Capgemini

Introduction to GitHub

Lesson 1

Lesson Objectives

Setting the Context

- What is Git – an overview
- What You Need to Know About Git

Introduction to GitHub

- What is GitHub
- Signup on GitHub
- Branches are used to propose changes to GitHub projects
- Create or delete branches directly on GitHub
- Understanding Pull requests
  - Create a pull request to propose
  - Collaborate on changes to a repository
- Creating a Pull request from a fork
- Request a review for changes
- Change the base branch to compare the changes
- Commit changes on a pull request branch

Setting the Context

What is Git – an overview : Git Configuration

- Levels of Git Configuration
- Local : repository level (--local)
  - For example : while using a personal GitHub account
  - "git config -- local user.email a@b.org"
  - "git config –- local – list" to view the local configuration
  - "cat .git/config" or "vi ~.git/config"

- Global : user level (--global)
  - For example : To know/reset your user level settings
    - "git config -- user.name <optionally new value>"
    - "git config –- user.email <optionally new value>"
    - "git config –- global – list"
    - "cat ~/.gitconfig" or "vi ~/.gitconfig"
    - "git config -- global pull.rebase true" defaulting pull to rebase

- System : all user level (--system)
  - For example : Setting UI color for all users
  - "git config – system color.ui true

Setting the Context

What is GitHub

- GitHub is a collaboration platform.
- From software to legal documents, you can count on GitHub to help you do your best work with the collaboration and security tools your team needs.
- With GitHub, you can keep projects completely private, invite the world to collaborate, and streamline every step of your project.

- GitHub is also a powerful version control tool.
- GitHub uses Git, the most popular open source version control software, to track every contribution and contributor to your project--so you know exactly where every line of code came from.

Setting the Context

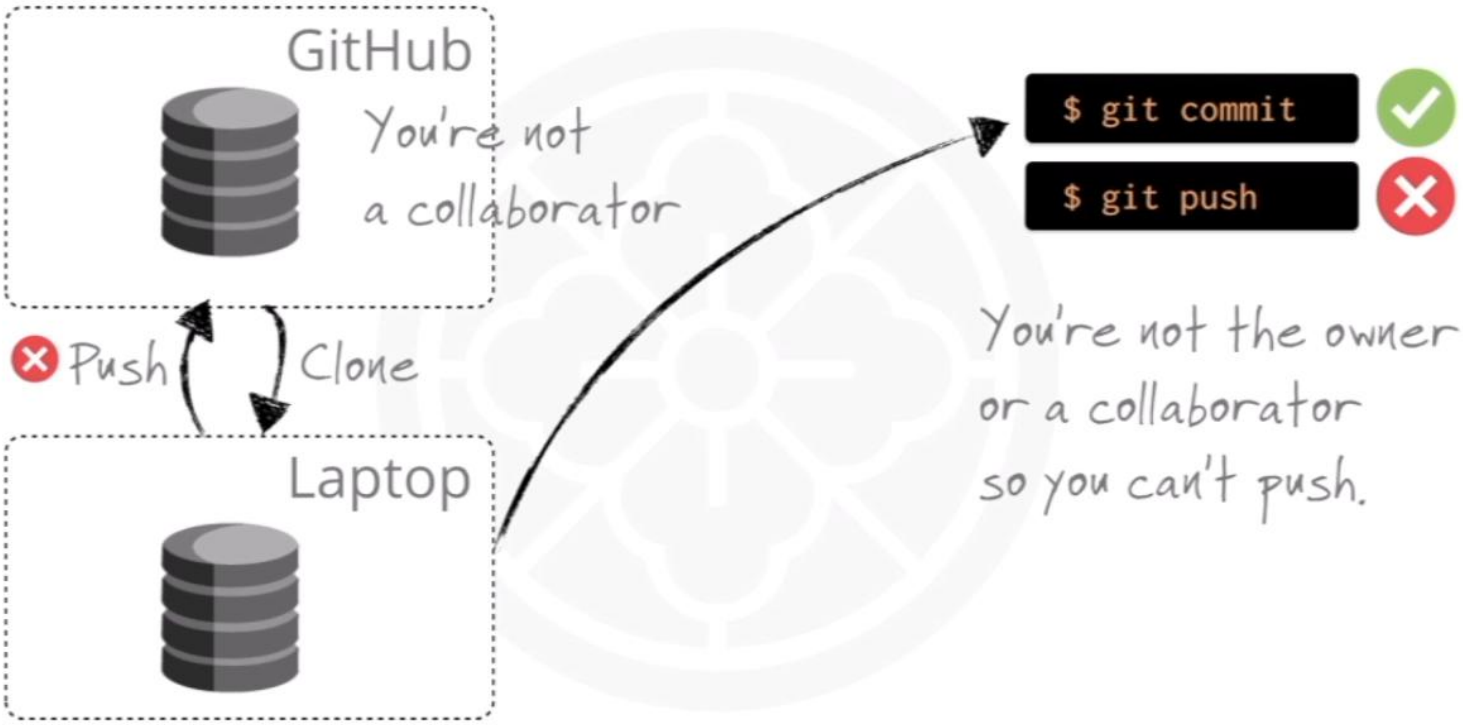
What is GitHub – Sharing a Git Configuration on GitHub

- Wiki Page
- Add the configuration files (dot files) to a repository on GitHub
- Global : user level (--global)
- System : all user level (--system)

Setting the Context

What is GitHub – Cloning :

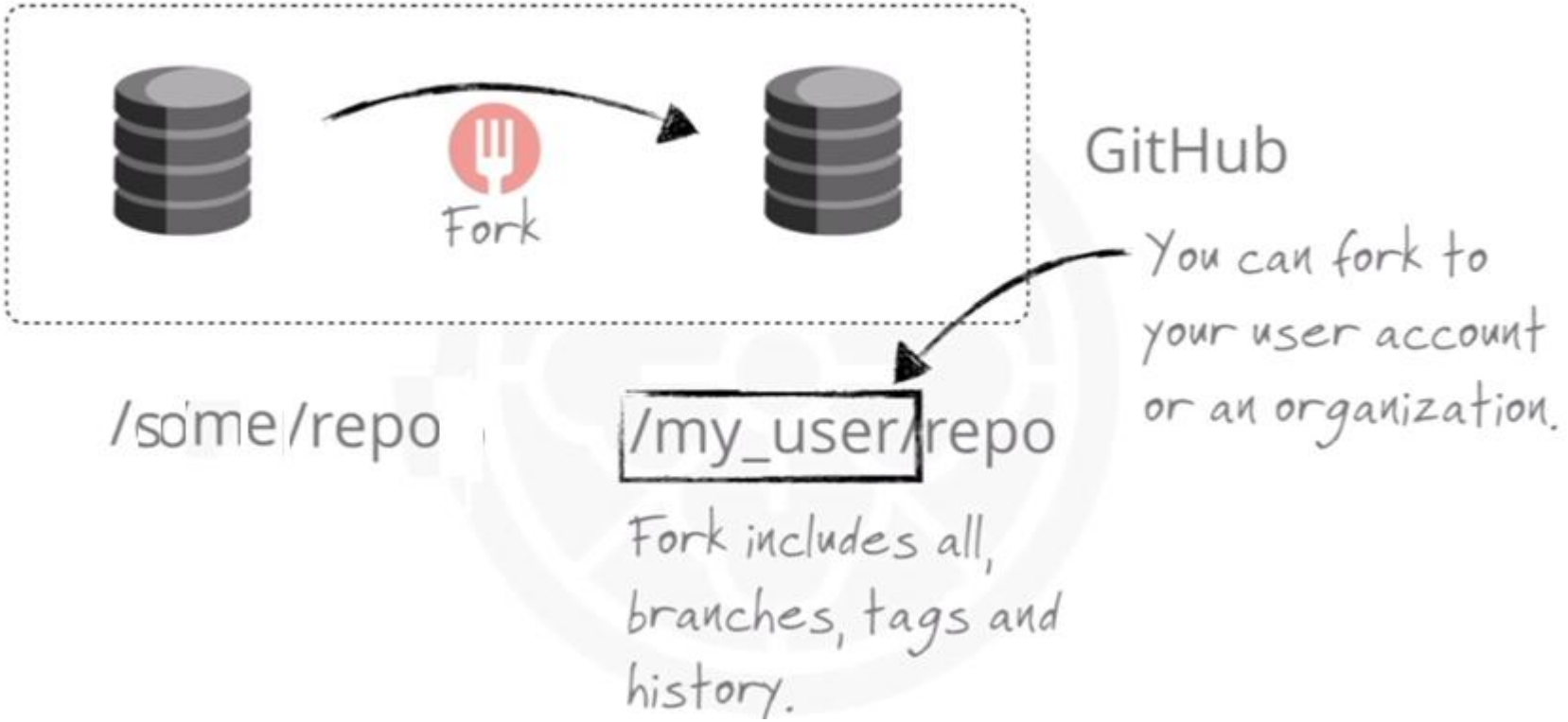
- When you are not a Ower/Collaborator on a public repo, you can
  - Clone the repo
  - Commit the changes locally
- However you can not push it back to the repository.



