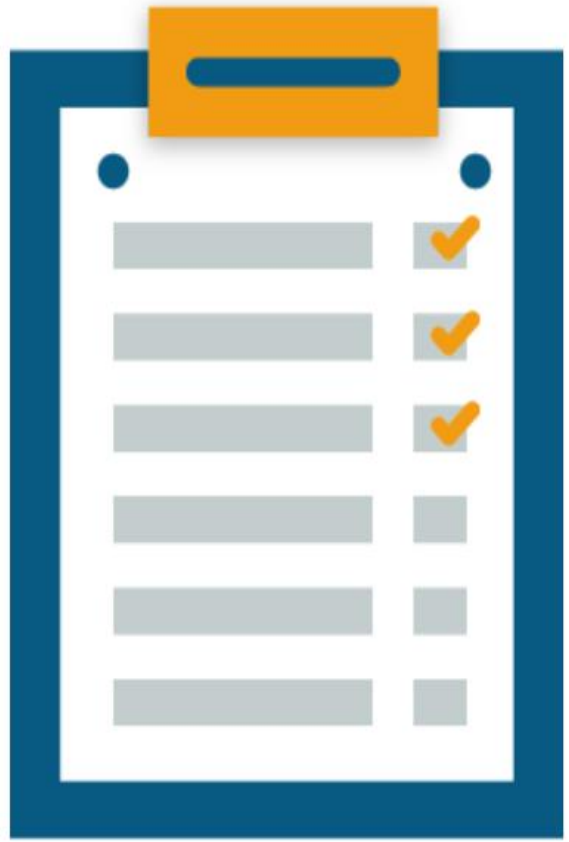
Module 1 – Version Control with Git

Topics

Following are the topics covered in this module:

- Version Control
- Git Introduction
- Git Installation
- Commonly used commands in Git
- Working with Remote repository



Why Need Version Control?

Why Need Version Control?

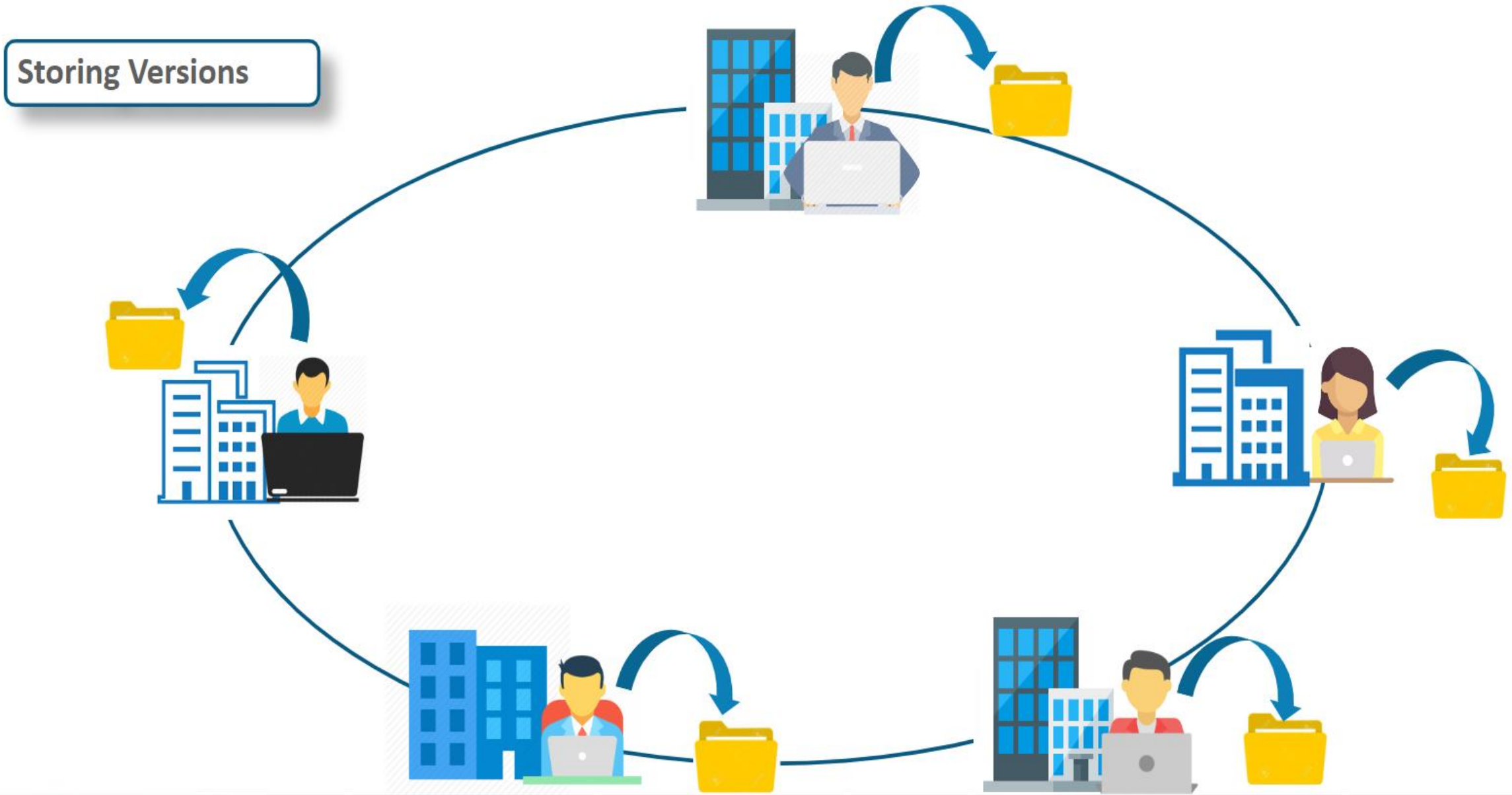
Let's consider a multinational company that has its offices and employees all around the globe

Why Need Version Control?

Problems that may arrise

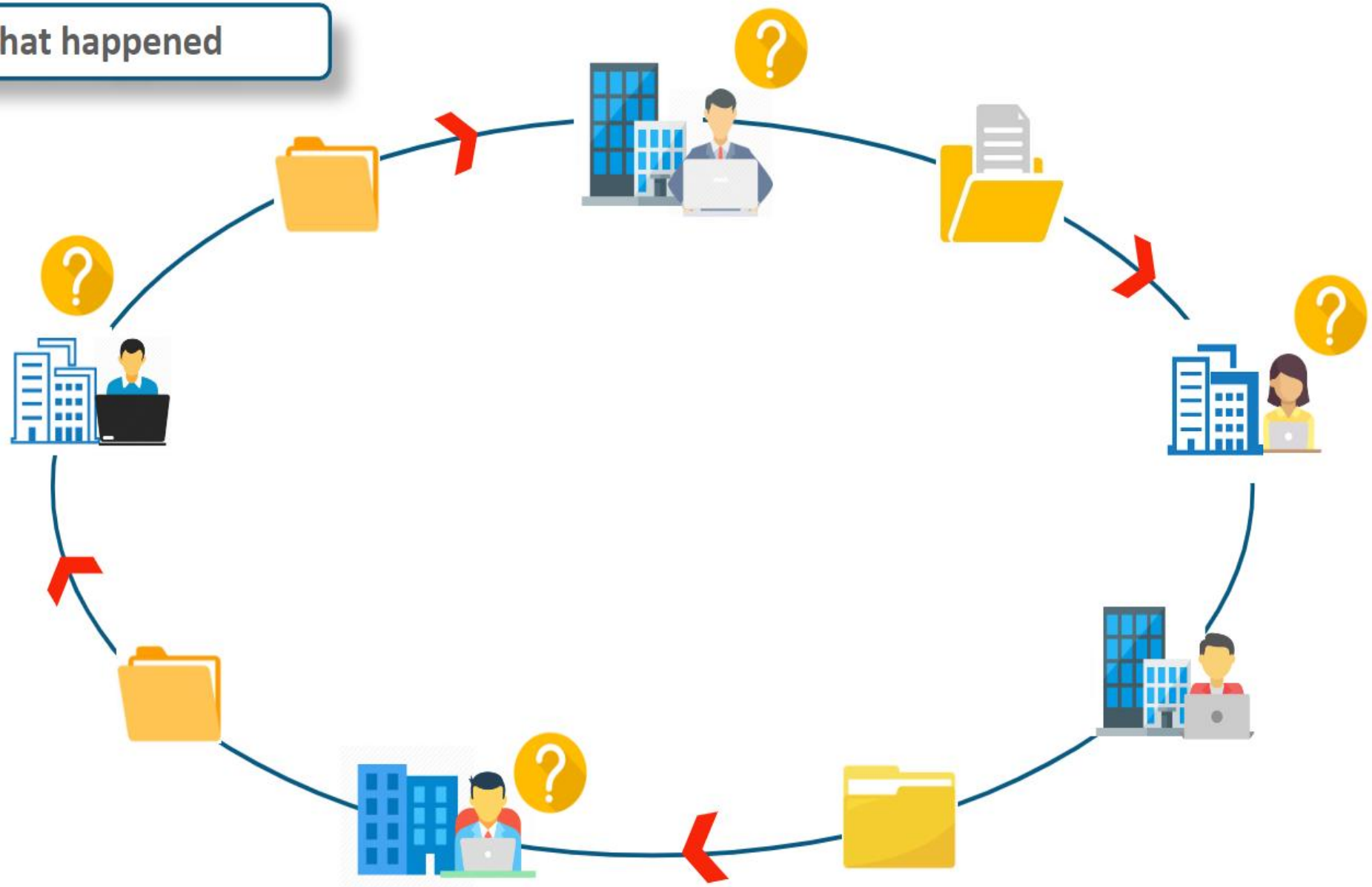
Collaboration

Why Need Version Control?



