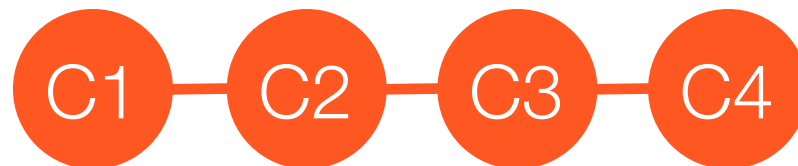# Visualizing Git Workflows

A visual guide to 539 workflows
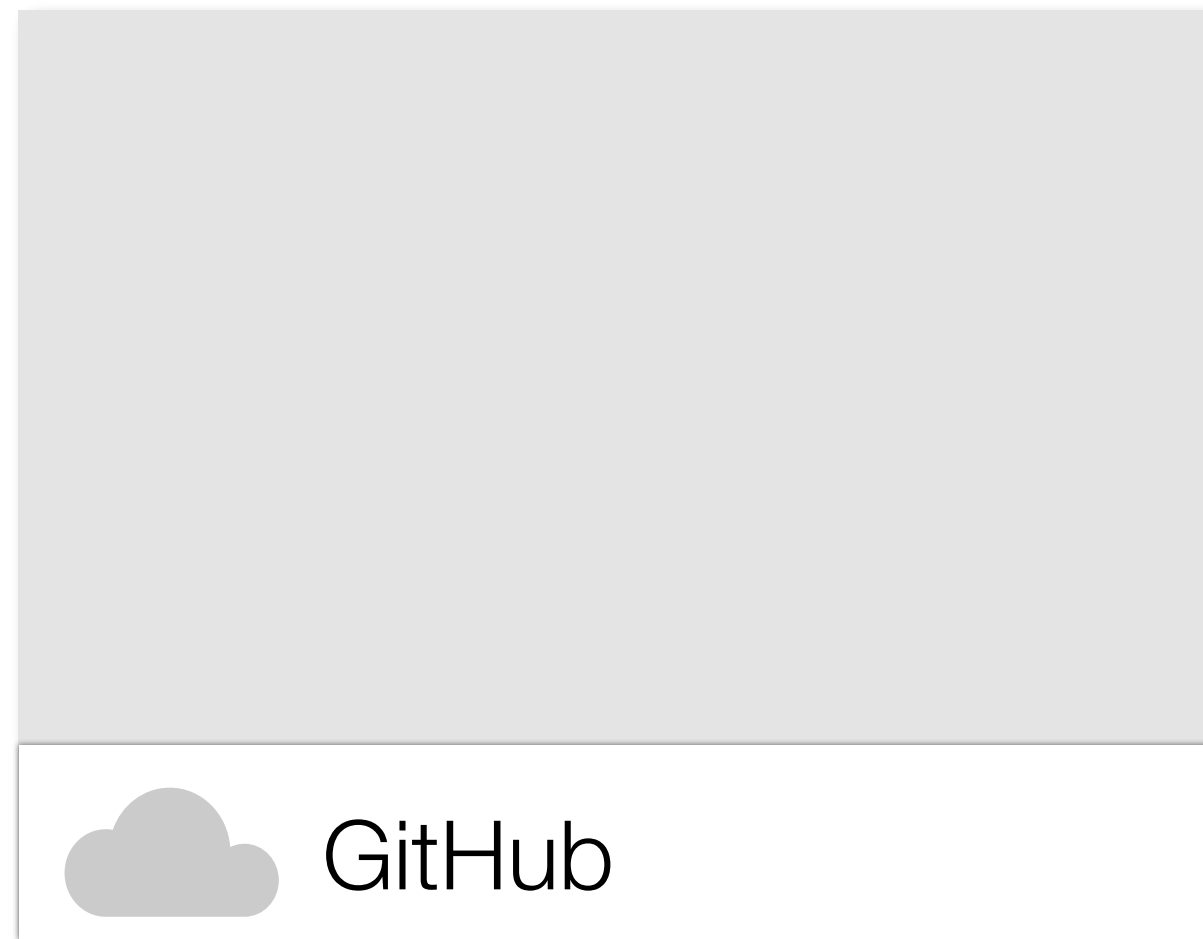
Table of Contents

Notation

Notation

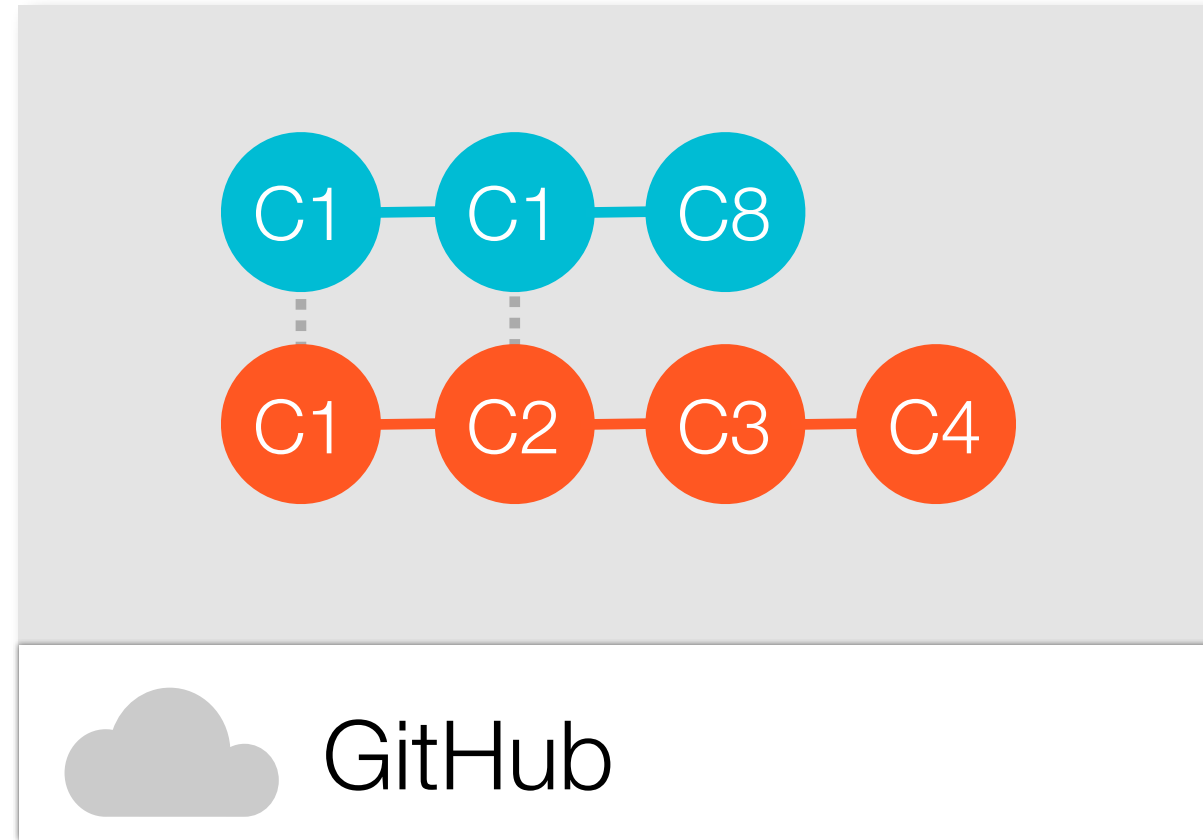This is a single commit. The text is the commit ID.

