LTSP Setup Guide

Contents

Articles

Article Licenses

Introduction

LTSP Setup Guide

A short overview on how to use this guide.

Introduction

Some background information about LTSP and file systems. This will help you to understand what you are doing in the following chapters.

Install

A step by step guide on how to install an Ubuntu system from the alternate CD.

Server Setup

This chapter explains how to setup the Server. Basic knowledge using the terminal is recommended. If you are new to Linux or Debian based systems, you should take a look at the administration section first.

Client Setup

This chapter explains how to setup the client chroots.

Administration

A quick overview on basic administration tasks like adding users, installing software and some useful terminal commands.

Introduction to LTSP and Thin Client Computing

Overview

One of the key technologies included in modern GNU/Linux operating systems is the Linux Terminal Server Project (LTSP) which allows you to boot thin clients from an LTSP server. For educational environments, LTSP lowers hardware costs by enabling the use of older or less powerful machines as thin clients, as well as reduced administration overhead by having only to install and maintain the software on the server. When a workstation fails, it can simply be replaced without data loss or re-installation of the operating system.

Thin client computing has been around for a long time in the UNIX world. Although the implementation has evolved quite a bit, the concept has remained the same:

1. The thin client only takes care of the basic functions like display, keyboard, mouse and sound.
2. The server does the heavy weightlifting. All the applications run on the server, and they simply display on the thin client.

Because the thin clients have a limited number of tasks to manage, the hardware for the thin client can be small and cheap. The thin clients themselves are basically maintenance free. They last longer because they have no storage with moving parts like hard disks. If they break no data is lost since nothing is stored on the client itself. Simply swap the client with another one and go back to work. If your thin client is stolen or put in the trash, no data ends up in the hands of unauthorized people.

The terminal server runs all applications and contains all the data. All the regular maintenance (software updates, administration) takes place on the terminal server. The number of thin clients that a terminal server can support is proportional to the power of the server. Because GNU/Linux makes efficient use of resources, you can support a surprising number of thin clients from a machine which might only be considered a powerful single user system running other operating systems. In a thin client computing environment, the stability of the server is essential. It's important to make sure that your server has good power facilities. As well, users who have the resources may decide to invest in multiple disks for RAID support.

Basic Concepts: Networks and Networking

Hardware

Networking works by breaking files and other data into little packets of information. These packets are transferred over a network. The difference between various types of networks is how they transfer packets. An important fact to remember is that a network will be only as fast as the slowest part. Making sure that your network setup matches your intended use case is an important consideration in an LTSP network.

Wired networking transfers packets over a cable that resembles a telephone cord, but with more wires. Wired networks can transfer packets at one of three possible speeds: 10 Mbit/sec, 100 Mbit/sec, or (Gigabit) 1000 Mbit/sec. A network is only useful if it can connect multiple computers. There are some pieces of hardware that allow multiple computers to be connected in a network. They look alike, but they function differently and, likewise, operate at different speeds.

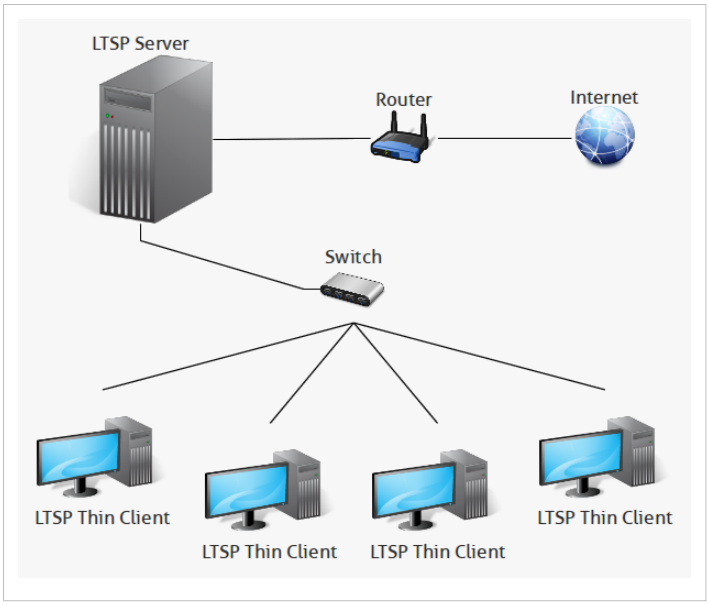
- **Switch** A switch looks a lot like a hub; it has a lot of ports in the front and usually has several small lights corresponding to each port. However, a switch is unlike a hub because it only makes a connection between the ports it needs to. A switch can have multiple connections at the same time. This allows a switch to be faster than a hub.
- **Router** A router is used to make a connection between two networks. Routers are also commonly used to connect a LAN (local area network) to the Internet.

Software

The most common network infrastructure services include:

- **DHCP (Dynamic Host Configuration Protocol)** Each computer on a network needs a unique identifier called an IP address. The IP address allows packets to be directed to the computer, much like a street address allows mail to be delivered to the correct house. An IP address follows a specific form: four groups of digits forming a number from 0 to 255. For example, a local IP address might be 192.168.2.50. For convenience, a computer's IP address can be given by a server running the Dynamic Host Configuration Protocol (DHCP) service. DHCP automatically provides network settings to the computers on the network. With DHCP, there is no need to keep track of each computer's IP address.
- **TFTP (Trivial File Transfer Protocol)** is a file transfer protocol known for its simplicity.[citation needed] It is generally used for automated transfer of configuration or boot files between machines in a local environment. It is an element of the Preboot Execution Environment (PXE) network boot protocol, where it is implemented in the firmware ROM / NVRAM of the host's network card. Due to the lack of security, it is dangerous to use it over the Internet. Thus, TFTP is generally only used on private, local networks.
- **PXE (Preboot Execution Environment)** is an environment to boot computers using a network interface independently of data storage devices or installed operating systems. The PXE protocol is approximately a combination of DHCP and TFTP, albeit with subtle modifications to both. DHCP is used to locate the appropriate boot server, while TFTP is necessary to download the initial bootstrap program and additional files.

How LTSP Works



LTSP is a collection of software that turns a normal GNU/Linux installation into a terminal server. This allows low-powered, low-cost thin-clients (or legacy hardware you already own) to be used as terminals to the thin-client server. LTSP is unique from other thin-client systems in that it is considered by many as the easiest to maintain. Other thin-client systems require each client to have software that boots the system to a point to be able to connect to the terminal server. This could be a full-blown operating system, or a minimal OS that simply provides an interface to connect to the server. Systems such as this generally require more maintenance and administration, as the local software that boots the thin-clients may become corrupt or contain bugs that require attention. LTSP, on the other hand, requires no client-side software. It requires only a PXE capable network interface, which many thin-clients and PCs have built-in already. This means that you need absolutely no physical storage media (hard disk, compact-flash, etc.) for your thin-client to boot to LTSP.

This significantly reduces the amount of administration required to keep your network running. The process of booting a thin- client to an LTSP server is as follows:

1. Thin-clients boot via a protocol called PXE (Pre-eXecution Environment)
2. PXE requests an IP address from a local DHCP server

3. The DHCP server passes additional parameters to the thin-client and downloads a Linux initramfs filesystem image via TFTP into a RAM disk on the client itself.
4. The thin-client then boots the downloaded Linux initramfs image, detects hardware, and connects to the LTSP server's X session (normally handled by LDM).

From here, all operations such as authenticating your username and password, launching applications, and viewing websites are actually handled on the LTSP server rather than the thin-client. The LTSP server transfers all graphical information to the thin-client over the network. This allows very low powered thin-clients to utilize the power of the server for all operations. It also allows for large client deployments with reduced overall resource utilization, as 50 thin-clients all running the popular OpenOffice suite under different sessions generally only require enough RAM for a single instance of OpenOffice (excluding per-user configuration which is minimal). The server shares memory between user sessions, so libraries for applications are only loaded once and referenced for each user session.

Further Information

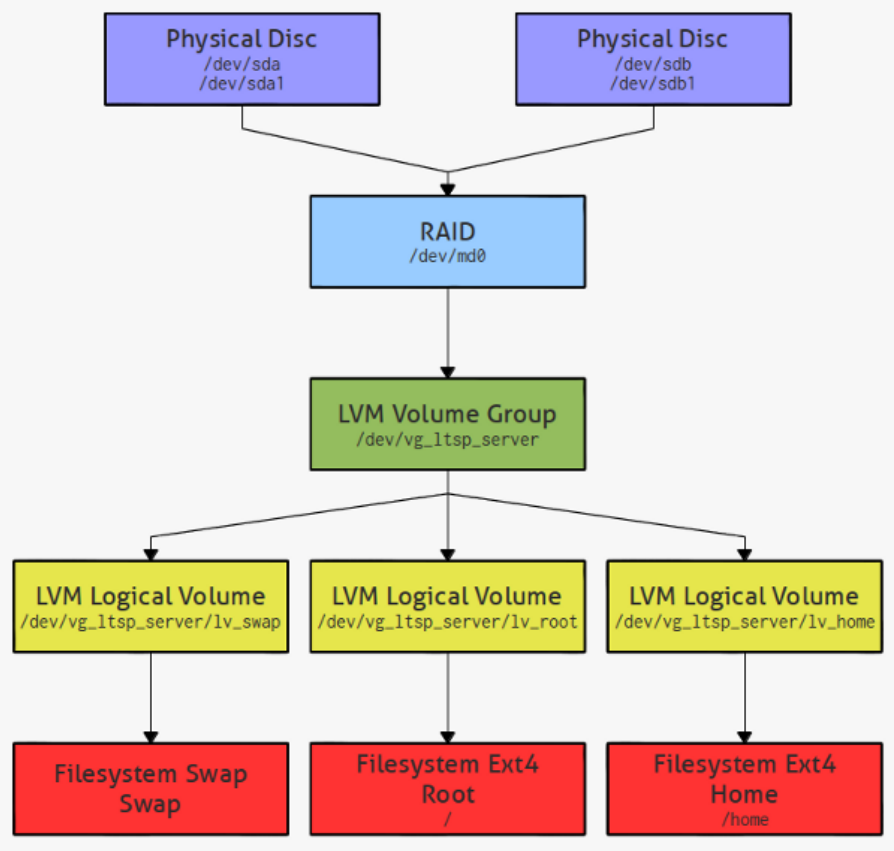
- Linux Terminal Server Project [1]
- LTSP Administrator's Reference [2]
- LTSP upstream development page [3]
- LTSP mailing lists [4]
- Ubuntu Help [5]

References

[1] http://www.ltsp.org/
[2] http://sourceforge.net/apps/mediawiki/ltsp/index.php?title=Ltsp_LtspDocumentationUpstream
[3] https://launchpad.net/ltsp
[4] http://sourceforge.net/mail/?group_id=17723
[5] https://help.ubuntu.com/community/UbuntuLTSP

Introduction to File Systems

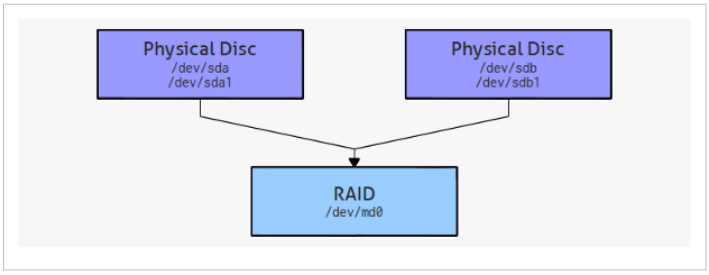
Overview



Schematic Overview

RAID

What is RAID?



RAID (redundant array of independent disks; originally redundant array of inexpensive disks) is a way of storing the same data in different places (thus, redundantly) on multiple hard disks. By placing data on multiple disks, I/O (input/output) operations can overlap in a balanced way, improving performance. Since multiple disks increases the mean time between failures, storing data redundantly also increases fault tolerance.

A RAID appears to the operating system to be a single logical hard disk. RAID employs the technique of disk striping, which involves partitioning each drive's storage space into units ranging from a sector (512 bytes) up to several megabytes. The stripes of all the disks are interleaved and addressed in order.

RAID-1

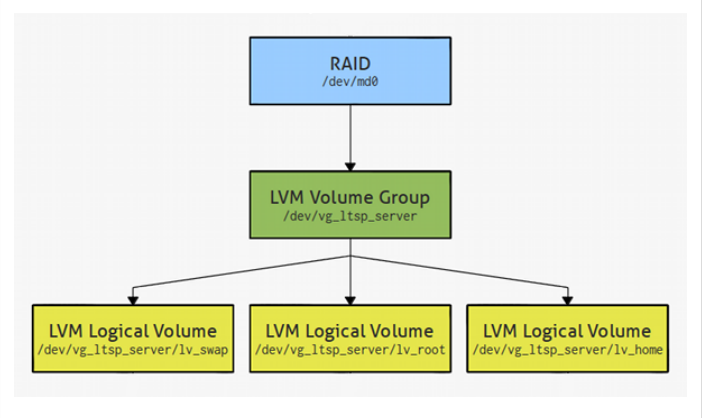
This type is also known as disk mirroring and consists of at least two drives that duplicate the storage of data. There is no striping. Read performance is improved since either disk can be read at the same time. Write performance is the same as for single disk storage. RAID-1 provides the best performance and the best fault-tolerance in a multi-user system. This allows overlapped disk I/O across drives.

Further Information

- Linux RAID Upstream [1]
- mdadm Introduction [2]
- Ubuntu Help [3]
- mdadm manpage [4]

LVM

What is LVM?



LVM (Logical Volume Management) makes use of the device-mapper feature of the Linux kernel to provide a system of partitions that is independent of the underlying disk's layout. With LVM you can abstract your storage space and have "virtual partitions" which makes it easier to extend and shrink partitions and add/remove partitions without worrying about whether you have enough contiguous space on a particular disk and without having to move other partition out of the way. This is strictly an ease-of-management issue: it does not provide any addition security.

The basic building blocks of LVM are:

- **Physical volume (PV)**: Partition on hard disk (or RAID device) on which you can have volume groups. It has a special header and is divided into physical extents. Think of physical volumes as big building blocks which can be used to build your hard drive.
- **Volume group (VG)**: Group of physical volumes that are used as storage volume (as one disk). They contain logical volumes. Think of volume groups as hard drives.
- **Logical volume (LV)**: A "virtual/logical partition" that resides in a volume group and is composed of physical extents. Think of logical volumes as normal partitions.

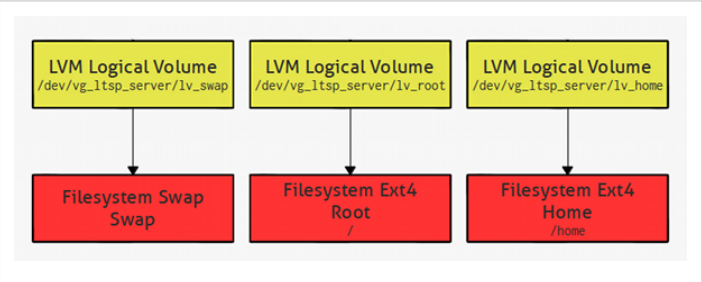
Further Information

- LVM Howto [5]
- lvm manpage [6]

File System Format

Ext4

Ext4 is the evolution of the most used Linux file system, Ext3. Ext3 was mostly about adding journaling to Ext2, but Ext4 modifies important data structures of the file system such as the ones destined to store the file data. The result is a file system with an improved design, better performance, reliability and features.



Mount Points

- **Root (mount point /)** In computer file systems, the root directory is the first or top-most directory in a hierarchy.
- **Home (mount point */home*)** Contains a subdirectory for each user on the system. A user's home directory is intended to contain that user's files; including text documents, music, pictures or videos, etc. It may also include their configuration files of preferred settings for any software they have used there and might have tailored to their liking: web browser bookmarks, favorite desktop wallpaper and themes, etc. The content of a user's home directory is protected by file system permissions, and by default is only accessible to that user and administrators.
- **Swap (no mount point)** Linux operating systems, use the term "swap" to describe both the act of moving memory pages between RAM and disk, and the region of a disk the pages are stored on. In some of those systems, it is common to dedicate an entire partition of a hard disk to swapping.

Further Information

- Ext4 Introduction [7]
- tune2fs manpage [8]
- Filesystem Hierarchy Standard [9]

References

[1] https://raid.wiki.kernel.org/
[2] https://en.wikipedia.org/wiki/Mdadm
[3] https://help.ubuntu.com/community/Installation/SoftwareRAID
[4] http://manpages.ubuntu.com/manpages/lucid/en/man8/mdadm.8.html
[5] http://tldp.org/HOWTO/LVM-HOWTO/
[6] http://manpages.ubuntu.com/manpages/lucid/en/man8/lvm.8.html
[7] http://kernelnewbies.org/Ext4
[8] http://manpages.ubuntu.com/manpages/lucid/en/man8/tune2fs.8.html
[9] https://en.wikipedia.org/wiki/Filesystem_Hierarchy_Standard

Install

Installer Setup

Overview

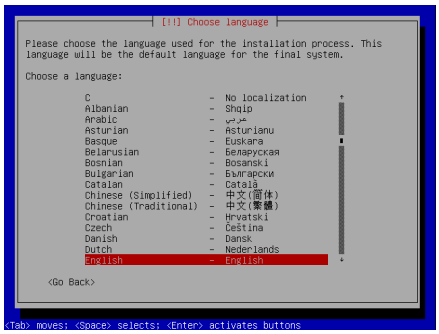
It is possible to install an LTSP server directly from the ubuntu-10.04.3-alternate-amd64 [1] CD

The installer will setup

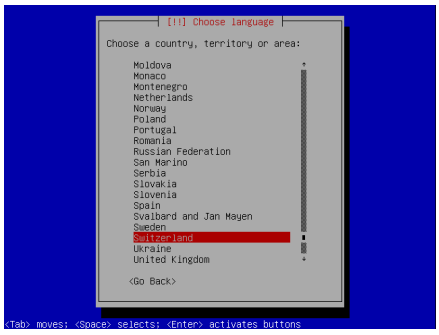
- an Ubuntu system with the ltsp-server packages and their dependencies
- a change root directory for the thin-clients
- a network boot environment with DHCP, PXE and TFTP

At the boot screen, press F4 and select *Install an LTSP server*

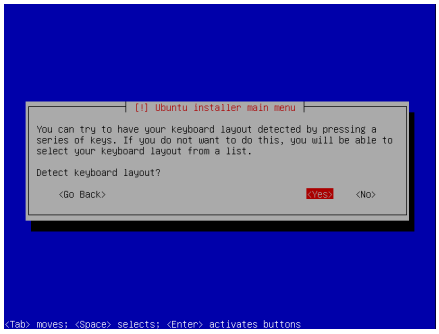
Locale



Select the default language for the installation process and for the final system.