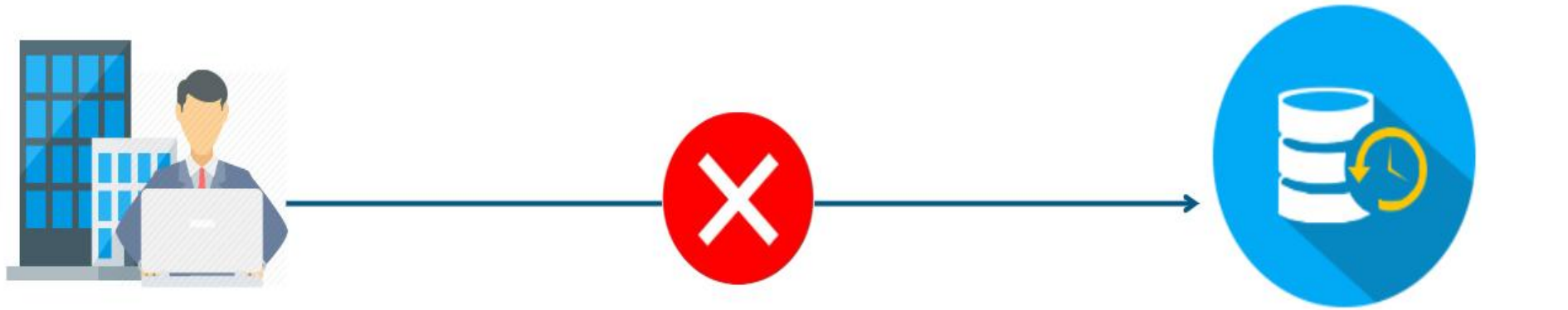
Why Need Version Control?

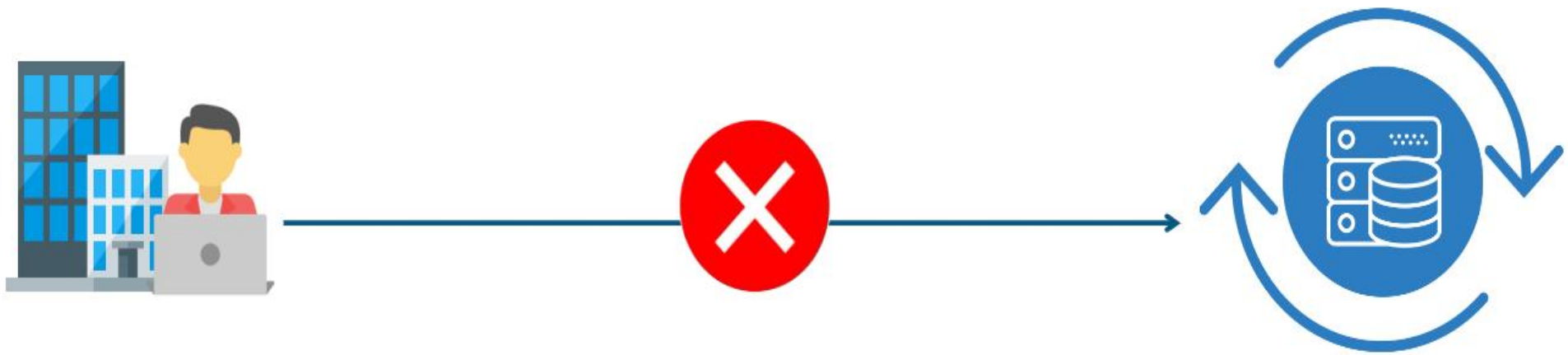
Figuring out what happened



Why Need Version Control?

Backup





Why Need Version Control?

Solution for all these problems

All these problems can be solved with the help of a "**Version Control System**"

Why Need Version Control?

- Version control helps the teams to solve these kinds of problems, by tracking every individual change by each contributor and helps prevent concurrent work from conflicting.
- A version control software supports a developer's preferred workflow without imposing one way of working.

Issues Without Version Control

- Once saved, all the changes made in the files are permanent and

  cannot be reverted back

- No record of what was done and by whom

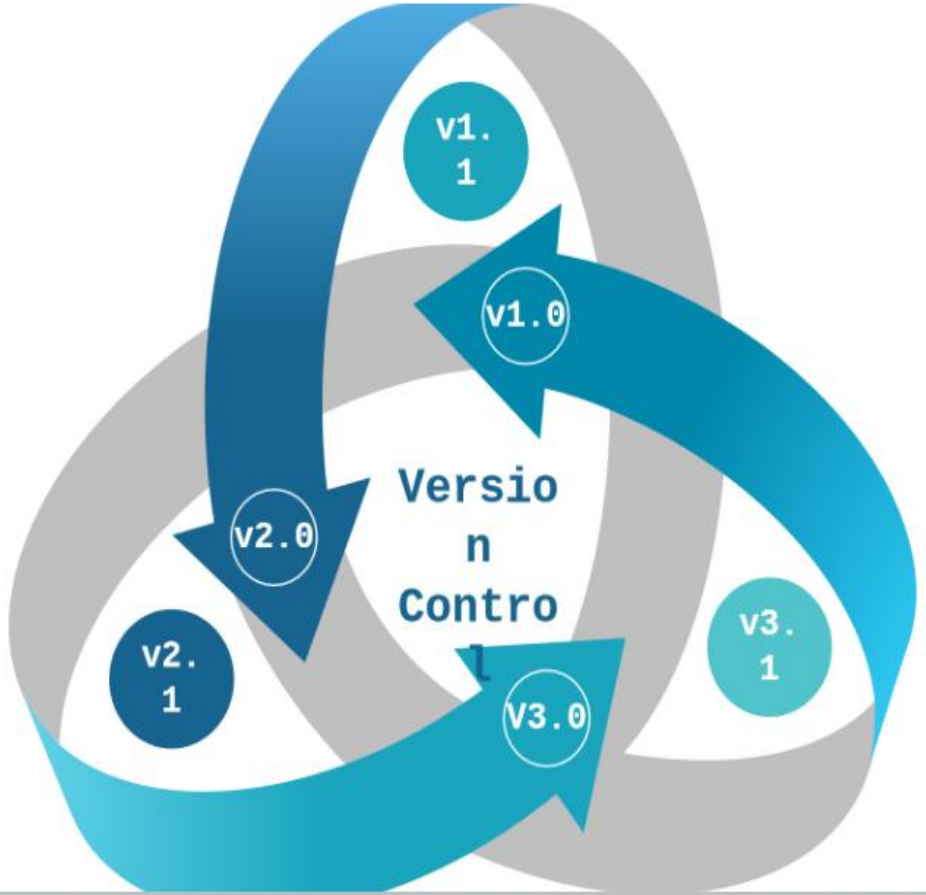
- Downtime that can occur because of a faulty update could cost

  Millions in losses

Version Control

What Is Version Control?

Version Control is a system that documents changes made to a file or a set of files. It allows multiple users to manage multiple revisions of the same unit of information. It is a snapshot of your project over time.

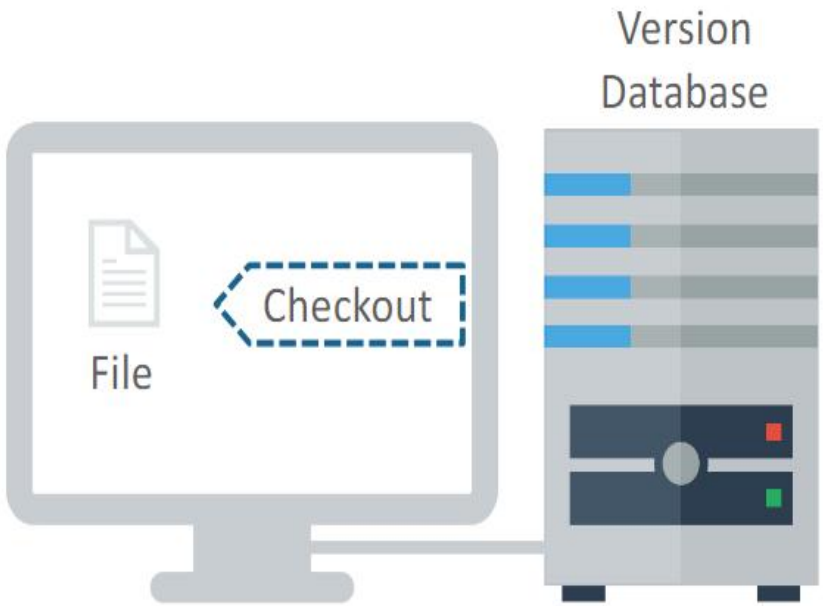


Version Control Types

Local Version Control (LVC)