Setting the Context

What is GitHub – Forking :

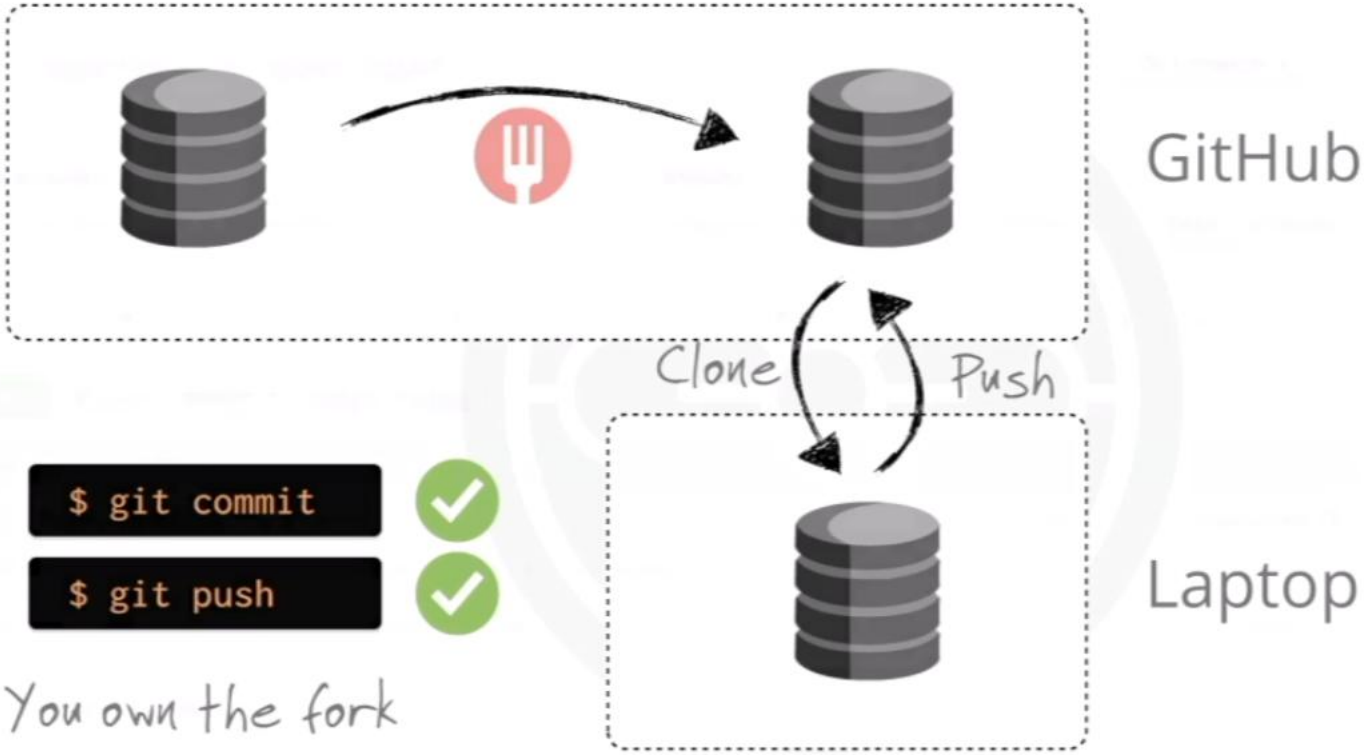
- You can fork a project on a public repo



Setting the Context

What is GitHub – Forking :

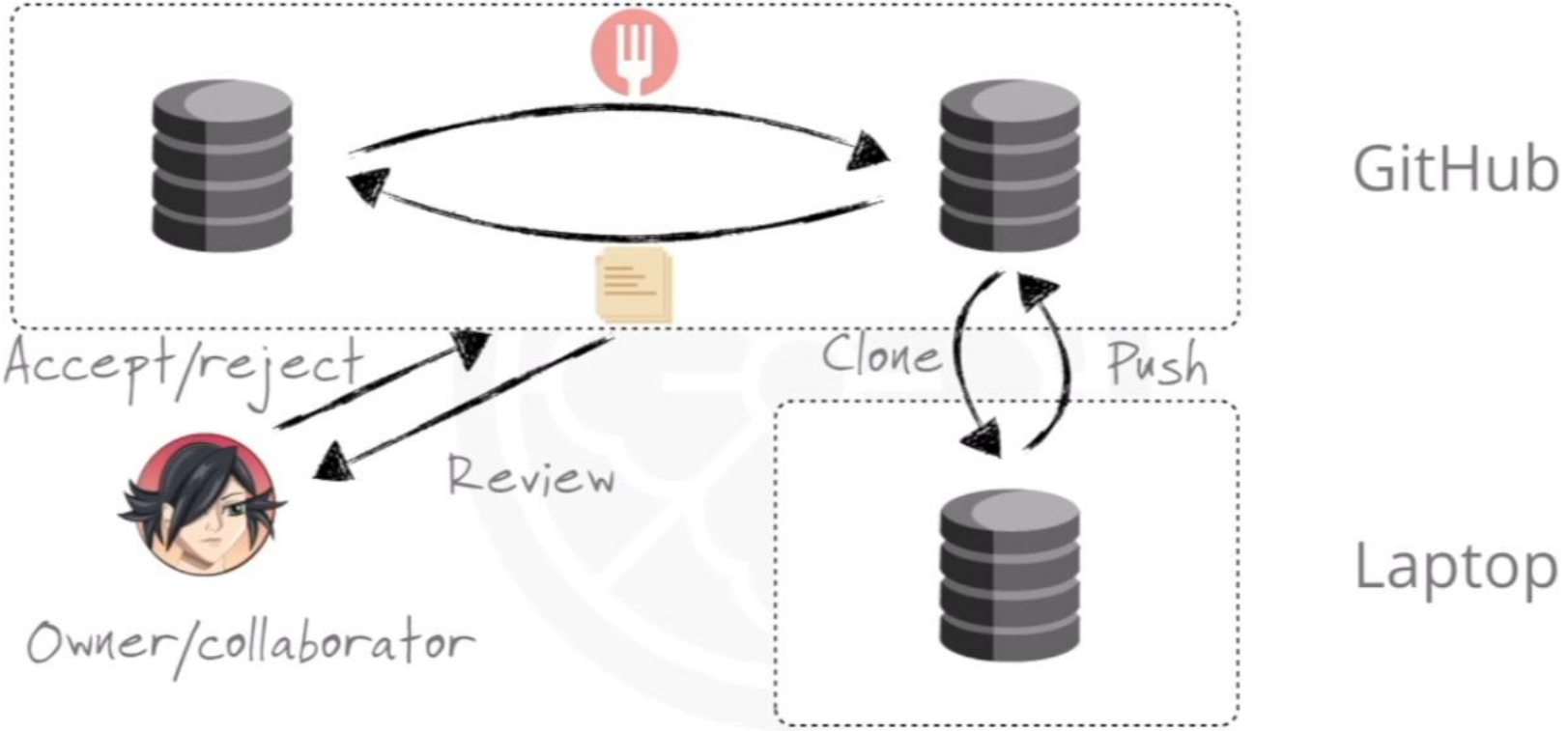
- You can fork a project on a public repo, you can
  - Clone the forked locally
  - Commit the changes
  - Push it back to the forked repo



