Remote Repository (GitHub)

Table of Contents

▶ Remote Repository (GitHub)

▶ Cloning a Remote Repository

▶ Remote Repo Operations

▶ Pull Request

CLARUSWAY©
WAY TO REINVENT YOURSELF

1 Recap- Git Workflow

Recap-Basic Commands

git init

git status

git add .

git rm --cached

git commit -m "abc"

git log

git checkout commitID

Local Machine

GitHub

Working directory

Staging area (index)

Git repository

Remote Git repository

git add

git commit

git checkout

CLARUSWAY©

WAY TO REINVENT YOURSELF
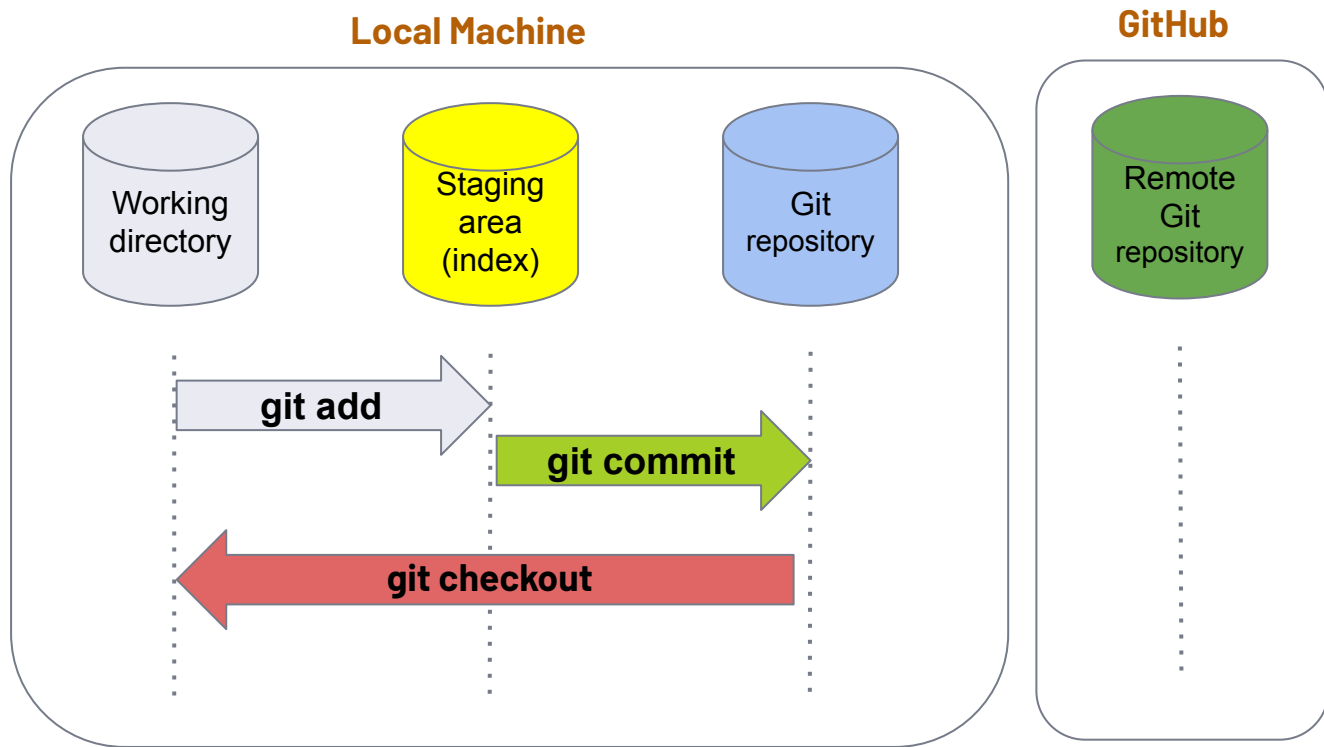
Recap-Branches

In Git, a branch is a new/separate version of the main/master repository.

