Refactoring: Git

CPSC 501: Advanced Programming Techniques
Fall 2020

Jonathan Hudson, Ph.D
Instructor
Department of Computer Science
University of Calgary

Tuesday, August 4, 2020

UNIVERSITY OF CALGARY

Not an acronym

Not an acronym

- (from the source code read-me)
- "git" can mean anything, depending on your mood.
  - random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
  - stupid. contemptible and despicable. simple. Take your pick from the dictionary of slang.
  - "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
  - "goddamn idiotic truckload of sh*t": when it breaks

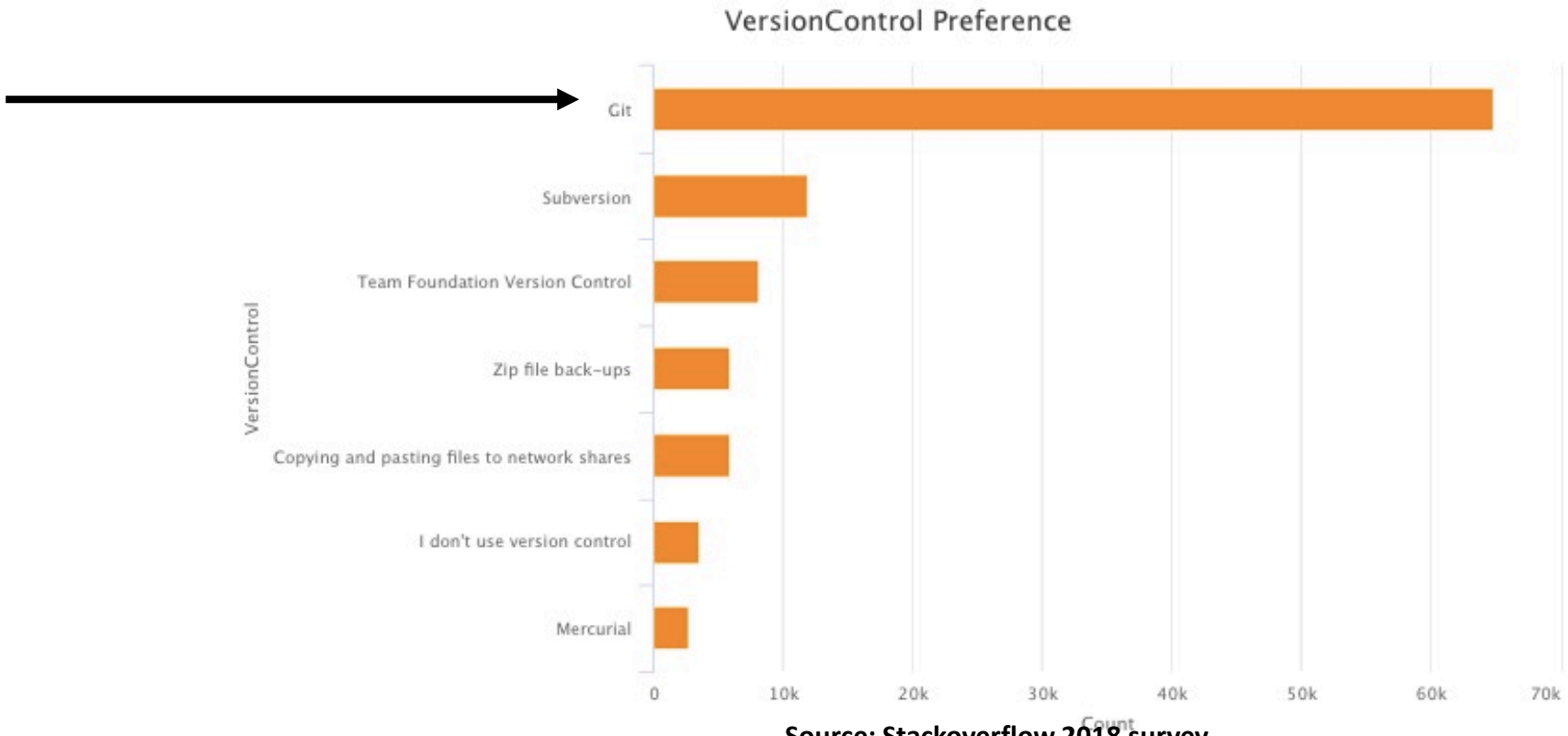
The Rise of Git

- *Git* is the most popular implementation of a distributed version control system.
- Development started in 2005 by Linus Torvalds.
  - Linux kernel source host dispute with BitKeeper
  - Same reason resulted in another DVCS -> Mercurial
- It is used by many popular open source projects as well as many commercial organizations.

1. Take Concurrent Versions System (CVS) as an example of what not to do; if in doubt, make the exact opposite decision.
2. Support a distributed, BitKeeper-like workflow. (‘He’s dead Jim’ -> BitKeeper)
3. Include very strong safeguards against corruption, either accidental or malicious.

Why Git?

- Git's the most popular version control system in the industry.
- Most popular VCS are similar to Git



Source: Stackoverflow 2018 survey

UNIVERSITY OF CALGARY

Why Git?

- Git is distributed
- i.e. there is generally are remote repo (like the single svn one) and a local repo on your own machine
  - SVN required repo to be only local, or only remote
  - GIT lets each developer have their own version of repo
  - Each developer can make changes and make commits to own repo and periodically push/pull from remote to bring together development
  - Frees programmer, code on a plane and still do multiple local commits

Source: Stackoverflow 2018 survey

This is how you do it

Git: *New* Version Control Terminology

1. SHA
2. Staging Area/Index

Git: *New* Version Control Terminology

1. SHA
2. Staging Area/Index

SHA

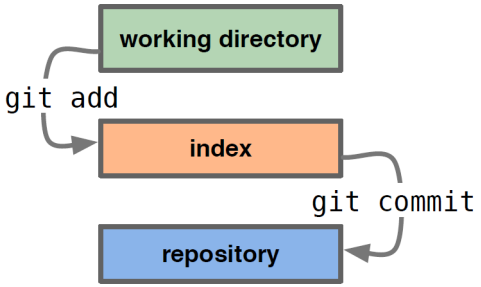
- A SHA is basically an ID number for each commit.
- Ex. E2adf8ae3e2e4ed40add75cc44cf9d0a869afeb6
- Instead of version numbering (SVN)

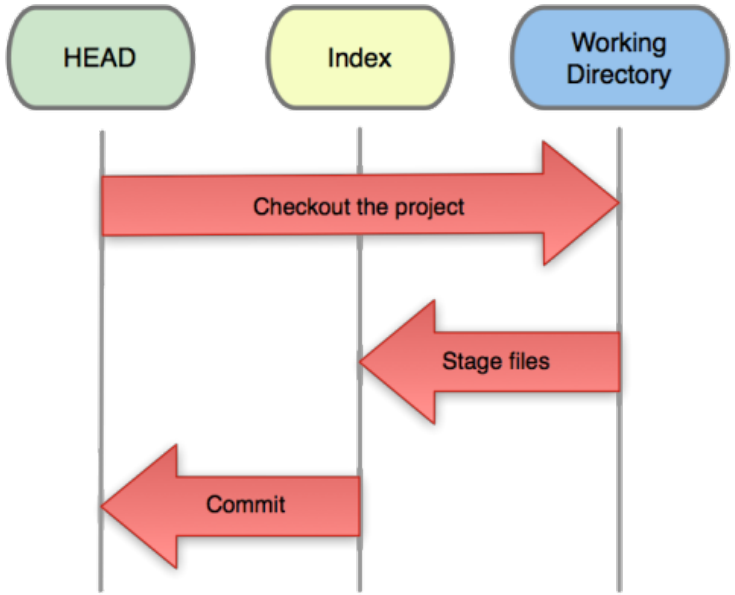
Staging Area

- You can think of the staging area as a prep table where Git will take the next commit.
- Files on the Staging Index are ready to be added to the repository.

Git: Getting Started

- Three trees of Git
  - The HEAD
    - last commit snapshot, next parent
  - Index
    - Proposed next commit snapshot
  - Working directory
    - Sandbox





