PXE Install Environment

Authors:
Jennifer Beattie
Edmund Sutcliffe

Summary

This document provides detailed steps to set up a PXE network boot environment, including DHCP, TFTP and HTTP servers. This will be sufficient to allow CentOS/RedHat kickstart installs to be performed over the network.

PXE boot environments are a well-known process. We provide a particular formulation of this process to ease the reader through the steps, and to provide the prerequisites for our later uses of the environment, such as diskless booting.

Overview

Creating a PXE network boot environment is an established process in a Linux environment, and the technical requirements fall into several fundamental stages:

- Pre-reqs: Download and unpack an ISO image of a known base OS build. We use CentOS 5.4 x86_64. A build machine to install, and a test machine to PXE boot are also needed. Both of these can be virtual machines.
- Set up a DHCP server in the network. We use the build machine for this, but it can be any DHCP server where you have the power to set additional DHCP options for `next-server` and `filename` .
- Set up a TFTP server. It is most convenient for this to be on the build machine since it must serve the kernel image for network booting, but this is not a strict requirement.
- Set up an HTTP server on the build machine. We use this for serving install packages during kickstart installs. NFS can be used as an alternative, but we do not describe that here.
- Configure the DHCP server to point at the pxelinux files on the TFTP server in order to boot to a PXE menu. Configure the TFTP server to serve a suitable kernel image that can install using the HTTP repository.
- Network boot your test image.

Prerequisites

You must have a build machine to install on, and a test machine to PXE boot. Both these machines will be reinstalled during the process.

DNS must be set up to refer to these machines, although bare IP addresses can be used as a fallback.

You must have downloaded the CentOS 5.4 x86_64 (64 bit) ISO image. This can be found via: http://isoredirect.centos.org/centos/5/isos/x86_64/

Filesystem Paths and Network Addresses

Several of the pathnames used in this install process are configurable. This document assumes particular locations for these, and for ease of reading, the explanations below use these paths without qualifying them as potentially variable names.

The following paths are used as standard:

TFTP root directory	`/tftpboot`
NFS base directory	`/data/bootimage`
HTTP boot server filesystem location	`/data/bootserv`
CentOS HTTP repository root	`/data/bootserv/centos`

The network topology is assumed to be on a private network `10.1.4.0/24` in this example. The machines being created are intended to be part of a domain `test.example.com`.

We further assume that the build machine will be acting as the DNS nameserver for the local network, and this is reflected in the `resolv.conf` file we create. The build machine itself is named `core` in this example (ie `core.test.example.com`) and is assumed to be on `10.1.4.1`.

Process

The following steps should be followed in sequence, once the pre-requisites have been met.

Install the build server from the ISO image

We are trying to make this setup process as simple and consistent as possible. As such, this document assumes you have installed the build machine itself from the same ISO that will be used to install other machines. This simplifies matters because a known set of packages is available. Of course, this initial "bootstrapping" install must be done manually.

We assume that you have performed a default CentOS install and selected at least the "Server" group of packages for installation.

Of course, it is perfectly possible to turn a pre-existing machine into a build machine if you satisfy the need for a DHCP, TFTP and HTTP server in an alternative manner, including using a Windows server rather than a Linux machine.

Log into your installed server image, and obtain root access.
For brevity, we refer in this document to commands such as `service`, `yum` and `chkconfig` assuming that they are present on your PATH.

Copy the ISO image to the local disk

In this guide we place the packages the build machine will serve under `/data` .

Inside this container, we create a `bootserv` directory to use as the HTTP repository root, and inside *this* we create a directory to hold the CentOS install packages. Here we have simply called it `centos` .

