Git flow in Framgia

Chu Anh Tuan @Framgia

Content

- Git basic
- Git flow in Framgia

Git Basic

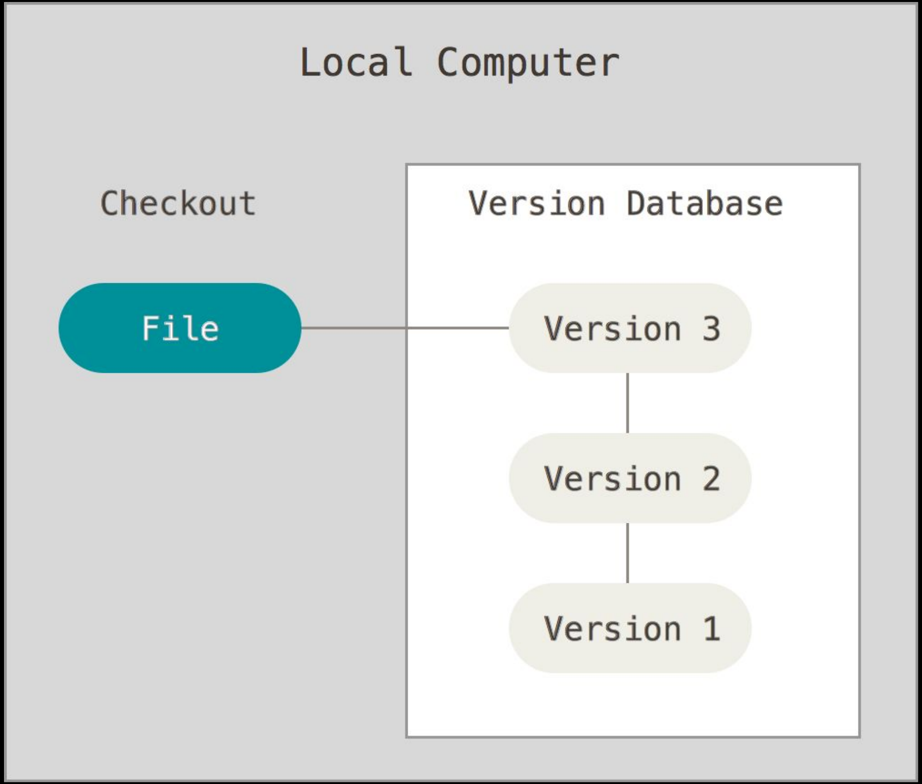
About Git

Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later

- CVS
- SVN
- Git

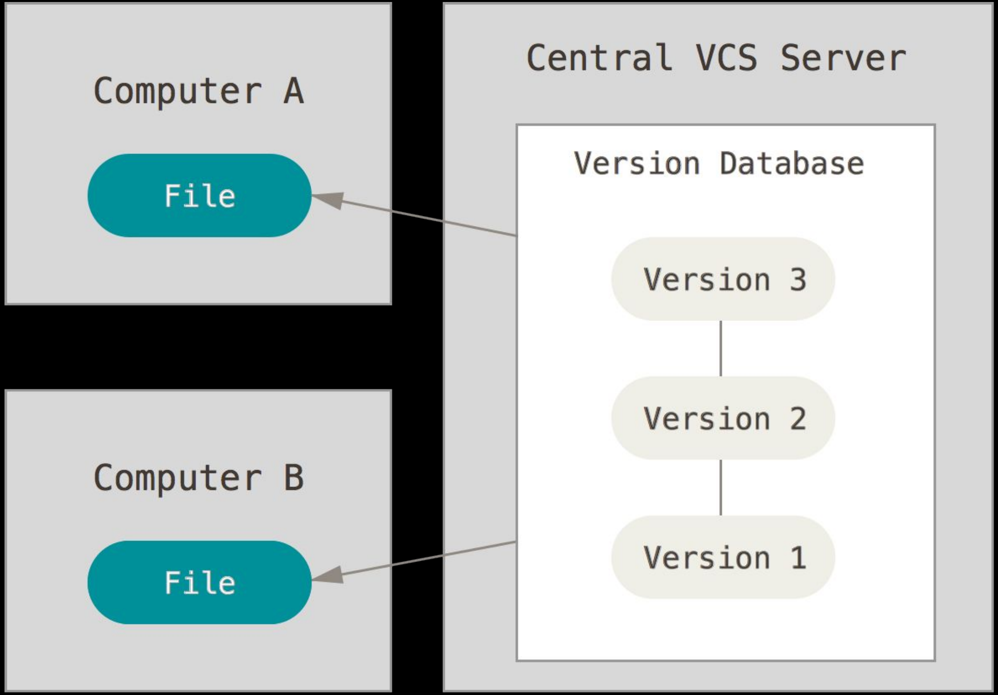
Local Version Control Systems

Keep all the changes to files under revision control



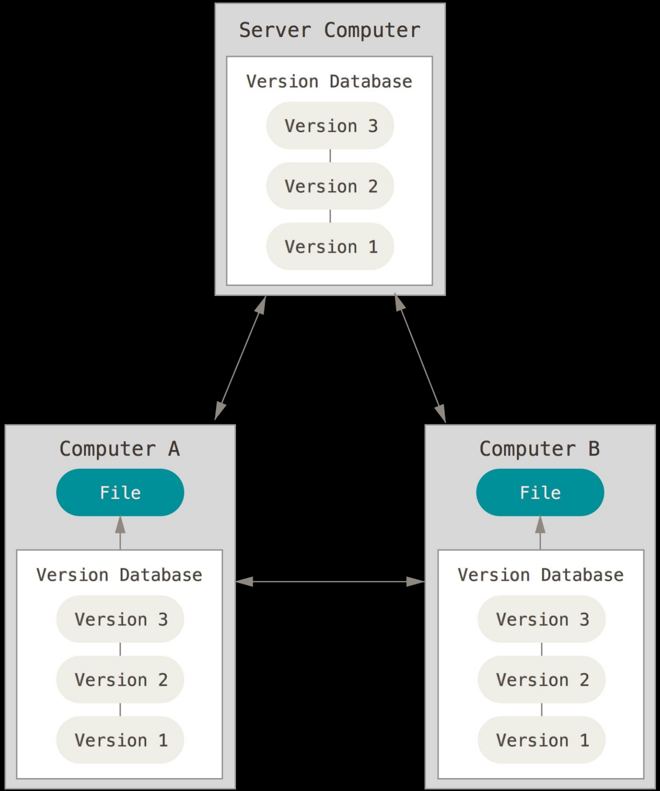
Centralized Version Control Systems

