

APACHE - 2 WEB SERVER

Kurs
+ Übungen
(Deutsch)

bei
WBS Training
München

Web Server Kurs

Übersicht

1 – Geeignete Partitionierung *home pages* und Logdateien

- /var, /home, /srv

2 – Benötigte Komponenten:

- *Web Server* (Apache2, tthttpd, ...)
 - Normal: http://....
 - *Secure SSL*: https://....
- *FTP Server* (vsftpd, pureftpd, proftpd, in.ftpd, wu-ftp)
 - Benutzer *login* für *Home Pages Upload*
 - *Anonymous* für separate *download*-Punkte im Verzeichnis

3 - Web Server

Theorie

- Was ist ein Web Server (tthttpd, MSInformation Server, Netscape etc.)
- TCP/IP und das HTTP/HTTPS Protokoll
- Was sind URL und URI
- Auflösung der Namen zur IP ohne Proxy
- Auflösung der Namen zur IP mit Proxy
- *Browser*
- Unterschiede zwischen: tthttpd, Apache 1.3 und Apache 2.
- Wie wird Apache benutzt
- Vorbereitungen für die Übungen:
- Start/Stop des Apache
- Erste Übung mit HTTP Protokoll
- Überwachung des `http` Protokolles mit Ethereal
- Konfigurations-Dateien
- Wo ist was
- Benutzer- und Gruppeneinstellungen für *client* Prozesse
- Modul Roles und seine Direktiven
- Apache Server Status
- Dokumentation
- Wichtige *core* Direktiven

Praxis

- Freigeben `server-status` und `server-info` Modul
- Vorbereitung für Übungen
- **Alias** Direktive
- Container (**Directory**, **Location**, **Files**)
- Options **FollowSymlinks**
- **DirectoryIndex**
- **Allow - Deny**
- **Public_html**
- Authentifizierung
- INDEXES
- **AllowOverride**
- Virtuelle *Hosts*
- CGI

- SUDO
- Überprüfung, ob das PHP Modul arbeitet
- Erzeugen von *web access reports* mit **Webalizer**
- *Secure* HTTP: HTTPS Konfiguration:
- Erzeugen eines temporär selbst signierten Zertifikates SSL:
- *Secure Webmail* Installation (**Squirrelmail**)
- Projekt Bilder-Galerie: IDS - CGI
- httpd2-prefork Befehls-Optionen
- Steuerung der gestarteten Prozesse und *clients* von Apache
- Apache in Debian basierenden Distributionen
- Schreiben der Apache Konfigurations-Skripte

4 - FTP server: Installation und Konfiguration vsftpd

- für *Web Pages upload*
- für *file downloads*

5 - Installation LAMP - Linux-Apache-MySQL-PHP

- Installation MySQL und PHPMyAdmin

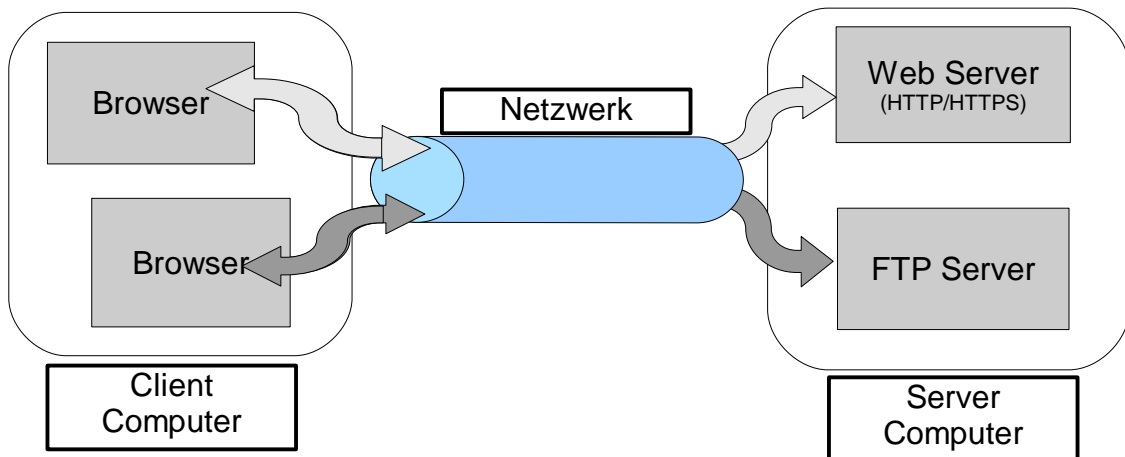
6 - Bezug und Installation WAMP-Windows-Apache-MySQL-PHP

7 - Bezug und Installation XAMP für MAC OS (nur Intel)

Inhaltsverzeichnis

Was ist ein Web Server	5
TCP/IP und das HTTP/HTTPS Protokoll.....	5
Was sind URL und URI	5
Browser.....	5
Unterschiede zwischen: httpd, Apache 1.3 und Apache 2.....	5
How to get Apache.....	6
Apache für Linux.....	6
Apache für Windows.....	6
Apache für MAC OS X.....	6
Installation Apache in SuSE.....	7
Start/Stop von Apache.....	7
Erste Übung mit HTTP Protokoll.....	7
Überwachung des http Protokolles mit Ethereal.....	7
Konfigurations-Dateien.....	7
Wo ist was.....	7
User und group Einstellungen für clients Prozesse.....	7
Modul.....	7
Apache Server Status.....	7
Dokumentation.....	7
Wichtige core Direktiven.....	8
Vorbereitung für Übungen.....	8
Einige troubleshooting Befehle:.....	8
Testen Sie die web server's home pages.....	8
ALIAS.....	8
Containers.....	9
Options FollowSymlinks.....	9
DirectoryIndex.....	9
Allow - Deny.....	10
Public_html.....	11
Authentication.....	11
INDEXES.....	12
AllowOverride.....	13
Virtual Hosts.....	14
cgitest.cgi.....	17
Erzeugen von web access Protokollen mit Webalizer.....	18
Secure HTTP: HTTPS Konfiguration:.....	19
Erzeugen eines temporären Zertifikates für SSL.....	19
Überprüfen, ob die PHP-Module arbeiten.....	20
.....	20
Secure Webmail Installation.....	21
IDS Images Galerie - CGI Installation.....	23
httpd2-prefork command Options.....	24
Controlling the started Prozesse und clients on Apache.....	24
Apache in Debian basierenden Distributionen.....	25
Writing Apache configuration scripts.....	25
LAMP - Linux-Apache-MySQL-PHP.....	26
WAMP-Windows-Apache-MySQL-PHP.....	28
Installation Apache2, PHP4 und MySQL in Windows2000/XP.....	28
Manuelle Installation für ganz Mutige!.....	28
Installation APACHE2.....	28
Installation MYSQL.....	28
Installation PHP4.....	28
Installation phpMyAdmin.....	28
CONFIGURATION.....	29
Apache2 (SuSE 10.1)	

- **Was ist ein Web Server** (thttpd, MSInformation Server, Netscape, AOL Server etc.)



- **TCP/IP und das HTTP/HTTPS Protokoll**

- TCP Transport-Protokoll des Paketes
- HTTP ist die Beschreibungssprache für den Informationsaustausch
- GET und POST sind die zwei häufigsten Befehle der *Browser* zum Anfordern einer Resource. (web *page*, image, pdf Dokument etc.)

