

SAN Boot Linux (iSCSI)

Synopsis

So I built this PXE server to eliminate my need for CD/DVD base installation media. That got me to thinking about SAN boot and how I could eliminate the need for a physical hard drive on some of my physical computers. I knew of 2 SAN boot options, Fiber Channel and iSCSI. I definitely can't afford the equipment necessary for Fiber Channel, so iSCSI was my choice. Here is how I installed and boot Ubuntu and CentOS using an iSCSI target disk.

Build the iSCSI Target Server

First make a virtual disk. I am putting my iscsi disk images in /tftpboot/iscsi. The command below will make a 20GB virtual disk IMG:

```
sudo dd if=/dev/zero of=/tftpboot/iscsi/centos.img bs=1M count=20000
```

Next, install and enable the iSCSI target software:

```
sudo apt-get install iscsitarget
sudo nano /etc/default/iscsitarget
```

make the following change

```
Change
ISCSITARGET_ENABLE=false
To
ISCSITARGET_ENABLE=true
```

Next we will define our iSCSI share. Everything in this file should be commented out. If for some reason it isn't, comment everything and put your own information using the following template. You can include more than one target, I have included both my centos.img and ubuntu.img

```
sudo nano /etc/iet/ietd.conf
```

```
Target iqn.2014-06.com.it-joe:centos-server
    IncomingUser <username> <password> ##Leave blank if you want no
authentication
    OutgoingUser
    Lun 0 Path=/tftpboot/iscsi/centos.img,Type=fileio
    Alias centos-server

Target iqn.2014-06.com.it-joe:ubuntu-desktop
    IncomingUser <username> <password> ##Leave blank if you want no
authentication
    OutgoingUser
    Lun 0 Path=/tftpboot/iscsi/ubuntu.img,Type=fileio
```

Alias ubuntu-desktop

Now you should be ready to start the iSCSI Target Server:

```
sudo /etc/init.d/iscsitarget start
```

Installing an OS on iSCSI target disk

I am going to fork these off because each OS installation is different. Choose your operating system by clicking on the tab

- Ubuntu
- CentOS

Ubuntu

Ubuntu doesn't come prepackaged with an iSCSI initiator. Because of this, there are a few extra steps to install and PXE boot an Ubuntu iSCSI drive. Nevertheless, it can be done, and here is how to do it.

Installation

First, boot to an Ubuntu LiveCD. I am using Ubuntu 13.04 that I set up to PXE boot in my [PXE Server](#) tutorial. Once your booted to the desktop we need to install open-iscsi and discover our iSCSI target disk that we created above.

Install and start open-iscsi

```
sudo su
apt-get install open-iscsi
/etc/init.d/open-iscsi start
```