Collaborate with developers on other systems. Have a single server that contains all the versioned files, and a number of clients that check out files from that central place



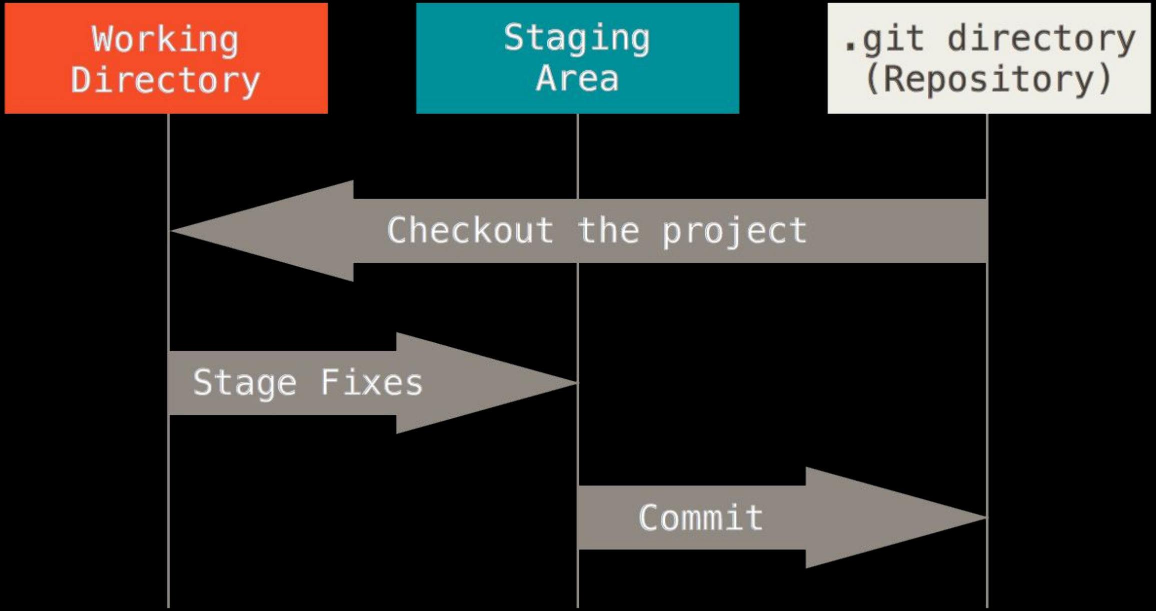
Distributed Version Control Systems

Fully mirror the repository, if any server dies, and these systems were collaborating via it, any of the client repositories can be copied back up to the server to restore it. Every checkout is really a full backup of all the data

