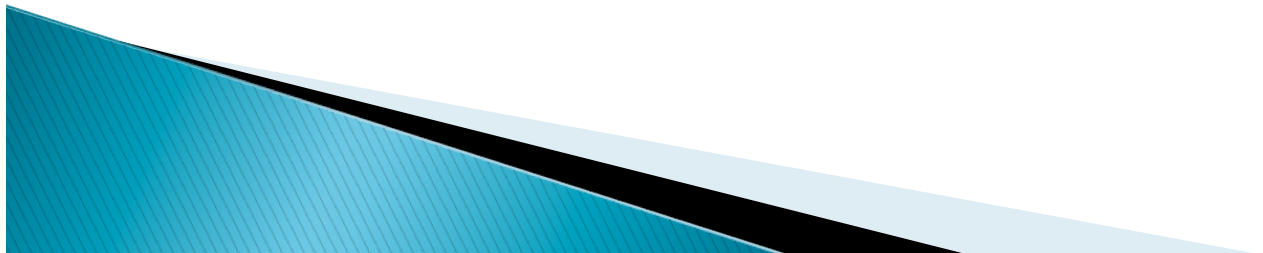
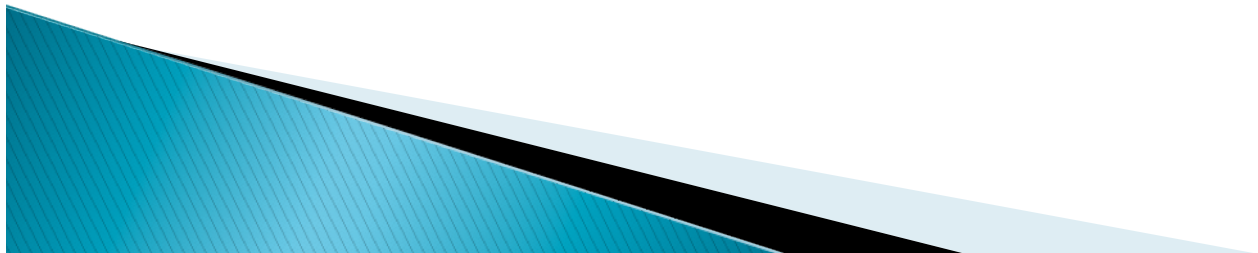
Subversion (SVN)

Système de gestion de version
Successeur de CVS



Sommaire

- Qu'entend-on par gestion de version ?
- Introduction à SVN
- Principes de base
- Différences avec CVS
- Commandes
- Exemples
- Configuration et accès à un dépôt



Qu'entend-on par gestion de version ?

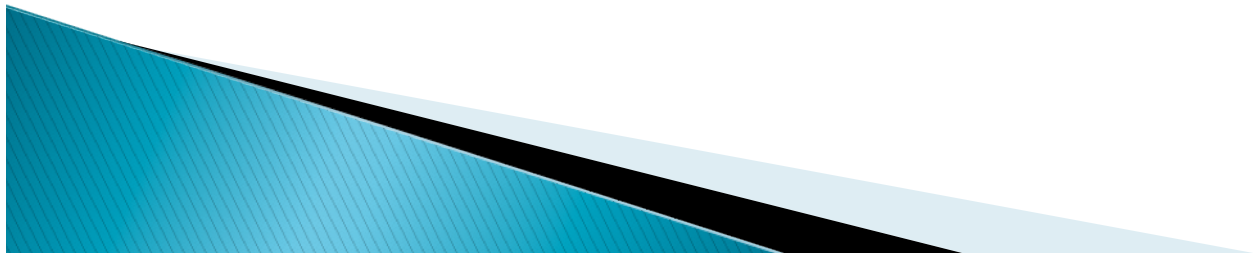
Trois principes de base :

- Garder la trace et l'historique des changements
- Accorder un accès public à l'information
- Conserver différentes versions d'un même ensemble de données

Quels types de données ?

Code source,

- Documentation
- Fichiers de configuration
- Plus généralement, tout type de données

