An Introduction to Git Talk

Ehsan Zandi

Deutsche Telekom IT

ehsan.zandi@telekom.de

23.08.2021

Overview

Git vs SVN

Git Basics

Git vs SVN

- Git is a fully distributed version control system (VCS)
- Each user (PC/Laptop) is an exact clone of the remote repository
  - Each user is a repository (log, revert, merge, branch, etc)
  - No network connection required, except to sync with central repo (pull/push/fetch)
  - merge and rebasing can be done offline
- Git is much faster than SVN
- Git's repositories are much smaller than SVN
- Git's branches are much simpler and less resource heavy than SVN
- Git is much better in branch auditing and merge handling
- As many backups as the number of users ()
- Content integrity using SHA-1 hash

Git vs SVN

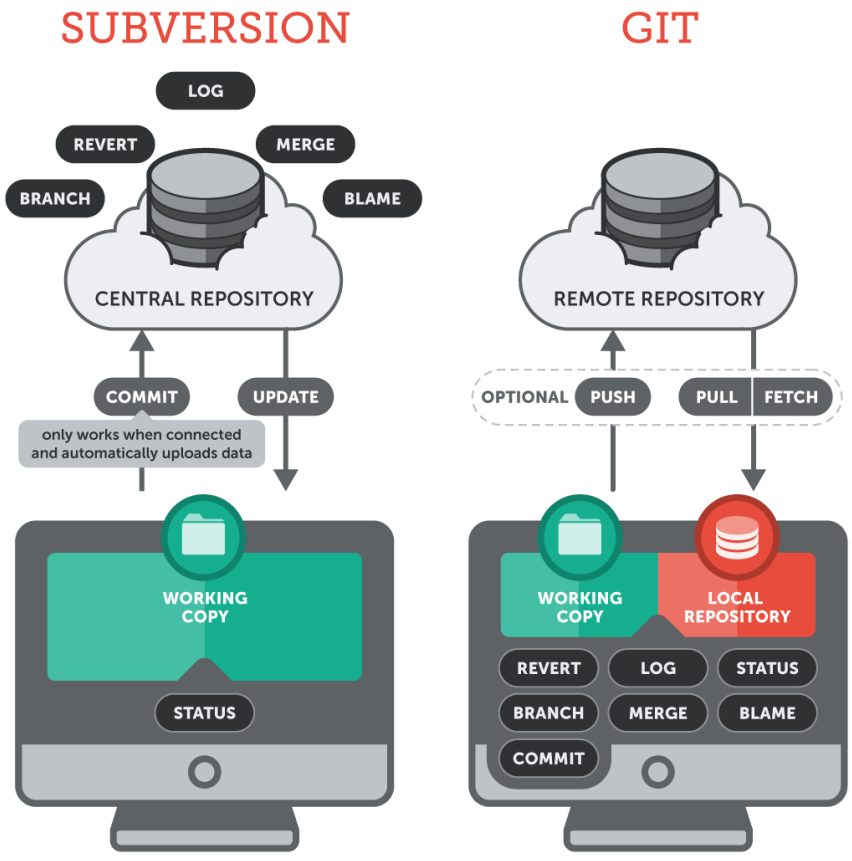

Figure : Centralized vs distributed VCS (Source: www.git-tower.com)

Git vs SVN

	SVN	Git
License	Open-source (Apache)	GNU
Distributed-ness	Centralized	Fully Distributed
Speed	✗	✓
Storage	✗	✓
Integrity Guarantee	✗	✓
Brnaching & merging	✗	✓
Stashing	✗	✓

Git Basics

Architecture

- Remote: The central repo (on a host machine/server, e.g., Github or Gitlab) → is identified by the alias "origin"
- Repository: The local repo (.git sub-directory inside your working directory), created by "git init" or "git clone", i.e., ceartion/clonining
- Index or staging area: State between the working directory and repository (after modifying and before commiting)
- Workspace or working directory: your local machine, including all directories, sub-directories, and files of your project

