Tutorial #4: Source Control

CS374: Introduction to HCI

2017. 5. 1

Hyeungshik Jung

Before start

- This tutorial will not cover how to create repo, make branch, merge, push…
    - You can learn it by yourself with the materials in the last slide
- Instead, I'll try to

Outline

- Why do we need Version Control System
- Git
- Repository / Commit / Branch
- Merge / Rebase / Conflict
- Remote repository
- Working with others
- Recommended Materials

Why do we need Version Control System

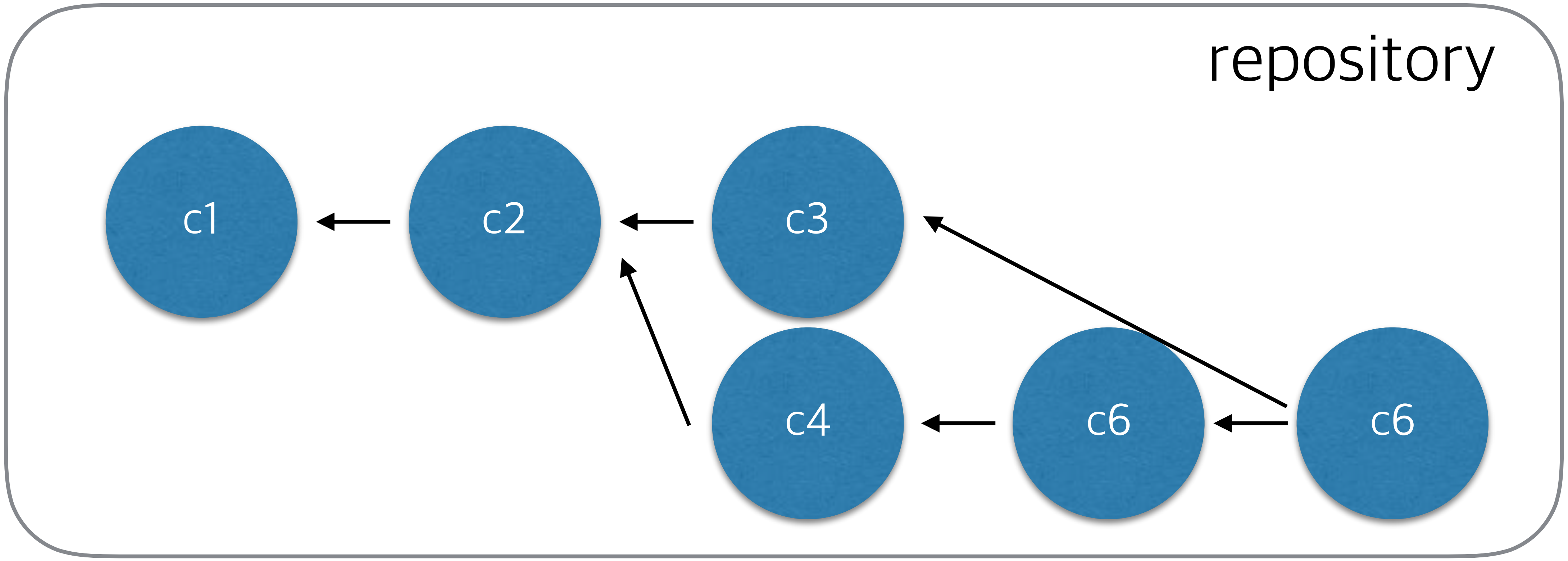
- Version Control System
  - Store versions of file and retrieve them
- Why we need it?
  - Roll back a change that caused a bug
  - Separate deployed version from development version
  - Document history of who did what when

Git

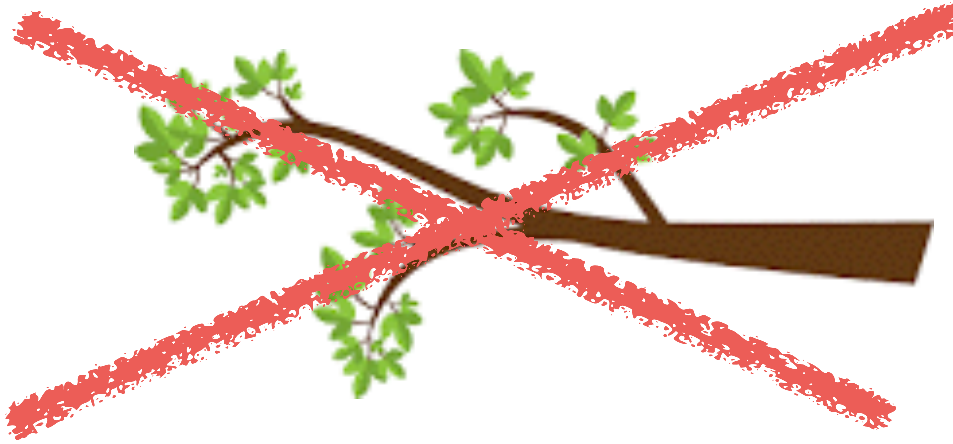
- Built by Linus Torvalds at 2005
- De facto standard in open source software
- But Git is not the only Version Control System.

Repository

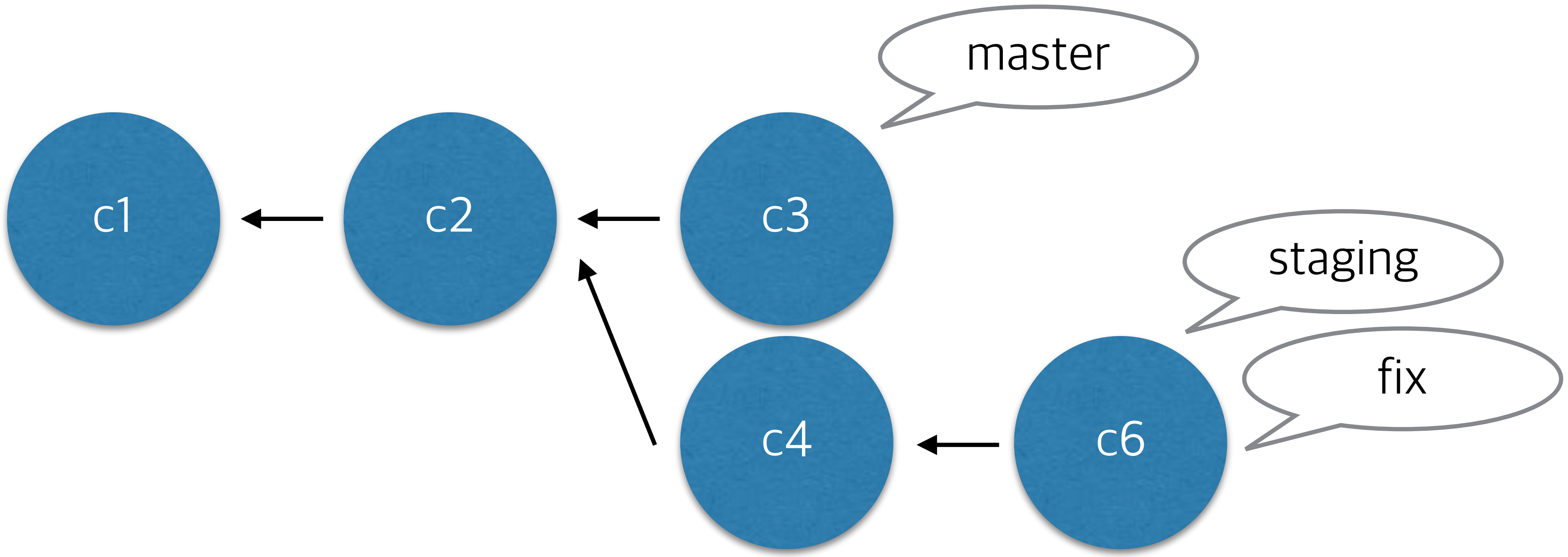
- **Repository** stores history of development
  - “Directed Acyclic Graph with Root”
- Node (“**Commit**”): A snapshot of codes from some point
- Edge: Point to the parent node



Branch

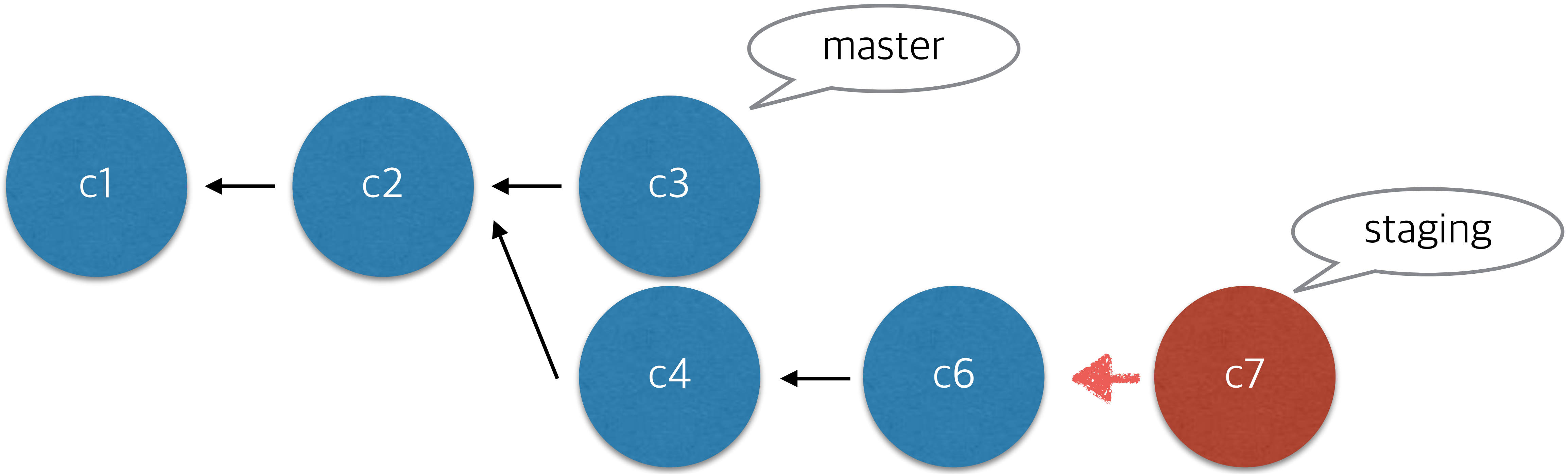


- Pointer for **A** commit (Not a series of commits)
- Creating a branch = Making a new pointer



