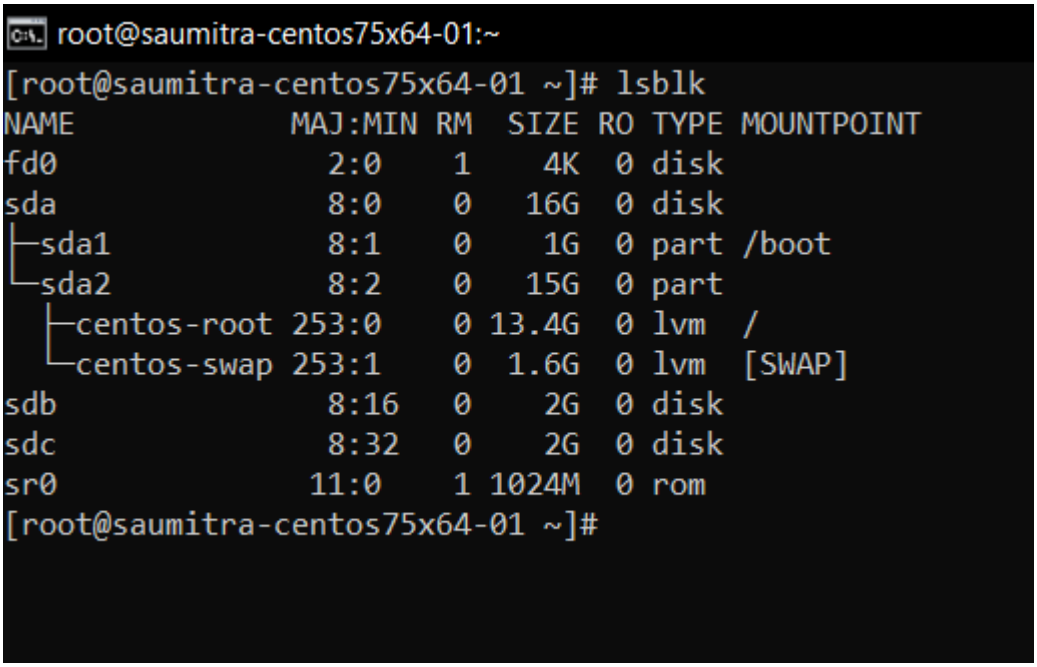
Working with Linux LVM

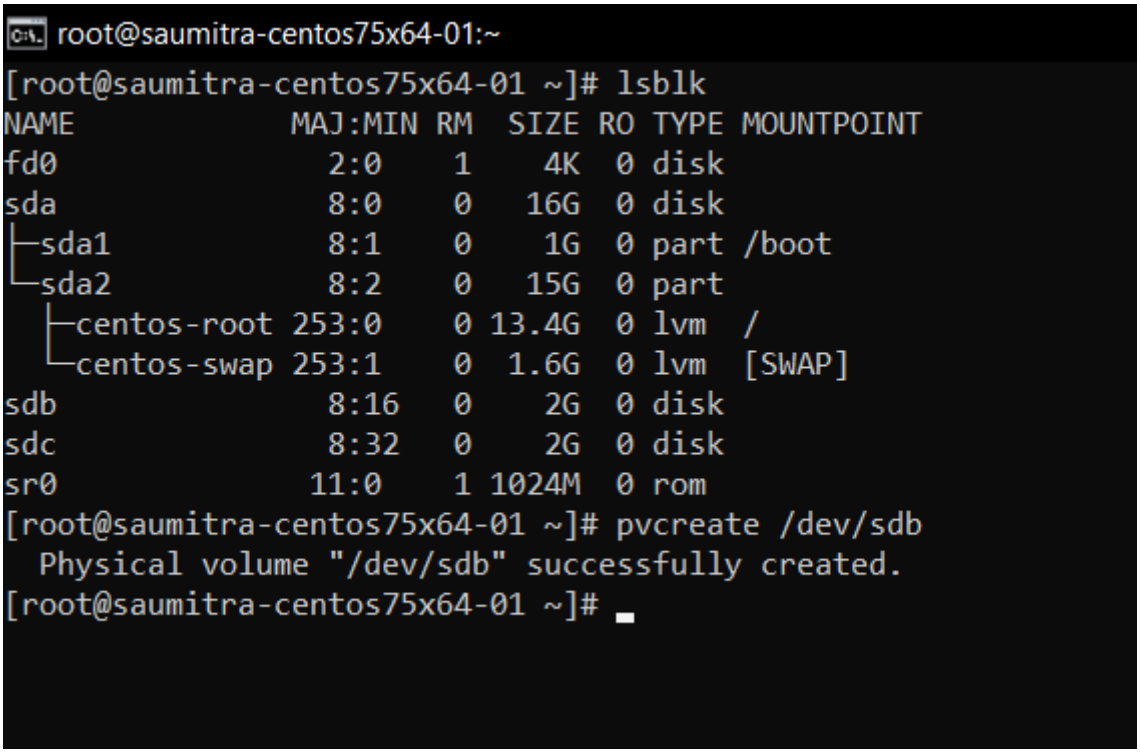
Reference: https://www.youtube.com/watch?v=MeltFN-bXrQ

