

Root Disk Mirror

- ◆ `pvcreate -B <pv>` Create a BDRA on the new PV
- ◆ `vgextend vg00 <pv>` Add disk to system volume group
- ◆ `mkboot <pv>` Add the LIF filesystem to the BDRA
- ◆ `lvextend -m 1 /dev/vg00/lvol1 <pv>` Mirror the boot LV
- ◆ `lvextend -m 1 /dev/vg00/lvol2 <pv>` Mirror the swap LV
- ◆ `lvextend -m 1 /dev/vg00/lvol3 <pv>` Mirror the root LV
- ◆ `lvlnboot -R /dev/vg00` Ensure BDRAs and LIFs are sync'ed
- ◆ `lvlnboot -v /dev/vg00` Verify BDRA configuration
- ◆ Mirror remaining filesystems in vg00



ROOT DISK MIRROR

Many server configurations rely on mirrored system volume groups. Even common MC/ServiceGuard cluster nodes have mirrored, internal disk, system volume groups.

The two main points to remember about mirroring a system volume group are to ensure the LIF filesystem is available on all bootable disks and the BDRAs on the bootable disks are in sync.

To create a mirror of the system volume group, there first must be enough disks and space available to store a duplicate of all of the logical volumes (including primary swap) already configured in vg00. Of course in a perfect environment all of the disks of vg00 would be of the same type and size. If vg00 consists of two disks, then two more of the same type and size would be used as their mirrors.

The first step is to create a BDRA on the disk to be used as a mirror of the bootable disk (use `pvcreate -B`) and then extend it into the system volume group. The next step is to create a LIF filesystem with the `isl` and `hpux` system loaders. As with creating a system disk, use the `mkboot` command on the new disk. It is important to realize that the LIF area (and, for that matter, the entire LVM “header”) is *not* part of the mirror. The LIF area exists within the BDRA and could be different, although, ideally they would be the same.

Create the mirrored logical volumes using the `lvextend -m 1` command specifying the mirror physical volume as the destination of the physical extents. Start with the boot logical volume (normally lvol1), then swap (normally lvol2), and so on. Since `lvextend` does not change the characteristics of the logical volumes, each retain their required parameters (contiguous, no bad block relocation, etc.). Just make sure the mirror of the boot logical volume resides as the first physical extents on the boot mirror disk and that the location of the root mirror logical volume starts at the same spot on the mirror disk as it does on the original disk.

Finally, to verify that the BDRAs are in sync, use `lvlnboot -R` and `lvlnboot -v`.

The example on the next page details these steps.

EXAMPLE - MIRROR SYSTEM VOLUME GROUP (TWO DISKS)

This example assumes a single-disk (c2t1d0) system volume group named `vg00` with `/stand` on `lvol1`, primary swap on `lvol2`, root on `lvol3`, and the rest of the filesystems on `lvol4` through `lvol8`. Another disk, `/dev/dsk/c2t6d0` will be added to the volume group. It is the same size as the original system disk.

```
# pvcreate -fB /dev/rdisk/c2t6d0      # Create an LVM "header" with a BDRA
# vgextend vg00 /dev/dsk/c2t6d0      # Include the disk into the VG
# mkboot /dev/dsk/c2t6d0             # Create a LIF on the new disk
# for i in 1 2 3 4 5 6 7 8; do        # Mirror all logical volumes in order
> lvextend -A n -m 1 /dev/vg00/lvol$i /dev/dsk/c2t6d0
> done                                # Control where extents allocate
#
# lvinboot -R                        # Ensure BDRAs are synced
# lvinboot -v

Boot Definitions for Volume Group /dev/vg00:
Physical Volumes belonging in Root Volume Group:
    /dev/dsk/c2t1d0 (8/16/5.1.0) -- Boot Disk
    /dev/dsk/c2t6d0 (8/16/5.6.0) -- Boot Disk
Boot: lvol1      on:      /dev/dsk/c2t1d0
                        /dev/dsk/c2t6d0
Root: lvol3      on:      /dev/dsk/c2t1d0
                        /dev/dsk/c2t6d0
Swap: lvol2      on:      /dev/dsk/c2t1d0
                        /dev/dsk/c2t6d0
Dump: lvol2      on:      /dev/dsk/c2t1d0, 0
# vgcfgbackup vg00      # Needed because of the -A n in lvextend above
```

QUORUM CONSIDERATIONS

Recall that if 50% or less of the physical volumes are available, LVM will not activate the volume group. This could be a problem for two-disk system volume group machines. If one of the disks fails and the machine reboots, it will not come back up because it will be unable to activate the system volume group.

The documented method of handling this is to change the string in the AUTO file so that the ISL calls `hpux` using `"hpux -lq"`. The `-lq` option will cause LVM to ignore quorum on the activation of the root volume group. Most documentation says to only make this change to the mirrored disk's AUTO file. Here is an example (continuing with the values from the previous example):

```
# mkboot -a "hpux -lq" /dev/rdisk/c2t6d0
```

However, there are a couple of differing views on the topic of two-disk volume groups and quorum. If only the mirrored disk has the `"hpux -lq"` string in its AUTO file, then what happens if it is the one that fails? The volume group will not activate.

So, what happens if the AUTO file is updated on both disks? It means the system will always boot ignoring quorum on the root volume group. Is it wise to make changes on a volume group that does not have quorum? No.

The final thought is, what does it matter? The system is only going to boot from the primary path anyway unless someone interacts with the console and the PDC. As long as the person doing the console intervention knows to interact with the ISL and use “hpux -lq” to boot the machine, there is no reason for it in the AUTO file.

These are just points to ponder. Probably the best method of handling quorum in a root volume group is to have an odd number of disks beginning with at least three. There would be no need for the AUTO file modification and the machine would boot as long as only one disk failed.