



Kostenloses eBook

LERNEN

apache

Free unaffiliated eBook created from
Stack Overflow contributors.

#apache

Inhaltsverzeichnis

Über.....	1
Kapitel 1: Erste Schritte mit Apache.....	2
Bemerkungen.....	2
Versionen.....	2
Verschiedene Apache-httpd-Versionen.....	2
Examples.....	2
Installation oder Setup.....	2
Ubuntu-Installation.....	2
Windows-Installation.....	2
CentOS-Installation.....	3
macOS Installation.....	3
[Ubuntu] Einfaches Hello World Beispiel.....	3
Installationsvoraussetzungen.....	3
HTML einrichten.....	3
Besuchen Sie Ihre Webseite.....	4
Um sicherzustellen, dass der Server aktiv ist.....	4
Kapitel 2: .htaccess-Dateien in Apache.....	5
Examples.....	5
Motor umschreiben.....	5
HTTPS erzwingen.....	5
Aktivieren Sie CORS.....	6
Voraussetzungen.....	7
301 Umleitung durch Htaccess.....	7
Kapitel 3: Apache Flume.....	9
Einführung.....	9
Examples.....	9
Streaming / Protokolldaten.....	9
Kapitel 4: So erstellen Sie einen virtuellen Host in Apache.....	10
Bemerkungen.....	10
Examples.....	10

Namensbasierte Konfiguration des virtuellen Hosts.....	10
Virtueller Host für PHP-Entwicklung.....	11
Virtueller Host in WAMP.....	12
1) IP-basierte vhosts 2) Mehrere vhosts mit demselben Port 3) Definieren von vhosts mit Ma.....	13
Erzwingen Sie HTTPS mithilfe eines virtuellen Hosts.....	14
Credits.....	16



You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [apache](#)

It is an unofficial and free apache ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official apache.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Kapitel 1: Erste Schritte mit Apache

Bemerkungen

In diesem Abschnitt erhalten Sie einen Überblick darüber, was Apache ist und warum ein Entwickler es verwenden möchte.

Es sollte auch alle großen Themen im Apache erwähnen und auf die verwandten Themen verweisen. Da die Dokumentation für Apache neu ist, müssen Sie möglicherweise erste Versionen dieser verwandten Themen erstellen.

Versionen

Verschiedene Apache-httpd-Versionen

Ausführung	Aktuelle Version	Veröffentlichung
1.3	1.3.42	1998-06-06
2,0	2.0.65	2002-04-06
2.2	2.2.32	2005-12-01
2.4	2.4.25	2012-02-21

Examples

Installation oder Setup

Detaillierte Anweisungen zum Einrichten oder Installieren von Apache.

Ubuntu-Installation

```
sudo apt-get install apache2
```

Windows-Installation

Schauen Sie sich den [WAMP](#)- Stack an. WAMP steht für Windows, Apache, MySQL, PhpMyAdmin.

CentOS-Installation

Apache 2.2 wird mit CentOS6 geliefert, wohingegen 2.4 mit CentOS7 zur Installation auf einem der beiden Betriebssysteme ausgeführt wird

```
yum -y install httpd
```

macOS Installation

macOS ist mit vorinstalliertem Apache ausgestattet, kann jedoch über Homebrew installiert werden

Wenn Sie bereits den integrierten Apache ausgeführt haben, müssen Sie ihn zuerst herunterfahren und alle Skripts zum automatischen Laden entfernen.

```
$ sudo apachectl stop
$ sudo launchctl unload -w /System/Library/LaunchDaemons/org.apache.httpd.plist 2>/dev/null
$ brew install httpd24 --with-privileged-ports --with-http2
```

[Ubuntu] Einfaches Hello World Beispiel

In diesem Beispiel werden Sie durch das Einrichten eines Backends für eine Hello World-HTML-Seite geführt.

Installationsvoraussetzungen

Ordnung zählt für diesen Schritt!

