Login with SSO **FAQ** 

Thread Tools

Please read before SSO login **Forum Activity Page** Advanced Search **Ubuntu Community Quick Links** Forum Community Other Support Social Media **Ubuntu Official Flavours Support** Forum The Ubuntu Forum Community Networking & Wireless HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes Page 1 of 3 Results 1 to 10 of 28 Last >>>

Thread: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

One person likes this. Be the first of your friends.

September 11th, 2013

Emblem Parade 🚇 A Carafe of Ubuntu

Join Date: Mar 2007

#1

Display

Beans: 108

# HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

## Note: This guide also applies to Ubuntu 14.04

There's a lot of information about LTSP (the Linux Terminal Server Project) out there, including quite a few quides, but nothing that really brought everything together well enough for me. In particular, they were bad at explaining the reasons for each step. So, here's my version!

Note that you do not need a dedicated Ubuntu server for this guide. I ran a terminal server just fine using Ubuntu 14.04 or 12.04 running inside VirtualBox. If you do so, just make sure the virtual machine is running with the network adapter in bridged mode, so that it is a full participant in the LAN.

### **HOW DOES LTSP WORK?**

When it does work it seems like magic, but it's worth understanding what LTSP does, because it's actually a rather simple (but beautiful!) combination of existing technologies and products.

The thin client runs applications on the server using X forwarding, in fact not much differently from Unix terminals in pre-PC days. You can easily try this technology without LTSP by using "ssh -X", as long as you have X running on the client. What LTSP adds to this is the initial handshaking, the creation of the client/server session. You thus require specialized thin/fat clients that include support for logging into the LTSP server. After this handshake, LTSP actually plays no further role. So, all a client really needs is this LTSP session creator, and then X. (I'm simplifying this a bit: LTSP adds support for forwarding USB devices and sharing network drives, too, but that stuff is all done within the X forwarding framework.)

So how do you get that specialized LTSP session creator and X on the client? One way is to install it manually, assuming the client has some storage. That's how BerryTerminal works: it's a tiny operating system that includes only that bare minimum. When it boots up, it will look for the LTSP server and display the login screen (or login automatically if you have that configured on the server).

### **NETWORK BOOT**

But obviously, many clients do not have storage. They can, however, automatically download their operating system over the network, using an technology called PXE. PXE has been included in practically every network card controller since the past decade. PXE, in turn, relies on two services: a DHCP server provides the client with its initial IP address, DNS, and PXE boot information. The client then uses TFTP to download the operating system image and boot it. So, in fact you will need three things to make this work: 1) a DHCP server; 2) a TFTP

server; 3) a client operating system image. The whole dance of getting an IP address, downloading the image and booting it doesn't take that long: within 30 seconds you should get a login screen, even for weak clients.

To complicate things, there is an important limitation: only a single DHCP server can exist on a LAN, and it's very likely that you already have one running on your router. If that's the case, you have two options: 1) you *might* be able to configure your router's DHCP server to send clients PXE information about your terminal server, an advanced option that unfortunately does not exist in cheap office routers; 2) you can use a PXE feature called "proxy DHCP", created exactly for this situation. It's very easy to set up, and indeed **that's what this guide is for**. (Note that some *very* old equipment does not have proxy DHCP support in its PXE, in which case you can't use this method. But that's very rare these days.)

(Actually, there is a third, more complicated option: you can create a special closed LAN just for your terminal server and clients. In this LAN, your terminal server will act as both the DHCP server and TFTP server. Remember: you don't need a router in this LAN, because clients--assuming they are thin and not fat--run all their software on the server. But this means that your terminal server likely has two network cards, one sitting on this client LAN, and the other sitting on your broader network, likely with access to the Internet. This is what the "ltsp-server-standalone" package is for, and where most guides for LTSP begin. Here's an excellent one if that's your situation. So, to reiterate, this guide assumes that you already have a working DHCP server on your network.)

