Atomic commits

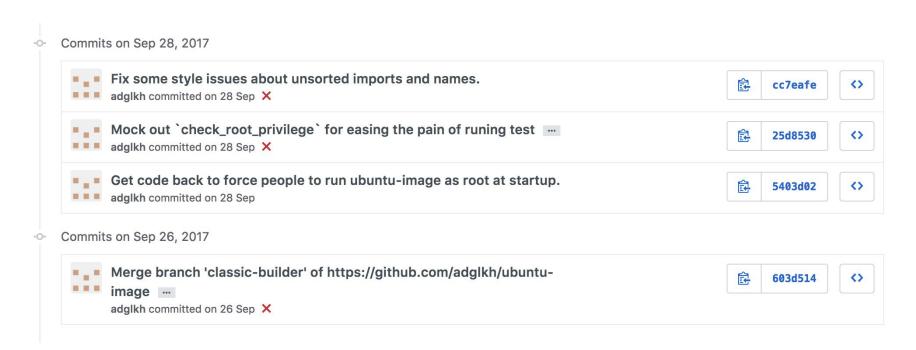
What they are and why they're important

Atomic commits

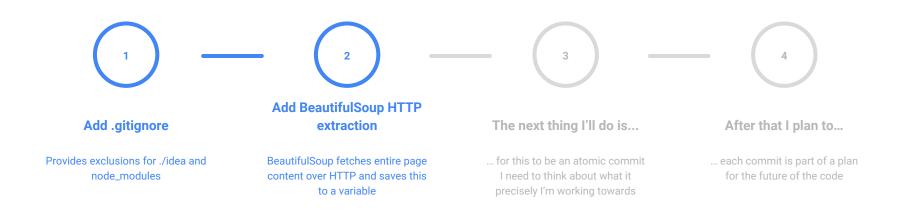
"Normally, you check in a group of files that **relate to a specific task or a bug you have fixed**. You **check them all in together**, along with a **meaningful log message** that will let folks in the future figure out **what** files changed and, more important, **why**. This kind of atomic commit will also help should you need to roll back the change.

Make sure all your unit tests still pass before checking in the code. One easy way for you to make sure the code in the control system is healthy is to use continuous integration."

Commits tell the 'story' of development



From one working state to a new working state



Clearly able to see what's been done and why

Forces you to be clear about what you're about to do and why

Related tools in Git

Git is great, but to really get the most of how it can help you, you'll benefit from doing atomic commits

- **Log** see the development story
- Interactive rebase rewrite history to tell a better story
- Revert reverse the effect of (any) previous commit(s)