- **Was sind URL und URI**

Uniform **R**esource **L**ocator

Uniform **R**esource **I**dentifier

Die Zeichenfolge identifiziert eine Resource über den Namen und kann die *location* enthalten

Beispiel einer URL: http:// www.elop.de /bilder/kopf1.jpg

1

2

3

1: Protokoll

2: Server-Adresse

3: Location und Resource(**URI**)

- **Browser**

Linux:

- Firefox, Opera, Konqueror, Galeon, Browsex, etc

Windows:

- MSIE, Firefox, Opera

MAC OS X:

- Safari, Opera, Firefox, Camino

Text(Linux):

- lynx, w3m

- **Unterschiede zwischen: thttpd, Apache 1.3 und Apache 2.**

- httpd : Enthalten im *kernel*, damit für andere Prozesse schnell erreichbar
- Apache 1.3
 - *Forking based server*
 - Ein zentraler Kern (*core*) und dynamische Module
- Apache 2.x
 - Enthält den *Core*, 2 *sub-core* (Server) und dynamische Module
 - *Core* : Zentraler Teil des Servers

- *Subcores*:

- apache2-prefork
 - Kopie von sich selbst als eigenen Prozess (*forking*) um einen neuen *client* zu bedienen.
 - Vorteile: Mehr Module sind kompatibel mit ihm.
 - Nachteile: Langsamer als der *worker*.
Benötigt mehr RAM
- apache2-worker
 - Startet einen Prozess like in *prefork* but each process can serve multiple *clients* instead of only one(as seen mit *prefork*). Each new process will in turn start sub-Prozesse called '*Threads*' to serve each new *client*.
 - Vorteile: Schneller als *prefork*
Speicher wird effizient genutzt
 - Nachteile: Weniger Module sind kompatibel zu ihm.
- apache2-perchild
 - Similar to '*worker*' except that each Virtual Host ist assigned a separate process which in turns uses separate '*Threads*' to serve der *clients*.
This allows also to assign a separate user und group per process which eliminates the need of SUExec.
 - Vorteile: Flexibel, kann verschiedene Benutzer über virtuell *Hosts* zuweisen.
Speicher wird effizient genutzt
 - Nachteile: Weniger Module sind kompatibel zu ihm.
- Dynamische Module
 - Pieces of software that can attach themselves to the *core* of Apache to add functionality.
Tip: Most modules come mit set of Direktiven that can be used in the Apache configuration file to control the behavior of the module.

• How to get Apache

- Apache für Linux
 - Most distributions provide already compiled Apache.
 - Source code of Apache can be found at: <http://www.apache.org>
- Apache für Windows can also be downloaded from:
<http://www.apachefriends.org/en/xampp-windows.html>
Das Paket enthält:
Apache HTTPD 2.2.3, MySQL 5.0.27, PHP 5.2.0 + 4.4.4 + PEAR + Switch, MiniPerl 5.8.7, Openssl 0.9.8d, phpMyAdmin 2.9.1.1, XAMPP Control Panel 2.3, Webalizer 2.01-10, Mercury Mail Transport System für Win32 und NetWare Systems v4.01a, FileZilla FTP Server 0.9.20, SQLite 2.8.15, ADODB 4.93, Zend Optimizer 3.0.2, XAMPP Security. für Windows 98, 2000, XP.
- Apache für MAC OS X can also be downloaded from:
<http://www.apachefriends.org/en/xampp-macosx.html>
Das Paket enthält:
Apache 2.2.3, MySQL 5.0.27, PHP 4.4.4, PHP 5.2.0, Perl 5.8.8, ProFTPD 1.3.0a, phpMyAdmin 2.9.1, OpenSSL 0.9.8d, GD 2.0.33, Freetype 2.1.10, libjpeg 6b, libpng 1.2.12, libungif-4.1.4, zlib 1.2.3, expat 2.0.0, Ming 0.3, Webalizer 2.01-10, pdf class 009e, mod_perl 2.0.2, SQLite 3.3.7, phpSQLiteAdmin 0.2, libconv-1.9.2, gdbm-1.8.3, libxml-2.6.26, libxslt-1.1.17, openldap-2.3.27, imap-2004g, gettext-0.15, libmccrypt-2.5.7, mhash-0.9.7.1, curl-7.15.5, zziplib-0.10.82, bzip2-1.0.3, freetds-0.64

- **Installation Apache in SuSE**

apache2, apache2-doc, apache2-mod_php5, apache2-prefork,
apache2-example-pages (Paket: libapr0 ist wegen Abhängigkeiten bereits installiert)

- **Start/Stop von Apache** (rcapache2, /etc/init.d/apache2, inserv apache2)

- **Erste Übung mit HTTP Protokoll:** telnet localhost 80 (GET /)

- **Überwachung des http Protokolles mit Ethereal**

- Installieren Sie Ethereal von CD/DVD
- Starten Sie einen *browser* und Ethereal
- Starten Sie die Datenaufzeichnung für eth0 und Filter *port 80*
- Geben Sie im *Browser* <http://www.suse.de> ein
- Stoppen Sie Ethereal Datenaufzeichnung und betrachten Sie die Pakete

- **Konfigurations-Dateien**

Apache 1.3: /etc/apache/httpd.conf

Apache 2.x: /etc/apache2/httpd.conf

oder /etc/apache2/apache2.conf

(Include /etc/apache2/sites-enabled/[^.#]*)

auch (nur bei SuSE): /etc/sysconfig/apache2

Apache Quellcode in:

/etc/httpd/conf/httpd.conf

Hinweis: Direktive 'Include *Dateiname*' ruft an dieser Stelle andere Datei auf.

Wichtiger Hinweis für httpd.conf:

Bei fehlerhaften Direktiven startet Apache nicht!

- **Wo ist was (basic only)**

- Haupt *server* /usr/sbin/httpd2-prefork (Apache 1.3 oder älter)
oder /usr/sbin/httpd2-worker (apache2-mod_php5 funktioniert noch nicht!)
- Haupt Konfigurationsdatei /etc/apache2/httpd.conf
- Module /usr/lib/apache2/xxxx.so
- Log Dateien /var/log/apache2/xxxx_log
- *Haupt Homepages* /srv/www/htdocs/

- **User und group Einstellungen für *clients* Prozesse:**

Direktiven:

User wwwrun oder www-data

Group wwwrun

- **Modul : *role* und seine Direktiven (**server-info**)**

- Ändern Sie in /etc/sysconfig/apache2:

APACHE_MODULES="..... status info"

APACHE_EXTENDED_STATUS="on"

- rcapache2 restart

- In *browser*: <http://localhost/server-info/>

- **Apache Server Status:** <http://localhost/server-status/?refresh=10>

- **Dokumentation:** <http://localhost/manual/>
oder <http://www.apache.org>
oder <http://www.opensuse.org>

Die lokale Dokumentation /usr/share/apache2/manual
(apache2-doc Paket)

