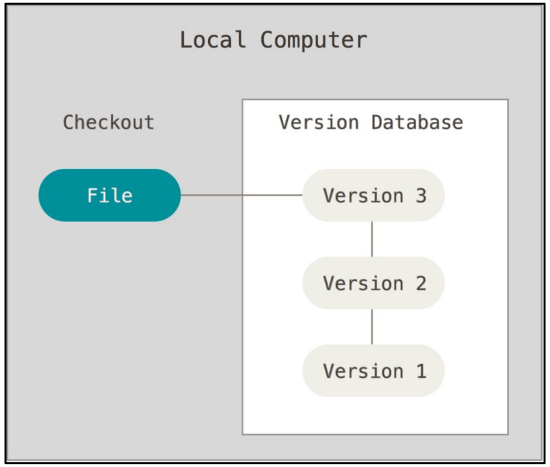
Introduction to git

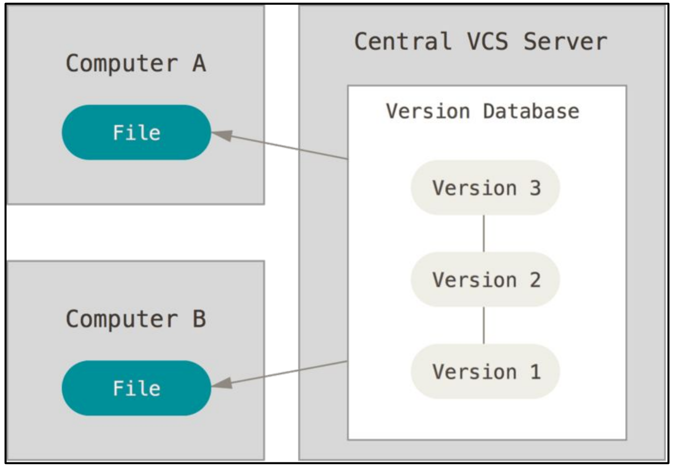
12 July 2022

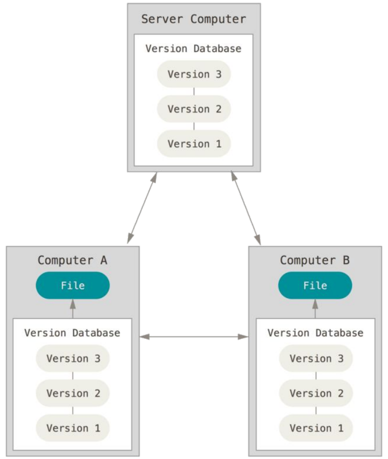
Version control systems

What is version control and why should you care?

Version control keeps history of changes







Local version control

- Lets you keep versions of files locally, create and merge new versions.
- Con: Works locally

Centralized version control systems

- Remote server keeps a database of all changes.
- Users checkout files locally, make changes and merge them.
- Con: Single point of failure

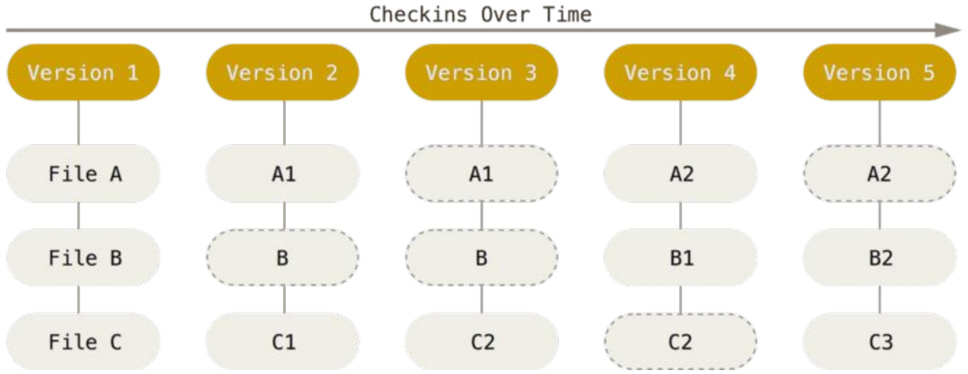
Distributed version control systems

- Remote server keeps a database of all changes
- Users checkout the entire repository, make changes and merge them in the remote server.

