Git

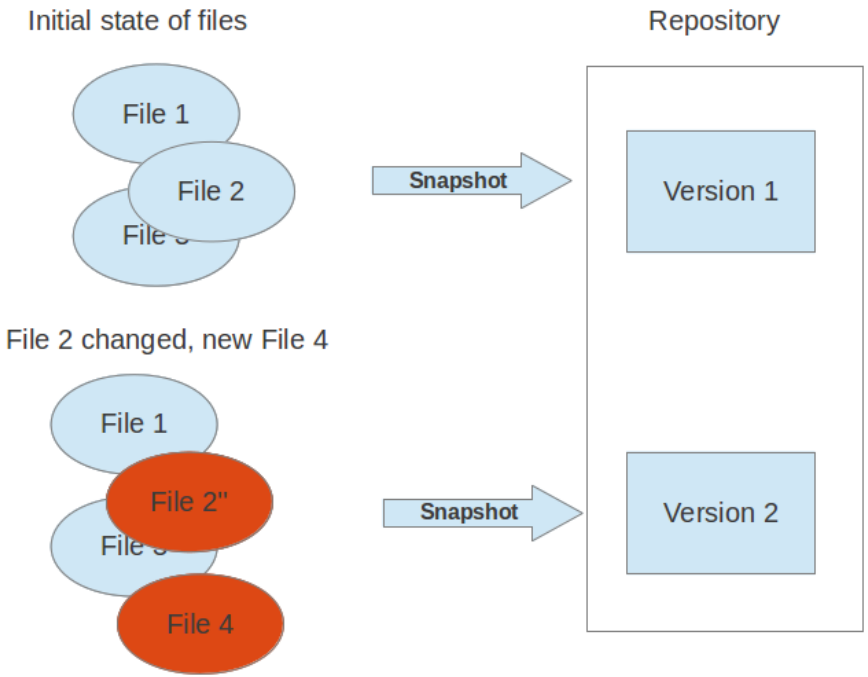
A Simple Introduction to

Daniel Tai, 2013

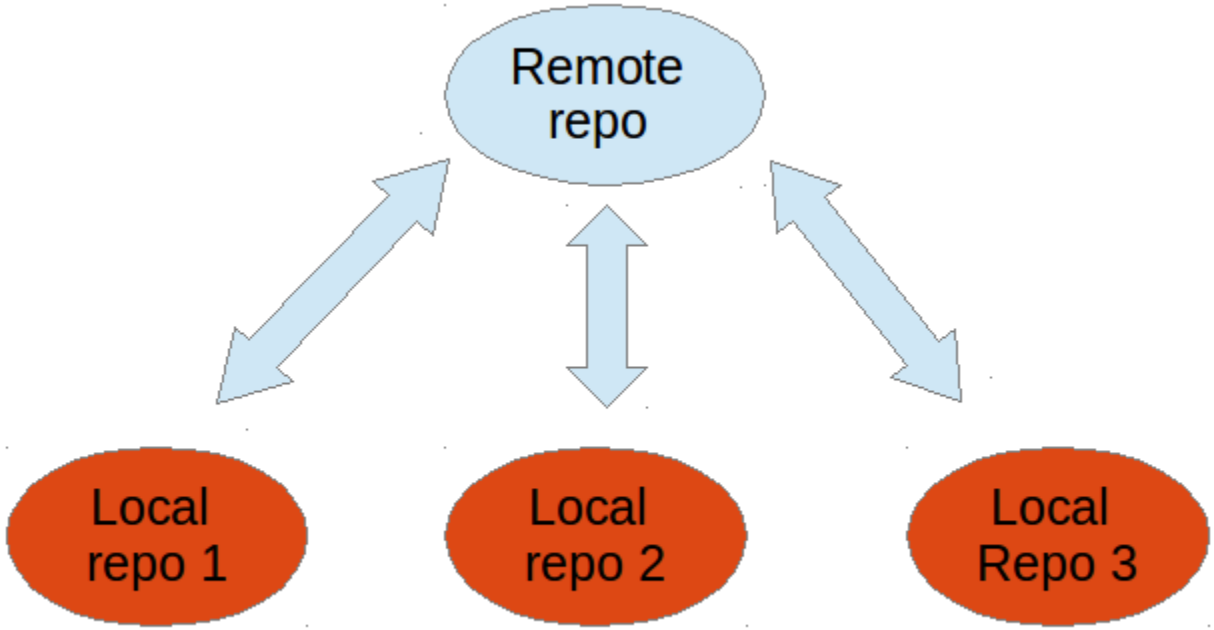
References

1. Lars Vogel, Git Tutorial
   http://www.vogella.com/articles/Git/article.html
2. Atlassian, Git Tutorial
   https://www.atlassian.com/git/
3. Github, Try Git yourself
   http://try.github.io/

What is version control system?