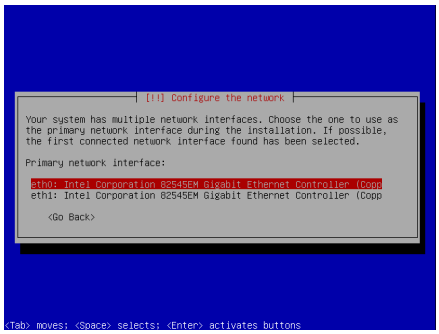
Select a country. This information will be used to setup the correct system time and the daylight saving time of your region.



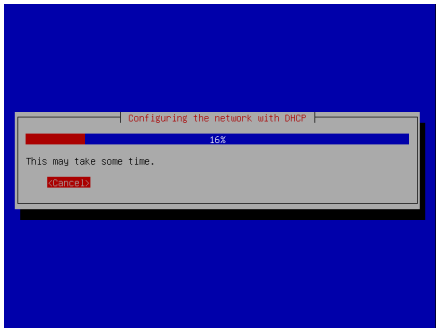
Select the default keyboard layout. You can either use auto detection or choose the layout from a list.

Network

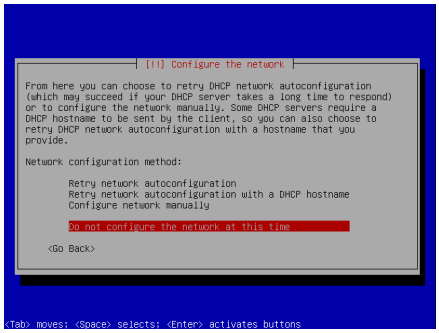
Disconnect all network cables from your computer. The network interfaces will be configured later in the running system. We don't rely on the auto configuration.



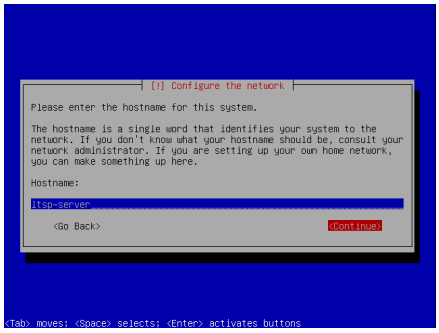
Configuring with DHCP will fail anyway, so just hit cancel.



Do not configure the network at this time.



The hostname is a unique name that identifies the machine to the local network. If your network admin did not give you a hostname to use, use something like *ltsp-server*.

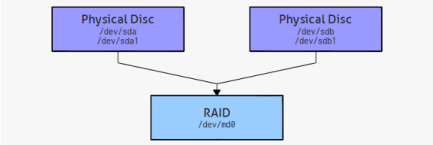


References

[1] http://www.ubuntu.com/download/ubuntu/alternative-download#alternate

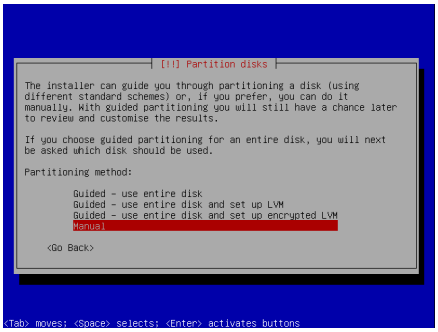
RAID Setup

Overview

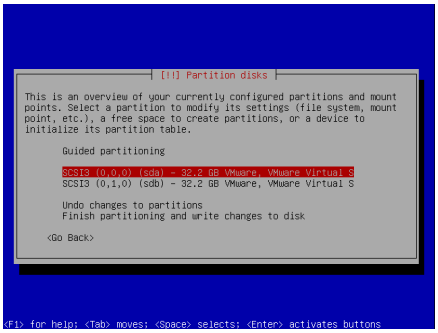


The RAID consists of at least two physical discs */dev/sda* and *dev/sdb*.

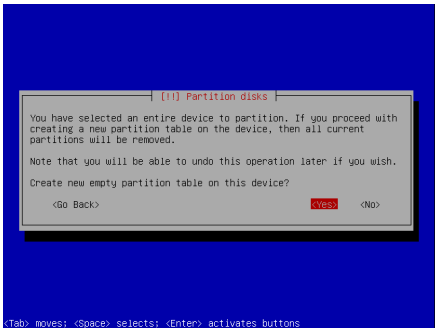
Partition Table



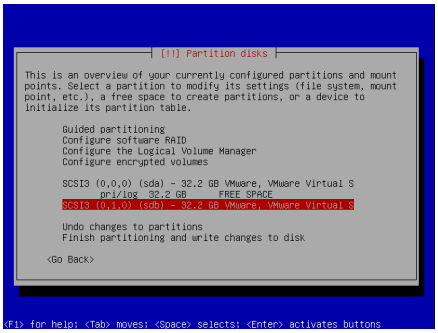
The manual configuration is necessary to setup a RAID



Delete all existing partitions on *sda*. This step will destroy all data on the disks.

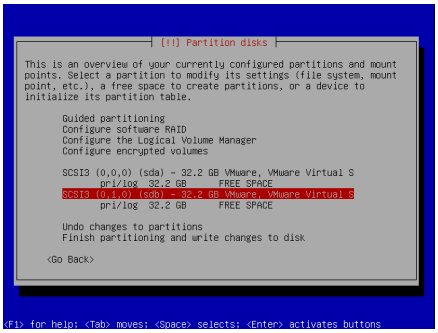


Create a new and empty partition table



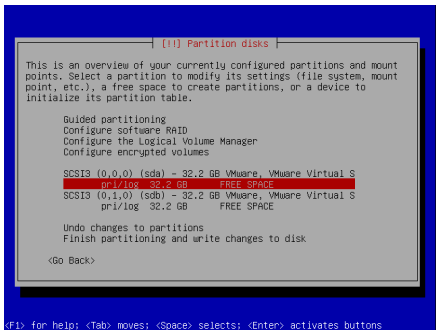
Repeat the steps above with the second device *sdb*

RAID

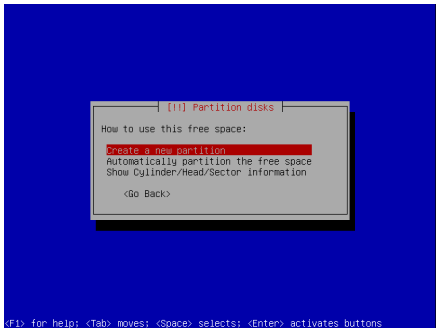


Both devices should contain *FREE SPACE* which reflects the empty partition table

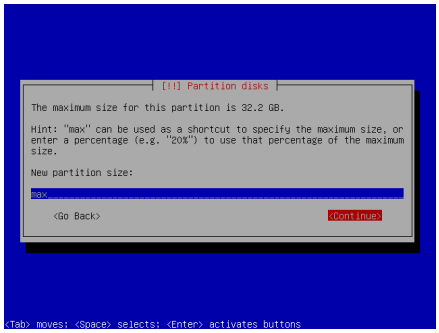
Now we create single partitions on both disks, select the *FREE SPACE* and hit enter.
If you are using the next LTS-Release of Ubuntu (12.04) you can skip this step and continue with MD Device. *The 10.04 Ubuntu installer has a bug and requires manual partitioning here.*



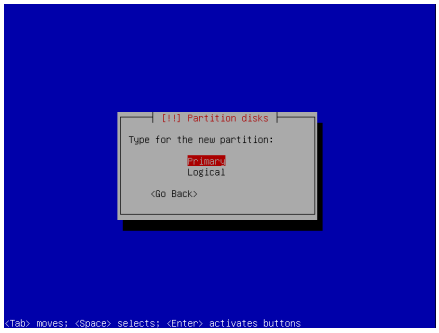
Choose *Create a new partition*



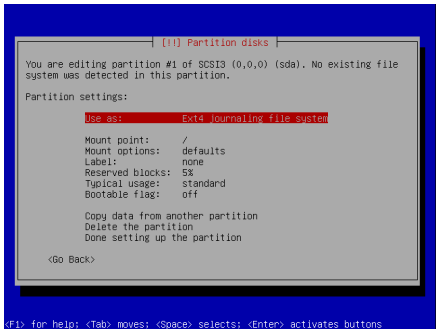
The partition uses the whole disk space, so enter *max* and press enter



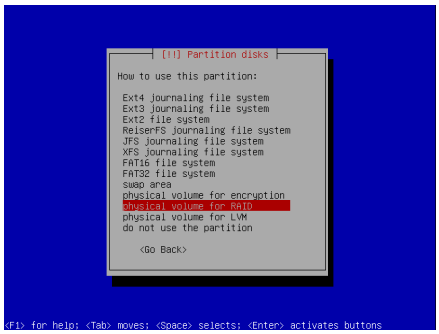
The partition is of type *primary*



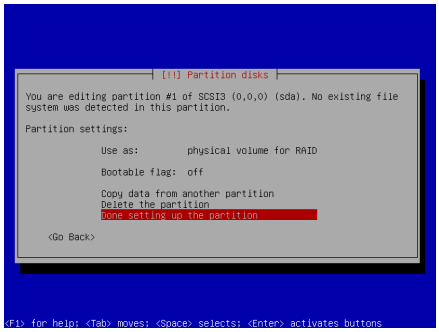
Select *Use as* to denote the partition for RAID setup.



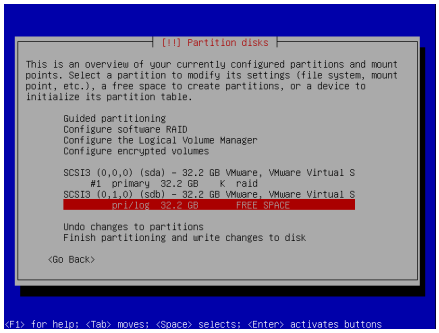
Select *physical volume for RAID*



Select *Done setting up the partition* to finish partitioning the device

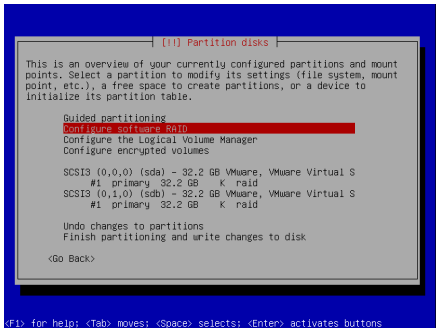


Repeat the above steps for the second hard disk (sdb)

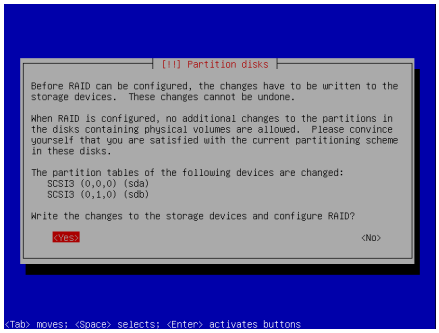


MD Device

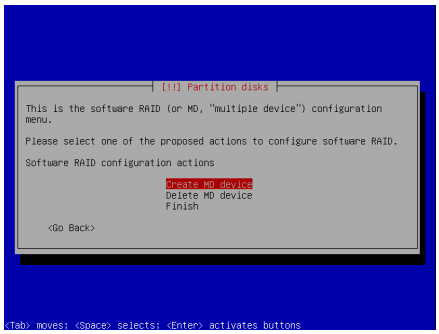
Finish the physical disk partitioning by choose *Configure software RAID*



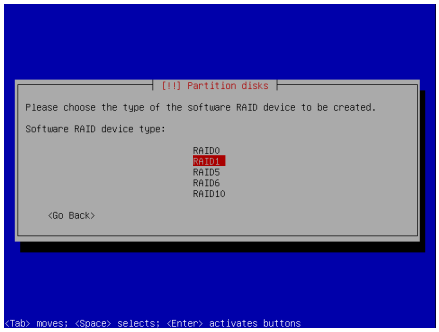
To continue, the previous setup has to be written to the disk. Choose yes to do so and continue configure the RAID.



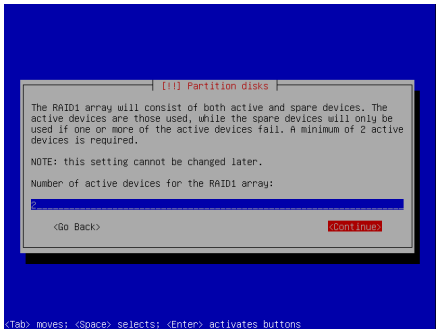
Linux Software RAID devices are implemented through the md (Multiple Devices) device driver



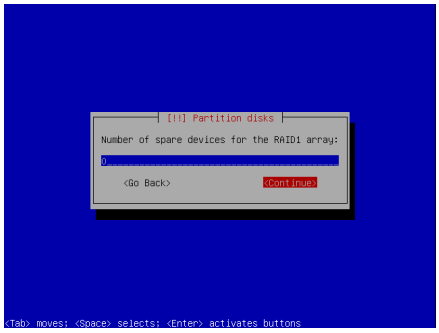
RAID1 will mirror the two assigned disks



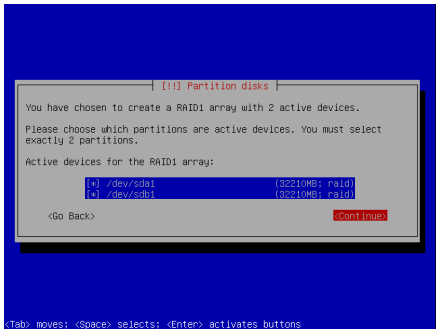
The RAID contains two devices



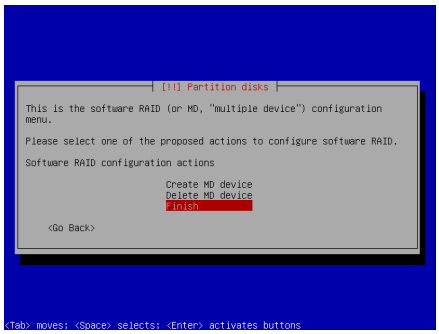
There are no spare devices



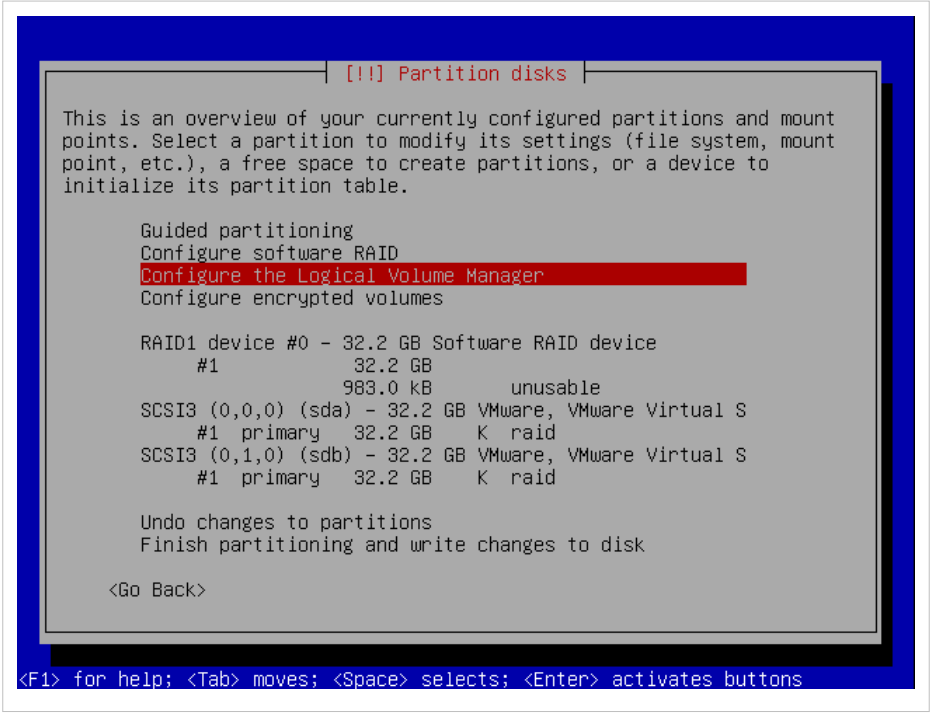
The two devices */dev/sda* and */dev/sdb* are assigned to the RAID



Choose *Finish* to write the changes to the disk



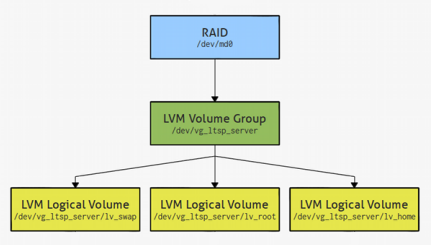
/dev/dm0 is now available as RAID device



LVM Setup

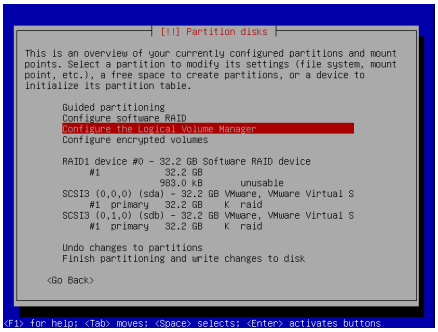
Overview

- We assign a Volume Group (VG) to */dev/dm0*
- We create three Logical Volumes (LV) for root,swap and home on the Volume Group *vg_ltsp_server*

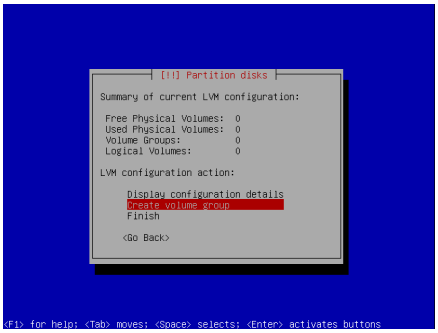


Volume Group (VG)

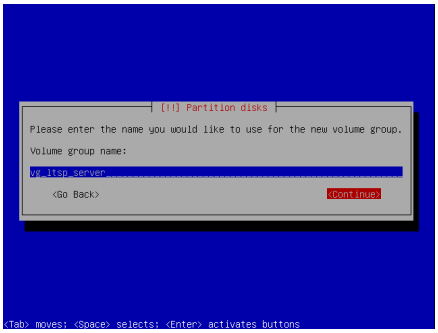
Configure the Volume Manager.



