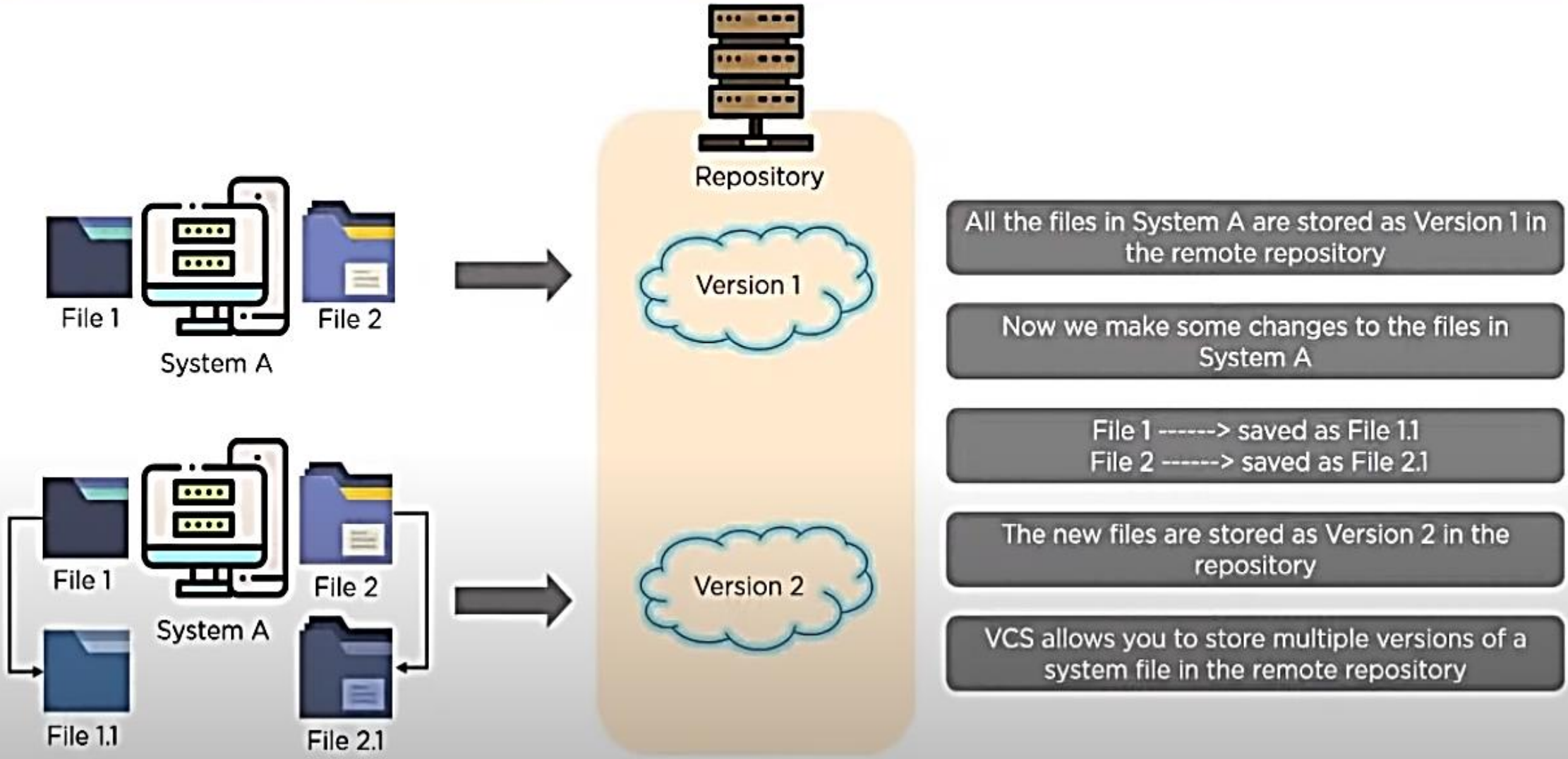
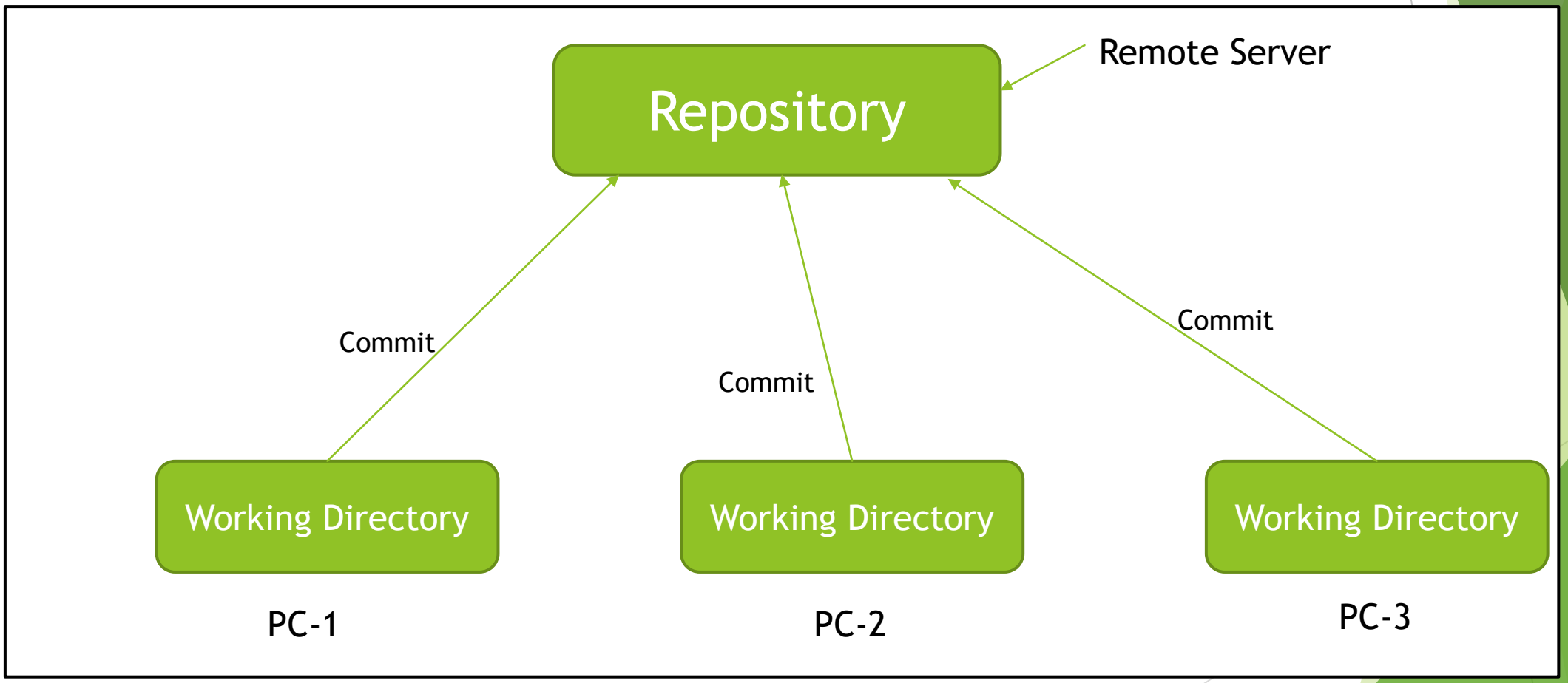
Prepared by: Ms.Pragati Patil

