

# Version Control

# What you will learn

- What is Git?
- Why do you need Git?
- How does it work?
- What git is not?
- Git workflow
- Getting started with git
- Git commands

THIS IS GIT. IT TRACKS COLLABORATIVE WORK  
ON PROJECTS THROUGH A BEAUTIFUL  
DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

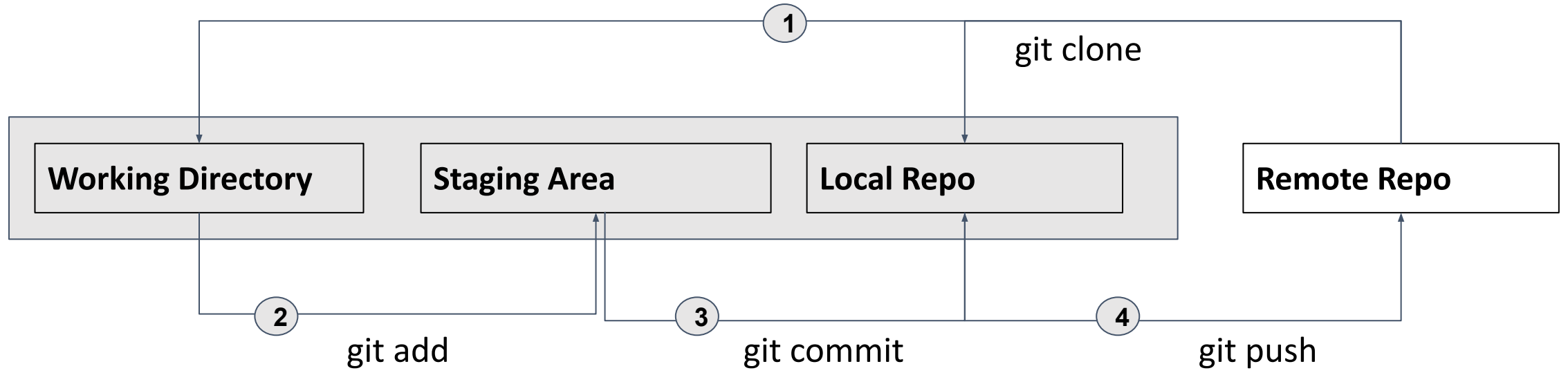
NO IDEA. JUST MEMORIZE THESE SHELL  
COMMANDS AND TYPE THEM TO SYNC UP.  
IF YOU GET ERRORS, SAVE YOUR WORK  
ELSEWHERE, DELETE THE PROJECT,  
AND DOWNLOAD A FRESH COPY.