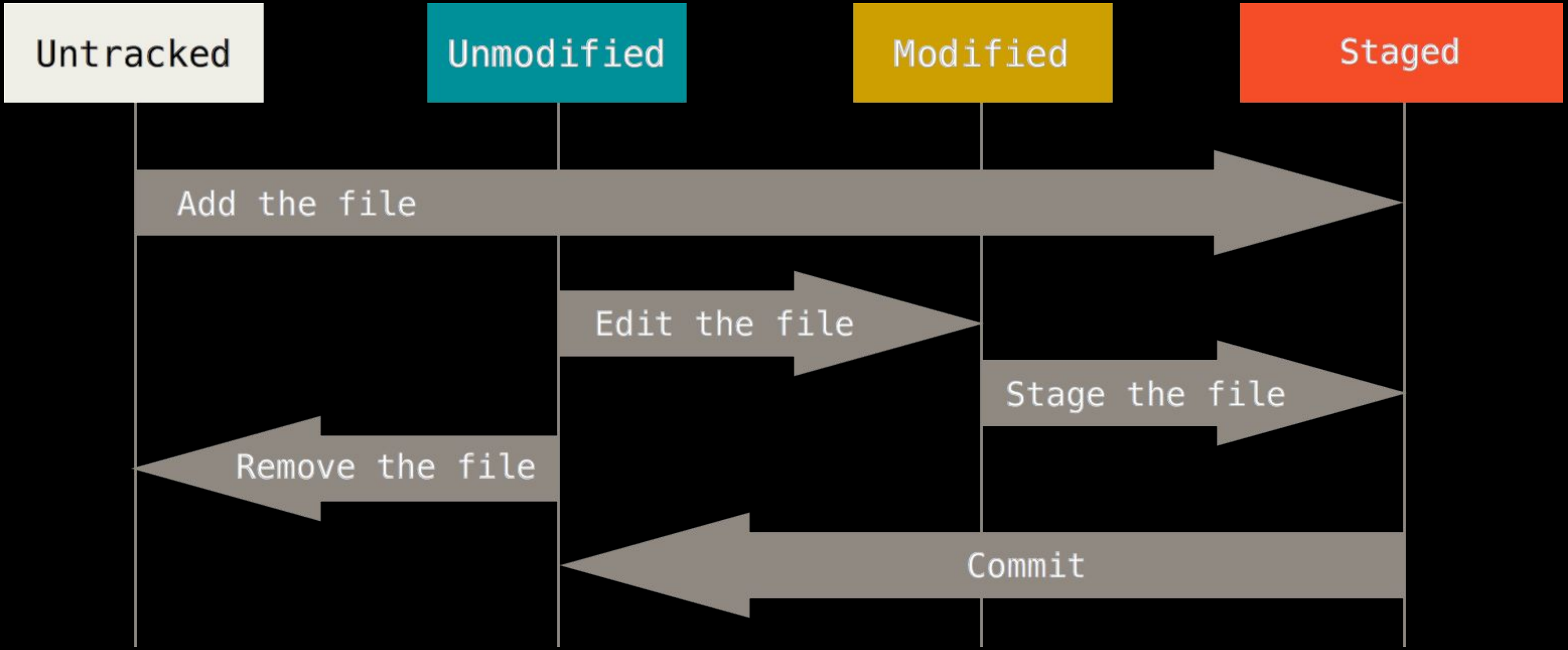


The Three States

```
$ git add
$ git commit
$ git checkout
```

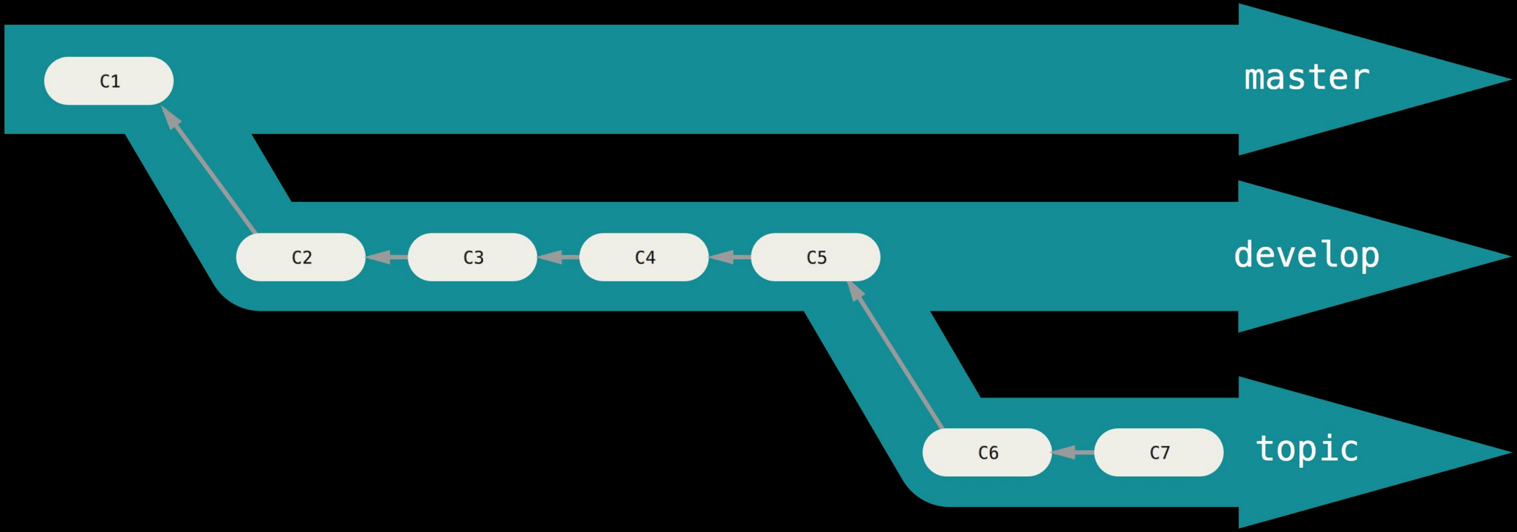


