
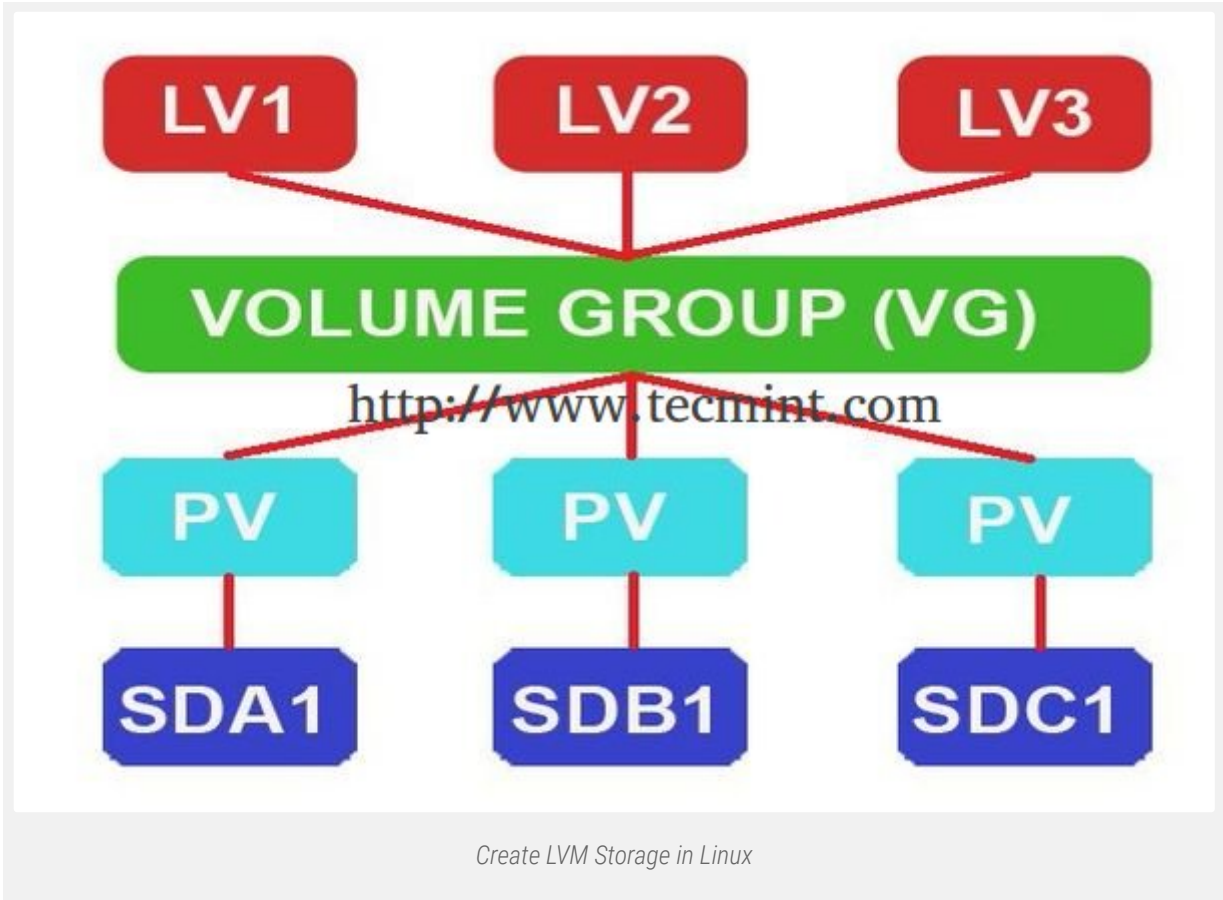


Setup Flexible Disk Storage with Logical Volume Management (LVM) in Linux – PART 1

by Babin Lonston | Published: July 30, 2014 | Last Updated: December 16, 2015

 Download Your Free eBooks NOW - [10 Free Linux eBooks for Administrators](#) | [4 Free Shell Scripting eBooks](#)

Logical Volume Management (LVM) makes it easier to manage disk space. If a file system needs more space, it can be added to its logical volumes from the free spaces in its volume group and the file system can be re-sized as we wish. If a disk starts to fail, replacement disk can be registered as a physical volume with the volume group and the logical volumes extents can be migrated to the new disk without data loss.



In a modern world every Server needs more space day by day for that we need to expand depending on our needs. Logical volumes can be use in RAID, SAN. A Physical Disk will be grouped to create a volume Group. Inside volume group we need to slice the space to create Logical volumes. While using logical volumes we can extend across multiple disks, logical volumes or reduce logical volumes in size with some commands without reformatting and re-partitioning the current disk. Volumes can stripes data across multiple disks this can increase the I/O stats.

LVM Features

- It is flexible to expand the space at any time.
- Any file systems can be installed and handle.
- Migration can be used to recover faulty disk.
- Restore the file system using Snapshot features to earlier stage. etc...

My Server Setup – Requirements

- Operating System – CentOS 6.5 with LVM Installation
- Server IP – 192.168.0.200

This series will be titled Preparation for the setting up LVM (Logical Volume Management) through Parts 1-6 and covers the following topics.