1. Creating Physical Volumes

Before adding our volumes to a volume group, we need to inform LVM that we intend to use the volume with it. For this we need to make our volume into a new Physical volume (PV). Using the command `pvcreate` storage can be initialized for use by LVM. This is our drive status before pvcreate:



We intend to use the `dev/sdb` drive and create a PV from it. This is our result:



We can view the status of our PVs with the help of `pvdisplay` command.

```
root@saumitra-centos75x64-01:~
[root@saumitra-centos75x64-01 ~]# pvdisplay
  --- Physical volume ---
  PV Name               /dev/sda2
  VG Name               centos
  PV Size               <15.00 GiB / not usable 3.00 MiB
  Allocatable           yes (but full)
  PE Size               4.00 MiB
  Total PE              3839
  Free PE               0
  Allocated PE          3839
  PV UUID               KXhVks-VjyO-I7yP-Ol0Y-MJPn-Cbeb-2EWed4

  "/dev/sdb" is a new physical volume of "2.00 GiB"
  --- NEW Physical volume ---
  PV Name               /dev/sdb
  VG Name
  PV Size               2.00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               GSd6rd-Xq4X-lUCO-kri5-fW5N-fuNc-RFx1TY

[root@saumitra-centos75x64-01 ~]#
```

2. Extending Volume Groups

Since our machine already has a VG `centos` available, we can add our newly created PV to this VG. This can be done using the `vgextend` command.

```
root@saumitra-centos75x64-01:~
[root@saumitra-centos75x64-01 ~]# vgextend centos /dev/sdb
  Volume group "centos" successfully extended
[root@saumitra-centos75x64-01 ~]#
```

Because we have extended the existing VG, the `vgdisplay` command shows the following output.

```
root@saumitra-centos75x64-01:~
[root@saumitra-centos75x64-01 ~]# df -h
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/centos-root   14G  1.2G   13G   9% /
devtmpfs                 908M     0  908M   0% /dev
tmpfs                    920M     0  920M   0% /dev/shm
tmpfs                    920M  8.8M  911M   1% /run
tmpfs                    920M     0  920M   0% /sys/fs/cgroup
/dev/sda1               1014M  142M  873M  14% /boot
tmpfs                    184M     0  184M   0% /run/user/0
[root@saumitra-centos75x64-01 ~]# vgdisplay
  --- Volume group ---
  VG Name               centos
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  4
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               16.99 GiB
  PE Size               4.00 MiB
  Total PE              4350
  Alloc PE / Size       3839 / <15.00 GiB
  Free  PE / Size       511 / <2.00 GiB
  VG UUID               jHgtR8-96h2-AXtm-NZyr-RVw5-kpB4-gtlfT7

[root@saumitra-centos75x64-01 ~]#
```

However, our mounted drive summary (using `df -h`) still does not show the drive.

```
root@saumitra-centos75x64-01:~
[root@saumitra-centos75x64-01 ~]# df -h
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/centos-root   14G  1.2G   13G   9% /
devtmpfs                 908M     0  908M   0% /dev
tmpfs                    920M     0  920M   0% /dev/shm
tmpfs                    920M  8.8M  911M   1% /run
tmpfs                    920M     0  920M   0% /sys/fs/cgroup
/dev/sda1               1014M  142M  873M  14% /boot
tmpfs                    184M     0  184M   0% /run/user/0
[root@saumitra-centos75x64-01 ~]#
```

This is because we haven't yet created a logical volume from the volume group. It is necessary to either create a new LV or extend an existing LV for them to be recognised in the `df -h` command.

