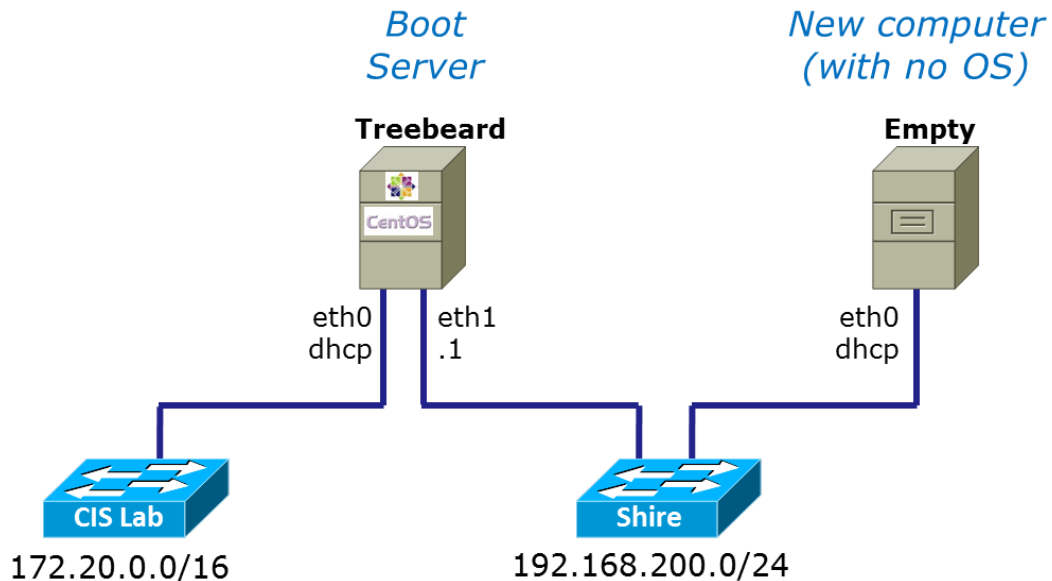**CentOS installation using PXE (203)**

This Howto documents the configuration of a PXE based boot server for doing Linux installations. PXE (Preboot Execution Environment) allows a new "empty" computer to use the network to boot up and install an operating system.

The PXE-based boot server (Treebeard) will have DHCP, TFTP and a web server configured and running. The "empty" computer will do a PXE boot, get an IP address using DHCP, then get the Linux kernel and initial RAM disk files using tftp. Next, HTTP is used to get installation files via the web server to do the install.

PXELINUX (a derivative of SYSLINUX) will be used to orchestrate the initial boot. The Anaconda installer using a kickstart file will automate the Linux installation.

The boot server, named Treebeard, will have repositories for the PXELINUX boot files, a kickstart file and all the files normally found on a Linux Distribution installation CD/DVD.

The Linux kernel (**vmlinuz**), initial RAM disk (**initrd.img**) and PXELINUX boot files are placed in the **/tftpboot** directory. These files will be transferred using TFTP to the new empty client when it first boots.

The kickstart files (**ks*.cfg**) and all the files found on a Linux distribution installation CD/DVD are placed in the **/var/www/html** directory. These files will be transferred using HTTP via the Apache web server to the new empty client.

Boot Server

Treebeard

/var/www/html/ — Installation and kickstart files

/var/lib/tftpboot — pxeboot files

Supplies

VMs:
- Treebeard (CentOS 6.4)
    - VMware version 8
    - One processor
    - 1 GB RAM
    - 16 GB SCSI HD
        - Thin Provisioning
        - VMware Paravirtual Controller
    - 2 NICs (E1000)

- Empty (no OS installed)
    - Choose Custom mode when creating
    - Select Red Hat Enterprise 6 (64 bit)
    - 1 GB MB RAM
    - 16 GB SCSI HD
        - Thin Provisioning
        - VMware Paravirtual Controller
    - 2 NICs (E1000)

ISOs:
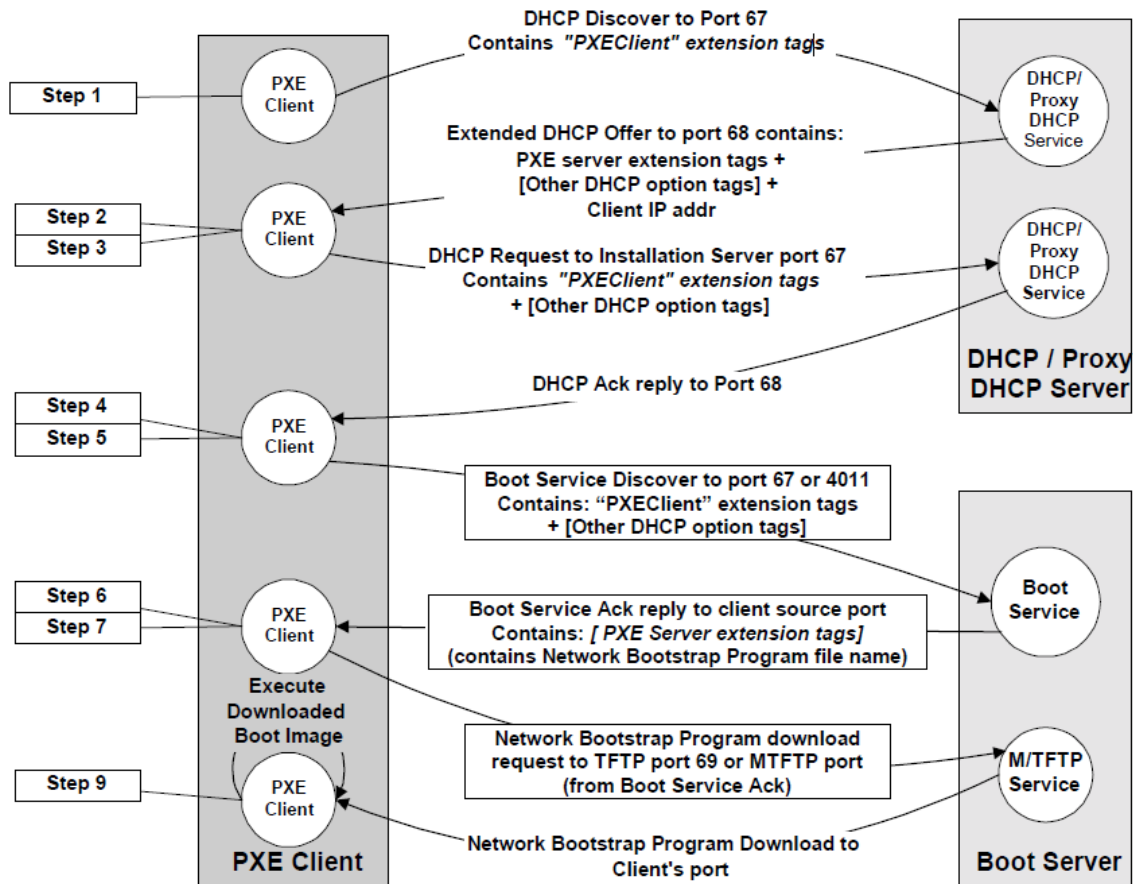- CentOS-6.4-x86_64-minimal (64-bit)

