

Far Horizons Lab Training: GitHub Tutorial

Lou Nigra

Get the App

- Go to github.com
- If you have an account...
 - Browser might log you in automatically
 - Sign out to get welcome page.
- Scroll down to find GitHub app download link...
 - Windows: It's an icon near the bottom. (Hmmm....)
- Download and start installing the app.

Sign Up

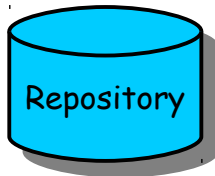
- While app is installing...
 - Browse back to welcome page if necessary (Windows)
- If you have an account, Sign in.
- If not, Sign up.
 - Username: Will tell you as you type if it's unique,
 - Email: Default is to keep private, so no worries... yet
 - Password:

Start the App

- Log in
-

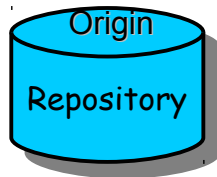
GitHub

- **Repository:** Compressed history of project “snapshots” (versions).
 - Origin (reference): Normally on GitHub
 - Local: Your working version, starts as **clone** of origin
- **Working Tree:** Visible files, sub-folders with current edits.
 - Extracted from **repository**.



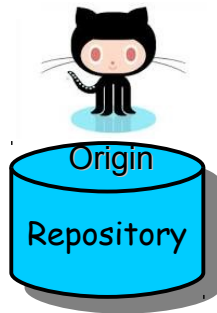
GitHub

- **Repository:** Compressed history of project “snapshots” (versions).
 - Origin (reference): Normally on GitHub
 - Local: Your working version, starts as **clone** of origin
- **Working Tree:** Visible files, sub-folders with current edits.
 - Extracted from **repository**.



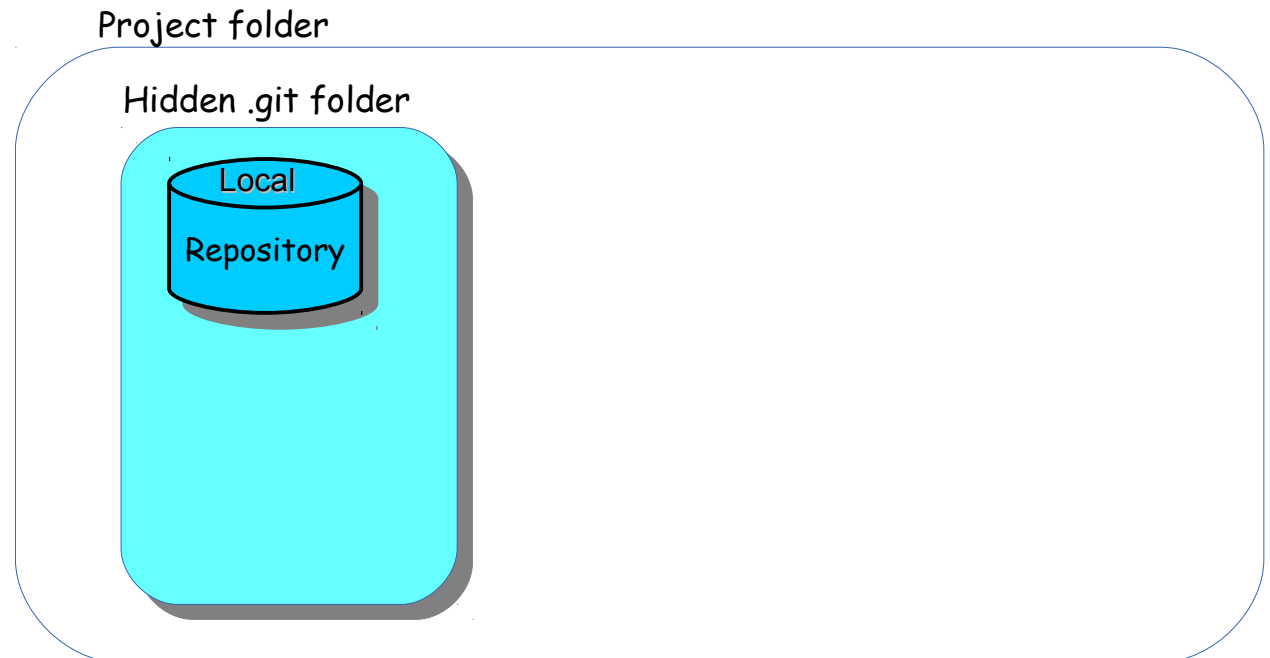
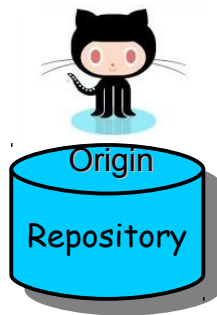
GitHub