Part 1: Setup Flexible Disk Storage with Logical Volume Management

Part 2: **How to Extend/Reduce LVM's in Linux**

Part 3: **How to Take Snapshot of Logical Volume and Restore in LVM**

Part 4: Setup 4hin Provisioning Volumes in LVM

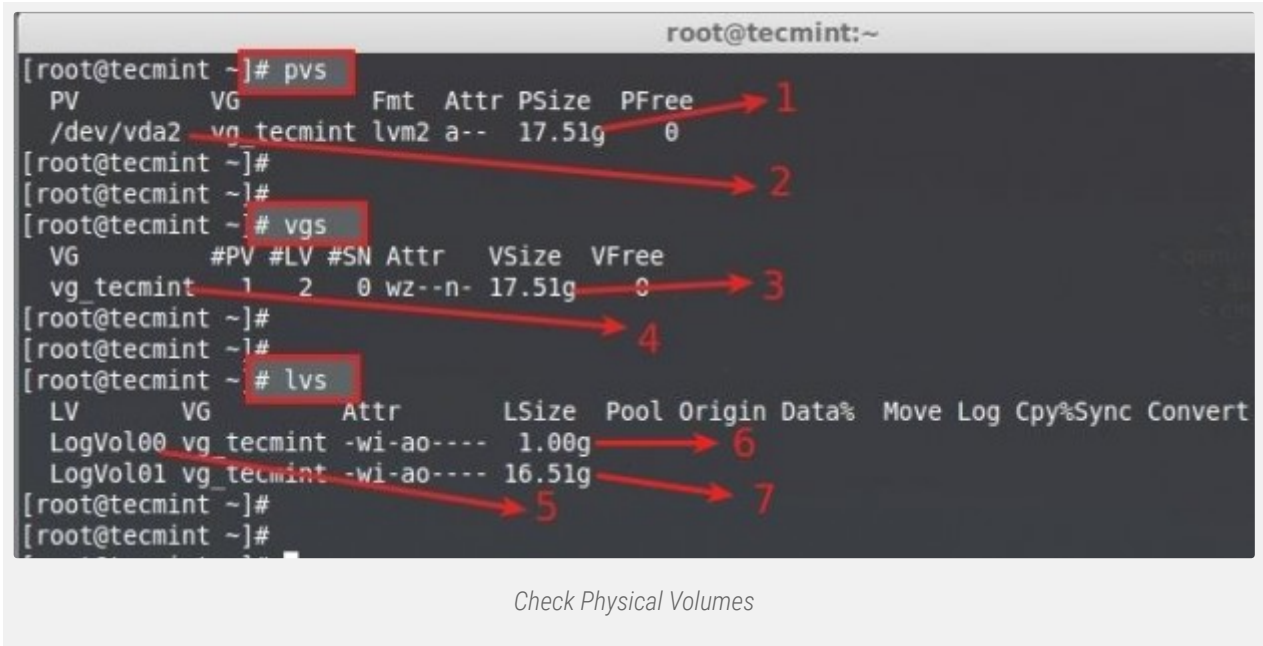
Part 5: Manage Multiple LVM Disks using Striping I/O

Part 6: Migrating LVM Partitions to New Logical Volume (Drive)

Creating LVM Disk Storage in Linux

1. We've used CentOS 6.5 Operating system using LVM in a Virtual Disk (VDA). Here we can see the Physical Volume (PV), Volume Group (VG), Logical Volume (LV) by using following command.

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



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

- Physical Disk Size (PV Size)
- Disk which used was Virtual Disk vda.
- Volume Group Size (VG Size)
- Volume Group name (vg_tecmint)
- Logical Volume name (LogVol00, LogVol01)
- LogVol00 Assigned for sawp with 1GB Size
- LogVol01 Assigned for / with 16.5GB

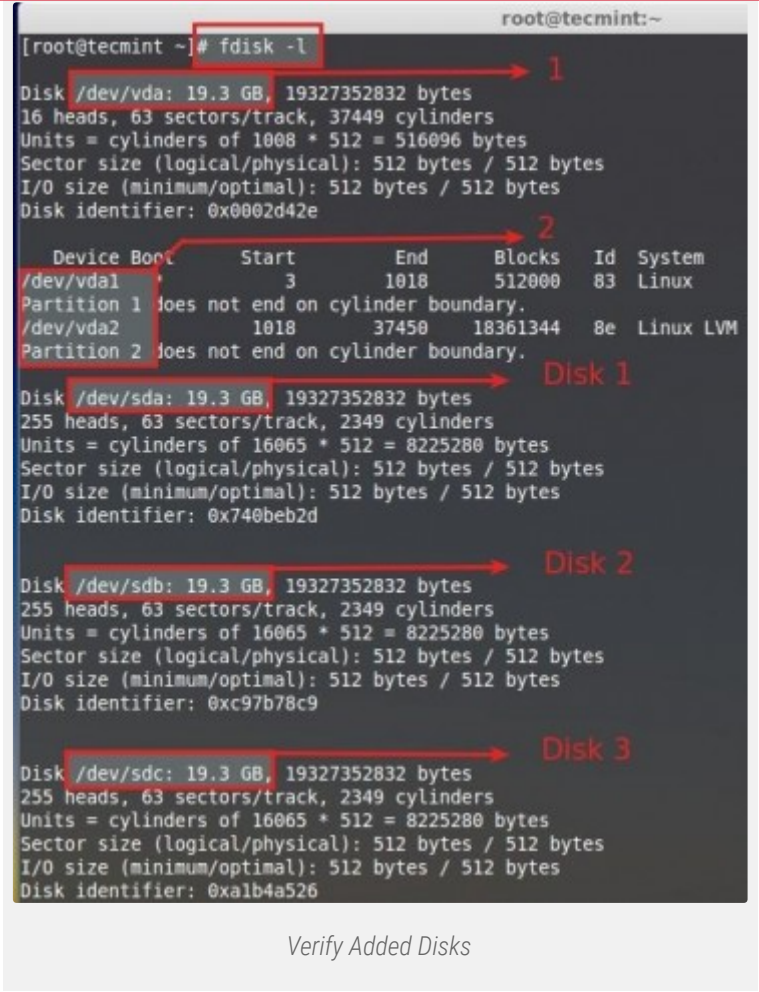
So, from here we come to know that there is not enough free space in VDA disk.

2. For Creating a **New Volume Group**, we need to add Additional **3 hard disks** in this server. It is not Compulsory to use 3 Drives just 1 is Enough to create a new **VG** and **LV** inside that vg, I am adding more here for demonstration purpose and for more feature command explanations.

Following are the disks I have added additionally.

