

RHEL - Add Disks & Expand Logical Volume

Clark Mercer 2017-05-04

Summary:

This document walks through adding physical disks to a physical Red Hat Enterprise Linux (RHEL) server. The disks are added in a RAID mirror for fault tolerance. The disks are then used to expand the "/data" partition of the RHEL server. A similar process can be used to add space to any LVM volume on any Linux server (physical or virtual).

Document Outline and Summary of the Process:

- Add new physical disks to the server
- Create a partition on the new disk
- Create a Physical Volume on the new partition
- Expand the Volume Group with the new Logical Volume
- Resize the Logical Volume
- Resize the filesystem
- Check the consistency

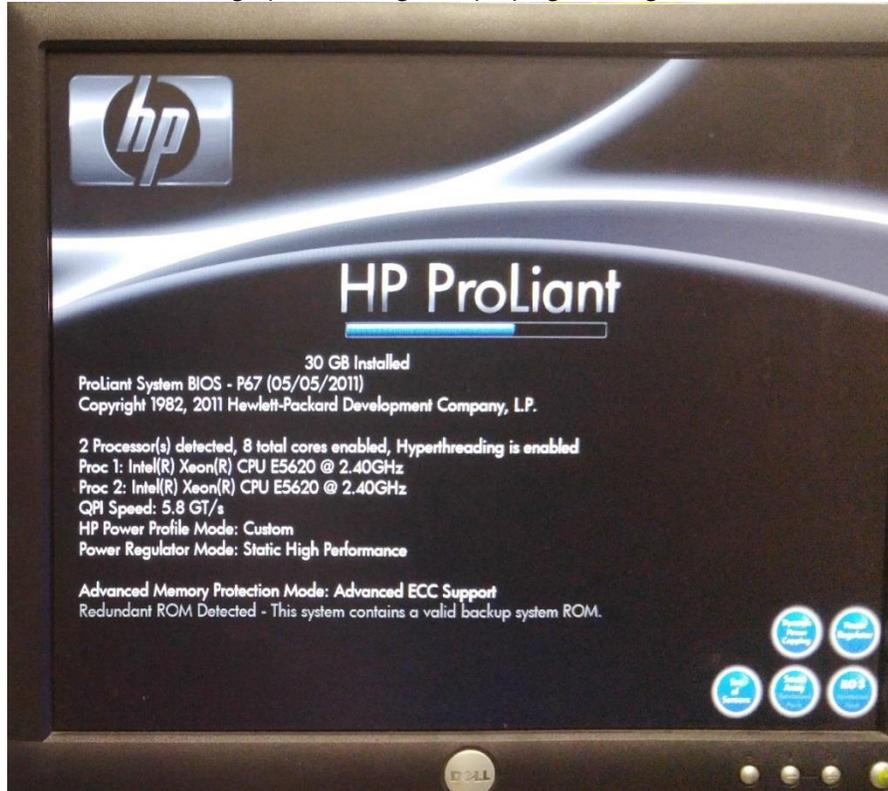
The diagram that I drew below explains the Linux LVM and lists some helpful commands at each step. The process from left to right is representative of what I demonstrate in this document.

Logical Volume Manager (LVM)

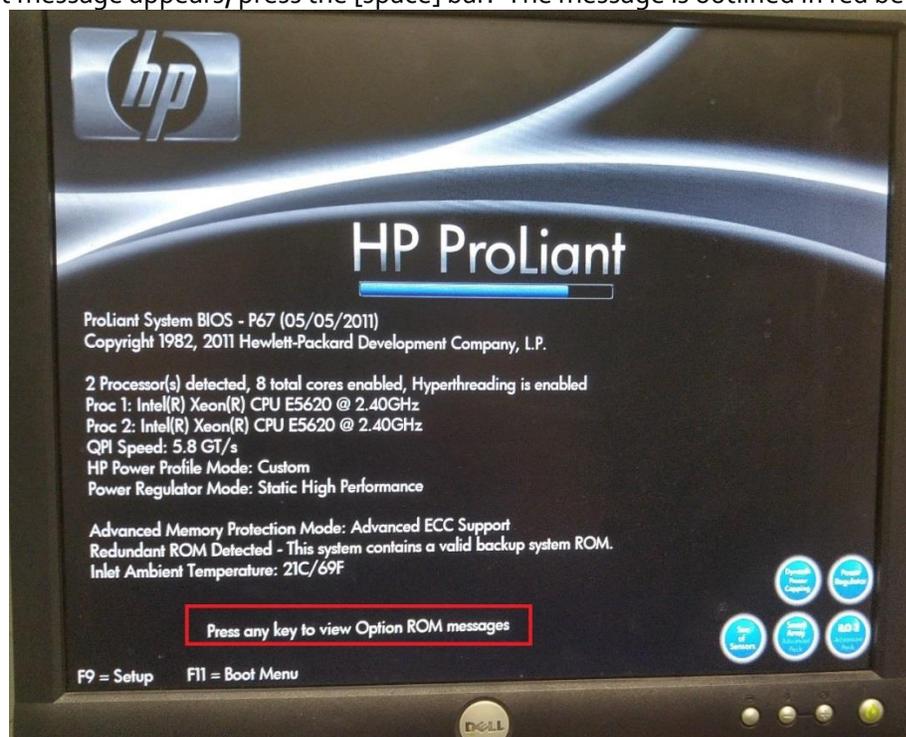
ITEM	HARD DRIVES	PARTITIONS PHYSICAL VOLUMES	VOLUME GROUPS	LOGICAL VOLUMES
DIAGRAM	<p>SINGLE DISK, RAID-DISK, OR VIRTUALMACHINE-DISK</p>	<p>PHYSICAL VOLUMES ARE CREATED ON ANY PARTITION THAT WILL BE PART OF A VOLUME GROUP</p>	<p>EACH PHYSICAL VOLUME IS GROUPED INTO ONE OR MORE VOLUME GROUPS</p>	<p>AS THE FINAL STEP LOGICAL VOLUMES ARE CREATED WITHIN A VOLUME GROUP</p>
COMMANDS		<p>pvs pvdisplay pvcreate pvresize pv[tab] [tab]</p>	<p>vgs vgdisplay vgcreate vgextend vg[tab] [tab]</p>	<p>lvs lvdisplay lvcreate lvextend lv[tab] [tab]</p>

I. Add new physical disks to the server

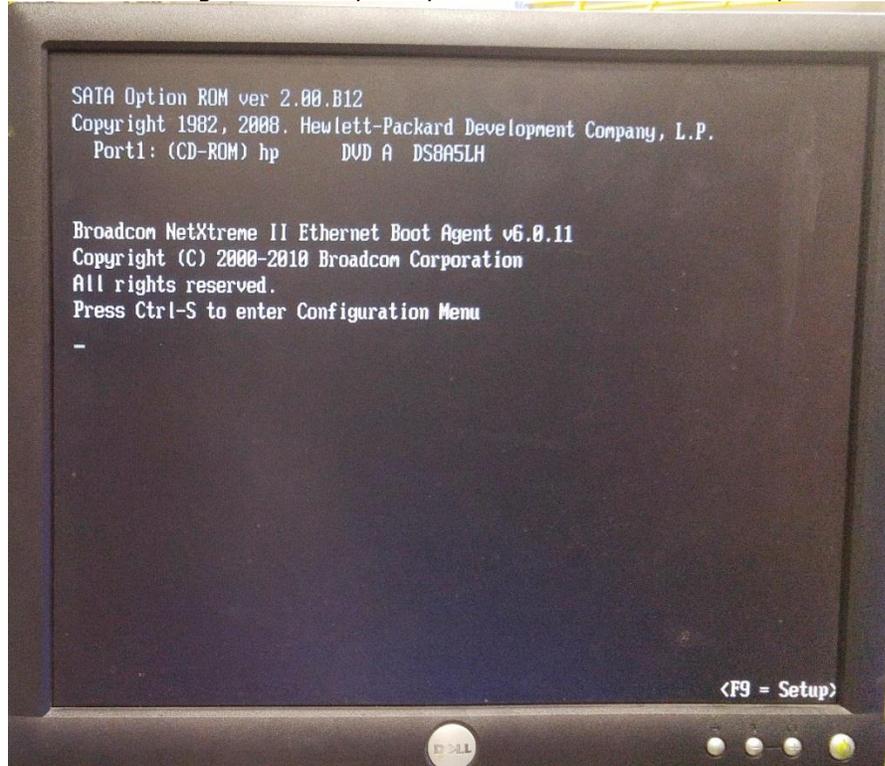
1. Power off the server and insert the two new disks into the 3rd and 4th slots.
2. Push firmly on the disks so that they are properly seated.
3. Power on the server.
4. While the server is booting up, it will begin displaying messages like this:



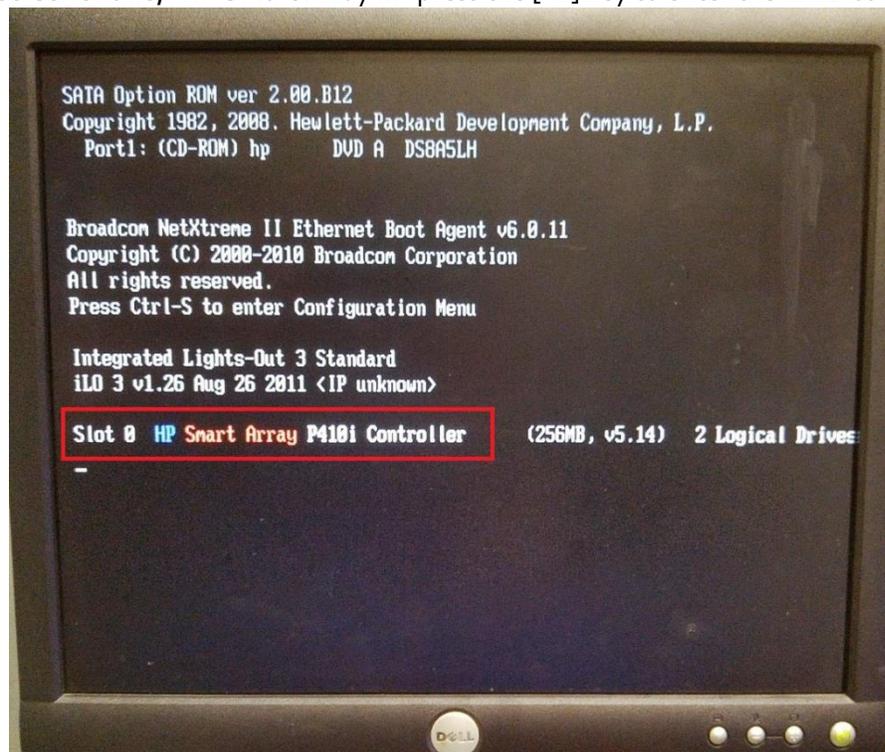
Watch the bottom of the screen for the message, "Press any key to view Option ROM messages". Once that message appears, press the [space] bar. The message is outlined in red below:



- The system now begins displaying various other messages that it would not have displayed otherwise. Get ready to tap the [F8] key, but do not do so yet. If you tap [F8] too early, it will enter the iLO Server Management utility and you'll have to start over from power off.

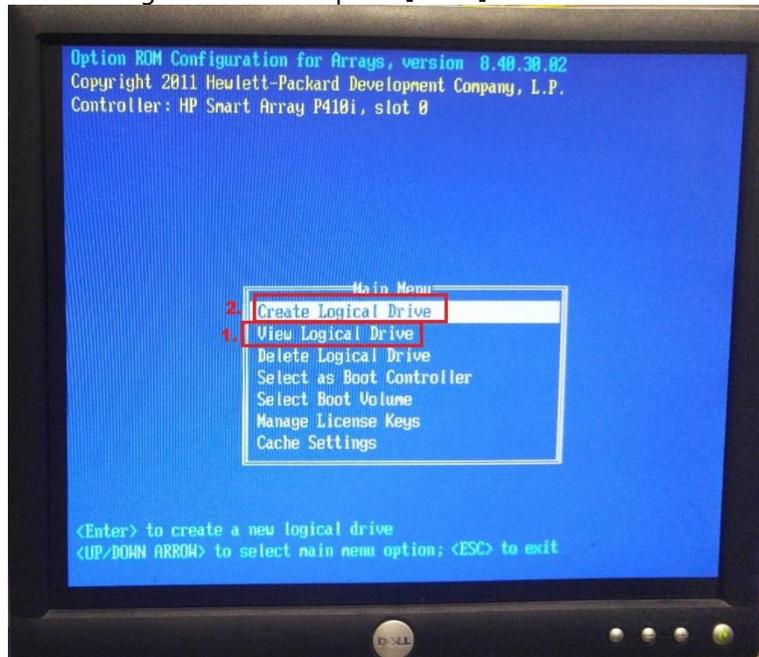


- Once the screen shows, "HP Smart Array ..." press the [F8] key to enter the RAID controller.

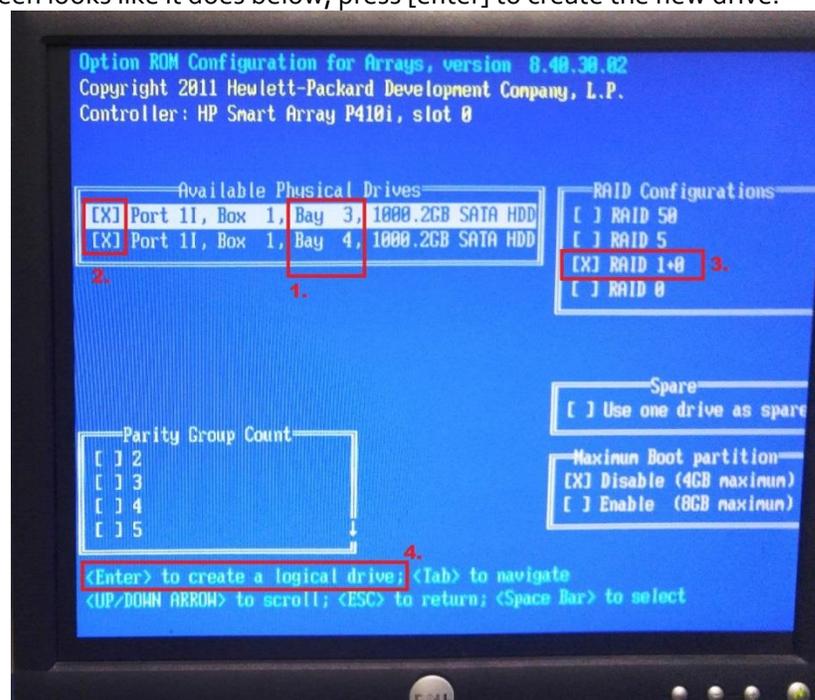


- From within the RAID controller to navigate around you will use the arrow keys and [tab]. The [space] bar checks boxes, function keys perform special operations, and [enter] and [esc] are used as well. If in doubt, read the bottom of the screen for navigation tips.

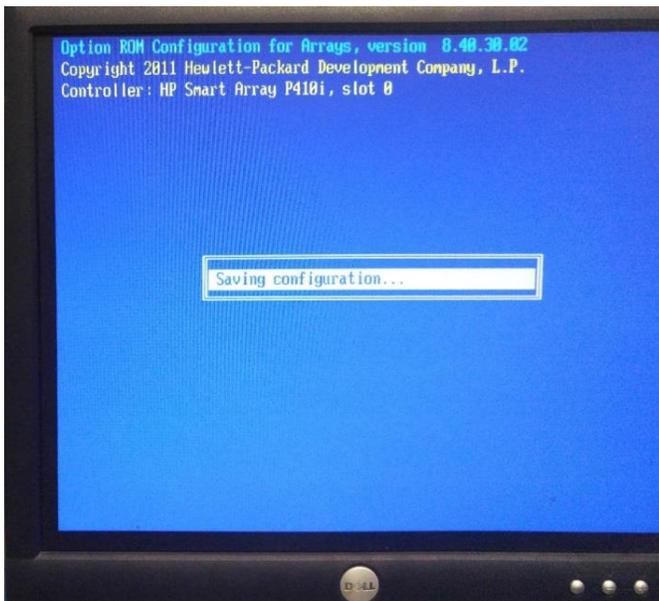
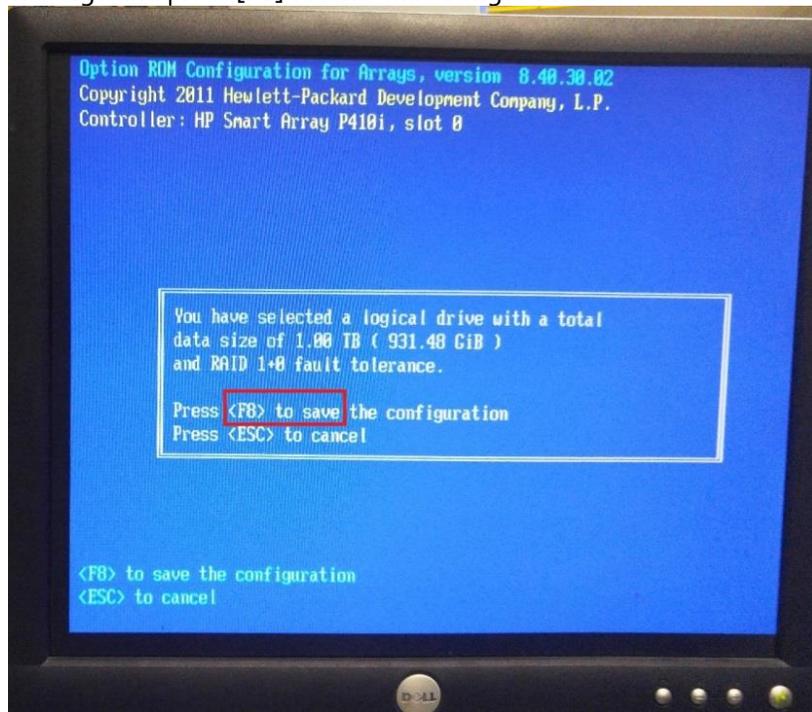
8. First select "View Logical Drive" to ensure everything looks as you expect. RAID disks or virtual/logical disks are displayed. If you only have one mirror, you would expect to only see one disk that shows "RAID 1+0". For the purposes of this document, RAID 1+0 should just be interpreted as a mirror or RAID₁ (RAID 1+0 or RAID 10 is outside the scope of this document).
9. After viewing the logical drive, press the [esc] key to return to the main menu.
10. Arrow up to "Create Logical Drive" and press [enter].