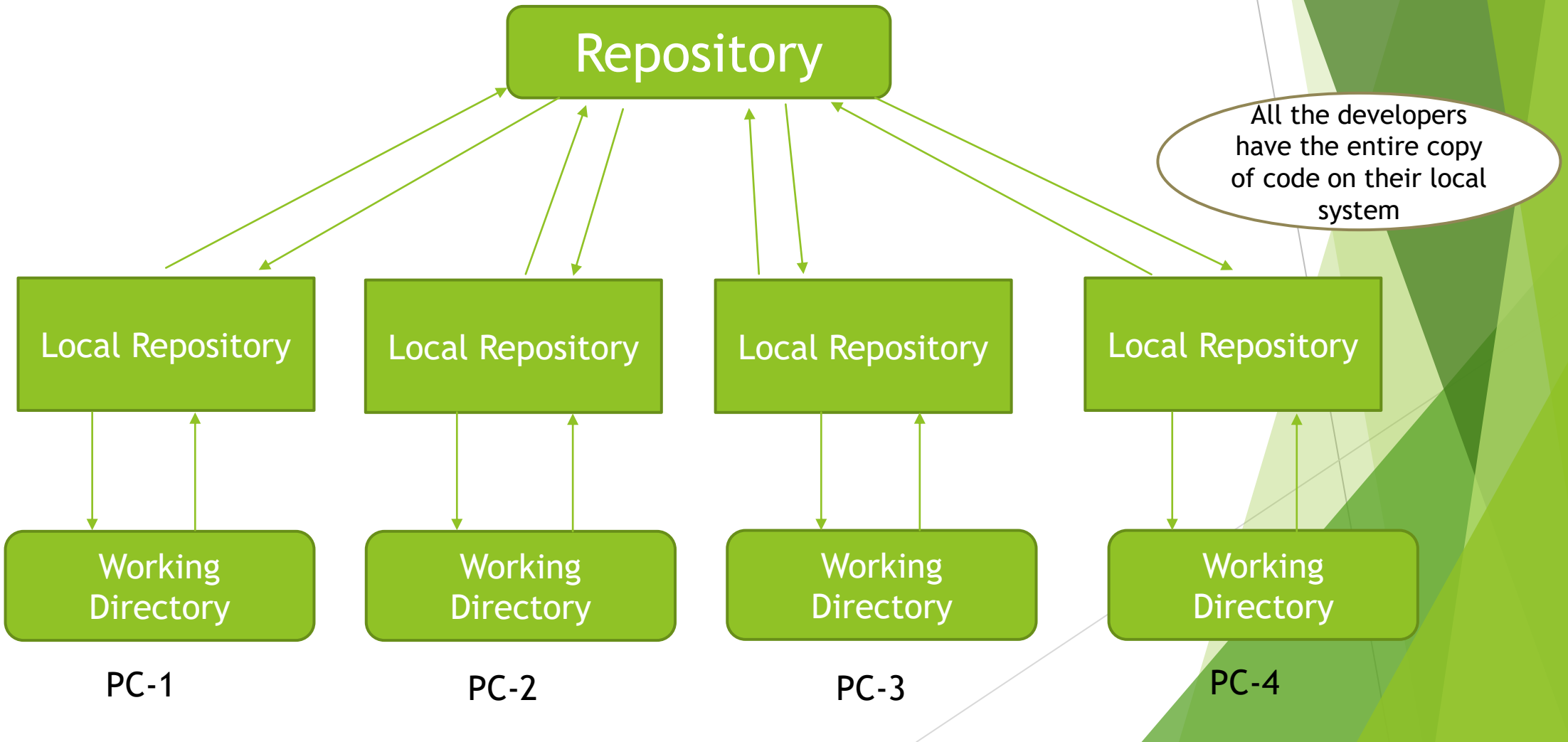
Version Control System (VCS)



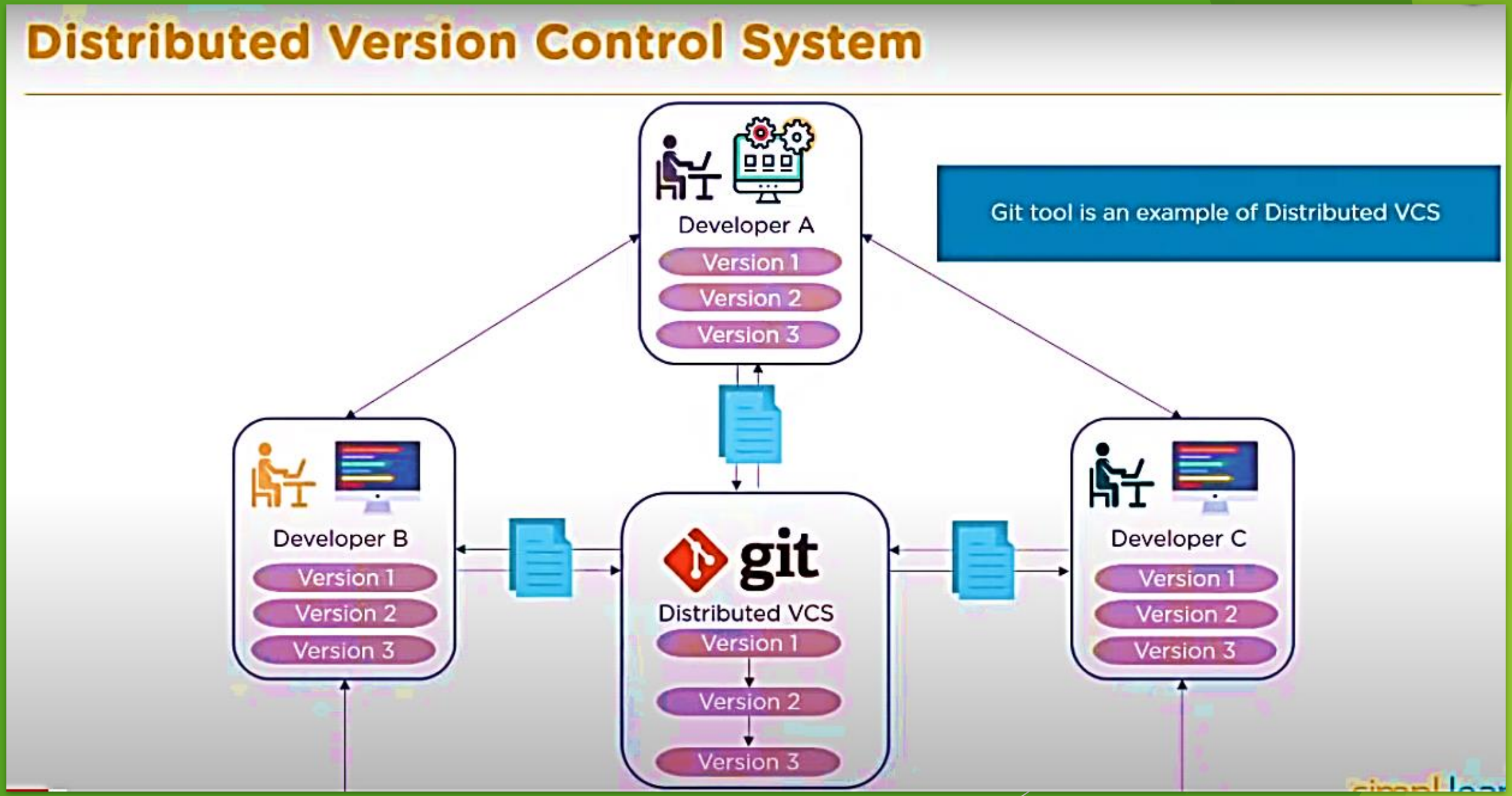
Centralized Version control system(CVCS)



