What is PXE Booting?

PXE, stands for **P**reboot e**X**ecution **E**nvironment, will help to install an operating system over the network without the need for physical media. This means no dealing with pesky bootable USB drives and ridiculous boot menu keys. This is useful for System Administrators because you are able to install a variety of operating systems by simply powering on the hardware and ensuring the NIC has an active connection. PXE environments need a **DHCP server** that distributes the IP addresses to the client systems, and a **TFTP server** that downloads the installation files to the PXE clients. Tonight we'll be creating a PXE boot server on a CentOS 7 box. We'll walk you through the steps below.

Virtual Network Editor

You'll need to make some changes to the Virtual Network on your host machine, namely you'll need to disable DHCP on your NAT'd Network and make note of the network information. Try and figure this one out on your own (a.k.a. Consult the googs). If you get stuck check out Appendix B for a walkthrough.

Download your ISO

Download which flavor of linux you'd like to PXE boot from mirrors.rit.edu, make sure to grab the full image .iso file and not the Live boot image.

Disabling SELinux and Firewalls

Stop the firewalld service from starting at boot:

```
systemctl disable firewalld
```

Stop the firewalld service:

```
systemctl stop firewalld
```

To disable SELinux you will need to edit the following file:

```
/etc/sysconfig/selinux
```

Change the **SELINUX** variable to **disabled**, save the file and reboot your machine.

Installing required packages

```
yum install dhcp httpd xinetd syslinux tftp-server -y
```

Statically assign your IP address

`/etc/sysconfig/network-scripts/ifcfg-ens[#]`

Configuring your PXE Server

First off, we'll need to copy some core files from the syslinux directory that we just installed (these are the bootloader files)

```
cd /usr/share/syslinux/
cp pxelinux.0 menu.c32 memdisk mboot.c32 chain.c32 /var/lib/tftpboot/
```

Now we need to enable our Trivial File Transfer Protocol (TFTP) server. With the text editor of your choice (vim master race) edit:

`/etc/xinetd.d/tftp`

Change the **`disable=yes`** parameter to **no**

Next we'll need to put the required files for the centos7 install somewhere so let's create a directory in `/var/lib/tftpboot/` called **centos7_x64**

```
mkdir /var/lib/tftpboot/centos7_x64
```

Switch to the directory where your .iso file is downloaded and execute the following command *you'll need to change the path after the loop portion to match your iso file location***

```
mount -o loop /root/CentOS-7.0-1406-x86_64-Everything.iso /mnt/
```

Once mounted, copy the contents of the directory to your tftpboot directory:

```
cp -fr /mnt/. /var/lib/tftpboot/centos7_x64/
```

Ensure the permission are correct for that directory:

```
chmod -R 755 /var/lib/tftpboot/centos7_x64/
```

Next we need to create an Apache configuration file so that we can point our PXE boot script to the correct directory under /etc/httpd/conf.d/ create the following file:

```
vim /etc/httpd/conf.d/pxeboot.conf
```

Enter the following configuration lines:

```
Alias /centos7_x64 /var/lib/tftpboot/centos7_x64/

<Directory /var/lib/tftpboot/centos7_x64>
Options Indexes FollowSymLinks
Order Deny,Allow
Require all granted
</Directory>
```

Now we need to create a configuration directory for our PXE server. This is what makes it so that we have a nice menu when we PXE boot. Create the following directory:

```
mkdir /var/lib/tftpboot/pxelinux.cfg
```

Now create the following file:

```
vim /var/lib/tftpboot/pxelinux.cfg/default
```

Enter the following configuration lines:

```
default menu.c32
prompt 0
timeout 300
ONTIMEOUT local

menu title ########## NextHop PXE Boot Menu ##########

label 1
menu label ^1) Install CentOS 7
kernel centos7_x64/images/pxeboot/vmlinuz
append initrd=centos7_x64/images/pxeboot/initrd.img
method=http://192.168.111.X/centos7_x64 devfs=nomount

label 2
menu label ^2) Boot from local drive localboot
```

The HTTP server you specify in this config file should be the static IP you assigned yourself*

Configuring your DHCP Server

Edit the dhcp configuration file:
vim /etc/dhcp/dhcpd.conf

Enter the following configuration lines:

```
#standard config
ddns-update-style interim;
ignore client-updates;
authoritative;
allow booting;
allow bootp;
allow unknown-clients;

#subnet declaration
subnet 192.168.111.0 netmask 255.255.255.0 {
      range 192.168.1.50 192.168.1.75;
      option routers 192.168.1.2;
      default-lease-time 600;
      max-lease-time 7200;

      #PXE Server IP Configuration
      next-server 192.168.111.X;
      filename “pxelinux.0”;
```

