

So installieren und sichern Sie Apache auf Debian11

Link: <https://tecadmin.net/how-to-install-apache-on-debian-11/>

Der Apache HTTP-Webserver ist einer der weit verbreiteten Webserver, insbesondere bei Linux-Distributionen, bei denen es sich um eine kostenlose, plattformübergreifende Plattform handelt, die von einer Vielzahl von Websites im Internet verwendet wird. Der Apache-Webserver verwendet HTTP, um die Anfrage zu verarbeiten und Webinformationen zu unterhalten. Apache hat unzählige nützliche Funktionen und seine Funktionalität kann mit zusätzlichen Modulen erweitert werden. Es ermöglicht Programmierern auch, ihre Arbeit im Internet zu veröffentlichen.

In diesem Artikel werden wir also die Installation des Apache-Webserver besprechen und wie man ihn nach der Installation auf Debian 11 sichert.

Anforderungen

Vor der Installation müssen Sie beim Debian-System mit Zugriff auf alle sudo-Berechtigungen angemeldet sein. Wir empfehlen außerdem, die [Ersteinrichtung des Servers](#) auf neu installierten Debian 11-Systemen abzuschließen.

Schritt 1 – Apache auf Debian installieren

Die neueste Version der Apache-Pakete ist im standardmäßigen Debian 11-Repository verfügbar. So können wir es direkt über den Paketmanager installieren.

Öffnen Sie nach der Anmeldung das Terminal und aktualisieren Sie den apt-Cache mit dem unten genannten Befehl:

```
sudo apt update
```

Installieren Sie nach der Aktualisierung des apt-Cache jetzt den Apache2 auf Ihrem Debian 11 Bullseye mit dem Befehl:

```
sudo apt install apache2
```

Drücken Sie „y“ für jede Bestätigung, die vom Installationsprogramm angefordert wird.

Sobald der Installationsvorgang abgeschlossen ist. Überprüfen Sie die installierte Apache-Version, indem Sie den folgenden Befehl ausführen:

```
apache2 -v  
Ausgabe:  
Serverversion: Apache/2.4.48 (Debian)
```

Server gebaut: 2021-08-12T11:51:47

Eine andere Möglichkeit, die Installation von Apache zu überprüfen, besteht darin, mit der IP-Adresse oder dem Hostnamen Ihres Servers auf die Apache2-Standardseite zuzugreifen. Wenn Sie Ihren Hostnamen nicht kennen, führen Sie zuerst den unten genannten Befehl aus:

```
hostname -I
```



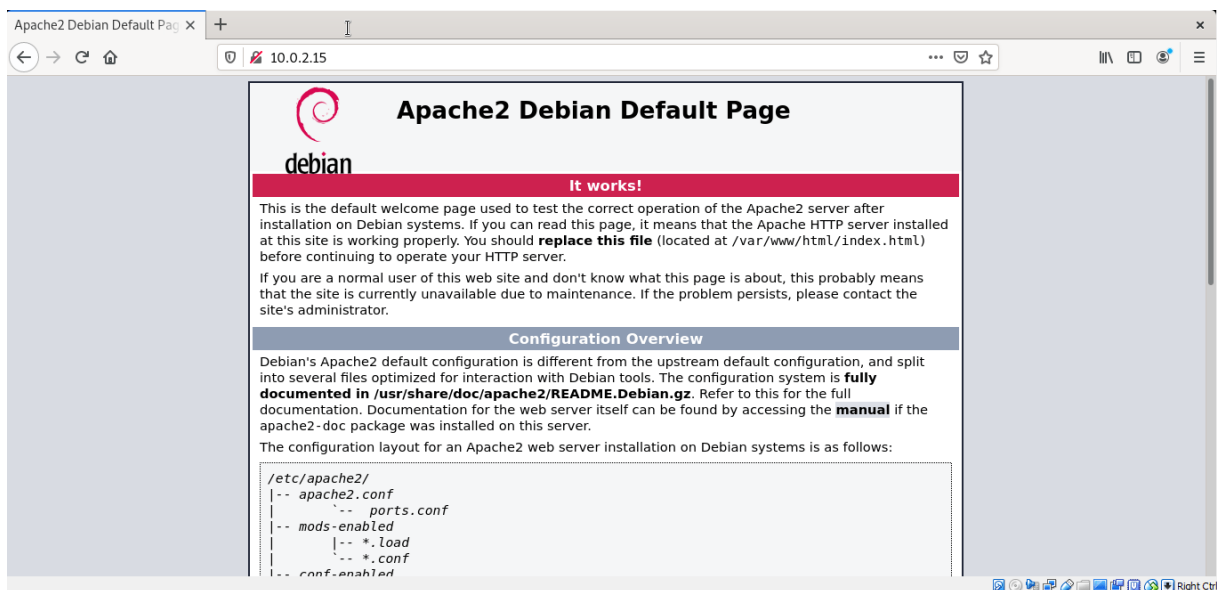
```

rahul@tecadmin:~$ hostname -I
10.0.2.15
rahul@tecadmin:~$

