Microsoft Learn
Student Ambassadors

Version Controlling using Git and GitHub

AGENDA

- What were the issues before Git came into existence?
- How to install Git?
- Introduction to Git
- Collaborate with GitHub
- Contribute with GitHub

What is Version Control?

A version-control system is a program or set of programs that tracks changes to a collection on files.

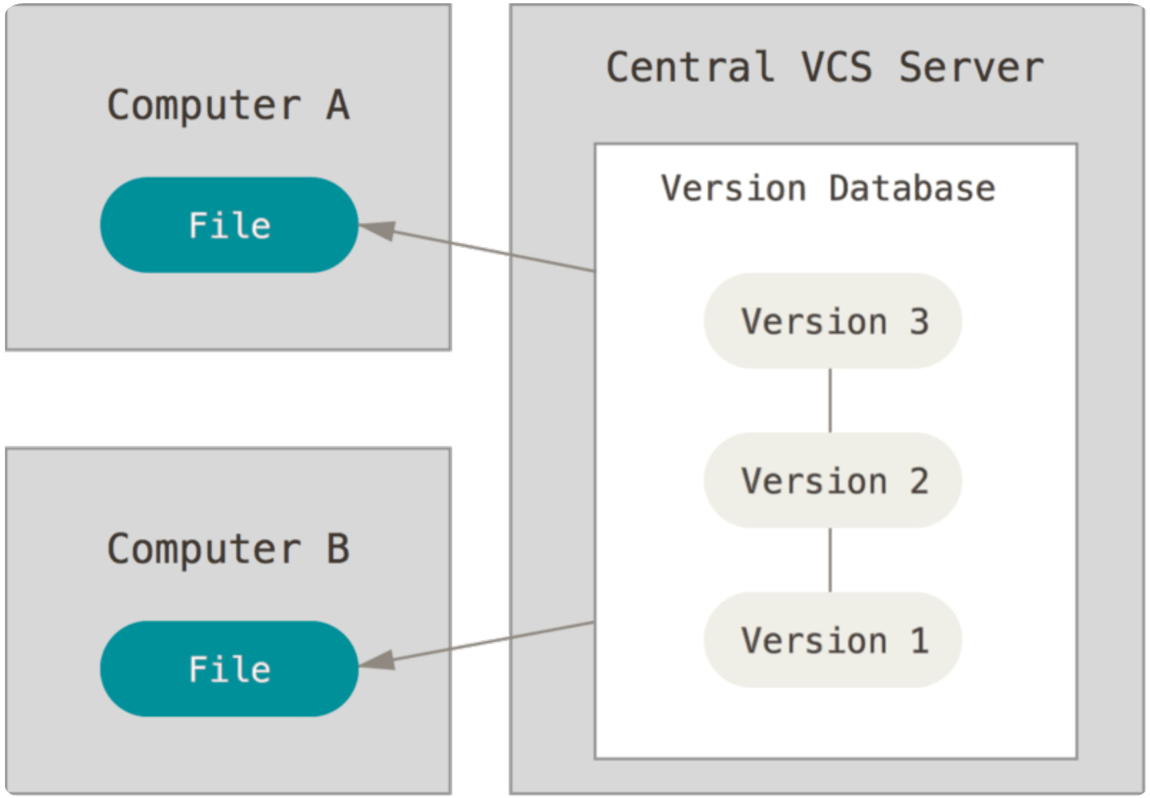
It allows you to:

- Revert selected files back to a previous state
- Compare changes over the time
- Recover older versions

