Collaborating Using the GitHub Flow

Gill Cleeren

CTO Xpirit Belgium

@gillcleeren – xpirit.com/gill

Overview

Working with branches

Understanding branching strategies

Using the GitHub Flow

Forking a repository

Working with Branches

Branching traditionally means diverging from the main development line

Expensive operation

   – Copy of source directory

Branching in Git

Much more lightweight
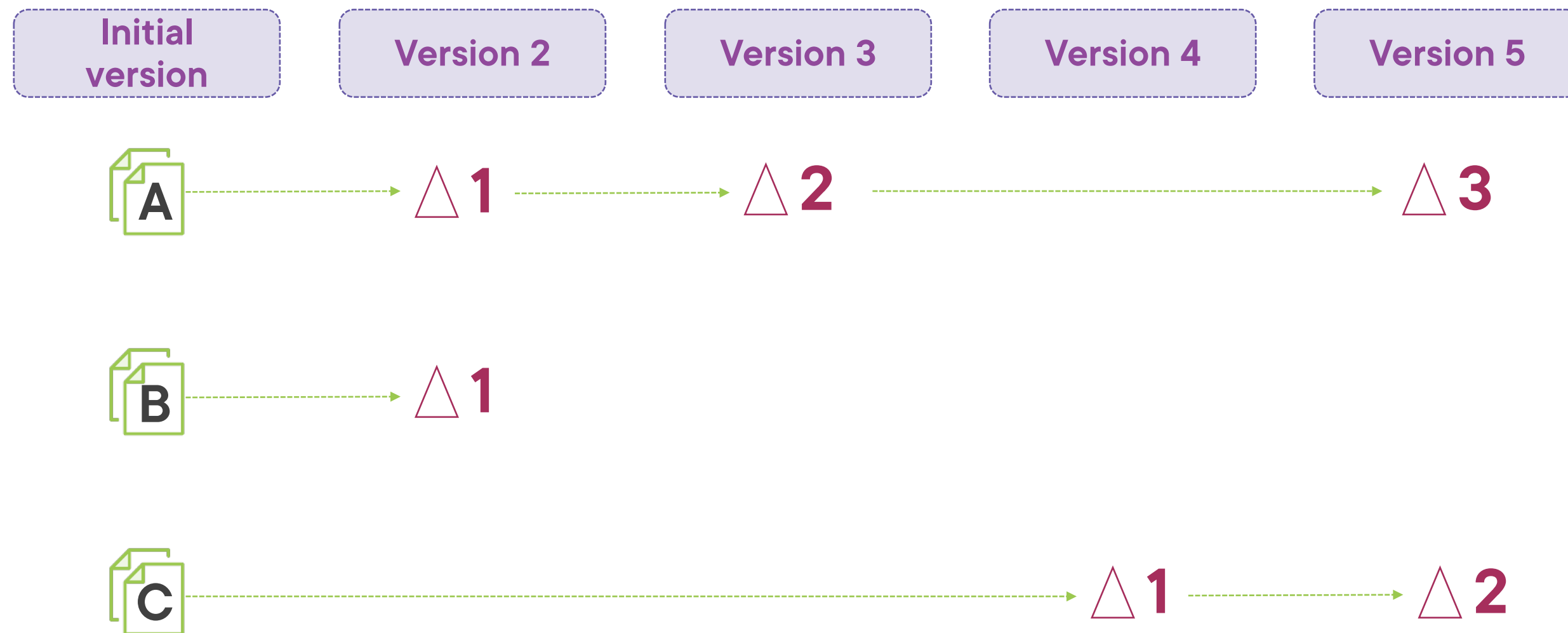
Branching is very fast
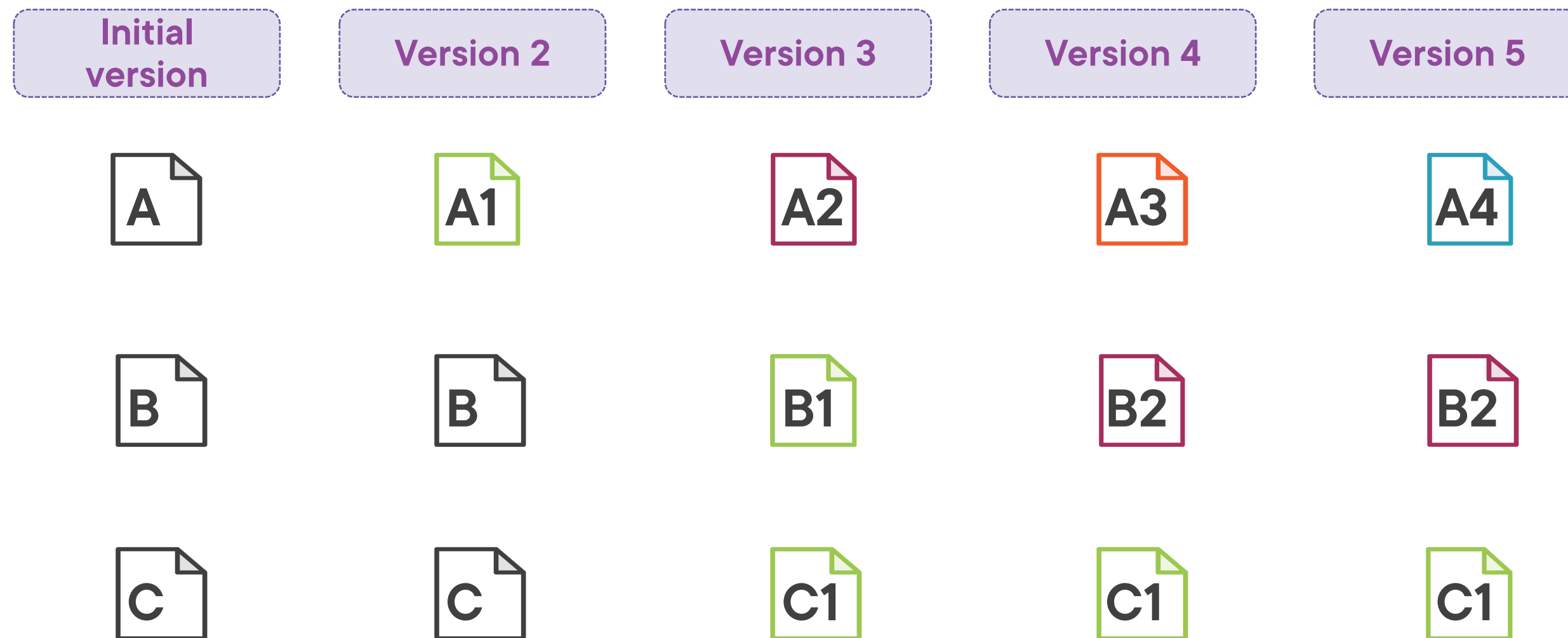
Encouraged to be used
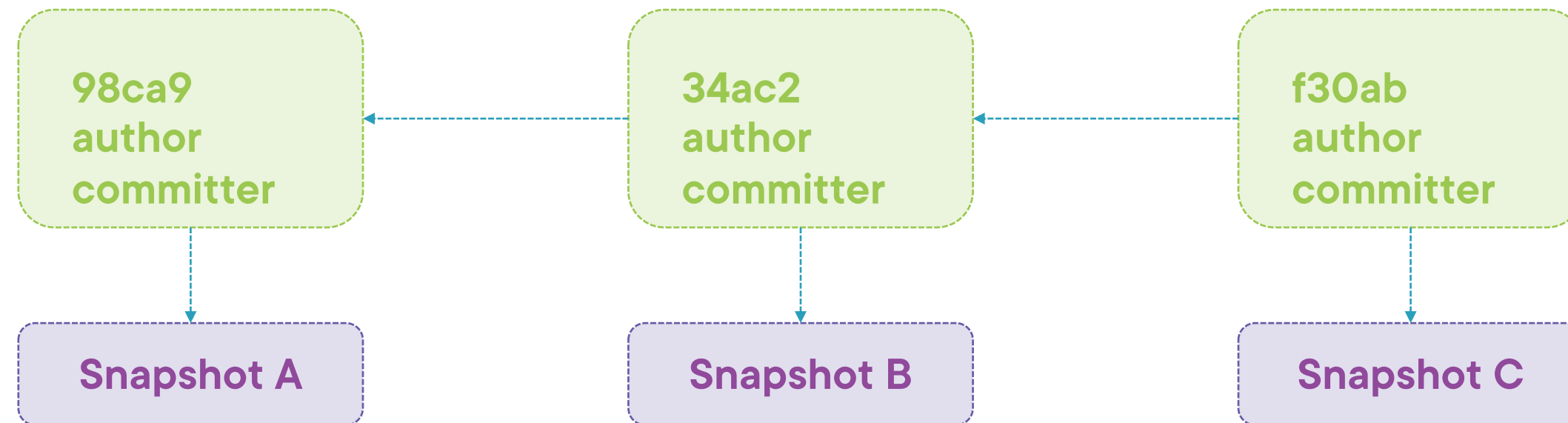
Works because of the way Git works
- Snapshots

