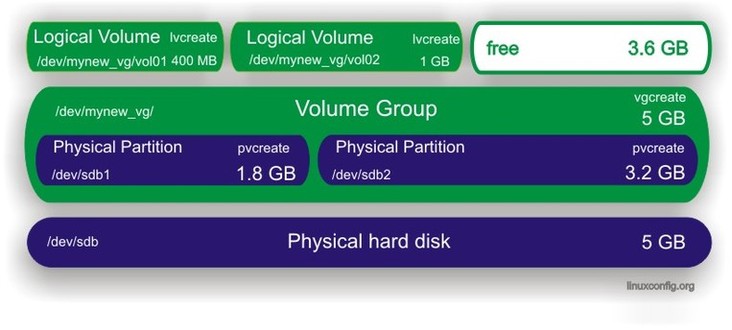
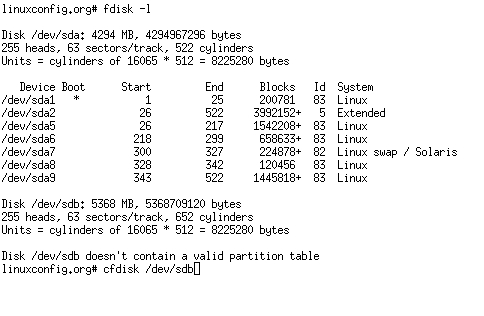
This article describes a basic logic behind a Linux logical volume manager by showing real examples of configuration and usage. Although Debian Linux will be used for this tutorial, you can also apply the same command line syntax with other Linux distributions such as Red Hat, Mandriva, SuSe Linux and others.

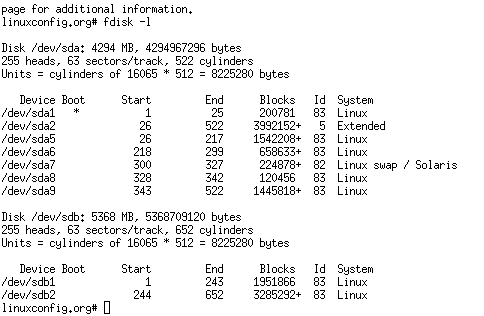
**1. This is what we are going to do**



**2. Create Partitions**

For this Linux lvm example you need an unpartitioned hard disk /dev/sdb. First you need to create physical volumes. To do this you need partitions or a whole disk. It is possible to run pvcreate command on /dev/sdb, but I prefer to use partitions and from partitions I later create physical volumes.

  
Use your preferred partitioning tool to create partitions. In this example I have used cfdisk.



Partitions are ready to use.

**3. Create physical volumes**

Use the pvcreate command to create physical volumes.

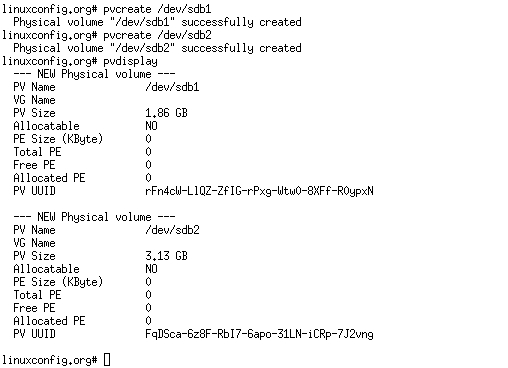
# pvcreate /dev/sdb1  
# pvcreate /dev/sdb2

The pvdisplay command displays all physical volumes on your system.

