I am using both SVN (Apache Subversion) and Git in my daily work, so I will cast a brick to attract jade for more valuable discussion.

------------------------------------------------This is a cut line----------------------------------------------------------

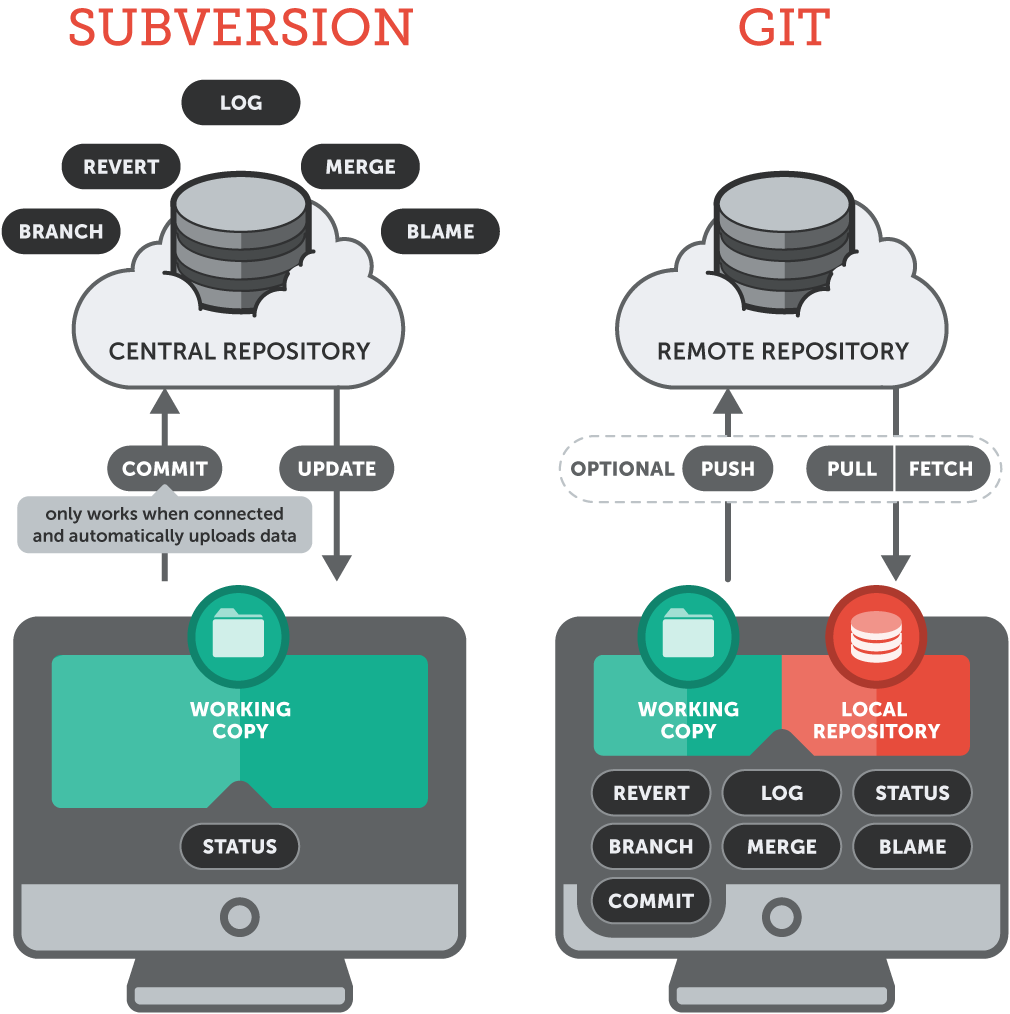
1. Main difference between SVN and Git

SVN:

* SVN is a centralized version control system.
* Must have a connection and the server must be up to commit changes
* SVN stores information as a list of file-based changes, also known as delta-based control.
* Liner commit revision. Revision number is, for instance, r886886.The revision increases sequentially.
* Single branch checkout

Git:

* Git is a distributed version control system
* Nearly all operations are local
* Git thinks of its data more like a series of snapshots of a miniature file system
* Revision number is, for instance, d0e4b10d5e994836152e5095f25c6306ec0f59c3. It is also commonly called as SHA and is essentially an output of hash function and appears to be random between commits.
* Full repository clone available locally



1. Advantages of Git over SVN

As mentioned, Git is a distributed, whereas SVN is a centralized system

* Developers can work offline with Git by committing any local changes to a local repository and save logs for each commit (version stored).

SVN developers will be stranded if they are not connected to the network, as they cannot commit their local code changes to repository and mark down as one log. They will have no choice but to wait for network resume.

