Git Tutorial

A Distributed Version-Control System

- Chapter 1 -

Getting Started

- Getting Started -

About Version Control

What is “ version control ” , and why should you care?

- A system that records changes to a file.
- You can recall specific versions later.

About Version Control – Local Version Control Systems

- The method that many people implement a VCS.
- Programmers method.
- RCS was one of most popular VCS tools.

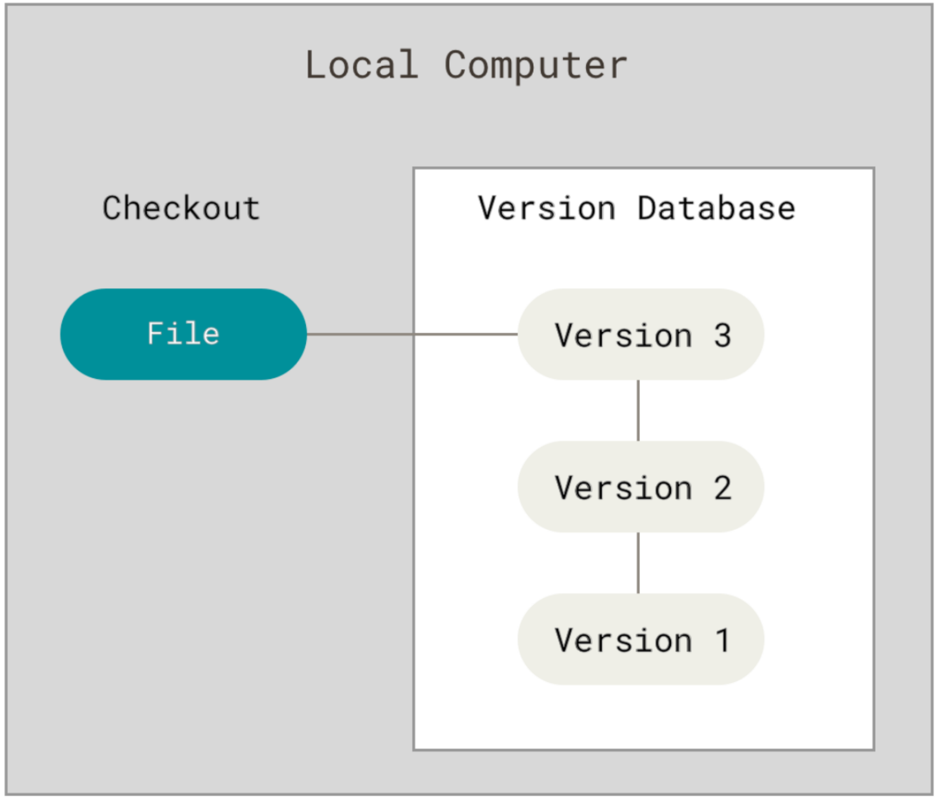


Figure 1. Local version control

- Need to collaborate with developers on other systems.
- So Centralized Version Control Systems (CVCSs) were developed.
- Advantages of CVCSs.
- Downsides of CVCSs.