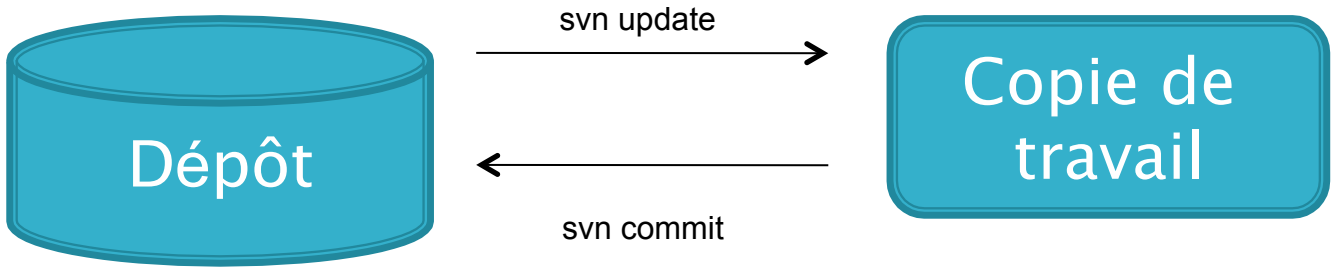


Terminologie

- Dépôt
  - Copie centralisée de tous les fichiers suivis, structurée sous forme d'arborescence de répertoires .
- Copie de travail
  - Copie locale des données pouvant être modifiée et synchronisée avec le dépôt. Contient des informations spéciales qui permettent la synchronisation.
- Version
  - Groupe de répertoires et de fichiers reflétant l'état du dépôt à un moment déterminé.

Principes de base

- Le dépôt est la copie maîtresse
- Tout le travail est effectué dans une copie de travail
- Les changements sont validés et apparaissent dans le dépôt (avec la commande *commit*)

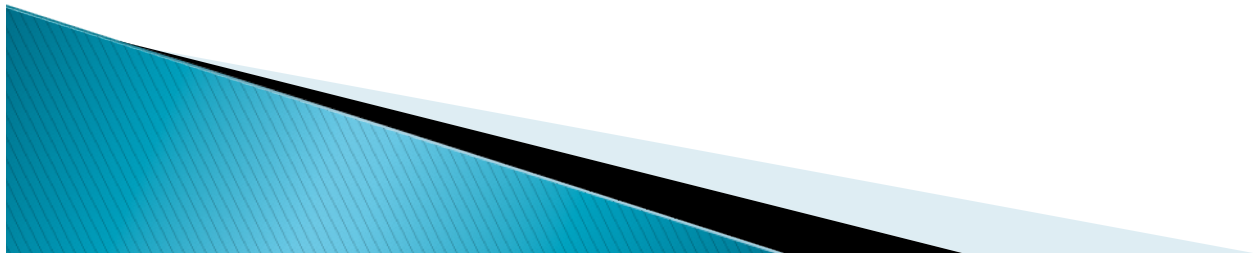


Gestion des modifications : états

- Aucune modification et à jour
  - La copie est identique au dépôt
  - Commande *commit* ou *update* inopérante
- Modifications locales et à jour
  - La copie locale a été modifiée et le dépôt n'a pas reçu les modifications.
  - Un *commit* met à jour le dépôt. *update* n'a aucun effet.
- Aucune modification et pas à jour
  - La copie locale n'a pas été modifiée, mais le dépôt l'a été.
  - *update* modifie l'état local, *commit* est sans effet.
- Modifications locales et pas à jour
  - Conflit ! Nécessité d'exécuter un *update*
  - Si SVN ne peut pas résoudre le conflit automatiquement, alors une résolution manuelle sera nécessaire.

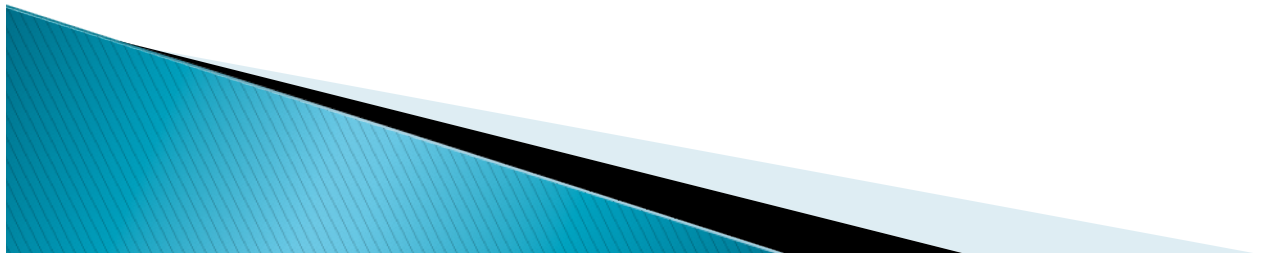
Exemple de session

- Contrôle initial du dépôt
  - svn checkout <*projet*>
  - vi <*myfile.conf*> (...modifications ...)
  - svn commit <*myfile.conf*> (*répercute les modifications*)
- Autres modifications :
  - svn update
  - vi <*myfile.conf*>
  - svn commit <*myfile.conf*>