Thus, we currently have:

```
[root@saumitra-centos75x64-01 ~]# pvscan
  PV /dev/sda2   VG centos          lvm2 [<15.00 GiB / 0    free]
  PV /dev/sdb    VG centos          lvm2 [<2.00 GiB / <2.00 GiB free]
  Total: 2 [16.99 GiB] / in use: 2 [16.99 GiB] / in no VG: 0 [0   ]
[root@saumitra-centos75x64-01 ~]#
```

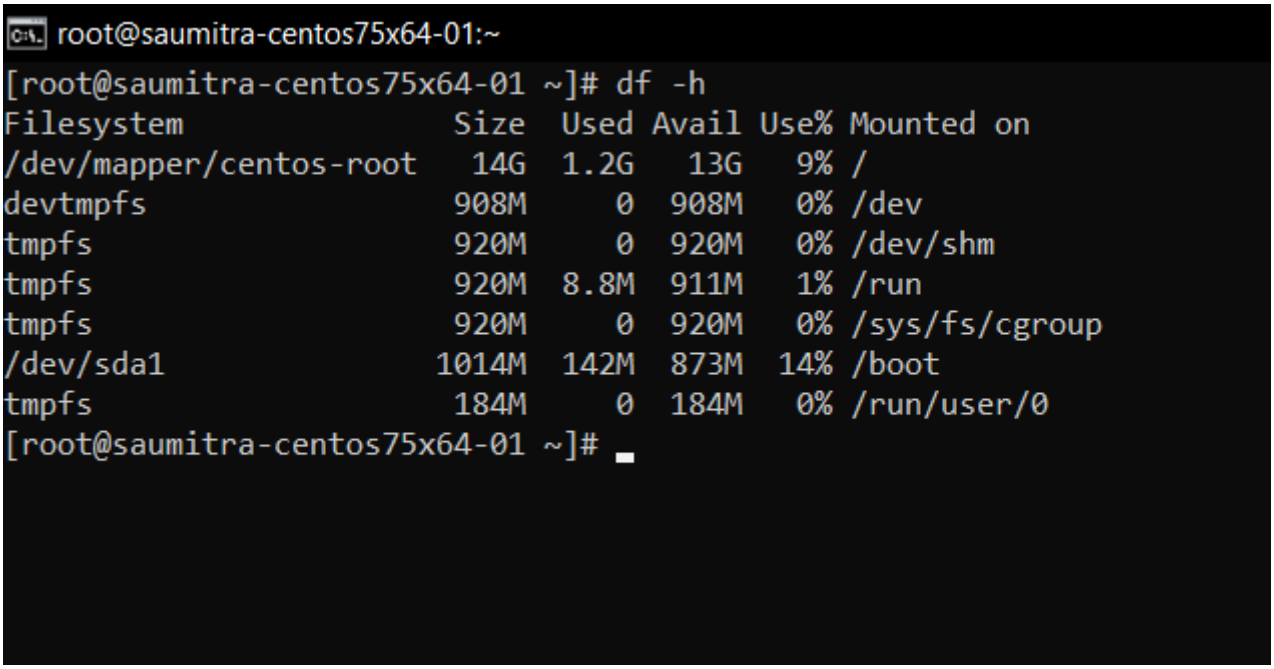
3. Logical Volumes

A. Extending existing LVs-

We can use a part of the 2GB free space from our VG and allocate it to existing LVs. Let us extend the `/dev/mapper/centos-root` by 512 MB.

```
root@saumitra-centos75x64-01:~
[root@saumitra-centos75x64-01 ~]# df -h
Filesystem               Size  Used Avail Use% Mounted on
/dev/mapper/centos-root   14G  1.2G   13G   9% /
devtmpfs                 908M     0  908M   0% /dev
tmpfs                    920M     0  920M   0% /dev/shm
tmpfs                    920M  8.8M  911M   1% /run
tmpfs                    920M     0  920M   0% /sys/fs/cgroup
/dev/sda1               1014M  142M  873M  14% /boot
tmpfs                    184M     0  184M   0% /run/user/0
[root@saumitra-centos75x64-01 ~]#
[root@saumitra-centos75x64-01 ~]#
[root@saumitra-centos75x64-01 ~]#
[root@saumitra-centos75x64-01 ~]#
[root@saumitra-centos75x64-01 ~]# lvextend -L +512M /dev/mapper/centos-root
  Size of logical volume centos/root changed from 13.39 GiB (3429 extents) to 13.89 GiB (3557 extents).
  Logical volume centos/root successfully resized.
[root@saumitra-centos75x64-01 ~]#
```

However, the `centos-root` volume still doesn't show any change in its size.



This is because, we also need to tell the filesystem to occupy all the remaining free space available in the `centos-root` LV. The filesystem present in the root directory is `xfs`. Thus, we perform the extension with the help of `xfs_growfs /dev/centos/root` command.

This grows our LV size as follows:

```
[root@saumitra-centos75x64-01 ~]# lvdisplay
  --- Logical volume ---
  LV Path                /dev/centos/swap
  LV Name                swap
  VG Name                centos
  LV UUID                GdqE4v-y9sE-bb3D-SnX7-IKOq-XPwr-YkmPOy
  LV Write Access        read/write
  LV Creation host, time localhost, 2020-11-10 00:20:00 -0500
  LV Status              available
  # open                 2
  LV Size                1.60 GiB
  Current LE             410
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:1

  --- Logical volume ---
  LV Path                /dev/centos/root
  LV Name                root
  VG Name                centos
  LV UUID                CZC0jJ-G4Ds-Gfo8-6css-N8Z4-8QCp-0qlFGJ
  LV Write Access        read/write
  LV Creation host, time localhost, 2020-11-10 00:20:00 -0500