Mount the CentOS ISO (here we assume it is available as a physical device) and copy the packages:

```
mkdir -p /data/bootserv/centos
mkdir -p /mnt/cdrom
mount -t iso9660 /dev/cdrom /mnt/cdrom -o ro
cp -r /mnt/cdrom/* /data/bootserv/centos/
umount /mnt/cdrom
```

Set up the DHCP server

First, we make sure it is installed and turned on:

```
yum -y install dhcp
chkconfig --level 345 dhcpd on
```

Now we need to put some sensible contents into the `/etc/dhcpd.conf` config file. Since we're setting up the DHCP daemon to run automatically at boot time, we really want to make sure that it only responds to selected clients, so the `deny unknown-clients;` config statement is crucial.

We also need to specify `not authoritative;` so that we know to ignore other requests, just in case we see any requests from clients asking for addresses on different network segments (if the network changed under our feet or other people are plugged into the same segment).

We specify the domain name, DNS servers, subnet mask, and router options, and we allow old fashioned bootp booting (in case it's needed). We don't do any dynamic DNS, or IP forwarding.

After that, we have to specify the IP address of the TFTP server using the `next-server` directive, and the `filename` of the PXE boot image for the client to actually request.

Finally, we specify individual host information by MAC address, so that we know which clients to serve (since we deny any unknown clients).

Here is the resulting config for the sample `10.1.4.0/24` network topology that was described at the beginning of this document, and test servers named `vapp1` and `vapp2` with MAC addresses `00:0c:29:01:02:03` and `00:0c:29:04:05:06` :

```
deny unknown-clients;
not authoritative;

option domain-name              "test.example.com";
option domain-name-servers      10.1.4.1;
option subnet-mask              255.255.255.0;

allow bootp;
#allow booting;

option ip-forwarding    false;  # No IP forwarding
option mask-supplier    false;  # Don't respond to ICMP Mask req

ddns-update-style none;

subnet 10.1.4.0 netmask 255.255.255.0 {
  option routers       10.1.4.1;
  option time-offset    -18000; # Eastern Standard Time
}

group {
  next-server 10.1.4.1;         # Name of your TFTP server
  filename "pxelinux.0";        # Name of the bootloader program

  host vapp1 {
        option host-name "vapp1.test.example.com";
        hardware ethernet 00:0c:29:01:02:03;
        fixed-address 10.1.4.101;
  }

  host vapp2 {
        option host-name "vapp2.test.example.com ";
        hardware ethernet 00:0c:29:04:05:06;
        fixed-address 10.1.4.102;
  }
}
```

NOTE: Obviously the MAC addresses for each host have to be manually configured, and in the case of VMWare Server you will need to power on the host to determine the automatically generated MAC address for each new VM instance.

Start (or restart) the DHCP service to pick up the new config:

```
service dhcpd start
```

We must also ensure that DHCP (UDP port 67 inbound) is reachable through the firewall.

You must therefore ensure that a line like:
`-A RH-Firewall-1-INPUT -p udp -m udp --dport 67  -j ACCEPT`
is present in `/etc/sysconfig/iptables` . We assume that outbound UDP is not restricted; otherwise you will need an additional line to allow UDP port 68 outbound.

Restart iptables if required (`service iptables restart`).

At this point we can verify that the test PXE boot host can be network booted, and it should be assigned a suitable address via DHCP if its MAC address matches the config. Its attempt to perform a TFTP request for the `pxelinux.0` file will of course fail until we have completed that step.

You can also test from another client on the same segment to make sure that only known clients are given any lease offers (say, using **DHCP Explorer** from a Windows box).

Set up the TFTP server

On CentOS, we use the `tftp-server` RPM, and the TFTP server runs out of `xinetd` rather than as a persistent daemon.

Ensure both are installed. Then we must comment out the disable = yes line in /etc/xinetd.d/tftp so that TFTP is available, and make sure that xinetd is available at boot time:

```
yum -y install xinetd tftp-server
perl -pi -e 's,(^\s*disable\s*=).*,$1 no,' /etc/xinetd.d/tftp
service xinetd restart
/sbin/chkconfig --level 345 xinetd on
```

Next, open up port 69 over UDP in iptables. A line like:
`-A RH-Firewall-1-INPUT -p udp -m udp --dport 69 -j ACCEPT`
must be present in `/etc/sysconfig/iptables` .