- **Wichtige core Direktiven:**

```
In /etc/apache2/default-server.conf
    DocumentRoot /srv/www/htdocs
In /etc/apache2/httpd.conf
    DirectoryIndex index.html
In /etc/apache2/uid.conf
    User wwwrun
    Group www
```

-----Praktische Übungen-----

- **Vorbereitung für Übungen:**

- Login als *root* und erstellen Sie das Verzeichnis /www
- Wechseln nach /etc/sysconfig/apache2
APACHE_SERVERNAME="laptop.linux.site" oder *local host name*
- Erzeugen einer leeren Datei /etc/apache2/conf.d/user.conf
 - Alternativ: Wechseln nach /etc/sysconfig/apache2:
APACHE_CONF_INCLUDE_FILES="/etc/apache2/user.conf"
und erzeugen Sie eine leere Datei: /etc/apache2/user.conf
- Laden und installieren Sie Opera aus dem Internet und w3m und lynx von der CD
- Starten Sie ein Terminalprogramm (konsole) mit Fensterverwaltung für:

```
⇒ mcedit /etc/apache2/conf.d/user.conf      Titel: USER.CONF
⇒ sudo su -      (für rcapache2 reload Befehl)  Titel: RELOAD/RESTART
⇒ sudo su - -c 'tail -f /var/log/apache2/error_log'
                                                    Titel: APACHE FEHLER
⇒ sudo su -      Titel: ROOT-BEFEHLE
⇒ sudo su - -c 'mc / /srv/www/htdocs'        Titel: ROOT-MC
```

- **Einige troubleshooting Befehle:**

```
netstat -ltupn | grep apache2
apache2 -S
watch 'netstat -ltupn | grep apache2 ;\
ps ax | grep apache2 | grep -v grep'
```

- **Testen Sie die web server's home pages:**

- Modifizieren Sie die Datei /srv/www/htdocs/index.html.de
- In *browser*: http://localhost/

ALIAS

Beispiel:

```
In user.conf:    alias /webicons/ /usr/share/apache2/icons/
In browser:      http://localhost/webicons/
```

Übung 2:

```
In user.conf:    alias /software/ /usr/share/doc/packages/
                  alias /web/ /www/
In browser:      http://localhost/software/
```

Resultat: ERROR 403 Access forbidden

```
In user.conf:    <Directory /usr/share/doc/packages>
                  Allow from all
                  Options +Indexes
                  </Directory>
                  <Directory /www>
```



```

        Allow from all
        Options +Indexes
    </Directory>

```

In *browser*: `http://localhost/software/`
`http://localhost/web/`

Containers

`<Directory>` Absolute Pfadangabe zu den *web* Dokumenten.

`<Location>` Relativer Pfad DocumentRoot des *server* (wird nach *Directory* und *Files* gelesen!)

`<Files>` Kann in `<Directory>` verschachtelt sein oder alleine stehen
(dann als Standard für server)

Hinweis: `<Files>` kann nicht in `<Location>` verschachtelt werden. Benutzen Sie
`<Location>` für Dateien.

zB. `<Location /usr/local/myprogs/readme.html>`

`<Limit>` Beliebiger Wert. Limitiert die Anzahl der benutzten HTTP Methoden

Options FollowSymlinks

Beispiel:

In Bash: `ln -s /home/[username]/Documents /srv/www/htdocs/userdocs`

In *browser*: `http://localhost/userdocs/` Nicht erlaubt!

In user.conf: `<Directory /srv/www/htdocs/>`
`Options +FollowSymlinks +Indexes`
`</Directory>`

In *browser*: `http://localhost/userdocs/` Erlaubt!

Hinweis: `FollowSymlinks` Funktioniert nicht innerhalb eines `<Location>` Blockes

DirectoryIndex

Übung 1:

In Bash: `ln -s /usr/share/doc/packages/bash /srv/www/htdocs/bash`

In user.conf: `<Directory /srv/www/htdocs/bash>`
`DirectoryIndex bashref.html`
`</Directory>`

In *browser*: `http://localhost/bash/`

Übung 2 (wenn benötigt):

Hinweis: Stellen Sie sicher, daß `w3m` und `lynx` sind

Befehl zum Erstellen eines symbolischen *links*:

```

ln -s /usr/share/doc/packages/w3m /srv/www/htdocs/w3m
ln -s /usr/share/doc/packages/lynx/lynx_help /srv/www/htdocs/lynx

```

In user.conf: `<Directory /srv/www/htdocs/w3m>`
`DirectoryIndex w3mhelp.html`
`</Directory>`

```
<Directory /srv/www/htdocs/lynx>
    DirectoryIndex Lynx_users_guide.html
</Directory>
```

Allow - Deny

Beispiel 1 (verbiete Zugriff auf eine einzelne Datei):

```
In user.conf:    alias /etc/ /etc/
                  <Directory /etc>
                    Options +Indexes
                    order allow,deny
                    allow from all
                    deny from localhost [dozent_host]
                    <Files fstab>
                      deny from all
                    </Files>
                    <FilesMatch ".*\.conf">
                      order deny,allow
                      deny from all
                      allow from localhost
                    </FilesMatch>
                  </Directory>
```

In *browser*: http://localhost/etc/ Erlaubt!
 Von Studenten: http://localhost/etc/fstab Nicht erlaubt!
 http://localhost/etc/

Nur für Dozent ist nicht erlaubt: http://[studenthost]/etc

Beispiel 2:

Erlaube den Zugriff auf das Verzeichnis (*location*) /cups/ für 4 aufeinander folgende Adressen:

```
In user.conf:    alias /cups/ /usr/share/doc/packages/cups/de/
                  <Location /cups>
                    order deny,allow
                    deny from all
                    allow from 192.168.71.12/30
                  </Location>
```

In *browser*: http://localhost/cups/ Nicht erlaubt!
 (nur *clients hosts* haben die Adresse 192.168.71.12 bis 192.168.71.15 und haben die Erlaubnis,
 für die Adresse http://[studenthost]/cups/

Übung 1: Erlaube Zugriff zu /sambahelp für die Kursteilnehmer

und gebe eine spezielle page an für Zutritt (DirectoryIndex): welcome.html

- Vergewissern Sie sich, daß das Paket samba-doc installiert ist.

```
- In user.conf:    alias /sambahelp/ /usr/share/samba/swat/help/
                  <Directory /usr/share/samba/swat/help/>
                    order deny,allow
                    deny from all
                    allow from 192.168.71.0/27 localhost
                    DirectoryIndex welcome.html
                  </Directory>
```

(nur *clients hosts* haben die Adresse 192.168.71.0 bis 192.168.71.31 und haben die Erlaubnis,
 für die Adresse http://[studenthost]/sambahelp/)

Wenn wir auf einen *link* klicken, kommen wir zu einem nicht existierenden Pfad:

```
http://servername/swat/help/manpages/smbd.8.html
```

Um diesen *link* bereitzustellen, müssen wir einen *alias* erzeugen:

```
alias /swat/ /usr/share/samba/swat/
```

Nun ist zwar die Fehlermeldung '*...NOT FOUND*' weg, aber dafür '*FORBIDDEN*'.