PXE Boot Sequence

Figure 2-1 PXE Boot

Source: http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf

Configure Treebeard to be a PXE boot server

1) Configure the network interfaces:
   o   eth0 is dhcp
   o   eth1 is static 192.168.200.1/24
   o   enable IP forwarding (`echo 1 > /proc/sys/net/ipv4/ip_forward`)

   Note: eth0 will connect to your physical network. eth1 will connect to the virtual Shire network. The Empty VM will also be connected to the Shire virtual network.

   Forwarding is enabled so the Empty VM will be able to get to the Internet via Treebeard.

   See Treebeard Network Interface Configuration below for more information.

2) Install (if necessary) and configure the DHCP service.

```
[root@treebeard ~]# yum install dhcp
```

Configure **/etc/dhcp/dhcpd.conf** file as shown in the example below. Make sure the next-server entry has the IP address of the boot (tftp) server. In this example we are using Treebeard at 192.168.200.1:

```
[root@treebeard 6.4]# cat /etc/dhcp/dhcpd.conf
ddns-update-style interim;

# Added for PXE boot support
allow booting;
allow bootp;
option option-128 code 128 = string;
option option-129 code 129 = text;
next-server 192.168.200.1;
filename "pxelinux.0";

ignore client-updates;
option time-offset              -25200; #PDT

subnet 192.168.200.0 netmask 255.255.255.0 {
option routers                  192.168.200.1;
option subnet-mask              255.255.255.0;
option domain-name              "localdomain";
option domain-name-servers      172.20.0.1;

range dynamic-bootp             192.168.200.150 192.168.200.199;
default-lease-time              21600;
max-lease-time                  43200;
}

[root@treebeard 6.4]#
```

Use **service dhcpd restart** after making the changes.
Use **chkconfig dhcpd on** so the service starts automatically during system boot.

3) Install and configure the tftp-server package. After a client gets an IP address using dhcp then tftp is used to pull down the linux kernel and initial RAM disk files.

```
[root@treebeard ~]# yum install tftp-server
```

tftp runs under the xinetd super-daemon umbrella. By default it is disabled. Edit the **/etc/xinetd.d/tftp** file and set "**disable**" to "**no**". In addition, add the **–vvv** option to increase what gets logged.

```
[root@treebeard ~]# cat /etc/xinetd.d/tftp
# default: off
# description: The tftp server serves files using the trivial file transfer \
#       protocol.  The tftp protocol is often used to boot diskless \
#       workstations, download configuration files to network-aware printers, \
#       and to start the installation process for some operating systems.
service tftp
{
        socket_type             = dgram
        protocol                = udp
```

```
        wait                    = yes
        user                    = root
        server                  = /usr/sbin/in.tftpd
        server_args             = -vvv -s /tftpboot
        disable                 = no
        per_source              = 11
        cps                     = 100 2
        flags                   = IPv4
}
[root@treebeard ~]#
```

Note: the **-s** is the *secure* option so tftp runs under chroot. **-vvv** is *very very verbose* so all transferred files get logged in **/var/log/messages**.

Restart the xinetd service so the tftp configuration change takes effect:

```
[root@treebeard ~]# service xinetd restart
Stopping xinetd:                                        [  OK  ]
Starting xinetd:                                        [  OK  ]
[root@treebeard ~]#
```

4) Install (if necessary) the http service (Apache web server) and configure it.

If you need to install the Apache web server:
- [root@treebeard ~]# **yum install httpd**
- Set the **ServerName** directive in **/etc/httpd/conf/httpd.conf**
  to your server's IP address and port 80 (e.g. **ServerName 192.168.200.1:80**)
- Start Apache with: **service httpd start**
- Restart at boot: **chkconfig httpd on**

To check status:
```
[root@treebeard ~]# service httpd status
httpd (pid 4092 4091 4090 4089 4088 4087 4086 4085 4083) is running...
[root@treebeard ~]#
```

To start Apache web server:
```
[root@treebeard ~]# service httpd start
Starting httpd:                                         [  OK  ]
[root@treebeard ~]#
```

5) The firewall needs to be adjusted to allow incoming http and tftp requests. Open TCP port 80 and UDP port 69 in the firewall.

```
[root@treebeard 6.4]# cat /etc/sysconfig/iptables
# Generated by iptables-save v1.4.7 on Mon Jul 15 17:55:39 2013
*nat
:PREROUTING ACCEPT [2706:282958]
:POSTROUTING ACCEPT [1:48]
:OUTPUT ACCEPT [11:1165]
-A POSTROUTING -o eth0 -j MASQUERADE
COMMIT
# Completed on Mon Jul 15 17:55:39 2013
# Generated by iptables-save v1.4.7 on Mon Jul 15 17:55:39 2013
*filter
```

```
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [3288:3066398]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p udp -m state --state NEW -m udp --dport 69 -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 80 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Mon Jul 15 17:55:39 2013
[root@treebeard 6.4]#
```

Using the command line is preferable. In addition to opening the ports 69 and 80 you can set up a NAT service for the 192.168.200.0/24 network and enable packet forwarding. Be sure to delete the rule on the FORWARD chain.