Types of Version-Control Systems

Centralized

Distributed



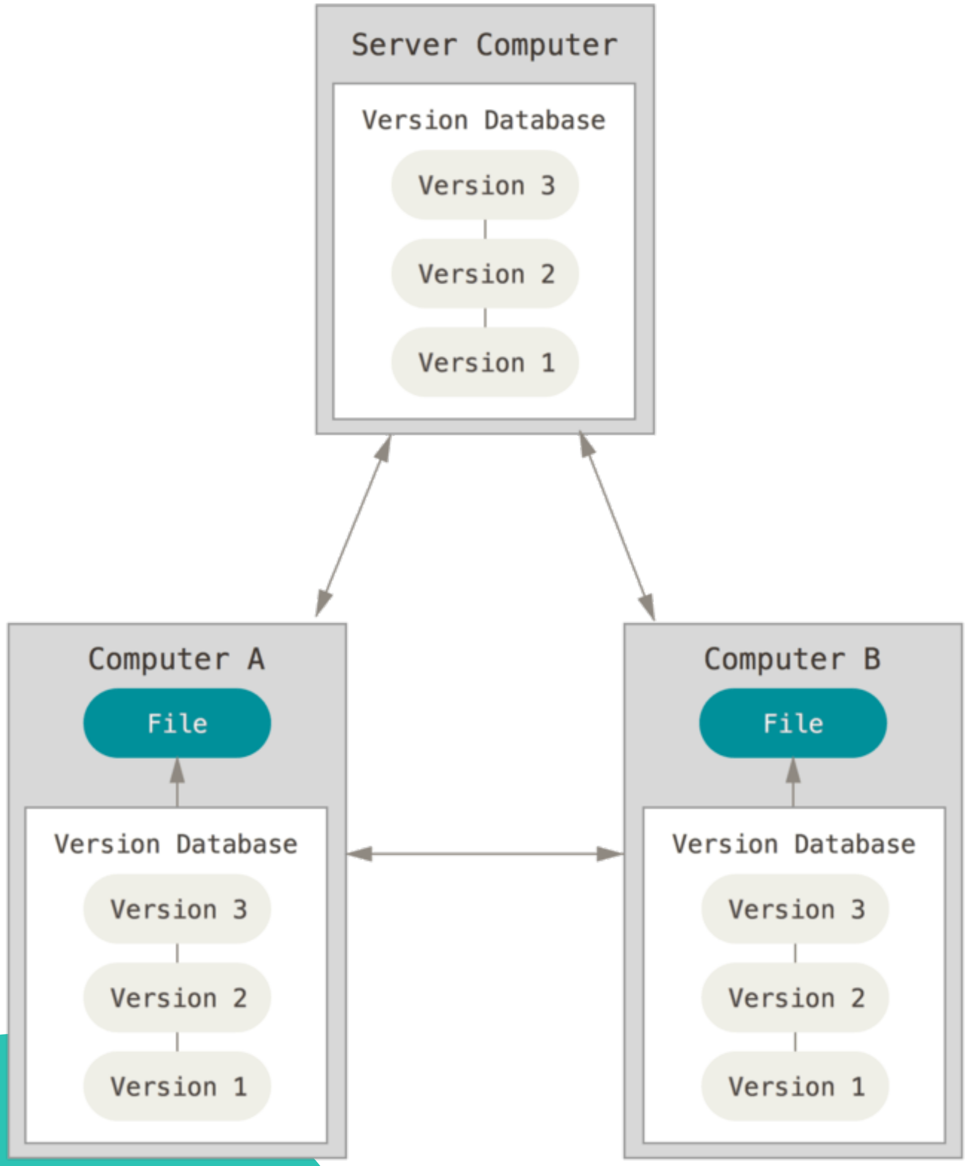
Centralized VCS

Centralized Version Systems are based on the idea that there is a single "central" copy of your project somewhere (probably on a server), and programmers will commit their changes to the central copy.

Popular Centralized VCSs

1. SVN
2. CVS
3. Perforce

- SVN is the most popular Centralized Version Control System.
- SVN can also work with GitHub.



Distributed VCS

Distributed Version Systems are based on the idea that there is a two copies of your project, one on a server and other client, and programmers will commit their changes to the local and then the central copy.

Popular Distributed VCSs

1. GIT
2. Mercurial
3. Bazaar

- Git is the most popular Distributed Version Control System and has became an industry standard.

What were the issues before Git came into existence?

- Untraceable previous updates
- Difficulty in collaboration
- Keeping live code and under development code in separate places and difficulty in merging those.