# pvdisplay

Alternatively the following command should be used:

# pvdisplay /dev/sdb1



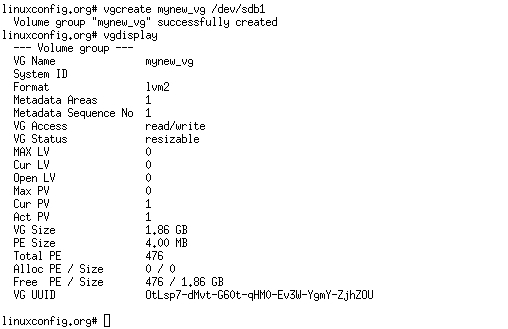
**4. Create Virtual Group**

At this stage you need to create a virtual group which will serve as a container for your physical volumes. To create a virtual group with the name "mynew\_vg" which will include /dev/sdb1 partition, you can issue the following command:

# vgcreate mynew\_vg /dev/sdb1

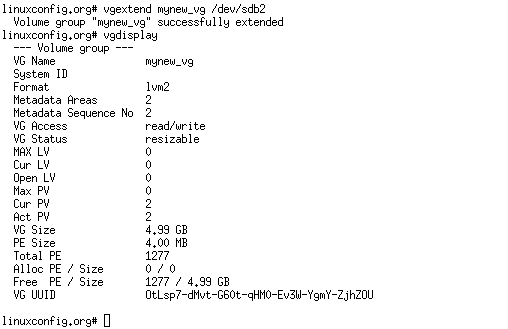
To include both partitions at once you can use this command:

# vgcreate mynew\_vg /dev/sdb1 /dev/sdb2



Feel free to add new physical volumes to a virtual group by using the vgextend command.

# vgextend mynew\_vg /dev/sdb2

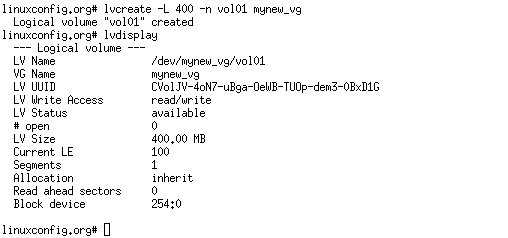


**5. Create Logical Volumes**

From your big cake (virtual group) you can cut pieces (logical volumes) which will be treated as a partitions for your linux system. To create a logical volume, named "vol01", with a size of 400 MB from the virtual group "mynew\_vg" use the following command:

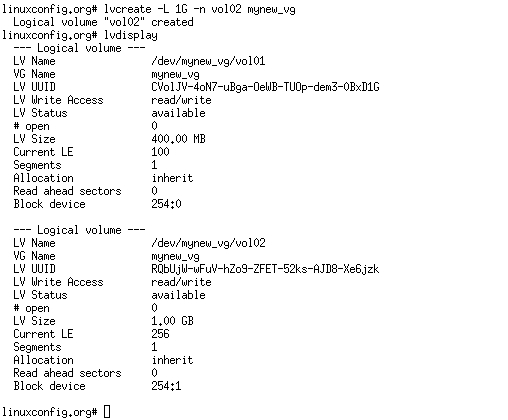
* create a logical volume of size 400 MB -L 400
* create a logical volume of size 4 GB -L 4G

# lvcreate -L 400 -n vol01 mynew\_vg

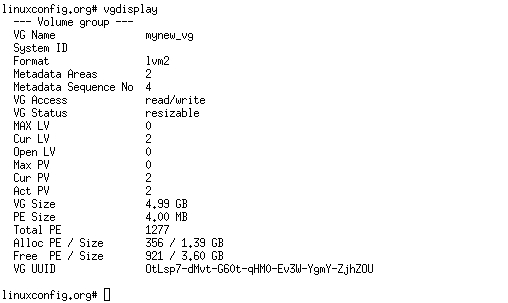


With a following example you will create a logical volume with a size of 1GB and with the name vol02:

# lvcreate -L 1000 -n vol02 mynew\_vg



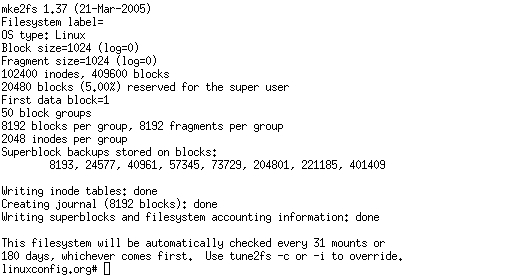
Note the free size in virtual group.