Traditional Source Management

The Concept of Snapshots in Git (and GitHub)

Initial version	Version 2	Version 3	Version 4	Version 5
A	A1	A2	A3	A4
B	B	B1	B2	B2
C	C	C1	C1	C1

Commits in Git

Branching in Git

Branch is a pointer to a commit

Default branch is main
- Points to latest commit
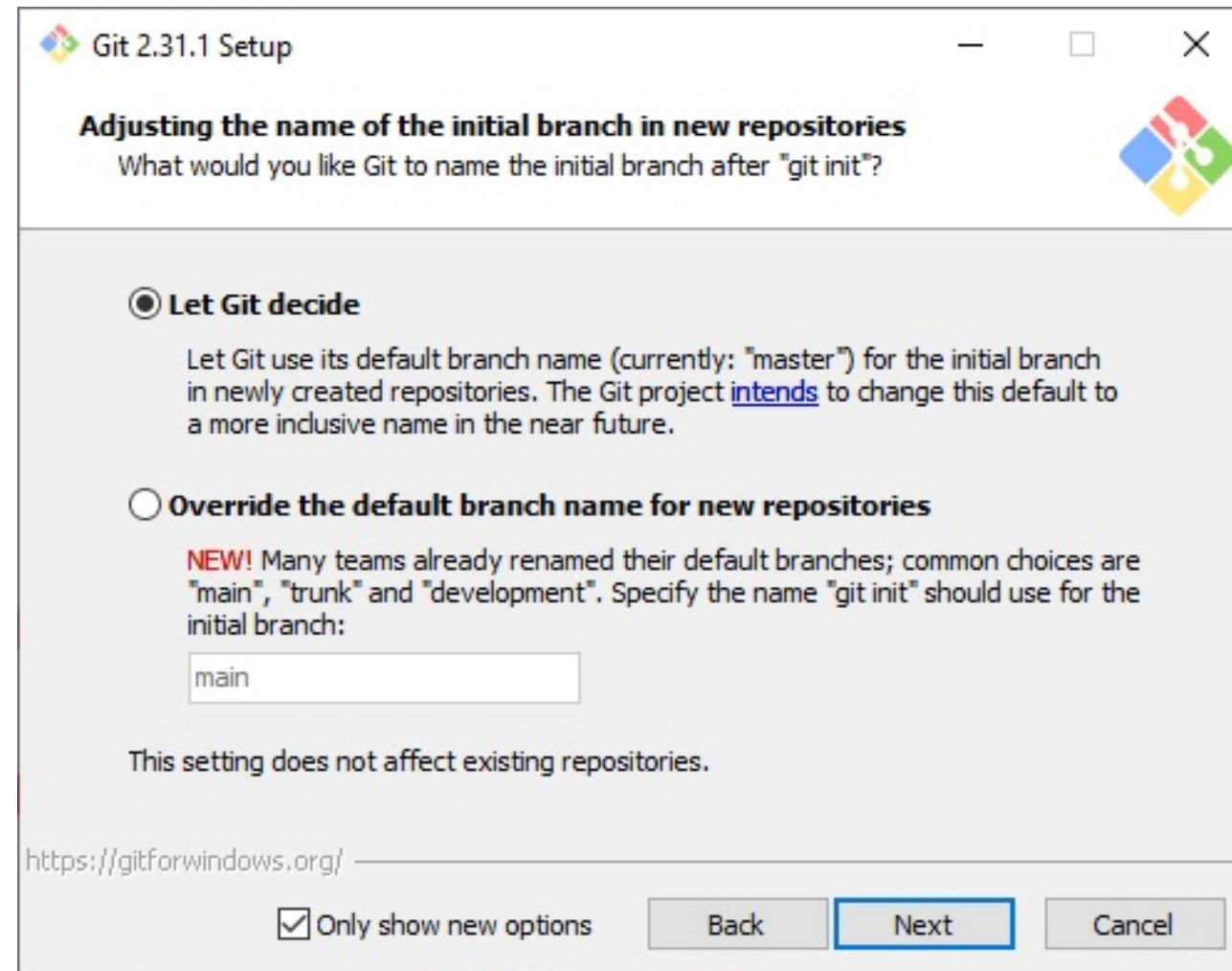- Moves forward with every commit