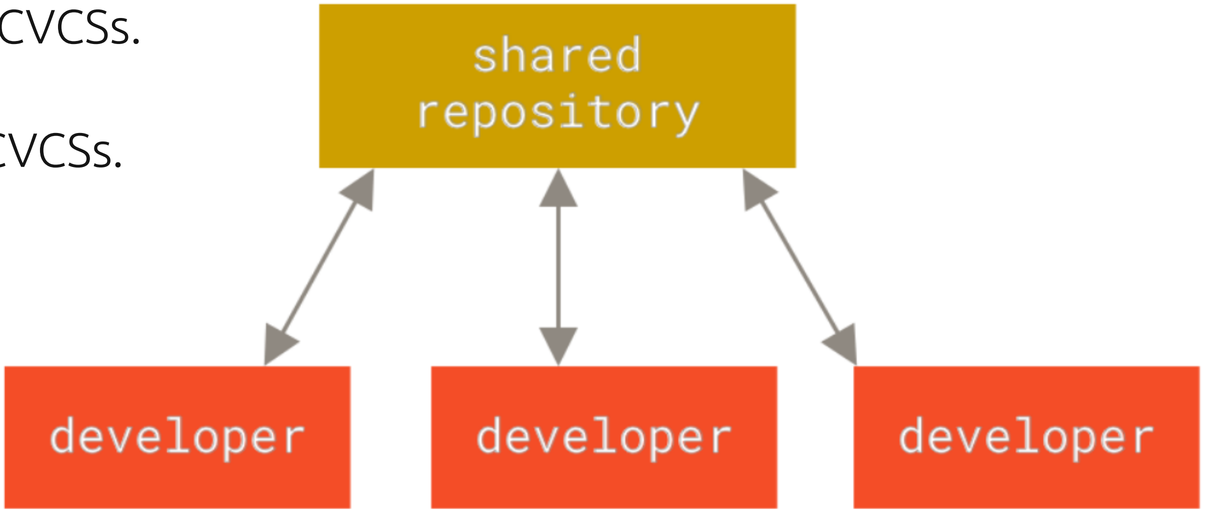


Figure 2. Centralized version control

Getting Started
About Version Control – Distributed Version Control Systems

- Because of all these reasons, DVCSs (such as Git) step in.
- They fullymirror the repository, including its full history.
- Every clone is really a full backup of all the data.
- Several remote repositories.
- Several types of workflows.

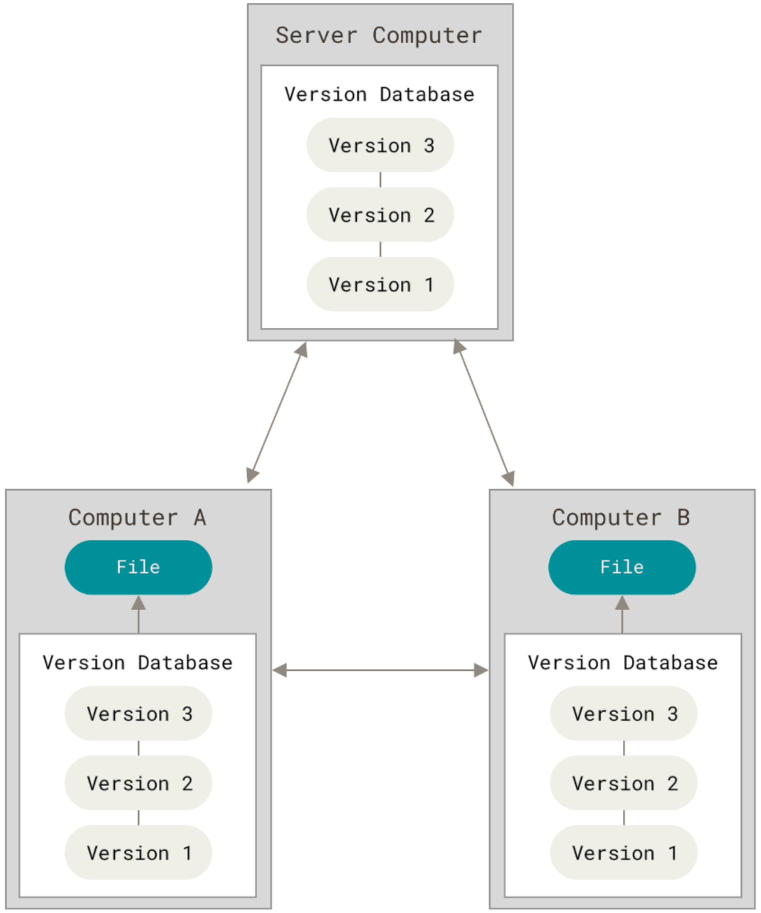


Figure 3. Distributed version control

- Getting Started -

A Short History of Git

A Short History of Git

- Git began with a bit of creative destruction.

- The story of Linux kernel and developing Git.

- Some goals of Git:
  - Speed.
  - Simple design.
  - Strong support for non-linear development.
  - Fully distributed.
  - Able to handle large projects like the Linux kernel efficiently.

- Git was born in 2005.

- Getting Started -

What is Git?

- Understanding Git and its fundamentals are very important.
- Clear your mind of the things you may know about other VCSs.
- It will help you avoid subtle confusion when using the tool.

The major difference between Git and other VCSs?