Git basics

Git fundamentals and glossary

git fundamentals



Definition

Git thinks of data more like a series of snapshots of a mini filesystem.
Every time you commit data, you create a snapshot.
The entire project history, combined with metadata is called a git **repository.**

Most operations in Git only need local files and resources to operate.

When you perform actions in Git, nearly all of them only **add** data to your repository.

The Three States

modified

The file is changed compared to the current snapshot.

staged

The file is marked to be included in the next snapshot.

committed

The file is part of a snapshot.

Sections of a git project

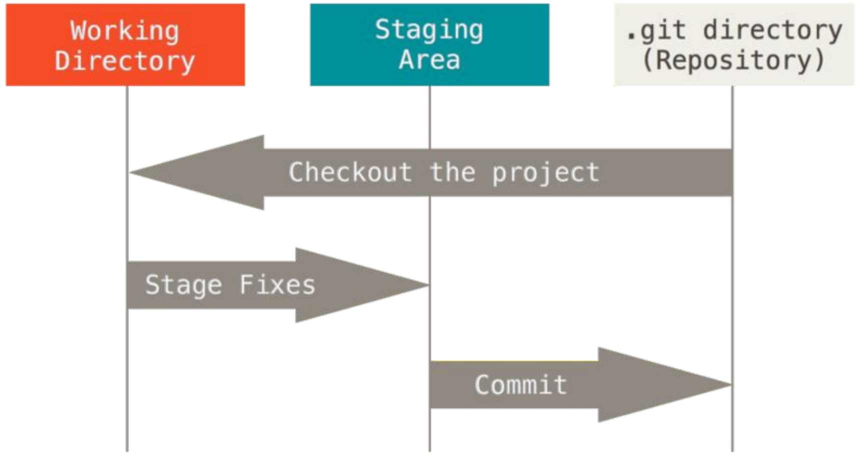
Creating commits

Working directory -> current snapshots
Staging area -> staged for next snapshots

checkout -> get snapshot from .git to working dir to make changes
commit -> create new commit (snapshot) and add it to .git locally

Files in the work directory can be 2 types:
tracked -> file was in the last snapshot, git knows about it
untracked -> file was not in the last snapshot, git does not know about it

Ignoring files -> specify them in .gitignore



Creating commits

Commands

Some git commands that are useful for creating commits.

git status
Get the status current status of the git project.

git add
Add a **modified** file from the working dir -> staging area

git commit
Create a snapshot from the files in the staging area.

git diff
Compare changes between two different snapshots.

git log
Get a history of changes.

git blame
Get a line by line visualisation of **who** modified every line of code.

Working with remotes

Remote repositories

Definition

Remote repositories are versions of your project that are hosted on the Internet or a network somewhere. This is, for example, GitHub.

They can be more than one.

clone -> cloning a repository means downloading a remote repository locally
origin -> this is the original repository that you cloned

```
$ git remote -v

bakkdoor  https://github.com/bakkdoor/grit (fetch)
bakkdoor  https://github.com/bakkdoor/grit (push)
cho45     https://github.com/cho45/grit (fetch)
cho45     https://github.com/cho45/grit (push)
defunkt   https://github.com/defunkt/grit (fetch)
defunkt   https://github.com/defunkt/grit (push)
koke      git://github.com/koke/grit.git (fetch)
koke      git://github.com/koke/grit.git (push)
origin    git@github.com:mojombo/grit.git (fetch)
origin    git@github.com:mojombo/grit.git (push)
```

