Version Control

Agenda

- What is version control
- What is Git
- How to use version control using Git
- What is Github
- How to use Github

What is Version Control ?

- A version control system is a tool that manages changes made to the files and directories in a project
- Version control record the changes made to your code overtime in a special database called repository or repo

Why Use a Version Control System ?

Generate backups

A side-effect of using a distributed version control is that it can act as a backup; every team member has a full-blown version of the project on his disk -

Team Collaboration

With a VCS, everybody on the team is able to work absolutely freely - on any file at any time. The VCS will later allow you to merge all the changes into a common version. There's no question where the latest version of a file or the whole project is. It's in a common, central place

Keep history and track changes

Having a record of the changes in a specific code file could help you and new contributors understand how a specific part of the code came to be. How did it start and develop over time to reach its current version

Test and experiment

When a team works on a software project, they often have different main project copy to develop new features on, test them, and make sure they function properly before this new feature can be added to the main project. This could save time as different aspects of the code can be developed simultaneously.
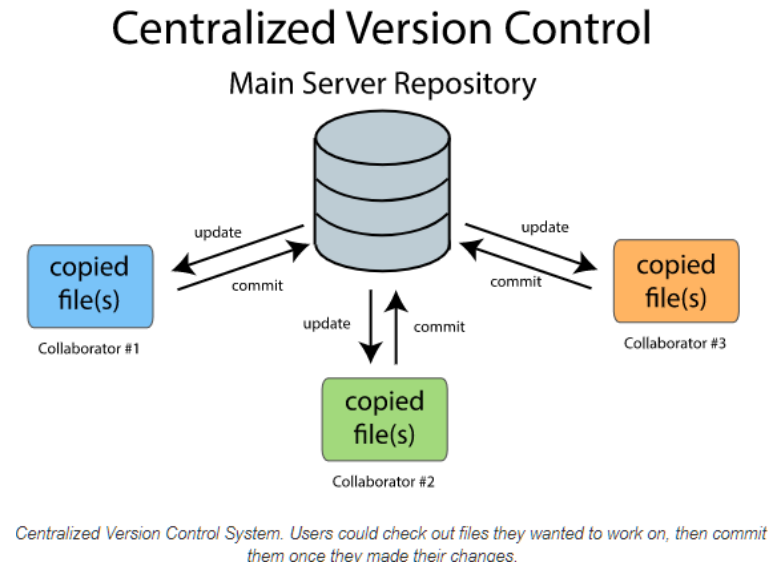
# The various types of the version control systems are:
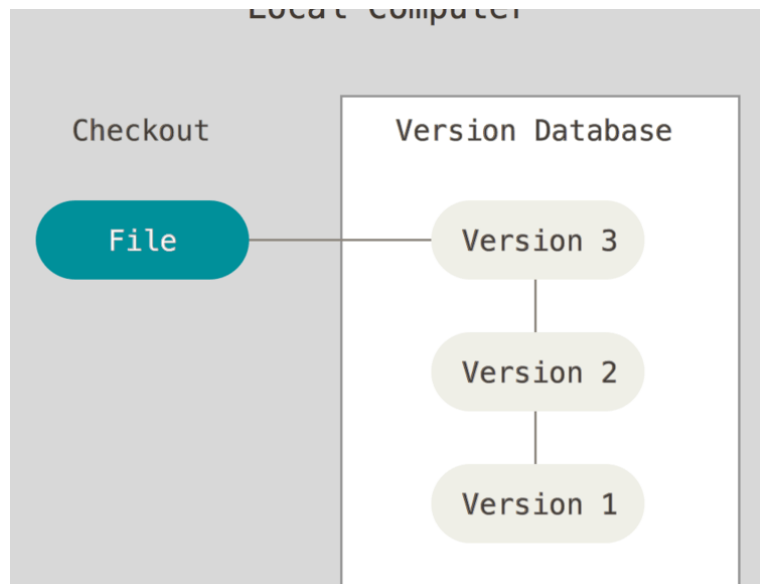
1. Local VCS

A local version control system is a local database located on your local computer, in which every file change is stored as a patch. Every patch set contains only the changes made to the file since its last version.