Discover and connect to your iSCSI target

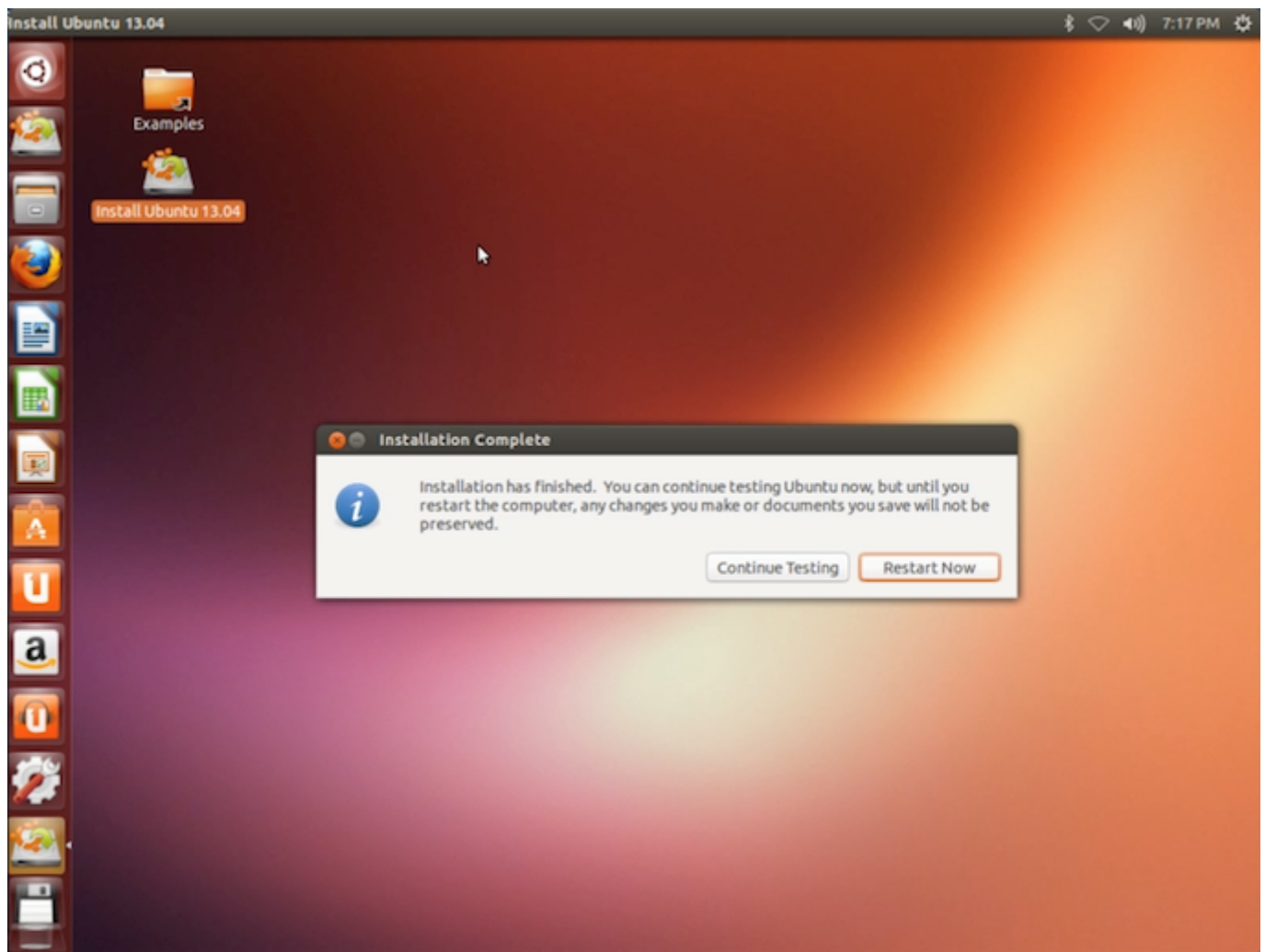
```
iscsiadm -m discovery -t st -p 192.168.0.50
iscsiadm -m node --targetname "iqn.2014-06.com.it-joe:ubuntu-desktop" --portal "192.168.0.50:3260" --login
```

Run fdisk to confirm your disk is connected

```
fdisk -l
```

Now that Ubuntu can see our iSCSI target disk, we can proceed with the install as we normally would.

When you get to the screen below, DO NOT REBOOT.



Next we need to install a few packages on our new installation. How do we do this without booting to our new installation? Simple, by using "chroot". chroot basically changes our working root (/) directory. By using chroot we have direct access to the file system of our new installation and any changes will effect it as if we were booted directly to it.

```
sudo su
mkdir /mnt/temp
mount /dev/sda1 /mnt/temp
chroot /mnt/temp
mount -t proc none /proc
echo "nameserver 192.168.0.1" >> /etc/resolv.conf
hostname ubuntu
apt-get install initramfs-tools open-iscsi sysv-rc-conf
echo "iscsi" >> /etc/initramfs-tools/modules
echo "InitiatorName=iqn.2014-06.com.it-joe:ubuntu-client1" >
/etc/iscsi/initiatorname.iscsi
touch /etc/iscsi/iscsi.initramfs
update-initramfs -u
```

Edit the /etc/fstab file to use /dev/sda1 rather than UUID for root

```
nano /etc/fstab
```

Look for a line similar to:

```
UUID=5c0bd3d6-b43b-48d4-ac82-cc9b364f4544 / ext4 errors=remount-  
ro 0 1
```

Replace with:

```
/dev/sda1 / ext4 errors=remount-ro 0 1
```

Make sure `/etc/network/interfaces` has the following lines

```
auto eth0  
iface eth0 inet manual
```

Copy the `vmlinuz` and `initrd.img` files to your PXE server

```
scp /boot/initrd.img-3.8.0-19-generic /boot/vmlinuz-3.8.0-19-generic  
root@192.168.0.50:/tftpboot/iscsi/
```

