Git and GitHub Basics

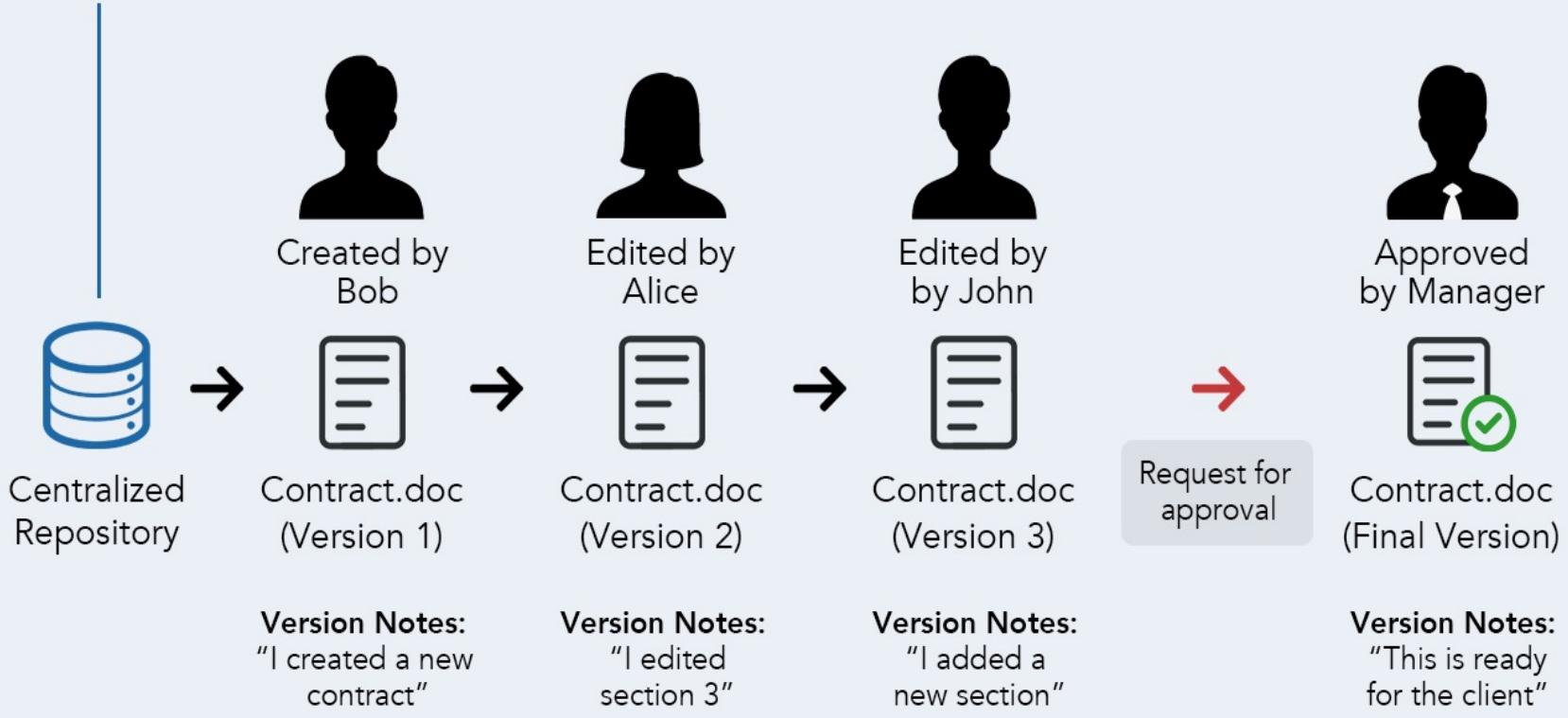
Myo Minn Oo

1. Version Control

- a system
- records changes to set of files over time
- recall specific versions later.

Source: https://www.documentlocator.com/img/knowledge/version-control-diagram.png

