Giting Git

Andrew Denner
Central Iowa Linux Users Group

January 2022

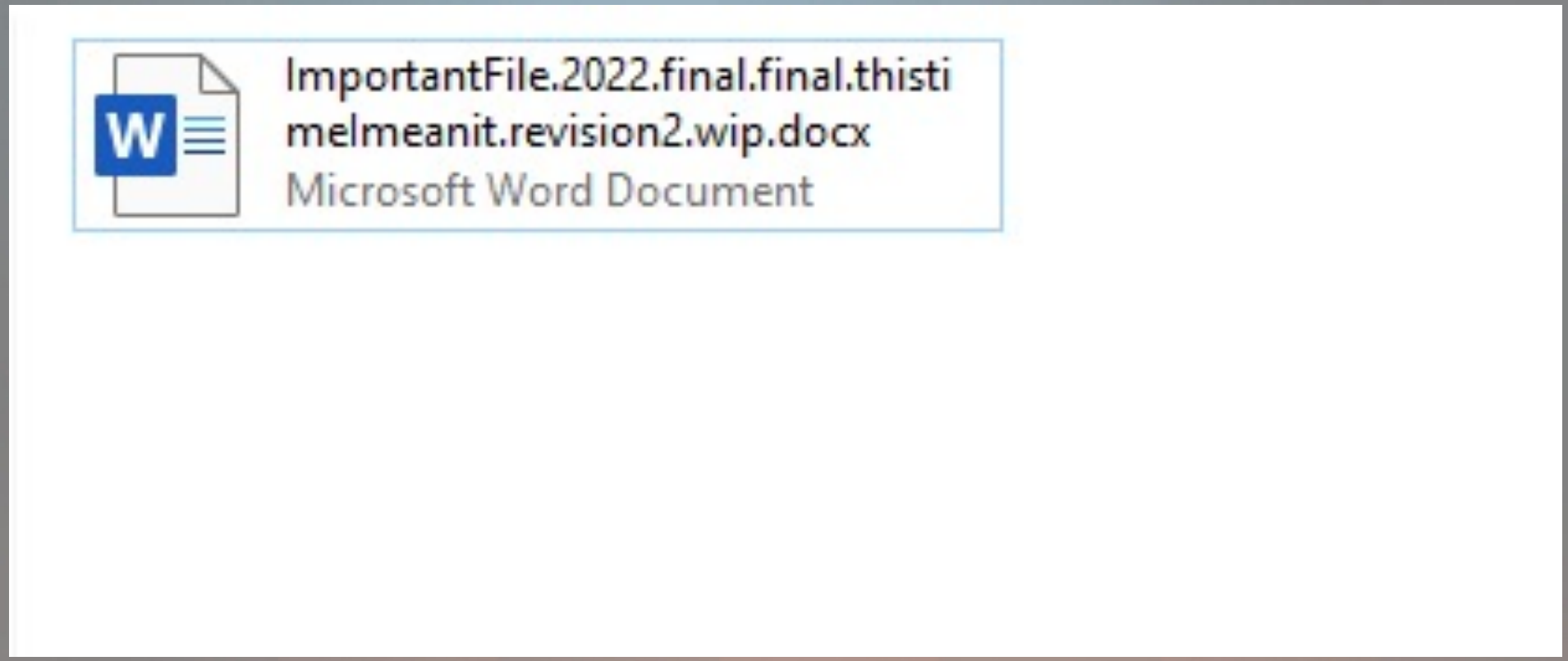
About me

Senior Scientific Software Developer

Linux aficionado

Slides posted to https://denner.co after

In the beginning…



There is a better way

In the beginning

- Source Code Control System (SCCS)—1972 Bell labs
- RCS—Networked version control shortly after
- CVS—1984 (last updated 1998)
- Subversion—founded 2000 by CollabNet
  - Directories are versioned
  - Atomic Commits
  - Branching and merges
  - File locking
  - Binary Diffs
  - Interactive merges
  - Still a central server though

Git a brief history

Git a brief history

- 2005 by Linus
- Goals
  - Speed
  - Data integrity
  - Support for distributed non-linear workflows
- Why? (From the Readme file)
  - Random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant.
  - Stupid. Contemptible and despicable. Simple. Take your pick from the dictionary of slang.
  - "Global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room.
  - "Goddamn idiotic truckload of sh*t": when it breaks.

NAME

git - the stupid content tracker

SYNOPSIS

```
git [--version] [--help] [-C <path>] [-c <name>=<value>]
    [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
    [-p|--paginate|-P|--no-pager] [--no-replace-objects] [--bare]
    [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
    [--super-prefix=<path>]
    <command> [<args>]
```

DESCRIPTION

Git is a fast, scalable, distributed revision control system with an unusually rich command set that provides both high-level operations and full access to internals.

See **gittutorial**(7) to get started, then see **giteveryday**(7) for a useful minimum set of commands. The **Git User's Manual**[1] has a more in-depth introduction.

After you mastered the basic concepts, you can come back to this page to learn what commands Git offers. You can learn more about individual Git commands with "git help command". **gitcli**(7) manual page gives you an overview of the command-line command syntax.

