GIT

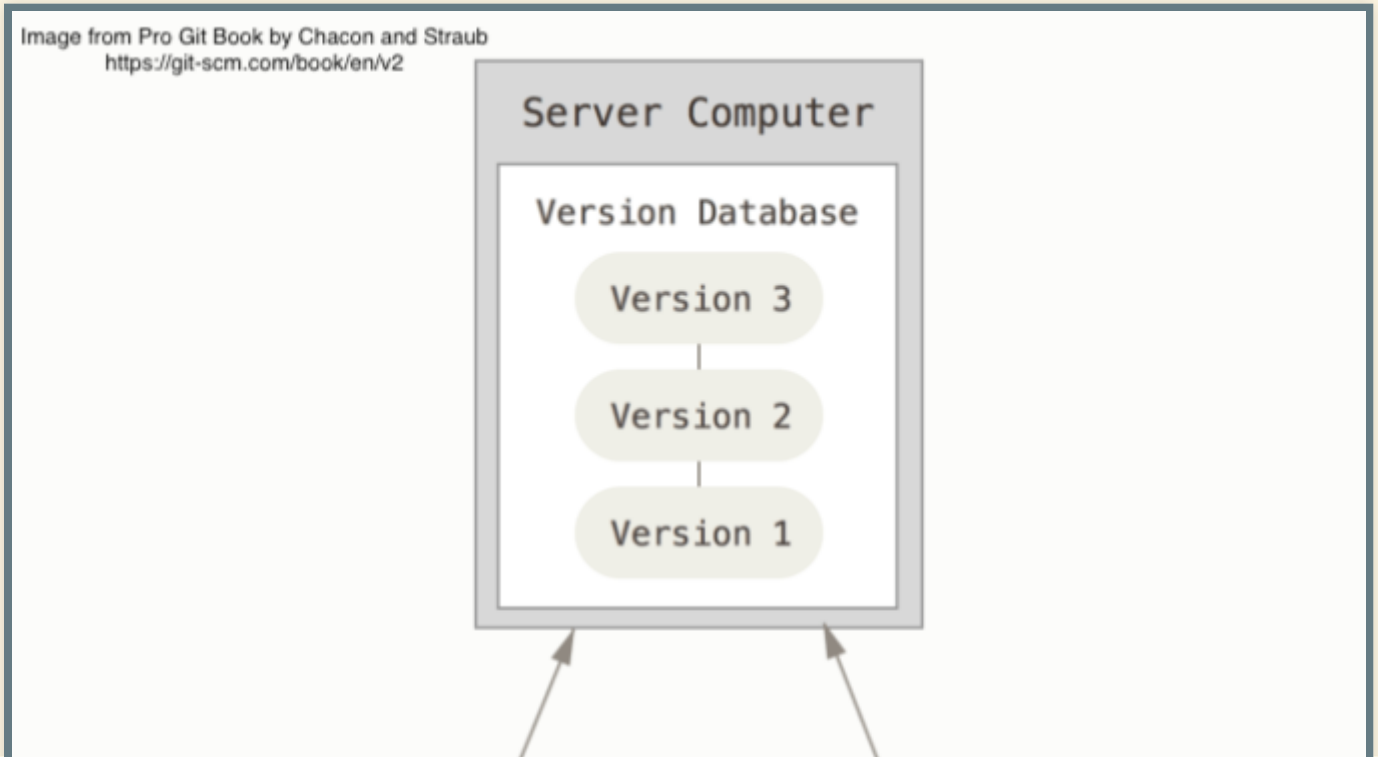
- Distributed Version Control System (DVCS)
- Free and open source