  LV Status              available
  # open                 1
  LV Size                13.89 GiB
  Current LE             3557
  Segments               2
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:0

[root@saumitra-centos75x64-01 ~]#
```

Creating new LVs-

With the remaining 1.5GB of free space in our VG, we can create a new LV. This is done with the help of the `lvcreate` command.

```
lvcreate VG_Name -l space_to_be_allocated -n name_of_LV
```

```
[root@saumitra-centos75x64-01 ~]# lvcreate centos -l 100%FREE -n lv_logs
  Logical volume "lv_logs" created.
[root@saumitra-centos75x64-01 ~]#
```

Similar to the previous step, here too we need to initialize a filesystem to the newly created LV. As is the trend, we use `xfs`.

```
[root@saumitra-centos75x64-01 ~]# mkfs.xfs -f -L XFS -b size=1024 /dev/mapper/centos-lv_logs
meta-data=/dev/mapper/centos-lv_logs isize=512    agcount=4, agsize=392192 blks
         =                       sectsz=512   attr=2, projid32bit=1
         =                       crc=1        finobt=0, sparse=0
data     =                       bsize=1024   blocks=1568768, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0 ftype=1
log      =internal log           bsize=1024   blocks=10240, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =none                   extsz=4096   blocks=0, rtextents=0
```

Now that we have a filesystem created, we need to mount this newly created LV. For that we need to create a mount point, and then mount the LV on it.

```
[root@saumitra-centos75x64-01 ~]# mkdir -p /mnt/extra/logs
[root@saumitra-centos75x64-01 ~]# mount /dev/mapper/centos-lv_logs /mnt/extra/logs
[root@saumitra-centos75x64-01 ~]# df -h
Filesystem                  Size  Used Avail Use% Mounted on
devtmpfs                    908M     0  908M   0% /dev
tmpfs                       920M     0  920M   0% /dev/shm
tmpfs                       920M  8.9M  911M   1% /run
tmpfs                       920M     0  920M   0% /sys/fs/cgroup
/dev/mapper/centos-root      14G  1.7G   13G  12% /
/dev/sda1                  1014M  173M  842M  18% /boot
tmpfs                       184M     0  184M   0% /run/user/0
/dev/mapper/centos-lv_logs  1.5G  4.5M  1.4G   1% /mnt/extra/logs
[root@saumitra-centos75x64-01 ~]#
```

4. Editing `fstab`

While we indeed have successfully mounted our newly created LV, it will not be mounted automatically the next time our server restarts. This is because our OS has no record of this new LV in its file system table (`fstab`). In order to make our system automatically mount our LV, we need to edit the `fstab` file in our system.