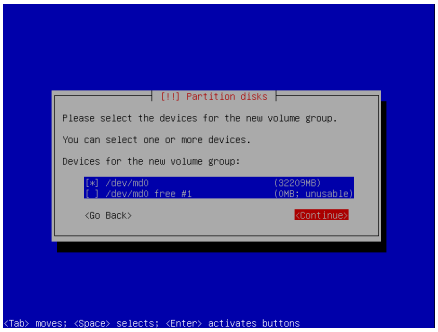
Create a new Volume Group.



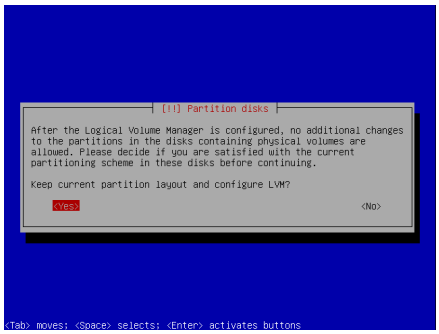
Name it *vg_ltsp_server*



Choose */dev/dm0* and ignore the second entry marked *unusable*

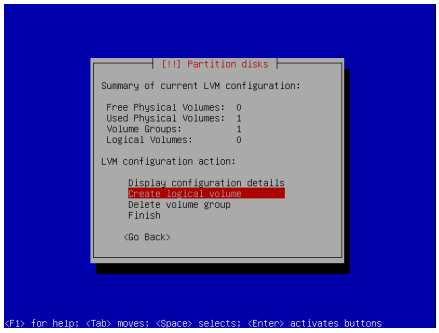


Confirm the operation

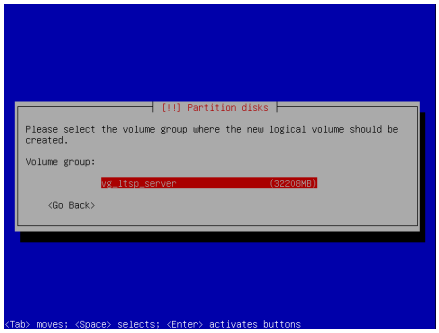


Logical Volume (LV) Swap

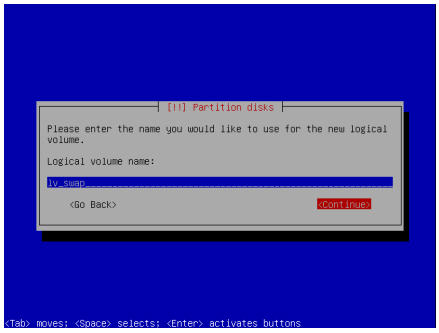
We create a Logical Volume with the name *lv_swap* on the Volume Group *vg_ltsp_server*. This LV will hold our swap which will be used if the server runs out of memory. Choose to *create logical volume*



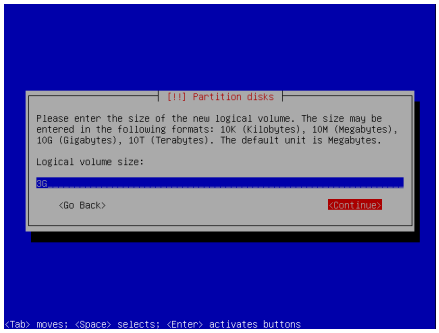
select *vg_ltsp_server* as *volume group*



use *lv_swap* as name



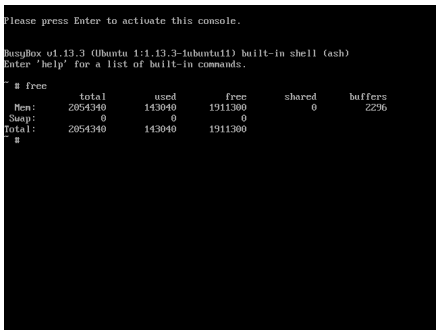
The size of this LV should match the amount of memory on your server and an additional one GB. In this example we have 2GB of memory and therefore create an LV with 3GB (2GB + 1GB).



To see how much RAM you have, press <ctrl+F2> to switch to *VT-2*, press enter and then type

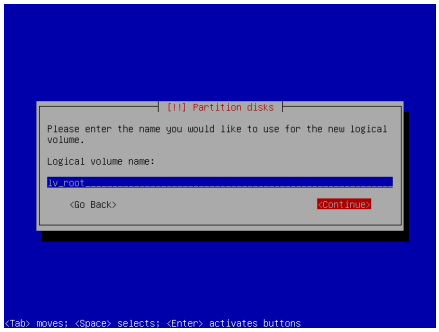
```
free -m
```

to display the amount in Megabytes. Press <ctrl+F1> to switch back to the installer (which resides on *VT-1*).

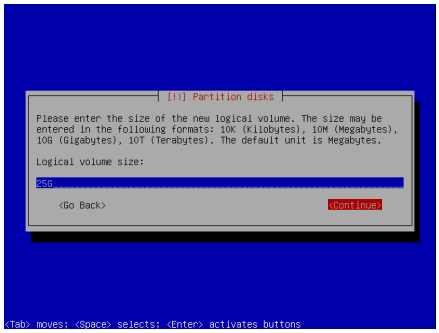


Logical Volume (LV) Root

We create another Logical Volume with the name *lv_root* on the Volume Group *vg_ltsp_server*. This LV will hold our root filesystem.

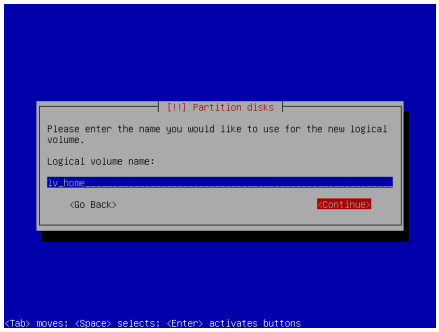


We recommend a size of 40GB which should be enough for the the operating system and all additional applications.

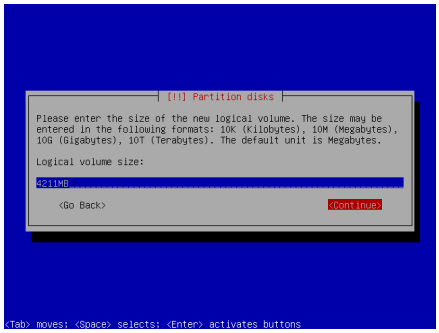


Logical Volume (LV) Home

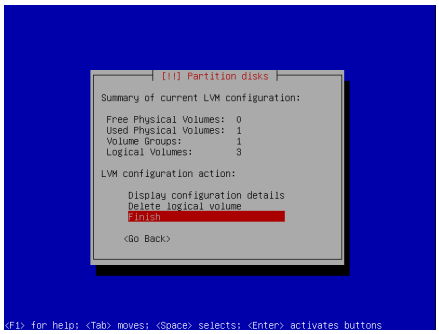
We create a Logical Volume with the name *lv_home* on the Volume Group *vg_ltsp_server*. This LV will hold all userdata.



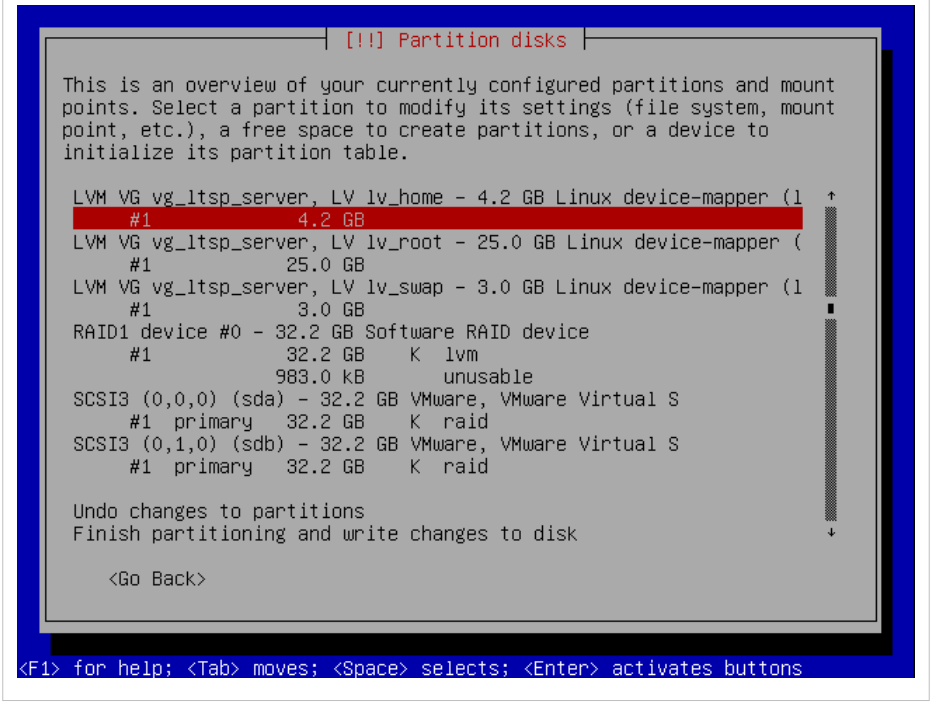
The default size for this LV is all empty space that is left after the previous steps. We don't change the proposed default value.



Choose to finish LVM configuration

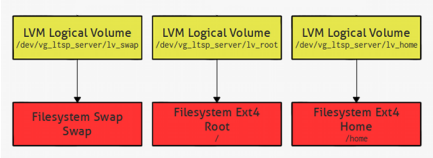


This now looks similar to this:



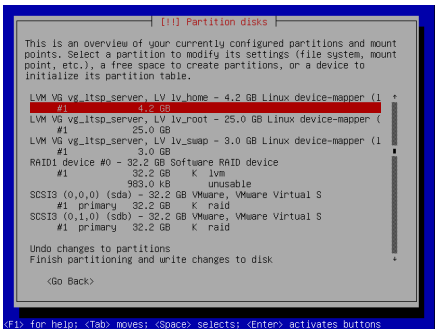
File System Setup

Overview

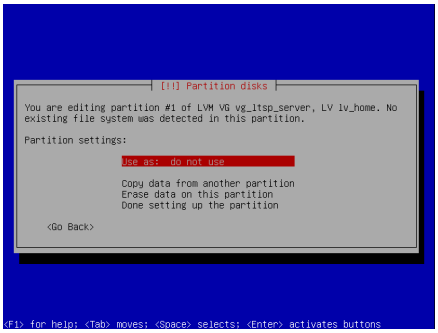


Home

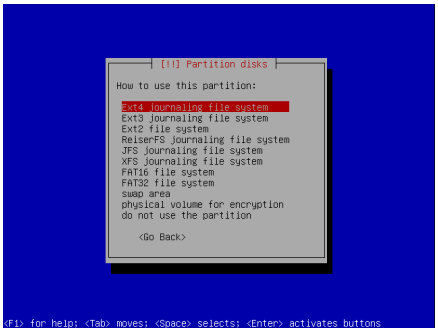
This partition will hold the /home directory. This is the place where all the user-specific files are stored. On a multi-user system, each user will have her own directory under /home select the logical volume *home*



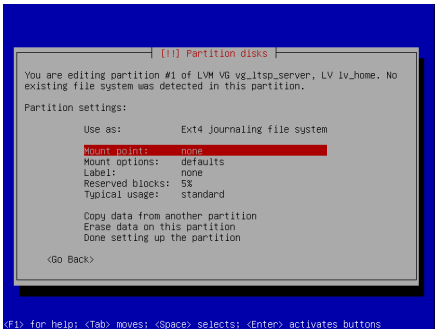
press enter to change the usage



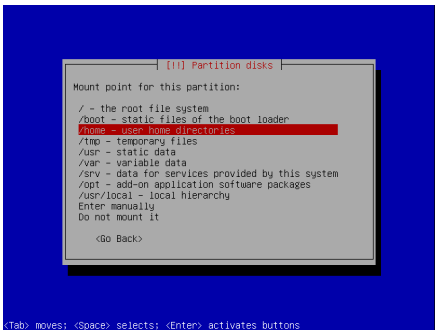
select *ext4*



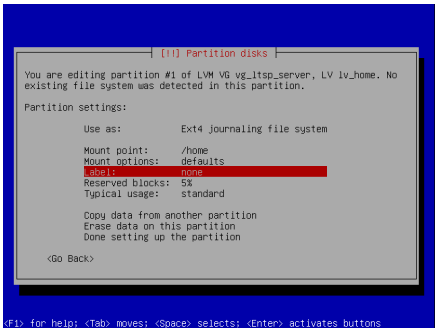
change the *Mount point*



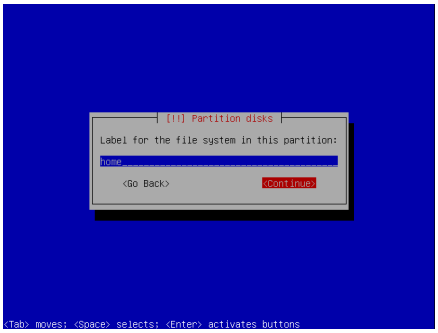
use it as */home*



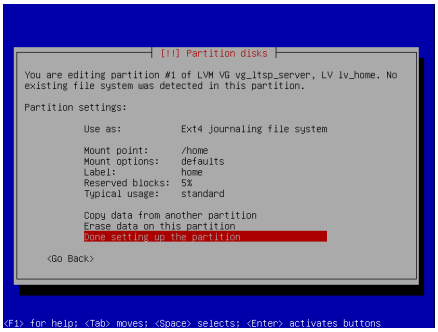
we recommend to set the Label



we recommend to use the vg-name *home*



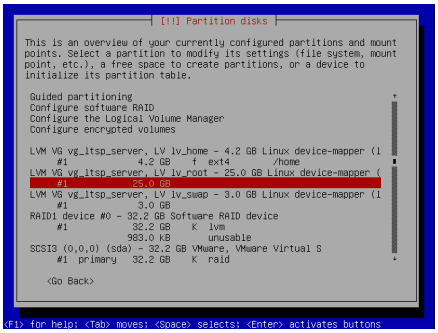
Finish the setup of the home partition



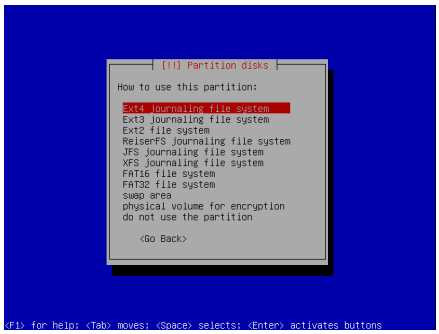
Root

The root file system is the top of the directory tree, and contains Linux and everything that is installed on the system. There must be a partition for the root directory. (Don't confuse this with the "root" user account, who is the administrator of the system. That's a different kind of root!)

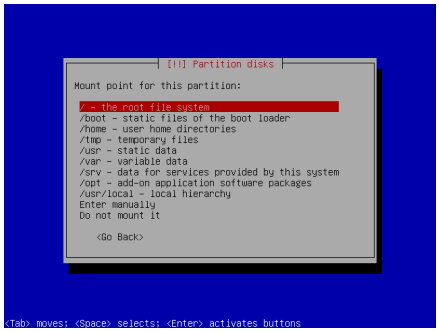
select the logical volume *home*



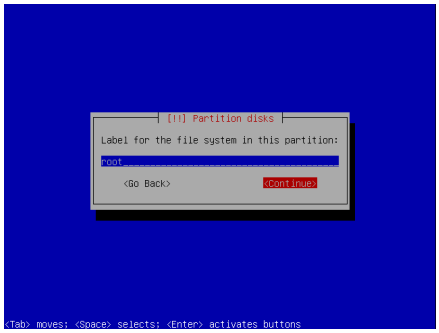
Use the filesystem as *ext4*



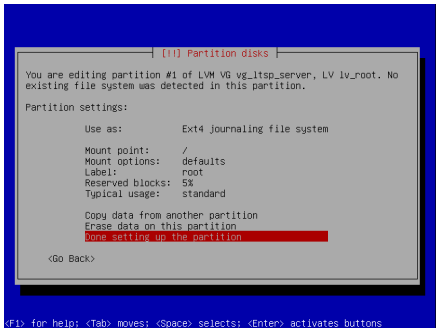
change the mountpoint to */ - the root file system*



we recommend to set the label to *root'*

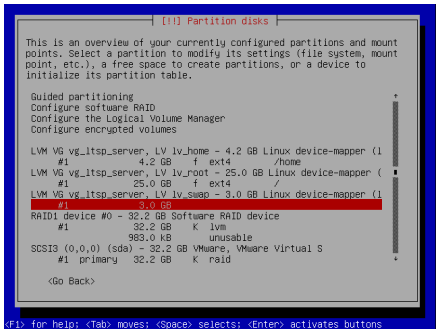


Finish the setup of the root partition

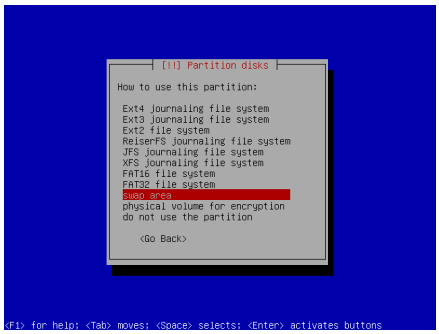


Swap

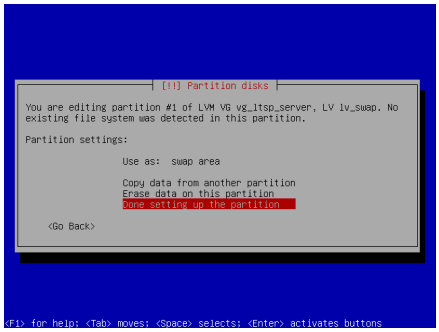
This partition will be used as Linux swap space, which is space on your hard drive that can be used as virtual memory. Virtual memory allows a computer to run large programs and perform complex tasks even if it does not have enough physical RAM select the logical volume *swap*



Choose to use the swap as swap area.

