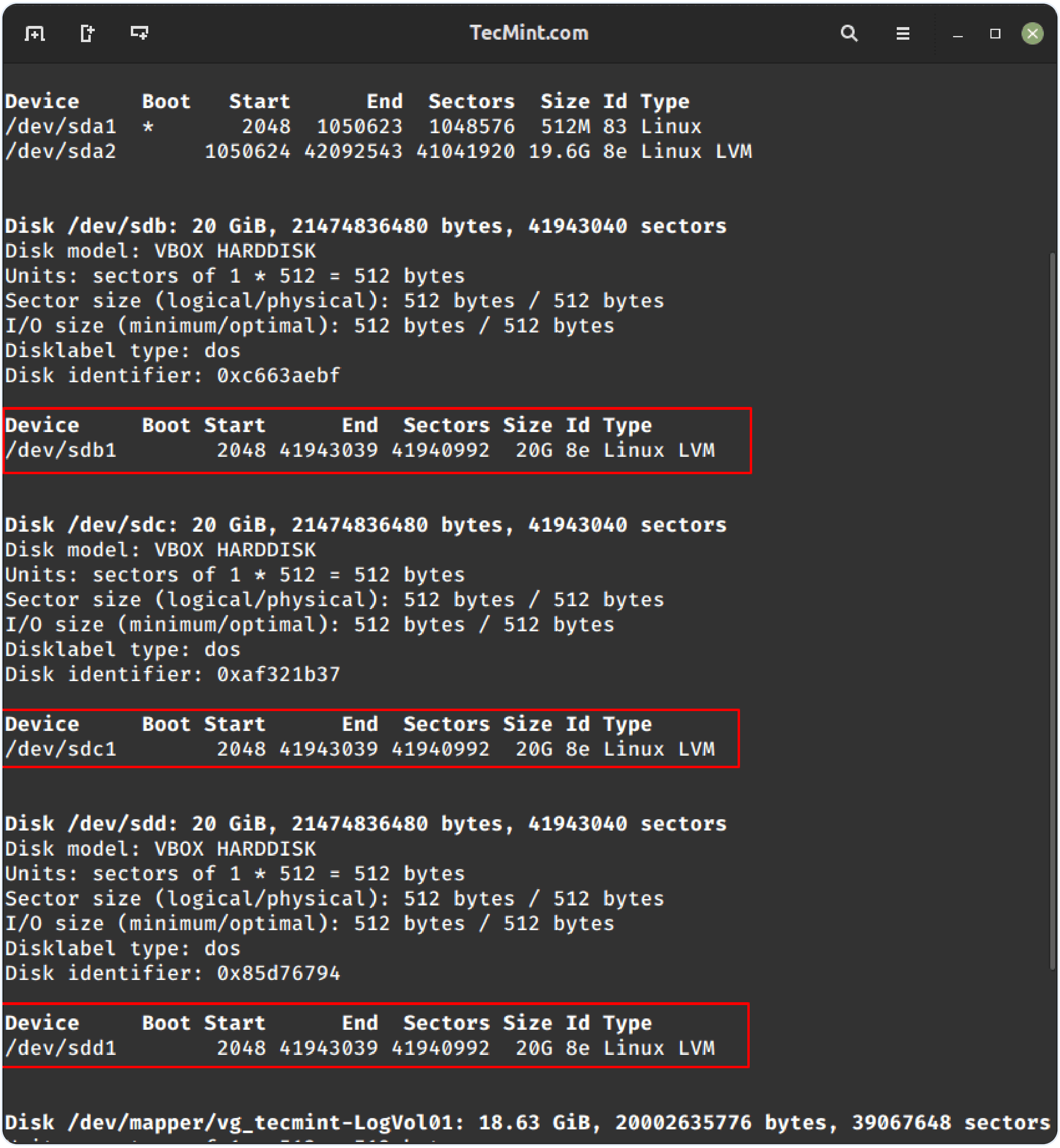
Next, follow the below steps to create a new partition.

- Choose `n` to create new.
- Choose `p` to create a primary partition.
- Choose which number of partitions we need to create.
- Press **enter** twice to use the full space of the disk.
- We need to change the type of newly created partition type `t`.

- Which number of partition need to change, choose the number which we created its `1`.
- Here we need to change the type, we need to create LVM so we going to use the type code of LVM as `8e`, if we do not know the type code Press `L` to list all types of codes.
- Print the partition that we created to just confirm.
- Here we can see the ID as 8e LINUX LVM.
- Write the changes and exit the fdisk.

7. Do the above steps for the other 2 disks `sdc` and `sdd` to create new partitions. Then restart the machine to verify the partition table using the fdisk command.

```
# fdisk -l
```



```
Device     Boot   Start      End  Sectors  Size Id Type
/dev/sda1  *       2048  1050623  1048576  512M 83 Linux
/dev/sda2       1050624 42092543 41041920 19.6G 8e Linux LVM


Disk /dev/sdb: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xc663aebf

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 41943039 41940992  20G 8e Linux LVM


Disk /dev/sdc: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xaf321b37

Device     Boot Start      End  Sectors Size Id Type
/dev/sdc1        2048 41943039 41940992  20G 8e Linux LVM


Disk /dev/sdd: 20 GiB, 21474836480 bytes, 41943040 sectors
Disk model: VBOX HARDDISK
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x85d76794

Device     Boot Start      End  Sectors Size Id Type
/dev/sdd1        2048 41943039 41940992  20G 8e Linux LVM


Disk /dev/mapper/vg_tecmint-LogVol01: 18.63 GiB, 20002635776 bytes, 39067648 sectors
```

Confirm Disk Partitions

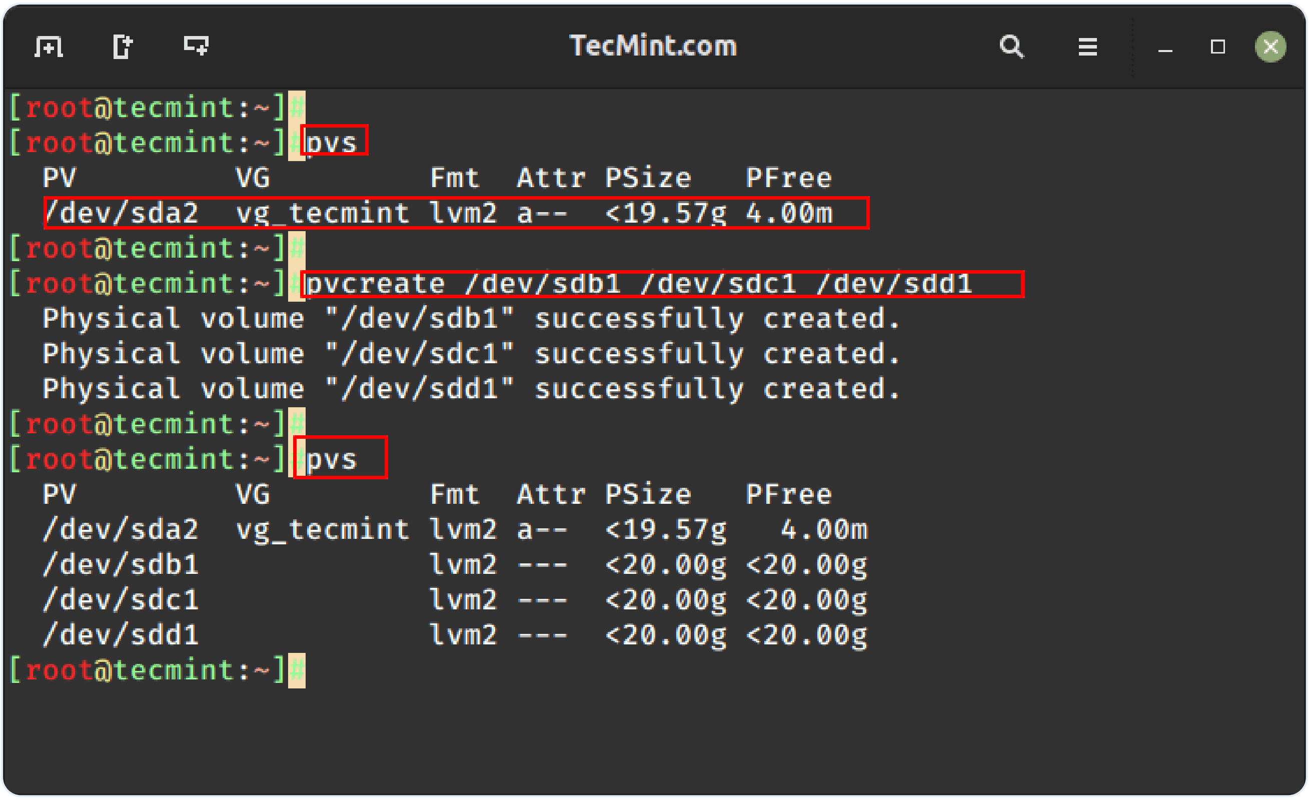
## Create LVM Physical Volume

8. Now, it's time to create **Physical Volumes** using all 3 disks. Here, I have listed the physical disks using the '**pvs**' command, and now only one default PV is listed.

```
# pvs
```

9. Then create the new physical disks and confirm the newly created physical disks.

```
# pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1
# pvs
```



```
[root@tecmint:~]#
[root@tecmint:~]#pvs
  PV         VG         Fmt  Attr PSize   PFree
  /dev/sda2  vg_tecmint lvm2 a--  <19.57g 4.00m
[root@tecmint:~]#
[root@tecmint:~]#pvcreate /dev/sdb1 /dev/sdc1 /dev/sdd1
  Physical volume "/dev/sdb1" successfully created.
  Physical volume "/dev/sdc1" successfully created.
  Physical volume "/dev/sdd1" successfully created.
[root@tecmint:~]#
[root@tecmint:~]#pvs
  PV         VG         Fmt  Attr PSize   PFree
  /dev/sda2  vg_tecmint lvm2 a--  <19.57g   4.00m
  /dev/sdb1             lvm2 ---  <20.00g <20.00g
  /dev/sdc1             lvm2 ---  <20.00g <20.00g
  /dev/sdd1             lvm2 ---  <20.00g <20.00g
[root@tecmint:~]#
```

Create LVM Physical Volumes

## Creating LVM Volume Groups

10. Create a Volume Group named **tecmint_add_vg** using the available free PV and a PE size of 32. To display the current volume groups, we can see that there is one volume group with 1 PV in use.