- `sudo apt-get install apache2`

HTML einrichten

Apache-Dateien leben in `/var/www/html/`. Lasst uns schnell dahin kommen. Stellen Sie sicher, dass Sie sich zuerst in Ihrem Stammverzeichnis befinden, `cd` und dann `cd /var/www/html/`.

In diesem `html` Verzeichnis werden alle Ihre Website-Dateien gespeichert. Lässt schnell eine einfache Hello World-Datei erstellen.

Geben Sie in Ihrem bevorzugten Texteditor Folgendes ein

```
<!DOCTYPE html>
<html>
<head>
```

```
<title>Hello World!</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

Speichern Sie diese Datei als `index.html` im aktuellen Verzeichnis und Sie können loslegen!

Besuchen Sie Ihre Webseite

Um die gerade erstellte Seite in Ihrem Browser zu besuchen, rufen Sie `localhost` . Wenn dies nicht funktioniert, versuchen Sie es mit `127.0.0.1` . Sie sollten "Hallo Welt" sehen. als `h1` . Sie sind fertig!

Um sicherzustellen, dass der Server aktiv ist.

Wenn Sie eine Meldung erhalten, dass der Browser keine Verbindung zum Server herstellen kann, überprüfen Sie zunächst, ob der Server aktiv ist.

```
$ ps -aef | grep httpd
```

Sie sollten einige `httpd`-Prozesse sehen, wenn Apache läuft.

Erste Schritte mit Apache online lesen: <https://riptutorial.com/de/apache/topic/964/erste-schritte-mit-apache>

Kapitel 2: .htaccess-Dateien in Apache

Examples

Motor umschreiben

Das RewriteEngine-Modul in Apache wird verwendet, um URLs und Pfade abhängig von den bereitgestellten Ausdrücken dynamisch neu zu schreiben:

```
<IfModule mod_rewrite.c>
RewriteEngine On
RewriteBase /
RewriteRule ^index\.php$ - [L]
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule . /index.php [END]
</IfModule>
```

Die obigen Regeln schreiben PHP-Dateien so um, dass ihre Erweiterung nicht mehr angezeigt wird, sodass index.php nur als nackte Domäne angezeigt wird (ähnlich dem Verhalten, das normalerweise in index.html auftritt). Die obige Regel wird mit WordPress ausgeliefert.

Beachten Sie, dass in Apache httpd 2.2.16 und höher der gesamte Block mithilfe der Direktive `FallbackResource` durch eine einzige Zeile ersetzt werden kann:

```
FallbackResource /index.php
```

HTTPS erzwingen

.htaccess kann verwendet werden, um die Umleitung Ihrer HTTP-Site zu HTTPS zu erzwingen.

Hier ist eine schnelle Methode, die den Code für Ihre Domain nicht bearbeiten muss:

```
RewriteEngine On
RewriteCond %{HTTPS} =off
RewriteRule ^ https://%{HTTP_HOST}%{REQUEST_URI} [L,R=301]
```

Warnung: Bei dem obigen Code wird davon `%{HTTP_HOST}` dass Sie `%{HTTP_HOST}` auf Ihre Domäne verwenden können.

Wenn Sie sicherstellen `%{HTTP_HOST}`, dass der Umleitungsspeicherort Ihre Domäne ist, ersetzen Sie `%{HTTP_HOST}` durch Ihre Domäne.

Der Code oben macht dies:

1. [Aktivieren Sie RewriteEngine](#) .
2. Fahren Sie fort, wenn die aktuelle Anforderung kein HTTPS verwendet.
3. `https://%{HTTP_HOST}%{REQUEST_URI}` Sie eine HTTP 301-Weiterleitung an

`https://%{HTTP_HOST}%{REQUEST_URI}` durch

- `%{HTTP_HOST}` ist der vom Browser angeforderte Host und
- `%{REQUEST_URI}` ist die vom Browser angeforderte URI (alles nach der Domäne).

Warnung: Ihre Webanwendung muss HTTPS-Anforderungen verarbeiten können, und Apache für Ihren Host sollte mit einem gültigen Standortzertifikat konfiguriert werden.

Beachten Sie, dass es wesentlich effizienter ist, einfach eine `Redirect` im `http-Vhost` durchzuführen, als diese mehrfachen Vergleiche pro Anforderung in einer `.htaccess`-Datei durchzuführen. Weitere Informationen zu dieser Technik finden Sie unter <http://wiki.apache.org/httpd/RedirectSSL> .

Aktivieren Sie CORS

Um **Cross-Origin Resource Sharing (CORS)** in Apache zu aktivieren, müssen Sie mindestens einen HTTP-Header festlegen, der ihn ändert (das Standardverhalten ist das Blockieren von CORS). Im folgenden Beispiel wird dieser HTTP-Header in `.htaccess` , er kann jedoch auch in Ihrer Site `your-site.conf` oder in der Apache-Konfigurationsdatei festgelegt werden. Unabhängig davon, wie Ihre Konfiguration aussieht, können Sie die relevanten HTTP-Header in jedem Apache-Konfigurationsblock festlegen, z. B. `<VirtualHost>` , `<Directory>` , `<Location>` und `<Files>` .

Es gibt einige CORS-bezogene HTTP-Header, die Sie in der Antwort zurückgeben können:

```
Access-Control-Allow-Origin
Access-Control-Allow-Credentials
Access-Control-Allow-Methods
Access-Control-Max-Age
Access-Control-Allow-Headers
Access-Control-Expose-Headers
```

Einige der oben genannten Anforderungen sind für "Preflight" -Anfragen erforderlich. Einige HTTP-Clients (dh moderne Browser) führen eine Anforderung **vor** der gewünschten Anfrage aus, um zu sehen, ob sie die Berechtigung haben, die eigentliche Anfrage auf dem Server auszuführen. Weitere [Informationen](https://en.wikipedia.org/wiki/Cross-origin_resource_sharing) zur Preflight-Anforderung finden Sie unter https://en.wikipedia.org/wiki/Cross-origin_resource_sharing .

Der wichtigste HTTP-Header, den wir benötigen, ist `Access-Control-Allow-Origin` und das werden wir festlegen. Das gleiche Prinzip gilt jedoch für alle (Sie müssen nur wissen, was Sie zurückgeben müssen).

Im folgenden Beispiel wird der erforderliche HTTP-Header innerhalb eines `<Directory>` - Konfigurationsblocks festgelegt, um einen SSL-geschützten Client mit dem vollständigen qualifizierten Domännennamen (FQDN) zu aktivieren:

```
<Directory /path/to/your/site/>
    Header set Access-Control-Allow-Origin "https://my.CLIENT.domain"
</Directory>
```

Nachdem wir dies auf dem **Server festgelegt haben** , können wir jetzt eine Anfrage von <https://riptutorial.com/de/home>

//my.client.domain an unseren Server [senden](#) , und er sollte antworten.

Hinweis: Viele Benutzer verwenden `Access-Control-Allow-Origin: "*"` ist ein Platzhalter, dh Anfragen von **ALLEN** Domänen sollten akzeptiert werden. Dies wird normalerweise nicht empfohlen, es sei denn, Sie führen eine öffentliche API oder ein Repository von Dateien aus. Beachten Sie auch den Kontext Ihrer HTTP-Headereinstellung. Vielleicht möchten Sie HTTP - Anforderungen für eine API ermöglichen, jedoch nicht für „hotlinking“ Bilder etc. Sie diesen Header festlegen können wo immer Sie wollen in Ihrem Apache - Konfigurationsfluss **nur** in bestimmten Situationen eingestellt. Zum Beispiel würde der CORS-HTTP-Header beispielsweise **nur dann** festgelegt, wenn der angeforderte Pfad **keine** Datei oder **kein** Verzeichnis ist (dies ist für eine öffentliche API geeignet, die das Image-Hotlinking nicht zulässt):

```
<Directory /path/to/your/site/>
  Options +FollowSymlinks
  Options +Indexes
  RewriteEngine On