- **Repository**: Compressed history of project “snapshots” (versions).
 - Origin (reference): Normally on GitHub
 - Local: Your working version, starts as **clone** of origin
- **Working Tree**: Visible files, sub-folders with current edits.
 - Extracted from **repository**.



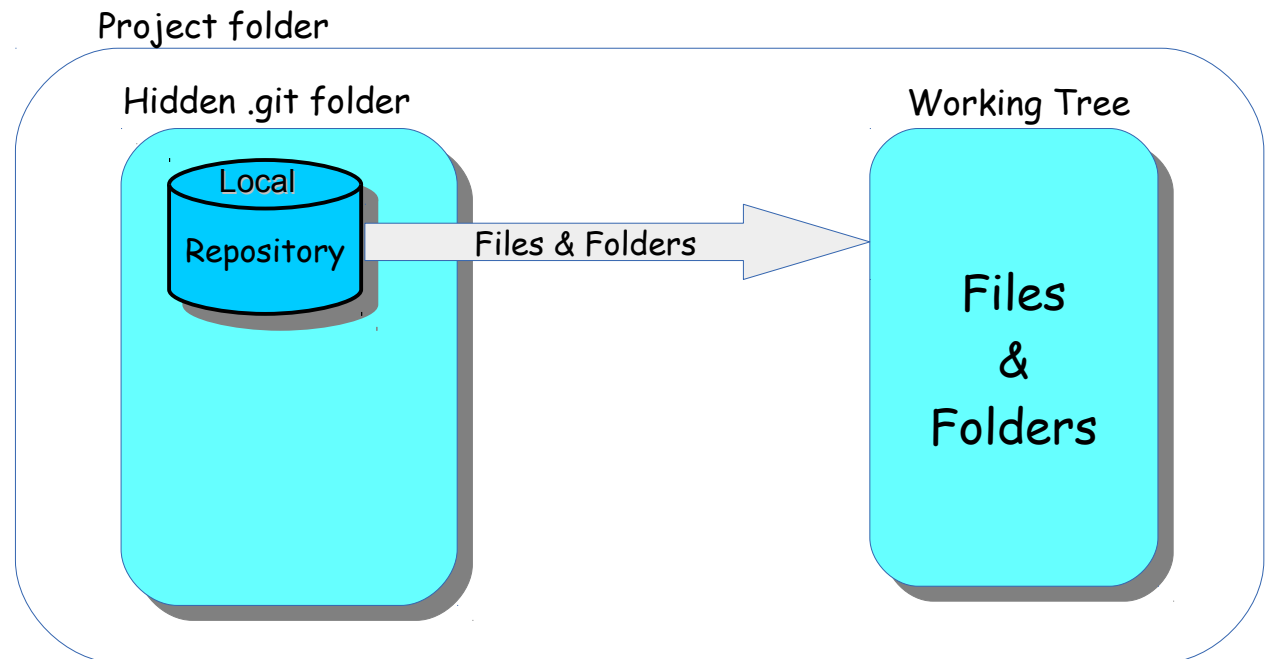
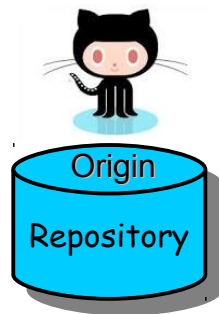
GitHub

- **Repository:** Compressed history of project “snapshots” (versions).
 - Origin (reference): Normally on GitHub
 - Local: Your working version, starts as **clone** of origin
- **Working Tree:** Visible files, sub-folders with current edits.
 - Extracted from **repository**.



