Distributed Version Control

Outline

- Philosophical differences
- Typical workflow with a distributed system

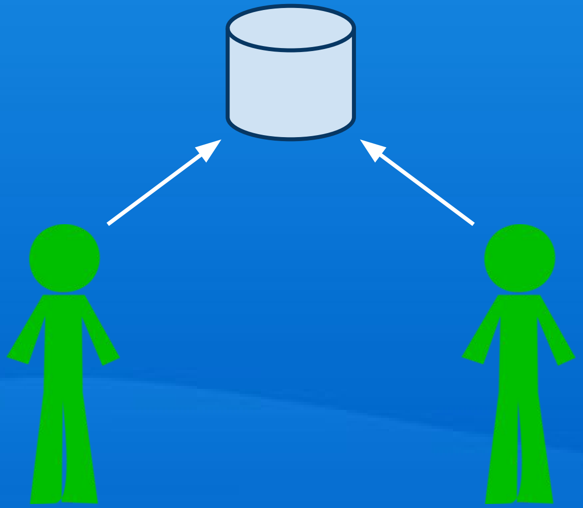
Outline

- Philosophical differences
- Typical workflow with a distributed system

Philosophical differences: repos

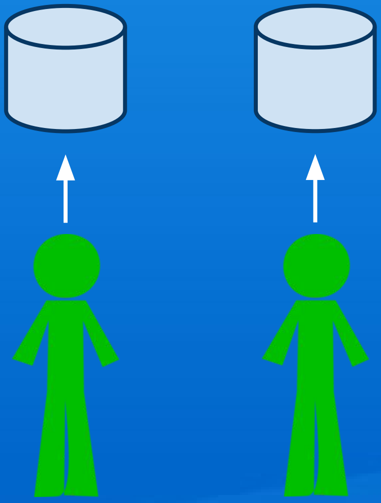
CVS/Subversion

- Everyone works with the same central repository



Mercurial/Git

- The repository is distributed - i.e. everyone has a copy of the repository



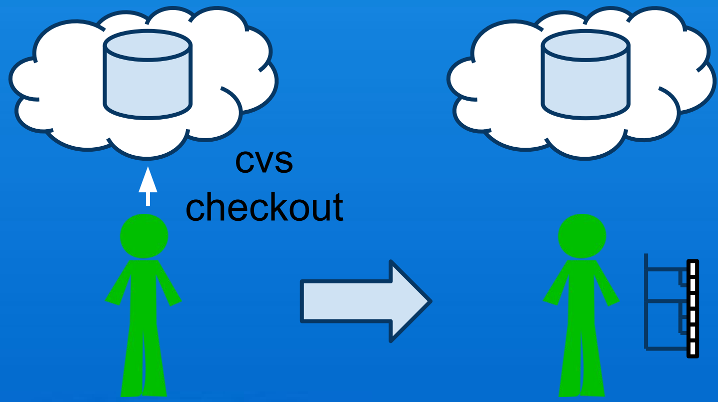
Philosophical differences: code

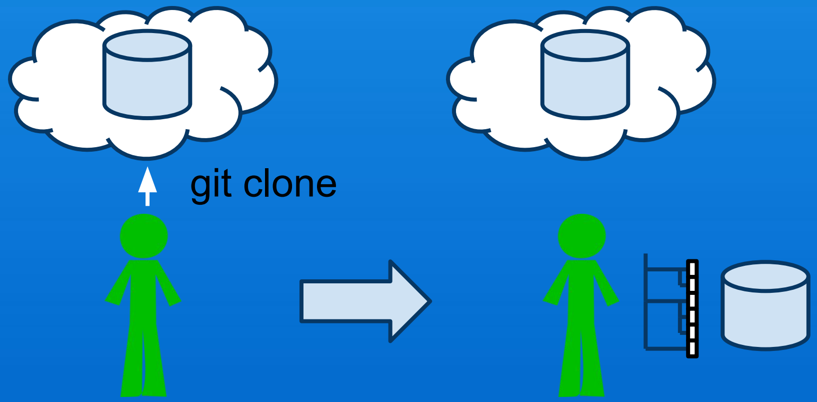
CVS/Subversion

- "checkout" gets you a copy of the latest and greatest code from the same central repository

Mercurial/Git

- "clone" of a particular repository gets you a full copy of the entire repository