This is a branch (i.e. sequence of commits).
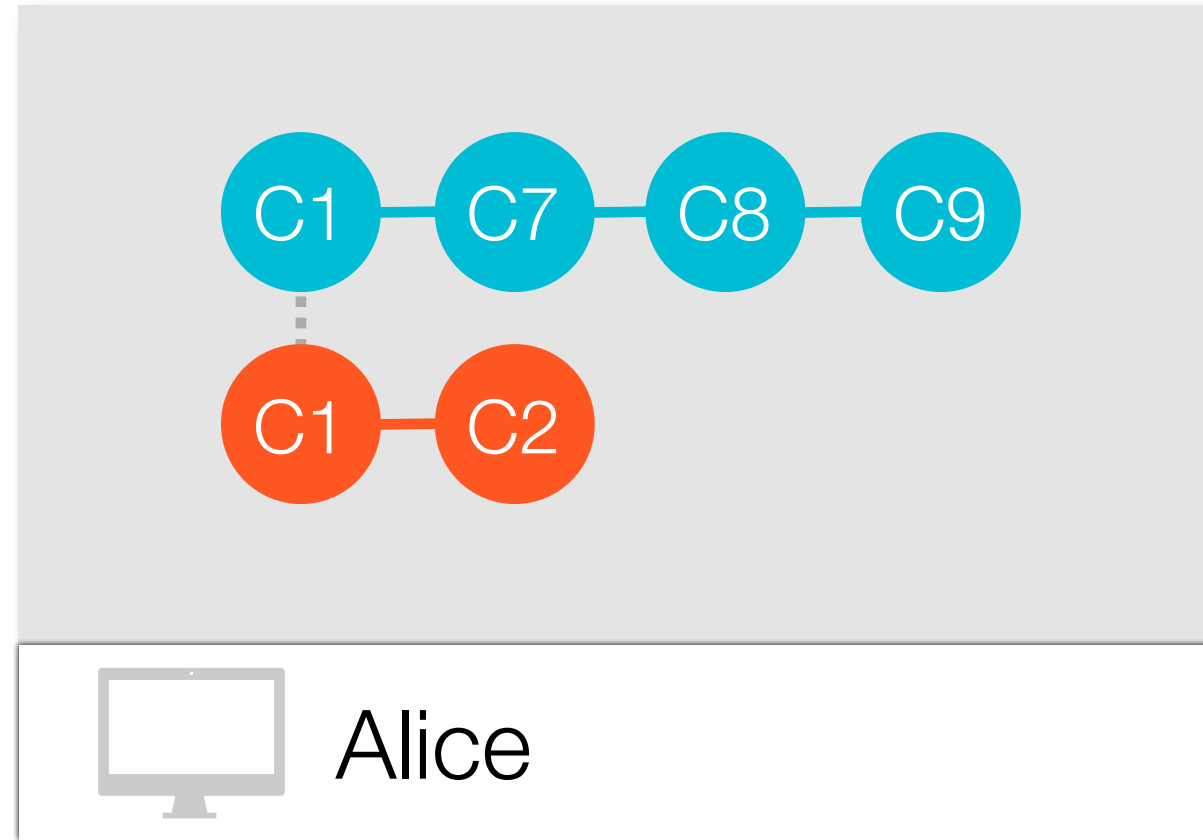Each branch has a different color.

Notation

These boxes show the state of a repository
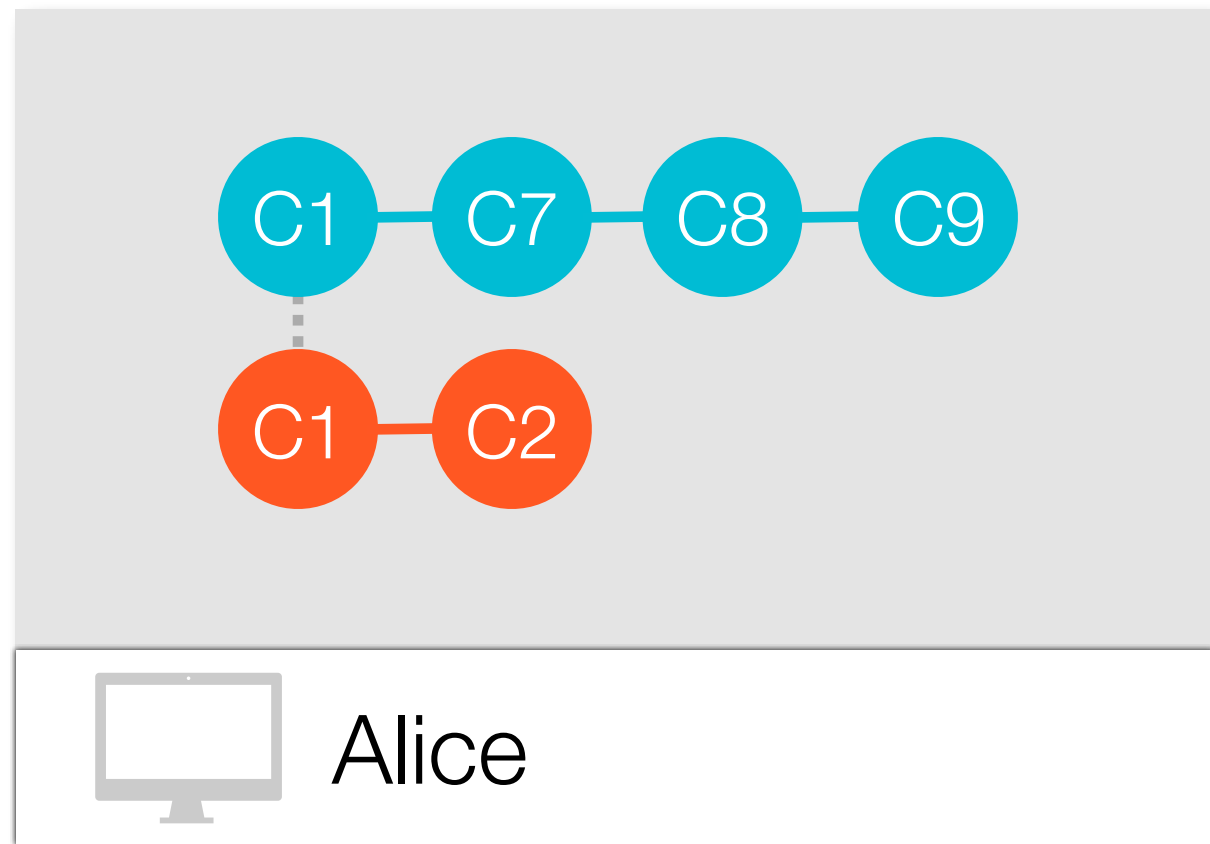at a given location.

