Git and Github Workshop

9th April 2022

Outline

- Introduction to Git and version control
- Setup and configure Git and Github account
- Practical session

The challenge

- Large, open-source projects may have hundreds of developers around the world maintaining it
  - E.g., Nearly **12,000** developers have contributed to the Linux kernel
  - The kernel is nearly **20 million** lines of code
  - Each release has ~**10,000 patches**
  - That's about **7 code changes per hour**.

  (Linux Kernel Development Report, Linux foundation, 2015)

Managing source code

- How can you manage changes to millions of lines of code, across hundreds of active developers, spread around the world?
- Answer: using a source code version control or repository system

Why not just keep copies in backup folders?

- Error prone
- Hard to roll back
- Cannot easily share code with others

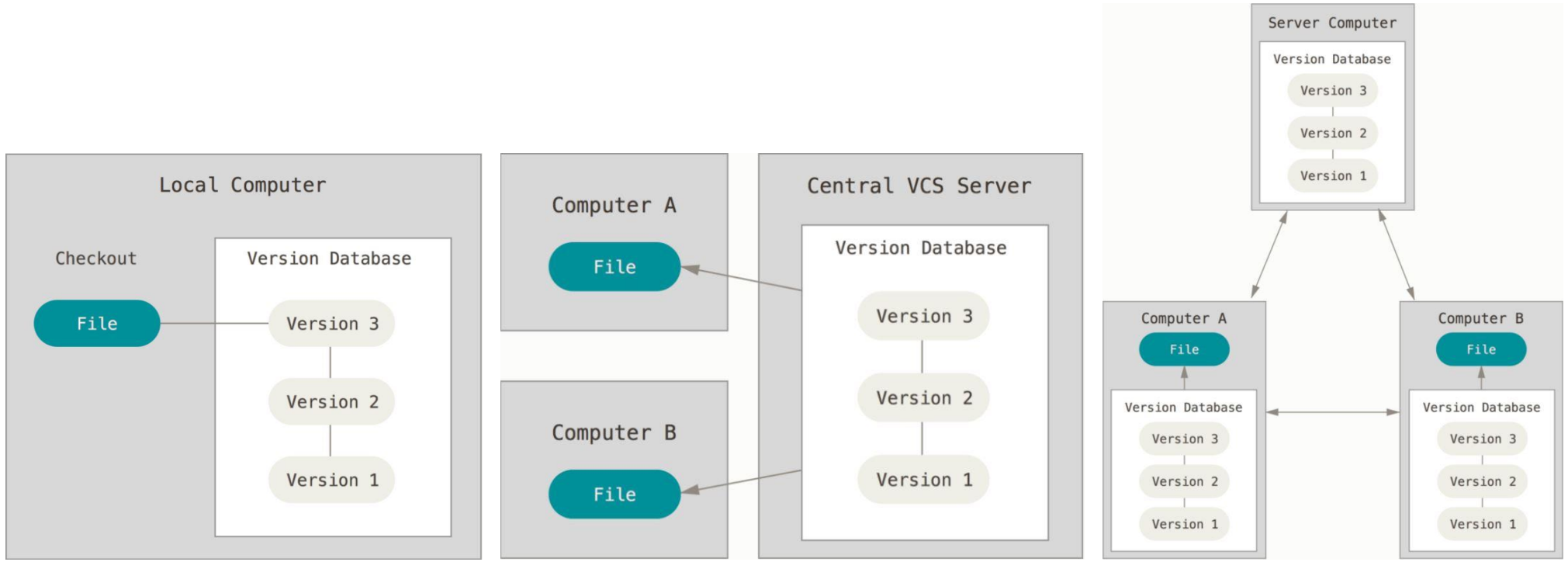
Or just use DropBox etc?

- While arguably better than just storing lots of folders, they are not designed specifically for source code management and automated testing etc.

