Git: Day 3

REMOTE REPOSITORIES

Concepts

A remote repository is a reference to another repository.

A remote is a shorthand name for a Git URL.

You can define any number of remotes in a repository

Once a remote is established, Git can transfer data from one repository to another using push and pull model.

To keep track of data from other repositories, Git uses tracking branches.

Each tracking branch in your repository is a local branch that serves as a proxy for a specific branch in a remote repository.

Remotes

Git uses remote and tracking branch to reference and facilitate the “connection” to another repository.

$ git remote: to manipulate remotes

Remotes

$ git remote

add: adds a remote

rm: deletes a remote

rename: rename a remote repo

-v: list all remotes

Remotes

In addition to git clone , other common Git commands that refer to remote repositories are:

- git fetch: Retrieves objects and their related metadata from a remote repository
- git pull: Fetch + Merge
- git push: Transfers objects and their related metadata to a remote repository
- git ls-remote: Shows references within a remote

Tracking branches

Used exclusively to follow the changes from another repository.

Not merge or make commits onto a tracking branch

Tracking branches

```
$ git fetch <remote-repo>
```

```
$ git checkout --track –b <local-branch> <remote-repo> / <remote-branch>
```

Push to a remote branches

Push

$ git push <repo> <branch>

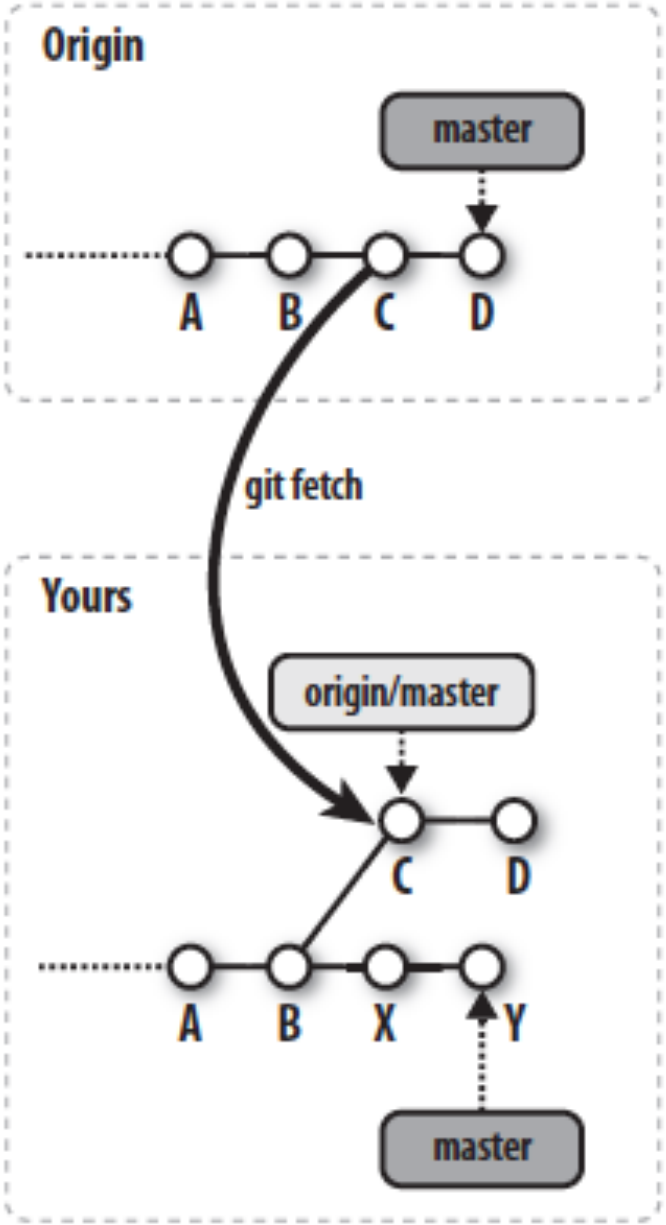
Force Push

$ git push –f <repo> <branch>

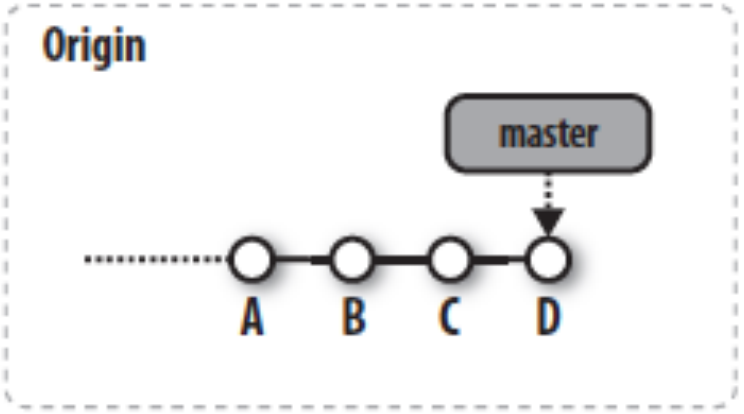
¡Be careful forcing the push!

Fetch

$ git fetch <repo> <branch>

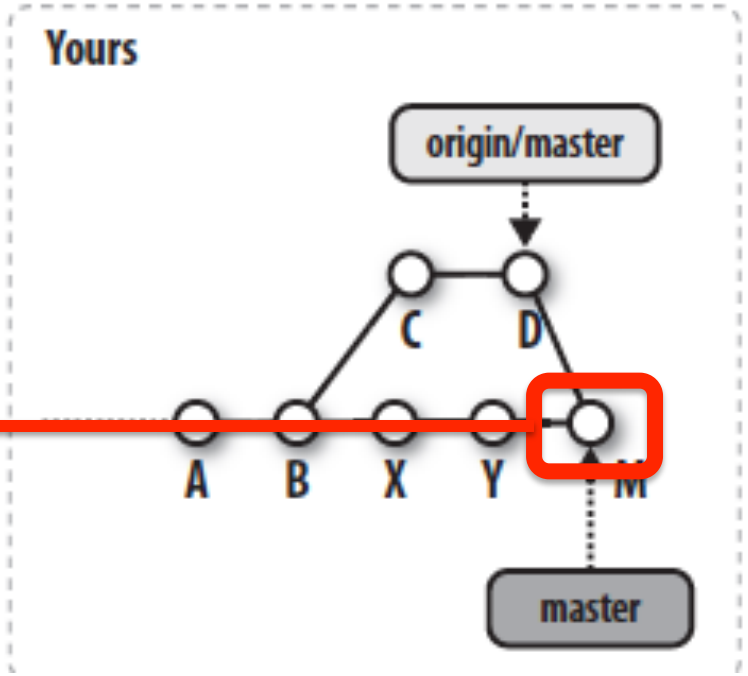


Merge



$ git merge <branch>

New commit is created



Abort a merge

```
$ git reset --hard ORIG_HEAD
```

Pull

$ git pull <repo> <branch>

Git Fetch + Git Merge → Exactly the same as doing separated

MERGE

What is a merge

Unifies two or more commit history branches.

Most often, a merge unites just two branches.

Git supports a merge of three, four or many branches at the same time

All the branches to be merged must be present in the same repository.

What is a merge

When modifications in one branch do not conflict in another branch → new commit

When branches conflict (alter the same line) Git does not resolve the dispute.

Merge examples

You have to switch to the target branch and execute the merge:

```
$ git checkout destinyBranch
```

```
$ git merge anotherBranch
```

Preparing for a Merge

First of begin a merge → tidy up working directory

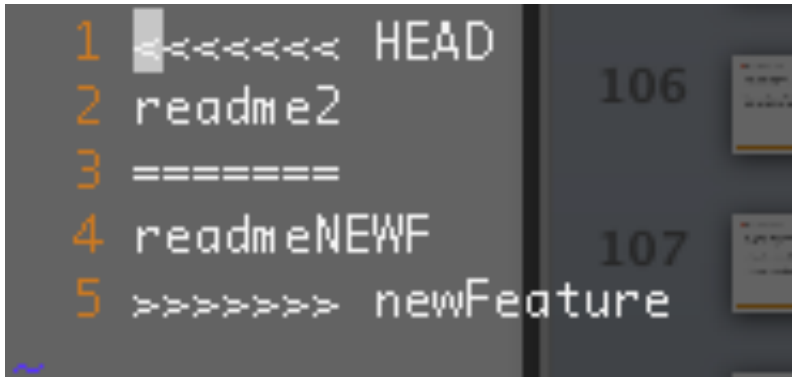
If you start a merge in a dirty state, Git may be unable to combine the changes from all the branches and those in your working directory or index in one pass.