Der Grund ist, daß das Verzeichnis `/usr/share/samba/swat/help/manpages` ein symbolischer *link* ist. Jetzt müssen wir erlauben, den symbolischen *links* zu folgen:

```
<Directory /usr/share/samba/swat/help/>
    Options +FollowSymLinks
</Directory>
```

Hinweis: Die Options `+FollowSymLinks` kann einfach zur vorherigen *directory container* Deklaration hinzugefügt werden.

Public_html

Beispiel:

Installiere das Paket `htdig` (wenn es nicht bereits installiert ist)

In bash: `cp -R /usr/share/doc/packages/htdig/htdocs/* ~/public_html`

In browser: `http://localhost/~[userlogin]`

Authentication

Erstelle den user: `user1 (mkdir user1)`

Ein Beispiel: Benutzer `/~user1/` Verzeichnis `(/home/user1/public_html)`

In `user.conf`:

```
<Location /~user1>
    allow from all
    AuthType Basic
    AuthName "Private Web Page"
    AuthUserFile /etc/apache2/.okusers
    Require valid-user
</Location>
```

In bash: `touch /etc/apache2/.okusers`
`htpasswd2 /etc/apache2/.okusers user1`

In browser: `http://localhost/~user1/`

Übung 2

In `user.conf`: `alias /doc/ /usr/share/doc/`

```
<Location /doc>
    Options +Indexes
    order deny,allow
    allow from all
    AuthType Basic
    AuthName "Private DOCS"
    AuthUserFile /etc/apache2/.okusers
    # AuthGroupFile /etc/apache2/.okgroups
    Require valid-user
    # Require user jennie otto
```

```
# Require group buero admin
# Satisfy any
# Satisfy all
</Location>
```

In bash:

```
touch /etc/apache2/.okusers
htpasswd2 /etc/apache2/.okusers jennie
htpasswd2 /etc/apache2/.okusers paul
htpasswd2 /etc/apache2/.okusers marie
htpasswd2 /etc/apache2/.okusers hans
htpasswd2 /etc/apache2/.okusers otto
echo "admin: hans otto" >> /etc/apache2/.okgroups
echo "buero: jennie marie" >> /etc/apache2/.okgroups
echo "dienst: paul hans" >> /etc/apache2/.okgroups
```

In browser: `http://localhost/doc/`

INDEXES

- Tip:
- Index-Direktiven funktionieren nicht innerhalb <Location> *container*.
 - Benutzen Sie <Directory>
 - Alias-Pfade können über <Directory> realen Verzeichnissen zugeordnet werden.
 - Options FollowSymLinks kann nicht innerhalb <Location> *container* benutzt werden.

Beispiel 1:

Kopieren Sie das ganze Verzeichnis linuxbuch vom Dozentenrechner nach Verzeichnis: /srv/www/htdocs/linuxbuch.

```
mkdir /srv/www/htdocs/linuxbuch
cp xxxxxx/linuxbuch/* /srv/www/htdocs/linuxbuch
```

In user.conf:

```
<Directory /srv/www/htdocs/linuxbuch>
    Options +Indexes
    DirectoryIndex dummy
    AddDescription "PDF Datei" .pdf .PDF
    AddDescription "<Font color=red>GIF Bilder</Font>" .gif .GIF
    AddDescription "<Font color=green>===START===</Font>" online.html
```

----- Erweiterte Index Features -----

```
IndexOptions      +FancyIndexing          \
                  +FoldersFirst           \
                  +IconsAreLinks          \
                  +SuppressSize           \
                  +SuppressLastModified   \
                  +ScanHTMLTitles         \
                  +NameWidth=*            \
                  +DescriptionWidth=*
HeaderName header.html
ReadmeName footer.html
IndexIgnore header.html footer.html *.txt *.log
</Directory>
```

Hinweis: Der '\ ' am Ende der Zeile zeigt an, daß die Anweisung auf der Folgezeile fortgesetzt wird. Danach darf nur ein <Enter> folgen. Kein Leerzeichen, Tabulator oder irgendein anderes Zeichen.

header.html

```
<HTML>
  <BODY background="green_paper.gif">
    <!-- KOMENTARE SIND HIER -->
    <H1><FONT color=red><CENTER> L I N U X - B U C H <BR>
    Inhalt</CENTER></FONT></H1>
    <IMG src=penguin.gif>
    <H3><FONT color=blue>Letzte Liste von LinuxBu.ch Dateien
    </FONT></H3>
    <BR>
    <HR><HR><HR>
  </BODY>
</HTML>
```

footer.html

```
<B><I>Bitte schreiben Sie Ihre Kommentare durch email:
<a href="mailto:michel@linuxint.com">michel@linuxint.com
</a></I></B>
```

Beispiel 2: (Kopieren Sie sich die Zeilen aus diesem Text)

Installiere das Paket opensuse-manual_de (wenn es nicht bereits installiert ist)

In user.conf:

```
alias /opensuse-buch/ /usr/share/doc/manual/opensuse-manual_de/manual/
<Directory /usr/share/doc/manual/opensuse-manual_de/manual/>
  Allow from all
  Options +Indexes
  DirectoryIndex dummy
  IndexOptions      +FancyIndexing      \
                   +FolderFirst         \
                   +IconsAreLinks        \
                   +SuppressLastModified \
                   +ScanHTMLTitles       \
                   +NameWidth=*          \
                   +DescriptionWidth=*
</Directory>
```

In *Browser*: <http://localhost/opensuse-buch/>

AllowOverride

Direktiven: AccessFileName .htaccess

Beispiel 1:

(Hinweis: Verfügbar in <Directory> aber nicht in <Location>)

In bash:

```
mkdir /srv/www/htdocs/override/
cp /usr/share/doc/packages/bash/bashref.html /srv/www/htdocs/override/
```

In *browser*: <http://localhost/override/> Datei ist sichtbar im Index

In user.conf:

```
<Directory /srv/www/htdocs/override>
  AllowOverride Indexes
</Directory>
```

In bash: touch /www/override/.htaccess
 vi /www/override/.htaccess

In .htaccess: DirectoryIndex bashref.html

In browser: http://localhost/override/
 Bash page (manual) wird angezeigt.

Virtual Hosts

Erzeugen Sie das *Virtual Host's ROOT* Verzeichnis und *log files* Verzeichnis:

```
mkdir /srv/www/www.virtual1.de
mkdir /var/log/apache2/www.virtual1.de/
```

- Tragen Sie `www.virtual1.de virtual1.de virtual1` in `/etc/hosts` ein
 z.B.

```
echo "$(hostname -i) www.virtual1.de virtual1.de virtual1" >> /etc/hosts
```
- Füllen Sie *Virtual host* mit *home pages*:

```
cp -v /usr/share/doc/packages/rpm/RPM-Tips/* /srv/www/www.virtual1.de/
```
- In `user.conf`

```
NameVirtualHost 192.168.71.27
<VirtualHost 192.168.71.27>
    ServerName www.virtual1.de
    ServerAlias virtual1.de virtual1
    DocumentRoot /srv/www/virtual1.de
    <Directory /srv/www/virtual1.de>
        Allow from all
        DirectoryIndex RPM-Tips.html
        # Platz für extra Direktiven für dieses Verzeichnis
    </Directory>
    TransferLog /var/log/apache2/www.virtual1.de/access_log
    ErrorLog /var/log/apache2/www.virtual1.de/error_log
</VirtualHost>
```