Commit

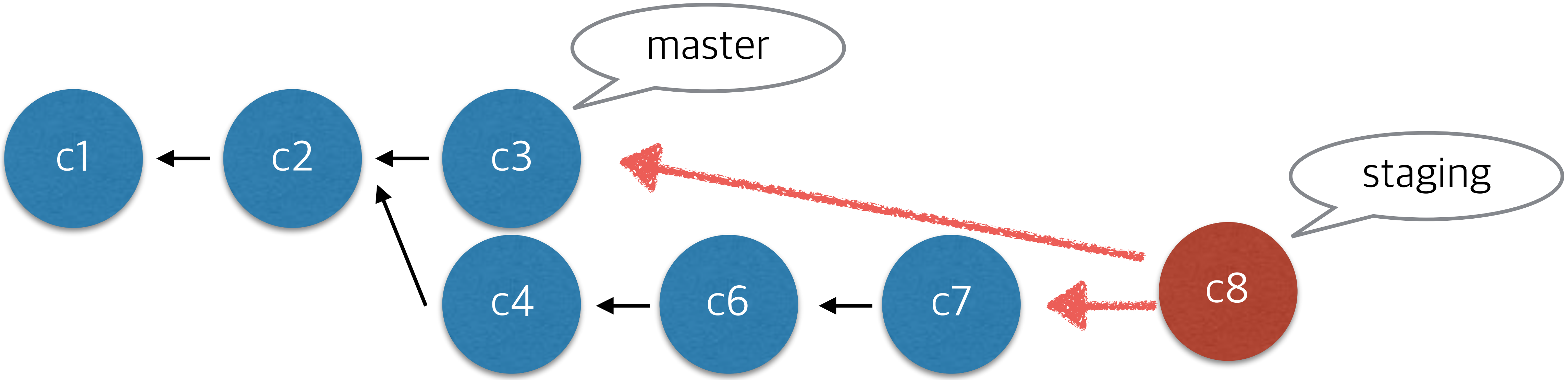
- Add a “fresh” commit with some change



Merge

- Add a commit by merging two branches (3-way-merge)

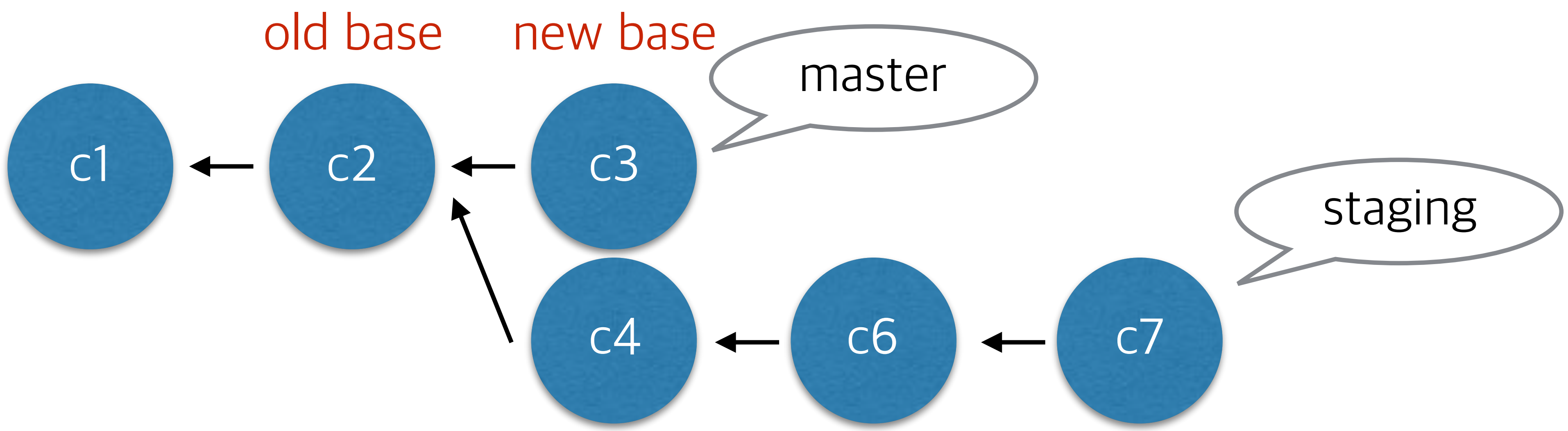
```
(staging) git merge master
```



Rebase

- "Re" + "Base": Set new base
- Base: common ancestor

```
(staging) git rebase master
```