PXE Setup

At this point we should have everything we need to get our PXE server setup. Just add an entry similar to the one below to your `pxelinux.cfg/default` file

```
LABEL Ubuntu  
KERNEL iscsi/vmlinuz-3.8.0-19-generic  
APPEND initrd=iscsi/initrd.img-3.8.0-19-generic ip=dhcp  
ISCSI_INITIATOR=iqn.2014-06.com.it-joe:ubuntu-client1  
ISCSI_TARGET_NAME=iqn.2014-06.com.it-joe:ubuntu-desktop  
ISCSI_TARGET_IP=192.168.0.50 ISCSI_TARGET_PORT=3260 root=/dev/sda1 rw
```

The LABEL can be whatever you would like. The KERNEL line just needs to be the path to the `vmlinuz` file (relative to the tftp server `/`). The APPEND line needs to be `initrd="the path to the initrd.img"` and the rest is copied directly from the kernel line in the `grub.conf` file we transferred earlier

CentOS

This is the first of 2 options to install CentOS to an iSCSI target. I prefer this method because there is no need to copy the `initramfs.img` or `vmlinuz` files to the PXE Server. This method also makes updating CentOS much easier. If you use CentOS Alt, updating the kernel would require you to copy over the new `initramfs.img` and `vmlinuz` files each time.

iPXE Setup

Rather than using the built in iSCSI initiator in CentOS to locate the iSCSI target, this option is going to use iPXE to connect to the iSCSI target and boot to our CentOS installation media.

Begin by going to rom-o-matic.eu to generate an iPXE image.

- Select: "Advanced, for experienced users"
- Choose an output format: "Linux kernel (SYSLINUX/GRUB/LILO) loadable image (.lkrn)"
- Choose a NIC type: "all-drivers"

Keep all the default options checked and add the following config options plus any others you think would be useful.

- PXE_STACK, PXE stack in iPXE - you want this
- DOWNLOAD_PROTO_NFS, Network File System Protocol
- SANBOOT_PROTO_ISCSI, iSCSI protocol

Copy the ipxe.lkrn file that is generated to the / of your tftp server. If you followed my previous tutorials than that would be /tftpboot/. Now that our ipxe.lkrn file is in place we just need to make an entry into our PXE menu and create an ipxe script. First lets make our ipxe script.

```
nano centos.ipxe
```

```
#!/ipxe
dhcp
set keep-san 1
sanhook iscsi:192.168.0.50::::iqn.2014-06.com.it-joe:centos-server
set boot-url tftp://192.168.0.50
kernel ${boot-url}//distros/centos-installer/isolinux/vmlinuz
method=nfs:192.168.0.50:/tftpboot/distros/centos-installer/ lang=en_US.UTF-8
keyboard=us ksdevice=eth0 ip=dhcp
initrd ${boot-url}/distros/centos-installer/isolinux/initrd.img
boot
```

Then the PXE menu entry:

```
nano nano /tftpboot/prelinux.cfg/default
```

```
LABEL iPXE CentOS
MENU LABEL iPXE CentOS
KERNEL ipxe.lkrn
INITRD ipxe.centos
```

Installation

To begin the installation, boot to your PXE Server and select the iPXE CentOS entry. After that continue to install CentOS as normal. CentOS will automatically see you iSCSI disk as a storage

option. After the installation is complete continue to the final step. PXE Setup Lastly we need to create another iPXE script and add an entry to the PXE menu.

```
nano /tftpboot/iscsi/centos.iscsi
```

```
#!ipxe
dhcp
set keep-san 1
sanboot iscsi:192.168.0.4:::iqn.2014-06.com.it-joe:centos-server
```

Finally, just add an entry to your PXE menu

```
nano /tftpboot/prelinux.cfg/default
```

```
LABEL CentOS iSCSI
MENU LABEL CentOS iSCSI
KERNEL ipxe.lkern
INITRD centos.iscsi
```

Please enable JavaScript to view the [comments powered by Disqus](#).

From:

<http://www.it-joe.com/> - **iT-Joe**

Permanent link:

http://www.it-joe.com/linux/iscsi_pxe_boot

Last update: **2018/04/01 03:11**