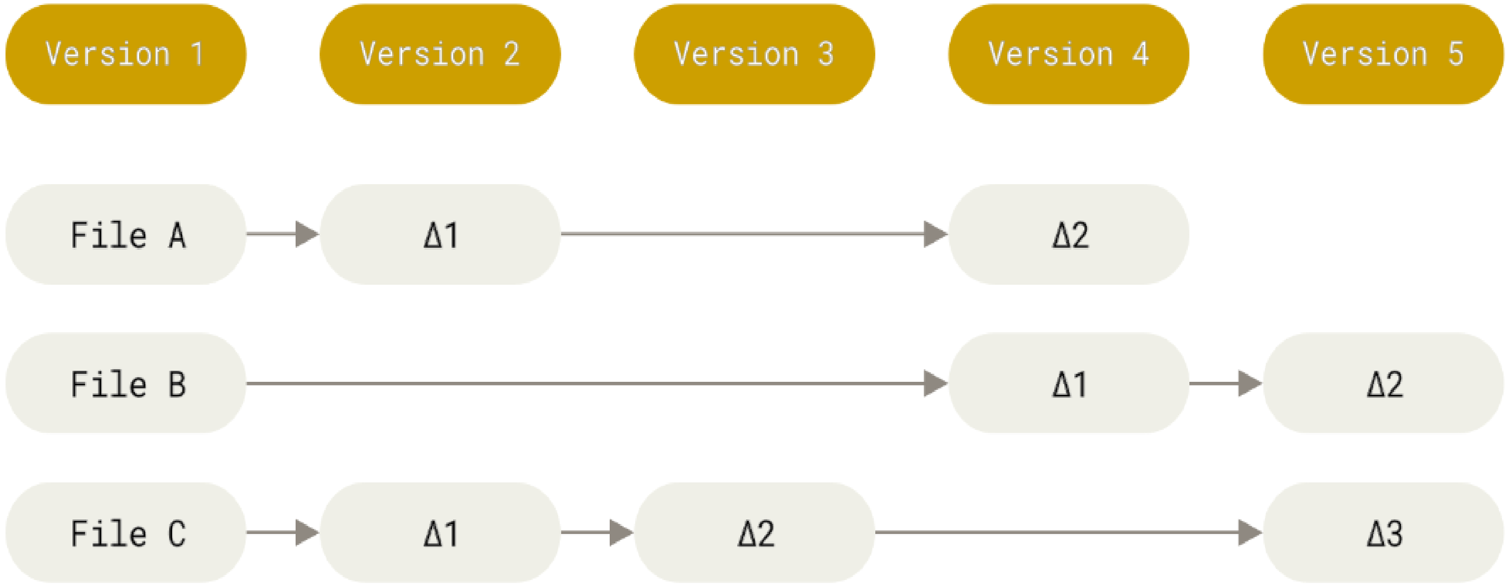


Figure 4. Storing data as changes to a base version of each file

Git thinks about its data more like a stream of snapshots.