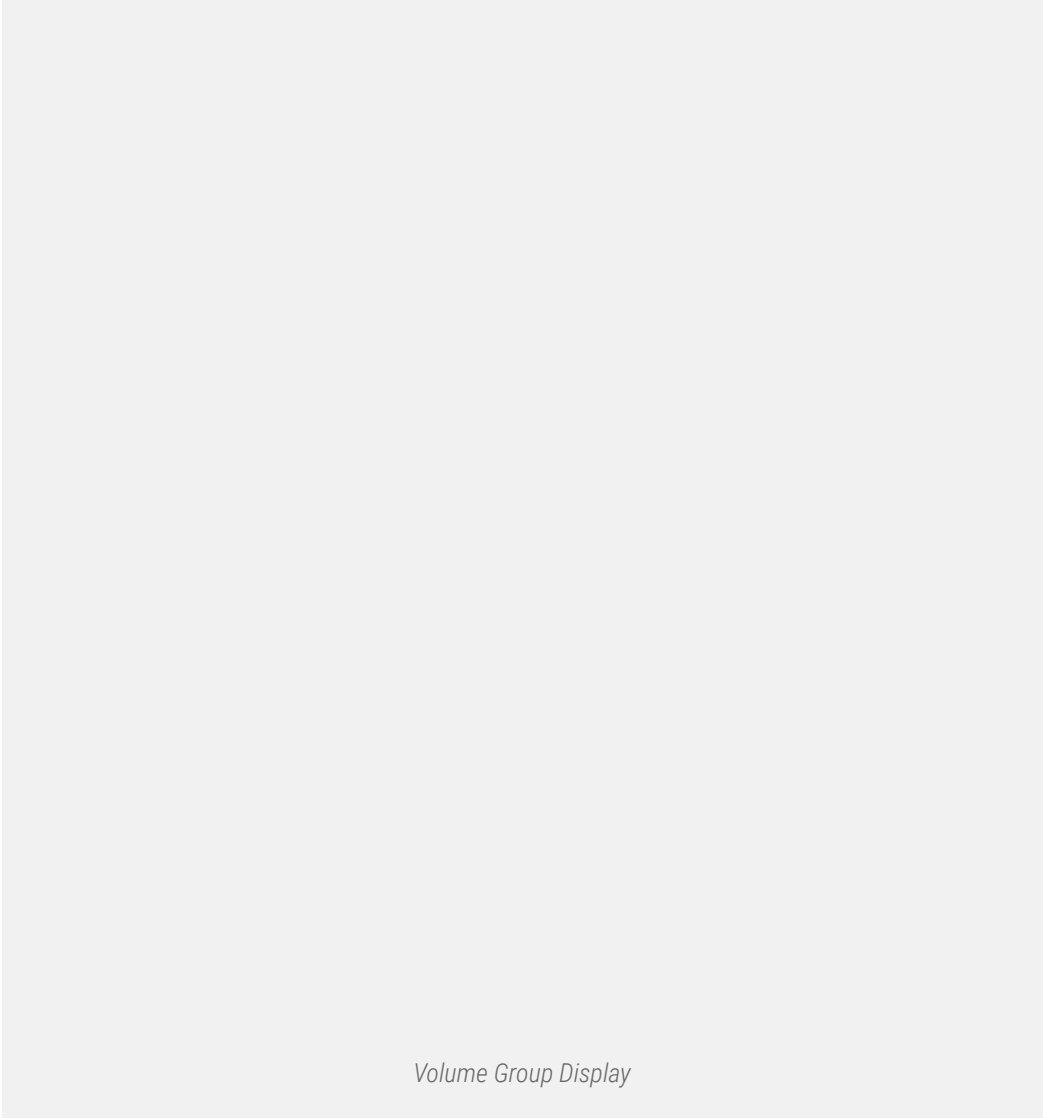
```
sda, sdb, sdc
```

```
# fdisk -l
```



- Default Disk using for Operating system (Centos6.5).
- Partitions defined in default Disk (vda1 = swap), (vda2 = /).
- Additionally added Disks are mentioned as Disk1, Disk2, Disk3.

Each and every Disks are 20 GB in Size. Default PE Size of a Volume Group is 4 MB, Volume group what we are using in this server is configured using default PE.



- **VG Name** – A Volume Group name.
- **Format** – LVM Architecture Used LVM2.
- **VG Access** – Volume Group is in Read and Write and ready to use.
- **VG Status** – Volume Group can be re-sized, We can Expand more if we need to add more space.
- **Cur LV** – Currently there was 2 Logical volumes in this Volume Group.
- **CurPV and Act PV** – Currently Using Physical Disk was 1 (vda), And its active, so what we can use this volume group.
- **PE Size** – Physical Extends, Size for a disk can be defined using PE or GB size, 4MB is the Default PE size of LVM. For example, if we need to create 5 GB size of logical volume we can use sum of 1280 PE, Don't you understand what I'm saying ?.

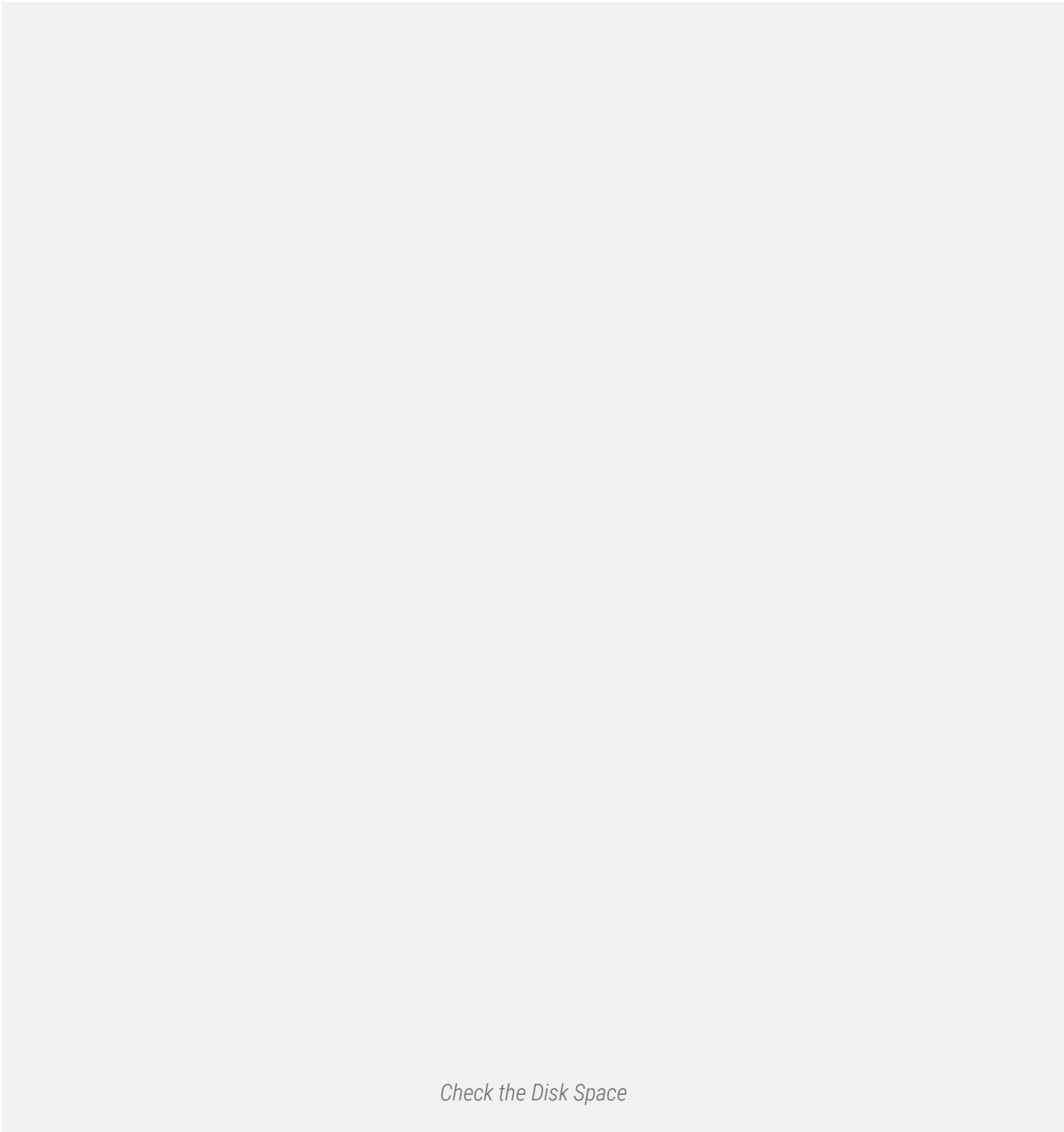
Here the Explanation -> 1024MB = 1GB, if so 1024MB x 5 = 5120PE = 5GB, Now Divide the 5120/4 = 1280, 4 is the Default PE Size.