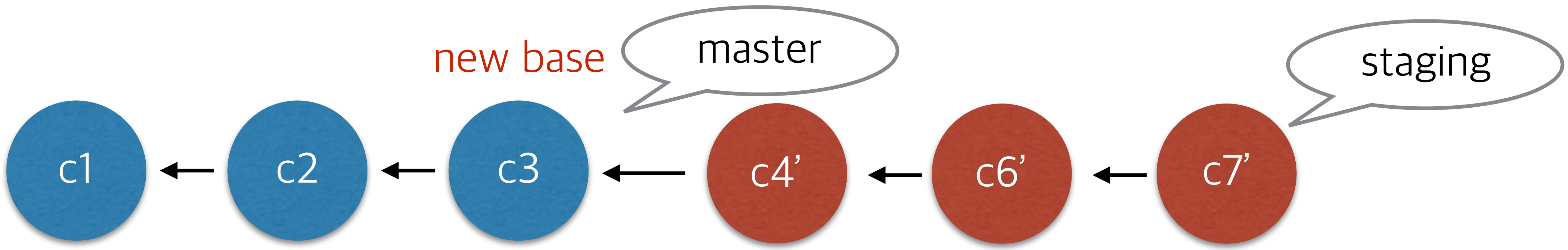


Rebase

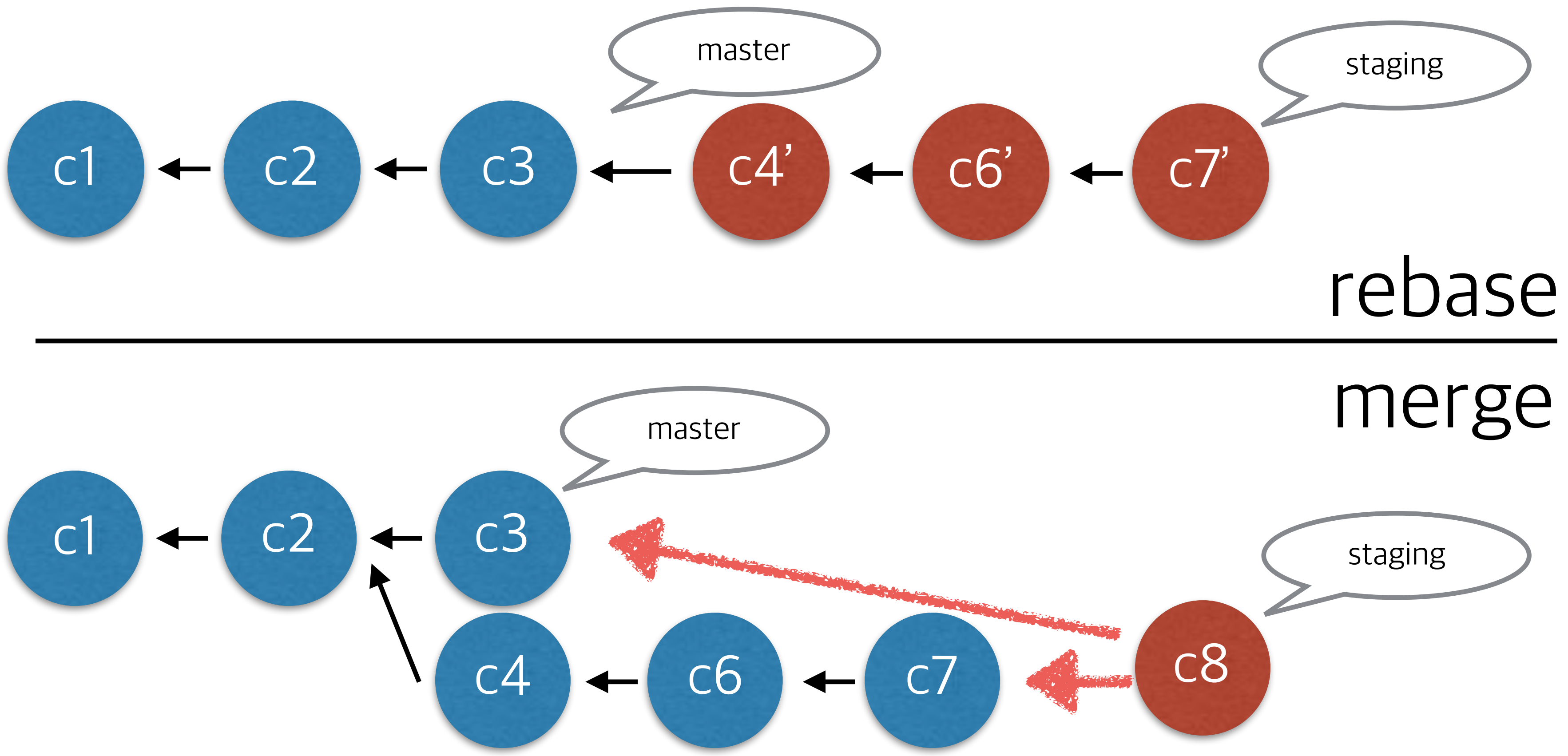
- “Re” + “Base”: Set new base

- Base: common ancestor

```
(staging) git rebase master
```

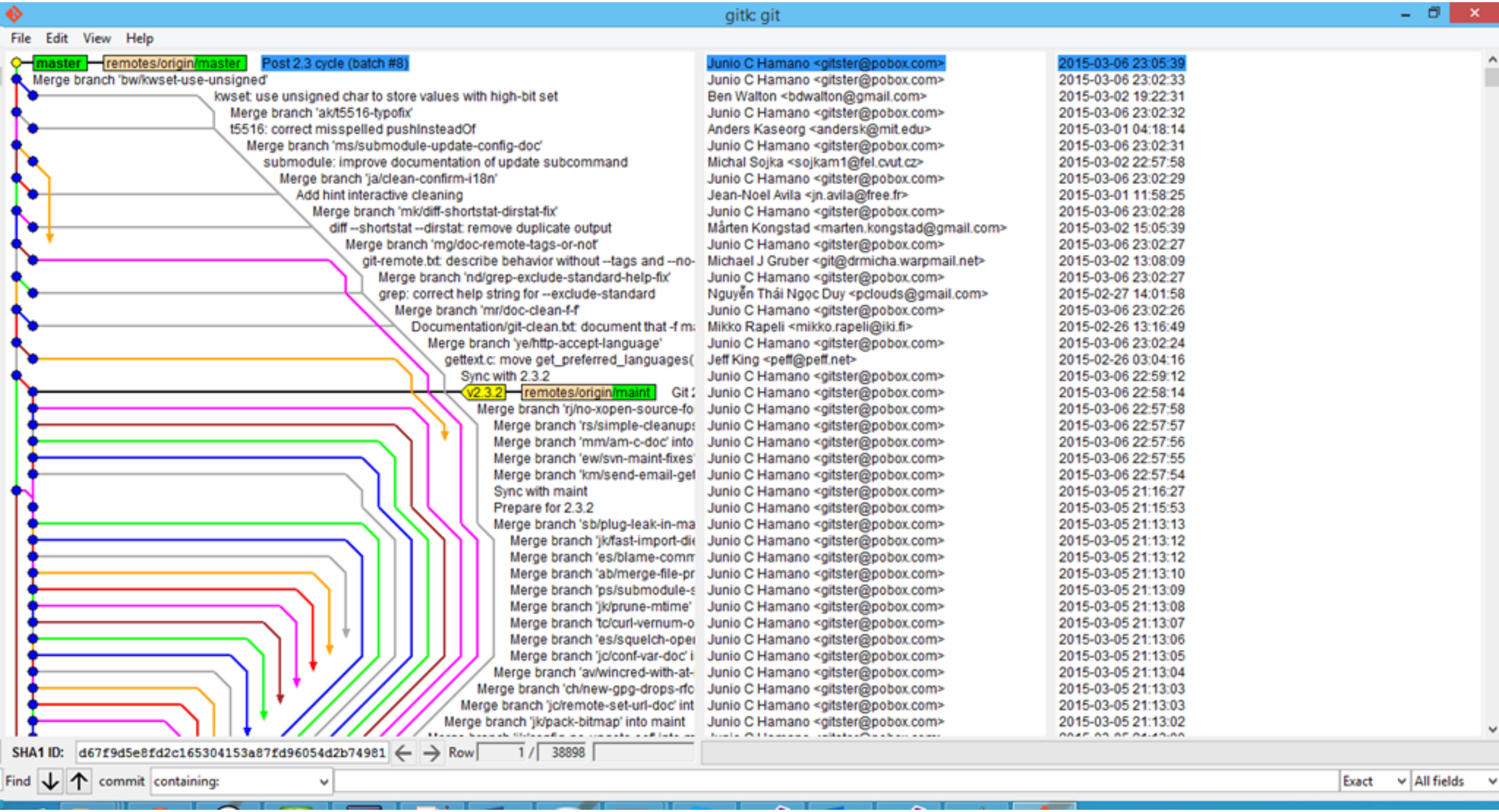


Merge and Rebase



- “Staging” branch after rebase(c7’) and merge(c8) has same content.

- But generally rebase is more recommended, since it’s history is much neater.



master remotes/origin/master Post 2.3 cycle (batch #8)	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:05:39
Merge branch 'bw/kwset-use-unsigned'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:33
kwset: use unsigned char to store values with high-bit set	Ben Walton <bdwalton@gmail.com>	2015-03-02 19:22:31
Merge branch 'ak/t5516-typofix'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:32
t5516: correct misspelled pushInsteadOf	Anders Kaseorg <andersk@mit.edu>	2015-03-01 04:18:14
Merge branch 'ms/submodule-update-config-doc'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:31
submodule: improve documentation of update subcommand	Michal Sojka <sojkam1@fel.cvut.cz>	2015-03-02 22:57:58
Merge branch 'ja/clean-confirm-i18n'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:29
Add hint interactive cleaning	Jean-Noel Avila <jn.avila@free.fr>	2015-03-01 11:58:25
Merge branch 'mk/diff-shortstat-dirstat-fix'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:28
diff --shortstat --dirstat: remove duplicate output	Mårten Kongstad <marten.kongstad@gmail.com>	2015-03-02 15:05:39
Merge branch 'mg/doc-remote-tags-or-not'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:27
git-remote.txt: describe behavior without --tags and --no-	Michael J Gruber <git@drmicha.warpmail.net>	2015-03-02 13:08:09
Merge branch 'nd/grep-exclude-standard-help-fix'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:27
grep: correct help string for --exclude-standard	Nguyễn Thái Ngọc Duy <pclouds@gmail.com>	2015-02-27 14:01:58
Merge branch 'mr/doc-clean-f-f'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:26
Documentation/git-clean.txt: document that -f ma	Mikko Rapeli <mikko.rapeli@iki.fi>	2015-02-26 13:16:49
Merge branch 'ye/http-accept-language'	Junio C Hamano <gitster@pobox.com>	2015-03-06 23:02:24
gettext.c: move get_preferred_languages(	Jeff King <peff@peff.net>	2015-02-26 03:04:16
Sync with 2.3.2	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:59:12
v2.3.2 remotes/origin/maint Git 2	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:58:14
Merge branch 'rj/no-xopen-source-fo	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:57:58
Merge branch 'rs/simple-cleanups	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:57:57
Merge branch 'mm/am-c-doc' into	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:57:56
Merge branch 'ew/svn-maint-fixes'	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:57:55
Merge branch 'km/send-email-get	Junio C Hamano <gitster@pobox.com>	2015-03-06 22:57:54
Sync with maint	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:16:27
Prepare for 2.3.2	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:15:53
Merge branch 'sb/plug-leak-in-ma	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:13
Merge branch 'jk/fast-import-die	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:12
Merge branch 'es/blame-comm	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:12
Merge branch 'ab/merge-file-pr	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:10
Merge branch 'ps/submodule-s	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:09
Merge branch 'jk/prune-mtime'	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:08
Merge branch 'tc/curl-vernum-o	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:07
Merge branch 'es/squelch-oper	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:06
Merge branch 'jc/conf-var-doc' i	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:05
Merge branch 'av/wincred-with-at-	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:04
Merge branch 'ch/new-gpg-drops-rfc	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:03
Merge branch 'jc/remote-set-url-doc' int	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:03
Merge branch 'jk/pack-bitmap' into maint	Junio C Hamano <gitster@pobox.com>	2015-03-05 21:13:02