Setting the Context

What is GitHub – Pull Request :

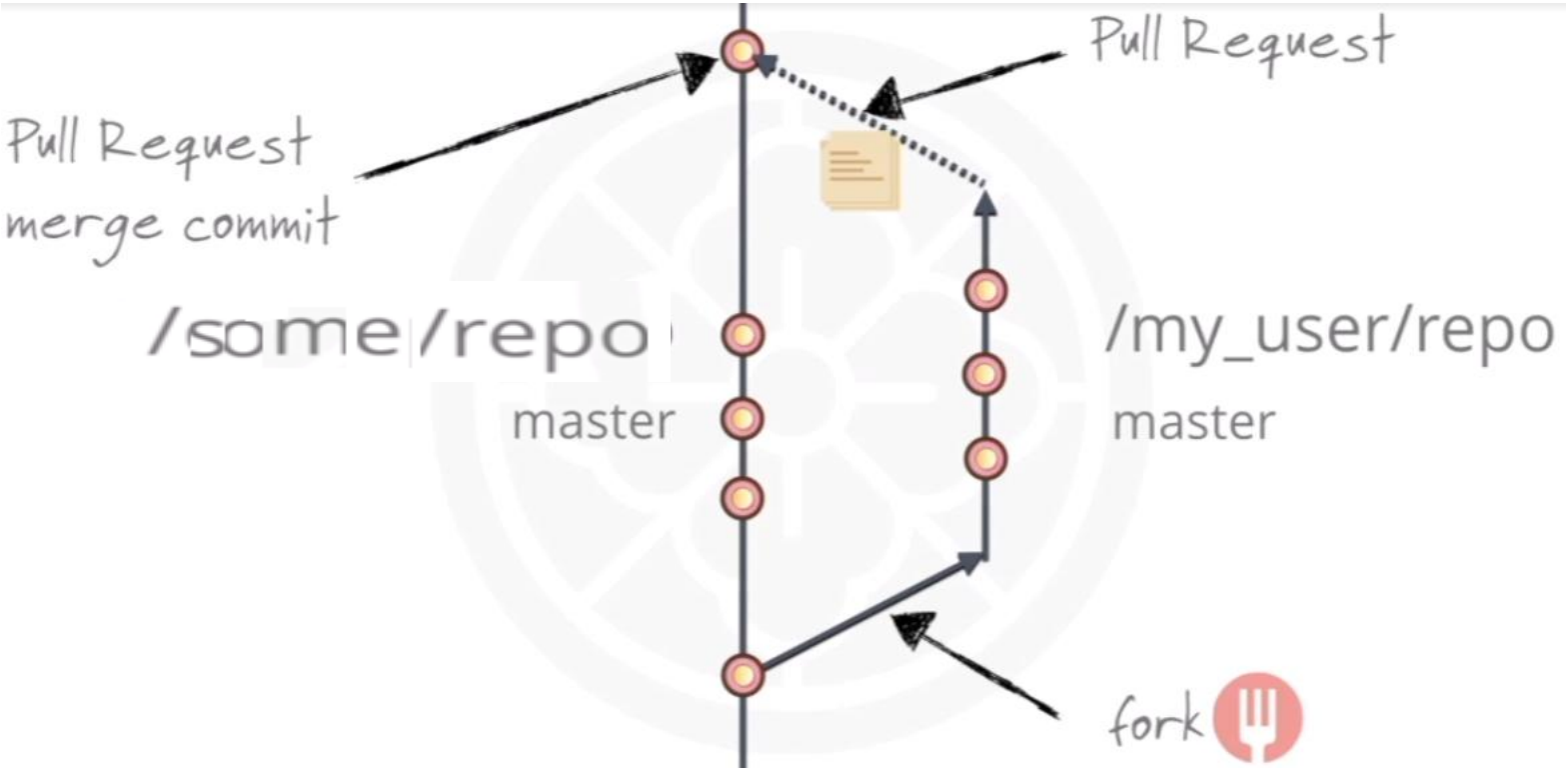
- You can fork a project on a public repo, you can
  - Clone and Commit the changes locally
  - Push it back to the forked repo
- By Submitting a Pull Request, you can
  - Send a Pull Request to original repo owner/collaborator
  - It is reviewed by owner/collaborator
  - Changes are Accepted ot Rejected



Setting the Context

What is GitHub – Pull Request :

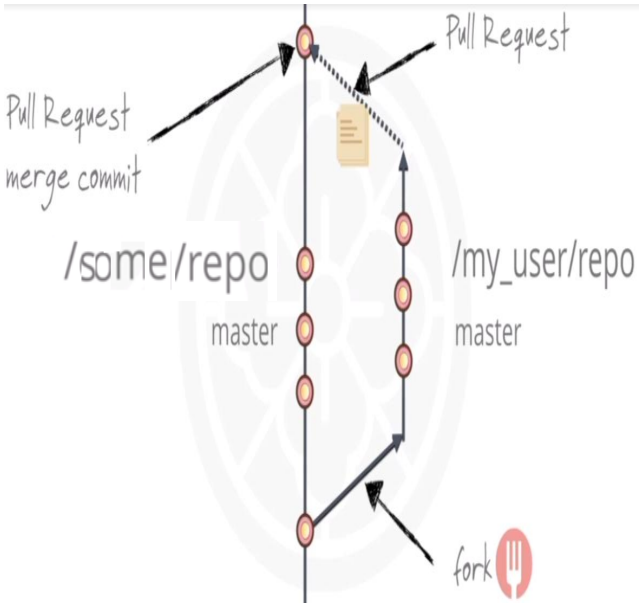
- We can think of Pull Request as a Branch
  - Forking creates a branch
  - You commit several times to the forked repo
- By Submitting a Pull Request, is like a Merge request
  - Forked repo branch is Merged into master Branch if Accepted



Setting the Context

What is GitHub – Pull Request :

- We can think of Pull Request as a Branch
  - Forking creates a branch
  - You commit several times to the forked repo
- By Submitting a Pull Request, is like a Merge request
  - Forked repo branch is Merged into master Branch if Accepted