Recording Changes to the Repository



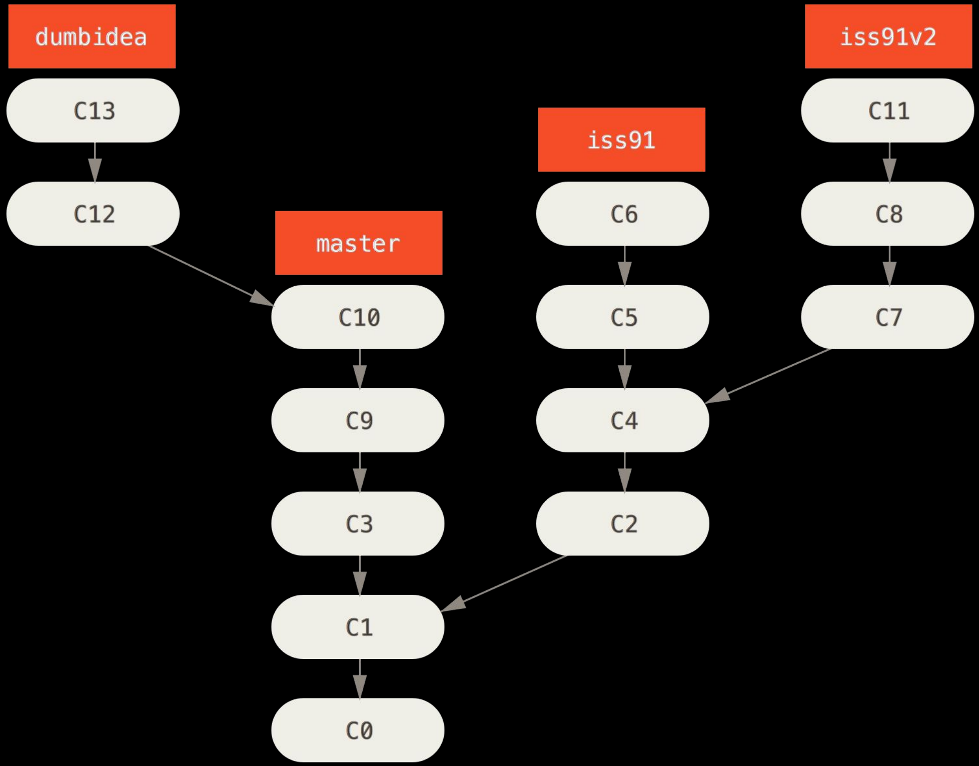
Git Branching Workflows

```
$ git branch develop
$ git checkout -b topic
```