Version Control

- **Version control**, also known as source control, is the practice of tracking and managing changes to software code.
- Version Control System (VCS) allows you to:
  - **revert files** back to a previous state
  - **revert the entire project** back to a previous state
  - **compare changes** over time, see who last modified something that might be causing a problem, who introduced an issue and when, and more.

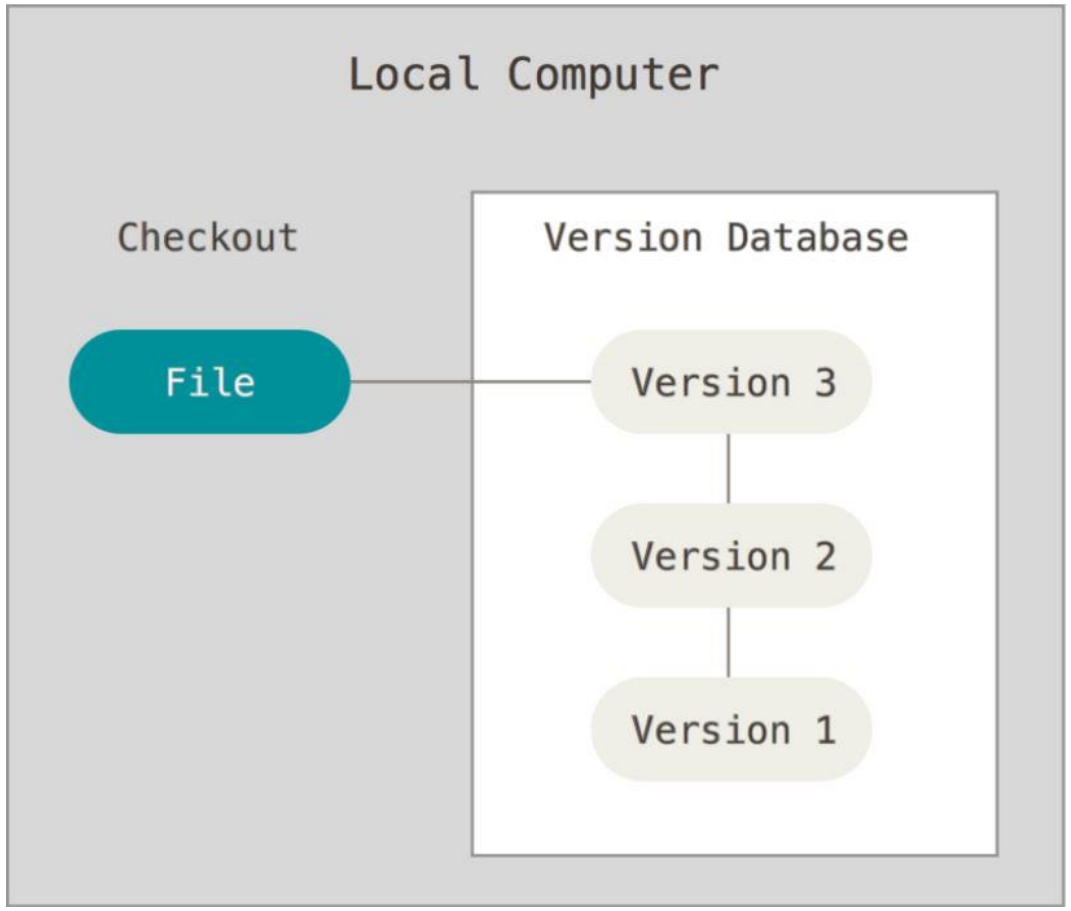
Types of VCS



Local Centralized Distributed

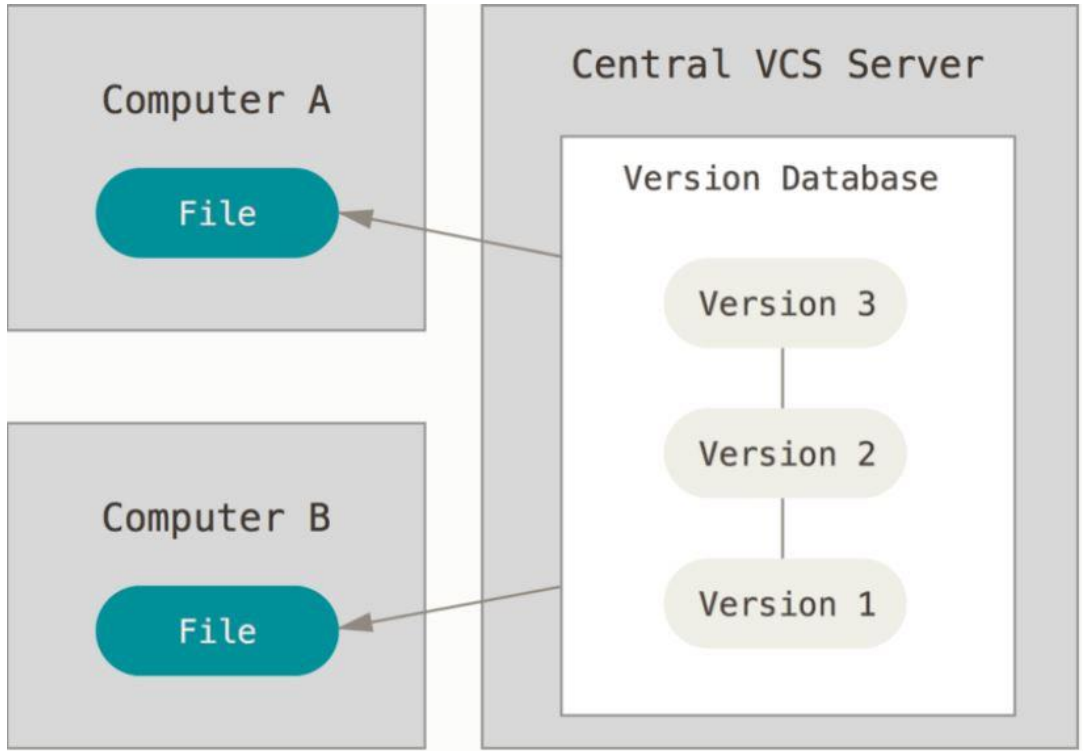
Local Version Control Systems

- A local version control system is a local database located on your local computer.
- It keeps a snapshot for each version in a linear history of your files, so when you change file, it creates a new version of it.