Anyone can comment on a PR

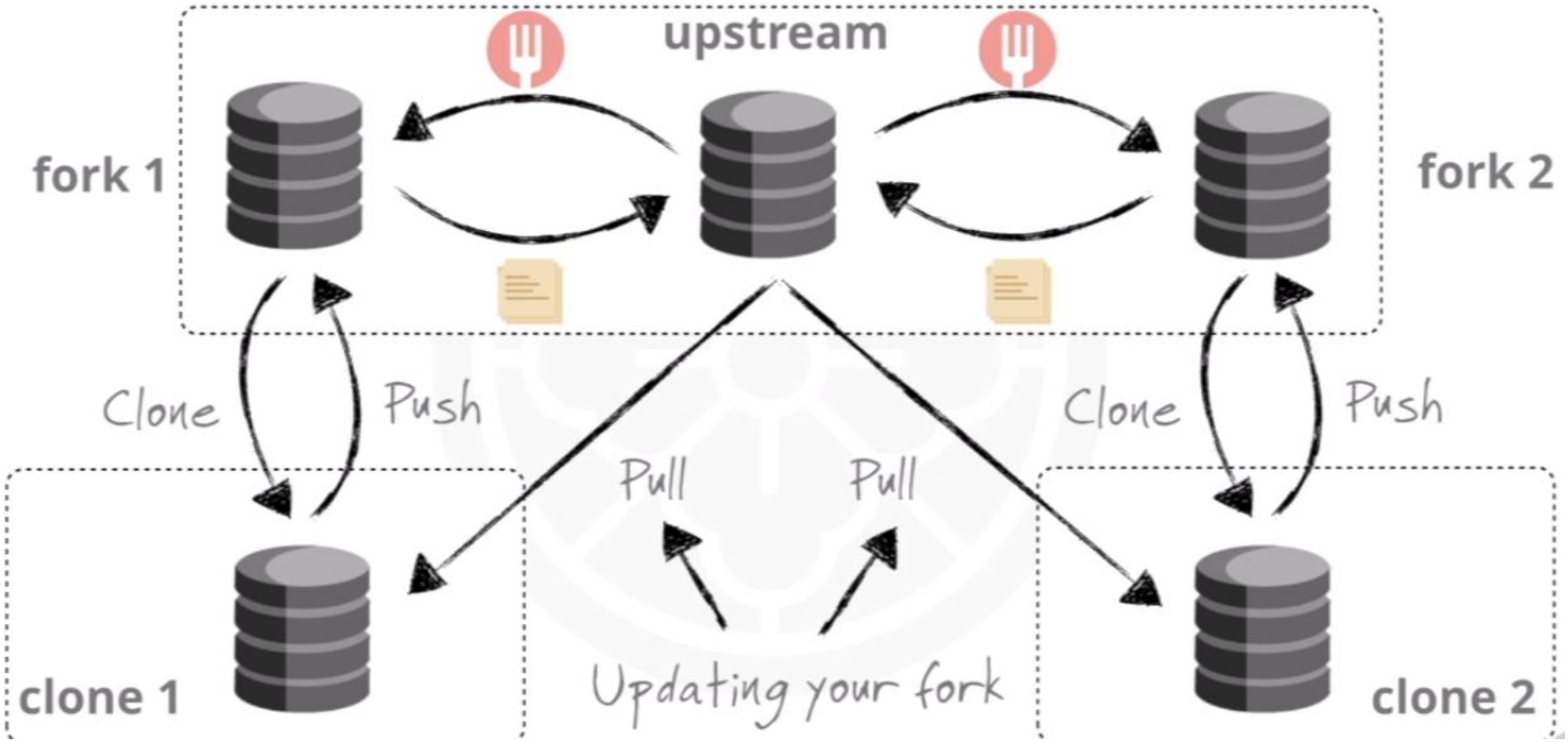
They can also checkout your branch, commit and add their changes to your PR.

The owner or any collaborator can merge in your PR.

Setting the Context

What is GitHub – Updating Fork with accepted Pull Requests(PR) :

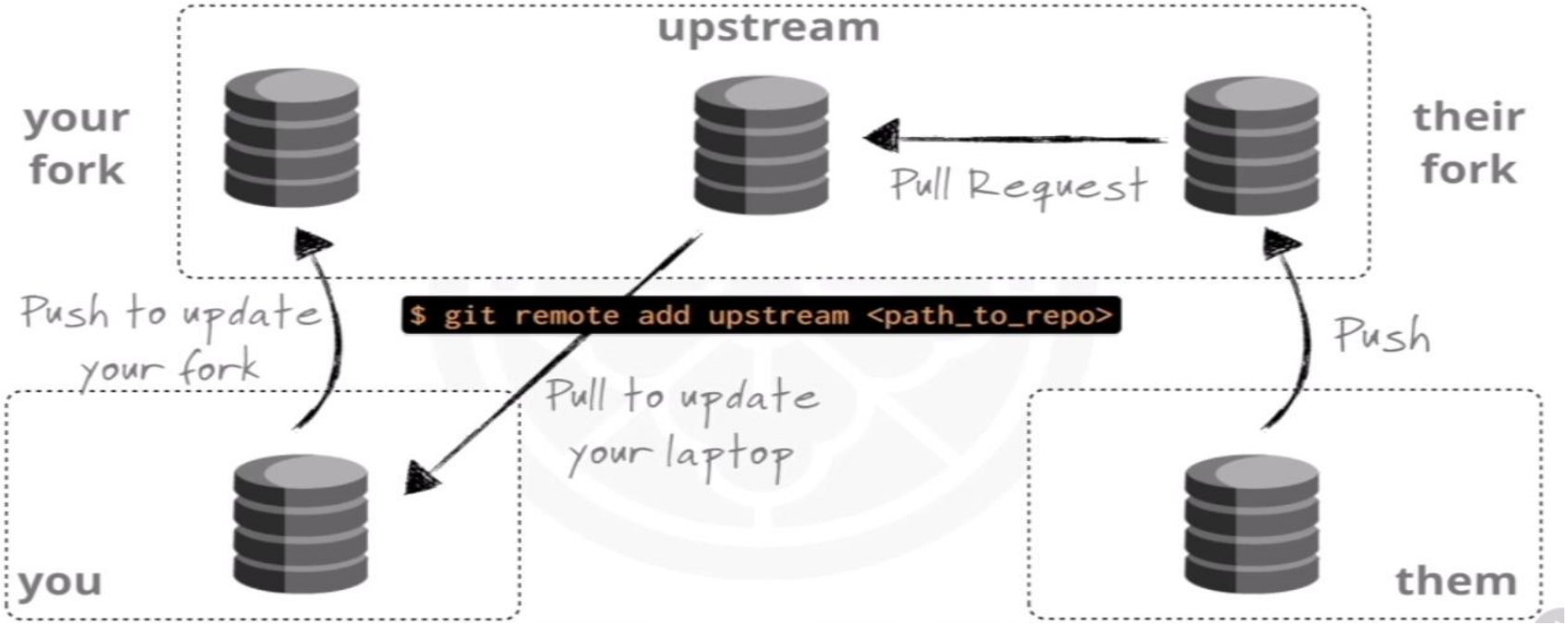
- Think of two devs have their own Forks of a upstream repo
  - After Forking they create their local clones
  - Push several times to the forked repo
- Created Pull Requests, which duly get accepted
  - Now the local repo has to be updated accordingly



Setting the Context

What is GitHub – Updating Fork with accepted Pull Requests(PR) :

- Think of two devs have forks of a upstream repo
  - After Forking they create their local clones
  - Push several times to the forked repo
- Care Pull Requests, which duly get accepted
  - Now the local repo has to be updated accordingly
  - On local clone, declare the original repo as its remote & fetch changes
  - Merge into local repo to update it
  - And then push to your fork to update



Setting the Context

What is GitHub – Updating Fork with accepted Pull Requests(PR) :

- Commands to update your Fork for every Pull Request accepted
  - On local clone, declare the original repo as its remote
  - Fetch the changes
  - Merge into local repo to update it
  - And then push to your fork to update