Use branches for
- Features
- Bugs
- Experiments

Branching in Git Continued
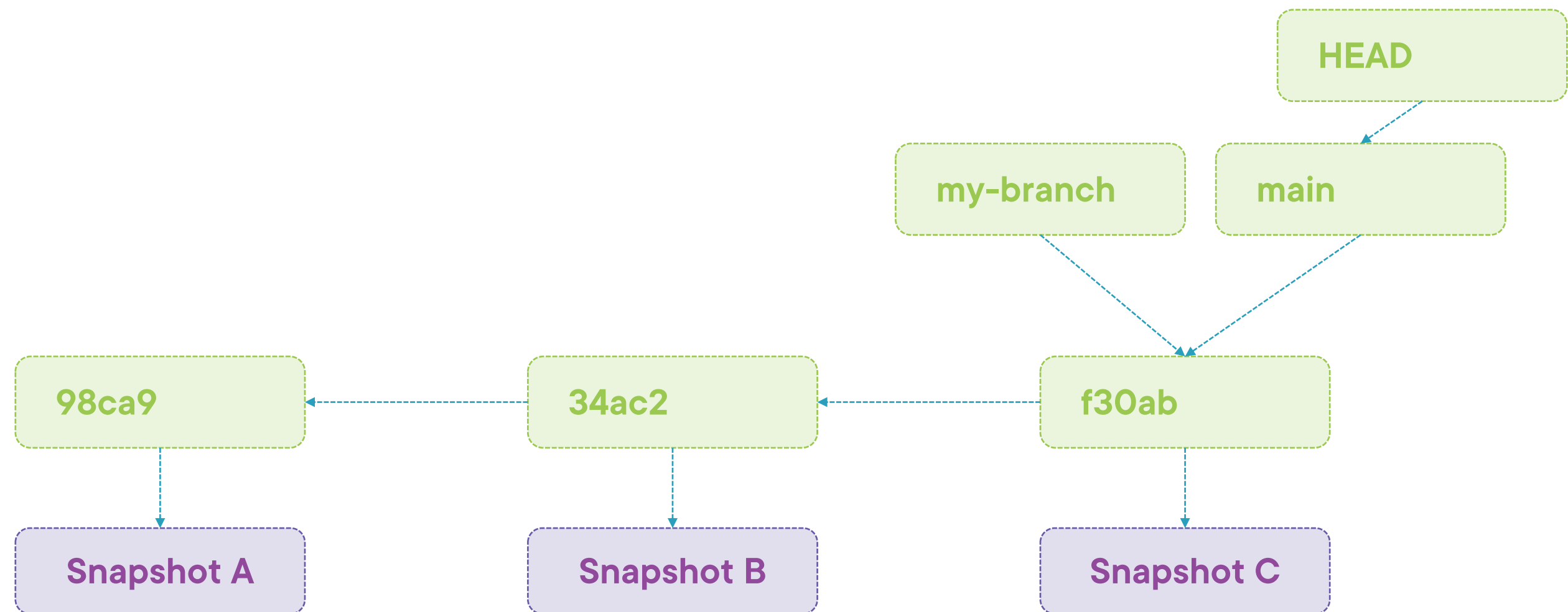
Default Branch Name