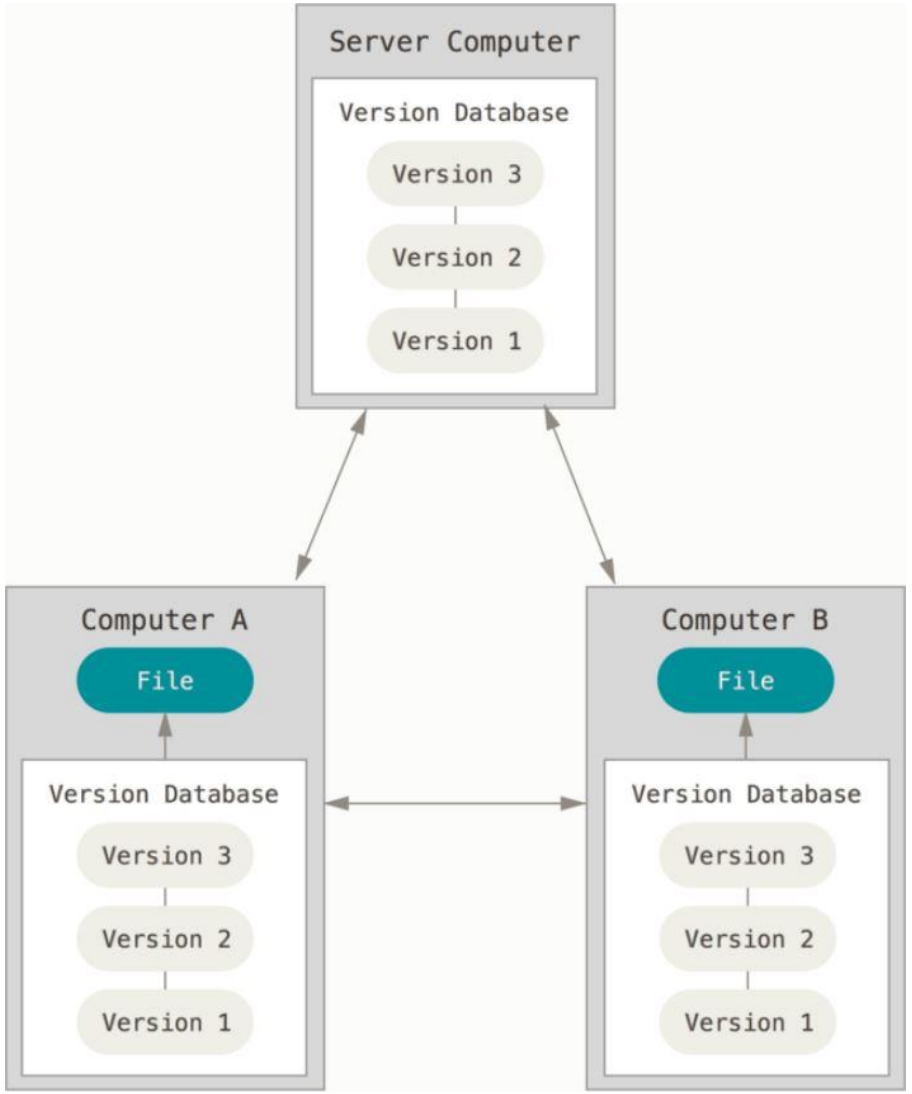
Centralized Version Control Systems

- A centralized version control system has a single server that contains all the file versions.
- To make edits, you need to get a local copy of latest version and then update server with your edits to create a new version.
- Example: Microsoft Team Foundation Server (TFS) and SVN.



Distributed Version Control Systems

- Each one holds local copy of files with its versions history (each one could be a server).
- They can make whatever changes they want and commit without affecting the remote repo. They first commit in their local repo and then push the changes to the remote repo.
- Example: Git



Git

- Git is a (free and open source!) distributed version control system.
- Designed originally for command line use.
- But there are various GUI clients available and web front ends for management, such as GitHub.

Who Should Use Git?

- **Anyone wanting to track edits**
  - Review a history log of changes made
  - View differences between versions
  - Review old versions
- **Anyone needing to share changes with collaborators**
- **Not as useful for tracking non-text files**
  - Images, Movie, Music, Fonts
  - PDF, PSD

Who Should Use Git?

- **Anyone wanting to track edits**
  - Review a history log of changes made
  - View differences between versions
  - Review old versions
- **Anyone needing to share changes with collaborators**
- **Not as useful for tracking non-text files**
  - Images, Movie, Music, Fonts
  - PDF, PSD

