GENERAL ASSEMBLY

INTRO TO DATA SCIENCE

SESSION 2.1: INTRO TO VERSION CONTROL

Rob Hall

DAT13 SF // March 11, 2015

INTRO TO VERSION CONTROL

VERSION CONTROL

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it?*

VERSION CONTROL

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*

VERSION CONTROL

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*
- *But ever had a co-worker do that...? ;-)*

VERSION CONTROL

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*
- *But ever had a co-worker do that...? ;-)*
- *...and then pushed that code to your production website?!?!*

VERSION CONTROL

Why do we care about version control?

- *Ever had a hard drive crash?*
- *Ever had a computer stolen?*
- *Ever deleted the wrong file?*
- *Ever modified code and unintentionally broken it? (of course not)*
- *But ever had a co-worker do that...? ;-)*
- *...and then pushed that code to your production website?!?!*

That's why we care about version control!!

Version control is a system that records changes to a file or set of files over time so that we can recall specific versions later.

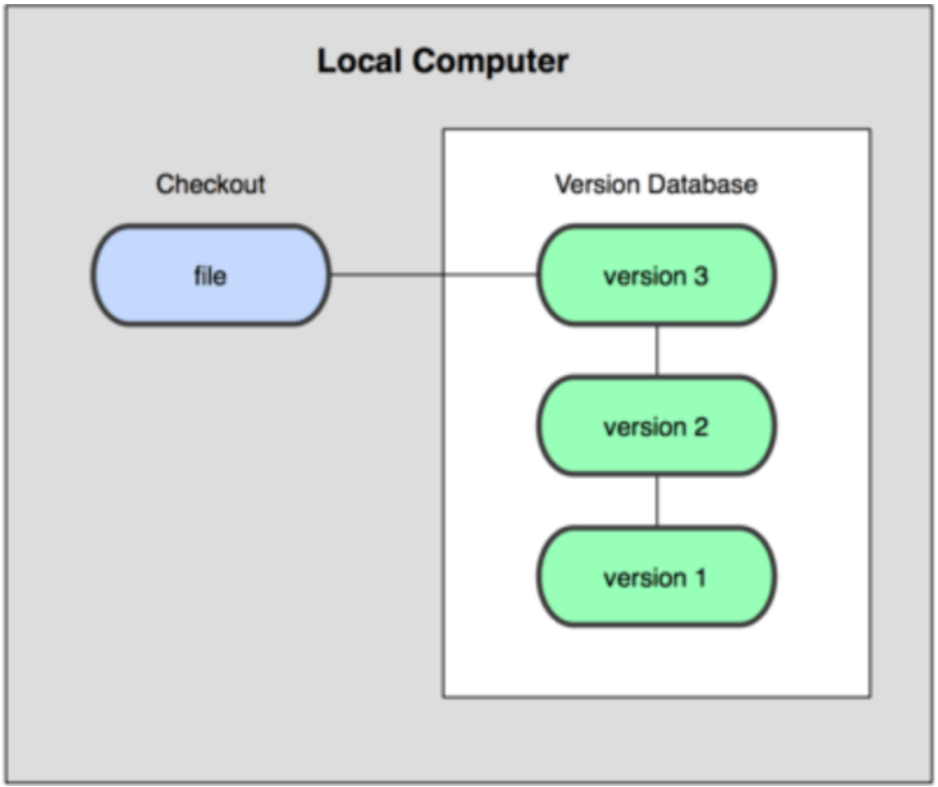
(Think of Time Machine for your Mac)