### **LET'S DO IT!**

Let's install LTSP, a proxy DHCP server, and a TFTP server:

### Code:

sudo apt-get install ltsp-server dnsmasq tftpd-hpa

(Though Dnsmasq can do both DHCP and TFTP, I've found it has a crucial bug when functioning as a TFTP server specifically with Ubuntu 14.04, so we'll be using tftpd-hpa instead for that.)

**Warning:** do *not* install "ltsp-server-standalone" package as is suggested by other guides, because that assumes that you want your terminal server to be a *full* DHCP server.

Now we need a client operating system image. As stated earlier, for thin clients all we need the is the LTSP session creator and X. Of course, we also need a kernel (Linux), basic drivers for keyboard, mouse, video card, audio, etc. It's still an operating system! Marvelously, LTSP on Ubuntu comes with a tool to automatically create such an image for you, based, of course, on Ubuntu. But it should be emphasized that this client operating system is nothing like the regular full-blown Ubuntu desktop distribution: it is stripped down and highly specialized to work purely as an LTSP client. Indeed, for thin clients its total size is less than 300 mb.

LTSP on Ubuntu gives you many options for customizing the client image, but the defaults should work just fine. (Customization would be necessary in case the default image doesn't support your thin clients' hardware for some reason. In that case, you might need to install additional drivers on your image.)

This is going to take a while, so sit back and relax:

Ubuntu 14.04:

### Code:

sudo ltsp-build-client

## Ubuntu 12.04:

(We'll assume in this guide that our thin clients are 32-bit x86 machines, which is most common. You can also build 64-bit x86 images, assuming you have clients that could support it, though there would be no serious advantage to it unless they are fat clients running heavy software. More commonly useful, perhaps, are images for ARM architecture clients. Simply replace the "i386" in this guide with the client architecture you are using, as long as it's supported by Ubuntu.)

```
sudo ltsp-build-client --arch i386
```

The command will download all packages for this stripped-down Ubuntu-based client operating system and install them under: /opt/ltsp/i386/. It will then compress this into /opt/ltsp/images/i386.img, the actual image to be sent to the client.

(Note that this image will be based on your *current* Ubuntu version, at the *current* state of packages in its repositories. Since this is merely a thin client operating system, it shouldn't really matter if the image is stays "old", as long as it works. Remember, the software is actually running on your server, not the client. However, it is possible to upgrade this image to latest Ubuntu packages, something you might want to do once in a while, especially if security is important. This guide won't cover that.)

Actually, we need to stop and explain one more aspect of the network booting process for LTSP on Ubuntu: it happens in two stages. In order to download, decompress and boot into the i386.img, we need a network operating system bootloader. This is provided by PXELINUX, a tiny Linux-based bootloader (really, a mini operating system) designed exactly for this use case. If you're used to desktop Ubuntu, consider that PXELINUX is similar to GRUB, but designed around network boot. The "ltsp-build-client" created it for us under /var/lib/tftpboot/ltsp/i386/. It is less than 15 mb in size. So, the client gets this pxelinux directly over TFTP and runs it, and PXELINUX then downloads i386.img, decompresses it into a RAM drive, and boots.

The configuration file for our PXELINUX is under /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default. Unfortunately, in the version of LTSP included in Ubuntu 14.04 and 12.04, it is generated without support for proxy DHCP by default. To fix this, you must run:

Ubuntu 14.04:

### Code:

```
sudo sed -i 's/ipappend 2/ipappend 3/g' /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default
```

Ubuntu 12.04:

# Code:

```
(cat <<EOF
ipappend 3
EOF
) | sudo tee -a /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default</pre>
```

If you update your client image in the future (with "ltsp-update-image") then you will need to run this command again.

(Documentation for the "ipappend" can be found here. The value "3" is the logical OR of "1" and "2".)

