

# Subversion auf dem Raspberry Pi einrichten

## Einleitung

Unter [Download](#) findet ihr eine PDF-Datei mit der Einrichtung und Benutzung von Subversion.

Hier möchte ich eine kurze Anleitung für die Einrichtung von Subversion auf dem Raspberry Pi schreiben.

Ich habe bei mir auf meinem Raspi Subversion für die kleinere Projekte eingerichtet, wobei der Zugriff über HTTPS-Protokoll stattfindet. Damit kann ich von Überall auf die Daten verschlüsselt zugreifen.

## Betriebssystem

Als Betriebssystem habe ich auf dem Raspi "Raspbian" eingerichtet. Aus dem Grund, dass dieses auf dem Debian-System basiert, sollte die folgende Anleitung auch für alle Debian-Systeme gelten.

Für meine Zwecke habe ich eine MicroSD-Karte mit der Größe von 128GB gewählt. Diese soll dafür ausreichen und ist halbwegs preiswert.

Bevor es mit der Installation und Einrichtung von Subversion weiter geht, sollte das Betriebssystem auf den neusten Stand gebracht werden. Dafür müssen folgende Befehle in der Shell ausgeführt werden:

```
sudo apt-get update
sudo apt-get upgrade
```

## Software

Für die Einrichtung von Subversion werden folgende Softwarepakete benötigt:

Paketname	Beschreibung
apache2	Webserver
libapache2-svn	Subversion-Library für Apache2

subversion	Subversion
subversion-tools	Subversiontools
rapidSVN	Grafischer Client. Dieses Paket kann zus. ausgewählt werden, wenn auch die grafische Desktop-Umgebung mitinstalliert wurde. auch auf dem Server die Verbindungen zum Subversion-Server aufgebaut und getestet werden.
vim	Erweiterter vi-Editor für die Shell

Mit dem folgenden Befehl werden die Pakete installiert:

```
sudo apt-get install apache2 libapache2-svn subversion subversion-  
tools vim
```

Das Betriebssystem installiert alle ausgewählten Pakete mit den automatisch ermittelten Abhängigkeiten.

Wenn die Installation fertig ist, muss überprüft werden, ob ein für die Authentifizierung benötigter Apache-Modul installiert und aktiviert ist, was mit dem folgenden Befehl erledigt wird:

```
sudo a2enmod authz_user
```

## Sicherheit

Um meinen Quellcode bei der Übertragung übers Internet vor fremden Augen zu schützen, möchte ich gerne den Datenverkehr verschlüsseln. Dafür muss das SSL-Protokoll für den Apache-Webserver mit folgenden Befehlen aktiviert werden:

```
sudo a2enmod ssl
```

```
sudo a2ensite default-ssl
```

Für die Verschlüsselung wird ein Zertifikat benötigt. Für mich reicht auch ein selbst erzeugter Zertifikat. Diesen erzeuge ich mit dem folgenden Befehl:

```
sudo openssl req -new -x509 -nodes -out /etc/ssl/certs/server.crt  
-keyout /etc/ssl/private/server.key -days 3650
```

Dabei müssen Sie noch ein paar Fragen beantworten, um z.B. Land, Name, usw. festzulegen. Wichtig ist die COMMON-Eingabe. Hier muss die Adresse des Servers angegeben werden:

z.B. www.test.com.

Aus Faulheit 😊 habe ich die Gültigkeit des Zertifikates auf 3650 Tage (ca. 10 Jahre) gesetzt. Ihr könnt aber die Dauer nach Belieben selbst anpassen.

Jetzt muss ich dem Apache noch mitteilen, welcher Zertifikat benutzt werden soll. Dafür muss ich mit dem Texteditor in der Datei "/etc/apache2/sites-available/default-ssl.conf" folgende 2 Zeilen finden

```
SSLCertificateFile /etc/ssl/certs/ssl-cert-snackoil.pem  
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snackoil.key
```

und diese so anpassen:

```
SSLCertificateFile /etc/ssl/certs/server.crt  
SSLCertificateKeyFile /etc/ssl/private/server.key
```

Zum Schluss muss ich den Web-Server neu starten, um die Konfiguration zu aktivieren:

```
sudo /etc/init.d/apache2 restart
```

Beim Aufruf der Seite im Browser muss ich das Zertifikat akzeptieren und dafür eine Ausnahme hinzufügen, da dieses nicht öffentlich signiert ist.

## Subversion einrichten

Jetzt bin ich an einem Punkt angekommen, an dem die Einrichtung der

Versionsverwaltung beginnt.