6) Install the syslinux package. The Syslinux project produces lightweight bootloaders that can be used to boot OS's form hard drives, CDs (ISOLINUX) and over the network (PXELINUX).

```
[root@treebeard ~]# yum install syslinux
```

8) Copy Syslinux files to the /tftpboot directory

```
[root@treebeard ~]# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
[root@treebeard ~]# cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot/
[root@treebeard ~]# cp /usr/share/syslinux/memdisk /var/lib/tftpboot/
[root@treebeard ~]# cp /usr/share/syslinux/mboot.c32 /var/lib/tftpboot/
[root@treebeard ~]# cp /usr/share/syslinux/chain.c32 /var/lib/tftpboot/
```

9) Make a new directory for the PXE boot menus

```
[root@treebeard ~]# mkdir /tftpboot/pxelinux.cfg
[root@treebeard ~]#
```

10) Create a file named default in that directory. Note the kernel and append commands must be on one line (even though they may be shown as wrapped in the examples below):

```
[root@treebeard 6.4]# cat /var/lib/tftpboot/pxelinux.cfg/default
default menu.c32
prompt 0

MENU TITLE PXE Menu

LABEL minimal
        MENU LABEL centos-6.4-64 (minimal install)
        kernel images/centos-6.4-64/vmlinuz
        append initrd=images/centos-6.4-64/initrd.img nomodeset ksdevice=bootif
ip=dhcp ks=http://192.168.200.1/mirrors/centos/6.4/ks.cfg

LABEL interactive
        MENU LABEL centos-6.4-64 (interactive)
        kernel images/centos-6.4-64/vmlinuz
```

```
        append initrd=images/centos-6.4-64/initrd.img nomodeset ksdevice=bootif
ip=dhcp ks=http://192.168.200.1/mirrors/centos/6.4/ks-interactive.cfg

[root@treebeard 6.4]#
```

11) Place Linux kernel(s) and initial RAM disk image(s) into **/tftpboot** directory for tftp transfers. For each Linux distribution to install make a directory for the kernel and initial ram disk images:

```
[root@treebeard ~]# mkdir -p /var/lib/tftpboot/images/centos-6.4-64/
```

12) For each Linux distribution to install, mount the installation CD and copy the **vmlinuz** (kernel) and **initrd.img** (initial RAM disk) image files to your /var/lib/tftpboot/images/ directory as shown in the example below:

```
root@treebeard ~]# mount /dev/cdrom /mnt
mount: block device /dev/sr0 is write-protected, mounting read-only

root@treebeard ~]# mount
/dev/mapper/vg_centos6net-lv_root on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw,rootcontext="system_u:object_r:tmpfs_t:s0")
/dev/sda1 on /boot type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
/dev/sr0 on /mnt type iso9660 (ro)

[root@treebeard ~]# cd /var/lib/tftpboot/images/centos-6.4-64/
[root@treebeard centos-6.4-64]# cp /mnt/isolinux/vmlinuz .
[root@treebeard centos-6.4-64]# cp /mnt/isolinux/initrd.img .

[root@treebeard ~]# tree /var/lib/tftpboot/
/var/lib/tftpboot/
├── chain.c32
├── images
│   └── centos-6.4-64
│       ├── initrd.img
│       └── vmlinuz
├── mboot.c32
├── memdisk
├── menu.c32
├── pxelinux.0
└── pxelinux.cfg
    └── default

3 directories, 8 files
[root@treebeard ~]#
```

13) Copy all the files on the Linux distribution DVD/CDs to your web document root location. These files will now be available using the http protocol to new clients.

```
[root@treebeard ~]# mkdir -p /var/www/html/mirrors/centos/6.4/
[root@treebeard ~]# cd /var/www/html/mirrors/centos/6.4/
[root@treebeard 6.4]# cp -r /mnt/* .
```