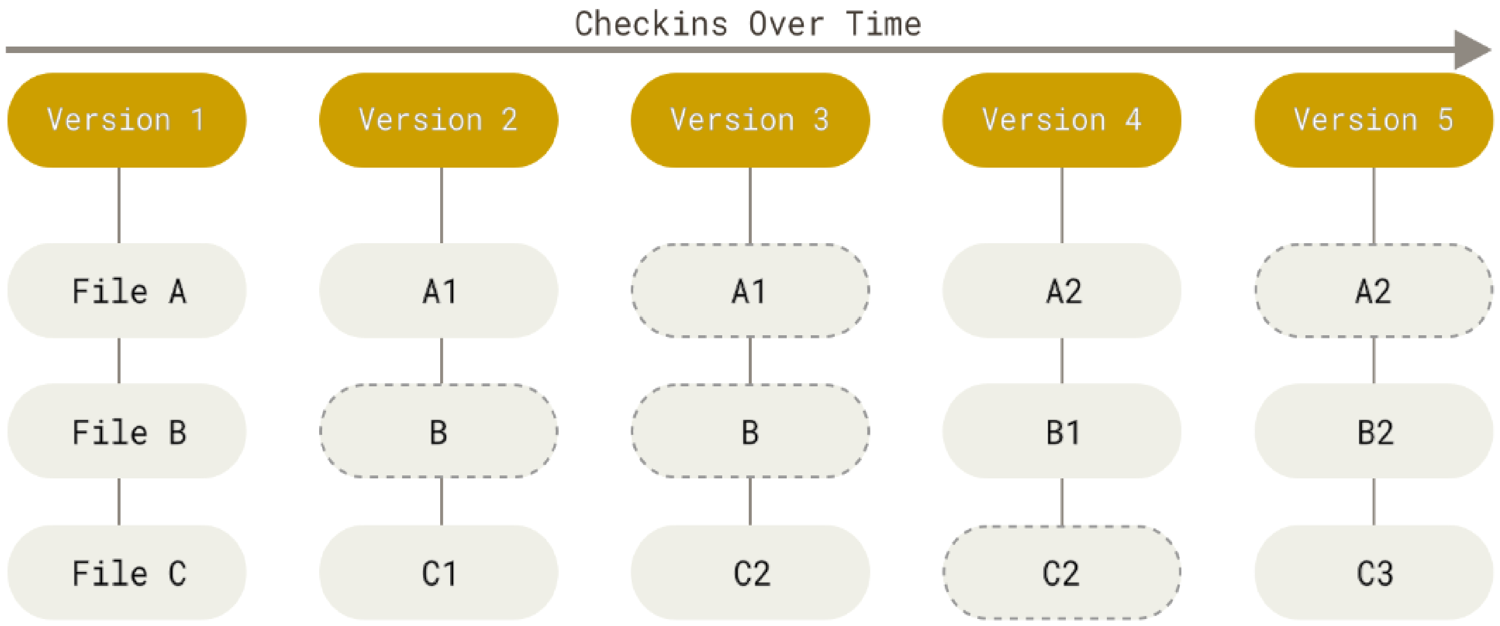


Figure 5. Storing data as snapshots of the project over time

What is Git? - Nearly Every Operation Is Local

- Most operations in Git need only local files to operate.
- For example, to browse the history of the project.
- So, it seems that the gods of speed have blessed Git with unworldly powers.

- Everything in Git is checksummed before it is stored and is then referred to by that checksum.
- For example, to browse the history of the project.
- The mechanism of Git checksumming is called SHA-1 hash.

```
24b9da6552252987aa493b52f8696cd6d3b00373
```

What is Git? - Git Generally Only Adds Data

- Nearly all of your actions only add data to the Git database.
- It is hard to make Git, erase data in any way.
- This makes using Git a joy.

- Git has three main states that your files can reside in:
  - Modified.
  - Staged.
  - Committed.

- The basic Git workflow goes something like this.

