Automount von Sambalaufwerken beim Login - inkl. Kerberos und Nomachine Terminalserver

Du möchtest dich gerne für unsere Hilfe erkenntlich zeigen . Gerne. Wir bedanken uns bei dir für deine Spende!

Spenden

Zum frei verfügbaren Apt-Repository

GITLAB:

LTSP mit Ubuntu

Einleitung

Das Linux Terminal Server Project dient als eine Art Framework, um eine Umgebung einzurichten, die es erlaubt, einfache Thin Clients (Rechner mit minimaler Ausstattung, meist ohne Festplatte) über das Netzwerk zu starten und Anwendungen auf dem Server zu benutzen. Die Vorteile bestehen darin, dass die Administratoren nur den Hauptserver verwalten müssen. Sämtliche Anwendungen müssen nur einmal installiert werden, um jedem zur Verfügung zu stehen. Wenn ein Client ausfällt, kann einfach ein neuer aufgestellt werden, und der Benutzer kann in wenigen Minuten die Arbeit wieder aufnehmen.

Voraussetzungen

Da sämtliche Aufgaben vom Client auf den Server übertragen werden, muss der Server über eine entsprechende Ausstattung verfügen. Dabei ist das Augenmerk weniger auf eine starke CPU zu legen als vielmehr auf Arbeitsspeicher und schnelle Festplatten. Je nach Anzahl der Clients sollte der Server über mehr als 1 GB RAM verfügen. Als Faustregel gilt 256 MB RAM für die Serveranwendungen und

zusätzlich ~30 MB RAM für jeden angeschlossenen Client. Werden viele grafische Anwendungen gleichzeitig verwendet, sollte man pro Client mit ~128 MB RAM rechnen. Ein Server, der 5 Clients bedient, sollte also mindestens ~512 MB RAM besitzen, ein Server für 20 Clients entsprechend ~1024 MB. Auch auf eine gute Datenrate der Festplatte(n) ist zu achten. IDE-Platten versagen ihren Dienst ab etwa 10 Clients aufgrund des zu hohen Datenaufkommens. SCSI- oder SATA-Festplatten sollten aber mit 20 Clients fertig werden. Bei den Festplatten ist weiter darauf zu achten, dass bei vielen Clients die Platten im Dauerlauf sind und somit mechanisch sehr beansprucht werden. Man sollte daher in Erwägung ziehen spezielle „Server"-Festplatten zu kaufen, die eine höhere Lebensdauer auch bei hoher Beanspruchung haben. Es ist auch zu empfehlen ein Netzwerk mit einer Datenübertragungsrate von 100 Mbit/s und mehr zu verwenden, da es aufgrund der X-Server Daten zu einem entsprechend hohen Datenverkehr kommt, und langsamere Netzwerke nicht mehr ausreichen.

Weiter sollte man beachten, dass die Clients, die über das Netzwerk booten wollen, einen Netzwerk-bootfähige Netzwerkkarte haben. Am besten wäre eine Karte, die entweder PXE oder Intel Etherboot unterstützt.

Installation

Eingesetztes Betriebssystem: Edubuntu 11.04 32bit mit PAE Kernel für mehr als 4GB RAM: Natürlich fragen sich jetzt mache wieso 32bit. Hierfür gibt es mehrere gute Gründe.

- Noch teilweise Inkompatiblität zu Kauf- und OpenSource-Programmen (z.B. Dudenkorrektor)
- Hat man bei der Clientstruktur eine Mischstruktur, (64 und 32bit) so muss man 2 Images zur Auslieferung bereitstellen, und demnach auch 2 pflegen.
- Nachdem Prozessorgeschwindigkeit nicht das Riesenthema am Desktop ist, und die meisten Progamme die Kerne ja nicht ganz ausnutzen können, ist es nicht nötig ein 64bit System hierfür zu installieren. Und Ausnahmen wie „ffmpeg" oder „Videoschnitt mit Cinelerre" benötigen ohnehin einen Fullstateclient.