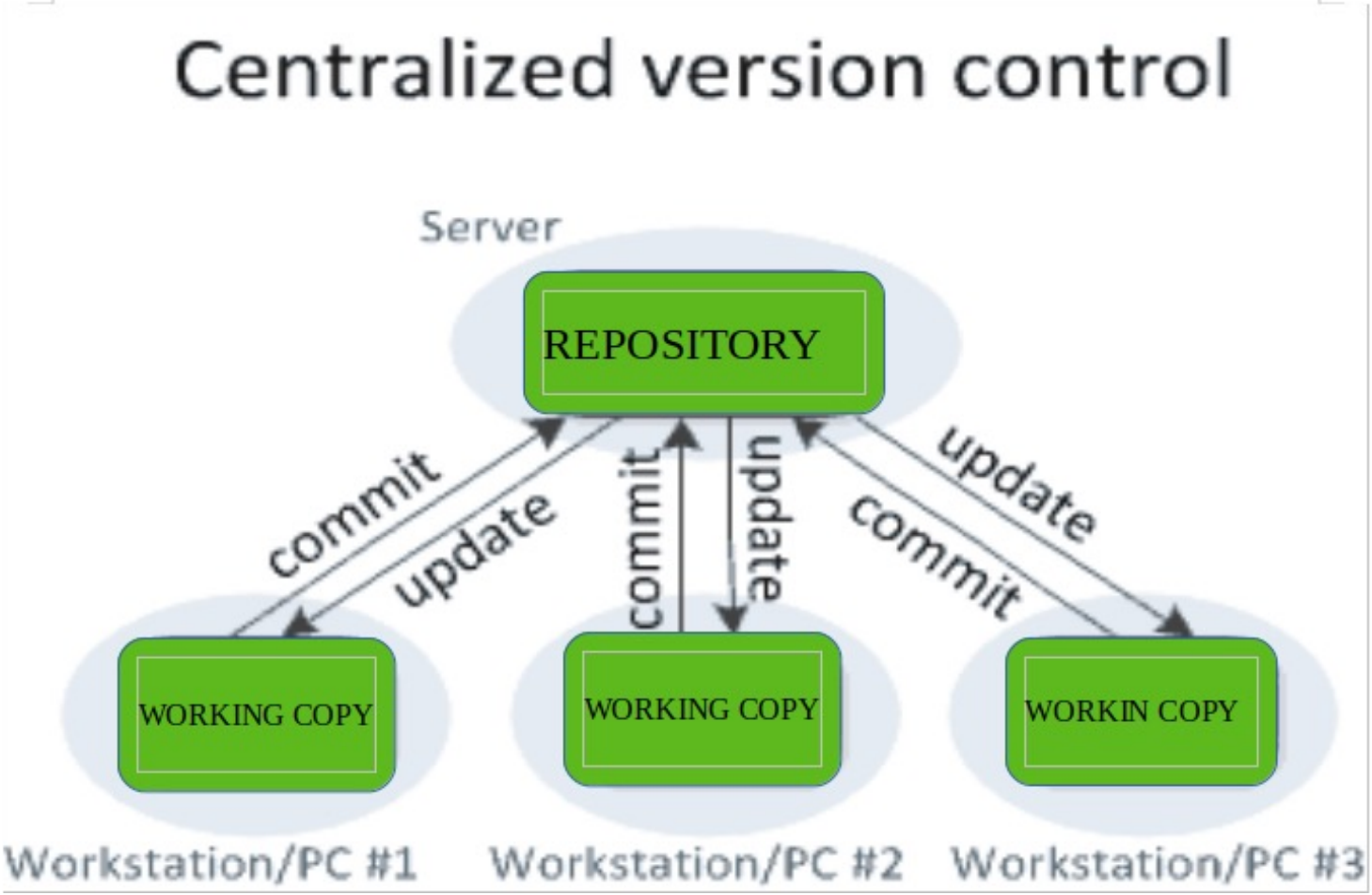
How Version Control Works



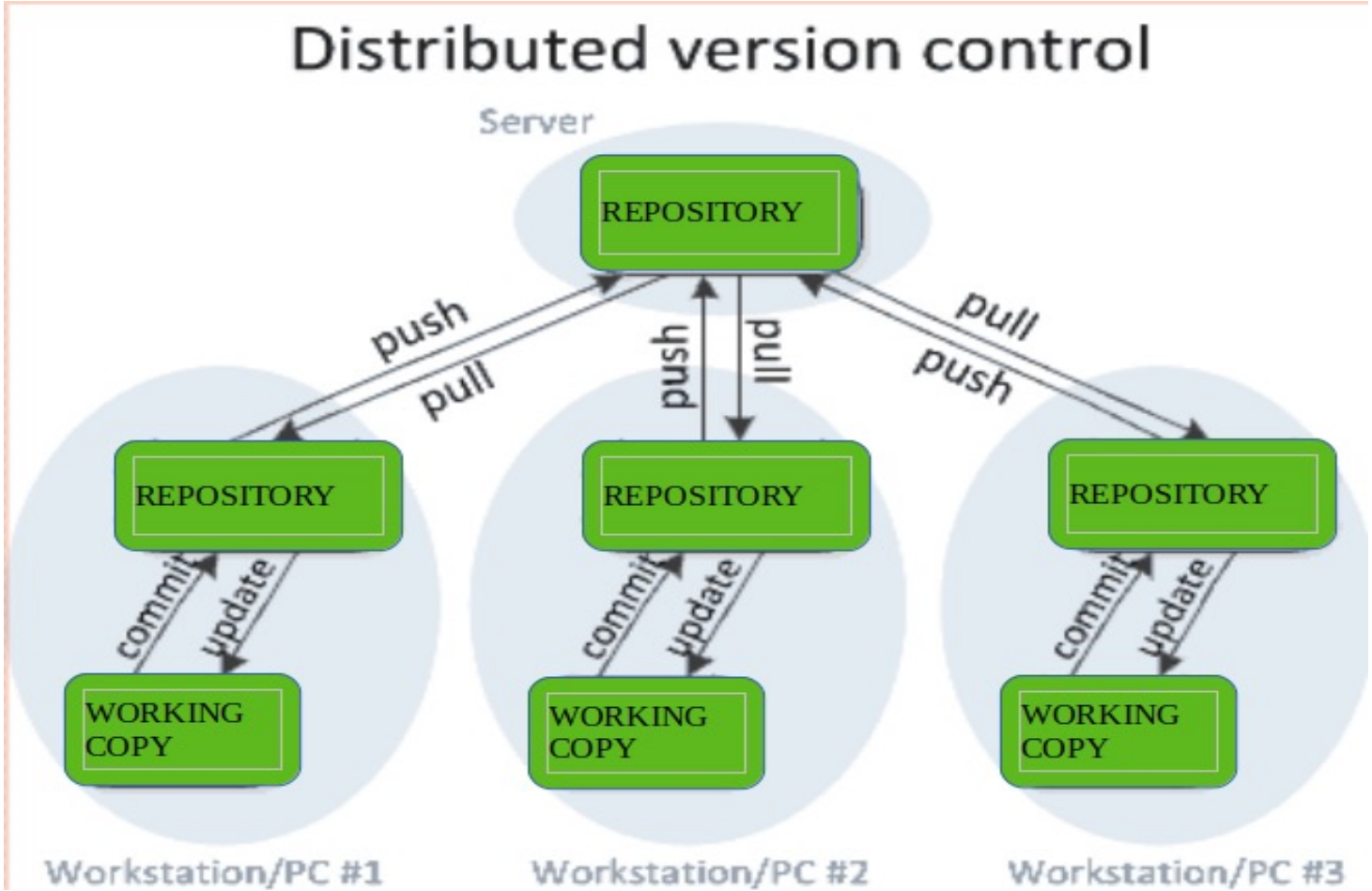
Why Use Version Control Software?