- **Total PE** – This Volume Group have.
- **Alloc PE** – Total PE Used, full PE already Used, 4482 x 4PE = 17928.
- **Free PE** – Here it's already used so there was no free PE.

3. Only vda used, Currently Centos Installed **/boot**, **/**, **swap**, in vda physical disk using lvm there were no space remaining in this disk.

```
# df -TH
```



Check the Disk Space

Above image shows the mount Point we are using **18GB** fully used for root, so there is no free space available.

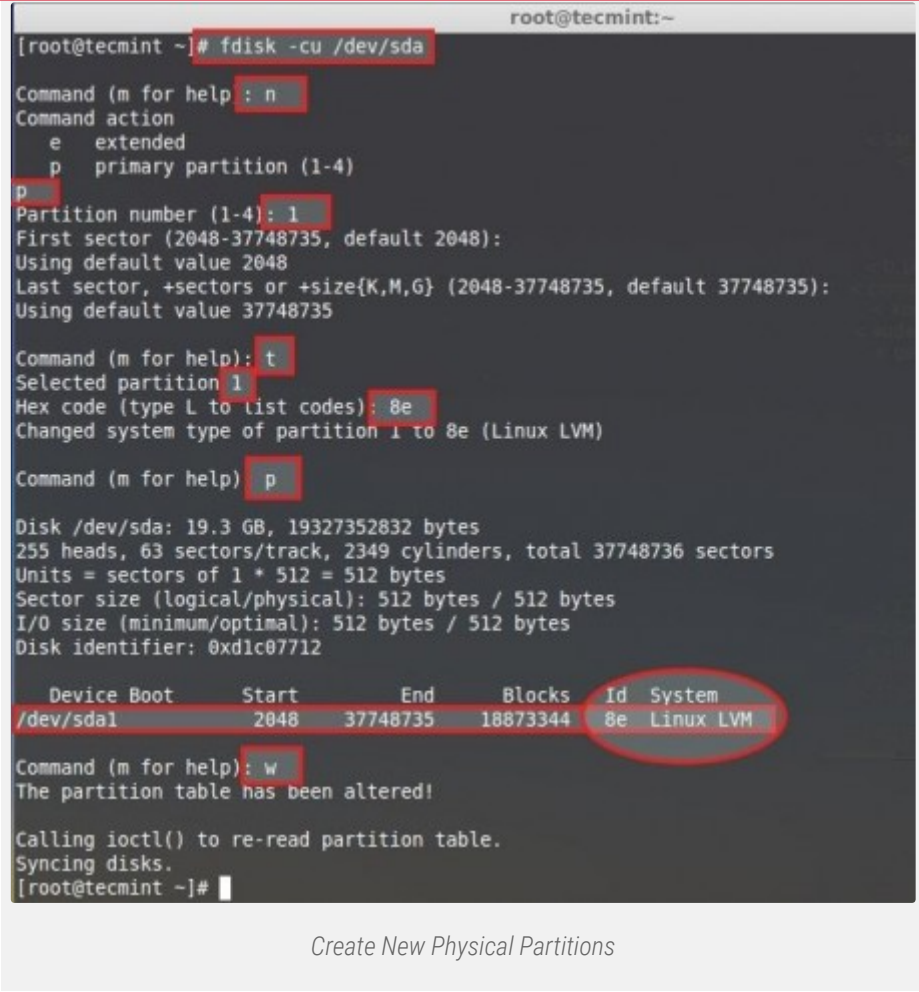
4. So let's, create new physical volume (**pv**), Volume Group (**vg**) in the name of **tecmint_add_vg** and create Logical Volumes (**lv**) in it, Here we can create 4 Logical Volumes in the name of **tecmint_documents**, **tecmint_manager** and **tecmint_public**.

We can extend the Volume Group of currently using VG to get more space. But here, what we are going to do is to Create new Volume Group and play around it, later we can see how to extend the file systems Volume group which is currently in use.

Before using a new Disk we need to partition the disk using fdisk.

```
# fdisk -cu /dev/sda
```

- **c** – Switch off DOS-compatible mode it is Recommend to include this Option.
- **u** – While listing the partition tables it will give us in sector instead of cylinder.

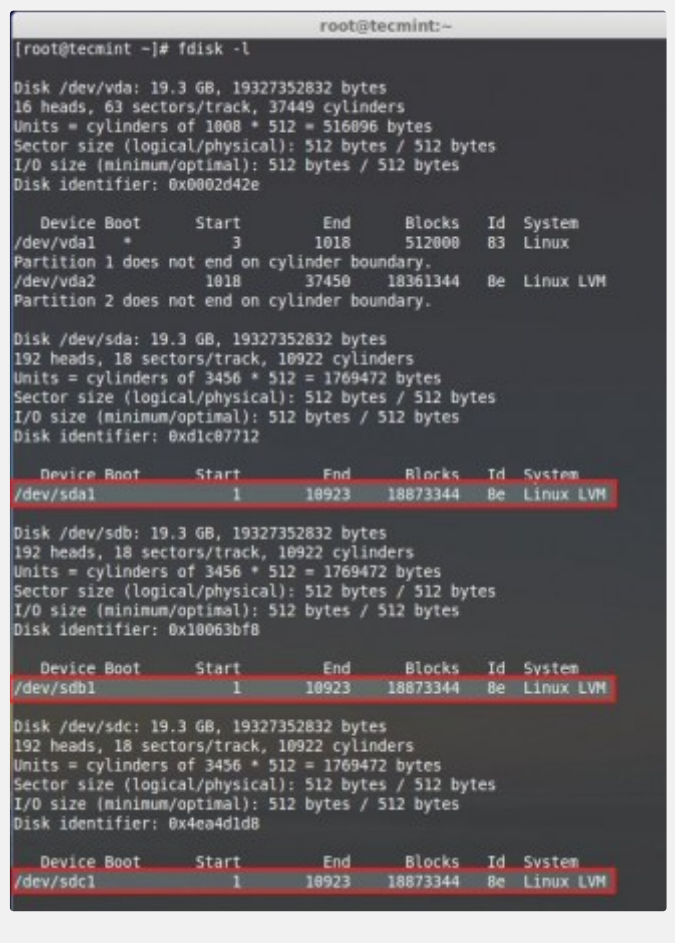


Next, follow the below steps to create new partition.