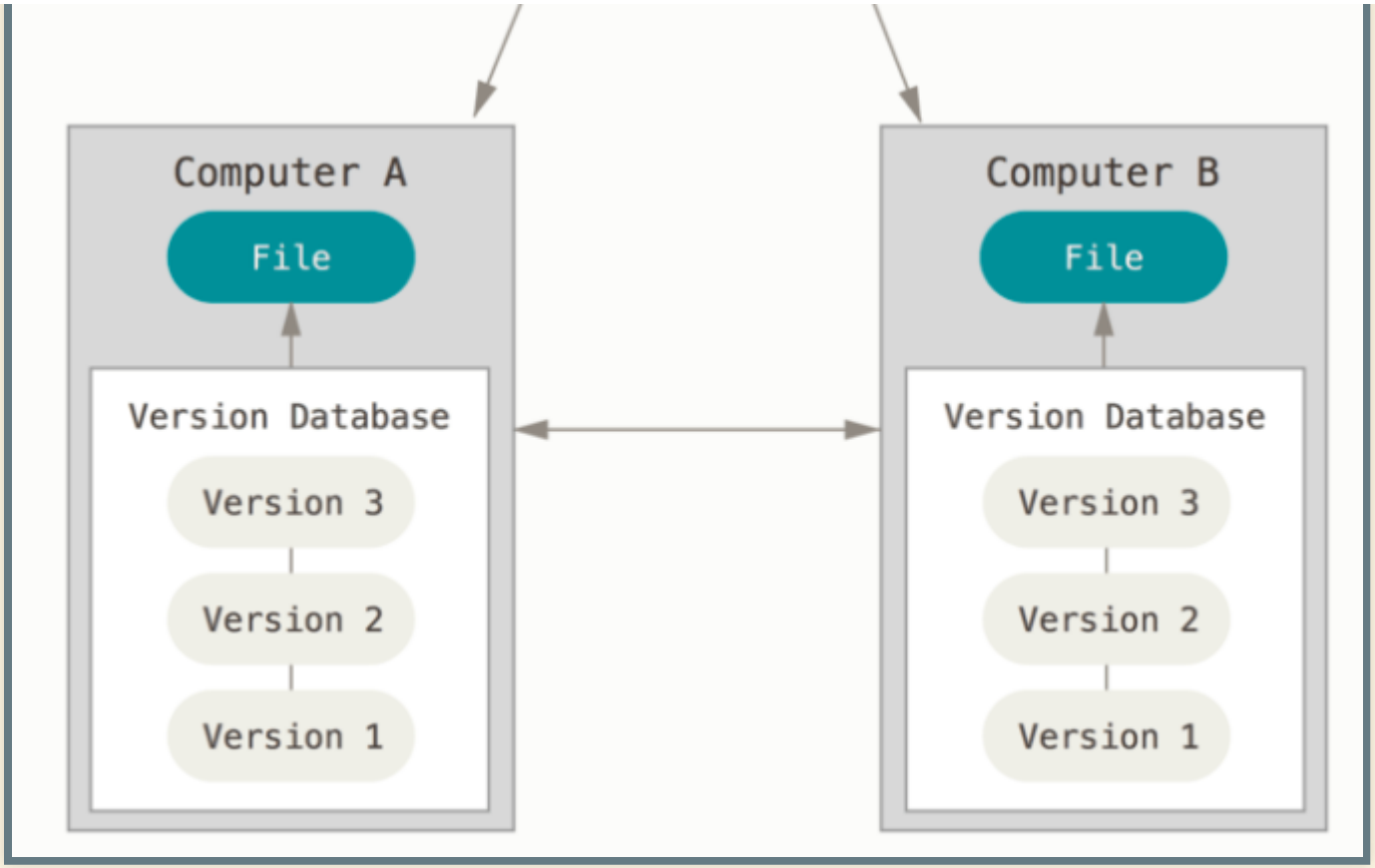
VERSION CONTROL SYSTEM

- Records changes to a file/set of files over time
- Allows viewing and recalling of history
- Who cares:
  - Rollback changes
  - Investigate bugs, bug fixes
  - Recover deleted files