You can name these directories as you wish but they need to be configured in your PXELINUX **default** file and kickstart file (**ks.cfg**)

```
[root@treebeard ~]# tree /var/www/html/mirrors/centos/6.4/
/var/www/html/mirrors/centos/6.4/
├── CentOS_BuildTag
├── EFI
│   ├── BOOT
│   │   ├── BOOTX64.conf
│   │   ├── BOOTX64.efi
│   │   ├── splash.xpm.gz
│   │   └── TRANS.TBL
│   └── TRANS.TBL
├── EULA
├── GPL
├── images
│   ├── efiboot.img
│   ├── efidisk.img
│   ├── install.img
│   ├── TRANS.TBL
│   └── updates.img
├── isolinux
│   ├── boot.cat
│   ├── boot.msg
│   ├── grub.conf
│   ├── initrd.img
│   ├── isolinux.bin
│   ├── isolinux.cfg
│   ├── memtest
│   ├── splash.jpg
│   ├── TRANS.TBL
│   ├── vesamenu.c32
│   └── vmlinuz
├── ks.cfg
├── ks-interactive.cfg
├── Packages
│   ├── acl-2.2.49-6.el6.x86_64.rpm
│   ├── aic94xx-firmware-30-2.el6.noarch.rpm
│   ├── atmel-firmware-1.3-7.el6.noarch.rpm
│   ├── attr-2.4.44-7.el6.x86_64.rpm
│   ├── audit-2.2-2.el6.x86_64.rpm
│   ├── audit-libs-2.2-2.el6.x86_64.rpm
│   ├── authconfig-6.1.12-13.el6.x86_64.rpm
│   ├── b43-openfwwf-5.2-4.el6.noarch.rpm
│   ├── basesystem-10.0-4.el6.noarch.rpm
│   ├── bash-4.1.2-14.el6.x86_64.rpm
│   ├── bfa-firmware-3.0.3.1-1.el6.noarch.rpm
│   ├── binutils-2.20.51.0.2-5.36.el6.x86_64.rpm
│   ├── bridge-utils-1.2-10.el6.x86_64.rpm
│   ├── bzip2-1.0.5-7.el6_0.x86_64.rpm
│   ├── bzip2-libs-1.0.5-7.el6_0.x86_64.rpm
│   ├── ca-certificates-2010.63-3.el6_1.5.noarch.rpm
│   ├── centos-release-6-4.el6.centos.10.x86_64.rpm
│   ├── checkpolicy-2.0.22-1.el6.x86_64.rpm
│   ├── chkconfig-1.3.49.3-2.el6.x86_64.rpm
│   ├── coreutils-8.4-19.el6.x86_64.rpm
│   ├── coreutils-libs-8.4-19.el6.x86_64.rpm
│   ├── cpio-2.10-11.el6_3.x86_64.rpm
│   ├── cracklib-2.8.16-4.el6.x86_64.rpm
│   ├── cracklib-dicts-2.8.16-4.el6.x86_64.rpm
│   ├── cronie-1.4.4-7.el6.x86_64.rpm
│   ├── cronie-anacron-1.4.4-7.el6.x86_64.rpm
│   ├── crontabs-1.10-33.el6.noarch.rpm
│   ├── cryptsetup-luks-1.2.0-7.el6.x86_64.rpm
```

```
├── cryptsetup-luks-libs-1.2.0-7.el6.x86_64.rpm
├── curl-7.19.7-35.el6.x86_64.rpm
├── cyrus-sasl-2.1.23-13.el6_3.1.x86_64.rpm
├── cyrus-sasl-lib-2.1.23-13.el6_3.1.x86_64.rpm
├── dash-0.5.5.1-4.el6.x86_64.rpm
├── db4-4.7.25-17.el6.x86_64.rpm
├── db4-utils-4.7.25-17.el6.x86_64.rpm
├── dbus-glib-0.86-5.el6.x86_64.rpm
├── dbus-libs-1.2.24-7.el6_3.x86_64.rpm
├── device-mapper-1.02.77-9.el6.x86_64.rpm
├── device-mapper-event-1.02.77-9.el6.x86_64.rpm
├── device-mapper-event-libs-1.02.77-9.el6.x86_64.rpm
├── device-mapper-libs-1.02.77-9.el6.x86_64.rpm
├── device-mapper-multipath-0.4.9-64.el6.x86_64.rpm
├── device-mapper-multipath-libs-0.4.9-64.el6.x86_64.rpm
├── device-mapper-persistent-data-0.1.4-1.el6.x86_64.rpm
├── dhclient-4.1.1-34.P1.el6.centos.x86_64.rpm
├── dhcp-common-4.1.1-34.P1.el6.centos.x86_64.rpm
├── diffutils-2.8.1-28.el6.x86_64.rpm
├── dracut-004-303.el6.noarch.rpm
├── dracut-kernel-004-303.el6.noarch.rpm
├── e2fsprogs-1.41.12-14.el6.x86_64.rpm
├── e2fsprogs-libs-1.41.12-14.el6.x86_64.rpm
├── efibootmgr-0.5.4-10.el6.x86_64.rpm
├── elfutils-libelf-0.152-1.el6.x86_64.rpm
├── ethtool-3.5-1.el6.x86_64.rpm
├── expat-2.0.1-11.el6_2.x86_64.rpm
├── file-5.04-15.el6.x86_64.rpm
├── file-libs-5.04-15.el6.x86_64.rpm
├── filesystem-2.4.30-3.el6.x86_64.rpm
├── findutils-4.4.2-6.el6.x86_64.rpm
├── fipscheck-1.2.0-7.el6.x86_64.rpm
├── fipscheck-lib-1.2.0-7.el6.x86_64.rpm
├── fuse-2.8.3-4.el6.x86_64.rpm
├── gamin-0.1.10-9.el6.x86_64.rpm
├── gawk-3.1.7-10.el6.x86_64.rpm
├── gdbm-1.8.0-36.el6.x86_64.rpm
├── glib2-2.22.5-7.el6.x86_64.rpm
├── glibc-2.12-1.107.el6.x86_64.rpm
├── glibc-common-2.12-1.107.el6.x86_64.rpm
├── gmp-4.3.1-7.el6_2.2.x86_64.rpm
├── gnupg2-2.0.14-4.el6.x86_64.rpm
