ProLUG – LVM and RAID

Required Materials

Rocky 9.3 – ProLUG Lab

root or sudo command access

EXERCISES (Warmup to quickly run through your system and familiarize yourself)

- 1. cd ~
- 2. mkdir lvm_lab
- 3. cd lvm_lab
- 4. touch somefile
- 5. echo "this is a string of text" > somefile
- 6. cat somefile
- echo "this is a string of text" > somefile Repeat 3 times
- cat somefile How many lines are there?
- Echo "this is a string of text" >> somefile Repeat 3 times
- 10. cat somefile

how many lines are there?

cheat with `cat somefile \mid wc –l`

- echo "this is our other test text" >> somefile repeat 3 times
- 12. cat somefile | nl how many lines are there?
- cat somefile | nl | grep test compare that with #14
- 14. cat somefile | grep test | nlIf you want to preserve positional lines in file (know how much you've cut out when you grep something, or generally be able to find it in the unfiltered file for context, always | nl | before

your grep

Pre Lab – Disk Speed tests

When using the ProLUG lab environment, you should always check that there are no other users on the system `w` or `who`.

After this, you may want to check the current state of the disks, as they retain their information even after a reboot resets the rest of the machine. `Isblk /dev/xvda`. If you need to wipe the disks, you should use fdisk or a similar partition utility. fdisk /dev/xvda

- p #print to see partitions
- d #delete partitions or information
- w #Write out the changes to the disk.

This is an aside, before the lab. This is a way to test different read or writes into or out of your filesystems as you create them. Different types of raid and different disk setups will give different speed of read and write. This is a simple way to test them. Use these throughout the lab in each mount for fun and understanding.

Write tests (saving off write data – rename /tmp/file each time):

- 1. Check /dev/xvda for a filesystem
- blkid /dev/xvda
- 2. If it does not have one, make one
 - mkfs.ext4 /dev/xvda
- 3. mkdir /space (If you don't have it. Lab will tell you to later as well)
- 4. mount /dev/xvda /space

Write Test:

5. for i in `seq 1 10`; do time dd if=/dev/zero of=/space/testfile\$i bs=1024k count=1000 | tee - a /tmp/speedtest1.basiclvm; done

Read tests:

6. for i in `seq 1 10`; do time dd if=/space/testfile\$i of=/dev/null; done

Cleanup:

7. for i in `seq 1 10`; do rm -rf /space/testfile\$i; done

IF you are re-creating a test without blowing away the filesystem, change the name or counting numbers of testfile because that's the only way to be sure there is not some type of filesystem caching going on to optimize. This is especially true in SAN write tests.*

LAB – start in root (#); cd /root

LVM explanation and use within the system

 Check physical volumes on your server (my output may vary) [root@rocky1~]#fdisk -l | grep -i xvd

Disk /dev/xvda: 15 GiB, 16106127360 bytes, 31457280 sectors Disk /dev/xvdb: 3 GiB, 3221225472 bytes, 6291456 sectors Disk /dev/xvdc: 3 GiB, 3221225472 bytes, 6291456 sectors Disk /dev/xvde: 3 GiB, 3221225472 bytes, 6291456 sectors

2. Looking at Logical Volume Management -

Logical Volume Management is an abstraction layer that looks a lot like how we carve up SAN disks for storage management. We have Physical Volumes that get grouped up into Volume Groups. We carve Volume Groups up to be presented as Logical Volumes. Here at the Logical Volume layer we can assign RAID functionality from our Physical Volumes attached to a Volume Group or do all kinds of different things that are "under the hood". Logical Volumes get filesystems formatting and are mounted to the OS.

There are many important commands for showing your physical volumes, volume groups, and logical volumes. The three simplest and easiest are:

[root@rocky1~]#pvs

[root@ rocky1 ~]#vgs

[root@ rocky1 ~]#lvs

With these you can see basic information that allows you to see how the disks are allocated. Why do you think there is no output from these commands the first time you run them? Try these next commands to see if you can figure out what is happening? To see more in depth information try pvdisplay, vgdisplay, and lvdisplay.

If there is still no output, it's because this system is not configured for LVM. You will notice that none of the disk you verified are attached are allocated to LVM yet. We'll do that next.

