COMP1531

👯 Coding Together - Git - Team Usage

Lecture 1.4

Author(s): Hayden Smith, Dr. Yuchao Jiang

In This Lecture

- **Why?** 🤔
  - Git is primarily useful when working with others, and working with others effectively is important

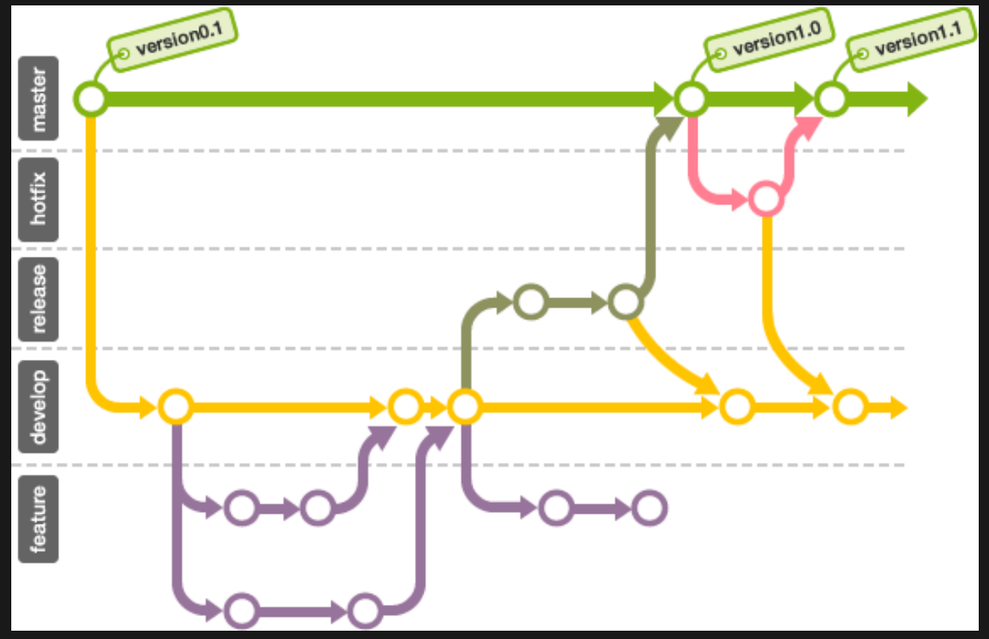
- **What?** 
  - Branching
  - Merging
  - Merge Requests

📺 Live Demo

Most of today's explanations will be covered via a live demo. If you want to follow a written guide, then please checkout Atlassian's git guide.

🌳 The Git Tree Model

- Git can be understood as a tree-like structure.
- Git is a collection of commits.
- Each **commit** has one parent. Each **commit** can have multiple children (i.e. **branches**)
- A **branch** essentially is just a pointer to a particular commit.
- To try and bring two separate **branches** together onto the same commit is a process of "**merging**"



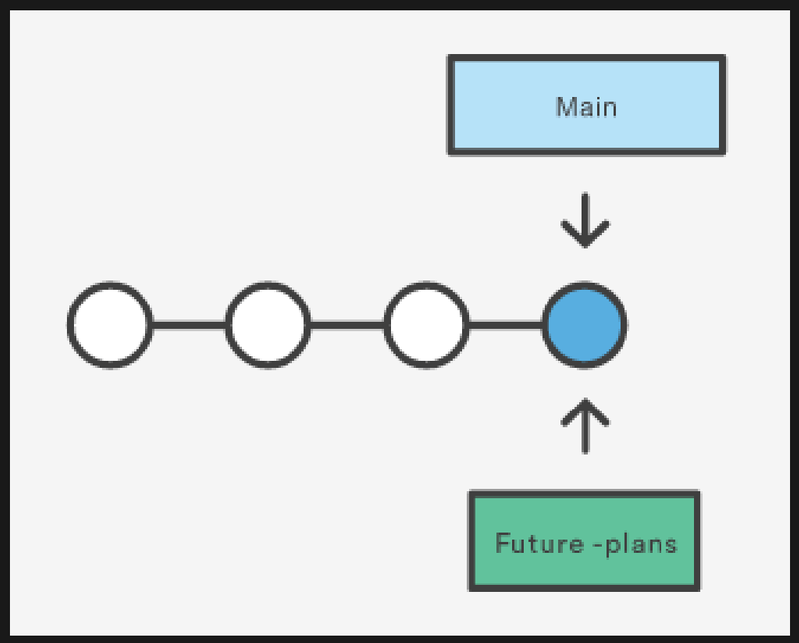
Source: https://github.com/frappe/charts/issues/180

 Branches

Your "master" branch is just a pointer to a particular commit on the tree (usually the latest).

