

Mise en oeuvre d'un serveur SubVersion

Le 30 mai 2008, à 16:18 par Ulhume...

L'objectif est ici de mettre en place un serveur SubVersion comprenant la mise en place du dépôt, la configuration d'apache et le paramétrage des messages de notifications.

Création du repository

Pour créer le dépôt, nous utilisons la commande `svnadmin` comme suit :

```
root# svnadmin create --fs-type fsfs /svnroot
root# ls /svnroot
  conf/ dav/ db/ format hooks/ locks/ README.txt
root#
```

Une fois la commande exécutée, le dépôt est préparé et n'attend plus que nos fichiers. Le paramètre `--fs-type fsfs` est indispensable si l'on ne veut pas se retrouver avec un dépôt utilisant le calamiteux BerkleyDB. Avec cette option, subversion va utiliser le système de fichier standard pour stocker ces données en toute sécurité.

Dans la mesure où c'est apache qui va devoir gérer ce dépôt, il est important de lui donner les droits sur le dossier `/svnroot`.

```
chown [1] apache:apache /svnroot -Rc
```

Configuration d'apache

Dans ce qui suit Apache est considéré comme installé et opérationnel. Mais pour permettre l'utilisation de Subversion, il nous faut cependant quelques modules supplémentaires : `apache-mod_dav` et `apache-mod_dav_svn`. En effet, [WebDav](#) [2] est le protocole utilisé par subversion pour communiquer ? webdav à Apache, et le second d'étendre le protocole pour gérer spécifiquement les dépôts subversion.

Ensuite, il nous faut modifier le fichier de configuration d'apache `/etc/httpd/conf/httpd2.conf` en ajoutant le bloc suivant:

```
<Location/subversion>
  DAV svn
  SVNPath /svnroot
  AuthzSVNAccessFile /svnroot/droits
</Location>
```

Cela a pour résultat d'ajouter à Apache une URL `http://mon_site/subversion` permettant l'accès au dépôt situé en `/svnroot`. L'option `AuthzSVNAccessFile` indique quant à elle le chemin vers un fichier texte contenant les droits associés au dépôt.

```
[/]
*   r
robert  rw

[/un_projet]
gaston  rw
ginette  r
```

exemple de fichier de droits subversion

Tels qu'on a défini les droits, Robert a un accès complet en lecture et écriture à tous les projets. Gaston a un accès en lecture/écriture au projet `un_projet` et Ginette (sexisme !!) n'a accès qu'en lecture à e même projet.

Maintenant pour que cela fonctionne, il faudrait qu'Apache ait le login de l'utilisateur. Il est donc nécessaire d'ajouter un bloc d'authentification à notre section `location`. Dans ce domaine, c'est à chacun de choisir : fichier apache, shadow, apache. jetez un ?il [ici](#) ^[3] pour avoir des idées sur quel système utiliser.

Une fois que tout est en place, il suffit de relancer apache et d'aller à l'URL

`http://mon_site/subversion`. En effet, Subversion fonctionne aussi bien en WebDav qu'en HTTP. Dans le second cas il est possible, si l'on a les bon droits, d'explorer le dépôt en lecture seule. Le premier cas, il faut utiliser un client subversion comme celui d'Eclipse.

Import d'un projet CVS existant

Si vous désirez garder l'historique des commit CVS, l'utilitaire à utiliser est `cvs2svn`. Il est suffisamment efficace pour avoir permis la migration d'un "petit projet" comme KDE... Sinon, vous pouvez utiliser le script suivant dont la syntaxe est :

```
convert module_cvs module_svn
```

Le code source du script `convert` est à modifier pour correspondre à vos emplacements :

```
#!/bin/sh

export CVSROOT=":pserver:user@serveur:/my_cvsroot"
export SVNROOT="http://serveur/my_svnroot/trunk/$2"
# Checkout du projet venant de cvs
cvs -d :pserver:yorannehia:cvsroot co [4] $1

# Enlever les éléments de controle CVS
find [5] $1 -name "CVS" -exec rm [6] -rf {} \;
find [5] $1 -name ".cvsignore" -exec rm [6] -rf {} \;

# Création du dossier projet
svnmkdir [7] $SVNROOT -m 'Création du projet'

# Import initial du projet vers SVN
svn import $ $SVNROOT -m 'import initial'

# suppression du checkout cvs temporaire et checkout de la version svn
```

```
|| rm [6] -rf $1
```

Rajouter des scripts de notification

Un aspect très sympathique de subversion est la facilité avec laquelle il est possible d'ajouter des actions sur les divers événements générés dans un dépôt.