Git: Basic Commands

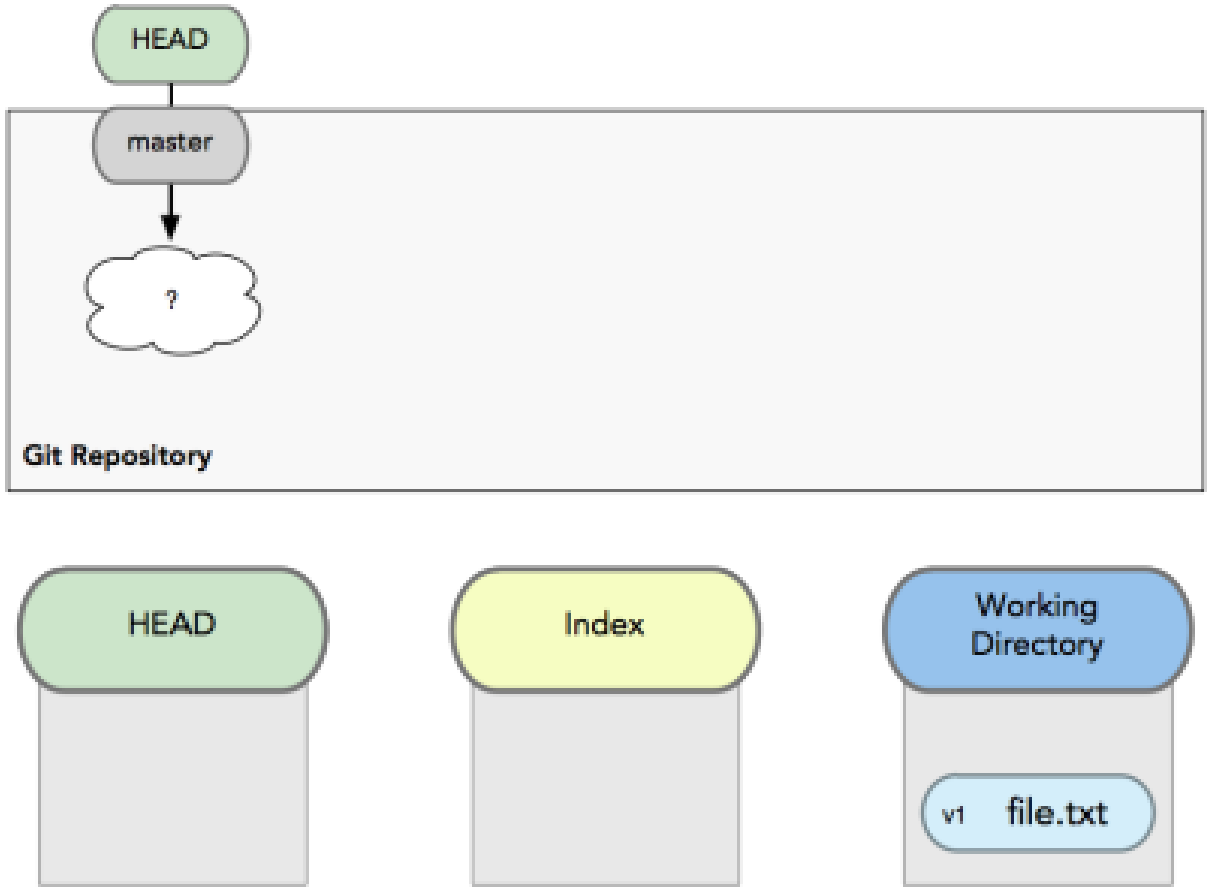
- **git init** – *Initialize a Git repository/working directory*
  - ***git init NAME***
- **git status** – *Status of your working directory*
  - ***git status***
- **git add <filename>** or **git add .** *(for all files in your working directory)*
- **git commit** – *Stash changes in your working directory*
- **git log** – *View your commit history*
- **git clone** – *Create an identical copy*

Git: A Basic Workflow

- A basic workflow
  - Init a repo (or clone an existing one)
  - Edit files
  - Stage the changes
  - Review your changes
  - Commit the changes

Git: A Basic Workflow

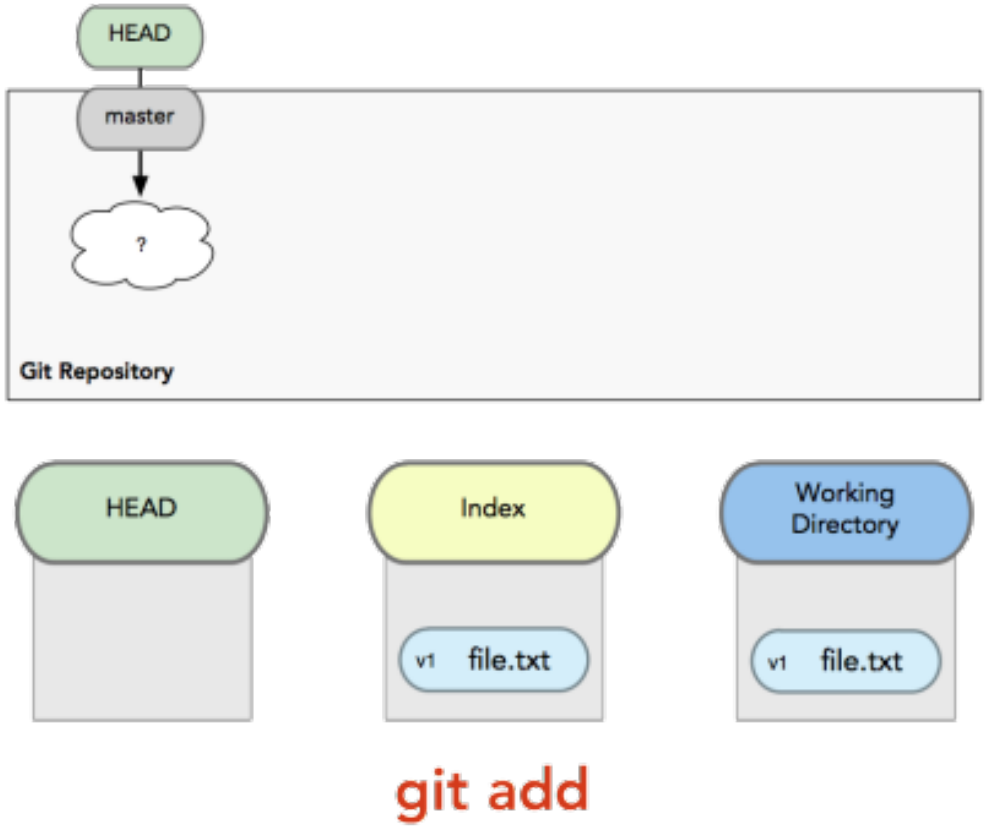
- A basic workflow
  - Edit files
  - Stage the changes
  - Review your changes
  - Commit the changes



Git: A Basic Workflow

- A basic workflow
  - Edit files
  - Stage the changes
  - Review your changes
  - Commit the changes

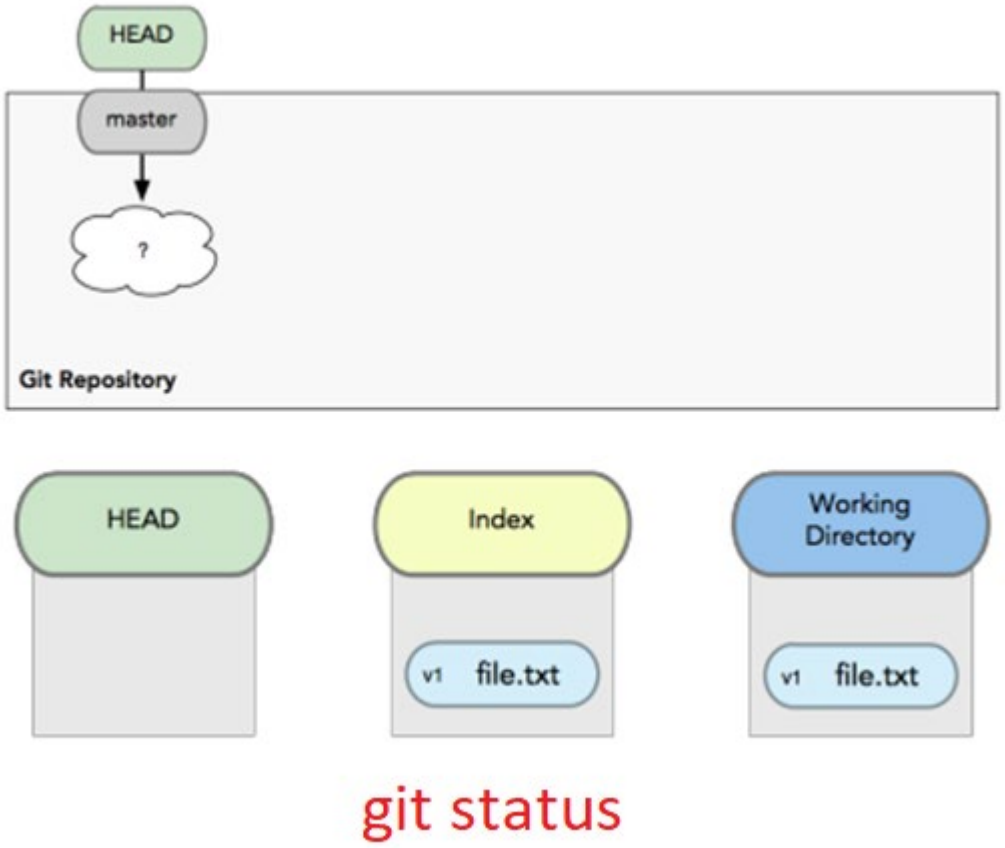
- Git add filename



Git: A Basic Workflow

- A basic workflow
  - Edit files
  - Stage the changes
  - Review your changes
  - Commit the changes