Git Branching Workflows

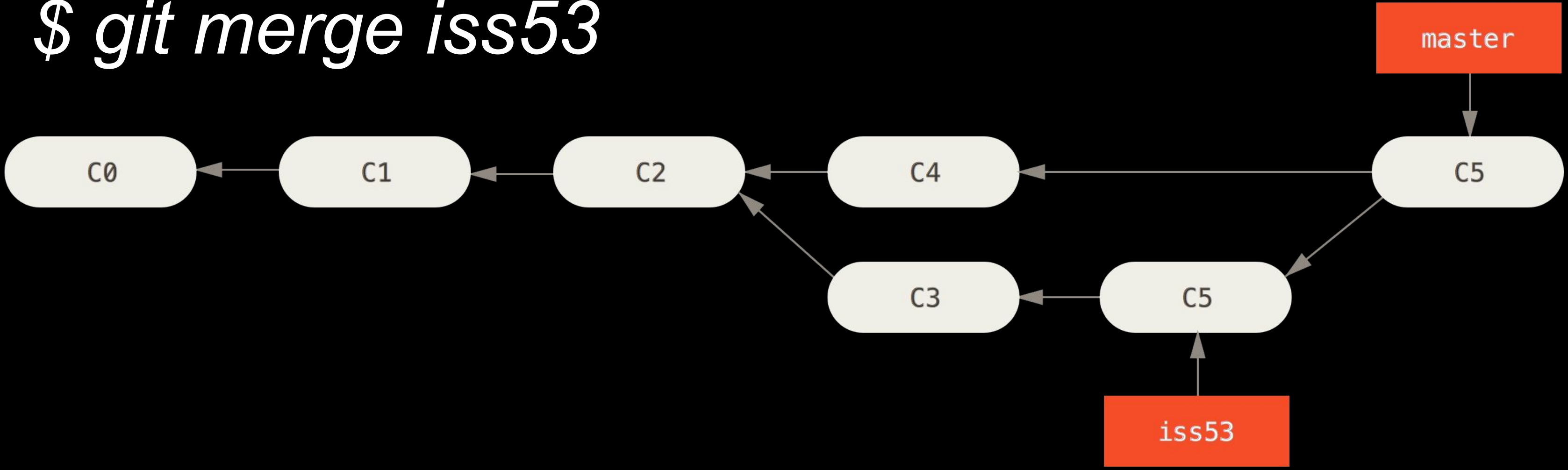
Multiple topic branches



Branching and Merging

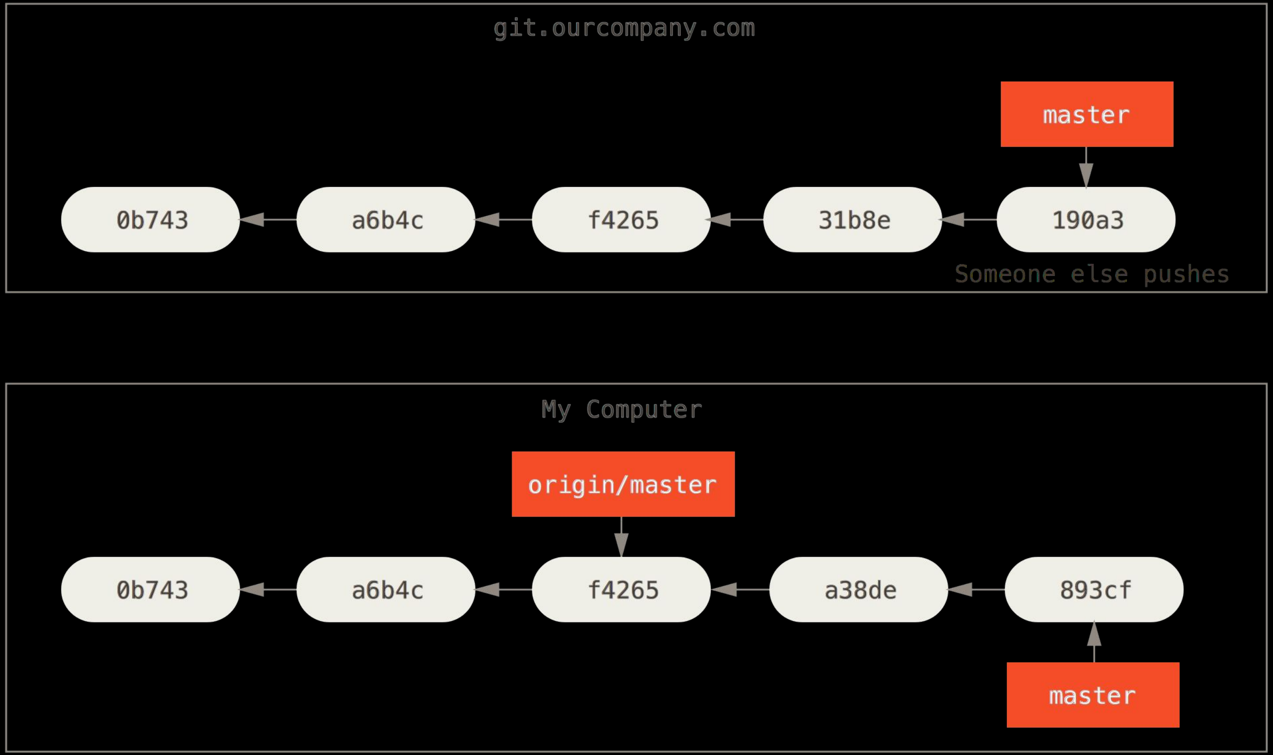
A merge commit

```
$ git checkout master
$ git merge iss53
```