- The practice of having the Version Database in the local computer

- Local database keeps a record of the changes made to files in version database

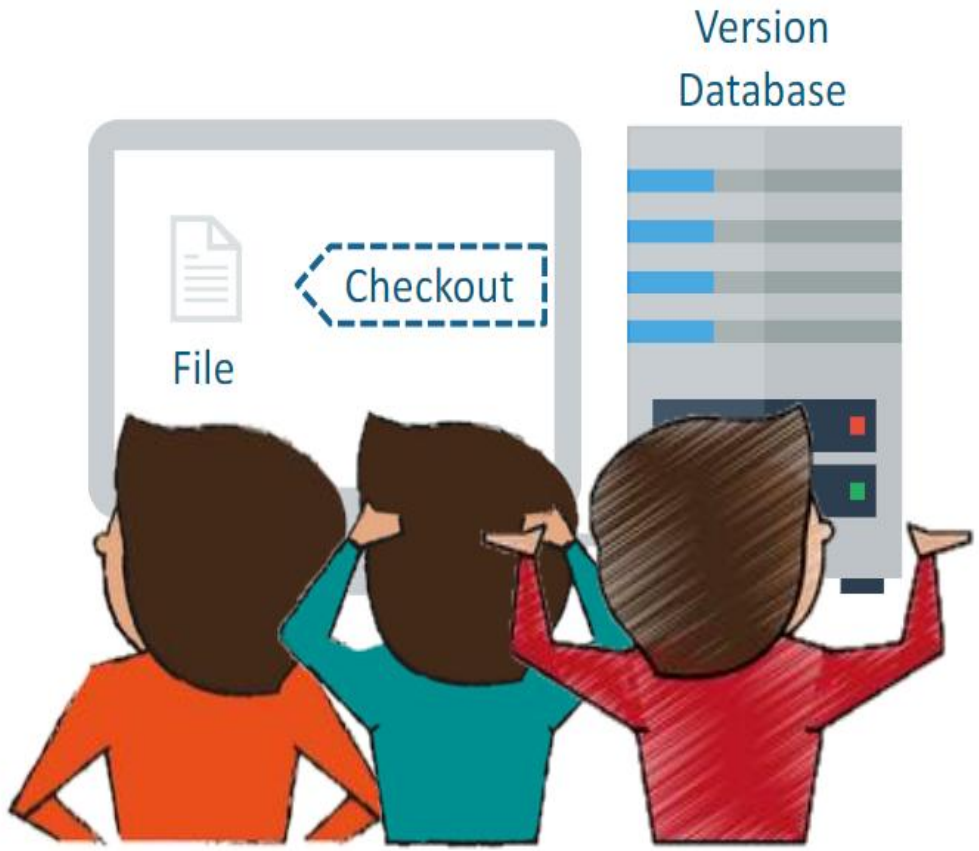


Version Control Types

Local Version Control: Issue

- **Issue:** Multiple people parallelly working on the same project

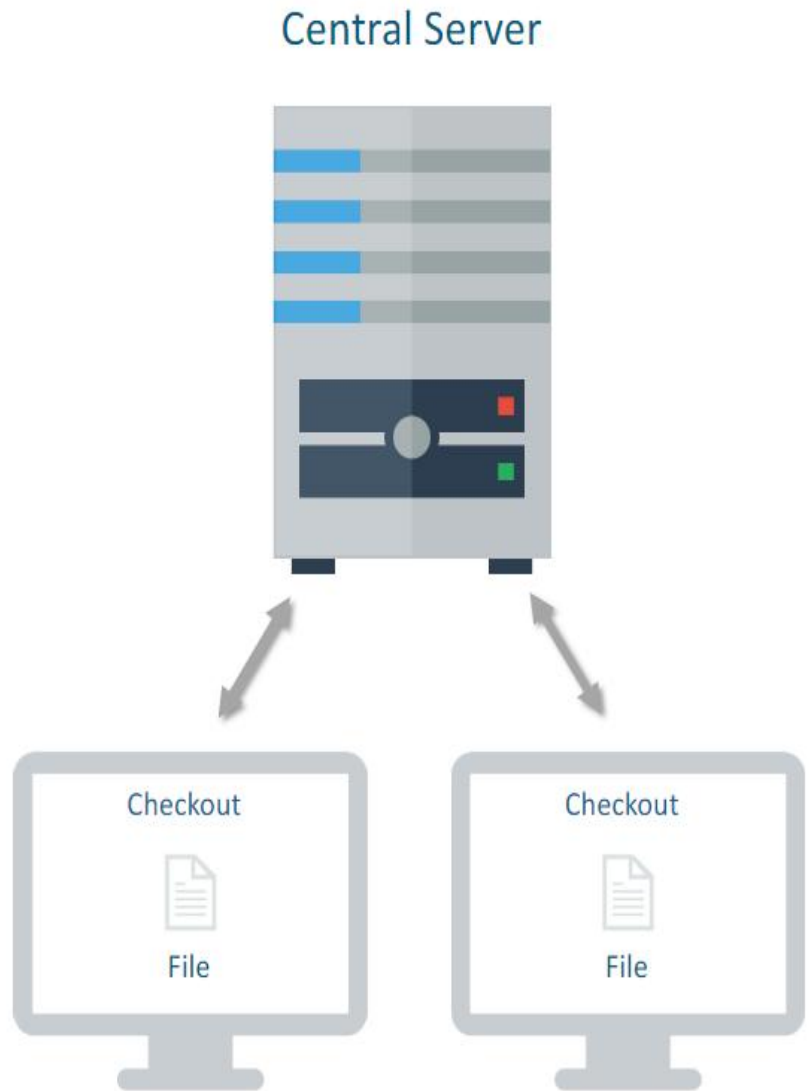
- **Solution:** Centralized Version Control



Version Control Types

Centralized Version Control (CVC)

- Local Version Control's issues are resolved by Centralized Version Control
- In CVC, a central repository is maintained where all the versioned files are kept
- Now users can checkout, and check-in files from their different computers at any time

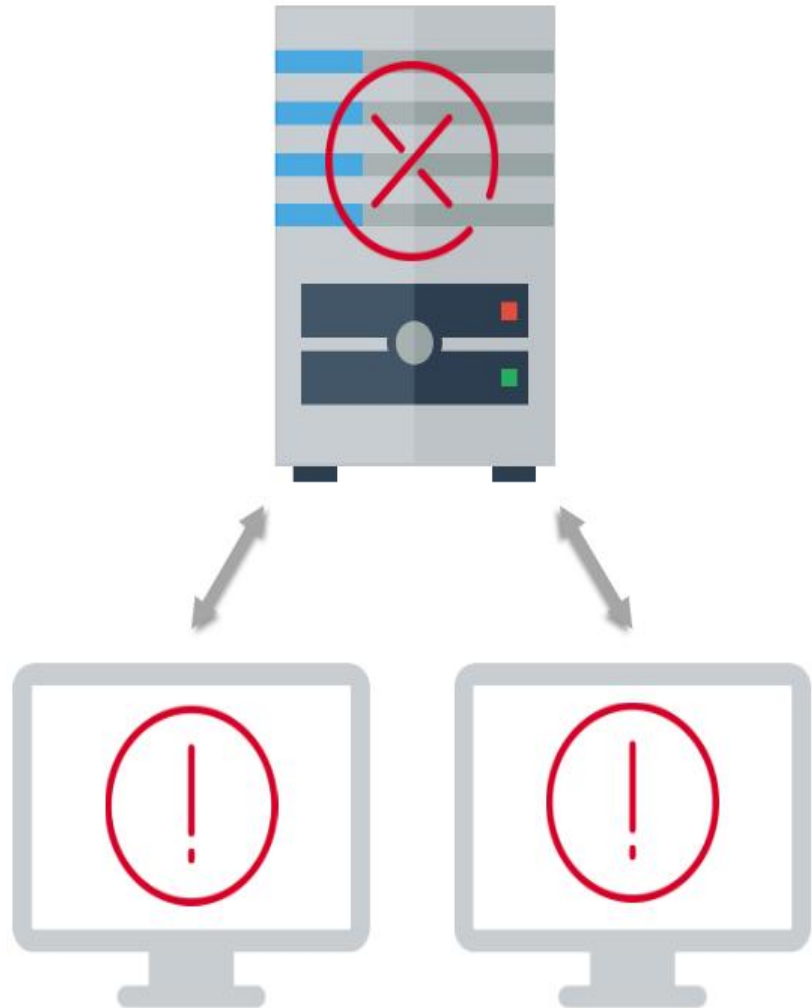


Version Control Types

Centralized Version Control: Issue

- **Issue:** In case of central server failure whole system goes down

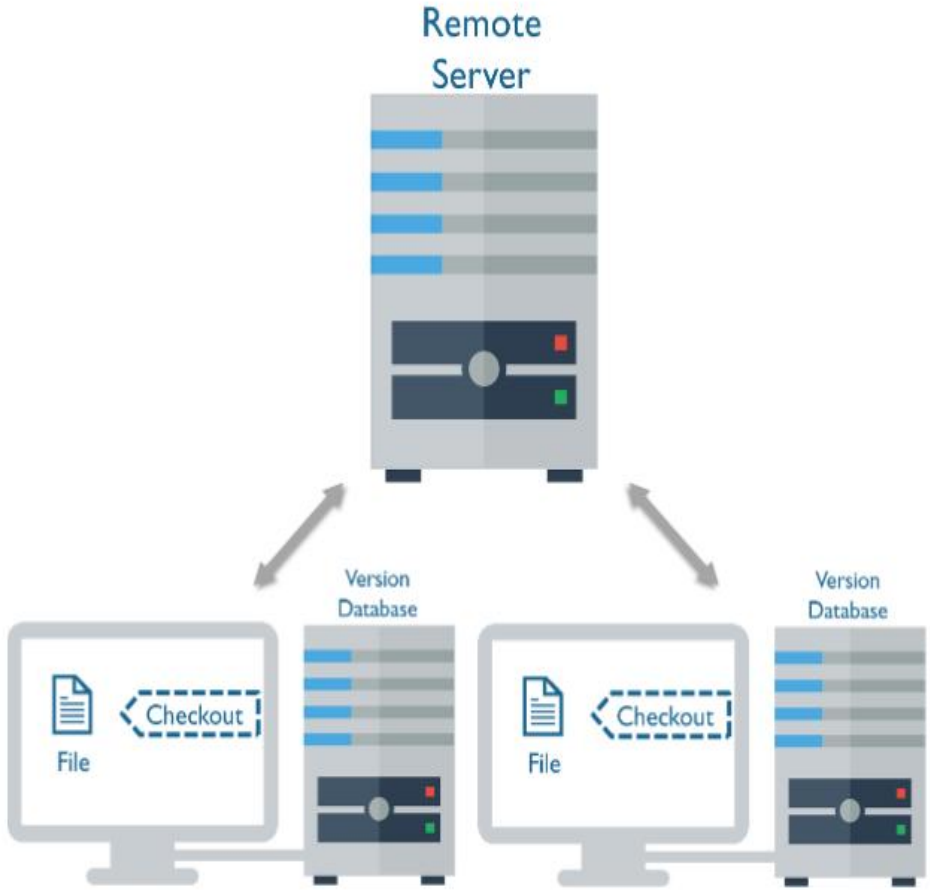
- **Solution:** Distributed Version Control