Add remote for upstream

```
$ git remote add upstream <path_to_repo>
```

Fetch changes

```
$ git fetch upstream
```

Merge them into master

```
$ git merge upstream/master master
```

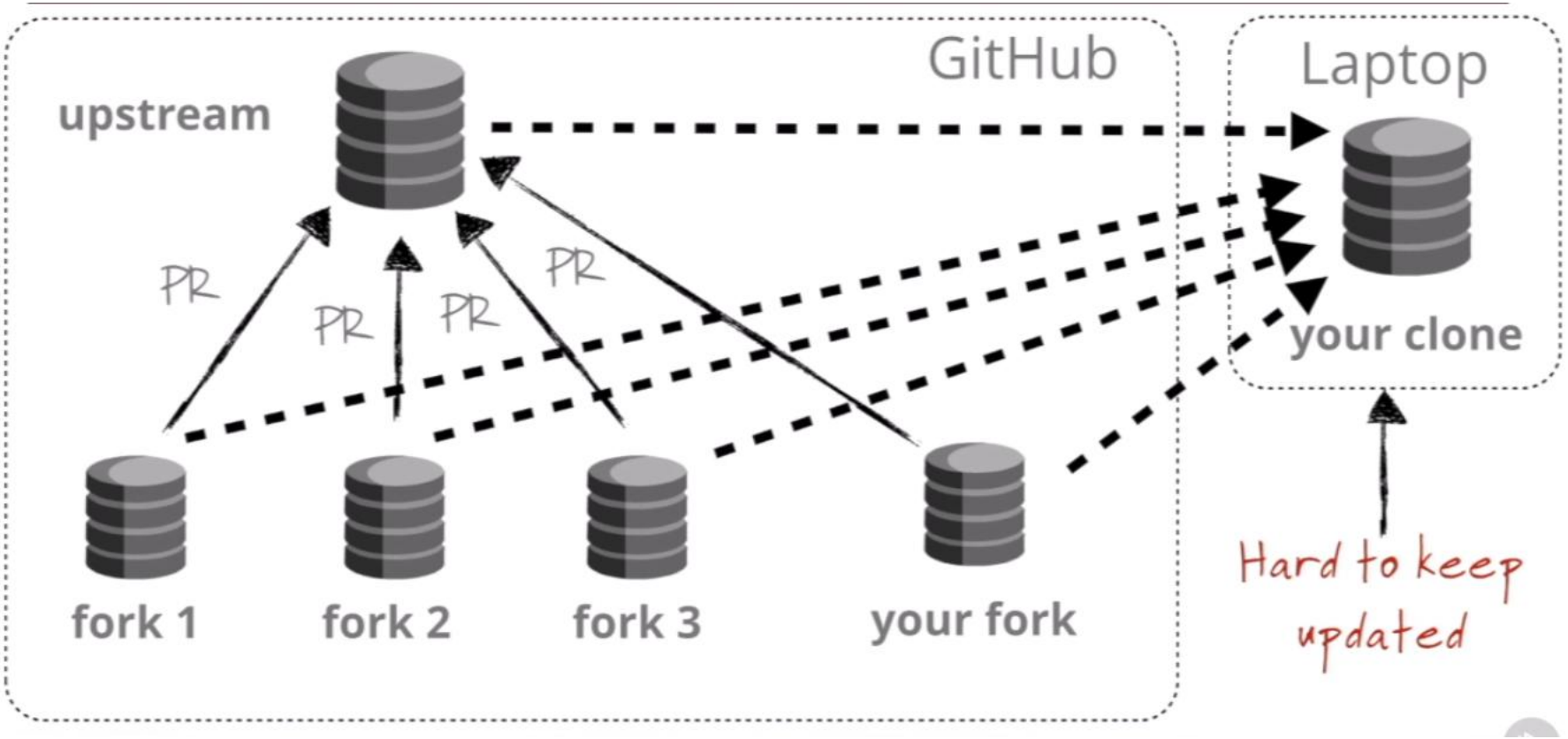
Push them to your remote

```
$ git push origin master
```

Setting the Context

What is GitHub – Multi Repository Workflow Challenges

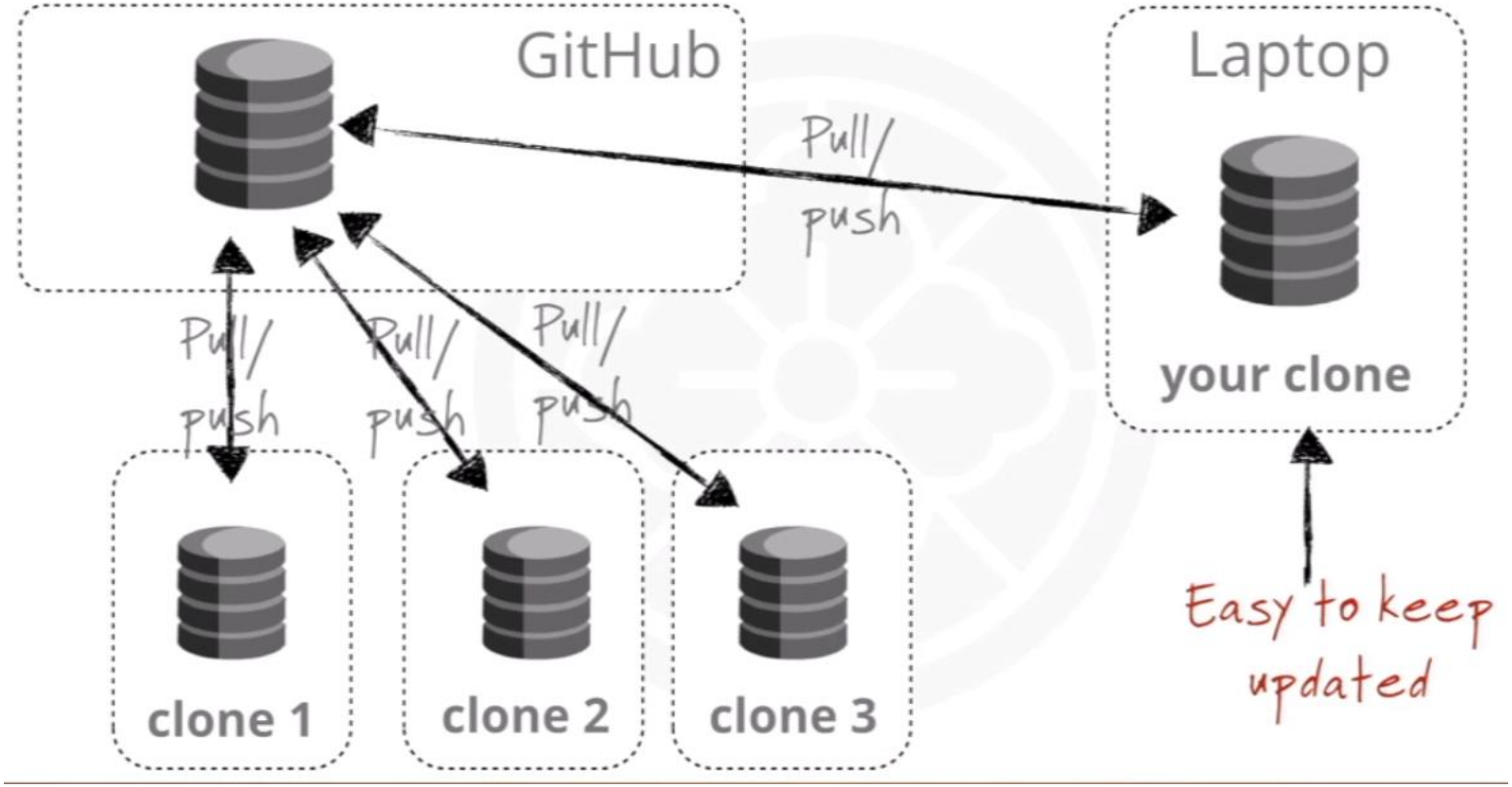
- Maintaining Multi repository Workflows is a challenging task
  - Especially while you try to keep your local codebase, up-to-date with PRs that are yet to be accepted from other forks in your team