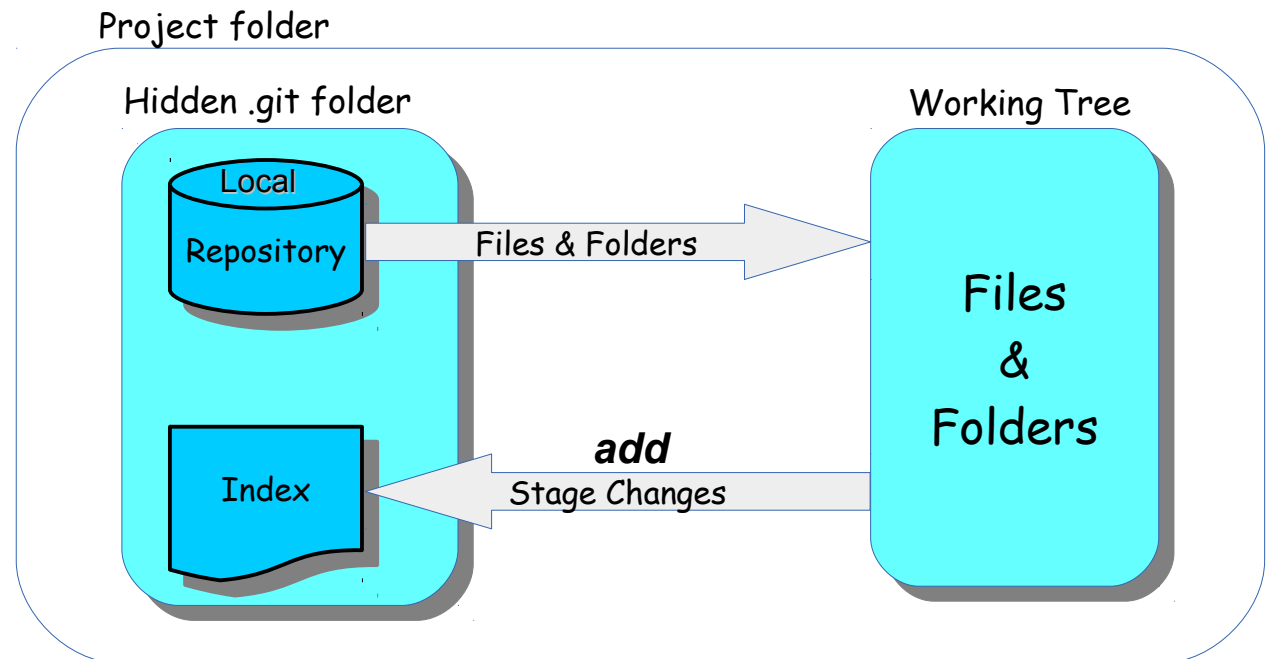
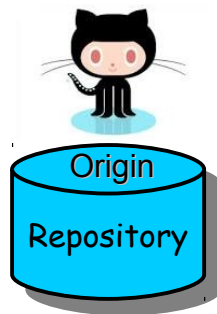
GitHub

- **Repository:** Compressed history of project “snapshots” (versions).
 - Origin (reference): Normally on GitHub
 - Local: Your working version, starts as **clone** of origin
- **Working Tree:** Visible files, sub-folders with current edits.
 - Extracted from **repository**.