 Creating and Carving up your LVM resources Disks for this lab are /dev/xvdb, /dev/xvdc, and /dev/xvdd. (but verify before continuing and adjust accordingly.) We can do individual pvcreates for each disk `pvcreate /dev/xvdb` but we can also loop over them with a simple loop as below. Use your drive letters.

```
[root@Rocky1 ~]#for disk in b c d
```

> do

```
> pvcreate /dev/xvd$disk
```

> done

```
Physical volume "/dev/xvdb" successfully created.
Creating devices file /etc/lvm/devices/system.devices
Physical volume "/dev/xvdc" successfully created.
Physical volume "/dev/xvde" successfully created.
```

[root@rocky1 ~]#pvs

#to see what we made

PV	VG	Fmt	Attr	PSize	PFree
/dev/xvdb		lvm2		3.00g	3.00g
/dev/xvdc		lvm2		3.00g	3.00g
/dev/xvde		lvm2		3.00g	3.00g

[root@ROCKY1 ~]#vgcreate VolGroupTest /dev/xvdb /dev/xvdc /dev/xvde

Volume group "VolGroupTest" successfully created

[root@ROCKY1~]#vgs

[root@ROCKY1 ~]#lvcreate -l +100%FREE -n lv_test VolGroupTest

```
Logical volume "lv test" created.
```

[root@ROCKY1 ~]#lvs

LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync Convert lv_test VolGroupTest -wi-a----- <8.99g

4. Formatting and mounting the filesystem

[root@ROCKY1 ~]#mkfs.ext4 /dev/mapper/VolGroupTest-lv_test

```
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
983040 inodes, 3929088 blocks
196454 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2151677952
120 block groups
```

```
[root@ROCKY1 ~]#mkdir /space #Created earlier
[root@ROCKY1 ~]#vi /etc/fstab
```

Add the following line

/dev/mapper/VolGroupTest-1	v test /space	ext4	defaults	0 0
,	· · · · · · · · · · · · · · · · · · ·			

[root@ROCKY1 ~]#mount -a

#If this command works, there will be no output. We use the df –h in the next command to verify the new filesystem exists. The use of mount -a and not manually mounting the filesystem from the command line is an old administration trick I picked up over the years. By setting our mount in /etc/fstab and then telling the system to mount everything we verify that this will come back up properly during a reboot. We have mounted and verified we have a persistent mount in one step.

[root@ROCKY1~]#df -h

[root@rocky1 ~]# df -h					
Filesystem	Size	Used	Avail	Use%	Mounted on
devtmpfs	4.0M	0	4.0M	0%	/dev
tmpfs	2.0G	0	2.0G	0%	/dev/shm
tmpfs	2.0G	8.5M	1.9G	1%	/run
tmpfs	2.0G	1.4G	557M	72%	/
tmpfs	2.0G	0	2.0G	0%	/run/shm
192.168.200.25:/home	44G	15G	30G	34%	/home
192.168.200.25:/opt	44G	15G	30G	34%	/opt
tmpfs	390M	0	390M	0%	/run/user/0
/dev/mapper/VolGroupTest-lv_test	8.8G	24K	8.3G	1%	/space

Good place to speed test and save off your data

5. Removing and breaking down the LVM to raw disks

The following command is one way to comment out the line in /etc/fstab. If you had to do this across multiple servers this could be useful. (OR YOU CAN JUST USE VI FOR SIMPLICITY)

[root@ROCKY1 ~]#grep lv_test /etc/fstab; perl -pi -e

"s/\/dev\/mapper\/VolGroupTest/#removed \/dev\/mapper\/VolGroupTest/" /etc/fstab; grep removed /etc/fstab

```
/dev/mapper/VolGroupTest-lv_test /space ext4 defaults 0 0
#removed dev/mapper/VolGroupTest-lv_test /space ext4 defaults
0 0
```

[root@ROCKY1 ~]#umount /space

[root@ROCKY1 ~]#lvremove /dev/mapper/VolGroupTest-lv_test

Do you really want to remove active logical volume VolGroupTest/lv_test? [y/n]: \boldsymbol{y}

Logical volume "lv_test" successfully removed

[root@ROCKY1 ~]#vgremove VolGroupTest

Volume group "VolGroupTest" successfully removed

[root@ROCKY1 ~]#for disk in c e f; do pvremove /dev/sd\$disk; done

Labels on physical volume "/dev/sdc" successfully wiped. Labels on physical volume "/dev/sde" successfully wiped. Labels on physical volume "/dev/sdf" successfully wiped.

Use your `pvs;vgs;lvs` commands to verify those volumes no longer exist.