Merge Conflicts

Git warns you about the conflict:

```
israelalcazar@Mac-Lamaro:~/dev/git/examples2 (master) $ git merge newFeature
Auto-merging README3.TXT
CONFLICT (content): Merge conflict in README3.TXT
Automatic merge failed; fix conflicts and then commit the result.
```

And marks the file:



Locating conflicted Files

Which files have got conflicts?

Git keeps track of problematic files by marking each one in the index as conflicted or unmerged

You can use one of these commands:

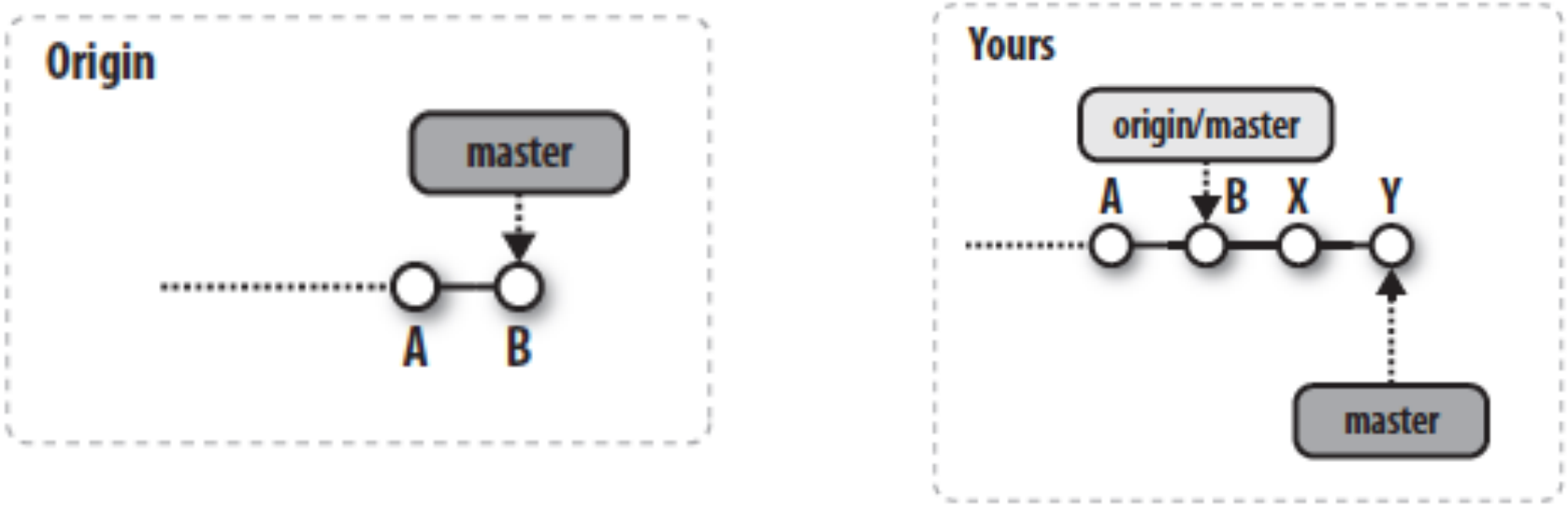
$ git status

$ git diff

$ git ls-files -u