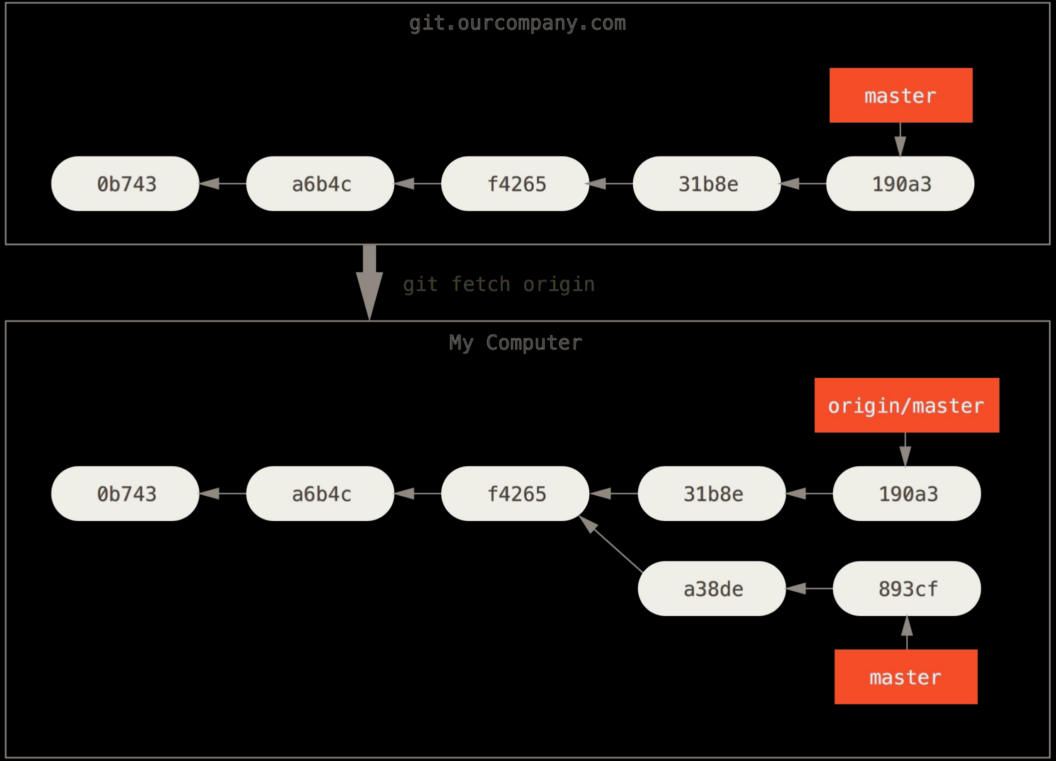
Remote Branches

Local and remote work can diverge



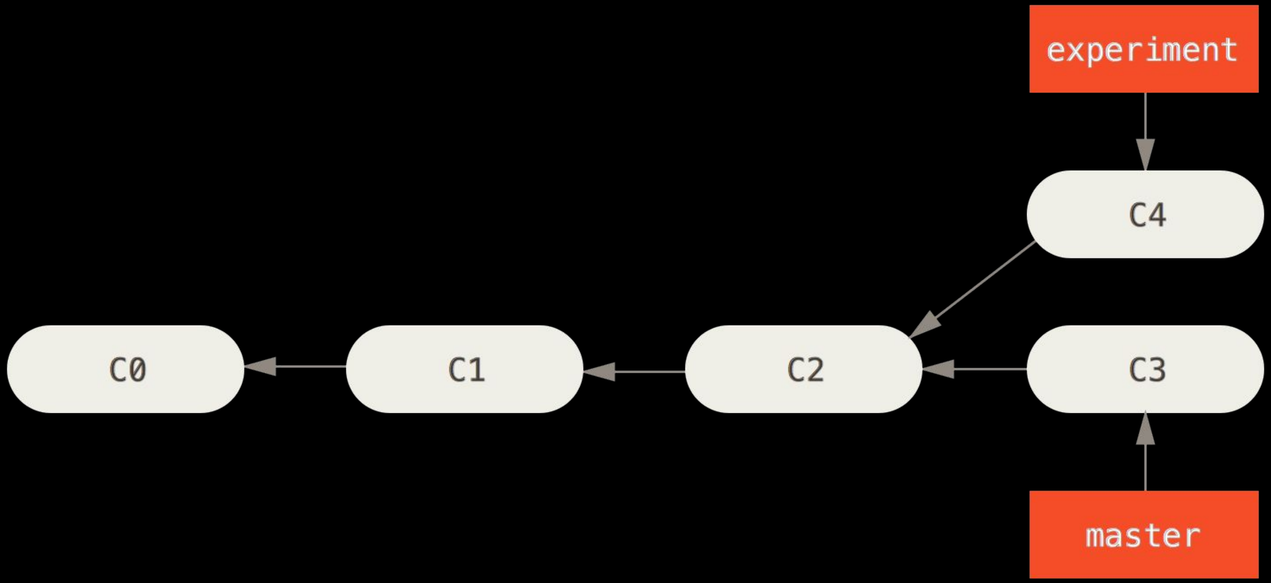
Remote Branches

git fetch updates your remote references



Rebasing

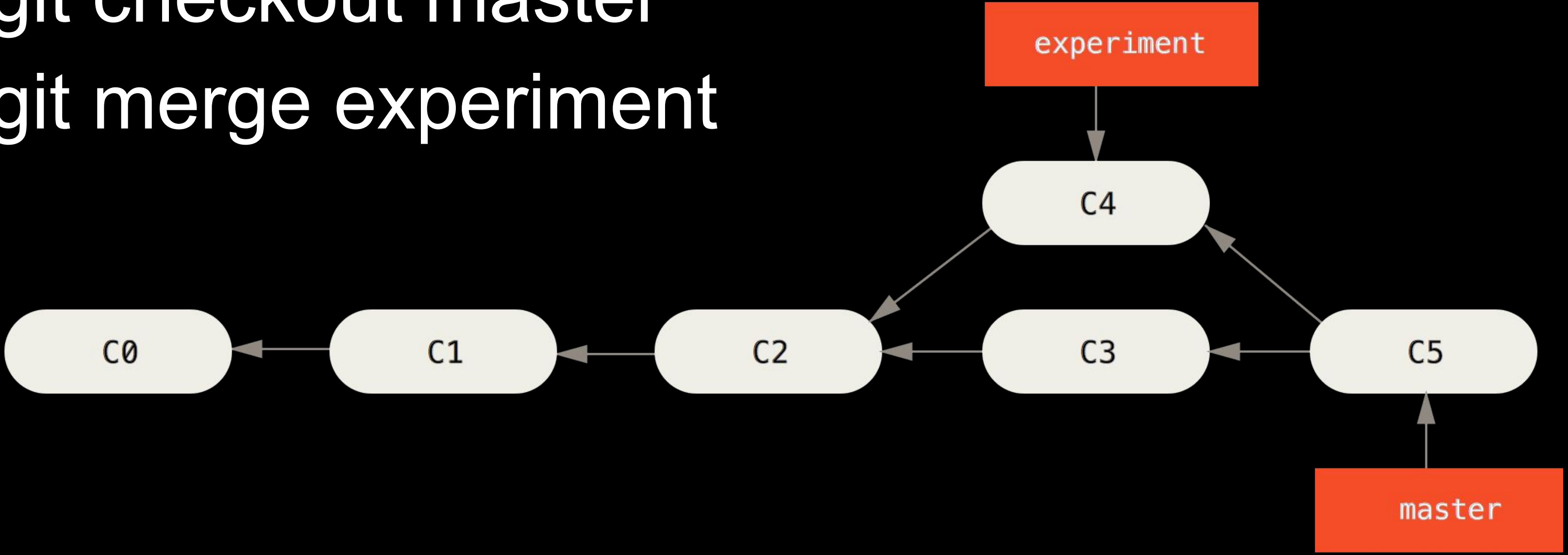
Simple divergent history



Rebasing

