How to configure LVM in linux rhel5 / Centos

Logical Volume Management[LVM]

Every system contains **Physical Volums[PV]**. Such as hard disks, partitions or external storages. Volume management treats **PV**s as sequences of chunks called **Physical Extents (PEs)**. There is an another concept also. **Logical Extents(LE)**. Each **LE** maps one-to-one **PE**. The system pools **LE**s into a **Volume Group (VG)**. We can extend this **VG** by adding a group of **Logical extents** to it from anywhere at anytime.

Uses of LVM:

1. Extending the partitions online
2. Grouping of hard disks
3. Reducing the partitions/hard dsik size (offline)
4. Increasing the performance
5. Taking Backup (SNAPSHOT)

Example:
Here we are going to discuss a **Volume Group(VG)** created from 3 **Physical Voumes(PV)**. And in that **VG** we'll create two **Logival Volumes(LV)** And mount it to **/linux1** and **/linux2** respectively.
Now first of all in our example, we have 3 partitions. In real industry it may be 3 different hard disks.
Let it be (1) **/dev/sda5** (2) **/dev/sda6** (3) **/dev/sda7** each of size **300Mb**.

```
[root@vm4 ~]# fdisk -l
/dev/sda5 429 465 297171 8e Linux LVM
/dev/sda6 466 502 297171 8e Linux LVM
/dev/sda7 503 539 297171 8e Linux LVM
```

Steps:
First we will convert this partions(hard disks) into **Physical Volumes(PV).** Then we'll create a **Volume Group (VG)** from those **PV s.** Then inside that **VG,** We'll create two **Logical Volumes (LV)** and we'll mount those for use.

Step1: Creating Physical Volume(PV)s.
Partitions or disks can be converted into **PV s** using the following Command.

```
#pvcreate PARTITION_NAMES
#pvcreate /dev/sda5 /dev/sda6 /dev/sda7
```

or as below

```
#pvcreate /dev/sda{5,6,7}
```

```
[root@vm4 ~]# pvcreate /dev/sda5 /dev/sda6 /dev/sda7
Physical volume "/dev/sda5" successfully created
Physical volume "/dev/sda6" successfully created
Physical volume "/dev/sda7" successfully created
```

Monitoring or verifying the PV s:

You can verify the **PV s** using following commands,

```
#pvscan
#pvdisplay
#pvs
[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 lvm2 -- 290.21M 290.21M
/dev/sda6 lvm2 -- 290.21M 290.21M
/dev/sda7 lvm2 -- 290.21M 290.21M
```

Step2: Creating Volume Group(VG):

The **Physical Volumes** are grouped into one to make it a **Volume Group(VG).** It can be done using the following command.

```
#vgcreate VG_NAME PV_NAMES
#vgcreate oracle /dev/sda5 /dev/sda6 /dev/sda7
```

or as below

```
#vgcreate oracle /dev/sda{5,6,7}
```

It will have a appoximate size of **900(300+300+300).** Some part will for writing headers LE and making LE-PE mapping.

```
[root@vm4 ~]# vgcreate oracle /dev/sda5 /dev/sda6 /dev/sda7
Volume group "oracle" successfully created
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
oracle 3 0 0 wz--n- 864.00M 864.00M
```

Monitoring or verifying the VG s:

You can verify the **VG s** using following commands,

```
#vgscan
#vgdisplay
#vgs
```

Output of #vgs is shown above the picture.

Step3: Creating Logical Volumes In Volume Group:

Now we got a volume group "**Oracle**" of size as the total size of all individual disks/partitions. Now we can create **Logical Volumes** or usable partitions inside it. We will create two logical Volumes **lvm1** and **lvm2** of size **100Mb** each.