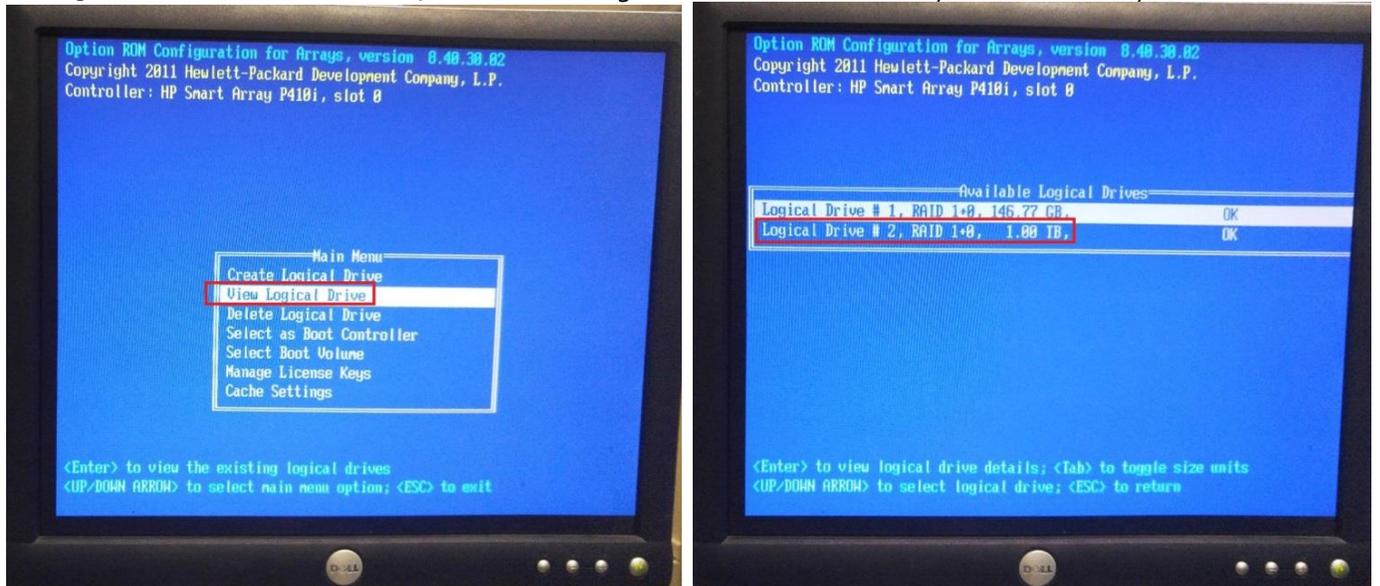
11. From the next screen the available disks will be shown as well as which physical bay they are in. This will enable you to visually identify on the front of the server that the disks are the ones you recently inserted.
12. Navigate with [tab] and the arrow keys to ensure that the box is checked on the left of each of the disks using [space] bar, and ensure the box for "RAID 1+0" is checked as well.
13. Once the screen looks like it does below, press [enter] to create the new drive.



14. Read the message and press [F8] to save the configuration.

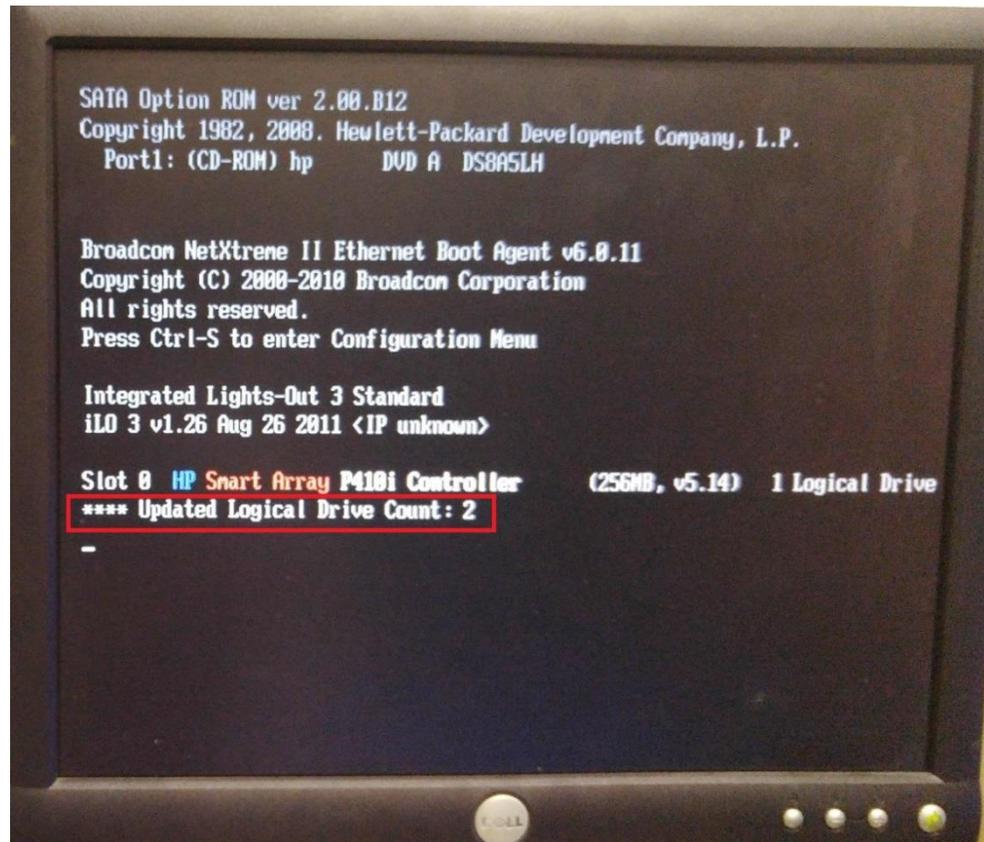


15. Now from the main menu, select "View Logical Drive" to ensure that you see the newly created mirror.



16. When you're happy with the result, from the main menu press the [esc] key to exit the RAID controller.

17. You may now notice that the controller displays the message, "**** Updated Logical Drive Count: 2" like this:



18. You may now boot into the operating system to complete the remaining steps that follow.

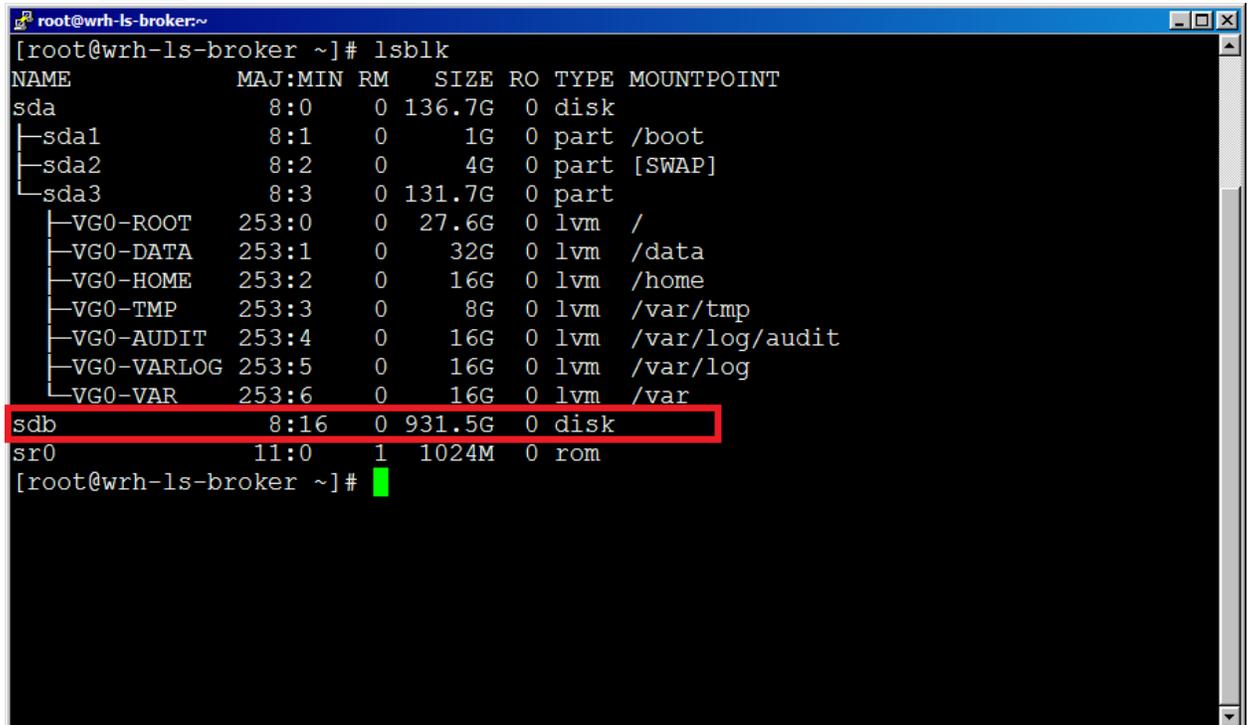
II. Create a partition on the new disk

Commands for this section:

```
fdisk      # list, create, delete partitions
lsblk      # list block devices
df -h      # list logical volume capacities
```

