

# Apache

## **Was soll dieses Dokument bringen?**

Die Fähigkeit einen Apacheserver zu installieren zu konfigurieren und so seinen eigenen Bedürfnissen anzupassen.

## **Was wird nicht beschrieben?**

Compilieren des Apache Quellcodes.  
Performance Tuning.

## **Voraussetzungen**

Apache Webserver Version 1.3.x  
SuSE Linux 6.4 und höher  
Grundkenntnisse des Dateisystems in Linux.

## **Installation**

### **... mit RPM- Paketen:**

Unter SuSE- Linux kann das Paket unter Yast ausgewählt werden und wird dann automatisch installiert und in seinen Grundzügen konfiguriert.

In der Komandozeile geschieht die Installation durch den Befehl:

```
rpm -i [--force] [ --allfiles] <Dateiname.rpm>
```

## **Steuern des Servers**

SuSE- Linux: **rcapache** [-restart | -start | -stop]

Eigene Installationen:

**Starten** : <Pfad/zu/Apache >/httpd -

f<Pfad/zu/den/Configurationsdateien>/httpd.conf

z.B. : `usr/local/apache/httpd -f /usr/local/apache/conf/httpd.conf`

**Stoppen** des Servers: `kill -TERM `cat <Pfad zu Apache Log- Dateien>/httpd.pid``

z.B. : `kill -TERM `cat /usr/local/apache/logs/httpd.pid``

## Grundsätzliche Konfigurationsbefehle

Befehl	Beschreibung
Alias <URL> <Verzeichnispfad>	Erlaubt es eine url an ein anderes Verzeichnis, als eines unter dem „DocumentRoot“, zu binden.
AccessFileName <Dateiname> [Dateiname] ...	Erlaubt es Verzeichnisse mit Paßwörtern zu sichern.
Allow from <Hostname   IP> [Hostname   IP] ...	Gibt an wem, in Verbindung mit der <Directory>-Anweisung, der Zugang zu einem Verzeichnis gewährt wird. Hostname kann dabei sein: all : Alle werden zugelassen. (Teilweise) IP- Adresse, (Teilweise) Domainnamen Netzmaske : z.B. 10.1.0.0/255.255.0.0
BindAddress <IP  Domainname>	Bindet den laufenden Server an eine IP- Adresse oder einem Domainnamen. Wird „*“ benutzt so werden alle IP- Adressen und Domainnamen gebunden.
DocumentRoot <Verzeichnispfad>	Setzt das Basisverzeichnis auf das vom Webserver aus zugegriffen wird. Hier liegen normalerweise alle Internetseiten.
deny from <Hostname   IP> [Hostname   IP] ...	Ist ähnlich der allow- Anweisung. Nur, dass hier angegeben wird wem der Zugang verwehrt bleibt.
DirectoryIndex <Dateiname> [Dateiname] ...	Die angegebenen Dateien werden immer dann aufgerufen, wenn nur das Verzeichnis angegeben wird. http://www.bsp.de/ ist das gleiche wie http://www.bsp.de/index.html
ErrorDocument <Error- Code> <Dateipfad>	Zum festlegen eines eigenen Dokumentes zum Anzeigen von Fehlern.
Group <Unix- Benutzergruppe > User <Unix- User>	„User“ und „Group“ bestimmen jeweils die Gruppe und den User, den alle Aktivitäten, die durch den Webserver zustande kommen, zugeordnet werden. Das heißt: Wenn ein X- belibieger Mensch auf dem Server via z.B. cgi Dateien erstellt dann haben diese diese Benutzer- und Gruppen- Zugehörigkeiten.
KeepAlive <on  off>	Soll die Verbindung nach einer bestimmten Zeit automatisch beendet werden?
KeepAliveTimeout <Sekunden>	So viele Sekunden wird gewartet bis die Verbindung vom Server zum Client getrennt wird.
Listen <IP- Adresse[:Port]>	Ähnlich wie BindAddress, nur hier können mehrere IP- Adressen an den Server gebunden werden. Es können durch die Angabe eines Ports auch mehrere Ports benutzt werden.
MaxKeepAliveRequests <Anzahl>	Hier wird die Anzahl der Maximalen Anfragen an den Server gesetzt wenn „KeepAlive“ auf „on“ geschaltet ist.
MaxClients <Anzahl>	Setzt die maximale Anzahl an Anfragen an den Server. Wenn mehr Anfragen ankommen werden diese in eine Warteschlange gesteckt.
MaxSpareServers <Anzahl>	Setzt die Maximale und Minimale Zahl an wartenden