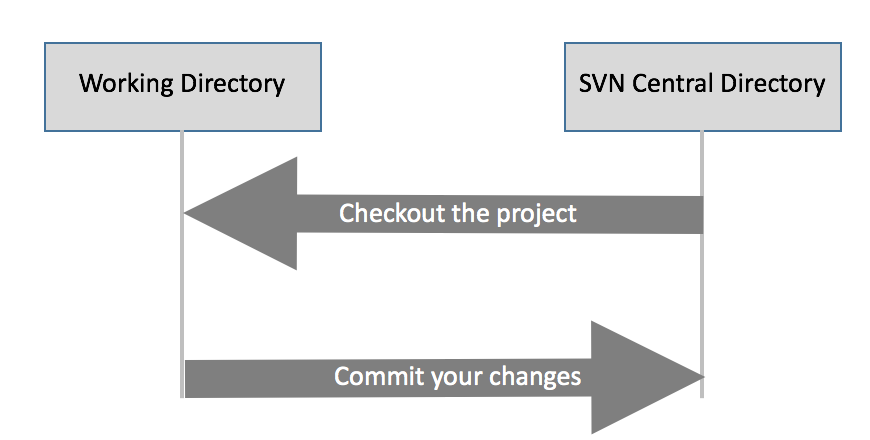
* The full history of Git is available to developers even when they are offline.
* With Git I can destroy the Remote Repository and I can be fully confident that it is resumable because every clone from developers on their local machine is a usable backup.

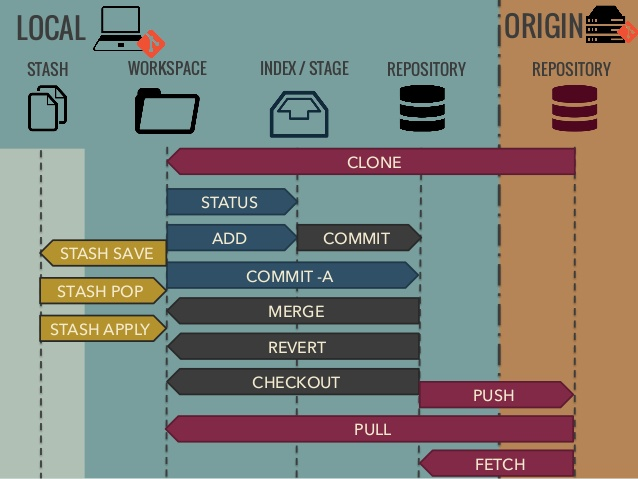
It is disastrous when some developer accidentally permanently deleted the Central Repository

* Git repositories are smaller than SVN.
* Git is faster and it enables advances workflows via features such as branching, forks and pull requests.
* Git branch system is lightweight and it can be quickly created/switched between branches by simply using command, git branch <branch1>, git checkout <branch1>. Whereas, SVN has to create a remote branch, then goes to your main local folder, and checkout to your folder.

1. Advantages of SVN over Git

* Simplicity in practice and also conceptually easier to for developers to pick up. SVN is a 2 stage system, as the below picture shows, which only involves working copy and central repository. A commit is indeed a commit. Unlike that Git is a 4 stage system, where it introduces the concept of local repository. In order to completely commit to the remote repository, you must git commit first, followed by git push.





* For logs numbering, SVN has a user-friendly and human-readable commit numbering system, only using digits.

Git d0e4b10d5e994836152e5095f25c6306ec0f59c3 vs SVN r886886

Manifestly, the latter is more meaningful to humans and simpler to track or remember.

1. Common commands for SVN and Git

* To checkout the remote repository

svn checkout <url>

git clone <url>

* To add file to remote repository

svn add <file>, svn commit -m “<msg>”

git add <file>, git commit -m “<msg>” , git push

* To remove a file

svn rm <file>, svn commit -m “<msg>”

git rm <file>, git commit -m “<msg>” , git push

* To get the latest update from remote repository to local repository

svn is not applicable

git fetch

* To get he latest update from remote repository to local working copy/workspace

svn up

git pull, equivalent to git fetch, followed by git merge

* To show logs

svn log

git log

* To show diff

svn diff

git diff

* To revert a local file

svn revert <file>

git checkout -- <file> //revert changes at workspace

git reset HEAD <file> //revert changes at staging area

git checkout <SHA> <file>, git revert <SHA> //revert a file at local repository

* To create a new branch from master

svn cp <master url> <branch url>, svn checkout <branch url>, cd <local branch>

git branch <branch>, git checkout <branch>