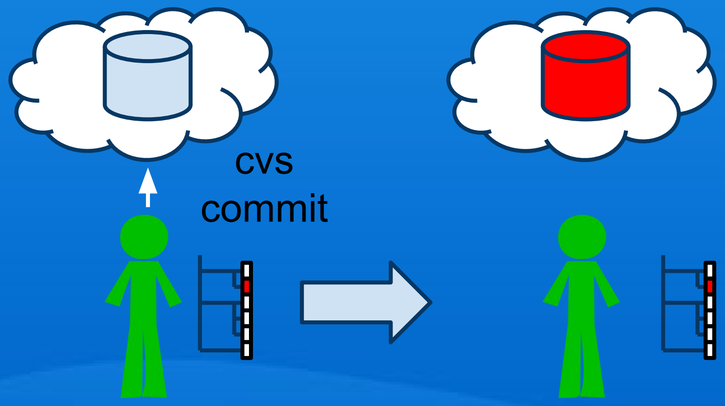
Philosophical differences: commit

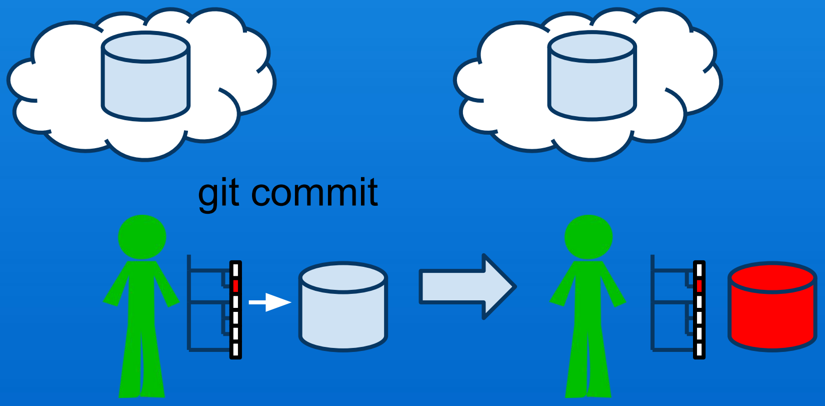
CVS/Subversion

- "commit" commits your changes to the central repository where everyone can see them

Mercurial/Git

- "commit" commits your changes to your own personal repository





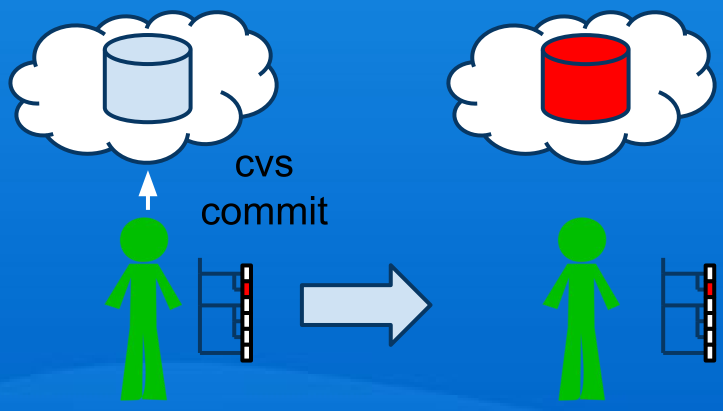
Philosophical differences: push

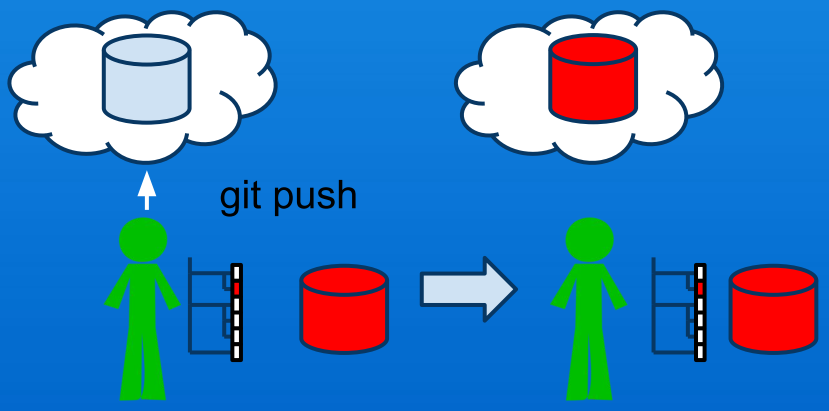
CVS/Subversion

- "commit" commits your changes to the central repository where everyone can see them