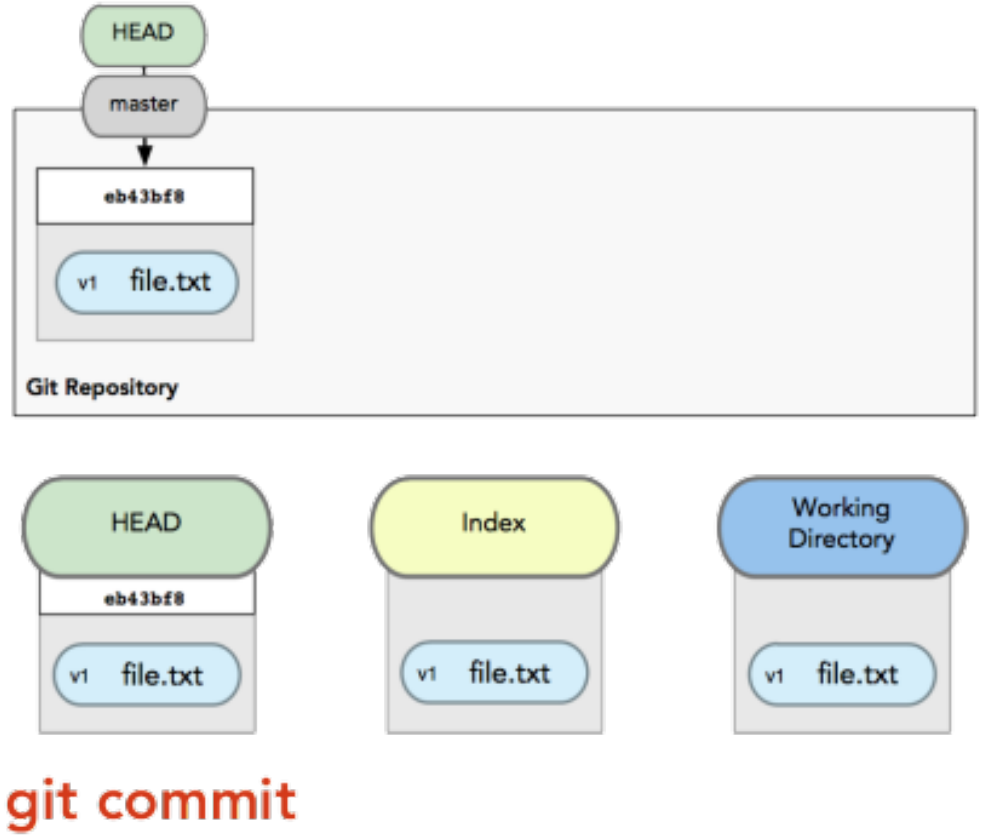
- Git status



Git: A Basic Workflow

- A basic workflow
  - Edit files
  - Stage the changes
  - Review your changes
  - Commit the changes

- Git commit



Git: Informational

- View changes
  - **git diff**
    - Show the difference between working directory and staged
  - **git diff --cached**
    - Show the difference between staged and the HEAD

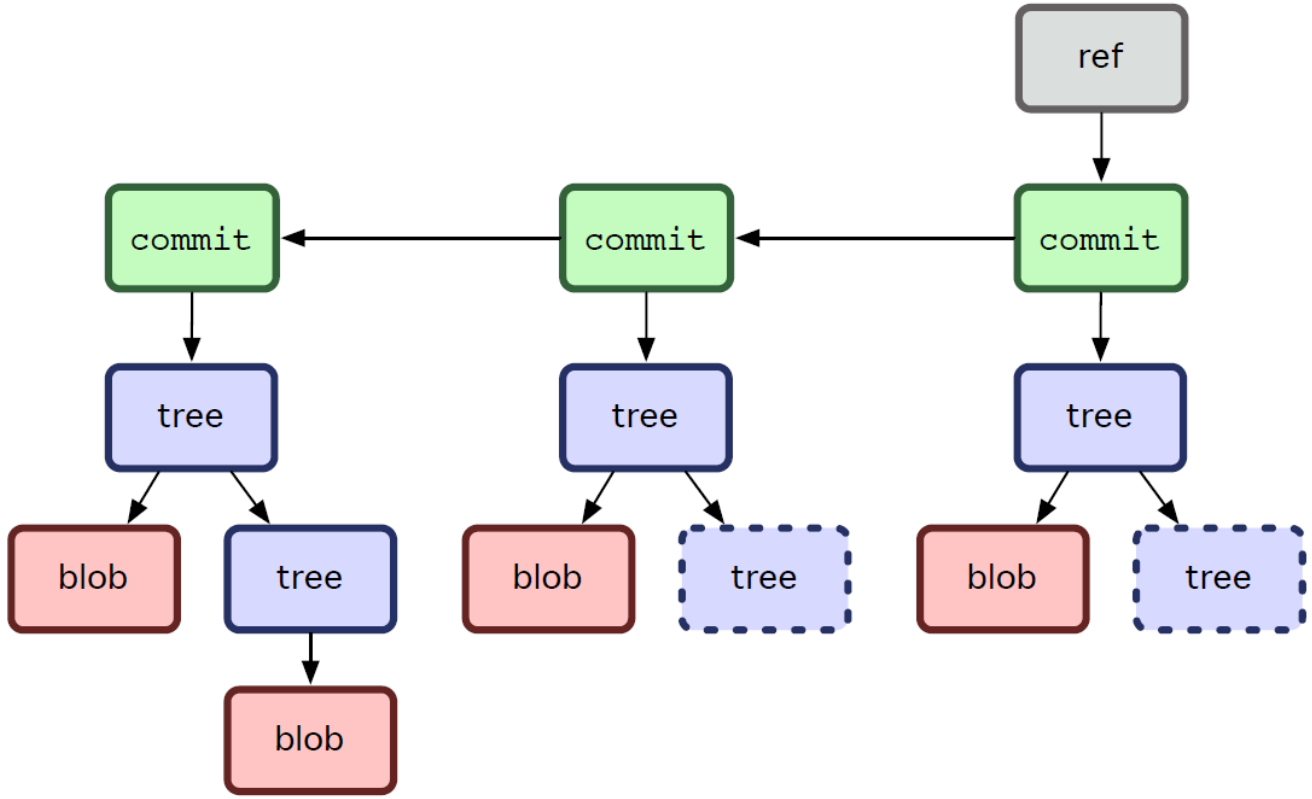
- View history
  - **git log**

Git: Revert

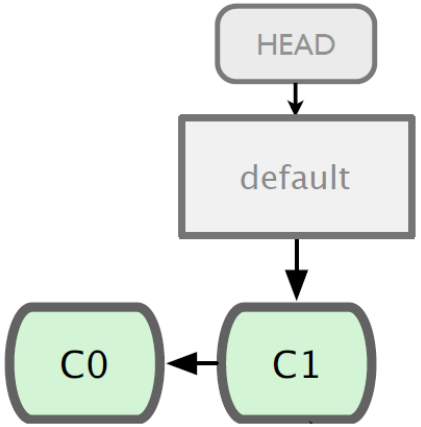
- Revert changes (Get back to a previous version)
  - **git checkout commit_hash**

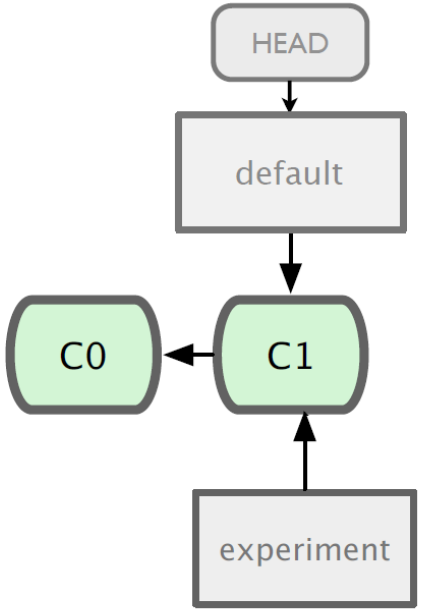
Git: Commit Tree

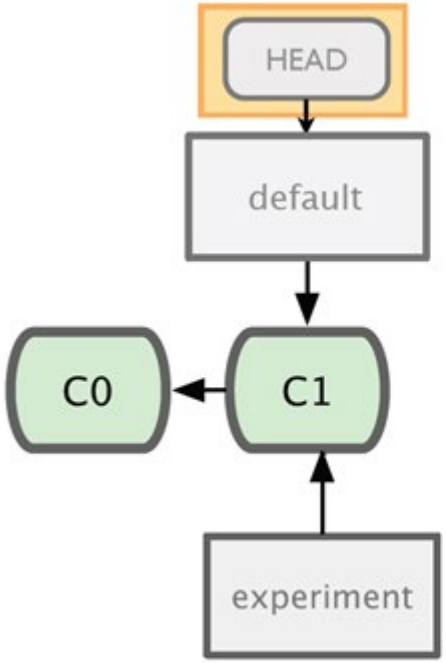
- Git sees commit this way…
- Branch annotates which commit we are working on