Notation

This means that GitHub is storing two branches.
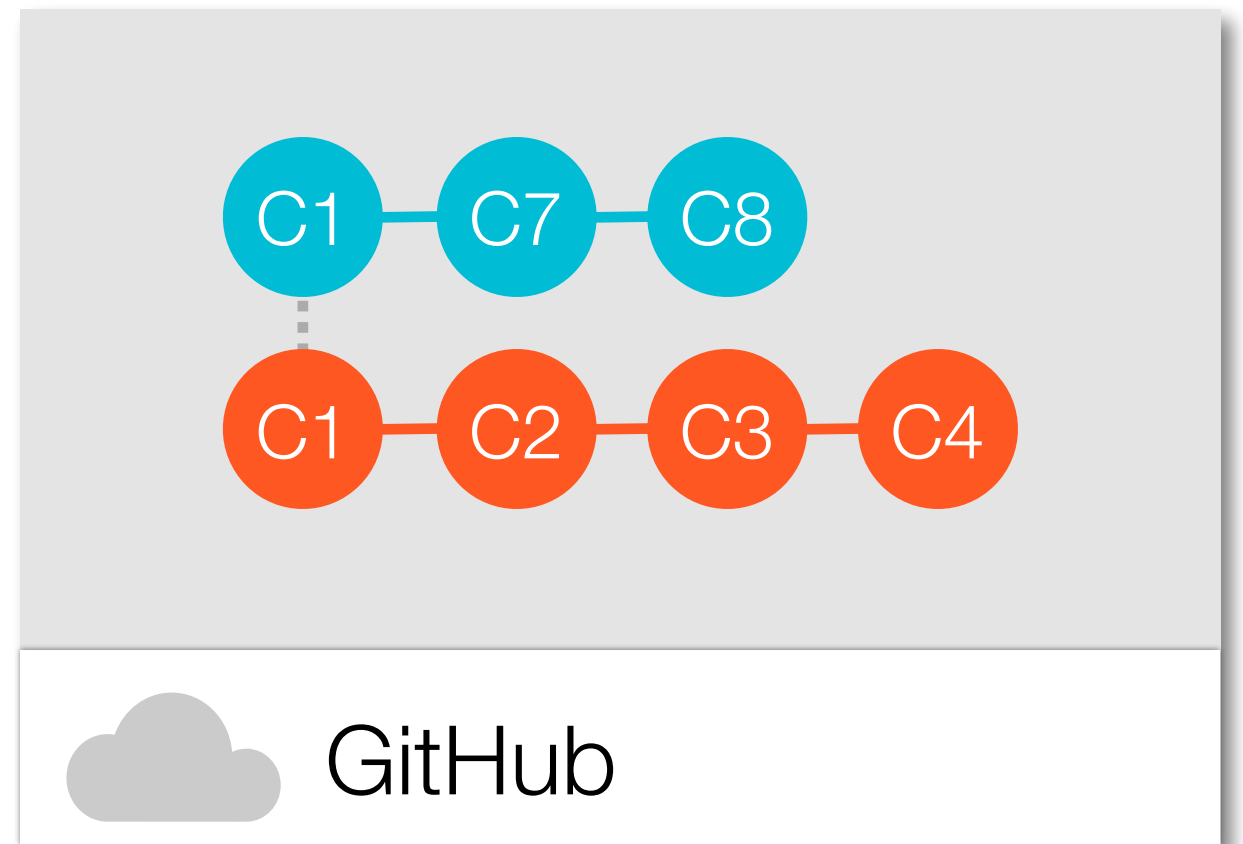One has 4 commits, the other has 3.

Notation

This means that Alice's computer is storing
local copies of the same two branches.
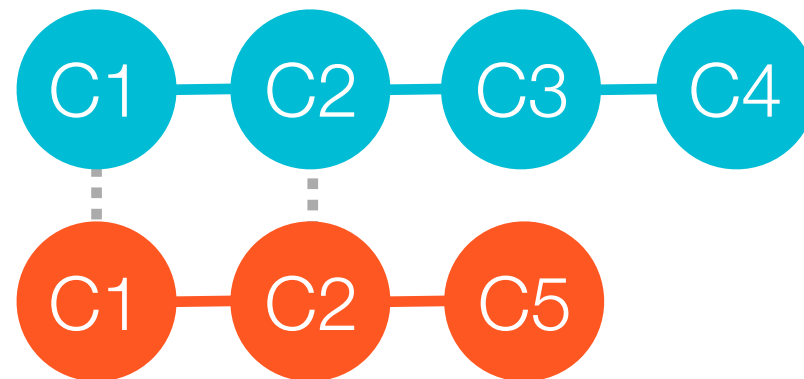
Notation

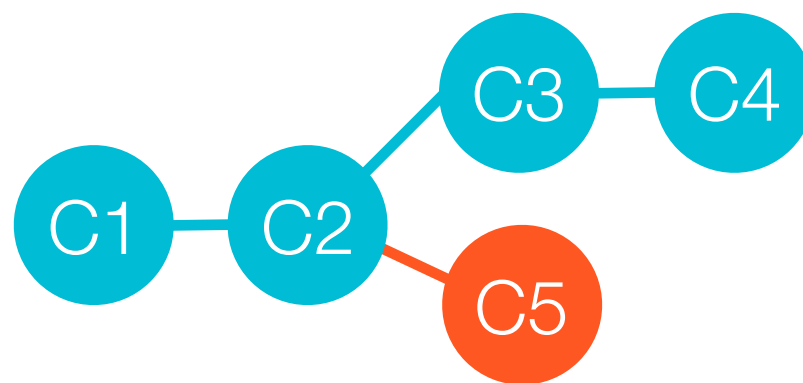
Local Copy

Remote Copy

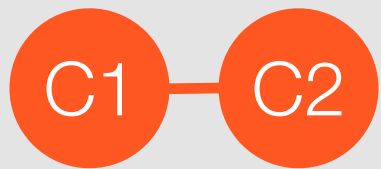Same branches, different commits.

Notation

I show branches like this:

Other tutorials may show this:

They mean the same thing

Collaboration Without Review or Branches
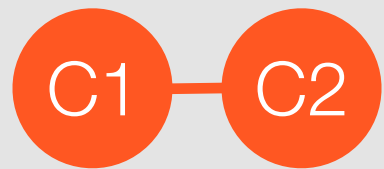
C1 — C2

GitHub

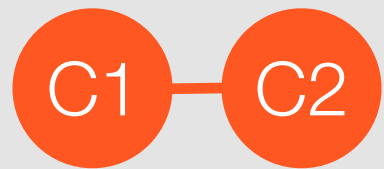There is a "master branch" on github with two commits.

Alice

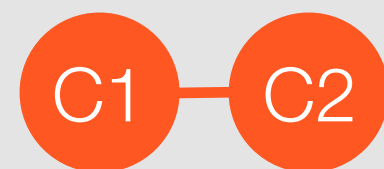Bob

C1 — C2

GitHub

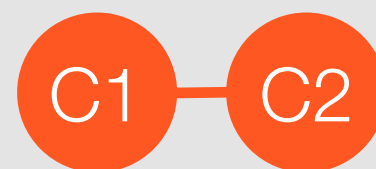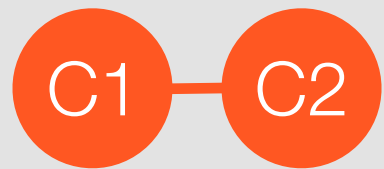Alice clones the repository on her computer.

C1 — C2
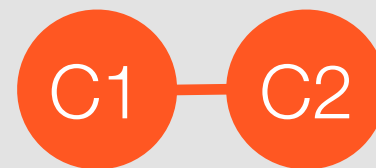
Alice

Bob

Bob clones the repository on his computer.