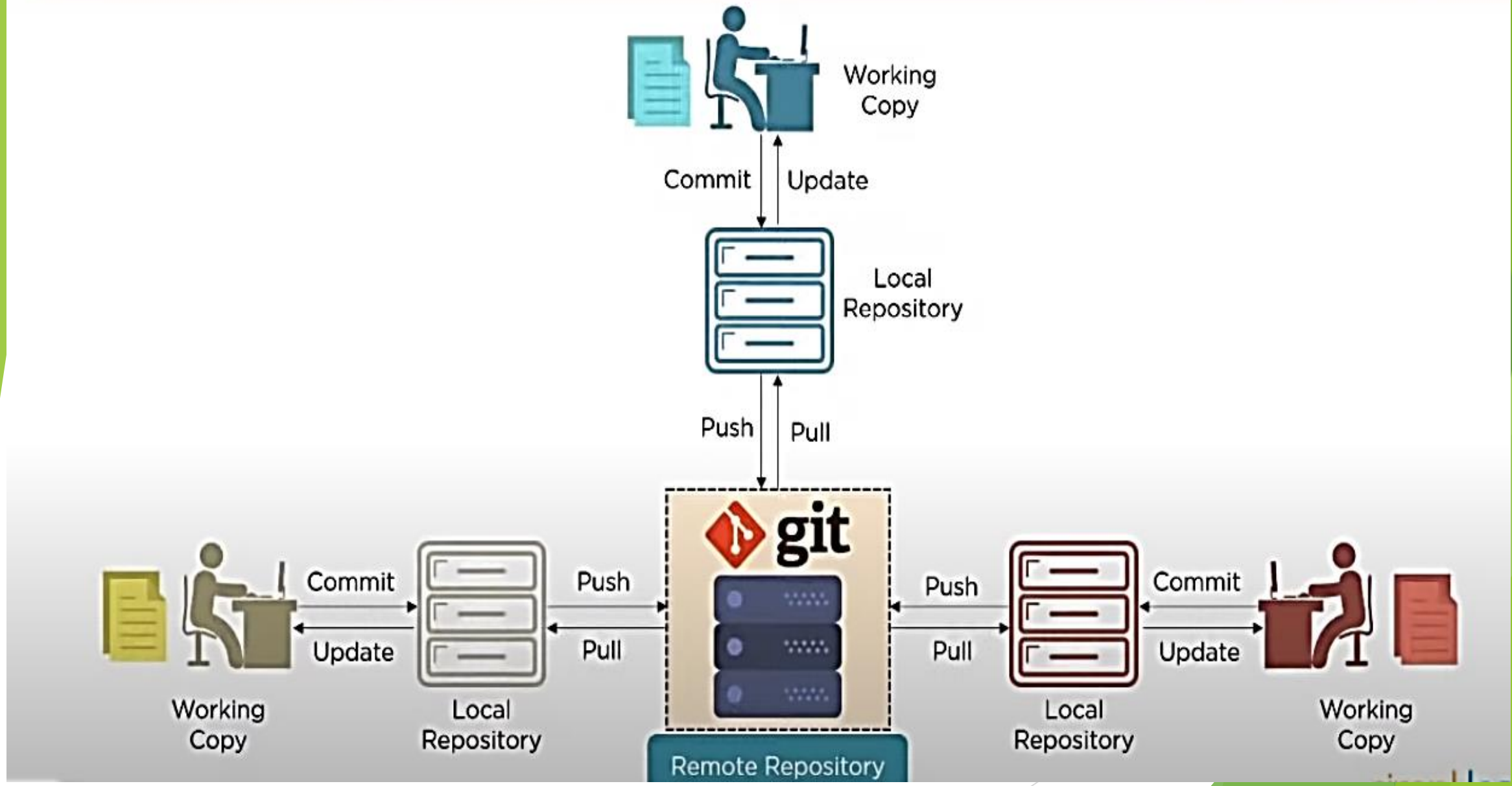
Distributed Version Control System(DVCS)



Distributed Version Control System



Distributed Version Control System



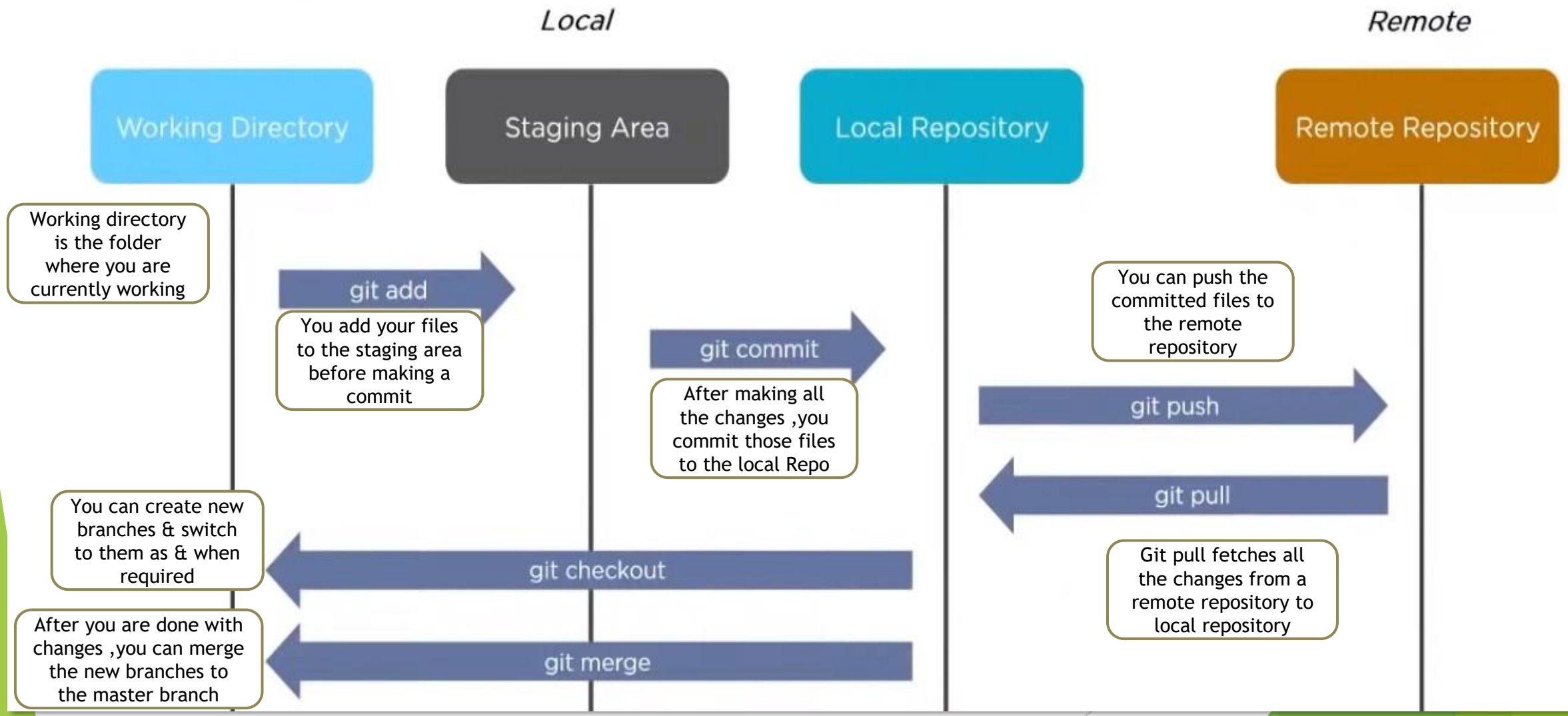
What is Git

- Git is distributed version control tools used for source code management.
- Git is used to track changes in the source code.
- Allows multiple developers to work together.
- Has the ability to handle large projects efficiently.