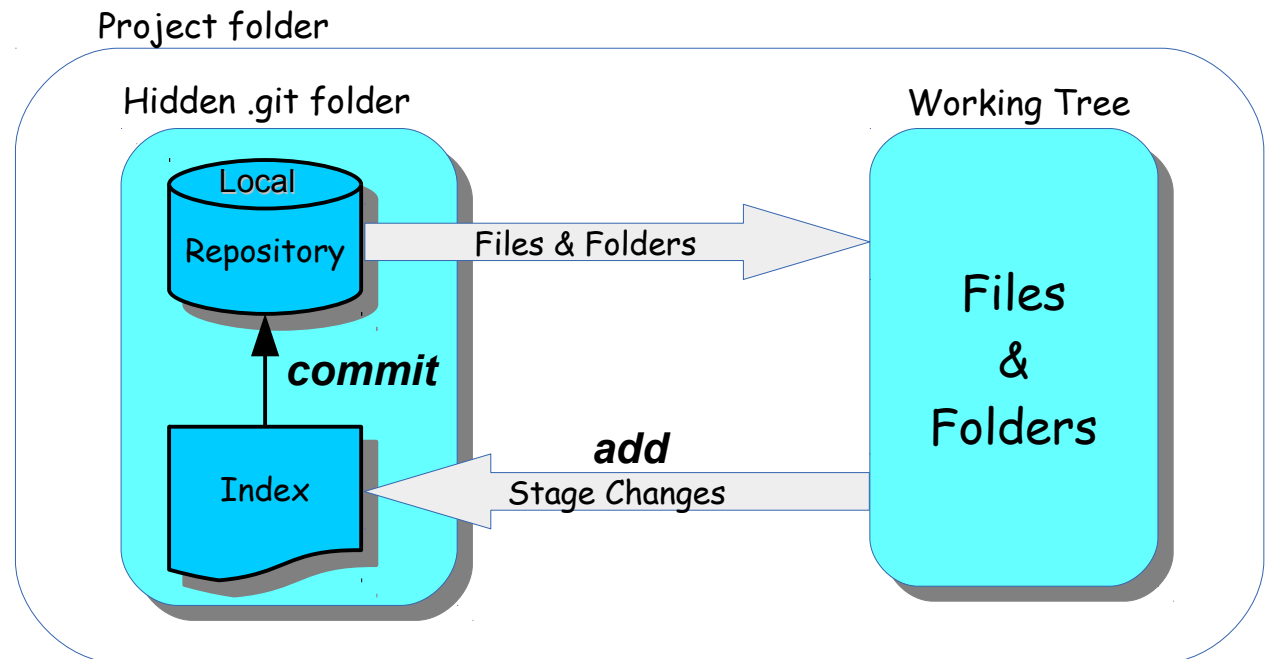
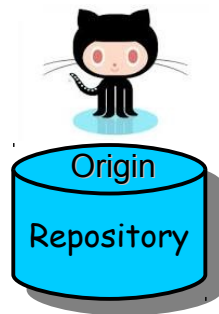
GitHub

- **Index:** Staging area for changes.
- **Commit:** A set of changes defining a version.
- **Checkout:** Restore files/folders from a commit.