VERSION CONTROL

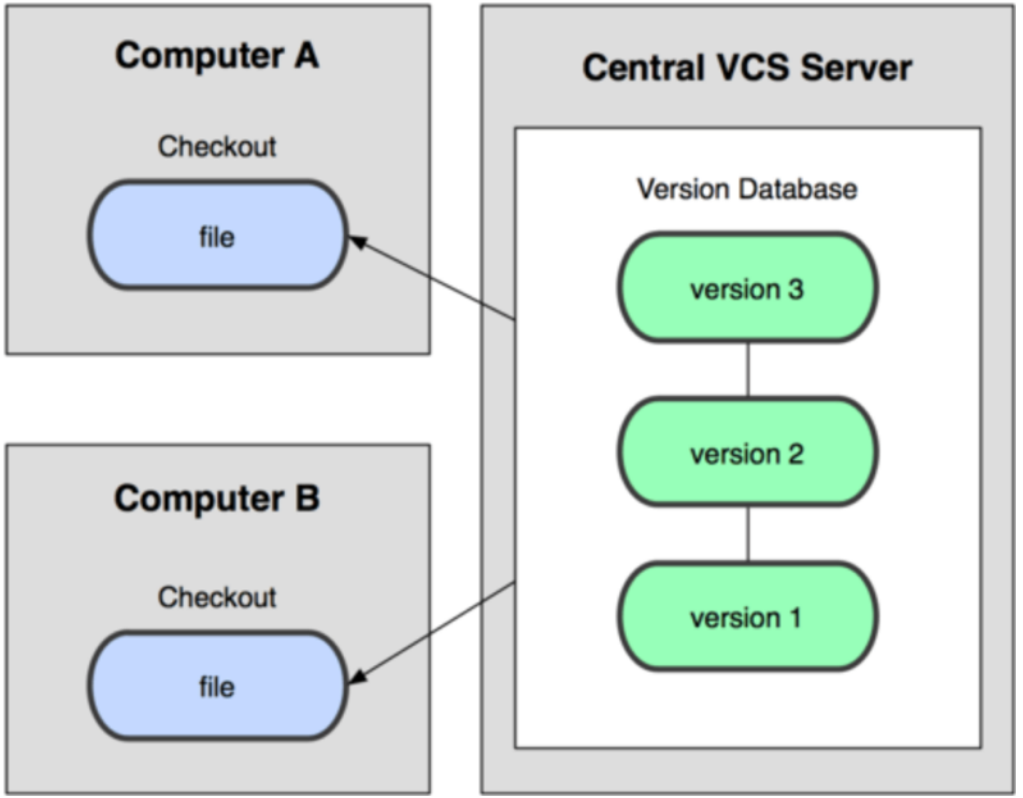
Version control systems (VCS) can be:

- *Local*
- *Centralized*
- *Distributed*

Local version control:

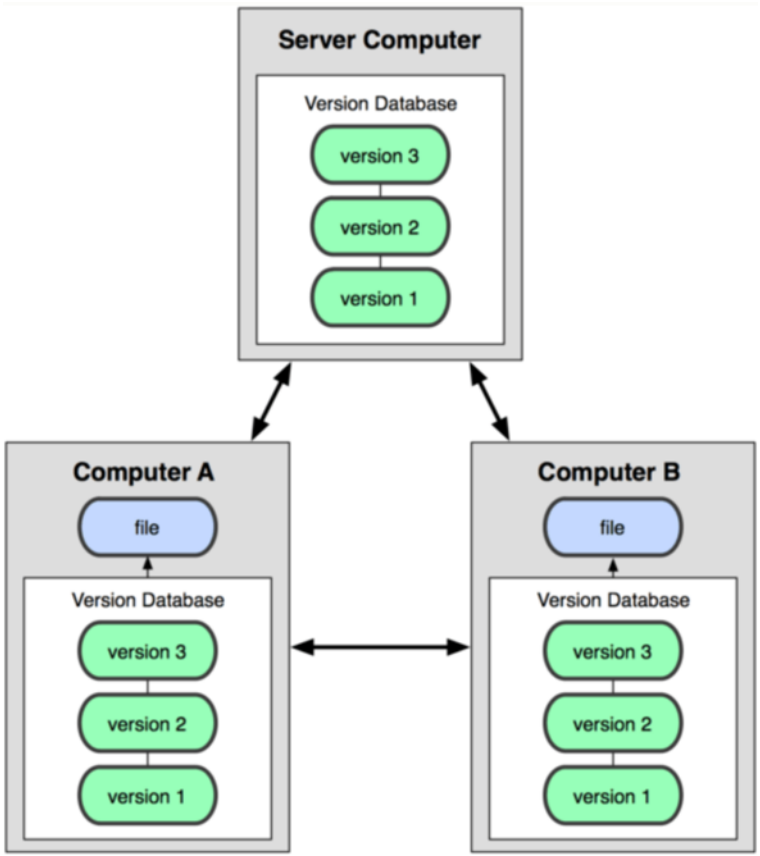


Centralized version control:



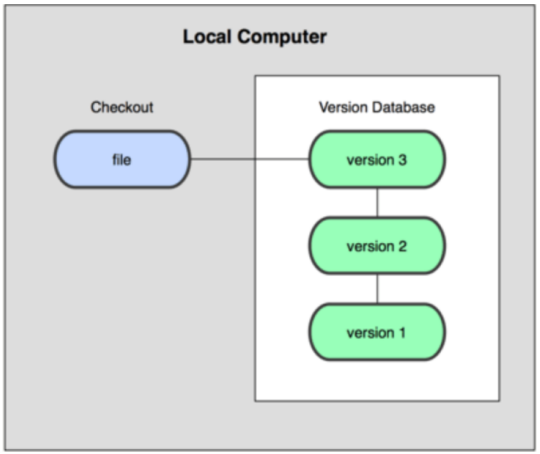
VERSION CONTROL

Distributed version control:

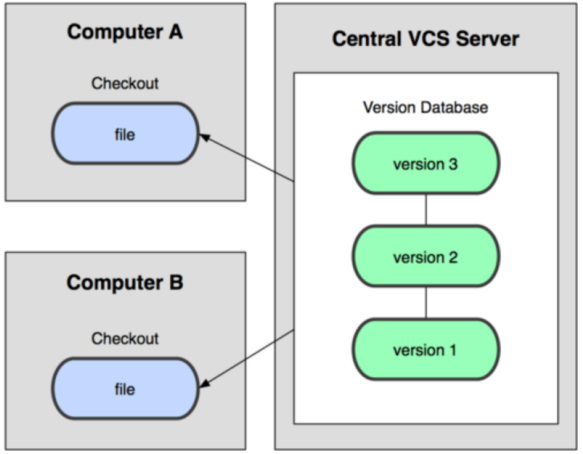


VERSION CONTROL

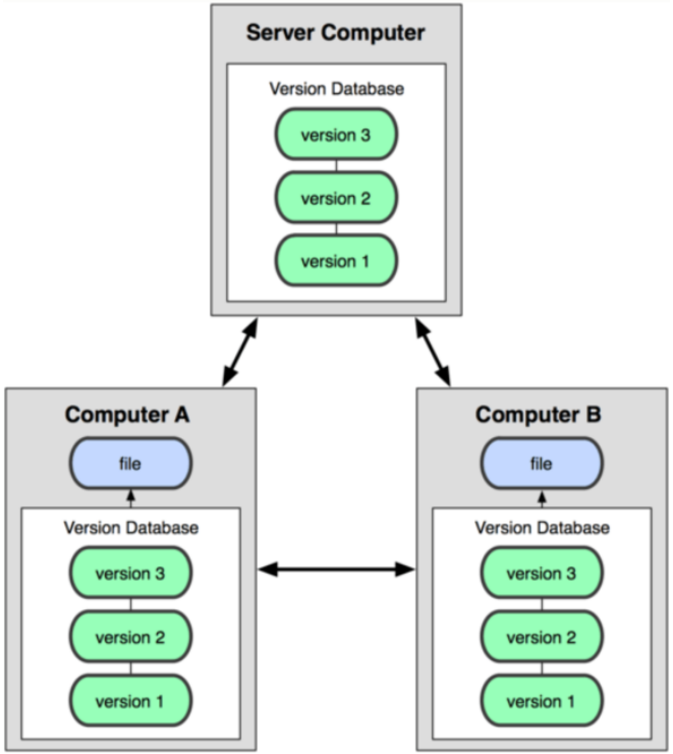
Local



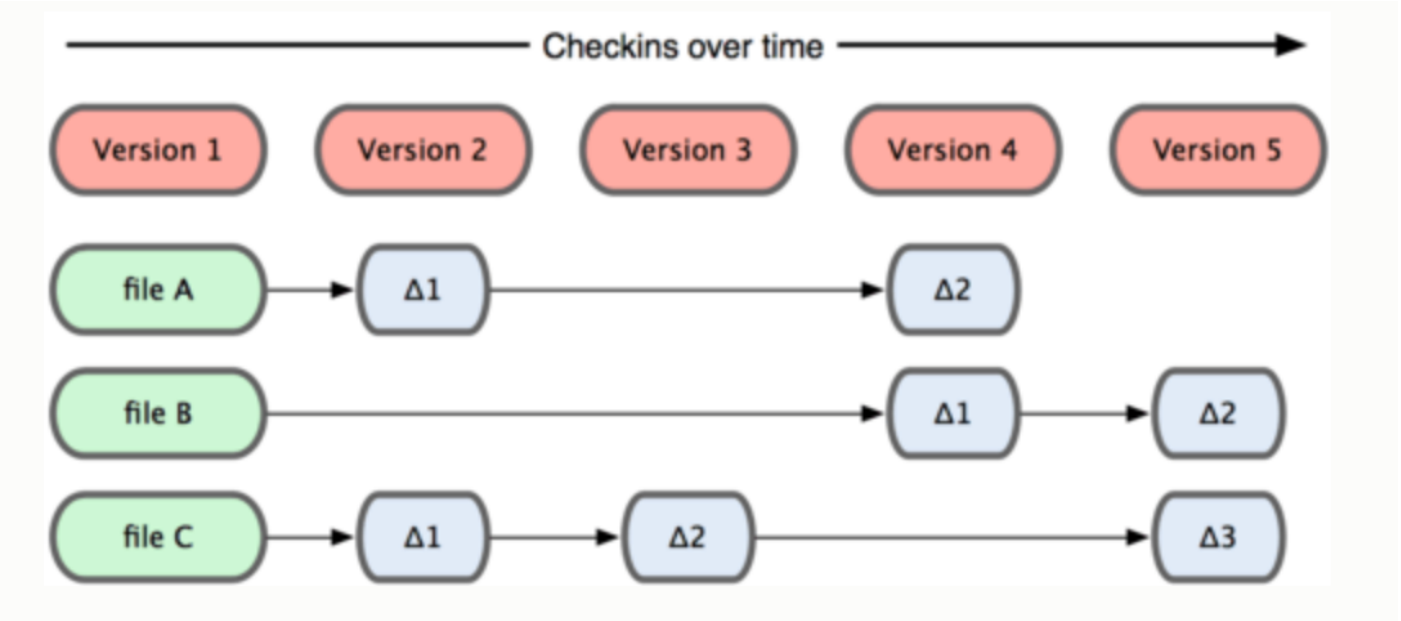
Centralized



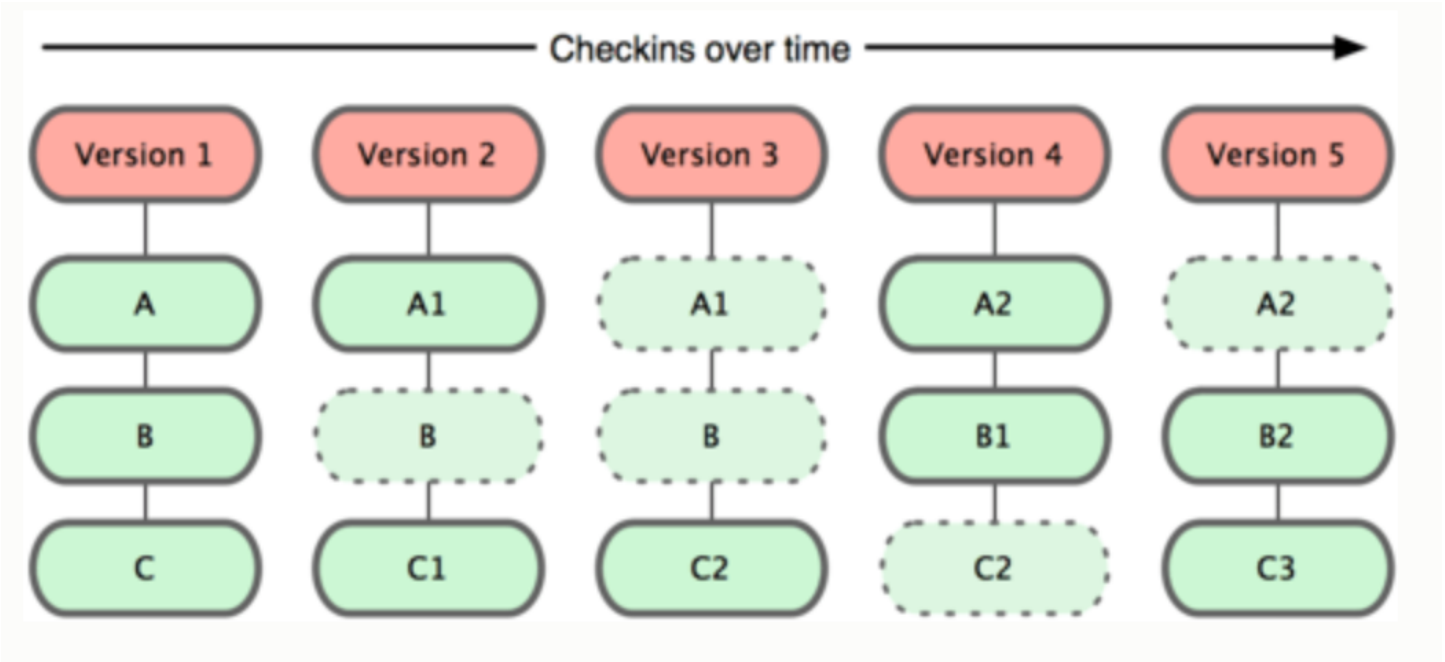
Distributed



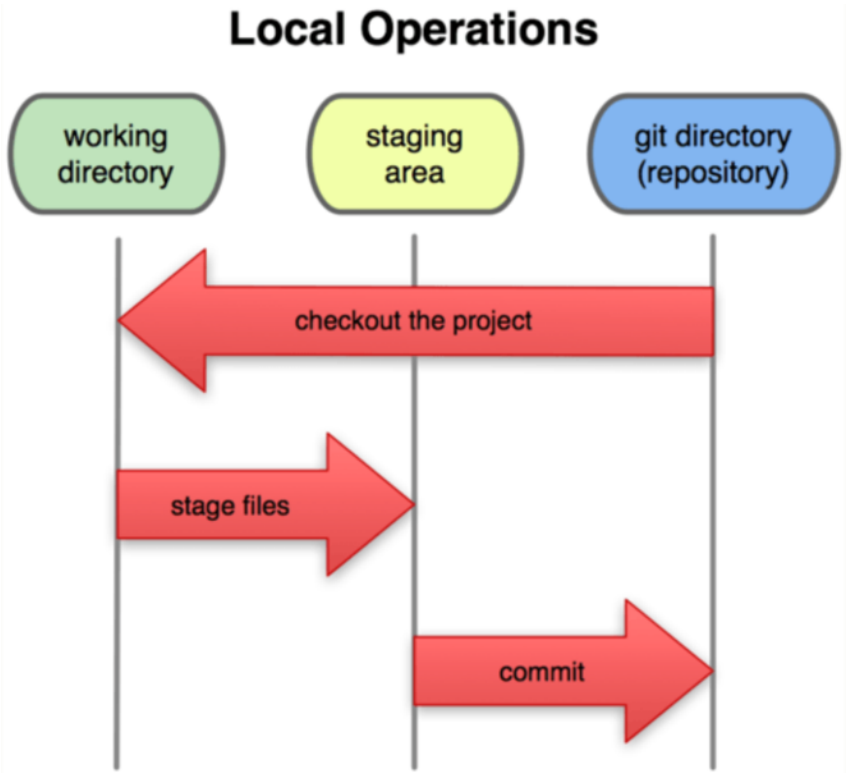
Traditional VCS' work in terms of files and differences



Git stores data as snapshots of a project over time



modified, staged, committed



The basic Git workflow

1. *You modify files in your working directory.*
2. *You stage those files by adding snapshots of the files to your staging area.*
3. *You do a commit, which takes the files in the staging area and stores that snapshot permanently to your Git directory.*

GIT COMMANDS

Main

git clone – clone a repo
git status – get status
git add – add changes to be pushed
git commit – commit the change with a comment
git push – push the change to github
git pull – pull remote changes from github

Others

git branch – see all branches
git checkout – checkout a branch
git merge – merge in another branch
git stash – stash changes
pull request – remote changes requested to be merged in