├── gpgme-1.1.8-3.el6.x86_64.rpm
├── grep-2.6.3-3.el6.x86_64.rpm
├── groff-1.18.1.4-21.el6.x86_64.rpm
├── grub-0.97-81.el6.x86_64.rpm
├── grubby-7.0.15-3.el6.x86_64.rpm
├── gzip-1.3.12-18.el6.x86_64.rpm
├── hwdata-0.233-7.9.el6.noarch.rpm
├── info-4.13a-8.el6.x86_64.rpm
├── initscripts-9.03.38-1.el6.centos.x86_64.rpm
├── iproute-2.6.32-23.el6.x86_64.rpm
├── iptables-1.4.7-9.el6.x86_64.rpm
├── iptables-ipv6-1.4.7-9.el6.x86_64.rpm
├── iputils-20071127-16.el6.x86_64.rpm
├── ipw2100-firmware-1.3-11.el6.noarch.rpm
├── ipw2200-firmware-3.1-4.el6.noarch.rpm
├── iscsi-initiator-utils-6.2.0.873-2.el6.x86_64.rpm
├── ivtv-firmware-20080701-20.2.noarch.rpm
├── iwl1000-firmware-39.31.5.1-1.el6.noarch.rpm
├── iwl100-firmware-39.31.5.1-1.el6.noarch.rpm
├── iwl3945-firmware-15.32.2.9-4.el6.noarch.rpm
├── iwl4965-firmware-228.61.2.24-2.1.el6.noarch.rpm
```

```
├── iwl5000-firmware-8.83.5.1_1-1.el6_1.1.noarch.rpm
├── iwl5150-firmware-8.24.2.2-1.el6.noarch.rpm
├── iwl6000-firmware-9.221.4.1-1.el6.noarch.rpm
├── iwl6000g2a-firmware-17.168.5.3-1.el6.noarch.rpm
├── iwl6050-firmware-41.28.5.1-2.el6.noarch.rpm
├── kbd-1.15-11.el6.x86_64.rpm
├── kbd-misc-1.15-11.el6.noarch.rpm
├── kernel-2.6.32-358.el6.x86_64.rpm
├── kernel-firmware-2.6.32-358.el6.noarch.rpm
├── keyutils-libs-1.4-4.el6.x86_64.rpm
├── kpartx-0.4.9-64.el6.x86_64.rpm
├── krb5-libs-1.10.3-10.el6.x86_64.rpm
├── less-436-10.el6.x86_64.rpm
├── libacl-2.2.49-6.el6.x86_64.rpm
├── libaio-0.3.107-10.el6.x86_64.rpm
├── libattr-2.4.44-7.el6.x86_64.rpm
├── libblkid-2.17.2-12.9.el6.x86_64.rpm
├── libcap-2.16-5.5.el6.x86_64.rpm
├── libcap-ng-0.6.4-3.el6_0.1.x86_64.rpm
├── libcom_err-1.41.12-14.el6.x86_64.rpm
├── libcurl-7.19.7-35.el6.x86_64.rpm
├── libdrm-2.4.39-1.el6.x86_64.rpm
├── libedit-2.11-4.20080712cvs.1.el6.x86_64.rpm
├── libertas-usb8388-firmware-5.110.22.p23-3.1.el6.noarch.rpm
├── libffi-3.0.5-3.2.el6.x86_64.rpm
├── libgcc-4.4.7-3.el6.x86_64.rpm
├── libgcrypt-1.4.5-9.el6_2.2.x86_64.rpm
├── libgpg-error-1.7-4.el6.x86_64.rpm
├── libidn-1.18-2.el6.x86_64.rpm
├── libnih-1.0.1-7.el6.x86_64.rpm
├── libpciaccess-0.13.1-2.el6.x86_64.rpm
├── libselinux-2.0.94-5.3.el6.x86_64.rpm
├── libselinux-utils-2.0.94-5.3.el6.x86_64.rpm
├── libsemanage-2.0.43-4.2.el6.x86_64.rpm
├── libsepol-2.0.41-4.el6.x86_64.rpm
├── libss-1.41.12-14.el6.x86_64.rpm
├── libssh2-1.4.2-1.el6.x86_64.rpm
├── libstdc++-4.4.7-3.el6.x86_64.rpm
├── libudev-147-2.46.el6.x86_64.rpm
├── libusb-0.1.12-23.el6.x86_64.rpm
├── libuser-0.56.13-5.el6.x86_64.rpm
├── libutempter-1.1.5-4.1.el6.x86_64.rpm
├── libuuid-2.17.2-12.9.el6.x86_64.rpm
├── libxml2-2.7.6-8.el6_3.4.x86_64.rpm
├── logrotate-3.7.8-16.el6.x86_64.rpm
├── lua-5.1.4-4.1.el6.x86_64.rpm
├── lvm2-2.02.98-9.el6.x86_64.rpm
├── lvm2-libs-2.02.98-9.el6.x86_64.rpm
├── m4-1.4.13-5.el6.x86_64.rpm
├── MAKEDEV-3.24-6.el6.x86_64.rpm
├── mdadm-3.2.5-4.el6.x86_64.rpm
├── mingetty-1.08-5.el6.x86_64.rpm
├── module-init-tools-3.9-21.el6.x86_64.rpm
├── mysql-libs-5.1.66-2.el6_3.x86_64.rpm
├── ncurses-5.7-3.20090208.el6.x86_64.rpm
├── ncurses-base-5.7-3.20090208.el6.x86_64.rpm
├── ncurses-libs-5.7-3.20090208.el6.x86_64.rpm
├── net-tools-1.60-110.el6_2.x86_64.rpm
├── newt-0.52.11-3.el6.x86_64.rpm
├── newt-python-0.52.11-3.el6.x86_64.rpm
├── nspr-4.9.2-1.el6.x86_64.rpm
├── nss-3.14.0.0-12.el6.x86_64.rpm
├── nss-softokn-3.12.9-11.el6.x86_64.rpm
```

```
├── nss-softokn-freebl-3.12.9-11.el6.x86_64.rpm
├── nss-sysinit-3.14.0.0-12.el6.x86_64.rpm
├── nss-tools-3.14.0.0-12.el6.x86_64.rpm
├── nss-util-3.14.0.0-2.el6.x86_64.rpm
├── openldap-2.4.23-31.el6.x86_64.rpm
├── openssh-5.3p1-84.1.el6.x86_64.rpm
├── openssh-clients-5.3p1-84.1.el6.x86_64.rpm
├── openssh-server-5.3p1-84.1.el6.x86_64.rpm