GitHub

Alice

Bob

C1 — **C2**

GitHub

Alice commits new code to her master branch

C1 — **C2** — **C3**

Alice

C1 — **C2**

Bob

Alice pushes the feature branch to GitHub for all to see.
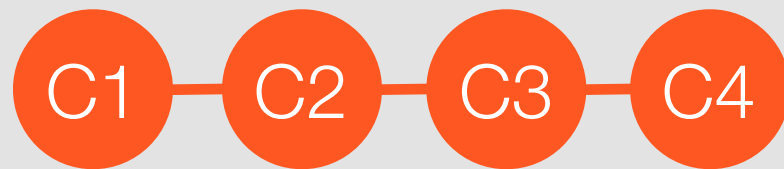
Bob fetches Alice's new changes from GitHub.

GitHub

C1 — C2 — C3

Alice

C1 — C2 — C3

Bob

C1 — C2 — C3

Bob builds some of his code on Alice's new commit.

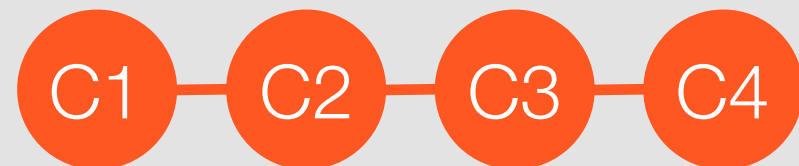C1 — C2 — C3

GitHub

C1 — C2 — C3

Alice
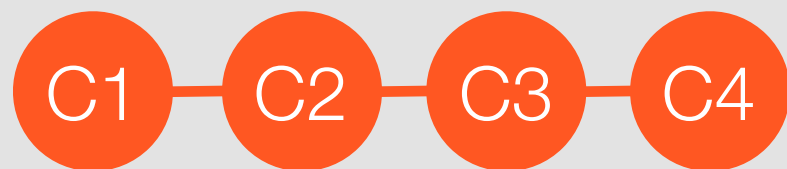
C1 — C2 — C3 — C4
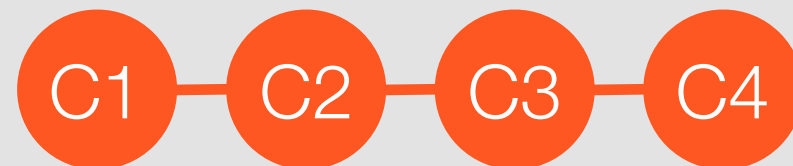
Bob

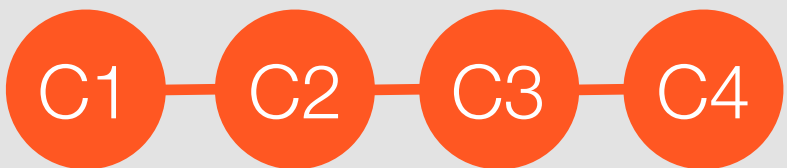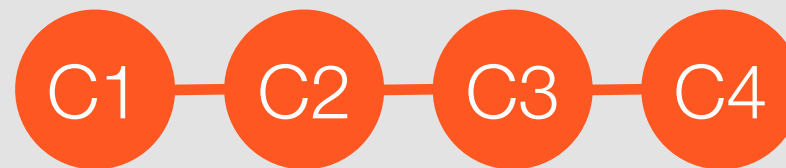Bob pushes his code to GitHub for Alice to see.

GitHub

Alice

Bob

Alice pulls Bob's changes, so now everyone has the same code.

GitHub

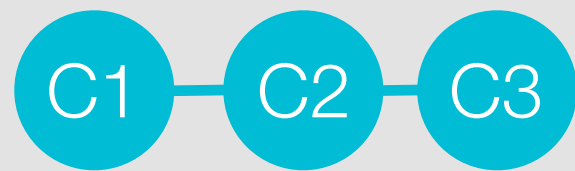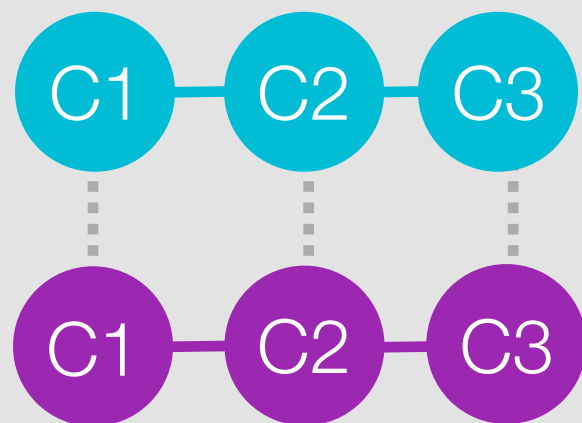Alice

Bob

Merge Conflicts

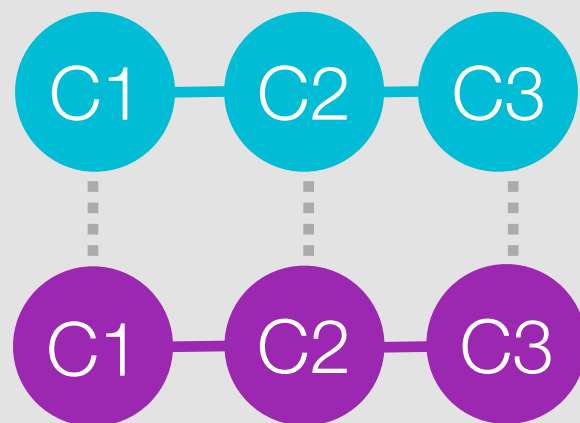What happens when we change the same file?

Imagine Alice is fixing a bug.

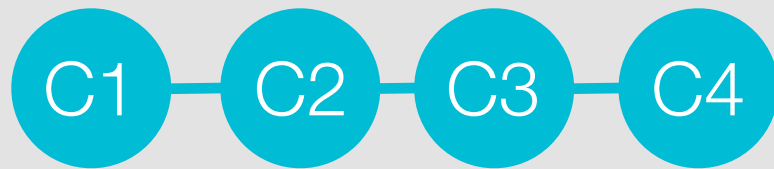She starts by checking out a new purple branch, based off of her local master.

C1 — C2 — C3 — C4

GitHub

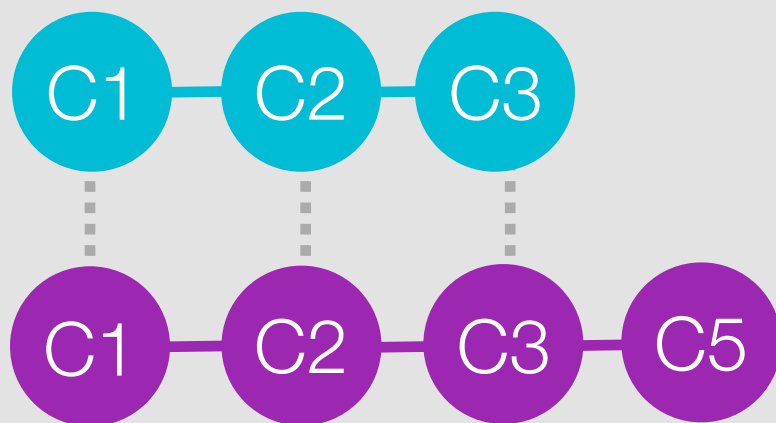Around the same time, bob pushes a new commit, C4, to master.