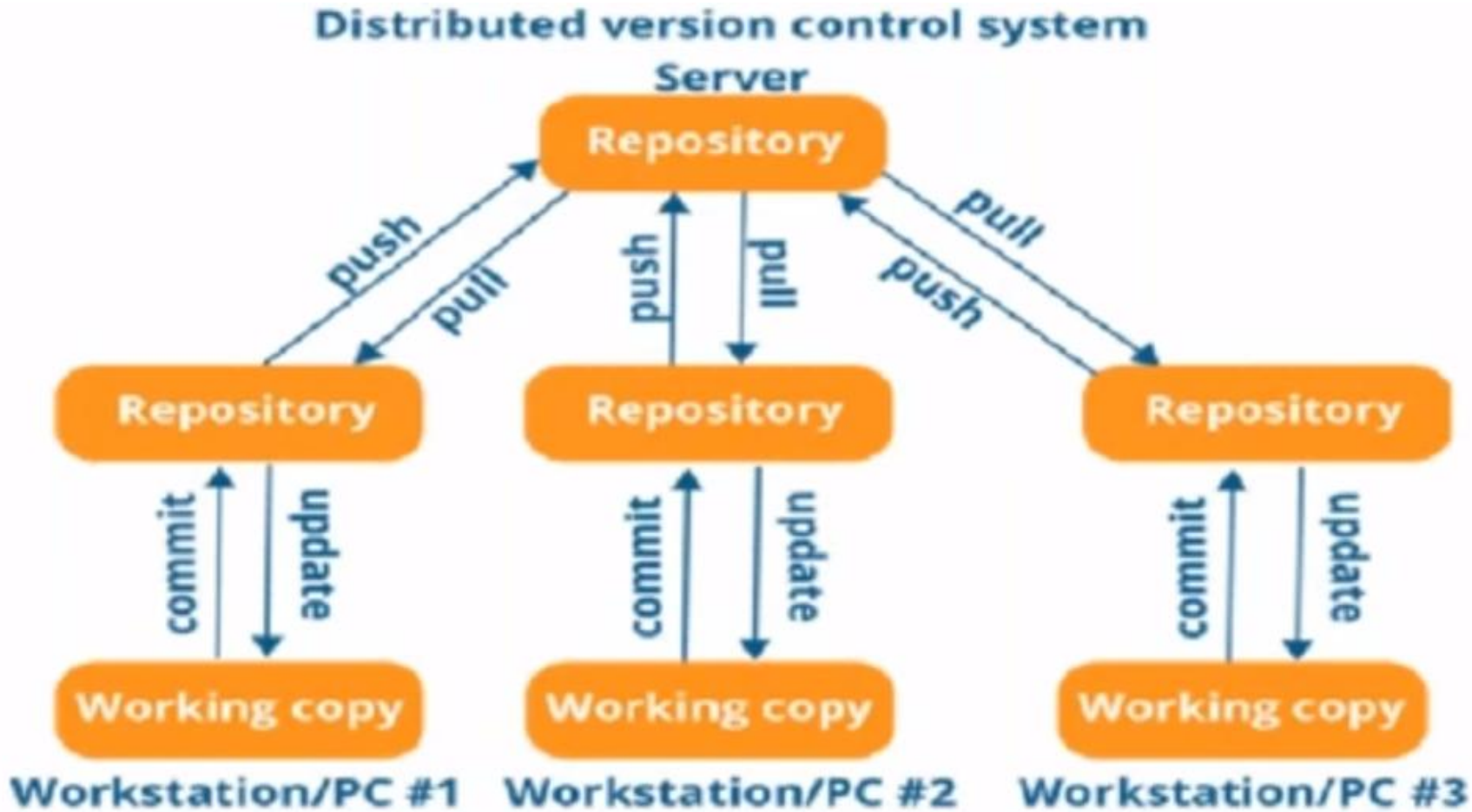


Version Control Types

Distributed Version Control

- Version Database is stored at every users' local system and at the remote server
- Users manipulate the local files and then upload the changes to the remote server
- If any of the servers die, a client server can be used to restore




Distributed version control system
Server
Repository
push
pull
push
pull
pull
push
Repository
Repository
Repository
commit
update
commit
update
commit
update
Working copy
Working copy
Working copy
Workstation/PC #1
Workstation/PC #2
Workstation/PC #3

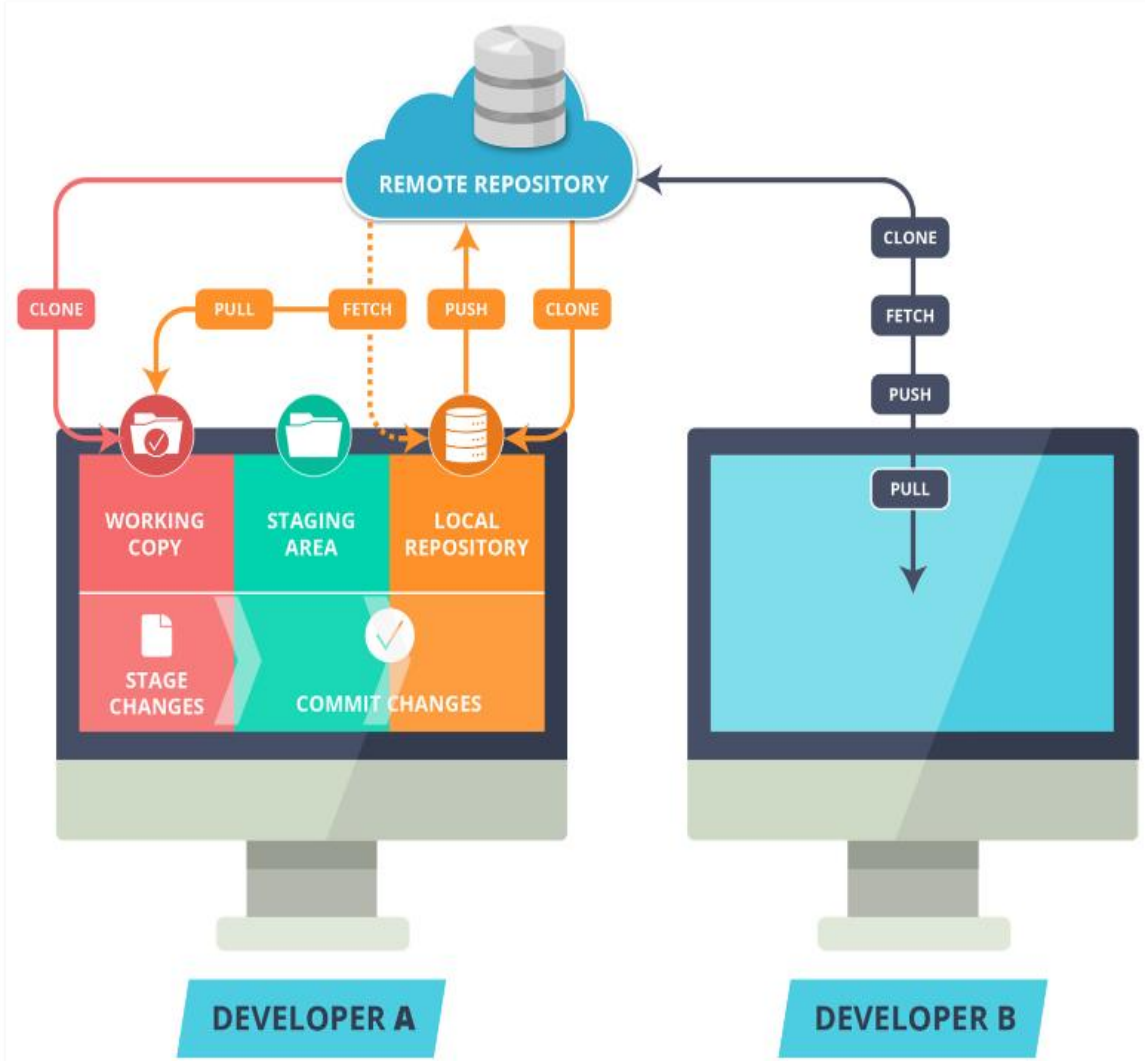
Introduction to Git

What is Git?