You can create your own branch if you want to continue on a separate thread of working, unrelated to the master branch.

```
1 git checkout -b new_branch_name
```

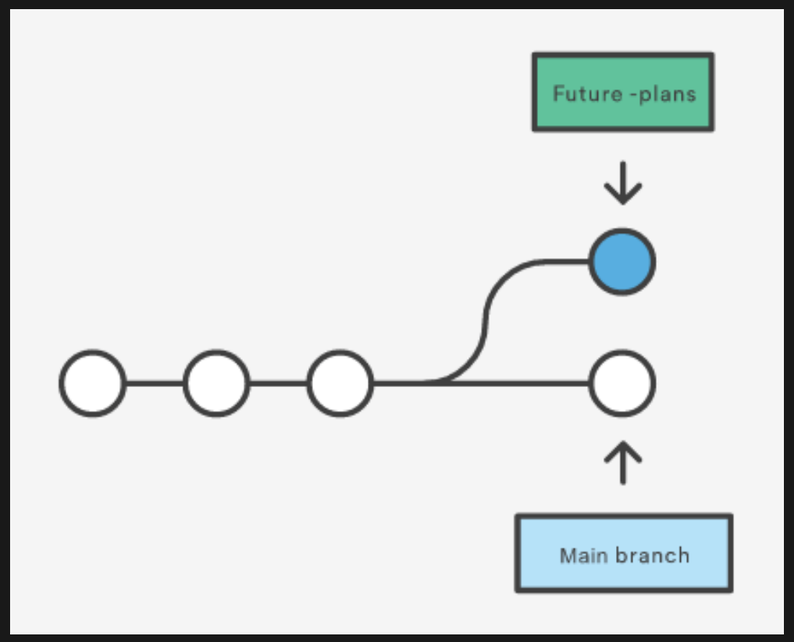


Source: Atlassian Git Guide

Branches

This then allows you to continue making commits on a separate "branch".

There is no limit for the number of branches you can have in a repository.



Source: Atlassian Git Guide

Branches

Your local repository can only "check out" (work with) a single branch at a time. You can swap between branches using the checkout command.

It's generally good practice to ensure you have no staged or unstaged changes on your branch before swapping to another.

```
1 git checkout branch_to_swap_to
```

💥 Merging

The process of "incorporating work on another branch into mine" is known as merging. The two most common cases of merging you'll see are:

- Merging master into your work whilst you develop on it (so you've integrated small changes often, rather than a big change suddenly)
- Merging your work into master once your branch is stable enough to merge into master

The merge command let's you **specify the branch you want merged into your current branch**.

```
1 git merge master
```