- Choose **n** to create new.
- Choose **p** to create a primary partition.
- Choose which number of partition 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 type codes.
- Print the Partition what we created to just confirm.
- Here we can see the ID as **8e** LINUX LVM.
- Write the changes and exit fdisk.

Do the above steps for other 2 disks sdb and sdc to create new partitions. Then Restart the machine to verify the partition table using fdisk command.

```
# fdisk -l
```



Creating Physical Volumes

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

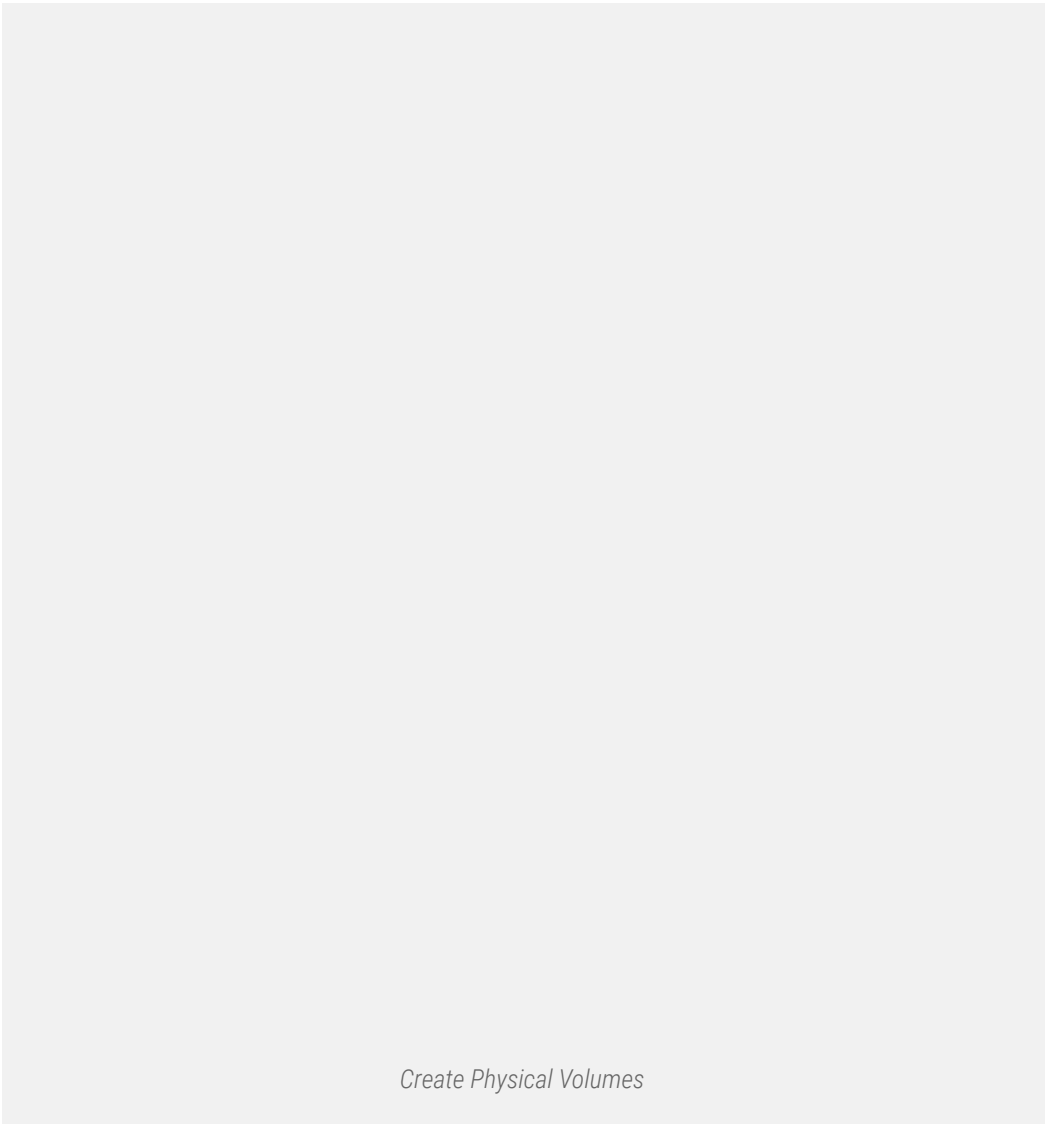
```
# pvs
```

Then create the new physical disks using command.

```
# pvcreate /dev/sda1 /dev/sdb1 /dev/sdc1
```

Once again list the disk to see the newly created Physical disks.

```
# pvs
```



Creating Volume Groups

6. Create Volume Group in the name of **tecmin****t_add_vg** using available free PV Create using PE size 32. To Display the current volume groups, we can see there is one volume group with 1 PV using.

```
# vgs
```

This will create the volume group using 32MB PE size in the name of **tecmin****t_add_vg** using 3 Physical volumes we created in last steps.

```
# vgcreate -s 32M tecmint_add_vg /dev/sda1 /dev/sdb1 /dev/sdc1
```

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

```
# vgs
```

```
root@tecmin:~
[root@tecmin ~]#
[root@tecmin ~]# vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  vg_tecmin  1   2   0 wz--n- 17.51g    0
[root@tecmin ~]#
[root@tecmin ~]#
[root@tecmin ~]# vgcreate -s 32M tecmin add vg /dev/sda1 /dev/sdb1 /dev/sdc1
Volume group "tecmin_add_vg" successfully created
[root@tecmin ~]#
[root@tecmin ~]#
[root@tecmin ~]# vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  tecmin_add_vg  3   0   0 wz--n- 53.91g 53.91g
  vg_tecmin  1   2   0 wz--n- 17.51g    0
[root@tecmin ~]#
[root@tecmin ~]#
```

Create Volume Groups

```
root@tecmin:~
[root@tecmin ~]#
[root@tecmin ~]#
[root@tecmin ~]# vgs
  VG      #PV #LV #SN Attr   VSize  VFree
  tecmin_add_vg  3   0   0 wz--n- 53.91g 53.91g
  vg_tecmin  1   2   0 wz--n- 17.51g    0
[root@tecmin ~]#
[root@tecmin ~]#
[root@tecmin ~]#
```

Verify 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.