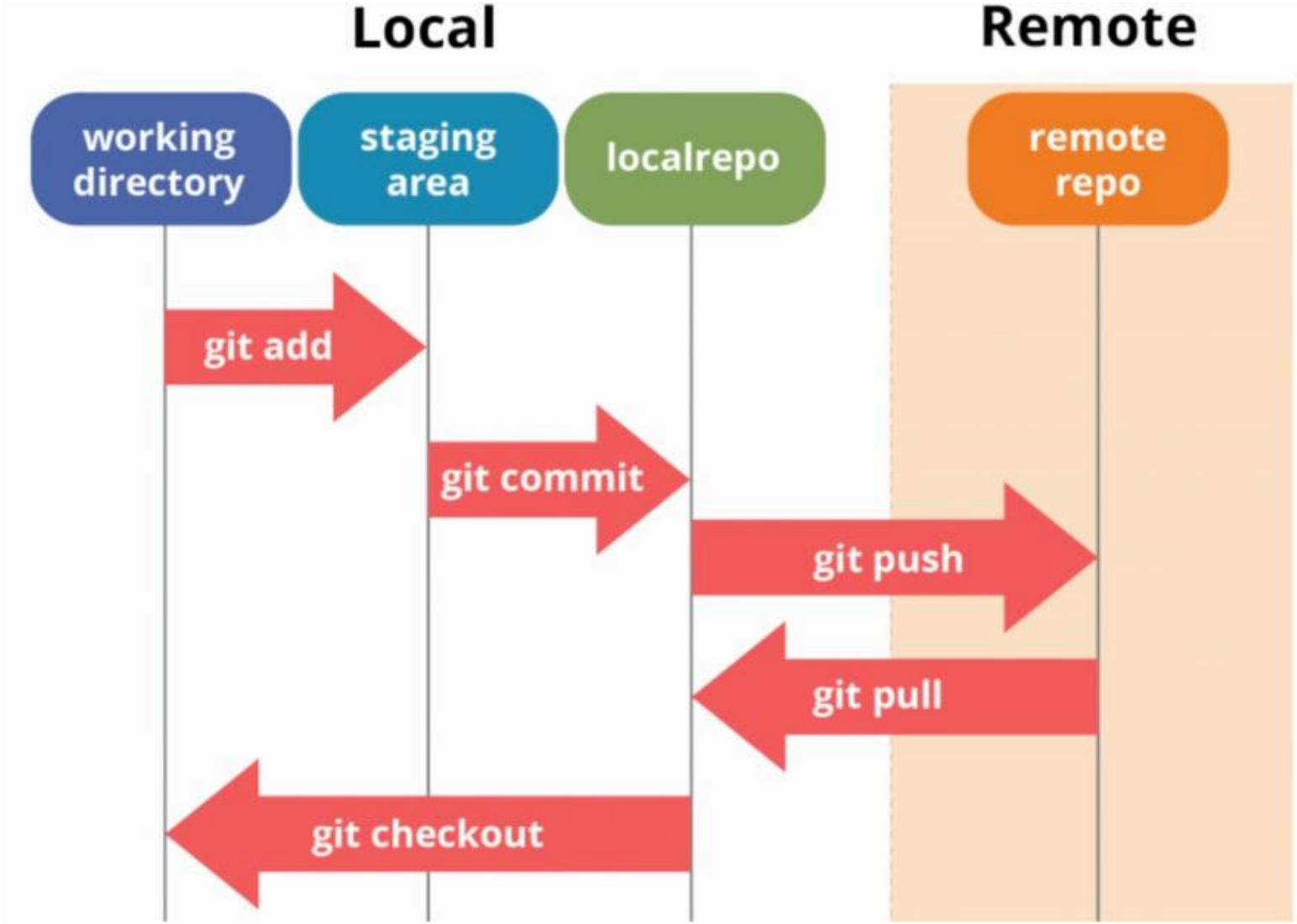
GitHub

GitHub.com is a site for online storage of Git repositories.

- You can create a remote repo there and push code to it.
- Many open-source projects use it, such as the Linux kernel.
- You can get free space for open-source projects, or you can pay for private projects.

Question: Do I always have to use GitHub to use Git?
– Answer: No! You can use Git locally for your own purposes

Git workflow



Further resources

- Git: some tips and helpful ideas
  - http://betterexplained.com/articles/aha-moments-when-learning-git/
  - https://www.tutorialspoint.com/git/git_basic_concepts.htm
  - https://www.atlassian.com/git/tutorials/learn-git-with-bitbucket-cloud

Question?