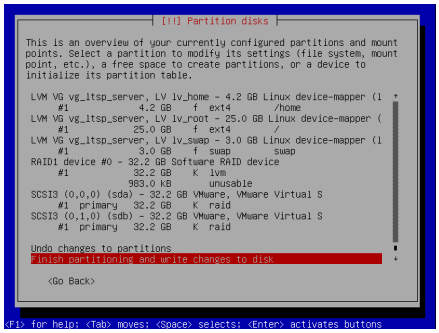


Finish the setup of the home

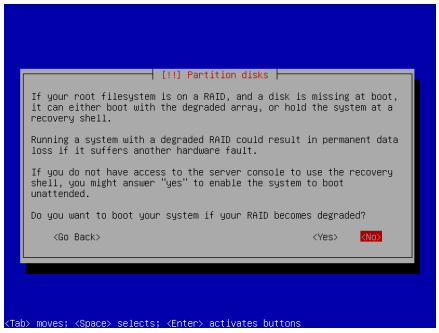


Write to Disc

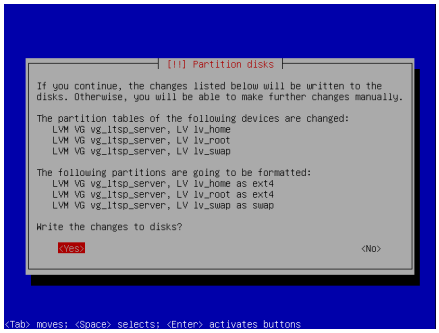
Finish the partitioning



To prevent Data loss, we suggest you fix the RAID before fully booting. Choose *No* to drop into a rescue system.



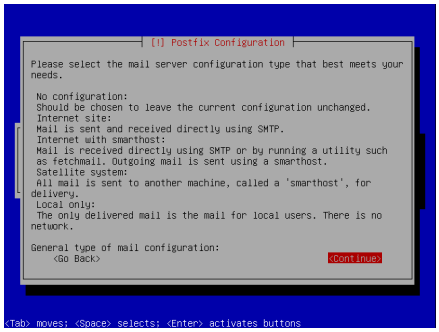
Review the changes and choose *Yes* to write them.



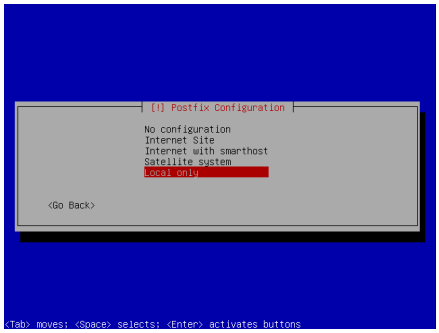
System Setup

Mail Server

Postfix is a free and open-source mail transfer agent (MTA) that routes and delivers electronic mail. It is intended as a fast, easier-to-administer, and secure alternative to the widely-used Sendmail.

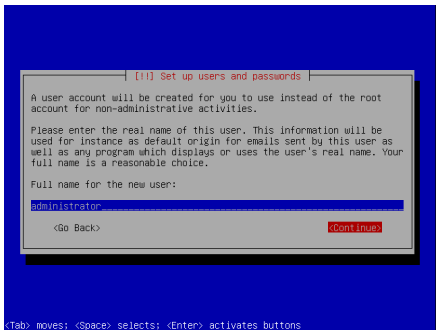


choose *local only*.

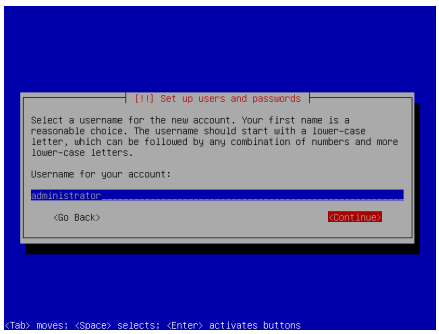


Administrator Account

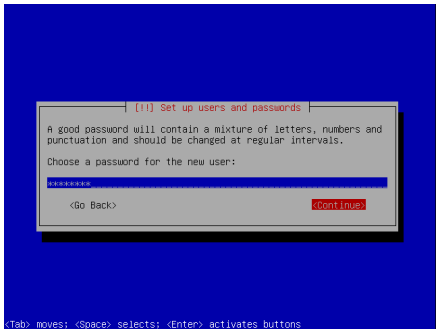
The administrative Account will be used for all system work.



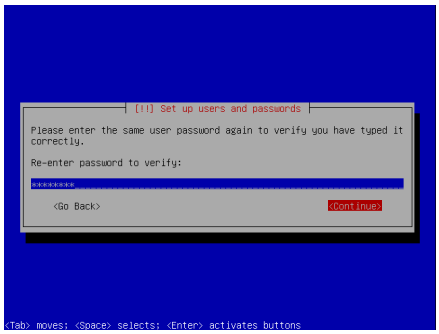
Use the suggested name for the unix acount.



Use a good password.

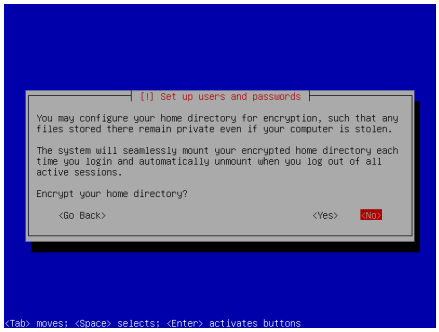


Confirm the password.



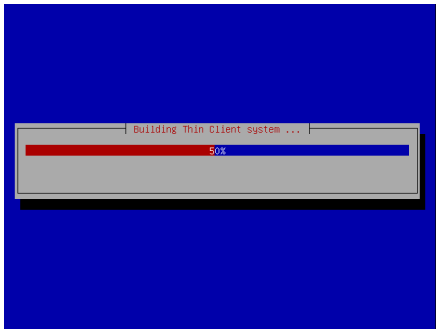
Encryption

Do not setup encryption for the administrative account. You want be able to access this account to fix things if somethings breaks.
However, its a good idea to use encrpytion for user accounts. See Administration->Add User for Details

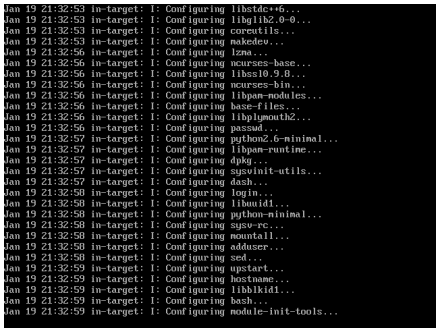


Thin Client

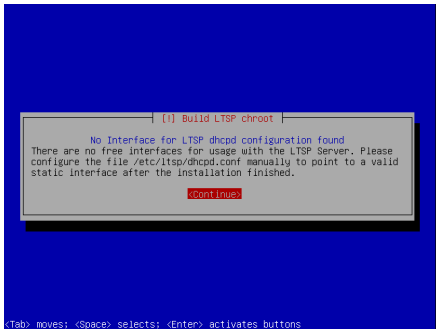
Building the client system will take some time.



To watch the progress, press <ctrl+F4> to switch to VT-4 Press <ctrl+F1> to switch back to the installer (which resides on VT-1).

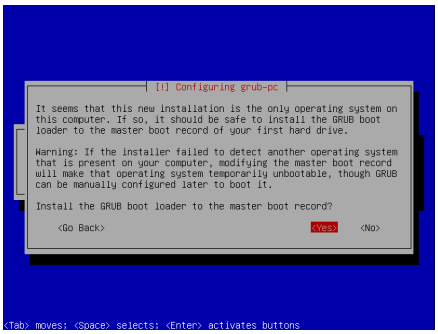


Ignore the error message and press *Continue*.

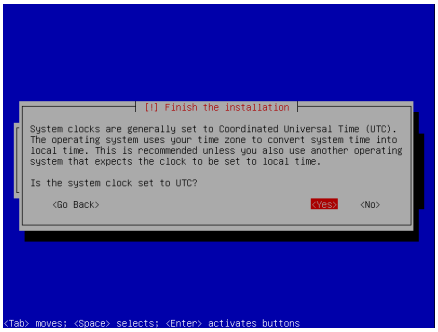


Finishing steps

Install *GRUB* to the master boot record.

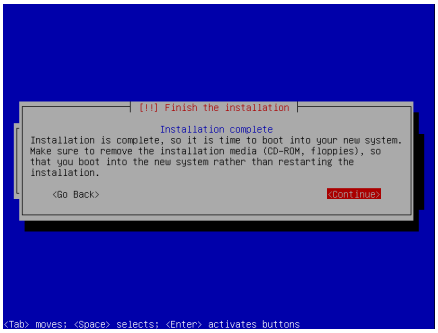


As we are the only System on the server, use UTC



Reboot

Remove the installation medium and reboot the system.



Server Setup

Networking

Manual vs Managed Networking

- Most modern Linux Distribution offer a easy way to automatically set-up networking. This method is best suited for mobile systems and workstations
- Unfortunately, the automatic process is not perfect and introduces trouble in a static set-up
  For example, one problem in systems with multiple network interfaces are server daemons not behaving nice.
- As most servers are never changing their network, it is therefore advisable to set-up networking manually
- Network setup usually is specific for every system. It is **very important** to properly document this as most network problems will not immediately occur and are hard to debug without proper documentation.

Basic Network Knowledge

Network Interfaces

Todays networking uses Ethernet with RJ45-plugs. This Guide assumes you have 2 network interfaces, one for external and one for internal traffic.
If your server only has 1 network interface, we strongly recommend to get a second network controller (cheap old 10MB cards are enough for external networking). This will save you a lot of brain-damage

To find out how linux names your network interfaces, use the ifconfig command

```
administrator@ltsp-server:~/$ ifconfig -a
eth0      Link encap:Ethernet  HWaddr 00:0c:29:04:8f:a2
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:04:8f:ac
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 B)  TX bytes:720 (720.0 B)
```

You should see the 2 interfaces named **eth0** and **eth1**, the third interface **lo** is a virtual internal interface and can be ignored.

To find out which interface corresponds to which name, connect one Interface with a counterpart (The LED in the interface should light on) and use ifconfig again

```
administrator@ltsp-server:~/$ ifconfig –a
eth0      Link encap:Ethernet  HWaddr 00:0c:29:04:8f:a2
          inet6 addr: fe80::20c:29ff:fe04:8fa2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:2178 (2.1 KB)

eth1      Link encap:Ethernet  HWaddr 00:0c:29:04:8f:ac
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:720 (720.0 B)  TX bytes:720 (720.0 B)
```

The one saying **RUNNING** is the one you connected. As you see, the virtual internal network lo0 is *connected* too.

For the remaining parts, we assume you will use eth0 for internal and eth1 for external connection. It may be a good idea to physically tag the plugs (use O and 1).
These names can change if you replace the cards!

IP (v4, v6)

To talk to each other, every computer on the network requires a number usually called *IP*. Each Network interface needs such a number to be used. Until recently, these numbers have been 32bit written as 4 numbers between 0 and 255 separated by dots.
Exapmles are

```
127.0.0.1
8.8.8.8
88.198.81.73
192.168.24.1
```

Devices called *routers* or *gateways* know where a computer with a given number is and *route* the traffic to that place. *IP addresses* usualy are *global* and must be unique in the world. A Organisation called Internet Assigned Numbers Authority [1] administrates who gets what numbers (actually, they delegate this work to *Regional Internet Registries*).

As there is a limited number of *IP addresses* and a lot of *internal networks* do not require worldwide networking, there are some special *IP ranges* which a router should never send out to other networks. This allows those ranges to be used without asking the IANA.

These Ranges are

RFC1918 name	IP address range	number of addresses	*classful* description	largest CIDR block (subnet mask)	host id size
24-bit block	10.0.0.0 – 10.255.255.255	16,777,216	single class A	10.0.0.0/8 (255.0.0.0)	24 bits
20-bit block	172.16.0.0 – 172.31.255.255	1,048,576	16 contiguous class Bs	172.16.0.0/12 (255.240.0.0)	20 bits
16-bit block	192.168.0.0 – 192.168.255.255	65,536	256 contiguous class Cs	192.168.0.0/16 (255.255.0.0)	16 bits

As the ltsp-clients do not require external network, we use a part of the 10.0.0.0/8 range for the internal networking.

Now, there is one quirk with this set-up. If the routers know where to send the Data, then everything needs to go trough the router, even if the other computer is on the same network. To tell a system what other Computers are on the same network (and can be spoken directly to) you specify a *subnet-mask*. Historically the *subnet* have been all devices on the same Ethernet link (the same cable). Using modern routers and switches has expanded this definition. If your *Local Area Network* uses the range from 192.168.1.1 to 192.168.1.254 then your subnet mask is 255.255.255.0. If you use the full range from 192.168.1.1 to 192.168.255.254 then your subnet mask is 255.255.0.0. It is important, that all systems on the same network use a correct setting for this. If set wrong networking usualy still works, but you will have weird connection problems. **if networking behaves strange always check if subnet masks are set correctly by using *ipconfig* .**

loopback

One more special range is *127.0.0.0/8* with addresses from 127.0.0.1 - 127.255.255.255. At least 127.0.0.1 is assigned to the loopback interface *lo0*. Talking to this IP is talking to yourself. A lot of Linux/Unix Applications work in a Server-Client fashion, and talk over the loopback with themselves. This design easily allows applications to work with other computers over the network and is the basis for the whole ltsp-set-up.

IPv6

32bit allow *4,294,967,295* (2^32 – 1) addresses. This is a lot, however humanity has reached this limit. Years ago, this problem was solved by a new type of IP addresses called IPv6. These addresses are 128bit long which allows 340,282,366,920,938,463,463,374,607,431,768,211,455 IP addresses. This should last some time.

As IPv6 is not deployed globally, IPv4 is still the preferred way to go. As we use a private IPv4 range in our internal network, we do not care about IPv6 at all. IPv6 comes into play if your external network provider only talks IPv6. Configuring IPv6 for end users is similar to IPv4.

The main difference with IPv4 is how addresses are written. As 128bit is 4 times 32, using numbers with dots is not comfortable. The new syntax uses Hex-Numbers and groups the IP by colons

```
2a00:1450:4001:c01:0:0:0:69
2a01:4f8:131:10e1:ff::75
::1
```

As a shortcut any chain of *:0:* can be simply shorted into *::*. The last address in the example is 0:0:0:0:0:0:0:1.

Since there are so many IPv6 addresses, it was decided to use some of them for special purposes to enhance networking. These special ranges are:

- *::1* like 127.0.0.1 this links back to yourself.
- *FF:: multicast* range. Data sent to these addresses will be sent to multiple systems. see below.
- *FE8::* Local Lan Scope. Similar to the IPv4 private ranges, these numbers are never sent outside the local network. You can use this IPs to connect to another computer on the same network, but not to everything outside. Every IPv6 enabled interface will **automatically** generate such a link local IP address.
- *FEC::* Local Site Scope. These are similar to the Local Lan Scope addresses above, but should be *routed inside the same organisation*. This is intended to connect multiple networks in the same organisation together (the idea was to generalize the *subnet* idea from above and make routing more simple). However, as *the same organisation* can not always be defined this feature of IPv6 is currently considerer *not in use*.

The *FF::* range has some special meaning.
Traffic sent to these will be sent and processed by nodes depending of their network *role*. Usually you can only use these IPs to send traffic to, but not to directly receive from them.
Some example usages are

- FF01::1 All nodes on the same interface (thats yourself. similar to loopback)
- FF02::1 All Nodes on the same Link (eg same subnet)
- FF02::2 All *routers* on the same network
- FF02::1:2 All dhcp agents (eg servers who assign ip addresses dynamically)

There are a lot more of those Multicast types, but since IPv6 is still not in wide use, they are usually not used.

Setup

Internal Networking

The internal network is used by the ltsp-clients to connect to the server and run applications.
As the workstations do not require external network access themselves (they will use the server), we will use a *private ip range*

IP

- The internal network will use a /24-subnet inside the 10.0.0.0/8 private range.
- The server itself will always have the same ip, it is common practice to use either the first (.1) or last (.254) ip available.

Open the network configuration file

```
administrator@ltsp-server:~/$ sudo nano /etc/network/interfaces
```

append these lines to the file.

```
auto eth0
iface eth0 inet static
  address 10.55.66.1
  netmask 255.255.255.0
```

in case there are already any lines containing eth0, delete them. You can add a comment by starting the line with a # char (remember to add a lot of documentation and comments to make your life easier). You can choose any number between 0 and 255 for the middle part (55.66) of the address
hit ctrl+x, y, enter to exit and save the file. Restart the network

```
administrator@ltsp-server:~/$ sudo /etc/init.d/networking restart
```

External Networking

External networks provide access to other networks to the ltsp-clients.
There are 2 common ways to setup a network, ask your network admin which to use.

If you have more a complex setup (for example connection sharing) you may want use Network Manager. In that case do not configure the external interface at all and skip the steps below.

DHCP

Using DHCP makes everything very easy. Most modern routers and modems will offer dhcp for configuration

```
administrator@ltsp-server:~/$ sudo nano /etc/network/interfaces
```

Append these lines to the file

```
auto eth1
iface eth1 inet dhcp
```

Close the file by hit ctrl+x, y, enter.
Now restart networking to check if everything works

```
/etc/init.d/networking restart
```

If all is fine, you should now have external connection on your server.

Specific IP

If your network does not offer dhcp, you need to get 3 values from your network administrator

- **ip** the ip your host shall use.
- **netmask** the network mask
- **gateway** the gateway to use (usualy the router/modem ip)

additionally you may get

- **broadcast** this can actually be calculated from ip and netmask

open the network confiuration file:

```
administrator@ltsp-server:~/$ sudo nano /etc/network/interfaces
```

Append these lines to the file

```
auto eth1
iface eth1 inet static
  address   ip
  netmask   netmask
  gateway   gateway
```

Close the file by hit ctrl+x, y, enter.
Now restart networking to check if everything works

```
administrator@ltsp-server:~/$ /etc/init.d/networking restart
```

If all is fine, you should now have external connection on your server.

in the rare case your network admin gives you **IPv6** data instead IPv4, you add them in the same way:

```
auto eth1
iface eth1 inet6 static
  address   v6ip
  netmask   v6netmask
  gateway   v6gateway
```

it says inet6 instead of inet!

if you have both IPv4 and IPv6 addresses, you use them both:

```
auto eth1
iface eth1 inet static
  address   ip
  netmask   netmask
  gateway   gateway
iface eth1 inet6 static
  address   v6ip
  netmask   v6netmask
  gateway   v6gateway
```

auto eth1 is only written once!

