

ISCSI for the Hobbyist

Jim Wildman

Ohio Linux Fest

October 11, 2008

jim.wildman@gmail.com



Introduction

- Unix user since 1985, Red Hat user since 1995
- RHCE (2x)
 - 1 endorsement towards RHCA
- Employed by JPMorgan Chase
 - Team leader for the Global Linux Engineering team
 - Standard disclaimer: nothing I say is to be taken as endorsement by JPMorgan Chase, etc.
- Frequent presenter at COLUG
 - Central Ohio Linux User Group (colug.net)
- Format for today.
 - 30 minute presentation, 30 minute demo.
 - There will be 'dead' time during some of the demo for the questions

What is iSCSI?

- From open-iscsi.org
 - “Open-iSCSI project is a high performance, transport independent, multi-platform implementation of RFC3720.”
 - Simple speak: SCSI commands over regular ethernet
- For our purposes: A way to collect bits and pieces of disk drives and access them as block devices across the network
- It is NOT by itself a cluster file system, though it may be used as part of one (ie, no lockd, etal)
- Note: We will be discussing iscsi in its simplest form

Terminology

- Targets
 - Iscsi drives presented to the network by the tgt (target daemon)
 - A host can provide multiple targets
 - Can be restricted to ip
 - Can use simple chap authentication
- Initiator
 - Machine accessing a target
 - Can access multiple targets on multiple machines on multiple networks
 - Can use multipath
 - All drives look identical (ie, it's a SCSI drive .. sdX)



Use case #1: using spare drive space

- Various systems
 - Wife has 160G drive, with 100G free
 - Kid has 100G drive with 60G free
 - Old desktop has 120G drive with 80G free
- Create a 50G iscsi target on each machine (2 on the wife's)
 - Can be
 - Traditional partition
 - LVM logical volume
 - Part of a Windows drive (single connection Windows target software is available for free)
- On the initiator
 - Bind to each target
 - Use mdadm to create RAID5 across the 4 drives
 - Can use LVM on top of the md device



Use case #2: create temporary or scratch space

- Systems same as case #1
- Create maximum sized iscsi target on each machine (100G, 60G, 80G)
- Use RAID0 to aggregate the drives into a single large volume
- Use LVM or single partition to utilize space



Use case #3: create NAS or storage device

- Purpose selected identical drives
 - a 2nd or 3rd drive in existing machines
 - USB drives (spinning or solid state)
 - Additional NIC in each machine
- Create identical iscsi targets
- Bind to multiple IP's on the initiator machine
- Configure multipathd on the initiator
- Use mdadm across the multipath devices
- Use LVM on top of mdadm



Status with RHEL (CentOS)

- Initiator software is in tech preview status for 5.2
- Target software is in production for 5
- Target software is also in 4.6+, but it uses a different command set
- Based on software from open-iscsi.org



Target command set – tgtadm

- Create target iqn
- Create lun against iqn
- 'back' the lun with storage
- 'bind' the iqn to an IP/setup authentication
- Report on status
- Notes
 - Does not run automatically on boot.
 - Must be added to `/etc/rc.d/rc.local` or similar



Initiator command set – iscsiadm

- 'discover' targets
- 'login' or 'bind' to targets
- Report on status
- Notes
 - When using iscsi drives in /etc/fstab, use the `_netdev` type
 - Manages a pseudo 'database' in /var/lib/iscsi



General notes

- Does not appear to work in Vmware-server guests
- Does work in Xen guests
- Do NOT set the partition type to lvm or RAID on target machines
 - 'local' kernel will try to assemble lvm or RAID and prevent tgtadm from using the device
- Red Hat implementation differs from open-iscsi
 - Use of /var/lib/iscsi
 - Documentation



General notes #2



- Reboot
 - Initiator handles reboot of targets ok
 - Tgtadm commands must be placed in `/etc/rc.d/rc.local`
 - Best to leave `iscsid` off on initiator and start manually (loooong timeouts if targets not available)
- DO NOT USE `/dev/sd?` It will change on reboot
 - Customize `udev` rules
 - Use `/dev/disk/by-path/<blah>`
 - Need to use `fdisk` on the initiator to set partition types, etc
 - Use `lvm.conf`, `mdadm.conf` with UUID's

'Lab' Environment

- Dell D630 Laptop
 - 40G USB drive
 - RHEL 5.2
 - 2G RAM
- 4 Xen target nodes (node1, 2, 3, 4)
 - 4G root (actually an LVM snapshot)
 - 1G drive for use with iscsi
- 1 Xen initiator node (node5)



Configurations to demo

- Back an iscsi target with
 - Partition
 - logical volume
 - file image
 - software raid
- RAID5 to 3 iscsi targets on partitions
- RAID5 to 3 iscsi targets on logical volumes



More resources

- open-iscsi.org -
 - Background information
 - Configuration is different than CentOS/Red Hat
- Mike Christie Red Hat Summit presentation
http://www.chpc.utah.edu/~brian/RedHatSummit2008/Mike_Christie.pdf
 - “Installation was Simple, but Optimizing the Root Session will Make You Cry”
- /usr/share/doc/scsi-target-utils-[version]/README

Scripts – assumptions

- Nodes are named node1, node2, node3, node4.
- IP addresses are 192.168.122.11, 12, 13, 14
- TID's are created by concatenating the last ip of the address and the LUN number (which is also the partition number except for the RAID1 devices)
- **NO ERROR CHECKING**
 - These are not production quality