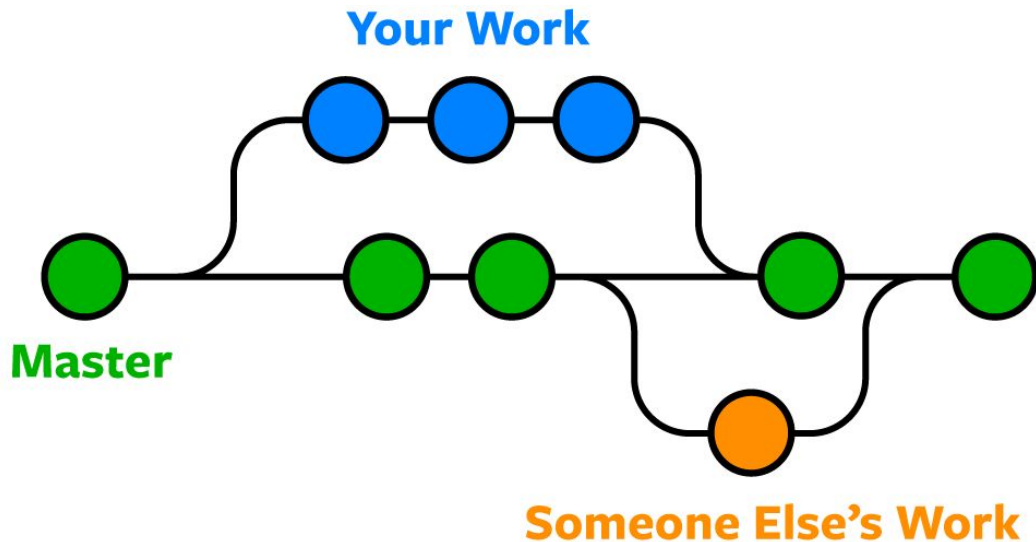git branch branch_name

git branch

git branch -r

git branch -a

git checkout branch_name

git checkout -b branch_name

git branch -d branch_name

git branch -D branch_name
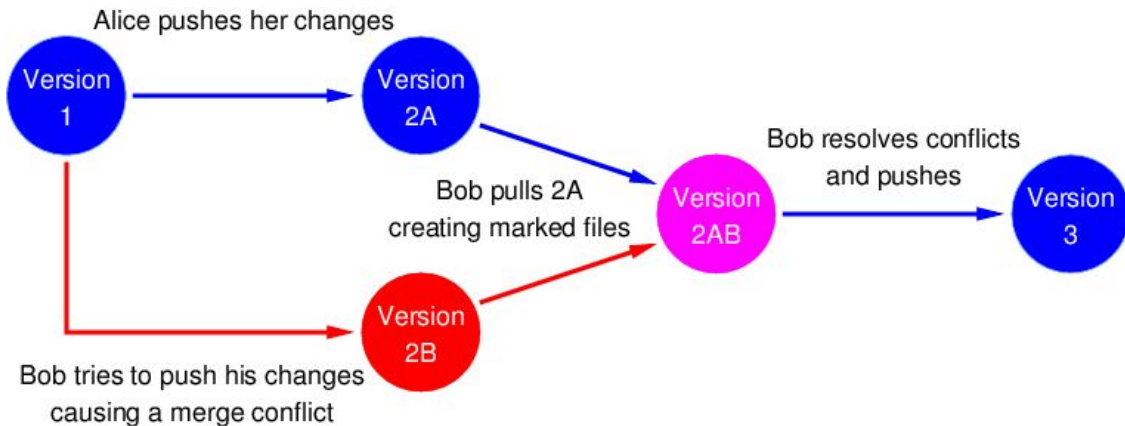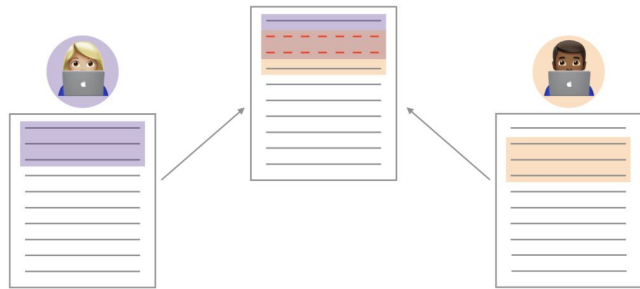
git merge branch_name

Merge Conflicts

Merge conflicts happen when you merge branches that have competing commits, and Git needs your help to decide which changes to incorporate in the final merge.

Same files were edited in both branches

2 Remote Repository (GitHub)

Github - Remote Repository

Git

&

GitHub

Distributed

version-control system

Repository hosting service

Github - Remote Repository

Bitbucket

+ Follow   + I use this

Stacks	Followers	Votes
25.8K	19.2K	2.8K

GitHub

+ Follow   + I use this

Stacks	Followers	Votes
132.1K	99.8K	10.1K

GitLab

+ Follow   + I use this

| Stacks	Followers	Votes
30.5K	23.4K	2.3K

Github - Remote Repository

GitHub vs GitLab vs BitBucket Interfaces

Generally, developers agree that GitLab and GitHub have the best interface. Many developers prefer GitHub because it's more popular and they are used to the navigation. However, teams using Jira benefit from having seamless integrations with the Atlassian ecosystem – so BitBucket is a natural choice.

So, here's a short summary of each platform's interface:

- **GitHub**: a lot of guides and tutorials, basic functionality can be accessed quickly, the interface is minimalistic and easy-to-understand. The downside is the lack of native integrations with Jira and a lack of advanced CI features.
- **BitBucket**: the interface can be confusing, but as the platform becomes more popular, there are a lot more resources. Teams who use Jira often prefer BitBucket because it provides much better visibility of the overall project.
- **GitLab**: the service isn't as established as competitors', so interface guides and tutorials are still lacking. However, the interface itself is very well-organized – even advanced add-ons are easy to locate.
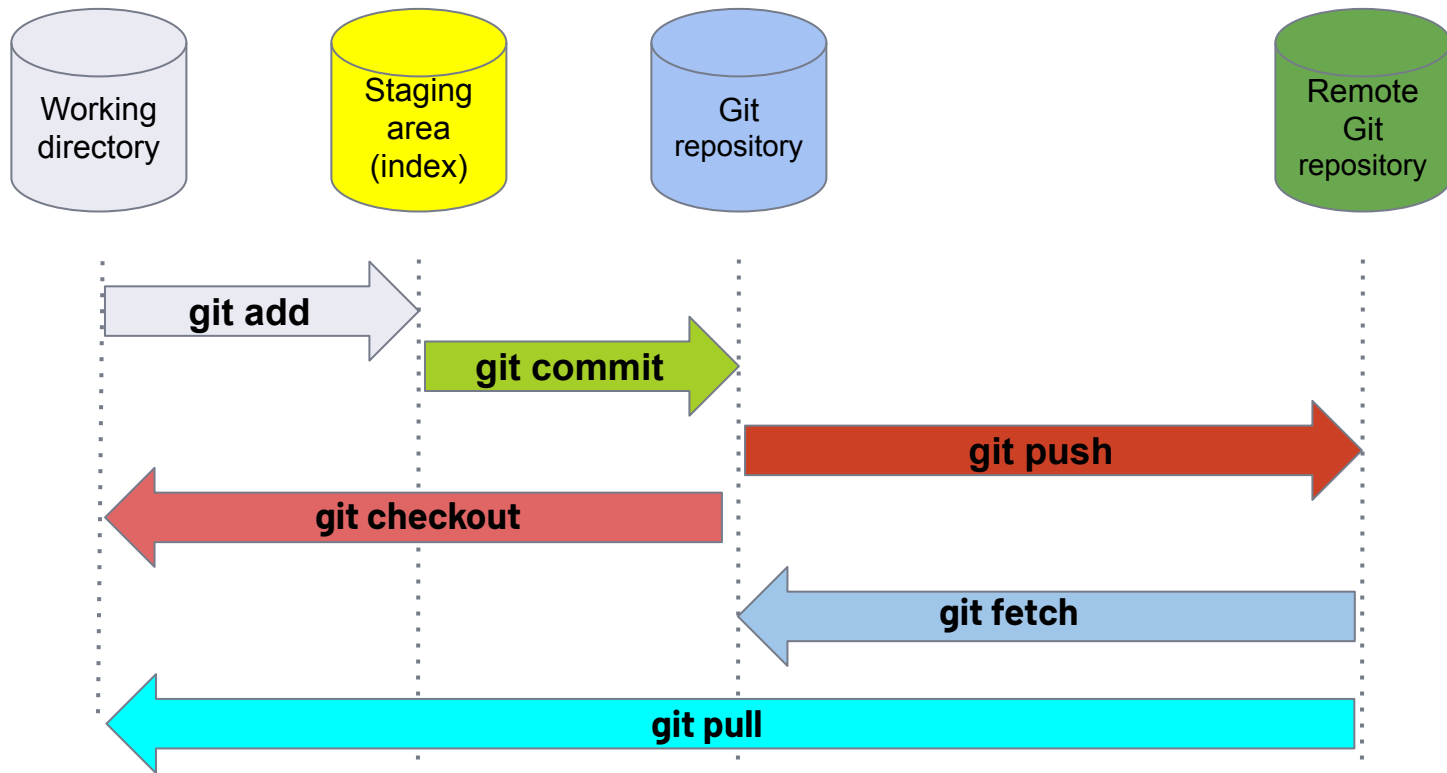
Github - Remote Repository