What is Git? - The Three States

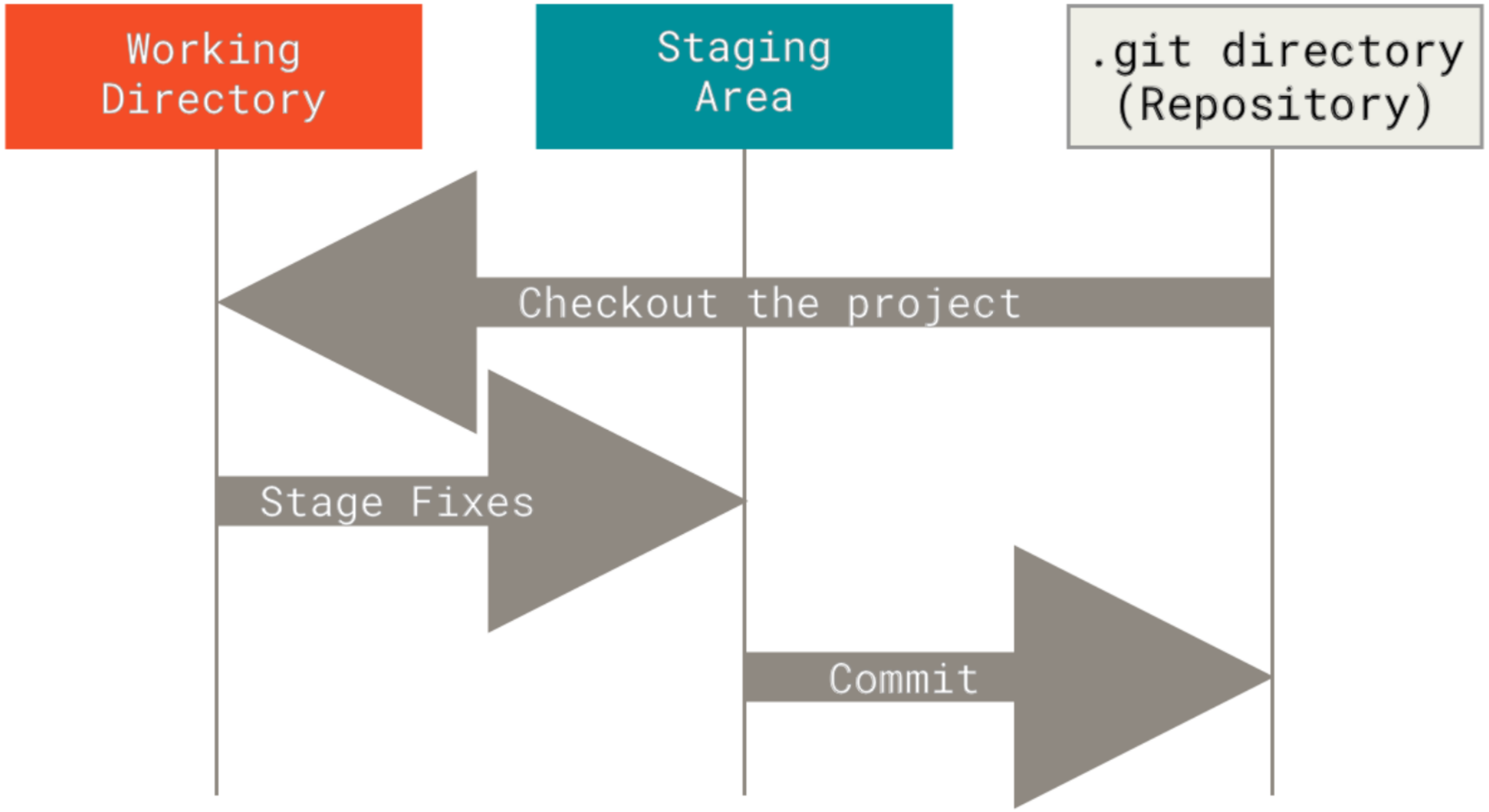


Figure 6. Working tree, staging area, and Git directory

- Getting Started -

The Command Line

- There are a lot of different ways to use Git:
  - Original command-line.
  - Other graphical user interfaces.

- Which one is better?

- Getting Started -

Installing Git

Lets do it ...

Page 17 of The Book

- Getting Started -

First-Time Git Setup

First-Time Git Setup

- Git config is a tool that lets you get and set configuration variables.

- These variables can be stored in three different places:
  - Unix OSs:
    - ✓ `[path]/etc/gitconfig` file
    - ✓ `~/.gitconfig` or `~/.config/git/config` file
    - ✓ `.git/config` file in your current repository
  - Windows:
    - ✓ `[path]/etc/gitconfig` file
    - ✓ `.gitconfig` file in `$HOME` directory (`C:\Users\$USER` for most people)

- You can view all of your settings and where they are coming from using:

```
$ git config --list --show-origin
```

First-Time Git Setup - Your Identity

- Set your user name and email address (Once and for all):

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

- Diffrent name and email for specefic projects.

First-Time Git Setup - Your Editor

- Set the default text editor (e.g. Vim):

```
$ git config --global core.editor vim
```

- On windows you must specify the full path (e.g. Nodepad++):

```
$ git config --global core.editor
"'C:/Program Files/Notepad++/notepad++.exe'
-multiInst -notabbar -nosession -noPlugin"
```

First-Time Git Setup - Branch Name

- The default branch name is “master”.
- To change that, use this command (e.g. main):

```
$ git config --global init.defaultBranch main
```

First-Time Git Setup - Checking Your Settings

- To check your configuration settings:

```
$ git config –list
user.name=John Doe
user.email=johndoe@example.com
color.status=auto
...
```

- You can also check a specific key's value:

```
$ git config user.name
John Doe
```

- Getting Started -

Getting Help

Getting Help

- There is three equivalent ways:

```
$ git help <verb>
$ git <verb> --help
$ man git-<verb>
```

- Or get help for specefic command (e.g. git config command):

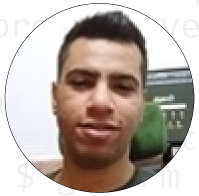
```
$ git help config
```

- Or get more concise help with -h command:

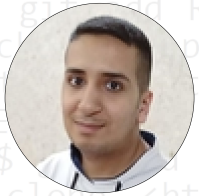
```
$ git add -h
usage: git add [<options>] [--] <pathspec>...

    -n, --dry-run         dry run
    -n, --dry-run         dry run
    -i, --interactive     interactive picking
    -p, --patch           select hunks interactively
    -e, --edit            edit current diff and apply
```

Group Members



Mohammad A. S. Minabi
bigm00bnd@gmail.com



Mohammad H. Bahrampour
bahrampour@pm.me

Hamid R. K. Pishghadam
kaveh@riseup.net