

# iPXE boot server

iPXE is an open-source Preboot Execution Environment implementation that has a lot more functionality than most firmware PXE clients. Apart from using tftp to download data, iPXE can also use HTTP. Other boot methods include booting from an iSCSI SAN, a fibre channel SAN using FCoE, an ATA over Ethernet (AoE) SAN, or a wireless network. It also has a scripting language so you can create complex boot scripts. iPXE is also used as the PXE client in QEMU, so with a iPXE server you can boot virtual machines directly from a webserver.

In this article, we will configure a web server as a primary boot server. Because not all PXE clients can use HTTP, we will setup a simple tftp server to serve (or chain-load) the iPXE client itself.

## Assumptions and prerequisites

- We will use *dnsmasq* as DHCP and tftp server. We only need a minimal tftp setup and dnsmasq has a built-in server.
- tftp root folder is `/srv/tftproot`.
- Web server is `www.mynetwork.com`. Web server root is `/var/www/htdocs`.
- We need a kernel and initrd file. Copy these from either a Slackware cd or mirror.

## Configure DHCP and tftp server

Add following section to `/etc/dnsmasq.conf`:

[dnsmasq.conf](#)

```
# PXE booting: setup tftp server
enable-tftp
tftp-root=/srv/tftproot

# set tag "ipxe" if request comes from iPXE ("iPXE" user class)
dhcp-userclass=set:ipxe,iPXE

# if PXE request came from regular firmware, serve iPXE firmware
dhcp-boot=tag:!ipxe,undionly.kpxe
# if PXE request comes from iPXE, direct it to boot from HTTP
dhcp-boot=tag:ipxe,http://www.mynetwork.com/pxe/boot.txt
```

The first lines will activate the built-in tftp server and define the root folder. Then a tag is set to determine if the boot request is coming from an iPXE client. If the request is not coming from iPXE, the iPXE client itself is downloaded and executed. If the request came from iPXE, a boot script is downloaded from the webserver and executed.

The above section deals specifically with PXE booting. You will also need to setup basic DHCP functionality; see the [DHCP server via dnsmasq](#) article for that. To activate dnsmasq, make sure script

/etc/rc.d/rc.dnsmasq is executable.

Create a new directory /srv/tftproot. Download the [undionly.kpxe](#) file from the iPXE website and copy to the /srv/tftproot directory. This is the iPXE client file which will be downloaded from the tftp server.

## Webserver setup

Create a new folder /var/www/htdocs/pxe. From the Slackware CD or mirror, copy the following files: isolinux/initrd.img and kernels/huge.s/bzImage into the pxe folder. Now create a new text file boot.txt with the following content:

[boot.txt](#)

```
#!ipxe

initrd http://www.mynetwork.com/pxe/initrd.img
chain http://www.mynetwork.com/pxe/bzImage load_ramdisk=1
prompt_ramdisk=0 rw printk.time=0 nomodeset SLACK_KERNEL=huge.s
```

This is all that is needed to create a PXE boot server. This is a very simple example that just boots the default huge installation kernel. To cater for more dynamic situations, you will have to create more complex boot scripts. Look at the examples page on the iPXE website.

## Complex scripts

A more complex script with a boot menu is shown below:

[boot.txt](#)

```
#!ipxe

set mirror http://www.mynetwork.com
set dir mirror

menu Slackware installation menu
item slackware64-14.1    Slackware64 14.1
item slackware64-14.0    Slackware64 14.0
item slackware64-current Slackware64 current
item slackware-14.1      Slackware 14.1
item slackware-14.0      Slackware 14.0
item slackware-current   Slackware current
choose slackversion || exit 0

# boot arguments
set args load_ramdisk=1 prompt_ramdisk=0 rw printk.time=0 nomodeset
```

```
SLACK_KERNEL=huge.s

# The APU system from PC Engines has the console on the serial port
set pcengines PC Engines
iseq ${manufacturer} ${pcengines} && iseq ${product} APU && set
extraargs console=ttyS0,115200n8 ||

# load initrd and boot kernel
initrd ${mirror}/${dir}/${slackversion}/isolinux/initrd.img
chain ${mirror}/${dir}/${slackversion}/kernels/huge.s/bzImage ${args}
${extraargs}
```

Note that you can also use a public Slackware mirror instead of hosting the files yourself. If you point to a slackware current mirror, you can boot the latest installation kernel directly from the internet !

## Sources

- The iPXE website: [ipxe.org](https://ipxe.org) with many [examples](#).
- Originally written by [Frank Donkers](#)

[howtos](#), [boot](#), [pxe](#), [ipxe](#), [author fdonkers](#)

From:  
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:  
[https://docs.slackware.com/howtos:network\\_services:ipxe](https://docs.slackware.com/howtos:network_services:ipxe)

Last update: **2014/11/14 09:43 (UTC)**