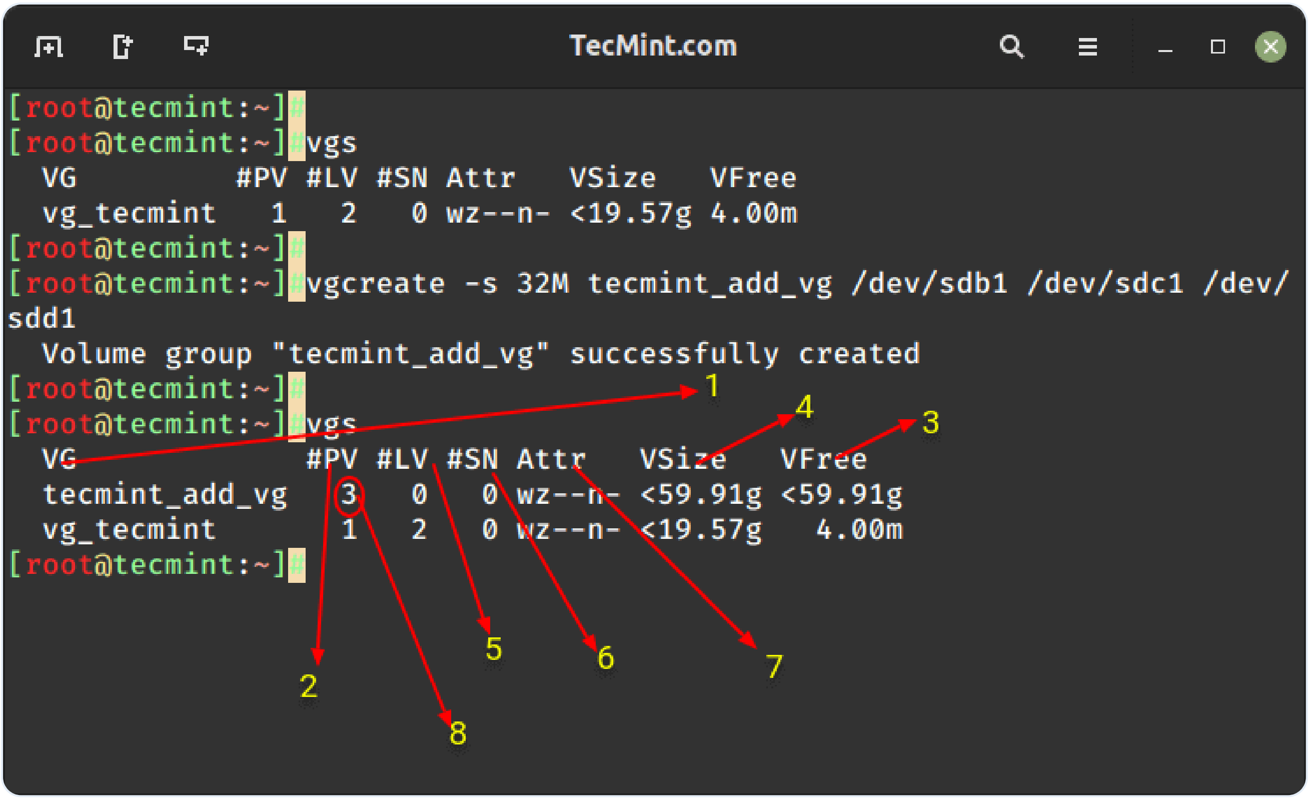
```
# vgs
```

11. This will create the volume group named **tecmint_add_vg** using a **32MB** PE size and the 3 physical volumes we created in the last steps.

```
# vgcreate -s 32M tecmint_add_vg /dev/sdb1 /dev/sdc1 /dev/sdd1
```

12. Next, verify the volume group by running the **vgs** command again.

```
# vgs
```



```
[root@tecmint:~]#
[root@tecmint:~]#vgs
  VG         #PV #LV #SN Attr   VSize   VFree
  vg_tecmint   1   2   0 wz--n- <19.57g 4.00m
[root@tecmint:~]#
[root@tecmint:~]#vgcreate -s 32M tecmint_add_vg /dev/sdb1 /dev/sdc1 /dev/
sdd1
  Volume group "tecmint_add_vg" successfully created
[root@tecmint:~]#
[root@tecmint:~]#vgs
  VG             #PV #LV #SN Attr   VSize   VFree
  tecmint_add_vg   3   0   0 wz--n- <59.91g <59.91g
  vg_tecmint       1   2   0 wz--n- <19.57g   4.00m
[root@tecmint:~]#
```

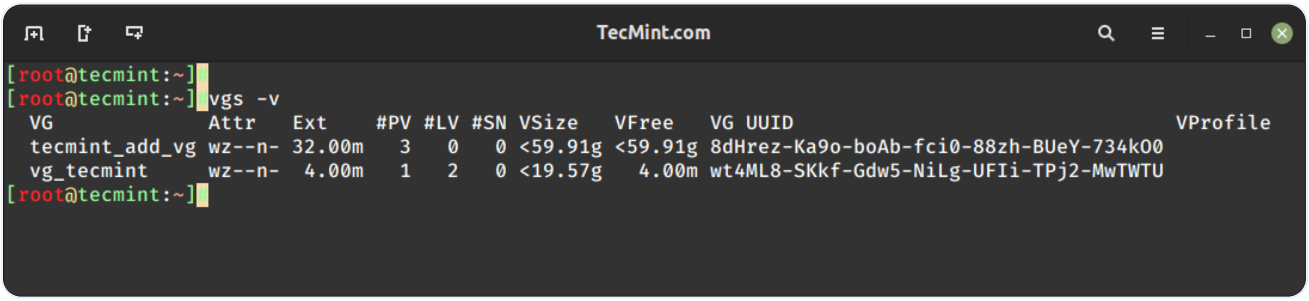
Confirm LVM Volume Groups

Understanding vgs command output:

- Volume Group name.
- Physical Volumes used in this Volume Group.
- Shows free space available in this volume group.
- Total Size of the Volume Group.
- Logical Volumes inside this volume group, Here we have not yet created so there is 0.
- SN = Number of Snapshots the volume group contains. (Later we can create a snapshot).
- Status of the Volume group as Writeable, readable, resizeable, exported, partial, and clustered, Here it is wz–n- that means w = Writable, z = resizeable.
- Number of Physical Volume (PV) used in this Volume Group.