Demo

Creating a branch on GitHub

The Commands for Branching

```
$ git branch [branch-name]


$ git checkout [branch-name]


$ git push -u [origin] [branch]
```
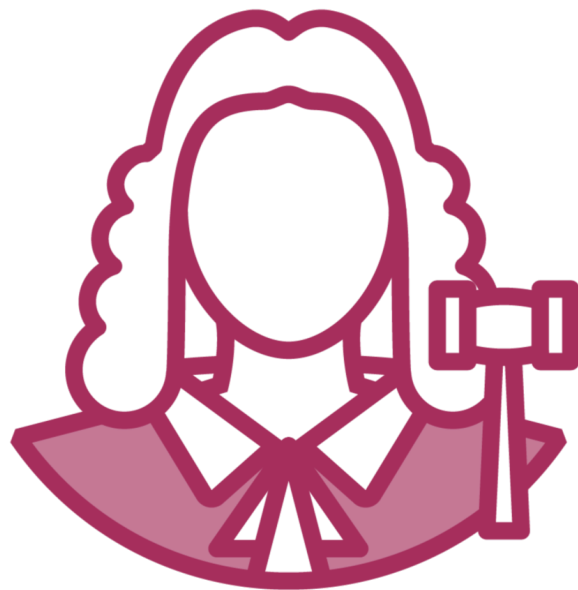
Demo