Git is an open-source Distributed Version Control System(DVCS) which records changes made to the files laying emphasis on **speed**, **data integrity** and **distributed**, **non-linear workflows**

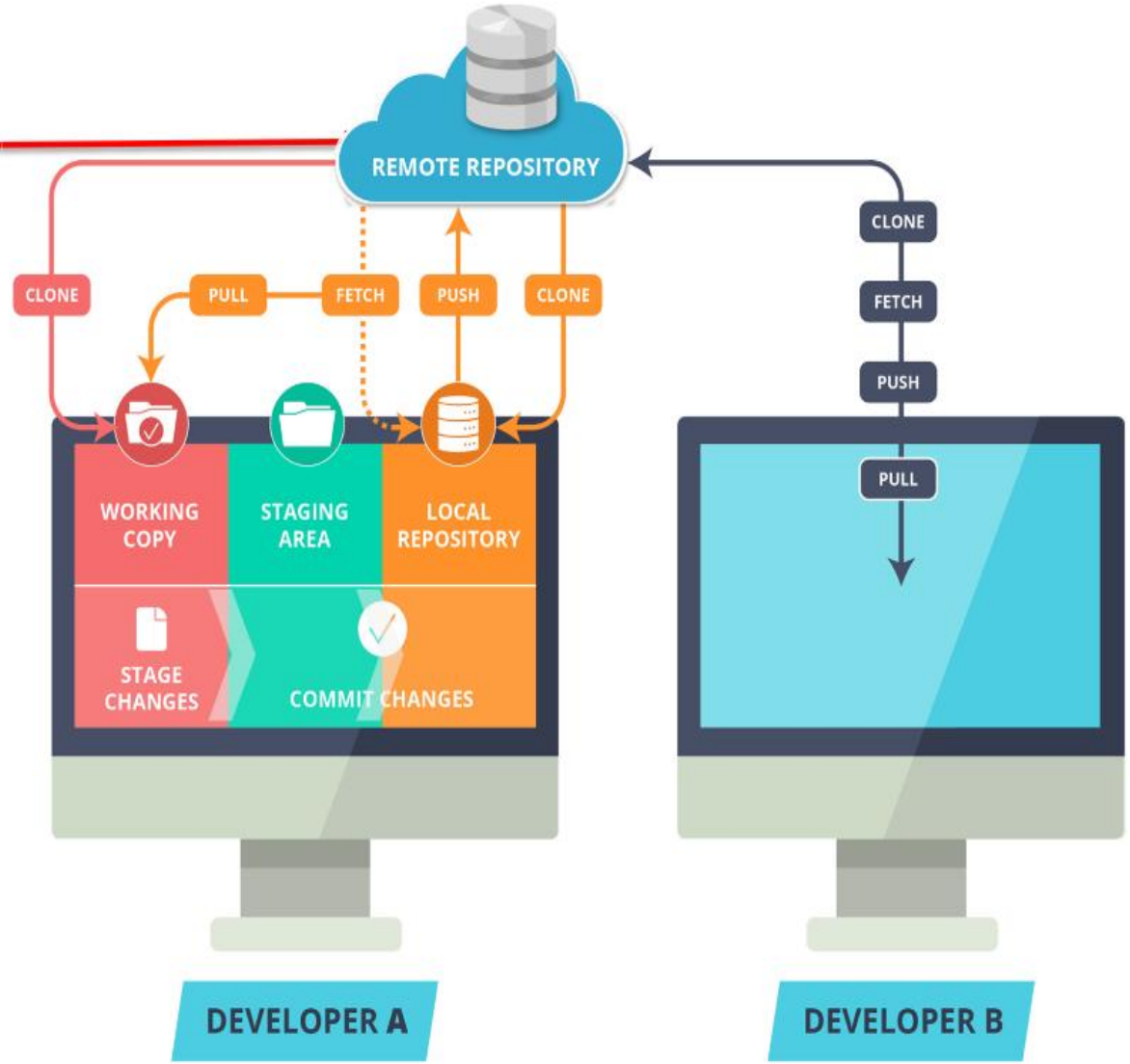
The Git File Workflow

- Use Git workflow to manage your project effectively
- Working with set of guidelines increases Git's consistency and productivity



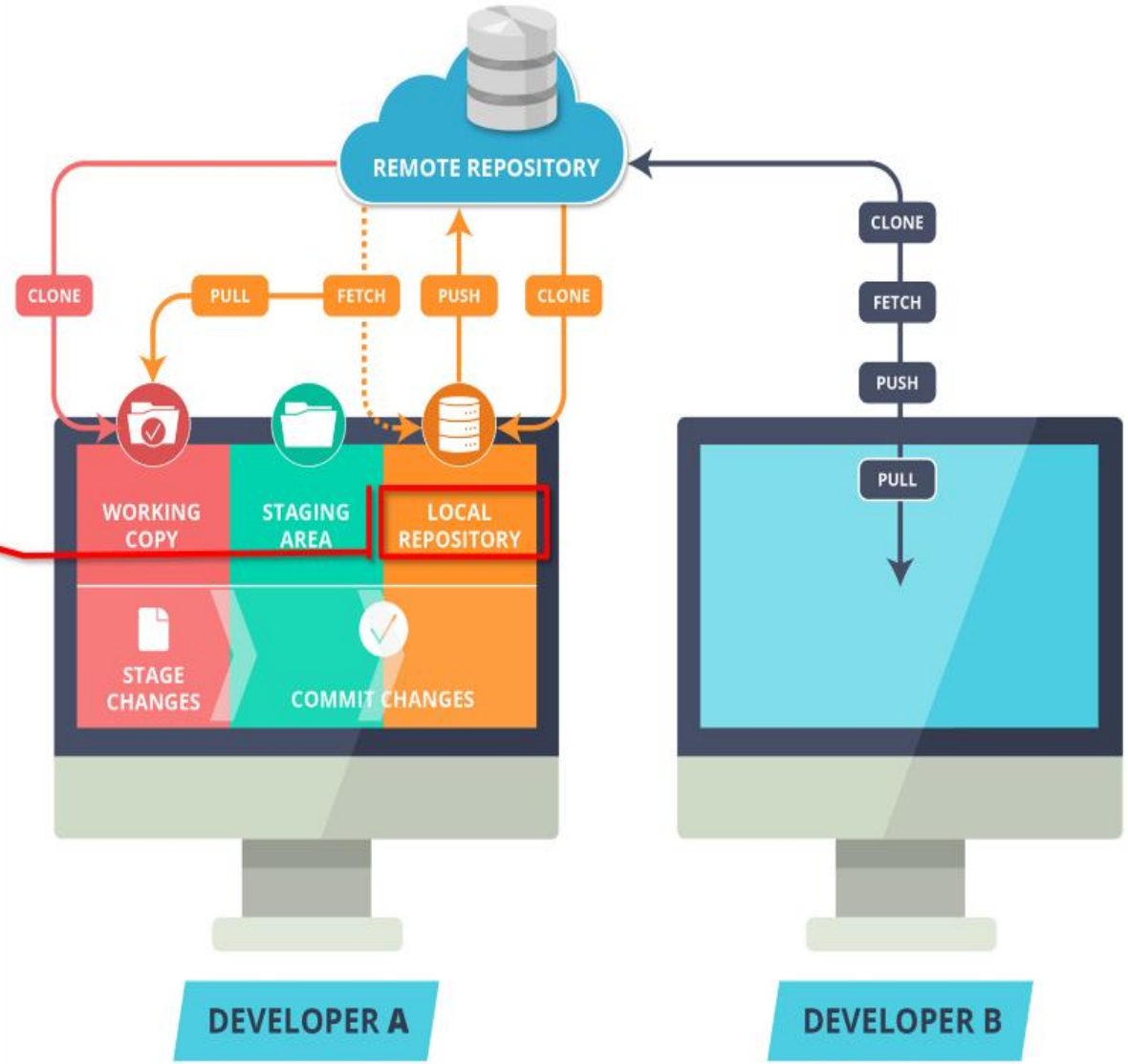
The Git File Workflow

The Remote Repository is the server where all the collaborators upload changes made to the files



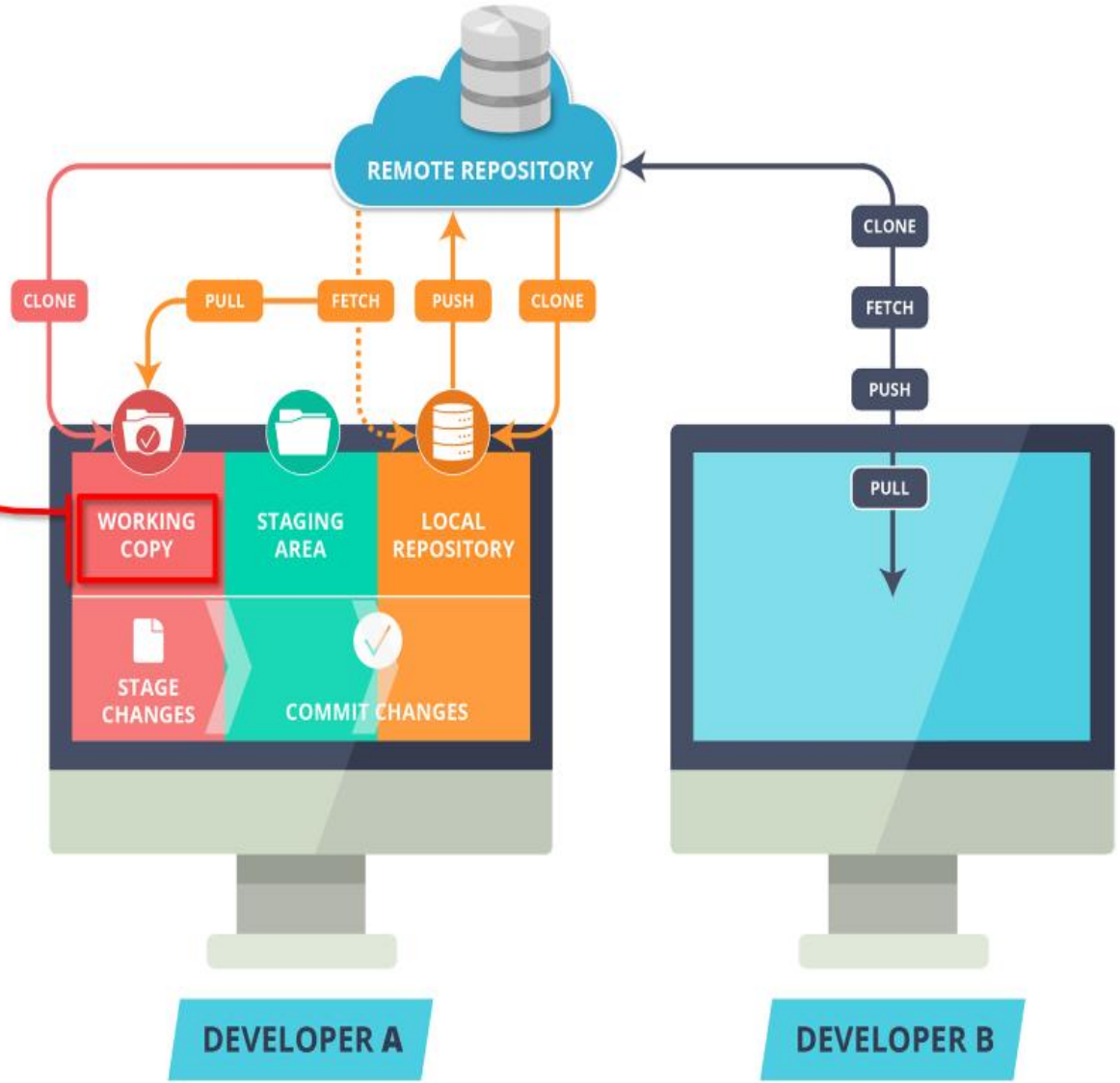
The Git File Workflow

- "**Local Repository**" is user's copy of the Version Database
- The user accesses all the files through local repository and then push the change made to the "**Remote Repository**"