Fast Forward

A simple linear history advancement operation



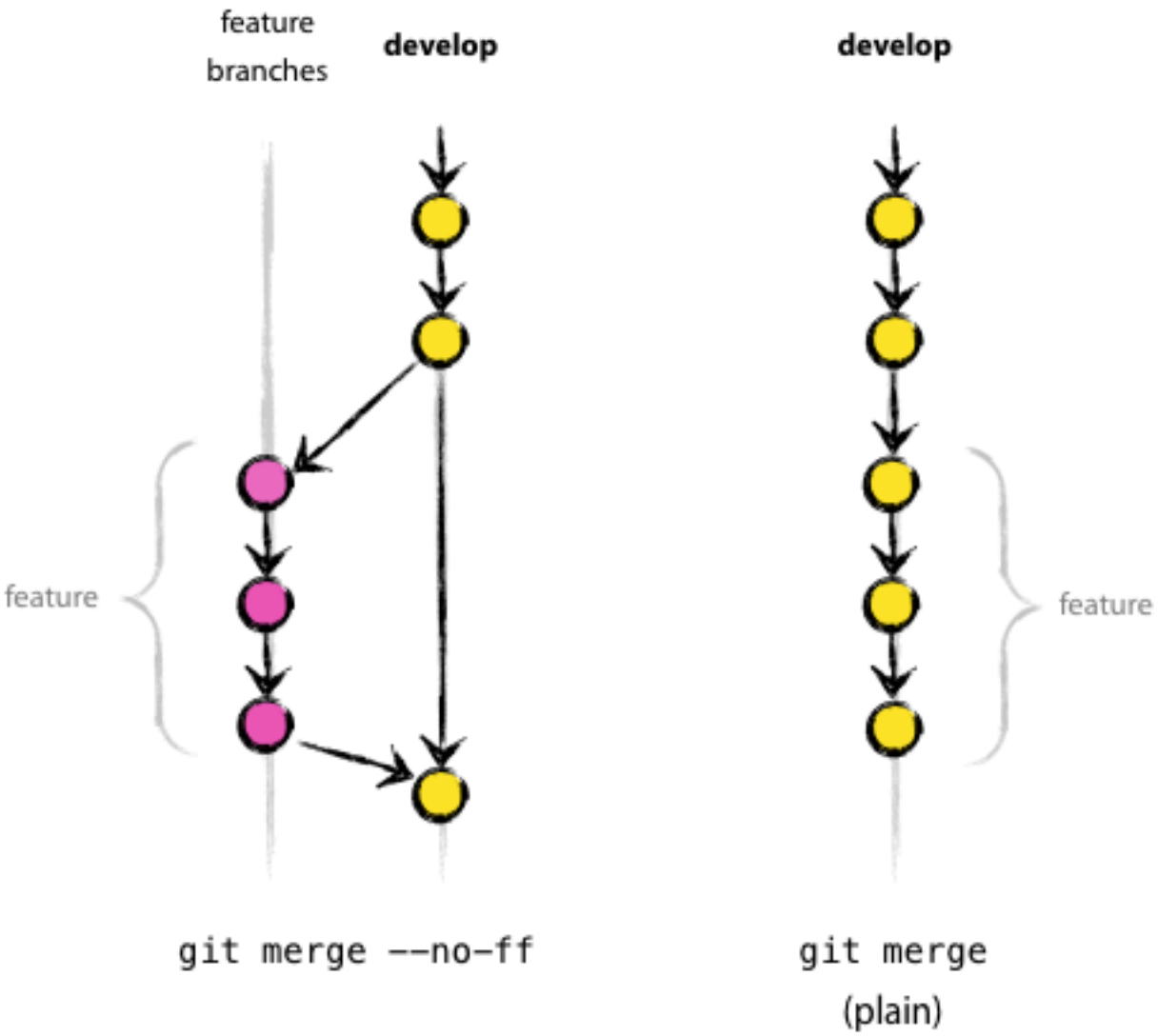
Fast forward has happened in origin from B to X

Fast Forward

If another developer has pushed some changes (C, D) you can’t push with fast forward.

You should merge your changes first.

Fast Forward Merging



Fast Forward Merging

- http://nathaniel.themccallums.org/2010/10/18/using-git-fast-forward-merging-to-keep-branches-in-sync/

- http://alblue.bandlem.com/2011/11/git-tip-of-week-git-flow.html

Preguntas

Israel Alcázar
israelalcazar@gmail.com
@ialcazar