Branches

How to merge changes more easily

git branches

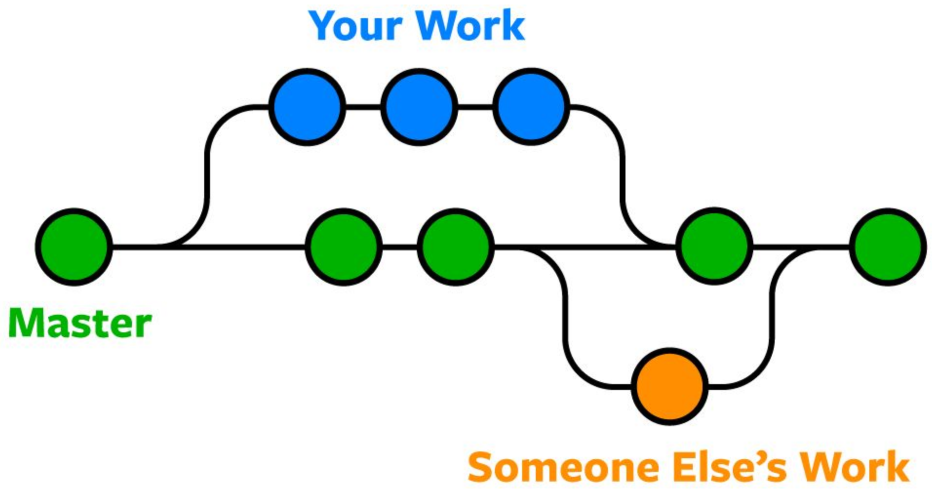
Definition

Having multiple people work on same project can make merging the changes difficult.
Branches are a way ease this process.

Every repository has one **main** branch.

When creating a new branch, we **diverge** from the main branch into a new one.
We make necessary changes and merge it into the main branch.

That way everything in the main branch is kept clean and deployable.



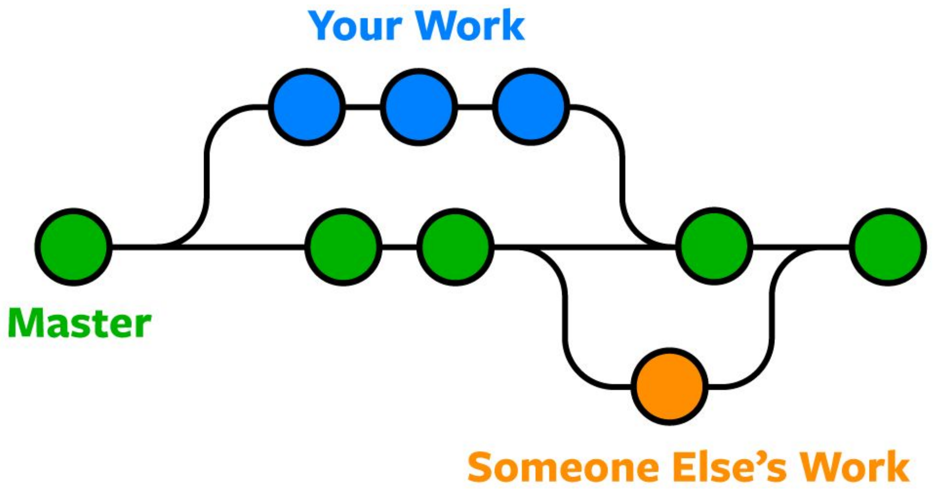
Merge conflicts

Definition

When we merge a branch into the main one, git will automatically try to merge both snapshots into one.

If it detects that the same file has been edited by two people at once, we get a **merge conflict**.

In that case, git will mark all conflicts inline in our project and make us merge them manually.



Working with branches

Concepts

Branches can exist locally and remotely.

In order to merge our changes, we need to **push** our branch remotely, so other people can find it.

When we want the latest version of a **remote** branch, we have **pull** it locally.

git pull
Downloads the new changes from the remote repository locally.

git push
Uploads the new local changes to a remote repository.

git checkout
Change the current working state to some other state. Can be a different commit or a different branch. This command is also used as an alias to create a new branch locally.

git rebase
Used to 'base' you current local branch on a different commit. Typically used to make commit history more linear.

Workflow

Your basic workflow when creating changes

Workflow

Create a new branch
Create, commit and test changes locally.

Push branch remotely

When ready for review, you can push you branch remotely, so other people can find it.

Open a pull request
Request a code review from a developer.

Merge
Merge the change into the main branch when everything is ok.

Tips

Keep commit messages clean and understandable

Doing this would make looking at commit history more easy in the future.

When opening a pull request always give context to the change you are making

DEMO