Proxy deaktivieren für: `www.virtual1.de virtual1.de virtual1`

Reload oder *restart* Apache: `rcapache2 reload`
 oder `rcapache2 restart`

Kontrolle der Liste *virtual hosts*: `httpd -S`

CGI

```
In bash:      mkdir /srv/www/www.cgiptest.net/
               mkdir /var/log/apache2/www.cgiptest.net/
               echo "192.168.10.50 www.cgiptest.net" >> /etc/hosts
```

In user.conf:

```
NameVirtualHost 192.168.10.50:80
<VirtualHost 192.168.10.50:80>
    ServerName www.cgiptest.net
    ServerAlias cgiptest.net
    DocumentRoot /srv/www/www.cgiptest.net
    <Directory /srv/www/www.cgiptest.net>
        Allow from all
        Options +execCGI
        AddHandler cgi-script .cgi
        DirectoryIndex cgiptest.cgi
    </Directory>
    TransferLog /var/log/apache2/www.cgiptest.net/access_log
    ErrorLog /var/log/apache2/www.cgiptest.net/error_log
</VirtualHost>
```

In bash:

```
touch /srv/www/www.cgiptest.net/cgiptest.cgi
chmod 755 /srv/www/www.cgiptest.net/cgiptest.cgi
```

To run root commands from the cgi you need to use sudo:

```
> visudo
    username ALL=NOPASSWD:/path/to/root/commmand parameters
```

Übung: Diese Übung kombiniert 3 themen: CGI, VirtualHost und Authentifizierung

Eine authentifizierte VirtualHost der erlaubt eine reboot von system durch zu führen.

```

Addr:                www.reboot.com
Login Name:          admin
CGI Script Name:      reboot.cgi
Web Space Verzeichnis: /srv/www/www.reboot.com/

```

Schritte dass zu realisieren:

- 1) Eine web-space (DocumentRoot verzeichnis)für dass cgi-script erzeugen:

```
mkdir /srv/www/www.reboot.com
```

- 2) Eine Log dateien Verzeichnis erzeugen:

```
mkdir /var/log/apache2/www.reboot.com
```

- 3) Neue Authentifizierung benutzer (admin) anlegen

```
htpasswd2 /etc/apache2/.okusers admin
```

- 4) Die IP für www.reboot.com in /etc/hosts eintragen

```
echo "192.168.10.50    www.reboot.com" >> /etc/hosts
```

- 5) Die Erlaubnis für wwwrun Benutzer die Befehl '/sbin/reboot' zu führen:

```
visudo
    wwwrun    ALL=NOPASSWD:/sbin/reboot
```

- 6) Die CGI script in DocumentRoot Verzeichnis anlegen:

(Inhalt von /srv/www/www.reboot.com/reboot.cgi)

```
#!/bin/bash
echo "Content-type: text/html"
echo
sudo /sbin/reboot
```

- 7) Die Ausführen Zugriffsrechte zu der CGI Script eingeben

```
chmod 755 /srv/www/www.reboot.com/reboot.cgi
```

- 8) Die VirtualHost configuration (in /etc/apache2/conf.d/user.conf)anlegen:

```

<VirtualHost 192.168.10.50>
    ServerName    www.reboot.com
    DocumentRoot  /srv/www/www.reboot.com
    <Directory    /srv/www/www.reboot.com>
        Allow from all
        DirectoryIndex reboot.com
        Options +execCGI
        AddHandler cgi-script .cgi
        AuthType Basic
        AuthName "REBOOT CGI Script"
        AuthUserFile /etc/apache2/.okusers
        Require user admin
    </Directory>
    TranferLog /var/log/apache2/www.reboot.com/access_log
    ErrorLog    /var/log/apache2/www.reboot.com/error_log
</VirtualHost>

```

- 9)Reload oder restart Apache2:

```

rcapache2 reload
oder rcapache2 restart

```


-----cgitest.cgi-----

```
#!/bin/sh      <----- Ab Hier, diese folgenden drei Zeilen müssen hier sein
echo Content-type: text/html
echo

# This above header and empty echo after it is VERY important.
# Otherwise Apache shows an error : Content Header missing
# The HTML code enclosed in quotes is also very important.....

echo "<HTML><HEAD><TITLE>CGI TEST Seite</TITLE></HEAD>"
echo "<BODY bgcolor=lightgrey>"

echo "<H1><CENTER>SYSTEM INFO</CENTER></H1><P>"
echo "<H3>Ihre IP Addr. ist :</H3><H2>$REMOTE_ADDR</H2><BR>"

#---Display content of /etc/fstab
echo "<HR><H3>System /etc/fstab</H3><BR><pre>"
/bin/cat /etc/fstab
echo "</pre><BR><HR>"

#---Display free space of all mounted disks in Linux -----
echo "<Center><H1>DISK SPACE</H1></Center><BR><pre>"
/bin/df -h
echo "</pre><HR>"

#---Display all the system Processes ---
echo "<Center><H3>PROCESS INFO</H3></Center><BR><pre>"
/bin/ps -eo pid,ppid,user,nice,args --sort user

#---Display Apache's user name and group
echo "</pre><BR><HR>"
echo "<H3>Apache ist $(id -nu) Benutzer. Er ist Mitglied von Gruppe $(id -ng)</H3>"
echo "<BR><HR>"
echo "<Center><H3>Inhalt von /etc Verzeichnis</H3></Center><PRE>"
ls -l /etc
echo "</pre><BR><HR>"

echo "<Center><H3>Aktive Ports</H3></Center><PRE>"
netstat -ltupn
echo "</pre><BR><HR>"
echo "</BODY></HTML>"
```

----- Erzeugen von *web access* Protokollen mit Webalizer -----

Notiz: der program `awstats` ist auch eine sehr gut Statistics Program für Web sogar mail u.s.w.

- Stellen Sie sicher dass der Packet 'webalizer' installiert ist.

- Run the commands:

```
echo "192.168.10.50 stats.cgitest.net" >> /etc/hosts
mkdir /srv/www/stats.cgitest.net
chown wwwrun. /srv/www/stats.cgitest.net
```

- ```
- in user.conf
```

```
#----- Statistics of www.cgitest.net -----
<VirtualHost 192.168.10.50>
 ServerName stats.cgitest.net
 DocumentRoot /srv/www/stats.cgitest.net
 <Directory /srv/www/stats.cgitest.net>
 DirectoryIndex webalize.html
 AuthName LogFiles
 AuthType Basic
 AuthUserFile /etc/apache2/.okusers
 require user jennie paul
 </Directory>
 ErrorLog /var/log/apache2/www.cgitest.net/stats_error.log
 TransferLog /var/log/apache2/www.cgitest.net/stats_access.log
</VirtualHost>
```

```
In /srv/www/stats.cgitest.net/webalize.html:
```

```
<html>
<head>
 <Title> Web Access Statistics</Title>
</head>
<body>

<center>
Webalize
JETZ !!!</center>

</body>
</html>
```

```
IN /srv/www/stats.cqitest.net/webalize.cgi:
```

```
#!/bin/sh
#--- Starts the webalizer for the present log file-----
echo "Content-Type: text/html"
echo
/usr/bin/webalizer -Q -n www.cgitest.net -o . \
 /var/log/apache2/www.cgitest.net/access.log &>/dev/null
cat index.html
```