The lvm1 and lvm2 can be created using the following commands.

```
#lvcreate -L SIZE -n LV_NAME VG_NAME
#lvcreate -L 100M -n lvm1 oracle
#lvcreate -L 100M -n lvm1 oracle
[root@vm4 ~]# lvcreate -L 100M -n lvm1 oracle
Logical volume "lvm1" created
[root@vm4 ~]# lvcreate -L 100M -n lvm2 oracle
Logical volume "lvm2" created
```

Monitoring or verifying the LV s:

You can verify the **LV s** using following commands,

```
#lvscan
#lvdisplay
#lvs
```

```
[root@vm4 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lvm1 oracle -wi-a- 100.00M
lvm2 oracle -wi-a- 100.00M
```

The Logical Voumes lvm1 and lvm2 should be formatted(making filesystem in those) before mounting it. Then only you can use those partitions.

Here formatting in ext3:

```
#mkfs.ext3 /dev/oracle/lvm1
#mkfs.ext3 /dev/oracle/lvm2
```

Making Mount Points:

```
#mkdir /linux1
#mkdir /linux2
```

Mounting (Temporary):

```
#mount /dev/oracle/lvm1 /linux1
#mount /dev/oracle/lvm2 /linux2
```

```
[root@vm4 ~]# mount
[Output truncated]
/dev/mapper/oracle-lvm1 on /linux1 type ext3 (rw)
/dev/mapper/oracle-lvm2 on /linux2 type ext3 (rw)
```

Extending a Logical Volume (Online):

Now We have a Volume group "oracle" of size about 900Mb. And two Logical vloumes lvm1 and lvm2 mounted on /linux1 and /linux2 respectively. Each having 100Mb size. Now we'll extend the size of lvm1 by 100Mb.
Extending size of a LV can be done online, That is by keeping them mounted. It can be achived by executing following command.

```
#lvextend -L +SIZE THE_PATH_OF_LV
#lvextend -L +100M /dev/oracle/lvm1
```

Before:

```
[root@vm4 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lvm1 oracle -wi-ao 100.00M
lvm2 oracle -wi-ao 100.00M
```

Executing:

```
[root@vm4 ~]# lvextend -L +100M /dev/oracle/lvm1
Extending logical volume lvm1 to 200.00 MB
Logical volume lvm1 successfully resized
```

After:

```
[root@vm4 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lvm1 oracle -wi-ao 200.00M
lvm2 oracle -wi-ao 100.00M
```

After executing above commands you can verify the changed size by any of following commands

```
#lvs, #lvdisplay, #lvscan
```

But if you check

```
#df -hT
```

it will be showing the old size only. Because the filesystem is updated the changed in Logical Volume. It can be updated by following command.

```
#resize2fs /dev/oracle/lvm1
```

NOTE: In case of extending the LV is resized first and the filesystem after that. But in case of shrinking a LV, filesystem is shrinked first followed by the shrink in LV.

Before:

```
[root@vm4 ~]# df -hT
Filesystem Type Size Used Avail Use% Mounted on
/dev/mapper/oracle-lvm1
ext3 97M 5.6M 87M 7% /linux1
/dev/mapper/oracle-lvm2
ext3 97M 5.6M 87M 7% /linux2
```

Executing:

```
[root@vm4 ~]# resize2fs /dev/oracle/lvm1
resize2fs 1.39 (29-May-2006)
Filesystem at /dev/oracle/lvm1 is mounted on /linux1; on-line resizing required
Performing an on-line resize of /dev/oracle/lvm1 to 204800 (1k) blocks.
The filesystem on /dev/oracle/lvm1 is now 204800 blocks long.
```

After:

```
[root@vm4 ~]# df -hT
Filesystem Type Size Used Avail Use% Mounted on
/dev/mapper/oracle-lvm1
ext3 194M 5.6M 179M 4% /linux1
/dev/mapper/oracle-lvm2
ext3 97M 5.6M 87M 7% /linux2
```

Shrinking a Logical Volume (Offline):

As we extended the size of Logical Volume, we can reduce the size also. But in later case, it can be done only offline. That is the LV should be unmounted for reducing its size.

