

# Apache2

- [Apache2 - SSL Zertifikat mit Let's encrypt für Webserver bereitstellen](#)
- [Apache2 - Webspaces passwortgeschützt mit einzelner Ausnahme](#)
- [Apache2 - SSL erzwingen bzw. HTTP-Zugriffe umleiten](#)
- [Apache 2 - SVN](#)

# Apache2 - SSL Zertifikat mit Let's encrypt für Webserver bereitstellen

1. letsencrypt über die Paketverwaltung installieren: `#aptitude install letsencrypt`
2. Zertifikat generieren:
  1. `# certbot certonly --manual --rsa-key-size 4096`
  2. Die Fragen beantworten und z.B. die Domain festlegen.
3. Verifikations-Datei auf dem Webserver anlegen:
  1. Hinweis: Die kryptischen Bezeichner sind bei euch anders - ersetzt sie einfach.
  2. `mkdir /var/www/letsencrypt/.well-known/acme-challenge`  
`cd /var/www/letsencrypt/.well-known/acme-challenge`  
`echo -n N_aDTbuhynhvqhGaqs10VHs1_Bl1A4Z9rHtFhFeV1cA.DGo-QeCJ79p7eoFfCvjK4Np9a_RsbbtjWteKY0QML0I >`  
`N_aDTbuhynhvqhGaqs10VHs1_Bl1A4Z9rHtFhFeV1cA`
4. Dem Apache2 noch die neuen Zertifikate beibringen:
  1. in eure config unter `/etc/apache2/site-enabled/eureseite.conf` folgende Zeilen unter HTTPS einfügen:  
`ssl_certificate /etc/letsencrypt/live/meinedomain.de/fullchain.pem;`  
`ssl_certificate_key /etc/letsencrypt/live/meinedomain.de/privkey.pem;`
5. Danach die Verzeichnisstruktur der Verifikations-Datei wieder löschen.
6. Apache2 Server neustarten: `#/etc/init.d/apache2 restart`

Fertig!

Quelle: <https://serverfault.com/questions/750902/how-to-use-lets-encrypt-dns-challenge-validation>

# Apache2 - Webspaces passwortgeschützt mit einzelner Ausnahme

Quelle: <https://stackoverflow.com/questions/2641646/how-to-accomplish-auth-type-none-in-apache-2-2>

Quelle: <https://gist.github.com/lokesh-webonise/5625636>

# Apache2 – SSL erzwingen bzw. HTTP-Zugriffe umleiten

## Der eigentliche vhost

Hier würden auch directory-, Authentication-, und ähnliche Anweisungen erfolgen.

```
<VirtualHost 127.0.0.1:443>
  ServerName localhost:443
  #
  SSLEngine On
      SSLCertificateFile /etc/apache2/ssl/apache.pem
</VirtualHost>
```

## Das ist der „Dummy“-vhost

Dieser **vhost** dient dann nur noch dazu um eine Umleitung auf den oberen „443-vhost“ zu machen.

```
<VirtualHost localhost:80>
  ServerName localhost
  #
  # Das folgende erzwingt SSL
  RewriteEngine On
  RewriteCond %{HTTPS} off
  RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
</VirtualHost>
```

Quelle: <http://blog.rvi-media.de/linux/apache2-ssl-erzwingen-bzw-http-zugriffe-umleiten/>

# Apache 2 - SVN

## SVN

### SVN Installation

- Subversion installieren:

```
apt-get update  
apt-get install subversion
```

### Speicherort für Repositories anlegen

- Ordner erstellen wo alle Repositories gespeichert werden sollen

```
# mkdir /var/svn-repos/
```

- Rechte für Apache2 vergeben

```
chown www-data:www-data /var/svn-repos
```

### SVN Repository erstellen

```
svnadmin create /var/svn-repos/repo123
```

### Dateirechte für Apache2 für Repository vergeben

```
chown -R www-data:www-data /var/svn-repos/repo123
```

## Apache 2 Vorbereitungen

# Apache SVN Modul installieren

```
apt-get install libapache2-svn
```

## Apache Module aktivieren

```
a2enmod dav  
a2enmod dav_svn  
a2enmod authz_svn
```

## Benutzerverzeichnis anlegen

- nun legen wir für den Webzugriff unsere User an

### Benutzerverzeichnis erstmalig anlegen mit der Option -c

Achtung! Bestehende Nutzer werden dabei überschrieben! Das ist zum Anlegen des Verzeichnisses, nicht eines neuen Users!

```
htpasswd -c /etc/apache2/dav_svn.passwd user1
```

- danach müsst ihr ein Passwort festlegen

### Weitere Benutzer in das Benutzerverzeichnis anlegen

```
htpasswd /etc/apache2/dav_svn.passwd user2
```

- danach müsst ihr ebenfalls ein Passwort festlegen

Mit diesen Benutzern könnt ihr jetzt arbeiten und Zugriffe steuern.

## Accessfile anlegen

- diese Datei beinhaltet die Regelungen welche User oder gruppe auf welche Repos oder Verzeichnisse zugreifen darf

- diese Datei unter /etc/apache2 speichern

- Beispiel: dav\_svn.accessfile

```
# Definitions of groups with users from operating system
[groups]
admins = user1,user2
testusers = user2

# Definitions of repositories

[user1:/]
user1 = rw
user2 = r

[user2:/]
user1 = r
user2 = rw

[test:/]
@testusers = rw
```

## Apache2 Konfiguration

Nun müsst ihr eine sog. VHOST Konfiguration anlegen und euren Ordner angeben, den wir am Anfang angelegt haben. In unserem Beispiel also `/var/svn-repos`.

Hier eine Beispiel Konfiguration:

```
<Location /svn>
    DAV svn
    SVNParentPath /var/svn-repos
    SVNListParentPath On

    AuthType Basic
    AuthName "SVN Authorization Realm"
    AuthUserFile /etc/apache2/dav_svn.passwd
    Require valid-user
    AuthzSVNAccessFile /etc/apache2/dav_svn.accessfile
</Location>
```

# Fertig!

Danach nochmal den Apache2 neustarten `/etc/init.d/apache2 restart` und ihr könnt euer SVN bequem für eure Teams einrichten ohne auf eurem Linuxsystem Unmengen an Usern anlegen zu müssen.