Git: Branching



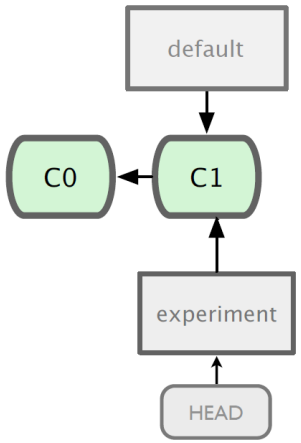




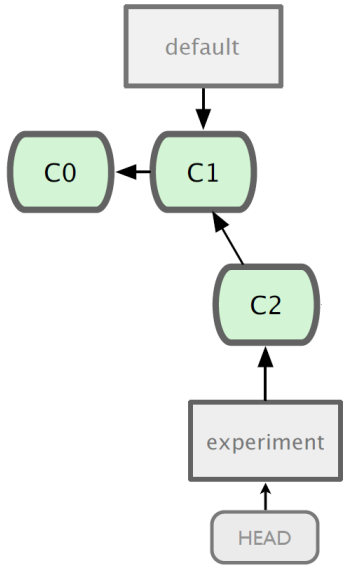

```
$ git branch
* default
  experiment
```

git branch experiment

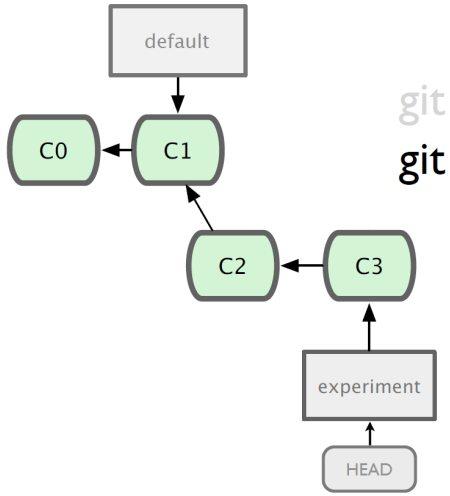
Git: Branching



git checkout experiment



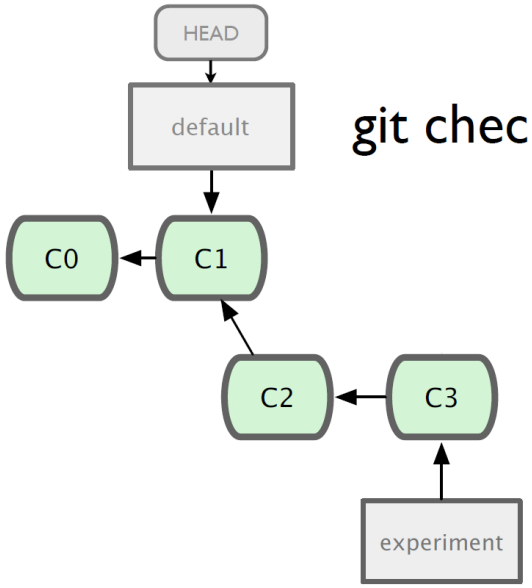
git commit



git commit

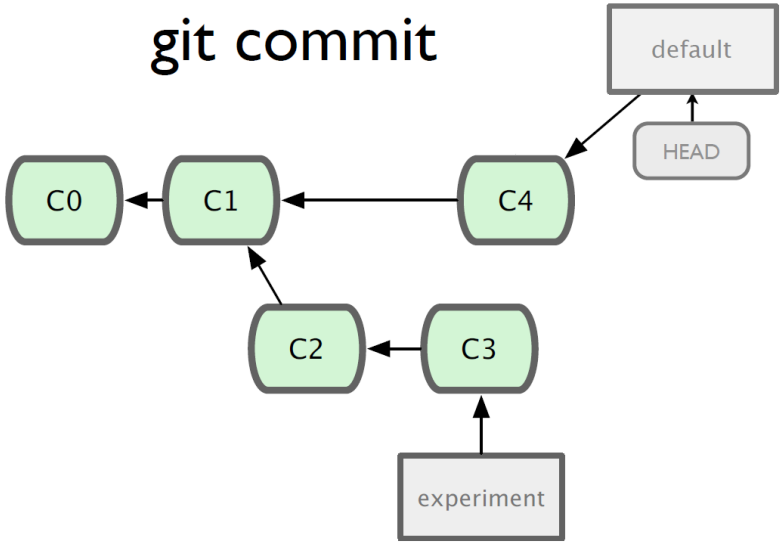
git commit

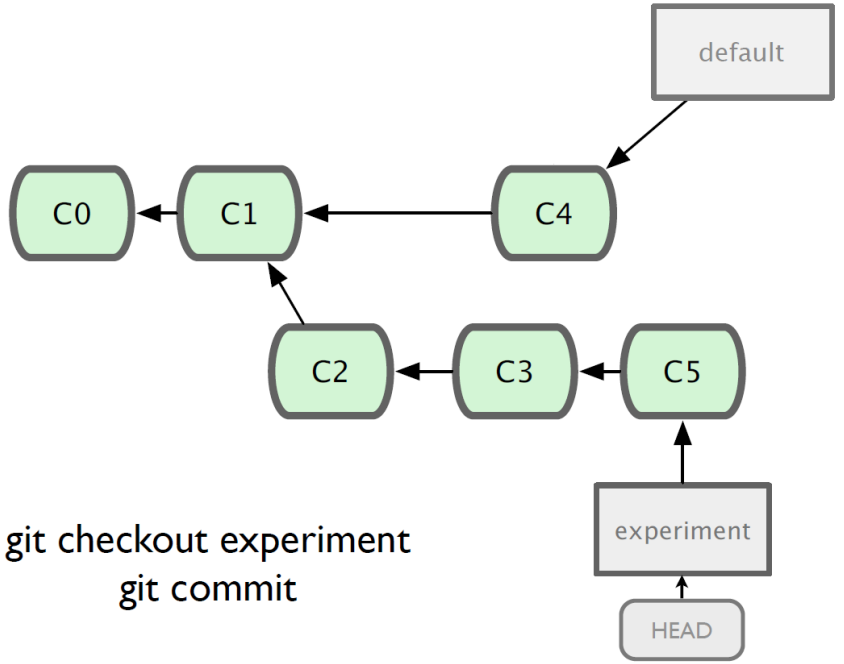
Git: Branching



git checkout default

git commit

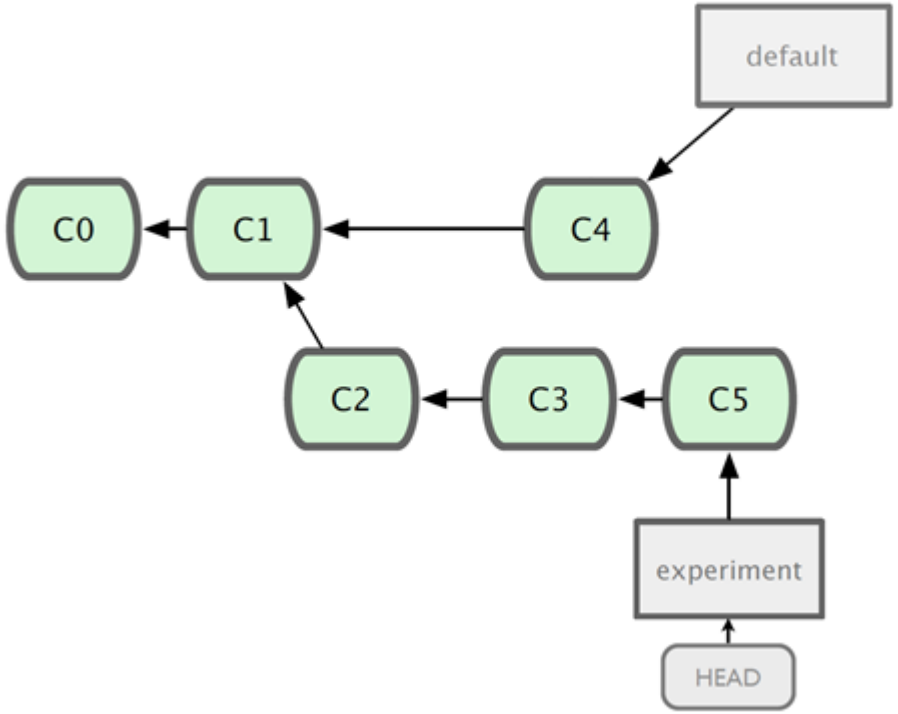




git checkout experiment
git commit

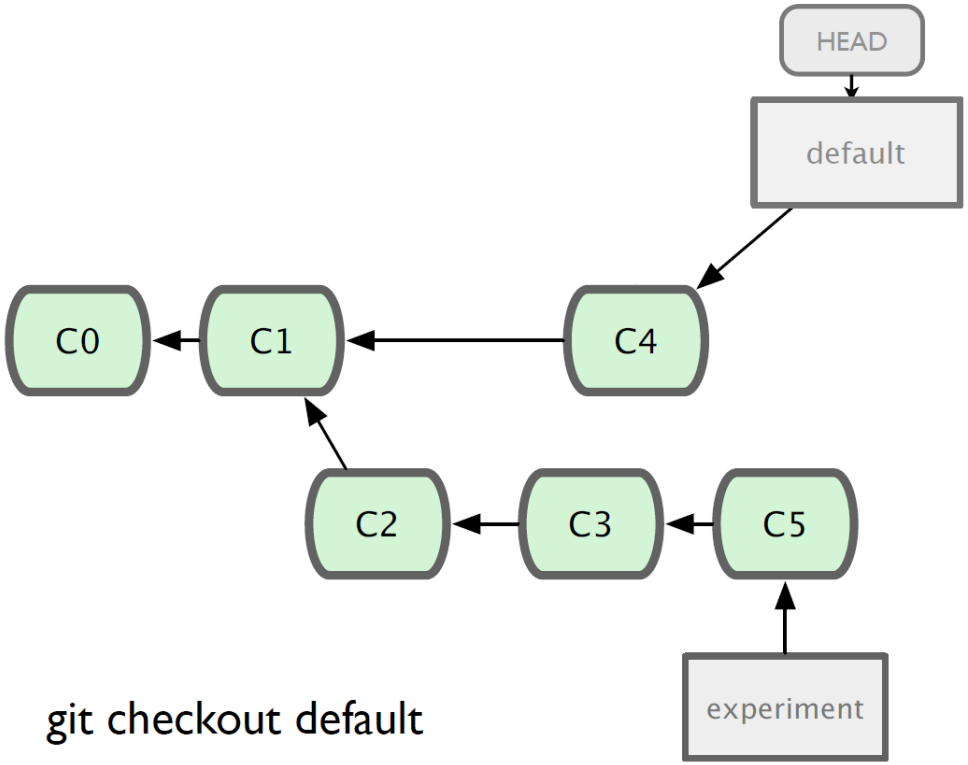
Git: Merging

- What do we do with this mess?
  - Merge them

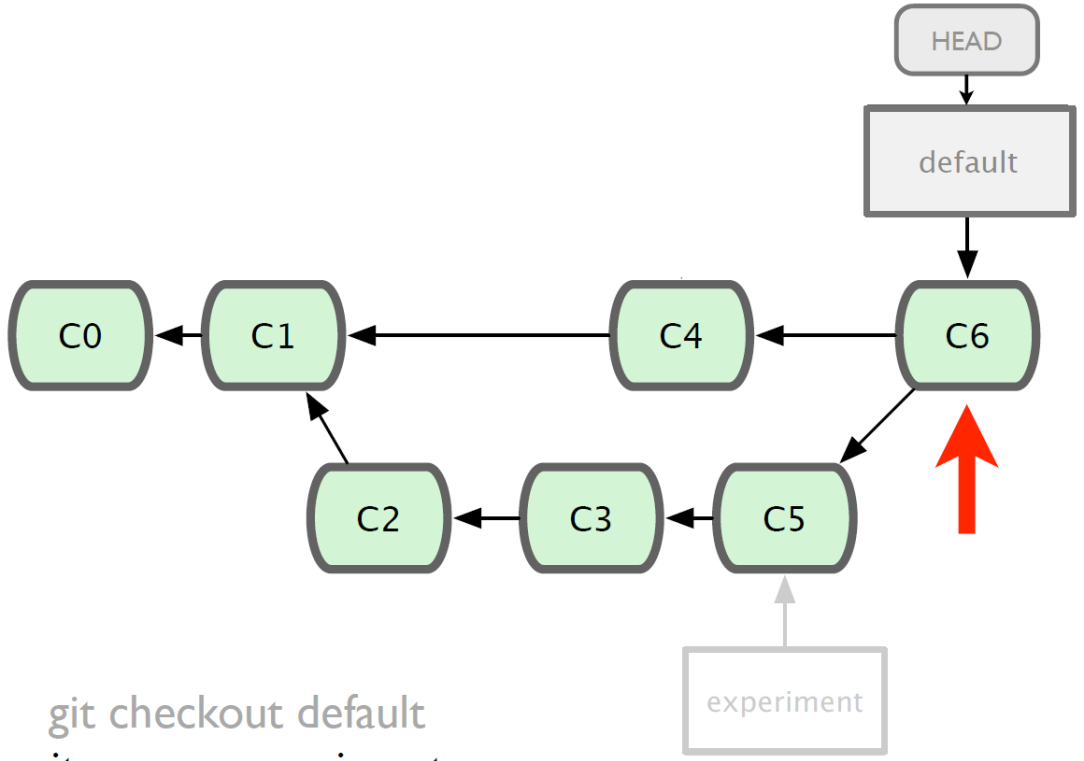


Git: Merging

- Steps to merge two branch
  - Checkout the branch you want to merge onto
  - Merge the branch you want to merge



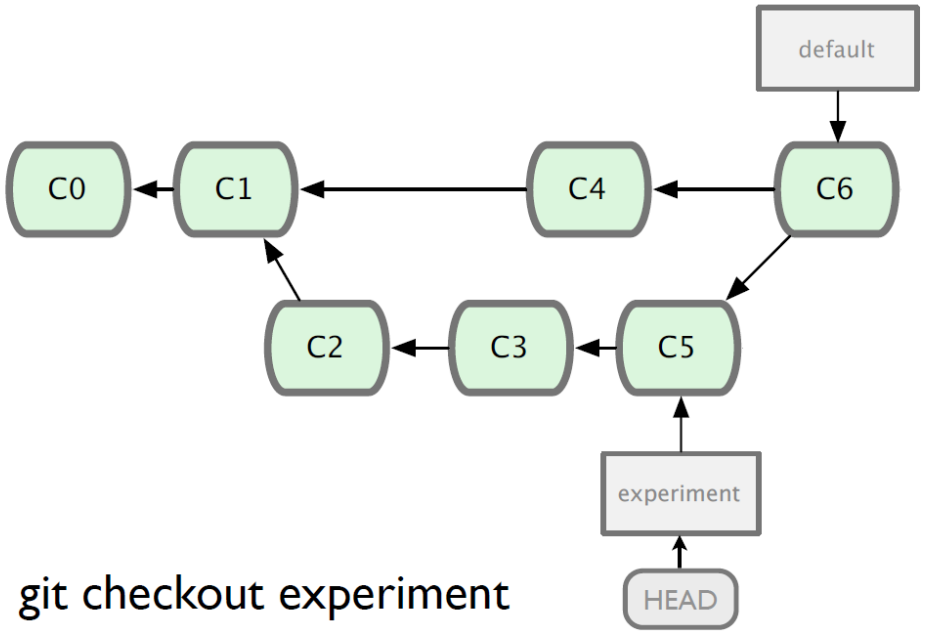
git checkout default



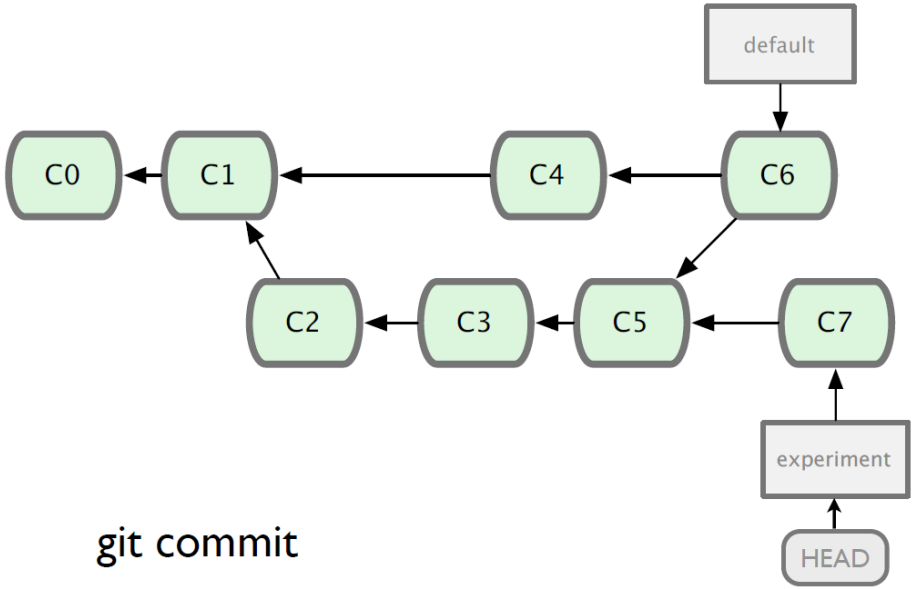
git checkout default
git merge experiment

Git: Merging

- We can continue working one whichever branch we want (the trunk **default** or on **experiment**)



git checkout experiment



git commit

Git: Branching and Merging

- Why this is cool?
  - Non-linear development

```
clone the code that is in production
create a branch for issue #53 (iss53)
work for 10 minutes
someone asks for a hotfix for issue #102
checkout ‘production’
create a branch (iss102)
fix the issue
checkout ‘production’, merge ‘iss102’
push ‘production’
checkout ‘iss53’ and keep working
```

GitHub, UofC GitLab

- It’s a hosting medium/website for your Git repositories
- Offers powerful collaborative abilities
- A good indicator of what you code/how much you code/quality of your code

Git: Working with a remote repository

- Remote?

The common central repository

By default, remote name is **origin** and default branch is **main** *(previously master)*.