7. To Display more information about volume group use command.

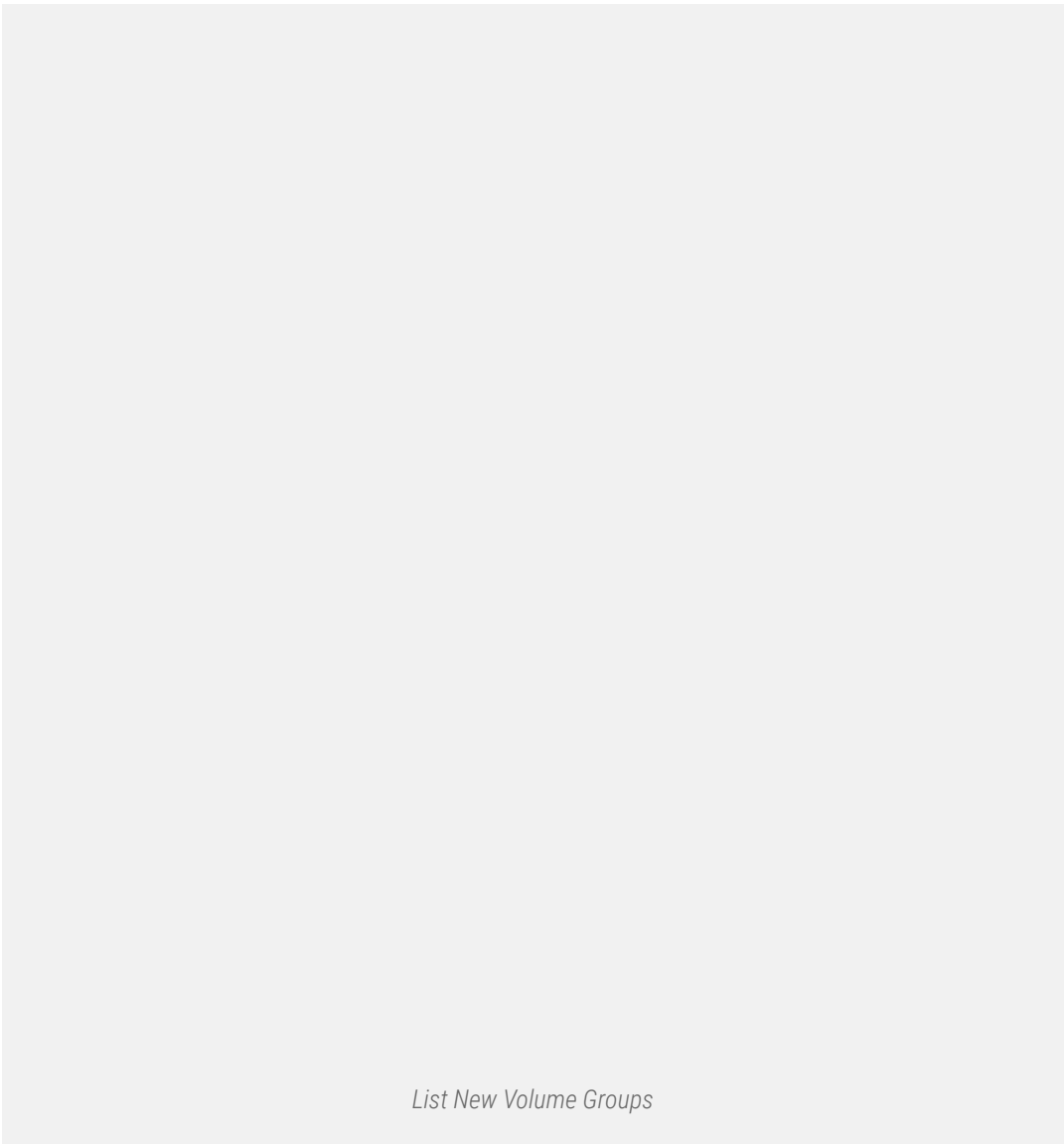
```
# vgs -v
```

```
root@tecmin:~
[root@tecmin ~]# vgs -v
Finding all volume groups
Finding volume group "vg_tecmin"
Finding volume group "tecmin_add_vg"
  VG      Attr  Ext  #PV #LV #SN VSize  VFree  VG UUID
  tecmin_add_vg wz--n- 32.00m  3   0   0 53.91g 53.91g cS3lgE-07EB-Ktlq-47PJ-h33Y-SVLG-izKXkp
  vg_tecmin  wz--n-  4.00m  1   2   0 17.51g    0 JZXiJe-uViS-DyDY-6qaX-zjF4-GkSU-hTZglt
[root@tecmin ~]#
```

Check Volume Group Information

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

```
# vgdisplay tecmint_add_vg
```

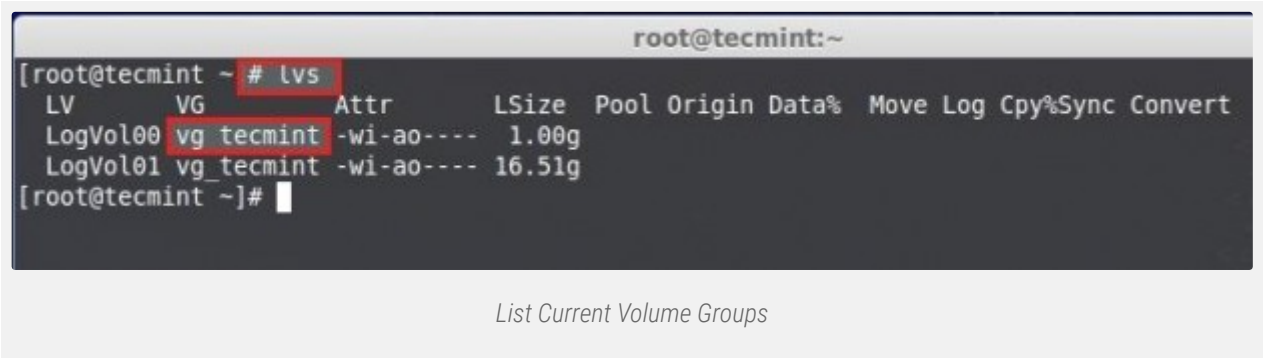


- Volume group name
- LVM Architecture used.
- It can be read and write state, ready to use.
- This volume group can be resizeable.
- No of Physical disk used and they are active.
- Volume Group total size.
- A Single PE size was 32 here.
- Total number of PE available in this volume group.
- Currently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

Creating Logical Volumes

9. Now, ceate 3 Logical Volumes in the name of **tecmint_documents**, **tecmint_manager** and **tecmint_public**. Here, we can see how to Create Logical Volumes Using PE size and Using GB Size. First, list the Current Logical Volumes using following command.

```
# lvs
```



10. These Logical Volumes are in **vg_tecmint** Volume Group. List and see how much free spaces are there to create logical volumes using **pvs** command.

```
# pvs
```