Distributed version control system

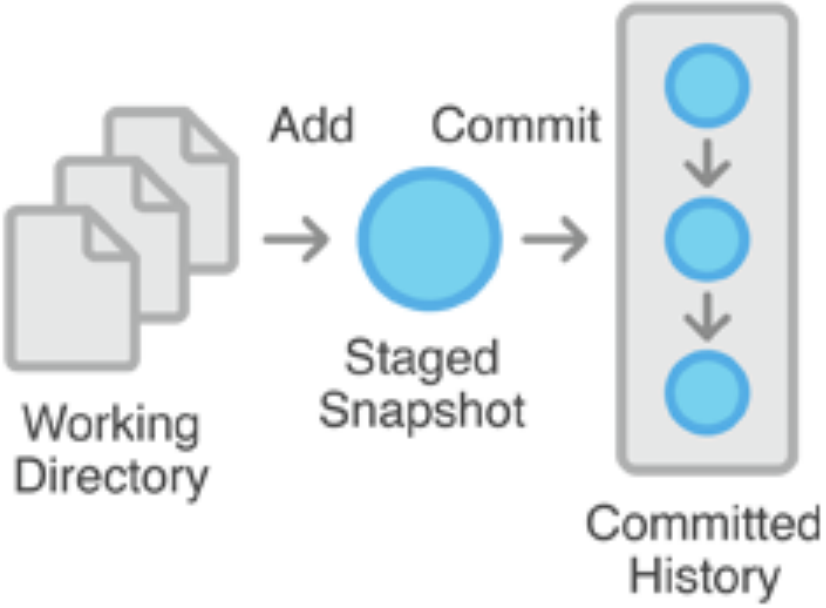


What is Git?

Simply put, Git is a distributed version control system

Local Repository Operations

Git uses "Staging" workflow



Commonly Used Commands

- **Initialize a repo**
  - `git init`
  - `git clone`
- **Add files to staging area**
  - `git add`
- **Creating a snapshot**
  - `git commit`
- **Un-track a file**
  - `git rm`
- **Configuring git**
  - `git config`
- **Checking status**
  - `git status`
  - `git diff`
  - `git log`

Check here for more detailed stuffs!

https://www.atlassian.com/git/tutorial/git-basics

Before start using git……

Setup your name and email so others can know who committed changes:

```
$ git config --global user.name "<name>"
$ git config --global user.email "<email>"
```

Initializing a repo

- `git init`
  - Create a repo locally
- `git clone <repo>`
  - Clone another repo
  - More on this later

Staging files and Reverting changes

- `git add <file>`
  - Add <file> to staging area
- `git reset <file>`
  - Remove <file> from staging area
  - See later slides for more about git reset

- Don’t get confused with: `git rm <file>`
  - Untrack and delete <file>

Creating a snapshot

- `git commit -m '<comment>'`
  - Create a snapshot of all files in staging area
  - Always add comments, so you and others can see what you've been doing

Checking Status

- `git status`
  - Check for status: staged files and new files
- `git diff`
  - Check for changes in files
- `git log`
  - Check commit log

Hands on Time!

Head to http://try.github.io and finish to challenge 9

Commonly Used Commands, cont.

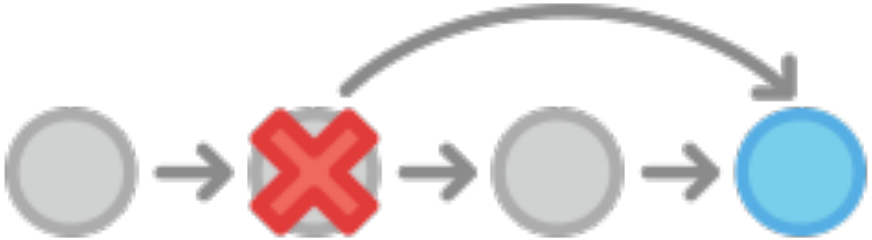
- **Viewing previous commits**
  - `git checkout`
- **Undo a commit**
  - `git revert`
- **Unstage or revert files**
  - `git reset`
- **Delete all un-tracked files**
  - `git clean`

Viewing previous commits

- `git checkout <commit>`
  - Checkout <commit>. You will be detached from master branch.
- `git checkout <commit> <file>`
  - Checkout <file> in <commit> and put it into stage area
  - Use this function to revive old version of file into new commits

Reverting old commits

- `git revert <commit>`
  - Revert to <commit> and make it as a new commit



Resetting commits

- `git reset <file>`
  - Remove <file> from staging area
  - Does not change file content
- `git reset`
  - Remove all files from staging area
  - Does not change file content
- `git reset` has some other dangerous functions, see here. Never use them if you don't fully understand it.

Working with Remote Repositories

Commands related to remote repos

- **Assigning remote repository**
  - `git remote add <url>`
- **Upload local commits**
  - `git push`
- **Download remote commits**
  - `git pull`

Working with Github

1. Apply a Github account here
   (Optionally, you can upgrade to education account here)
2. Make sure that you've setup your name and email (see this slide)
3. Follow the instructions here

Connect with remote repo

- Case 1: You've started a local repo. You want to push to remote repo
  - `git remote add origin <url>`
    - The <url> can be obtained on Github page
  - `git push -u origin master`
- Case 2: You are cloning other's remote repo
  - `git clone <url>`

Synchronizing changes

- `git pull`
  - Pull (download) remote commits and merge them into working directory
  - Merging is usually done automatically
- `git push`
  - Push (upload) your local commits

Hands on Time!

Back to http://try.github.io and finish to challenge 17

Open a repo on Github and try playing around your self

Branch and other stuff……

Check out the references for more!