

SelfLinux-0.10.0



Linux als WWW-Server

Autor: Dirk Hebenstreit (*dirk.hebenstreit@epost.de*)
Formatierung: Matthias Nuessler (*m.nuessler@web.de*)
Lizenz: GFDL

Linux bietet sich als Betriebssystem zum Betreiben eines Webservers geradezu an. Die am weitesten verbreitete Webserver-Software, der Apache, ist ebenso wie Linux frei verfügbar.

Deshalb erfolgt hier eine kurze Einführung in dessen Konfiguration mit Beschreibung der wichtigsten Konfigurationsdateien.

Inhaltsverzeichnis

1 Was ist eigentlich der Apache?

2 Grundkonfiguration


- 2.1 Übersicht der Konfigurationsdateien
- 2.2 Wichtige globale Parameter in der httpd.conf
- 2.3 Wichtige Webserverparameter
- 2.4 Wichtige Zugriffsberechtigungsparameter
- 2.5 Zugriffsschutz für Verzeichnisse

3 Aufgaben

1 Was ist eigentliche der Apache?

Der bekannteste Dienst des Internet ist zweifelsohne das **WWW** oder **World Wide Web** (Welt Weites Warten ;-)). Es ist eine klassische Client-Server-Anwendung, bei welcher der Client passiv Informationen vom Server bezieht und diese in der Regel nicht verändern kann. Grundlage sind ein zustandsloses Protokoll (**HTTP**=Hypertext Transfer Protocol) und die Seitenbeschreibungssprache **HTML** (Hypertext Markup Language).

Der Server hat die Aufgabe, die Anfragen der Form `Dienst://server.in.domäne/Pfad_zur_Seite` zu beantworten und die gewünschte Seite auszuliefern. Ferner können dynamische Seiten generiert werden, z.B. durch eine Datenbankanwendung, welche vom WWW-Server abgefragt wird. Um besondere Dienste zu realisieren, können auch komplette Programme auf dem Webserver gestartet werden, in der Regel über das **CGI** (Common Gateway Interface). Dies erlaubt es, vom Client aus bestimmte Prozesse auf einem Server anzustoßen.

Seit der Erfindung des WWW Ende der 80iger Jahre am  [CERN](http://cern.ch) in der Schweiz, hat sich der Dienst vom reinen Userinterface für Informationssysteme zu einer globalen und plattformübergreifenden Benutzeroberfläche gewandelt. Damit wurden die Inhalte und Aufgaben der Web-Server immer weiter ausgedehnt, und es entstanden verschiedene Lösungsansätze und Programme.

Der am weitesten verbreitete Server im Internet ist mit über 60% der **Apache**. Er ist als **GNU**-Software frei verfügbar und hat sich als "A patchy Server" aus dem NCSA-Server entwickelt, der auch die Basis für andere Entwicklungen darstellt. Zwischenzeitlich ist er auf nahezu alle Plattformen portiert worden, auch eine NT-Variante ist verfügbar.

Der Apache stellt in der SuSE-Distribution den Standard-Webserver dar und wird in der Serie **n** installiert.

2 Grundkonfiguration

Die Konfigurationsdateien des Apache befinden sich im Verzeichnis `/etc/httpd`. Dort liegt nach der Installation die Datei `httpd.conf`.

Bei älteren Installationen gibt es darüber hinaus noch weitere Dateien:

- * `access.conf`
- * `magic`
- * `mime.types`
- * `srn.conf`

2.1 Übersicht der Konfigurationsdateien

Der Apache verwendet eine Konfigurationsdatei `httpd.conf`. Hier werden die Grundeinstellungen für den Server vorgenommen.

Manchmal, besonders in älteren Installationen, sieht man noch weitere Dateien:

1. `access.conf`
Wie der Name es erahnen läßt, werden hier die Zugriffsrechte auf den Server vergeben.
2. `httpd.conf`
Hier werden die Grundeinstellungen für den Server vorgenommen.
3. `magic`
Diese Datei wird vom Apache benutzt, um den Typ einer auszuliefernden Datei zu ermitteln.
4. `mime.type`
Hier stehen die MIME-Typen (Multipurpose Internet Media Extensions).
5. `srn.conf`
Ergänzungen zur Grundkonfiguration (Server Ressource Management).