Setting the Context

What is GitHub – Introducing Single Repository Workflow

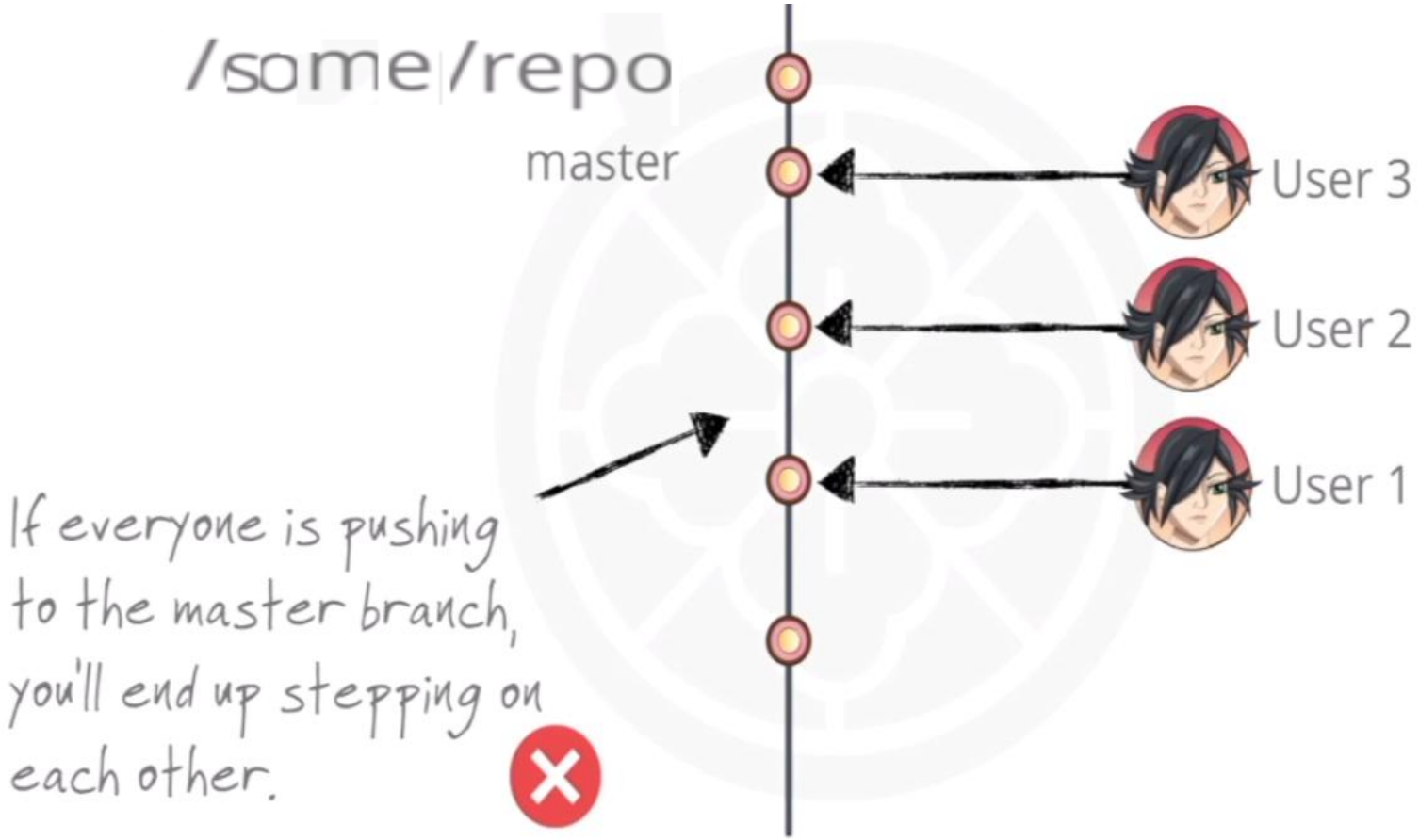
- It is easy to use Single repository Workflows
  - Especially while collaborating with central repository, in your team, every one can close the central repo, pull or push the changes on it. NO Fork required in this case.



Setting the Context

What is GitHub – Causion : Single Repository Workflow

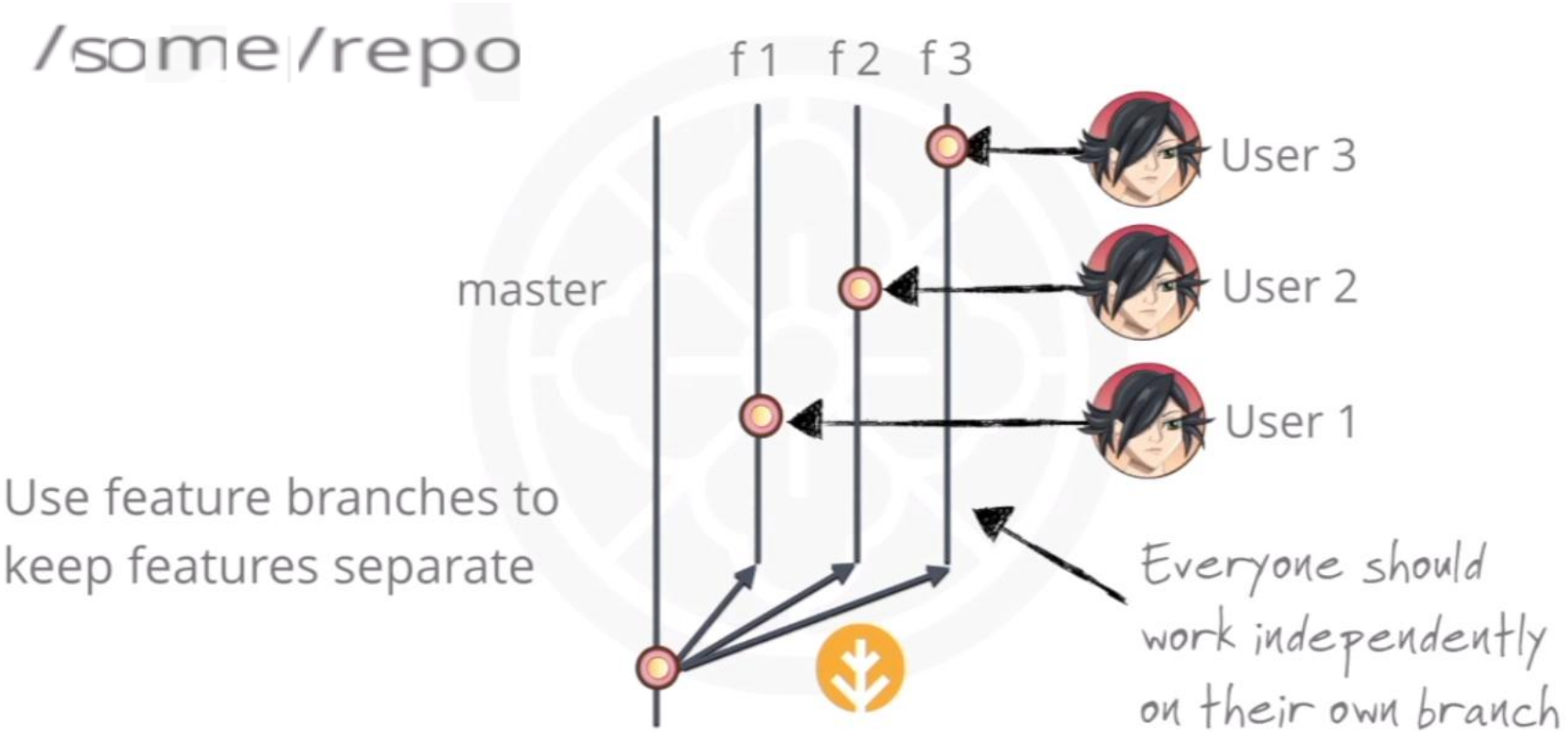
- Due to use Single repository Workflows, there can be regular Conflicts



Setting the Context

What is GitHub – Feature Branches

- Feature Branches can used for independent development on Single repository Workflows