Difference between Git and GitHub

Git	GitHub
1. Git is a software tool	1. GitHub is a service
2. It is installed on the local system	2. It is hosted on the web
3. It is used to manage different versions of the source code	3. It is used to have a copy of the local repository code
4. It provides the command line to interact with the files	4. It provides a graphical interface to store a file

Git Architecture



Advantages of Git

- Free & open source.
- Fast & small in size

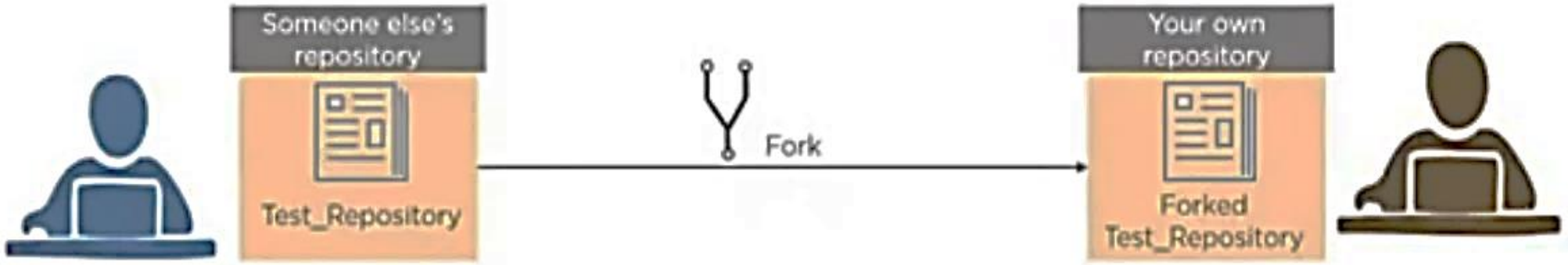
(As most of the operations are performed locally, therefore it is fast)

- Security – Git uses a common cryptographic hash function called secure hash system(SHA-1) to name and identify objects within database.
- No need of powerful hardware.
- Easier branching – if we create a new branch, it will copy all the codes to the new branch.

Fork and Clone

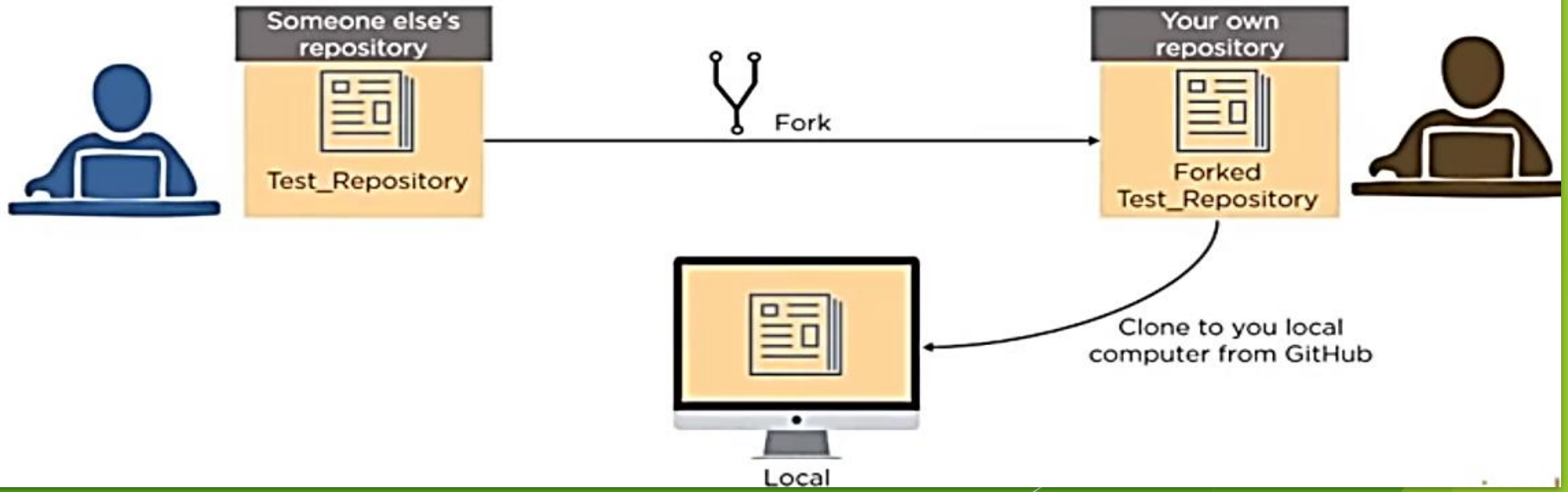
Fork and Clone

Git allows you to fork an open source repository. When you fork a repository, you create a copy of it on your GitHub account.

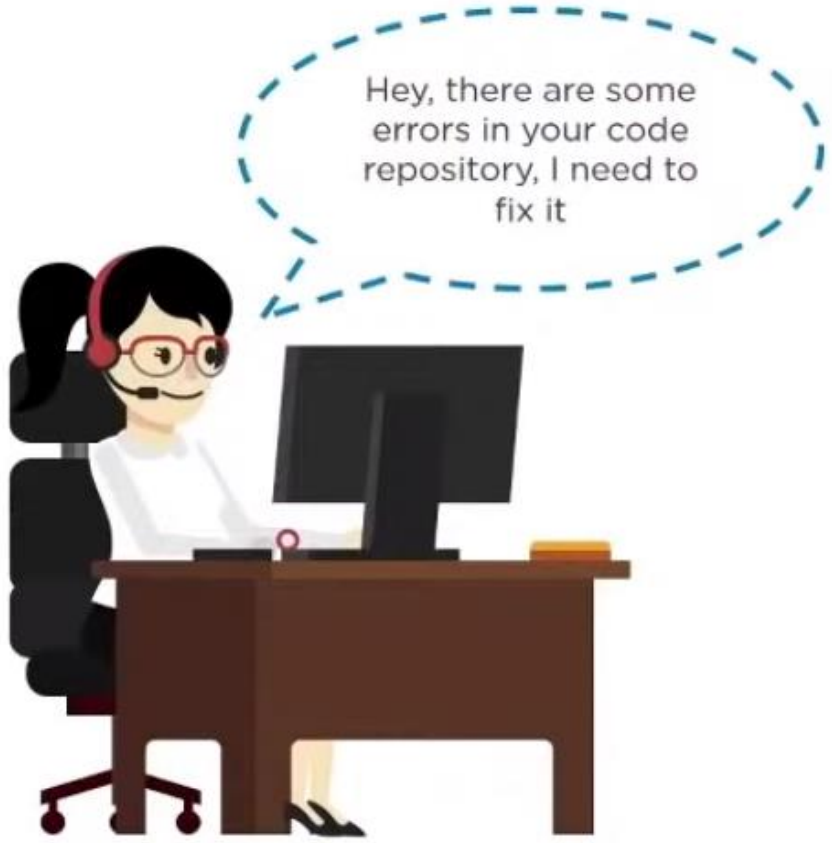


Fork and Clone

After you fork a repository, you can clone it and have a copy of it on your local system.