OK! Now it's time to configure Dnsmasq. Edit /etc/dnsmasq.d/ltsp.conf (the file isn't there, you must create it) and put something like this:

```
#
# Dnsmasq running as a proxy DHCP and TFTP server
#
# See: http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html
#
# TFTP
#
# This might work instead of tftpd-hpa:
```

```
#enable-tftp
#tftp-root=/var/lib/tftpboot

#
# DHCP
#

DHCP proxy on this network
dhcp-range=192.168.1.0,proxy

# Tell PXE clients not to use multicast discovery
# See section 3.2.3.1 in http://tools.ietf.org/html/draft-henry-remote-boot-protocol-0
dhcp-option=vendor:PXEClient,6,2b

# Better support for old or broken DHCP clients
```

You will need to change "dhcp-range" to the subnetwork of your LAN. Also change "i386" and "x86PC" to something else if you are using a different client architecture image. Now, restart dnsmasq:

### Code:

```
sudo service dnsmasq restart
```

Note that on Ubuntu 14.04, there is currently a bug with tftpd-hpa, with this solution.

And ... you should be good to go. Start your clients with network boot: they should get an IP address from your router's DHCP server, then get PXE information from your proxy DHCP server (Dnsmasq), then boot PXELINUX from your TFTP server (also Dnsmasq), then download and boot into the i386.img stipped-down-Ubuntu-based client operating system, and then display the login screen.

### **EXTRAS**

By default, LTSP on Ubuntu uses an ssh tunnel for secure X forwarding, but you can switch to "direct X" mode for better scalability and performance, at the expense of reduced security on the LAN. Edit /var/lib/tftpboot/ltsp/i386/lts.conf (the file will not exist, but it's OK to create it) and put this:

### Code:

```
[Default]
LDM_DIRECTX = True
```

Now we need to regenerate the image and not forget to fix the proxy DHCP issue:

Ubuntu 14.04:

# Code:

```
sudo ltsp-update-image
sudo sed -i 's/ipappend 2/ipappend 3/g' /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default
```

Ubuntu 12.04:

```
sudo ltsp-update-image --arch=i386
(cat <<EOF
ipappend 3
EOF
) | sudo tee -a /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default</pre>
```

The next time clients boot, they will be using this new image.

### **NEED GUEST USERS FOR A PUBLIC LAB?**

I've created a separate guide for doing that.

Last edited by Emblem Parade; June 19th, 2014 at 01:04 PM. Reason: Ubuntu 14.04

**Adv Reply** 

November 29th, 2013

robin.peters First Cup of Ubuntu

Join Date: Nov 2013

Beans:

#2

# Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

Hi - nice HowTo, but can you make one for the configuration with a i386 terminal server and Raspberry Pi (armhf architecture)? I use Ubuntu Server and U-Boot (RasPi), but I have no idea what I'm doing wrong.

Thanks - Robin

Adv Reply

November 29th, 2013



Join Date: Mar 2007

#3

Beans: 108

## Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

The setup is identical for an i386 terminal server: there should be no difference on the server side.

However, Raspberry Pi introduces a special limitation: you cannot create an Ubuntu-based client image for it (for the same reason Ubuntu won't run on a Raspberry).

The solution is to install BerryTerminal on your Raspberries. It's a very minimal image that could work on small, cheap SD cards. With it, the Raspberry would boot straight into the LTSP logon. Simple!

**Adv Reply** 

March 2nd, 2014



Join Date: Dec 2005

> Beans: 10

#4

## Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

Hi,

Thanks for the great how-to.

I've just now tried to work my way through it, but unfortunately, I didn't quite end up where I thought I would.

In short, upon starting the machine that I want to use as a thin client, I got:

### Code:

PXE-E53: No boot filename received

PXE-MOF: Exiting Intel PXE ROM.

## My situation is this:

I have two computers, one running Linux the other Windows 7. Both are connected to a home-network router through which they share an internet connection. Both computers are plugged into the router with an ethernet cable (and both also have a wireless connection to the router, though I can't remember why - but I digress). Suffice to say: the router is acting as the LAN's DHCP server.