13. To display more information about the volume group use the command.

```
# vgs -v
```

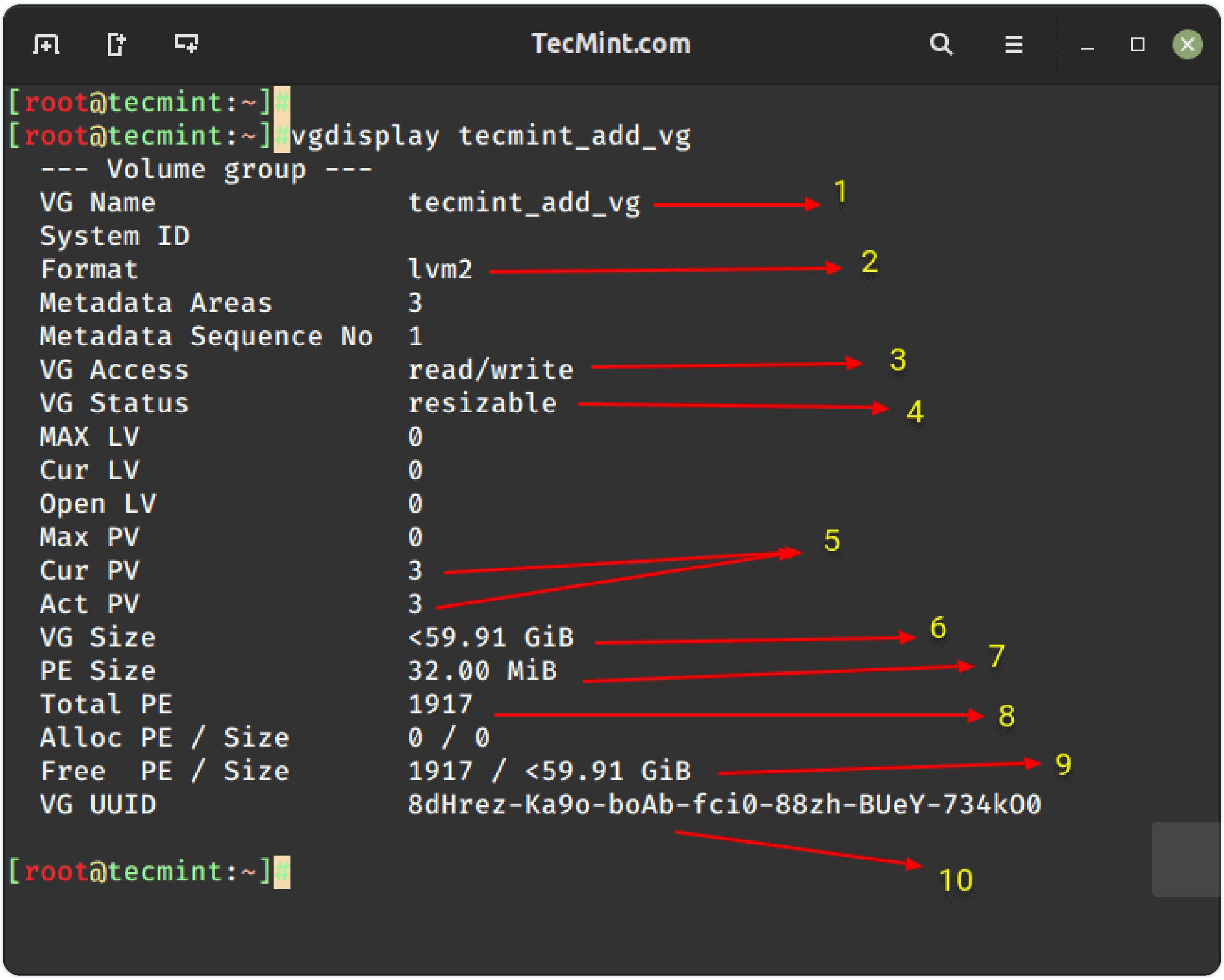


```
[root@tecmint:~]#
[root@tecmint:~]#vgs -v
  VG             Attr   Ext     #PV #LV #SN VSize   VFree   VG UUID                                VProfile
  tecmint_add_vg wz--n- 32.00m    3   0   0 <59.91g <59.91g 8dHrez-Ka9o-boAb-fci0-88zh-BUeY-734kO0
  vg_tecmint     wz--n-  4.00m    1   2   0 <19.57g   4.00m wt4ML8-SKkf-Gdw5-NiLg-UFIi-TPj2-MwTWTU
[root@tecmint:~]#
```

View LVM Volume Groups Info

**14.** To get more information about newly created volume groups, run the following command.

```
# vgdisplay tecmint_add_vg
```



```
[root@tecmint:~]#
[root@tecmint:~]#vgdisplay tecmint_add_vg
  --- Volume group ---
  VG Name               tecmint_add_vg
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               <59.91 GiB
  PE Size               32.00 MiB
  Total PE              1917
  Alloc PE / Size       0 / 0
  Free  PE / Size       1917 / <59.91 GiB
  VG UUID               8dHrez-Ka9o-boAb-fci0-88zh-BUeY-734kO0

[root@tecmint:~]#
```

List LVM Volume Groups

Here, is the description of each parameter shown in the above screenshot.

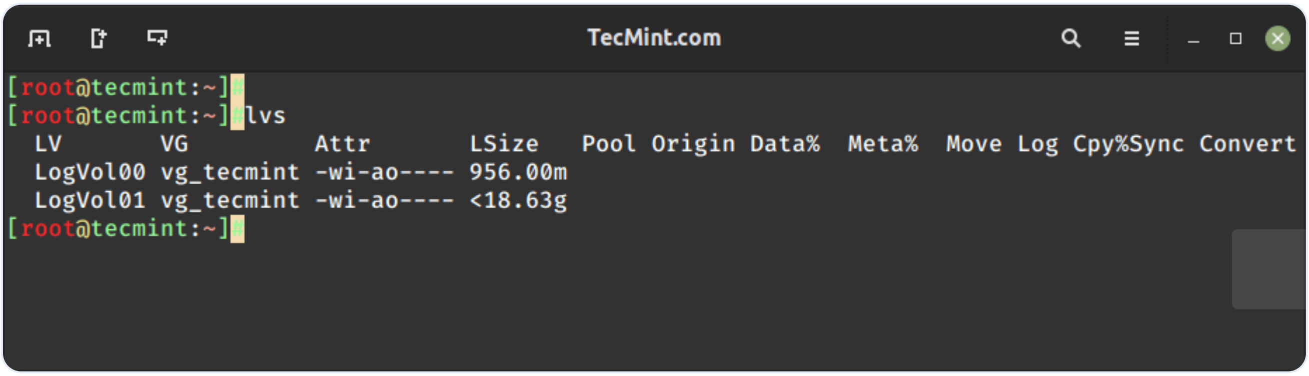
- Volume group name
- LVM Architecture used.
- It can be read and write state, ready to use.
- This volume group can be resizeable.
- No Physical disk was used and they are active.
- Volume Group total size.
- A Single PE size was 32 here.
- Total number of PE available in this volume group.
- Currently, we have not created any LV inside this VG so it's totally free.
- UUID of this volume group.

## Creating LVM Logical Volumes

15. Now, create 3 Logical Volumes named **tecmint_documents**, **tecmint_manager**, and **tecmint_public**. Here, we will demonstrate how to create Logical Volumes using both PE size and GB size.

First, list the current Logical Volumes using the following command..

```
# lvs
```

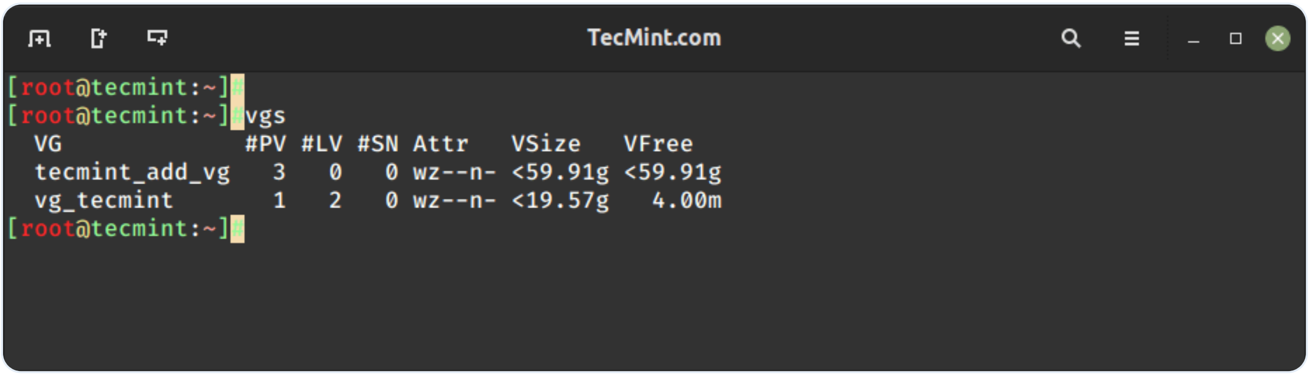


```
[root@tecmint:~]#
[root@tecmint:~]#lvs
  LV       VG         Attr       LSize    Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  LogVol00 vg_tecmint -wi-ao---- 956.00m
  LogVol01 vg_tecmint -wi-ao---- <18.63g
[root@tecmint:~]#
```

List LVM Logical Volumes

16. These Logical Volumes are in the **vg_tecmint** Volume Group. To see how much free space is available to create logical volumes, list the Volume Group and available Physical Volumes using the '**vgs**' command.

```
# vgs
```



```
[root@tecmint:~]#
[root@tecmint:~]#vgs
  VG             #PV #LV #SN Attr   VSize   VFree
  tecmint_add_vg   3   0   0 wz--n- <59.91g <59.91g
  vg_tecmint       1   2   0 wz--n- <19.57g   4.00m
[root@tecmint:~]#
```

