# CST8177 – Linux II

Quota, LVM

# Topics

- Quota
- LVM (Logical Volume Manager)

# Quotas

- https://access.redhat.com/knowledge/docs/en-US/Red_Hat_Enterprise_Linux/6/html/Storage_Administration_Guide/ch-disk-quotas.html
- Quotas give us the ability to keep track of users' disk usage: both blocks (disk space) and inodes (number of files)
- `quota` rpm must be installed
- For both blocks and inodes, quotas allow hard limits and soft limits:
  - Soft limit: user is allowed to exceed a soft limit, but they will be warned, and after a grace period, they cannot increase usage
  - Hard limit: user is never allowed to exceed the hard limit
- We enable quotas for a file system
- Quotas can be applied to users and/or groups
- System administrator can report on all users' disk usage status
- Each user can see their own disk usage status (quota information)

# Turning quotas on (and off)

- Example: enabling quotas on /home (separate /home filesystem)
  - In `/etc/fstab`, add the `usrquota,grpquota` mount options for the file system mounted on the /home mount point
  - Initialize the quota database files for `/home` with the command

```
quotacheck -cug /home
```

    - `c`: don't read quota files, create new quota database files
    - `u`: do user quotas
    - `g`: do group quotas
  - Turn quotas on
  - `quotaon -vaug # turn quotas on`
    - `v`: display a message for each filesystem affected
    - `a`: turn quotas on for all automatically mounted file systems according to `/etc/fstab`
    - `u:` user quotas
    - `g:` group quotas
  - `repquota -a    # report on quotas`
  - Turn quotas off
  - `quotaoff -vaug # turn quotas off`
  - quotaoff -vaug; quotacheck -vaug; quotaon -vaug  #single user mode

# Setting Quotas

- To set a quota for a user, as root

```
edquota username
```

  - where
    - you'll see (example) DO NOT edit blocks or inodes, just soft and hard limits!

```
Disk quotas for user tgk (uid 107):
Filesystem       blocks      soft      hard     inodes     soft     hard
/dev/sda8           108      1000      2000          1        0        0
```

or this command can be used in scripts

```
setquota -u username soft hard isoft ihard  fs
```

  - where
    - username is the name of the user
    - soft is the block soft limit
    - hard is the block hard limit
    - isoft is the inode soft limit
    - ihard is the inode hard limit
    - fs is the file system mount point (e.g. /home)

# Quota Grace Period

- To set the grace period for all users

```
edquota -t     # edit grace period
```

  - where you'll see something like this (note units)

```
Grace period before enforcing soft limits for users:
Time units may be: days, hours, minutes, or seconds
Filesystem                                     Block grace period     Inode grace period
/dev/mapper/VolGroup00-LogVol00                      8days                   8days
```

- To set the grace period for an individual user

```
edquota -T tgk
```

  - where you'll see something like this (note units)

```
Times to enforce softlimit for user tgk (uid 498):
Time units may be: days, hours, minutes, or seconds
Filesystem                          block grace        inode grace
/dev/mapper/VolGroup00-LogVol00         unset            unset
```

# quota and repquota commands

- individual users can check their individual quota status with `quota` command:
  - shows
    - block usage and limits
    - inode usage and limits
    - remainder on grace period if over soft limit
- System administrator can print report of all users quota status (see also `warnquota`):
  - `repquota -a`
  - shows for each user what they've used, soft limits, hard limits, and remainder of grace periods if that user has entered one of their grace periods