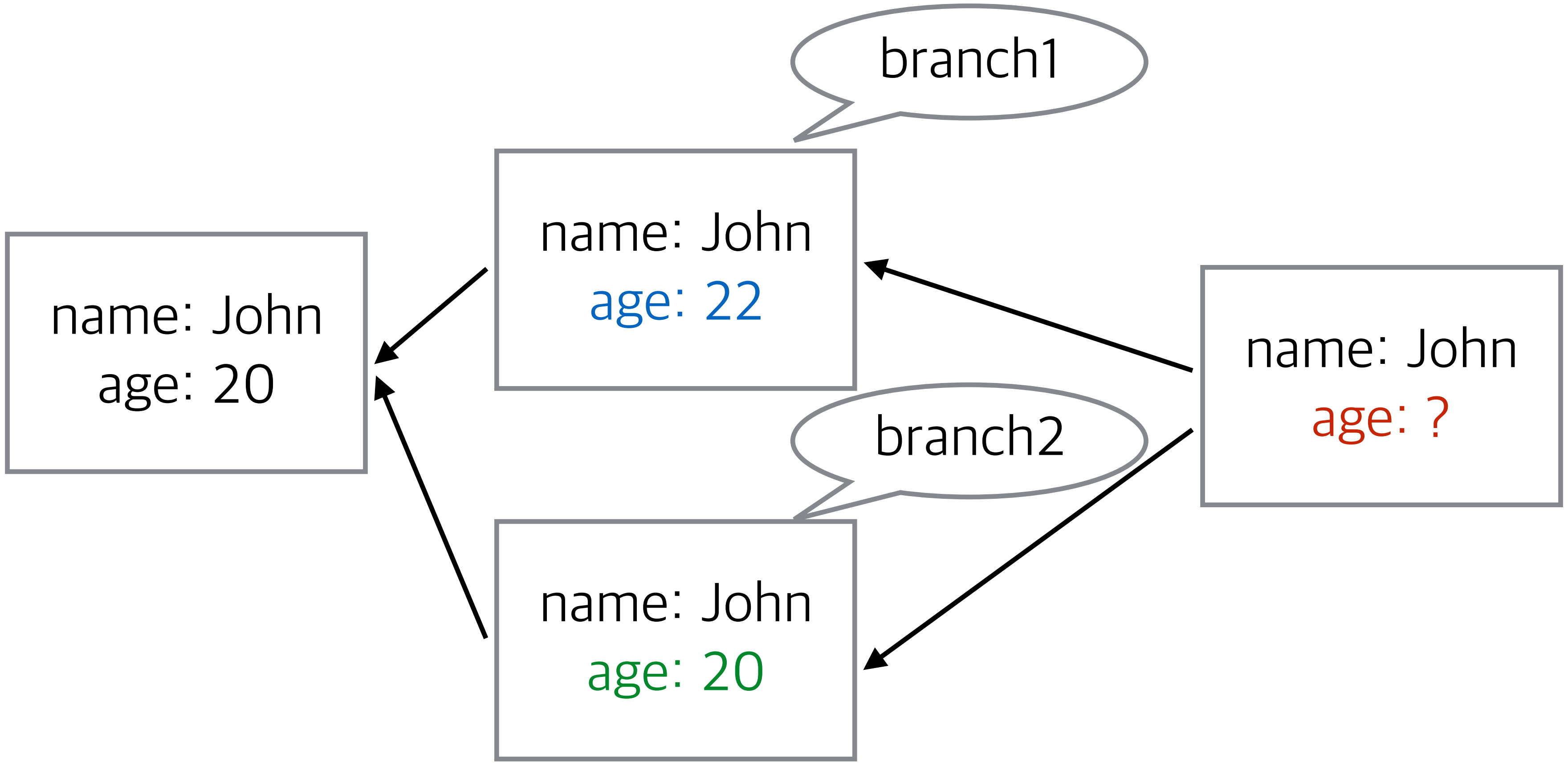
SHA1 ID: d67f9d5e8fd2c165304153a87fd96054d2b74981 Row 1 / 38898

Find commit containing: Exact All fields

http://www.tugberkugurlu.com/archive/resistance-against-london-tube-map-commit-history-a-k-a--git-merge-hell

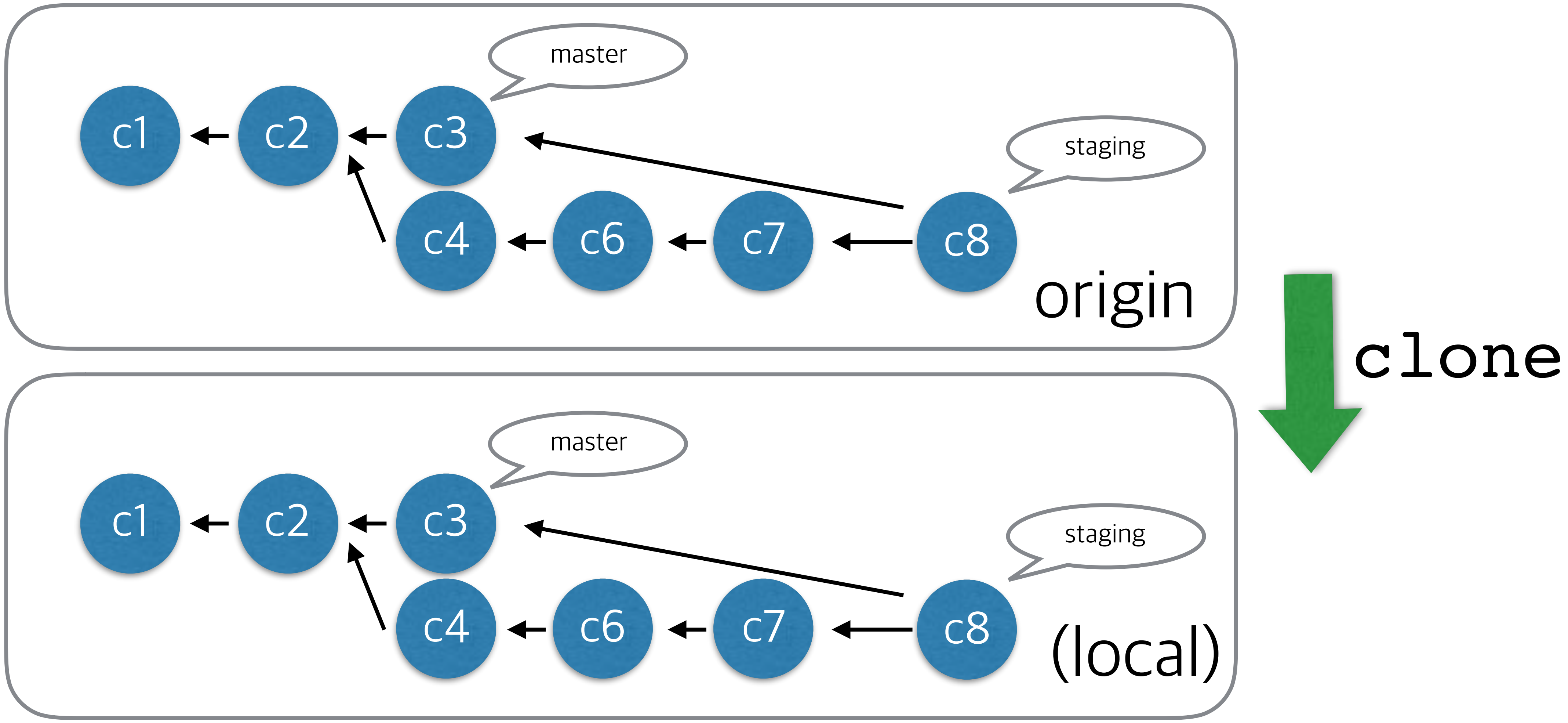
Conflict

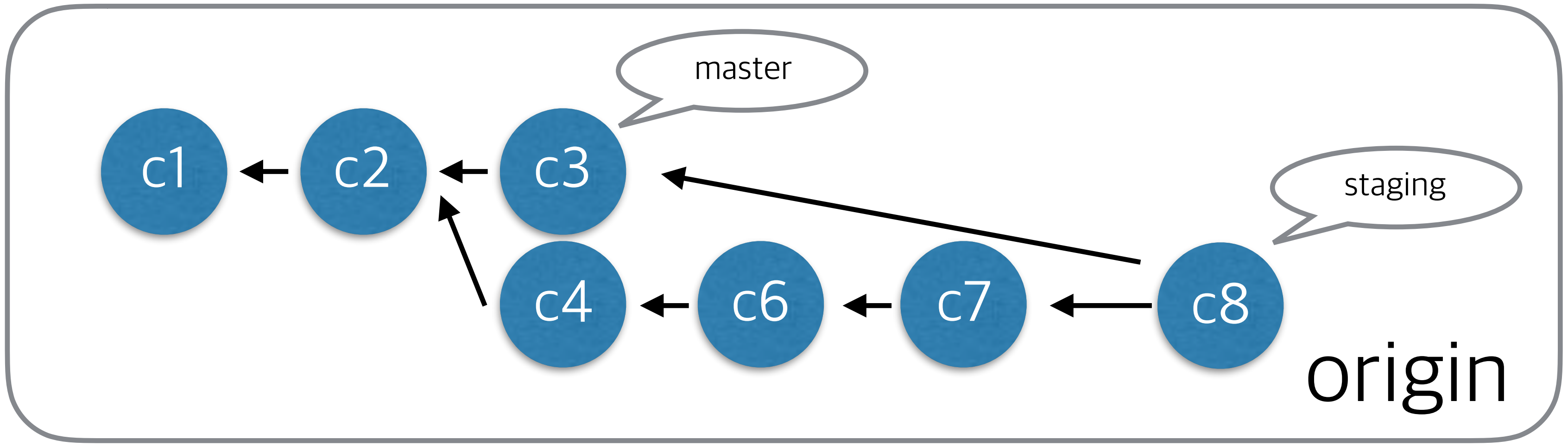
- Don’t get scared
- Conflict happens when Git failed to merge
- Usually when two commits changed same part of a file
- Developer should choose which one to follow