Setting the Context

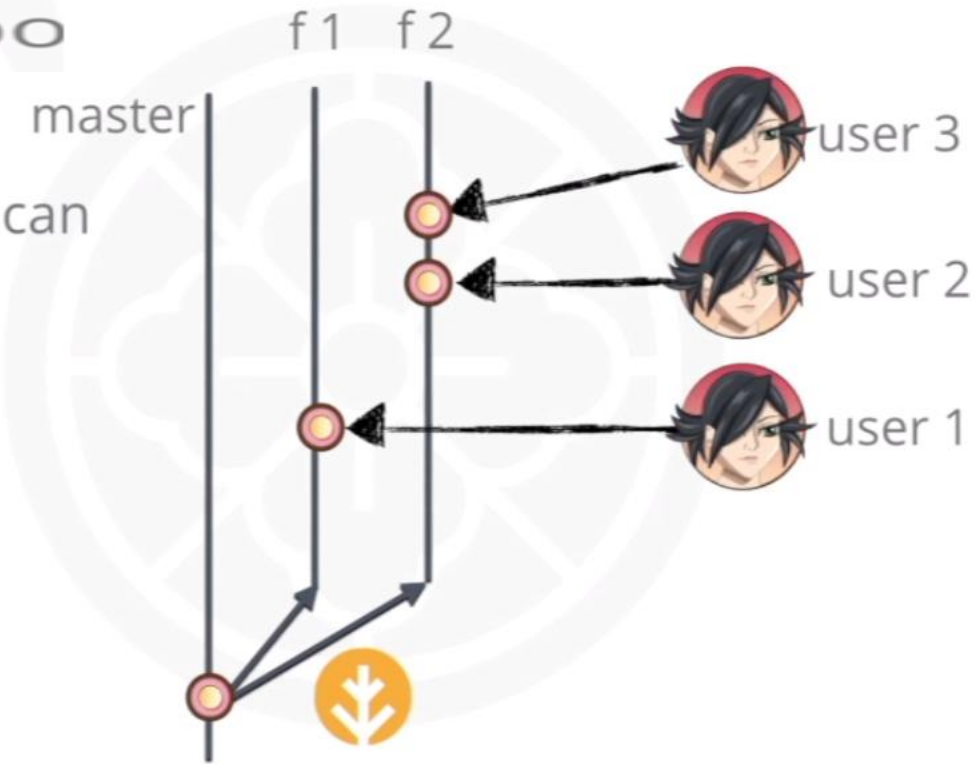
What is GitHub – Collaborating on Feature Branches

- Feature Branches can used for independent development on Single repository Workflows. Each developer can ….
  - Fetch the changes from the remoter repo
  - List all existing branches
  - Checkout the branch that has been recently pushed to GitHub

/some/repo

Multiple developers can collaborate on each feature branch.

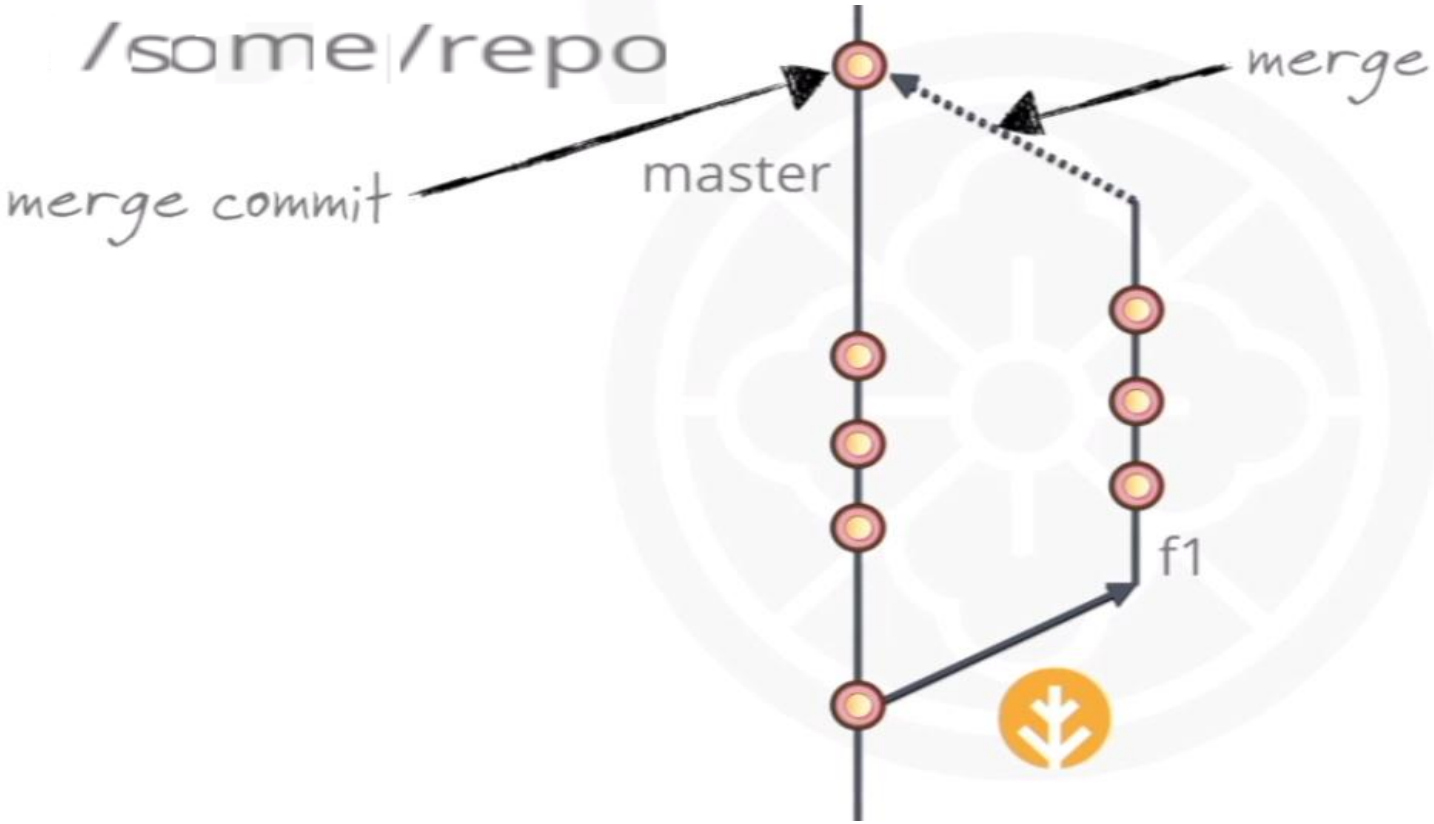
```
$ git fetch
$ git branch -a
$ git checkout f2
```



