Notes

Generally Do

* Create things
* Save things
* Edit things
* Save the thing again

Benefits of Git

* Git is a fast and modern implementation of version control
* Git provides a history of content changes.
* Git facilitates collaborative changes of file
* Git is easy to use for any type of knowledge worker.

Local Git

* Distributed so that connectivity doesn’t block work
* Easy so that learning its commands can happen progressively

Distributed Git

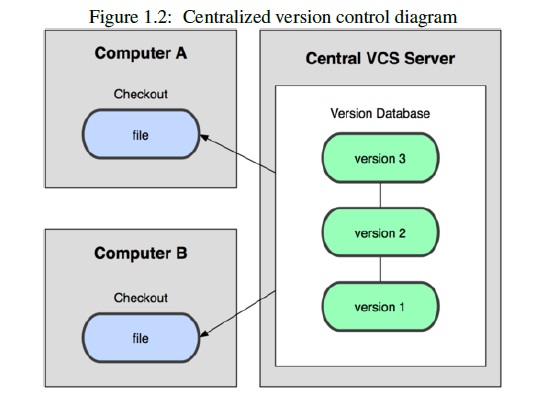
* Team-centric so that collaboration happens naturally

Version Control

* Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later.

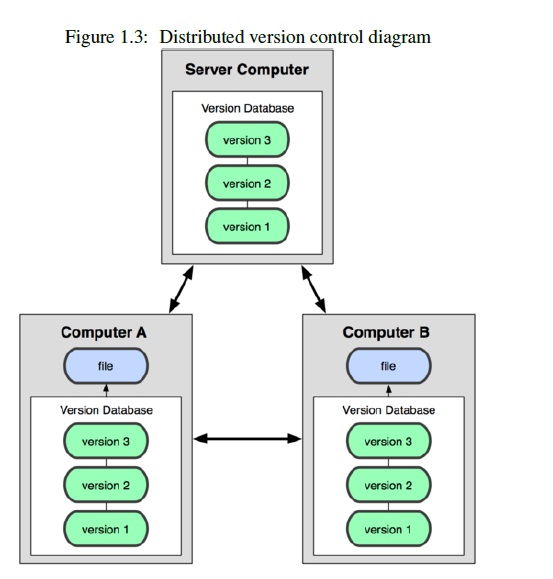
Centralized Version Control Systems

* CVS, Subversion, and Perforce, have a single server that contains all the versioned files, and a number of clients that check out files from that central place.
* Example



Distributed Version Control Systems

* Clients don’t just check out the latest snapshot of the files: they fully mirror the repository.
* Thus if any server dies, and these systems were collaborating via it, any of the client repositories can be copied back up to the server to restore it.
* Every checkout is really a full backup of all the data
* Example

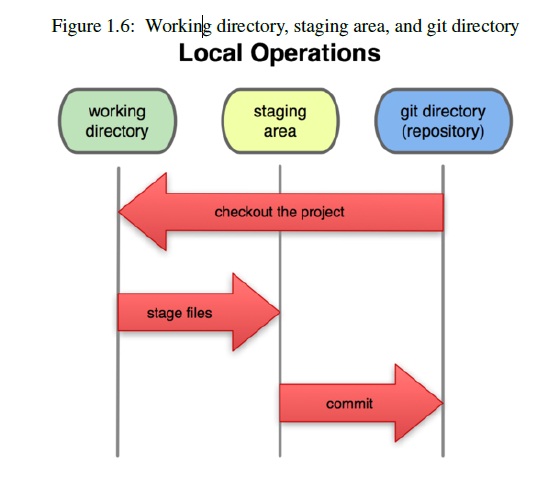


Advantages of Git

* Speed
* Simple design
* Strong support for non-linear development (thousands of parallel branches)
* Fully distributed
* Able to handle large projects like the Linux kernel efficiently (speed and data size)

The Three States of Git

* Committed, modified, and staged.
* Committed means that the data is safely stored in your local database.
* Modified means that you have changed the file but have not committed it to your database yet.
* Staged means that you have marked a modified file in its current version to go into your next commit snapshot.
* This leads us to the three main sections of a Git project: the Git directory, the working directory, and the staging area.
* Example



Git Directory

* The Git directory is where Git stores the metadata and object database for your project. This is the most important part of Git, and it is what is copied when you clone a repository from another computer.
* The working directory is a single checkout of one version of the project. These files are pulled out of the compressed database in the Git directory and placed on disk for you to use or modify.
* The staging area is a simple file, generally contained in your Git directory that stores information about what will go into your next commit. It’s sometimes referred to as the index, but it’s becoming standard to refer to it as the staging area.

The basic Git workflow goes something like this:

* You modify files in your working directory.
* You stage the files, adding snapshots of them to your staging area.
* You do a commit, which takes the files as they are in the staging area and stores that snapshot permanently to your Git directory.
* If a particular version of a file is in the git directory, it’s considered committed. If it’s modified but has been added to the staging area, it is staged. And if it was changed since it was checked out but has not been staged, it is modified