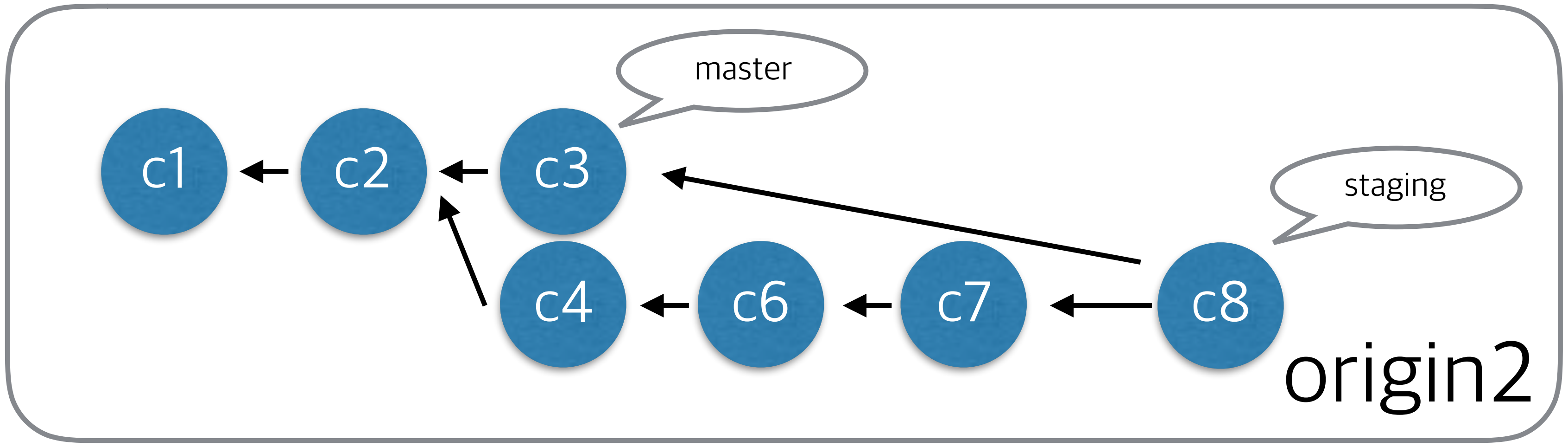


Remote Repository

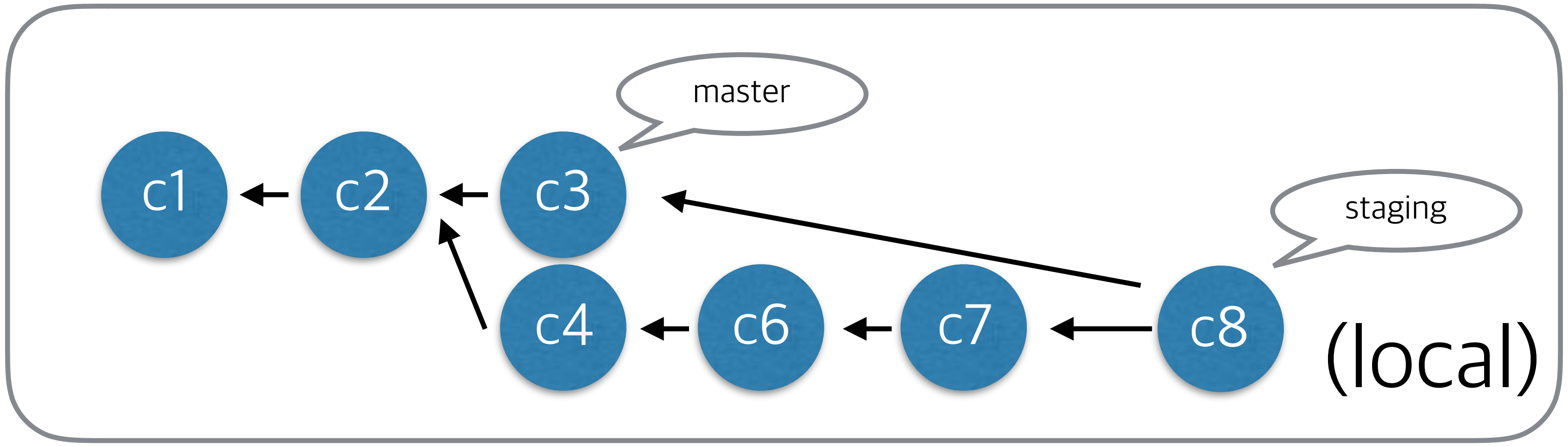
- Git is a distributed version control system
  - Repository can be at server or developer's system
  - But they are equal repositories





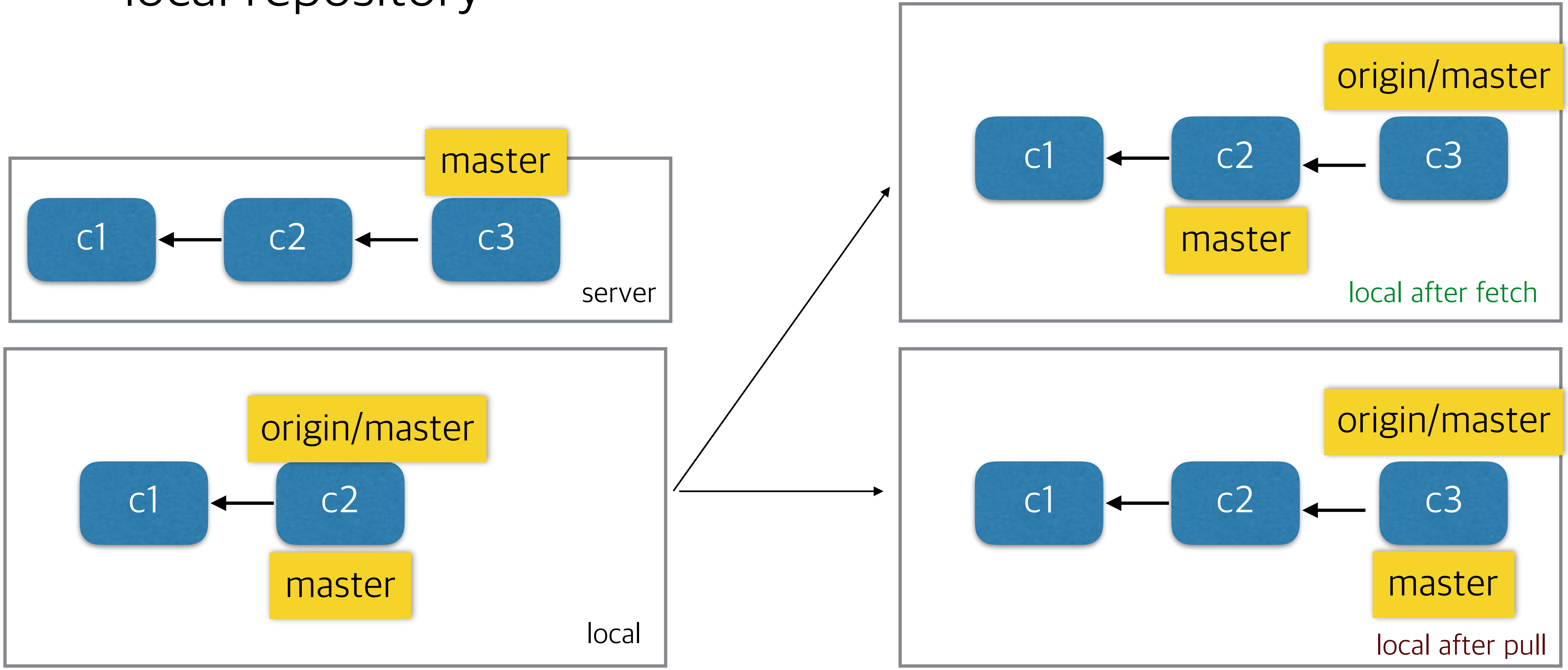


We can have multiple remote repositories.
`origin` is just one of them.



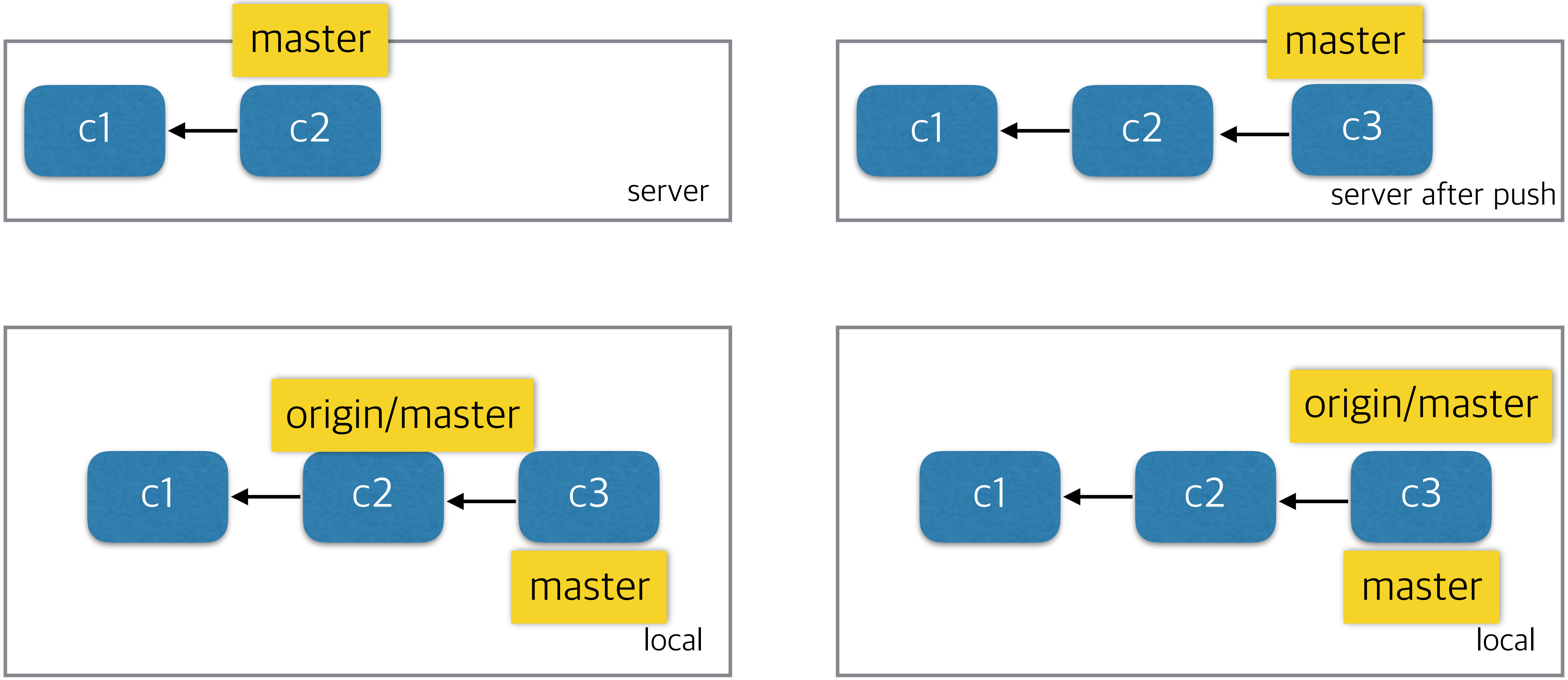
Fetch, Pull

- Fetch: Just download all the content from remote repository
- Pull: Fetch and apply commits from remote repository to local repository