**Secure HTTP: HTTPS Konfiguration:**

Kontrolle, ob das ssl-Modul in folgende Datei eingetragen ist:

```
/etc/sysconfig/apache2
APACHE_MODULES= ssl
APACHE_SERVER_FLAGS="SSL"
```

In /etc/apache2/ssl-global.conf make sure the following directives are correct:

```
#----- SSL Global Directives -----
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl .crl
SSLPassPhraseDialog builtin
SSLSessionCache shmcb:/var/lib/apache2/ssl_scache(512000)
SSLSessionCacheTimeout 600
SSLMutex default
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
```

Make a copy of the SSL configuration template:

```
cp /etc/apache2/vhosts.d/vhost-ssl.template \
 /etc/apache2/vhosts.d/vhost-ssl.conf
```

Create the SSL documents directory: /srv/www/ssl-docs/

```
mkdir /srv/www/ssl-docs/suse
cp -a /usr/share/doc/manual/opensuse-manual_de/manual/* \
 /srv/www/ssl-docs/suse/
```

Adjust the Apache SSL virtual host configuration file:

```
/etc/apache2/vhosts.d/vhost-ssl.conf
#----- SSL Virtual Host -----
<VirtualHost _default_:443>
 DocumentRoot /srv/www/ssl-docs/
 SSLEngine on
 SSLCipherSuite \
 ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
 SSLCertificateFile /etc/apache2/ssl.crt/server.crt
 SSLCertificateKeyFile /etc/apache2/ssl.key/server.key
 <Files ~ "\.(cgi|shtml|phtml|php3?)$" >
 SSLOptions +StdEnvVars
 </Files>
 SetEnvIf User-Agent ".*MSIE.*" \
 nokeepalive ssl-unclean-shutdown \
 downgrade-1.0 force-response-1.0
 <Directory /srv/www/ssl-docs/manual>
 allow from all
 Options +Indexes
 </Directory>
```

**Erzeugen eines temporären Zertifikates für SSL:**

Sie bauen Ihre eigenen mit folgenden Befehlen:

(3 Keys will be created in the current directory)

```
openssl genrsa 1024 > server.key
openssl req -new -key server.key -out server.csr (The ComonName(CN) should be the FQDN of server.)
openssl req -new -x509 -days 1460 -key server.key -in server.csr -out server.crt
```

Zum Testen des Inhaltes der Zertifikate:

```
openssl x509 -noout -text -in server.crt
```

Kopieren Sie beide folgendermaßen:

```
cp server.key /etc/apache2/ssl.key/
cp server.crt /etc/apache2/ssl.crt/
```

## ----- Überprüfen, ob die PHP-Module arbeiten -----

1) Erzeugen Sie eine Testdatei (php) im vorhanden Verzeichnis DocumentRoot.

```
echo "<?php" >/srv/www/htdocs/phptest.php
echo "<info()" >>/srv/www/htdocs/phptest.php
echo "?>" >>/srv/www/htdocs/phptest.php
```

2) In *Browser*:

`http://localhost/phptest.php`

Should give a full *home page* on the configuration of the PHP module.

## phpMyAdmin

1) Install the phpMyAdmin package

2) Move the Directory phpMyAdmin the SSL Documents area:

```
mv /srv/www/htdocs/phpMyAdmin /srv/www/ssl-docs/
```

```
#====Redirection of phpMyAdmin =====
<Directory /srv/www/ssl-docs/phpMyAdmin/>
Options -Indexes
AllowOverride All
DirectoryIndex index.php
Allow from All
AuthName "MySQL Admin"
AuthType Basic
AuthUserFile /etc/apache2/.okusers
Require valid-user
Satisfy all
</Directory>

#====Umleitung von www.webmail.com =====
<Directory /srv/www/ssl-docs/squirrelmail>
Allow from all
php_flag register_globals off
Options Indexes FollowSymLinks
DirectoryIndex index.php
<Files configtest.php>
order deny,allow
deny from all
allow from 127.0.0.1
</Files>
</Directory>
#-----
ErrorLog /var/log/apache2/https_error.log
TransferLog /var/log/apache2/https_access.log
</VirtualHost>
```

**Secure Webmail Installation (Squirrelmail)**(Benutzung des `mod_rewrite` Modul)

- Kontrolle, ob das `ssl`, `rewrite` und `php5` Modul in folgende Datei eingetragen sind:

`/etc/sysconfig/apache2``APACHE_MODULES= ssl rewrite php5 .....`

- In `bash`:

`mkdir /srv/www/ssl-docs``echo "192.168.10.50 webmail.net www.webmail.net" >> /etc/hosts`

- Squirrelmail packet installieren und nach SSL Dokument Bereich verschieben:

`mv /srv/www/htdocs/squirrelmail /srv/www/ssl-docs/squirrelmail`

- In `user.conf`

```
===== www.webmail.net =====
<VirtualHost 192.168.10.50:80>
 ServerName www.webmail.net
 ServerAlias webmail.net
 RewriteEngine on
 RewriteCond %{HTTPS} !^on$ [NC]
 RewriteRule . https://www.webmail.net/squirrelmail/ [L]
</VirtualHost>
```

Die Aufrufe werden zum `https` `VirtualHost` umgelenkt, der oben durch das Unterverzeichnis (`/squirrelmail`) als squirrelmail web page angegeben wurde.

**Preparing a web space for:**

**User:** marie  
**Web site:** www.kunden1.de

**Create the ftp user marie**

```
useradd -d /srv/www/www.kunden1.de marie
passwd marie
```

**Create a web space and give it to marie**

```
mkdir /srv/www/www.kunden1.de
chown /srv/www/www.kunden1.de
```

**Create a log directory for www.kunden1.de**

```
mkdir /var/log/apache2/www.kunden1.de
```

**Start creating the Virtual Host (using bash commands):**

```
echo "<VirtualHost 192.168.10.50:80>" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " ServerName www.kunden1.de" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " DocumentRoot /srv/www/www.kunden1.de" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " <Directory /srv/www/www.kunden1.de" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " Allow from all" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " DirectoryIndex /srv/www/www.kunden1.de" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " </Directory>" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " TransferLog /var/log/apache2/www.kunden1.de" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo " ErrorLog /var/log/apache2/www.kunden1.de" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
echo "</VirtualHost>" >>/etc/apache2/vhosts.d/www.kunden1.de.conf
```

**IDS Images Galerie - CGI Installation**

1 ) Installieren Sie folgende Pakete von den Original SuSE CDs.

```
perl-Image-Info
perl-PerlMagick (Benötigt ImageMagick Paket)
jpeg (benutzt Programm jpegtran)
```

2 ) Entpacken Sie das ISD System `ids.tar.gz` in das Verzeichnis:

```
/srv/www/htdocs/ids/
```

3 ) Konfigurieren des *Virtual Host* in Apache2 folgendermaßen:

```
<VirtualHost 192.168.100.70>
 ServerName pictures.linux.site
 DocumentRoot /srv/www/htdocs/ids
 <Directory /srv/www/htdocs/ids>
 order deny,allow
 allow from all
 DirectoryIndex index.cgi
 Options +ExecCGI +FollowSymLinks
 AddHandler cgi-script .cgi
 AllowOverride AuthConfig Limit
 </Directory>
</VirtualHost>
```