(default name - **origin**)
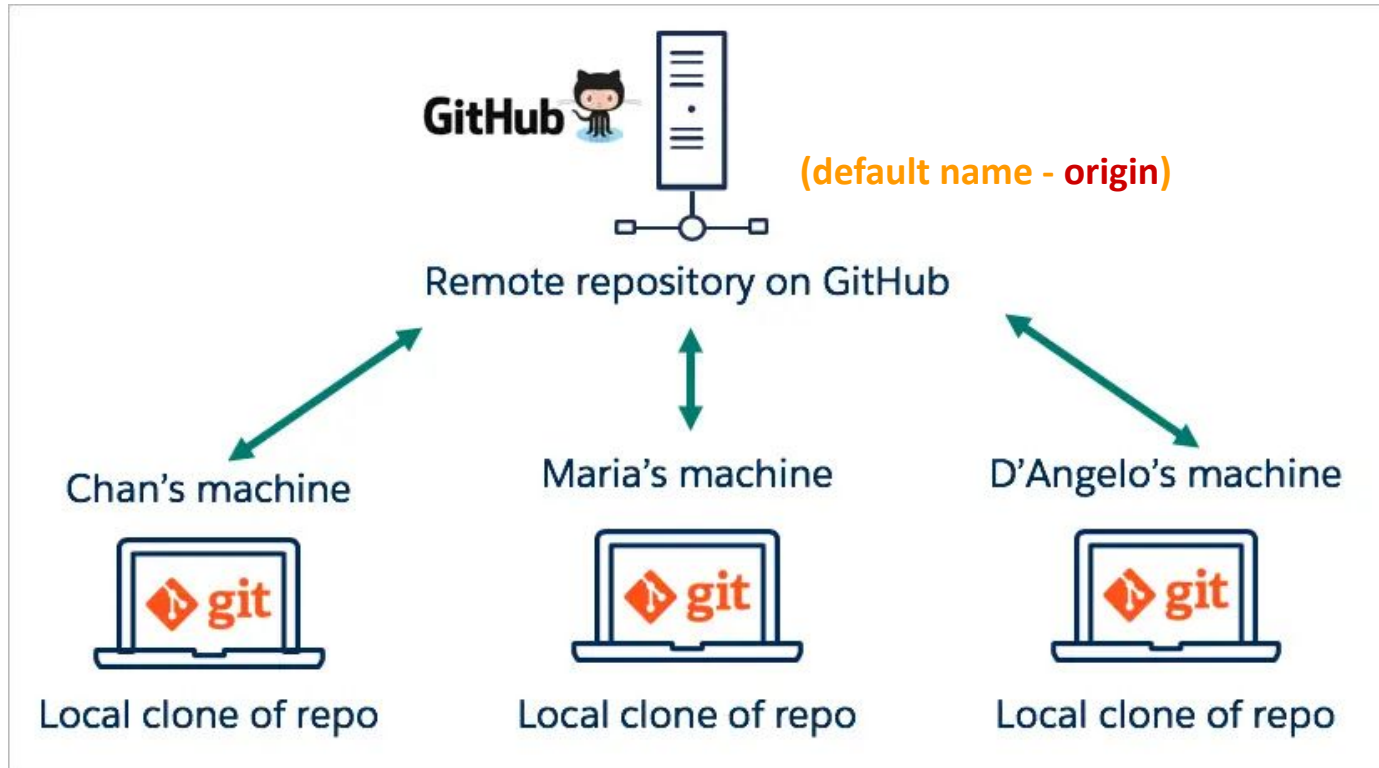
Github - Remote Repository

➔ Act of copying a repository from remote server to your local machine is called **cloning**

➔ Cloning allows team to work together

➔ Downloading commits from others : **fetch, merge**

➔ Downloading commits from others : **pull (fetch + merge)**

➔ Uploading your commits (local changes) to remote : **push**

Github - Remote Repository

Github - Remote Repository

Git fetch command helps the user download commits, refs, and files from the remote repository to the local repository.

Git merge command is the positive conclusion of your decision to incorporate the changes you saw using the Git fetch command. Let me straighten it out. Once the user is ready to accept the changes from the remote repository, they can merge these changes to the local repository.
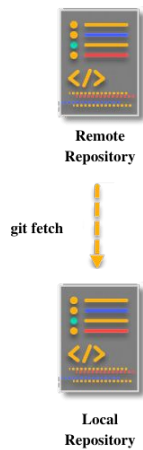
Git Fetch
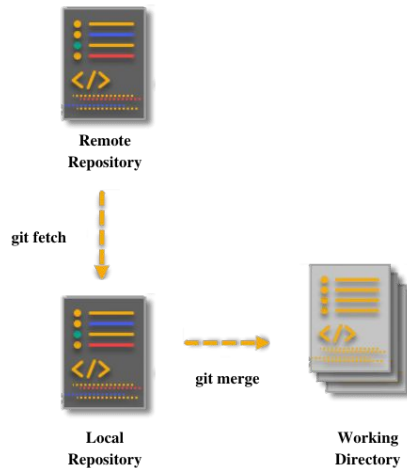
Git Merge

3 Pull Request

Github - Pull Request

➔ A pull request – also referred to as a merge request – is an event that takes place in software development when a contributor/developer is ready to begin the process of merging new code changes with the main project repository.