Subversion a normalement créé un dossier `hooks` dans votre dossier `/svnroot`. Ce dossier contient déjà les modèles des cinq événements qu'il est capable de prendre en charge. Par défaut ces fichiers ont l'extension `.tmpl` ce qui les rends inactifs. Il faut donc ôter cette extension sur chacun des fichiers que vous désirez utiliser :

- `post-commit`, appelé après un commit.
- `post-lock`, appelé après un lock.
- `pre-lock`, appelé avant un lock.
- `pre-unlock`, appelé avant un unlock.
- `post-revprop`, appelé après un changement de révision.
- `pre-commit`, appelé avant la fin du commit.
- `start-commit`, appelé avant le commit.
- `pre-revprop-change`, appelé avant un changement de révision.

Comme exemple, imaginons que nous voulons envoyer un courriel aux développeurs d'un projet à chaque commit dans le dépôt. Pour cela nous allons créer un fichier `/svnroot/hooks/post-commit` :

```
#!/bin/sh

REPOS="$1"
REV="$2"

usr bin svnnotify -r $REV -C -d -H HTML::ColorDiff -p $REPOS -x gaston
mon_site.com=projet1 --from no-reply mon_site.com --reply-to no-reply
mon_site.com
```

`svnnotify` fait partie du projet [perl-svn-notify](#) [8] qu'il faut donc installer. Pas la peine d'aller le télécharger, il est sûrement inclus dans les dépôts de votre distribution.

Une fois installé, il faut changer les droits sur ce hook pour permettre à apache de l'exécuter. Le plus simple est de relancer la commande `chmod` que nous avons utilisé au début pour tout le dépôt :

```
|| chown [1] apache:apache svnroot -Rc
```

Maintenant à chaque commit, si ce dernier appartient au projet `projet1`, cela provoquera un envoi de courriel à gaston. On peut spécifier plusieurs arguments `-x` ainsi que plusieurs adresses séparées par des virgules. Par exemple `-x toto@mondomaine,tutu@mondomain=mon_module`.

Erreur \"502 Bad Gateway\"

Si vous avez droit un jour à un `svn: COPY of package.sh: 502 Bad Gateway` (<https://mondomaine.subversion.com/dev>), il est probable que vous ayez tenté une copie/déplacement à travers un accès HTTPS au dépôt subversion. Hors dans le cas du https, un cas qui

pose problème, celui de l'utilisation de reverse proxy sur le serveur cible (la cible est une machine apache A qui est accédée via HTTPS et qui relaye les requêtes sur une machine apache B qui gère le dépôt), ou alors de mauvaise définition du nom de serveur au niveau de la cible (leservername du vhost est mal configuré). Bref, pas beaucoup de solutions là dessus, juste de ne pas le faire 😊 . Cela ne pose problème que sur les commande COPY et MOVE, assez rare donc. Dans ces deux cas, préférez un accès pl us proche du serveur de dépôt en HTTP ou en SVN.

Conclusion

Voilà, c'est tout. Votre dépôt subversion est maintenant pleinement opérationnel. Enfin 😊 j'espère .

Liens:

- [1] <http://pwet.fr/man/linux/commandes/chown>
- [2] <http://fr.wikipedia.org/wiki/WebDav>
- [3] <http://artisan.karma-lab.net/node/22>
- [4] <http://pwet.fr/man/linux/commandes/co>
- [5] <http://pwet.fr/man/linux/commandes/find>
- [6] <http://pwet.fr/man/linux/commandes/rm>
- [7] <http://pwet.fr/man/linux/commandes/mkdir>
- [8] <http://search.cpan.org/dist/SVN-Notify/lib/SVN/Notify.pm>