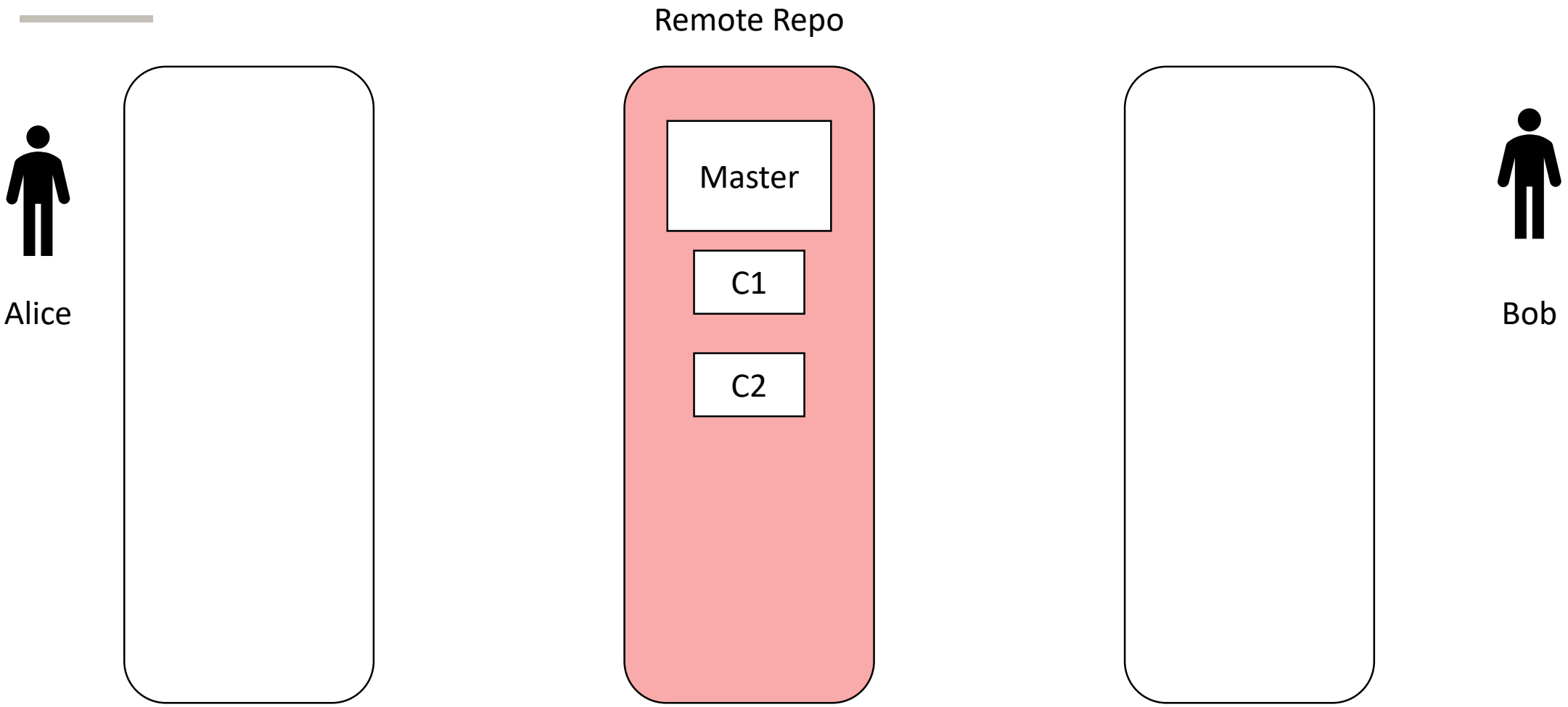
How to access GitHub/UofC GitLab

- Access on https://github.com/ or https://gitlab.cpsc.ucalgary.ca
- Get a clone link
- github.com/intley/Version_Control_Workshop

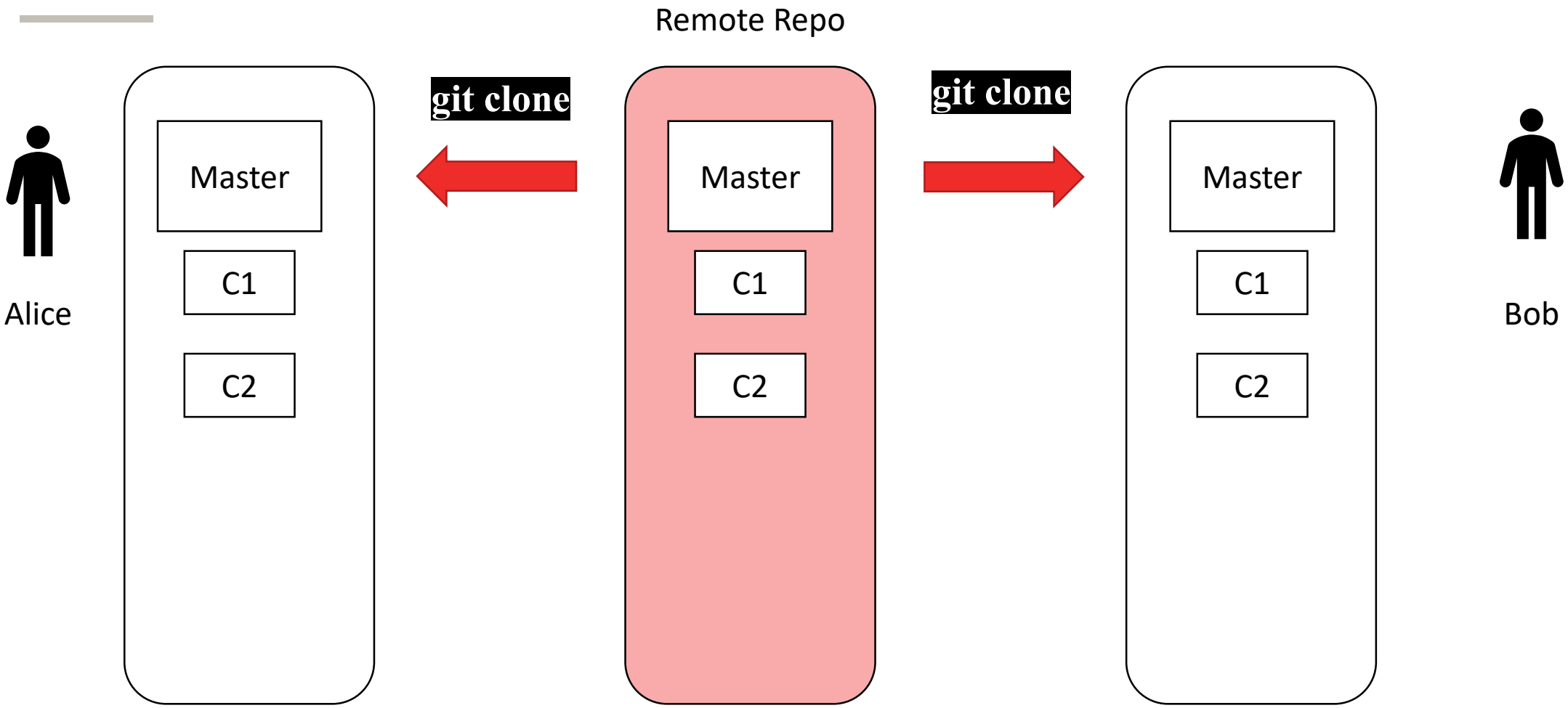
Git: Remote Commands

- git push – push your changes into the remote repository
- git pull – pull your latest changes from the remote repository