A formatted and hyperlinked copy of the latest Git documentation can be viewed at **https://git.github.io/htmldocs/git.html** or **https://git-scm.com/docs**.

OPTIONS

--version
: Prints the Git suite version that the git program came from.

--help
: Prints the synopsis and a list of the most commonly used commands. If the option **--all** or **-a** is given then all available commands are printed. If a Git command is named this option will bring up the manual page for that command.

  Other options are available to control how the manual page is displayed. See **git-help**(1) for more information, because **git --help ...** is converted internally into **git help ...**.

-C <path>
: Run as if git was started in <path> instead of the current working directory. When multiple **-C** options are given, each subsequent non-absolute **-C <path>** is interpreted relative to the preceding **-C <path>**. If <path> is present but empty, e.g. **-C ""**, then the current working directory is left unchanged.

  This option affects options that expect path name like **--git-dir** and **--work-tree** in that their interpretations of the path names would be made relative to the working directory caused by the **-C** option. For example the following invocations are equivalent:

  ```
  git --git-dir=a.git --work-tree=b -C c status
  ```

Git, the stuff that is missing

- Centralized servers have added on features (Azure Devops, Gitlab, Github, et.al.
  - Protected branches (restricted pushes)
  - Pull requests
  - Backlogs/Scrum or Kanban boards
  - Gated Check-ins
  - CI/CD
  - Secrets
  - Security Scans
  - User management

Git the good, the bad, the ugly

- Good: Standalone, you can make changes offline
- Good: Keeps track only of the deltas between versions of a file (really good for txt)
- Good: You have the whole repository right there
- Bad: What is the delta of a binary object like jpeg, executable, or movie?
- Bad: You have the whole repository right there (lots of baggage)

Corey Quinn
@QuinnyPig

If I do a `git pull` on @elastic's Logstash repository, it pulls down 113.16 MiB from GitHub.

Fully 66MB of that is a binary WAR file erroneously committed to git in 2013.

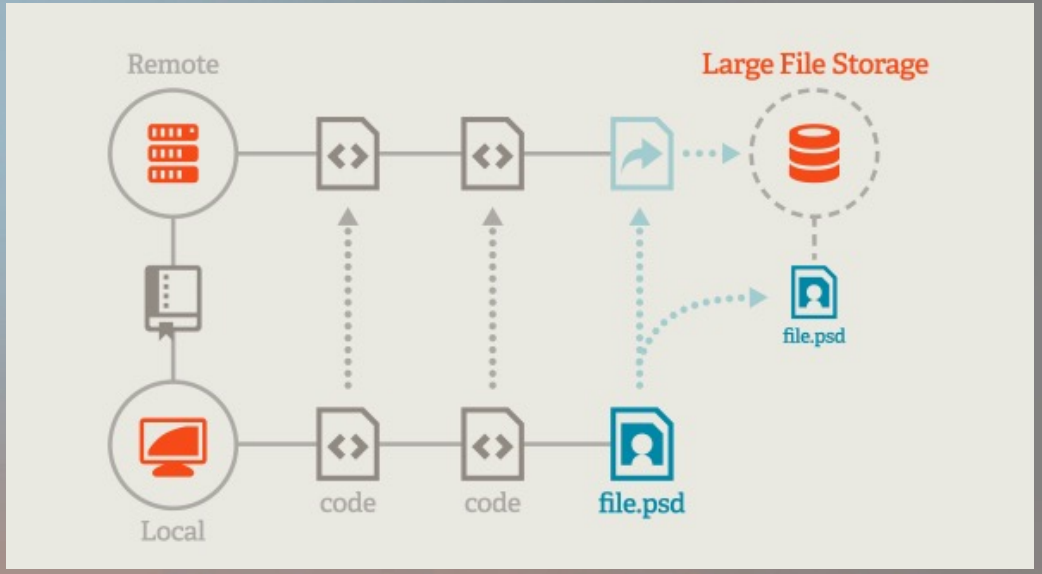
That's right. Over *half* of every transfer of that wildly popular repo is a file some moron messed up.