[root@ROCKY1 ~]#pvs;vgs;lvs

PV	VG	Fmt Attr	PSize	PFree				
/dev/sda2	VolGroup(0 lvm2 a	17.48g	4.00m	n			
/dev/sdb	VolGroup(1 lvm2 a	20.00g	96.00m	n			
VG	#PV #LV #	SN Attr V	Size VF	ree				
VolGroup0	0 1 9	0 wzn- 1	7.48g 4	.00m				
VolGroup0	1 1 1	0 wzn- 2	0.00g 96	.00m				
LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move Log
LogVol00	VolGroup00	-wi-ao	2.50g					
LogVol01	VolGroup00	-wi-ao	1000.00m					
LogVol02	VolGroup00	-wi-ao	5.00g					
LogVol03	VolGroup00	-wi-ao	1.00g					
LogVol04	VolGroup00	-wi-ao	5.00g					
LogVol05	VolGroup00	-wi-ao	1.00g					
LogVol06	VolGroup00	-wi-ao	1.00g					
LogVol07	VolGroup00	-wi-ao	512.00m					
LogVol08	VolGroup00	-wi-ao	512.00m					
lv_app	VolGroup01	-wi-ao	19.90g					

More complex types of LVM

LVM can also be used to raid disks

1. Create a RAID 5 filesystem and mount it to the OS (For brevity's sake we will be limiting show commands from here on out, please use pvs,vgs,lvs often for your own understanding)

[root@ROCKY1 ~]#for disk in c e f; do pvcreate /dev/sd\$disk; done

Physical volume "/dev/sdc" successfully created. Physical volume "/dev/sde" successfully created. Physical volume "/dev/sdf" successfully created.

vgcreate VolGroupTest /dev/sdc /dev/sde /dev/sdf lvcreate -l +100%FREE --type raid5 -n lv_test VolGroupTest mkfs.xfs /dev/mapper/VolGroupTest-lv_test

vi /etc/fstab

fix the /space directory to have these parameters (change ext4 to xfs) /dev/mapper/VolGroupTest-lv_test /space xfs defaults 0 0

[root@ROCKY1~]#df -h

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/VolGroup00-LogVol08	488M	34M	419M	8%	/var/log/audit
/dev/mapper/VolGroupTest-lv_test	<mark>10G</mark>	33M	10G	1%	/space

Since we're now using RAID 5 we would expect to see the size no longer match the full 15GB, 10GB is much more of a RAID 5 value 66% of raw disk space.

To verify our raid levels we use lvs

[root@ROCKY1~]#lvs

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%Sync
LogVol00	VolGroup00	-wi-ao	2.50g							
LogVol01	VolGroup00	-wi-ao	1000.00m							
LogVol02	VolGroup00	-wi-ao	5.00g							
LogVol03	VolGroup00	-wi-ao	1.00g							
LogVol04	VolGroup00	-wi-ao	5.00g							
LogVol05	VolGroup00	-wi-ao	1.00g							
LogVol06	VolGroup00	-wi-ao	1.00g							
LogVol07	VolGroup00	-wi-ao	512.00m							
LogVol08	VolGroup00	-wi-ao	512.00m							
lv_app	VolGroup01	-wi-ao	19.90g							
lv_test	VolGroupTest	rwi-ao <mark>r</mark>	9.98g							100.00

Spend 5 min reading the `man lvs` page to read up on raid levels and what they can accomplish. To run RAID 5 3 disks are needed. To run RAID 6 at least 4 disks are needed. *Good place to speed test and save off your data*

2. Set the system back to raw disks

Unmount /space and remove entry from /etc/fstab

```
[root@ROCKY1 ~]#lvremove /dev/mapper/VolGroupTest-lv_test
Do you really want to remove active logical volume VolGroupTest/lv_test? [y/n]:
y
```

Logical volume "lv_test" successfully removed

[root@ROCKY1 ~]#vgremove VolGroupTest

Volume group "VolGroupTest" successfully removed

[root@ROCKY1 ~]#for disk in c e f; do pvremove /dev/sd\$disk; done

Labels on physical volume "/dev/sdc" successfully wiped. Labels on physical volume "/dev/sde" successfully wiped. Labels on physical volume "/dev/sdf" successfully wiped.

Working with MDADM as another RAID option

There could be a reason to use MDADM on the system. For example you want raid handled outside of your LVM so that you can bring in sets of new disks already raided and treat them as their own Physical Volumes. Think, "I want to add another layer of abstraction so that even my LVM is unaware of the RAID levels." This has special use case, but is still useful to understand.

May have to install mdadm yum: yum install mdadm

1. Create a raid5 with MDADM

[root@ROCKY1 ~]#mdadm --create -l raid5 /dev/md0 -n 3 /dev/sdc /dev/sdf mdadm: Defaulting to version 1.2 metadata mdadm: array /dev/md0 started.

2. Add newly created /dev/md0 raid to LVM

This is same as any other add. The only difference here is that LVM is unaware of the lower level RAID that is happening.