├── openssl-1.0.0-27.el6.x86_64.rpm
├── pam-1.1.1-13.el6.x86_64.rpm
├── passwd-0.77-4.el6_2.2.x86_64.rpm
├── pciutils-libs-3.1.10-2.el6.x86_64.rpm
├── pcre-7.8-6.el6.x86_64.rpm
├── pinentry-0.7.6-6.el6.x86_64.rpm
├── plymouth-0.8.3-27.el6.centos.x86_64.rpm
├── plymouth-core-libs-0.8.3-27.el6.centos.x86_64.rpm
├── plymouth-scripts-0.8.3-27.el6.centos.x86_64.rpm
├── policycoreutils-2.0.83-19.30.el6.x86_64.rpm
├── popt-1.13-7.el6.x86_64.rpm
├── postfix-2.6.6-2.2.el6_1.x86_64.rpm
├── procps-3.2.8-25.el6.x86_64.rpm
├── psmisc-22.6-15.el6_0.1.x86_64.rpm
├── pth-2.0.7-9.3.el6.x86_64.rpm
├── pygpgme-0.1-18.20090824bzr68.el6.x86_64.rpm
├── python-2.6.6-36.el6.x86_64.rpm
├── python-iniparse-0.3.1-2.1.el6.noarch.rpm
├── python-libs-2.6.6-36.el6.x86_64.rpm
├── python-pycurl-7.19.0-8.el6.x86_64.rpm
├── python-urlgrabber-3.9.1-8.el6.noarch.rpm
├── ql2100-firmware-1.19.38-3.1.el6.noarch.rpm
├── ql2200-firmware-2.02.08-3.1.el6.noarch.rpm
├── ql23xx-firmware-3.03.27-3.1.el6.noarch.rpm
├── ql2400-firmware-5.08.00-1.el6.noarch.rpm
├── ql2500-firmware-5.08.00-1.el6.noarch.rpm
├── readline-6.0-4.el6.x86_64.rpm
├── redhat-logos-60.0.14-12.el6.centos.noarch.rpm
├── rootfiles-8.1-6.1.el6.noarch.rpm
├── rpm-4.8.0-32.el6.x86_64.rpm
├── rpm-libs-4.8.0-32.el6.x86_64.rpm
├── rpm-python-4.8.0-32.el6.x86_64.rpm
├── rsyslog-5.8.10-6.el6.x86_64.rpm
├── rt61pci-firmware-1.2-7.el6.noarch.rpm
├── rt73usb-firmware-1.8-7.el6.noarch.rpm
├── sed-4.2.1-10.el6.x86_64.rpm
├── selinux-policy-3.7.19-195.el6.noarch.rpm
├── selinux-policy-targeted-3.7.19-195.el6.noarch.rpm
├── setup-2.8.14-20.el6.noarch.rpm
├── shadow-utils-4.1.4.2-13.el6.x86_64.rpm
├── slang-2.2.1-1.el6.x86_64.rpm
├── sqlite-3.6.20-1.el6.x86_64.rpm
├── sudo-1.8.6p3-7.el6.x86_64.rpm
├── system-config-firewall-base-1.2.27-5.el6.noarch.rpm
├── sysvinit-tools-2.87-4.dsf.el6.x86_64.rpm
├── tar-1.23-11.el6.x86_64.rpm
├── tcp_wrappers-libs-7.6-57.el6.x86_64.rpm
├── TRANS.TBL
├── tzdata-2012j-1.el6.noarch.rpm
├── udev-147-2.46.el6.x86_64.rpm
├── upstart-0.6.5-12.el6.x86_64.rpm
├── ustr-1.0.4-9.1.el6.x86_64.rpm
├── util-linux-ng-2.17.2-12.9.el6.x86_64.rpm
├── vim-minimal-7.2.411-1.8.el6.x86_64.rpm
├── which-2.19-6.el6.x86_64.rpm
```

```
│       ├── xfsprogs-3.1.1-10.el6.x86_64.rpm
│       ├── xorg-x11-drv-ati-firmware-6.99.99-1.el6.noarch.rpm
│       ├── xz-libs-4.999.9-0.3.beta.20091007git.el6.x86_64.rpm
│       ├── yum-3.2.29-40.el6.centos.noarch.rpm
│       ├── yum-metadata-parser-1.1.2-16.el6.x86_64.rpm
│       ├── yum-plugin-fastestmirror-1.1.30-14.el6.noarch.rpm
│       ├── zd1211-firmware-1.4-4.el6.noarch.rpm
│       └── zlib-1.2.3-29.el6.x86_64.rpm
├── RELEASE-NOTES-en-US.html
├── repodata
│       ├── 2bf9e31398aa64a1c8aa0ef6b7d962db7cb300e9c5a069aa71b7648f84d73906-other.xml.gz
│       ├── 34bae2d3c9c78e04ed2429923bc095005af1b166d1a354422c4c04274bae0f59-c6-minimal-
x86_64.xml
│       ├── 441c2721450dbfd99b7e9d7322c397fff31ee88dbe681f427b355a72866424de-
other.sqlite.bz2
│       ├── 5149a913a6298651ca56bc2e206dc592d9f5c4c72203e62c6957f8e874ba74e3-
filelists.sqlite.bz2
│       ├── 86dbd6361ab95972a3ae9af0872db0fbdf982d8dc124565acffd9700ed2f47ff-
primary.sqlite.bz2
│       ├── 967c01a5c36dbb12330480abc45d3017d4ec67af6785f1f5011e7e61e5e5c929-
primary.xml.gz
│       ├── b4408eb763ccbe450903df7b0c6937fbb3f9fe84cec5d4731ecb749866443ec9-
filelists.xml.gz
│       ├── ce2d698b9fb1413b668443e88835a0642cea8f387c7f25cc946f56dd93f109bb-c6-minimal-
x86_64.xml.gz
│       ├── repomd.xml
│       └── TRANS.TBL
├── RPM-GPG-KEY-CentOS-6
├── RPM-GPG-KEY-CentOS-Debug-6
├── RPM-GPG-KEY-CentOS-Security-6
├── RPM-GPG-KEY-CentOS-Testing-6
└── TRANS.TBL

6 directories, 267 files
[root@treebeard ~]#
```