➔ PR is a Github's feature not Git's feature

➔ It allows you to contribute to other projects

**Create Branch**   **Create Pull Request**   **Merge Branch**
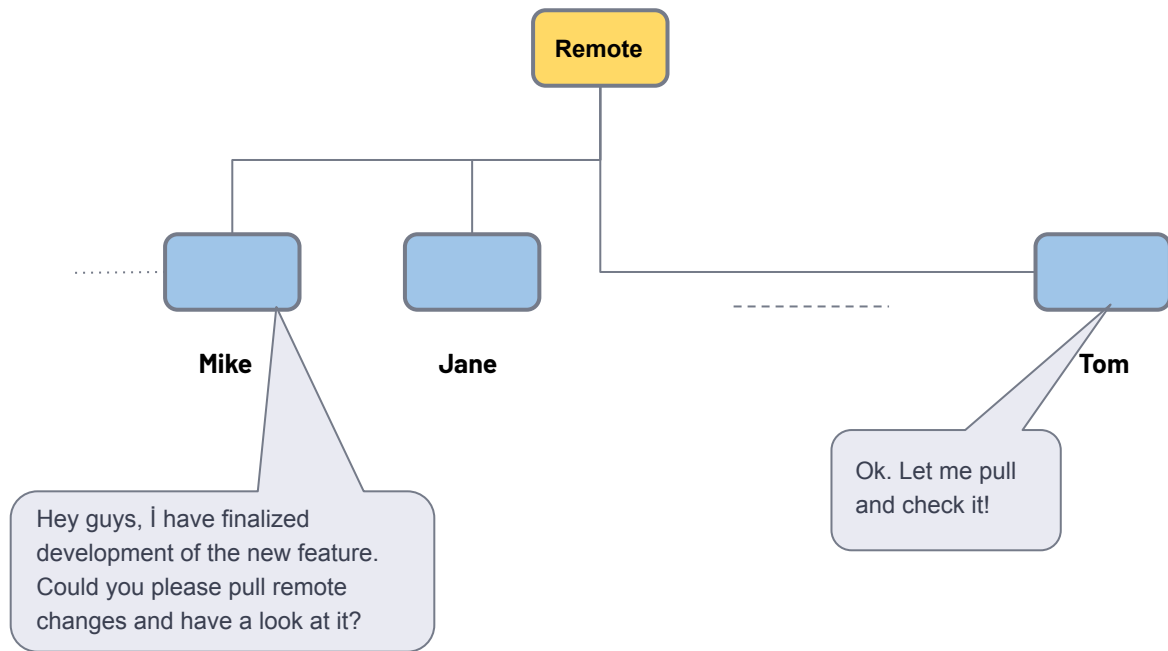
Github - Pull Request

➔ **Pull Requests (PR)** let you tell others about changes you've pushed to a branch in a repository on GitHub

➔ You create a pull request to propose and collaborate on changes to a repository. These changes are proposed in a branch, which ensures that the master branch only contains finished and approved work.

➔ *The importance of pull requests cannot be overstated*. They are an essential part of the software development process, helping to ensure that relevant parties review code changes before they are merged into the main codebase. This helps to avoid bugs and other problems that could potentially cause serious issues.

How collaborators communicate?
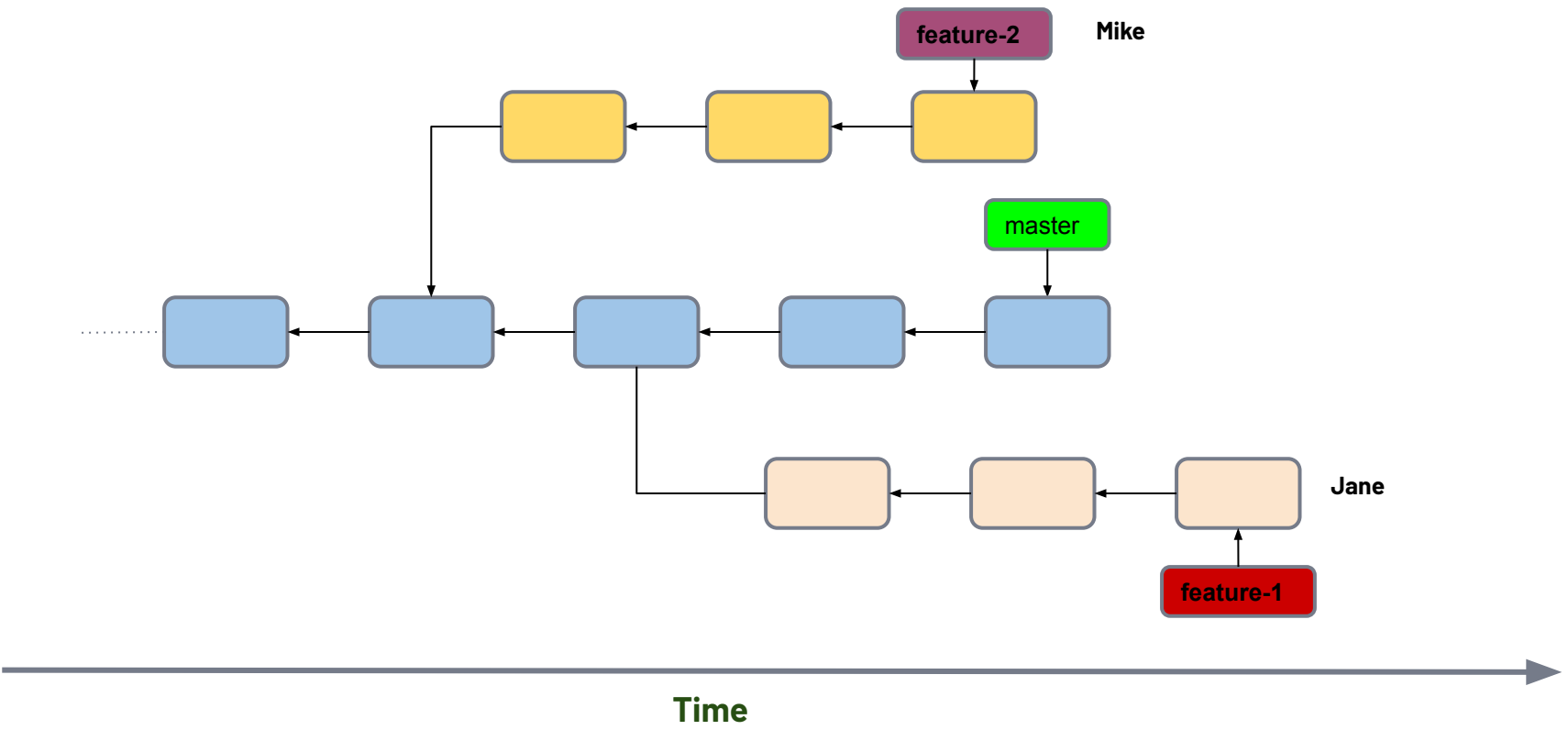
Why "pull" request?