Restart iptables if required (`service iptables restart`).

Set up the PXE image and menu files

Now that we have a TFTP server, we need to copy the PXE boot images over for use on it. CentOS provides these in the `syslinux` package:

In fact, the system-config-netboot package also provides a copy of the pxelinux files inside the `/tftpboot/linux-install` directory, but we choose to keep control of the file structure for clarity, and make our own copies of the files we need.

Install the PXE image files

We install `syslinux`, and copy the `pxelinux.0` boot file into the tftpboot directory. Note that we have to copy it rather than use a symlink, because we are letting `xinetd` use its default secure setting which chroots into `/tftpboot`.
We also create a directory named `pxelinux.cfg` which we will need in the next step.

```
yum -y install syslinux
cp /usr/lib/syslinux/pxelinux.0 /tftpboot/pxelinux.0
mkdir -p /tftpboot/pxelinux.cfg
```

PXE config file locations

The next thing we need is the PXE config directory.
The PXE README file explains how you can provide a config file for specific machines, or groups of machines, based on their MAC address, or their IP address or a prefix of their IP address:

The PXELINUX boot image first tries to load a config file like:
`pxelinux.cfg/01-00-0c-29-01-02-03`
based on the MAC address.

Note: the MAC address is prefixed with `01-` signifying Ethernet in the above filename.

If that fails, it turns the DHCP-assigned IP address (in this example we show 192.168.1.129) into a big-endian hex string, and tries to load a more general config file by progressively stripping off one character at a time, like this:

```
pxelinux.cfg/C0A80181
pxelinux.cfg/C0A8018
pxelinux.cfg/C0A801
pxelinux.cfg/C0A80
pxelinux.cfg/C0A8
pxelinux.cfg/C0A
pxelinux.cfg/C0
pxelinux.cfg/C
```

If all of these methods fail, it finally tries to load
`pxelinux.cfg/default`
which is the only file we use in this document.

Configure PXE menu files

PXE booting allows you to display a menu screen (and in fact to change that text in response to function keys), before prompting the user for text input to select one of the available network boot options. You can also use escape codes to clear the screen, change text colors etc.

The menu config file syntax and display escape codes are described in the `syslinux` distribution, in the text file `syslinux.doc` , with a few additions in the file `pxelinux.doc` .

After some global options, you specify a series of `label` directives describing the kernel images that can be booted; each label has a corresponding `kernel` statement to say where the image file is, and an `append` statement for any kernel parameters. Other options can be found in the help file.

In addition, the file specifies a default label to boot from if the user simply presses Enter at the prompt, a prompt option to say whether or not to show the `boot:` text prompt, and optional timeout specifications.

You can use `display` ***filename*** to dump the text of a descriptive menu to the screen. Note that this file is relative to the TFTP root; it's not in the `pxelinux.cfg` subdirectory.

This file can also contain escape characters, although this does require you to know how to insert such characters in your chosen text editor (for example, on Windows, you could use Microsoft Visual Studio in binary mode to generate initial characters in the file; copy and paste then lets you edit the remainder of the file in text mode). Editors on other platforms have other means of accomplishing the same goal.

The interesting escape characters are "Ctrl-L", which is decimal 12, hex 0C - this clears the screen. It is often good to put this at the start of the menu text, to clear out the previous boot messages.

To change the text color, you need a sequence of three bytes:
A color escape byte, followed by a background color byte, followed by a foreground color byte.
The color escape byte is "Ctrl-O", which is decimal 15, hex 0F.

The background and foreground color bytes are actually derived from the ASCII text character corresponding to their color, so it's easier to treat these directly as ASCII characters following the binary escape code, rather than looking at the byte value directly. For example, black is the character zero (`0`) which would have ASCII value hex 30, decimal 48.

