Introduction to Version Control with Git

Andreas Skielboe

Dark Cosmology Centre
Niels Bohr Institute

January 17, 2012

Why Use Version Control?

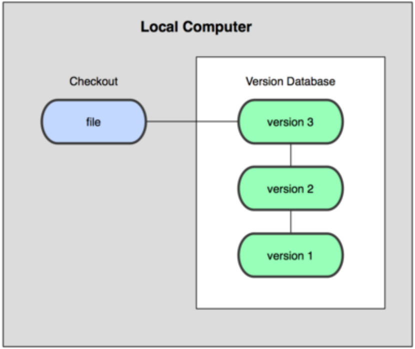
A Version Control System (VCS) is an integrated fool-proof framework for

- Backup and Restore
- Short and long-term undo
- Tracking changes
- Synchronization
- Collaborating
- Sandboxing

... with minimal overhead.

Local Version Control Systems

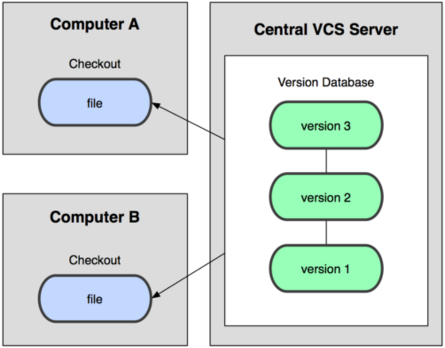
Conventional version control systems provides some of these features by making a local database with all changes made to files.



Any file can be recreated by getting changes from the database and patch them up.

Centralized Version Control Systems

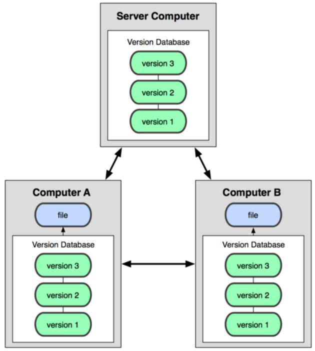
To enable synchronization and collaborative features the database is stored on a central VCS server, where everyone works in the same database.



Introduces problems: single point of failure, inability to work offline.

Distributed Version Control Systems

To overcome problems related to centralization, distributed VCSs (DVCSs) were invented. Keeping a complete copy of database in every working directory.



Actually the most **simple** and most **powerful** implementation of any VCS.

Git Basics

Git Basics

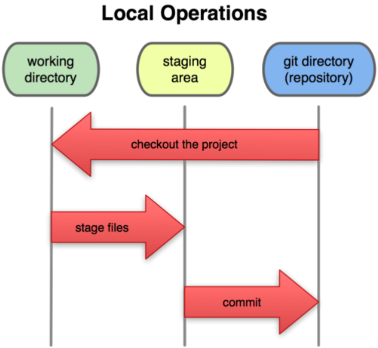
Git Basics - The Git Workflow

The simplest use of Git:

- **Modify** files in your *working directory*.
- **Stage** the files, adding snapshots of them to your *staging area*.
- **Commit**, takes files in the staging area and stores that snapshot permanently to your *Git directory*.

Git Basics - The Three States

The three basic states of files in your Git repository:



Git Basics

Basic characteristics of Git:

- Nearly every operation is local (speed, availability)
- Committed snapshots are always kept (integrity)
- Strong support for non-linear development (branching, merging, experimentation)

Hands-on

Hands-on with Git

First-Time Git Setup

Before using Git for the first time:

Pick your identity

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

Check your settings

```
$ git config --list
```

Get help

```
$ git help <verb>
```