List Volume Groups

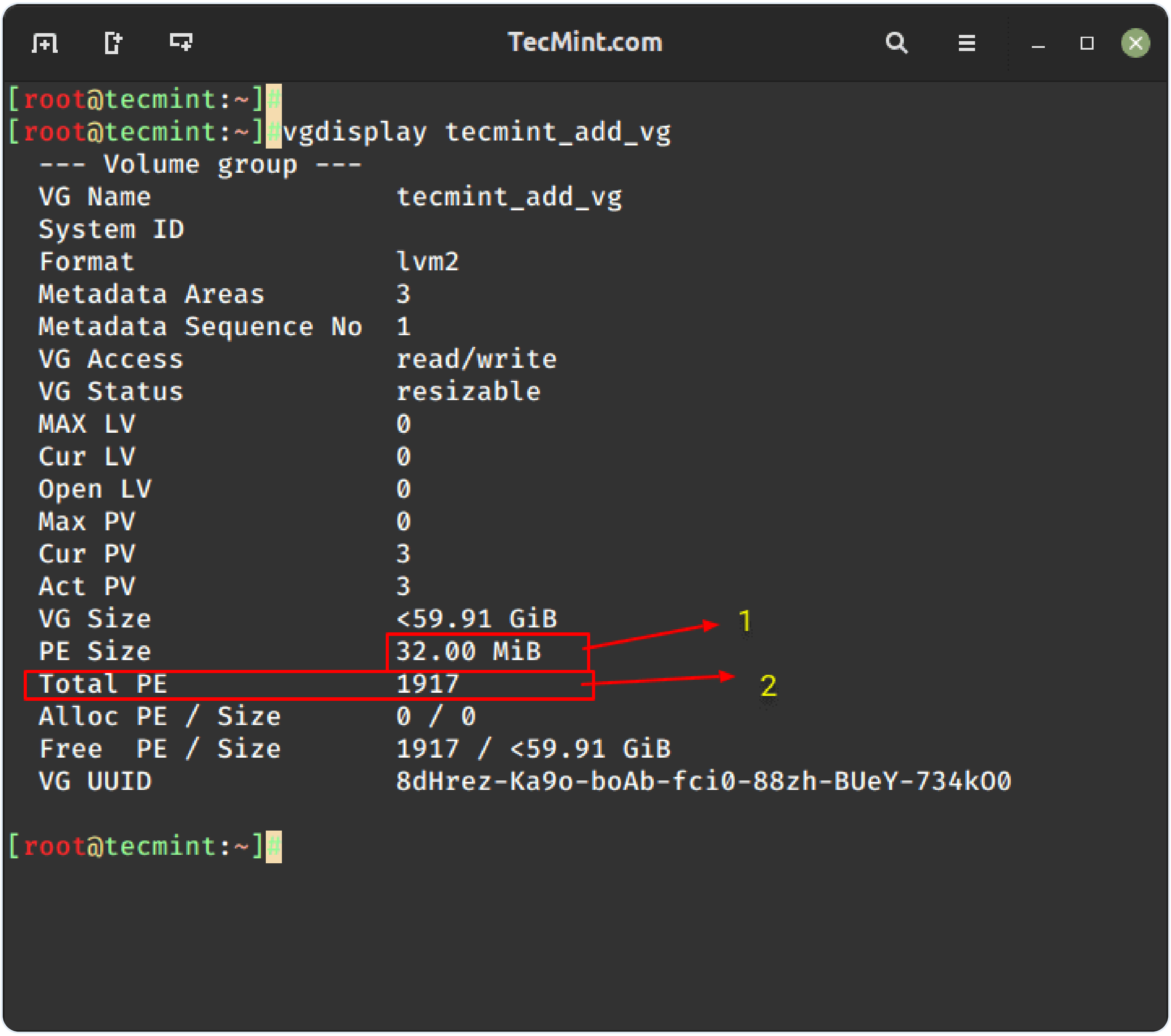
The volume group size is almost 60GB, and it is unused, so we can create LVs in it. Let us divide the volume group into equal sizes to create 3 Logical Volumes. That means 60GB/3 = 20GB. Each Logical Volume will be 20GB in size after creation.

## Method 1: Creating Logical Volumes using PE Size

First, let us create Logical Volumes using the Physical Extent (PE) size. We need to know the default PE size assigned to this Volume Group and the total available PEs to create new Logical Volumes.

Run the following command to get this information.

```
# vgdisplay tecmint_add_vg
```



```
[root@tecmint:~]#
[root@tecmint:~]#vgdisplay tecmint_add_vg
  --- Volume group ---
  VG Name               tecmint_add_vg
  System ID
  Format                lvm2
  Metadata Areas        3
  Metadata Sequence No  1
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                0
  Open LV               0
  Max PV                0
  Cur PV                3
  Act PV                3
  VG Size               <59.91 GiB
  PE Size               32.00 MiB
  Total PE              1917
  Alloc PE / Size       0 / 0
  Free  PE / Size       1917 / <59.91 GiB
  VG UUID               8dHrez-Ka9o-boAb-fci0-88zh-BUeY-734kO0

[root@tecmint:~]#
```

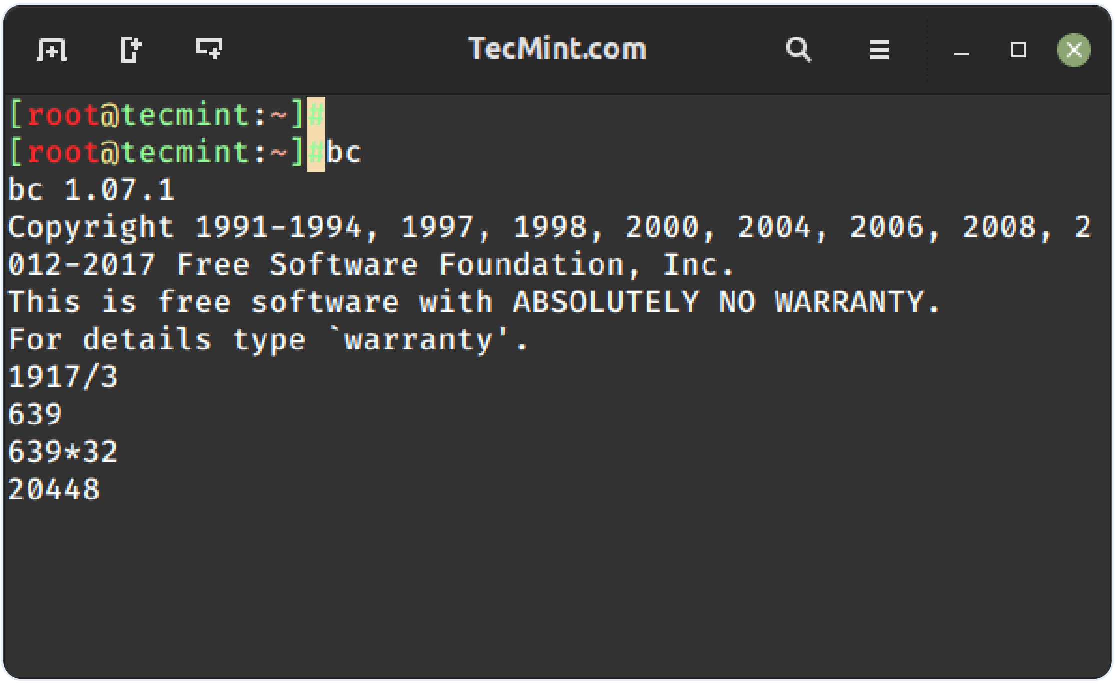
Create a New Logical Volume

- The default PE Assigned for this VG is 32MB, Here Single PE size will be 32MB.
- Total Available PE is 1917.

Just do and see a little Calculation using the bc command.

```
# bc
```

```
1917PE/3 = 639 PE.
639 PE x 32MB = 20448 --> 20GB
```



```
[root@tecmint:~]#
[root@tecmint:~]#bc
bc 1.07.1
Copyright 1991-1994, 1997, 1998, 2000, 2004, 2006, 2008, 2
012-2017 Free Software Foundation, Inc.
This is free software with ABSOLUTELY NO WARRANTY.
For details type `warranty'.
1917/3
639
639*32
20448
```

Calculate Disk Space

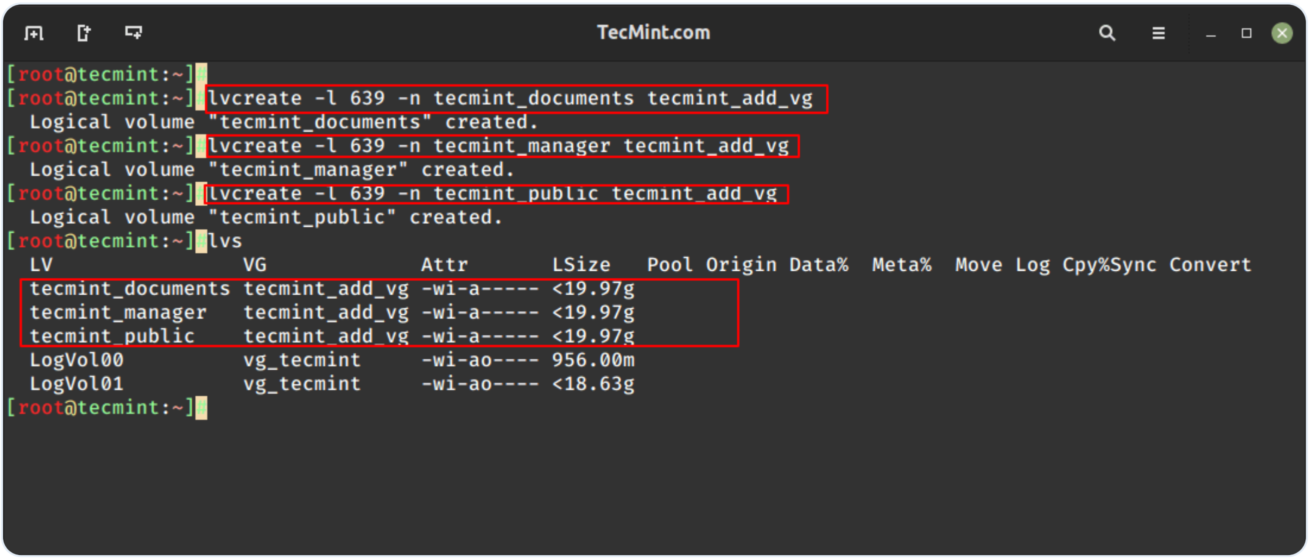
Press CRTL+D to exit from bc.