  #Make sure it's not a specific file or directory that they're trying to reach
  RewriteCond %{SCRIPT_FILENAME} !-f
  RewriteCond %{SCRIPT_FILENAME} !-d
  Header set Access-Control-Allow-Origin "*"
  RewriteRule ^(.*)$ index.php/$1 [L]
</Directory>
```

Voraussetzungen

Sie müssen [mod_headers](#) installiert und aktiviert haben: `a2enmod headers`

301 Umleitung durch Htaccess

Der HTTP-Antwortstatuscode 301 dauerhaft verschoben wird für die permanente URL-Umleitung verwendet. Dies bedeutet, dass aktuelle Links oder Datensätze mit der URL, für die die Antwort empfangen wurde, aktualisiert werden sollten. Die neue URL sollte im Feld "Ort" der Antwort angegeben werden. Die 301-Umleitung gilt als bewährte Methode für das Upgrade von Benutzern von HTTP auf HTTPS. Schreiben Sie diesen Code in die htaccess-Datei für PHP-APACHE

```
Redirect 301 /oldpage/ /newpage/
```

Hier ist ein Beispiel, in dem eine htaccess-Datei verwendet wird, um mit einem an die Domäne angefügten SSL an ein Nicht-WWW umzuleiten.

```
RewriteEngine On
RewriteCond %{HTTPS} off
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ http://%1/$1 [R=301,L]

RewriteCond %{HTTPS} on
RewriteCond %{HTTP_HOST} ^www\.(.*)$ [NC]
RewriteRule ^(.*)$ https://%1/$1 [R=301,L]

RewriteEngine On
```

```
RewriteCond %{SERVER_PORT} 80  
RewriteRule ^(.*)$ https://example.com/$1 [R,L]
```

.htaccess-Dateien in Apache online lesen: <https://riptutorial.com/de/apache/topic/2089/-htaccess-dateien-in-apache>

Kapitel 3: Apache Flume

Einführung

Apache Flume ist ein Werkzeug-, Dienst- und Datenaufnahmemechanismus zum Sammeln und Transportieren großer Mengen von Streaming-Daten wie Protokolldateien, Ereignissen (usw.) aus verschiedenen Quellen zu einem **zentralen Datenspeicher** .

Flume ist ein äußerst zuverlässiges, verteiltes und konfigurierbares Werkzeug. Es ist hauptsächlich zum Kopieren von Streaming-Daten (Protokolldaten) von verschiedenen Webservern nach `HDFS` .

Examples

Streaming / Protokolldaten

Im Allgemeinen werden die meisten zu analysierenden Daten von verschiedenen Datenquellen wie Anwendungsservern, Websites für soziale Netzwerke, Cloud-Servern und Unternehmensservern erzeugt. Diese Daten werden in Form von Protokolldateien und Ereignissen angezeigt.

Protokolldatei - Eine Protokolldatei ist im Allgemeinen eine Datei, in der Ereignisse / Aktionen aufgelistet sind, die in einem Betriebssystem auftreten. Beispielsweise listen Webserver jede an den Server gerichtete Anforderung in den Protokolldateien auf.

Beim Sammeln solcher Protokolldaten können wir Informationen erhalten über:

die Anwendungsleistung und lokalisieren verschiedene Software- und Hardwarefehler. das Nutzerverhalten und bessere geschäftliche Erkenntnisse ableiten. Die herkömmliche Methode zum Übertragen von Daten in das HDFS-System ist die Verwendung des Befehls `put`. Lassen Sie uns sehen, wie Sie den Befehl `put` verwenden.

Apache Flume online lesen: <https://riptutorial.com/de/apache/topic/9630/apache-flume>

Kapitel 4: So erstellen Sie einen virtuellen Host in Apache

Bemerkungen

Der Haupteinstiegspunkt für `VirtualHost` von [Apache](#) befindet sich in der [Dokumentation zu Apache Virtual Host](#). Von dort aus haben Sie eine allgemeine Dokumentation zur Konfiguration des virtuellen Hosts sowie eine Referenzdokumentation zu `VirtualHost` und zugehörigen Anweisungen.

Examples

Namensbasierte Konfiguration des virtuellen Hosts

Namensbasiertes virtuelles Hosting auf Apache wird auf der [Apache-Website](#) als solches beschrieben:

Beim namenbasierten virtuellen Hosting muss der Server den Hostnamen als Teil der HTTP-Header melden. Bei Verwendung dieser Technik können viele verschiedene Hosts dieselbe IP-Adresse verwenden.

Daher können auf diese Weise mehrere Websites auf einem Server gehostet werden. Unter Ubuntu befinden sich die Konfigurationsdateien in `/etc/apache2/sites-available`. In diesem Verzeichnis finden Sie `000-default.conf`. Dies ist die Standardkonfiguration. Alle Anforderungen werden an diese Konfigurationsdatei gesendet, bis andere eingerichtet wurden.

Um einen virtuellen Host einzurichten, wird hier **example.com** verwendet, Sie sollten ihn jedoch durch Ihre **domain.com** ersetzen. Kopieren Sie die Standarddatei:

```
cp 000-default.conf example.com.conf
```

Die Konfigurationsdatei kann die folgenden Anweisungen enthalten:

```
<VirtualHost *:80>
    ServerAdmin admin@example.com
    ServerName example.com
    ServerAlias www.example.com

    DocumentRoot /var/www/example.com/html

    ErrorLog /var/log/apache/logs/error.log
    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache/logs/access.log combined
</VirtualHost>
```

- Die erste Zeile gibt an, dass alle Anforderungen an Port 80 (Standard-HTTP-Port) abgeglichen werden sollen. Sie können auch eine IP-Adresse anstelle von * verwenden, die die IP des Servers ist.
- `ServerAdmin` sind die Kontaktdaten des Website-Administrators, die zur Anzeige mit HTTP-Fehlermeldungen verwendet werden.
- `ServerName` ist der Domainname der Website.
- `ServerAlias` ist ein sekundärer Name der Website, normalerweise `www.domain.com`
- `DocumentRoot` ist der Stammordner, der geladen wird, wenn wir eine Website durchsuchen.
- `ErrorLog` ist die Datei, in die Fehler geleitet werden
- `LogLevel` . ist die Ebene der Fehler, die an das Protokoll gesendet werden sollen
- `CustomLog` ist die Datei, in die Zugriffsinformationen geleitet werden

Bearbeiten Sie die Datei und ersetzen Sie `example.com` mit Ihrem Website-Domännennamen und dem entsprechenden Verzeichnis für die Websitedateien.

Speichern Sie die Datei und aktivieren Sie die Site mit dem folgenden Apache-Befehl:

```
sudo a2ensite example.com.conf
```

Apache neu laden

```
sudo service apache2 reload
```

Noch ein paar Dinge, die überprüft werden müssen:

- Stellen Sie sicher, dass Ihr DNS für Ihre Domain für die richtige IP-Adresse eingerichtet ist (dies kann einige Zeit dauern).
- Stellen Sie sicher, dass Ihr Port 80 in der Firewall geöffnet ist
- Stellen Sie sicher, dass Ihre Dateiberechtigungen für die Serverdateien richtig eingerichtet sind. Eigentümer sollte `www-data` sein: `www-data` und Verzeichnisberechtigungen sollten 750 sein und Dateiberechtigungen sollten 640 sein.

Ihr virtueller Host sollte betriebsbereit sein! Sie können dies für andere Websites auf demselben Server mit einer anderen Konfigurationsdatei (mit derselben Namenskonvention) und verschiedenen Verzeichnissen unter `/var/www` wiederholen.

Virtueller Host für PHP-Entwicklung

In diesem Beispiel wird die PHP-Fehlerprotokollierung in einer virtuellen Host-Site für die Entwicklung und das Debugging gesteuert. Annahmen

- Das PHP-Modul wurde installiert.
- Entwicklungsumgebung ist nicht für die Produktion gedacht.