1. In order to perform the disk and LVM steps, you must first become root, so do so now.
2. Now use the following command to identify which block device is the new disk.

```
lsblk
```



```
root@wrh-ls-broker:~# lsblk
NAME                                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda                                  8:0      0 136.7G 0 disk
├─sda1                               8:1      0    1G 0 part /boot
├─sda2                               8:2      0    4G 0 part [SWAP]
└─sda3                               8:3      0 131.7G 0 part
   ├─VG0-ROOT                       253:0    0  27.6G 0 lvm  /
   ├─VG0-DATA                       253:1    0   32G 0 lvm  /data
   ├─VG0-HOME                       253:2    0   16G 0 lvm  /home
   ├─VG0-TMP                        253:3    0    8G 0 lvm  /var/tmp
   ├─VG0-AUDIT                     253:4    0   16G 0 lvm  /var/log/audit
   ├─VG0-VARLOG                    253:5    0   16G 0 lvm  /var/log
   └─VG0-VAR                        253:6    0   16G 0 lvm  /var
sdb                                  8:16     0 931.5G 0 disk
sr0                                  11:0     1 1024M 0 rom
[root@wrh-ls-broker ~]#
```

3. In the above picture, you can see the new disk is sdb. The full device name is "/dev/sdb".
4. We will now create a partition on the new disk using the following command:

```
fdisk /dev/sdb
```
5. It is important to note that any configuration changes made in fdisk are only in memory and are not actually written to disk until you press the 'w' key. If you make a mistake, just quit out with the 'q' key. This way none of your changes will take effect and you can start over from scratch. For this reason, exercise caution before using the 'w' menu option as it will permanently write the changes to disk.
6. Now that you are in fdisk, tap 'm' to display the menu list of options.
7. Now tap 'p' to print the current partitions.
8. You should see that there are currently no partitions present.

```

root@wrh-ls-broker:~
m print this menu
n add a new partition
o create a new empty DOS partition table
p print the partition table
q quit without saving changes
s create a new empty Sun disklabel
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)

Command (m for help): p

Disk /dev/sdb: 1000.2 GB, 1000171331584 bytes, 1953459632 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: dos
Disk identifier: 0x4eb8aac9

Device Boot      Start         End      Blocks   Id  System
no partitions

Command (m for help): █

```

9. We are now going to create a new partition by tapping 'n'. You will then be asked a series of questions. Tap [enter] to accept all of the default answers. This will create a single primary partition utilizing the entire disk.
10. Once you've created the partition, tap 'p' to print the partition table.

```

root@wrh-ls-broker:~
p primary (0 primary, 0 extended, 4 free)
e extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-1953459631, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-1953459631, default 1953459631):
Using default value 1953459631
Partition 1 of type Linux and of size 931.5 GiB is set

Command (m for help): p

Disk /dev/sdb: 1000.2 GB, 1000171331584 bytes, 1953459632 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk label type: dos
Disk identifier: 0x4eb8aac9

Device Boot      Start         End      Blocks   Id  System
/dev/sdb1        2048    1953459631    976728792    83  Linux

Command (m for help): █

```

11. We now need to change the partition type from "Linux" to "Linux LVM", so tap 't' for type.
12. Enter '8e' for the "Hex code" and press [enter].

13. Now press 'p' to ensure that the partition shows "Linux LVM"

```
root@wrh-ls-broker:~  
Disk label type: dos  
Disk identifier: 0x4eb8aac9  
  
Device Boot      Start          End      Blocks      Id System  
/dev/sdb1        2048      1953459631    976728792    83  Linux  
  
Command (m for help): t  
Selected partition 1  
Hex code (type L to list all codes): 8e  
Changed type of partition 'Linux' to 'Linux LVM'  
  
Command (m for help): p  
  
Disk /dev/sdb: 1000.2 GB, 1000171331584 bytes, 1953459632 sectors  
Units = sectors of 1 * 512 = 512 bytes  
Sector size (logical/physical): 512 bytes / 4096 bytes  
I/O size (minimum/optimal): 4096 bytes / 4096 bytes  
Disk label type: dos  
Disk identifier: 0x4eb8aac9  
  
Device Boot      Start          End      Blocks      Id System  
/dev/sdb1        2048      1953459631    976728792    8e Linux LVM  
  
Command (m for help): █
```

14. If there is a problem or for some reason you do not want to make any of these changes, tap 'q', but if you're sure everything is correct, tap 'w' to write the changes and exit.

III. Create a Physical Volume on the new partition

Commands for this section:

```
pvs # physical volume summary  
pvdisk # physical volume details  
pvcreate # create a physical volume  
pv[tab] [tab] # list pv commands
```

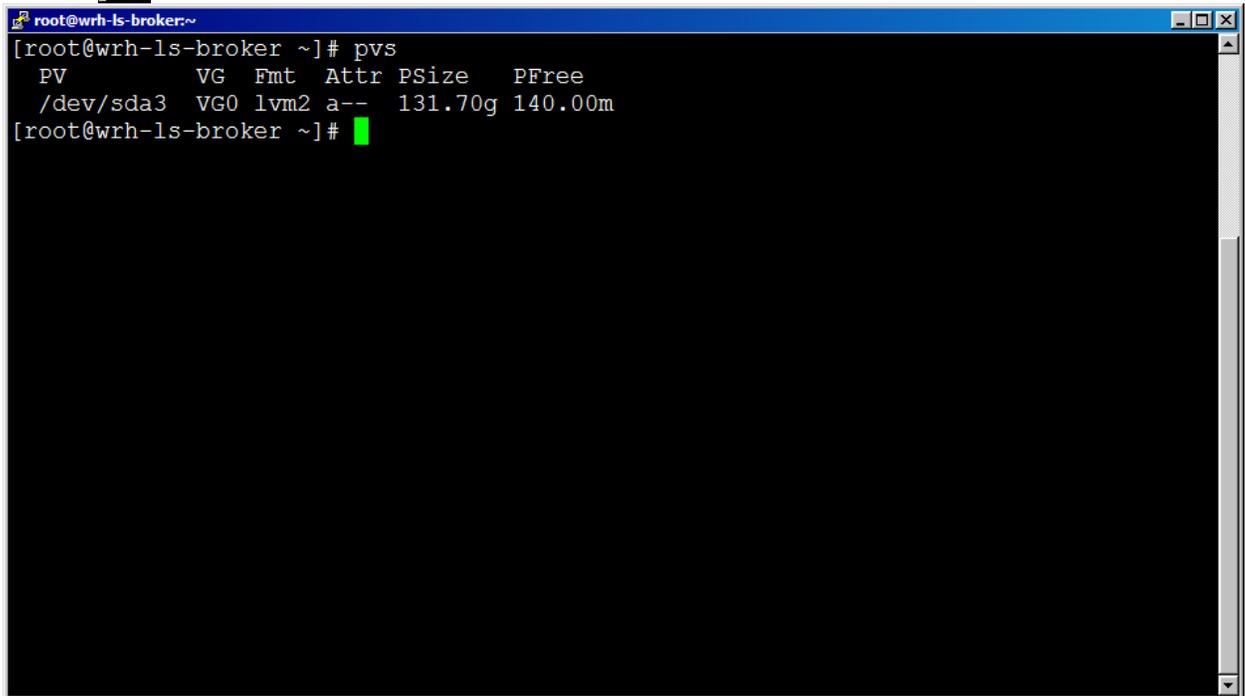
1. View the block devices again with the command:

```
lsblk
```

```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# lsblk  
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sda              8:0    0 136.7G  0 disk  
├─sda1           8:1    0    1G  0 part /boot  
├─sda2           8:2    0    4G  0 part [SWAP]  
└─sda3           8:3    0 131.7G  0 part  
   ├─VG0-ROOT    253:0   0  27.6G  0 lvm  /  
   ├─VG0-DATA    253:1   0   32G  0 lvm  /data  
   ├─VG0-HOME    253:2   0   16G  0 lvm  /home  
   ├─VG0-TMP     253:3   0    8G  0 lvm  /var/tmp  
   ├─VG0-AUDIT   253:4   0   16G  0 lvm  /var/log/audit  
   ├─VG0-VARLOG  253:5   0   16G  0 lvm  /var/log  
   └─VG0-VAR     253:6   0   16G  0 lvm  /var  
sdb              8:16   0 931.5G  0 disk  
└─sdb1           8:17   0 931.5G  0 part  
sr0             11:0    1 1024M  0 rom  
[root@wrh-ls-broker ~]# █
```