For the sake of science (proof-of-concept), I want to turn the Linux machine into an LTSP terminal server and the Windows 7 machine into a thin client.

Following your instructions, I've done this:

- On the Linux machine, sudo apt-get install ltsp-server dnsmasq
- 2. sudo ltsp-build-client --arch i386
- 3. (cat <<EOF ipappend 3 EOF

```
) | sudo tee -a /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default
```

This step I don't fully understand except that it is necessary because in "the version of LTSP included in Ubuntu 12.04, [the configuration file for our PXELINUX] is generated without support for proxy DHCP."

How, though, does just the appending of ipappend 3 to end of the file linked to by the soft link at /var/lib/tftpboot/ltsp/i386/pxelinux.cfg/default ensure I'll be provided with a functioning "network operating system bootloader", ie, PXELINUX? Given that, using "a PXE feature called 'proxy DHCP', [is] **what this guide is for**", could you please expand on this seemingly vital step?

For the moment, I will have faith.

Next step.

4. "Edit /etc/dnsmasq.d/ltsp.conf"

You wrote "edit". Is that to say that this file should have been created by the apt-get install or ltsp-build-client commands and we need just to change it? For me, at this point, there was no such file at that location.

So I created a file named ltsp.conf in the /etc/dnsmasq.d directory and put into it what you have in your post.

However, for the dhop-range line, you suggest that we change it to the "subnetwork of your LAN". I'm not entirely clear on how to figure out what the subnetwork of my LAN is, but my guess is this: to get to my router's user interface, I open http://192.168.0.1 in a web browser; therein, I can find that my router's IP address on the LAN is 192.168.0.1; so I'm guessing the "subnetwork of your LAN" is 192.168.0 and so I made the dhcp-range line look like this:

dhcp-range=192.168.0.0

Is that what you mean by "change 'dhcp-range' to the subnetwork of your LAN"?

5. sudo service dnsmasq restart

The response was

[ ok ] Restarting DNS forwarder and DHCP server: dnsmasq.

Moreover, when I then ran

sudo /etc/init.d/dnsmasq status

I got

[ ok ] Checking DNS forwarder and DHCP server: dnsmasq[....] (running).

So it looks like I'm ready for the next step.

6. "Start your clients with network boot"

Assuming that I've got everything right so far, this is where I could most use your help.

I did this:

I turned on the Windows 7 (thin-client-to-be) machine and repeatedly pressed F2 to enter its BIOS SETUP UTILITY. Went to its Boot screen and there, guessing completely, made two changes: (A) For Boot Device Priority, I moved [Network: Atheros Bo] from last place to first place: and (B), for good measure, I [Enabled] something called OnBoard LAN Boot ROM (which had until then been [Disabled], upon then saving and exiting, I saw the machine seemingly try to [not boot Windows as per usual], but that effort ended with

## Code:

PXE-E53: No boot filename received

PXE-MOF: Exiting Intel PXE ROM.

That message then disappeared and Windows 7 started up.

So close and yet so far?

**Adv Reply** 

March 3rd, 2014

#5

Join Date: Mar 2007

> Beans: 108



# Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

It seems like you did things correctly. On the client, you should be seeing the menu we defined in ltsp.conf ("Press F8 for boot menu", etc.)

There is a chance that your "Windows 7 computer" has an NIC with firmware that does not support newer features of PXE, such as ProxyDHCP. If it's possible at all, try putting other computers on your LAN and having them boot from the network. Laptops should be fine! The newer the computer, the more likely it will work.

To answer your question in #3: I learned about it here, but you'll see the solution repeated in many other guides, too. If you're really, really curious, the documentation for this option is here. (Why 3? It is a bitwise OR of both options 1 and 2, which we both want enabled.)

Adv Reply

March 3rd, 2014

Grugs & First Cup of Ubuntu

#6

Join Date: Dec 2005

Beans:

# Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

Hi,

Thanks for writing . . .