Let us now create 3 Logical Volumes using 639 PE's. Here `-l` used to extend the size and `-n` to assign a logical volume name.

```
# lvcreate -l 639 -n tecmint_documents tecmint_add_vg
# lvcreate -l 639 -n tecmint_manager tecmint_add_vg
# lvcreate -l 639 -n tecmint_public tecmint_add_vg
```

List the created Logical Volumes using lvs command.

```
# lvs
```



```
[root@tecmint:~]
[root@tecmint:~]lvcreate -l 639 -n tecmint_documents tecmint_add_vg
  Logical volume "tecmint_documents" created.
[root@tecmint:~]lvcreate -l 639 -n tecmint_manager tecmint_add_vg
  Logical volume "tecmint_manager" created.
[root@tecmint:~]lvcreate -l 639 -n tecmint_public tecmint_add_vg
  Logical volume "tecmint_public" created.
[root@tecmint:~]lvs
  LV                VG             Attr       LSize   Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert
  tecmint_documents tecmint_add_vg -wi-a----- <19.97g
  tecmint_manager   tecmint_add_vg -wi-a----- <19.97g
  tecmint_public    tecmint_add_vg -wi-a----- <19.97g
  LogVol00          vg_tecmint     -wi-ao---- 956.00m
  LogVol01          vg_tecmint     -wi-ao---- <18.63g
[root@tecmint:~]
```

List Created Logical Volumes

## Method 2: Creating Logical Volumes using GB Size

While creating Logical Volume using GB size we cannot get the exact size. So, the better way is to create using extend.

```
# lvcreate -L 20G -n tecmint_documents tecmint_add_vg
# lvcreate -L 20G -n tecmint_manager tecmint_add_vg
# lvcreate -L 20G -n tecmint_public tecmint_add_vg
# lvcreate -L 20G -n tecmint_public tecmint_add_vg
```

List the Created logical Volumes using lvs command.

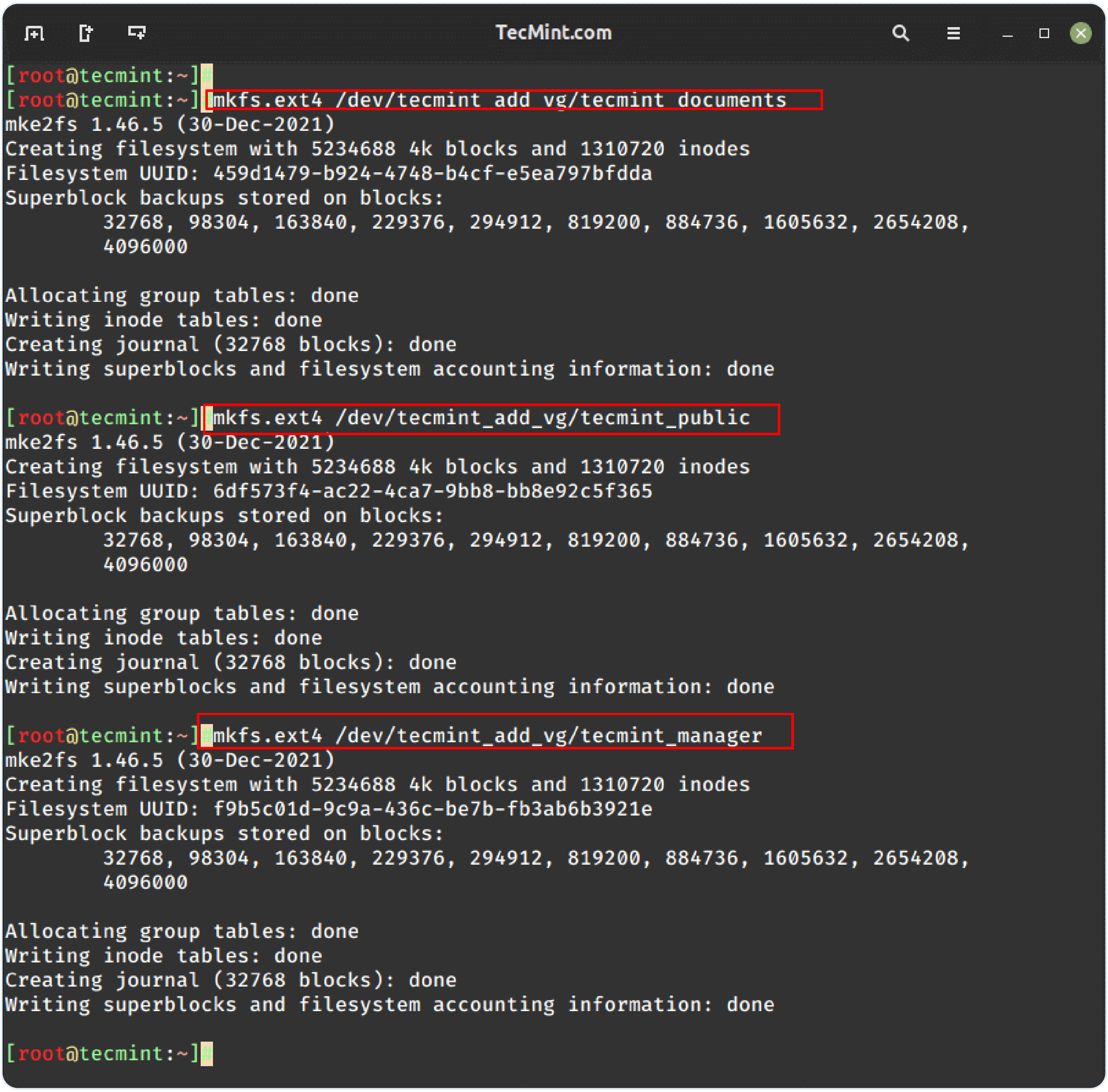
```
# lvs
```

Here, we can see while creating the 3rd LV we can't Round-up to 20GB, it is because of small changes in size, but this issue will be ignored while creating LV using Extend size.

## Creating File System

17. For using the logical volumes we need to format. Here I am using the ext4 file-system to create the volumes and going to mount them under /mnt/.

```
# mkfs.ext4 /dev/tecmint_add_vg/tecmint_documents
# mkfs.ext4 /dev/tecmint_add_vg/tecmint_public
# mkfs.ext4 /dev/tecmint_add_vg/tecmint_manager
```



```
[root@tecmint:~]
[root@tecmint:~]mkfs.ext4 /dev/tecmint_add_vg/tecmint_documents
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5234688 4k blocks and 1310720 inodes
Filesystem UUID: 459d1479-b924-4748-b4cf-e5ea797bfdda
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@tecmint:~]mkfs.ext4 /dev/tecmint_add_vg/tecmint_public
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5234688 4k blocks and 1310720 inodes
Filesystem UUID: 6df573f4-ac22-4ca7-9bb8-bb8e92c5f365
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@tecmint:~]mkfs.ext4 /dev/tecmint_add_vg/tecmint_manager
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5234688 4k blocks and 1310720 inodes
Filesystem UUID: f9b5c01d-9c9a-436c-be7b-fb3ab6b3921e
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

[root@tecmint:~]
```

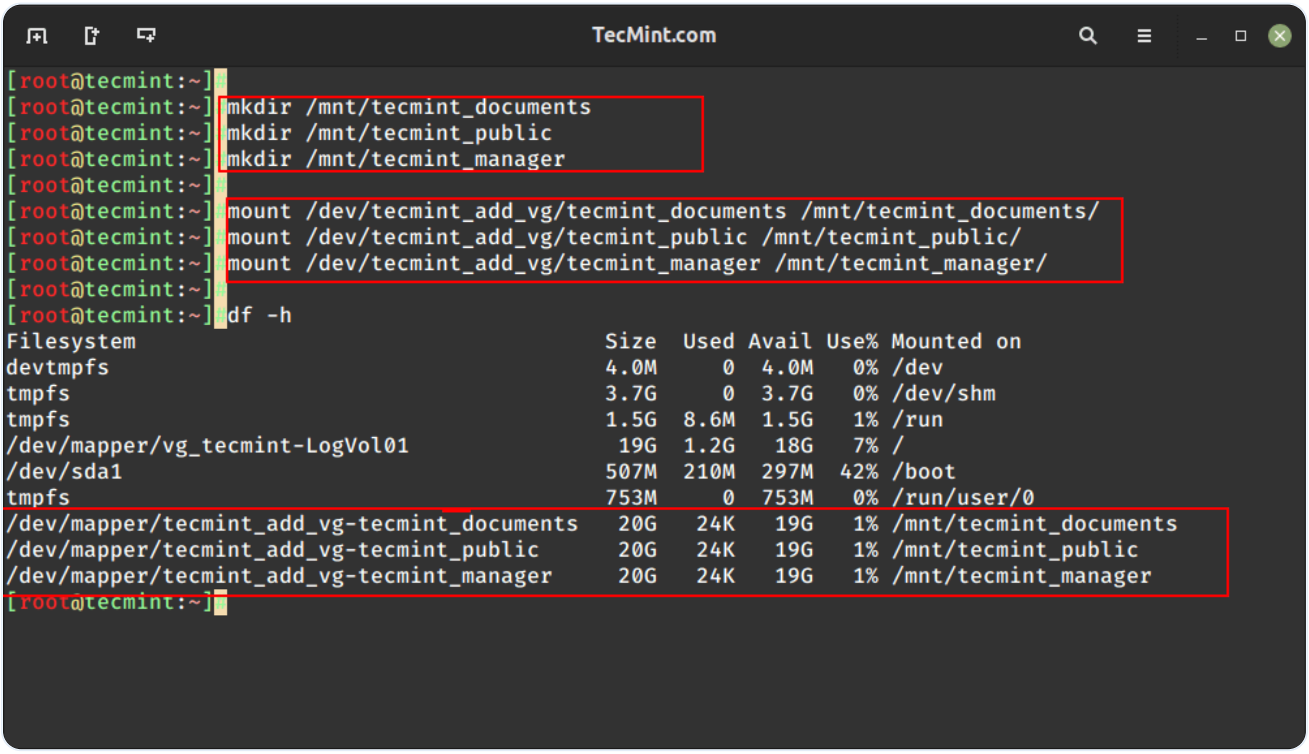
Create Ext4 File System

18. Let us create directories in /mnt and mount the Logical volumes that we have created file-system.

```
# mount /dev/tecmint_add_vg/tecmint_documents /mnt/tecmint_documents/
# mount /dev/tecmint_add_vg/tecmint_public /mnt/tecmint_public/
# mount /dev/tecmint_add_vg/tecmint_manager /mnt/tecmint_manager/
```

