Distributed Version Control Systems

In Git clients don't just check out the latest snapshot of the files; rather, they fully mirror the repository, including its full history.

Every clone is really a full backup of all the data.

Git specs

Everything in Git is checksummed before it is stored and is then referred to by that checksum

Git uses for this checksumming an SHA-1 hash and is calculated based on the contents of a file or directory structure in Git.

Git stores everything in its database not by file name but by the hash value of its contents

When you do actions in Git, nearly all of them only add data to the Git database

Snapshots, Not Differences

Every time you commit Git basically takes a picture of what all your files look like at that moment and stores a reference to that snapshot.

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

The Three States

Git has three main states that your files can reside in:

- **Modified** means that you have changed the file but have not committed it to your database yet.

- **Staged** means that you have marked a modified file in its current version to go into your next commit snapshot.

- **Committed** means that the data is safely stored in your local database.

The Three Sections

This leads us to the three main sections of a Git project: the **working tree**, the **staging area**, and the **Git directory**.

The **working tree** is a single checkout of one version of the project.

The **staging area** is a file, generally contained in your Git directory, that stores information about what will go into your next commit.

The **Git directory** is where Git stores the metadata and object database for your project.