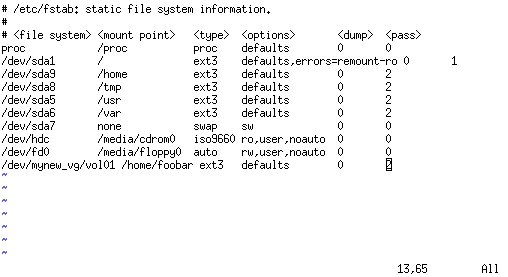
**6. Create File system on logical volumes**

The logical volume is almost ready to use. All you need to do is to create a filesystem.:

# mkfs.ext3 -m 0 /dev/mynew\_vg/vol01

the -m option specifies the percentage reserved for the super-user, set this to 0 if you wish not to waste any space, the default is 5%. 

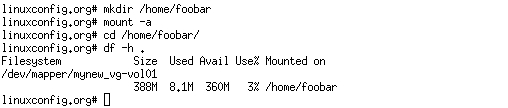
**7. Edit /etc/fstab**

Add an entry for your newly created logical volume into /etc/fstab  


**7.1. Mount logical volumes**

Before you mount do not forget to create a mount point.

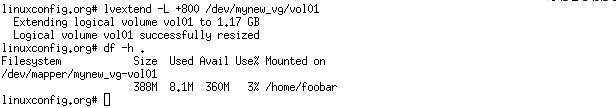
# mkdir /home/foobar



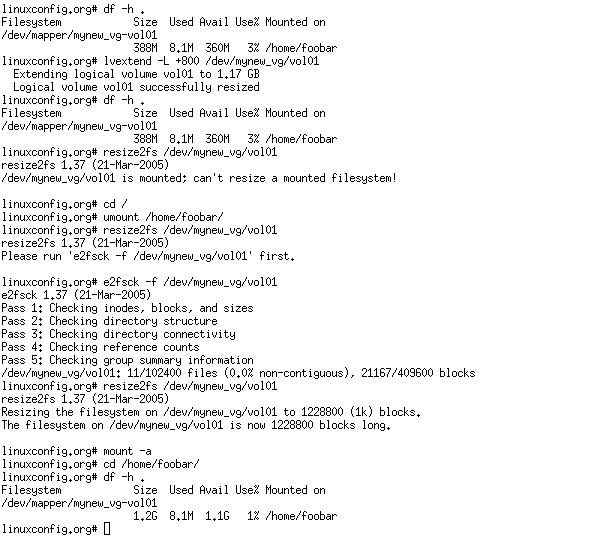
**8. Extend logical volume**

The biggest advantage of logical volume manager is that you can extend your logical volumes any time you are running out of the space. To increase the size of a logical volume by another 800 MB you can run this command:

# lvextend -L +800 /dev/mynew\_vg/vol01

  
The command above does not actually increase the physical size of volume, to do that you need to:

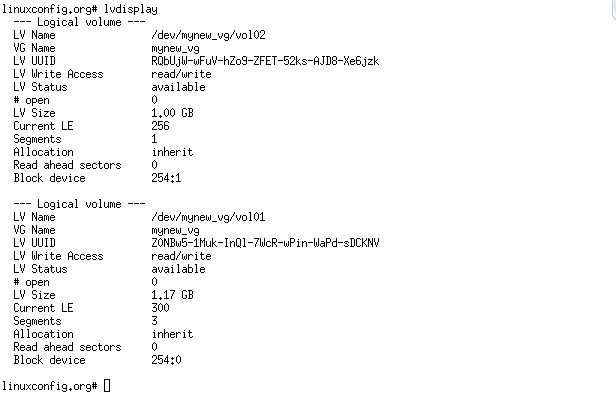
# resize2fs /dev/mynew\_vg/vol01

Look at the figure below to see what problems you may encounter when extending a volume:  


**9. Remove logical volume**

The command lvremove can be used to remove logical volumes. Make sure that before you attempt to remove logical volumes your logical volume does not have any valuable data stored on it, moreover, make sure the volume is unmounted.

# lvdisplay



# lvremove /dev/mynew\_vg/vol02