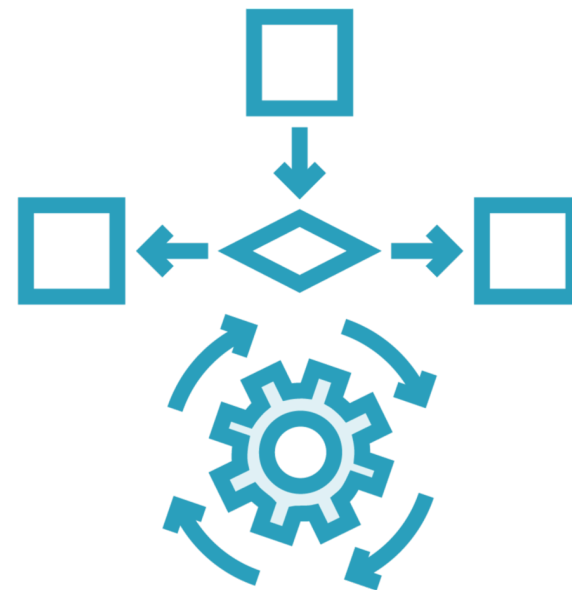**Creating local branches**

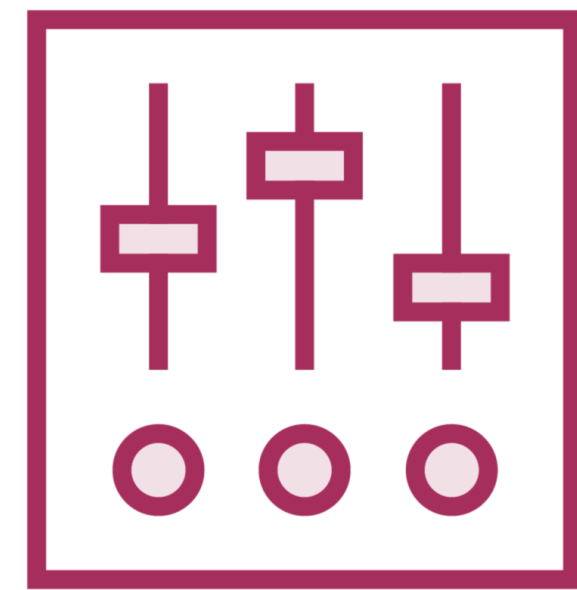**Pushing the branch to GitHub**

Understanding Branching Strategies

Defining a Branching Strategy

Set of rules

Define workflow for branches

Different options exist

Branching Strategies

Centralized workflow

Gitflow workflow
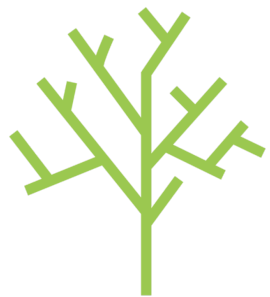
Forking workflow

GitHub workflow

Using the GitHub Flow

Understanding the GitHub Workflow
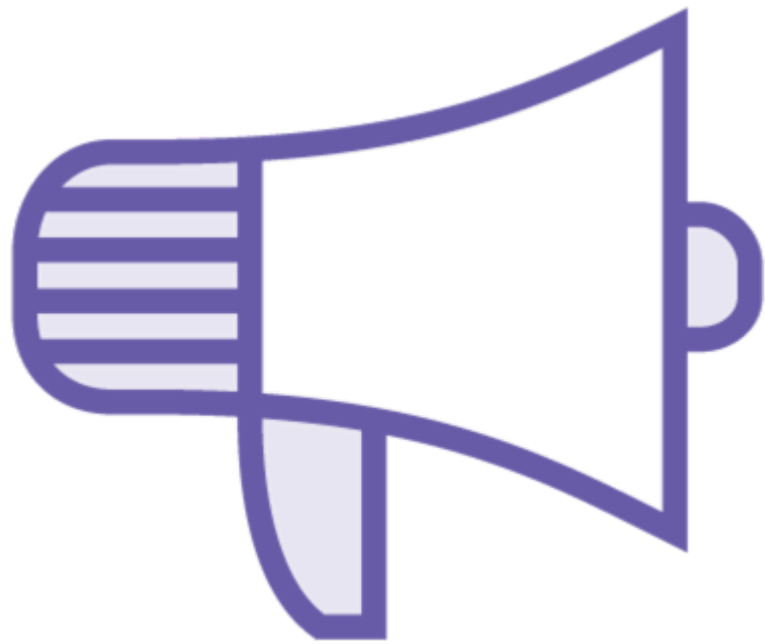
Lightweight

Favored by GitHub
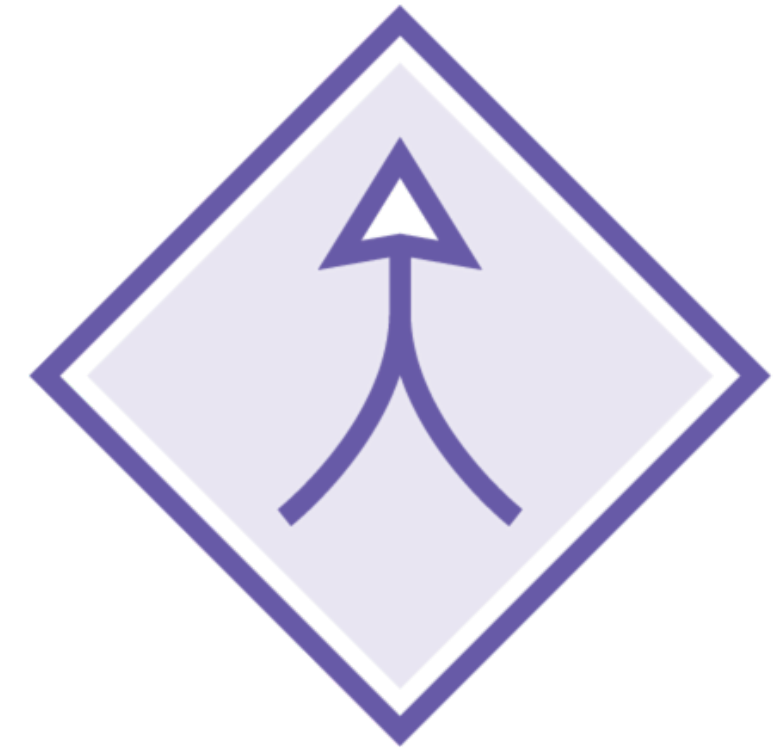
Branches for bugs, features...

Evolves around pull requests

Pull Requests

Announce a push to a branch

**Discuss
Review changes
Add more commits**

Merge

GitHub Support for Pull Requests

GitHub Flow

Commits → Pull request → Reviews

Demo

Submitting a pull request

Demo

Reviewing the pull request

Attributing the pull request

Sometimes, You'll Have a Merge Conflict

Add more commits by pushing to the **test-branch** branch on **GillCleeren/sample-repo**.

⚠ **This branch has conflicts that must be resolved**
Use the web editor or the command line to resolve conflicts.