SVN et le dépôt

- Les clients peuvent accéder en local ou via le réseau
- Variable d'environnement SVNROOT :
- SVNROOT=
  - `/svn/myproject` # disque local
  - `svn://svnserver/svn/myproject` # via svnserve
  - svn+ssh:// `svnserver/svn/myproject` # via SSH

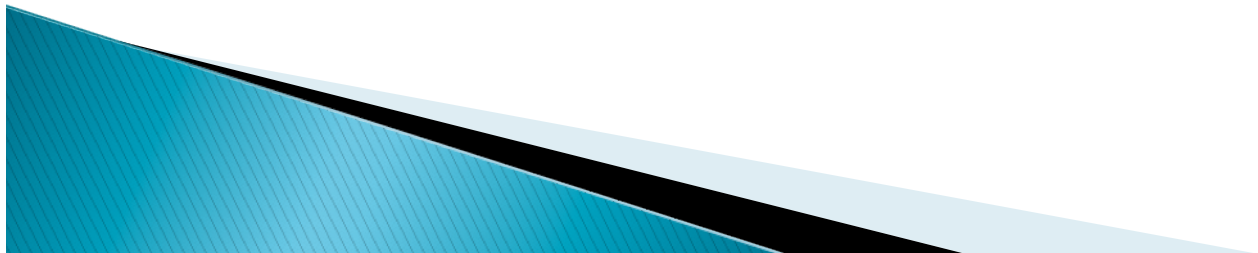


Création d'un dépôt

- Installation
  - #apt-get install subversion
  - #svncreate <dépôt>
  - Modifiez <dépôt>/
- Création en tant que "service"
  - Créez /etc/init.d/subversion, qui comprend essentiellement
    - svnserve -d -r <dépôt>
  - #chkconfig --add subversion
  - #chkconfig -level 2345 subversion on
- Modification des autorisations
  - Modifiez <dépôt>/conf/svnserve.conf
  - Spécifiez le fichier de mots de passe :
    - [général]
    - password-db = <fichierutilisateur>
    - realm = exemple realm
  - Créez des utilisateurs :
    - [utilisateurs]
    - pedro = foopassword
    - sandra = barpassword

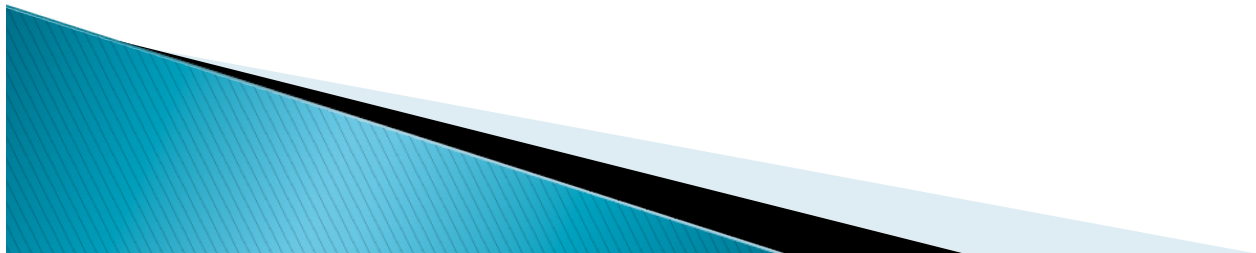
SVN – clients

- Il existe des clients pour la plupart des systèmes d'exploitation :
  - svn (UNIX)
  - TortoiseSVN (Windows)
  - ...
- Accès local ou via le réseau



Commandes SVN

- import
  - Importe un nouveau projet depuis un dépôt
- checkout (co)
  - Copie le dépôt dans un répertoire local
- update (up)
  - Met à jour la copie locale à partir du dépôt
- add
  - Crée un nouveau fichier ou répertoire dans la copie locale
- delete
  - Supprime un fichier de la copie locale
- commit
  - Met à jour le dépôt à partir de la copie locale

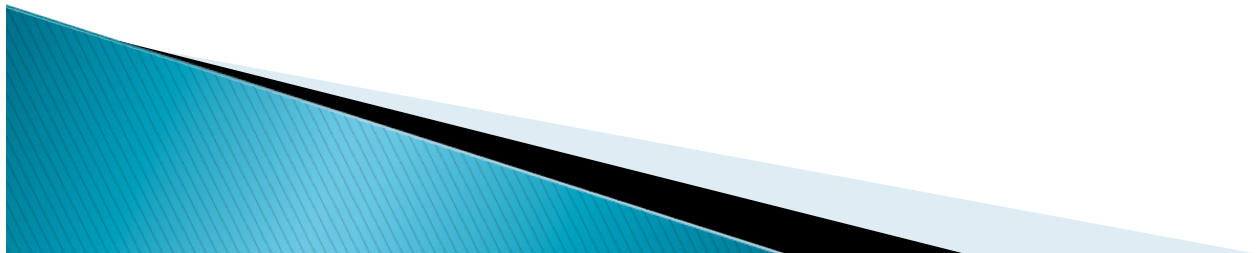


Autres commandes utiles

- mkdir
  - Crée un répertoire dans la copie locale.
- status
  - Affiche la version et l'état du fichier.