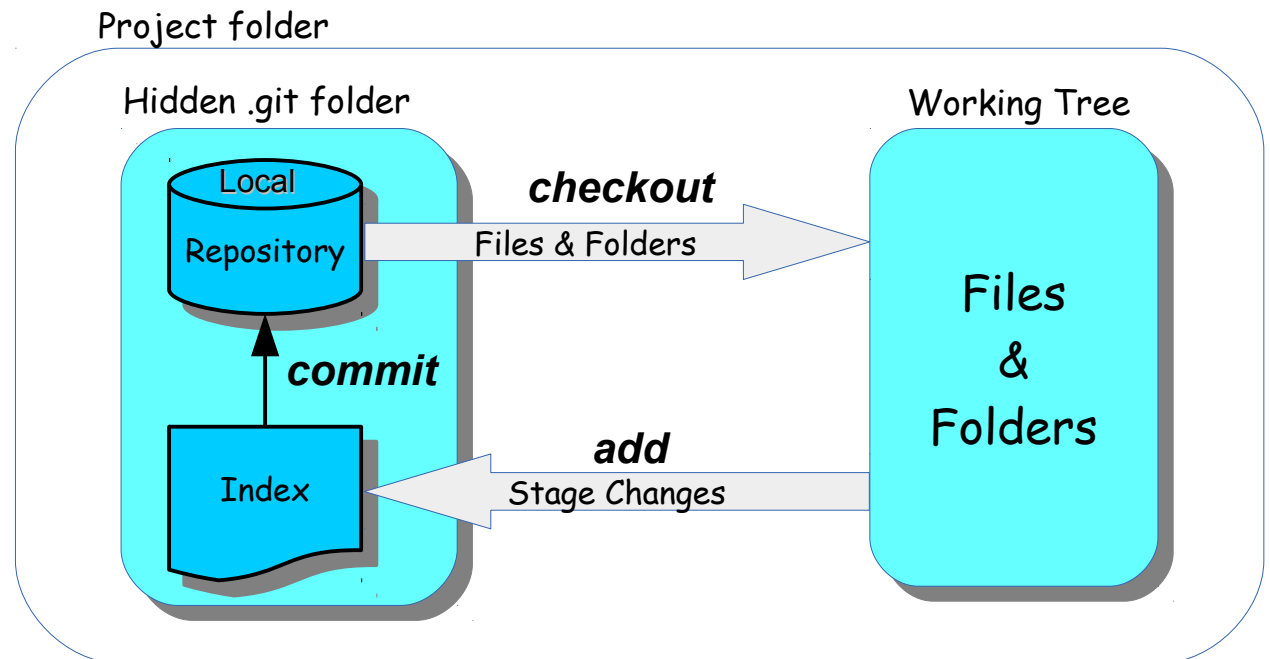
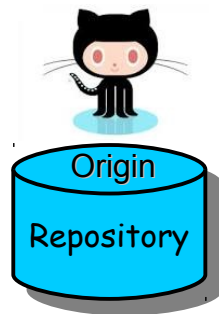
GitHub

- **Index:** Staging area for changes.
- **Commit:** A set of changes defining a version.
- **Checkout:** Restore files/folders from a commit.