[root@ROCKY1 ~]#pvcreate /dev/md0

Physical volume "/dev/md0" successfully created.

[root@ROCKY1 ~]#vgcreate VolGroupTest /dev/md0

Volume group "VolGroupTest" successfully created

[root@ROCKY1 ~]#lvcreate -l +100%Free -n lv_test VolGroupTest

Logical volume "lv_test" created.

[root@ROCKY1~]#lvs

VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log
VolGroup00	-wi-ao	2.50g						
VolGroup00	-wi-ao	1000.00m						
VolGroup00	-wi-ao	5.00g						
VolGroup00	-wi-ao	1.00g						
VolGroup00	-wi-ao	5.00g						
VolGroup00	-wi-ao	1.00g						
VolGroup00	-wi-ao	1.00g						
VolGroup00	-wi-ao	512.00m						
VolGroup00	-wi-ao	512.00m						
VolGroup01	-wi-ao	19.90g						
VolGroupTest	-wi-a <mark>-</mark>	<mark>9.99g</mark>						
	VG VolGroup00 VolGroup00 VolGroup00 VolGroup00 VolGroup00 VolGroup00 VolGroup00 VolGroup00 VolGroup01 VolGroupTest	VG Attr VolGroup00 -wi-ao VolGroup01 -wi-ao VolGroup01 -wi-ao	VG Attr LSize VolGroup00 -wi-ao 2.50g VolGroup00 -wi-ao 1000.00m VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 512.00m VolGroup01 -wi-ao 512.00m VolGroup02 -wi-ao 19.90g VolGroup1 -wi-ao 19.90g	VG Attr LSize Pool VolGroup00 -wi-ao 2.50g VolGroup00 -wi-ao 1000.00m VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 512.00m VolGroup01 -wi-ao 512.00m VolGroup01 -wi-ao 9.99g	VG Attr LSize Pool Origin VolGroup00 -wi-ao 2.50g VolGroup00 -wi-ao 1000.00m VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 512.00m VolGroup01 -wi-ao 512.00m VolGroup01 -wi-ao 19.90g VolGroupTest -wi-a 9.99g	VG Attr LSize Pool Origin Data% VolGroup00 -wi-ao 2.50g VolGroup00 -wi-ao 1000.00m VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 512.00m VolGroup01 -wi-ao 512.00m VolGroup01 -wi-ao 19.90g VolGroupTest -wi-a 9.99g	VG Attr LSize Pool Origin Data% Meta% VolGroup00 -wi-ao 2.50g VolGroup00 -wi-ao 1000.00m VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 5.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 1.00g VolGroup00 -wi-ao 512.00m VolGroup01 -wi-ao 512.00m VolGroup01 -wi-ao 19.90g VolGroupTest -wi-a 9.99g	VG Attr LSize Pool Origin Data% Meta% Move VolGroup00 -wi-ao 2.50g <t< td=""></t<>

Note that LVM does not see that it is dealing with a raid system, but the size is still 10g instead of 15g $\,$

Fix your /etc/fstab to read

/dev/mapper/VolGroupTest-lv_test /space xfs defaults 0 0

[root@ROCKY1 ~]#mkfs.xfs /dev/mapper/VolGroupTest-lv_test

meta-data	a=/dev/mapper/VolGroupTes	st-lv_test isi	lze=512 agcount=16,
agsize=10	63712 blks		
	=	sectsz=512	attr=2, projid32bit=1
	=	crc=1	finobt=0, sparse=0
data	=	bsize=4096	<pre>blocks=2618368, imaxpct=25</pre>
	=	sunit=128	swidth=256 blks
naming	=version 2	bsize=4096	ascii-ci=0 ftype=1
log	=internal log	bsize=4096	blocks=2560, version=2
	=	sectsz=512	<pre>sunit=8 blks, lazy-count=1</pre>
realtime	=none	extsz=4096	blocks=0, rtextents=0

[root@ROCKY1 ~]#mount -a

Good place to speed test and save off your data

3. Setting the MDADM to persist through reboots #not in our lab environment though#

[root@ROCKY1 space]#mdadm --detail --scan >> /etc/mdadm.conf

[root@ROCKY1 space]#cat /etc/mdadm.conf

ARRAY /dev/md0 metadata=1.2 name=ROCKY1:0 UUID=03583924:533e5338:8d363715:09a8b834

Verify with `df -h` ensure that your /space is mounted.

There is no procedure in this lab for breaking down this MDADM RAID.

You are root/administrator on your machine, and you do not care about the data on this RAID. Can you use the internet/man pages/or other documentation to take this raid down safely and clear those disks?

Can you document your steps so that you or others could come back and do this process again?