# SAN(LVM+RAID+DRBD+HEARTBEAT+AOE)

Krishna Kumar

September 12, 2009

**Abstract**

This document describes how we can club more and more softwares in SAN so that we can get maximum benfit from it.There are obvious advantages in terms of acessing and backup if we use RAID, LVM, HEARTBEAT and DRBD together. AoE will provide a HA SAN which is worth from many angles.

## 1 System Configuration and objective

There are three systems used in this configuration: CENTOS(10.0.0.243), CENTOS1(10.0.0.122) AND CENTOS2(10.0.0.128). CENTOS2 acts as a primary server while CENTOS1 work as a secondary server. CENTOS works as a client machine. Our objective of this experiment is to create a raid device on CENTOS2 (raid level 1), make a lvm on it and export it over network. Client which is sitting on CENTOS won't feel any disturbance even if CENTOS2 will crash because he will be able to get all of these services from CENTOS1.

## 2 Step by step how to

These are the following steps and procedures which be have two follow for acheiving the above results:

### 2.1 RAID and LVM setup on CENTOS2 and CENTOS1

On CENTOS2 /dev/hda3 and /dev/hda9 are used for creation of a RAID level1.
Run the following command to create a RAID1:
[root@CENTOS2 ]# mdadm -C /dev/md0 -l 1 -n 2 /dev/hda3 /dev/hda9
Verify its status by following command:
[root@CENTOS2 ]# cat /proc/mdstat
[root@CENTOS2 ]# mdadm –detail /dev/md0
If everything is fine then go ahead for LVM by following commands:
[root@CENTOS2 ]# pvcreate /dev/md0
[root@CENTOS2 ]# vgcreate volumgrp /dev/md0
[root@CENTOS1 ]# lvcreate -L 1000M -n logicalvol volumgrp
Verify above things by following commands:
[root@CENTOS2 ]# pvs

[root@CENTOS2 ]# vgs
[root@CENTOS2 ]# lvs
Make similar raid and lvm on CENTOS1:
[root@CENTOS1 ]# mdadm -C /dev/md0 -l1 -n2 /dev/sda3 /dev/sda5
[root@CENTOS1 ]# pvcreate /dev/md0
[root@CENTOS1 ]# vgcreate volumgrp /dev/md0
[root@CENTOS1 ]# lvcreate -L 1000M -n logicalvol volumgrp

## 2.2 DRBD setup on CENTOS2 and CENTOS1

After making the logical volume group we have to setup DRBD and start it on both the machines, primary as well as on secondary so that exact replica will be maintained on secondary node. This will help us in case of failover. The very first step is to edit drbd.conf file on both the machine as follows:

resource testing { # name of resources
protocol C;
on CENTOS2 {
device /dev/drbd0; # Name of DRBD device
disk /dev/volumgrp/logicalvol; # Partition to use, which was created using fdisk
address 10.0.0.128:7788; # IP addres and port number used by drbd
meta-disk /dev/loop0 [0];
}
on CENTOS1{
device /dev/drbd0;
disk /dev/volumgrp/logicalvol;
address 10.0.0.122:7788;
meta-disk /dev/loop0 [0];
}
disk {
on-io-error detach;
}
net {
max-buffers 2048;
ko-count 4;
}
syncer {
rate 10M;
al-extents 257;
}
startup {
wfc-timeout 0;
degr-wfc-timeout 120; # 2 minutos. }
}

[root@CENTOS2 ]# ls -l new1.img
[root@CENTOS2 ]# losetup /dev/loop0 new1.img
[root@CENTOS2 ]# drbdadm create-md testing
[root@CENTOS2 ]# /etc/init.d/drbd restart

[root@CENTOS2 ]# cat /proc/drbd

Repeat above steps on CENTOS1. Take a 200MB new1.img file (by using dd if=/dev/zero of=new.img bs=1M count=200), make a loop device from this and start DRBD on secondary node. At this time the output of cat /proc/drbd will show both the device as secondary device. Now we have everything ready for the block device, execpt the filesystem. So, make a file system on drbd block device on both the primary and secondary node. Run following command on CENTOS2:

[root@CENTOS2 ]# drbdadm – –overwrite-data-of-peer primary all
[root@CENTOS2 ]# cat /proc/drbd
[root@CENTOS2 ]# mkfs.ext3 /dev/drbd0
[root@CENTOS2 ]# drbdadm secondary all

Run following command on secondary node:
[root@CENTOS1 ]# drbdadm primary all
[root@CENTOS1 ]# mkfs.ext3 /dev/drbd0
[root@CENTOS1 ]# drbdadm secondary all

At this time we have drbd device prepared with file system on both the nodes. But both the device is in secondary state. So, convert one device in primary state (here on CENTOS2).
[root@CENTOS2 ]# drbdadm primary all
[root@CENTOS2 ]# cat /proc/drbd

Now we have everything ready for DRBD on both the nodes. Its time to start the heartbeat on primary as well as on secondary node:

## 2.3 Heartbeat setup on CENTOS2 and CENTOS1

1 Edit /etc/ha.d/ha.cf file as follows:
debugfile /var/log/ha-debug
logfile /var/log/ha-log
logfacility local0
keepalive 2
deadtime 30
initdead 120
bcast eth0 # Linux
auto_failback on
node CENTOS2
node CENTOS1

2 Edit /etc/ha.d/ha.cf file as follows:
CENTOS2 10.0.0.255 drbddisk::testing aoeinit.sh

3 Edit /etc/ha.d/authkeys file as follows:
auth 2
2 crc
The file permission of authkeys is as follows:

[root@CENTOS2 ]# chmod 600 /etc/ha.d/authkeys

4 Make a new file /etc/ha.d/resource.d/aoeinit.sh as follows:
case "$1" in
"start")
vbladed 0 3 eth0:0 /dev/drbd0
;;
"stop")
kill `pidof vblade`
;;
*)
echo "usage: `basename $0` start—stop" 1>&2
;;
esac