```
<VirtualHost *:80>
    ServerName example.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/www/domains/example.com/apache.error.log
    CustomLog /var/www/domains/example.com/apache.access.log common
```

```
php_flag log_errors on
php_flag display_errors on
php_value error_reporting 2147483647
php_value error_log /var/www/domains/example.com/php.error.log
</VirtualHost>
```

Hinweis: Die Konfiguration des virtuellen Hosts ist nur für die Entwicklung vorgesehen, da `display_errors` aktiviert ist und Sie dies nicht in der Produktion möchten.

Virtueller Host in WAMP

Vorausgesetzt, Sie arbeiten mit Windows 7 PC

Schritt 1: GOTO -> C: \ Windows \ System32 \ drivers \ etc Wenn Sie eine Datei mit dem Namen "hosts" finden, kopieren Sie sie und fügen Sie sie an derselben Stelle ein. Dort wird eine Kopie der Hosts erstellt.

Jetzt müssen wir einige Änderungen an dieser Datei vornehmen, aber wenn Sie versuchen, sie mit einem Editor wie Notepad oder Notepad ++ zu bearbeiten, können Sie die Datei nicht speichern.

Kopieren Sie nun dieselbe Datei erneut und fügen Sie sie auf Ihrem Desktop ein. Jetzt können Sie diese Datei problemlos bearbeiten.

Sie finden einen oder mehrere Einträge wie: 127.0.0.1 localhost In dieser Datei. Fügen Sie nun eine weitere Zeile unter dieser Zeile hinzu, zum Beispiel: 127.0.0.1 myproject1.local Auf diese Weise haben Sie eine neue Unterdomäne "myproject1.local" definiert, die anstelle von "localhost / myproject1" verwendet werden kann.

Schritt 2: Okay, jetzt ist es an der Zeit, den Root-Pfad für den Zugriff auf diese neu erstellte Domäne zu definieren, oder? GOTO: C: \ wamp \ bin \ apache \ Your-Apache-Version \ conf \ extra Hier finden Sie eine Datei mit dem Namen "httpd-vhosts". Öffnen Sie es im Editor und fügen Sie die folgenden Zeilen ein.

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummys.example.com
    DocumentRoot "c:/wamp/www/myproject1/"
    ServerName myproject1.local
    ErrorLog "logs/myproject1.local-error.log"
    CustomLog "logs/myproject1.local.log" common
</VirtualHost>
```

Nun sind Sie fast da, um auf das Projekt zuzugreifen, das unter "c: / wamp / www / myproject1 /" liegt.

Schritt 3: GOTO: C: \ wamp \ bin \ apache \ your-Apache-Version \ conf

Suchen Sie eine Datei mit dem Namen „**httpd.conf**“, kopieren Sie sie und fügen Sie sie zur Sicherheit an derselben Stelle ein. Öffnen Sie die Datei im Editor, und suchen Sie nach dem Wort "**Virtuelle Hosts**". Unten finden Sie die Zeile "**Include conf / extra / httpd-vhosts.conf**". Wenn

es kommentiert ist, machen Sie es unkommentiert und starten Sie die Dienste Ihres Wamp-Servers neu.

Gehen Sie zu Ihrem Webbrowser und schreiben Sie `myproject1.local`. Das Projekt wird jetzt ausgeführt.

Nun könnte es vorkommen, dass Ihr localhost nicht funktioniert, wenn Sie localhost als URL verwenden. Keine Sorgen... fügen Sie diesen Code in die Datei „`httpd-vhosts`“ ein .

```
<VirtualHost *:80>
    ServerAdmin webmaster@dummy.example.com
    DocumentRoot "c:/wamp/www"
    ServerName localhost
    ErrorLog "logs/localhost-error.log"
    CustomLog "logs/localhost.log" common
</VirtualHost>
```

Starten Sie alle Dienste von WAMP erneut, die Arbeit ist erledigt.