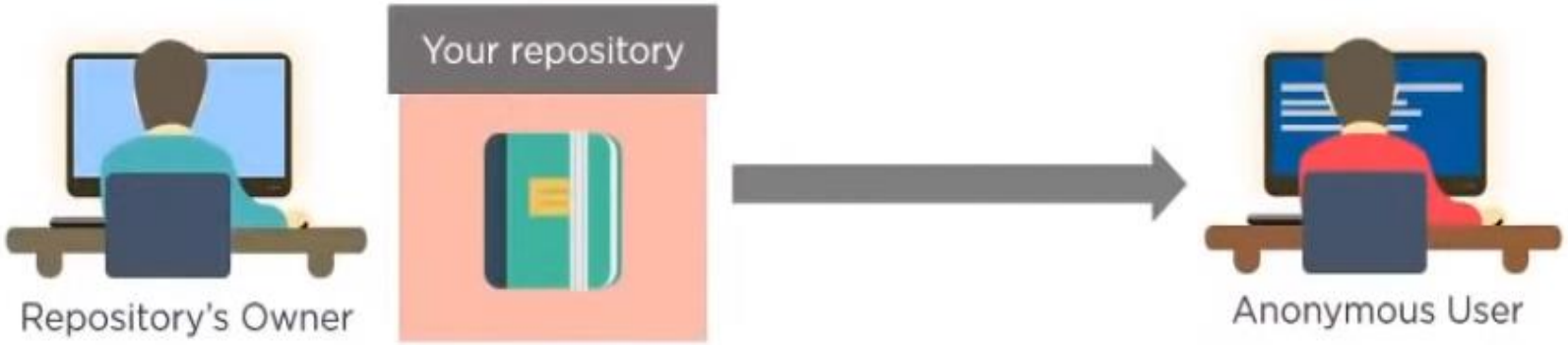
Collaborators



Collaborators

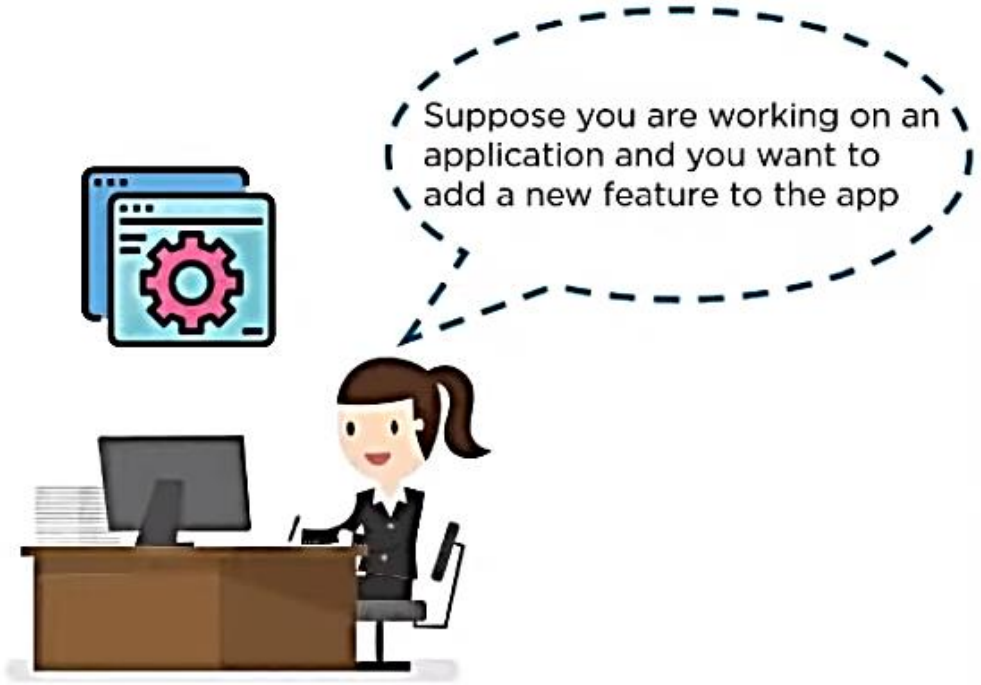
GitHub allows you to work with users from all over the world at any given time

Collaborators are GitHub users who are given permission to edit a repository owned by someone else

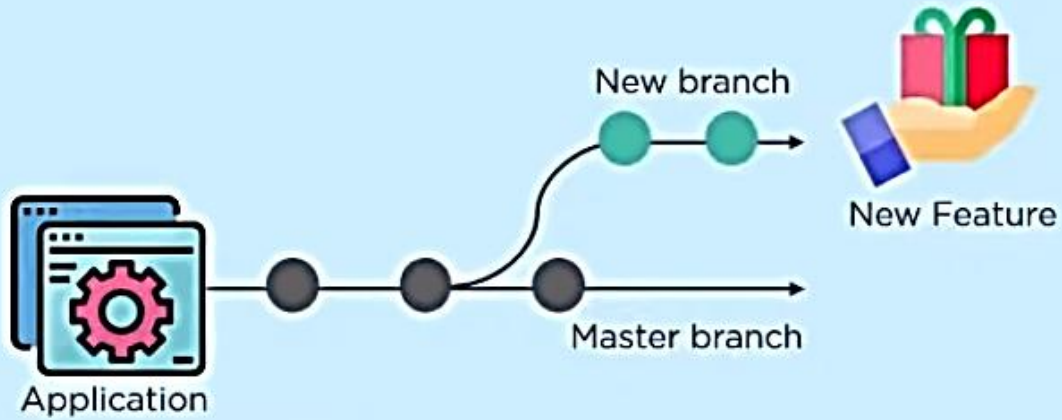


- ❑ Access all the files
- ❑ View the files
- ❑ Make changes to the files
- ❑ Commit the files