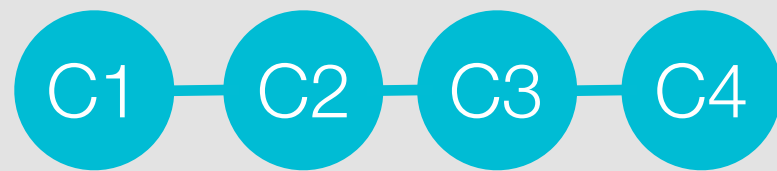Now the remote master (on GitHub) is out of sync with Alice's local master branch.

C1 — C2 — C3

C1 — C2 — C3

Alice

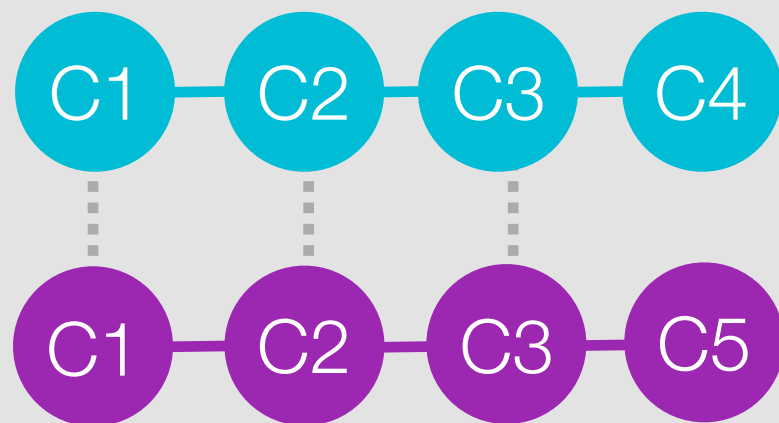GitHub

C1 — C2 — C3 — C4

Alice

C1 — C2 — C3
C1 — C2 — C3 — C5

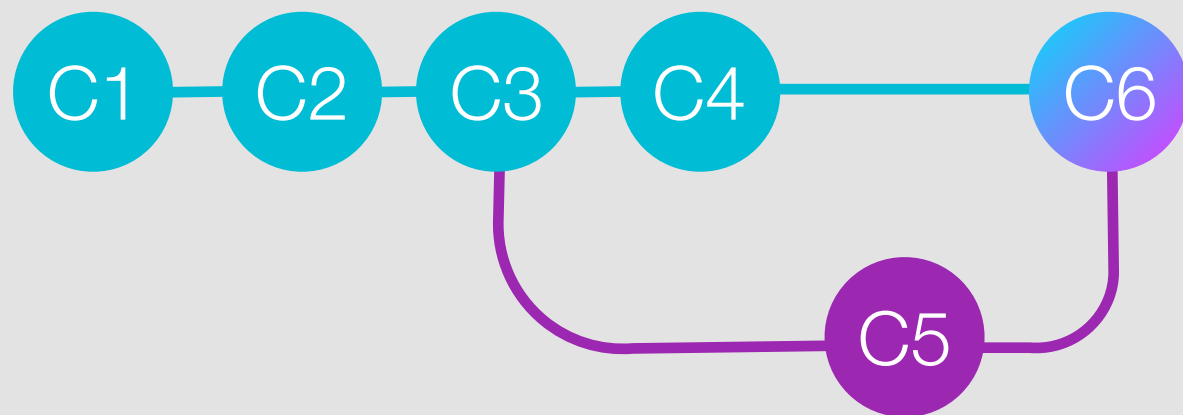Alice creates a new commit, C5, to fix the bug on her new branch.

Now Alice pulls Bob's updated changes from GitHub to her computer.

Bob put the new feature on the master branch, so when Alice pulls, it goes to her local master branch.

Notice how the last commit of the bug fix branch is different from master.

C1 — C2 — C3 — C4
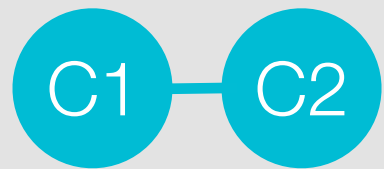
GitHub

C1 — C2 — C3 — C4 ———— C6
         C5

Alice

Now Alice adds her local bug fix to her local master branch.

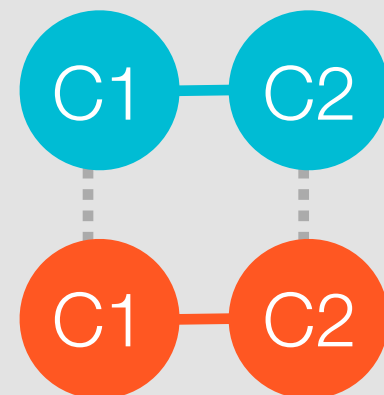She does this by **merging** her changes from her bug fix branch to her master branch.

If there is a conflict with Bob's changes, this results in a special **merge commit**.

Requesting Code Review

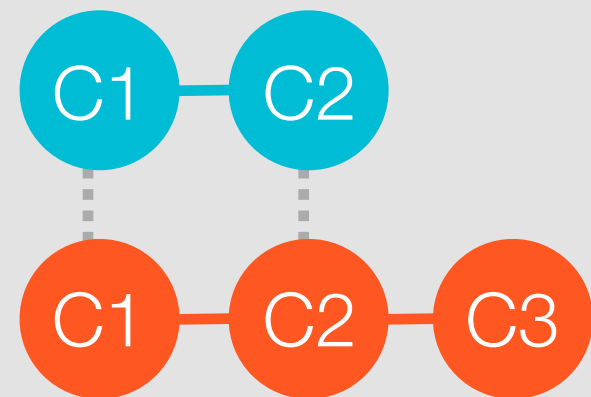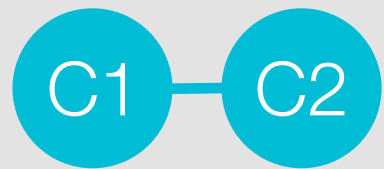Do this for every change outside of SkunkWorks!
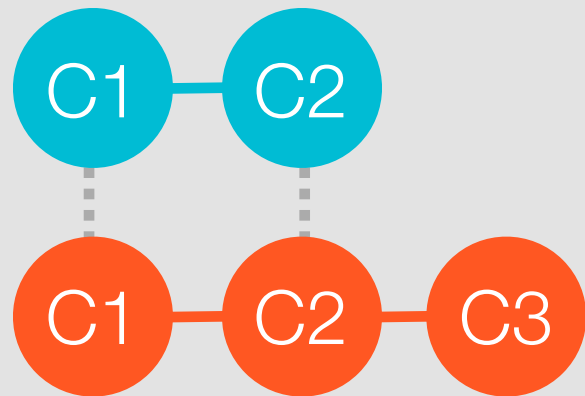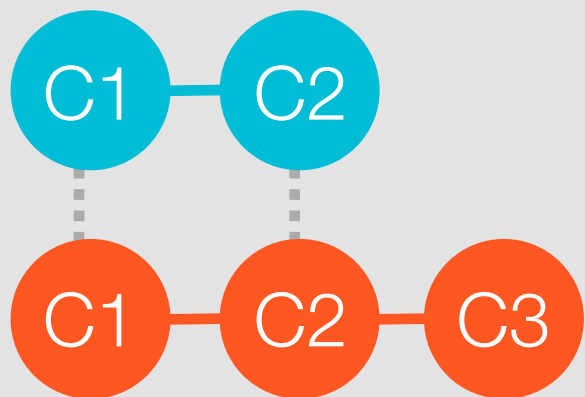
GitHub

Alice creates a new "feature" branch.