19. List and confirm the Mount point using.

```
# df -h
```



```
[root@tecmint:~]#
[root@tecmint:~]#mkdir /mnt/tecmint_documents
[root@tecmint:~]#mkdir /mnt/tecmint_public
[root@tecmint:~]#mkdir /mnt/tecmint_manager
[root@tecmint:~]#
[root@tecmint:~]#mount /dev/tecmint_add_vg/tecmint_documents /mnt/tecmint_documents/
[root@tecmint:~]#mount /dev/tecmint_add_vg/tecmint_public /mnt/tecmint_public/
[root@tecmint:~]#mount /dev/tecmint_add_vg/tecmint_manager /mnt/tecmint_manager/
[root@tecmint:~]#
[root@tecmint:~]#df -h
Filesystem                                   Size  Used Avail Use% Mounted on
devtmpfs                                     4.0M     0  4.0M   0% /dev
tmpfs                                        3.7G     0  3.7G   0% /dev/shm
tmpfs                                        1.5G  8.6M  1.5G   1% /run
/dev/mapper/vg_tecmint-LogVol01               19G  1.2G   18G   7% /
/dev/sda1                                    507M  210M  297M  42% /boot
tmpfs                                        753M     0  753M   0% /run/user/0
/dev/mapper/tecmint_add_vg-tecmint_documents   20G   24K   19G   1% /mnt/tecmint_documents
/dev/mapper/tecmint_add_vg-tecmint_public      20G   24K   19G   1% /mnt/tecmint_public
/dev/mapper/tecmint_add_vg-tecmint_manager     20G   24K   19G   1% /mnt/tecmint_manager
[root@tecmint:~]#
```

Mount Logical Volumes

## Permanent Mounting of Logical Volumes

20. It’s now temporarily mounted, for permanent mount, we need to add the entry in fstab, for that let us get the mount entry from mtab using

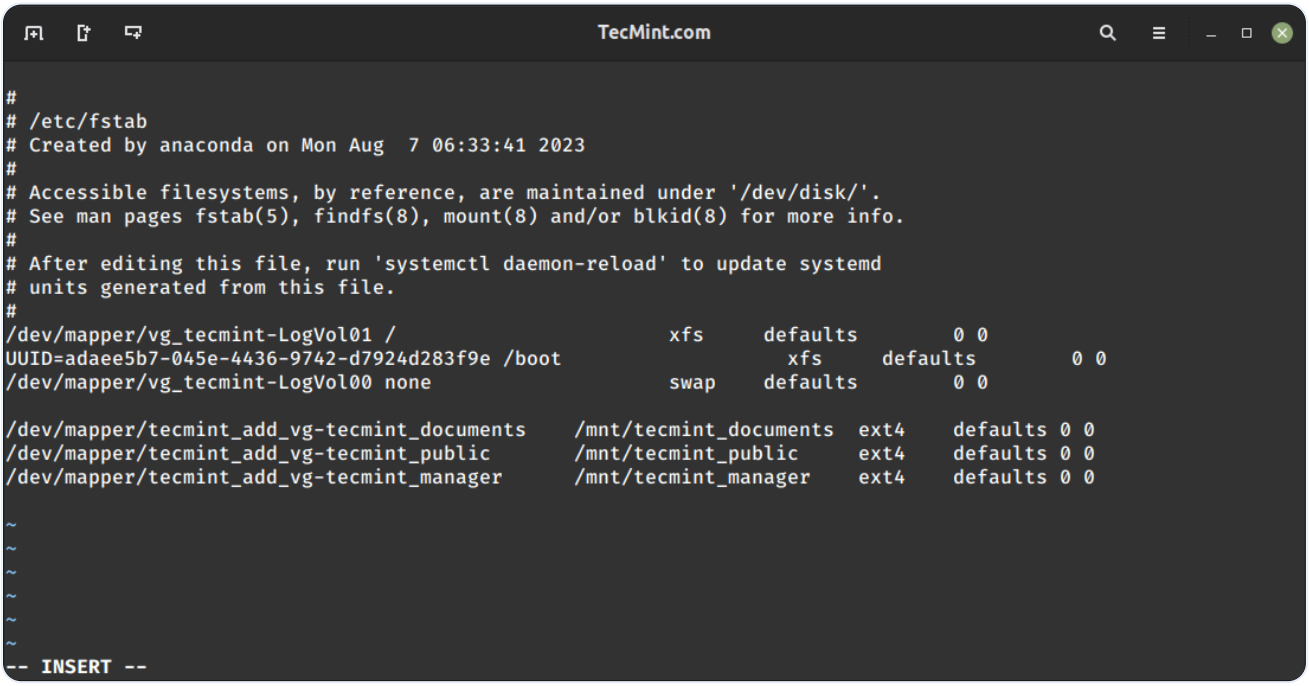
```
# cat /etc/mtab
```

21. We need to make slight changes in the fstab entry while entering the mount entry contents copies from mtab, we need to change the rw to defaults

```
# vi /etc/fstab
```

Our fstab entries should look similar to the below sample.

```
/dev/mapper/tecmint_add_vg-tecmint_documents    /mnt/tecmint_documents   ext
/dev/mapper/tecmint_add_vg-tecmint_public       /mnt/tecmint_public      ext
/dev/mapper/tecmint_add_vg-tecmint_manager      /mnt/tecmint_manager     ext
```

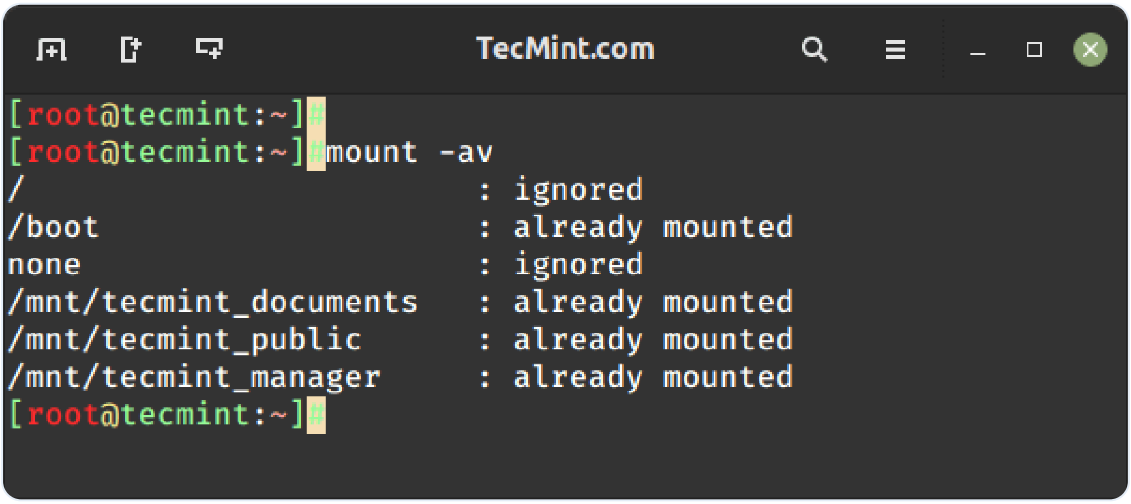


```
#
# /etc/fstab
# Created by anaconda on Mon Aug  7 06:33:41 2023
#
# Accessible filesystems, by reference, are maintained under '/dev/disk/'.
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/vg_tecmint-LogVol01 /                       xfs     defaults        0 0
UUID=adaee5b7-045e-4436-9742-d7924d283f9e /boot                   xfs     defaults        0 0
/dev/mapper/vg_tecmint-LogVol00 none                    swap    defaults        0 0

/dev/mapper/tecmint_add_vg-tecmint_documents    /mnt/tecmint_documents  ext4    defaults 0 0
/dev/mapper/tecmint_add_vg-tecmint_public       /mnt/tecmint_public     ext4    defaults 0 0
/dev/mapper/tecmint_add_vg-tecmint_manager      /mnt/tecmint_manager    ext4    defaults 0 0

~
~
~
~
~
~
-- INSERT --
```

Permanent Mount Logical Volumes

22. Finally, run the command **mount -a** to check for the fstab entry before restarting.

```
# mount -av
```



```
[root@tecmint:~]#
[root@tecmint:~]#mount -av
/                        : ignored
/boot                    : already mounted
none                     : ignored
/mnt/tecmint_documents   : already mounted
/mnt/tecmint_public      : already mounted
/mnt/tecmint_manager     : already mounted
[root@tecmint:~]#
```

Confirm Mount Points

Here we have seen how to set up flexible storage with logical volumes by using physical disk to physical volume, physical volume to the volume group, and volume group to logical volumes.

In my upcoming future articles, I will see how to extend the volume group, and logical volumes, reduce logical volume, take snapshots, and restore from snapshots. Till then stay updated to TecMint for more such awesome articles.

PREVIOUS ARTICLE:

How to Setup SSH Passwordless Login in Linux [3 Easy Steps]

NEXT ARTICLE:

What are Ext2, Ext3 & Ext4? How to Create and Convert Linux File Systems

Ravi Saive

I am an experienced GNU/Linux expert and a full-stack software developer with over a decade in the field of Linux and Open Source technologies

*Each tutorial at TecMint is created by a team of experienced Linux system administrators* so that it meets our high-quality *standards*.