```

Überprüfen Sie die IP-Adresse des lokalen Systems

Geben Sie den Hostnamen oder die IP-Adresse Ihres Servers in die URL-Leiste des Browsers ein und drücken Sie die Eingabetaste. Die Apache2 Debian-Standardseite wird wie unten gezeigt geöffnet:



Apache-Standardseite auf Debian 11

Schritt 2 – Verwalten des Apache-Dienstes

Nach erfolgreicher Installation kann der Apache-Dienst mit „systemctl“-Befehlen verwaltet werden, führen Sie den unten genannten Befehl aus, um den Status des Servers zu überprüfen:

```
sudo systemctl status apache2.service
```

```

rahul@tecadmin: ~
rahul@tecadmin:~$ sudo systemctl status apache2.service
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor prese
   Active: active (running) since Thu 2021-08-19 05:59:44 EDT; 5min ago
     Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2606 (apache2)
    Tasks: 55 (limit: 2181)
   Memory: 10.8M
     CPU: 188ms
   CGroup: /system.slice/apache2.service
           └─2606 /usr/sbin/apache2 -k start
             └─2608 /usr/sbin/apache2 -k start
               └─2609 /usr/sbin/apache2 -k start

Aug 19 05:59:44 tecadmin systemd[1]: Starting The Apache HTTP Server...
Aug 19 05:59:44 tecadmin apachectl[2605]: AH00558: apache2: Could not reliably
Aug 19 05:59:44 tecadmin systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)

```

Überprüfen Sie den Apache-Dienststatus auf Debian 11

Drücken „q“ zum Beenden Einige Befehle zum Verwalten des Apache-Dienstes in Debian 11 sind:

Um den Server zu starten, verwenden Sie den Befehl:

```
sudo systemctl start apache2.service
```

Um den Dienst zu stoppen, ersetzen Sie in ähnlicher Weise start durch ein stop im obigen Befehl:

```
sudo systemctl stop apache2.service
```

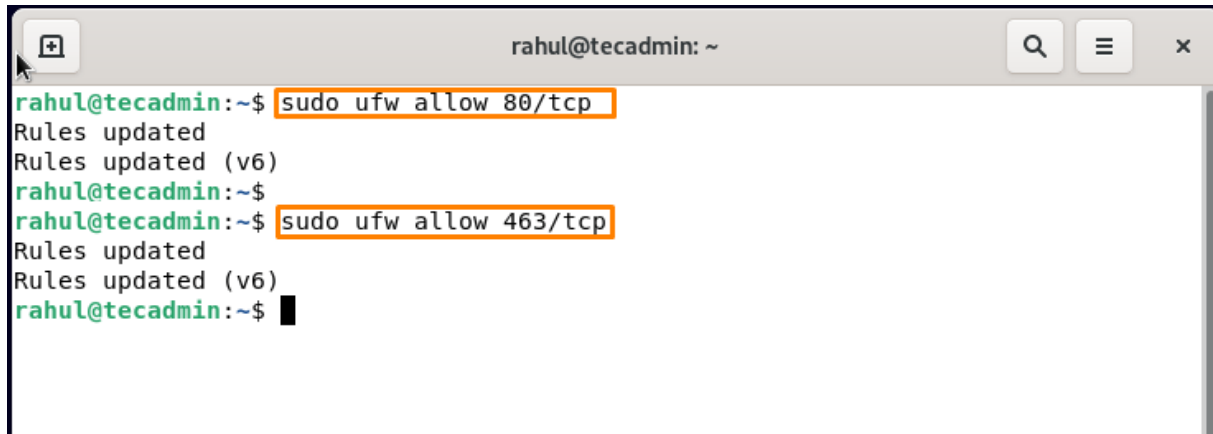
Der Dienst kann neu gestartet werden mit:

```
sudo systemctl restart apache2.service
```

Schritt 3 – Konfigurieren der Firewall-Einstellungen

Wenn Ihr System über eine Firewall verfügt, müssen Sie den Zugriff auf bestimmte Webports autorisieren, damit externe Benutzer sie verwenden können. Führen Sie den unten genannten Befehl aus, um Port 80 (HTTP) und 443 (HTTPS) im Debian-Terminal zuzulassen:

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
```

A terminal window titled 'rahul@tecadmin: ~' showing two commands being executed. The first command is 'sudo ufw allow 80/tcp' and the second is 'sudo ufw allow 463/tcp'. Both commands are highlighted with an orange box. The output for each command is 'Rules updated' and 'Rules updated (v6)'.