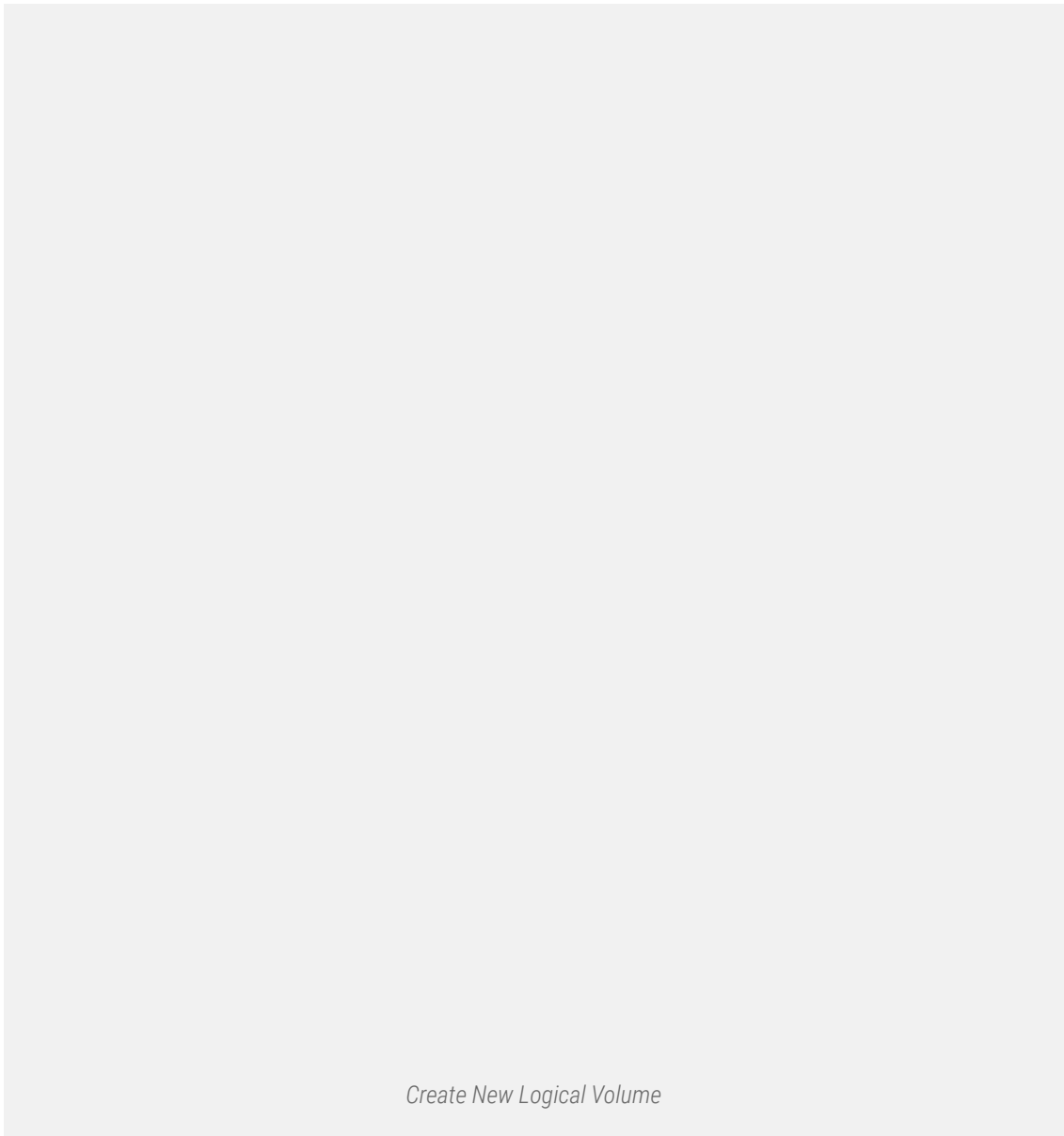



11. Volume group size is **54GB** and its unused, So we can Create LV in it. Let us divide volume group to equal size to create 3 Logical Volumes. That means **54GB/3 = 18GB**, A single Logical Volume will be 18GB in Size after Creation.

Method 1: Creating Logical Volumes using PE Size’s

First let us create Logical Volumes Using Physical Extends (PE) size. We need to know Default PE size assigned for this Volume Group and Total PE available to create new Logical Volumes, Run the command to get the info using.

```
# vgdisplay tecmint_add_vg
```



- Default PE Assigned for this VG is 32MB, Here Single PE size will be 32MB.
- Total Available PE is 1725.

Just do and see a little Calculation using bc command.



```
# bc
```

```
1725PE/3 = 575 PE.  
575 PE x 32MB = 18400 --> 18GB
```

Calculate Disk Space

Press **CRTL+D** to Exit from **bc**. Let us now Create 3 Logical Volumes using 575 PE's.

```
# lvcreate -l (Extend size) -n (name_of_logical_volume) (volume_group)  
# lvcreate -l 575 -n tecmint_documents tecmint_add_vg  
# lvcreate -l 575 -n tecmint_manager tecmint_add_vg  
# lvcreate -l 575 -n tecmint_public tecmint_add_vg
```

- **-l** – Creating using Extent Size
- **-n** – Give a Logical Volume name.

List the Created Logical Volumes using **lvs** command.

```
# lvs
```

root@tecmint:~

```
[root@tecmint ~]#  
[root@tecmint ~]# lvcreate -l 575 -n tecmint_documents tecmint_add_vg  
Logical volume "tecmint_documents" created  
[root@tecmint ~]#  
[root@tecmint ~]# lvcreate -l 575 -n tecmint_manager tecmint_add_vg  
Logical volume "tecmint_manager" created  
[root@tecmint ~]#  
[root@tecmint ~]# lvcreate -l 575 -n tecmint_public tecmint_add_vg  
Logical volume "tecmint_public" created  
[root@tecmint ~]#  
[root@tecmint ~]#  
[root@tecmint ~]# lvs  
LV VG Attr LSize Pool Origin Data% Move Log  
tecmint_documents tecmint_add_vg -wi-a----- 17.97g  
tecmint_manager tecmint_add_vg -wi-a----- 17.97g  
tecmint_public tecmint_add_vg -wi-a----- 17.97g  
LogVol00 vg_tecmint -wi-ao----- 1.00g  
LogVol01 vg_tecmint -wi-ao----- 16.51g  
[root@tecmint ~]#
```

List Created Logical Volumes

Method 2: Creating Logical Volumes using GB Size's

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

```
# lvcreate -L 18G -n tecmint_documents tecmint_add_vg
# lvcreate -L 18G -n tecmint_manager tecmint_add_vg
# lvcreate -L 18G -n tecmint_public tecmint_add_vg
# lvcreate -L 17.8G -n tecmint_public tecmint_add_vg
```