If you wish to create a `menu.txt` file, copy it into `/tftpboot` and uncomment the `display` line in the following config file.

Then, copy a config file similar to this to `/tftpboot/pxelinux.cfg/default` :

```
# pxelinux Menu

default localboot

# display menu.txt
prompt 1

# Timeout is disabled right now;
# since we only PXE boot at install time so far, we want admins
# to see this menu and make a specific install choice.
# timeout 200
# totaltimeout 6000

label centos
  kernel centos/vmlinuz
  append initrd=centos/initrd.img ksdevice=eth0 load_ramdisk=1 network

label centos-min
  kernel centos/vmlinuz
  append initrd=centos/initrd.img load_ramdisk=1 network \
         ksdevice=eth0 ks=http://core/bootserv/kickstart/minimum.cfg

label localboot
  localboot 0
```

Note: The `append` line under `centos-min` is still a single line, but has been wrapped using a `\` in the above text for clarity.

The `centos-min` stanza shows how we will later provide a kickstart file via HTTP, to allow automated, one-click OS installs.

The `centos` stanza will cause the OS to load the kernel and then enter attended install mode, requesting the network location of the install repository (to which we will be able to provide an HTTP server address once we have configured the web server).

Provide kernel images

For each boot option, now we have to supply the actual kernel images to boot into.
The filenames in the PXE menu config are relative to the TFTP root dir, so for example with

```
   kernel centos/vmlinuz
```

you would copy `vmlinuz` into `/tftpboot/centos` .

For CentOS, these can be found in the directory `images/pxeboot` in the install image (on the first ISO CD image of the set, if you are using CD rather than DVD images).

In our environment, we will copy these files from `/data/bootserv/centos` where we installed them:

```
mkdir /tftpboot/centos

cp /data/bootserv/centos/images/pxeboot/vmlinuz  \
   /tftpboot/centos/vmlinuz

cp /data/bootserv/centos/images/pxeboot/initrd.img \
   /tftpboot/centos/initrd.img
```

At this point, it should be possible to PXE boot the client into the menu, select the `centos` install option and successfully boot the OS image specified. For this CentOS installer this gets you to the point where you need to specify an installation method, and we want to use HTTP as the simplest, most direct network option.

Set up HTTP server

For CentOS installs, what we need is simply an HTTP location which contains the unpacked contents of the ISO images (if using CD rather than DVD images, unpack these all in one directory, ie "on top of" each other).

We already unpacked the ISO into `/data/bootserv/centos` at the beginning of this document, so all that remains is to install Apache, and configure an alias in the Apache config pointing to this location.

Install Apache:

```
yum -y install httpd
```

We want to see the images under paths like http://core/bootserv/centos/ , so we add the following to the Apache config:

```
<Directory /data/bootserv>
  Options Indexes FollowSymLinks
  Order allow,deny
  AllowOverride None
  Allow from all
</Directory>

Alias /bootserv /data/bootserv
```

Restart Apache: `service httpd restart`

We must also ensure that TCP port 80 is reachable in iptables, with a line like:

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp
                        --dport 80 -j ACCEPT
```

or equivalent, although you may have already configured this during the initial install. (The above line is wrapped onto two lines, but is a single line in `/etc/sysconfig/iptables`).

Finally, we assume here that SELinux is disabled; otherwise, it will be necessary to change SELinux permissions for the /data/bootserv file tree also.

Once Apache is restarted, it should then be possible to specify in the CentOS network boot installer a host ("Web site name") of
`core.test.example.com` (or `10.1.4.1`)
and a path ("CentOS directory") of
`/bootserv/centos`
to have it find the packages and complete the install.

Once you have installed a target host, note that a kickstart config file, suitable for use in the `centos-min` install option described above, can be found in `/root/anaconda-ks.cfg` , and this can be edited, with reference to the kickstart configuration guide at:

http://www.centos.org/docs/5/html/Installation_Guide-en-US/s1-kickstart2-file.html

Here is a simple kickstart file, generated by selecting just the Server and Cluster package groups during the original install. The `clearpart`, `part`, `volgroup` and `logvol` directives have been uncommented from the autogenerated file, so that the machine's local disk will be automatically repartitioned any time the machine is reinstalled.

Based on the PXE menu config created above, the contents of this file could be placed in `/data/bootserv/centos-min.cfg` to provide an auto install template named `centos-min`.