Mercurial/Git

- "push" moves your changes into another repository

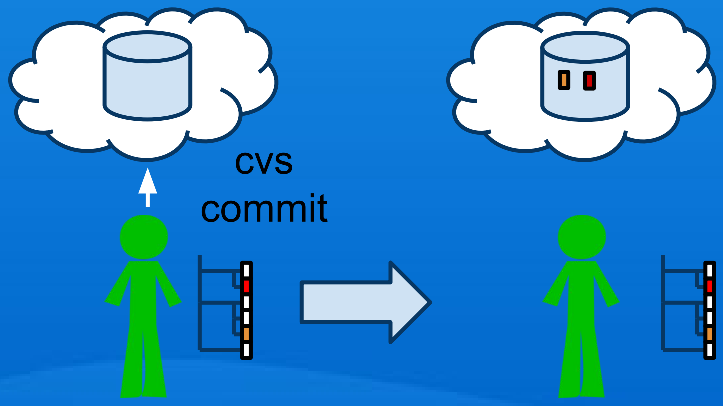




Philosophical differences: changesets

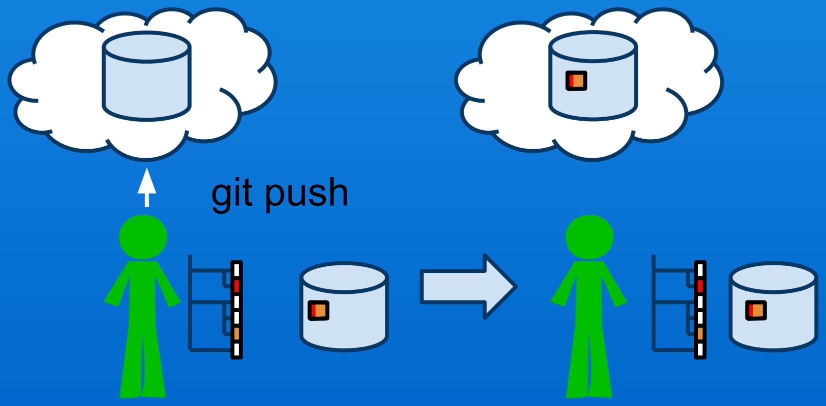
CVS (not Subversion!)

- "commit" commits your changes to the central repository one file at a time.



Mercurial/Git

- "push" commits your changes to the central repository one *changeset* at a time.