2. Like the picture above, you should now see the partition that you just created as sdb1.
3. View the current physical volume summary with the command:

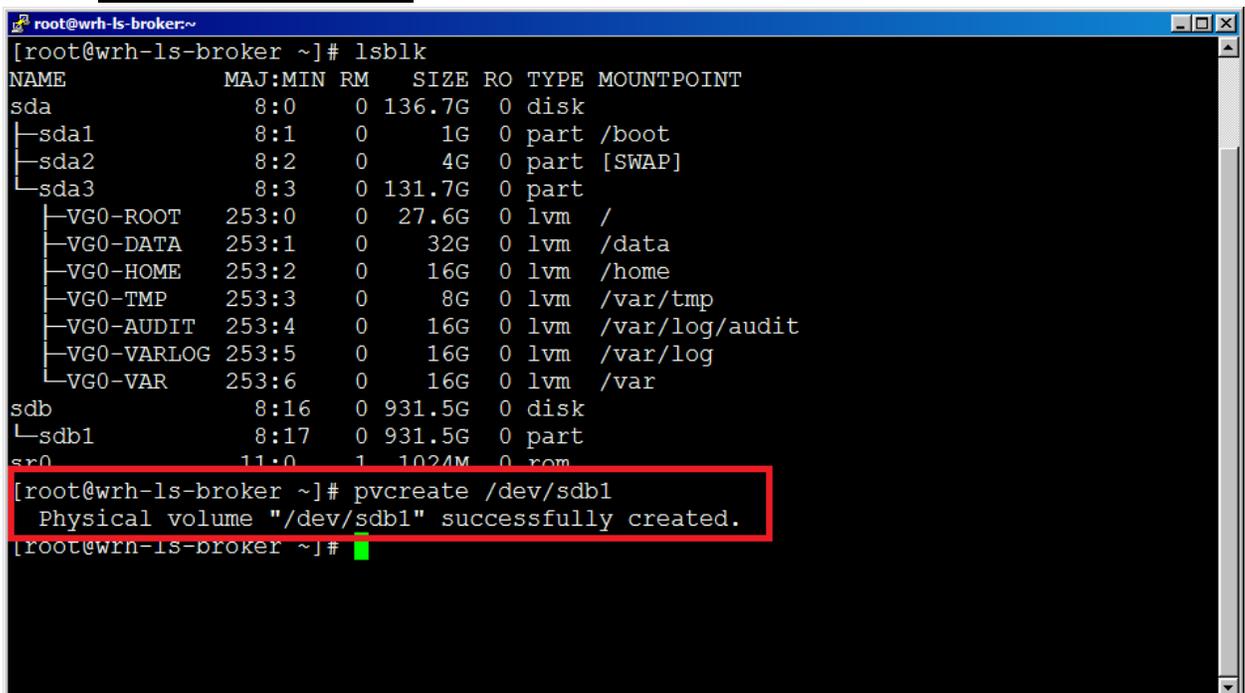
```
pvs
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# pvs  
PV          VG Fmt Attr PSize  PFree  
/dev/sda3  VG0 lvm2 a-- 131.70g 140.00m  
[root@wrh-ls-broker ~]#
```

4. Create a new physical volume from the new partition with the following command:

```
pvcreate /dev/sdb1
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# lsblk  
NAME            MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT  
sda              8:0    0 136.7G  0 disk  
├─sda1           8:1    0    1G  0 part /boot  
├─sda2           8:2    0    4G  0 part [SWAP]  
└─sda3           8:3    0 131.7G  0 part  
   ├─VG0-ROOT    253:0    0  27.6G  0 lvm /  
   ├─VG0-DATA    253:1    0   32G  0 lvm /data  
   ├─VG0-HOME    253:2    0   16G  0 lvm /home  
   ├─VG0-TMP     253:3    0    8G  0 lvm /var/tmp  
   ├─VG0-AUDIT  253:4    0   16G  0 lvm /var/log/audit  
   ├─VG0-VARLOG  253:5    0   16G  0 lvm /var/log  
   └─VG0-VAR     253:6    0   16G  0 lvm /var  
sdb              8:16    0 931.5G  0 disk  
└─sdb1           8:17    0 931.5G  0 part  
sr0             11:0    1   1024M  0 rom  
[root@wrh-ls-broker ~]# pvcreate /dev/sdb1  
Physical volume "/dev/sdb1" successfully created.  
[root@wrh-ls-broker ~]#
```

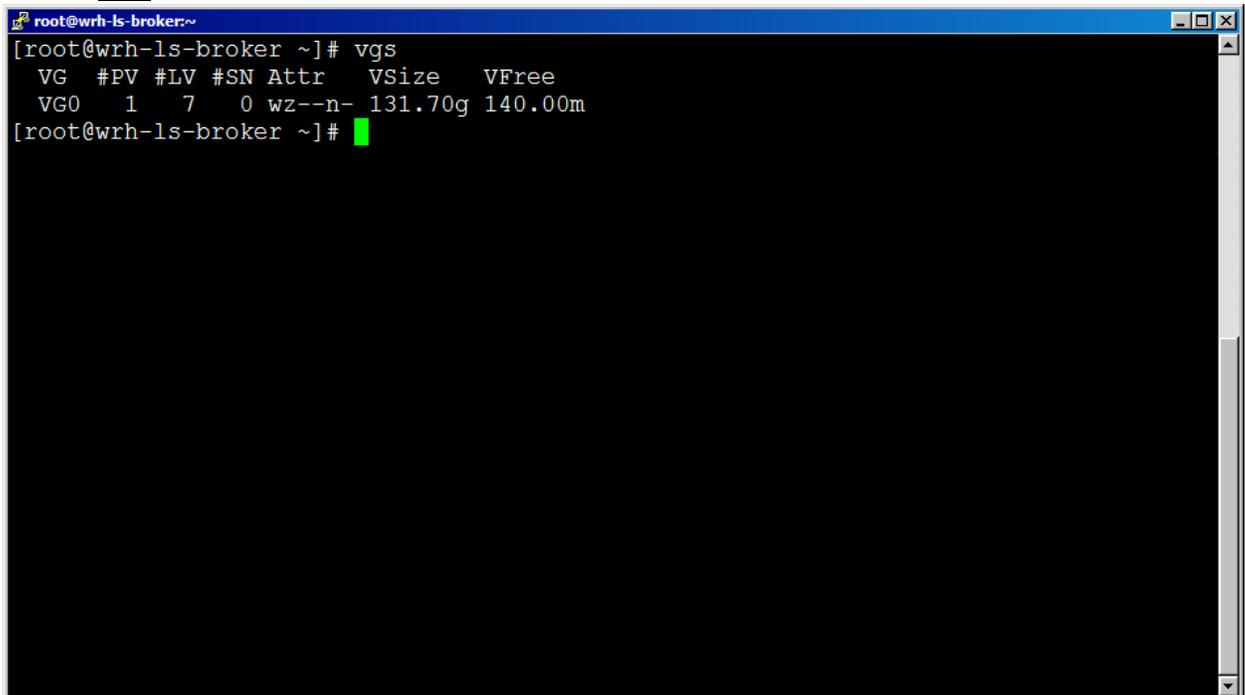
IV. Expand the Volume Group with the new Logical Volume

Commands for this section:

```
vgs # volume group summary
vgdisplay # volume group details
vgextend # extend the volume group
vg[tab] [tab] # list vg commands
```