References

[1] http://www.iana.org

Mirror and Update

Mirror Setup

It is recommended to use a Internet connection for updating. If this is not possible, a local device like USB hard disks can be used.
The installer already has prepared everything to use internet updating and you may skip ahead to Updating.

Using a external local device

To use a USB hard disk, you need add the Repository on the USB Hard Disk to the System. Read more about repositories in the Repositories Administration Part -> Package Management.
Creating a local repository is not a simple Task, The default Ubuntu Repositories contain about 400 Gigabyte of Software. See below for more information on how to create and update a local Repository.

As described in Repositories Administration Part -> Package Management you add the repository in */etc/apt/sources.list.d/*

If your Disk is prepared as described below, just copy the *repository file* from the disk

```
administrator@ltsp-server:~/$ sudo cp /media/ubuntu-ltsp/usbmirror.list /etc/apt/sources.list.d/
```

alternatively you can use nano

```
administrator@ltsp-server:~/$ sudo nano /etc/apt/sources.list.d/usbmirror.list
```

and insert the repository

```
deb file:///path lucid main restricted universe multiverse
deb-src file:///path lucid main restricted universe multiverse
deb file:///path lucid-updates main restricted universe multiverse
deb-src file:///path lucid-updates main restricted universe multiverse
deb file:///path lucid-security main restricted universe multiverse
deb-src file:///path lucid-security main restricted universe multiverse
deb file:///path lucid partner
```

where **path** points to the directory containing the mirror (it contains subdirectories like *dist*, *pool*).
Usually the **path** is something like */media/ubuntu-ltsp/mirror*.
To not use your internet connection for updating aven if available,make sure to disable autoupdate as described in Repositories Administration Part -> Package Management

Updating

First let apt update its package list

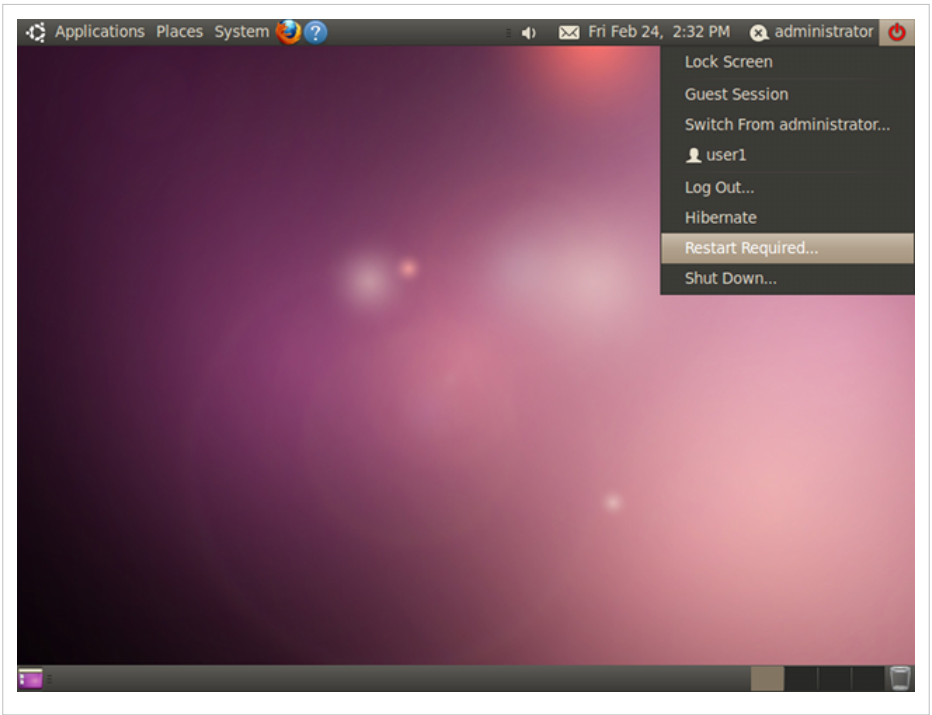
```
administrator@ltsp-server:~/$ sudo apt-get update
```

If you are using a local device, you may see multiple errors about files not found. You can safely ignore them.

Now you update all packages

```
administrator@ltsp-server:~/$ sudo apt-get upgrade
```

In case you updated your *kernel* you may need to reboot. This would be indicated by a *red session manager*



It may be a *good idea* to now install the ltsp documentation using

```
administrator@ltsp-server:~/$ sudo apt-get install ltsp-docs
```

Build & Maintain a local Mirror

Creating a ubuntu mirror is easy. All it requires is a big harddisk (> 500 Gigabyte recommended) and a big network pipe (> 1 megabit/sec) and some time (about 4 days with 1megabit/sec).

All you have to do is to copy the files from a official ubuntu mirror to the local disk. One could simply use wget and let it run for 4 days, however, as the ubuntu mirrors are updated on a daily basis, this would lead to a inconsistent copy where some files (from the start) are older than others (those at the end). This may lead to nonworking local Repositories and is hard to update.

The solution to the problem is called rsync. rsync is a tool which checks for differences and copies changed files and folders over the network. As it does compare the files, it only copies things actually changed. If the transfer takes long time, it can be run again until all files are synchronized.

Rsync still will have one problem. All Repositories have a *index file* where all available packages are listed. If this file is copied before the package files, the Repository can not be used until all packages are copied too. The solution is simply to copy those index files **after** all package file synchronization is done. As typing all these rsync commands is a call for errors, there exist official scripts to do this. Using these will ensure, that the local mirror can always be used, even if a synchronization process was interrupted in the middle.

A secondary mirror is used to host Ubuntu Release Images. This holds installation and live disk images for Ubuntu and some variants of it, in various formats for *cd*, *dvd*, *netboot*. The images from there can be used to experiment with variations of Ubuntu and the live images can help administrators with debugging client-related problems.

Making a *powerpc* mirror is not as straightforward as *x86*. The server hosting the *powerpc* packages is too hosting other architectures like *sparc* or *arm*. As this would be way too much, we use a small script called debmirror (from the debian project) to mirror only a part of the server.

The same applies to the partners repository. The partners repository may not be legal to mirror in your country.

Prepare harddisk

This is a **one-time task!** If your hard disk already contains a mirror, created with this procedure, move on to the next section about updates.

1. First we need to prepare the external Harddisk.
   - Use the DiskUtil tool to format your partition as ext4 and name it *ubuntu-ltsp*. The name is important as it is used in the scripts! Using the same name for multiple disks allow easy exchange of them.
   - You should use the disk exclusively for the mirror. Using it for other things you risk data loss by mistake. Copy 400 gigabytes of Data is a big task, so you do not want loose everything by mistake.
2. your disk should now be mounted at */media/ubuntu-ltsp*
3. create a basic directory structure
   - a folder called script
   - a folder called mirror
   - a folder called keyrings
   - *optional* a folder called powerpc
   - *optional* a folder called partners
   - *optional* a folder called releases
   - *optional* a folder called documentation

```
administrator@ltsp-server:~/$ cd /media/ubuntu-ltsp
```

```
$ sudo mkdir script mirror keyrings powerpc partners releases documentation
```

4. get the rsync scripts from *Appendix -> Scripts*
   - copy into scripts/
   - customize the mirror urls.
5. run the sync script

```
administrator@ltsp-server:~/$ sudo /media/ubuntu-ltsp/script/update.sh
```

   add *-v* to see what is going on
   - Wait....

The rsync process can be interrupted anytime by hit *ctrl+c* and resume with

```
administrator@ltsp-server:~/$ sudo /media/ubuntu-ltsp/script/update.sh
```

Update script

With the setup above, updating the hard disk is as easy as run the script again

```
administrator@ltsp-server:~/$ sudo /media/ubuntu-ltsp/script/update.sh
```

Again, interrupt the process with *ctrl+c* if required. If the process does not finish, no *index files* are overwritten and nothing was deleted. This ensures the disk is always in a usable state.

Updating the mirror on other systems

Warning: *The mirror is formated for Linux. Using it with non-linux systems may lead to trouble including* ***data loss on mirror****. Be carefull!*

Any modern Linux using a recent Distribution is capable to update the main-mirror. If you plug in the Disk, it should be mounted at */media/ubuntu-ltsp*. Now you can just run the update script.

```
admin@somehost:~/$ sudo /media/ubuntu-ltsp/script/update.sh
```

You must be able to run this command as root using the *sudo* command.

LTSP Server

In order to turn a computer into a thin client, we need to run a mini version of GNU/Linux on the workstation. It needs to boot this mini version of GNU/Linux over the network, since it probably won't have a hard drive on it's own. This mini GNU/Linux installation needs to live somewhere, and the best place for it is on the server. This scaled-down GNU/Linux installation, customized so that it's efficient to boot over the network, is called a **chroot** environment. You can have several of them, based upon several different CPU architectures. They'll normally live under /opt/ltsp on the server, with sub directories for each of the architectures. For instance, if you have a lab full of old Power PC Macs, and older PC's, you'll have an /opt/ltsp/ppc and an /opt/ltsp/ i386 directory on the server.

This is the LTSP project's preferred area to store the chroot. The reason why it is called a chroot environment is that to install it, the GNU/Linux command chroot is called to actually set the installation root to /opt/ltsp/<arch>. From there, a scaled-down version of the distribution is installed. What this means is that for you to manage the chroot, performing such things as updates, all you need to do is use the chroot command to change the root of your installation. Then you can use all your tools like you normally would.

lts.conf

Some simple and often used customization can be done using the config file

```
/var/lib/tftpboot/ltsp/i386/lts.conf
```

You do not need any *chroot* to edit this and changes should take place immediately. The file may not exist and can be safely created.

This file is used by all clients. It can contain global settings for all clients. For more information install the package *ltsp-docs* and see

```
man lts.conf
```

To add client-specific settings, you specify the client by its *MAC* address. The *MAC* is a globaly unique identifier (looking similar to *IPv6* addresses) for your *network device*.

If your client is running and you know its *IP* address (displayed on the bottom-right on the login screen), you use

```
administrator@ltsp-server:~/$ ping -c1 client ip
...
administrator@ltsp-server:~/$ arp -an
? (10.55.66.20) at e0:46:9a:4e:7f:a9 [ether] on eth0
? (10.55.66.21) at 00:11:24:40:f5:62 [ether] on eth0
? (10.55.66.22) at <incomplete> on eth0
...
```

This tells you the mac of the clients with *IP* ending in 20 and 21. The client with *IP* 22 was tried to reach, but did not answer (usually this tells you, that the client does not have a valid network connection)

lts.conf is usually used to enable *printers*, load *kernel-modules* or specify graphics settings in case the auto detection fails. See the Example below.

```
[00:11:25:93:CF:00]
    PRINTER_0_dEVICE=/dev/usblp0
```

Setup dhcp for the boot process

The clients will use dhcp to get network information. dhcp can additionaly offer a *boot-server* to start a *pxe boot process*

```
administrator@ltsp-server:~/$ sudo nano /etc/ltsp/dhcp.conf
```

Edit the subnet part to contain

```
subnet 10.55.66.0 netmask 255.255.255.0 {
  range 10.55.66.20 10.55.66.200;
  option domain-name-servers 10.55.66.1;
  option routers 10.55.66.1;
  next-server 10.55.66.1;
  server-name "10.55.66.1";

  if substring (option vendor-class-identifier, 0, 9) = "AAPLBSDPC" {
    filename "/ltsp/powerpc/yaboot-fix";
    option root-path "/opt/ltsp/powerpc";
    option vendor-class-identifier "AAPLBSDPC";
    option vendor-encapsulated-options 01:01:02:08:04:01:00:00:01:82:05:69:6d:61:63:34;
  } else {
    filename    "/ltsp/i386/pxelinux.0";
    option root-path "/opt/ltsp/i386";
    }
```

}

Client Setup

X86 Client

Setup the Build Environment

By default, the ltsp build scripts will build a system for the same architecture as the host. As most ltsp-servers are 64 Bit capabable, you will get *amd64* clients. Most Client Systems are usually still 32 Bit. Fortunately, 64Bit computers can run 32 Bit Linux Systems (this is not true for powerpc!). Therefore it is advisable to create 32Bit images for all clients. Specify these options in /etc/ltsp/ltsp-build-client.conf (*you may need create the file*)

```
ARCH=i386
```

If you want use a local attached Repository to retrieve the system (else you need internet connection), add the mirror to the ltsp-build-client.conf

```
MIRROR=file:///media/ubuntu-ltsp/mirror/
```

A minor annoyance in ltsp-update-image would now include the full mirror into the *chroot* resulting in a 500 Gigabyte image. to prevent this, edit /etc/ltsp/ltsp-update-image and add

```
ARCH=i386
EX_DIRS=/media
```

Warning: Do **not** delete the directory */opt/ltsp/i386* while your mirror is mounted!

Build the client

As mentioned above, if your server is x86 you should build the client image on the server itself.
the ltsp installation offers a script to do this

```
administrator@ltsp-server:~/$ sudo ltsp-build-client
```

This script will

- create a new directory /opt/ltsp/<arch>
- install a minimal linux system into this directory
- **chroot** into this directory
- install packages required for ltsp
- leave the **chroot**
- create a boot image by packing everything into one file in /opt/ltsp/images/<arch>.img

PowerPC Client

chrooting as described above is a nice tool to build the client images. However if your server is a Intel x86 based system, and you want build a chroot for ppc (some apple computers), this does not work. If you chroot into the directory you use the applications and libraries inside this directory. As these are for ppc and you are on a x86 host, this can not work. Therefore you need to boot an ubuntu livecd for ppc on one of your ppc clients, build the chroot from there and copy it over to your x86 based server. You can use our prebuild ppc chroot image from the offline mirror hard disc and skip the *Live Environment* step. The fixed yaboot binary is on the disc as well.

Live Environment

Start one of your ppc clients from the ubuntu livecd for ppc and install the ltsp-server-standalone package. To boot from the CD, press *C* during startup. At the yaboot prompt type *live*. If this doesn't work, see the section about the *Open Firmware*.

```
sudo apt-get upgrade
sudo apt-get update
sudo apt-get install ltsp-server-standalone
```

Mount an usbstick on */opt/ltsp*. The stick needs to be formated ext4. You can reformat the filesystem with *System -> Administation -> Disk Utility*.

```
sudo umount /media/NAME_OF_STICK
sudo mount /dev/sda1 /opt/ltsp/
```

Typically to generate a chroot you would run *sudo ltsp-build-client*. However in this instance, things will be slightly different. Within a PowerPC environment, the mirrors that PowerPC needs to complete the chroot build are unavailable within the default command, so they must be specified manually by utilizing the --mirror tag with the necessary URLs.

```
sudo ltsp-build-client --mirror=http://ports.ubuntu.com/ubuntu-ports \
  --security-mirror=http://ports.ubuntu.com/ubuntu-ports \
  --updates-mirror=http://ports.ubuntu.com/ubuntu-ports
```

or if a offline mirror is available

```
sudo ltsp-build-client --mirror=file:///media/ubuntu-ltsp/ports \
--security-mirror=file:///media/ubuntu-ltsp/ports \
--updates-mirror=file:///media/ubuntu-ltsp/ports
```

You need to specify the portnumber while building the image because the default one may be allready used on your ltsp-server for another arch.

```
sudo ltsp-update-image --port 2200
```

If the built is complete, unmount the usbstick and remove the livecd from your ppc client.

```
sudo umount /opt/ltsp/
```

Yaboot

There is also a known issue with the default yaboot file. Yaboot is utilized to boot the PowerPC clients into the LTSP environment. The netboot images for powerpc and powerpc64 have exceeded 6MB uncompressed for either the kernel (powerpc64) or the ramdisk (powerpc). Yaboot is unable to do a netboot with those images since they exceed a limitation within yaboot that limits the uncompressed images to 6MB each

An updated binary that works is available on the bug report on the Ubuntu bugtracker [1].

Download Yaboot 1.3.13a-1ubuntu4 patched for >6MB initrd [2] and save it as *yaboot-fix*.

Copy to the LTSP Server

Copy the *powerpc* directory from the usb stick to */opt/ltsp/* on your server and run

```
sudo ltsp-update-sshkeys
sudo ltsp-update-image --arch powerpc
```

to regenerate the ssh keys and the image. There may be *errors and warnings* because the script fails to enter the chroot. **Just ignore them**.

Copy the *yaboot-fix* binary to */var/lib/tftpboot/ltsp/powerpc*

Make sure that the line

```
filename "/ltsp/powerpc/yaboot-fix
```

is present in your */etc/ltsp/dhcp.conf*

That's it. To boot your ppc client over the network, press *N* while booting your client. In case there are problems with the networkboot, see the next section about the *open firmware*.

Open Firmware

Password

If the open firmware password ist set, you are not allowed to boot over the network or from a cdrom. To reset the password, boot into the administrator account of your osx install and open a terminal.

Print NVRAM settings with:

```
sudo nvram -p
```

To set a firmware password:

```
sudo nvram security-password=mypass
sudo nvram security-mode=command
```

To reset a firmware password:

```
sudo nvram -d security-password
sudo nvram security-mode=none
```

Boot

To boot into open firmware, press the following keys at startup:

```
Command + Option + O + F
```

To boot explicitly via ethernet:

```
boot enet:0
```

Or, if your TFTP server is different from your DHCP server, you'll need to specify its IP address:

```
boot enet:10.55.66.1,yaboot
```

Alternatively, set your boot device (semi-permanently) to enet:0:

```
setenv boot-device enet:0
boot
```

If the configuration on the server is correct, it should be enough to press *N* during startup to boot over the network.

Further Information

- UbuntuLTSP PowerPC [3]
- yaboot cannot handle kernels and/or initrds >6MB uncompressed for netbooting [4]

References

[1] https://bugs.launchpad.net/ubuntu/+source/yaboot/+bug/26426/
[2] https://bugs.launchpad.net/ubuntu/+source/yaboot/+bug/26426/+attachment/420679/+files/yaboot
[3] https://help.ubuntu.com/community/UbuntuLTSP/PowerPC
[4] https://bugs.launchpad.net/ubuntu/+source/yaboot/+bug/26426

Customize Client Environment

Chroot

The default installation will work with most Hardware, but *most* is not **all**. If your clients do not boot or show hardware related crashes or gltiches, you may need customize your chroot.
After successfully creation of the client-image, the ltsp-build-client-script will leave the files inside */opt/ltsp/<arch>/* you now can change things in there.
You may ommit the mounting of dev, sys or proc depending on the command you run

```
administrator@ltsp-server:~/$ sudo mount /dev -o bind /opt/arch/dev
administrator@ltsp-server:~/$ sudo chroot /opt/ltsp/arch /bin/bash
root@ltsp-server:/$ mount none -t proc /proc
root@ltsp-server:/$ mount none -t sysfs /sys
root@ltsp-server:/$ do some edits/installations/etc
root@ltsp-server:/$ umount /proc /sys
root@ltsp-server:$ exit
administrator@ltsp-server:~/$ sudo umount /opt/arch/dev
```

now you need update the image file to catch up the changes.

```
administrator@ltsp-server:~/$ ltsp-update-image --excludedir=/media
```

if you did update kernels you additionally need

```
administrator@ltsp-server:~/$ ltsp-update.kernel
```

Now your clients boot with the new image.