Branch in Git

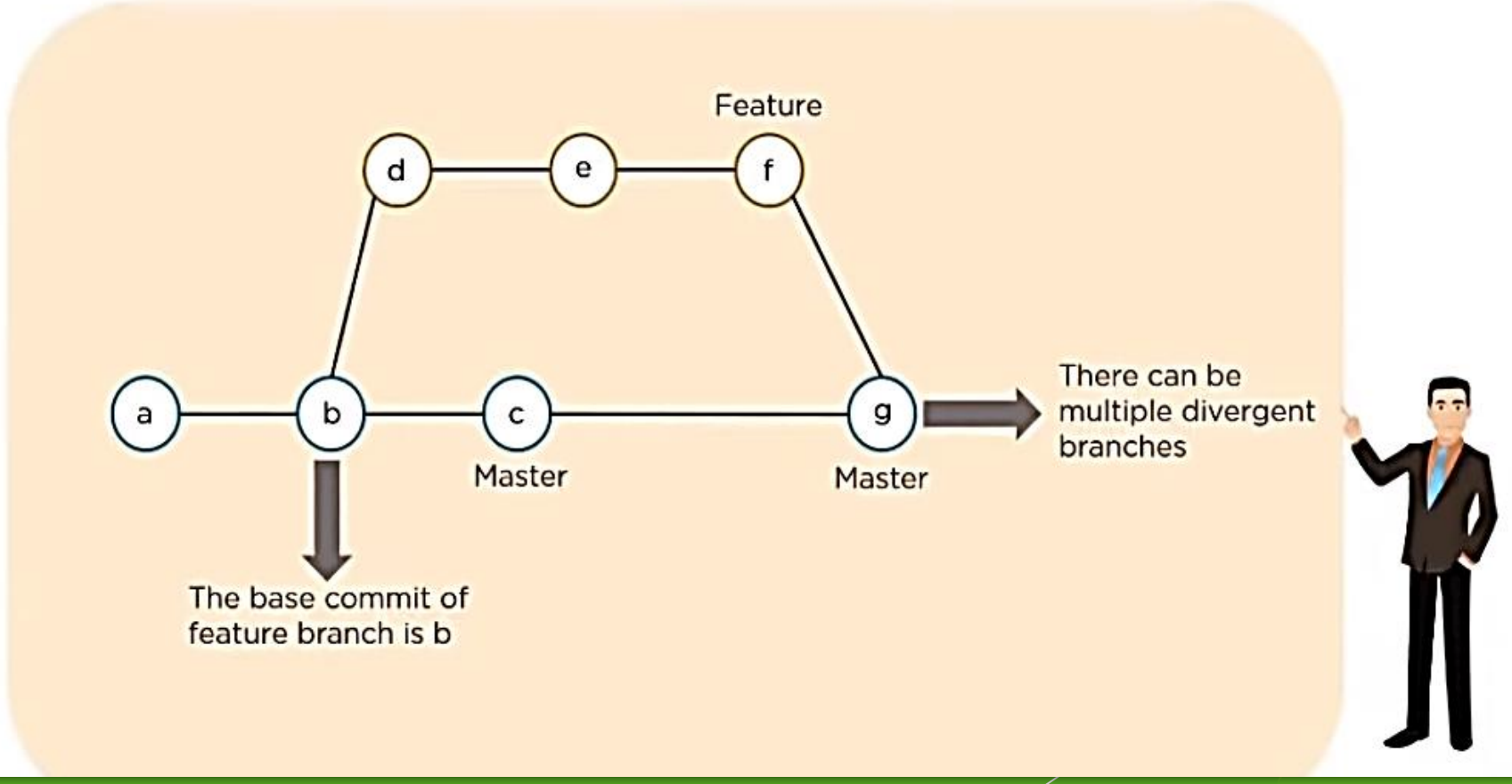


You can create a new branch and build the new feature on that branch

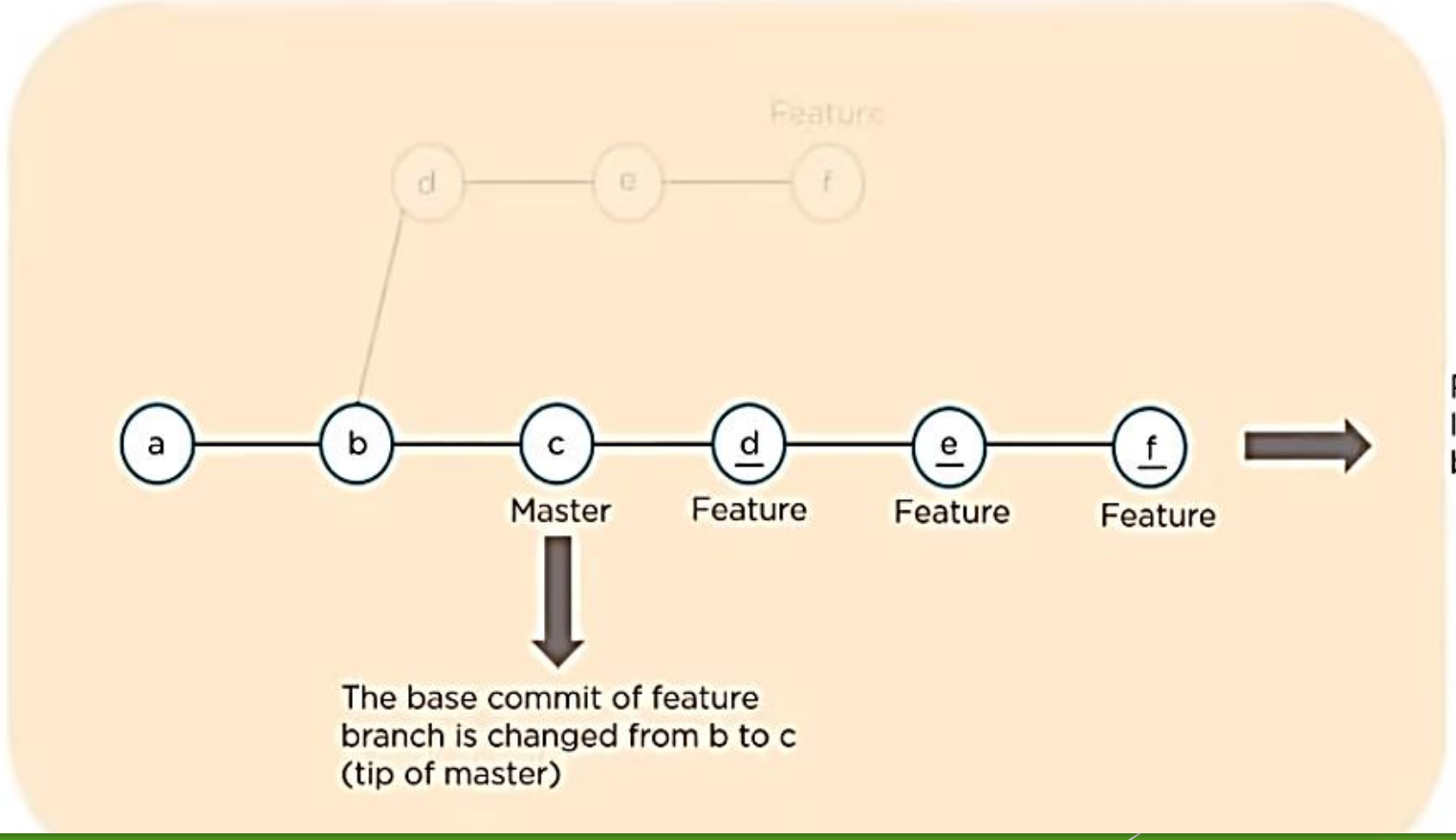


- By default, you always work on the master branch
- The circles on the branch represent various commits made on the branch

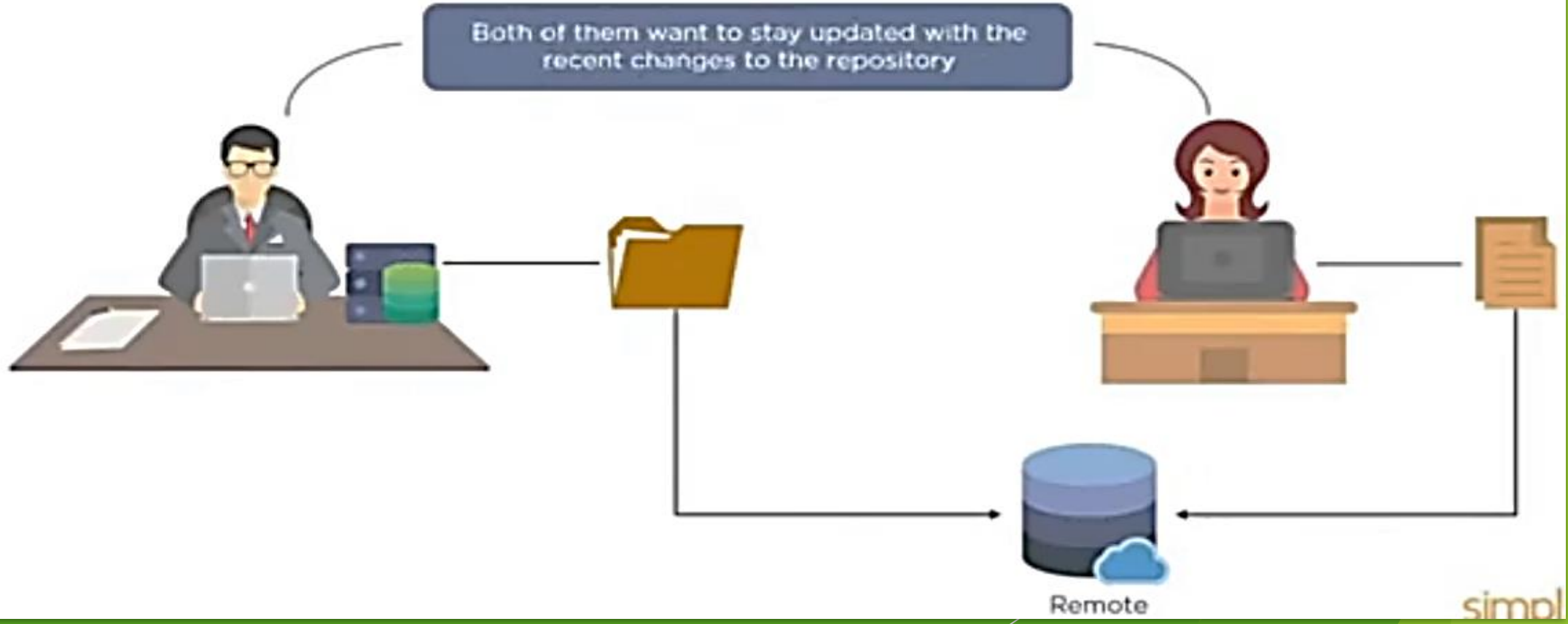
Git Merge


Feature
d
e
f
a
b
c
g
Master
Master
There can be multiple divergent branches
The base commit of feature branch is b