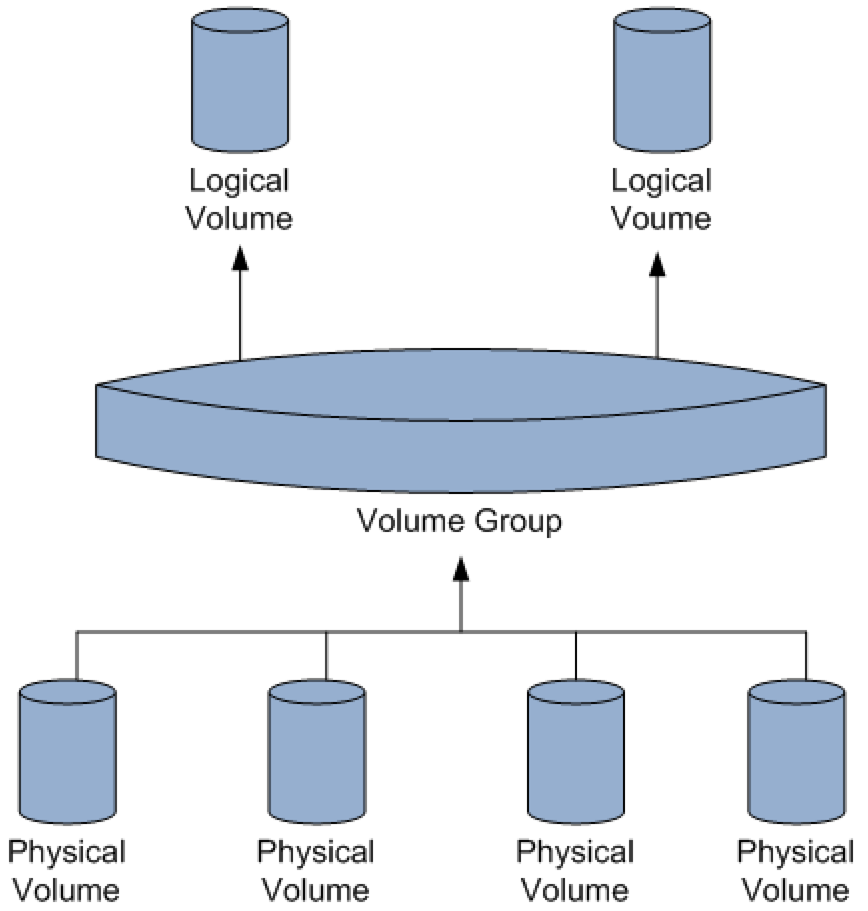
# LVM basics

- Logical Volume Manager
- LVM tutorial:
  - http://www.howtoforge.com/linux_lvm
- disk partitions are physical volumes
- one or more physical volumes forms a volume group
- a volume group can be divided into logical volumes
- We create file systems on the logical volumes

# Extents

- With LVM, we deal with space in logical and physical volumes in terms of "extents"
- Logical Volumes: LE or Logical Extents
- Physical Volumes: PE or Physical Extents
- Extents are the little pieces of space that can be managed: divided up into volumes, added to volumes