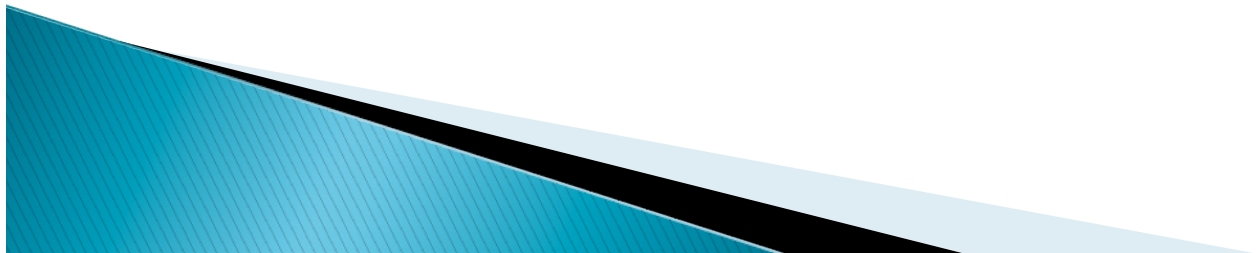
- diff
  - Indique les différences entre un élément local (fichier, répertoire) et cet élément dans le dépôt.
- log
  - Affiche l'historique des modifications d'un ou de plusieurs fichiers.

- Nombreuses autres commandes : copy, export…



Cycle de travail

- Mise à jour de la copie de travail
  - svn update
- Modification de votre copie locale
  - svn add
  - svn delete
  - svn copy
  - svn move
- Contrôle de vos modifications
  - svn status
  - svn diff
  - svn revert
- Combinaison de vos modifications avec celles d'autres personnes
  - svn merge
  - svn resolve
- Finalisation de vos modifications et intégration dans le dépôt
  - svn commit



Avantages & différences par rapport à CVS

- CVS gère uniquement les modifications apportées aux fichiers.
- SVN crée un système de fichiers virtuel qui comprend des répertoires.

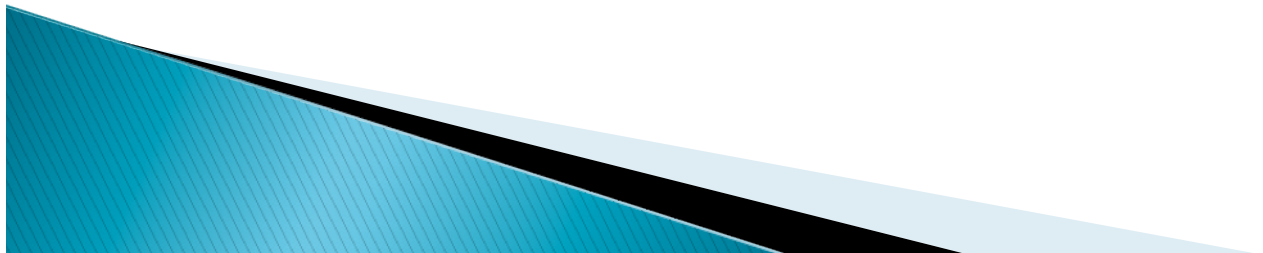
- CVS ne gère pas les changements de nom, ni les copies de fichiers.
- Le mode de gestion des répertoires par SVN autorise les changements de nom et les copies de fichiers.

- SVN permet une gestion "atomique" des modifications : acceptation de toutes les modifications ou d'aucune modification.
- CVS n'offre pas de fonctionnalité similaire.

- De façon générale SVN offre une plus grande souplesse d'accès, par exemple HTTP via Apache et les avantages qui en découlent.

Conclusions

- Un système de gestion de version sophistiqué
- Très utile pour les programmeurs
- Bon nombre de fonctions de haut niveau ne sont pas utilisées par les administrateurs de réseau
- En réalité, CVS et Subversion peuvent tous deux être utilisés pour faciliter l'administration de réseau.
- On ne peut cependant ignorer que :
  - L'outil le plus populaire est l'outil qui reçoit le plus de soutien
  - Beaucoup d'entre nous apportent un soutien aux équipes de programmation dans le cadre de notre travail quotidien



Références

- "Version Control with Subversion" – O'Reilly
- Téléchargeable gratuitement sur le site http://svnbook.red-bean.com