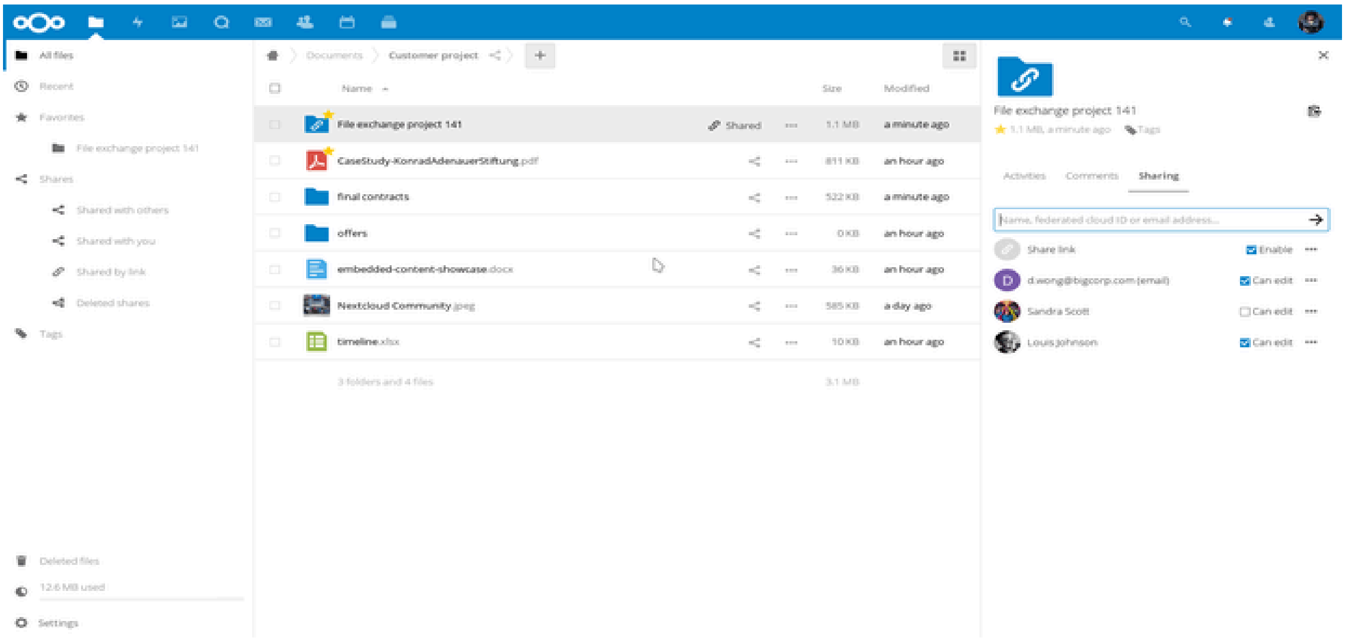
Join the TecMint Weekly Newsletter (More Than 156,129 Linux Enthusiasts Have Subscribed)

Was this article helpful? Please add a comment or buy me a coffee to show your appreciation.

## Related Posts

### How to Install OwnCloud to Create Own Cloud Storage in Linux

### How to Install OwnCloud on Ubuntu 18.04



| Name | | | Size | Modified |
|---|---|---|---|---|
| File exchange project 141 | Shared | ··· | 1.1 MB | a minute ago |
| CaseStudy-KonradAdenauerStiftung.pdf | | ··· | 811 KB | an hour ago |
| final contracts | | ··· | 522 KB | a minute ago |
| offers | | ··· | 0 KB | an hour ago |
| embedded-content-showcase.docx | | ··· | 36 KB | an hour ago |
| Nextcloud Community.jpeg | | ··· | 585 KB | a day ago |
| timeline.xlsx | | ··· | 10 KB | an hour ago |
| 3 folders and 4 files | | | 3.1 MB | |

File exchange project 141

1.1 MB, a minute ago Tags

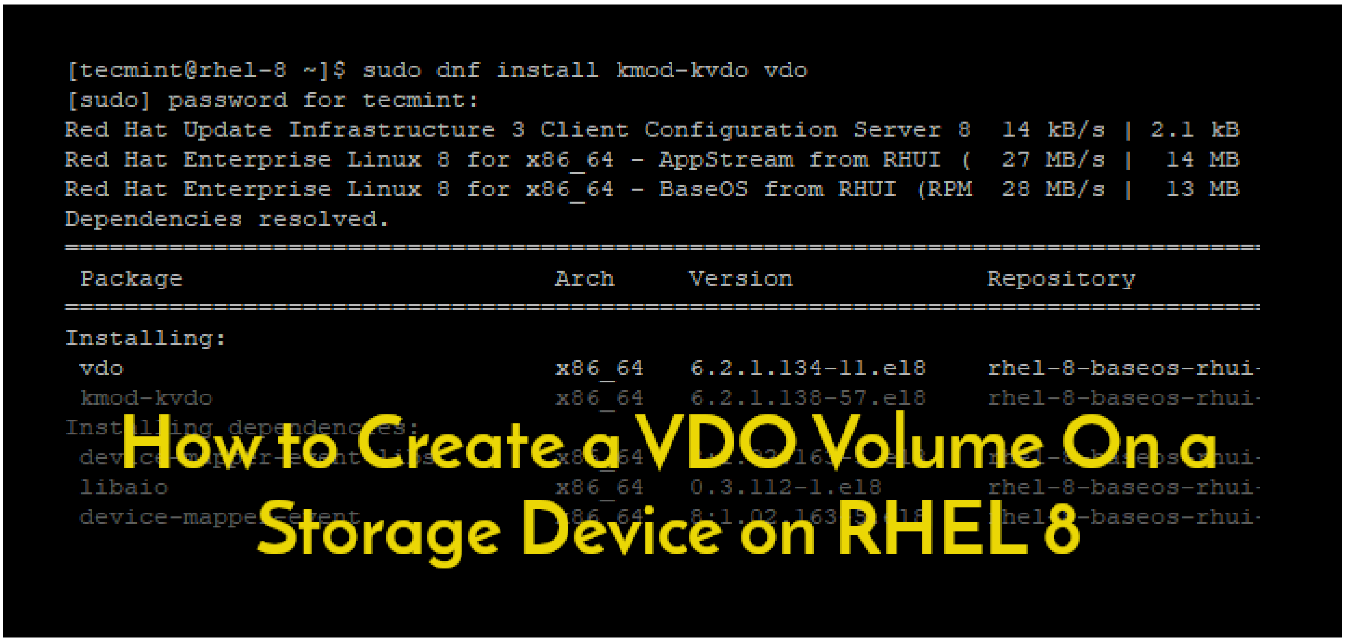
Name, federated cloud ID or email address...

| | | |
|---|---|---|
| Share link | ☑ Enable | ··· |
| d.wong@bigcorp.com (email) | ☑ Can edit | ··· |
| Sandra Scott | □ Can edit | ··· |
| Louis Johnson | ☑ Can edit | ··· |

## How to Install NextCloud on CentOS 8

## How to Install OwnCloud in Debian 10

How to Install OwnCloud on CentOS 8



```
[tecmint@rhel-8 ~]$ sudo dnf install kmod-kvdo vdo
[sudo] password for tecmint:
Red Hat Update Infrastructure 3 Client Configuration Server 8  14 kB/s | 2.1 kB
Red Hat Enterprise Linux 8 for x86_64 - AppStream from RHUI (  27 MB/s |  14 MB
Red Hat Enterprise Linux 8 for x86_64 - BaseOS from RHUI (RPM  28 MB/s |  13 MB
Dependencies resolved.
================================================================================
 Package                        Arch      Version              Repository
================================================================================
Installing:
 vdo                            x86_64    6.2.1.134-11.el8     rhel-8-baseos-rhui-
 kmod-kvdo                      x86_64    6.2.1.138-57.el8     rhel-8-baseos-rhui-
Installing dependencies:
 device-mapper-event-libs       x86_64    8:1.02.163-5.el8     rhel-8-baseos-rhui-
 libaio                         x86_64    0.3.112-1.el8        rhel-8-baseos-rhui-
 device-mapper-event            x86_64    8:1.02.163-5.el8     rhel-8-baseos-rhui-
```

How to Create a VDO Volume On a Storage Device on RHEL 8

## 24 Comments

Leave a Reply

Majid Khatib Shahidi

December 1, 2019 at 1:25 am

Thanks, very helpful article…

Reply

**Madhav**

June 10, 2019 at 9:42 am

What is the purpose of `/etc/mtab` file? what is difference between **fstab** and mtab?

Reply

**Priyashree**

February 4, 2018 at 11:17 pm

Can we use a full disk for LVM without partitioning or is it necessary to partition the disk for LVM

Reply

**Bobin Lonston**

February 5, 2018 at 12:48 pm

@ Priyashree,

We can use full disk without partitioning.

Reply

**Imad**

December 12, 2017 at 9:19 pm

Thanks, please add more information about why we should choose a bigger or smaller PE size.

Reply

**Bobin Lonston**

January 2, 2018 at 11:50 pm

@Imad,

Bigger PE size will be good, a large number of extents will slow down the tools but it won't affect any I/O performance. Large size file system large value of PE is good by default it will select 4 MB.

Regards,

Reply

**swati**

September 15, 2017 at 7:05 pm

Hello,

Good article.

