

Making LTSP 5 work in fedora 8

- SUBHODIP BISWAS

While currently LTSP 5 supports only ubuntu and debian i.e there isn't any integration of LTSP 5 in fedora (work in progress) or CentOS yet . I tried to give it a shot and landed in the following.

I will be describing a step by step procedure to get to a working model of the same.

1) INSTALLATION:

Installation is pretty easy . Just download the ltsp 5 tarball from .

{http://ltsp.mirrors.tds.net/pub/ltsp/ltsp-5.0/ltsp_ubuntu_6.10_i386-2.tar.bz2}

This is for ubuntu 6.10 .

2) Extract the content of this compressed archive to /opt/ltsp/i386 .

Remember /opt/ltsp/i386 does not exist by default .so create it .

N.B . The above step is just for ease of understanding .it can be anything you like .

3) Now copy the content of /boot under /opt/ltsp/i386 to /tftpboot in fedora filesystem.

4) Now download two small scripts .

A] one is **ltsp-update-sshkeys** .get it from the url below .

```
http://ltsp.mirrors.tds.net/pub/ltsp/ltsp-5.0/ltsp-update-sshkeys-2.tar.bz2
```

N.B. LTSP 5 uses a more secure login now so it uses ssh which is more secure as compared to XDMCP. This simply sets up the ssh_known_hosts file to know about the server.

B] The other is **ltsp-update-kernels** .get it from url below.

```
http://ltsp.mirrors.tds.net/pub/ltsp/ltsp-5.0/ltsp-update-kernels-2.tar.bz2
```

N.B. This will copy the kernels from the ltsp tree to a place where your TFTP server can get them.

5) Now take the above mentioned scripts and dump them at `/usr/local/sbin/` .

6) Now run the scripts as root

```
#ltsp-update-kernels
```

```
#ltsp-update-sshkeys
```

CONFIGURING DHCP

This is the most important part of configuring a LTSP server . Because this serves the connection between Host and Client . In order to configure this part ,do the following

- 1) Select an ip for server ,that will serve as dhcp server .preferably select a class three ip range .
- 2) Select a suitable subnet and dns

```
option subnet-mask          255.255.255.0;
option domain-name-servers  10.82.6.1;
```

3) Select a domain name ,This is optional .here is a example

```
option domain-name          "choiceofyours.org";
```

4) Now the most important part , point to the root path i.e. From where it will fetch the filesystem via tftp , here it will be :

```
next-server                 10.82.6.1;
option root-path            "/opt/ltsp/i386";
```

N.B. Unlike previous version of LTSP<=4.2 .Here option root path does not contain the server ip i.e it **should not** be

```
option root-path            10.82.6.1:/opt/ltsp/i386  ;
```

5) Now point to kernel which tftp will fetch . In my case its ;

```
subnet 10.82.6.0 netmask 255.255.255.0 {
    range    10.82.6.100    10.82.6.199;
    if substring (option vendor-class-identifier, 0, 9) =
    "PXEClient" {
        filename "/ltsp/i386/pxelinux.0";
    }
}
```

6) Now that you are ready with your dhcp . Make sure that your dhcp server is running by using this as super user.

```
# /etc/init.d/dhcpd start
```

CONFIGURING NFS :

This actually serves the filesystem over the network . NFS can be configured by writing a proper **/etc/exports** file. Here is what i used in my **/etc/exports** file.

```
# LTSP-begin ##

/opt/ltsp
10.82.6.0/255.255.255.0(ro,no_root_squash, sync)
/var/opt/ltsp/swapfiles
10.82.6.0/255.255.255.0(rw,no_root_squash, async)

## LTSP-end ##
```

Now configure the **/etc/hosts** . This actually specifies the mapping of some host names to IP addresses before DNS can be referenced. This mapping is kept in the **/etc/hosts** file .

An example is like this :

```
127.0.0.1 home-server    localhost.localdomain  localhost
::1 localhost6.localdomain6 localhost6
10.82.6.2 ws002.ltsp     ws002
10.82.6.3 ws003.ltsp     ws003
10.82.6.4 ws004.ltsp     ws004
```

Now configure **/etc/host.allow** file ;

This file contains access rules which are used to allow or deny connections to network services that either use the **tcp_wrappers library** or that have been started through a **tcp_wrappers-enabled xinetd**.

An example of **/etc/hosts.allow** file :

```
## LTSP-begin ##
bootpd:      0.0.0.0
in.tftpd:    10.82.6.
portmap:     10.82.6.
## LTSP-end ##
```

SOME TWEAKS :

One most important point is ,

xinetd service must be enabled and running in order to have a xsession running in thin clients .