Merging to integrate diverged work history

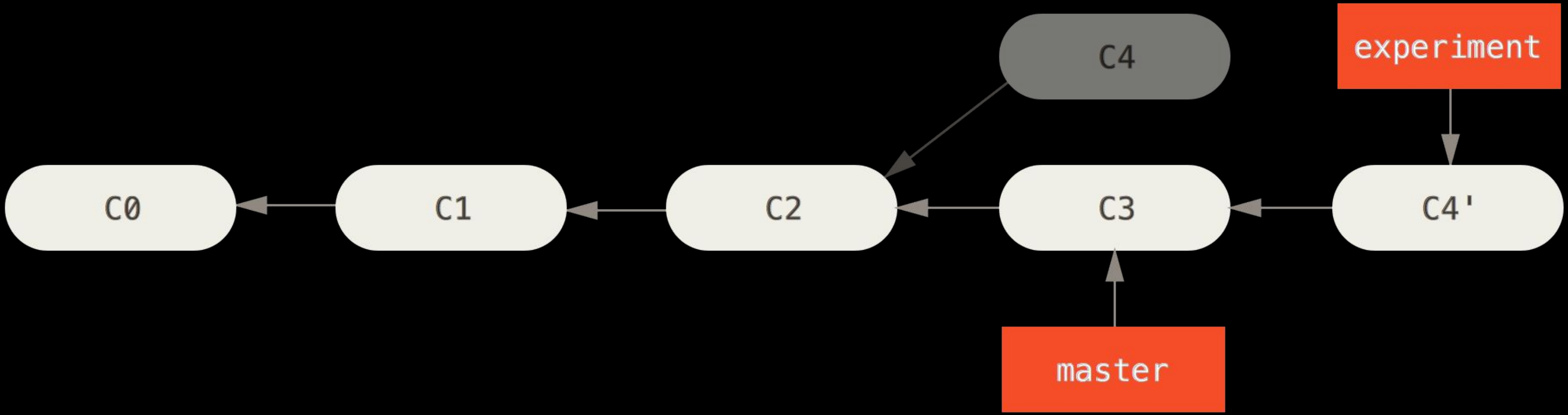
```
$ git checkout master
$ git merge experiment
```



Rebasing

Rebasing the change introduced in C4 onto C3

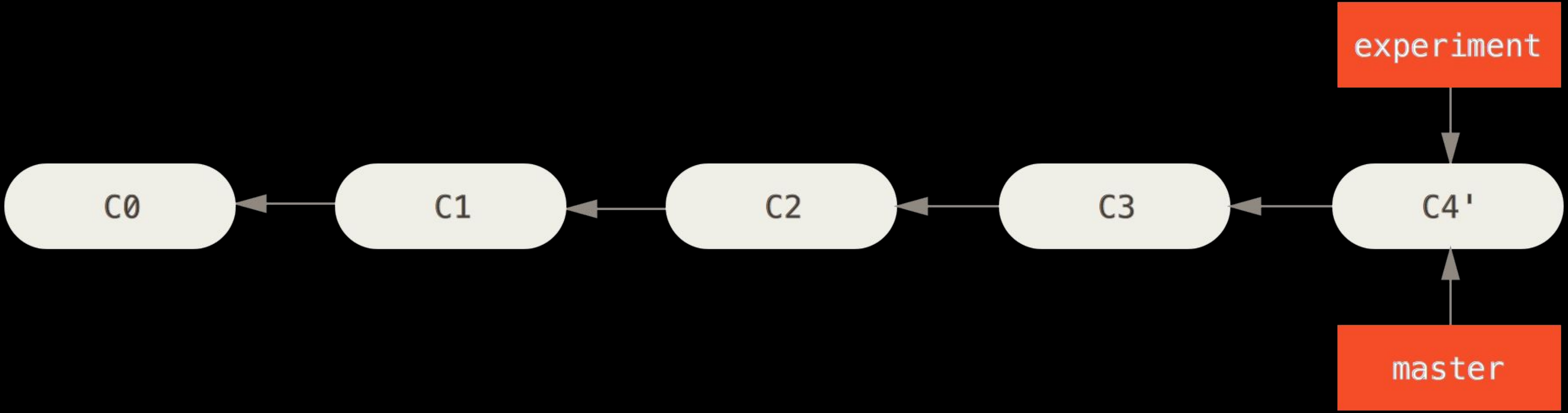
```
$ git checkout experiment
$ git rebase master
```