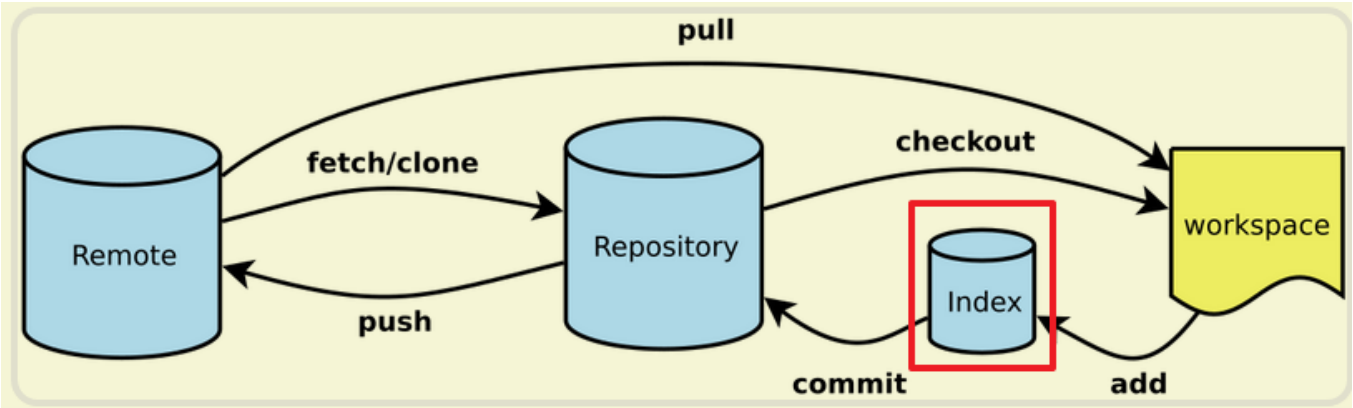

Figure : Git architecture (Source: www.stackoverflow.com)

Git Basics

Definitions

- **origin**: A shorthand name for the remote repo

  $git remote show (shows "origin" as output)

  $git remote show origin (shows detailed info on origin)
- **branch**: A movable pointer to a commit
- **master (or sometimes main)**: Default name of the (first) branch: can be changed
- **HEAD**: A special pointer that tells on (the tip of) which branch you are.
- **origin/HEAD**: A special pointer that tells on which branch the remote repo is.

Git Basics

Initializing a repo

- Creating a local repo (without any remote)

  $git init (creates .git sub-directory)

  $echo "hello world." >> firstFile.txt (makes changes in working area)

  $git status (You see that your commit has some hash value)

  $git add firstFile.txt (puts your changes into staging area)

  $git status (You see that your commit has some hash value)

  $git commit -m "A proper message" (Now you have your first commit)

  $git status (A clean repo and one commit with a hash value)

Git Basics

Add/Commit

- **git add**: To add a new file or modified into the staging (index) area. It makes the changes ready for commiting.

  $git add FILE_NAME

  $git add . (adds all the changes current directory and sub-directories)

- **git commit**: To put the staged files into the (local) repo. Such changes can be tracked, i.e., revert, log, etc.

  $git commit -m "A proper message"

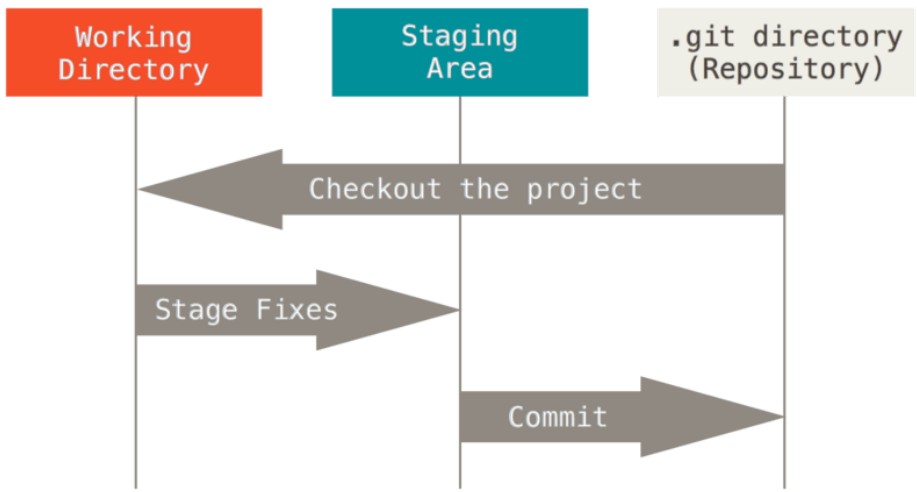

Figure : Git areas (Source: https://git-scm.com)