MinSpareServers <Anzahl>	Klienten an den Server. Wenn mehr oder weniger Anfragen sind werden die entsprechenden Klienten abgelehnt. Eine Änderung ist nicht auf „normal“ besuchten Servern nötig. Außerdem funktionieren sie nicht aus Windows Plattformen.
MaxRequestsPerChild <Anzahl>	Gibt die maximale Anzahl der Anfragen pro verbundenen Client an.
NameVirtualHost <IP  Hostname[:Port]>	Wird für Namensbasierte- virtuelle- Hostkonfiguration gebraucht. Es <i>kann</i> ein Hostname verwendet werden. Es wird aber empfohlen, dass IP- Adressen benutzt werden. Spezifiziert für die angegebene Adresse einen virtuellen Host. Optional kann noch eine Portnummer angegeben werden.
order <Reihenfolge>	Bestimmt die Reihenfolge in der „allow“ und „deny“ abgearbeitet werden.
PidFile <Dateipfad>	In dieser Datei wird die „Pid“ des laufenden Webservers abgespeichert.
Port <Nummer>	<Nummer> gibt einen speziellen Port an, auf den der Webserver hören soll.
ScriptAlias <Url>- <Verzeichnispfad>	Verhält sich wie „Alias“. Nur, dass das Verzeichnis zusätzlich als cgi- Verzeichnis gekennzeichnet wird.
ServerAlias <Hostname>	Setzt alternative Namen. Wird mit <VirtualHost> benutzt.
ServerName <Domainname>	Setzt den Hostname des Servers. Kann vom realen Namen des Servers unterschiedlich sein. Es kann auch die IP- Adresse benutzt werden.
ServerPath <Webpfad>	Alle Aufrufe des primären Webservers auf diesen Pfad werden an einem Virtuellen Webserver umgeleitet. Wird mit <VirtualHost> benutzt.
ServerRoot <Verzeichnispfad>	Setzt den Basispfad zu den Server. Er enthält typischerweise die Unterordner conf/“ und logs/“ und htdocs/“.
ServerType <Typ>	Setzt die Art auf der Apache läuft. Entweder „standalone“ oder „inetd“. Es ist zu empfehlen „standalone“ zu nutzen, da „inetd“ nicht überall funktioniert.
StartServers <Anzahl>	Anzahl der beim Start zu startenden Klient- Prozesse.
Timeout <Sekunden>	Setzt die Zeit die der Apacheserver auf folgende Dinge wartet:  Die Zeit die gebraucht wird um eine GET- Anfrage zu bearbeiten.  Die Zeit zwischen dem Empfang zweier TCP- Pakete bei einer POST- oder PUT- Anfrage.  Die Zeit zwischen ACKs bei der Übertragung von TCP- Paketen.
<IfDefine <[!]Parameter> >	Wenn eine bestimmte Variable/ Parameter definiert

<code>&lt;/IfDefine&gt;</code>	ist und den Wert „true“ besitzt, dann werden die Anweisungen die hier aufgeführt werden ausgeführt.
<code>&lt;Directory &lt;Verzeichnispfad&gt; &lt;/Directory&gt;</code>	Alle in Anweisungen, die zwischen <code>&lt;Directory&gt;</code> und <code>&lt;/Directory&gt;</code> erfolgen werden auf das angegebene Verzeichnis und seine Unterverzeichnisse angewandt. Es können parallel auch sogenannte Wildcards oder auch Platzhalter benutzt werden. Dies sind z.B. „*“ oder „?“ Aber auch „[ ]“.
<code>&lt;Location &lt;URL&gt; &gt;&lt;/Location&gt;</code>	Ist ähnlich der <code>&lt;Directory&gt;</code> - Anweisung. Nur hier geschieht die Kontrolle auf Basis einer URL des Servers.
<code>&lt;VirtualHost &lt;IP   Hostname&gt;[:Port]&gt; &lt;/VirtualHost&gt;</code>	Anweisungen, die hier stehen gelten nur für einen bestimmten virtuellen Webserver. Innerhalb können alle Anweisungen benutzt werden, die auch für Hauptserver benutzt werden können.