DISTRIBUTED VERSION CONTROL

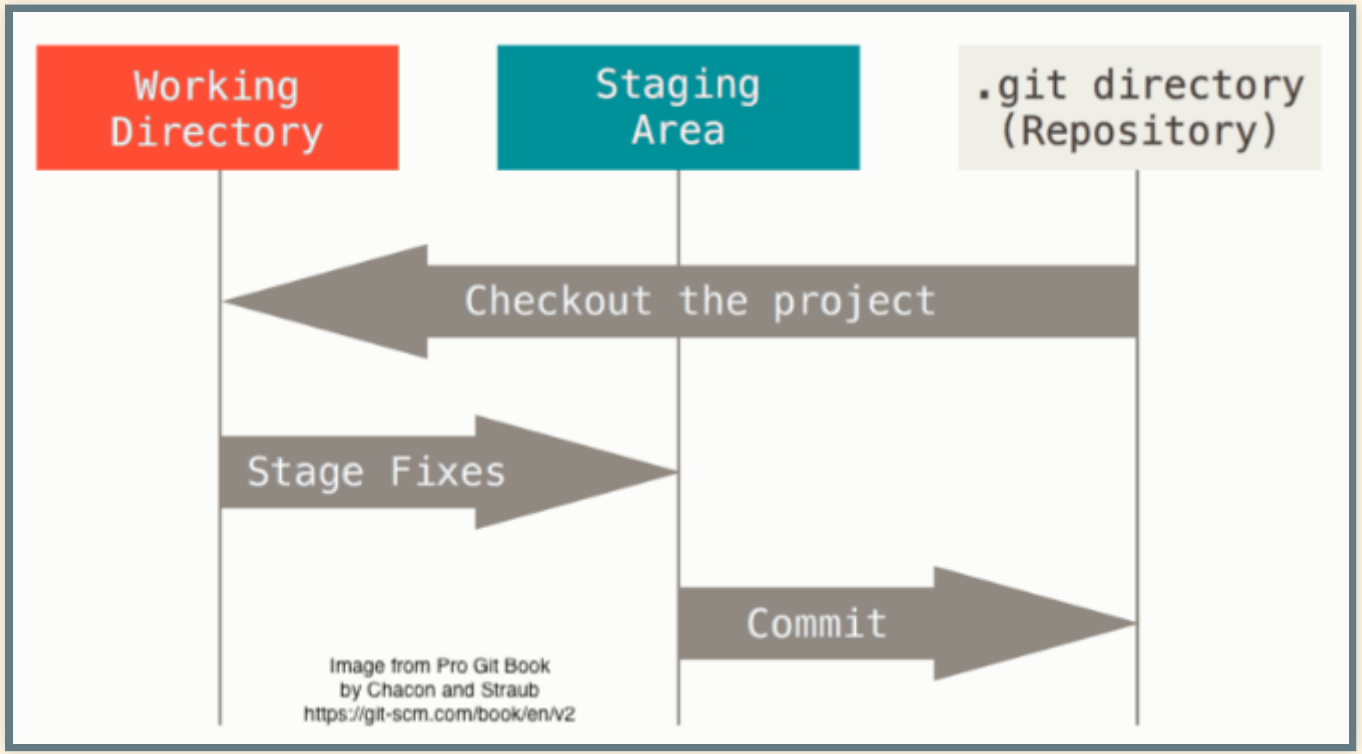
- each client fully mirrors repository, including full history
- each clone is full backup
- allows for working, recording changes offline




Computer A
File
Version Database
Version 3
Version 2
Version 1
Computer B
File
Version Database
Version 3
Version 2
Version 1

FILE STATES - LOCAL

- Modified: changed, but not committed
- Staged: modified files marked to be committed
- Committed: version of file stored in local repo



HOW IT WORKS - BIG IDEA

- Committing -> like taking a picture of all files
- Links to old file if no change
- Underneath:
  - eventually objects are packed
  - packing basically stores as changes from closest overall state of project

FIRST TIME USING GIT

- Install
- Configure (specify name, email, default editor)
- Most likely want to do for all of your repos on this computer
  - `--global` option
- Can specify for:
  - whole system (`--system`)
  - single repo (`--local`)

CONFIGURING GIT

- How to configure:

```
git config --global user.name "Erin Carrier"

git config --global user.email carrieer@gvsu.edu
```

- View configuration:

```
git config --list
```

COMMON REFERENCES

- `HEAD` - reference to tip of current branch
- index - staging area
- SHA-1 hashes - for referencing commit hashes

BASIC OPERATIONS LOCALLY

- `git init` -> initalize new repo
- `git add` -> stage changes to file (or add untracked file)
- `git rm` -> remove
- `git mv` -> rename or move
- `git commit` -> commit staged changes

BASIC OPERATIONS LOCALLY (CONT.)

- `git status`
- `git diff`: view changes
- `git log`: view list of commits

MINILAB 5

In the minilab folder.