Danke & Prost Chintan Gor

1) IP-basierte vhosts 2) Mehrere vhosts mit demselben Port 3) Definieren von vhosts mit Makro (Apache2.4)

1) IP-basierte vhosts

```
<VirtualHost 192.168.13.37>
    ServerName example.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/log/example.com/error.log
    CustomLog /var/log/example.com/access.log common
</VirtualHost>

<VirtualHost 192.168.47.11>
    ServerName otherurl.com
    DocumentRoot /srv/www/htdocs/otherurl.com/html
    ErrorLog /var/log/otherurl.com/error.log
    CustomLog /var/log/otherurl.com/access.log common
</VirtualHost>
```

Ändern Sie einfach den Port in Ihre IP-Adresse (n). Der Port ist für die Entscheidung, welcher vhost ausgewählt wird, irrelevant.

2) Mehrere vhosts mit demselben Port

Da NameVirtualHost nicht mehr benötigt wird, können Sie einfach mehrere vhosts mit demselben Port schreiben.

```
<VirtualHost *:80>
    DocumentRoot /srv/www/htdocs/otherurl.com/html
    ErrorLog /var/log/otherurl.com/error.log
    CustomLog /var/log/otherurl.com/access.log common
</VirtualHost>
```



```

<VirtualHost *:80>
    ServerName example.com
    ServerAlias ex1.com ex2.com
    DocumentRoot /var/www/domains/example.com/html
    ErrorLog /var/log/example.com/error.log
    CustomLog /var/log/example.com/access.log common
</VirtualHost>

```

Hier gilt das Gegenteil: Die IP ist irrelevant, aber wenn die Anfrage an Port 80 eingeht, wird der eingegebene Name ausgewertet. Haben Sie ex1.com angerufen, wird der 2. vhost ausgewählt. Und wenn Sie eine andere URL aufgerufen haben (wie otherurl.com, aber auch example3.com), wird die erste URL ausgewählt. Sie können diesen vhost als "Fallback" verwenden, wenn Sie möchten.

3) vhosts mit Makro definieren (Apache2.4)

```

<Macro VHost $port $host>
    <VirtualHost *:$port>
        Servername $host
        DocumentRoot /srv/www/htdocs/$host
        ErrorLog /var/log/$host/error.log
    </VirtualHost>
</Macro>

Use VHost 80 example.com
Use VHost 443 secure_example.com

```

Erstellt zwei vhosts, einen für Port 80 und einen für 443, und legt die verwendeten Variablen entsprechend fest.

Erzwingen Sie HTTPS mithilfe eines virtuellen Hosts

Verwenden Sie **Redirect**, um Benutzer zu zwingen, sich mit der sicheren URL zu verbinden.

```

<VirtualHost *:80>
    ServerName example.com
    SSLProxyEngine on
    Redirect permanent / https://secure_example.com/
</VirtualHost>

```

Der Rest der Konfiguration kann im virtuellen SSL-Host (Port 443) abgelegt werden, da alles umgeleitet wird.

```

<VirtualHost _default_:443>
    ServerName secure_example.com
    ServerAdmin webmaster@example.com
    DocumentRoot /var/www/domains/secure_example.com/html
    ErrorLog /var/log/secure_example.com/error.log
    CustomLog /var/log/secure_example.com/access.log common
    SSLEngine On
    ...
</VirtualHost>

```

So erstellen Sie einen virtuellen Host in Apache online lesen:

<https://riptutorial.com/de/apache/topic/4856/so-erstellen-sie-einen-virtuellen-host-in-apache>

Credits

S. No	Kapitel	Contributors
1	Erste Schritte mit Apache	Community , fab , Flamewires , hjpotter92 , James , Katie , Kuhan , Nicholas Qiao , Rich Bowen
2	.htaccess-Dateien in Apache	Chintan Gor , Deltik , ezra-s , Luke Bearl , Rich Bowen , SimpleAnecdote
3	Apache Flume	Vinod Kumar
4	So erstellen Sie einen virtuellen Host in Apache	Chintan Gor , Clutch , fab , Harikrishnan , Hello Fishy , Katie , Olaf Dietsche