Scripts – setup the disk

```
TID=`ifconfig eth0 | grep "inet addr" | cut -c33-34`
HN=`hostname -s`
IQN=iqn.2008-10.private.build
INITIATOR=192.168.122.1

# setup partition 1 as a logical volume
echo "Setting up vg and lv"
pvcreate /dev/xvdb1
vgcreate vg_iscsi /dev/xvdb1
lvcreate -L +200M vg_iscsi -n lv_iscsi

# Make a filesystem on partition 2 and mount it
echo "Creating filesystem"
mke2fs -j /dev/xvdb2
mkdir -p /iscsi
mount /dev/xvdb2 /iscsi

echo "Create image file"
dd if=/dev/zero of=/iscsi/iscsi.img bs=4096 count=40000

echo "Creating RAID1"
mdadm -C /dev/md0 -f -n 2 -l 1 /dev/xvdb3 /dev/xvdb4

echo "make sure tgt is running"
chkconfig tgt on
service tgt restart
```

Scripts – lvm target

```
TID=`ifconfig eth0 | grep "inet addr" | cut -c33-34`  
HN=`hostname -s`  
IQN=iqn.2008-10.private.build  
INITIATOR=192.168.122.1
```

```
echo "lv backed lun"  
tgtadm --lld iscsi --op new --mode target --tid ${TID}1 -T $IQN.$HN:disk1  
tgtadm --lld iscsi --op new --mode logicalunit --tid ${TID}1 --lun 1 -b /dev/mapper/vg_iscsi-lv_iscsi  
tgtadm --lld iscsi --op bind --mode target --tid ${TID}1 -I ALL  
echo "display the output"  
tgtadm --lld iscsi --op show --mode target
```

Scripts – image target

```
TID=`ifconfig eth0 | grep "inet addr" | cut -c33-34`  
HN=`hostname -s`  
IQN=iqn.2008-10.private.build  
INITIATOR=192.168.122.1
```

```
echo "image backed lun"  
tgtadm --lld iscsi --op new --mode target --tid ${TID}2 -T $IQN.$HN:disk2  
tgtadm --lld iscsi --op new --mode logicalunit --tid ${TID}2 --lun 2 -b /iscsi/iscsi.img  
tgtadm --lld iscsi --op bind --mode target --tid ${TID}2 -I ALL
```

```
echo "display the output"  
tgtadm --lld iscsi --op show --mode target
```

Scripts – partition target

```
TID=`ifconfig eth0 | grep "inet addr" | cut -c33-34`  
HN=`hostname -s`  
IQN=iqn.2008-10.private.build  
INITIATOR=192.168.122.1
```

```
echo "Create partition backed lun"
```

```
tgtadm --lld iscsi --op new --mode target --tid ${TID}3 -T $IQN.$HN:disk3  
tgtadm --lld iscsi --op new --mode logicalunit --tid ${TID}3 --lun 3 -b /dev/xvdb3  
tgtadm --lld iscsi --op bind --mode target --tid ${TID}3 -I ALL
```

```
echo "Create partition backed lun"
```

```
tgtadm --lld iscsi --op new --mode target --tid ${TID}4 -T $IQN.$HN:disk4  
tgtadm --lld iscsi --op new --mode logicalunit --tid ${TID}4 --lun 4 -b /dev/xvdb4  
tgtadm --lld iscsi --op bind --mode target --tid ${TID}4 -I ALL
```

```
echo "display the output"
```

```
tgtadm --lld iscsi --op show --mode target
```

Scripts – image target

```
TID=`ifconfig eth0 | grep "inet addr" | cut -c33-34`  
HN=`hostname -s`  
IQN=iqn.2008-10.private.build  
INITIATOR=192.168.122.1  
  
echo "Create md0 backed target"  
tgtadm --lld iscsi --op new --mode target --tid ${TID}3 -T $IQN.$HN:disk3  
tgtadm --lld iscsi --op new --mode logicalunit --tid ${TID}3 --lun 3 -b /dev/md0  
tgtadm --lld iscsi --op bind --mode target --tid ${TID}3 -I ALL  
  
echo "display the output"  
tgtadm --lld iscsi --op show --mode target
```

Scripts – discovery and login

```
echo "Making sure iscsid is dead.  service iscsid stop is not clean"
service iscsid stop
killall iscsid

echo "Setting up /etc/iscsi/initiatorname.iscsi"
cat > /etc/iscsi/initiatorname.iscsi << EOF
InitiatorName=iqn.2008-10.private.build
InitiatorAlias=node4
EOF

echo "Starting iscsid"
service iscsid start

echo "Doing discovery"
for i in 11 12 13 14; do iscsiadm -m discovery -t st -p 192.168.122.$i; done
iscsiadm -m node

echo "logging in"
iscsiadm -m node -L all

echo "listing disks"
fdisk -l | grep ^Disk | grep bytes | grep -v partition
```

Scripts – not written

- Target

- To create the md0 for the mirror
- Once correct, need to add to /etc/rc.d/rc.local or a proper /etc/rc.d initscript

- Initiator

- sfdisk/fdisk
 - Once logged in, all drives need to have a correct partition created
- Assemble the devices into RAID, create pv/lv, filesystems, etc
- Add UUID's to /etc/mdadm.conf
 - mdadm -D /dev/md0 to find the uuid
- Use _netdev in /etc/fstab