Benötigten Pakete aussuchen und installieren

- tsp-server
- ltsp-server-standalone (nur falls kein geeigneter DHCP-Server im LAN bereits aktiv ist)
- openssh-server
- nbd-server (ab Gutsy, s. LTSP without NFS)
- ltspfs und ltspfsd (falls man an die Clienten lokale Geräte anbinden will; z.B. USB-Sticks)

aptitude install ltsp-server openssh-server nbd-server ltspfs ltspfsd

LTSP-Umgebung installieren

Bevor man die Umgebung installiert sollte man sich im klaren sein was man möchte. Zur Installation legt man sich zuerst eine Konfigurationsdatei an. Möchte man nur einen Terminalserver haben, muss

man diese nicht unbedingt erstellen.

sudo apt-key adv –recv-keys –keyserver keyserver.ubuntu.com 0C5A2783

```
nano /etc/ltsp/ltsp-build-client.conf

# The chroot architecture.
ARCH=i386

# ubuntu-desktop and edubuntu-desktop are tested.
# If you test with [k|x]ubuntu-desktop, edit this page and mention if it
worked OK.
# kubuntu lucid (10.10) working okay.
FAT_CLIENT_DESKTOPS="ubuntu-desktop"

# Space separated list of programs to install.
# The java plugin installation contained in ubuntu-restricted-extras
# needs some special care, so let's use it as an example.
LATE_PACKAGES="
    ubuntu-restricted-extras
    ldm-ubuntu-theme
    mesa-utils
    devede
    sauerbraten
    nfs-client
    language-support-de
    language-pack-gnome-de
    language-pack-kde-de
"

# This is needed to answer "yes" to the Java EULA.
# We'll create that file in the next step.
DEBCONF_SEEDS="/etc/ltsp/debconf.seeds"


# This uses the server apt cache to speed up downloading.
# This locks the servers dpkg, so you can't use apt on
# the server while building the chroot.
MOUNT_PACKAGE_DIR="/var/cache/apt/archives/"

#EXTRA_MIRROR="
#    http://ppa.launchpad.net/stgraber/ppa/ubuntu $DIST main
#"
```

Weiters erstellen wir die Datei **/etc/ltsp/debconf.seeds** um Lizenzen zu akzeptieren.

```
# Do you agree with the DLJ license terms?
sun-java6-bin   shared/accepted-sun-dlj-v1-1    boolean true
sun-java6-jre   shared/accepted-sun-dlj-v1-1    boolean true
```

Die bereits installierten LTSP-Pakete enthalten nur Werkzeuge, um eine LTSP-Umgebung zu

installieren. Um die für LTSP benötigten Dateien nun zu installieren, ruft man in einem Terminal folgenden Befehl auf:

```
ltsp-build-client
```

Der Befehl ltsp-build-client installiert ein Client-Betriebssystem nach **/opt/ltsp/i386**. Die zu bootenden Kernel und die Unterstützung für PXE wird analog in **/var/lib/tftpboot/ltsp/i386** installiert.

Anschliessend muss für die Clienten der Zugriff auf den Server per ssh erlaubt werden:

```
ltsp-update-sshkeys
```

Zusätzliches 32bit-System in einer 64bit-Umgebung

Will man unter einem 64-Bit Server zusätzlich ein System für 32-Bit Clienten installieren, so tut man dies mit dem Befehl:

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

Clients boofähig machen

Dies ist ein leichtes, hat man einen DHCPserver, braucht man nur den Hostnamen/IP-Adresse des TFTPs (in diesem Fall wäre das ja ibm_darkwolf.lan) und die Startfile in den DHCP eintragen.

```
192.168.1.XX
ltsp/i386/pxelinux.0
```

Weiterführende Konfiguration und Andwendung

NBD

Ab Ubuntu 8.04LTS übernimmt der nbd die Bereitstellung des LTSP-Images, nicht mehr NFS. Der Befehl **sudo ltsp-update-image** erstellt diese in **/opt/ltsp/images**. Bei Client-Rechnern, die nicht dem Hostsystem entsprechen, muss dem Befehl die Architektur mitgeteilt werden, z.B.:

```
ltsp-update-image --arch i386
```

Nach Änderungen mittels **chroot** in **z.B. /opt/ltsp/i386** (Setzen des root-Passwortes, Installation von neuen Paketen, ...) muss obiger Befehl erneut ausgeführt werden, damit die Änderungen im Image

übernommen werden. Der nbd stellt seine Dienste via des **inetd** bereit. In der Konfigurationsdatei **/etc/inetd.conf** muss ergo ein Eintrag für diese Freigabe vorhanden sein. Ein Eintrag für zwei unterschiedliche Architekturen sieht so aus:

```
2000    stream  tcp     nowait  nobody /usr/sbin/tcpd /usr/sbin/nbdrootd
/opt/ltsp/images/amd64.img
2001    stream  tcp     nowait  nobody /usr/sbin/tcpd /usr/sbin/nbdrootd
/opt/ltsp/images/i386.img
```

