Cloning Complex Linux Servers

Cloning A Linux Machine That Has A Complex Storage Setup

Where I work we have Cent OS servers whose drives are in various software raid and LVM2 configurations. I was recently asked to prepare to backup/migrate some of these servers using "ghost like" technologies. As I began working with various tools (clonezilla, g4l, etc) I quickly learned that all of our storage flexibility came with a trade off of great complexity. None of the tools available were flexible enough to do the job the way I wanted it to (backup used space only and work properly with mdadm), so I decided to do things by hand. The server we will be cloning in this article has 3 SCSI drives in which root (/), boot (/boot) and swap are in raid 1 arrays and then everything else across all the drives is in a raid 5 array. md0 is boot, md1 is root, md2 is the raid5 array and md3 is the raid 1 swap array. On top of the raid 5 array, md2, are LVM volumes.

I wrote this article in hopes that it will help others (and myself after I forget) who might have to do a similar task in the future. These steps worked for me but they may not work for you. In other words, this text comes with no express or implied warranty. Just back up your data and be careful, and if something should go terribly wrong, don't come crying to me--you have been warned.

What you will need:

- Two servers with similar hardware. The hard drives in the replacement/destination server must be the same size or larger than the drives in the source server. Also, if the hardware is significantly different, the destination server may not boot when you are through with the procedure even though the data was copied properly. This is usually due to kernel or initrid modules/inbuilt differences or due to boot loaders needing to be updated to reflect the new configuration.
- Physical access to the old machine and its replacement
- The ability to take both machines offline for a period of time
- A copy of the System Rescue CD and an optical drive in each server
- Access to a NFS share, external hard drive, or some type of temporary storage
- Lots of free time

Lets get started.

1) Go to the old server and print out some of its current vital storage information. You may need this stuff when you boot off systemrescue CD on either of the machines. It is probably a good idea to get:

```
cat /proc/partitions
cat /proc/mdstat
cat /etc/mdadm.conf
```

```
fdisk -l
cat /etc/fstab
mount
lvdisplay
vgdisplay
pvdisplay
lvscan
vgscan
pvscan
mdadm --detail /dev/md0
mdadm --detail /dev/md1
mdadm --detail /dev/md2, etc
```

2) Boot the systemrescue CD in the old server.

3) Get the system on the network.

```
dhclient eth0
```

Or you can do the *ifconfig* commands if you want.

4) Mount the space you will be using to temporarily store the images of the server you are cloning. Since we will be using *partimage* each image should take up no more than the used space on the original partition you are backing up. IE: If you have a 50Gb partition and *df* shows that your users are only using 20 gigs of it the image for that partition should be less than 20 gigs because *partimage* only backs up space that is used and then it compresses the image itself. Once you have an external hard drive or a NFS server with the right capacity picked out mount it to /mnt/sysimages. I will sending my data over ssh using sshfs, so look at my steps below, modify them to fit your needs, and then issue a *df* to make sure it worked.

```
mkdir /mnt/sysimages
sshfs root@server:/md0/sysimages /mnt/sysimages
```

5) Copy the partition tables from all of your hard drives to the mounted directory.

```
sfdisk -d /dev/sda > /mnt/sysimages/sda.partitions
sfdisk -d /dev/sdb > /mnt/sysimages/sdb.partitions
sfdisk -d /dev/sdc > /mnt/sysimages/sdc.partitions
etc
```

Modify the lines above to fit your needs and then *cat* the files to make sure they look right.

