repo:

- local: local computer

- remote: server

basic command:

- git init: create a repository in local

- git clone: get source from net

- git pull: sync with net to local

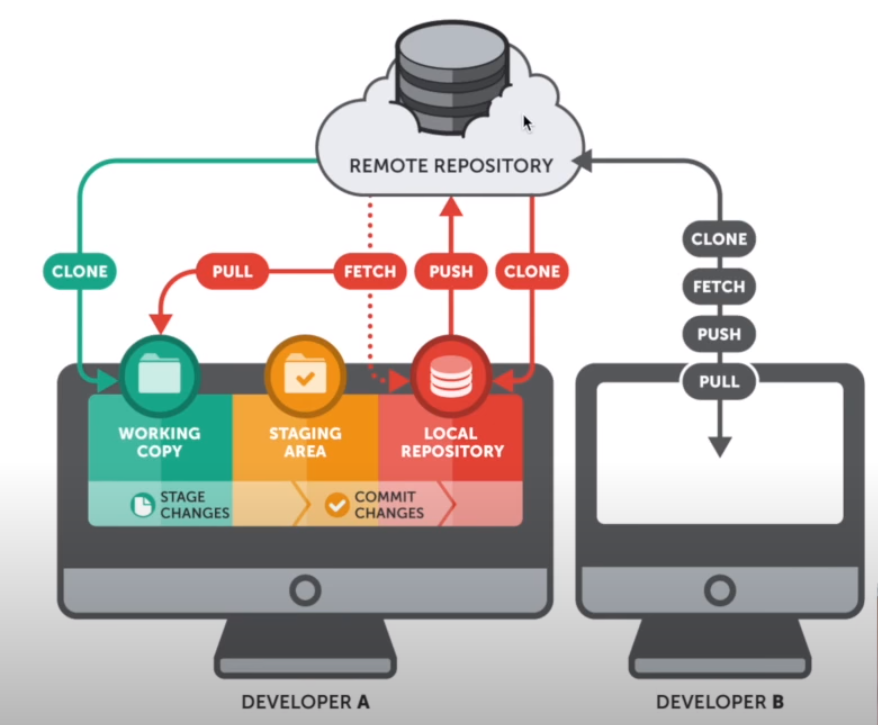
- git add/ git add . + commit: adjust file

- git push: sync from local with net

- git log: check file status

Architecture

- REMOTE REPOSITORY: can be Github or Gitlab,...



Demo

- At directory you want to push source code: *git init*

- To see status file: *git status*

- To put the file into Staging area: *git add ./ git add <file\_name>*

- git commit -m “Description for commit” (At this step, source will be in Local Repository, but network still have yet)

- To check file: *git log*

*­*- To sync net with local: git remote add origin <https://github.com/nhoangthinh/tma-git-tip.git> : mount origin tag to URL to informed for local know that: “There is a remote named origin and its url is <https…>, if you push the code on it, you will have to push to this url address.”

- git push -u origin master: push code from master branch in local on repo of github.

Resolve Conflict

If someone edit on the same file on repo, and you also edit on the same file on local. Then you:

- git add .

- git commit -m “Description”

- git push

=> ! [rejected] master -> master (fetch first)

error: failed to push some refs to 'https://github.com/nhoangthinh/tma-git-tip.git'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

- try: git pull . Then source in local will show 2 version: the first is local, the second is repo, so you will choose 1 or 2.

- try: git add .

- try: git commit -m “description”

- try: git push

ADVANCED COMMAND

- git branch

- git checkout

- git merge

**PULL REQUEST**: Show the code changes, others can see what they have changed in the code, if the code is ok, people will accept the pull request - ie merge those changes into the branch master.

Example: Add a new feature.

- git branch <name\_of\_new\_branch>

- git branch: to see all current branchs

- git checkout <name\_of\_new\_branch>: to switch code to new branch.

- git add .

- git commit -m “description”

- git push: error

fatal: The current branch branch-for-new-feature has no upstream branch.

To push the current branch and set the remote as upstream.

Error happen because on repo don’t have branch name is <name\_of\_new\_branch>. So to solve this problem. git push --set-upstream origin <name\_of\_new\_branch> to create this branch on repo.

Go to repo on github, you will se a pull request, if source code OK, you can merge to branch master now.

- git checkout master: to back branch master

- git merge <name\_of\_new\_branch>

- git push

**CREATE PULL REQUEST**:

- git branch <name\_of\_new\_branch>

- git branch: to see all current branchs

- git checkout <name\_of\_new\_branch>: to switch code to new branch.

- git add .

- git commit -m “description”

- git push --set-upstream origin <name\_of\_new\_branch>