

Git

Git is a **distributed version-control system** for tracking changes in source code during software development. It is designed for coordinating work among programmers, but it can be used to track changes in any set of files. Its goals include speed, data integrity, and support for distributed, non-linear workflows.



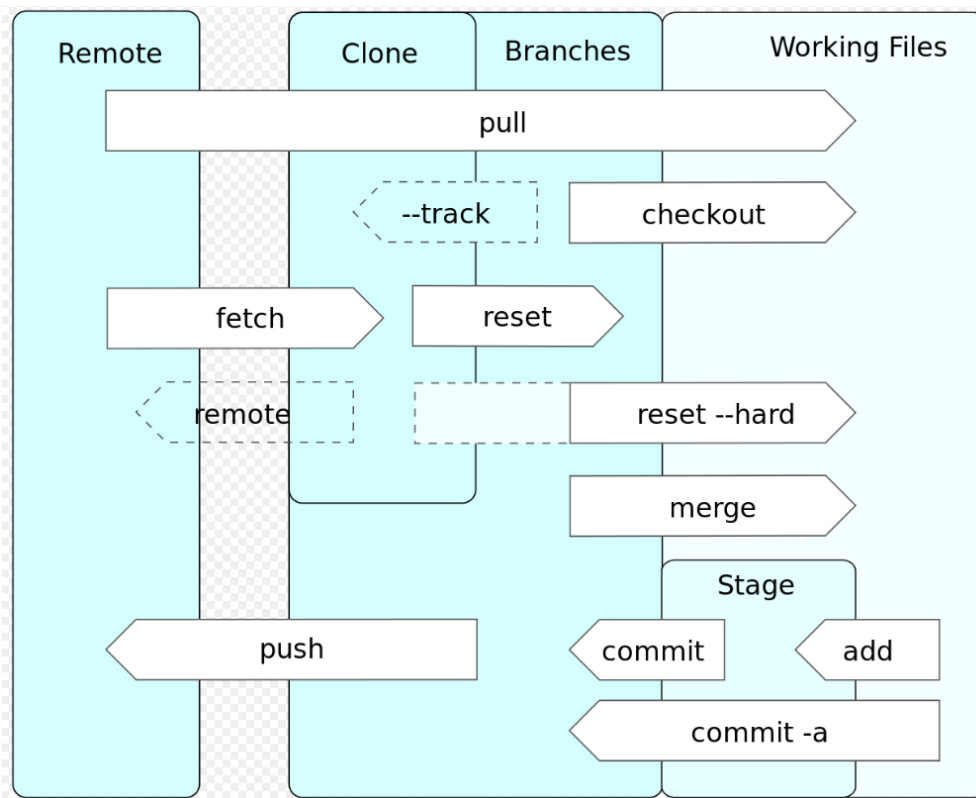
If you are familiar with SVN(Subversion) or CVS(Concurrent Version System), you may have no difficulty in using Git.

The characteristics are as followings:

- Strong support for non-linear development
- Distributed development
- Compatibility with existent systems and protocols such as HTTP, FTP, SSH and etc.
- Efficient handling of large projects
- Cryptographic authentication of history
- Toolkit-based design
- Plug-gable merge strategies
- Garbage accumulates until collected
- Periodic explicit object packing

Data flows

- Git has a data revision control system in storage level with its version information. And you will be faced an unfamiliar words like fork, clone and branch.
 - A **fork** is a copy of a repository that allows you to freely experiment with changes without affecting the original project. You can contribute back to the original project using **pull requests**. Forking is a first step for collaborate with your colleagues.
 - A **clone** is a copy of all the code on the master branch - it is an exact replica of the code on git server. Unlike forking, you won't be able to pull down changes from the original repository you cloned from, and if the project is owned by someone else you won't be able to contribute back to it unless you are specifically invited as a collaborator. Cloning is ideal to quickly get your own copy of a repository where you may not be contributing to the original project.
 - A **branch** is a slightly changed or modified section of code that meets different objectives.
- **git clone** command copies an existing Git repository - it is primarily used to point to an existing repo and make a clone or copy of that repo at in a new directory, at another location. The original repository can be located on the local file system or on remote machine accessible supported protocols.
- **git commit** goes to local branch.
- **git push** goes to remote master.



Followings are useful content can understand about the basic concept of Git at YouTube

Other information

- [Block accessing .git directory in the Apache Server](#) — You should block accessing your precious source code when you use Apache web server and manage your code based in Git - meaning Hacker can easily stole your precious information from your source code directly when you have .git on your repository directory.
- [Clone a git repository to another directory](#) — To clone to another directory at git, just add the target directory in the clone command.
- [Setup Git Client with TortoiseGit on Windows 10](#) — TortoiseGit is a kind of family project with TortoiseSVN and TortoiseCVN, so if you are familiar with that, you will be easy to use it.
- [Setup Git server on CentOS 7](#) — Git is a part of CentOS 7 repository, so you can install it easily by yum.
- [Setup Git server on CentOS 8](#) — Just like CentOS 7, CentOS 8 provides git as a part of repository. You can simply install it by yum - it is super straightforward and easy in terms of server's point of view. It requires just creating an account for git and running couple of commands.
- [Useful git command scripts in bash](#) — git provides a straight forward and strong commands, but sometimes scripts will help you to make things simpler.