Configuring diskless clients with Red Hat Enterprise Linux

Dave Kline

Beginning with Red Hat Enterprise Linux 5.1, Red Hat supports booting Linux via **NFSROOT**. This allows clients to mount root filesystems remotely on NFS servers. Combined with PXE (Preboot eXecution Environment), clients can boot and operate without local storage of any kind. Additionally, when using storage snapshots, many clients can operate from a single filesystem.

OVERVIEW

The process of enabling diskless operation (sometimes called netbooting) includes a number of steps. The client powers on, requests network and boot information, downloads a kernel and **initrd**, and finally mounts a root filesystem via NFS. This is similar to PXE booting for the purposes of kickstarting installations. However, rather than performing an installation, the client runs normally.

This article outlines how to create both a server capable of serving diskless clients, and a single diskless client.

1: CLIENT CONFIGURATION

The first step is to create a normal RHEL host. Install Red Hat Enterprise Linux 6 onto a host. After you've customized the host you will need to allow your image to be booted via NFS. The **dracut-network** package provides the necessary network components for RHEL 6 clients to mount network root filesystems:

```
# yum install dracut-network -y
```

If creating a RHEL 5 diskless client, you should install the **busybox-anaconda** package, and disable SELinux. In general, Red Hat does not recommend disabling SELinux. As such, you should carefully consider the security implications of this action.

The last step is to create an **initrd**. Note that in this example, the path on the NFS server that will hold our client is **/usr/local/netboot**.

RHEL 6 clients

Use the following command for a RHEL 6 client, noting the path and IP address of the NFS server:

```
# dracut -f /boot/netboot6.img `uname -r` root=dhcp root-
path=nfs:<server_ip>:/usr/local/netboot/
```

RHEL 5 clients

If we were creating a RHEL 5 client, we would use the **mkinitrd** command instead:

```
# mkinitrd -f --with=virtio_net --with=nfs --net-dev=eth0
--rootdev=<server_ip>:/usr/local/netboot/ --rootfs=nfs /boot/netboot.img `uname
-r`
```

In the **mkinitrd** example above, we have explicitly specified **virtio_net.** This is to ensure the **initrd** is created with the proper network driver for our environment. If your diskless client uses a different network driver, you can specify that it be included with the **--with=** parameter. Multiple drivers can be specified this way.

2: SERVER CONFIGURATION

A diskless server provides the following services:

- DHCP/PXE
- TFTP
- NFS

These requirements are easily handled by Red Hat Enterprise Linux, though you can also make use of any existing infrastructure that may provide them. We will be using a RHEL 6 host as the server, though the commands and package names needed are very similar on RHEL 5.

First, install needed packages, and ensure necessary daemons start at boot:

```
# yum install dhcp syslinux tftp-server -y
# yum groupinstall "NFS file server" -y
# chkconfig dhcpd on
# chkconfig tftp on
# chkconfig nfs on
```

Next, create directories to host filesystems and configuration information, and copy the **pxelinux.0** binary in place:

```
# mkdir /usr/local/netboot/
# mkdir /var/lib/tftpboot/pxelinux.cfg/
# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot/
```

Copy over the configured client we created earlier. A simple way of doing so is to use **rsync** from the server**:**

```
# rsync -av --progress --exclude=/proc --exclude=/sys root@CLIENT_HOST:/
/usr/local/netboot/
```

The DHCP service provides clients with network information, as well as the location of the PXELINUX network boot loader. The following is an example of a **dhcpd.conf** file, found under **/etc/dhcp/dhcpd.conf**:

```
ddns-update-style interim;
ignore client-updates;
allow booting;
allow bootp;
subnet 192.168.122.0 netmask 255.255.255.0 {
       option routers   192.168.122.1;
       option subnet-mask  255.255.255.0;
       option nis-domain  "domain.org";
       option domain-name  "domain.org";
       option domain-name-servers 192.168.122.1;
       option time-offset  -18000; # Eastern Standard Time
       default-lease-time 21600;
       max-lease-time 43200;

       host netboot6 {
    next-server <server_ip>;
    hardware ethernet <client_mac_address>;
    fixed-address <client_ip_address>;
    filename "pxelinux.0";
       }
}
```

In this example we have specified a client, **netboot6**, with the IP address of **<client_ip_address>**. The **next-server** parameter specifies where **pxelinux.0** can be downloaded (directly from the same server is quite common), and the MAC address that corresponds to **netboot6**: **<client_mac_address>**. Additional clients can be placed under different **host** declarations, provided that unique IP and MAC addresses are specified.

Once finished, restart dhcpd and look for errors in **/var/log/messages**:

```
# service dhcpd restart
```

Our client will mount its root filesystem via NFS, so we need to add the following to **/etc/exports**:

```
/usr/local/netboot/       <client_ip_address>(rw,async,no_root_squash)
```

Once complete, reload your changes with **exportfs -ra** or **service nfsd restart**. You can verify your exports by running **exportfs**.

Next we will enable extra verbosity for our tftp server, which is useful when clients initially request files.

Add **-vv** to following line under **/etc/xinetd.d/tftp**:

```
server_args       = -vv -s /var/lib/tftpboot
```

The client's kernel, and the **initrd** that we created earlier, should be copied and served via tftp with the following commands:

```
# cp /usr/local/netboot/boot/netboot6.img /var/lib/tftpboot/
# cp /usr/local/netboot/boot/vmlinuz-2.6.32-71.14.1.el6.x86_64
/var/lib/tftpboot
```

We will now create a configuration file that our client retrieves during boot. Create the following file as **/var/lib/tftpboot/pxelinux.cfg/default**:

```
default netboot6
timeout 30
prompt 1

label netboot6
kernel /vmlinuz-2.6.32-71.14.1.el6.x86_64
append initrd=/netboot6.img rw root=nfs:<nfs_server>:/usr/local/netboot/
selinux=0 enforcing=0
```

In this example, we have specified the NFS server and path, a timeout of three seconds, and a **netboot6** stanza containing a kernel and **initrd** declaration.

With our client's filesystem in place, we need to make a few server-side changes. We will need to create directories and an **/etc/fstab/** file, and disable the primary interface. Perform the following:

```
# mkdir /usr/local/netboot/{proc,sys}
# vi /usr/local/netboot/etc/fstab

<nfs_server>:/usr/local/netboot/ /     nfs   defaults    1 1
tmpfs          /dev/shm        tmpfs  defaults    0 0
devpts         /dev/pts        devpts gid=5,mode=620 0 0
sysfs          /sys          sysfs  defaults    0 0
proc          /proc          proc  defaults    0 0

# vi /usr/local/netboot/etc/sysconfig/network-scripts/ifcfg-eth0

DEVICE="eth0"
HWADDR="<client_mac_address">
```

```
NM_CONTROLLED="yes"
ONBOOT="no"
BOOTPROTO="dhcp"
```

With these steps in place, we're ready to boot our client.

3: BOOTING AND TROUBLESHOOTING THE DISKLESS CLIENT

With the server and client filesystem properly configured, we can now boot over the network. Before the boot begins, you may wish to view logs on the server with **tail -f /var/log/messages**.

Because of the large amount of configuration required, it's common for users to make mistakes. To avoid the more common errors, check that:

- the firewall settings permit NFS and DHCP traffic
- the server's SELinux settings aren't interfering with the client's ability to download files
- when copying files to **/var/lib/tftpboot/**, the files have read permissions
- MAC addresses were recorded properly in **/etc/dhcp/dhcpd.conf**
- the NFS server settings and kernel settings are properly configured in **/var/lib/tftpboot/pxelinux.cfg/default**.

Your diskless client is now ready to use.