

# **#VERSION CONTROL**

*by*



**STUDIOPRESENT**

# ***{ WHAT IS GIT? }***

***Most widely used modern version control system in the world today is Git.***

***Git development began in April 2005. It is developed by Linus Torvalds who wanted a better alternative for BitKeeper (source control system that he was using).***

- ***Version control, also known as source control, is the practice of tracking and managing changes to software code.***
- ***Version control systems are software tools that help software teams manage changes to source code over time.***

***Version control systems help software teams work faster and smarter. They are very useful for all types of teams since they help them to reduce development time and increase successful **deployments**.***



***# Deployment in software and web development means pushing changes or updates from one deployment environment to another.***

***Version control software keeps track of every modification to the code in a special kind of database. If a mistake is made, developers can turn back the clock and compare earlier versions of the code.***

# ***{ BENEFITS OF VERSION CONTROL }***

- ***A complete long-term change history of every file***
- ***Branching and merging. Enabling developers to verify that the changes on each branch do not conflict.***
- ***Being able to trace each change made to the software***
- ***Bug tracing***

# **{ BENEFITS OF USING GIT }**

- ***Performance***
- ***Security***
- ***Flexibility***

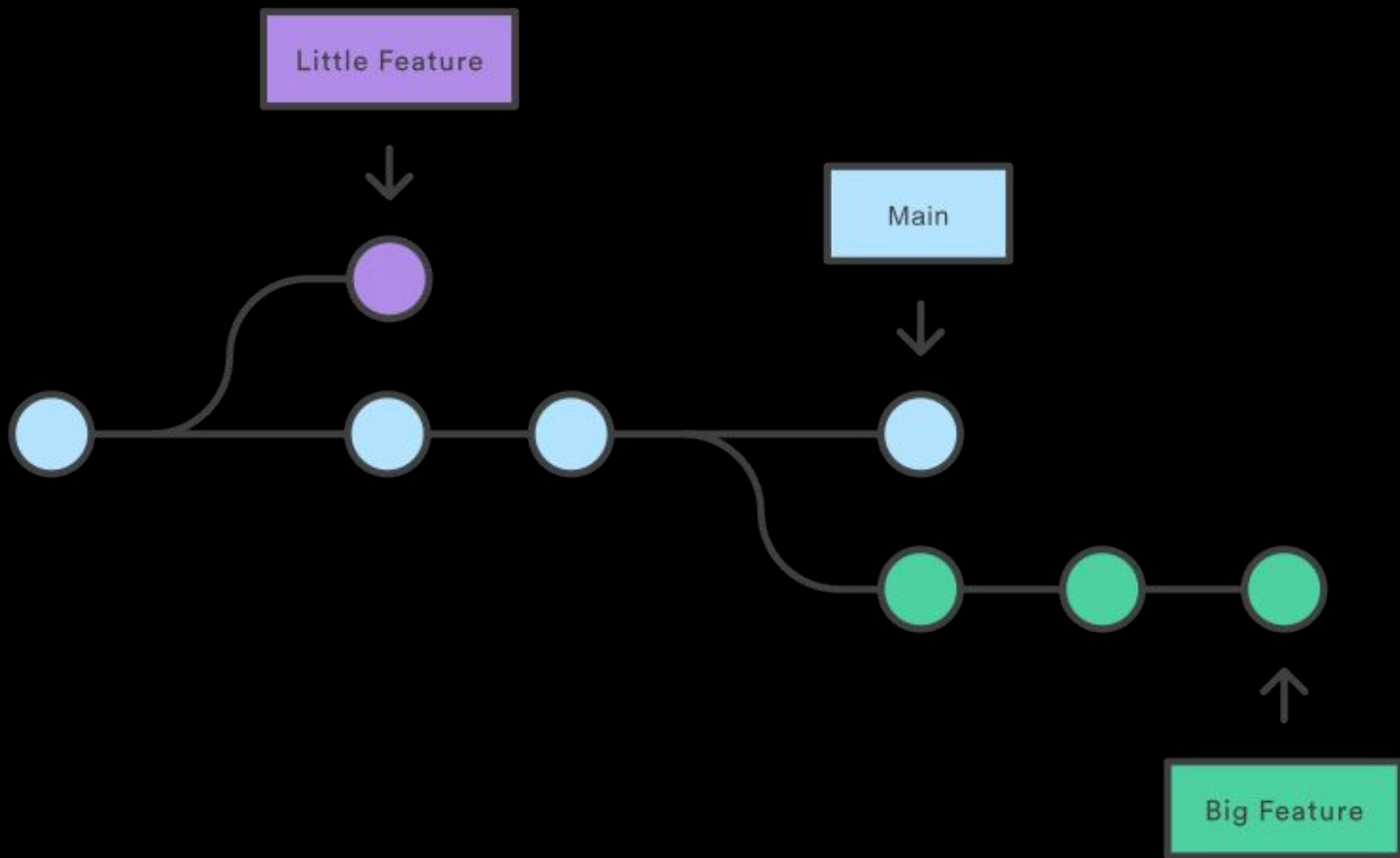


## ***{ WHAT IS GIT REPOSITORY? }***

***A Git repository is the .git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time. Meaning, if you delete the .git/ folder, then you delete your project's history.***

## ***{ WHAT IS GIT BRANCH? }***

***Git branches are effectively a pointer to a snapshot of your changes. When you want to add a new feature or fix a bug—no matter how big or how small—you spawn a new branch to encapsulate your changes.***