6) See if sysrescure brought up your raid arrays properly.

```
cat /proc/mdstat
```

If it did not, you will have to bring them up manually. Hopefully the command below will help you do this. Increase the number each time for a new array.

```
mdadm --assemble -m 0 /dev/md0
mdadm --assemble -m 1 /dev/md1
```

```
etc
```

If that doesn't work, these articles may help.
http://www.linuxjournal.com/article/8874
http://andrewsblog.org/2008/11/16/mounting-a-linux-software-raid-array-lvm/
http://www.howtoforge.com/recover_data_from_raid_lvm_partitions

7) Bring up LVM.

```
vgchange -a y
/etc/init.d/lvm stop
/etc/init.d/lvm start
```

Make sure you see the volumes by using *lvdisplay*. There should also be some "dm" devices in *cat /proc/partitions* if things are working correctly. If this does not go right by the numbers you may have to manually rebuild the lvm config files by using the info that LVM stores at the beginning of the disks. See the links below for more info.
http://www.howtoforge.com/recover_data_from_raid_lvm_partitions
http://www.linuxjournal.com/article/8874

8) Backup your LVM configuration.

```
vgcfgbackup --file /mnt/sysimages/lvmcfg.backup
```

cat the file and make sure it looks right.

9) Backup the MBRs for your boot drives. Or for all your drives if you don't know.

```
dd if=/dev/sda of=/mnt/sysimages/sda.mbr bs=512 count=1
dd if=/dev/sdb of=/mnt/sysimages/sdb.mbr bs=512 count=1
etc.
```

10) Make sure none of the partitions you want to backup are mounted.

```
mount
```

11) Backup your partitions using partimage. Note that you can specify normal partitions, raid arrays or LVM volumes using the commands below. You want to copy the partition at the highest level possible.

```
partimage -d -b -z1 -M save /dev/md0 /mnt/sysimages/md0.image
partimage -d -b -z1 -M save /dev/md1 /mnt/sysimages/md1.image
partimage -d -b -z1 -M save /dev/testgroup/tesvolume /mnt/sysimages/testvolume.image
```

That last one was for an LVM volume.

-d means don't ask for a description
-b means batch. don't ask dumb questions.
-z1 means use standard image compression.
-M means don't backup the MBR. We already did that.

Glance at the file sizes of the data on your temporary storage device and make sure they seem reasonable.

12) Cleanly shutdown the source server.

Make sure all data is done writing.

```
sync
sync
```

Unmount your temporary storage space.

```
cd /
umount /mnt/sysimages
```

Make sure nothing else is mounted.

```
mount
```

Shutdown LVM.

```
vgchange -an
/etc/init.d/lvm stop
```

Stop your raid arrays.

```
mdadm -S /dev/md0
mdadm -S /dev/md1
etc
```

Shutdown the machine.

```
shutdown -h now
```

13) Boot the replacement server with the system rescue CD.

14) Get the system on the network.

```
dhclient eth0
```

Or you can do the *ifconfig* commands if you want.

15) Mount the space you are using to temporarily store the partition images to /mnt/sysimages. I will getting my data over ssh using sshfs, so look at my steps below, modify them to fit your needs, and then issue a *df* to make sure it worked.

```
mkdir /mnt/sysimages
sshfs root@server:/md0/sysimages /mnt/sysimages
```

16) Double check to make sure there is no data you need on the drives in the destination server.
IF THERE IS IT WILL BE LOST IN THE NEXT STEPS.

17) Make sure nothing is mounted and stop any raid arrays that may be running.

```
mount
mdadm -S /dev/md?
swapoff -a
```

NOTE: None of this should be necessary if you are using clean new drives.

18) Restore the MBRs to your boot drives. Or for all your drives if you don't know.

```
dd if= /mnt/sysimages/sda.mbr of=/dev/sda bs=512 count=1
dd if= /mnt/sysimages/sdb.mbr of=/dev/sdb bs=512 count=1
etc
```

19) Restore the partition tables to your drives.

```
sfdisk /dev/sda < /mnt/sysimages/sda.partitions
sfdisk /dev/sdb < /mnt/sysimages/sdb.partitions
sfdisk /dev/sdc < /mnt/sysimages/sdc.partitions
```

NOTE: I had to use --force on some drives, not sure why.

Now try to update the partition information that the system knows about by issuing the command *partprobe*.

Verify that things look right by doing a *fdisk -l* and a *cat /proc/partitions*. Reboot and check again if things don't look right.