1. First view the volume group summary using the command:

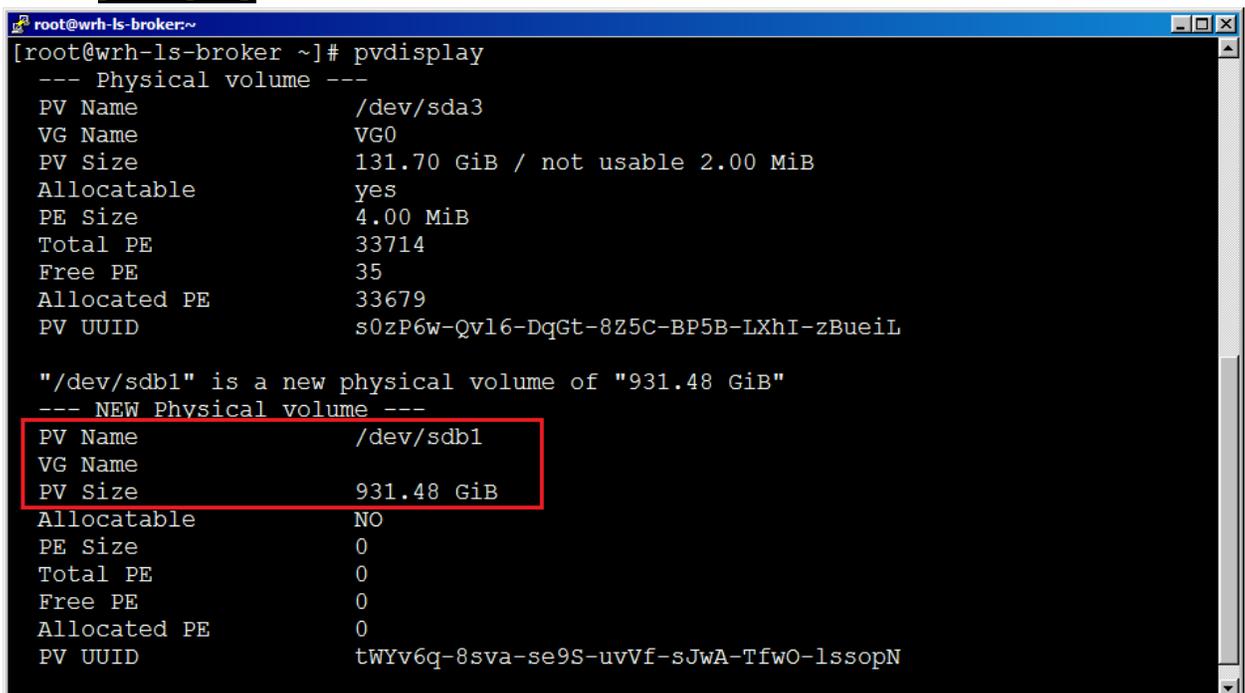
```
vgs
```



```
root@wrh-ls-broker:~
[root@wrh-ls-broker ~]# vgs
  VG #PV #LV #SN Attr   VSize   VFree
  VG0  1   7   0 wz--n- 131.70g 140.00m
[root@wrh-ls-broker ~]#
```

2. Note that the name is "VG0". Now view your physical volumes with the command:

```
pvdisplay
```



```
root@wrh-ls-broker:~
[root@wrh-ls-broker ~]# pvdisplay
--- Physical volume ---
PV Name           /dev/sda3
VG Name           VG0
PV Size           131.70 GiB / not usable 2.00 MiB
Allocatable       yes
PE Size           4.00 MiB
Total PE          33714
Free PE           35
Allocated PE      33679
PV UUID           s0zP6w-Qv16-DqGt-8Z5C-BP5B-LXhI-zBueiL

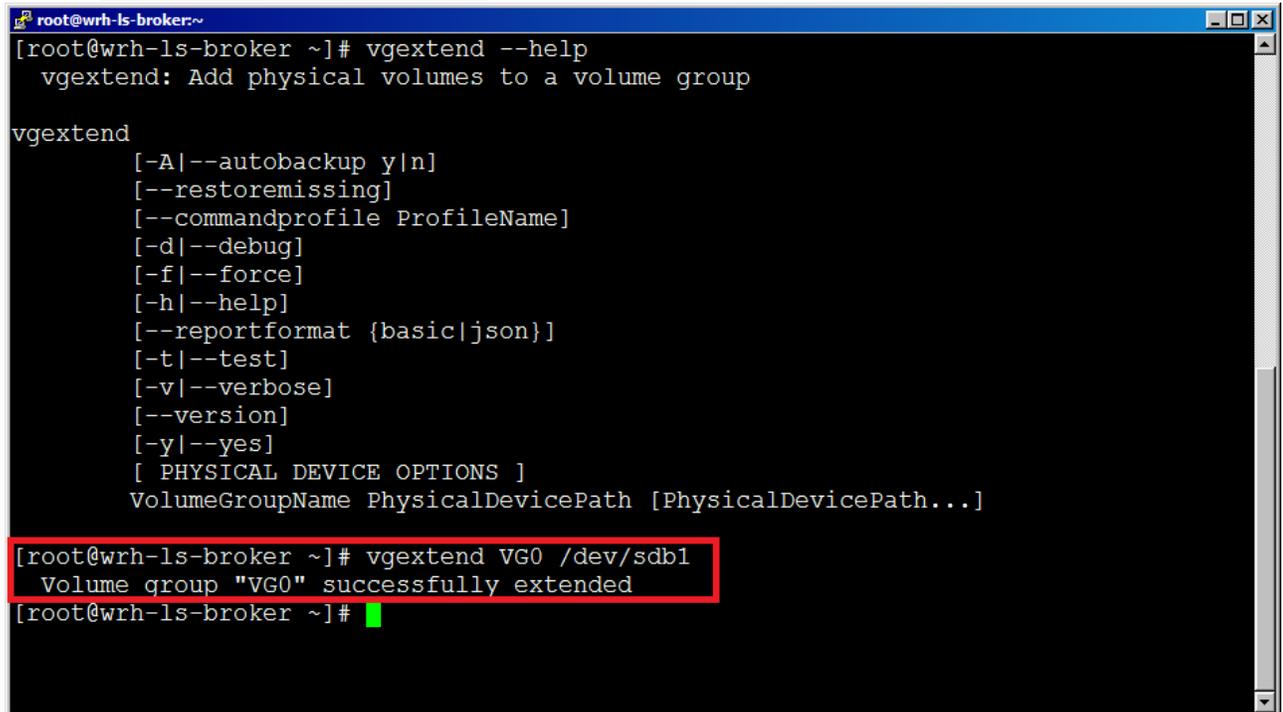
"/dev/sdb1" is a new physical volume of "931.48 GiB"
--- NEW Physical volume ---
PV Name           /dev/sdb1
VG Name
PV Size           931.48 GiB
Allocatable       NO
PE Size           0
Total PE          0
Free PE           0
Allocated PE      0
PV UUID           tWYv6q-8sva-se9S-uvVf-sJwA-TfwO-lssopN
```

3. View the syntax of the next command using the "--help" like this:

```
vgextend --help
```

4. Extend the volume group using the following command:

```
vgextend VG0 /dev/sdb1
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# vgextend --help  
vgextend: Add physical volumes to a volume group  
  
vgextend  
    [-A|--autobackup y|n]  
    [--restoremissing]  
    [--commandprofile ProfileName]  
    [-d|--debug]  
    [-f|--force]  
    [-h|--help]  
    [--reportformat {basic|json}]  
    [-t|--test]  
    [-v|--verbose]  
    [--version]  
    [-y|--yes]  
    [ PHYSICAL DEVICE OPTIONS ]  
    VolumeGroupName PhysicalDevicePath [PhysicalDevicePath...]  
  
[root@wrh-ls-broker ~]# vgextend VG0 /dev/sdb1  
Volume group "VG0" successfully extended  
[root@wrh-ls-broker ~]#
```

5. Now view the volume group to see the new size using the commands:

```
vgs  
vgdisplay
```

V. Resize the Logical Volume

Commands for this section:

```
lvs          # logical volume summary  
lvdisplay   # logical volume details  
lvextend    # extend a logical volume  
lv[tab] [tab] # list lv commands
```

Now that you have a larger volume group, you can expand any of your logical volumes. The following demonstrates using all of the new volume group space to enlarge the 'data' partition.

1. View the current 32GB size of the data partition (physical volume) and details with the commands:

```
lvs  
lvdisplay | less
```

```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# lvs  
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert  
AUDIT VG0 -wi-ao---- 16.00g  
DATA VG0 -wi-ao---- 32.00g  
HOME VG0 -wi-ao---- 16.00g  
ROOT VG0 -wi-ao---- 27.56g  
TMP VG0 -wi-ao---- 8.00g  
VAR VG0 -wi-ao---- 16.00g  
VARLOG VG0 -wi-ao---- 16.00g  
[root@wrh-ls-broker ~]#
```

```
root@wrh-ls-broker:~  
--- Logical volume ---  
LV Path /dev/VG0/DATA  
LV Name DATA  
VG Name VG0  
LV UUID 0FqxIy-KesA-2xSm-cDKp-sYS0-Y3Cd-qz8KLF  
LV Write Access read/write  
LV Creation host, time wrh-ls-erebor2.wrh.noaa.gov, 2017-03-08 15:07:45 +0000  
LV Status available  
# open 1  
LV Size 32.00 GiB  
Current LE 8192  
Segments 1  
Allocation inherit  
Read ahead sectors auto  
- currently set to 256  
Block device 253:1  
  
--- Logical volume ---  
LV Path /dev/VG0/HOME  
LV Name HOME  
VG Name VG0  
LV UUID fbQfdx-eYaV-2KXR-EpaP-Be9c-yr00-1aBe3h  
:
```

