

Installer Subversion et Trac

Ce tutoriel explique comment installer le gestionnaire de révisions et de source Subversion (svn) et Trac, un ensemble d'applications associés à SVN contenant un wiki, un collecteur de bugs, etc.

Installation

```
apt-get update
apt-get install trac libapache2-svn subversion subversion-tools
```

Création du dépôt pour subversion

Dans mon cas, je veux que mon dépôt subversion soit dans /home/svn/nomduprojet

Ainsi quand je voudrais créer d'autres projets, je les mettrais dans l'arbo /home/svn/ avec des noms distincts.

Je fais donc :

```
# mkdir /home/svn
# chown www-data: -R /home/svn (pour pouvoir commiter via le module webdav d'apache)
# svnadmin create --fs-type fsfs /home/svn/monprojet
```

Mon projet, avant import dans svn, est dans /home/moi/monprojet.

Dans ce répertoire, il faut créer 3 répertoires : trunk, branches et tags :

```
mkdir trunk branches tags
```

Il vous faut ensuite déplacer tous les fichiers de votre projet dans /trunk/.

Pour les importer, il suffit de procéder de la façon suivante :

```
nicolas@ange:~$ svn import /home/moi/monprojet file:///home/svn/monprojet/
Adding monprojet
Adding monprojet/trunk
Adding monprojet/trunk/i18n
Adding monprojet/trunk/i18n/francais.php
Adding monprojet/trunk/test.php
Adding monprojet/trunk/config
Adding monprojet/trunk/config/config.php
Adding monprojet/trunk/inscription.php
Adding monprojet/trunk/fonctions.php
Adding monprojet/trunk/fonctions
Adding monprojet/trunk/fonctions/authentication.php
Adding monprojet/trunk/fonctions/mysql.php
Adding monprojet/trunk/login.php
Adding monprojet/branches
Adding monprojet/tags
Committed revision 1.
nicolas@ange:~$
```

Pour vérifier que votre projet est bien créé dans svn :

```
nicolas@ange:~$ svn list --verbose file:///home/svn/monprojet
```

Voilà mon projet est dans subversion. Par contre pour travailler dessus, je dois d'abord réaliser un check-out pour me synchroniser avec Subversion.

Si je devais créer un deuxième projet, il me suffirait de répéter l'étape ci-dessus une seconde fois.

Mon répertoire projet initial ne me sert alors plus à rien - je peux donc le supprimer après en avoir fait une sauvegarde au cas où 😊

Je me mets donc dans mon nouveau répertoire de travail pour un de mes projets et je fais :

```
svn checkout file:///home/svn/monprojet/trunk
```

avec :

- /home/svn/ : mon dépôt pour subversion
- monprojet : mon projet
- trunk : le lieu où les fichiers sont initialement stockés

A ce stade, je considère que Subversion fonctionne et par conséquent que l'on peut passer à l'étape suivante. Pour vous familiariser avec Subversion, je vous invite à consulter le book de Subversion (voir lien en bas de page).

Paramétrage de Subversion dans Apache

Activer le support webdav et subversion d'apache

Vérifiez que les modules de SVN sont bien chargés dans apache :

```
# a2enmod dav
# a2enmod dav_fs
# a2enmod dav_svn
# a2enmod rewrite
```

Paramétrer votre hôte virtuel

Dans la configuration du virtual host de votre serveur apache, ajoutez :

```
<Location /svn>
  DAV svn
  SVNPath /home/svn/monprojet
  AuthType Basic
  AuthName "Mon Projet"
  AuthUserFile /home/svn/monprojet.htpasswd
  <LimitExcept GET PROPFIND OPTIONS REPORT>
    Require valid-user
  </LimitExcept>
</Location>
```

Relancer apache :

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

Allez à <http://localhost/svn/> [<http://localhost/svn/>] et vous devez voir votre projet apparaître.

Initialiser Trac

Création de l'instance Trac

Je veux que trac soit installé dans /home/trac donc je le crée sous l'utilisateur root :

```
# mkdir /home/trac/
```

Je veux ensuite initialiser mon environnement trac dans /home/trac/monprojet :

```
# trac-admin /home/trac/monprojet initenv
Creating a new Trac environment at /home/trac/monprojet
Trac will first ask a few questions about your environment in order to initialize and prepare the project.
Please enter the name of your project. This name will be used in page titles and descriptions.
Project Name [My Project]> Mon Projet
Please specify the absolute path to the project Subversion repository. Repository must be local, and trac's
Path to repository [/var/svn/test]> /home/svn/monprojet
Please enter location of Trac page templates. Default is the location of the site-wide templates installed
Templates directory [/usr/share/trac/templates]>
Creating and Initializing Project
[...]
```

Pour que trac fonctionne correctement, il faut que l'utilisateur utilisé par apache aient les droits de lecture et écriture :

```
# chown www-data: -R /home/trac/
```