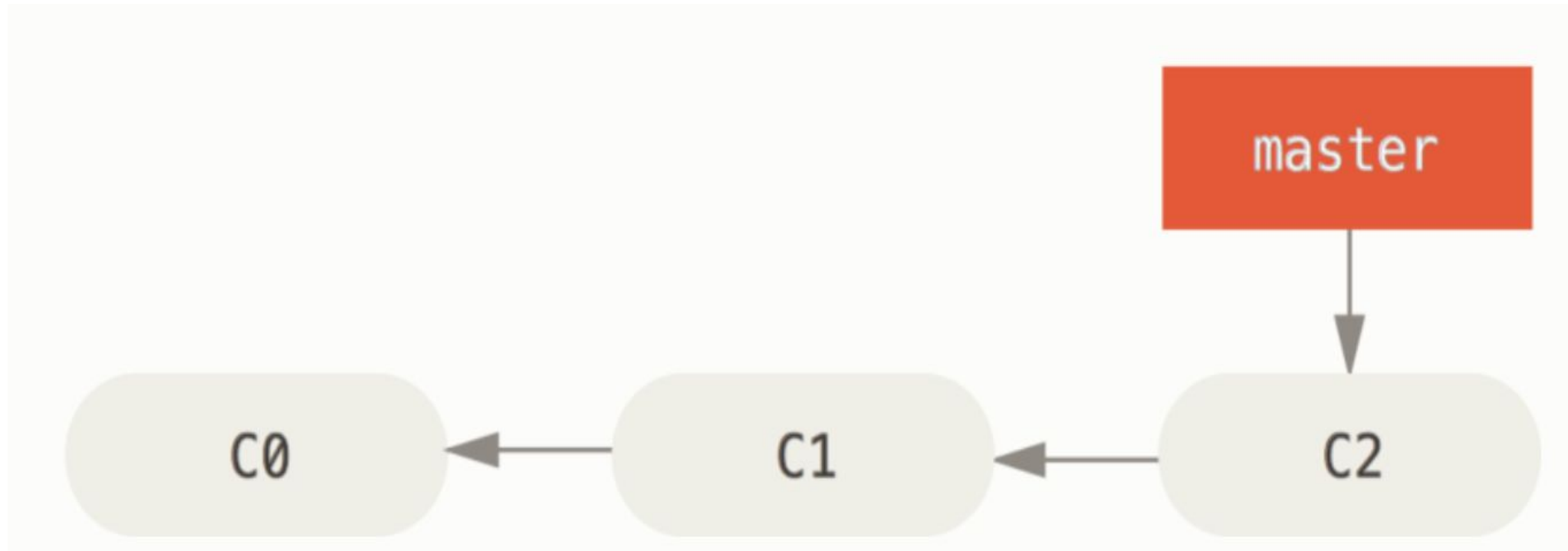
# Git

- Free and open source distributed version control system
- Why GIT?
  - Distributed
  - Performance
  - Reliability
  - Branching