For Shrinking a Volume:

1. Filesystem must be reduced first
2. Requires a filesystem check and cannot be performed online
3. #lvreduce can then reduce the volume

Now We have a Volume group "oracle" of size about 900Mb. And two Logical vloumes lvm1 and lvm2 mounted on /linux1 and /linux2 respectively. Size of lvm1 is 200Mb and size of lvm2 is 100Mb. Now we'll reduce the size of lvm2 by 40Mb.

Current status:

```
[root@vm4 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lvm1 oracle -wi-ao 200.00M
lvm2 oracle -wi-ao 100.00M
```

Step1: Unmont the volume:

```
#umount /linux2
```

Step2: Checking the filesystem:

```
#e2fsck -f LV_Path
#fsck -f /dev/oracle/lvm2
```

```
[root@vm4 ~]# e2fsck -f /dev/oracle/lvm2
e2fsck 1.39 (29-May-2006)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/oracle/lvm2: 11/25688 files (9.1% non-contiguous), 8914/102400 blocks
```

This is to make sure that the filesystem is in a consistent state.

Step3: Resizing the filesystem:

```
#resize2fs LV_Path Final_size
#resize2fs /dev/oracle/lvm2 60M
```

[Total size was 100Mb. Reduction 40Mb. So final size is 100-40=60]

```
[root@vm4 ~]# resize2fs /dev/oracle/lvm2 60M
resize2fs 1.39 (29-May-2006)
Resizing the filesystem on /dev/oracle/lvm2 to 61440 (1k) blocks.
The filesystem on /dev/oracle/lvm2 is now 61440 blocks long.
```

Step4: Now reduce the LV using #lvreduce

```
#lvreduce -L Size LV_Path
#lvreduce -L 40M /dev/oracle/lvm2
```

```
[root@vm4 ~]# lvreduce -L 40M /dev/oracle/lvm2
WARNING: Reducing active logical volume to 40.00 MB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lvm2? [y/n]: y
Reducing logical volume lvm2 to 40.00 MB
Logical volume lvm2 successfully resized
```

Before Reducing:

```
root@vm4 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lvm1 oracle -wi-ao 200.00M
lvm2 oracle -wi-ao 100.00M
```

After Reducing:

```
[root@vm4 ~]# lvs
LV VG Attr LSize Origin Snap% Move Log Copy% Convert
lvm1 oracle -wi-ao 200.00M
lvm2 oracle -wi-a- 40.00M
```

After mounting:

```
[root@vm4 ~]# df -hT

/dev/mapper/oracle-lvm1
ext3 194M 5.6M 179M 4% /linux1
/dev/mapper/oracle-lvm2
ext3 59M 5.3M 50M 10% /linux2
```

Adding a Pysical Volume to a VG:

Now we have a Volume Group "oracle" of size 900Mb. Suppose we used upto the maximum usable size. So we need to extend the size of the VG.
In this case we will create a new partition/a new hard disk, and will make it a Physical Volume and add it to the Volume group.
Current status:

```
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
oracle 3 2 0 wz--n- 864.00M 624.00M
```

Now we created one more partition with id LVM.

```
/dev/sda5 429 465 297171 8e Linux LVM
/dev/sda6 466 502 297171 8e Linux LVM
/dev/sda7 503 539 297171 8e Linux LVM
/dev/sda8 540 576 297171 8e Linux LVM
```

Step1:
First we have to convert it to physical volume.

```
#pvcreate /dev/sda8
```

```
[root@vm4 ~]# pvcreate /dev/sda8
Physical volume "/dev/sda8" successfully created
```

Current status of all Physical Volumes:

```
[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 oracle lvm2 a- 288.00M 88.00M
/dev/sda6 oracle lvm2 a- 288.00M 248.00M
/dev/sda7 oracle lvm2 a- 288.00M 288.00M
/dev/sda8 lvm2 -- 290.21M 290.21M
```

See /dev/sda8 is not the part of the VG oracle

Step2:
Now we will add the PV /dev/sda8 to Volume Group "oracle"

```
#vgextend -v oracle /dev/sda8
```