Git Rebase

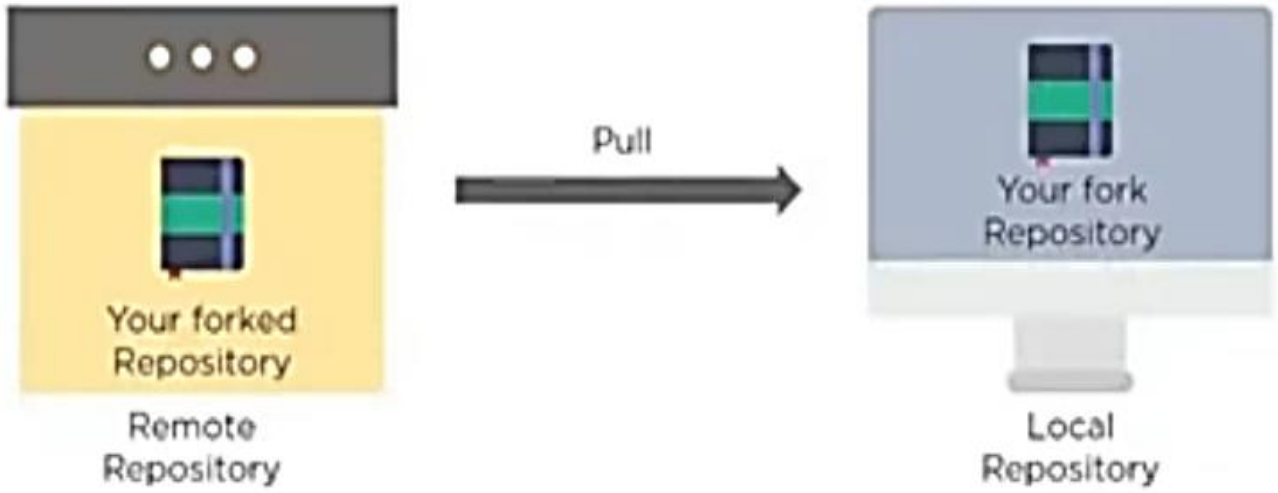

Feature
d
e
f
a
b
c
d
e
f
Master
Feature
Feature
Feature
The base commit of feature branch is changed from b to c (tip of master)
Resulting history is linear with no divergent branches

Pull from a Remote



Pull from a Remote

You can pull in any changes that have been made from your forked remote repository to the local repository