How to install Git?

https://git-scm.com/

Introduction to git

Git configuration on Local Machine:

- git config user.name “<Username>”
- git config user.email “<Email id>”

Introduction to git

Most helpful git commands:

- git help
- git status
- git log [branchname]

Introduction to git

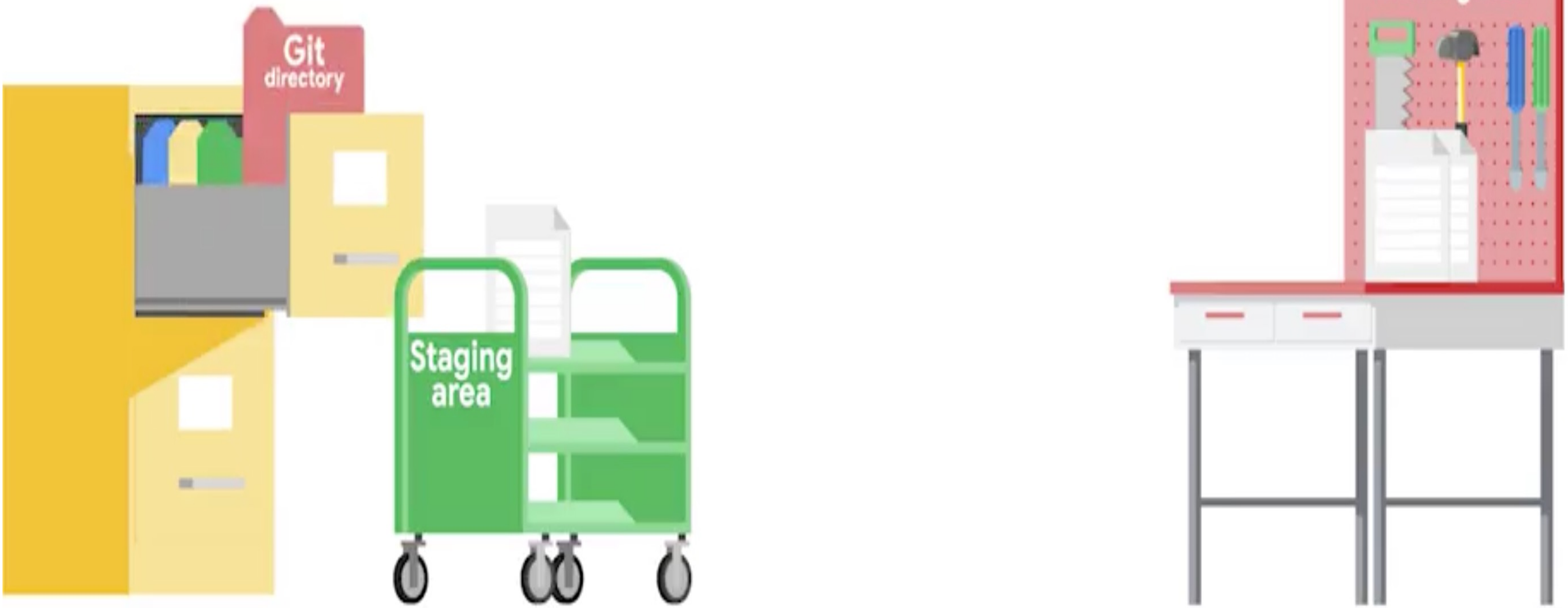
Git implementation on Local Machine:

- git init [projectname]
- git add <filename>
- git commit –m “<message>”

Introduction to git

How to Rollback to previous commit in GIT:

- git checkout <filename>
- git reset HEAD~<No. of commits back>
- Git revert <SHA Code>


Git directory
Staging area
Working tree

Collaborate with GitHub

Creating a GitHub Account

https://github.com/

Collaborate with GitHub

Connecting Git from local machine with GitHub:

- git remote add <remotename> <URL>
- git pull <remotename> <branchname>
- git push <remotename> <branchname>

Collaborate with GitHub

Branches

- git branch <branchname>
- git checkout <branchname>
- git merge <branchname>

Contribute with GitHub

Steps :

1. Visit GitHub
2. Fork
3. Make changes
4. Send a pull request

Recap

- What is VCS
- GIT & GitHub
- Collaborations with GitHub
- Contribution with GitHub
- GitHub Pages
- GitHub Profile Readme

Quiz Time

Thankyou