- show the changes made to the code over time
- allow to backtrack if necessary and undo those changes
- all versions are stored on a central server

Source: Version Control Systems - https://www.geeksforgeeks.org/version-control-systems/



Source: Version Control Systems - https://www.geeksforgeeks.org/version-control-systems/



2. Git

- free and open source
- basis of distributed development
- offer a full-fledged repository with complete history and full version-tracking capabilities
- Allows a team of people to work together, all using the same files

Git repository

- **Working directory**

Your local directory where you make the project (write code) and make changes to it.

- **Staging Area**

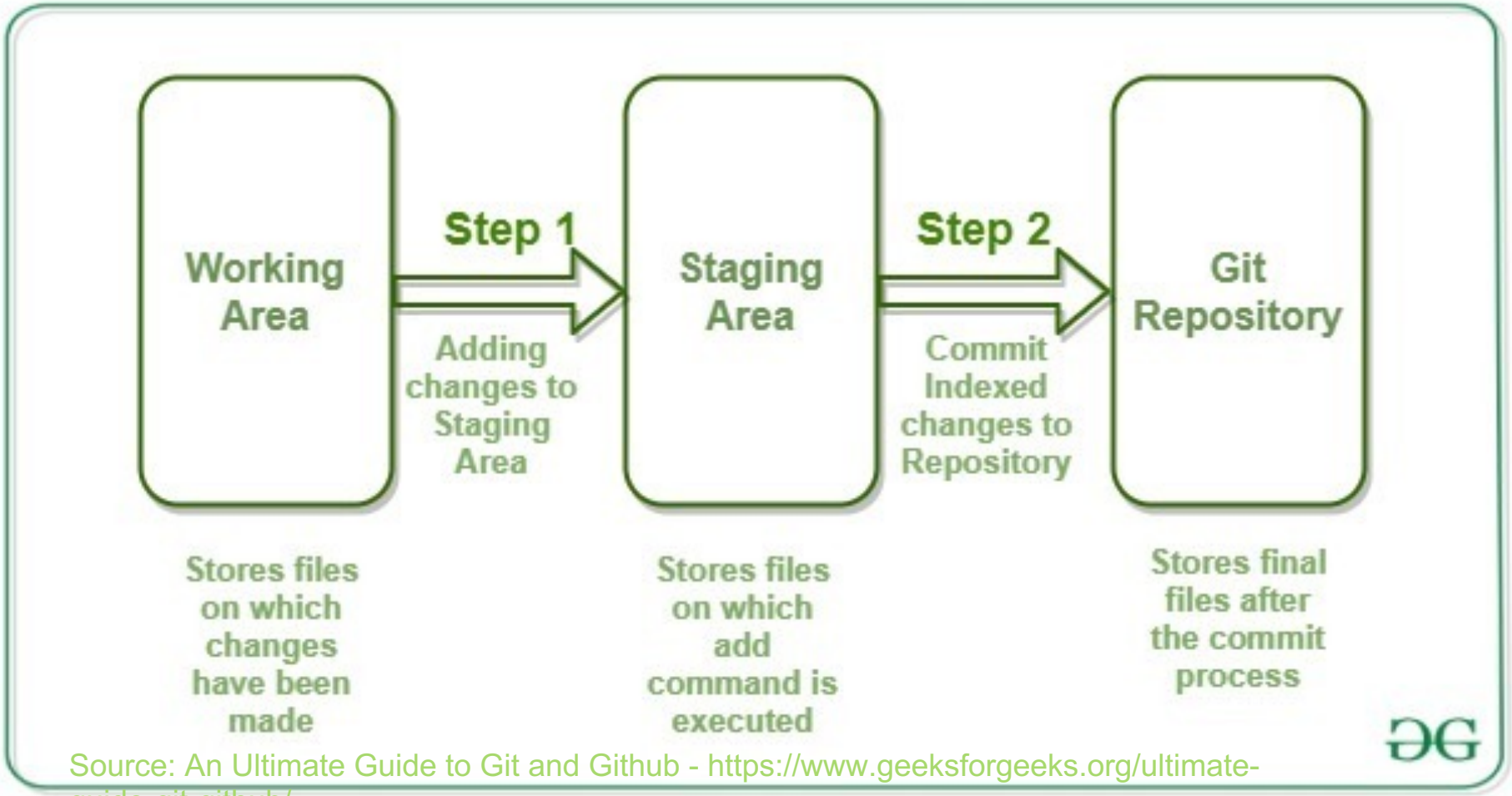
An area where you first need to put your project before committing. This is used for code review by other team members.