Push

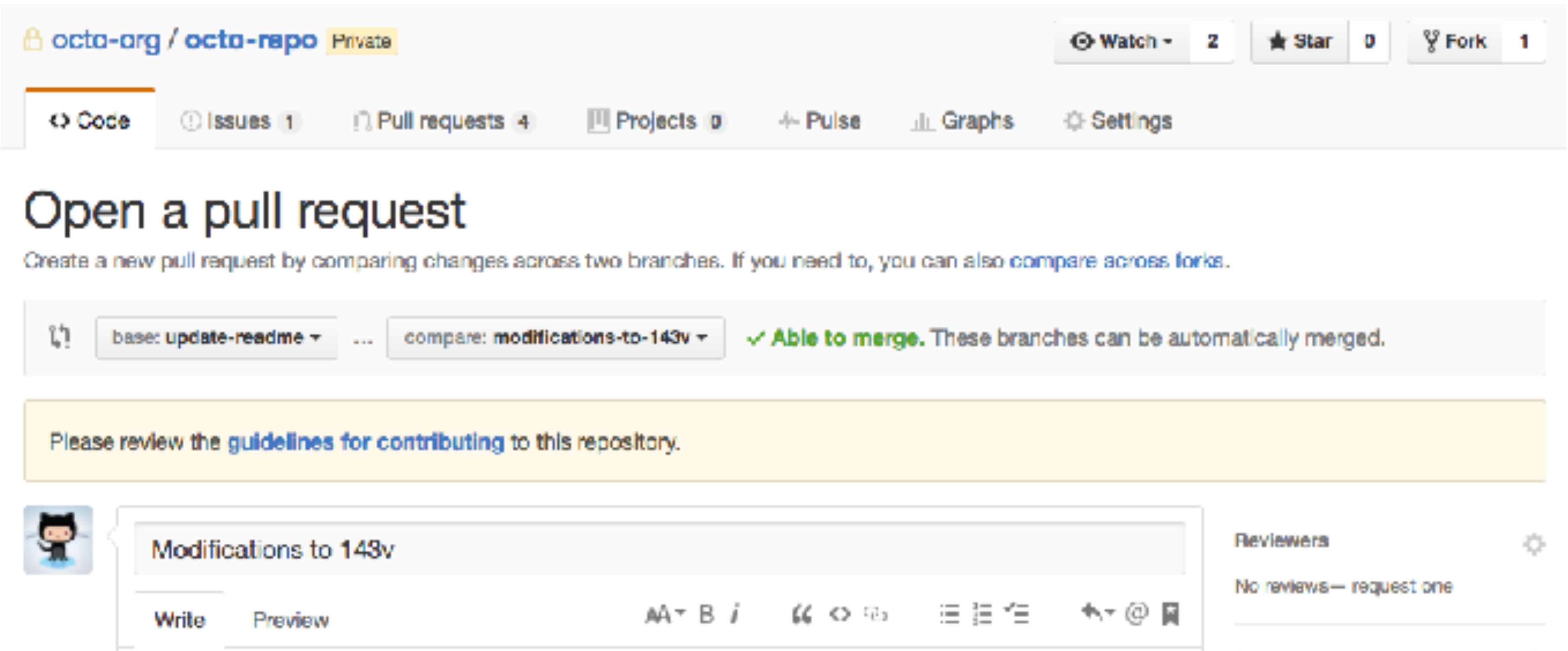
- Apply local commits to remote repo



Pull Request

- What if a developer doesn't have permission to push? (Or we cannot give permission to anyone?)

- Asking to apply changes to a branch.



Open a pull request

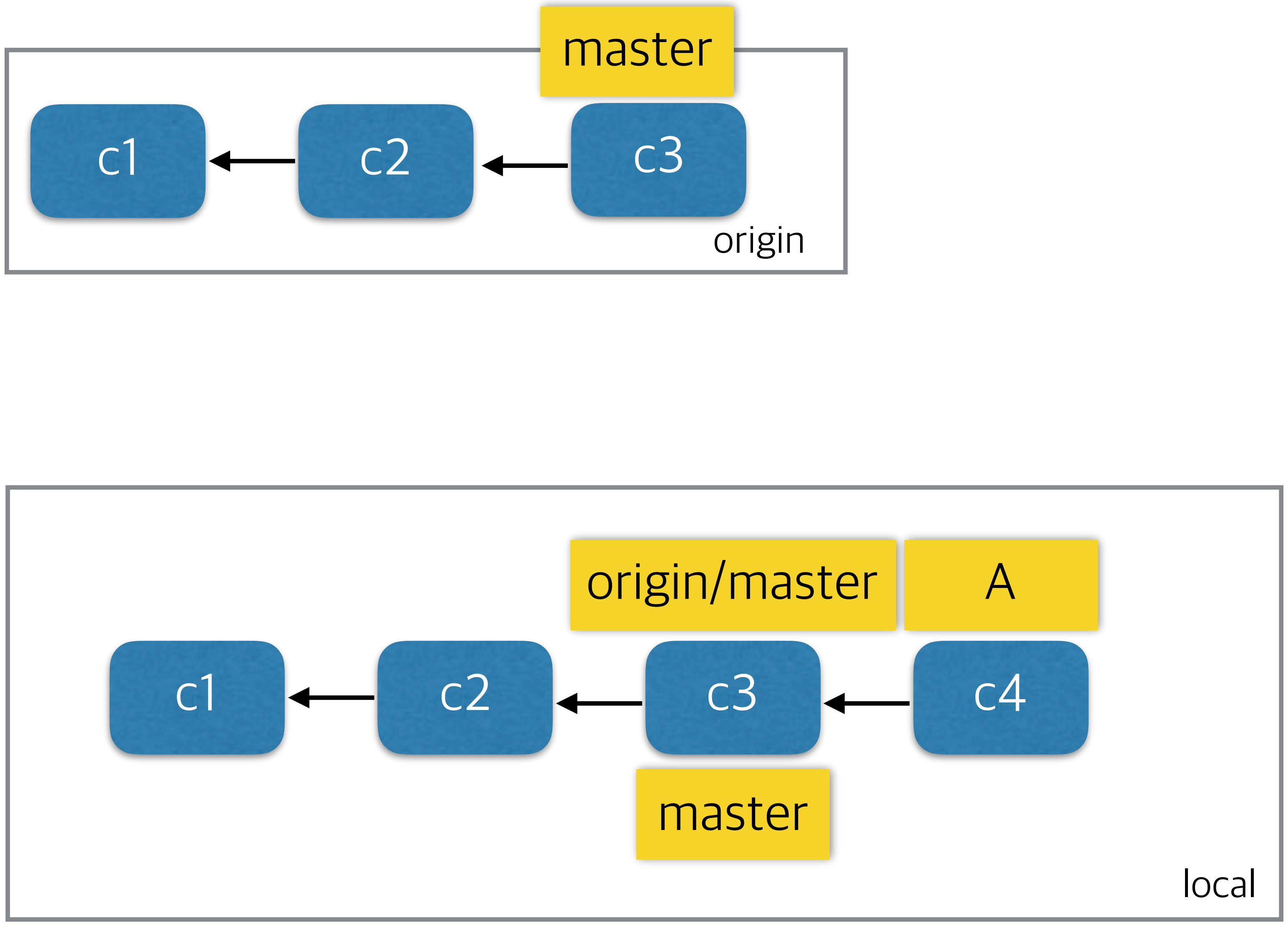
Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: update-readme ... compare: modifications-to-143v

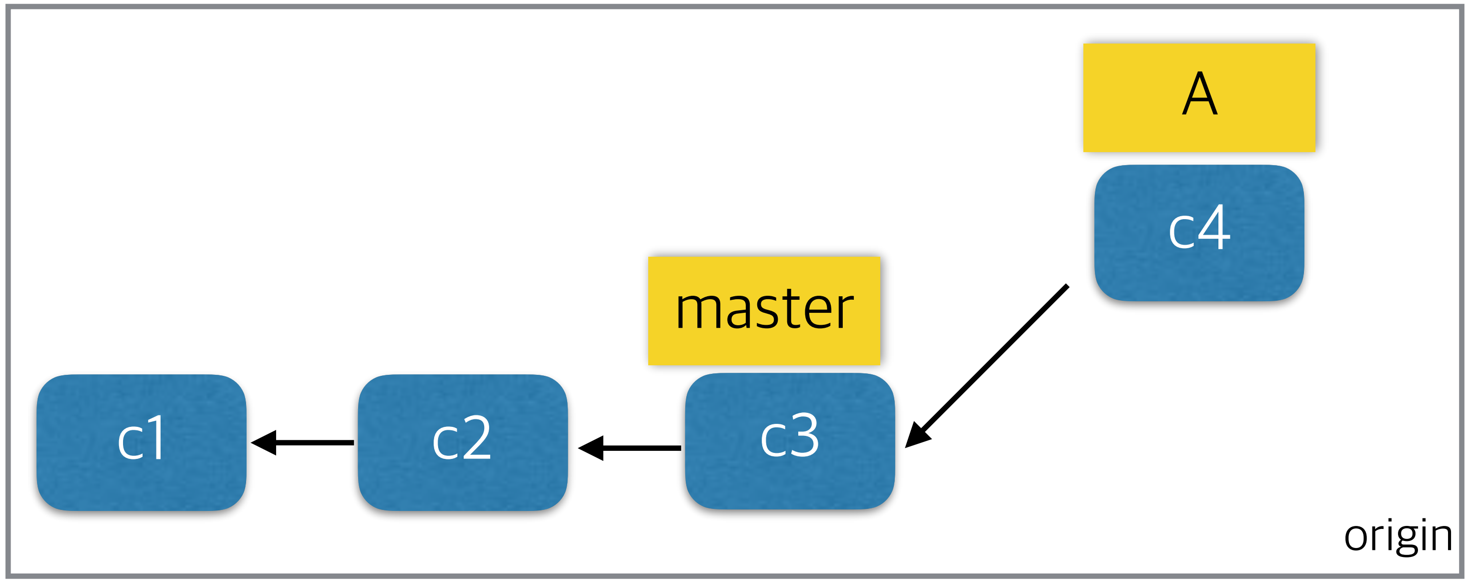
✓ Able to merge. These branches can be automatically merged.