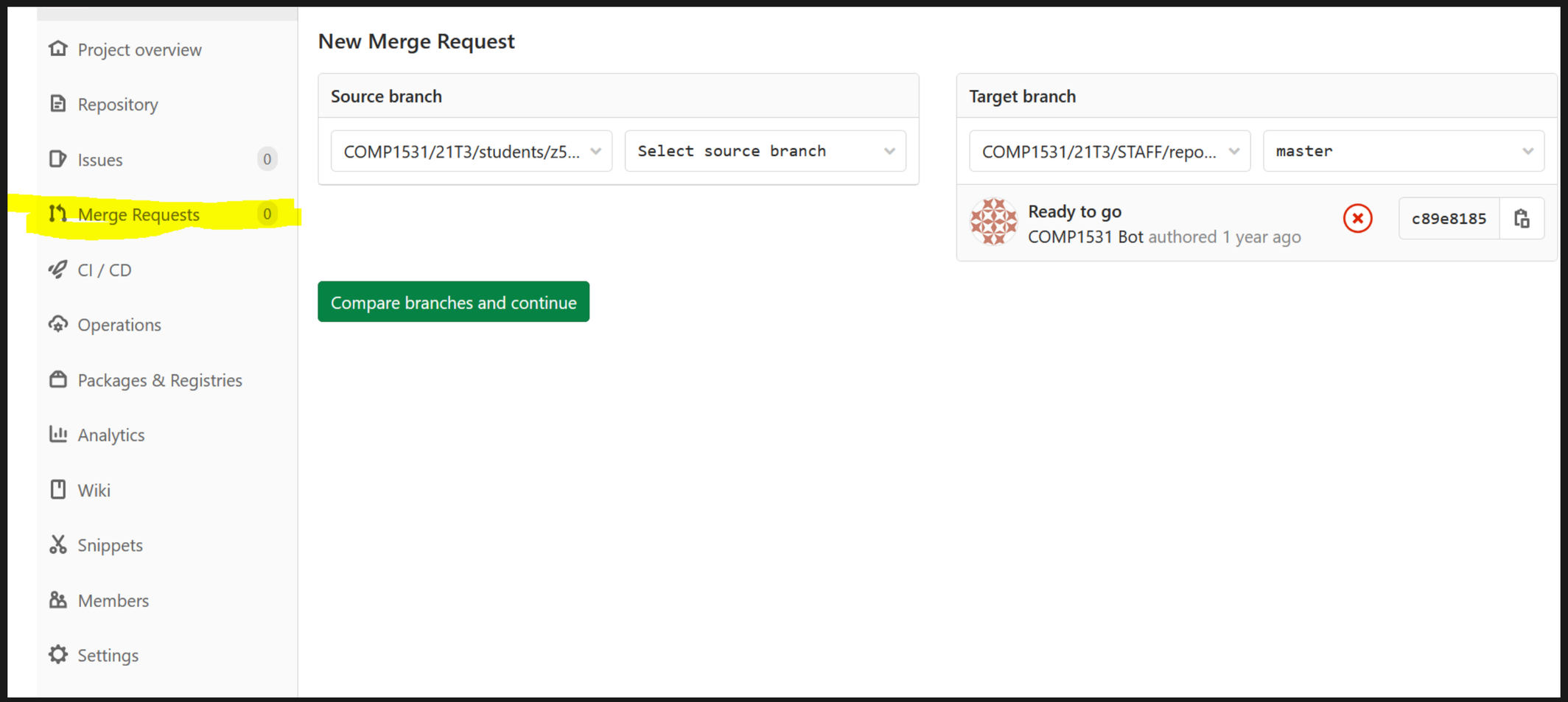
💥 Merging

The following describe a scenarios of scenarios with respect to merging between your working branch and master

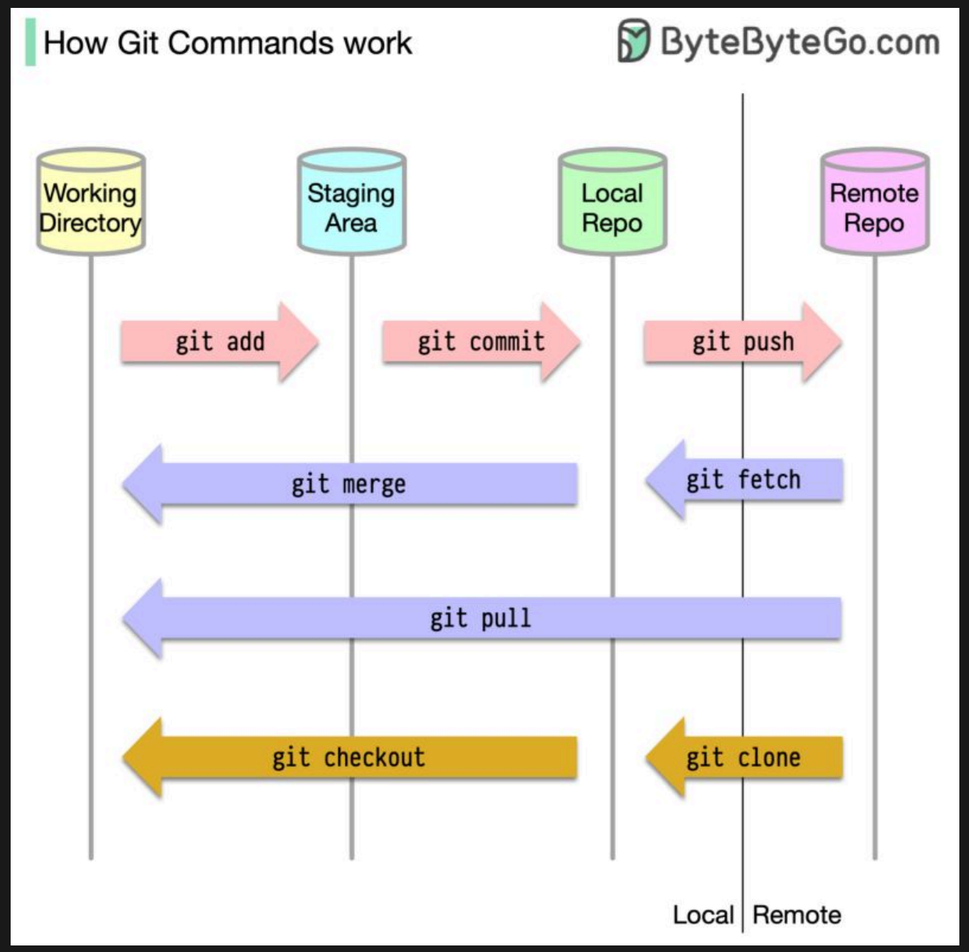
#	Commits made on your branch	Commits made on master branch	Command & Outcome
1	Yes	No	Nothing to do
2	No	Yes	from your branch, git merge master Will "fast forward" merge (i.e. simply bring your branch pointer to the same commit as master, effectively no merge)
3	Yes	Yes	from your branch, git merge master Will merge master into your branch, but a merge commit will get made (either automatically or manually)
5	Yes	Yes	from master branch, git merge your_branch Will merge your branch into master, but a merge commit will get made (either automatically or manually)

🙋🏿 Merge Requests

In most industries, you cannot merge your branch into master via the command line. Instead, we allow our git site (e.g. gitlab) to do this via a Merge Request (a web-based GUI that helps manage merges into master)



🙋🏿 How Git Commands Work



👂🏼 Feedback

Or go to the form here.