In dieser Konfiguration werden die Images an die Ports 2000 und 2001 gebunden. Zugriffsbeschränkungen auf den nbd legt man am einfachsten in der Datei **/etc/hosts.allow** fest. Nachdem alle Anpassungen durchgeführt wurden, müssen sämtliche Dienste, die zum Betrieb eines LTSP-Servers nötig sind, neu gestartet werden:

```
restart tftpd-hpa
/etc/init.d/openbsd-inetd restart
```

Softwarepakete für den Client nachinstallieren

Das Beispiel mit den wichtigsten Paketen und den optionalen Grafiktreiber von Nvidia.

```
export LTSP_HANDLE_DAEMONS=false
chroot /opt/ltsp/i386
mount -t proc proc /proc
apt-get install nvidia-glx-185 nvidia-185-libvdpau

## chroot /opt/ltsp/i386 apt-get install nvidia-current aptitude ssh htop
nmap nvidia-settings nagios-nrpe-server nagios-plugins elinks nano
language-pack-gnome-de language-pack-kde-de language-support-de skype
supertuxkart smc frozen-bubble devede nautilus-clamscan nautilus-gksu
nautilus-image-converter

umount /proc
exit
```

Mit dieser Methode lassen sich beliebige Pakete aus den Repositories installieren. Ab Ubuntu 8.04LTS müssen Änderungen, die man am LTSP-Filesystem vornimmt, ins Image übernommen werden. Dazu ruft man den den Befehl ltsp-update-image auf, der ein neues Image generiert. Bei heterogenen Installationen muss man dem Befehl die Architektur mitteilen. Beispiel:

```
ltsp-update-kernels
ltsp-update-image --arch i386
```

LTS.CONF

Nachdem man den Server gemäß LTSP konfiguriert hat, kann man nun die Clienten davon booten lassen. Ab und zu sind jedoch noch einige Feineinstellungen nötig, damit man komfortabel mit den

Clienten arbeiten kann. Dazu gehören zum Beispiel das verwendete Sprachschema, die Bildschirmauflösung und das Anbinden von Peripheriegeräten an die Clienten. Diese Optionen übergibt man mittels der Datei lts.conf, welche auf dem Server liegt. Falls man die Datei benötigt, so muss man sie mit einem Editor [1] anlegen und bearbeiten.

Die LTS.conf wird auch hier näher beschrieben.

Bis Feisty befindet sich die Datei im Ordner /opt/ltsp/i386/etc/; seit Gutsy ist sie unter /var/lib/tftpboot/ltsp/i386/ zu finden. Statt i386 kann im Pfad natürlich auch amd64 stehen - je nach verwendeter Architektur der Client-Rechner.

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

Hier eine Beispieldatei:

```
[default]
LDM_THEME=edubuntu
XKBMODEL = pc105
XKBLAYOUT = de
CONSOLE_KEYMAP = de
LDM_DIRECTX = true
#[20:cf:30:81:4d:75]
#LTSP_FATCLIENT=false
XSERVER = nvidia
MODULE_01 = nvidia
X4_MODULE_01 = glx
```

LDAP Anbindung (PAMconfig)

common-account

```
account         sufficient      pam_ldap.so
account         required        pam_unix.so
```

common-auth

```
auth    sufficient      pam_ldap.so
auth    required        pam_unix.so use_first_pass nullok_secure
```

common-password

```
password   sufficient pam_ldap.so
password   required   pam_unix.so use_first_pass nullok obscure min=4 max=8
md5
```

common-session

```
session required        pam_unix.so
session optional        pam_foreground.so
session         required        pam_mkhomedir.so skel=/etc/skel umask=0077
```

Aufgaben nach dem gebauten Image

- Samba installieren und konfigurieren
- Rootpasswort setzen
- Lokale Administratoren hinzufügen
- MDNS Domäne setzen
- Wake on Lan einstellen
- Gnome-Panel in den Autostart geben (/etc/xdg/autostart)
  http://wiki.ubuntuusers.de/autostart#Fuer-alle-Benutzer
- Programme die nicht benötigt werden aus der Chroot entfernen
- Man darf lokal (am Squashfs) nicht den gleichen Benutzer angelegt haben wie auf dem Server

Links

- Customizing thin client behaviour
- https://help.ubuntu.com/community/UbuntuLTSP
- Introduction to available software
- How do LTSP Fat Clients work?
- Ubuntu LTSP FatClients WICHTIG
- VirtualGL 3D Grafikberrechnung über das Netzwerk
- UbuntuLTSP mit AtomIon
- LTS.CONF

Gitlab Project

LTSP - **Namespace:** public-projects

Last activity: 2023-08-15 - 18:06:20