Please review the guidelines for contributing to this repository.

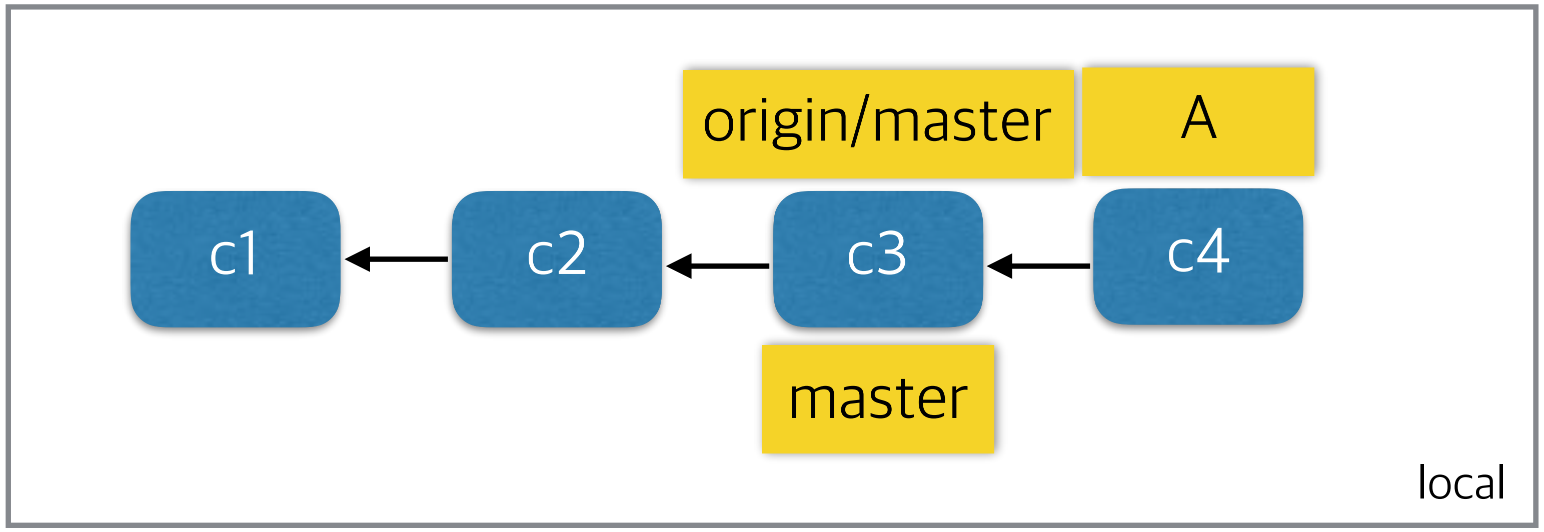
Modifications to 143v

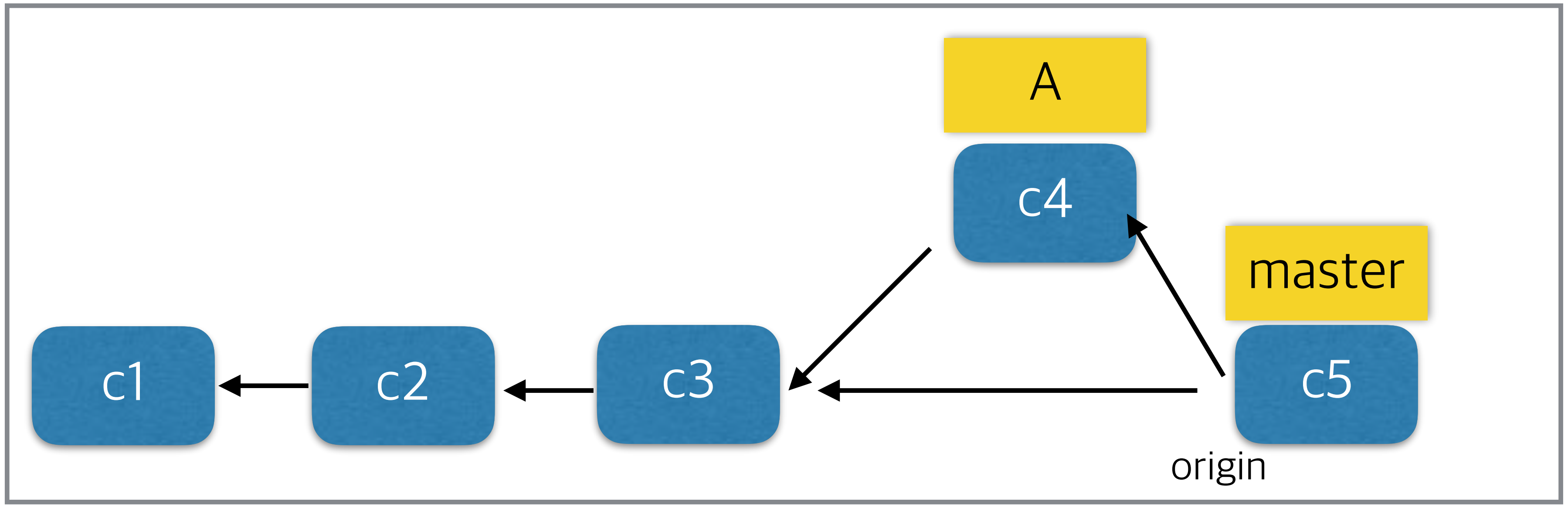


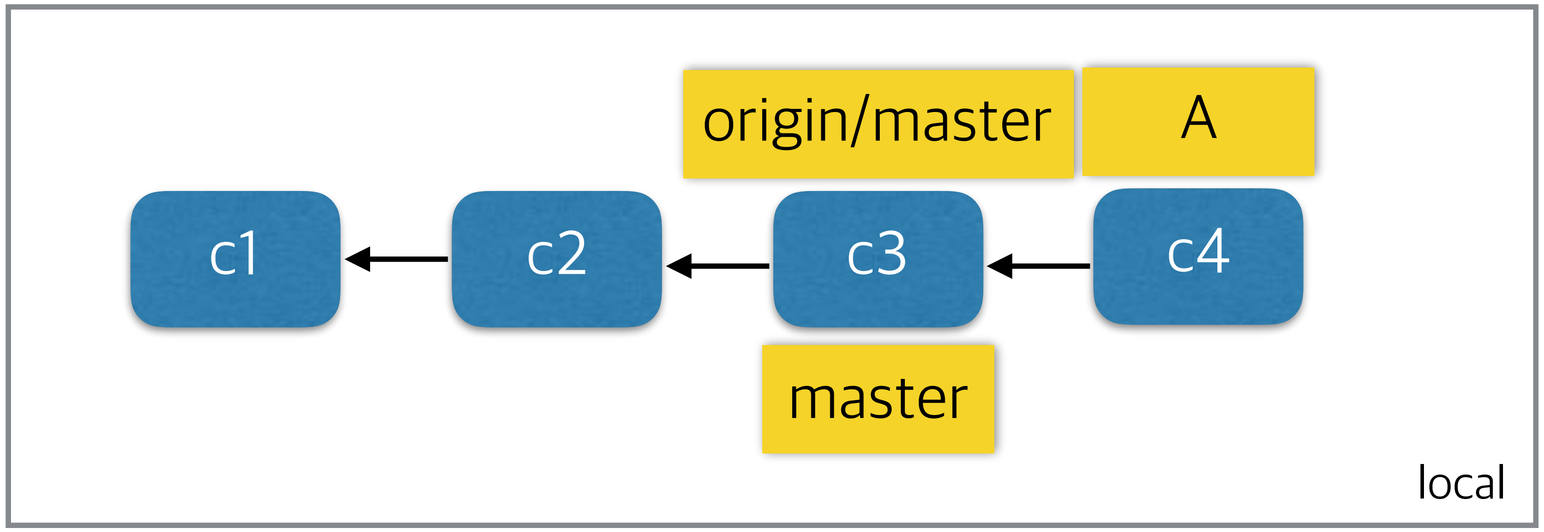
Has a new local branch A,
but cannot push it to master