Nach der Installation mittels `YaST`, `rpm` oder einem anderen Verfahren ist der Server bereits einsatzbereit. Er wird beim Starten des Rechners automatisch in Runlevel 2 gestartet und kann per Hand über das Script `/etc/rc.d/apache` beeinflusst werden. Dieses wertet die Variable `START_HTTPD` aus, welche in `rc.config` auf `YES` gesetzt werden muß (ist standardmäßig der Fall). Danach benutzt der Apache das Verzeichnis `/usr/local/httpd/htdocs` als Basisverzeichnis für die HTML-Dateien. Der Aufruf vom Browser aus sollte dann bereits folgendes Ergebnis liefern:

Um den Server herunterzufahren geben sie `/etc/rc.d/apache stop` ein.

Wenn Sie ihn wieder starten wollen, so geht dies analog mit `/etc/rc.d/apache start`.

Um im laufenden Betrieb Änderungen an den Konfigurationsdateien wirksam werden zu lassen, können Sie mit `/etc/rc.d/apache restart` einen Neustart des Servers veranlassen.

Entgegen älteren Versionen des Apache wird hierbei lediglich der Vaterprozeß von der Änderung informiert (mittels `kill -HUP PID-des-Vater-httpd`) und nicht die gesamte Prozeßgruppe gekillt und neu gestartet.

2.2 Wichtige globale Parameter in der `httpd.conf`

Wie Sie bereits gesehen haben, ist der Apache bereits direkt nach der Installation funktionsbereit. Um ihn in das lokale Netz einzubinden und auf die zu erwartende Last einzustellen, sollten aber einige Änderungen erfolgen. Folgende Parameter in der Datei `httpd.conf` sind interessant:

LoadModule, modulname Pfad zum Modul

Wie eingangs erwähnt, zeichnet sich der Apache durch Modularität und leichte Erweiterbarkeit aus. Über diese Parameter in `httpd.conf` kann der Leistungsumfang des Apache bestimmt werden, indem einzelne Module ein- oder ausgeblendet werden. Dies kann in sensiblen Bereichen (Internet-Server) besonders interessant werden, wenn sich z.B. ein Modul als potentielle Sicherheitslücke erwiesen hat. Anstatt den gesamten Server vom Netz zu nehmen (was einem Verlust der Web-Präsenz und damit geschäftlichen Verlusten gleichkommt) muß lediglich das betroffene Modul deaktiviert und der Apache neu gestartet werden. Damit fehlt zwar ein Teil der Funktionalität, aber nicht das gesamte System.

ServerType standalone

Standardmäßig läuft der `httpd` bei SuSE als Standalone-Server, d.h. er wird über die rc-Skripte gestartet und bleibt bis zum Beenden durch Aufruf des Shutdown-Skriptes im Speicher. Alternativ hierzu kann auf Servern, welche lediglich für besondere Zwecke und nur zeitweise als Webserver laufen, auch ein Start über den `inetd` erreicht werden. Neben der entsprechenden Einrichtung in der Datei `/etc/inetd.conf` muß in der `httpd.conf` der Wert `standalone` durch `inetd` ersetzt werden.

Port 80

Der Standard-Port für das HTTP ist 80. Ist es aus besonderen Gründen erforderlich, hiervon abzuweichen, kann hier ein alternativer Port angegeben werden. Beachten Sie dabei, daß diese Änderung den Clients bekannt gemacht wird (Browser-Konfiguration) und das der gewünschte Port auf der Servermaschine auch frei ist.

**User wwwrun
Group nogroup**

Der `httpd`-Prozeß sollte aus Sicherheitsgründen nicht unter root-Kennung laufen. Da aber für die Benutzung von Ports unterhalb von 1023 eben dieses notwendig ist, startet lediglich der Vaterprozeß als root und wechselt anschließend nach User und Group. Der Standarduser `wwwrun` ist bei SuSE bereits eingerichtet, der Wert `-2` führt in die Gruppe `nogroup`, welche ebenfalls auf Ihrem System vorhanden sein sollte.

ServerAdmin you@your.address

Fehler und Probleme sendet der `httpd` per Mail an diese Adresse. Sie taucht auch in den automatisch generierten Fehlerseiten für den Benutzer auf und sagt ihm, an wen er sich wenden soll, falls Probleme auftauchen. Im Internet hat sich die Konvention `webmaster@ihre.domain` durchgesetzt, wobei webmaster in der Datei `/etc/aliases` auf einen realen Benutzer verweist.