20) Get out those files you printed. Its time to manually re-create our raid arrays.

```
mdadm -C -R -f /dev/md0 -l1 -n2 /dev/sda1 /dev/sdb1
mdadm -C -R -f /dev/md1 -l1 -n2 /dev/sda2 /dev/sdb2
mdadm -C -R -f /dev/md2 -l5 -n3 /dev/sda5 /dev/sdb5 /dev/sdc1
mdadm -C -R -f /dev/md3 -l1 -n2 /dev/sda3 /dev/sdb3
```

-C is create
-R is run
-f is force
-l is level 1,5,0,10
-n is number of devices

You might have to break out the man pages if you have an advanced setup with spare drives or non-default chunk sizes. You may also have to power up the source server again and do a side by side comparison to figure out what to do if your print outs and your memory aren't too good.

21) Watch your arrays rebuild.

```
cat /proc/mdstat
```

22) Recreate your LVM phsyical volumes.
Grab the physical volume uuids from the lvmcfg.backup file (*cat /mnt/sysimages/lvmcfg.backup*) and issue commands like this:

```
pvcreate --uuid UUID_FROM_BACKUP_FILE /dev/md2
```

Check this with *pvdisplay*.

23) Restore your LVM groups.

```
vgcfgrestore --file /mnt/sysimages/lvmcfg.backup testgroup
```

You have to issue a command like that for every volume group.

Verify with *lvdisplay*.

24) Activate LVM.

```
vgchange -ay
```

25) Restore your partitions using partimage.

```
partimage -e -b restore /dev/md0 /mnt/sysimages/md0.image
partimage -e -b restore /dev/md1 /mnt/sysimages/md1.image
partimage -e -b restore /dev/testgroup/tesvolume /mnt/sysimages/testvolume.image
```

-e means erase empty blocks
-b means don't ask stupid questions

26) Re-format your swap partition(s)/array(s).

```
mkswap /dev/md3
```

27) Grub is finicky with raid 1 arrays and with being copied over with *dd*. Reinstall it.

```
/sbin/grub
root (hd0,0)
setup (hd0)
root (hd0,0)
setup (hd1)
quit
```

NOTE: This example is for two drives that have /boot in a raid 1 array. Modify it to fit your needs.

28) Cleanly reboot the destination server with your fingers crossed.

Make sure all data is done writing.

```
sync
sync
```

Unmount your temporary storage space.

```
cd /
umount /mnt/sysimages
```

Make sure nothing else is mounted.

```
mount
```

Shutdown LVM.

```
vgchange -an
/etc/init.d/lvm stop
```

Stop your raid arrays.

```
mdadm -S /dev/md0
mdadm -S /dev/md1
etc
```

Reboot the machine.

```
shutdown -r now
```

29) Did it come up properly?

If not, boot the sysrescue CD mount things and see if the data is there. If it is it is likely a kernel, initrd, bootloader, or fstab issue stopping the machine from booting. If the data isn't there properly, re-check your steps and better luck with the next round.

Sites used while writing this:

- http://www.partimage.org/forums/viewtopic.php?t=46
- http://www.sysresccd.org/Sysresccd-manual-en_Network
- http://www.ducea.com/2006/10/09/partition-table-backup/
- http://www.linuxjournal.com/article/8874
- http://crazedmuleproductions.blogspot.com/2008/11/fedora-9-prep-using-partimage-wraid.html
- http://andrewsblog.org/2008/11/16/mounting-a-linux-software-raid-array-lvm/
- http://www.howtoforge.com/recover_data_from_raid_lvm_partitions
- http://www.linuxjournal.com/article/8874
- http://docs.hp.com/en/B2355-90692/vgcfgbackup.1M.html
- http://www.cyberciti.biz/tips/linux-how-to-backup-hard-disk-partition-table-

mbr.html

- http://ubuntu.wordpress.com/2005/10/20/backing-up-the-mbr/
- http://docs.hp.com/en/B2355-90692/vgcfgrestore.1M.html
- http://linux.about.com/library/cmd/blcmdl8_vgcfgrestore.htm
- http://www.noah.org/wiki/Dd_-_Destroyer_of_Disks