[-v is for verbose. To see what is happening]

```
[root@vm4 ~]# vgextend -v oracle /dev/sda8
Checking for volume group "oracle"
Archiving volume group "oracle" metadata (seqno 5).
Wiping cache of LVM-capable devices
Adding physical volume '/dev/sda8' to volume group 'oracle'
Volume group "oracle" will be extended by 1 new physical volumes
```

```
Creating volume group backup "/etc/lvm/backup/oracle" (seqno 6).
Volume group "oracle" successfully extended
```

After extending the VG with new Physical Volume:

```
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr Vsize VFree
oracle 4 2 0 wz--n- 1.12G 912.00M
```

```
[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 oracle lvm2 a- 288.00M 88.00M
/dev/sda6 oracle lvm2 a- 288.00M 248.00M
/dev/sda7 oracle lvm2 a- 288.00M 288.00M
/dev/sda8 oracle lvm2 a- 288.00M 288.00M
```

Now /dev/sda8 became a part of VG oracle.

Removing a Pysical volume form a VG:

Before removing a physical volume from a volume group, you can make sure that the physical volume is not used by any logical volumes by using the **`#pvdisplay`** command. If the physical volume is still being used you will have to migrate the data to another physical volume using the **`#pvmove`** command. Then use the `vgreduce` command to remove the physical volume.

Current status of Pysical Volumes:

```
[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 oracle lvm2 a- 288.00M 88.00M
/dev/sda6 oracle lvm2 a- 288.00M 248.00M
/dev/sda7 oracle lvm2 a- 288.00M 288.00M
/dev/sda8 oracle lvm2 a- 288.00M 288.00M
```

In this example we will remove the Pysical Volume /dev/sda5 in which some Physical Extents are already used by some LV. So They need to be migrated.

The status of /dev/sds5 is

```
[root@vm4 ~]# pvdisplay /dev/sda5
--- Physical volume ---
PV Name /dev/sda5
VG Name oracle
PV Size 290.21 MB / not usable 2.21 MB
Allocatable yes
PE Size (KByte) 4096
Total PE 72
```

```
Free PE 22
Allocated PE 50
PV UUID 9l5HlF-h8Of-2J6D-TDr4-BY34-cREh-7U5zxm
```

Step1:

Migrate the used Pes using **#pvmove** command

```
[root@vm4 ~]# pvmove -v /dev/sda5
Finding volume group "oracle"
Archiving volume group "oracle" metadata (seqno 6).
[output truncated]
Creating volume group backup "/etc/lvm/backup/oracle" (seqno 9).
```

Step2:

Now reduce the Volume Group size by command **#vgreduce**

```
#vgreduce VG_name Removing_PV_Path
#vgreduce oracle /dev/sda5
```

```
[root@vm4 ~]# vgreduce oracle /dev/sda5
Removed "/dev/sda5" from volume group "oracle"
```

After Remvoing PV:

```
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
oracle 3 2 0 wz--n- 864.00M 624.00M
```

```
[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 lvm2 -- 290.21M 290.21M /Now not part of oracle VG
/dev/sda6 oracle lvm2 a- 288.00M 248.00M
/dev/sda7 oracle lvm2 a- 288.00M 88.00M
/dev/sda8 oracle lvm2 a- 288.00M 288.00M
```

Merging two Volume Groups:

Two different Volume Groups can be merged to a single Volume Group.
Suppose we have two VGs "oracle" and "linux".

Current status of VGs are:

```
VG #PV #LV #SN Attr VSize VFree
linux 1 0 0 wz--n- 288.00M 288.00M
oracle 3 2 0 wz—n- 864.00M 624.00M
```

Now we are going to merge "oracle" and "linux" to get a single VG linux.
We are using the command **#vgmerge** for this.
#vgmerge merges two existing volume groups. The inactive SourceVolumeGroupName will be merged into the DestinationVolumeGroupName if physical extent sizes are equal and physical and logical volume summaries of both volume groups fit into DestinationVolumeGroupName's limits.