Setting the Context

What is GitHub – Merging the Development back to Master

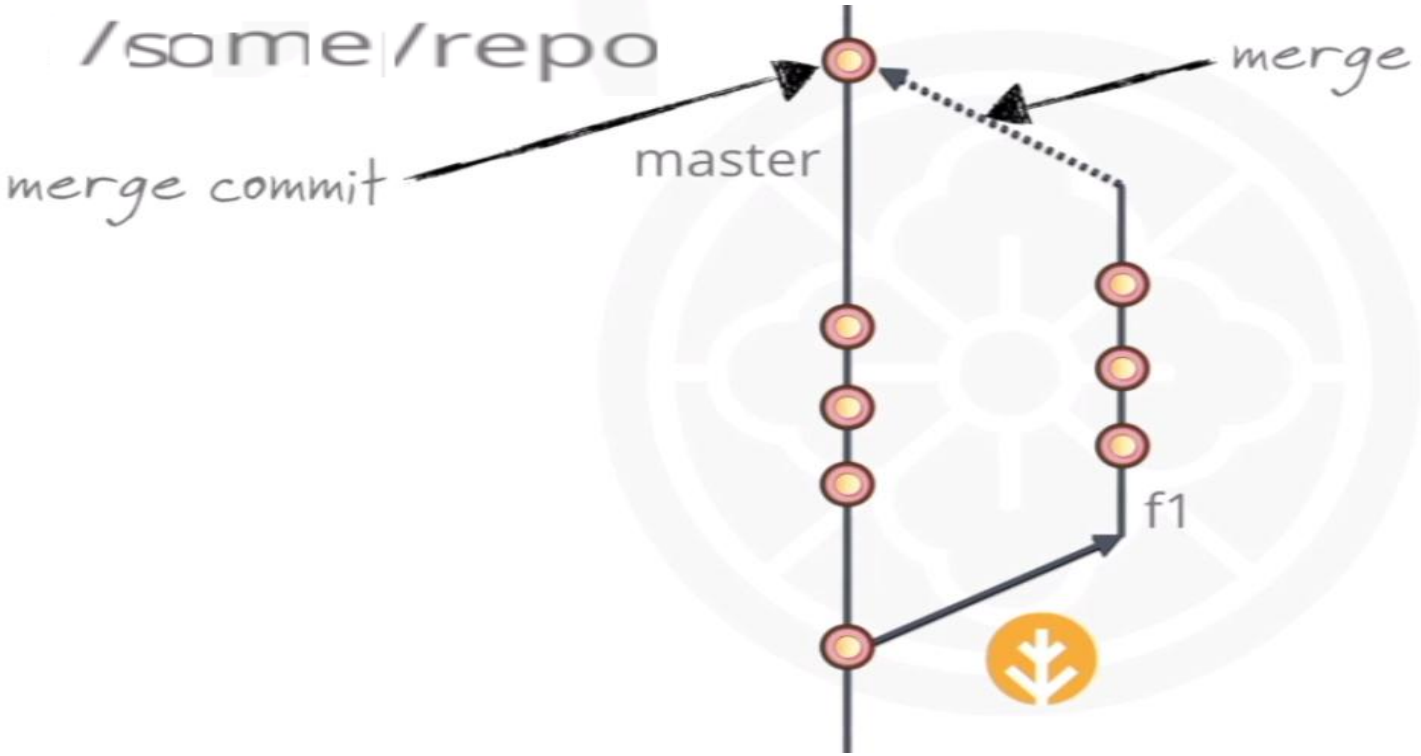
- Once a Feature is done …
  - Assuming that the right branch is checkout
  - Merge the changes on the Master branch
  - Push the changes to GitHub



Setting the Context

What is GitHub – Pull Requests are used for Review on GitHub

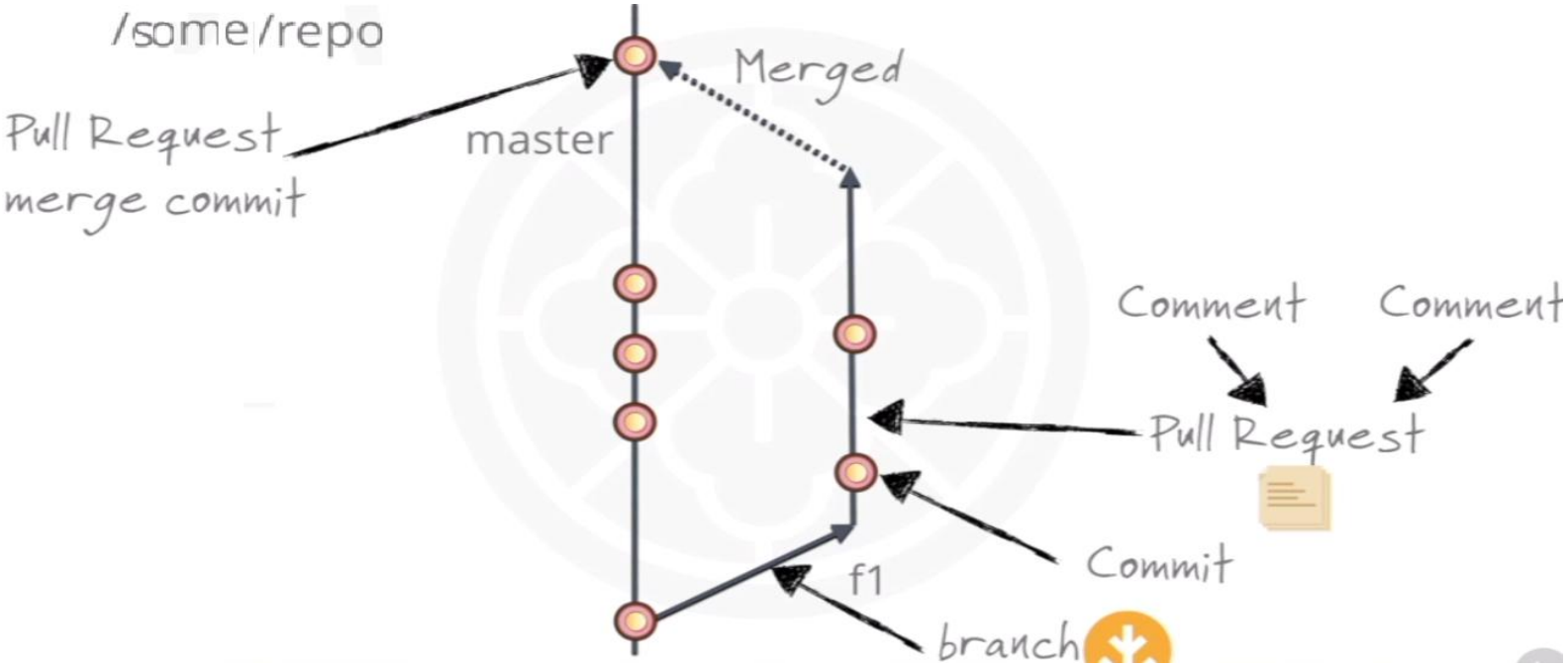
- Pull Requests can be used for review.
- With Pull Request you can have multple developers collaborating on a branch and commenting on it directly on GiHub
- Every developer creates own franch and work on single repo
- Once work is done, create a Pull request and merge into the master branch



Setting the Context

What is GitHub – Collaborating on Pull Requests

- For Working on a single repo in a team
  - Create a branch, Do the first commit
  - Create a pull request, you will receive comments
  - Make changes , Push them upto GitHub
  - These changes will automatically be added to the Pull Request.
  - This is because, Pull requests point head of a branch not to a particular commit
  - Then either merge through GitHub GUI or commit merge and push through the command prompt CUI



Setting the Context

What is GitHub – Commands used while Collaborating on Pull Requests

Download all branches from GitHub

```
$ git fetch
```

View all of the branches

includes remote tracking branches in red

```
$ git branch -a
```

Checkout a local copy of a remote branch

```
$ git checkout <branch_name>
```

Test code, make any changes and then commit and push changes

```
$ <make edits>
$ git commit
$ git push
```

Summary : We discussed …

Setting the Context

- What is Git – an overview
- What You Need to Know About Git

Introduction to GitHub

- What is GitHub
- Signup on GitHub
- Branches are used to propose changes to GitHub projects
- Create or delete branches directly on GitHub
- Understanding Pull requests
  - Create a pull request to propose
  - Collaborate on changes to a repository
- Creating a Pull request from a fork
- Request a review for changes
- Change the base branch to compare the changes
- Commit changes on a pull request branch

About Capgemini

With more than 190,000 people, Capgemini is present in over 40 countries and celebrates its 50th Anniversary year in 2017. A global leader in consulting, technology and outsourcing services, the Group reported 2016 global revenues of EUR 12.5 billion. Together with its clients, Capgemini creates and delivers business, technology and digital solutions that fit their needs, enabling them to achieve innovation and competitiveness. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Learn more about us at

www.capgemini.com

People matter, results count.