```
# Kickstart file automatically generated by anaconda.

install
url --url http://core/bootserv/centos
lang en_US.UTF-8
keyboard uk
network --device eth0 --bootproto dhcp
rootpw --iscrypted $1$9WOk6Ag5$oQ441G0PFxPq.KwkzXvQc/
firewall --enabled --port=22:tcp
authconfig --enableshadow --enablemd5
selinux --enforcing
timezone --utc Europe/London
bootloader --location=mbr --driveorder=sda

clearpart --all --drives=sda
part /boot --fstype ext3 --size=100 --ondisk=sda
part pv.2 --size=0 --grow --ondisk=sda
volgroup VolGroup00 --pesize=32768 pv.2
logvol swap --fstype swap --name=LogVol01 --vgname=VolGroup00 --
size=512 --grow --maxsize=1024
logvol / --fstype ext3 --name=LogVol00 --vgname=VolGroup00 --
size=1024 --grow

%packages
@base
@clustering
@core
@dns-server
@dialup
@editors
@ftp-server
@legacy-network-server
@mail-server
@network-server
@news-server
@server-cfg
@text-internet
@web-server
@smb-server
keyutils
trousers
fipscheck
device-mapper-multipath Sample boot flow
```

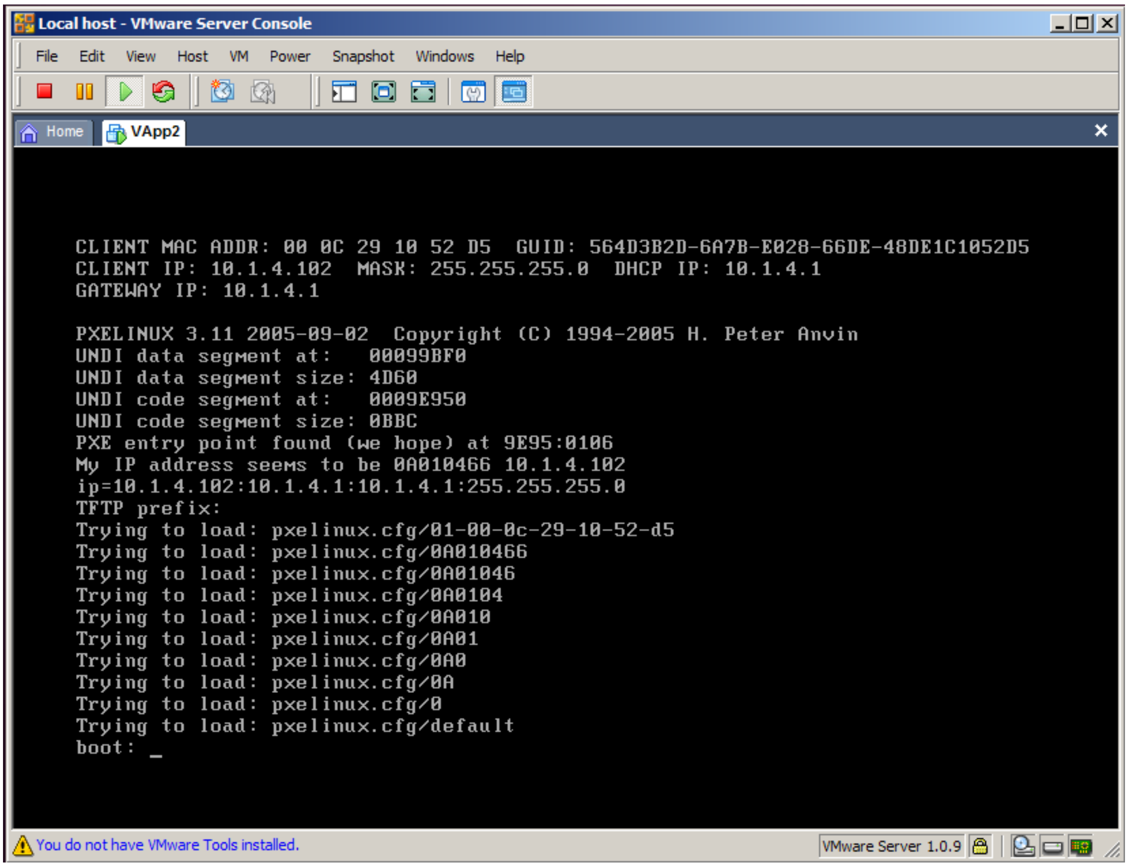
Sample boot flow