14) Place your kickstart file where it can be retrieved using your web server. A kickstart file contains the answers to questions you normally answer during an interactive install. This allows the entire installation to be automated. In this example, the kickstart file is placed in the same location as the distribution files.

```
[root@treebeard ~]# cat /var/www/html/mirrors/centos/6.4/ks.cfg
# Kickstart file automatically generated by anaconda.

install
url --url http://192.168.200.1/mirrors/centos/6.4/
lang en_US.UTF-8
keyboard us

network --onboot yes --device eth0 --bootproto dhcp --hostname=romeo

rootpw  --iscrypted
$6$MNvw9Xv9s5Q7RxI5$PrBYPsPta9FGkCduZC2Corbw2T3IiANLdCTBvVFA4j6atzmRpyFqMMjMZoXDfdAuoi
t89botl1.W1kpcZPtGJ1

firewall --service=ssh
authconfig --enableshadow --passalgo=sha512
selinux --enforcing
timezone --utc America/Los_Angeles

bootloader --location=mbr --driveorder=sda
```

```
zerombr
clearpart --all --drives=sda

# Method 1 - automatic partioning
#autopart

# Method 2 - fixed physical partitions
#part swap --size=512
#part /home --fstype=ext4 --size=1024
#part /boot --fstype=ext2 --size=128       --asprimary
#part /      --fstype=ext4 --size=1 --grow

# Method 3 - Logical Volumes
part swap        --size=512
part /boot       --size=128       --asprimary
part pv.01       --size=1 --grow
volgroup vg00 pv.01
logvol /         --vgname=vg00   --fstype=ext3   --size=1 --grow     --name=lvroot
logvol /home     --vgname=vg00   --fstype=ext3   --size=512          --name=lvhome


%packages --nobase
@core
%end
[root@treebeard ~]# cat /var/www/html/mirrors/centos/6.4/ks-interactive.cfg
# Kickstart file automatically generated by anaconda.

install
interactive
url --url http://192.168.200.1/mirrors/centos/6.4/
lang en_US.UTF-8
keyboard us

network --onboot yes --device eth0 --bootproto dhcp --hostname=romeo

rootpw  --iscrypted
$6$MNvw9Xv9s5Q7RxI5$PrBYPsPta9FGkCduZC2Corbw2T3IiANLdCTBvVFA4j6atzmRpyFqMMjMZoXDfdAuoi
t89botl1.W1kpcZPtGJ1

firewall --service=ssh
authconfig --enableshadow --passalgo=sha512
selinux --enforcing
timezone --utc America/Los_Angeles

bootloader --location=mbr --driveorder=sda

zerombr
clearpart --all --drives=sda

# Method 1 - automatic partioning
#autopart

# Method 2 - fixed physical partitions
#part swap --size=512
#part /home --fstype=ext4 --size=1024
#part /boot --fstype=ext2 --size=128       --asprimary
#part /      --fstype=ext4 --size=1 --grow

# Method 3 - Logical Volumes
part swap        --size=512
part /boot       --size=128       --asprimary
part pv.01       --size=1 --grow
```
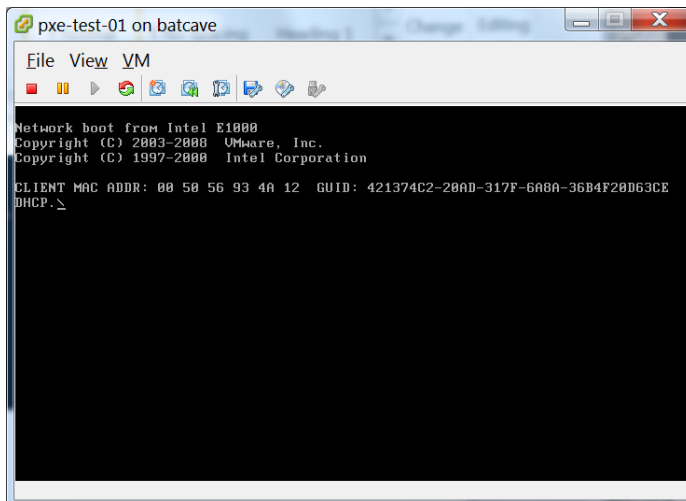
```
volgroup vg00 pv.01
logvol /        --vgname=vg00   --fstype=ext3   --size=1 --grow    --name=lvroot
logvol /home    --vgname=vg00   --fstype=ext3   --size=512         --name=lvhome


%packages --nobase
@core
%end
[root@treebeard ~]#
```
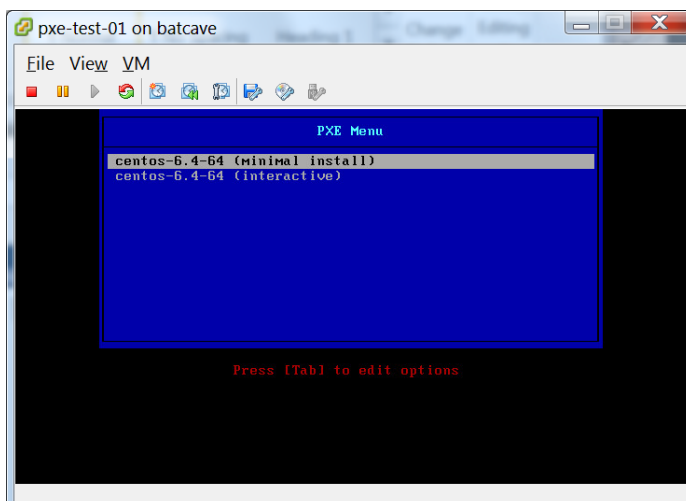
Installing Linux into new VMs

