

DEBIAN: Web-Projekte mit SUBVERSION

cplinux // 23.07.2007 13:46

Kategorie: **SVN / CVS**

In diesem Tutorial zeige ich, wie man relativ einfach einen Subversion-Server für seine **Web-Projekte** aufsetzen kann. Installiert habe ich das Ganze auf einem Debian Etch System.

Erst mal werden wir root.

Dann müssen erst einmal die erforderlichen Pakete installiert werden **apt-get install subversion subversion-tools**

Jetzt legt man das SVN-Repository - hier nenne ich es einfach **projekt** - an. Ich würde vorschlagen, es in /var/svn/ anzulegen:

```
mkdir /var/svn
```

```
cd /var/svn
```

```
svnadmin create projekt
```

Damit haben wir schon mal das Wichtigste erledigt. Jetzt sollte das Repository angepasst werden:

Zunächste editiert man nun die Datei **conf/svnserv.conf**:

```
[general]
anon-access = none #Anonymer Zugriff auf das Repository ist nicht erlaubt
auth-access = write #Authentifizierte Benutzer bekommen Schreibberechtigung
password-db = passwd #wie sollen die Benutzer authentifiziert werden? In unserem Fall über ein passwd-File
realm = projekt.example.net #hier die Domain eintragen
```

Damit sich die Benutzer nun auch anmelden können, müssen wir das passwd-File **conf/passwd** anlegen:

```
[users]
testbenutzer = testpasswort #Benutzername = Passwort im Klartext
root = rootpasswort #Ich arbeite in der Konfiguration lokal mit root
```

Jetzt kommt ein wichtiger Punkt, der speziell für Web-Projekte wichtig ist. Ich gehe davon aus, dass wir den Checkout später so durchführen, dass das Webserver-Verzeichnis /var/www/projekt heißt. Das läßt sich ja individuell anpassen.

Es wird nun die Datei **hooks/post-commit** angelegt:

```
#!/bin/sh
REPOS="$1"
REV="$2"
/usr/bin/svn update /var/www/projekt >> /var/svn/projekt/commit.log 2>&1
```

Anmerkung:

Die Commits werden in meiner Konfiguration in die Datei commit.log gespeichert. Die Datei wird mit der Zeit ziemlich groß. Also sollte man die Datei z.B. mittels Cron-Job hin und wieder leeren.

Bsp. für ein Shell-Script:

```
#!/bin/bash
gzip -9 -f /var/svn/projekt/commit.log
touch /var/svn/projekt/commit.log
chmod 777 /var/svn/projekt/commit.log
```

Das Post-Commit-Script bewirkt nun, dass das lokale Arbeitsverzeichnis /var/www/projekt bei jedem Commit in die SVN-Datenbank automatisch aktualisiert wird.

Auf diese Weise sind Änderungen direkt auf dem Webserver sichtbar.

So weit so gut. Nun kümmern wir uns um die Berechtigungen:

```
chgrp -R subversion /var/svn/projekt
```

```
chown -R root /var/svn/projekt
```

In meiner Konfiguration arbeite ich mit dem Benutzer root, der sich neben der Gruppe root auch in der Gruppe subversion befindet. Das sollte, wenn man diese Konfiguration 1:1 übernimmt auf jeden Fall noch angepasst werden:

In der Datei **/etc/group** sollten der **Gruppe subversion alle Benutzer hinzugefügt** werden die auf den Server Zugriff bekommen, bzw. die in der **conf/passwd** stehen.

Jetzt wird es Zeit, den Dienst zum ersten Mal zu starten. Für die Zukunft empfiehlt es sich, ein kleines Script dafür anzulegen.

nohup svnservice -r /var/svn/projekt/ -d --listen-port 3690

Der Standard-Port für den Dienst ist 3690. Möchte man später mehrere SVN-Projekte auf einem Server steuern, kann man die Ports auch variieren.

Jetzt führen wir einen Checkout durch:

cd /var/www

svn checkout svn://localhost:3690 ./

Statt localhost kann man auch die IP oder den Namen des Servers verwenden. Es wird nun noch das SVN-Passwort für den Benutzer root benötigt, dann folgt der Checkout.

Ist der Checkout komplett, passt man nun einmalig die Berechtigungen an:

chgrp -R subversion /var/www/projekt

chown -R root /var/www/projekt

chmod -R 777 /var/www/projekt

Es muß nicht unbedingt 777 (Vollzugriff) für die Berechtigungen sein. Das kann man je nach Bedarf anpassen.

Gut!

Nun sollte man von einem Client aus einen Checkout durchführen. Werden nun Änderungen an das Repository übertragen, sollten die Änderungen auf dem Webserver sofort sichtbar sein.

Falles es wider Erwarten Probleme gibt, empfiehlt es sich, einen Blick in die Datei commit.log zu werfen.

Die häufigste Ursache für Fehler sind bei Subversion meiner Erfahrung nach Berechtigungsprobleme.

Noch ein paar Hinweise für den Webserver:

In der lokalen Arbeitskopie auf dem Webserver gibt es versteckte Verzeichnisse mit dem Namen **.svn**. Es empfiehlt sich, den Zugriff auf diese Verzeichnisse per **.htaccess** zu unterbinden, z.B. so:

```
<Directory ~ ".svn">
    Order allow,deny
    Deny from all
</Directory>
```