ServerName www@ihre.domain

Hier tragen Sie den qualifizierten Namen Ihres Servers ein. Beachten Sie dabei, daß dieser aufgelöst werden können muß, d.h., entweder über DNS oder in der Datei `/etc/hosts` zugeordnet wird.

ServerRoot "/usr/local/httpd"

Dies ist das Basisverzeichnis der `httpd`-Installation. Wenn nicht anders angegeben (s.u.) werden Logdateien u.ä. abwärts von diesem Verzeichnis gesucht.

ErrorLog /var/log/httpd/error_log

Der Pfad zu der Fehler-Logdatei des Apache.

LogLevel warn

LogFormat: **UNIX**-Programme können sehr geschwätzig sein, was ihren Zustand und eventuelle Fehler betrifft. Hierfür existiert seit BSD-UNIX der Mechanismus des SYSLOG-Schreibers, der diese Meldungen protokolliert. Daneben erzeugen wichtige Serverprozesse aber auch eigene Log-Dateien und geben diesen ein entsprechendes Format. Mittels LogLevel kann die "Geschwätzigkeit" des `httpd` von wenig (`emerg` = emergency; nur "Lebensbedrohliche" Meldungen werden generiert), bis viel (`debug` = debugging; jede Kleinigkeit wird protokolliert) beeinflusst werden. Standard ist `warn`, was im Normalbetrieb ausreichend ist. Hinsichtlich des Formates der Log-Dateien hat sich das *Common Log Format* (CLF) durchgesetzt, welches von vielen Auswertungsprogrammen unterstützt wird. Änderungen an den Standardwerten sind nicht ratsam, es sei den, man ist sich der Auswirkungen bewußt.

CustomLog /var/log/httpd.access_log common

Diese Log-Datei hat eine Sonderstellung, da in ihr sämtliche Zugriffe auf den Server protokolliert werden.

StartServers 5**MinSpareServers 5**

Gibt die Anzahl der anfangs zu startenden und danach mindestens verfügbaren Child-Prozesse an. Der Apache erzeugt nach dem Start die hier angegebene Anzahl von Subprozessen und teilt die Anfragen danach dynamisch auf diese auf, so daß eine entsprechende Lastverteilung erfolgt.

MaxSpareServers 10

Wenn die vorhandenen Childprozesse nicht mehr ausreichen, erzeugt der Vaterprozeß weitere Kinder. Geht die Last dann wieder zurück, können die nunmehr leerlaufenden Prozesse wieder beendet werden. Wenn mehr als die hier angegebenen Child-Prozesse im Leerlauf sind, werden sie bis zu diesem Wert hin wieder beendet.

Die Datei `httpd.conf` bietet noch weitere Einstellungsmöglichkeiten. Zu Details sei hier auf die entsprechende Literatur und die mitgelieferte Dokumentation unter `/usr/doc/packages/apache`, sowie die HTML-Online-Hilfe verwiesen.

2.3 Wichtige Webserverparameter

Bei älteren Installationen sind diese Parameter in der Datei `srvm.conf` zu finden, normalerweise stehen diese jedoch auch in der `httpd.conf`.

DocumentRoot "/usr/local/httpd/htdocs"

Hier wird das Wurzelverzeichnis für die HTML-Dokumente festgelegt (Document Root). Alle Anfragen der Clients werden relativ zu diesem Verzeichnis beantwortet (`http://www.your.domain/path_to_file/file.html`). Bei eine Anfrage der Form `http://www.your.domain` wird in diesem Verzeichnis die Index-Datei (s.u.) gesucht.

UserDir public_html

Auf öffentlichen Webservern, aber auch in Unternehmen, wird die Möglichkeit angeboten, Benutzerseiten von Anwendern in das Netz zu stellen. Dazu wird in deren Home-Verzeichnis ein Unterverzeichnis `public_html` angelegt und mit der URL `http://www.server.name/~username/` kann dann auf dieses Verzeichnis zugegriffen werden. Wenn uns der Eintrag `public_html` nicht gefällt, so können wir ihn hier ändern.