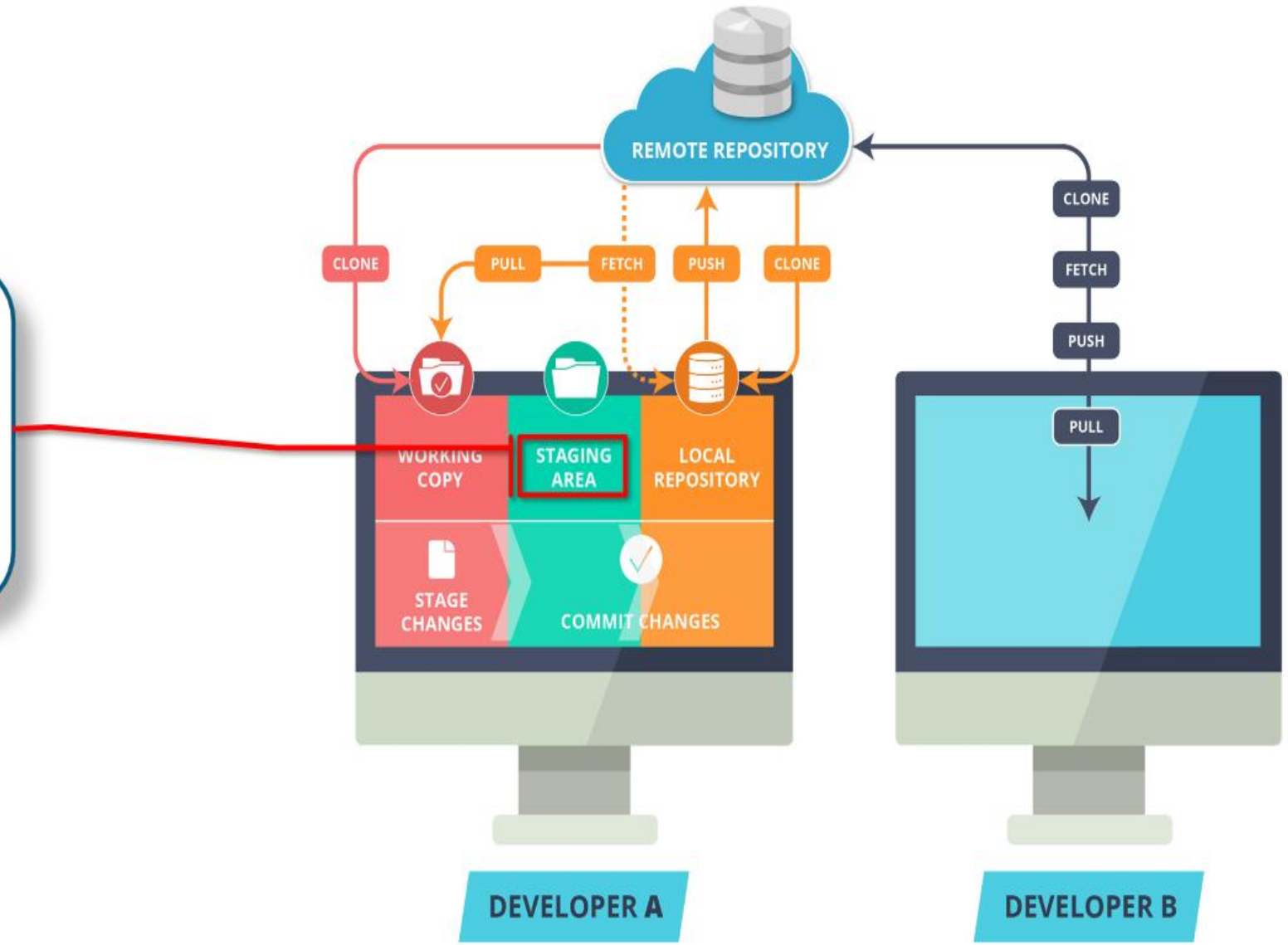
The Git File Workflow

- **"Workspace"** is user's active directory
- The user modifies existing files and creates new files in this space. Git tracks these changes compared to your Local Repository



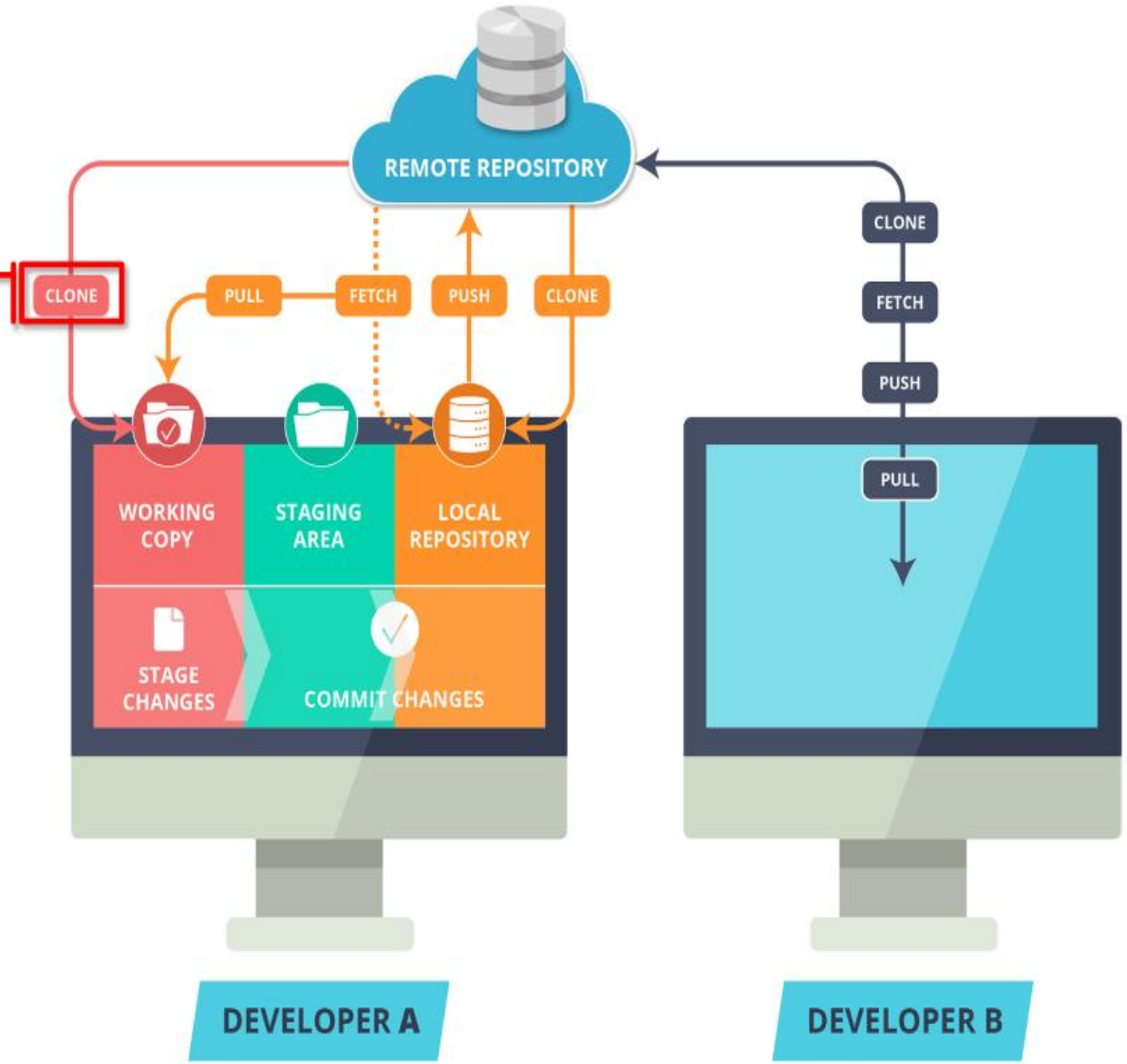
The Git File Workflow

Stage is a place where all the modified files marked to be committed are placed.