# Git Concepts

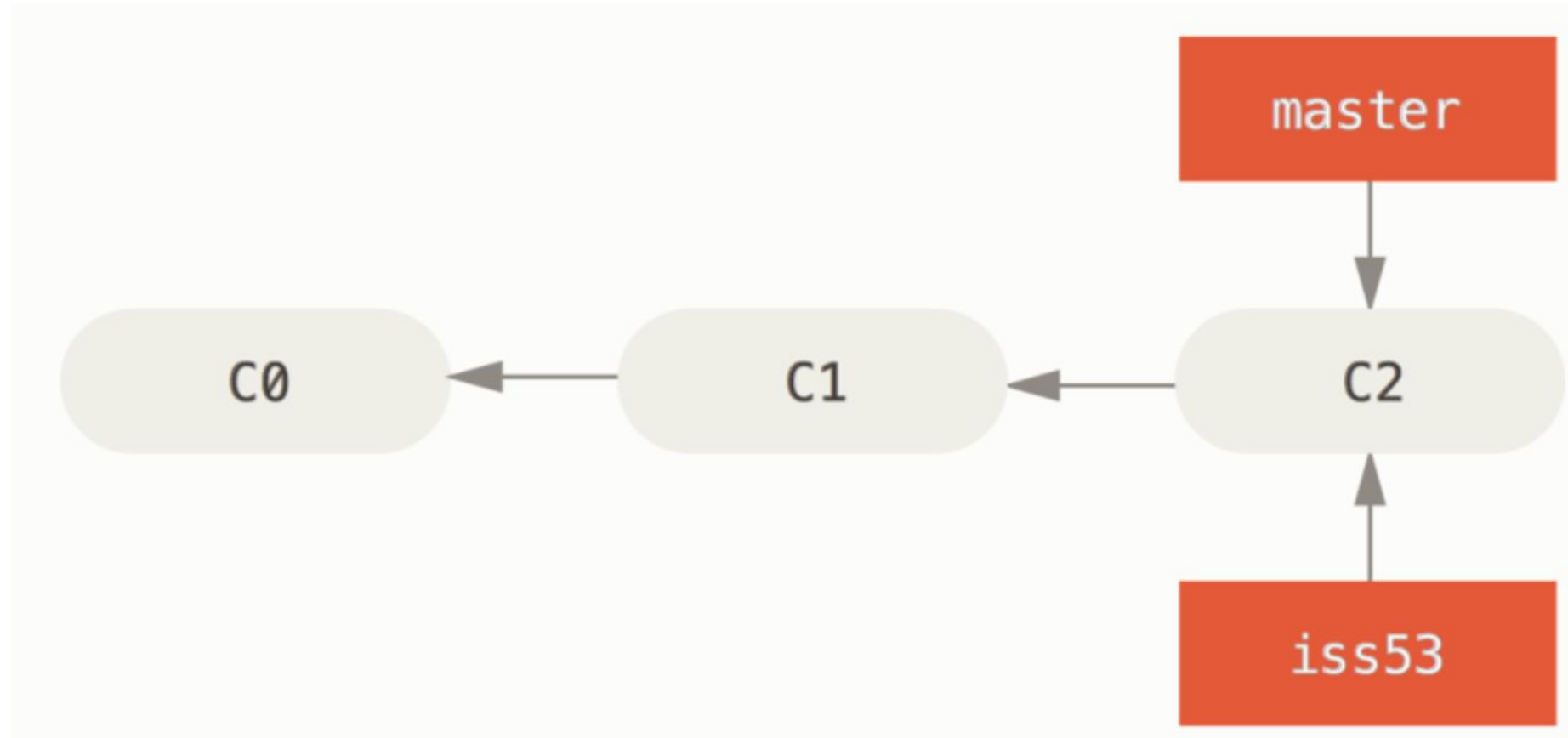


# Git Branches

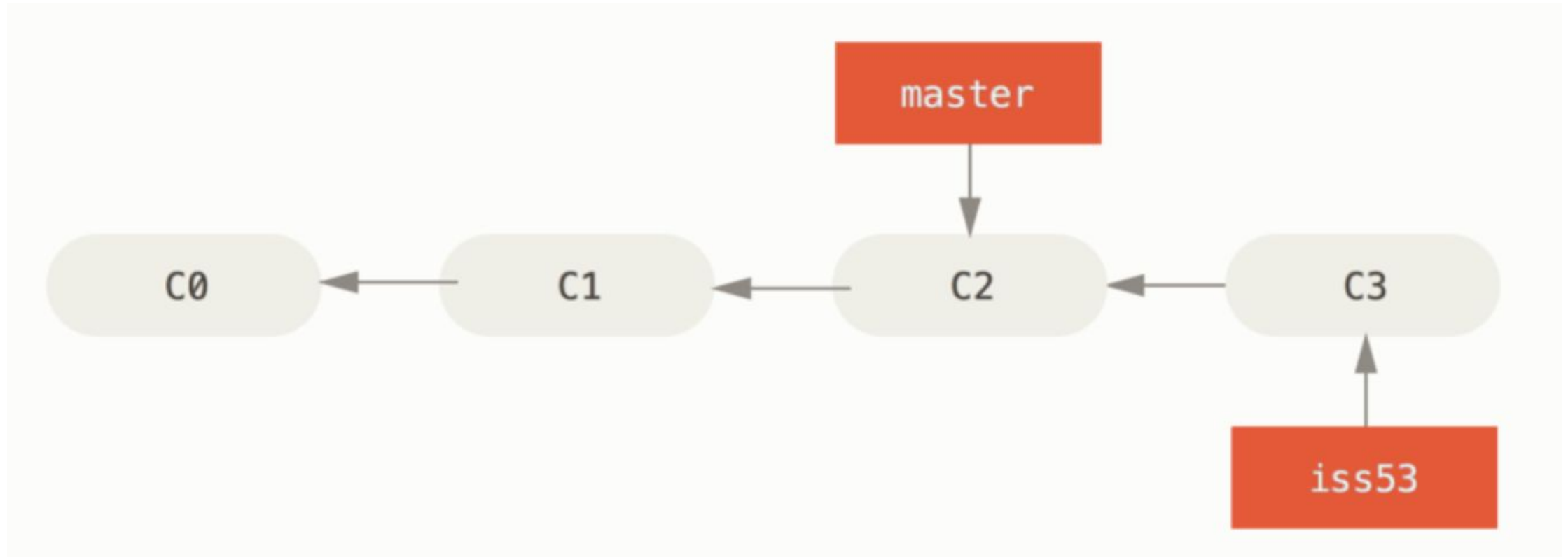


source: <https://git-scm.com/book/en/v2/Git-Branching-Basic-Branching-and-Merging>