DirectoryIndex index.html

Mit dieser Einstellung legen Sie fest, welche Dateien als sog. Indexdateien betrachtet werden. Bei Aufruf einer URL, die nur einen Pfad als Ziel enthält (z.B. `http://www.my.domain/verzeichnis/`) sucht der Apache in `<verzeichnis>` nach einer Datei, die einen der unter **DirectoryIndex** aufgeführten Namen hat. Mehrere Möglichkeiten können durch Leerzeichen getrennt angegeben werden, z.B. `index.html index.htm anfang.html`.

AddIcon [Pfad_zum_Icon]

Diese Einstellungen betreffen die Zuordnung von Icons zu einzelnen Dateien/MIME-Typen. Wenn vom Client auf ein Verzeichnis zugegriffen wird, in dem keine Datei `index.html` (s.o.) liegt, so bekommt er ein Inhaltsverzeichnis angezeigt. Mit den genannten Einträgen können die auszuliefernden Symbole beeinflusst oder geändert werden. Im Normalfall sind die voreingestellten Werte aber ausreichend und aufgrund der weiten Verbreitung des Apache eine Art Quasi-Standard. Man kann aber Symbole für eigene Dateitypen nach belieben hinzufügen.

HeaderName HEADER

Um dem Anwender eine kurze Erläuterung des Verzeichnisinhaltes zu geben, kann dort eine README-Datei angelegt werden. Apache stellt diese dann dem eigentlichen Verzeichnisinhalt voran.

Mit diesen einfachen Hilfsmitteln kann z.B. eine einfache Erklärung für die Verzeichnisstruktur eines FTP-Servers aufgebaut werden.

AccessFileName .htaccess

Der Name der bereits erwähnten `.htaccess`-Datei ist ebenfalls frei wählbar. Inhaltlich entsprechen diese Dateien den Directory-Anweisungen aus der Datei `httpd.conf` und dürfen demnach die gleichen Inhalte besitzen. Um zu verhindern, daß Anwender eigenmächtig riskante Einstellungen für ihr `public_html` vornehmen ist die Auswertung dieser Dateien aber in `httpd.conf` in der Regel gesperrt (s. `AllowOverride`). Ist die Benutzung von `.htaccess`-Dateien erlaubt, so könnten diese folgenden Inhalt haben:

.htaccess
<pre>order deny,allow deny from all allow from localhost</pre>

Damit erlauben wir nur dem Rechner `localhost` den Zugriff auf unser Verzeichnis NSG.

2.4 Wichtige Zugriffsberechtigungsparameter

Bei älteren Installationen sind diese Parameter in der Datei `access.conf` zu finden, normalerweise stehen diese jedoch auch in der `httpd.conf`.

Es wird der Zugriff auf Dateien und Verzeichnisse geregelt. Sie korreliert dabei mit den `.htaccess`-Dateien in den Verzeichnissen, welche einen User-bezogenen Zugriff gewähren.

Jede der nun folgenden Anweisungen kann für einzelne Unterverzeichnisse (ab `DocumentRoot`) gesetzt

werden. Hierfür ist das betroffene Verzeichnis jeweils mit `<Directory "Pfad">` einzuleiten und die Beschreibung mittels `</Directory>` als Block zu beenden. Die Datei `access.conf` beginnt mit einem Abschnitt für die Grundeinstellungen ab `DocumentRoot` und setzt dabei restriktive Werte:

Auszug httpd.conf

```
<Directory>
Options None
AllowOverride None
</Directory>
```

Hierfür wird die Wirkung von `.htaccess`-Dateien für `DocumentRoot` festgelegt. Default (des Apache, nicht SuSE!) ist `All`, d.h., die `.htaccess`-Dateien werden gelesen und der komplette Inhalt wird interpretiert. Wenn auf dem Server das Erstellen privater Benutzerseiten gewünscht ist, dann ist dieser Wert aber nicht zu empfehlen, da die Anwender sonst z.B. die Nutzung von eigenen CGI-Skripten einschalten könnten. Daher ist bei SuSE diese Einstellung auf `None` gesetzt, d.h., `.htaccess`-Dateien werden ignoriert.

Jetzt werden mit `<Directory "/usr/local/httpd/htdocs">` die Einstellungen für `/usr/local/httpd/htdocs` (also wieder unser `DocumentRoot`) genauer bestimmt.