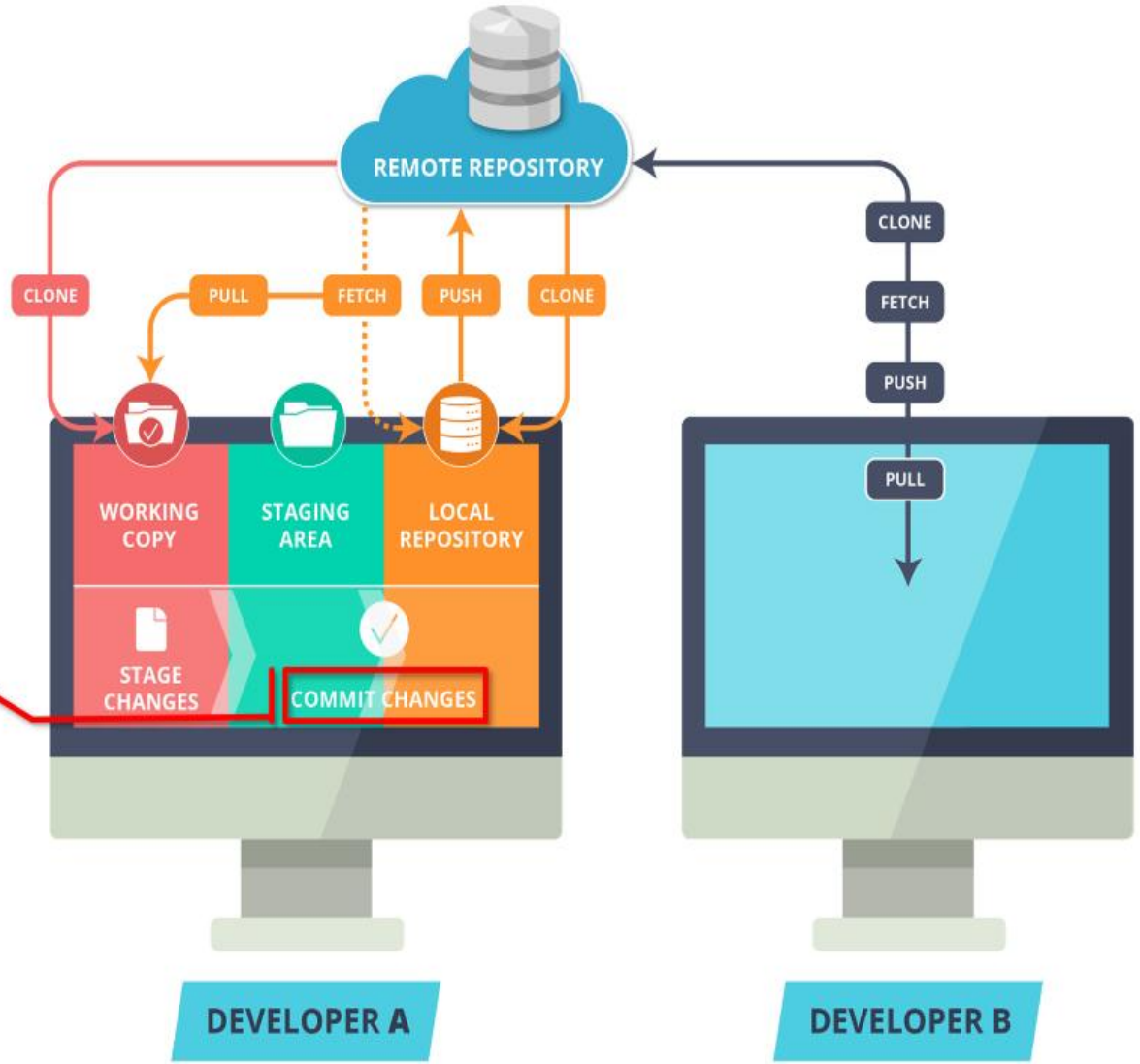
The Git File Workflow

Clone command creates a copy of an existing Remote Repository inside the Local Repository.



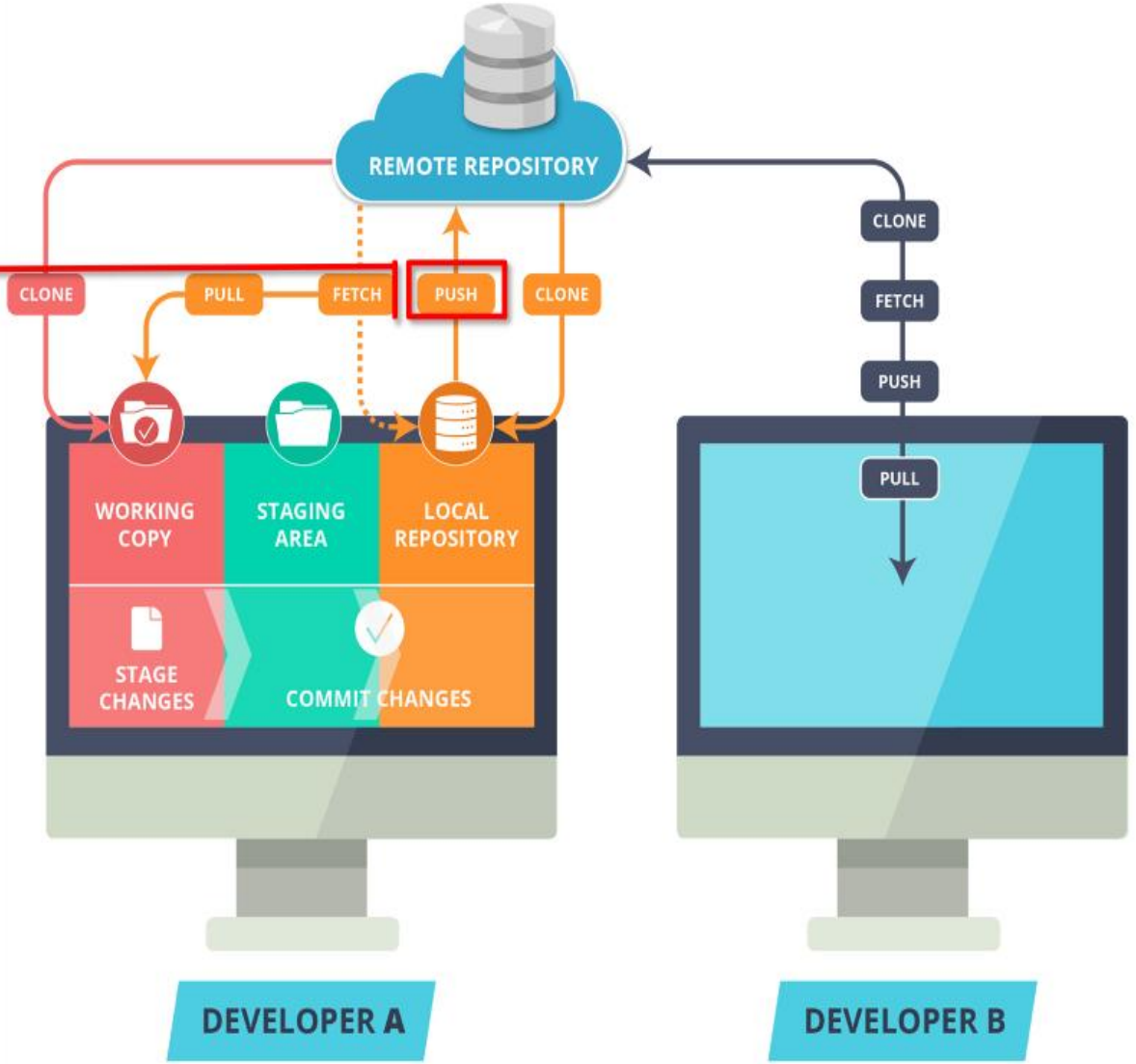
The Git File Workflow

Commit command commits all the files in the staging area to the local repository.