# **{ GIT STATES }**

- ***The Modified State – This is when you add,remove, or change something in a file. Git has noticed the change but you have not informed Git of the change yet.***
- ***The Staged State – This is when you inform Git of the change and tell Git to take note of that change. Commonly known as the "Staging Area", files in this stage are called the Staged Files. You can modify a file even after staging it, which lets you see the file both in a modified state and a staged state. The modifications made after the staging will not appear after a commit has been made.***
- ***The Committed State – This is when Git has saved your modified changes.***

# **{ GIT COMMANDS }**

**# git clone <repository url>**

**# git fetch**

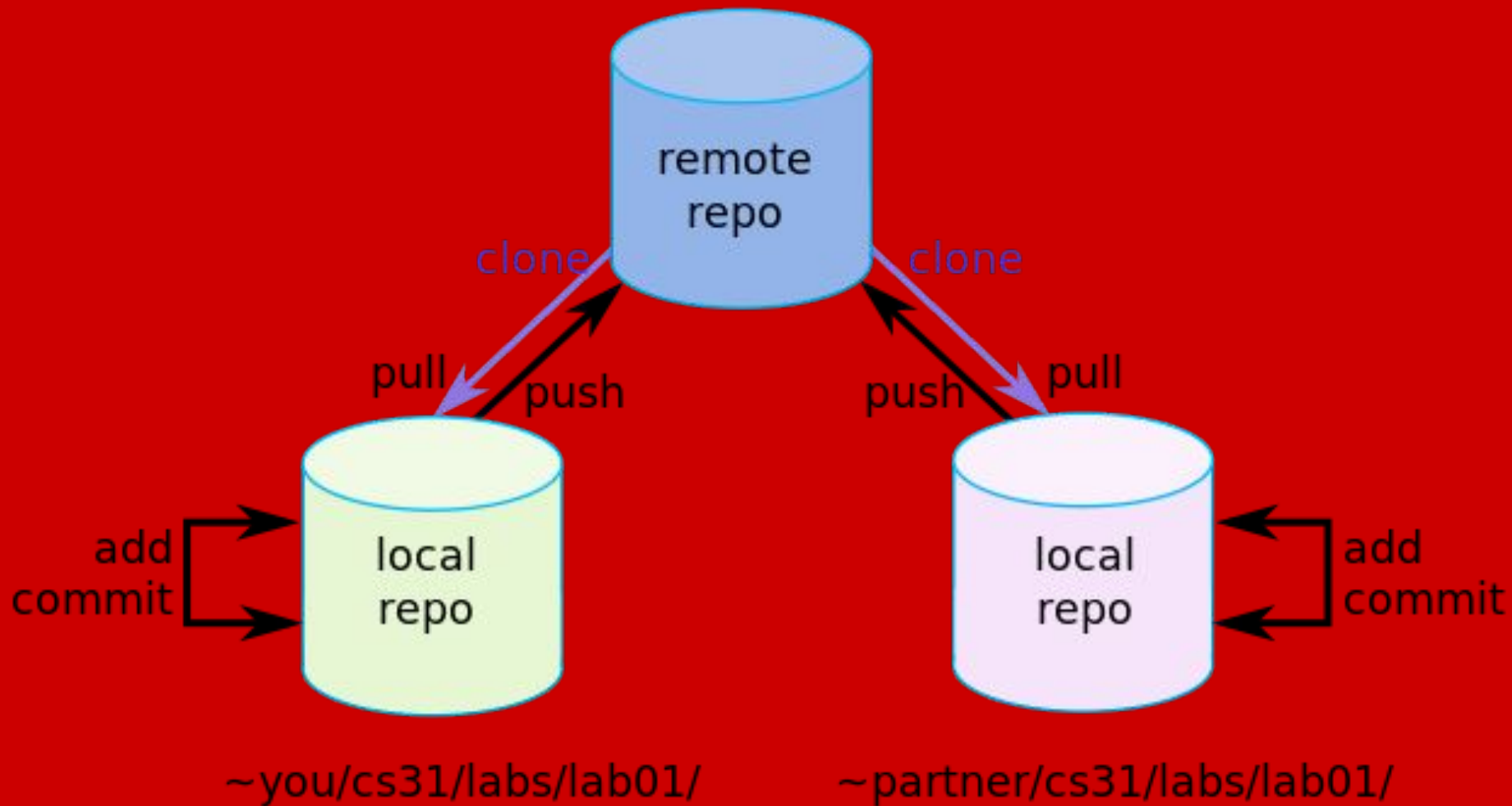
**# git checkout <branch name>**

**# git add**

**# git commit -m "<commit message>"**

**# git push**

**# git pull**



# **{ GIT VS GITHUB }**

**# Git is a free, open-source software distributed version control system (DVCS) designed to manage all source code history. It can keep a history of commits, can reverse changes, and lets developers share code.**

**# GitHub, on the other hand, is a web-based hosting service for Git repositories. It is delivered through a software as a service (SaaS) business model, was started in 2008.**

## **{ CLI VS GUI }**

- ***Git CLI: CLI stands for Command Line Interface. You open a terminal, type commands and tell Git what to do. This is the default interface and the one you get when you install Git.***
- ***Git GUI: GUI stands for Graphical User Interface. Git GUIs are programs that offer an interactive Git experience. You often get visualization of the state of your repository, commit history, and branches. Instead of writing your commands in a terminal, you can click on buttons and menu items.***



# **{ DEMO }**

- ***Create new GitHub profile***
- ***Create new repository***
- ***Install GitCMD***
- ***Create new folder with a file  
and push to new repository***