Now that you have the PXE boot server configure it is time to try it out.

1) Create a new VM named empty with at least 512 MB of RAM and 5 GB for the hard drive.

2) Edit the VM settings and configure the Network Adapter 1 to connect to Shire

3) Start the VM

3) Select the first install option

4) Sit back and enjoy the install. The hard drive will be partitioned, the OS packages installed.

Treebeard Network Interface Configuration

The two NICs are permanently configured for dhcp and static addresses. Note your MAC addresses will differ as they are randomly generated by VMware for each VM:

```
[root@treebeard ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE="eth0"
BOOTPROTO="dhcp"
HWADDR="00:50:56:93:A3:17"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
UUID="59e5b891-90fb-4e4d-9ef6-c27b68d02653"
[root@treebeard ~]#

[root@treebeard ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth1
DEVICE="eth1"
BOOTPROTO="static"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
NM_CONTROLLED="no"
ONBOOT="yes"
TYPE="Ethernet"
IPADDR=192.168.200.1
NETMASK=255.255.255.0
[root@treebeard ~]#

[root@treebeard ~]# cat /etc/sysconfig/network
NETWORKING=yes
HOSTNAME=treebeard.localdomain
[root@treebeard ~]#
```

Remember to use the following command to have the NIC configuration files take effect:

```
[root@treebeard ~]# service network restart
Shutting down interface eth0:                        [  OK  ]
Shutting down interface eth1:                        [  OK  ]
Shutting down loopback interface:                    [  OK  ]
Bringing up loopback interface:                      [  OK  ]
Bringing up interface eth0:
Determining IP information for eth0... done.
                                                     [  OK  ]
Bringing up interface eth1:                          [  OK  ]
[root@treebeard ~]#
```

IP forwarding is enabled permanently by editing the **/etc/sysctl.conf** file:

```
[root@treebeard ~]# cat /etc/sysctl.conf
# Kernel sysctl configuration file for Red Hat Linux
#
# For binary values, 0 is disabled, 1 is enabled.  See sysctl(8) and
# sysctl.conf(5) for more details.

# Controls IP packet forwarding
net.ipv4.ip_forward = 1

# Controls source route verification
net.ipv4.conf.default.rp_filter = 1
```

```
# Do not accept source routing
net.ipv4.conf.default.accept_source_route = 0

# Controls the System Request debugging functionality of the kernel
kernel.sysrq = 0

# Controls whether core dumps will append the PID to the core filename.
# Useful for debugging multi-threaded applications.
kernel.core_uses_pid = 1

# Controls the use of TCP syncookies
net.ipv4.tcp_syncookies = 1

# Disable netfilter on bridges.
net.bridge.bridge-nf-call-ip6tables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-arptables = 0

# Controls the default maxmimum size of a mesage queue
kernel.msgmnb = 65536

# Controls the maximum size of a message, in bytes
kernel.msgmax = 65536

# Controls the maximum shared segment size, in bytes
kernel.shmmax = 68719476736

# Controls the maximum number of shared memory segments, in pages
kernel.shmall = 4294967296
[root@treebeard ~]#
```

Remember to use the following command to have the **/etc/sysctl.conf** configuration changes take effect:

```
[root@treebeard ~]# sysctl -p
net.ipv4.ip_forward = 1
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
kernel.sysrq = 0
kernel.core_uses_pid = 1
net.ipv4.tcp_syncookies = 1
error: "net.bridge.bridge-nf-call-ip6tables" is an unknown key
error: "net.bridge.bridge-nf-call-iptables" is an unknown key
error: "net.bridge.bridge-nf-call-arptables" is an unknown key
kernel.msgmnb = 65536
kernel.msgmax = 65536
kernel.shmmax = 68719476736
kernel.shmall = 4294967296
[root@treebeard ~]#
```

Use ifconfig to make sure your network settings are active:

```
[root@treebeard ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:56:93:A3:17
          inet addr:172.20.4.63  Bcast:172.20.255.255  Mask:255.255.0.0
          inet6 addr: 2001:470:82a6:1:250:56ff:fe93:a317/64 Scope:Global
          inet6 addr: fe80::250:56ff:fe93:a317/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:79393 errors:0 dropped:0 overruns:0 frame:0
          TX packets:36569 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

```
          RX bytes:47942276 (45.7 MiB)  TX bytes:4416465 (4.2 MiB)

eth1      Link encap:Ethernet  HWaddr 00:50:56:93:C6:11
          inet addr:192.168.200.1  Bcast:192.168.200.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe93:c611/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:870938 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2402007 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:55473192 (52.9 MiB)  TX bytes:3587522988 (3.3 GiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:328 (328.0 b)  TX bytes:328 (328.0 b)
```

References

CentOS PXE Setup
- http://wiki.centos.org/HowTos/PXE/PXE_Setup

SYSLINUX
- http://syslinux.zytor.com/wiki/index.php/SYSLINUX

Fedora Anaconda/Kickstart
- http://fedoraproject.org/wiki/Anaconda/Kickstart

RedHat EL4 Kickstart Options
- https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html

Intel PXE 2.1 Specification
- http://download.intel.com/design/archives/wfm/downloads/pxespec.pdf

Setting up a PXE-Boot Server (Net Llama!)
- http://linux-sxs.org/internet_serving/pxeboot.html

Automate RHEL Based OS Deployments with PXE Boot and Kickstart (Josh Bolling's Blog)
- http://www.joshbolling.com/blog/2013/01/automate-rhel-based-os-deployments-with-pxe-boot-and-kickstart/

Automated Installation of CentOS 6.x And Kickstart File (Unattended Installation)
- http://ktaraghi.blogspot.com/2012/09/automated-installation-of-centos-6x-and.html

Red Hat Enterprise Linux 6 Installation Guide
- https://access.redhat.com/site/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/index.html