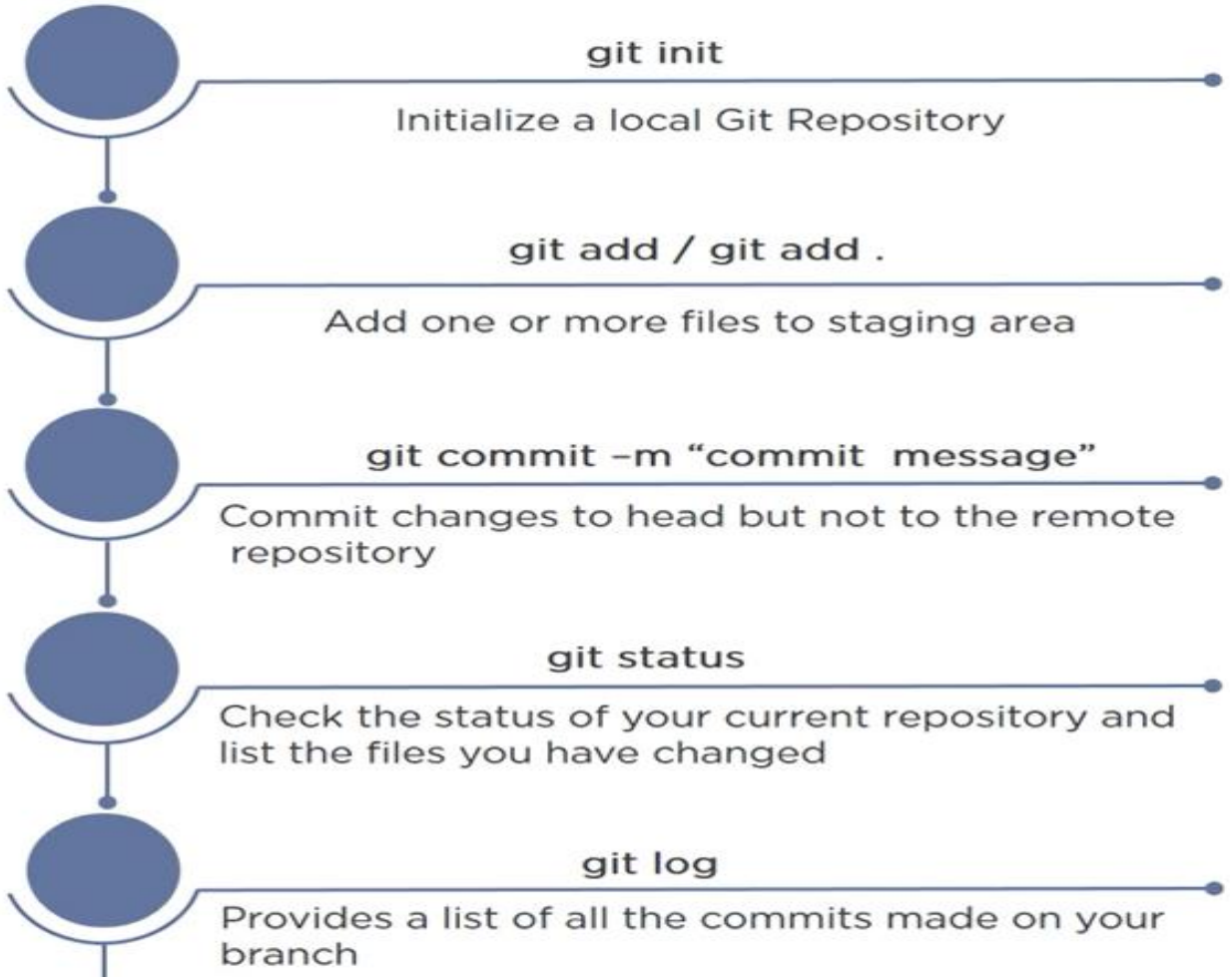


- Use the following command to check if there has been any change

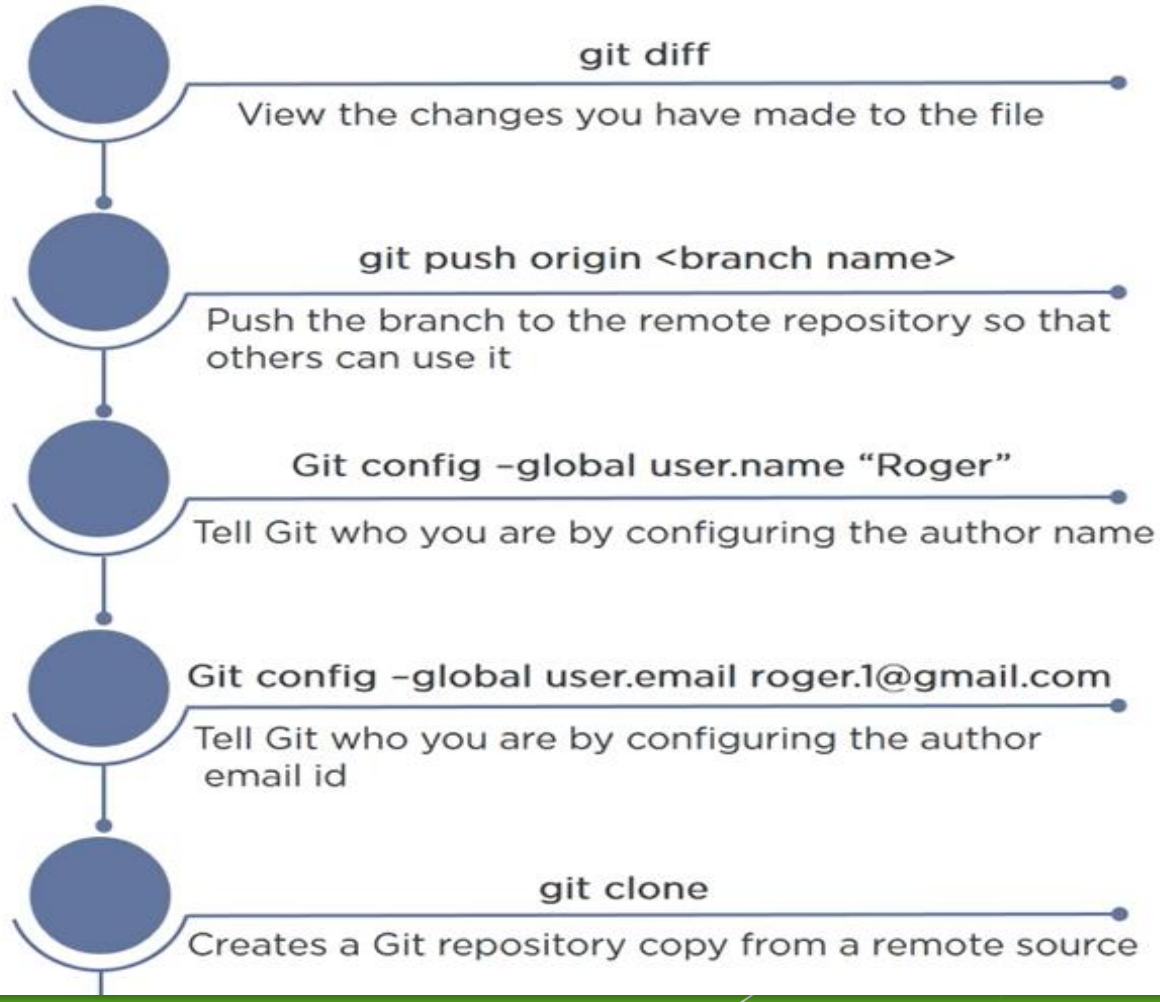
$ git pull <RemoteName> <BranchName>

- If there is no change, it will notify "Already up-to-date". If there is an change, it will merge those changes to your local repository

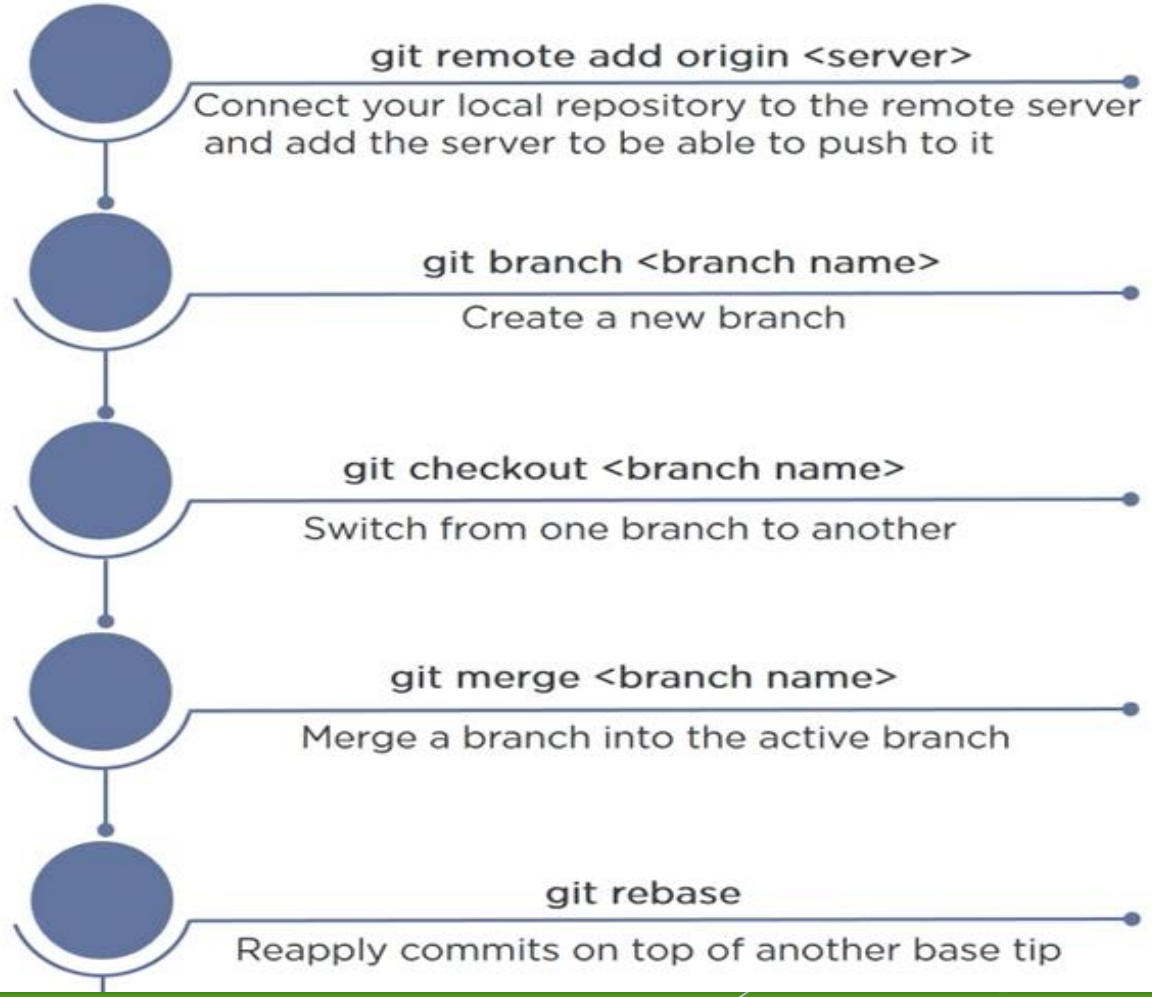
Popular Git Commands



Popular Git Commands



Popular Git Commands



References:

- **Git Tutorial For Beginners | What is Git and GitHub? | Git Tutorial | DevOps Tutorial | Simplilearn**