The NEXT-SERVER variable you define should be the static IP you assigned yourself*

Alright, time to restart those services!

```
systemctl restart xinetd
systemctl restart httpd
systemctl restart dhcpd
```

If this were real life we’d make sure they start at boot as well:

```
systemctl enable xinetd
systemctl enable httpd
systemctl enable dhcpd
```

Configuring a virtual machine for PXE boot

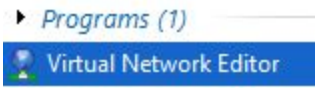
Now time to test! Create a new virtual machine and attempt to PXE boot it. This is relatively the same process as creating a normal VM, except we don't select an installation media. See if you can figure it out on your own. If you get stuck check out Appendix B for a walkthrough.

Advanced section Kickstart config

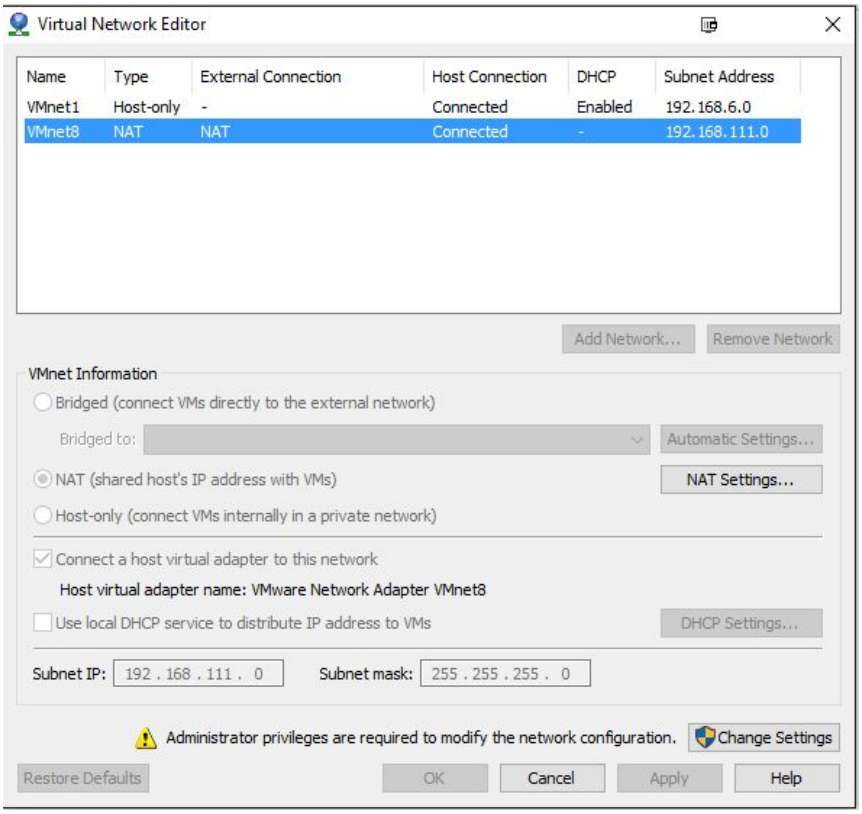
Google how to create a kickstart config. Create one to do some magic for you. Add that line to your PXE menu config file via ks=/path/kickstart.cfg

Appendix A: How to setup Virtual Network Editor

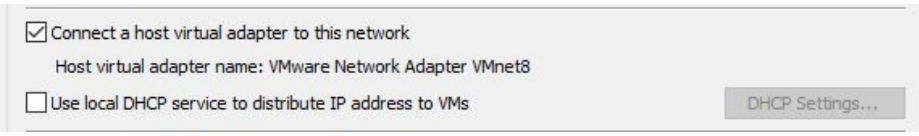
1. **Hit the Windows Key -> and search for Virtual Network Editor**



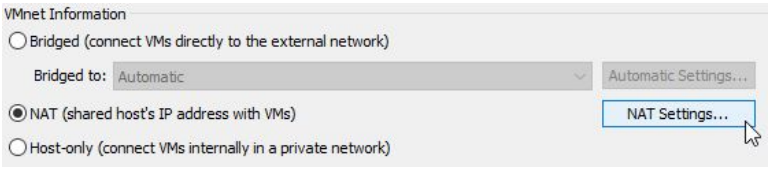
2. **In the menu select your NAT network and click Change Settings**