4 ) Kontrolle ob Rechte der ISD Verzeichnisse richtig sind:

```
drwxr-xr-x 11 wwwrun root 456 Apr 27 2002 .
drwxr-xr-x 12 root root 1712 Feb 13 19:44 ..
drwxr-xr-x 3 root root 208 Apr 27 2002 admin
drwxr-xr-x 2 wwwrun root 48 Mar 23 2002 album-data
drwxr-xr-x 2 wwwrun root 48 Mar 23 2002 albums
-rw-r--r-- 1 wwwrun root 737 Mar 23 2002 ids.conf
-rw-r--r-- 1 root root 39426 Dec 16 2001 idsShared.pm
drwxr-xr-x 2 wwwrun root 48 Mar 23 2002 image-cache
-rwxr-xr-x 1 root root 43790 Oct 6 2001 index.cgi
drwxr-xr-x 2 root root 392 Apr 27 2002 localizations
drwxr-xr-x 2 wwwrun root 80 Apr 28 2002 logs
drwxr-xr-x 2 wwwrun root 184 Apr 27 2002 postcomment
-rwxr-xr-x 1 root root 3436 Oct 6 2001 previewmaker.pl
drwxr-xr-x 3 root root 256 Apr 27 2002 site-images
-rw-r--r-- 1 wwwrun root 393 Dec 16 2001 site_news.txt
drwxr-xr-x 9 root root 232 Apr 27 2002 themes
```

5 ) Erzeugen des IDS's Administrator *password*

```
htpasswd2 -c /etc/apache2/htpasswd.ids admin
```

6 ) Kopieren des *images*-Verzeichnisses nach `/srv/www/htdocs/ids/albums`

als Unterverzeichnis, das *root*-Verzeichnis darf nur Verzeichnisse enthalten.

Hinweis: Vergewissern Sie sich, daß der Eigentümer für Verzeichnisse und Dateien `wwwrun` ist.

7) Test des lokalen Zugriffes durch Eintrag in `/etc/hosts`:

```
192.168.100.70 pictures.linux.site
```

8) *Restart* Apache

```
rcapache2 restart
```

9) In *browser* eingeben:

```
http://pictures.linux.site
```

**httpd2-prefork command Options****Syntax:** `/usr/sbin/httpd2 -options`**Optionen:**

- D *name* Definiere Name für Benutzung in <IfDefine *name*> Direktiven. Dieser wird benutzt, um verschiedene *server global* Einstellungen unterscheiden zu können und wählt die richtige Konfiguration beim Start des Apache aus.
- d *ServerRootDir* Angabe eines alternativen ServerRoot Verzeichnis.
- f *ConfigFile* Angabe einer alternativen Konfigurationsdatei.(ServerConfigFile)
- C *Directive* Führt diese Direktive vor dem Lesen der Konfigurationsdateien aus.
- c *Directive* Führt diese Direktive nach dem Lesen der Konfigurationsdateien aus.
- v Zeigt Apache's Versionsnummer
- h Zeigt die möglichen Kommandozeilen-Optionen.
- l (klein L) Zeigt die *compiled-in* Module
- L (groß L) Zeigt die *core* (Apache-Kern) Konfiguration
- s Zeigt *virtual hosts* Einstellungen
- t Testet den Syntax der Konfigurationsdateien; sonst keine Wirkung.

**Controlling the started Prozesse und *clients* on Apache****# prefork MPM (Multi Processing Modul)**

# StartServers ..... Anzahl der zu startenden *server* Prozesse  
 # MinSpareServers ..... Minimum der freien *server* Prozesse, die gehalten werden  
 # MaxSpareServers ..... Maximum der freien *server* Prozesse, die gehalten werden  
 # MaxClients ..... Maximale Anzahl der *server* Prozesse, die gestartet werden dürfen  
 # MaxRequestsPerChild .. Maximale Anzahl der Anfragen (Kind-) je *server* Prozess

```
<IfModule prefork.c>
 StartServers 5
 MinSpareServers 5
 MaxSpareServers 10
 MaxClients 20
 MaxRequestsPerChild 0
</IfModule>
```

**# pthread MPM**

# StartServers ..... Anzahl der zu startenden *server* Prozesse  
 # MaxClients ..... Maximale Anzahl der *server* Prozesse, die gestartet werden dürfen  
 # MinSpareThreads ..... Minimum der freien *worker threads*, die gehalten werden  
 # MaxSpareThreads ..... Maximum der freien *worker threads*, die gehalten werden  
 # ThreadsPerChild ..... Feste Anzahl der *worker threads* je *server* Prozess  
 # MaxRequestsPerChild .. Maximale Anzahl der Anfragen je *server* Prozess

```
<IfModule worker.c>
 StartServers 2
 MaxClients 150
 MinSpareThreads 25
 MaxSpareThreads 75
 ThreadsPerChild 25
 MaxRequestsPerChild 0
</IfModule>
```



## **Apache in Debian basierenden Distributionen**

- Auswahl Modul und *config files*

### **Extras:**

**Writing Apache configuration scripts**  
(Siehe Bash Shell Kursunterlagen)

**LAMP - Linux-Apache-MySQL-PHP**

1) Installieren Sie folgende Pakete:

apache2	mysql	phpMyAdmin
apache2-prefork	mysql-client	php5-mysql
apache2-mod_php5	mysql-shared	

phpMyAdmin kann entweder über einen *VirtualHost* oder direkt mit dem *main server* benutzt werden.

**Via VirtualHost**

2a) Entweder einen Verweis des *Virtualhost* zum `/srv/www/htdocs/phpMyAdmin` Verzeichnis in den Direktiven:

```
NameVirtualHost 192.168.71.10 (nur wenn er nicht bereits existiert)
<VirtualHost 192.168.71.10>
 ServerName phpmyadmin.linux.site
 DocumentRoot /srv/www/htdocs/phpMyAdmin
 <Directory /srv/www/htdocs/phpMyAdmin>
 Allow from all
 DirectoryIndex index.php
 </Directory>
</VirtualHost>
```

2b) Oder fügen Sie die lokale IP und den server-Name in `/etc/hosts` ein

```
zB. 129.168.71.10 phpmyadmin.linux.site
```

**Mittels Main Apache server:**

(`http://localhost/phpMyAdmin/`)

4) Starten Sie den *mysql daemon* mit dem Befehl:

```
rcmysql start
```

5) Ändern Sie das *root password* in MySQL mit folgendem Befehl:

```
mysqladmin -u root password newpasswd
mysqladmin -p -u root -h localhost password newpasswd
```

Drücken Sie **<Enter>** wenn nach dem *password* gefragt wird.

6) Ändern Sie das *root password* und die Absolute URI in

`/srv/www/htdocs/phpMyAdmin/config.inc.php`

zB.

```
$cfg['PmaAbsoluteUri'] = 'http://phpmyadmin.linux.site';
```

oder

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';
```

```
$cfg['Servers'][$i]['user'] = 'root'; // MySQL user
$cfg['Servers'][$i]['password'] = 'newpasswd'; // MySQL password
```

7) *Start* oder *Restart* Apache Server.