Once this infrastructure is in place, you can edit the BIOS options of your target host so that it network boots (often, this simply means hitting Escape during BIOS startup to change the boot order).

The PXE stack on the network card will then try and obtain an IP address over DHCP (in this example the server has recognised the machine's MAC address and allocated it 10.1.4.102).

The DHCP options next-server and filename then cause the machine to load `pxelinux.0` as a bootstrap file, which then tries to load a config file from inside the `pxelinux.cfg` directory on the server, as described previously in this document. The screenshot below shows this point, where the PXE prompt is now waiting for the administrator to type in a suitable boot label (such as `centos-min`) to boot a selected kernel.

PXE booting a VMWare Server instance



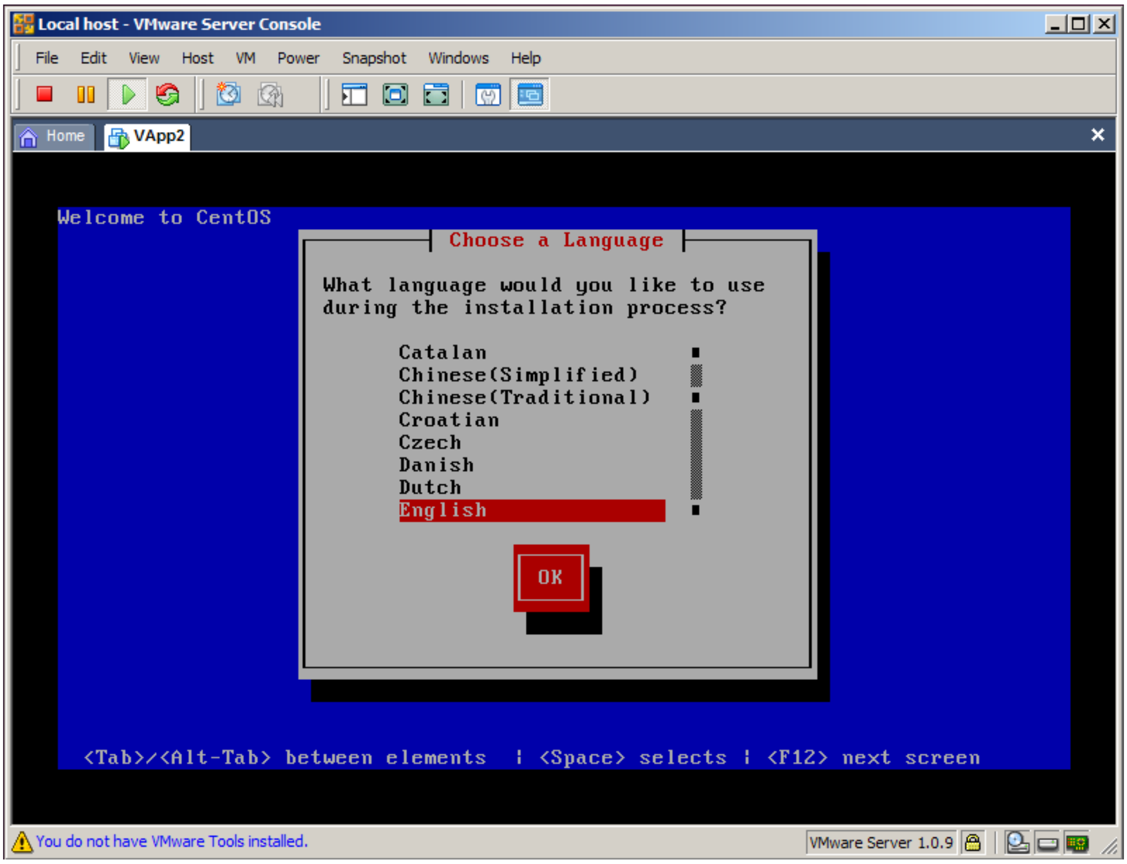
```
CLIENT MAC ADDR: 00 0C 29 10 52 D5  GUID: 564D3B2D-6A7B-E028-66DE-48DE1C1052D5
CLIENT IP: 10.1.4.102  MASK: 255.255.255.0  DHCP IP: 10.1.4.1
GATEWAY IP: 10.1.4.1