- **Local Repository**

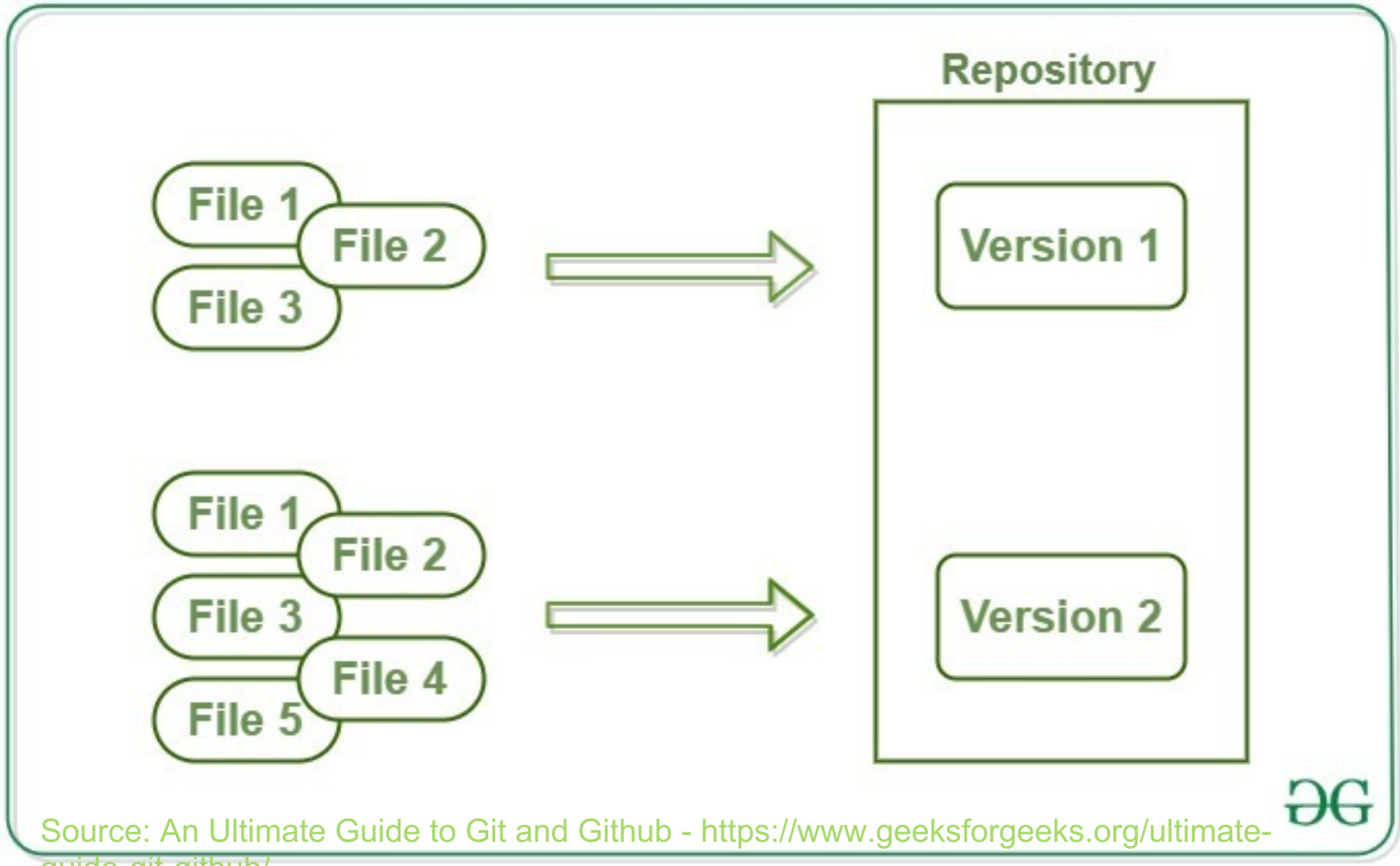
Your local repository where you commit changes to the project before pushing them to central repository on Github.

- **Central Repository**

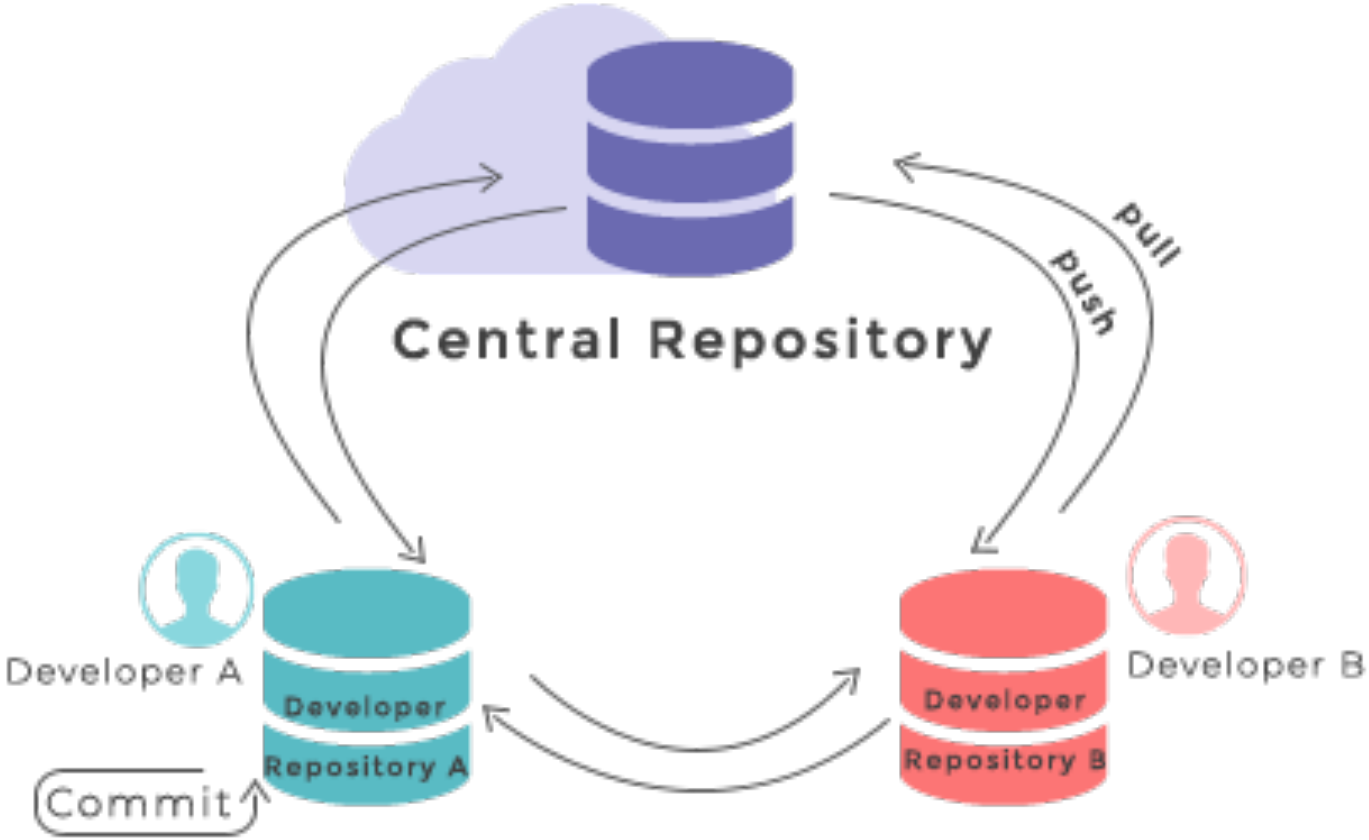
Main project on the central server, a copy of which is with every team member as local repository.