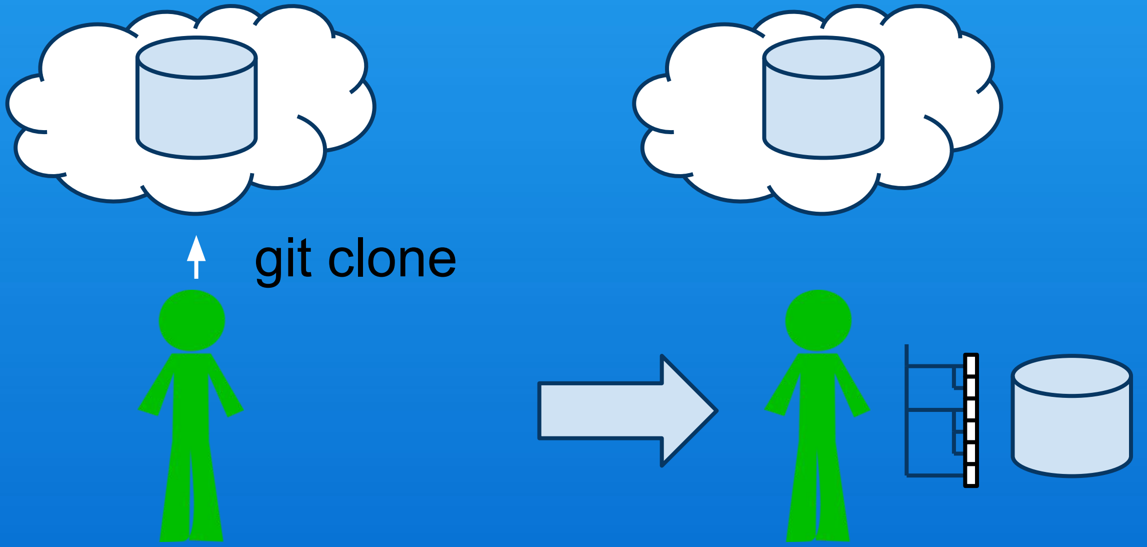


Outline

- Philosophical differences
- Typical workflow with a distributed system

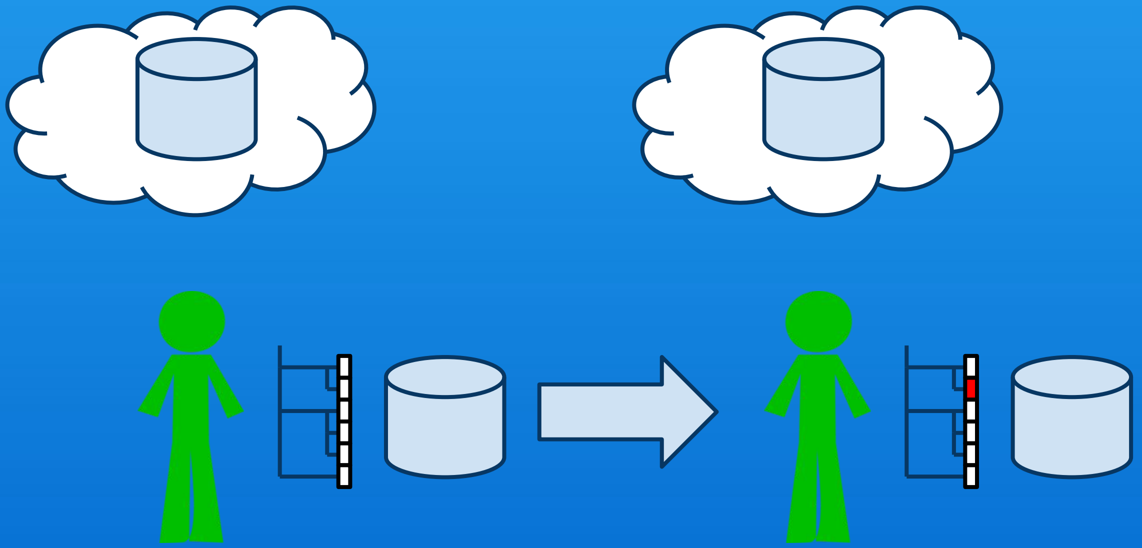
Typical Workflow in Git

1. Clone the central repository.



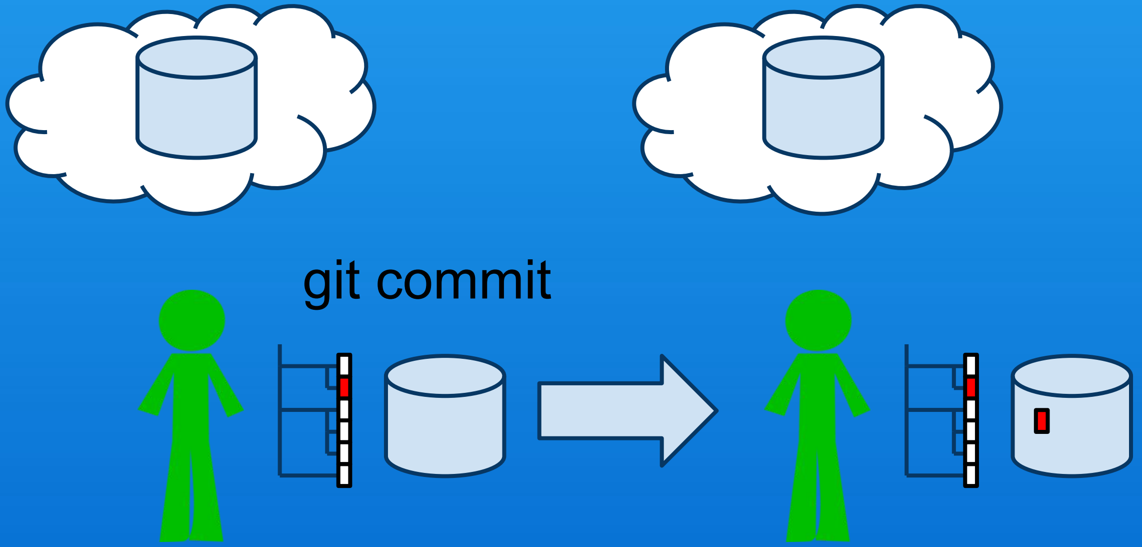
Typical Workflow in Git

2. Edit some code...



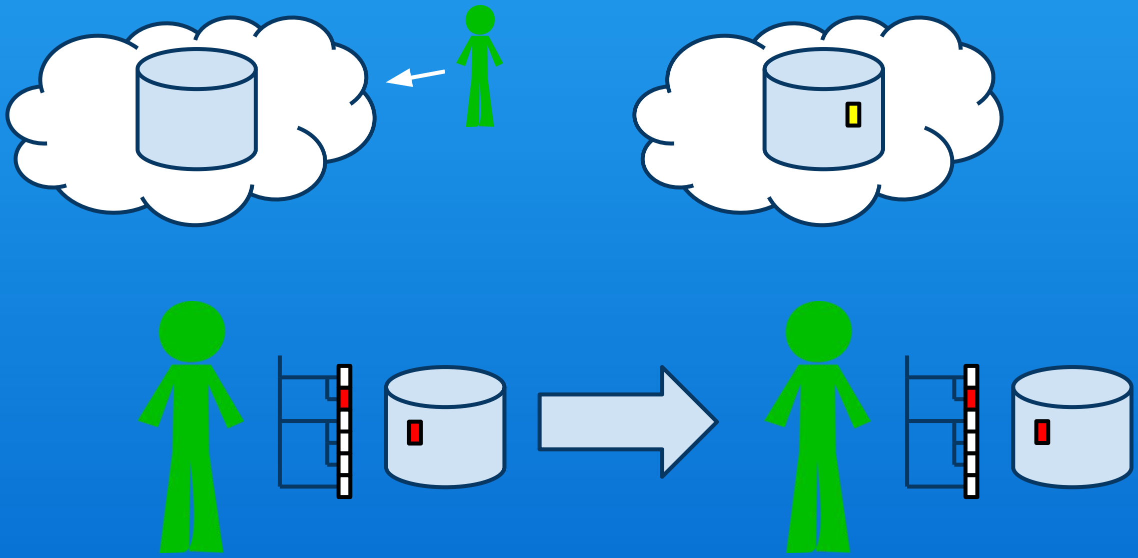
Typical Workflow in Git

3. Commit locally...



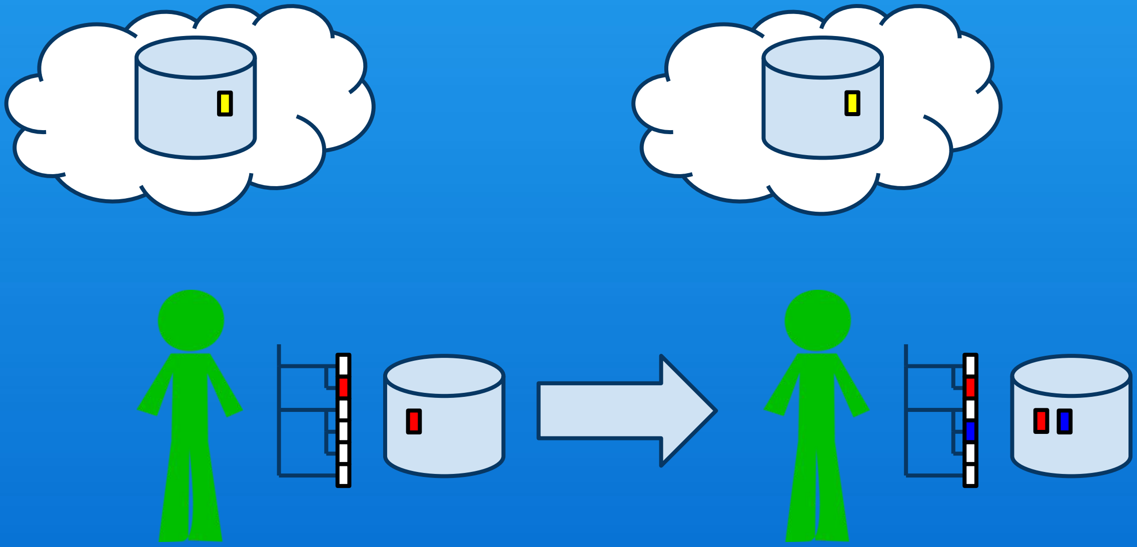
Typical Workflow in Git