PXELINUX 3.11 2005-09-02  Copyright (C) 1994-2005 H. Peter Anvin
UNDI data segment at:   00099BF0
UNDI data segment size: 4D60
UNDI code segment at:   0009E950
UNDI code segment size: 0BBC
PXE entry point found (we hope) at 9E95:0106
My IP address seems to be 0A010466 10.1.4.102
ip=10.1.4.102:10.1.4.1:10.1.4.1:255.255.255.0
TFTP prefix:
Trying to load: pxelinux.cfg/01-00-0c-29-10-52-d5
Trying to load: pxelinux.cfg/0A010466
Trying to load: pxelinux.cfg/0A01046
Trying to load: pxelinux.cfg/0A0104
Trying to load: pxelinux.cfg/0A010
Trying to load: pxelinux.cfg/0A01
Trying to load: pxelinux.cfg/0A0
Trying to load: pxelinux.cfg/0A
Trying to load: pxelinux.cfg/0
Trying to load: pxelinux.cfg/default
boot: _
```

⚠ You do not have VMware Tools installed.

If you boot into the default `centos` kernel, you will automatically be dropped into the menu-driven manual network install, and you can specify HTTP as the install method, with `core.test.example.com` as the server name and `/bootserv/centos` as the CentOS directory.

The CentOS install menu via a network boot


Local host - VMware Server Console
File
Edit
View
Host
VM
Power
Snapshot
Windows
Help
Home
VApp2
You do not have VMware Tools installed.
VMware Server 1.0.9

```
Welcome to CentOS

Choose a Language

What language would you like to use
during the installation process?

Catalan
Chinese(Simplified)
Chinese(Traditional)
Croatian
Czech
Danish
Dutch
English

OK

<Tab>/<Alt-Tab> between elements  | <Space> selects | <F12> next screen
```

Alternatively, once you have a kickstart menu in place such as `centos-min`, you can enter this at a boot prompt and the install will take place with no further prompting.

References

A search on `PXELINUX SYSLINUX` in Google should include the following page in the results:
`http://syslinux.zytor.com/pxe.php`

This is the PXELINUX instruction page by the author, H Peter Anvin.
If you are using a different Linux distribution that does not include syslinux by default, you can download it from:
`http://www.kernel.org/pub/linux/utils/boot/syslinux/`
and it includes the pxelinux.0 boot images you need to start.

The CentOS kickstart guide may be found at:
`http://www.centos.org/docs/5/html/Installation_Guide-en-US/s1-kickstart2-file.html`