Zuerst lege ich ein Verzeichnis für die Repositories an, in dem dann weiteres Unterverzeichnis je Projekt angelegt wird:

```
sudo mkdir /var/svn
sudo mkdir /var/svn/projek1
sudo mkdir /var/svn/projek2
```

Jetzt erstelle ich je Projekt ein Repository, in dem folgende Befehle benutzt werden:

```
sudo svnadmin create /var/svn/projekt1
sudo svnadmin create /var/svn/projekt2
```

Da die Verzeichnisse und Repositories von dem Benutzer „root“ angelegt wurden, müssen dem Webserver die Zugriffsrechte (Benutzer und Gruppe) erteilt werden, damit dieser die Repositories lesen und beschreiben kann:

```
sudo chown -vR www-data:www-data /var/svn/*
```

## Benutzer einrichten

Für den Zugriff auf Subversion möchte ich eine Anmeldung verlangen, um das ganze noch mal abzusichern. Mit folgendem Befehl wird die Datei erstellt, in der die Benutzer inkl. Kennwörter abgelegt werden, wobei der Parameter "-c" die Datei neu erstellen lässt und nur beim ersten Aufruf benutzt werden darf:

```
sudo htpasswd -cm /etc/apache2/dav_svn.passwd user1
sudo htpasswd -m /etc/apache2/dav_svn.passwd user2
```

Beim Ausführen des Befehls gebe ich für jeden Benutzer ein Kennwort ein.

## Repositories dem Webserver bekanntmachen

Um auf die Repositories zugreifen zu können, muss ich diese dem Webserver bekanntmachen.

Die Konfiguration erfolgt in der Datei "/etc/apache2/mods-

available/dav\_svn.conf", die nach der Installation schon vorhanden ist und die Beispielkonfiguration mit Beschreibung enthält.

Ich lasse das Beispiel unverändert und füge am Ende der Datei folgenden Inhalt hinzu:

```
<Location /projekt1>
#Repository 1
DAV svn
SVNPath /var/svn/projekt1
AuthType Basic
AuthName „Projekt 1“
AuthUserFile /etc/apache2/dav_svn.passwd
AuthzSVNAccessFile /etc/apache2/dav_svn_projekt1.authz
Require valid-user
</Location>
<Location /projekt2>
#Repository 2
DAV svn
SVNPath /var/svn/projekt2
AuthType Basic
AuthName „Projekt 2“
AuthUserFile /etc/apache2/dav_svn.passwd
AuthzSVNAccessFile /etc/apache2/dav_svn_projekt2.authz
Require valid-user
</Location>
```

In dieser Konfiguration ist noch eine Datei "/etc/apache2/dav\_svn\_projektX.authz" je Projekt aufgetaucht, die ich noch nicht erwähnt habe. In dieser Datei werden die Zugriffsrechte auf das Repository verwaltet.

Mit dem folgenden Befehl erstelle ich die Datei:

```
sudo vim /etc/apache2/dav_svn_projekt1.authz
```

und füge folgenden Inhalt ein:

```
[/]
#Das ist Projekt1
#Einzelnen Benutzern die Rechte zuteilen
user1 = rw
user2 = r
#für die restlichen Benutzer den Zugriff sperren
* =
```

Das gleiche führe ich auch für die Datei

"/etc/apache2/dav\_svn\_projekt2.authz" durch und vergebe die passenden Zugriffsrechte.

Bei der Rechtevergabe habe ich "r" für Leserechte, "w" für Schreibrechte und " " für keine Rechte benutzt.

Jetzt muss ich nur noch den Apache-Server neu starten, um die Konfiguration zu aktivieren:

```
sudo /etc/init.d/apache2 restart
```

## Verbindung zu den Repositories

Nach der Einrichtung der Repositories und des Webservers kann ich schon auf diese zugreifen. Der einfachste Zugriff zum Testen ist ein Aufruf der Repository-Adressen im Browser, z.B.

```
https://192.168.0.2/projekt1/  
https://192.168.0.2/projekt2/
```

Die IP-Adresse ist durch die richtige von dem Raspi bzw. durch eine externe Adresse zu ersetzen. Jetzt muss noch die Weiterleitung für den Port 443 in dem Router für den Zugriff aus dem Internet eingerichtet werden.

Der Browser-Zugriff ist aber nicht für die Arbeit eines Entwicklers geeignet, weswegen dafür andere Tools benutzt werden müssen. Ich könnte natürlich bei mir auf dem Windows-PC auch Subversion installieren und über die mitgelieferten Konsolen-Tools darauf zugreifen. Das ist aber bestimmt bei der täglichen Arbeit unbequem. Deswegen habe ich nach einer Alternative gesucht und gefunden. Es gibt sehr viele Client-Programme mit grafischer Oberfläche, die sogar kostenlos benutzt werden können, z.B. RapidSVN, TortoiseSVN usw..

Für den Zugriff auf die Repositories aus den Programmen werden die gleichen https-Adresse benutzt.