```
rahul@tecadmin:~$ sudo ufw allow 80/tcp
Rules updated
Rules updated (v6)
rahul@tecadmin:~$ sudo ufw allow 463/tcp
Rules updated
Rules updated (v6)
rahul@tecadmin:~$
```

HTTP- und HTTPS-Port in UFW zulassen

Überprüfen Sie nun, indem Sie den Status überprüfen:

```
sudo ufw status
```

wenn es nicht aktiv ist, um es zu verwenden:

```
sudo ufw enable
```

Schritt 4 – Erstellen eines virtuellen Hosts in Apache

In Apache ermöglichen virtuelle Hosts den Betrieb zahlreicher Websites auf einem einzigen Server. Auf dem Apache-Webserver erstellen wir einen virtuellen Host. Um dies zu erreichen, erstellen wir zunächst eine Website namens sample.com mit dem Serverblock, der standardmäßig mit Apache geliefert wird.

Beginnen wir damit, den ersten virtuellen Host Ihres Apache-Servers einzurichten. Wir verwenden die Beispieldomain als „sample.com“, aber Sie können sie nach Ihren Wünschen benennen:

```
sudo mkdir -p /var/www/sample.com
```

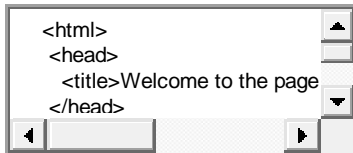
Ändern Sie nun die Berechtigungen und den Besitzer mit dem unten genannten Befehl:

```
sudo chown -R www-data:www-data /var/www/sample.com
sudo chmod -R 755 /var/www/sample.com
```

Wenn Sie den unten genannten Befehl ausführen, um unsere Website testdomain.info zu testen, erstellen wir nun eine Beispiel-Indexseite. Um dies zu erreichen, verwenden wir den Nano-Editor, um eine HTML-Datei zu generieren, die so aussieht:

```
sudo nano /var/www/sample.com/index.html
```

Fügen Sie den unten genannten Inhalt in die Indexseite ein und drücken Sie Strg+O, um die Datei zu speichern, und Strg+X, um die Datei zu verlassen und zum Terminal zurückzukehren:



```
<html>
<head>
  <title>Welcome to the page
</head>
```

```
1 <html>
2 <head>
3  <title>Welcome to the page sample.com!</title>
4 </head>
5 <body>
6  <h1>Congratulations! Your sample.com server succeeded!</h1>
7 </body>
8 </html>
```

Wenn Sie den unten genannten Befehl in einem Terminal ausführen, erstellen wir eine virtuelle Hostdatei, die den Inhalt des Servers bereitstellt:

```
sudo nano /etc/apache2/sites-available/sample.com.conf
```

Eine Textdatei wird geöffnet, fügen Sie den folgenden Inhalt ein:



```
<VirtualHost *:80>
  ServerAdmin admin@sample.c
  ServerName sample.com
  ServerAlias www.sampe.com
```

```
1 <VirtualHost *:80>
2  ServerAdmin admin@sample.com
3  ServerName sample.com
4  ServerAlias www.sampe.com
5  DocumentRoot /var/www/sample.com
6  ErrorLog ${APACHE_LOG_DIR}/error.log
7  CustomLog ${APACHE_LOG_DIR}/access.log combined
8 </VirtualHost>
```

Drücken Sie **Strg+O** , um die Datei zu speichern, und **Strg+X** , um die Datei zu verlassen und zum Terminal zurückzukehren.

Schritt 5 – Aktivieren der Domänenkonfiguration

Führen Sie den folgenden Befehl aus, um die virtuelle Hostdatei zu aktivieren:

```
sudo a2ensite sample.com.conf
```

Deaktivieren Sie die standardmäßige Apache-Konfiguration, indem Sie den unten genannten Befehl ausführen:

```
sudo a2dissite 000-default.conf
```

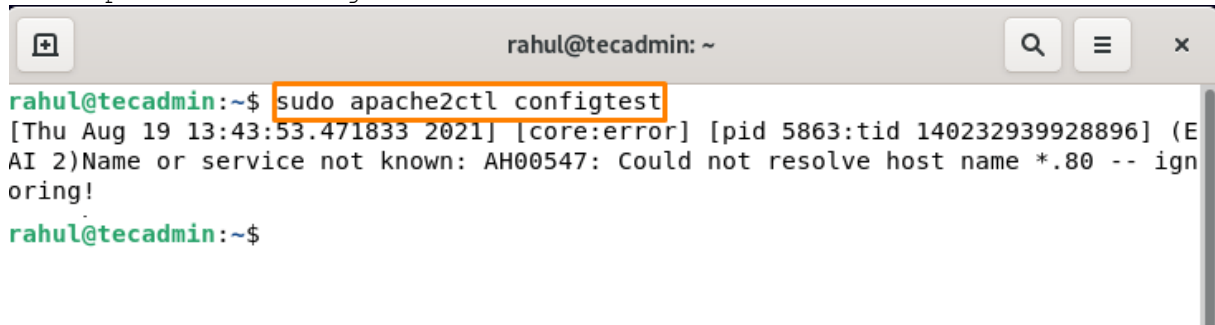
Neue Änderungen an Apache werden angewendet, indem der unten erwähnte Befehl ausgeführt wird:

```
sudo systemctl restart apache2
```

Schritt 6 – Hostnamenfehler beheben

Jetzt müssen wir unsere Konfiguration auf Syntaxfehler überprüfen, um die Konfiguration zu testen, führen Sie den unten genannten Befehl aus:

```
sudo apache2ctl configtest
```



```
rahul@tecadmin:~$ sudo apache2ctl configtest
[Thu Aug 19 13:43:53.471833 2021] [core:error] [pid 5863:tid 140232939928896] (EAI 2)Name or service not known: AH00547: Could not resolve host name *.80 -- ignoring!
rahul@tecadmin:~$
```

Problem mit System-Hostname konnte mit Apache nicht behoben werden

Dies wird einen Fehler verursachen, aber keine Sorge, wir werden ihn beheben. Erstellen Sie eine neue Konfiguration „servername.conf“ und bearbeiten Sie sie in einem Texteditor:

```
sudo nano /etc/apache2/conf-available/servername.conf
```

Fügen Sie nun folgenden Inhalt in die Datei ein:

```
Servername sample.com
```

Drücken Sie Strg+O, um die Datei zu speichern, und Strg+X, um die Datei zu verlassen. Stellen Sie sicher, dass Sie „sample.com“ durch Ihren tatsächlichen Domainnamen ersetzen. Führen Sie nun den unten genannten Befehl aus, um den Namen des conf-Servers zu aktivieren:

```
sudo a2enconf servername
```

Führen Sie nun den obigen Befehl erneut aus, um die Konfiguration zu testen:

```
sudo apache2ctl configtest
```

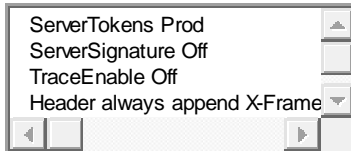
Sie werden sehen, dass der Hostnamenfehler jetzt behoben ist.

Schritt 7 – So sichern Sie Apache2 unter Debian 11

Um den Apache-Server zu sichern, bearbeiten Sie die „**security.conf**“ und führen Sie den unten genannten Befehl aus, um die Datei zu öffnen:

```
sudo nano /etc/apache2/conf-enabled/security.conf
```

Fügen Sie den folgenden Inhalt in die Datei ein oder aktualisieren Sie ihn:



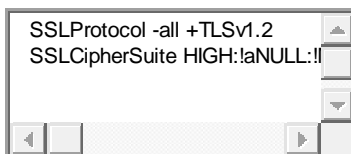
- 1 ServerTokens Prod
- 2 ServerSignature Off
- 3 TraceEnable Off
- 4 Header always append X-Frame-Options SAMEORIGIN
- 5 Header always set X-XSS-Protection: "1; mode=block"
- 6 Header always set X-Content-Type-Options: "nosniff"
- 7 Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains"
- 8 Header always edit Set-Cookie ^(.*)\$ \$1;HttpOnly;Secure

Speichern Sie die Datei und schließen Sie sie.

Stellen Sie die serverweite SSLCipherSuite und das SSL-Protokoll so ein, dass sichere Chiffren verwendet werden, um die Website durch Bearbeiten bereitzustellen `ssl.conf` Datei:

```
sudo nano /etc/apache2/mods-enabled/ssl.conf
```

Fügen Sie nun den unten geschriebenen Inhalt in die Datei ein und drücken Sie Strg+O, um die Datei zu speichern, und Strg+X, um die Datei zu verlassen:



- 1 SSLProtocol -all +TLSv1.2
- 2 SSLCipherSuite HIGH:!aNULL:!MD5

Führen Sie nun den Reload-Befehl von Apache aus, um die Konfiguration zu speichern:

```
sudo systemctl restart apache2.service
```

Das ist es. Sie haben den Apache-Server erfolgreich installiert und gesichert.

Fazit

Apache Web Server ist ein Open-Source-Server, der von vielen Websites im Internet verwendet wird und es Entwicklern ermöglicht, ihre Arbeit im Internet zu veröffentlichen. Dieser Server ist auf allen Betriebssystemen verfügbar, aber in diesem Artikel besprechen wir seine Installation auf der neuesten Version von Debian (Linux OS) und erklären auch, wie man ihn nach seiner erfolgreichen Installation testet und sichert. Nachdem Sie diese Anleitung durchgearbeitet haben, können Sie Apache2 erfolgreich auf Debian 11 Bullseye installieren und den Server konfigurieren.