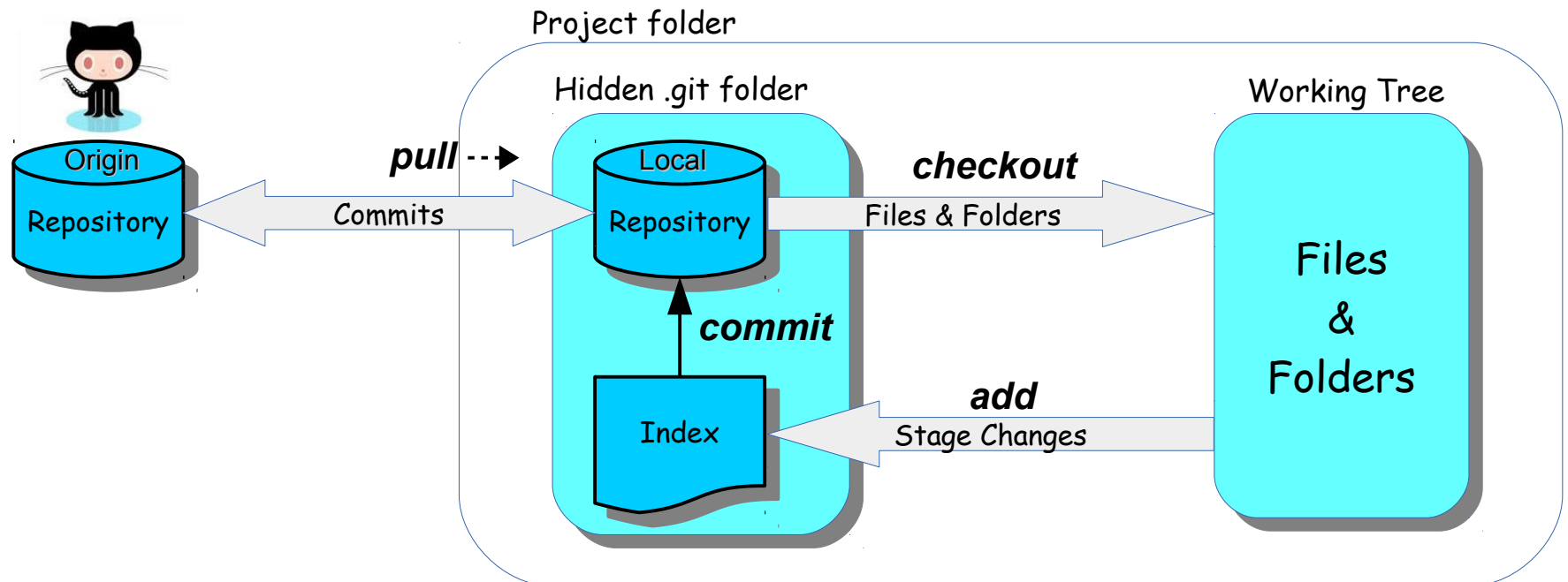
GitHub

- **Index:** Staging area for changes.
- **Commit:** A set of changes defining a version.
- **Checkout:** Restore files/folders from a commit.



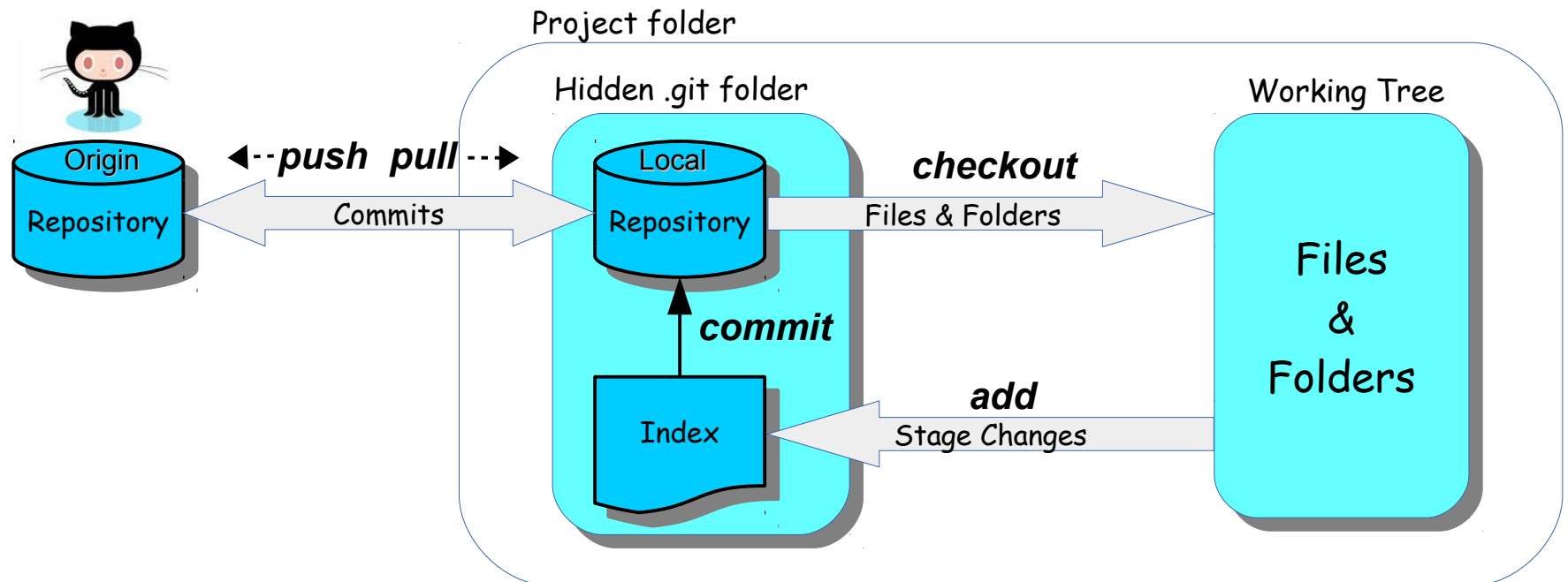
GitHub

- **Pull:** Synchronizes local repository with origin repository
 - Adds origin commits made after your last pull.
 - If un-pushed local commits, both sets of commits merged, conflicts highlighted.
- **Push:** Synchronizes origin with the local repository
 - Reverse of Pull, but if conflicts, won't work. Pull and resolve conflicts locally first.