4. Continue to work. Someone else puts a change in the central repository …



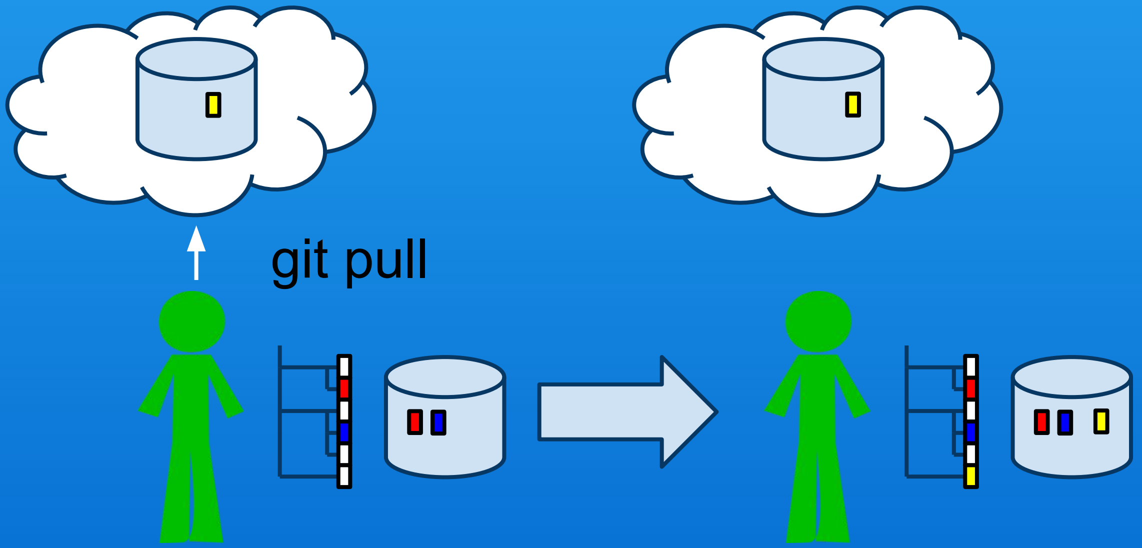
Typical Workflow in Git

5. Edit and commit until the code is in a stable (=tested!) state ...



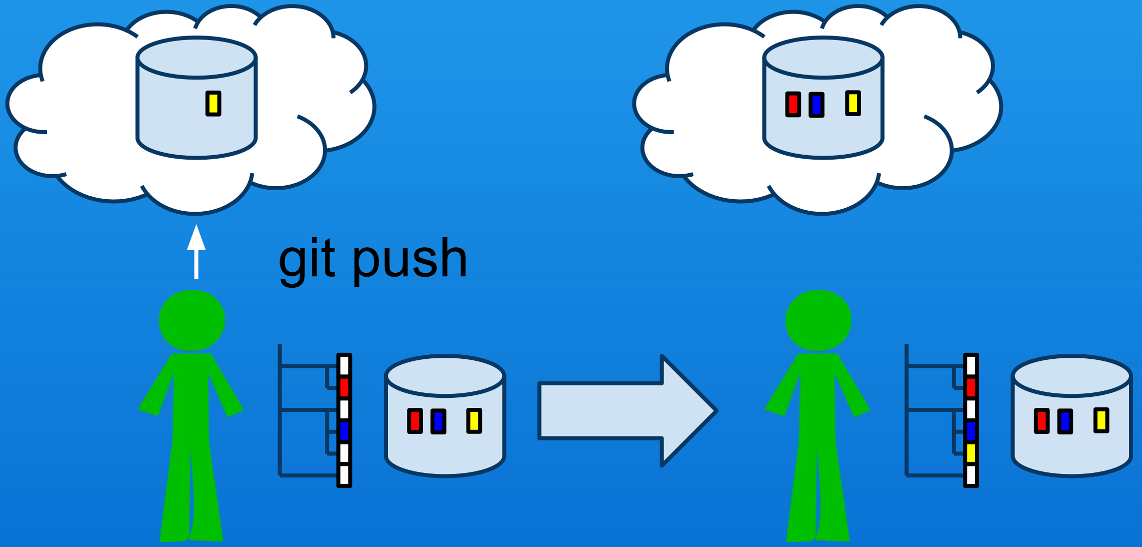
Typical Workflow in Git

6. Pull changes from the central repository.



Typical Workflow in Git

8. Push your updates to the central repository, where everyone else can see them.



Typical Workflow in Git

Caveat: There are many ways to setup your source control workflow with Mercurial/GIT!

- Repositories are cheap, so you can have just about as many as you like.
- You can think about a group having development, test, and deployment repositories, for example.
- Individual developers can have an "incoming" or clean repository, a development repository, and an outgoing repository.
- E.g. - http://nvie.com/posts/a-successful-git-branching-model/