Upload new branch A,
and make **PULL REQUEST**



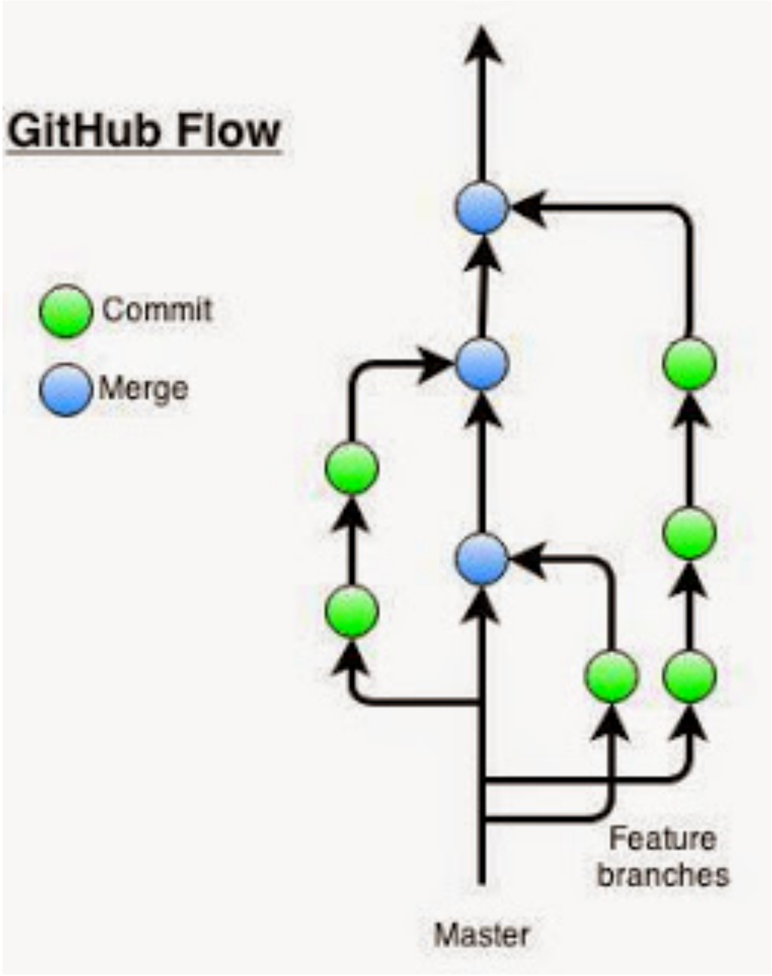


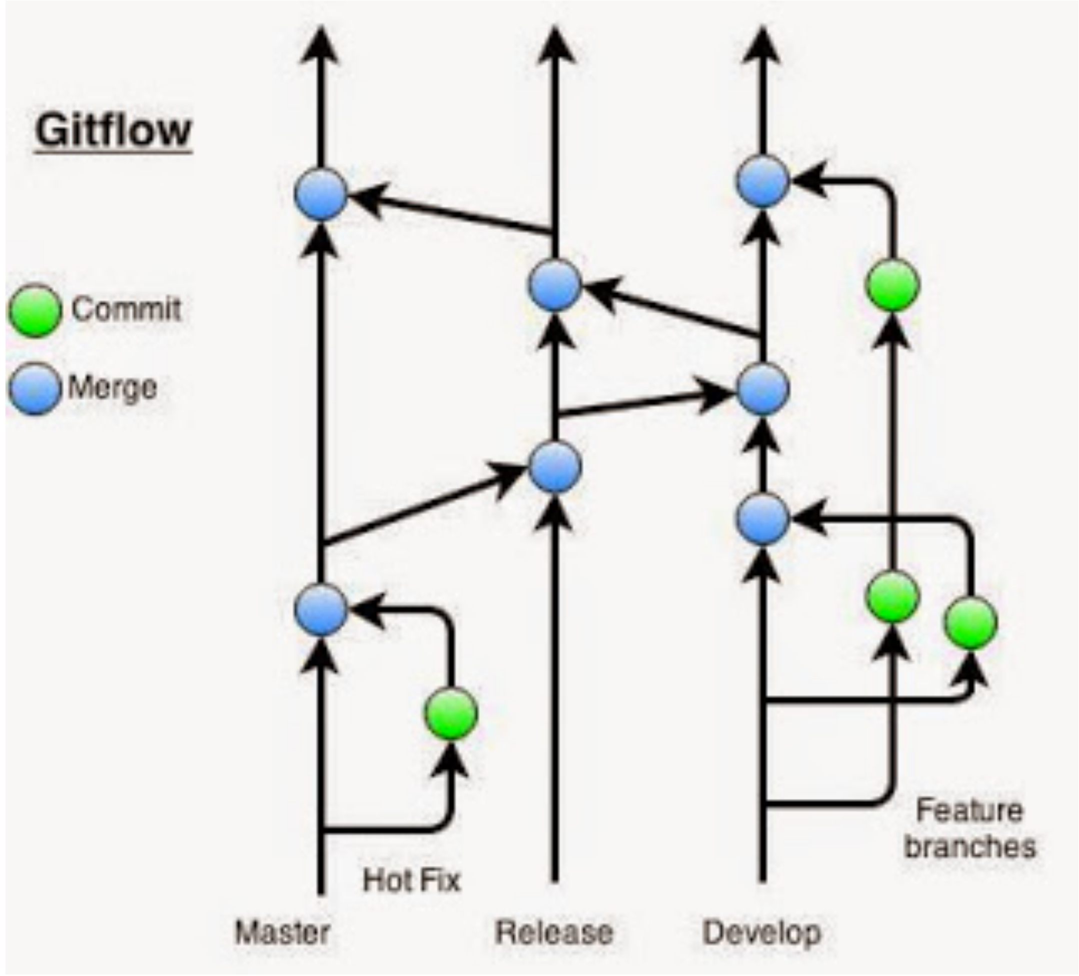


After PR is accepted, new branch is merged to master

Working with others

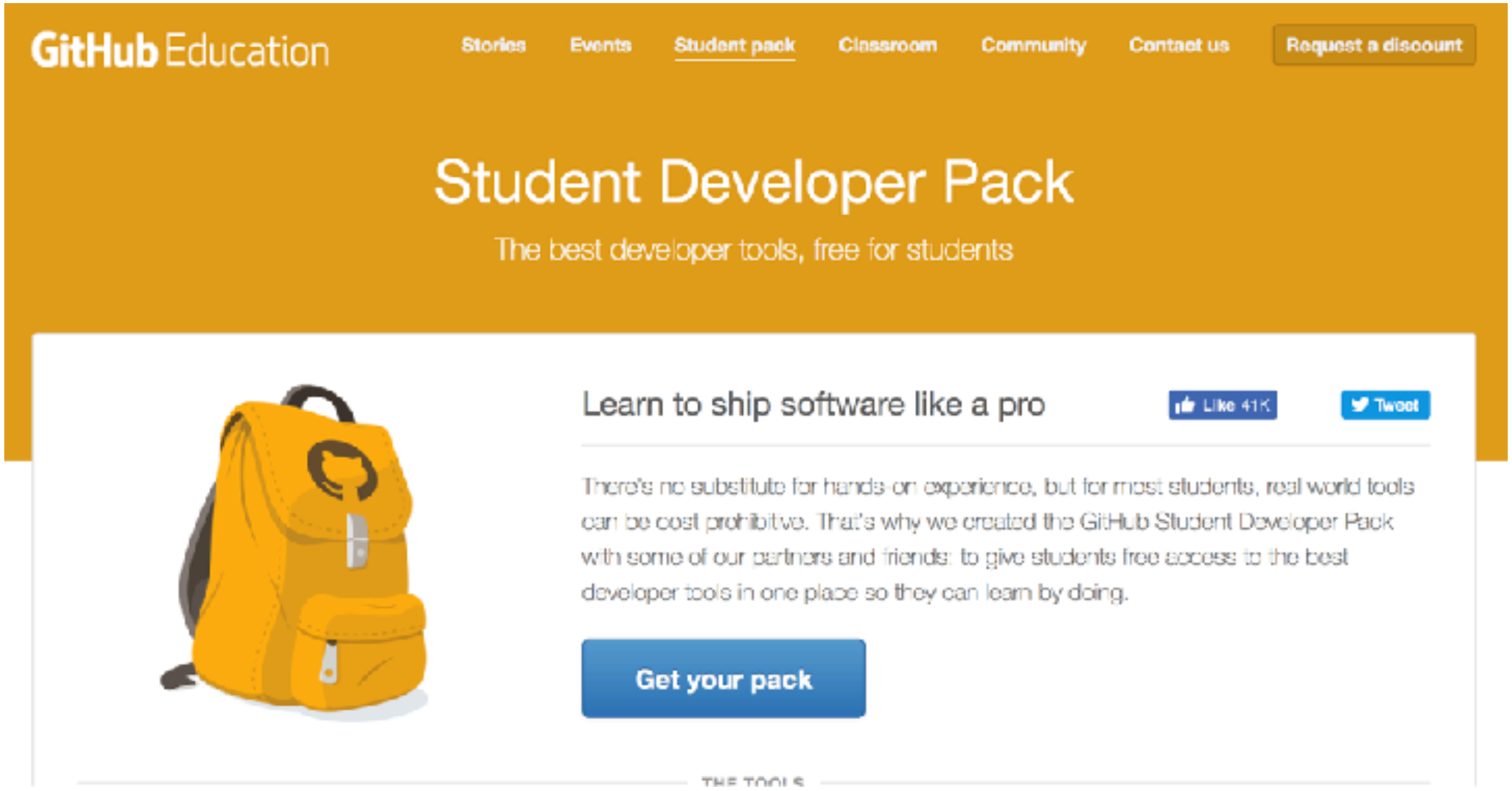
- There are many workflows on the web.
- But in every workflow, “master” branch is production-ready.





http://blog.endpoint.com/2014/05/git-workflows-that-work.html

Since you are a student



Student Developer Pack

The best developer tools, free for students

Learn to ship software like a pro

There's no substitute for hands-on experience, but for most students, real world tools can be cost prohibitive. That's why we created the GitHub Student Developer Pack with some of our partners and friends: to give students free access to the best developer tools in one place so they can learn by doing.

Free unlimited private repositories + alpha

Recommended Materials

- Pro Git online book (https://git-scm.com/book/en/v2)
- https://try.github.io
  (Easy interactive tutorial)
- http://learngitbranching.js.org/
  (Difficult interactive tutorial)
- (Korean) Effective Git
  (from NDC 2016. Includes so many useful tips)