List the Created logical Volumes using lvs command.

```
# lvs
```

root@tecmint:~

```
[root@tecmint ~]#
[root@tecmint ~]# lvcreate -L 18G -n tecmint_documents tecmint_add_vg
Logical volume "tecmint_documents" created
[root@tecmint ~]#
[root@tecmint ~]# lvcreate -L 18G -n tecmint_manager tecmint_add_vg
Logical volume "tecmint_manager" created
[root@tecmint ~]#
[root@tecmint ~]# lvcreate -L 18G -n tecmint_public tecmint_add_vg
Volume group "tecmint_add_vg" has insufficient free space (573 extents): 576 required.
[root@tecmint ~]#
[root@tecmint ~]# lvcreate -L 17.8G -n tecmint_public tecmint_add_vg
Rounding up size to full physical extent 17.81 GiB
Logical volume "tecmint_public" created
[root@tecmint ~]#
[root@tecmint ~]#
[root@tecmint ~]# lvs
  LV          VG          Attr      LSize   Pool Origin Data%  Move Log Cpy%Sync
  tecmint_documents  tecmint_add_vg -wi-a----- 18.00g
  tecmint_manager    tecmint_add_vg -wi-a----- 18.00g
  tecmint_public     tecmint_add_vg -wi-a----- 17.81g
  LogVol00          vg_tecmint     -wi-ao----  1.00g
  LogVol01          vg_tecmint     -wi-ao---- 16.51g
```

Verify Created Logical Volumes

Here, we can see while creating 3rd LV we can't Round-up to 18GB, It is because of small changes in size, But this issue will be ignored while creating LV using Extend size.

Creating File System

12. For using the logical volumes we need to format. Here I am using ext4 file-system to create the volumes and going to mount 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@tecmin:~

```
[root@tecmin ~]# mkfs.ext4 /dev/tecmin add vg/tecmin documents
mke2fs 1.41.12 (17-May-2010)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
1179648 inodes, 4710400 blocks
235520 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
144 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000

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

This filesystem will be automatically checked every 38 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@tecmin ~]#
```

Create Ext4 File System

13. Let us Create Directories in /mnt and Mount the Logical volumes what we have created file-system.

```
# mount /dev/tecmin_add_vg/tecmin_documents /mnt/tecmin_documents/
# mount /dev/tecmin_add_vg/tecmin_public /mnt/tecmin_public/
# mount /dev/tecmin_add_vg/tecmin_manager /mnt/tecmin_manager/
```

List and confirm the Mount point using.

```
# df -h
```

root@tecmin:~

```
[root@tecmin ~]#
[root@tecmin ~]# mount /dev/tecmin_add_vg/tecmin_documents /mnt/tecmin_documents/
[root@tecmin ~]#
[root@tecmin ~]# mount /dev/tecmin_add_vg/tecmin_public /mnt/tecmin_public/
[root@tecmin ~]#
[root@tecmin ~]# mount /dev/tecmin_add_vg/tecmin_manager /mnt/tecmin_manager/
[root@tecmin ~]#
[root@tecmin ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/vg_tecmin-LogVol01	17G	2.2G	14G	14%	/
tmpfs	939M	0	939M	0%	/dev/shm
/dev/vda1	485M	39M	421M	9%	/boot
/dev/mapper/tecmin_add_vg-tecmin_documents	18G	172M	17G	2%	/mnt/tecmin_documents
/dev/mapper/tecmin_add_vg-tecmin_public	18G	172M	17G	2%	/mnt/tecmin_public
/dev/mapper/tecmin_add_vg-tecmin_manager	18G	172M	17G	2%	/mnt/tecmin_manager

```
[root@tecmin ~]#
```

Mount Logical Volumes

Permanent Mounting

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
```

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

```
# vim /etc/fstab
```


Our fstab Entry want to be similar to below sample. Save and exit from fstab using wq!.

```
/dev/mapper/tecmin_add_vg-tecmin_documents /mnt/tecmin_documents ext4 defaults 0 0
/dev/mapper/tecmin_add_vg-tecmin_public /mnt/tecmin_public ext4 defaults 0 0
/dev/mapper/tecmin_add_vg-tecmin_manager /mnt/tecmin_manager ext4 defaults 0 0
```

```
root@tecmin:~
[root@tecmin ~]#
[root@tecmin ~]# cat /etc/mtab
/dev/mapper/vg_tecmin-LogVol01 / ext4 rw 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw,gid=5,mode=620 0 0
tmpfs /dev/shm tmpfs rw,rootcontext="system_u:object_r:tmpfs_t:s0" 0 0
/dev/vdal /boot ext4 rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
/dev/mapper/tecmin_add_vg-tecmin_documents /mnt/tecmin_documents ext4 rw 0 0
/dev/mapper/tecmin_add_vg-tecmin_public /mnt/tecmin_public ext4 rw 0 0
/dev/mapper/tecmin_add_vg-tecmin_manager /mnt/tecmin_manager ext4 rw 0 0
[root@tecmin ~]#
```

Get mtab Mount Entry

```
root@tecmin:~
[root@tecmin ~]#
[root@tecmin ~]# vim /etc/fstab
[root@tecmin ~]#
[root@tecmin ~]#
```

Open fstab File

```
root@tecmin:~
#
# /etc/fstab
# Created by anaconda on Sat Jul 19 10:00:45 2014
#
# 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
#
/dev/mapper/vg_tecmin-LogVol01 / ext4 defaults 1 1
UUID=f09b1b8e-92f4-471b-8c39-b843521c9888 /boot ext4 defaults 1 1
/dev/mapper/vg_tecmin-LogVol00 swap swap defaults 0 0
tmpfs /dev/shm tmpfs defaults 0 0
devpts /dev/pts devpts gid=5,mode=620 0 0
sysfs /sys sysfs defaults 0 0
proc /proc proc defaults 0 0
/dev/mapper/tecmin_add_vg-tecmin_documents /mnt/tecmin_documents ext4 defaults 0 0
/dev/mapper/tecmin_add_vg-tecmin_public /mnt/tecmin_public ext4 defaults 0 0
/dev/mapper/tecmin_add_vg-tecmin_manager /mnt/tecmin_manager ext4 defaults 0 0
```

Add Auto Mount Entry

Execute the command `mount -a` to check for the fstab entry before restart.

```
# mount -av
```

```
root@tecmin:~
[root@tecmin ~]#
[root@tecmin ~]# mount -av
mount: UUID=f09b1b8e-92f4-471b-8c39-b843521c9888 already mounted on /boot
mount: tmpfs already mounted on /dev/shm
mount: devpts already mounted on /dev/pts
mount: sysfs already mounted on /sys
mount: proc already mounted on /proc
mount: /dev/mapper/tecmin_add_vg-tecmin_documents already mounted on /mnt/tecmin_documents
mount: /dev/mapper/tecmin_add_vg-tecmin_public already mounted on /mnt/tecmin_public
mount: /dev/mapper/tecmin_add_vg-tecmin_manager already mounted on /mnt/tecmin_manager
nothing was mounted
[root@tecmin ~]#
```

Verify fstab Entry

Here we have seen how to setup flexible storage with logical volumes by using physical disk to physical volume, physical volume to volume group, volume group to logical volumes.

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