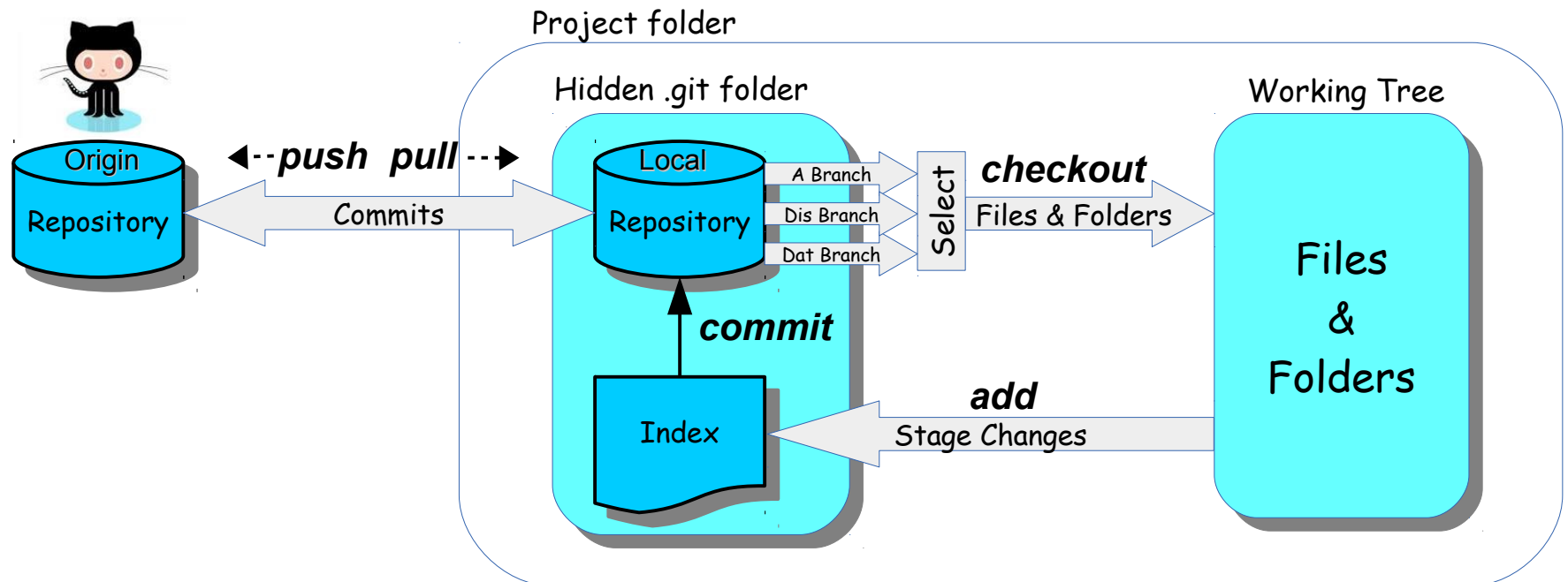
GitHub

- **Pull:** Synchronizes local repository with origin repository
 - Adds origin commits made after your last pull.
 - If un-pushed local commits, both sets of commits merged, conflicts highlighted.
- **Push:** Synchronizes origin with the local repository
 - Reverse of Pull, but if conflicts, won't work. Pull and resolve conflicts locally first.