I have 1 query. Can we mount more than 1 directory in same logical volume?

For example: /mnt_techmint_documents , /mnt_techmint_public on same logical volume (/dev/mapper/techmint_add_vg_techmint_documents)

Reply

**Babin Lonston**

September 19, 2017 at 10:57 pm

@Swati,

No, You can’t mount a single Logical Volume to multiple directories.

Thanks & Regards,
Babin Lonston

Reply

**Macfa**

August 8, 2017 at 8:53 pm

Thx, It’s helpful lol

Reply

**Raghu**

November 2, 2016 at 10:25 pm

please send interview questions and answers to my mail…

Reply

**Harivansh**

October 14, 2016 at 3:56 pm

How to check how much space is utilize via LV in Sectors and tracks ?
if am removing volume from vg there output is /dev/vdb2 is Still in use ?

Reply

**mohammad**

September 11, 2016 at 2:54 pm

hello.
In No 10 you told that :
“List and see how much free spaces are there to create logical volumes using pvs command”
But its picture is output of vga command

please correct the sentens and picture.

Thank you.

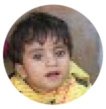
Reply

**Nazri**

February 29, 2016 at 12:14 pm

Thank you very much for the details. Now it's easy for me to understand LVM concept.

Reply



**Umesh**

November 19, 2015 at 1:26 pm

Very Good… Keep it UP

Reply

**Dmitry**

August 27, 2015 at 7:19 pm

Hi

Please tell – should I use system-storage-manager (ssm) for Centos 7?

Thank you

Reply

**Babin Lonston**

October 30, 2015 at 4:04 pm

@Dmitry

System Storage manager is to manage your lvm, btrfs, dm, encrypted vls and more, If you are familiar with LVM no need of SSM

Reply

**Mario**

September 2, 2014 at 6:27 am

Hi Babin,

good post.
I have some doubts when trying to repeat your steps. I'm playing with oVirt 3.4 Glusterized, and CentOS 6.5 .
In my two physical hosts, added a new FC LUN with 200 GB.

How do I use it?
How do I configure multipath? I see two paths to the same LUN.

```
[root @ CentOS-H1 ~] # lsscsi
[0: 0: 0: 0] disk HP LOGICAL VOLUME 3.55 / dev / sda
[0: 1: 0: 0] B320i storage HP 3:55 –
[5: 0: 0: 0] cd / dvd hp DVDRAM GT80N EA02 / dev / sr0
[7: 0: 0: 0] disk VRAID 0533 DGC / dev / sdb
[7: 0: 1: 0] disk VRAID 0533 DGC / dev / sdd
[8: 0: 0: 0] disk HP iLO LUN 0 2:09 00 Media / dev / sdc
```

```
[root @ CentOS-H1 ~] # multipath -ll
36006016043a036009f784f47cf0ce411 dm-2 DGC, VRAID
size = 200G features = ‘1 queue_if_no_path’ hwhandler = “1 emc ‘wp = rw
| – + – Policy = “round-robin 0 ‘prio = 1 status = active
| `- 7: 0: 1: 0 sdd 8:48 ready active running
```

```
`- + – Policy =” round-robin 0 ‘prio = 0 status = enabled
  `- 7: 0: 0: 0 SDB enable ready running 8:16
```

```
[root@CentOS-H1 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/mapper/36006016043a036009f784f47cf0ce411p1 lvm2 a– 200.00g 200.00g
/dev/sda2 vg_centosh1 lvm2 a– 558.24g 0
```

I’m not a Linux guy, but wanted to test the oVirt 3.4, Glusterized.

Can you give me some tips please?

Thanks in advance.

Reply

**Babin Lonston**

August 5, 2015 at 8:54 pm

good post.

// Thanks buddy //

I have some doubts when trying to repeat your steps. I’m playing with oVirt 3.4 Glusterized, and CentOS 6.5 .
In my two physical hosts, added a new FC LUN with 200 GB.

How do I use it?
How do I configure multipath? I see two paths to the same LUN.

// If you get the LUN in two paths mean already picked by multipath. If you need to assign a alias you can edit the file under /etc/multipath.conf and give a alias find the LUN id under /dev/disks/by-id/ then give the LUN id in multipath.conf. Even we can change the permission to some other users. //

```
[root @ CentOS-H1 ~] # lsscsi
[0: 0: 0: 0] disk HP LOGICAL VOLUME 3.55 / dev / sda
[0: 1: 0: 0] B320i storage HP 3:55 –
[5: 0: 0: 0] cd / dvd hp DVDRAM GT80N EA02 / dev / sr0
[7: 0: 0: 0] disk VRAID 0533 DGC / dev / sdb
[7: 0: 1: 0] disk VRAID 0533 DGC / dev / sdd
[8: 0: 0: 0] disk HP iLO LUN 0 2:09 00 Media / dev / sdc
```

```
[root @ CentOS-H1 ~] # multipath -ll
36006016043a036009f784f47cf0ce411 dm-2 DGC, VRAID
size = 200G features = ‘1 queue_if_no_path’ hwhandler = “1 emc ‘wp = rw
| – + – Policy = “round-robin 0 ‘prio = 1 status = active
| `- 7: 0: 1: 0 sdd 8:48 ready active running
`- + – Policy =” round-robin 0 ‘prio = 0 status = enabled
`- 7: 0: 0: 0 SDB enable ready running 8:16
```

// babin 36006016043a036009f784f47cf0ce411 dm-2 DGC, VRAID // In this line you will get the alias which you have assigned in multipath.conf

// dm-2 // this is the device we going to format and create a mount point

Then to create a mount point we need to do the below steps.

```
# pvcreate /dev/mapper/dm-2
# vgcreate volgrp /dev/mapper/dm-2
# lvcreate -L +200G -n logvol01 /dev/mapper/dm-2
```

then format it and mount it under any mount point

```
# mkfs.ext4 /dev/mapper/volgrp-logvol01
```

mount it

```
# mount /dev/mapper/volgrp-logvol01 /mnt/babin
```

//

```
[root@CentOS-H1 ~]# pvs
PV VG Fmt Attr PSize PFree
```

/dev/mapper/36006016043a036009f784f47cf0ce411p1 lvm2 a– 200.00g 200.00g
/dev/sda2 vg_centosh1 lvm2 a– 558.24g 0

Above step is fine but not using a alias and it will get confuse once you have more storage disks in same machine.

//

Welcome..

Reply

**JuanSys76**
August 3, 2014 at 12:13 am

excelente….

Reply

**Babin Lonston**
August 3, 2014 at 7:20 pm

@JuanSys76 Thanks dear, Follow us to get more on LVM

Reply

**Babin Lonston**
August 1, 2014 at 4:44 pm

Welcome bro, sorry for the typo will clear ..

Reply

**Oliver**

August 1, 2014 at 2:52 pm

Hi,

nice article, heres a little addition:

You want to use option “-c” of fdisk to make sure that there is a correct alignment (dividable with 2), without this option your partition starts at 63 which leads to a misaligned disk and reduced performance.

Greetings
Oliver

Reply

**Babin Lonston**

August 3, 2014 at 7:19 pm

@oliver yes the option -c used there already, please have a look.

Reply

**Medhansh**

July 31, 2014 at 10:57 pm

HI,

Article is good except some typos.

Keep posting such a good article.

Reply

## Got Something to Say? Join the Discussion...

*Thank you for taking the time to share your thoughts with us. We appreciate your decision to leave a comment and value your contribution to the discussion. It's important to note that we moderate all comments in accordance with our comment policy to ensure a respectful and constructive conversation.*

*Rest assured that your email address will remain private and will not be published or shared with anyone. We prioritize the privacy and security of our users.*

Name *

Email *

□ Save my name, email, and website in this browser for the next time I comment.

Post Comment

Search...

## Linux Commands and Tools

How to Use 'at' Command to Schedule a Task on Given or Later Time in Linux

How to Create a Systemd Service in Linux

How to Delete all Text in a File Using Vi/Vim Editor

How to Fix "Command 'pip3' not found" Error in Linux

How to Find Out Who is Using a File in Linux

How to Control Systemd Services on Remote Linux Server

## Linux Server Monitoring Tools

TCPflow – Analyze and Debug Network Traffic in Linux

Collectl: An Advanced All-in-One Performance Monitoring Tool for Linux

How to Add Linux Host to Nagios Monitoring Server Using NRPE Plugin

Setting Up Real-Time Monitoring with 'Ganglia' for Grids and Clusters of Linux Servers

Inxi – A Powerful Feature-Rich Commandline System Information Tool for Linux

How to Monitor System Usage, Outages and Troubleshoot Linux Servers – Part 9

## Learn Linux Tricks & Tips

How to Create a Password Protected ZIP File in Linux

How to Test Website Loading Speed in Linux Terminal

How to Disable SELinux Temporarily or Permanently

Progress – Show Percentage of Copied Data for (cp, mv, dd, tar) Commands

How to Check Bad Sectors or Bad Blocks on Hard Disk in Linux

Useful Commands to Create Commandline Chat Server and Remove Unwanted Packages in Linux

## Best Linux Tools

Top 7 Apps to Install for Your Nextcloud Instance

5 Best Open-Source Project Management Tools for Linux in 2024

5 Best Mathematical Equation and Formula Writing Tools for Linux

7 Best CCleaner Alternatives for Ubuntu

11 Best IP Address Management Tools for Linux Network

31 Best File Managers and Explorers [GUI + CLI] for Linux in 2024