## Zugriffskontrolle

Durch die Kombination von „order“, „deny“ und „allow“ kann für jedes beliebige Verzeichnis oder jede Datei festgelegt werden, wer von wo darauf zugreifen darf. Bei der Definition spielt die Reihenfolge von „deny“ und „allow“ eine wesentliche Rolle. So wird( in unserem Beispiel unten) **zuerst allen** der Zugang verwehrt. Um **dann die Ausnahmen** zu definieren. Hier dürfen dann alle Rechner, welche mit „meine-domain.com“ enden auf das Verzeichnis zugreifen. Auf die gleiche Weise können natürlich auch zuerst alle Rechner zugellassen werden, um danach bestimmte Rechner oder Subnetze auszuschließen.

```
<Location /status>
  SetHandler server-status
  # Die Reihenfolge von „deny“ und „allow“
  order deny,allow
  # wer darf nicht aufrufen?
  deny from all
  # und wer darf
  allow from .meine-domain.com
</Location>
```

## Mimetypes

Damit ein Browser auch etwas mit den ganzen verschiedenen Dateitypen etwas anfangen kann, Kann ihm über das Modul „mod\_mime.c“ gesagt werden, um was für ein Typ es sich bei einer bestimmten Datenamenserweiterung handelt. So kann der jeweilige Client darauf reagieren. Zum Beispiel nicht versuchen eine „\*.exe“ anzuzeigen.

```
<IfModule mod_mime.c>
    AddEncoding x-compress Z
    AddEncoding x-gzip gz tgz

    AddLanguage en .en
    AddLanguage de .de
    # Zuordnen eines bestimmten Zeichensatzes zu einer Sprache
    AddLanguage pl .po
    AddCharset ISO-8859-2 .iso-pl
    .
    .
    .
    AddLanguage cz .cz
</IfModule mod_negotiation.c>
    # Hier können Prioritäten für die verschiedenen
    # Sprachen vergeben werden.
    LanguagePriority en fr de el pl pt pt-br ca es sv
</IfModule>

# Setzt bestimmte Dateiendungen auf sogenannte „Mime-Types“.
# So können sie, wie oben erwähnt von den Browsern
# entsprechend behandelt werden.
<IfDefine PHP>
    AddType application/x-httpd-php3 .php3
    AddType application/x-httpd-php3-source .phps
    AddType application/x-httpd-php3 .phtml
</IfDefine>

AddType application/x-tar .tgz
AddType application/futuresplash .spl
AddType text/vnd.wap.wml wml
AddType text/vnd.wap.wmlscript wmls
AddType application/vnd.wap.wmlc wmlc
AddType application/vnd.wap.wmlscriptc wmlsc
Addtype image/vnd.wap.wbmp wbmp
AddType text/html .shtml
AddType application/x-schockwave-flash .swf

# Hier können spezifischen Dateiendungen "Handler"
# zugewiesen
# werden. So können z.B. cgi- Skripte als solche behandelt
# werden und werden nicht als reine Textform dargestellt.
AddHandler cgi-script .cgi
AddHandler server-parsed .shtml
AddHandler server-parsed .html
</IfModule>
```

## VirtualHosts

Es gibt verschiedene Möglichkeiten einen virtuellen Webserver einzurichten. Dies kann dadurch geschehen, dass noch eine Apache- Konfigurationsdatei angelegt wird und ein weiterer Server mit dieser Datei gestartet wird.

Es können aber auch innerhalb einer Konfigurationsdatei ein oder mehrere virtuelle Webserver definiert werden, die dann mit dem Hauptserver gestartet werden.

