Documentation / Installation

# Installation

Assumptions:
LTSP Server: 192.168.10.1
BOINCTasks Host: 192.168.1.200
BOINC Node Range: 192.168.10.10-100

> 📝 **Note** for LTSP5 users: it's possible to install the new LTSP in parallel with LTSP5, with the exception of the /etc/dnsmasq.d/ltsp-server-dnsmasq.conf file, which will need to be deleted before generating a new one with `ltsp dnsmasq`.

All of the terminal commands in the wiki should be run as root, which means you should initially run `sudo -i` on Ubuntu or `su -` on Debian.

don't require that sudo'ers have to enter password:
run 'visudo' and change %sudo line to '%sudo   ALL=(ALL:ALL) NOPASSWD: ALL'

## Server OS installation

The LTSP server can be headless, but it's usually better to install the operating system using a "desktop" .iso and not a "server" one. All desktop environments should work fine, but MATE and GNOME receive the most testing. Any .deb-based distribution that uses systemd should work; i.e. from Ubuntu 16.04 and Debian Jessie and onward.

In case you end up choosing Ubuntu MATE 20.04, @alkisg suggests running the following commands after installation, to save some RAM for older clients:

```
apt purge --yes --auto-remove indicator-application mate-hud snapd
apt install --yes synaptic
```

## Adding the LTSP PPA

The LTSP PPA is where stable upstream LTSP releases are published. It's mandatory for distributions before 2020 that have the older LTSP5, and optional but recommended to have in newer distributions. Follow the ppa page to add it to your sources, then continue reading here.

## Installing LTSP server packages

```
#add ppa source with:
add-apt-repository ppa:ltsp
apt update

apt install --install-recommends ltsp ltsp-binaries dnsmasq nfs-kernel-server openssh-server squashfs-tools ethtool net-tools boinc-client
```

```
create an inital ltsp.conf file from the template:
install -m 0660 -g sudo /usr/share/ltsp/common/ltsp/ltsp.conf /etc/ltsp/ltsp.conf

nano /etc/ltsp/ltsp.conf
add this under the [server] section to keep ssh keys in the image and allow SSH login to client,
see here for a potentially better way to handle this: https://github.com/ltsp/ltsp/discussions/310#discussioncomment-101549
OMIT_IMAGE_EXCLUDES="etc/ssh/ssh_host_*"

add this under the [client] section:
HOSTNAME=boinc-%{IP}
KEEP_SYSTEM_SERVICES="ssh boinc-client rsyslog"

add this section for each client:
[192.168.10.10]
FSTAB_1="192.168.10.1:/home/boinc-10 /var/lib/boinc-client nfs defaults,nolock,rsize=32768,wsize=32768 0 0"

set up client template directory, delete all boinc config files from regular conf dir
mkdir /home/template
mkdir /home/template/slots
mv /var/lib/boinc-client/remote_hosts.cfg /home/template
mv /var/lib/boinc-client/gui_rpc_auth.cfg /home/template
mv /var/lib/boinc-client/cc_config.xml /home/template
mv /var/lib/boinc-client/ca-bundle.crt /home/template
mv /var/lib/boinc-client/global_prefs_override /home/template
rm /var/lib/boinc-client/* -R
chmod 777 /home/template/*
chown boinc:boinc /home/template/slots
chmod 755 /home/template/slots

edit gui_rpci_auth.cfg to just be the word 'password'.  edit remote_hosts.cfg to include 192.168.1.200
follow steps at end for adding new client to create a directory for them and copy these template files
```

## Network configuration

There are two popular methods to configure LTSP networking. One is to avoid any configuration; this usually means that you have a single NIC on the LTSP server and an external DHCP server, for example a router, pfsense, or a Windows server. In this case, run the following command:

```
nano /usr/share/ltsp/server/dnsmasq/ltsp-dnsmasq.conf:
add 'dhcp-range=192.168.10.0,static' to only hand IPs to MAC addresses specified
comment out any existing dhcp-ranges and 'dhcp-range=set:proxy' line.
add 'dhcp-option=option:router,192.168.10.254'
add 192.168.1.200 as DNS server instead of 0.0.0.0
add 'dhcp-host=f4:8e:38:79:64:6f,192.168.10.10' for each host
save and then run 'ltsp dnsmasq'

edit the boinc-client startup script to add a delay to ensure NFS mount is ready:

systemctl edit --full boinc-client.service

add 'ExecStartPre=/bin/sleep 30' under the [Service] section

reload daemons with 'systemctl daemon-reload'
```

## Maintaining a client image

LTSP supports three methods to maintain a client image. They are documented in the ltsp image man page. You can use either one or all of them. In short, they are:

- Chrootless (previously pnp): use the server root (/) as the template for the clients. It's the easiest method if it suits your needs, as you maintain only one operating system, not two (server and image).
- ~~Raw virtual machine image: graphically maintain e.g. a VirtualBox VM.~~
- ~~Chroot: maintain a chroot directory using console commands. Note that the LTSP5 `ltsp-build-client` command is no longer supported, see the man page.~~

In the virtual machine and chroot cases, you're supposed to install the ltsp package to the image, by adding the LTSP PPA and running `apt install --install-recommends ltsp epoptes-client`, without specifying any other services. When the image is ready, to export it in *squashfs* format and make it available to the clients over NFS, run the following commands.

For chrootless:

~~`ltsp image /`~~

```
ltsp image /,,/boot,subdir=boot
```

see: https://github.com/ltsp/ltsp/issues/43

~~Virtual machines need to be symlinked before running `ltsp image`:~~

```
ln -s "/home/user/VirtualBox VMs/debian/debian-flat.vmdk" /srv/ltsp/debian.img
ltsp image debian
```

~~For a chroot in /srv/ltsp/x86_32:~~

```
ltsp image x86_32
```

You need to run these commands every time you install new software or updates to your image and want to export its updated version.

## Configuring iPXE

After you create your initial image, or if you ever create additional images, run the following command to generate an iPXE menu and copy the iPXE binaries in TFTP:

```
ltsp ipxe
```

In LTSP5, syslinux was used, but iPXE replaced it as it's much more powerful. You can read more about it in the ltsp ipxe man page.

## NFS server configuration

To configure the LTSP server to serve the images or chroots over NFS, run:

```
ltsp nfs
```

For finetuning options, see the ltsp nfs man page.

## Generate ltsp.img

A new procedure that wasn't there in LTSP5 is provided by the following command:

```
ltsp initrd
```

This compresses /usr/share/ltsp, /etc/ltsp, /etc/{passwd,group} and the server public SSH keys into /srv/tftp/ltsp/ltsp.img, which is transferred as an "additional initrd" to the clients when they boot. You can read about its benefits in its man page, for now keep in mind that you need to run `ltsp initrd` after each LTSP package update, or when you add new users, or if you create or modify /etc/ltsp/ltsp.conf, which replaced the LTSP 5 "lts.conf".

## Questions

Questions? Start a discussion or come to IRC live chat.

To view system logs: journalctl -b

create NFS export for boinc data folders. nano /etc/exports add this line:

```
/home   192.168.10.0/24(rw,async,no_root_squash,no_subtree_check)
```

to add a new client:

```
mkdir /home/boinc-10
cp -r /home/template/* /home/boinc-10/
chown root:root -R /home/boinc-10
chmod 777 -R /home/boinc-10
chmod 755 /home/boinc-10/slots
chown boinc:boinc /home/boinc-10/slots
```

nano /usr/share/ltsp/server/dnsmasq/ltsp-dnsmasq.conf to add reserved IP for new client
nano /etc/ltsp/ltsp.conf and add entry for new client

run ltsp initrd