Source: An Ultimate Guide to Git and Github - https://www.geeksforgeeks.org/ultimate-guide-git-github/



Source: An Ultimate Guide to Git and Github - https://www.geeksforgeeks.org/ultimate-guide-git-github/



Source: W3docs - Git Repository | W3Docs Git Online Tutorial

Some commands

which relate to repository structure:

```
git add
// transfers your project from working directory
// to staging area.
```

```
git commit
// transfers your project from staging area to
// Local Repository.
```

```
git push
// transfers project from local to central repository.
// (requires internet)
```

More commands

```
git commit -a -m "message for commit"
-a: commit all files and for files that have been
     staged earlier need not to be git add once more
-a option does that automatically.
```

```
git push origin master -> pushes your files to
                          github master branch
git push origin anyOtherBranch -> pushes any
                        other branch to github.
git log ; to see all your commits
```

```
git checkout commitObject(first 8 bits) file.txt->
revert back to this previous commit  for file file.txt
```

3. GitHub

- a repository hosting service tool
- features collaboration and access control
- designed for the developers and to help them track their changes into a project through the repository.

Some features of GitHub

- specifies milestone & labels to projects
- comparison view between branches
- publish and host websites
- syntax highlight feature.
- third-party API integrations for bug tracking and cloud hosting