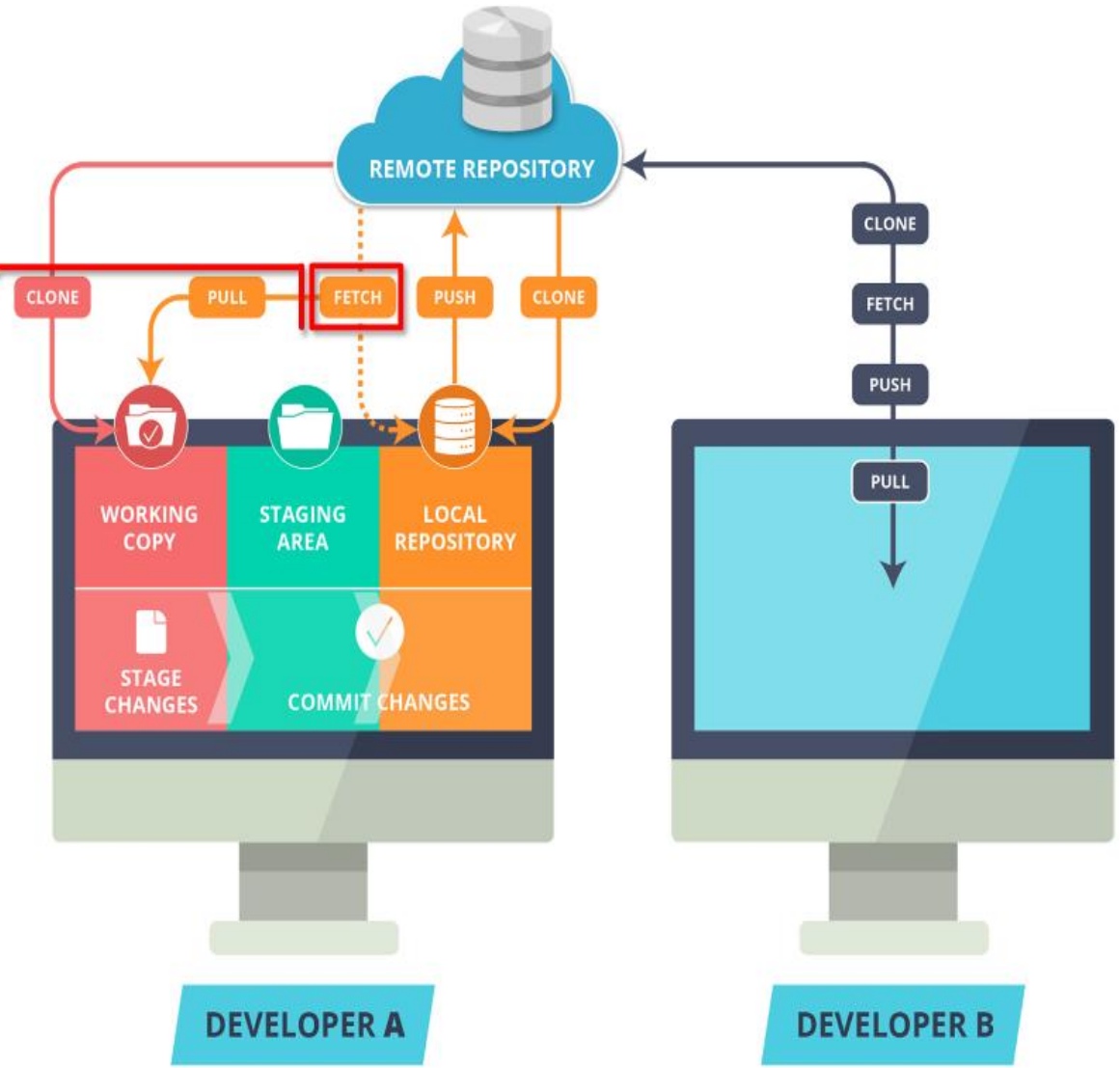
The Git File Workflow

Push command pushes all the changes made in the Local Repository to the Remote Repository



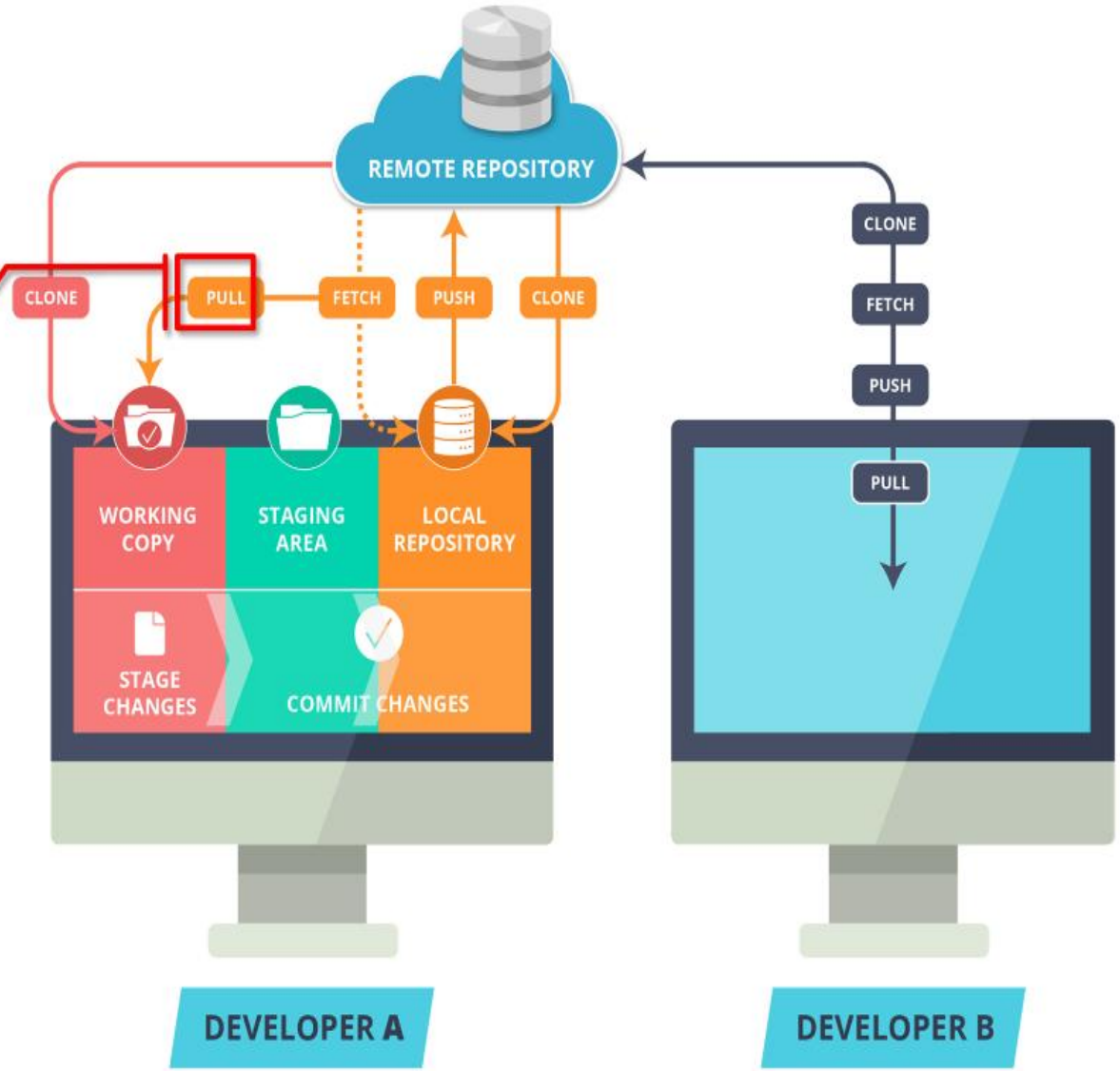
The Git File Workflow

Fetch command collects the changes made in the Remote repository and copies them to the Local Repository. This command doesn't affect our Workspace.

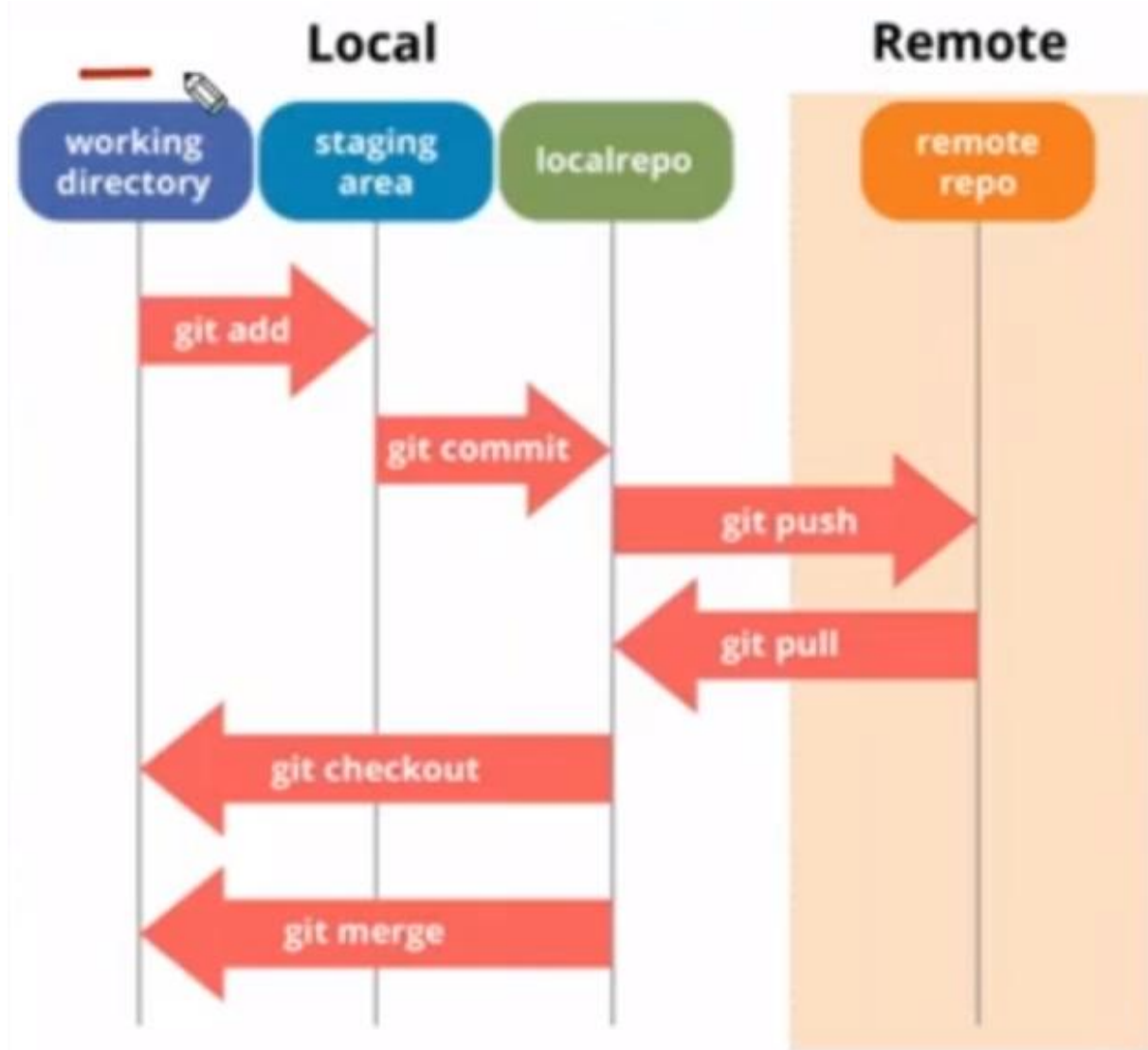


The Git File Workflow

- Pull like Fetch, gets all the changes from the remote repository and copies them to the Local Repository
- Pull merges those changes to the current working directory



Git Common Commands


Local
Remote
working directory
staging area
localrepo
remote repo
git add
git commit
git push
git pull
git checkout
git merge