Options Indexes FollowSymLinks

Mit der Options-Anweisung werden verschiedene Einstellungen eingeleitet. In diesem Beispiel bedeutet `Indexes`, daß der Server bei Aufruf einer URL mit einem Directory als Ziel (z.B. `http://www.my.domain/Verzeichnis/`) in `<DocumentRoot/Verzeichnis>` eine Datei `index.html` sucht und diese zurückliefert, ist diese nicht vorhanden, aber mit einer Fehlermeldung (Document not found) antwortet.

Die Option `FollowSymLinks` erlaubt es, aus dem `DocumentRoot` mittels symbolischer Links auszubrechen, sofern die Zugriffsrechte entsprechend gesetzt sind. Damit können Sie bei Platzproblemen einen Teilbaum Ihres Webservers auf eine andere Partition oder gar ein über `NFS/SAMBA` gemountes Verzeichnis legen.

Auszug httpd.conf

```
order allow,deny
allow from all
```

Hiermit legen Sie fest, wer alles auf Ihren Server zugreifen darf. Zuerst wird die Reihenfolge bestimmt, in der die folgenden Regeln ausgewertet werden. Bei `order allow,deny` werden zuerst die unter `allow` aufgeführten Regeln und danach erst die unter `deny` stehenden angewendet. Der Eintrag `order deny,allow` dreht diese Reihenfolge um.

Die einzelnen Regeln besitzen die Grundsyntax `allow|deny from domain|all`. Wollen wir z.B. nur den Rechnern aus unserer eigenen Domain den Zugriff erlauben, so schreibe man:

.htaccess

```
order deny,allow
deny from all
allow from my.domain
```


Um nur bestimmten Domains den Zugriff zu verweigern schreibe man:

Auszug httpd.conf

```
order allow,deny
allow from all
deny from bad.domain other.domain 123.123.40
```

Wie man sieht ist neben der Angabe des symbolischen Namens auch die Netzadresse des (un)erwünschten Netzes möglich.

2.5 Zugriffsschutz für Verzeichnisse

Aus der Kombination von `httpd.conf` (beziehungsweise `access.conf`) und `.htaccess`-Dateien ergibt sich der Zugriffsschutz für die Verzeichnisstruktur. Um nur einer bestimmten Benutzergruppe oder einzelnen Anwendern den Zugriff auf Verzeichnisse zu gewähren, sind folgenden Einstellungen notwendig:

In der Datei `access.conf` wird das Verzeichnis aus den allgemeingültigen Einstellungen herausgelöst.

Eintrag in access.conf:

```
<Directory /usr/local/httpd/htdocs/NSG>
AllowOverride All
AuthName NSG
AuthType Basic
AuthUserFile /etc/httpd/passwd
AuthGroupFile /etc/httpd/group
Require user hebi:nsg
</Directory>
```

AllowOverride

All ermöglicht es, mittels `.htaccess` weitere Einstellungen innerhalb des Verzeichnisses anzulegen.

AuthName NSG

legt einen Namen für das Login-Feld an.

AuthType

Basic stellt den Autorisierungsmechanismus ein (s.u.).

AuthUserFile

sagt, wo die Paßwortdatei zu suchen ist.

AuthGroupFile

legt fest, in welcher Datei die Usergruppen zu suchen sind.

Require user

legt fest, welche Benutzer (aus `AuthUserFile`) Zugriff haben sollen. Der Apache setzt bei der Authorisierung auf verschiedene Mechanismen. Bereits in der Standardausführung enthält er die Verfahren

```
Basic (mod_auth)
dbm (mod_auth_dbm)
db (mod_auth_db)
```

Weitere Möglichkeiten können über Module eingebunden werden. Die verbreitetste ist wohl der Weg über eine Datenbank, wobei die für nicht-kommerzielle Zwecke freie Implementierung **mSQL** bereits als Module `mod_auth_mysql` zum Standardumfang des Apache gehört.