2. View the `lvextend` command syntax by using the command:
`lvextend --help`
3. Now extend the `/data` partition (logical volume) to use all remaining space on the volume group with the below command:
`lvextend -l +100%FREE /dev/VG0/DATA`

```
root@wrh-ls-broker~  
[-h|--help]  
[-i|--stripes Stripes [-I|--stripesize StripeSize]]  
{-l|--extents [+]LogicalExtentsNumber[%{VG|LV|PVS|FREE|ORIGIN}] |  
-L|--size [+]LogicalVolumeSize[bBsSkKmMgGtTpPeE]}  
--poolmetadatasize [+]MetadataVolumeSize[bBsSkKmMgG]}  
[-m|--mirrors Mirrors]  
[--nosync]  
[--use-policies]  
[-n|--nofsck]  
[--noudevsync]  
[--reportformat {basic|json}]  
[-r|--resizefs]  
[-t|--test]  
[--type VolumeType]  
[-v|--verbose]  
[--version]  
LogicalVolume[Path] [ PhysicalVolumePath... ]  
  
[root@wrh-ls-broker ~]# lvextend -l +100%FREE /dev/VG0/DATA  
Size of logical volume VG0/DATA changed from 32.00 GiB (8192 extents) to 963.61 GiB (246685 extents).  
Logical volume VG0/DATA successfully resized.  
[root@wrh-ls-broker ~]#
```

4. Now check the logical volume summary showing the new size with the command:

`lvs`

```
root@wrh-ls-broker~  
[root@wrh-ls-broker ~]# lvs  
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert  
AUDIT VG0 -wi-ao---- 16.00g  
DATA VG0 -wi-ao---- 963.61g  
HOME VG0 -wi-ao---- 16.00g  
ROOT VG0 -wi-ao---- 27.56g  
TMP VG0 -wi-ao---- 8.00g  
VAR VG0 -wi-ao---- 16.00g  
VARLOG VG0 -wi-ao---- 16.00g  
[root@wrh-ls-broker ~]#
```

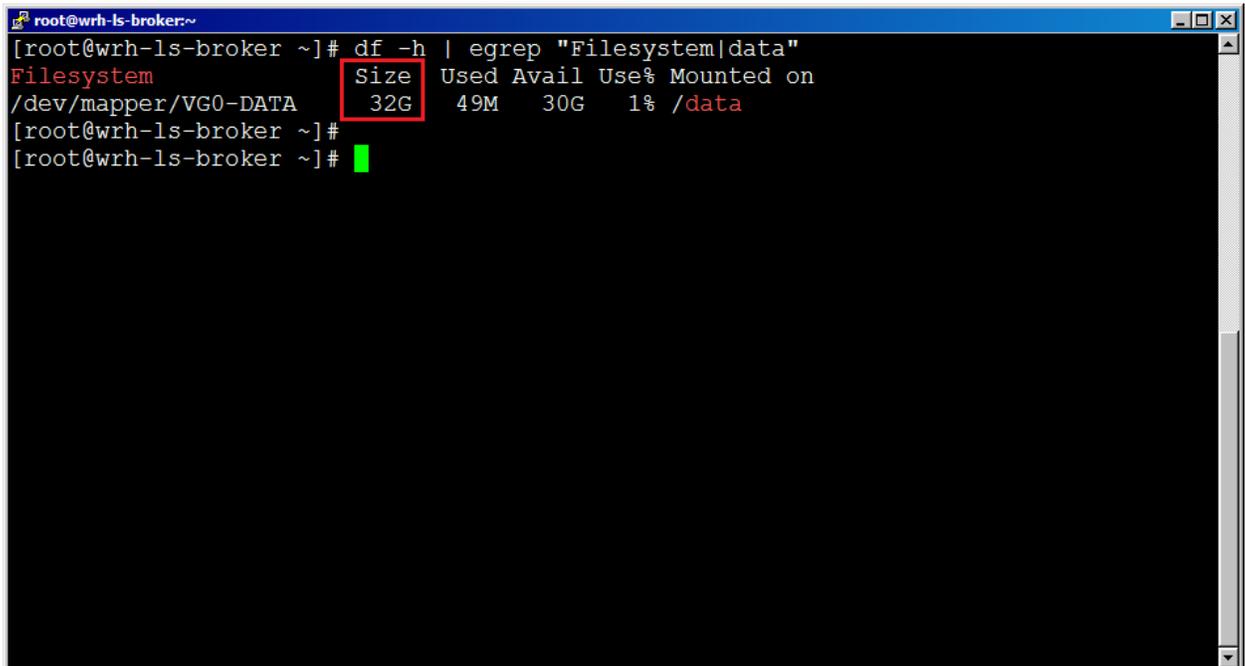
VI. Resize the filesystem

Commands for this section:

```
resize2fs # resize filesystem to match the logical volume  
df -h     # list partition capacities (filesystem)
```

1. View the current filesystem size of the "/data" partition using the df command:

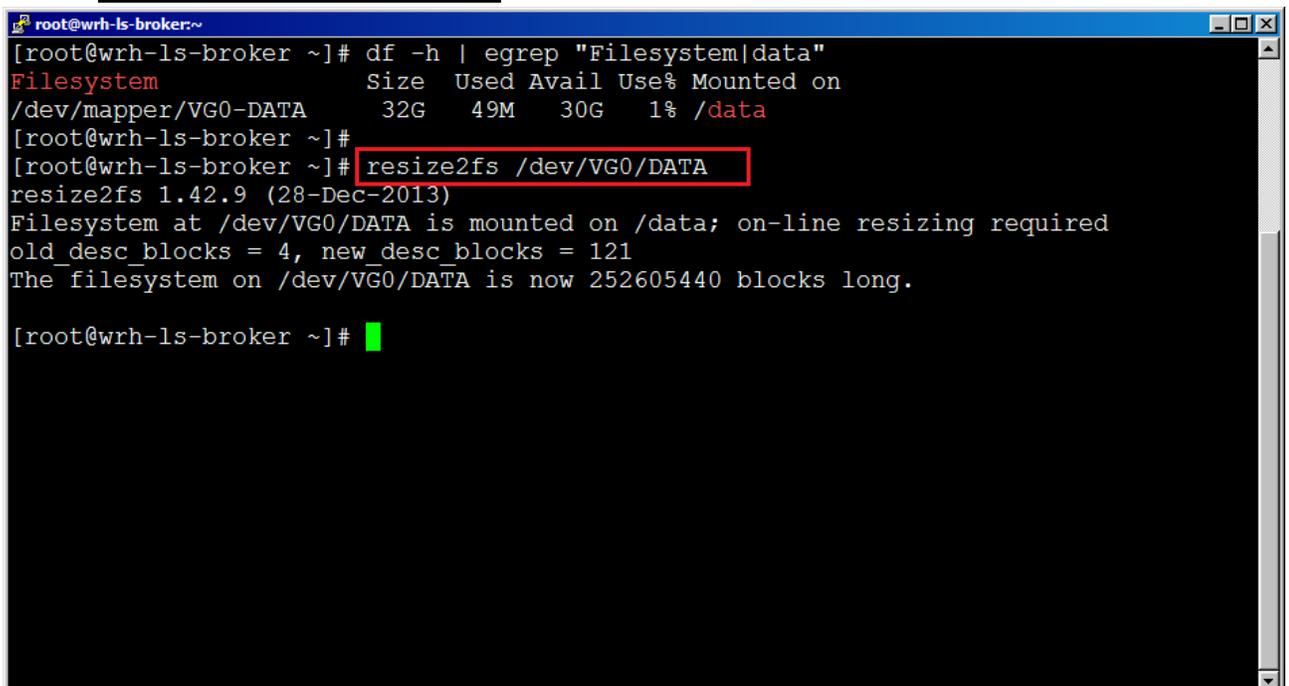
```
df -h | egrep "Filesystem|data"
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# df -h | egrep "Filesystem|data"  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/VG0-DATA 32G   49M   30G   1% /data  
[root@wrh-ls-broker ~]#  
[root@wrh-ls-broker ~]#
```

2. Notice that the filesystem does not yet know that the physical volume is actually much larger now. Resize the filesystem to match the underlying physical volume with the command:

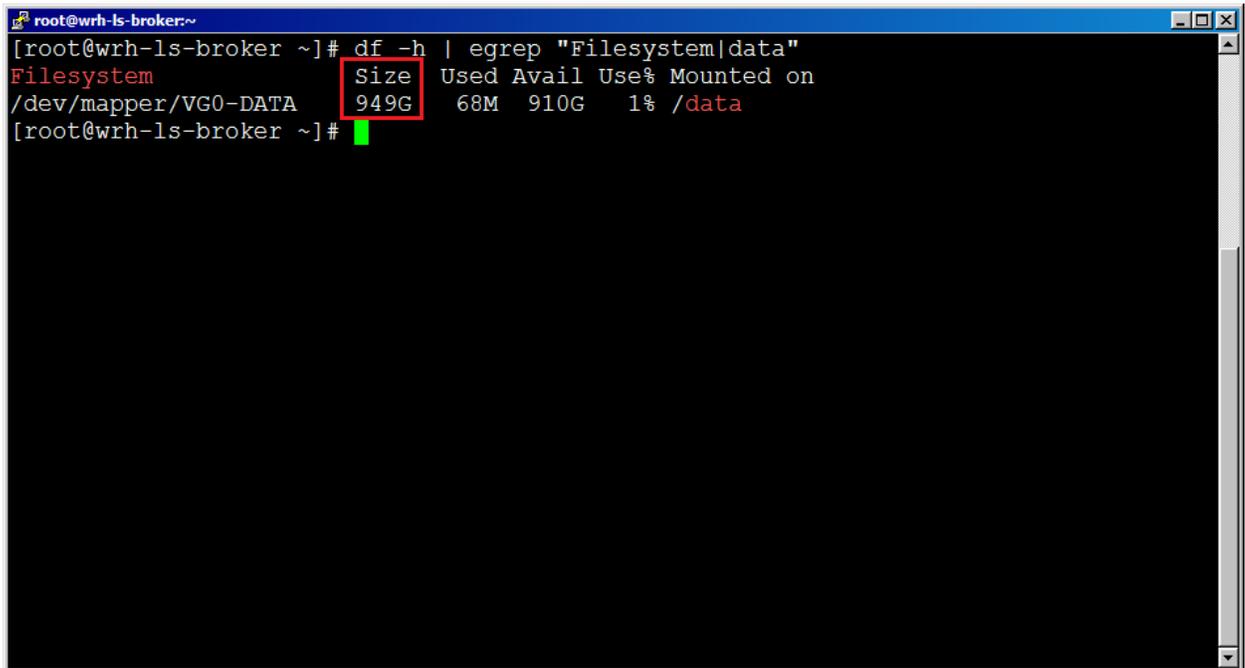
```
resize2fs /dev/VG0/DATA
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# df -h | egrep "Filesystem|data"  
Filesystem      Size  Used Avail Use% Mounted on  
/dev/mapper/VG0-DATA 32G   49M   30G   1% /data  
[root@wrh-ls-broker ~]#  
[root@wrh-ls-broker ~]# resize2fs /dev/VG0/DATA  
resize2fs 1.42.9 (28-Dec-2013)  
Filesystem at /dev/VG0/DATA is mounted on /data; on-line resizing required  
old_desc_blocks = 4, new_desc_blocks = 121  
The filesystem on /dev/VG0/DATA is now 252605440 blocks long.  
  
[root@wrh-ls-broker ~]#
```

- Now repeat the command from step 1 to verify that the filesystem now sees the correct size:

```
df -h | egrep "Filesystem|data"
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# df -h | egrep "Filesystem|data"  
Filesystem          Size      Used Avail Use% Mounted on  
/dev/mapper/VG0-DATA 949G      68M   910G   1% /data  
[root@wrh-ls-broker ~]#
```

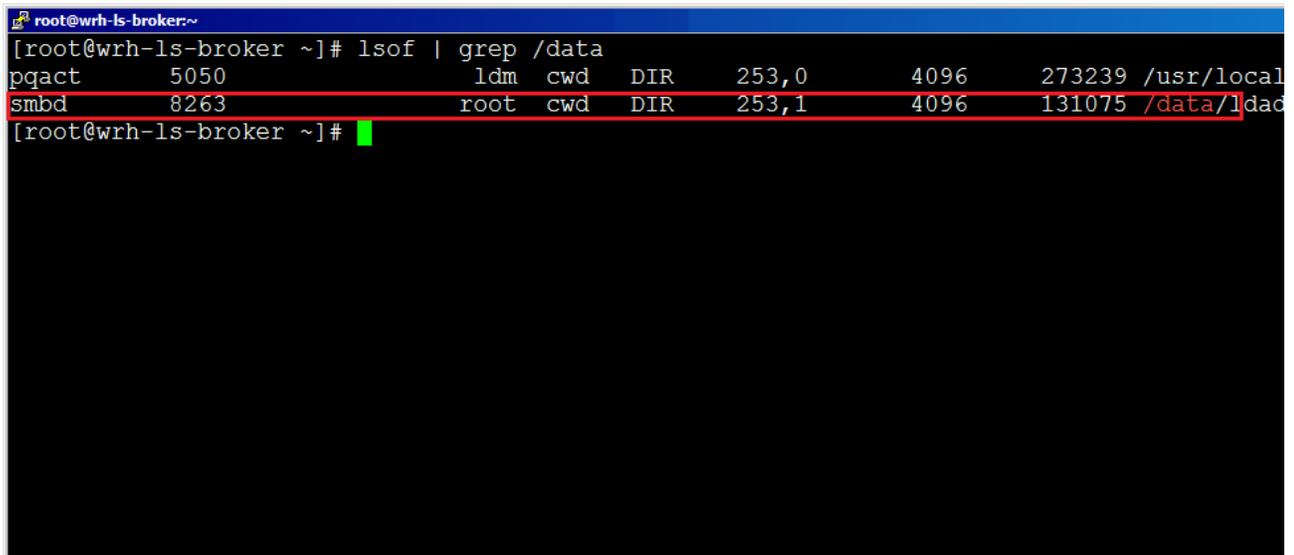
VII. Check the consistency

Commands for this section:

```
e2fsck # check the filesystem to ensure there are no problems
```

- In order to check the filesystem on "/data", we must first unmount it. In order to do that we should ensure that nothing is accessing files on that partition. A useful command for this is:

```
lsof | grep /data
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# lsof | grep /data  
pgact    5050      ldm  cwd  DIR    253,0    4096    273239 /usr/local  
smbd     8263      root cwd  DIR    253,1    4096    131075 /data/ldad  
[root@wrh-ls-broker ~]#
```

- As you can see above, the smb daemon is using data, so we must stop it with the command:

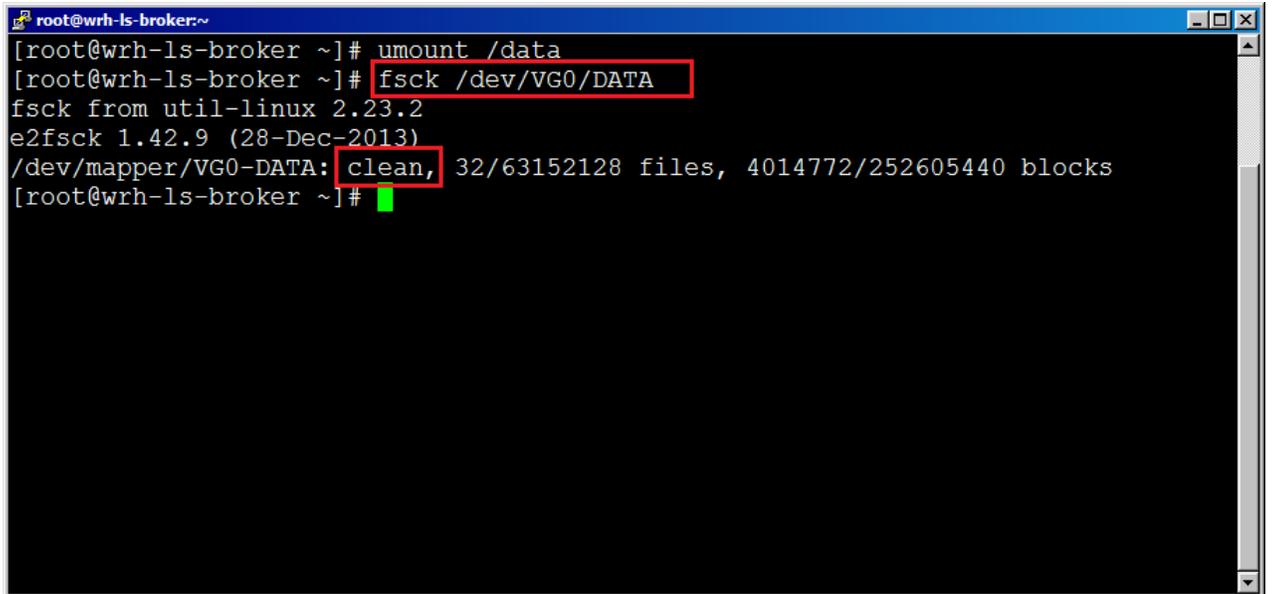
```
systemctl stop centrifydc-samba
```

3. Once everything has been stopped from accessing "/data", unmount it with the command:

```
umount /data
```

4. Now you can perform a filesystem check using the following command:

```
fscck /dev/VG0/DATA
```



```
root@wrh-ls-broker:~  
[root@wrh-ls-broker ~]# umount /data  
[root@wrh-ls-broker ~]# fscck /dev/VG0/DATA  
fscck from util-linux 2.23.2  
e2fsck 1.42.9 (28-Dec-2013)  
/dev/mapper/VG0-DATA: clean, 32/63152128 files, 4014772/252605440 blocks  
[root@wrh-ls-broker ~]#
```

5. You can now mount "/data" again using the following command:

```
mount -a
```

6. At this point I suggest rebooting the server to ensure that everything comes up as expected now that you've made these disk changes.