

Git

Kiran Vasudev¹

¹Hochschule Bonn-Rhein-Sieg

March 20, 2018

Outline

Introduction

Main states of a Git repository

States of files in your working directory

Some important commands

References

Introduction I

- ▶ Most widely used Version Control System(VCS)
- ▶ Takes snapshots of the system.

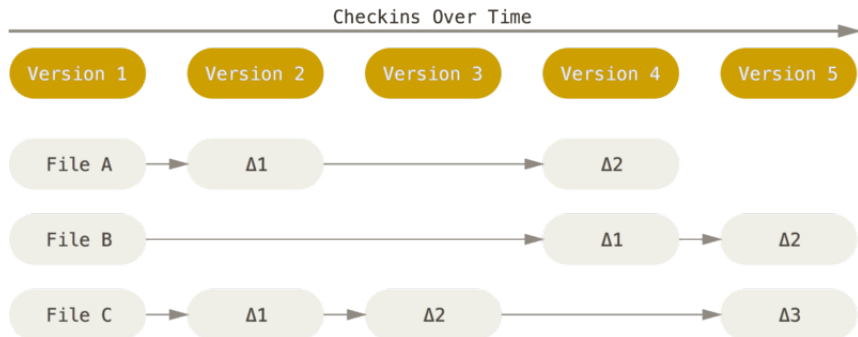


Figure: Status of files in other VCS[1]

Introduction II

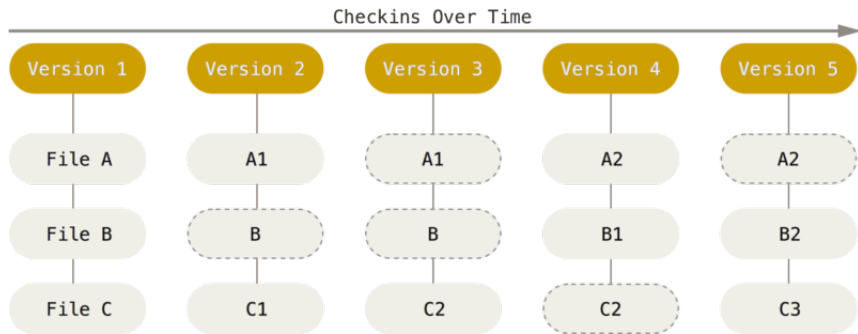


Figure: Different versions in the form of snapshots[1]

Main states of a Git repository

- ▶ Working directory
- ▶ Staging Area
- ▶ .git directory (repository)

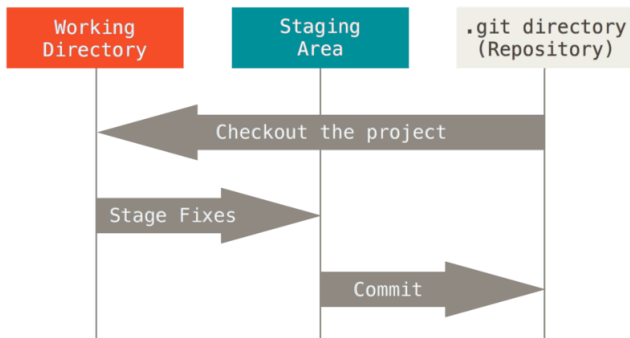


Figure: Working directory, staging area and the .git directory[1]

States of files in your working directory

- Files can either be tracked or untracked.

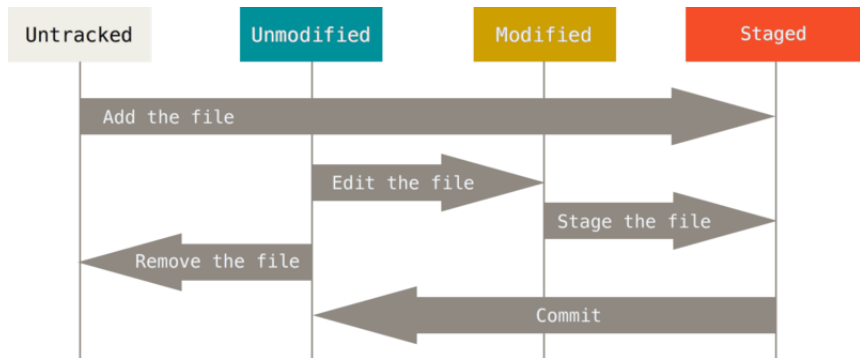


Figure: Lifecycle of the status of files[2]

Some important commands I

Initializing

- ▶ **git config --global user.name "user-name"**
- ▶ **git config --global user.email "email-id"**
- ▶ **git init**
Initializes folder to be recognized as a git folder

Cloning

- ▶ **git clone** <remote_location> <clone_name>
Clones a repository into a new directory
- ▶ **git pull** <repository_name>
Does a git fetch and then a merge
- ▶ **git fetch** < target_branch >
Gathers commits from target branch and stores them in your local repository but does not merge them with the current branch

Some important commands II

Tracking/removing files

- ▶ **git add** <f_name>
Adds the file to the stage
- ▶ **git rm** <f_name>
Removes the file from the working tree
- ▶ **git mv** <f_name>
Moves/renames the file in the working tree

Committing changes

- ▶ **git commit -m "message here"**
Commits the file to the repository

Checking file status

- ▶ **git status**
To list out the status of files and folders
- ▶ **git diff**
Shows changes between commits

Some important commands III

Tagging

- ▶ **git tag** <tag_name>
Used to tag a commit

Branching

- ▶ **git branch**
Lists the available branches. Can also be used to create a new branch with the argument <branch_name>
- ▶ **git checkout** <b_name>
Switches to another branch
- ▶ **git merge** < b_name >
Merges two development histories together

Viewing commit history

- ▶ **git log**
Lists the recent commit history

Some important commands IV

Working with remotes

- ▶ **git remote add** $\langle \text{remote_name} \rangle$ $\langle \text{remote_location} \rangle$
Remotes are used to track repositories
- ▶ **git remote -v**
Lists all the remotes
- ▶ **git push** $\langle \text{remote_name} \rangle$ $\langle \text{branch_name} \rangle$
Pushes local changes to the remote

Lets get our hands dirty !

References



Git basics



Recording changes in Git