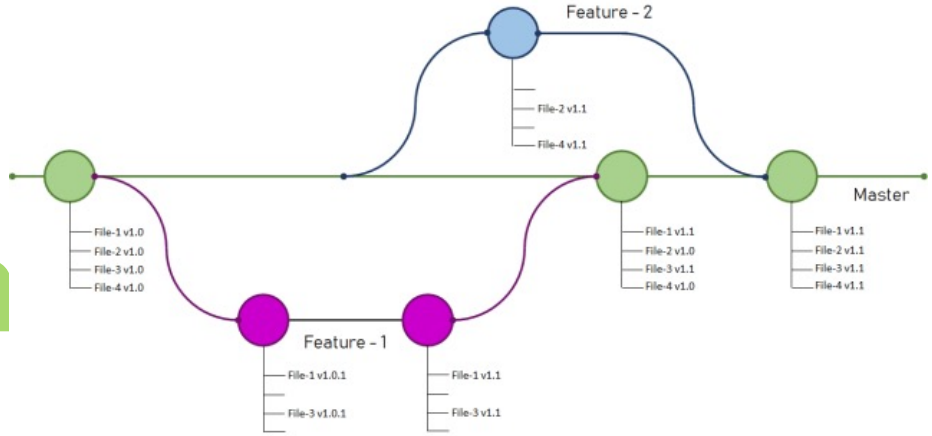
Task 1

1. Log in to GitHub.com
2. Create a repository named “my-first-github”
3. Add README file
4. Commit the changes.

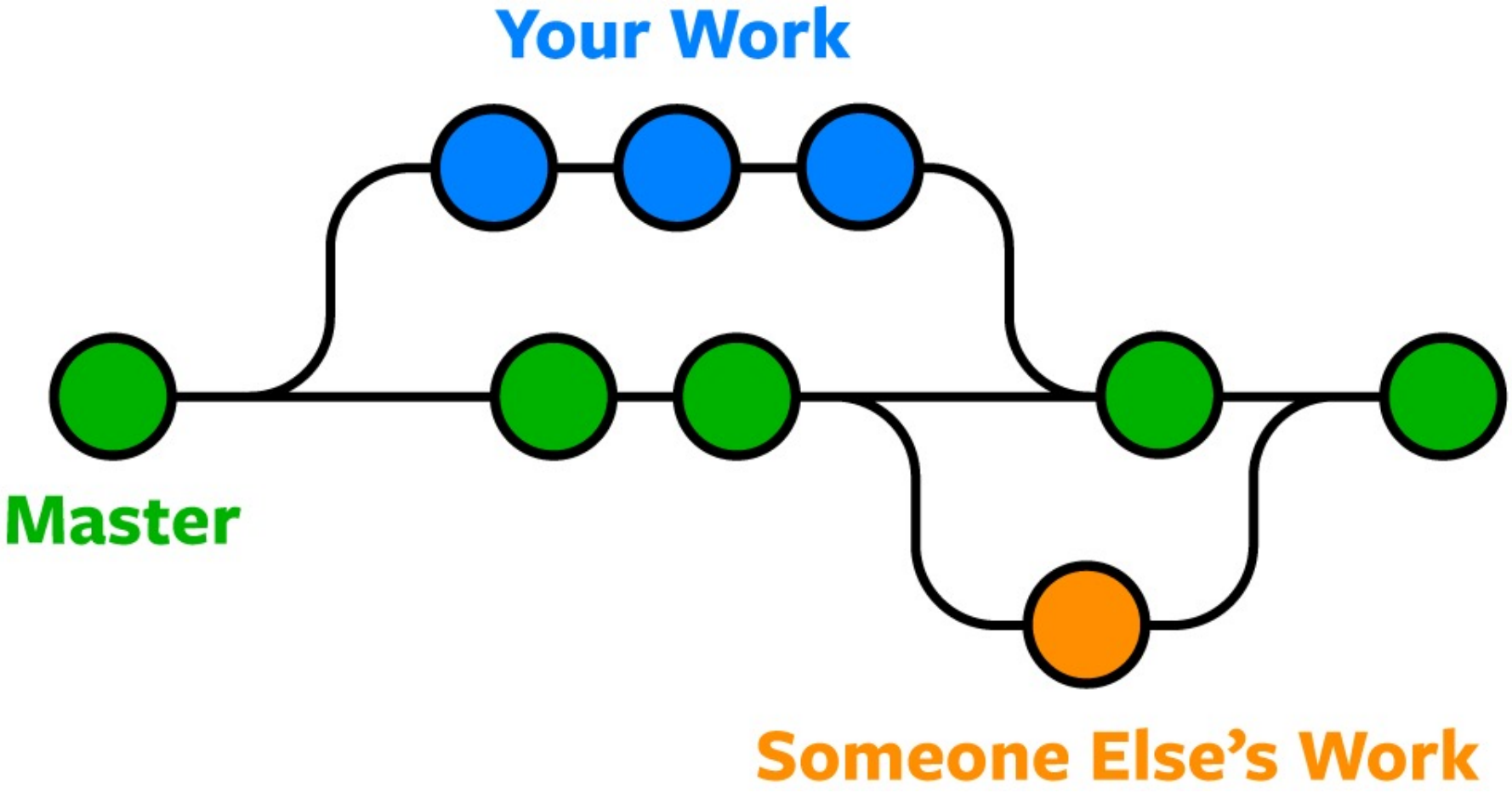
Task 2

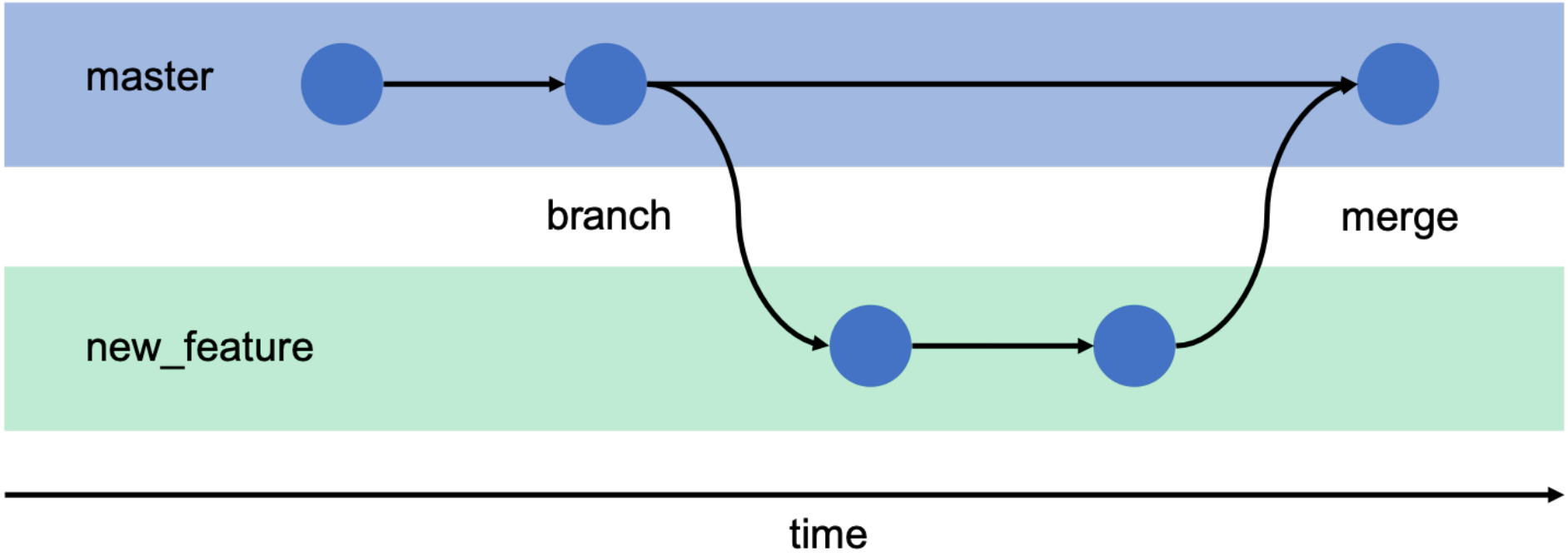
1. Open GitHub Desktop
2. Clone your repo “my-first-github”
3. Edit README file
4. Commit changes & push to origin

4. Git branch



- branches are a part of daily development process
- a snapshot of your changes
- make unstable code harder to get merged into the main code base