Copy all the above file from primary node to secondry node:
5 [root@CENTOS2 ha.d]# cd /etc/ha.d
6 [root@CENTOS2 ha.d]# scp ha.cf authkeys haresources root@CENTOS1:/etc/ha.d/
7 [root@CENTOS2 resource.d]# cd /etc/ha.d/resource.d
8 [root@CENTOS2 resource.d]# scp aoeinit.sh root@CENTOS1:/etc/ha.d/resource.d/

Start the heartbeat on both the nodes by following command:
9 [root@CENTOS2 resource.d]# /etc/init.d/heartbeat start
10 [root@CENTOS1 ]# /etc/init.d/heartbeat restart

## 2.4 To acess the exported block device on client side

Try following command at primary node to acess the exported device.
[root@CENTOS aoe6-72]# modprobe aoe

If heartbeat is running on primary and secondary node then one should be able to ping 10.0.0.255(virtual ip) and he should also be able to see exported block device.
[root@CENTOS aoe6-72]# ping 10.0.0.255
PING 10.0.0.255 (10.0.0.255) 56(84) bytes of data.
64 bytes from 10.0.0.255: icmp_seq=1 ttl=64 time=0.148 ms
64 bytes from 10.0.0.255: icmp_seq=2 ttl=64 time=0.143 ms
64 bytes from 10.0.0.255: icmp_seq=3 ttl=64 time=0.147 ms
64 bytes from 10.0.0.255: icmp_seq=4 ttl=64 time=0.148 ms
64 bytes from 10.0.0.255: icmp_seq=5 ttl=64 time=0.145 ms
[root@CENTOS aoe6-72]# aoe-stat
e0.3 5.239GB eth0 1024 up
[root@CENTOS aoe6-72]# ls -l /dev/etherd/e0.3
brw-r—— 1 root disk 152, 0 2009-09-08 16:20 /dev/etherd/e0.3
Mount the exported block device and do some read write operation as follows:
[root@CENTOS aoe6-72]# mount /dev/etherd/e0.3 /mnt/
[root@CENTOS aoe6-72]# ls -l /mnt/

## 2.5 Checking the failover across the two nodes

Now its time to see the magic. We are doing some write operation from client machine(CENTOS) to primary node(CENTOS2). While doing write operation, we will plug off the network cable. The idle case is user should not lose any data and his write operation should not disturb. These are the following steps to perform the failover phenomenon.

[root@CENTOS aoe6-72]# cd /mnt/
[root@CENTOS mnt]# dd if=/dev/zero of=new1.img bs=1M count=200

While copying the file block by block, plug off the network cable. Try to see the contents of /dev/drbd0 on secondary node as well as client node when dd has finished. Run following command on client node(CENTOS):
[root@CENTOS aoe6-72]# ls -l /mnt/
total 208132
-rw-r–r– 1 root root 0 2009-09-07 16:10 chintoocandy
-rw-r–r– 1 root root 0 2009-09-06 01:09 lo
drwx—— 2 root root 16384 2009-06-16 14:05 lost+found
-rw-r–r– 1 root root 38 2009-09-06 01:02 mm
-rw-r–r– 1 root root 0 2009-09-06 01:48 new
-rw-r–r– 1 root root 209715200 2009-09-08 17:30 new1.img
-rw-r–r– 1 root root 107 2009-06-16 15:02 newfile
-rw-r–r– 1 root root 47 2009-09-06 01:49 pk
-rw-r–r– 1 root root 3169720 2009-09-07 16:12 vmlinuz

Run following command on secondary node:
[root@CENTOS1 ]# ifconfig eth0:0
eth0:0 Link encap:Ethernet HWaddr 00:14:85:7C:89:0E
inet addr:10.0.0.255 Bcast:10.0.3.255 Mask:255.255.252.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
Interrupt:16 Base address:0x4000
[root@CENTOS1 ]# mount /dev/drbd0 /mnt/
[root@CENTOS1 ]# ls -l /mnt/
total 208132
-rw-r–r– 1 root root 0 Sep 7 16:10 chintoocandy
-rw-r–r– 1 root root 0 Sep 6 01:09 lo
drwx—— 2 root root 16384 Jun 16 14:05 lost+found
-rw-r–r– 1 root root 38 Sep 6 01:02 mm
-rw-r–r– 1 root root 0 Sep 6 01:48 new
-rw-r–r– 1 root root 209715200 Sep 8 17:30 new1.img
-rw-r–r– 1 root root 107 Jun 16 15:02 newfile
-rw-r–r– 1 root root 47 Sep 6 01:49 pk
-rw-r–r– 1 root root 3169720 Sep 7 16:12 vmlinuz

So, what we see that even if primary node crashes, our data will be saved on secondary node. Client doesn't feel any disturabnce while working on exported block device. Virtual ip has also shifted from primary node to secondary node.

## 2.6 After reboot starting of raid and lvm

If we reboot the system, then we have to manually run following command to restart lvm and raid on both the nodes :
[root@CENTOS1 ]# mdadm –assemble /dev/md0 /dev/sda3 /dev/sda5
[root@CENTOS1 ]# lvm vgscan
[root@CENTOS1 ]# lvm vgchange -ay
[root@CENTOS1 ]# lvm lvscan
[root@CENTOS1 ]# losetup /dev/loop0 new.img
[root@CENTOS1 ]# /etc/init.d/drbd restart
[root@CENTOS1 ]# cat /proc/drbd

Make DRBD primary on CENTOS2 by following commands
[root@CENTOS2 ]# drbdadm – –overwrite-data-of-peer primary all