See below or *Appendix: Troubleshooting* for some examples.

Example: setting the default Language

Setting the default language for your system is done using *dpkg-reconfigure*. This application does not require any mountpoints, so the whole stuff from above boils down to:

```
administrator@ltsp-server:~/$ sudo chroot /opt/ltsp/i386 /bin/bash
root@ltsp-server:/$ dpkg-reconfigure console-setup
root@ltsp-server:/$ exit
```

Or even faster

```
administrator@ltsp-server:~/$ sudo chroot /opt/ltsp/i386 dpkg-reconfigure console-setup
```

Don't forget to update the image administrator@ltsp-server:~/$ ltsp-update-image --excludedir=/media

Administration

Using The Terminal

Why use the command-line?

"Under Linux there are GUIs (graphical user interfaces), where you can point and click and drag, and hopefully get work done without first reading lots of documentation. The traditional Unix environment is a CLI (command line interface), where you type commands to tell the computer what to do. That is faster and more powerful, but requires finding out what the commands are." -- from **man intro(1)**

There are many varieties of Linux, but almost all of them use similar commands that can be entered from a command-line interface terminal.

There are also many graphical user interfaces (GUIs), but each of them works differently and there is little standardization between them. Experienced users who work with many different Linux distributions therefore find it easier to learn commands that can be used in all varieties of Ubuntu and, indeed, in other Linux distributions as well.

For the novice, commands-line interface commands can appear daunting:

```
sudo gobbledegook blah_blah -w -t -h --long-switch aWkward/ComBinationOf/mixedCase/underscores_strokes/and.dots
```

However, it is important to note that even experienced users often cut and paste commands (from a guide or manual) into the command-line terminal; they do not memorize them.

It is important, of course, to know how to use the command-line terminal - and anyone who can manage typing, backspacing, and cutting and pasting can manage the command-line terminal (it is not more difficult than that).

This page will outline a few crafty shortcuts which can make using a command-line interface easier.

Starting a Terminal

In Gnome

Gnome is the Classic Desktop Environment for Ubuntu 11.04 (Natty) and is the default DE in earlier releases, such as Ubuntu 10.04 LTS (Lucid).

Applications menu -> Accessories -> Terminal.

In Unity

Unity is the default Desktop Environment used in 11.04. Where systems are not ready for Unity they revert to Gnome which is also used in previous releases such as Ubuntu 10.04 LTS (Lucid), see previous sub-section.

The easiest way to open the Terminal is to use the 'search' function on the dash. Or you can click on the 'More Apps' button, click on the 'See more results' by the installed section, and find it in that list of applications. A third way, available after you click on the 'More Apps' button, is to go to the search bar, and see that the far right end of it says 'All Applications'. You then click on that, and you'll see the full list. Then you can go to Accessories > Terminal after that. So, the methods in Unity are:

Dash -> Search for Terminal

Dash -> More Apps -> 'See More Results' -> Terminal

Dash -> More Apps -> Accessories -> Terminal

Keyboard Shortcut: Ctl + Alt + T

From another computer

*SSH (**S**ecure **SH**ell)* is a network protocol which provides a replacement for insecure remote login and command execution facilities, such as telnet, rlogin and rsh. SSH encrypts traffic in both directions, preventing traffic sniffing and password theft. SSH also offers several additional useful features:

- Compression: traffic may be optionally compressed at the stream level.
- Public key authentication: optionally replacing password authentication.
- Authentication of the server: making ”man-in-the-middle” attack more difficult
- Port forwarding: arbitrary TCP sessions can be forwarded over an SSH connection.
- X11 forwarding: SSH can forward your X11 sessions too.
- File transfer: the SSH protocol family includes two file transfer protocols.

This makes *SSH* the perfect tool for remote server administration. You can move the noisy server into a secure, cool location in the basement. For Windows System, a programm called *PuTty* exists, offering a ssh to connect to servers. By default SSH offers you a terminal as you get on the local system.

As *ssh* does not require any graphics, you can connect to a system not having *'any* graphics or input devices. This is how most Server System in remote Data-centers are administrated. Make yourself familiar with *ssh* and you can administrate your System from anywhere in the world require only a computer and internet!

If your Graphical Environment is unuseable

Linux Systems contain a Feature called *virtual console*. *VT*s can be access by press <ctrl+alt+Fn> where *Fn* is one of the F-Keys (F1,F2...).
By default on *VT* 1-4 you get a terminal login window. On *VT* 7 you usually find your graphical environment. The other numbers may be disabled or offer more terminals.

This is most useful if your Systems graphical environment is not usable.

Syntax Convention

Terminal commands and their output will be displayed in boxes. For example

```
administrator@ltsp-server:~/dir$ ls -lisa
total 8
5767383 4 drwxr-xr-x  2 administrator administrator 4096 Jan 24 22:13 .
5767169 4 drwxr-xr-x 44 administrator administrator 4096 Jan 24 22:13 ..
5767703 0 -rw-r--r--  1 administrator administrator    0 Jan 24 22:13 .hidden
5767415 0 -rw-r--r--  1 administrator administrator    0 Jan 24 22:13 file1.txt
5767687 0 -rw-r--r--  1 administrator administrator    0 Jan 24 22:13 file2.txt
```

In the example above, you first see the *prompt*

```
administrator@ltsp-server:~/dir$
```

This shows you who you are (administrator), what this systems hostname is (ltsp-server) and where you are (~/dir is a folder called dir in your home directory).
The text after the *prompt* is a command to type

```
ls -lisa
```

In this case the command displays a list of all files in the current directory including the total number of files. The lasting part is the output of the command.

Sometimes you need type commands including some variable string (a username not known to the writer of this document for example).
Those variables will be typed in **bold** and you should replace them by the proper value

```
administrator@ltsp-server:~/$ echo "USERNAME"
```

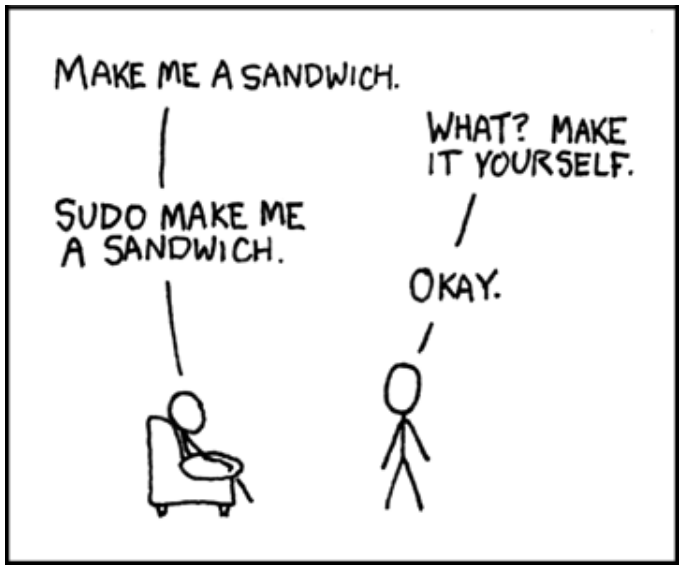
You will type something like

```
administrator@ltsp-server:~/$ echo "jonny"
```

Commands

sudo: Executing Commands with Elevated Privileges

- Most of the following commands will need to be prefaced with the **sudo** command. This elevates privileges to the root-user administrative level temporarily, which is necessary when working with directories or files not owned by your user account. When using sudo you will be prompted for your password. Only users with sudo (administrative) privileges will be able to use this command. In the installation setup decribed before, the administrative user was called *administrator*.



Famous xkcd [1] comic strip about the power of sudo

File & Directory Commands

- The tilde (~) symbol stands for your home directory. If you are *user*, then the tilde (~) stands for /home/*user*
- **pwd**: The **pwd** command will allow you to know in which directory you're located (**pwd** stands for "print working directory"). Example: **"pwd"** in the Desktop directory will show "~/Desktop". Note that the Gnome Terminal also displays this information in the title bar of its window. A useful gnemonic is "present working directory."
- **ls**: The **ls** command will show you ('list') the files in your current directory. Used with certain options, you can see sizes of files, when files were made, and permissions of files. Example: **"ls ~"** will show you the files that are in your home directory.
- **cd**: The **cd** command will allow you to change directories. When you open a terminal you will be in your home directory. To move around the file system you will use **cd**. Examples:
  - To navigate into the root directory, use **"cd /"**
  - To navigate to your home directory, use **"cd"** or **"cd ~"**

  - To navigate up one directory level, use **"cd .."**
  - To navigate to the previous directory (or back), use **"cd -"**
  - To navigate through multiple levels of directory at once, specify the full directory path that you want to go to. For example, use, **"cd /var/www"** to go directly to the /www subdirectory of /var/. As another example, **"cd ~/Desktop"** will move you to the Desktop subdirectory inside your home directory.
- **cp**: The **cp** command will make a copy of a file for you. Example: **"cp file foo"** will make an exact copy of "file" and name it "foo", but the file "file" will still be there. If you are copying a directory, you must use **"cp -r directory foo"** (copy recursively). (To understand what "recursively" means, think of it this way: to copy the directory and all its files and subdirectories and all their files and subdirectories of the subdirectories and all their files, and on and on, "recursively")
- **mv**: The **mv** command will move a file to a different location or will rename a file. Examples are as follows: **"mv file foo"** will rename the file "file" to "foo". **"mv foo ~/Desktop"** will move the file "foo" to your Desktop directory but will not rename it. You must specify a new file name to rename a file.
  - To save on typing, you can substitute '~' in place of the home directory.
  - Note that if you are using **mv** with **sudo** you can use the ~ shortcut, because the terminal expands the ~ to your home directory. However, when you open a root shell with **sudo -i** or **sudo -s**, ~ will refer to the root account's home directory, not your own.
- **rm**: Use this command to remove or delete a file in your directory. **warning:** files deleted with rm are irrecoverably deleted! you can not undo this command as you may be used to in the desktop.
- **rmdir**: The **rmdir** command will delete an *empty* directory. To delete a directory and all of its contents recursively, use **rm -r** instead.
- **mkdir**: The **mkdir** command will allow you to create directories. Example: **"mkdir music"** will create a directory called "music".
- **man**: The **man** command is used to show you the manual of other commands. Try **"man man"** to get the man page for **man** itself. See the "**Man** & Getting Help" section down the page for more information.

System Information Commands

- **df**: The **df** command displays filesystem disk space usage for all mounted partitions. "**df -h**" is probably the most useful - it uses megabytes (M) and gigabytes (G) instead of blocks to report. (**-h** means "human-readable")
- **du**: The **du** command displays the disk usage for a directory. It can either display the space used for all subdirectories or the total for the directory you run it on.
  - **-s** means "Summary" and **-h** means "Human Readable"

```
user@users-desktop:~$ du /media/floppy
1032    /media/floppy/files
1036    /media/floppy/
user@users-desktop:~$ du -sh /media/floppy
1.1M    /media/floppy/
```

- **free**: The **free** command displays the amount of free and used memory in the system. **"free -m"** will give the information using megabytes, which is probably most useful for current computers.
- **top**: The **top** command displays information on your Linux system, running processes and system resources, including CPU, RAM & swap usage and total number of tasks being run. To exit **top**, press **"q"**.
- **uname -a**: The **uname** command with the **-a** option prints all system information, including machine name, kernel name & version, and a few other details. Most useful for checking which kernel you're using.
- **lsb_release -a**: The **lsb_release** command with the **-a** option prints version information for the Linux release you're running, for example:

```
user@computer:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 11.10
Release:        11.10
Codename:       oneiric
```

- **ip addr** reports on your system's network interfaces.

Options

The default behaviour for a command may usually be modified by adding a ***--option*** to the command. The **ls** command for example has an **-s** option so that **"ls -s"** will include file sizes in the listing. There is also a **-h** option to get those sizes in a "human readable" format.

Options can be grouped in clusters so **"ls -sh"** is exactly the same command as **"ls -s -h"**. Most options have a long version, prefixed with two dashes instead of one, so even **"ls --size --human-readable"** is the same command.

"Man" and getting help

man ***command***, **info** ***command*** and ***command*** **--help** are the most important tools at the command line.

Nearly every command and application in Linux will have a man (manual) file, so finding them is as simple as typing **"man "command""** to bring up a longer manual entry for the specified command. For example, **"man mv"** will bring up the **mv** (Move) manual.

Move up and down the man file with the arrow keys, and quit back to the command prompt with **"q"**.

"man man" will bring up the manual entry for the **man** command, which is a good place to start!

"man intro" is especially useful - it displays the "Introduction to user commands" which is a well-written, fairly brief introduction to the Linux command line.

There are also **info** pages, which are generally more in-depth than **man** pages. Try **"info info"** for the introduction to info pages.

Some software developers prefer **info** to **man** (for instance, GNU developers), so if you find a very widely used command or app that doesn't have a **man** page, it's worth checking for an **info** page.

Virtually all commands understand the **-h** (or **--help**) option which will produce a short usage description of the command and it's options, then exit back to the command prompt. Try **"man -h"** or **"man --help"** to see this in action.

Caveat: It's possible (but rare) that a program doesn't understand the -h option to mean help. For this reason, check for a ***man*** *or* ***info*** *page first, and try the long option --help before -h.*

Searching for man files

If you aren't sure which command or application you need to use, you can try searching the man files.

- **man -k** ***foo*** will search the man files for *foo*. Try **"man -k nautilus"** to see how this works.
  - Note that this is the same as doing **apropos** ***command***.
- **man -f** ***foo*** searches only the titles of your system's man files. Try **"man -f gnome"**, for example.
  - Note that this is the same as doing **whatis** ***command***.

Other Useful Things

Pasting in commands

Often, you will be referred to instructions that require commands to be pasted into the terminal. You might be wondering why the text you've copied from a web page using **ctrl+C** won't paste in with **ctrl+V**. Surely you don't have to type in all those nasty commands and filenames? Relax. **ctrl+shift+V** pastes into a Gnome terminal; you can also do Middle Button Click on your mouse (both buttons simultaneously on a two-button mouse) or Right Click and select *Paste* from the menu. However, if you want to avoid the mouse and yet paste it, use "Shift+Insert", to paste the command. If you have to copy it from another terminal / webpage, you can use "Ctrl+Insert" to copy.

Save on typing

Up Arrow or **ctrl+p**	Scrolls through the commands you've entered previously.
Down Arrow or **ctrl+n**	Takes you back to a more recent command.
Enter	When you have the command you want.
tab	A very useful feature. It autocompletes any commands or filenames, if there's only one option, or else gives you a list of options.
ctrl+r	Searches for commands you've already typed. When you have entered a very long, complex command and need to repeat it, using this key combination and then typing a portion of the command will search through your command history. When you find it, simply press **Enter**.
History	less **for a scrollable list.**

Change the text

The mouse won't work. Use the Left/Right arrow keys to move around the line.

When the cursor is where you want it in the line, typing *inserts* text - ie it doesn't overtype what's already there.

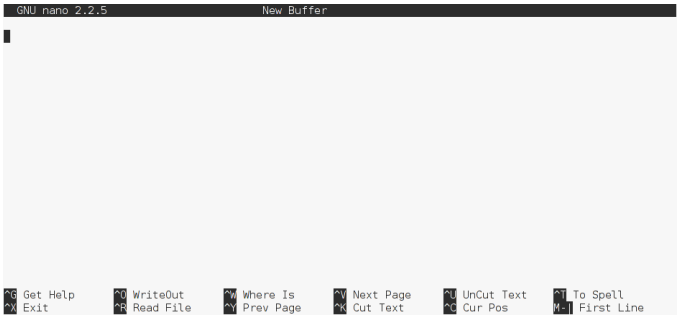
ctrl+a or **Home**	Moves the cursor to the *start* of a line.
ctrl+e or **End**	Moves the cursor to the ***e**nd* of a line.
ctrl+b	Moves to the **b**eginning of the previous or current word.
ctrl+k	Deletes from the current cursor position to the end of the line.
ctrl+u	Deletes the whole of the current line.
ctrl+w	Deletes the word before the cursor.

File editing

there are multiple editors available on the command line. As suggests by the strip above, the easiest to use is nano.

```
administrator@ltsp-server:~/dir$ nano file_to_edit
```

Use the arrow keys to navigate, and just type your text.



nano - a easy to use command line editor

On the bottom you see a list of available commands. Use them by hit ctrl together with the key. Fro example, to Exit when done you would hit ctrl+x. nano will ask you if you want save the file, hit Y to save or N to discard the editing (again, all available commands are listed on the bottom).

Further Information

- UsingTheTerminal [2]

References

[1] http://www.xkcd.com
[2] https://help.ubuntu.com/community/UsingTheTerminal

Add Users

Information

When Ubuntu is initially installed, a single user account was created. It is possible to create additional user logins on your system.