Resolve conflicts

Conflicting files

README.md

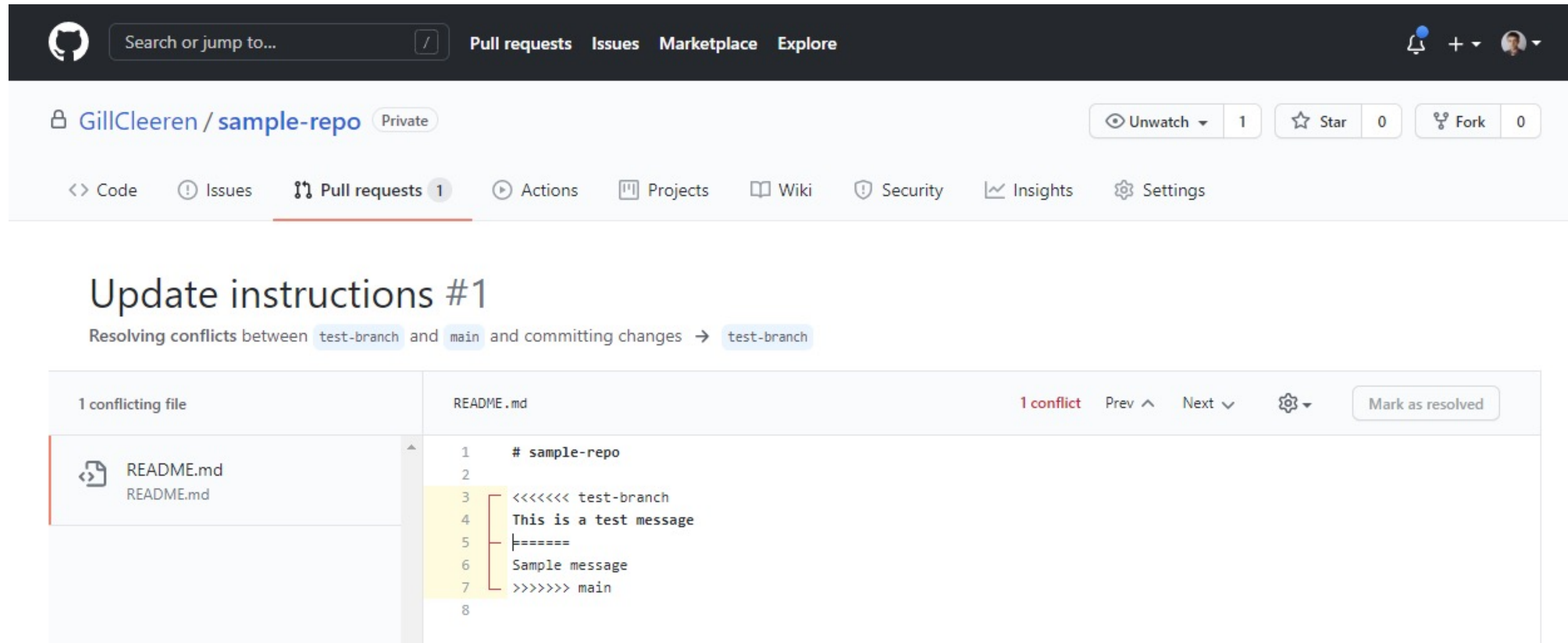**Merge pull request** ▾   You can also open this in GitHub Desktop or view command line instructions.