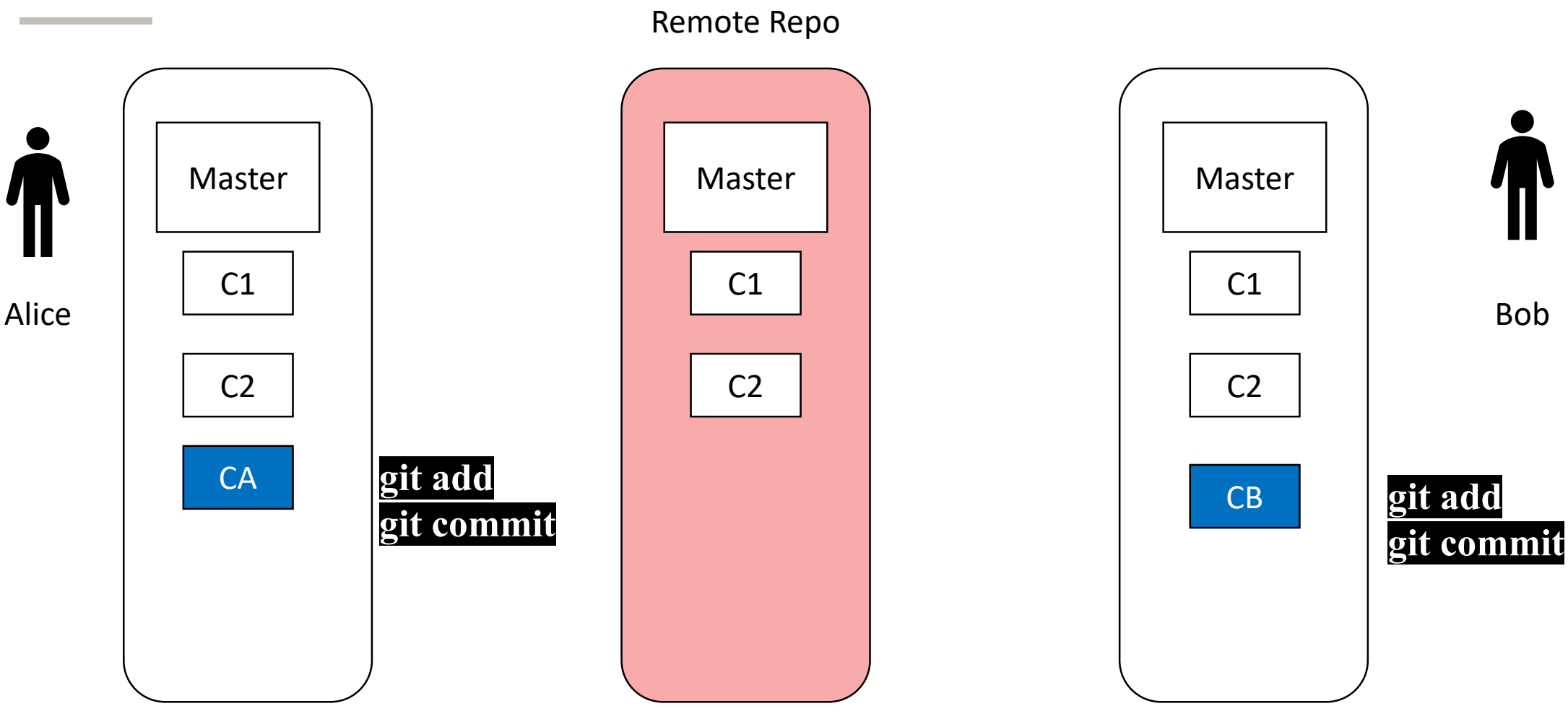
Git: Collaborate



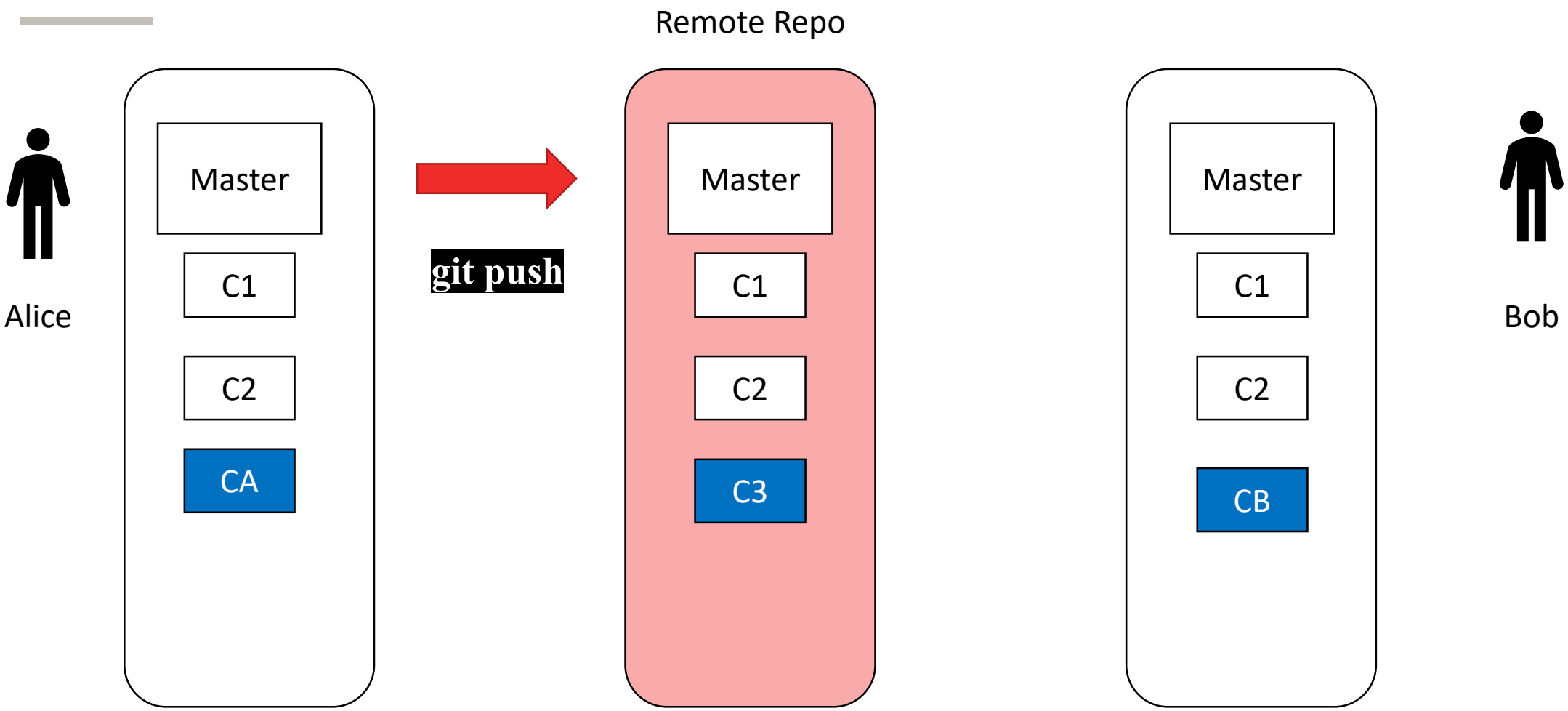
Git: Collaborate



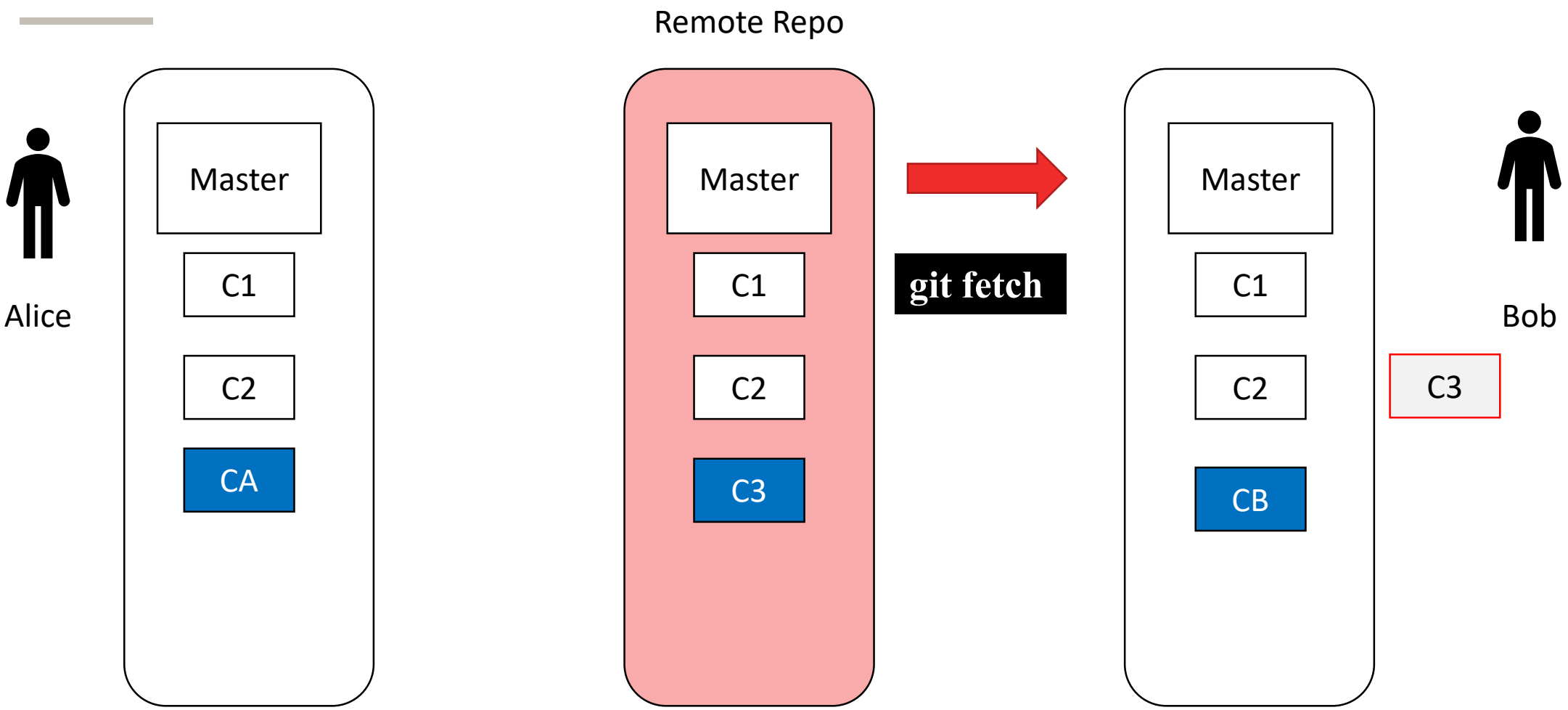
Git: Collaborate



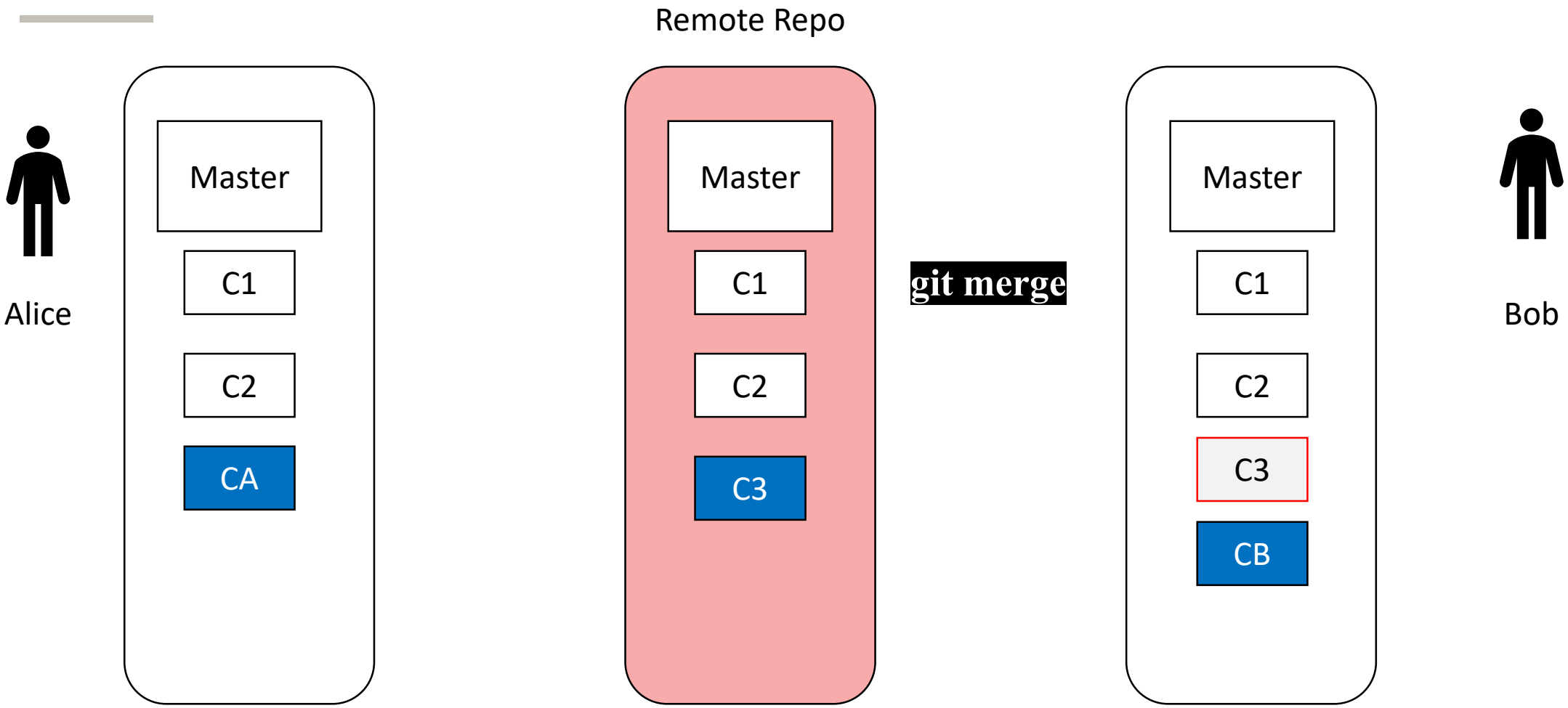
Git: Collaborate



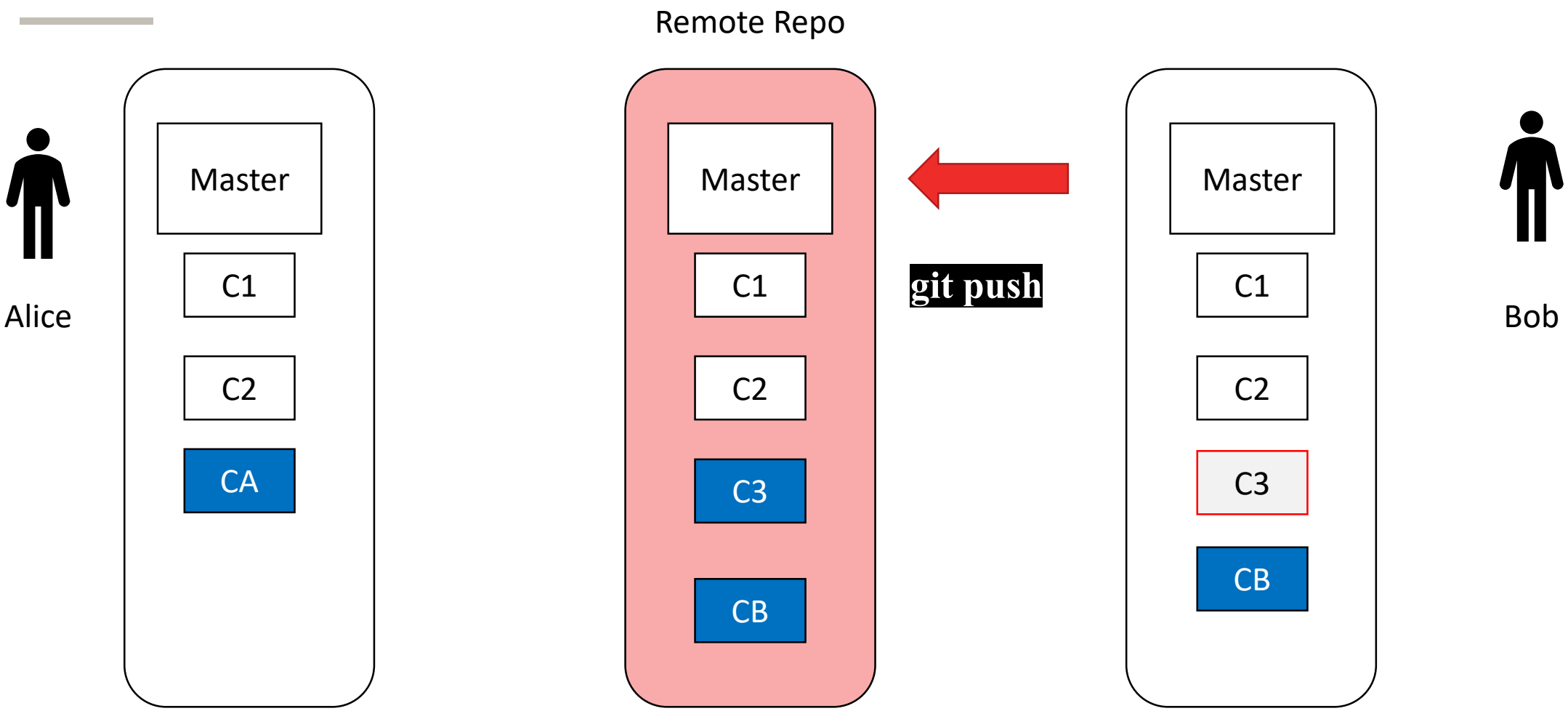