Rebasing

Fast-forwarding the master branch

```
$ git checkout master
$ git merge experiment
```

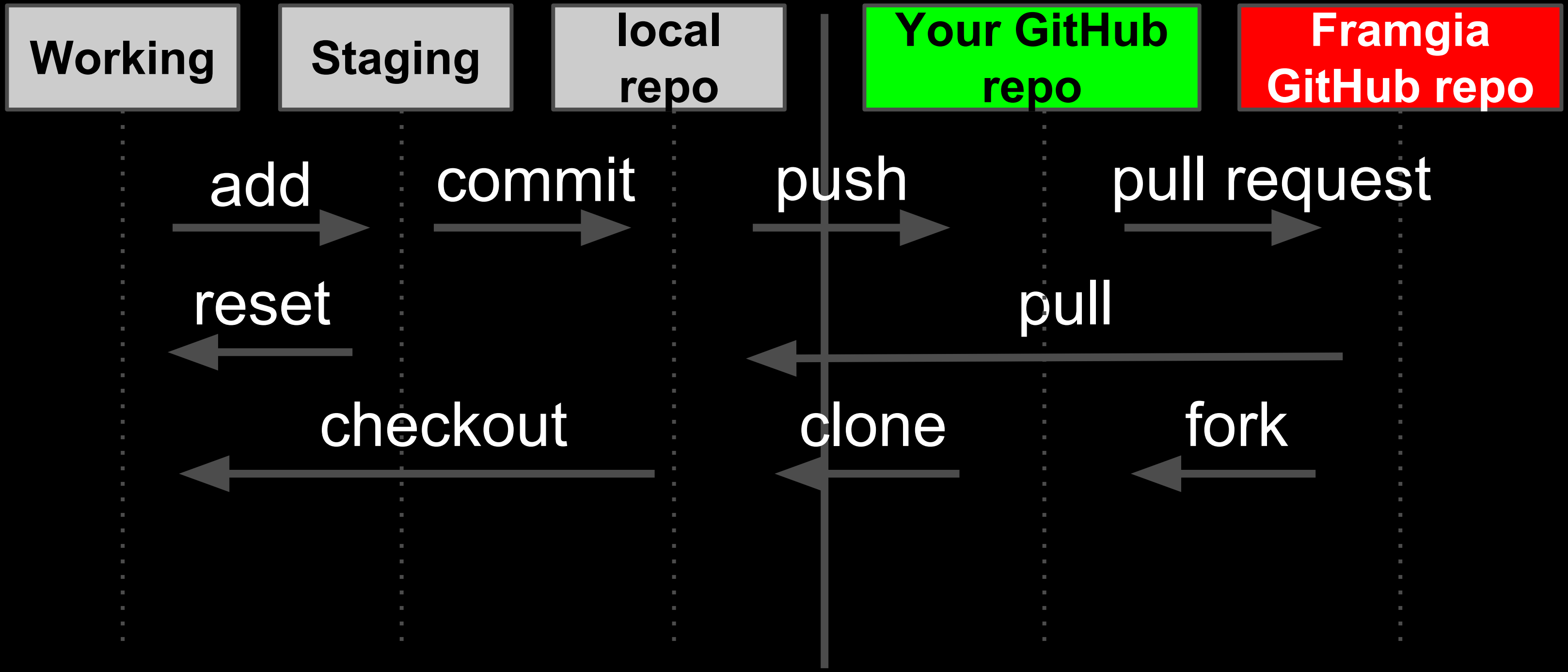


Rebasing

There is no difference in the end product of the integration, but rebasing makes for a cleaner history.
If you examine the log of a rebased branch, it looks like a linear history: it appears that all the work happened in series, even when it originally happened in parallel.

Git flow in Framgia

Git flow in Framgia



Git flow in Framgia

1. Fork
2. clone
3. remote add framgia
*4. checkout framgia/develop

Git flow in Framgia

4. checkout framgia/develop
   If default branch is master:
   4.1.1 checkout -b develop
   4.1.2 pull framgia develop

   4.2.1 fetch framgia develop
   4.2.2 checkout framgia/develop
   4.2.3 checkout -b develop

Git flow in Framgia

5. checkout -b new_branch
6. add/commit
*7. rebase develop
*8. push origin new_branch
9. make pull request

Git flow in Framgia

7. rebase develop
    - 7.1 checkout develop
    - 7.2 pull framgia develop
    - 7.3 checkout new_branch
    - 7.4 rebase develop

Git flow in Framgia

7.4 rebase develop
  If conflict:
   7.4.1 (no branch)
   7.4.2 fix conflict
   7.4.3 add
   7.4.4 rebase --continue

Git flow in Framgia

- push with force update
  push origin new_branch -f

- 1 commit/1 pull request
  commit --amend
  or
  rebase -i

Reference

Git book:

http://git-scm.com/book

Git repository service:

https://github.com/