2. Centralized VCS

System with a single server containing all versions of the code files

Example : Subversion ,Team foundation server
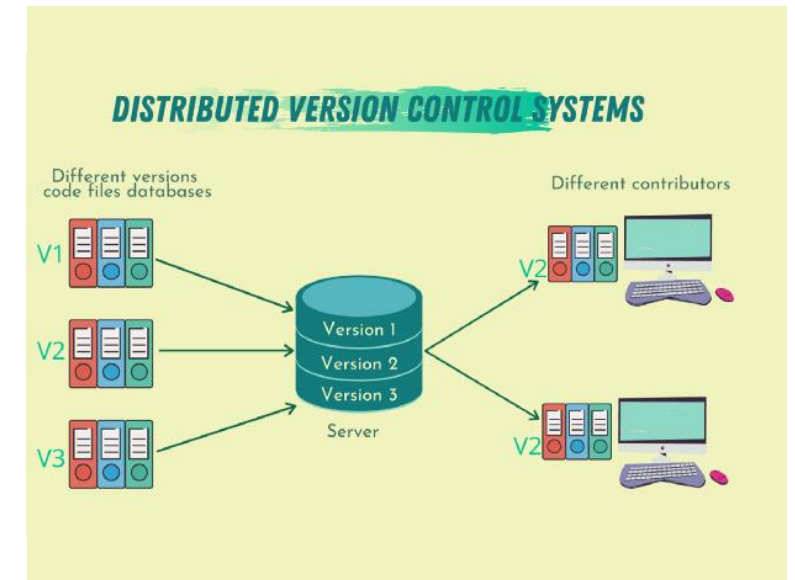
3. Distributed VCS

one or more main server containing the code files.

Example: Git, Mercurial, or Bazaar

Git

- The most popular version-control system.
- Git is free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git strength

- Nothing that is saved to Git is ever lost, so you can always go back to see which results were generated by which versions of your programs.
- Git automatically notifies you when your work conflicts with someone else's, so it's harder (but not impossible) to accidentally overwrite work.
- Git can synchronize work done by different people on different machines, so it scales as your team does.

How to use Git

1 Command line

2 Code editors and IDE (Pycharm and Vs code)

3 GUI (GitKraken , Sourcetree)

Git Workflow
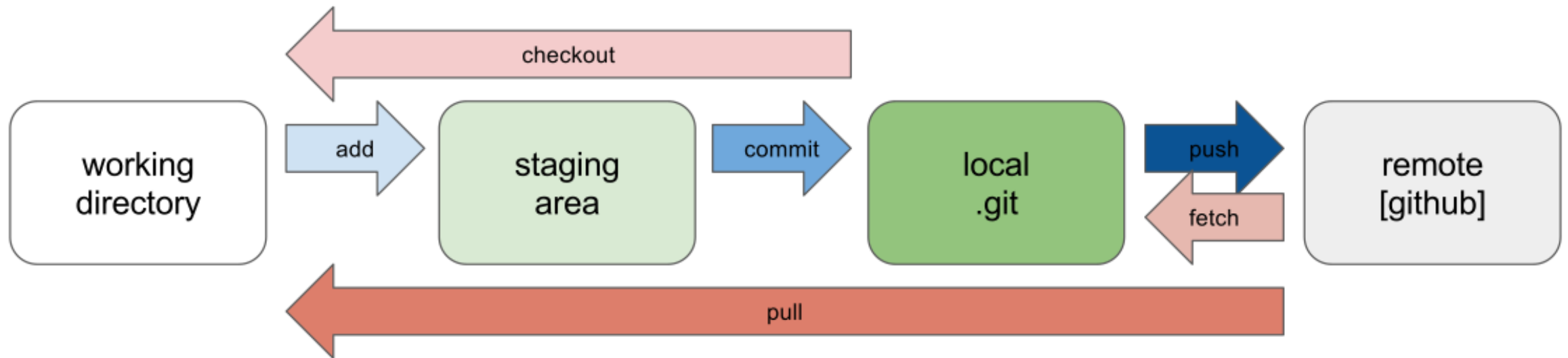
GIT LAB / PRACTICAL

https://github.com/Heytec/ADS_REV_code/blob/main/03%20Lesson%20Version%20Control/03%20lesson%20Version%20Control%20(%20Git%20%26%20Github).ipynb

AFRICA DATA SCHOOL

Github

- **GitHub** is a code hosting platform for version control and collaboration.
- GitHub is a company founded in 2008 https://github.com/
- Microsoft revealed earlier this 2018 that it's acquiring GitHub for $7.5 billion.
- Github makes tools which integrate with git.
- There are many other alternatives to GitHub, such as GitLab and BitBucket

What is the difference between Git and GitHub?

Git is a version control system that lets you manage and keep track of your source code history. GitHub is a cloud-based hosting service that lets you manage Git repositories. If you have open-source projects that use Git, then GitHub is designed to help you better manage them.

GIT LAB /PRACTICAL

https://github.com/Heytec/ADS_REV_code/blob/main/03%20Lesson%20Ve
rsion%20Control/03%20lesson%20Version%20Control%20(%20Git%20%2
6%20Github).ipynb