git

Reuven M. Lerner, PhD
reuven@lerner.co.il • http://lerner.co.il/

Who am I?

- Open-source Web/database programmer, consultant, lecturer
- Git user (and enthusiast) since 2007
- I work regularly with Ruby, Python, JavaScript, and PostgreSQL (and several others)
- Learn more, read my blog, and subscribe to my newsletter: http://lerner.co.il

What is version control?

- Time machine
- Undo mistakes
- Work in parallel
- Try new things without spoiling the project
- Let people collaborate without stepping on each others' toes

Open-source version control

- RCS — locked files, one machine
- CVS — optimistic, non-locking, network
- Subversion — improved CVS with atomic commits, renaming

Git history

- Linus Torvalds and his tarballs
- BitKeeper (proprietary)
- Git for Linux kernel
- Now used by many open-source projects
- Also in use by many organizations
- Wildly popular — for good reason!

Git itself

- Web site: http://git-scm.com/
- The site contains links to downloads for most operating systems
- However, many operating systems come with their own, pre-installed versions of Git.

Git version

- Try to have a fairly recent version — although for most people, Git hasn't changed much in years

```
$ git --version

git version 2.15.0
```

- The version doesn't change *that* much, so don't worry about it hugely.

Repositories

- The cornerstone of Git
- Everything takes place “in” a repository
- Contains files, commits, history, etc.
- If you’re used to CVS or SVN, then “repository” might imply a network server
- Git doesn’t have servers, just repositories
- You need to know about repositories!

Create a repository!

- Go into a directory
- Type “git init”
- That’s it!
- That directory now functions as a Git repository
- Meaning: We can track the history of any file or directory within it

Creating a repository

```
$ git init

Initialized empty Git repository in
Courses/Git/Programs/simple/.git/
```

Yes, it's this easy

- Many Unix users and system administrators make everything a Git repository

- /etc directory, home directory, .emacs directory… if it's a directory, it can be a Git repository, as well

- Git adds very little overhead, and ensures your files are always safe. Why not?

.git

- Everything about a Git repository, except for the files themselves, lives in the ".git" directory.
- The history, the commits, the logs … everything.
- Remove this directory, and your version-control history is gone!

Beware!

- The .git directory contains a wealth of information
- You normally don’t need to enter into it
    - But we will do so today!
- Remember: Modifying or removing things in this directory can have catastrophic consequences!

Inside .git

- You normally don't need to look at or modify the .git directory — but there is one file you'll likely need to look at, for debugging or general work
- .git/config contains all of your Git repository configuration
- It's a regular text file, which you can edit using any text editor (e.g., Emacs or vim)

Default .git/config

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
    ignorecase = true
    precomposeunicode = true
```

Notice

- Format is from Microsoft INI files, with [sections] and then name=value
- Don't worry about the contents now, but notice that we only have a single section, [core]
- You can edit this file manually — but you probably don't want to do so. Rather, try to use Git's various commands to manipulate it.

OK, now what?

- Commit multiple versions of a file to the repository
- Look at the changes that were made to that file over time

Config changes

- git config abc.def ghi # adds
- git config abc.def zzz # updates
- git config --unset abc.def # removes key-value
- git config --remove-section abc # removes section

First, set your config

- Before you continue, you need to tell Git your name and e-mail address

- These are put in the commit logs; without them, you'll get warnings, and the logs will look funny:

```
$ git config --global user.name "Reuven Lerner"

$ git config --global user.email reuven@lerner.co.il
```

Where does this go?

- Each Git repository has a .git/config file
- Each user has a ~/.gitconfig file
- When you make "global" setting changes, you're updating your ~/.gitconfig file
- If you work on more than one computer, you'll probably want to copy this file around.

In other words:

- The following sets [foo] bar = abc in the current Git repository:

  ```
  $ git config foo.bar abc
  ```

- The following sets [foo] bar = abc in your personal ~/.gitconfig

  ```
  $ git config --global foo.bar abc
  ```

- Your personal ~/.gitconfig sets default values; a particular repository's configuration has priority

See current config

```
# Show all config

git config --list

# Show only local config

git config --local --list

# Show only global config

git config --global --list
```

Get config of one pair

```
# Retrieve the value of a section-name

git config user.name
```

The story so far

- Any directory can be turned into a Git repository (and thus managed by Git) with "git init", which creates a ".git" directory.
- The ".git/config" file is the configuration for that repository.
- You can change the repository-specific configuration with "git config"
- You also have a ".gitconfig" file in your home directory, which applies to all of your repositories
- Modify global .gitconfig with "`git config --global`"