Intégrer Trac avec Apache (cgi)

Editez ensuite le fichier de configuration d'apache (vous devez normalement déjà avoir la partie dédiée à Subversion) :

```
## TRAC #
Alias /trac "/usr/share/trac/htdocs"
ScriptAlias /monprojet /usr/share/trac/cgi-bin/trac.cgi
<Location "/monprojet">
    SetEnv TRAC_ENV "/home/trac/monprojet"
</Location>

<Directory "/usr/share/trac/htdocs">
    Options Indexes MultiViews
    AllowOverride All
    Order allow,deny
    Allow from all
</Directory>

# You need something like this to authenticate users #
<Location "/cgi-bin/trac.cgi/login">
    AuthType Basic
    AuthName "project"
    AuthUserFile /home/svn/monprojet.htpasswd
    Require valid-user
</Location>
```

Redémarrer apache pour que ces éléments soient pris en compte :

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

Intégrer Trac avec Apache (mod_python)

Installer mod_python :

```
apt-get install libapache2-mod-python
a2enmod mod_python
/etc/init.d/apache2 reload
```

Editez ensuite le fichier de configuration de votre VirtualHost :

```
<Location /trac>
    SetHandler mod_python
    PythonHandler trac.web.modpython_frontend
    PythonOption TracEnv /home/nicos/django-trac
    PythonOption TracUriRoot /
    PythonPath "sys.path + ['/home/nicos/django-trac']"
</Location>

    <Location /trac/login>
        AuthType Basic
        AuthName "Mon Projet"
        AuthUserFile /home/svn/monprojet.htpasswd
        Require valid-user
    </Location>
```

Que vous ayez installé Trac en CGI ou avec mod_python

Redémarrer apache pour que ces éléments soient pris en compte :

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

Rendez vous alors sur <http://localhost/monprojet/> [<http://localhost/monprojet/>] et vous devez avoir une magnifique interface Trac et dans "Browser source" vous devez voir votre projet.

Sécurité et politique d'accès

Gestion globale des utilisateurs (authentification)

Trac, par défaut, gère ses utilisateurs via l'authentification apache (couple de fichiers .htaccess/.htpasswd). Il est néanmoins possible d'utiliser une base MySQL, un annuaire LDAP pour authentifier les utilisateurs mais tel n'est pas mon besoin.

Pour l'accès à trac et subversion, j'ai défini dans la configuration des mes hôtes virtuels un lien vers

un fichier /home/svn/monprojet.htpasswd, dans lequel seront stockés un couple login / mot de passe.

Il faut créer ce fichier :

```
htpasswd2 -c /home/svn/monprojet.htpasswd monutilisateur
New password :
Re-type password for user monutilisateur :
```

Pour ajouter d'autres utilisateurs :

```
htpasswd2 /home/svn/monprojet.htpasswd utilisateur2
New password :
Re-type password for user utilisateur2 :
```

Relancez apache pour que vos modifications soient prises en compte :

```
/etc/init.d/apache2 reload
```

Rendez vous sur votre instance Trac et cliquez sur "login" et authentifiez vous...

Gestion des droits des utilisateurs dans Trac

Il faut savoir que, par défaut, un profil anonyme (anonymous) peut tout faire sur votre instance Trac. Vous n'avez certainement pas envie que cela soit possible.

Pour visualiser les permissions, il vous suffit de taper dans une console :

```
trac-admin /home/trac/monprojet permission list
```

Cela donne par ex :

```
nicolas@ange:~$ sudo trac-admin /home/trac/monprojet/ permission list
User Action -----
anonymous BROWSER VIEW
anonymous CHANGESET VIEW
anonymous FILE VIEW
anonymous LOG VIEW
anonymous MILESTONE VIEW
anonymous REPORT_SQL VIEW
anonymous REPORT VIEW
anonymous ROADMAP VIEW
anonymous SEARCH VIEW
anonymous TICKET_CREATE
anonymous TICKET_MODIFY
anonymous TICKET_VIEW
anonymous TIMELINE VIEW
anonymous WIKI_CREATE
anonymous WIKI_MODIFY
anonymous WIKI_VIEW
Available actions: BROWSER_VIEW, CHANGESET_VIEW, CONFIG_VIEW, FILE_VIEW, LOG_VIEW, MILESTONE_ADMIN, MILESTONE_CREATE, MILESTONE_DELETE, MILESTONE_EDIT, MILESTONE_VIEW, REPORT_CREATE, REPORT_DELETE, REPORT_EDIT, REPORT_VIEW, ROADMAP_CREATE, ROADMAP_DELETE, ROADMAP_EDIT, ROADMAP_VIEW, SEARCH_CREATE, SEARCH_DELETE, SEARCH_EDIT, SEARCH_VIEW, TICKET_CREATE, TICKET_DELETE, TICKET_EDIT, TICKET_MODIFY, TICKET_VIEW, TIMELINE_CREATE, TIMELINE_DELETE, TIMELINE_EDIT, TIMELINE_VIEW, WIKI_CREATE, WIKI_DELETE, WIKI_EDIT, WIKI_MODIFY, WIKI_VIEW
nicolas@ange:~$
```

On voit ici qu'un anonyme peut créer/modifier une page de wiki et un ticket, or ce n'est pas ce que je souhaite.