👥 Originally Posted by Emblem Parade 🗾

(Why 3? It is a bitwise OR of both options 1 and 2, which we both want enabled.)

... which gives peace-of-mind that 3 is not just some arbitrarily chosen number.

But thank you even more for writing . . .

👥 Originally Posted by Emblem Parade 🗾

On the client, you should be seeing the menu we defined in ltsp.conf ("Press F8 for boot menu", etc.)

... which brought to my attention that the ltsp.conf file should not end with...

Code:

# Better support for old or broken DHCP clients dhcp-no-override

... but rather with ...

```
# Better support for old or broken DHCP clients
dhcp-no-override

# Enable this for better debugging
#log-dhcp

# 
# PXE
#

# Note the file paths are relative to our "tftp-root" and that ".0" will be appended

pxe-prompt="Press F8 for boot menu", 3
pxe-service=x86PC, "Boot from network", /ltsp/i386/pxelinux
pxe-service=x86PC, "Boot from local hard disk"
```

... damn near-invisible scroll bar! Or maybe I just have tunnel vision. But I digress.

So having put *all* of what you suggested for ltsp.conf into it and re-running service dnsmasq restart, I returned to the thin client machine.

Turned it on. It booted Windows 7. Turn it off and on again, all the while connected to router via ethernet cable, re-entered the BIOS setup, left all as it was (still set to boot from the network), exited from the BIOS set up and got this on the screen instead of Windows 7:

### Code:

```
Intel UNDI, PXE 2.1 (build 082)
Copyright (C) 1997-2000 Intel Corporation
For Atheros PCIE Ethernet Controller v1.0.0.5(01/22/09)
CLIENT MAC ADDR: <thin client computer ethernet nic mac addr> GUID: <a quid>
CLIENT IP: <thin_client_computer_ip_address> MASK: 255.255.255.0
DHCP IP: <router lan ip address> PROXY IP: <terminal server computer ip address>
GATEWAY IP: <router lan ip address>
Auto-select:
     Boot from network
BOOT SERVER IP: <terminal_server_computer_ip_address>
!PXE entry point found (we hope) at 9CC4:0106 via plan A
UNDI code segment at 9CC4 len 2E9E
UNDI data segment at 9475 len 84F0
Getting cached packet 01 02 03
My IP address seems to be <thin_client_computer_ip_address>
ip=<thin_client_computer_ip_address>:<terminal_server_computer_ip_address>:<router_lan
BOOTIF=<a_hypen_separated_number>
SYSUUID=<the_same_guid_as_above_but_with_lower_case_letters>
TFTP prefix: /ltsp/i386
Trying to load: pxelinux.cfg/default
                                                     ok
```

then all that disappeared and was replaced with

```
connect: Connection refused
read: Connection refused
read: Connection refused
read: Connection refused
read: Connection refused
```

ad nauseum, until

```
Code:
```

```
NFS over TCP not available from <router_lan_ip_address> connect: Connection refused
```

Googled that and have, therefore, started to think that it is related to something that showed up when I ran sudo apt-get install ltsp-server dnsmasq

specifically this big \*\* WARNING \*\*:

### Code:

```
Unpacking ltsp-server (5.5.0-1) ...
Processing triggers for man-db (2.6.6-1) ...
Setting up nbd-server (1:3.7-1) ...

Creating config file /etc/nbd-server/config with new version

** (process:14414): WARNING **: Could not parse config file: The config file does not

** Message: No configured exports; quitting.
nbd-server.
Setting up nfs-kernel-server (1:1.2.8-6) ...

Creating config file /etc/exports with new version

Creating config file /etc/default/nfs-kernel-server with new version

[warn] Not starting NFS kernel daemon: no exports. ... (warning).

Setting up squashfs-tools (1:4 2+20130409-2)
```

I was reminded of the above when, guessing, I ran

/etc/init.d/nfs-kernel-server restart

and got in response:

## Code:

```
[ ok ] Stopping NFS kernel daemon: mountd nfsd.
[ ok ] Unexporting directories for NFS kernel daemon....
[warn] Not starting NFS kernel daemon: no exports. ... (warning).
```

I mention all this on the chance that it might be immediately obvious to you what the problem is.