Alice

GitHub

Alice commits new code to her feature branch

Alice
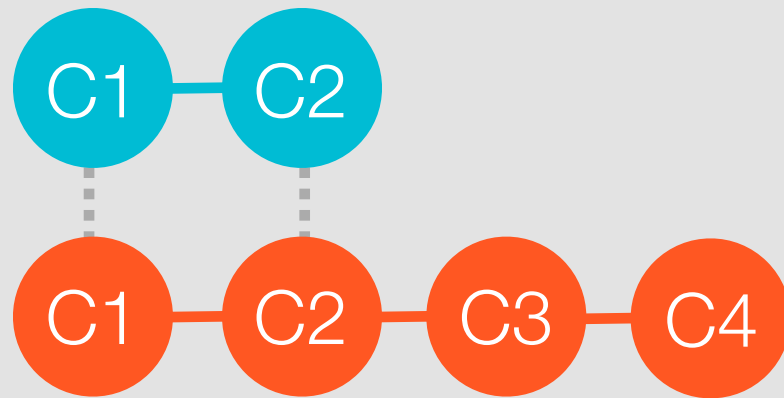
Alice pushes the feature branch to GitHub for all to see.
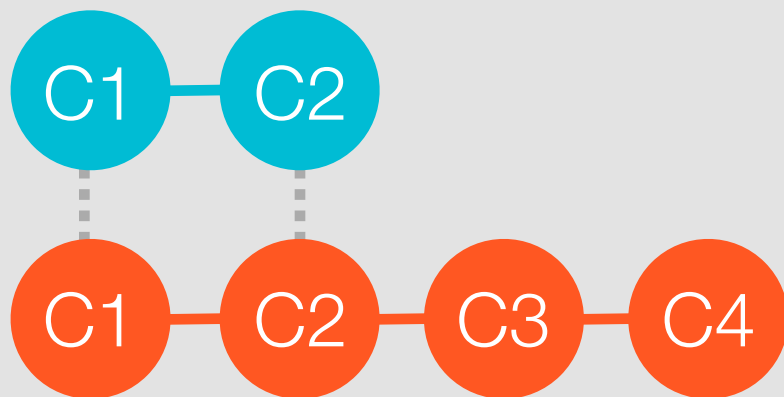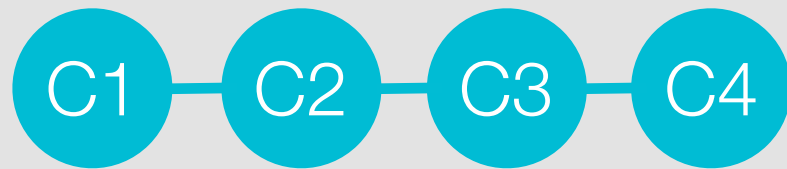
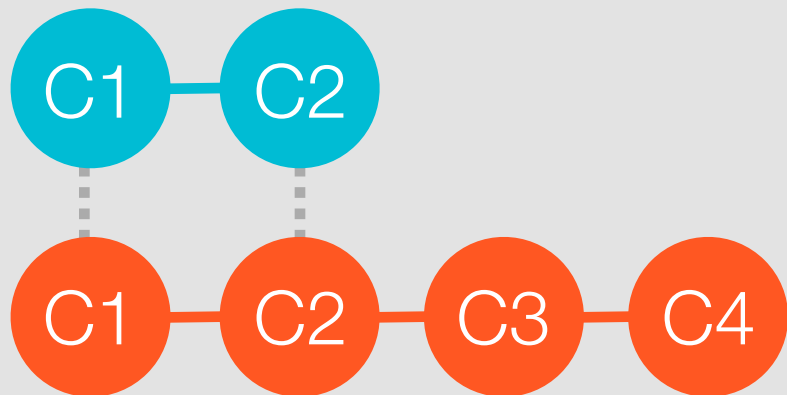She creates a **pull request** and asks for a review.

GitHub

Alice

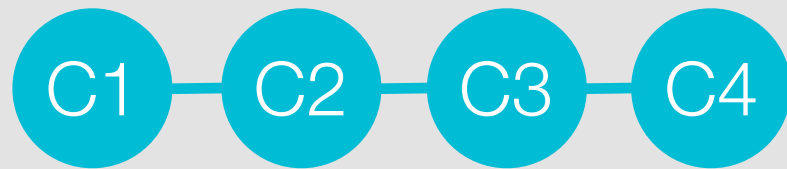After addressing a few comments in a new commit, the Koz says "Ayyyy that code looks good."

GitHub

Alice

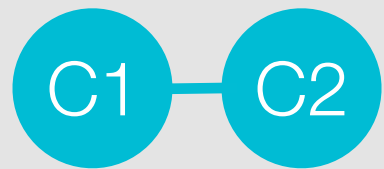After Alice addresses any review comments and obtains approval, the Koz merges her changes to master.

C1 — C2 — C3 — C4

GitHub

Alice now pulls the new master from GitHub, and deletes her old branch.
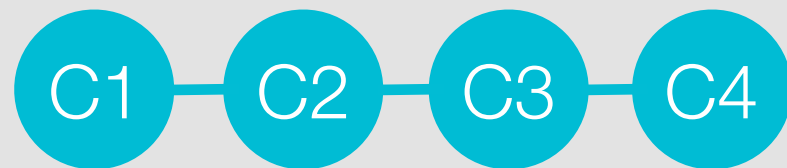
Now everyone has the same code.

C1 — C2 — C3 — C4

Alice

Collaboration With Multiple branches

C1 — C2

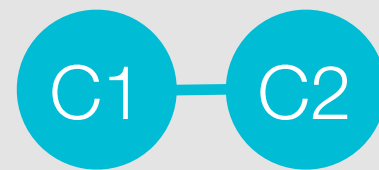GitHub

There is a "master branch" on github with two commits.
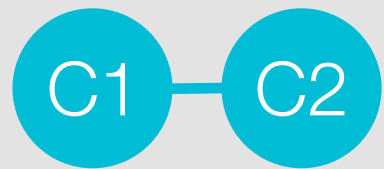
Alice

Bob
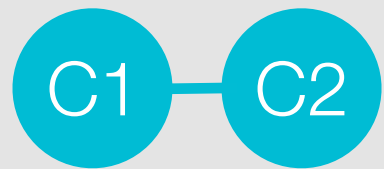
Both Alice and Bob clone the repository on their computers.
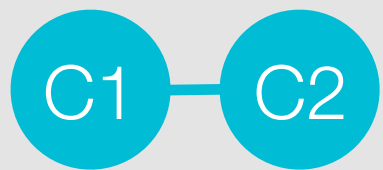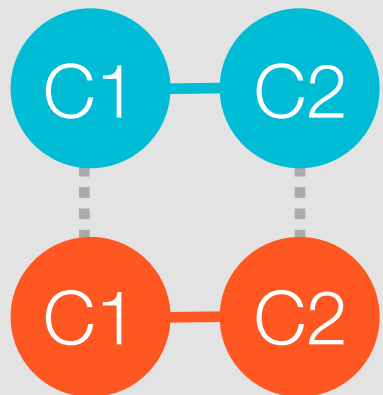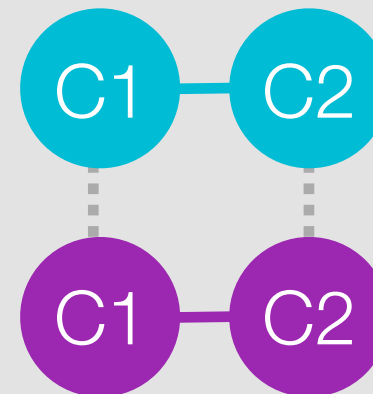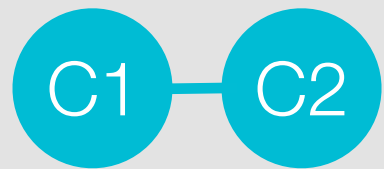
GitHub

Alice

Bob

Alice creates a local orange branch.