#vgmerge -v databases my_vg

merges the inactive volume group named "my_vg" into the active or inactive volume group named "databases" giving verbose runtime information.

Step1:

we are going to merge "oracle" and "linux" to get a single VG linux.
Make the source Volume Group inactive. In this case source is oracle.
#vgchange -a n oracle
umount the LV which uses the oracle VG then only u can inactivate it.

```
[root@vm4 ~]# umount /linux1
[root@vm4 ~]# umount /linux2
[root@vm4 ~]# vgchange -a n oracle
0 logical volume(s) in volume group "oracle" now active
```

#vgmerge Destination Source

we are going to merge "oracle" and "linux" to get a single VG linux. So
#vgmaerge -v linux oracle

```
[root@vm4 ~]# vgmerge -v linux oracle
Checking for volume group "linux"
Checking for volume group "oracle"
Archiving volume group "oracle" metadata (seqno 10).
Archiving volume group "linux" metadata (seqno 1).
Writing out updated volume group
Creating volume group backup "/etc/lvm/backup/linux" (seqno 2).
Volume group "oracle" successfully merged into "linux"
```

```
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
linux 4 2 0 wz--n- 1.12G 912.00M
```

```
[root@vm4 ~]# vgchange -a y linux
2 logical volume(s) in volume group "linux" now active
```

```
[root@vm4 ~]# mount /dev/linux/lvm1 /linux1
[root@vm4 ~]# mount /dev/linux/lvm2 /linux2
[root@vm4 ~]# mount
[output truncated]
/dev/mapper/linux-lvm1 on /linux1 type ext3 (rw)
/dev/mapper/linux-lvm2 on /linux2 type ext3 (rw)
```

```
[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 lvm2 -- 290.21M 290.21M
/dev/sda6 linux lvm2 a- 288.00M 248.00M
/dev/sda7 linux lvm2 a- 288.00M 88.00M
```

```
/dev/sda8 linux lvm2 a- 288.00M 288.00M
/dev/sda9 linux lvm2 a- 288.00M 288.00M
```

Spliting a Volume Group into two:

We can split a Volume Group into two Volume Groups using the command **#vgsplit**
We have one VG **oracle** with 4 PVs. We will split that VG oracle to oracle and another VG **redhat**.
The PVs **/dev/sda8** and **/dev/sda9** will be moved to redhat.
The syntax is as follows:

```
#vgsplit EXISTING_VG NEW_VG PATH OF PVs_TO _BE_MOVED
#vgsplit oracle redhat /dev/sda8 /dev/sda9
[root@vm4 ~]# vgsplit linux redhat /dev/sda8 /dev/sda9
New volume group "redhat" successfully split from "linux"
```

Before:

```
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
linux 4 2 0 wz--n- 1.12G 912.00M

[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 lvm2 -- 290.21M 290.21M
/dev/sda6 linux lvm2 a- 288.00M 248.00M
/dev/sda7 linux lvm2 a- 288.00M 88.00M
/dev/sda8 linux lvm2 a- 288.00M 288.00M
/dev/sda9 linux lvm2 a- 288.00M 288.00M
```

After:

```
[root@vm4 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
linux 2 2 0 wz--n- 576.00M 336.00M
redhat 2 0 0 wz--n- 576.00M 576.00M

[root@vm4 ~]# pvs
PV VG Fmt Attr PSize PFree
/dev/sda5 lvm2 -- 290.21M 290.21M
/dev/sda6 linux lvm2 a- 288.00M 248.00M
/dev/sda7 linux lvm2 a- 288.00M 88.00M
/dev/sda8 redhat lvm2 a- 288.00M 288.00M
/dev/sda9 redhat lvm2 a- 288.00M 288.00M

[root@vm4 ~]# vgchange -a y linux
2 logical volume(s) in volume group "linux" now active
[root@vm4 ~]# vgchange -a y redhat
0 logical volume(s) in volume group "redhat" now active
```