# creating new branch

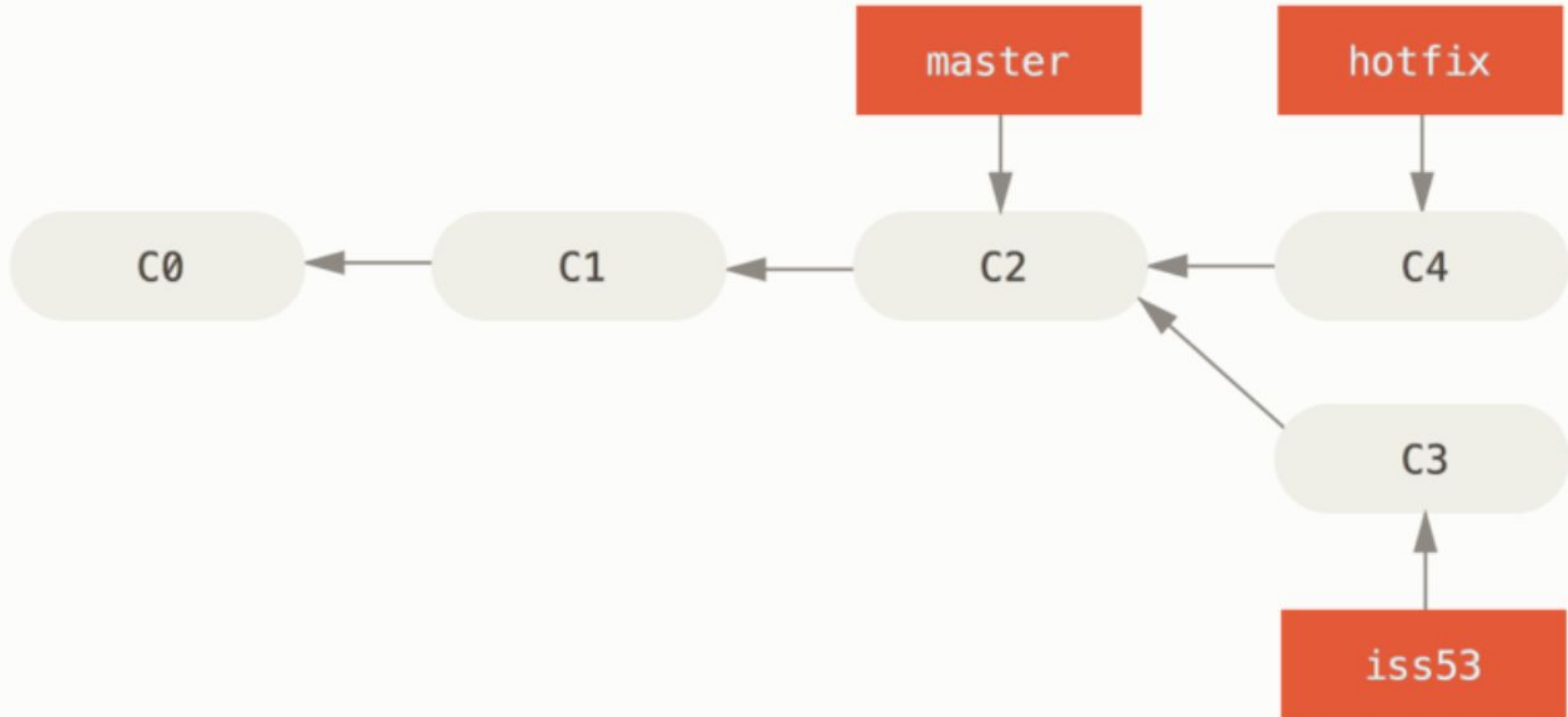


# Working with branches

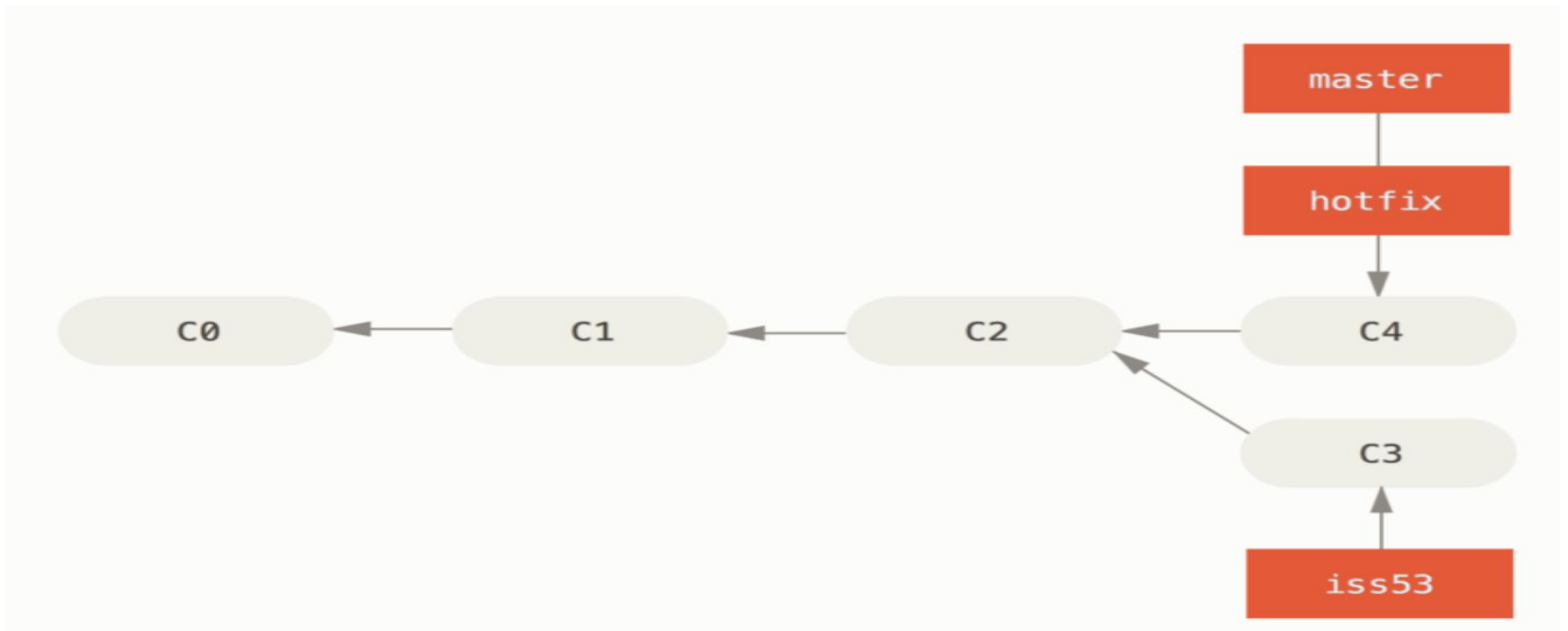




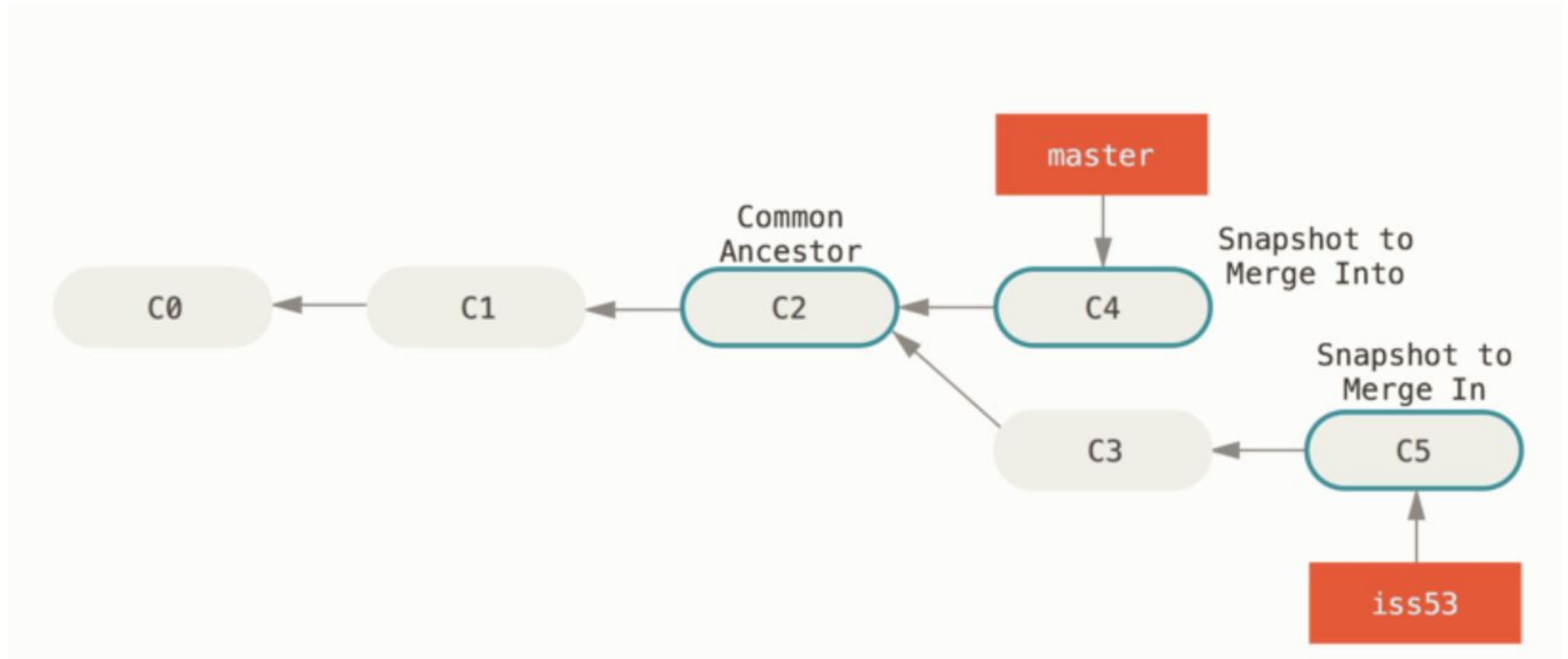
# Hot fix branch based on master



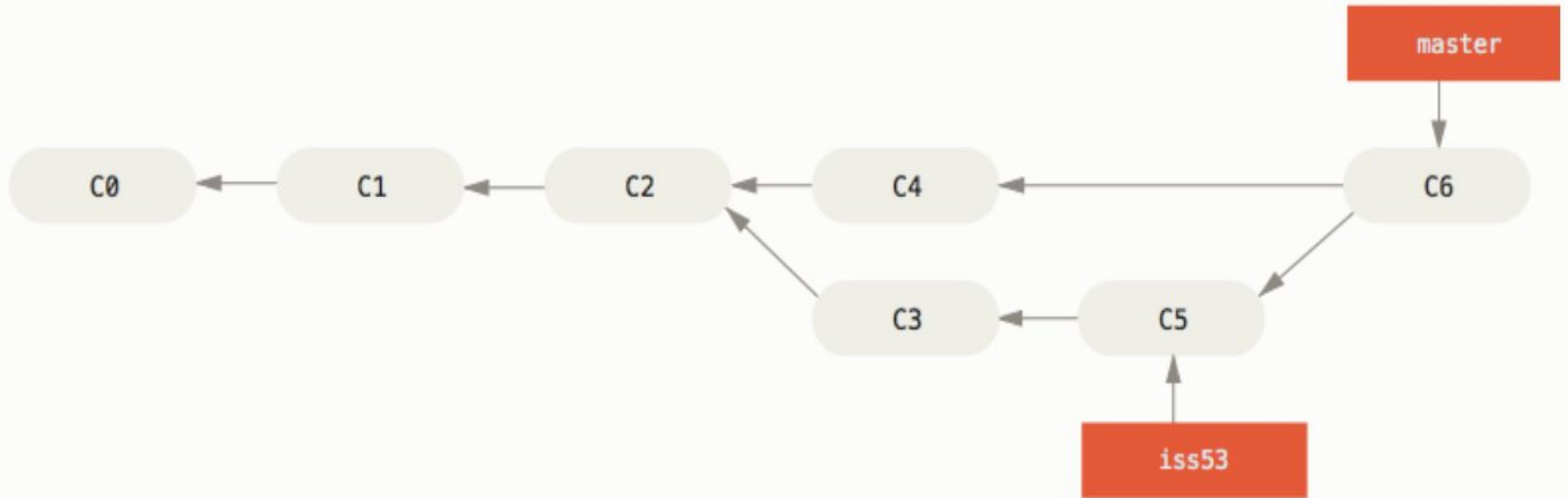
# After Hotfix merge to Master



# Merging iss53 into master



# Merging iss53 into master



Keywords	Description
git	Open-source distributed version-control system, used to store code in repositories
GitHub, GitLab and Bitbucket	Platform for hosting and collaborating on Git repositories
staging	Proposed files/directories that you'd like to commit
commit	Saving all staged files/directories to your local repository
branch	An independent line of development, so you can develop features isolated from each other. Master branch is the default.
clone	Local version of a repository, including all commits and branches
remote	Common repository on eg. Github that all team members to keep that changes in sync with
fork	Copy of a repository owned by a different user
pull request	A method of submitting contributions to a repository
HEAD	Represents your current working directory