Git: Collaborate



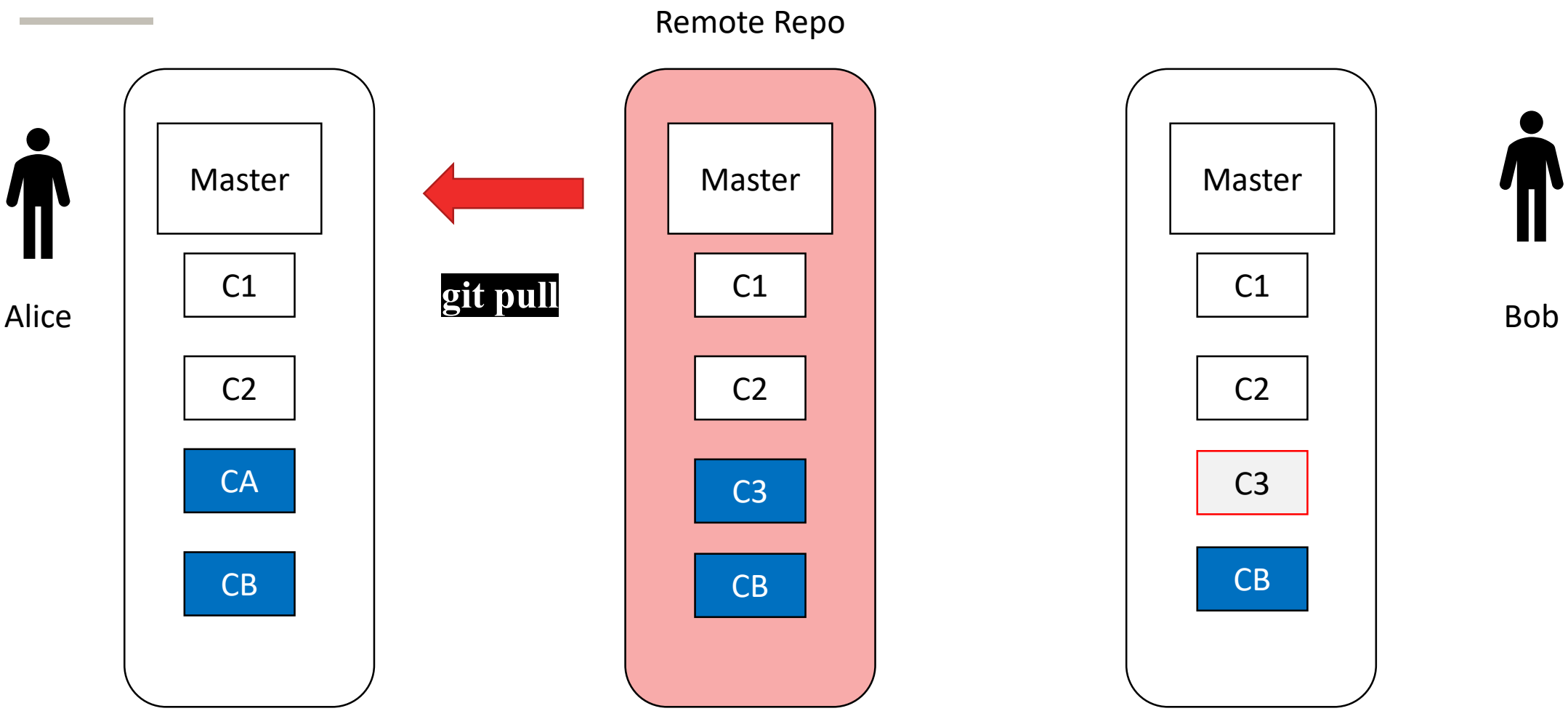
Git: Collaborate



Git: Collaborate



Git: Collaborate



Onward to … comparison.

Jonathan Hudson
jwhudson@ucalgary.ca
https://pages.cpsc.ucalgary.ca/~hudsonj/

UNIVERSITY OF CALGARY