```
rcapache2 restart
```

8) Starten Sie einen *browser* und schreiben Sie die Adresse:

```
http://phpmyadmin.linux.site oder http://localhost/phpMyAdmin/
```

9) Schützen Sie den Zugriff auf phpMyAdmin:

**Via VirtualHost**

In dem bereits konfigurierten VirtualHost *instructions block*:

```
<Directory /srv/www/htdocs/phpMyAdmin>
 Allow from all
 DirectoryIndex index.php
 AuthType Basic
 AuthName "mysql administration"
 AuthUserFile "/etc/apache2/.okmysql"
 require user marie
 satisfy all
</Directory>
```

### Via Haupt Apache server

In einer *apache* Konfigurationsdatei:

```
<Location /phpMyAdmin>
 Allow from all
 DirectoryIndex index.php
 AuthType Basic
 AuthName "mysql administration"
 AuthUserFile "/etc/apache2/.okmysql"
 require user marie
 satisfy all
</Location>
```

- Geben Sie ein *password* für marie (als *mysql administrator*)

```
touch /etc/apache2/.okmysql
htpasswd2 /etc/apache2/.okmysql marie
```

## WAMP-Windows-Apache-MySQL-PHP

### Installation Apache2, PHP4 und MySQL in Windows2000/XP

Hier finden Sie die Website, von der Sie das Installationspaket mit allen Komponenten von Apache2, PHP4, Mysql, PhpMyAdmin, Webalizer und Extras finden:

<http://www.apachefriends.org/de/xampp-windows.html>

Die letzte Variante beim schreiben dieses Dokumentes war: V. 1.4.11

Starten Sie Datei und alles wird automatisch erledigt.

### Manuelle Installation für ganz Mutige!

Hinweis: *This installation cuts corners and does things that might not be necessary. I've used this method and got a working setup.*

Für andere Alternativen schauen Sie bitte hier nach:

<http://www.php.net/manual/en/install.apache2.php>

### Installation APACHE2

- 1) Download the latest stable Apache für Windows from the internet:

<http://www.apache.org>

It's a .msi file und can be installed immediately

- 2) In my case, I kept the recommended install path of Apache:

C:/programm(e)/Apache Group/Apache2/

### Installation MYSQL

- 1) Download the MySQL windows installer from the Internet:

<http://dev.mysql.com/downloads/mysql/4.0.html>

It's a ZIP file

- 2) Unpack the ZIP file into a temporary Verzeichnis und run the program SETUP.EXE

I installed it in C:\mysql Verzeichnis

### Installation PHP4(as apache module)

- 1) Download the ZIP file from:

<http://www.php.net/downloads.php>

- 2) Unzip the files into C:/php Verzeichnis

- 3) copy all the .dll files from C:/php/dlls und C:/php/sapi to C:/php Verzeichnis

### Installation phpMyAdmin

- 1) Download the latest version of phpMyAdmin ZIP file from:

[http://www.phpmyadmin.net/home\\_page/](http://www.phpmyadmin.net/home_page/)

- 2) Unzip the file into the Verzeichnis:

C:\Programme\Apache Group\Apache2\htdocs\phpmyadmin\

- 3) Editieren Sie the config.inc.php in the phpmyadmin Verzeichnis und:

- add the proper full URI of the phpMyAdmin:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin/';
```

Optional:(If you leave the root *password* blank in MySQL then leave it blank here also)

- add the MySQL root (administrator) *password* in clear text in:

```
$cfg['Servers'][$i]['user'] = 'root'; // MySQL user
```

```
$cfg['Servers'][$i]['password'] = ''; // MySQL password
```

(only needed with 'config' auth\_type)

## CONFIGURATION

1) Editieren Sie die Datei `httpd.conf` und:

### Das Wichtigste in der Übersicht:

Datei (Verzeichnis)	Bedeutung
<code>\(mini)xampp\apache\logs</code>	Logverzeichnis für Apache und PHP.
<code>\(mini)xampp\cgi-bin\</code>	Das CGI-BIN-Verzeichnis.
<code>\(mini)xampp\apache\conf\httpd.conf</code>	Die zentrale Konfigurations-Datei für den Apache.
<code>\(mini)xampp\apache\bin\php.ini</code>	Die zentrale Konfigurations-Datei für PHP mit MOD_PHP
<code>\(mini)xampp\htdocs\</code>	Das zentrale Dokumentenverzeichnis für HTML, PHP, CGI ..
<code>\(mini)xampp\install\</code>	Notwendig für das erste Setup ...
<code>\(mini)xampp\mysql\</code>	Der Datenbankserver MySQL
<code>\(mini)xampp\perl\</code>	Das Perl-Verzeichnis.
<code>\(mini)xampp\php\</code>	php.exe + dlls + pear usw.
<code>\(mini)xampp\phpmyadmin\config.inc.php</code>	Die Konfigurations-Datei für phpMyAdmin.
<code>\(mini)xampp\tmp</code>	Temp. Ordner für PHP-Uploads, Sessions usw.
<code>\(mini)xampp\moddav</code>	MOD-DAV Beispielordner.

- Ändern Sie folgende Einstellungen:

```
DocumentRoot "C:\Programme\Apache Group\Apache2\htdocs"
DirectoryIndex index.html index.html.var index.php
```

- Fügen Sie folgende Einstellungen hinzu:

```
LoadModule php4_module c:/php/php4apache2.dll
LoadFile C:/php/php4ts.dll
AddType application/x-httpd-php .php .php3 .php4 .phtml
```

2) Editieren Sie die Datei `C:\php\php.ini`

- Wenn erforderlich, ändern Sie den Eintrag `extension_dir` auf Ihr Verzeichnis `php-install-dir`, oder wo Sie Ihre '`php_*.dll`' Dateien zu finden

zB.: `c:\php`

Der Eintrag ist ca. bei Zeile 442 zu finden

- Setzen Sie '`doc_root`' zum Verzeichnis Ihres Webserver `document_root`.

In unserem Fall: `C:/Programme/Apache Group/Apache2/htdocs`

3) Erzeugen der Datei `C:\MY.CNF` und fügen Sie folgende Zeilen ein:

```
[mysqld]
basedir=C:/mysql/
datadir=C:/mysql/data/
```

**Tip:** Siehe MY.CNF Beispiel bei XAMPP weiter unten

## Starten WAMP

1) Öffnen Sie das DOS Fenster ( `cmd` ) und geben Sie folgenden Befehl ein:

```
C:\mysql\bin\mysqld.exe
```

Wenn alles *ok* ist, dann kan ein *icon* erzeugt werden, um wahlweise über den 'Autostartordener' oder per Hand zu starten.

2) Starten Sie Apache mittels Windows Startmenü

3) Starten Sie einen *browser* und und geben Sie folgende Adresse ein:

```
http://localhost/phpmyadmin/
```

Nun haben Sie eine phpMyAdmin *web site* mit voller Kontrolle über die MySQL Datenbank.  
Damit bestätigen Sie, daß Apache, PHP und MySQL alle funktionieren...so weit, so gut!