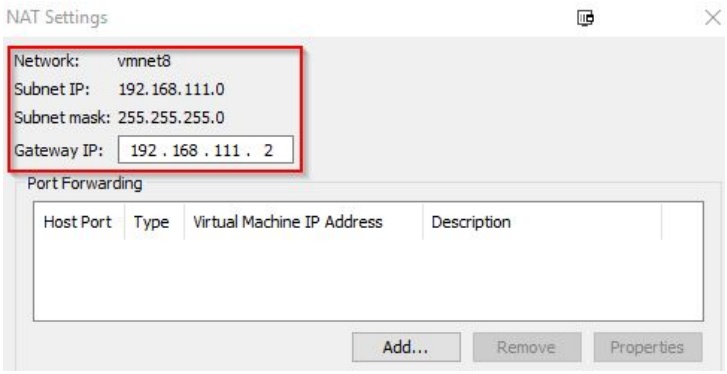
3. **Turn off the local DHCP server for VMs (Since we'll be hosting our own)**



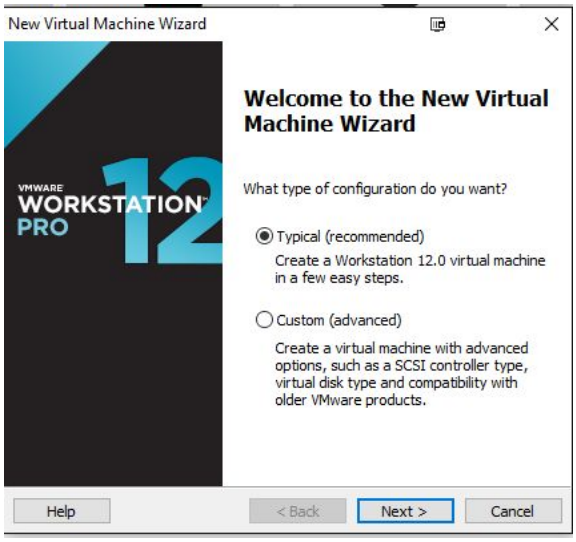
4. **Click on NAT Settings**

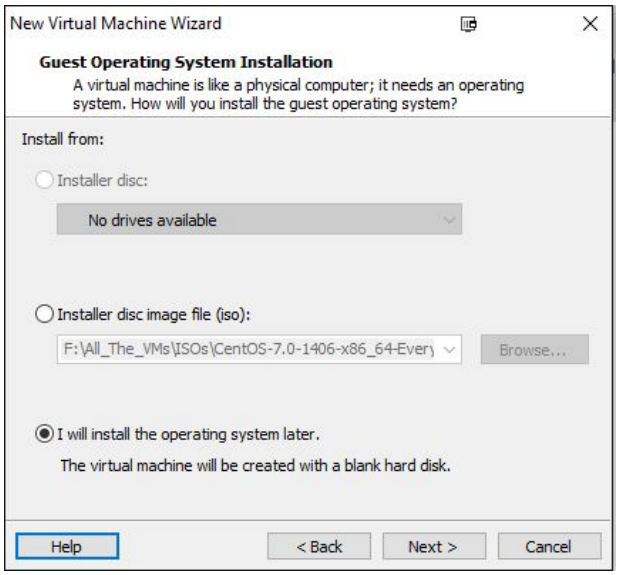


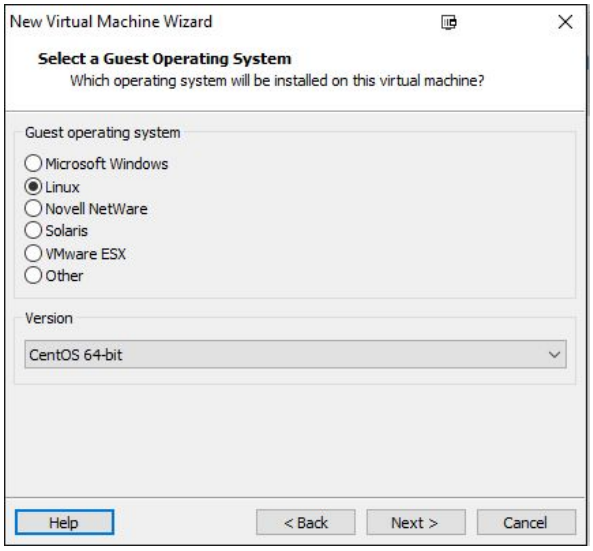
5. **Make note of the Network Information and Gateway address for you host and DHCP files**