# LVM Logical Volume Components



# Adding disks and LVM

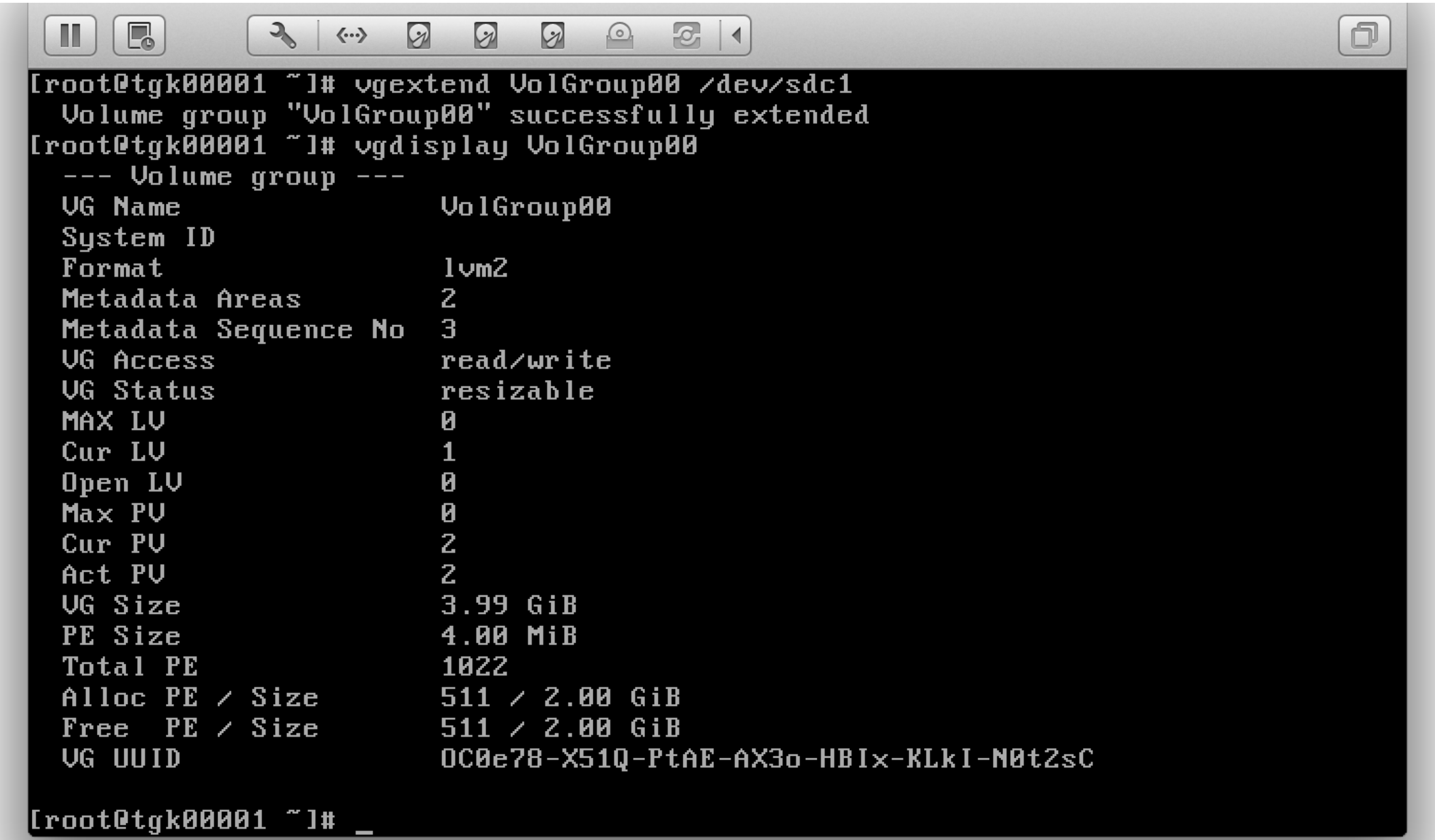
- Let's explore LVM by adding a disk and putting it under LVM control
- We'll create a file system on that logical volume
- Then we'll add yet another disk and grow that file system so it uses the added space
- physical volume commands /sbin/pv*
- volume group commands /sbin/vg*
- **logical volume commands /sbin/lv***
- **Examples**
  - `lvdisplay   # show logical volumes`
  - `pvdisplay   # show physical volumes`