**Adv Reply** 

Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

You've definitely moved forward!

The error message suggests that somewhere else you may have not copy-pasted a complete config file.

My suggestion to you is to start from scratch with a clean installation of Ubuntu 12.04 and follow the instructions here to the letter. I'll note that you do not have to install Ubuntu on "metal," but instead can install it inside a virtual machine. That way, it's easy to simply delete the VM if it doesn't work and start over. Great for experimentation. I recommend using VirtualBox: it's especially easy to use.

In case you're worried: I did test this guide using a VM, too. (Actually, you can even run the LTSP client in a VM! The Virtualbox BIOS support PXE boot.)

**Adv Reply** 

April 25th, 2014

felix10 🚨 First Cup of Ubuntu









#8

Join Date: Apr 2014

Beans:

Dealis

# Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

Hi, nice tutorial. I ran into a situation. I already have a Windows DHCP server. Do I need to tweak anything on the Windows DHCP? I didn't get the LTS to work that's the reason I'm asking the question. Also, my LTS is on Virtual server. Does that make any different?

Oh, and how about

# DHCP proxy on this network dhcp-range=192.168.1.0, proxy

Is the the IP of the Windows DHCP or the LTS-server? Thanks!

Adv Reply

April 26th, 2014



#9

Join Date: Mar 2007

> Beans: 108

# Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

Yes, it should work: the guide is exactly for situations where you already have a DHCP server existing on the LAN.

And it should work fine in a virtual machine, as long as the VM's network card is in bridged mode. You absolutely need the terminal server on the same LAN as the DHCP server and the terminals.

Then the dhcp-range would exactly be the range of the LAN as provided by the Windows DHCP server and used by the terminal server.

**Adv Reply** 

May 7th, 2014

felix10 🚨 First Cup of Ubuntu









#10

Join Date: Apr 2014

Beans:

# Re: HOWTO: Terminal server using Ubuntu 12.04 and LTSP in 15 minutes

I'm absolutely did something wrong with my configuration. I have my Ubuntu 14.04 LTS on network 192.168.110.0 and Windows DHCP on 192.168.101.0 and my pxe client are on the same as the LTS. How would I configure the ltsp.conf:

here's mine:

#Dnsmasq running as a proxy DHCP and TFTP server

#see: http://www.thekelleys.org.uk/dnsmasq...smasq-man.html

#

#TFTP

#

enable-tftp

tftp-root=/var/lib/tftpboot

**#DHCP** 

#set port to 0 so that dnsmasq knows its not being used as dns server

#port=0

#log-dhcp

**#DHCP** proxy on this network

dhcp-range=192.168.110.0,proxy

#Tell PXE clients not to use multicast discovery

#See section 3.2.3.1 in https://tools.ietf.org/html/draft-he...ot-protocol-00

dhcp-option=vendor@XEClient,6,2b

#Better support for old or broken DHCP clients

dhcp-no-overide

#Enable this fore better debugging

#log-dhcp

Service dnsmasq restart = fail.. what do i need to change?

**Adv Reply** 

Page 1 of 3 | 1 | 2 | 3

**Quick Navigation** 

Networking & Wireless | Top

« Previous Thread | Next Thread »

**Bookmarks** 



**Posting Permissions** 

You may not post new threads You may not post replies Vall may not pack attachments

**BB code** is On Smilies are On FINACT code is On All times are GMT +1. The time now is 05:06 AM.

vBulletin ©2000 - 2018, Jelsoft Enterprises Ltd. Ubuntu Logo, Ubuntu and Canonical © Canonical Ltd. Tango Icons © Tango Desktop Project. User contributions on this site are licensed under the Creative Commons Attribution Share Alike 4.0 International License. For details and our forum data attribution, retention and privacy policy, see here