Options

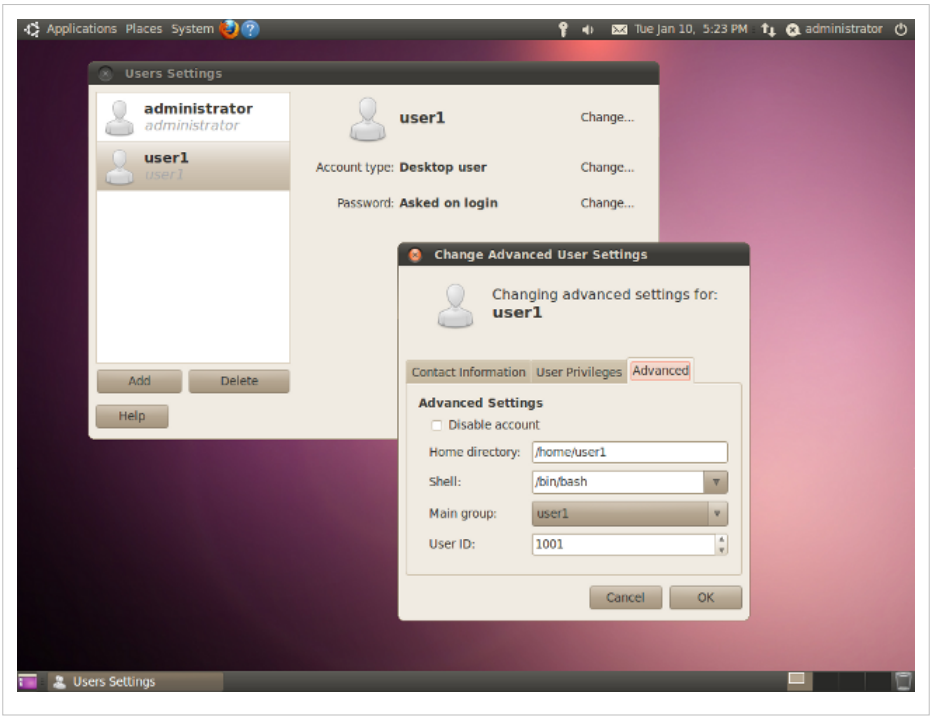
There is more than one way to add a user, however this wiki page will briefly discuss the easiest and most common ways. The two ways shown are:

1. Graphical
2. Command-line

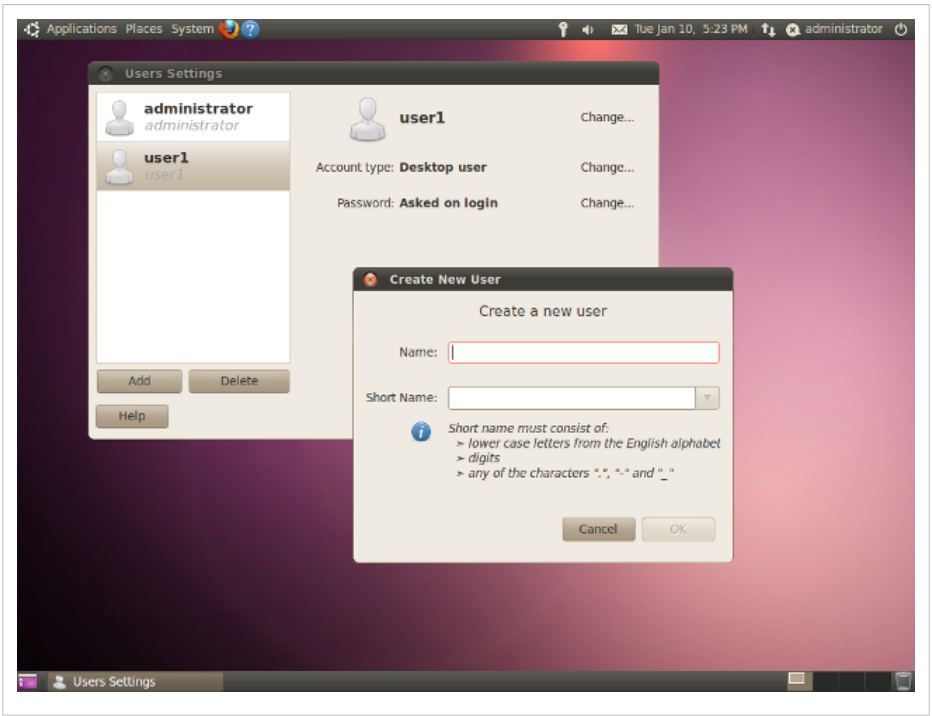
Graphical is the easiest solution, while the command-line offers the possibilities of multiple tasks at one time.

Graphical Ubuntu

Location: *System > Administration > Users and Groups*



Press + *Add User*, this will open up the *User Account Editor*. The minimum requirements are Username and password. For the Username, do **not** use spaces, and **do** use ASCII characters.



Command-line

To add a user you must use the sudo command (for an explanation of what that means, see the RootSudo page). Here are the commands:

To add a user. **NOTE:** do not use the useradd command.

```
sudo adduser <username>
```

To see the options for adding a user try the man command.

```
man adduser
```

Here is a useful example of the useradd command. Why use useradd? It gives a few more options for special cases. To add a user, give her a standard home directory in the /home folder and specify the shell she accesses by default do this:

```
sudo useradd username -m -s /bin/bash
sudo passwd username
```

Groups

You might also wish to create a new group for your users.

```
sudo addgroup <groupname>
```

To add a new user to a existing group you would do this:

```
sudo adduser <username> audio
```

To add an existing user to an existing group you can use the same command:

```
sudo adduser <username> <groupname>
```

or

```
sudo usermod -aG <groupname> <username>
```

Further Information

- AddUsersHowto [1]

References

[1] https://help.ubuntu.com/community/AddUsersHowto

Package Management

Synaptic is a graphical front-end to apt [1], the package management system in Ubuntu. It combines the point-and-click simplicity of the graphical user interface with the power of the *apt-get* command line tool. You can install, remove, configure, or upgrade software packages, browse, sort and search the list of available software packages, manage repositories, or upgrade the whole system. You can queue up a number of actions before you execute them. Synaptic will inform you about dependencies (additional packages required by the software package you have chosen) as well as conflicts with other packages that are already installed on your system.

Note for 11.10 release and above

Synaptic is no longer installed by default in Ubuntu 11.10, however it is still useful in some situations.

You can install it from the command line with

```
administrator@ltsp-server:~/$ sudo apt-get install synaptic
```

Getting Started

To launch Synaptic, choose **System > Administration >** "**Synaptic Package Manager**"

Or if you are using the Unity interface, open the dash and search for **synaptic**.

The main window is divided into three sections: a package browser on the left, the package list on the upper right, and package details on the lower right. The status bar shows you the system state at a glance.

And if you would like to see more details about a package, use a mouse right-click on a package and choose **Properties**

Browsing the package database

To browse the (very large) list of available packages by category, section, package status, custom filters, or recent searches. Click on the corresponding button at the bottom of the left window pane. You can also create your own filters.

To search for packages by name or description, click on the **Search** button in the toolbar or use the "Quick search" field in the toolbar.

You can examine a number of package details in the lower right window pane such as its size, its dependencies, recommended or suggested additional packages, and a short description.

Repositories

Repositories are like shops (currently almost always free), warehouses or archives full of almost all the software, drivers, codecs, libraries and other packages that are available for the release of Ubuntu (or whichever distro [2]) you are using.

It is up to the package project's team to get their package into the repositories for the various releases of the various distros. Also it is up to the team developing a release to make sure that the repositories have plenty of the right sort of packages to meet people's needs. Think of a shop. Companies need to get their product onto shelves and shops need to show they have plenty to sell. This saves individual users from having to identify the 'correct' website for a product and assess whether or not the site has been compromised and whether they are getting a genuine product or something stuffed full of malware.

The standard repositories are all free as are the Medibuntu repositories and most others that are easily found and added. Theoretically there is nothing stopping a games manufacturer (for example) setting up a one-off or monthly charge for accessing a specialist repository. The ethics and licensing agreements would have to be looked into if people or companies chose this route.

Adding or removing repositories is fairly easy.

Managing Repositories

Either from the top taskbar

System - Administrator - "Software Sources"

or from inside Synaptic using the **Settings** menu, select **Repositories**

The first tab of the pop-up is where Cd/Dvds can be added (or removed) as repositories to search. Usually the various online "repos" are already included so you might be installing a more recent version of something that is listed on the Cd/dvd and also in the online repos. The package manager will automatically choose the most recent by default although one of the other tabs in the pop-up can change the priorities, or in the **Preferences** pop-up from the **Setttings** menu.

As a front-end to apt, Synaptic uses the system-wide list of software repositories file located at

```
/etc/apt/sources.list
```

or from files ending in .list inside the directory

```
/etc/apt/sources.list.d
```

Updating the Package List

Click **Reload** or press **Ctrl + R** to make Synaptic aware of the latest updates and any changes to the list of repositories.

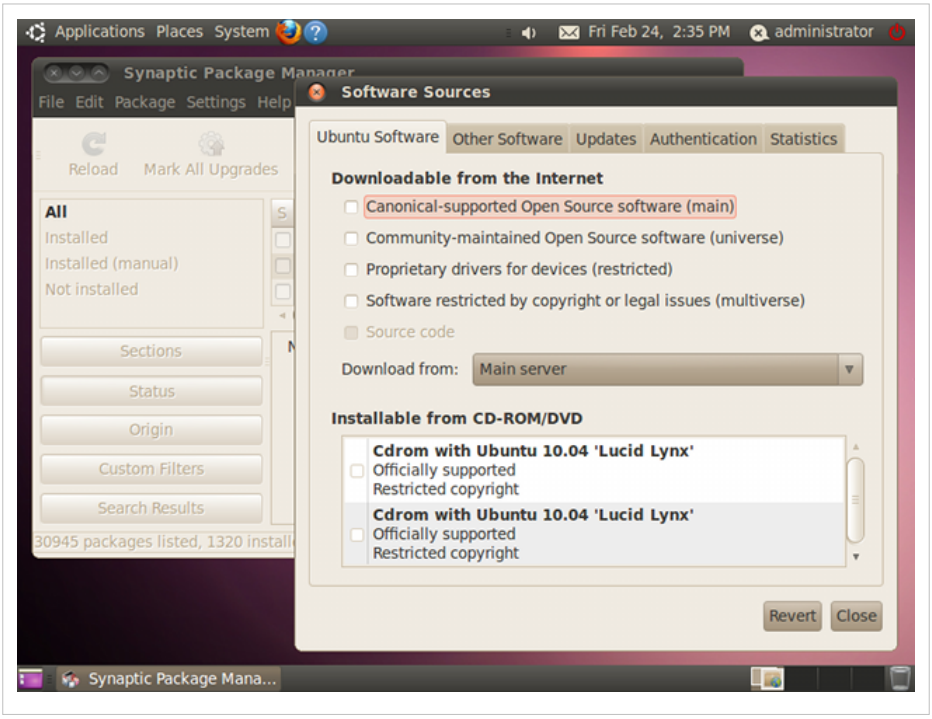
Although this is done automatically when you open Synaptic it can be a good idea to update the database from time to time while running Synaptic. The database is a list of packages on your system to keep track of installed software. All the package managers share the same list and the same list of repositories but each one presents the information slightly differently. The main time you will need to update the lists is when/if you have made changes to the list of software channels or if you have made changes to Synaptic's configuration.

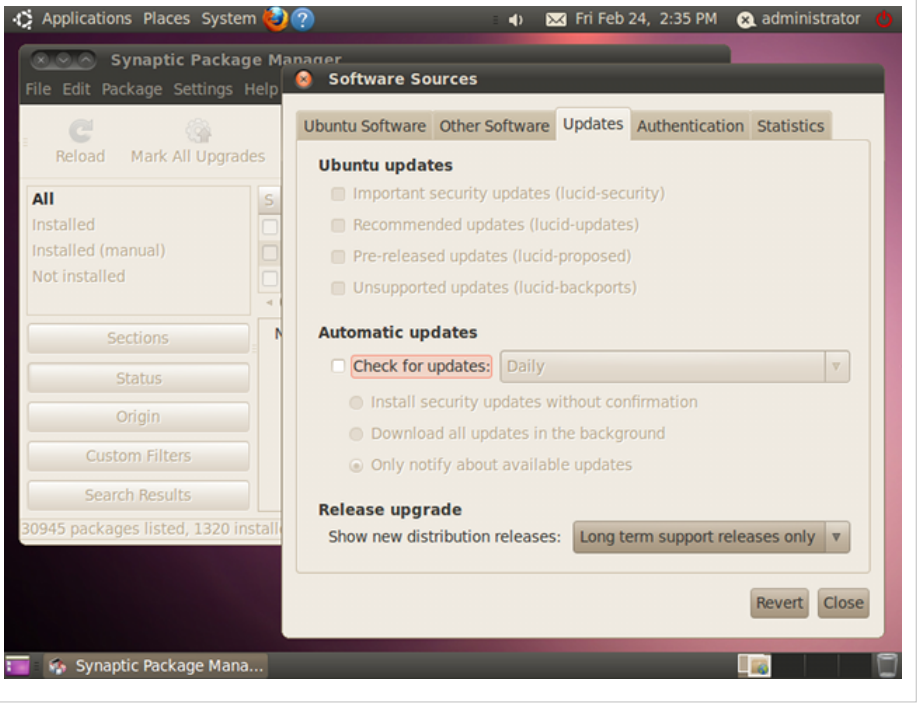
This can be done from command line with

```
administrator@ltsp-server:~/$ sudo apt-get update
```

Disable Internet Updates

To disable updates even with internet available open Synaptics and navigate to *Settings->Repositories*. Disable all repositories expect the local mirror (starting with *file://*) and disable autoupdates in the *Updates* pane.





Adding Packages

Mark the Chosen Packages

1. Right-click on each chosen package and choose **Mark for Installation** from the context menu, or press **Ctrl + I**. If the package requires the installation of another package, a dialog box appears. Synaptic will warn you if your choices conflict with packages that are already installed on your system. If this is the case, the dialog box will show you any packages that need to be removed. If you are **not sure** you don't need the package(s) that will be removed, make sure you look up its function and use before you apply the changes you've made.
2. Click **Mark** to allow the installation of the additional package(s).

Installing Packages

The mirror configured above needs be available. If you have internet connection the system will try use it if not configured otherwise. See above for details on disable network updating.
If you use a local mirror, the mirror must be available for installation of packages (you can still browse packages while the mirror is not present).

1. Once you are satisfied with your choices, click on **Apply** in the toolbar or press **Ctrl + P**.
2. A dialog box appears with a summary of changes that will be made.
3. Confirm the changes by clicking on **Apply**.

How to force the installation of a package version

To force the installation of a package different from the one chosen by Synaptic, do the following:

1. Click **Reload** or press **Ctrl + R** to make Synaptic aware of the latest updates.
2. Select the package. Choose **Force Version** from the **Package** menu.
3. Select the version you would like to use. To confirm your decision, click **Force**.
4. Click **Apply** on the toolbar or press **Ctrl + P**. A dialog appears with a summary of the changes that will be made to your system. To confirm, press **Apply**.

Remove or Uninstall Packages

To **remove** or **completely remove** a package

1. Right-click on the package and choose **Mark for Removal** from the context menu
2. Once you're satisfied with your choices, click on **Apply** in the toolbar or press **Ctrl + P**.
3. A dialog box appears with a summary of changes that will be made. Confirm the changes by clicking on **Apply**.

The **Mark for Complete Removal** option instructs Synaptic to remove any configuration files associated with the package as well.

To Upgrade a Package

Click **Reload** or press **Ctrl + R** to make Synaptic aware of the latest updates.

1. Right-click on the package and choose **Mark for Upgrade** from the context menu, or press **Ctrl + U**.
2. Once you're satisfied with your choices, click on **Apply** in the toolbar or press **Ctrl + P**.
3. A dialog box appears with a summary of changes that will be made. Confirm the changes by clicking on **Apply**.

How to keep your system up-to-date, including the Kernel

Synaptic provides two methods for upgrading your system:

- *Smart Upgrade (Dist-Upgrade) -- recommended* The smart upgrade method tries to resolve package conflicts. This includes installing additional dependencies (required packages) if needed and preferring packages with higher priority. Smart Upgrade has the same effect as the *apt-get dist-upgrade* tool on the command line.
- *Default Upgrade* The default upgrade method marks upgrades of installed packages only. If the new version of a package depends on not installed packages or coflicts with an already installed package, it will not be upgraded.

Dist-upgrade is the default upgrade method used by Synaptic. To change the upgrade method, choose **Preferences** from the **Settings** menu, then click on the **General** tab and adjust the **System upgrade** entry.

To upgrade your system with Synaptic:

1. Click **Reload** or press **Ctrl + R** to make Synaptic aware of the latest updates.
2. Click **Mark all Upgrades** or press **Ctrl + G**.
3. Click **Apply** on the toolbar or press **Ctrl + P**. A dialog appears with a summary of the changes that will be made to your system.
4. Click **Apply** to confirm the changes and go ahead with the upgrade.

Synaptic works hand-in-hand with **Update-Notifier** and **Update-Manager** to inform you about updates to the software installed on your system.

Update-Notifier displays an icon in the notification area when updates are available. It will display a pop-up message and an icon in the notification area on your GNOME desktop.

If you right click on the icon you see all the functions available, including what I think is a sensible default, Show Updates. This option runs another new program, **Upgrade-Manager**. The upgrade manager shows what packages the system wants to upgrade, which you can deselect from being upgraded if you wish.

To update your system, click on the notification icon, enter your password and click OK.

Once all the packages are downloaded, the dpkg system installs the packages in the hidden terminal. Questions that previously required interaction with the terminal now provide a Debconf interface using the GNOME frontend. The upgrade process never requires the user to access a terminal, which helps new users become comfortable with upgrading their system.

Ubuntu also makes basic APT configuration much easier for new users. It offers an interface to add or edit the entries in /etc/apt/sources.list which can be accessed through Synaptic or the preferences menu of the update-notifier. It shows the repositories that are currently listed, and provides an interface to edit them or add new repositories.

You can change the frequency of update checks in the **Software Preferences** panel. Choose **Repositories** from the **Settings** menu, then click on the **Internet Updates** tab and adjust the *Check for updates* entry.

Information Gathering

Locating software on your system

Synaptic can tell you about every file that belongs to a software package it knows about and show you where it is located on your system. Search the database for the software package you are interested in and select it in Synaptic's main window. Next, click on the *Installed Files* tab to see a list of all files and where they are.

View the Changelog of a Package

Applies to native Debian (**.deb**) packages only

You can view the changelog of a package with Synaptic. Select a package, then choose **Download Changelog** from the **Package** menu.

View History

You can easily keep track of all changes made to your system's software configuration. To view the history log file, choose *History* from the *File* menu.

The history log only shows changes made with Synaptic.

Troubleshooting

How to fix broken packages

'Broken packages' are packages that have unsatisfied dependencies. If broken packages are detected, Synaptic will not allow any further changes to the system until all broken packages have been fixed.

To fix broken packages:

1. Choose **Edit > Fix Broken Packages** from the menu.
2. Choose **Apply Marked Changes** from the **Edit** menu or press **Ctrl + P**.
3. Confirm the summary of changes and click **Apply**.

How to free disk space

Settings -> Preferences -> Files -> Delete downloaded packages after installation then click Delete Cached Package Files.