Github - Pull Request Process

Time

Mike

Creates new local branch

feature-1

Commits changes to feature-1 locally

commit

Mike is happy with changes and feature works as expected

Pushes changes to remote by creating remote feature-1 branch

feature-1

Creates pull request to start review process by other collaborators

Mike requests Jane to review newly opened pull request

Jane

Jane starts review of the Mike's pull request

Optionally **pulls** updates and checkouts **feature-1** branch to verify how new feature works.

Add some comments for specific blocks of code and asks for changes

comments

Github - Pull Request Process

Time

Mike

Mike is notified about comments and requested changes

Makes additional changes requested by Jane

Pushes changes to remote

commit

Jane

Jane is notified about new commits

Happy with new changes and **approves** pull request

Mike

Merges changes from the feature-1 branch to the main **master** or **release** branch

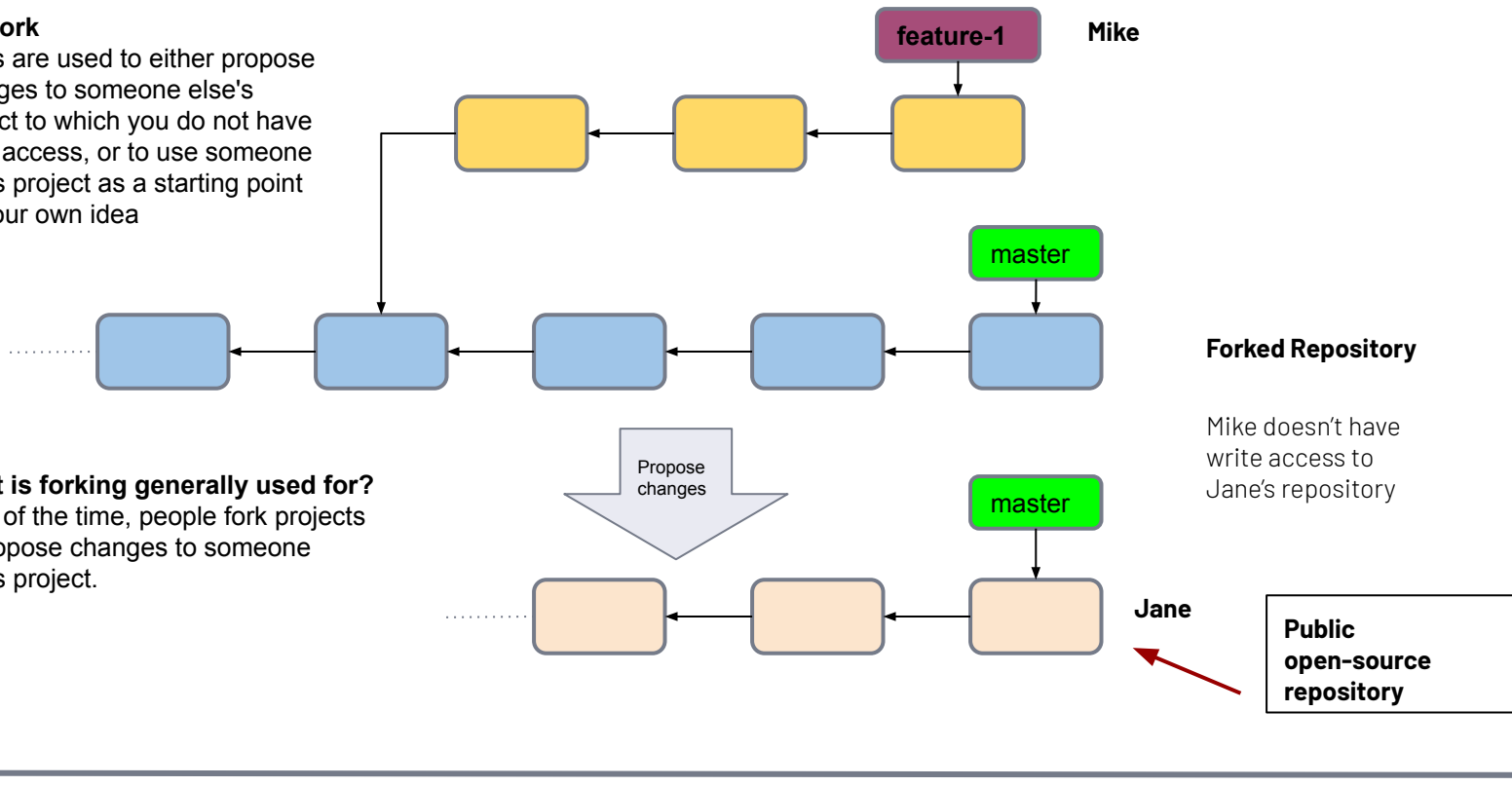Closes pull request and deletes feature-1 branch

New feature implemented !