Um die für das Verfahren Basic notwendigen Dateien anzulegen benötigt man das Programm `htpasswd` und einen beliebigen Editor. Das Anlegen einer Paßwortdatei erfolgt mit

```
root@linux / # htpasswd -c [Pfad_zur_Datei] Username
```

Danach kann ohne den Schalter `-c` ein neuer Benutzer angelegt werden. Hierbei wird der bei UNIX standardmäßig implementierte crypt(3)-Mechanismus verwendet. Das Ergebnis sieht dann wie folgt aus:

[Pfad_zur_Datei]
nobody:8Y0H8cYsTFzNw hhebi:wN95S.Ux00ZH6 hhe:.NQ/RSk5MnfvQ

Für eine Gruppendatei benötigt man lediglich einen Editor wie z.B. den `vi`:

Gruppendatei
admin:hebi he other:he nobody

Im Verzeichnis (hier: `/usr/local/httpd/htdocs/NSG`) kann mit der Datei `.htaccess` der Zugriff weiter kontrolliert werden:

Satisfy All

Order deny,allow

deny from all

allow from my.domain

Damit können jetzt nur noch Clients aus der Domain `my.domain` auf das Verzeichnis zugreifen. Ersetzt man bei `Satisfy` den Wert `All` durch `Any` ist aus der Domain `my.domain` der Zugriff ohne Einschränkung erlaubt, alle anderen Rechner benötigen eine gültige Kombination aus Username und Password.

3 Aufgaben

1. Installieren Sie den Apache von der CD, indem Sie **YaST** benutzen. Kontrollieren Sie, ob sich alle Dateien an den richtigen Orten befinden.
2. Starten Sie den Apache über das Script `/etc/rc.d/apache`. Was müssen Sie beachten? Wie können Sie verhindern, daß der Apache beim Hochfahren des Rechners gestartet wird?
3. Ändern Sie folgende Einstellungen in der Datei `httpd.conf`:

Änderungen in httpd.conf
<pre>ServerName = wwwXY.nsg.edu (Wobei XY Ihrer Platznummer entspricht) ServerAdmin = webmaster@wwwYX.nsg.edu</pre>

Was müssen Sie beachten, damit diese Einstellungen auch funktionieren?

4. Ändern Sie folgenden Einstellungen:

Änderungen in httpd.conf
<pre>StartServer = 10 MinSpareServers = 15</pre>

Stoppen Sie den Apache und starten Sie ihn erneut. Wie können Sie die Auswirkungen Ihrer Änderungen kontrollieren und wie sehen diese aus?

5. Erzeugen Sie im "DocumentRoot" ein Verzeichnis **Eigene_Dateien** und erstellen Sie darin eine Datei **index.html**. Benutzen Sie hierfür die Vorlage `/home/ablagen/ZENTRAL/html_body.html`. Arbeiten Sie dabei als root.
6. Versuchen Sie, mittels Netscape die eben erstellte Datei aufzurufen. Benutzen Sie dabei sowohl eine absolute URL (http://localhost/Eigene_Dateien/index.html) als auch den einfachen Verzeichnisaufruf (http://localhost/Eigene_Dateien/). Erfolg?
7. Ergänzen Sie die Voreinstellungen für Indexdateien um die Einträge **index.htm** und **anfang.html**. Erzeugen Sie durch Kopieren diese Dateien in **Eigene_Dateien** und verändern Sie diese, so daß man einen Unterschied erkennt. Versuchen Sie auf http://localhost/Eigene_Dateien/ mit dem Netscape zuzugreifen und stellen Sie fest, welche Datei angezeigt wird. Verändern Sie die Reihenfolge der Indexdateien in `httpd.conf` und versuchen Sie es erneut. Was stellen Sie fest?
8. Legen Sie einen Benutzer "*Peter Maier*" (Username: **pmaier**, Homeverzeichnis: `/home/pmaier`) an und erzeugen Sie ein Unterverzeichnis `/home/pmaier/public_html`. Erzeugen Sie eine Datei **index.html** in diesem Unterverzeichnis (benutzen Sie hierfür wieder die Vorlage `html_body.html`) und versuchen Sie, mit Netscape auf diese zuzugreifen.
9. Ergänzen Sie die Datei `httpd.conf` um folgende Zeilen:

Ergänzung in httpd.conf

```
<Directory /usr/local/httpd/htdocs/NSG>
    AllowOverride All
    AuthName NSG
    AuthType Basic
    AuthUserFile /etc/httpd/passwd
    AuthGroupFile /etc/httpd/group
    Require user hebi nsg
</Directory>
```

Legen Sie die darin beschriebenen Paßwort- und Gruppendateien mit den entsprechenden Benutzern an. Starten Sie den Apache neu und versuchen Sie, auf das Verzeichnis zuzugreifen.