Typical conflicts

– Editing on the same line

– Editing an already deleted file

Merge conflicts needs to be solved before a merge can happen

Merging Inside GitHub

Local Diff Tools

KDiff3

vimdiff3

P4Merge

Beyond Compare

Demo

Merging with conflicts using GitHub

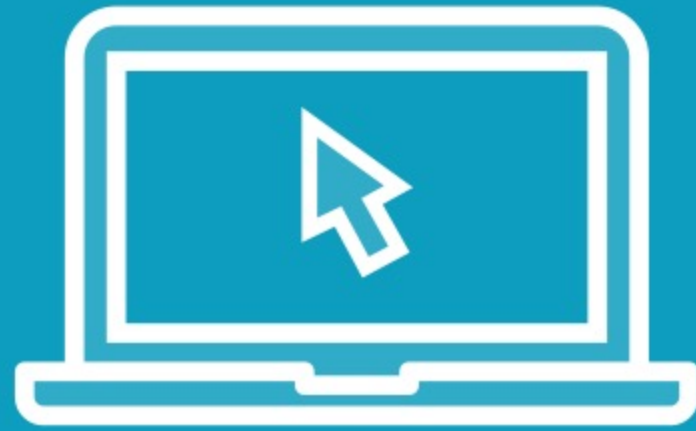**Merging locally**

Demo

Creating a PR template

Creating a PR based on a template

Demo

Reverting a merged PR

Exploring the history

Using the blame feature on GitHub

Demo

Setting the default branch

Restoring branches

Forking a Repository

Copy of a repo

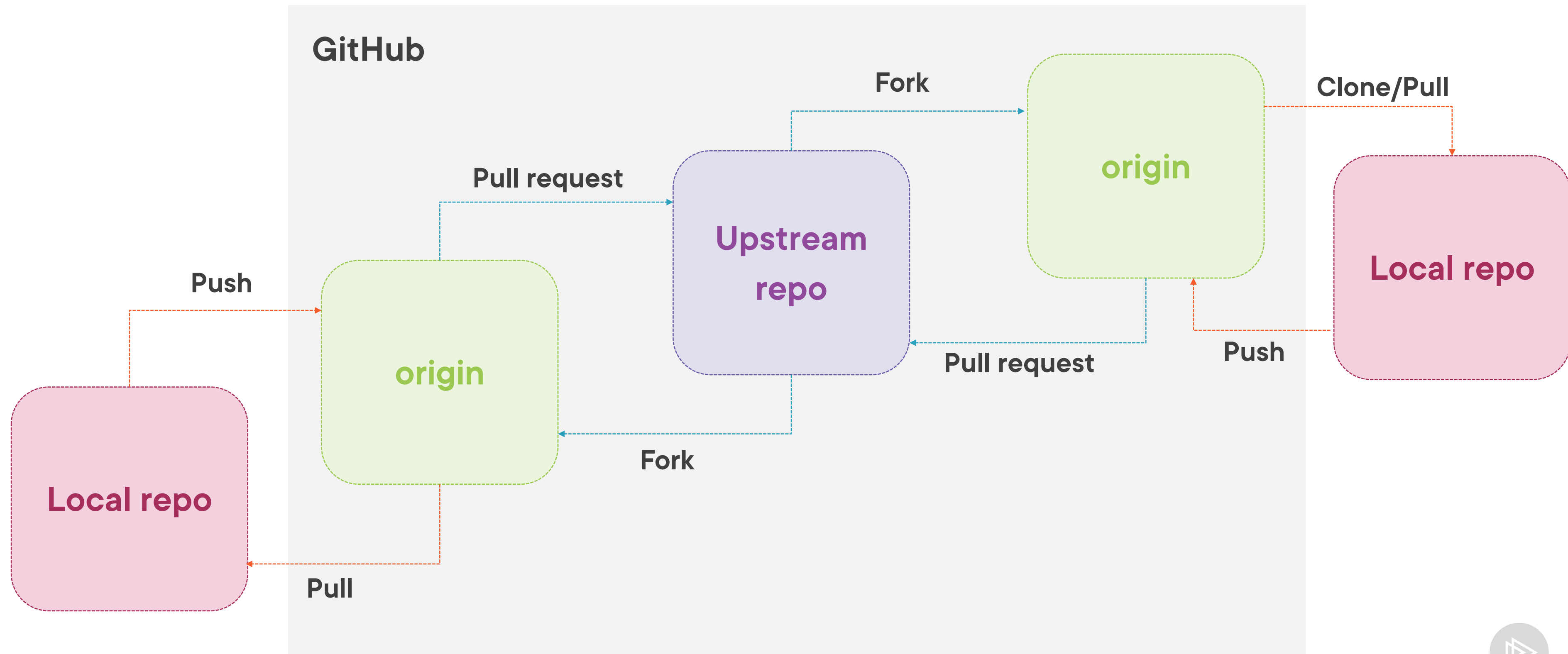**Doesn't impact the original repo**

Changes can be merged again via pull requests

Available to do for anyone

- It's your repository now

Forking on GitHub

Demo

Forking a repository on GitHub

Making changes in your own branch

Summary

Git has baked-in support for easy branching

GitHub Flow is based on branches and pull requests

Emphasizes the collaborative aspect

Forking creates a copy of a repository

Up next:
Managing issues