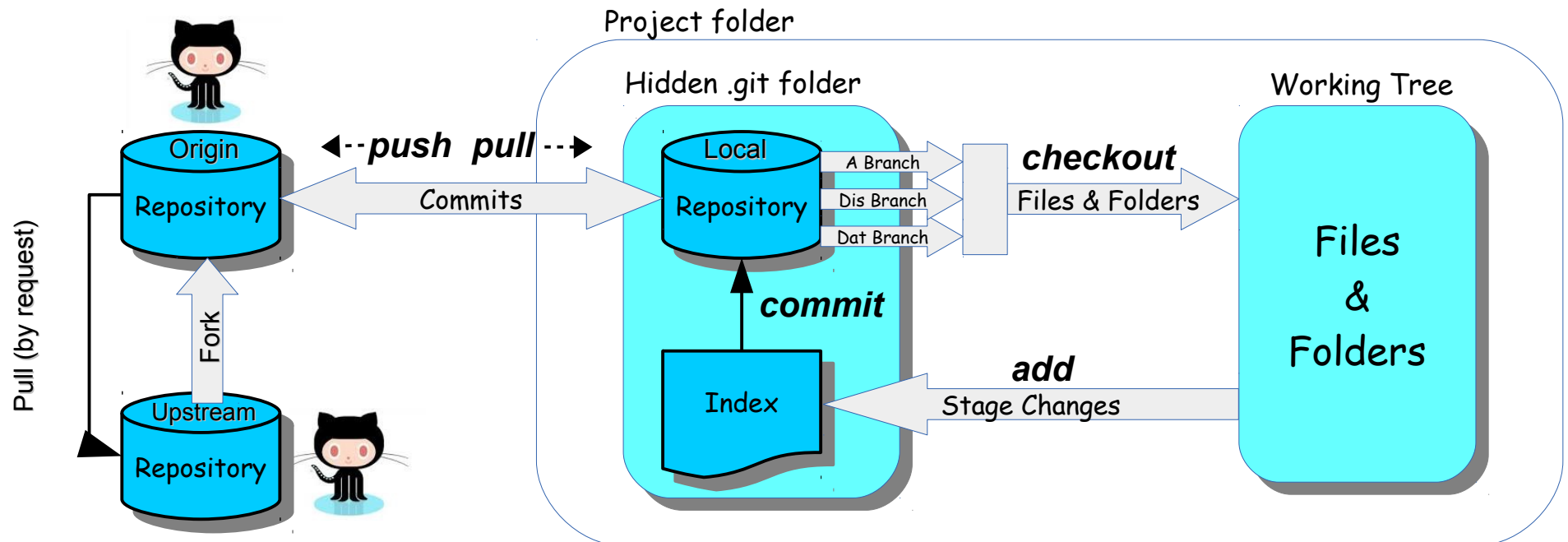
GitHub

- **Branch:** Separate version history split off from another branch.
 - Created from a commit on another branch
 - Parallel chain of commits - effectively another repo.
 - Can be merged with any other branch with usual conflict issues.



GitHub

- **Fork:** Copy a repository to your GitHub account.
 - Becomes your origin: Commit, push, pull just like any repo.
 - Original repo referred to as “upstream”.
 - Used to work on a repo without direct push access (not on the team).
 - Make a customized version, propose changes to upstream team with pull request.



GitHub

- Each Google Drive project folder is a repository.
- Some sub-directories excluded from the repository.
 - i.e., data, reference docs, ... large files that don't change.
 - “.gitignore” file says what to exclude.
- Sub-projects are git submodules within project repo
 - Independent stand-alone repositories.
 - Note: Users will need some basic command-line stuff to manage.
 - Stand-alone entities within the larger project repository.
 - Program code, Eagle design, Solidworks design, others?
 - Will enhance visibility.

GitHub Organizations

- All contributors must be GitHub *users*.
 - i.e., have an account.
- Collaborate with an *organization* on GitHub.
 - Effectively a shared user account.
 - Can *have* multiple *owners*.
 - Contributors are *members*.
 - *Members* assigned to *teams*.
 - *Teams* have access to *repositories*
 - Three levels of access: *read*, *write*, *admin*

Collaboration

Management

Our Organization

- AdlerFarHorizons
 - As a non-profit set to “Bronze” level at no cost.
 - Thanks, Arfon!
 - Up to 10 *private repositories*.
 - Proprietary collaborations.
 - Anything we don't want public.

Collaboration