

Plattform:
openSUSE 10.2 (X86-64)

Basisinstalltion

- openSUSE Base System
- YaST System Administraion
- openSUSE Software Management
- Enterprise Software Management (ZENworks)

zusätzlich installierte Programme

- midnight commander (mc)
- subversion (subversion)
- mytop (MySQL TOP)

Für Stud.IP benötigte Pakete

- apache2 (Prefork)
- apache2-mod_php5
- autoconf
- gcc
- libyaz-devel
- make
- mysql (5.0.26)
- php5-devel
- php5-mysql (5.2.0)
- php5-gettext (5.2.0)
- php5-sysvshm
- php5-sysvsem
- php5-xsl
- tcpd-devel
- tetex (beinhaltet dvips bereits)
- te_latex
- ImageMagick
- fop
- unzip
- yaz
- zip

Mailserver Konfiguration

- **smtp.uni-halle.de** eingetragen
- Eingehende/Ausgehende Mails ermöglichen (Firewall-Port muss für Senderverify offen sein)

```
yast
--> Mail Transfer Agent
[x] entfernte SMTP Verbindungen akzeptieren
[x] Firewall Port öffnen
```

TEMP-Verzeichnis automatisch leeren

/etc/sysconfig/cron

MAX_DAYS_IN_TMP = "1"

Temp wird so jeden Tag aufgeräumt, wenn 1 Tag nicht darauf zugegriffen. Dateien von root werden nicht angerührt.

./SuSEconfig nicht vergessen!

phpMyAdmin 2.9.2 (opt)

Optional, aber empfohlen:

phpMyAdmin 2.9.2 installiert? und über die SSL-Verbindung das root-Passwort für die DB gesetzt.

Basisinstalltion

zusätzlich installierte Programme
Für Stud.IP benötigte Pakete
Mailserver Konfiguration
TEMP-Verzeichnis automatisch leeren
phpMyAdmin 2.9.2 (opt)
Stud.IP Quelltexte installieren
Ordnerstruktur
Schreibrechte setzen
Apache Webserver konfigurieren
Modul Konfiguration
SSL-Schlüssel einspielen
MLU Server-Konfiguration
Symlink zur Stud.IP Konfigurationsdatei
Stud.IP Datenbank
Anpassen der Konfigurationsdateien
Bibliotheksanbindung (YAZ) einbauen

alle %-Nutzer gelöscht

Stud.IP Quelltexte installieren

Wie im entsprechenden Artikel? beschrieben, wird die Software nach /srv/www/studip installiert.

Ordnerstruktur

```
/cli      ... Command Line Interface: für Konfig und CRON benötigte Skripte
/config   ... Konfigurationsdateien (Stud.IP + Apache)
/data     ... Nutzdaten (archiv, externe Seiten, hochgeladene Dateien)
/db       ... Dump der Datenbankstruktur
/doc      ... verschiedene Anleitungen
/lib      ... Programmbibliotheken, Klassen, Plugins
/local    ... Shellskripte zum Erzeugen der Übersetzungsdateien
/public   ... für die WWW-Öffentlichkeit
/vendor   ... externe Bibliotheken
```

Schreibrechte setzen

In bestimmte Bereiche muss der Webserver-User schreiben können:

/data/(archiv, exter_config, upload_doc)
/public/pictures/(smiley, tex, user-bilder, banner)

```
cd /srv/www/studip
chown -R wwwrun:root data
cd public
chown -R wwwrun:root pictures
```

Apache Webserver konfigurieren

Modul Konfiguration

/etc/sysconfig/apache2 editieren: Modulkonfiguration und SSL

```
APACHE_MODULES="actions alias authz_host deflate dir env expires
headers include log_config mime negotiation rewrite setenvif ssl suexec
userdir php5"

APACHE_SERVER_FLAGS="SSL"
```

SuSEconfig macht die Sache aktiv

SSL-Schlüssel einspielen

- /etc/apache2/ssl.key
- /etc/apache2/ssl.crt/ca.crt
- /etc/apache2/ssl.crt/server.crt

MLU Server-Konfiguration

/etc/apache2/vhosts.d/mlu.conf einspielen

Konfig für serverseitige gzip Output-Komprimierung

Konfig für vhosts (www, ref, projekte etc.) und für SSL-Betrieb eingespielt

Hier wird Stud.IP als DocumentRoot für die Virtual Hosts eingestellt!

Symmlink zur Stud.IP Konfigurationsdatei

```
cd /etc/apache2/conf.d
ln -s /srv/www/studip/config/studip-httpd.conf studip-httpd.conf
```

Stud.IP Datenbank

1. Datenbank einspielen

```
mysql -u root -p
> CREATE DATABASE `studip` DEFAULT CHARACTER SET latin1 COLLATE latin1_german2_ci;
> quit;
$mysql -u root -p studip < studip.sql
$mysql -u root -p studip < studip_root_user.sql
$mysql -u root -p studip < studip_default_data.sql
$mysql -u root -p studip < studip_resources_default_data.sql
```

2. Datenbanknutzer anlegen

```
CREATE USER 'studipuser'@ 'localhost' IDENTIFIED BY '*****';

GRANT USAGE ON * . * TO 'studipuser'@ 'localhost' IDENTIFIED BY '*****'
WITH MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USI

GRANT SELECT , INSERT , UPDATE , DELETE , CREATE TEMPORARY TABLES, DROP ON `studip`
'studipuser'@'localhost';
```

Anpassen der Konfigurationsdateien

- /srv/www/studip/config/config.inc.php
- /srv/www/studip/config/config_local.inc.php
- /srv/www/studip/config/config_room_groups.inc.php
- /srv/www/studip/config/studip-httpd.conf

Bibliotheksanbindung (YAZ) einbauen

1. Folgende Pakete werden benötigt:

- yaz
- php5-devel, autoconf (für phpize kommando)
- gcc, make (für ./configure + make)
- libyaz-devel (für yaz-config)
- tcpd-devel (für lwrap-libraries (TCP wrapper Library) make)

2. Die PHP-Schnittstelle PHP-yaz herunterladen `$wget http://pecl.php.net/get/yaz-1.0.8.tgz`

3. Das PHP shared module kompilieren

```
$phpize
$./configure
$make
```

Auf unserem Server war es noch erforderlich, das generierte Makefile per Hand zu anzupassen, dass er in das Verzeichnis /usr/lib64 nach der libcrypto.so und der libssl.so schaut. Im Detail sieht das so aus:

```
# Alte Zeile im Makefile
# YAZ_SHARED_LIBADD = -lnsl -lwrap -lcrypto.so -lssl.so -lyaz
# Neue Zeile im Makefile
YAZ_SHARED_LIBADD = -lnsl -lwrap -L/usr/lib64/libcrypto.so -/usr/lib64/libssl.so
```

Für die Option --libdir=/usr/lib64 im ./configure Aufruf hat er sich scheinbar nicht interessiert.

4. Die erstellte Datei **yaz.so** in /usr/lib64/php5/extensions kopieren
5. in /etc/php5/conf.d eine entsprechende **yaz.ini** mit folgendem Inhalt anlegen

```
; comment out next line to disable yaz extension in php
extension=yaz.so
```