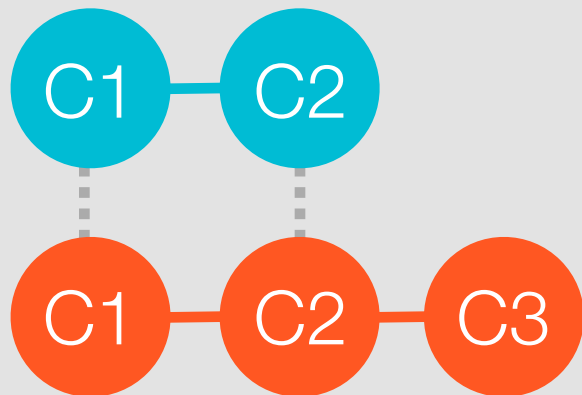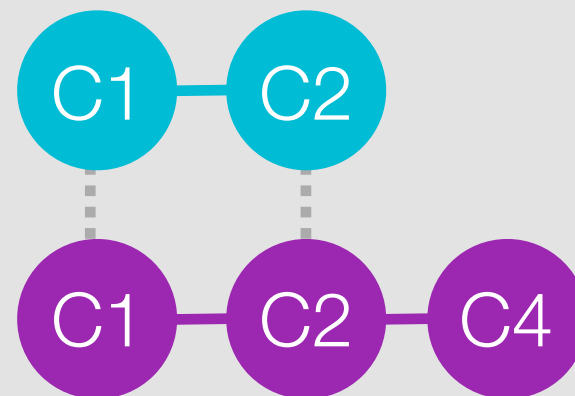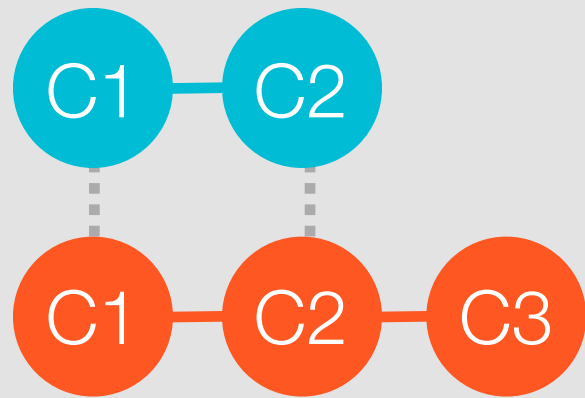Bob creates a local purple branch.

GitHub

Alice

Bob

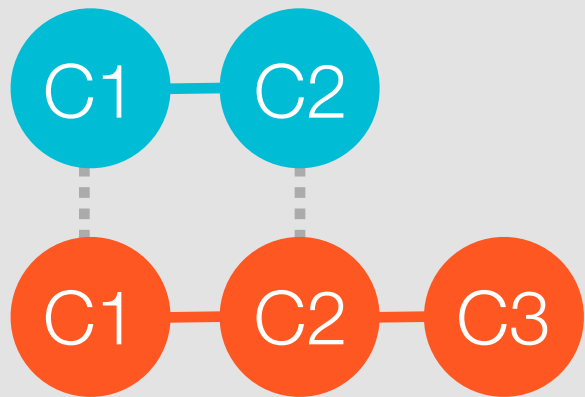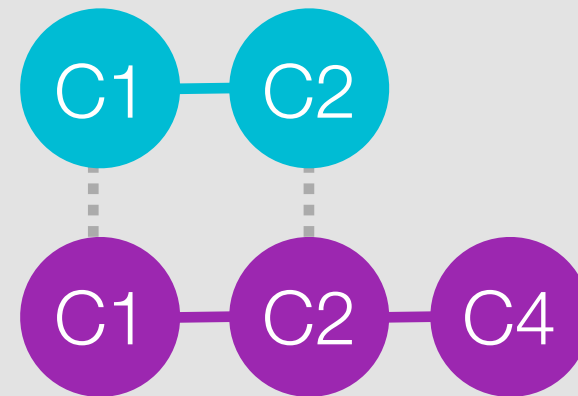They both commit new code to each of their branches.

GitHub

Alice

Bob

Alice pushes the feature branch to GitHub for all to see.
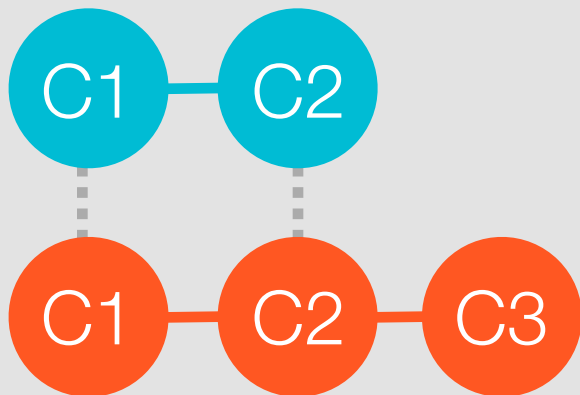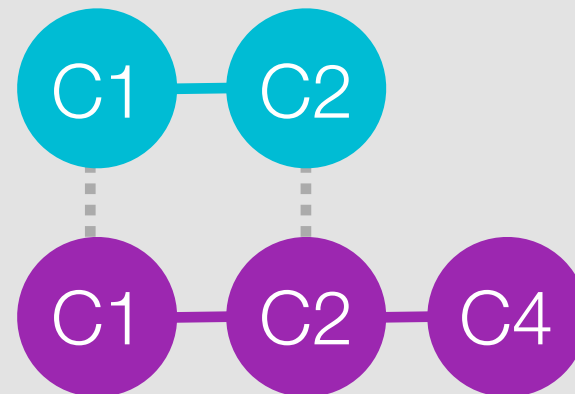
She requests a pull request.

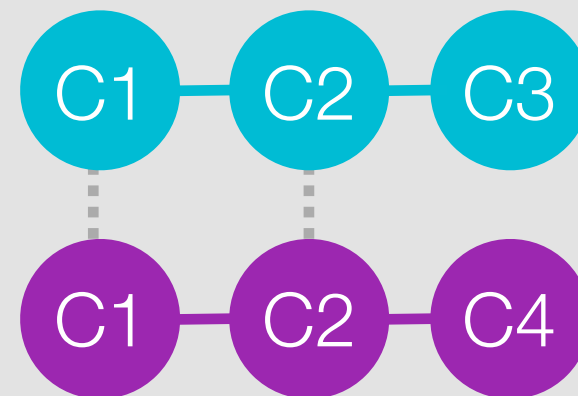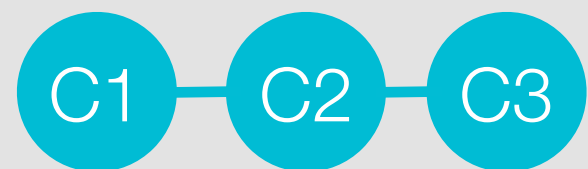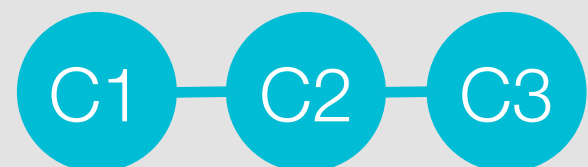The course staff merges Alice's changes with the master branch on GitHub.