Broken Upgrade or Installation

What to do if an installation process fails and you find it is no longer possible to install or remove packages:

- Open a Terminal and type the following commands, pressing the Return or Enter key after each (you may have to type in your password):

```
sudo dpkg --configure -a
sudo apt-get install -f
```

Switching to Ubuntu from Fedora

Ubuntu uses apt-get instead of yum to find, download, and install packages and their dependencies.

Note that, unlike yum, apt-get is only for packages available in repositories - it cannot handle packages you have already downloaded. The dpkg command is used instead.

Below is a table of equivalent commands for package management on both Ubuntu/Debian and Red Hat/Fedora systems.

Task	Red Hat/Fedora	Ubuntu
Adding, Removing and Upgrading Packages		
Refresh list of available packages	Yum refreshes each time it's used	apt-get update
Install a package from a repository	yum install package_name	apt-get install package_name
Install a package file	yum install package.rpm rpm -i package.rpm	dpkg --install package.deb
Remove a package	rpm -e package_name	apt-get remove package_name
Check for package upgrades	yum check-update	apt-get -s upgrade apt-get -s dist-upgrade
Upgrade packages	yum update rpm -Uvh **args**	apt-get dist-upgrade
Upgrade the entire system	yum upgrade	apt-get dist-upgrade
Package Information		
Get information about an available package	yum search package_name	apt-cache search package_name
Show available packages	yum list available	apt-cache dumpavail
List all installed packages	yum list installed rpm -qa	dpkg --list
Get information about a package	yum info package_name	apt-cache show package_name
Get information about an installed package	rpm -qi package_name	dpkg --status package_name
List files in an installed package	rpm -ql package_name	dpkg --listfiles package_name
List documentation files in an installed package	rpm -qd package_name	-
List configuration files in an installed package	rpm -qc package_name	-
Show the packages a given package depends on	rpm -qR package_name	apt-cache depends
Show other packages that depend on a given package (reverse dependency)	rpm -q -whatrequires **args**	apt-cache rdepends
Package File Information		
Get information about a package file	rpm -qpi package.rpm	dpkg --info package.deb
List files in a package file	rpm -qpl package.rpm	dpkg --contents package.deb

List documentation files in a package file	rpm -qpd package.rpm	-
List configuration files in a package file	rpm -qpc package.rpm	-
Extract files in a package	-	dpkg-deb --extract package.deb dir-to-extract-to
Find package that installed a file	rpm -qf filename	dpkg --search filename
Find package that provides a particular file	yum provides filename	apt-file search filename
Misc. Packaging System Tools		
Show stats about the package cache	-	apt-cache stats
Verify all installed packages	rpm -Va	debsums
Remove packages from the local cache directory	yum clean packages	apt-get clean
Remove only obsolete packages from the local cache directory	-	apt-get autoclean
Remove header files from the local cache directory	yum clean headers	apt-file purge
General Packaging System Information		
Package file extension	*.rpm	*.deb
Repository location configuration	/etc/yum.conf	/etc/apt/sources.list

Further Information

- SynapticHowto [3]
- SwitchingToUbuntuFromLinuxRedHatEnterpriseLinuxAndFedora [4]

References

[1] http://www.debian.org/doc/user-manuals#apt-howto
[2] http://distrowatch.com
[3] https://help.ubuntu.com/community/SynapticHowto
[4] https://help.ubuntu.com/community/SwitchingToUbuntu/FromLinux/RedHatEnterpriseLinuxAndFedora

Software

Desktop Environment

Ubuntu

Ubuntu [1] is an entirely open source operating system built around the Linux kernel and based on the Debian Linux distribution. The Ubuntu community is gathered around the ideals enshrined in the Ubuntu Philosophy: that software should be available free of charge, that software tools should be usable by people in their local language and despite any disabilities, and that people should have the freedom to customize and alter their software in whatever way they see fit.

Ubuntu is sponsored by the UK-based company Canonical Ltd., owned by South African entrepreneur Mark Shuttleworth. Canonical generates revenue by selling technical support and services related to Ubuntu, while the operating system itself is entirely free of charge.

Gnome

GNOME [2] is a desktop environment and graphical user interface that runs on top of a computer operating system. It is composed entirely of free and open source software. It is an international project that includes creating software development frameworks, selecting application software for the desktop, and working on the programs that manage application launching, file handling, and window and task management.

Used by millions of people across the world, it is the most popular desktop environment for GNU/Linux and UNIX-type operating systems. The desktop has been utilised in successful, large-scale enterprise and public deployments, and the project's developer technologies are utilised in a large number of popular mobile devices.

Default Software

Ubuntu will install a default set of Software Packages for everyday work.

Text and Layout

Use gedit [3] for simple and fast plain text editing.

Open Office Writer [4] can open most known text formats, including Microsoft Files. Useful for everyday office use. Ubuntu will switch to Libre Office [5] in the near future.

Image and Photo

F-Spot [6] simplifies digital photography by providing intuitive tools to help you share, touch-up, find and organize your images.

Open Office Draw [4] allows simple image manipulation. Useful for simple graphics and office integration. Ubuntu will switch to Libre Office [5] in the near future.

Multimedia

Rhythmbox [7] is an integrated music management application, originally inspired by Apple's iTunes.


Totem Movie Player [8] is a media player for GNOME. It has support for most audio and video codecs including DVDs among many others.

Internet


Firefox [9], the worlds best Internet Browser.


Evolution [10] is the GNOME groupware application that provides mail, calendaring and addressbook functionality.


Empathy [11] is a messaging program which supports text, voice, and video chat and file transfers over many different protocols.

Administration


Synaptic [3] is a graphical front-end to apt, the package management system in Ubuntu.


GNOME Terminal [12] is a terminal emulation application to access a UNIX shell in the GNOME environment.

Additional Software

Text and Layout


Use scribus [13] if you need advanced layout features. Useful for creating flyers, handouts, formulas, etc.


Texmaker [14] is a free, modern and cross-platform LaTeX editor that integrates many tools needed to develop documents with LaTeX, in just one application.


Okular [15] is a universal document viewer with support for advanced document features, such as annotations, forms, and embedded files.


Adobe® Reader® [16] software is the free global standard for reliably viewing, printing, and commenting on PDF documents.


Kiwix [17] is an offline reader for Web content. It's especially intended to make Wikipedia available offline (see features). *The required .deb package to install Kiwix can be found on the external mirror inside the custom folder*

Image and Photo


The **G**NU **I**mage **M**anipulation **P**rogram [18] allows advanced image editing for Professional use.


Inkscape [19], an Open Source vector graphics editor, with capabilities similar to Illustrator, CorelDraw, or Xara X, using the W3C standard Scalable Vector Graphics (SVG) file format.

Multimedia


VLC [20] is a free and open source cross-platform multimedia player and framework that plays most multimedia files as well as DVD, Audio CD, VCD, and various streaming protocols.


Audacity [21] is free, open source software for recording and editing sounds.

Internet


Thunderbird [22] is a free email application that's easy to set up and customize


Pidgin [23] is an easy to use and free chat client used by millions. Connect to AIM, MSN, Yahoo, and more chat networks all at once.


KompoZer [24] is a complete web authoring system that combines web file management and easy-to-use WYSIWYG web page editing.

References

[1] https://help.ubuntu.com/
[2] http://library.gnome.org/users/
[3] http://library.gnome.org/users/gedit/
[4] http://www.openoffice.org/
[5] https://www.libreoffice.org/
[6] http://f-spot.org/
[7] http://projects.gnome.org/rhythmbox/
[8] http://library.gnome.org/users/totem/
[9] https://www.mozilla.org/firefox/
[10] http://library.gnome.org/users/evolution/
[11] http://library.gnome.org/users/empathy/
[12] http://library.gnome.org/users/gnome-terminal/
[13] http://www.scribus.net/
[14] http://www.xm1math.net/texmaker/
[15] http://okular.kde.org/
[16] https://www.adobe.com/products/reader.html
[17] http://www.kiwix.org/index.php/Main_Page/en
[18] http://www.gimp.org/
[19] http://inkscape.org/
[20] http://www.videolan.org/
[21] http://audacity.sourceforge.net/
[22] https://www.mozilla.org/thunderbird/
[23] http://www.pidgin.im/
[24] http://kompozer.net/

Appendix

Troubleshooting

LTSP is a complex System involving networking, multiple Architectures and much more hassle.

If something does not work, the best help is experience. Unfortunately Experience is hard to teach. This section will show how some real world Problems are solved. They should give you some advice on how to debug a Problem.

Example 1: The client does not want boot

Trying to boot a MacosX powerpc did not work. The powerpc chroot environment was successfully built using the instructions in the Chapter *Server Setup -> LTSP setup*. Pressing *n* at boot showed up a *netboot logo* but the mac never booted the linux. After a long time, it continued booting MacOSX from the local Harddisk.

1. Check if the client gets a IP and other settings at all.
   In the booted MacosX you check *Apple-Menu -> Preferences-> Network Settings*. If your Mac got a IP from the server it should be seen here (*10.55.66.X* in our case). If a address from the private range *192.168.X.Y* is used, the client never got a DHCP-Reply from the server and one needs check if the dhcp on the server side is running. If your client does not have any Operating System installed at all, you should try using a live boot CD to see if the dhcp comes trough
   in our case the client got a valid IP 10.55.66.24
2. Now we know, that dhcp is basicaly working, but somehow the client does not understand the boot options offered. Unfortunately DHCP can offer a lot options to the clients, but does not log a lot of information. This makes it hard to debug.
   The best way is to look at the network traffic itself. This is done by using a application called wireshark. Wireshark will listen on a ethernet interface for any packages and display them. You can install wireshark trough synaptics.
   in our case, this showed, that the options we adjusted in the dhcp.conf file did not get sent properly (wrong boot-file name and wrong boot method). This indicated, that we started dhcp with wrong options, a manual restart of the dhcp server did solve the boot problem.

Example 2: Bad Graphic Output

The client did boot into Linux, and showed the login screen. However there already had been graphical bugs, but it worked. After trying to log in, the whole graphical environment went nuts.

The best way to debug something like this is to examine the running chroot on the client.

Enable root in chroot

Linux contains a usually ignored feature called *virtual consoles*. By pressing <*ctrl+alt+F1*> the system will switch to *virtual console number 1*. *Press* <*ctrl+alt+F2*> for a second *virtual console*.
The graphical environment itself too runs on a virtual console, most linux uses number 7 (*ctrl+alt+F7*), but it can reside on higher numbers for various reasons.

If you switch to a *virtual console* you usualy are greeted by a text login. As the usual ltsp-users only exists at the server, you can not login here. Contrary to your graphical environment, these *virtual consoles* run on the local host and access the local linux. You are in the *chroot* environment described above.

The default setup is to not allow any logins on this system. This is a **important** security consideration.

To debug client problems, it is nice to have access to local log files and analyze the hardware. The easiest way is to enable the root user in the chroot environment.

```
administrator@ltsp-server:~/$ sudo chroot /opt/ltsp/i386/ /bin/bash
root@ltsp-server:/$ passwd
Enter new UNIX password:
Retype new UNIX password:
root@ltsp-server:/$ exit
administrator@ltsp-server:~/$ sudo ltsp-update-client
```

From now on you can change to a *virtual console* and login as user *root* with the password you have set.

Use the console on client

Depending on your problem, you now can search for problems on the client.

For hardware related troubles, the first thing to look at are the kernel logs. Those are shown with

```
root@ltsp20:~/$ dmesg
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
.. A lot of text.
[    1.616071] agpgart-intel 0000:00:00.0: Intel 830M Chipset
[    1.616350] agpgart-intel 0000:00:00.0: detected 8060K stolen memory
[    1.622499] agpgart-intel 0000:00:00.0: AGP aperture is 128M @ 0xd8000000
[    1.623980] [drm] Initialized drm 1.1.0 20060810
[    1.674699] e100: eth0: e100_probe: addr 0xd0100000, irq 20, MAC addr 00:30:05:38:1a:a3
[    1.676735] [drm] i915 disabling kernel modesetting for known bad device.
.. More text
```

In our case this revealed that our graphic card uses the i915 driver and it knows about a problem with this card.

Next we want find out more information about the card itself

```
root@ltsp20:~/$ lspci -v
...
00:02.0 VGA compatible controller: Intel Corporation
82845G/GL[Brookdale-G]/GE Chipset Integrated Graphics Device (rev 01)
      Subsystem: Fujitsu Technology Solutions Device 1003
      Control: I/O+ Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop-
ParErr- Stepping- SERR- FastB2B- DisINTx-
      Status: Cap+ 66MHz- UDF- FastB2B+ ParErr- DEVSEL=fast >TAbort-
 <TAbort- <MAbort- >SERR- <PERR- INTx-
       Latency: 0
       Interrupt: pin A routed to IRQ 16
       Region 0: Memory at d8000000 (32-bit, prefetchable) [size=128M]
       Region 1: Memory at d0000000 (32-bit, non-prefetchable)
[size=512K]
       Capabilities: [d0] Power Management version 1
             Flags: PMEClk- DSI+ D1- D2- AuxCurrent=0mA
PME(D0-,D1-,D2-,D3hot-,D3cold-)
             Status: D0 PME-Enable- DSel=0 DScale=0 PME-
       Kernel modules: i915
```

```
 ...

In case you want to save the ouput into a file (see below on how to
transfer these files to the server), you append a output redirection to
 the command
 root@ltsp20:~/$ dmesg '''> dmesg_output.txt
```

This will save the output from the dmesg call into the file *dmesg_output.txt*

Another place to look at are the files in */var/log/*

```
root@ltsp20:~/$ cd /var/log
root@ltsp20:/var/log$ ls
...
root@ltsp20:/var/log/$ more syslog
root@ltsp20:/var/log/$ more Xorg.7.log
```

You may want copy some of these files to the server for easier analysis

```
root@ltsp20:~/$ scp path/to/file/on/client your_server_login_name@server-ip:~/
root@ltsp20:~/$ scp dmesg_output.txt administrator@10.55.66.1:~/
```

You need to type your users password. The file will be copied into your home directory.

Research

In most cases you are not the first person with a certain problem. Try search for error messages from dmesg or log files with google. Usually you should find something. If you get too many results, remove generic parts like times. You may want add *ubuntu lucid* (for 10.04 LTS) or *ubuntu oneiric* for (12.04 LTS) at start of your query.

Ubuntu uses Launchpad [1] for support and bug reporting. There you will find a lot of people reporting bugs and sometimes even answers to the posted problems. You may try register and ask questions. Be prepared to offer additional information if asked for.

Some problems are easy to solve, some are hard. In our case we finally found the information, that this is a known problem with the kernel in use and we shall need update the kernel on the client image.
This is not perfect. Updating software on the client makes the chroot non-standard, and requires **good** and **detailed** documentation. Else you will surely forget about what you did and have to do the research again.
If possible try not to customize the chroot.

Implement solution

As mentioned above, we have to update our kernel. We found a Ubuntu kernel package at http://packages.ubuntu.com, http://security.ubuntu.com/ubuntu/pool/main/l/linux/linux-image-3.0.0-15-generic_3.0.0-15.26_i386.deb
Make sure, that you

- get the correct architecture (i386, powerpc, etc)
- do not add repositories to the chroot, but just download the *.deb* file.
  .deb files are single packages to be installed.

Copy the image into the roots directory in the chroot */opt/ltsp/i386/root* and install it into the chroot

```
administrator@ltsp-server:~/$ sudo cp Downloads/linux-image-3.0.0-15-generic_3.0.0-15.26_i386.deb /opt/ltsp/i386/root/

administrator@ltsp-server:~/$ sudo mount /dev -o bind /opt/arch/dev

administrator@ltsp-server:~/$ sudo chroot /opt/ltsp/arch /bin/bash

root@ltsp-server:/$ mount none -t proc /proc
```

```
root@ltsp-server:/$ mount none -t sysfs /sys
root@ltsp-server:/$ dpkg -i /root/linux-image-3.0.0-15-generic_3.0.0-15.26_i386.deb
root@ltsp-server:/$ umount /proc /sys
root@ltsp-server:/$ exit
administrator@ltsp-server:~/$ sudo umount /opt/arch/dev
administrator@ltsp-server:~/$ sudo ltsp-update-image
```

Disable root in chroot environment

For security reasons it is a bad idea to have root login allowed on the local client. It does not open immediate security holes, but it allows easier attacks. Therefore it is **strongly** advised to disable root after successful debugging. In the chroot do

```
root@ltsp-server:/ passwd --lock root
```

to disable password login.

Example 3: Immediate logout after login

The client did boot, but on some system just logged a user out right after login.

In the servers authentication-log at

```
/var/log/auth.log
```

The following line did not look correct

```
ltsp-server sshd[6725]: channel 17: open failed: administratively prohibited: open failed
```

Googling for this error brought up a lot recommendations to update the chroot with

```
administrator@ltsp-server:~/$ ltsp-update-sshkeys
administrator@ltsp-server:~/$ ltsp-update-image
```

Unfortunately this did not solve our Problem. Digging deeper into the google results, revealed this URL https:// / answers. launchpad. net/ ltsp/ + question/ 115286 where a poster suggested to forcefully disable *compiz*. *compiz* is part of gnome and allows fast 3D-Graphics on the Desktop. As most clients do not have good enough graphic cards to use 3D-Acceleration, this feature is usually auto-disabled by gnome. However in our case this auto-disable did not work, and even crashed the server some times. The post suggested that running the command

```
administrator@ltsp-server:~/$ sudo gconftool-2 --direct \
--config-source xml:readwrite:-/etc/gconf/gconf.xml.mandatory --type string \
--set /desktop/gnome/session/required_components/windowmanager metacity
```

enforces the use of the old window manager called *metacity* to be used instead of *compiz*. *metacity* does not use any 3D-Graphics and therefore does not crash the server anymore.

This is a good example for use copy&paste for solving problems. The line above uses mutliple switches, some are not obvious. Some times solutions in forums are just like this. A single line of code to be run and all is fine. As you usually do not know if a given command really solves your given problem it is strongly recommended to try such things on a *test system*. If no Hardware is available, you may want to build a *Virtual Machine* to test things.

References

[1] https://launchpad.net/

Contact

Contact

Web

- Hanfi [1]
- Poncho [2]
- LTSP Docu [3]

Mail

ltsp@spahan.ch [4]

References

[1] https://spahan.ch/
[2] https://poncho.spahan.ch/main/poncho.html
[3] https://spahan.ch/ltspdocu/
[4] mailto:ltsp@spahan.ch

License