First, we need to refer to our LV block using a UUID. This can be obtained using the `lsblk` command.

```
root@saumitra-centos75x64-01:~
[root@saumitra-centos75x64-01 ~]# blkid /dev/mapper/centos-lv_logs
/dev/mapper/centos-lv_logs: LABEL="XFS" UUID="f1242cb7-f8eb-4a84-a300-7a71e1f13fe4" TYPE="xfs"
[root@saumitra-centos75x64-01 ~]#
```

We create a backup of our `fstab` file in case something goes wrong. And we unmount our LV in order to test it later.

```
[root@saumitra-centos75x64-01 ~]# cp /etc/fstab /etc/fstab.bk
[root@saumitra-centos75x64-01 ~]# umount /mnt/extra/logs/
```

Then, we edit the `fstab` file. We mention our LV with its UUID, we mention the mount point, the default flags and the priority. The number 2 indicates that this LV is given a low priority when mounting.(As the root LVs are more important and must be given a high priority).

```
root@saumitra-centos75x64-01:~
  GNU nano 2.3.1                                          File: /etc/fstab

#
# /etc/fstab
# Created by anaconda on Tue Nov 10 00:20:02 2020
#
# 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/centos-root /                       xfs     defaults        0 0
UUID=cb7f49c0-3834-4815-b87e-c35459d1635e /boot                   xfs     defaults        0 0
/dev/mapper/centos-swap swap                    swap    defaults        0 0
UUID=f1242cb7-f8eb-4a84-a300-7a71e1f13fe4 /mnt/extra/logs xfs defaults 0 2
```

In order to check if our `fstab` edit has worked, we try the `mount -a` command. This command tells linux to mount all the drives mentioned in its `fstab` file.

The `df -h` command is used to check the status of the mounted drives.

```
[root@saumitra-centos75x64-01 ~]# df -h
Filesystem                Size  Used Avail Use% Mounted on
devtmpfs                  908M     0  908M   0% /dev
tmpfs                     920M     0  920M   0% /dev/shm
tmpfs                     920M  8.8M  911M   1% /run
tmpfs                     920M     0  920M   0% /sys/fs/cgroup
/dev/mapper/centos-root    14G  1.7G   13G  12% /
/dev/sda1                1014M  173M  842M  18% /boot
tmpfs                     184M     0  184M   0% /run/user/0
/dev/mapper/centos-lv_logs  1.5G  8.1M  1.5G   1% /mnt/extra/logs
[root@saumitra-centos75x64-01 ~]#
```

5. Snapshots

LVM also allows us to take snapshots of our LVs in case we make a change that breaks some files, and a need arises to roll back to the previous status of the files.

In order to create a snapshot, we again have to use the `lvcreate` command, but with some options. Creating a snapshot requires us to have unclaimed space.

```
lvcreate path_to_LV -L size -s -n name_of_snapshot
```

In order to view all the existing snapshots, we use the `lvs` command. A default snapshot is created the moment we create a new LV. All the snapshots that we manually create are also displayed here. Their origin parameter tells us the difference.

Similar to LVs, we can also mount snapshots to a mount point. this is how we can recover the last system state.

In order to recover data from the snapshot to the original directory from where the snapshot was taken, we use the following command:

```
lvconvert --merge path_to_snapshot
```

This rolls back the origin volume of the snapshot to the state the snapshot was in. In order to refresh the LVs so that our change can be seen, we need to deactivate the LV and then reactivate it. This is done by the follwing 2 commands:

```
lvchange -an path_to_LV lvchange -ay path_to_LV
```

Now we remount the drive in order to see the changes.