# Configuration

---

Key/Command	Description
<code>git config --global user.name [name]</code>	Set author name to be used for all commits
<code>git config --global user.email [email]</code>	Set author email to be used for all commits

# Core Commands

Key/Command	Description
git init [directory]	Creates new local repository
git clone [repo]	Creates local copy of remote repository
git add [directory]	Stages specific [directory]
git add [file]	Stages specific [file]
git add -A	Stages all changed files
git add .	Stages new and changed files, NOT deleted files
git add -u	Stages changed and deleted files, NOT new files
git commit -m "[message]"	Commit everything that is staged
git status	Shows status of changes as untracked, modified or staged

# Synchronization of Changes

---

Key/Command	Description
git fetch	Downloads all history from the remote branches
git merge	Merges remote branch into current local branch
git pull	Downloads all history from the remote branch and merges into the current local branch
git push	Pushes all the commits from the current local branch to its remote equivalent
git-cherry-pick	Apply the changes introduced by some existing commits



# Undo Changes

Key/Command	Description
git checkout -- [file]	Replace file with contents from HEAD
git revert [commit]	Create new commit that undoes changes made in [commit], then apply it to the current branch
git reset [file]	Remove [file] from staging area
git reset --hard HEAD	Removes all local changes in working directory
git reset --hard [commit]	Reset your HEAD pointer to previous commit and discard all changes since then

# Branches

Key/Command	Description
git branch [branch]	Create a new branch
git checkout [branch]	Switch to that branch
git checkout [branch] -b	Create and checkout new branch
git merge [branch]	Merge [branch] into current branch
git branch -d [branch]	Deletes the [branch]
git push origin [branch]	Push [branch] to remote
git branch	Lists local branches
git branch -r	Lists remote branches
git branch -a	Lists local and remote branches

# History

Key/Command	Description
git log	Lists version history for the current branch
git log --author=[name]	Lists version history for the current branch from certain author
git log --oneline	Lists compressed version history for the current branch
git show [commit]	Outputs metadata and content changes of the specified commit
git blame [file]	Shows who changed what and when in file

# What you will learn

- Setup Git
- Fork a repo
- Setup access token
- Create Branch
- Make changes in local
- Push changes to repo

# Setting up Git

- [Download and install the latest version of Git](#)
- [Set your username in Git.](#)
  - `git config --global user.name "YOUR_NAME"`
- [Set your commit email address in Git](#)
  - `git config --global user.email "YOUR_EMAIL"`

# Setting up Github

- Github Repo: <https://github.com/edwinMarquez/YearUpJava>
- Create a fork from <https://github.com/edwinMarquez/YearUpJava>
- Create a personal access token (Github -> Settings ->Developer Settings->Personal Access Tokens ->Generate New Token->Select all->Copy token)
- Add Collaborators: Settings -> Collaborators (Add VidhyaRamamoorthy, mtsaber01)

# Local Git setup

- Start cloning repo:
  - git clone <https://github.com/yourname/YearUpJava.git>
- Setup Remote:
  - git remote add upstream <https://github.com/edwinMarquez/YearUpJava>
- Create your own branch from main
  - git checkout -b student/YOURLASTNAME (same as git branch student/YOURLASTNAME followed by git checkout student/YOURLASTNAME)
  - If you get any permission denied error in accessing the remote repo use the following command to set the remote repo

git remote set-url origin

<https://<githubtoken>@github.com/<username>/<repositoryname>.git>

# References

- <https://git-scm.com/book/en/v2>
- <https://git-scm.com/docs>
- <https://training.github.com/downloads/github-git-cheat-sheet/>
- <https://docs.github.com/en/get-started/quickstart/set-up-git>
- [\*\*https://docs.github.com/en/get-started/quickstart/hello-world\*\*](https://docs.github.com/en/get-started/quickstart/hello-world)
- <https://docs.github.com/en/get-started/getting-started-with-git/associating-text-editors-with-git>
- <https://www.youtube.com/watch?v=4XpnKHJAok8>
- <https://www.sourcetreeapp.com/>
- <https://www.atlassian.com/git/tutorials/atlassian-git-cheatsheet>