# create the PV and VG and LV

- create a partition /dev/sdb1
- pvcreate /dev/sdb1
  - create the physical volume
- vgcreate VolGroup00 /dev/sdb1
  - add /dev/sdb1 physical volume to a new volume group called VolGroup00
- lvcreate –l 100%FREE –n LogVol00 VolGroup00
  - use 100% of the free space of VolGroup00 to create a new logical volume named LogVol00
  - creates /dev/VolGroup00/LogVol00 on which we can make a filesystem
- mkfs –t ext4 /dev/VolGroup00/LogVol00

# Growing a file system

- add yet another disk (say /dev/sdc)
- partition /dev/sdc to create /dev/sdc1
- Create the new physical volume
  - pvcreate /dev/sdc1
- Add this new physical volume to a volume group (in this case VolGroup00):
  - vgextend VolGroup00 /dev/sdc1
- See how many free extents (Free PE) are available in this volume group (VolGroup00)
  - vgdisplay VolGroup00

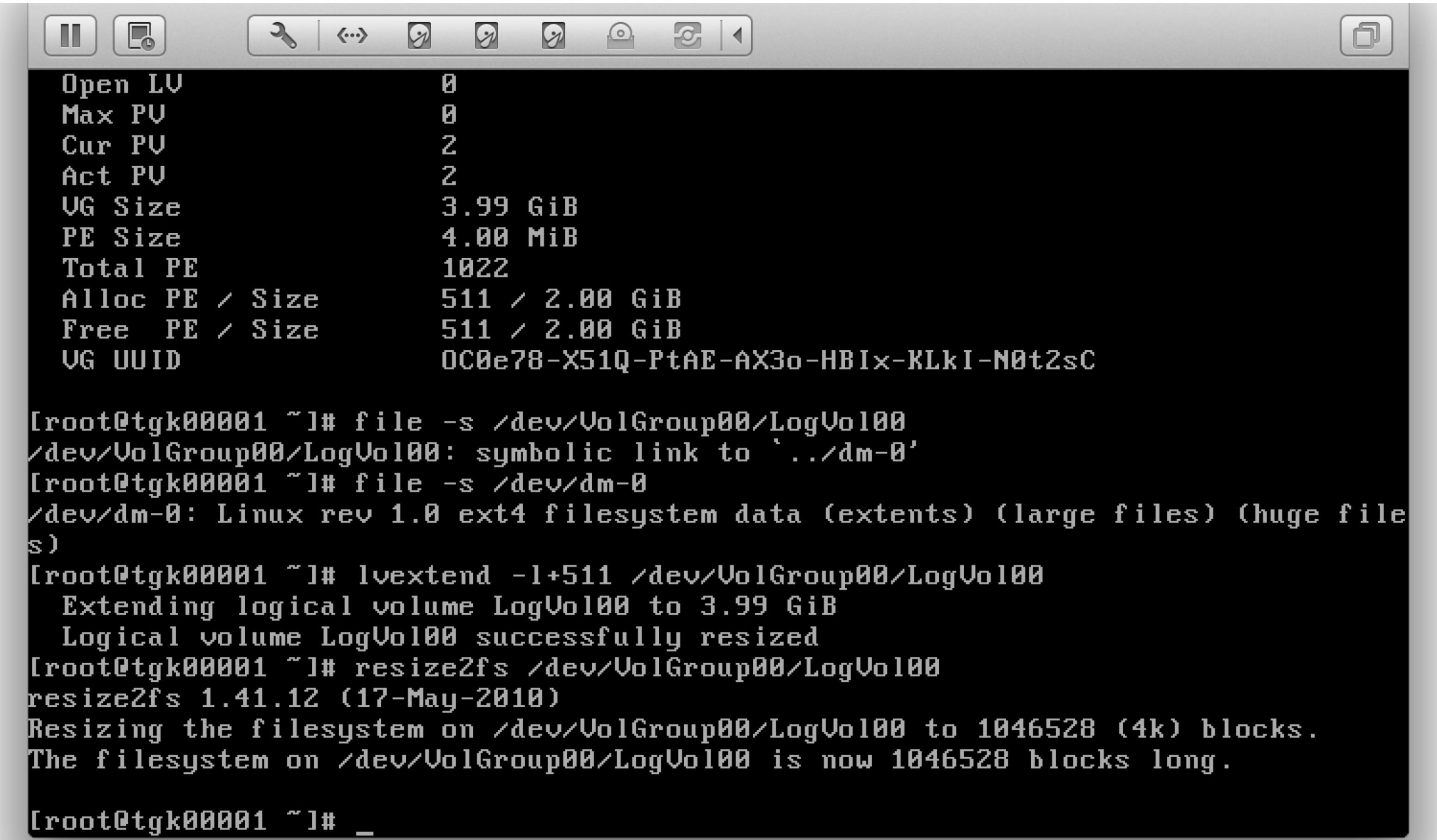


```
[root@tgk00001 ~]# vgextend VolGroup00 /dev/sdc1
  Volume group "VolGroup00" successfully extended
[root@tgk00001 ~]# vgdisplay VolGroup00
  --- Volume group ---
  VG Name               VolGroup00
  System ID
  Format                lvm2
  Metadata Areas        2
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                1
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.99 GiB
  PE Size               4.00 MiB
  Total PE              1022
  Alloc PE / Size       511 / 2.00 GiB
  Free  PE / Size       511 / 2.00 GiB
  VG UUID               OC0e78-X51Q-PtAE-AX3o-HBIx-KLkI-N0t2sC

[root@tgk00001 ~]# _
```

# Growing a file system (cont'd)

- Suppose the previous "vgdisplay" command showed that VolGroup00 had 511 free extents ("Free PE") and we use them all:
  - lvextend –l+511 /dev/VolGroup00/LogVol00
- Now LogVol00 is bigger, but the filesystem we created before is still the same size.
- Grow the filesystem (ext4) to fill the added space:
  - resize2fs /dev/VolGroup00/LogVol00
  - Now the filesystem is bigger, occupying the new disk space too



```
  Open LV               0
  Max PV                0
  Cur PV                2
  Act PV                2
  VG Size               3.99 GiB
  PE Size               4.00 MiB
  Total PE              1022
  Alloc PE / Size       511 / 2.00 GiB
  Free  PE / Size       511 / 2.00 GiB
  VG UUID               OC0e78-X51Q-PtAE-AX3o-HBIx-KLkI-N0t2sC

[root@tgk00001 ~]# file -s /dev/VolGroup00/LogVol00
/dev/VolGroup00/LogVol00: symbolic link to `../dm-0'
[root@tgk00001 ~]# file -s /dev/dm-0
/dev/dm-0: Linux rev 1.0 ext4 filesystem data (extents) (large files) (huge file
s)
[root@tgk00001 ~]# lvextend -l+511 /dev/VolGroup00/LogVol00
  Extending logical volume LogVol00 to 3.99 GiB
  Logical volume LogVol00 successfully resized
[root@tgk00001 ~]# resize2fs /dev/VolGroup00/LogVol00
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/VolGroup00/LogVol00 to 1046528 (4k) blocks.
The filesystem on /dev/VolGroup00/LogVol00 is now 1046528 blocks long.

[root@tgk00001 ~]# _
```