CLARUSWAY©

WAY TO REINVENT YOURSELF

21

How collaborators communicate?

Git Fork
Forks are used to either propose changes to someone else's project to which you do not have write access, or to use someone else's project as a starting point for your own idea

What is forking generally used for?
Most of the time, people fork projects to propose changes to someone else's project.

feature-1   Mike

master

Forked Repository

Mike doesn't have write access to Jane's repository

Propose changes

master

Jane

Public open-source repository

Time

What is Origin in Git?

In Git, "origin" is a shorthand name for the remote repository that a project was originally cloned from. More precisely, it is used instead of that original repository URL - and thereby makes referencing much easier.

In the following example, the URL parameter to the "clone" command becomes the "origin" for the cloned local repository:

git clone https://github.com/gittower/git-crash-course.git

Connecting your local with remote

➜ connect to remote repo

git remote add origin Repo address

git remote -v

origin = alias for your repo address

➜ first push

git push -u origin master

➜ remove remote origin

git remote rm origin

What is Upstream in Git?

Like the literal meaning, upstream refers to something which points to where it originated from.

In the git world, upstream refers to the original repo or a branch.

For example, when you clone from Github, the remote Github repo is upstream for the cloned local copy.When you clone a Forked repository to your local, the forked repository is considered as the remote origin, and the repository you forked from is upstream.
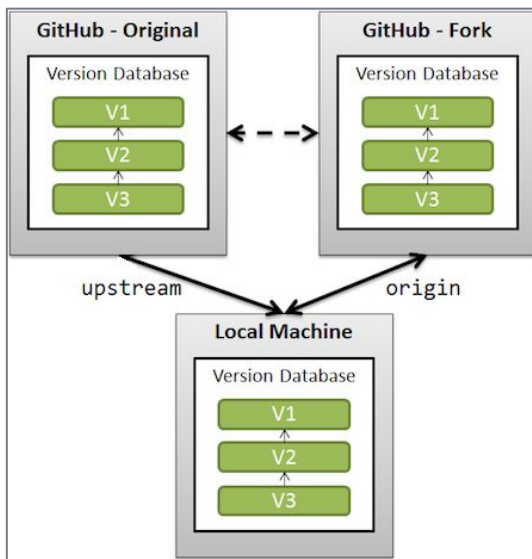
Git Repository Upstream

Whenever you clone a git repository, you get a local copy in your system. So, for your local copy, the actual repository is upstream.
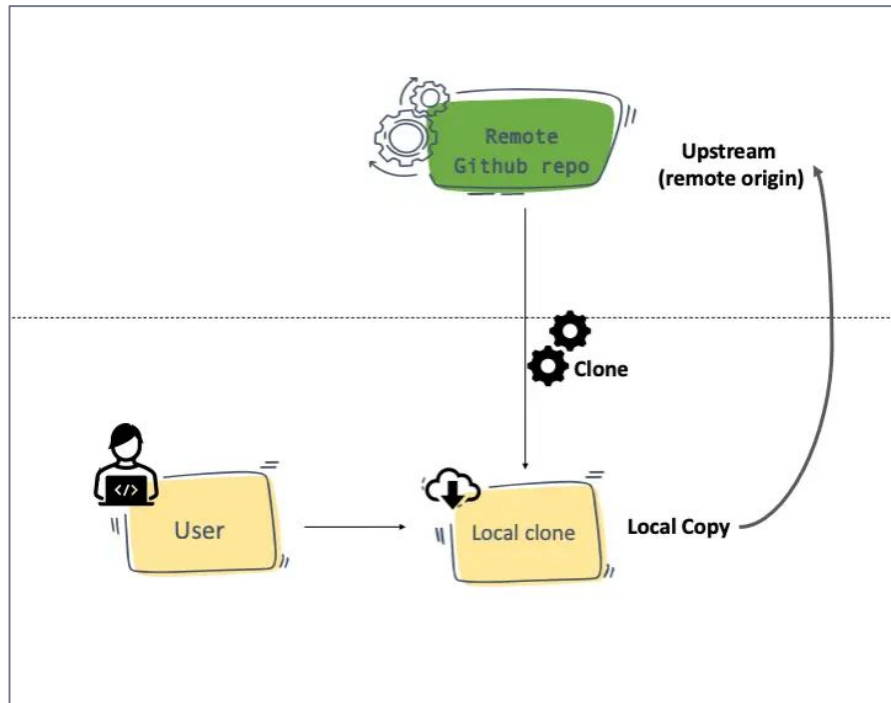
Git Forked Repository Upstream

When you clone a Forked repository to your local, the forked repository is considered as the remote origin, and the repository you forked from is upstream.