Your Work
Master
Someone Else's Work

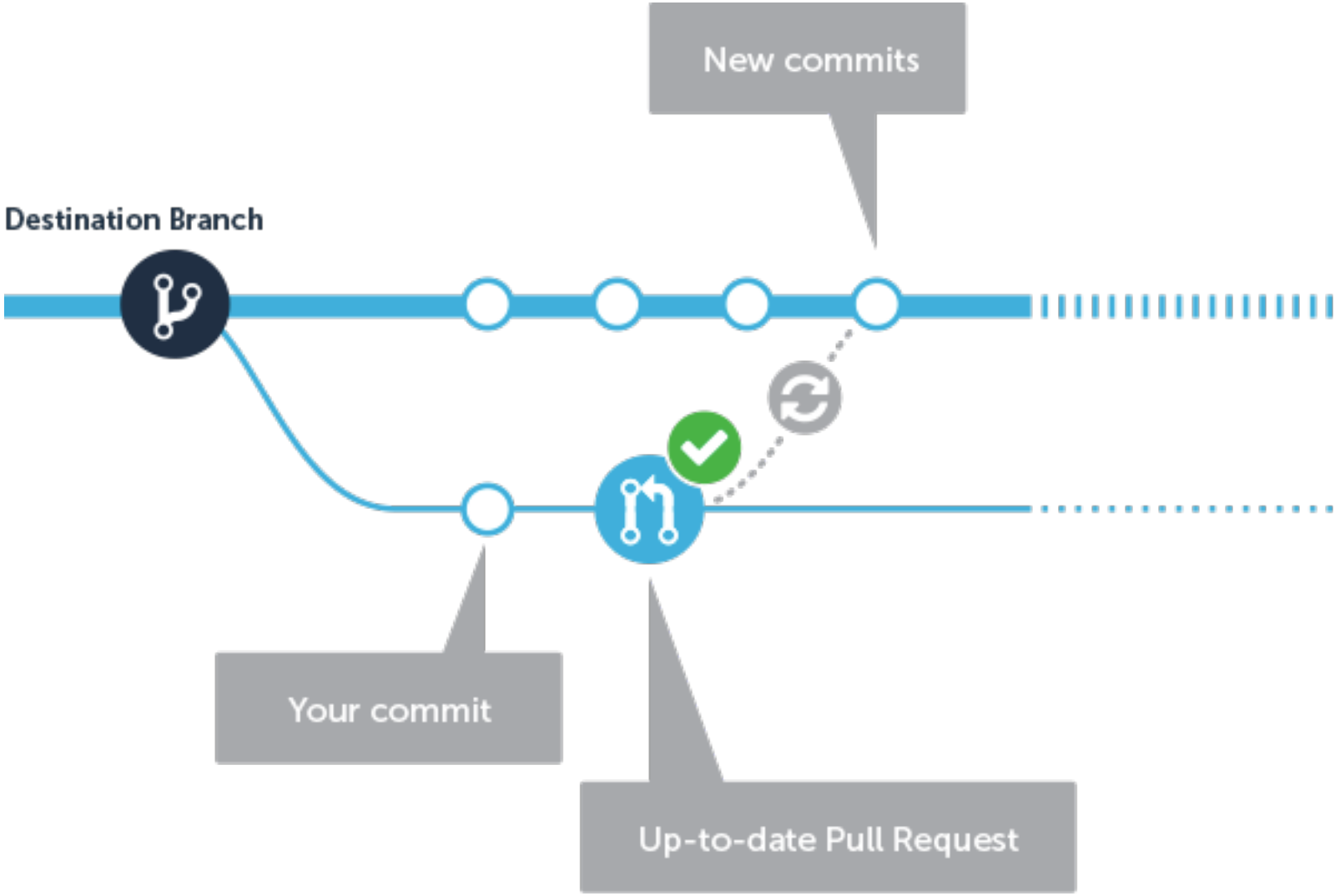

master
branch
merge
new_feature
time

Task 3

1. Create a branch of your repo
2. Name it “edited”
3. Edit README file
4. Commit changes

5. Pull request & merge

- git's way of putting a forked history back together again
- review code before merging into the main branch
- merge to the main branch