Both Alice and Bob pull down the changes from master.

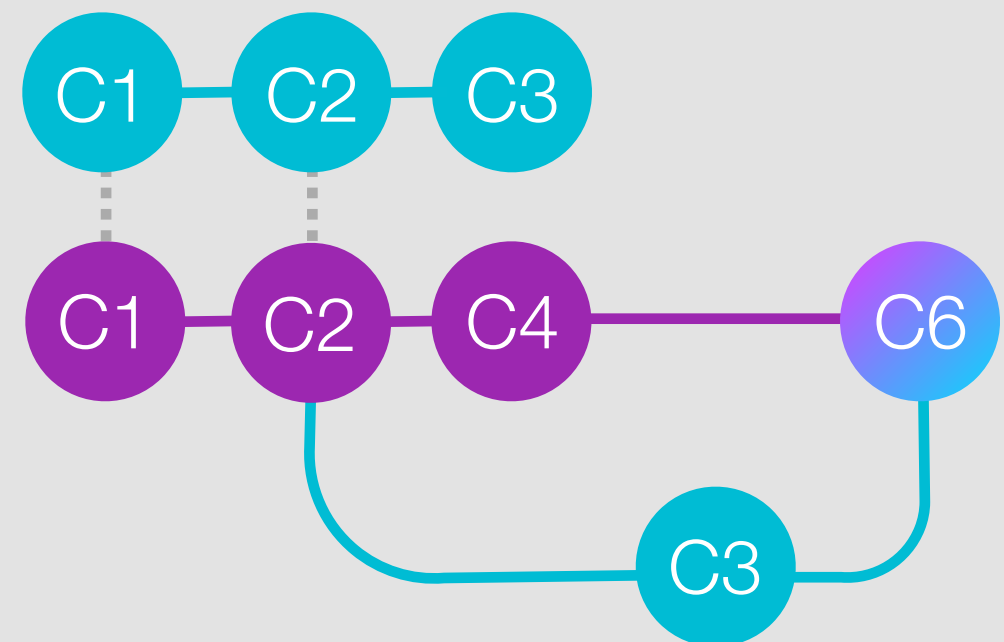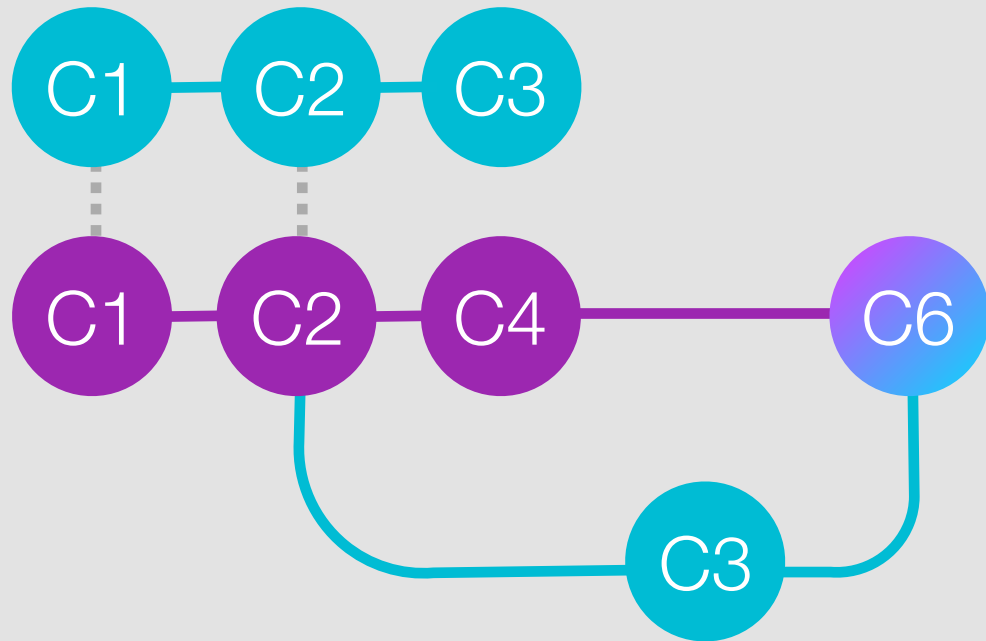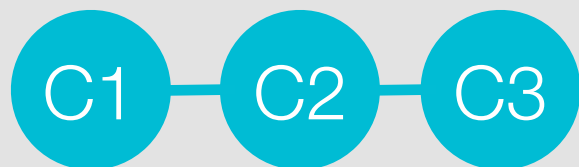Bob merges master into his feature branch.

GitHub

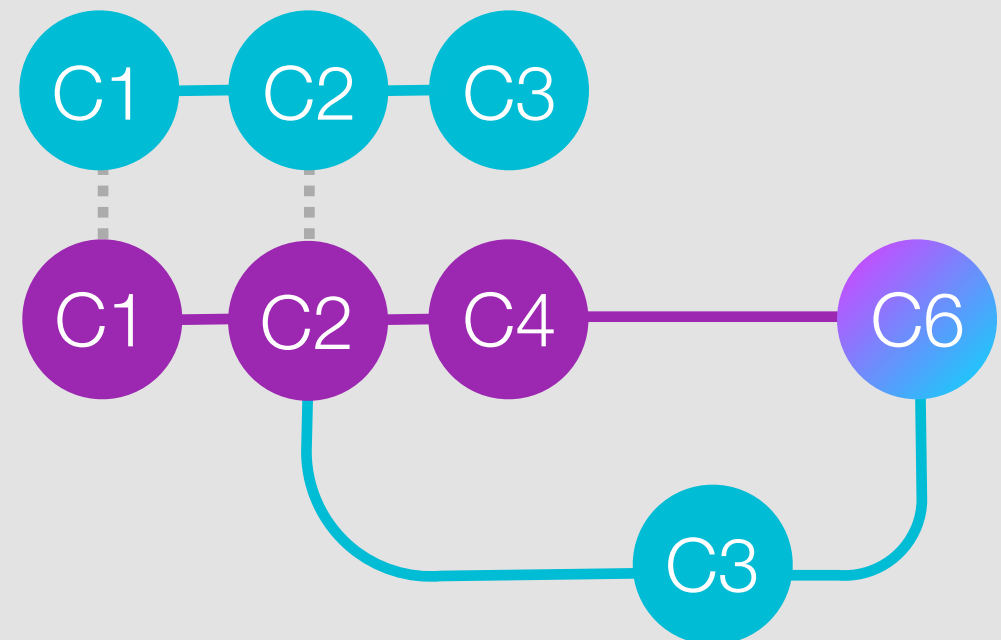Alice

Bob

Bob pushes his feature branch to master and asks for a PR.