This workflow is common in open-source development and even in normal project developments.

For example, If you wanted to contribute to an open-source project, this is what you do.
- ➔ You would fork the actual project repository.
- ➔ Then you clone from the forked repository.
- ➔ Meanwhile, to keep up with the main open-source repository, you pull the changes from it through your git upstream config.
- ➔ You push your changes to a branch in the forked repository.
- ➔ Then you would raise a PR to the actual project repository from your forked repository.
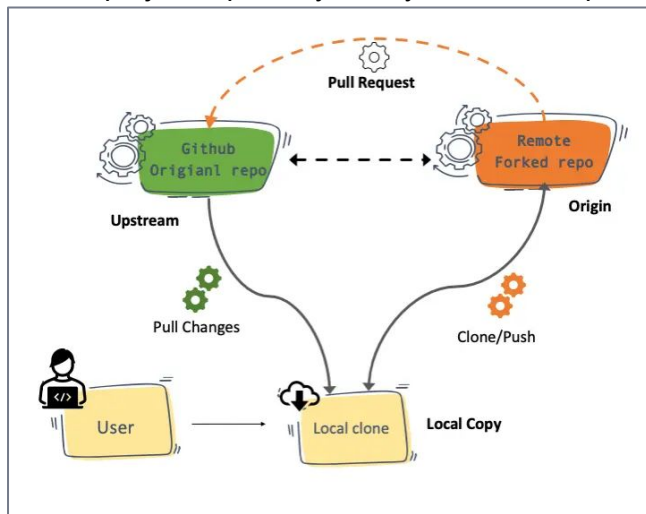
Git Basics - Summary
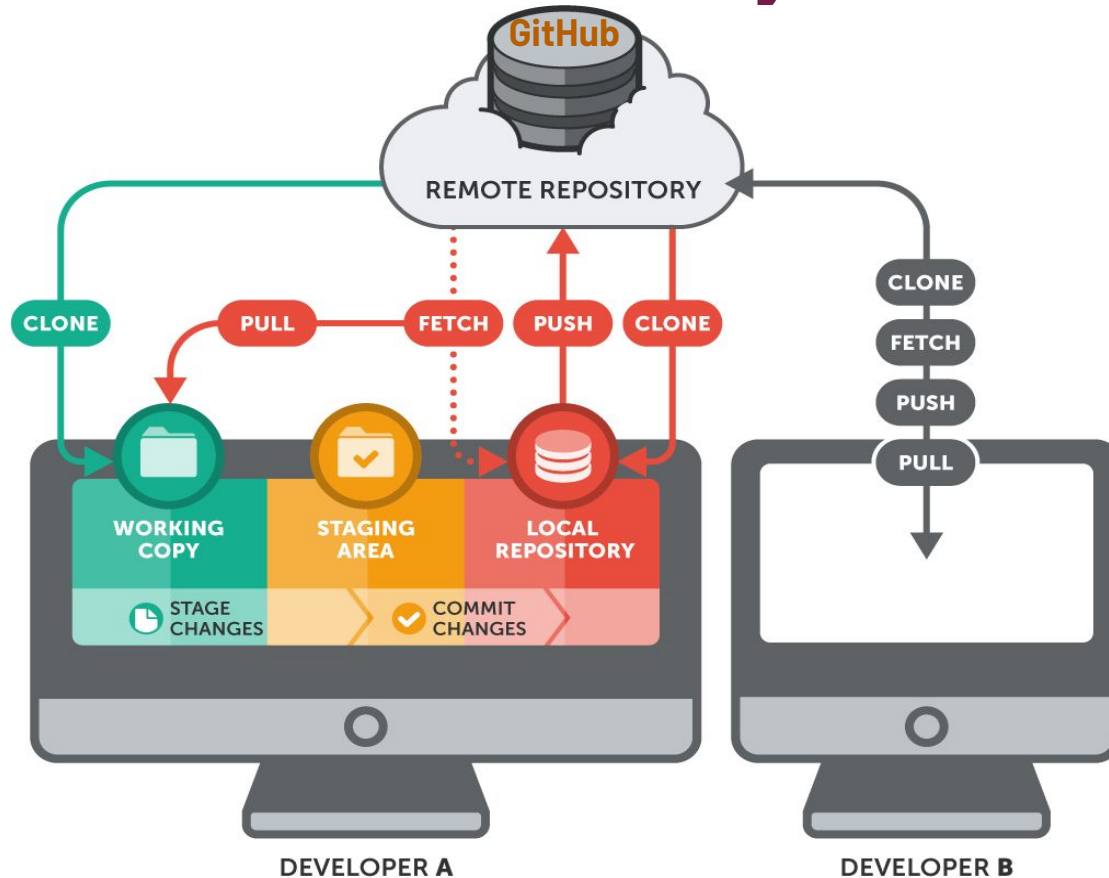
THANKS!

Any questions?

You can find me at:

- ▸ sumod@clarusway.com
- ▸ Slack - @sumod

CLARUSWAY©

WAY TO REINVENT YOURSELF