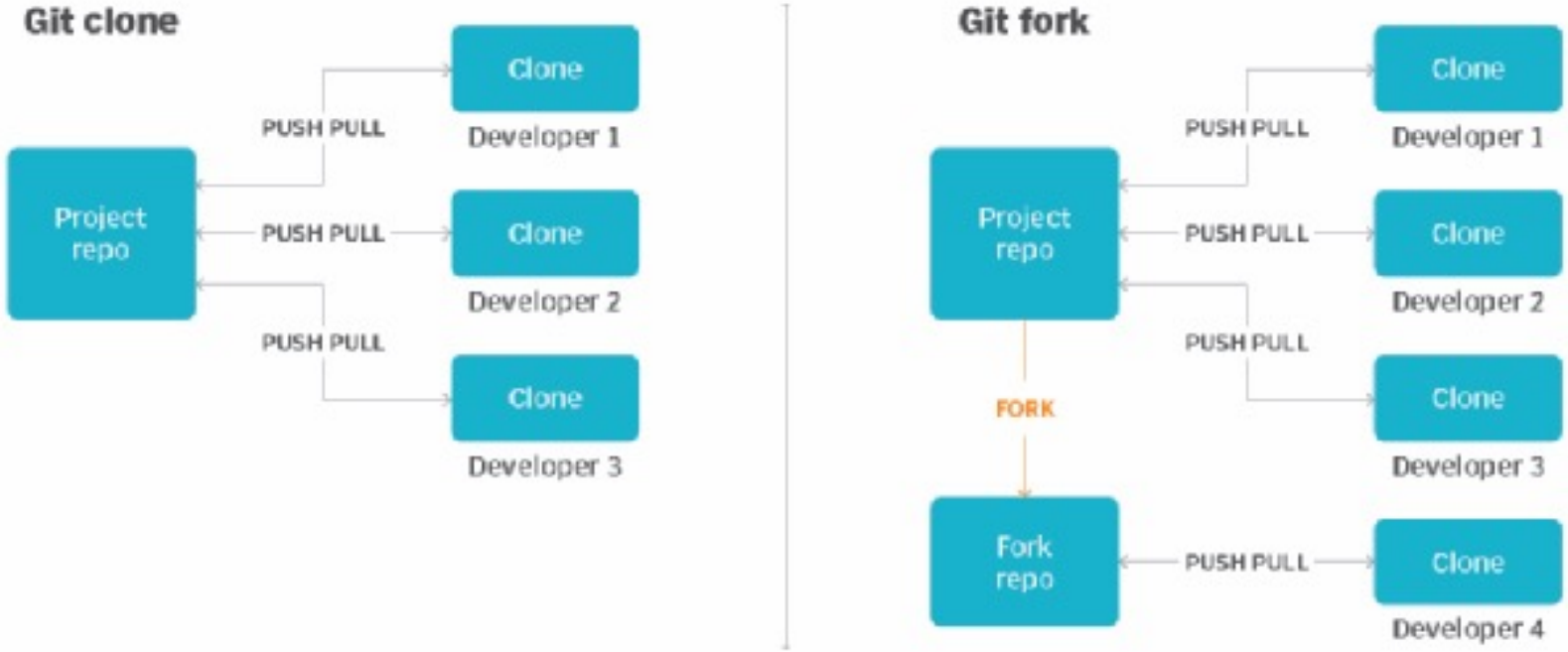

New commits
Destination Branch
Your commit
Up-to-date Pull Request

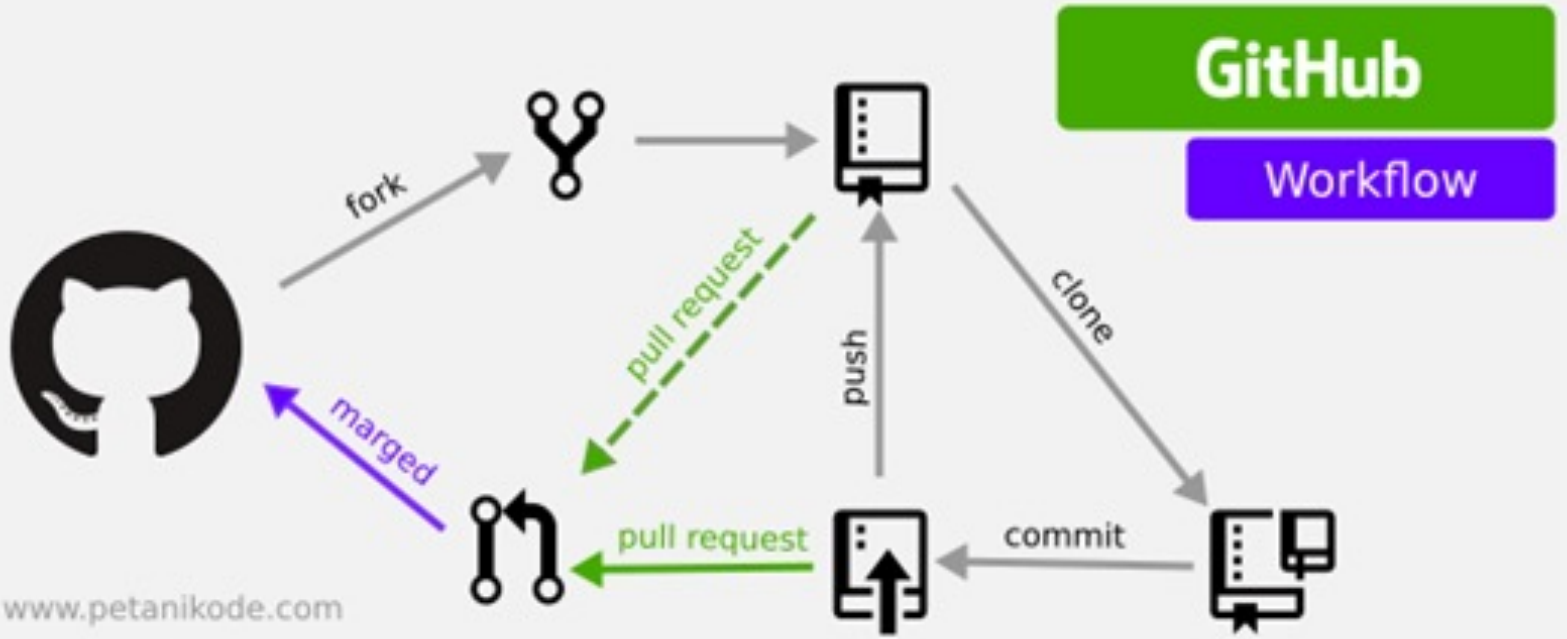
Task 4

1. Create a pull request on previous changes in task 3.
2. Review the changes
3. Confirm merge

Git clone vs. fork

Developers who work on a common codebase will clone the repository and then perform push and pull operations to synchronize their changes. In contrast, a fork creates a new codebase and updates to the fork are not synchronized with the original repo.





Source: W3docs - Git Repository | W3Docs Git Online Tutorial

Thanks!

Any questions?

You can find me at:

- http://myominnoo.github.io/
- dr.myominnoo@gmail.com