Für diesen Fall gibt es wiederum verschiedene Möglichkeiten so einen virtuellen Webserver zu definieren. Einmal auf Grundlage von IP- Adressen und dann noch auf Basis von Domainnamen. Dann kann man diese beiden Arten noch mischen, was aber hier nicht behandelt wird.

```
NameVirtualHost 192.168.0.34
```

```
<VirtualHost 192.168.0.34>
# setzen des Namen des virtuellen Webservers
ServerName sonne.alpha
# Verzeichnis, in dem die Dokumente zu finden sind
DocumentRoot "/usr/local/httpd/htdocs"
</VirtualHost>
```

```
<VirtualHost 192.168.0.34>
# setzen des Namen des virtuellen Webservers
ServerName security.alpha
# Verzeichnis, in dem die Dokumente zu finden sind
DocumentRoot "/usr/local/httpd/htdocs.security"
</VirtualHost>
```

```
<VirtualHost ip.address.of.host.some_domain.com>
# E- Mail des Server Admins
ServerAdmin webmaster@host.some_domain.com
# Verzeichnis, in dem die Dokumente zu finden sind
DocumentRoot /www/docs/host.some_domain.com
# setzen des Namen des virtuellen Webservers
ServerName host.some_domain.com
# Hier werden die Log- Dateien gesichert
ErrorLog logs/host.some_domain.com-error_log
CustomLog logs/host.some_domain.com-access_log common
</VirtualHost>
```

## Apache Server als Http- Proxy

Der Apache- Server kann auch als Proxy dienen. Dazu müssen lediglich ein paar zeilen in die Konfigurationsdatei eingefügt werden.

Natürlich ist es auch möglich dies in Verbindung mit einem Virtuellen Webserver zu gestalten.

```
<IfModule mod_proxy.c>
# Soll der Server als ein Proxy arbeiten, oder nicht?
# Setzt „ProxaPss“ nicht außer Kraft!
# Argumente : <on | off>
ProxyRequests On

# So werden fremde Server auf einem Lokalen angezeigt.
# „http://proxy/meine-sub-domain/index.html“ ist jetzt das
# gleiche wie „http://sub.meine-domain.com/index.html“
# Argumente : <Pfad> <URL>
ProxyPass /meine-sub-domain http://sub.meine-domain.com

# Wer kann auf den Proxy zugreifen?
# Der Stern ist ein Platzhalter für alle Ports
<Directory proxy:*>
    Order deny, allow
    Deny from all
    Allow from .meine-domain.com
</Directory>

# Steuert ob im Kopf z.B. einer E- Mail "Via:" angezeigt
# wird. Dieses Feature wurde eingeführt um den Weg durch
# mehrere Proxy- Server zurück zu verfolgen.
# "Full" fügt die Version des Apache- Server hinzu.
# "Block" entfernt alle eventuell schon vorhandene „Via“s
# Argumente : <Off | On | Full | Block>
ProxyVia On

# Dieser Pfad gibt an, wo die Dateien „gecacht“ werden.
# Ist kein Pfad angegeben wird nicht gecacht
# Argumente : <Verzeichnispfad >
CacheRoot "/var/cache/http"

# Legt die Größe des Cache's fest
# Argumente : <Größe in KByte>
CacheSize 5

# Gibt die Zeit an bis der Cache wieder überprüft wird.
# Erst dann werden alle Daten gelöscht die über
# „CacheSize“ liegen.
# Argumnte : <Zeit in Stunden>
CacheGcInterval 4.5

# Nach dieser Zeit werden die Dateien im Cache
# aufgefrischt.
# Argumnte : <Zeit in Stunden>
CacheMaxExpire 24
```

```
# Wenn der originale Http- Server keine Zeit für das
# Auslaufen eines Dokumentes vorsieht. Wird dies nach
# folgender Formel ermittelt.
# expiry-period = time-since-last-modification * <factor>
# Argumnte : <factor>
CacheLastModifiedFactor 0.1

# Die Vorgabe, wann ein Dokument „ausgelaufen“ ist.
# Argumnte : <Zeit in Stunden>
CacheDefaultExpire 1

# Wie tief werden die Seiten gecacht?
# Argumnte : <levels>
CacheDirLevels 3

# Gibt an von wo nicht gecacht wird.
# Argumnte : <Wörter | Hostsnamen | IP- Adressen>
NoCache a_domain.com another_domain.edu
joes.garage_sale.com
```

```
</IfModule>
```

Beispiel für die Verbindung von Virtuellen Webserver und Proxy.

```
NameVirtualHost 192.168.0.34
```

```
<VirtualHost 192.168.0.34>
  ServerName proxy.meine-domain.com
  DocumentRoot "/usr/local/httpd/htdocs.proxy"
  <IfModule mod_proxy.c>
    ProxyRequests On
    <Directory proxy:*>
      Order deny, allow
      Deny from all
      Allow from .meine-domain.com
    </Directory>
  </ IfModule>
</VirtualHost>
```

## Quellen

Apache core