Je souhaite que :

- L'internaute anonyme n'ait qu'un accès en lecture
- Les développeurs aient un accès en lecture/écriture.

Supprimons déjà les droits de modification/création pour les anonymes :

```
trac-admin /home/trac/monprojet permission remove anonymous WIKI_CREATE WIKI_MODIFY TICKET_CREATE TICKET_MODIFY
```

Ajoutons l'utilisateur que j'ai créé plus tôt dans un groupe "developpeur" que je crée par la même occasion :

```
trac-admin /home permission add monutilisateur developpeur
```

Donnons ensuite des permissions au groupe developpeur pour la création/modification de Tickets et des pages du wiki :

```
trac-admin /chemin/de/mon/instance/trac permission add developpeur WIKI_CREATE WIKI_MODIFY TICKET_CREATE
```

Note : on pourrait donner des permissions par utilisateur de la même manière mais je préfère gérer des groupes, c'est plus évolutif...

Il ne vous reste plus qu'à affiner les droits en fonction de vos souhaits...

Notes :

- Les droits en *_ADMIN (ex : WIKI_ADMIN), donne tous les droits inférieurs. Donc WIKI_ADMIN = WIKI_VIEW + WIKI_CREATE + WIKI_MODIFY + WIKI_DELETE
- Le droit TRAC_ADMIN donne tous les droits sur l'ensemble de l'application ; à manier avec précaution !

Gestion des droits des utilisateurs dans Subversion

Editez le fichier /home/svn/monprojet/conf/svnserve.conf pour avoir les options suivantes (j'ai enlevé les tonnes de commentaires pour un gain évident de place) :

```
[general]
anon-access = read
auth-access = write
password-db = passwd
realm = monprojet realm
```

Editez ensuite le fichier /home/svn/monprojet/conf/passwd (passwd = nom du fichier définit à l'azimut de la ligne password-db) et ajoutez-y votre utilisateur

```
### This file is an example password file for svnserve.
### Its format is similar to that of svnserve.conf. As shown in the
### example below it contains one section labelled [users].
### The name and password for each user follow, one account per line.
[users]
monutilisateur = monmotdepasse
```

Tester votre serveur

A ce stade, en anonyme, vous devez pouvoir mener un checkout des 2 façons suivantes :

- `svn co http://localhost/svn/trunk [http://localhost/svn/trunk] monprojet`
- `svn co file:///home/svn/monprojet/svn/trunk monprojet`

Par contre, pour commiter, il vous faut vous identifier depuis votre client SVN (Tortoise, plugin Subclipse pour Eclipse, etc) et dans ce cas, je vous renvoie à la documentation de votre logiciel... ☺

En savoir plus...

- Book Subversion [<http://svnbook.red-bean.com/>]
- Gérer un ou plusieurs projets dans SVN [<http://svnbook.red-bean.com/en/1.0/ch05s04.html>]
- Check-out initial [<http://svnbook.red-bean.com/en/1.0/ch03s04.html>]
- Trac on Debian [<http://projects.edgewall.com/trac/wiki/TracOnDebian>]
- Trac Guide [<http://projects.edgewall.com/trac/wiki/TracGuide>]
- Trac:Install [<http://projects.edgewall.com/trac/wiki/TracInstall>]
- Trac:Avoir différents projets sur une seule machine [<http://projects.edgewall.com/trac/wiki/TracMultipleProjects>]
- Trac : mod_python [<http://trac.edgewall.org/wiki/TracModPython>]
- Trac : gestion des utilisateurs [<http://projects.edgewall.com/trac/wiki/TracCgi>] (en bas de page)
- Trac : Permissions [<http://projects.edgewall.com/trac/wiki/TracPermissions>]
- SVN Book : Authentification & Authorisation [<http://svnbook.red-bean.com/en/1.1/ch06s03.html#svn-ch-6-sect-3.2>]

Discussion

ytek @ enimatek, [2007/05/28 16:03](#):

Merci pour ce tutorial, Il serait intéressant de préciser que le projet (import initial) doit être organisé comme la destination (fichiers sources du projet dans un /trunk). Une autre question. Je n'ais rien sous <http://localhost/svn> [<http://localhost/svn>] après l'étape "paramétrage de subversion dans apache" en revanche sous <http://localhost/monprojet/trunk> [<http://localhost/monprojet/trunk>], il y a bien les fichiers importés. Est-ce une erreur du tuto ou de moi ? (rien trouvé de mon côté)