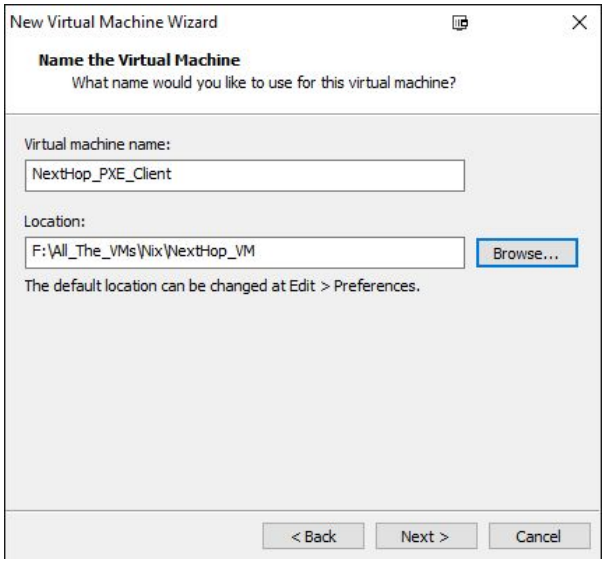


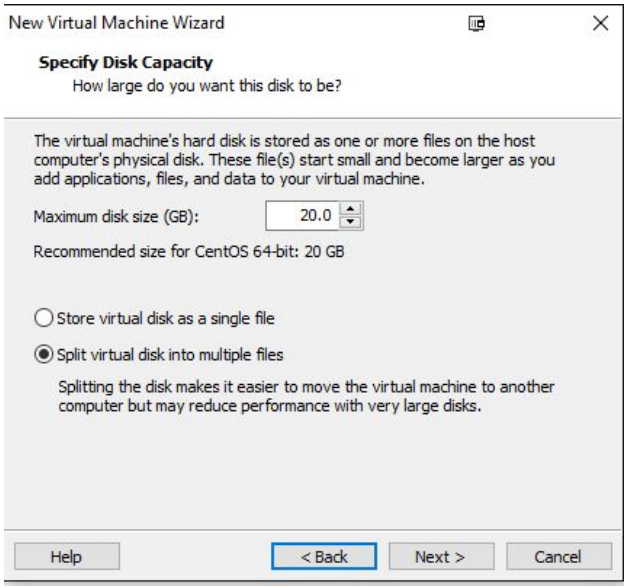
Appendix B: How to setup a Virtual Machine for PXE boot

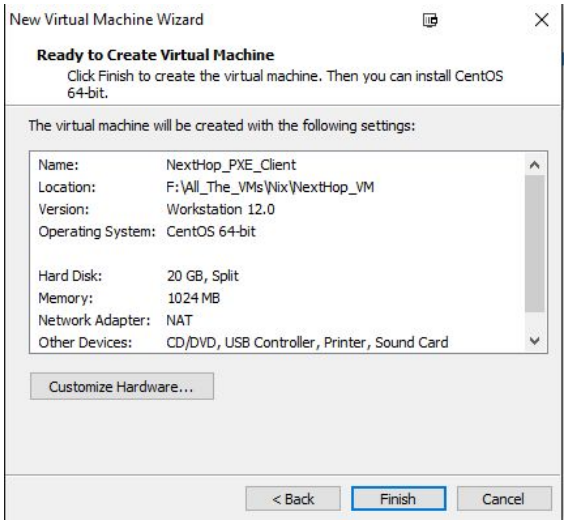





New Virtual Machine Wizard
Select a Guest Operating System
Which operating system will be installed on this virtual machine?
Guest operating system
Microsoft Windows
Linux
Novell NetWare
Solaris
VMware ESX
Other
Version
CentOS 64-bit
Help
< Back
Next >
Cancel


New Virtual Machine Wizard
Name the Virtual Machine
What name would you like to use for this virtual machine?
Virtual machine name:
NextHop_PXE_Client
Location:
F:\All_The_VMs\Nix\NextHop_VM
Browse...
The default location can be changed at Edit > Preferences.
< Back
Next >
Cancel


New Virtual Machine Wizard
Specify Disk Capacity
How large do you want this disk to be?
The virtual machine's hard disk is stored as one or more files on the host computer's physical disk. These file(s) start small and become larger as you add applications, files, and data to your virtual machine.
Maximum disk size (GB):
20.0
Recommended size for CentOS 64-bit: 20 GB
Store virtual disk as a single file
Split virtual disk into multiple files
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.
Help
< Back
Next >
Cancel


New Virtual Machine Wizard
Ready to Create Virtual Machine
Click Finish to create the virtual machine. Then you can install CentOS 64-bit.
The virtual machine will be created with the following settings:
Name: NextHop_PXE_Client
Location: F:\All_The_VMs\Nix\NextHop_VM
Version: Workstation 12.0
Operating System: CentOS 64-bit
Hard Disk: 20 GB, Split
Memory: 1024 MB
Network Adapter: NAT
Other Devices: CD/DVD, USB Controller, Printer, Sound Card
Customize Hardware...
< Back
Finish
Cancel