In order to do so there are some points to take care of: -

As this a ubuntu version running so there are certain changes than a normal fedora version.

a) Ubuntu uses **inetd** instead of **xinetd** so take care of that . Also from this version NBD is used and is fired over NFS. This is taken care of by installing **nbdrpm** which can be found here ^[1].¹

b)Ubuntu has **/etc/x11/xsession** but fedora has **/etc/x11/xinit/xsession** , so it is important to create link between them in order to have a xsession firing up. This can be done by

```
# ln -s /etc/X11/xinit/xsession /etc/X11/xsession
```

Now that most configurations are ready . Fire up the thin client . At this point actually ubuntu bits are executed .if every things all right you will be getting a ubuntu login screen. At this point it is necessary that a ssh authentication do takes place otherwise no login allowed .

¹ A fedora rpm and src rpm is found at <http://subhodip.fedorapeople.org/>

If you run `ltsp-update-sshkeys` for the first time in Fedora you will get the following error

```
ltsp-update-sshkeys: line 6: tempfile: command not found
```

Now it can be solved by downloading and installing **tempfile rpm** ^[2] from here .

Another problem encountered is

```
/var/log/syslog: No such file or directory.
```

This can be easily overcome by either creating a blank file as `/var/log/syslog` .

Now execute the command

```
#ltsp-update-sshkeys
```

The output will be something like the following.

```
Setting up ssh_known_hosts in:
```

```
  /opt/ltsp/i386
```

```
2
```

One important point , you may face problem while logging in . It is better to enable² password authentication for ssh . In order to enable this

```
#vi /etc/ssh/ssh_config
```

and remove comment from line 25 which states

```
password authentication yes
```

Now run the command after you are sure of all changes .

```
#ltsp-update-kernels
```

you will get a output like the one below.

² Normal rpm and src rpm is available at <http://subhodip.fedorapeople.org>

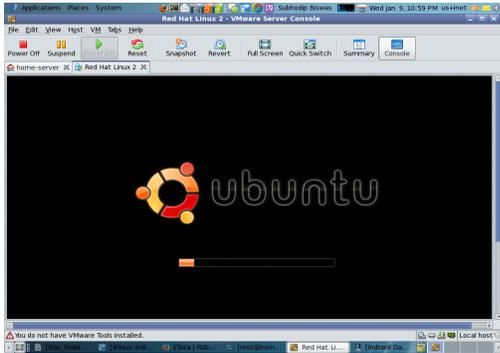
² Thanks to Jam (jam@trigger.ws) for providing me the centos rpms

Updating /tftpboot directories for chroot: /opt/ltsp/i386

Now you are ready to have nice fedora session up and running .

Screen shots:

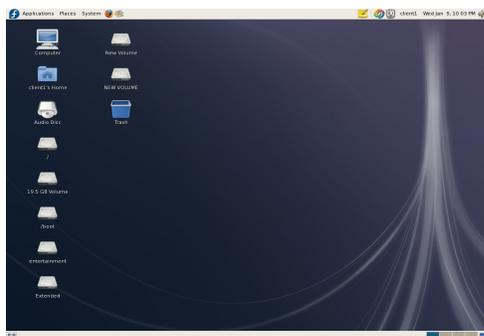
This shows ubuntu bits being executed .



This is login screen fired up



This is a thin client desktop .it is a fedora 8 desktop .



Annexure 1 : /etc/dhcpd.conf

```
# dhcpd.conf

ddns-update-style          ad-hoc;
allow booting;
allow bootp
option subnet-mask         255.255.255.0;
option broadcast-address   10.82.6.255;
option routers             10.82.6.1;
option domain-name-servers 10.82.6.1;
option domain-name        "subhodip-server.org";    # You
really should fix this
#option option-128 code 128 = string;
#option option-129 code 129 = text;

get-lease-hostnames        true;

next-server                10.82.6.1;
#option root-path
"10.82.6.1:/opt/ltsp/ubuntu_6.10_i386/";
option root-path           "/opt/ltsp/i386";

subnet 10.82.6.0 netmask 255.255.255.0 {
    range 10.82.6.100 10.82.6.199;
    if substring (option vendor-class-identifier, 0, 9) =
"PXEClient" {
#       filename "/lts/2.6.20.9-ltsp-1/pxelinux.0";

        filename "/ltsp/i386/pxelinux.0";
    }
}
```

```

    }
    else{
#         filename "/lts/vmlinuz-2.6.20.9-ltsp-1";
        filename "/ltsp/i386/nbi.img";
    }
}

# If you need to pass parameters on the kernel command line, you
can
# do it with option-129.  In order for Etherboot to look at
option-129,
# you MUST have option-128 set to a specific value.  The value is
a
# special Etherboot signature of 'e4:45:74:68:00:00'.
#
# Add these two lines to the host entry that needs kernel
parameters
#
#         option option-128         e4:45:74:68:00:00;         # NOT a
mac address
#         option option-129         "NIC=ne IO=0x300";
#

```