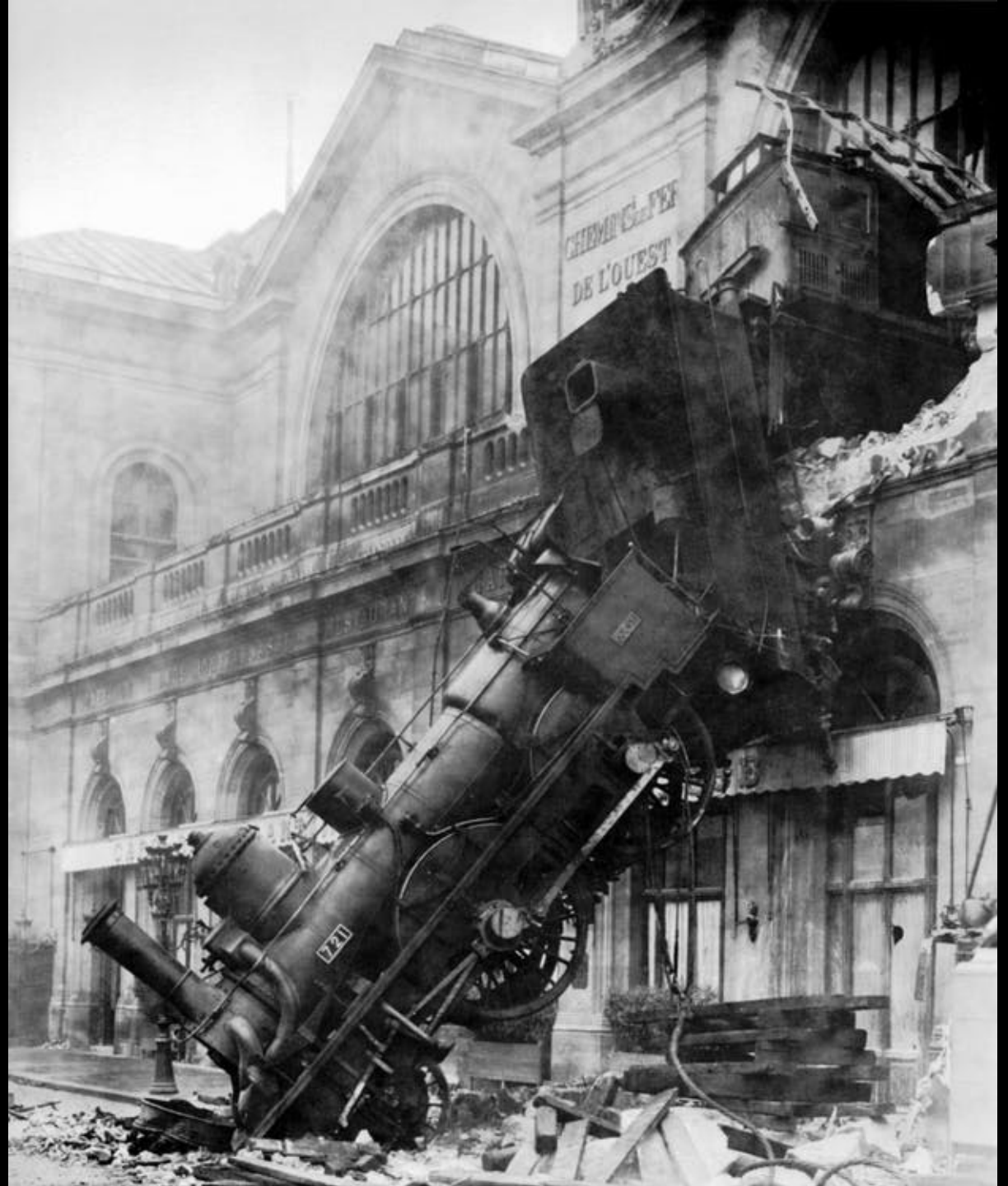
2:58 PM · Aug 19, 2021 · Twitter Web App

81 Retweets **14** Quote Tweets **772** Likes

Git lfs

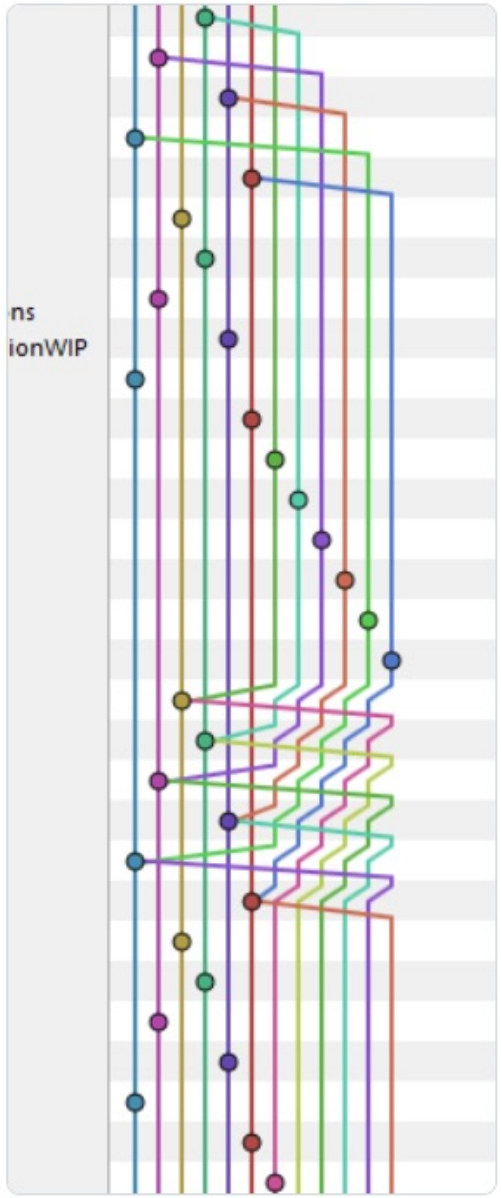
- One way around this is an extension Git large file storage. (really only helpful when using centralized server like github)
- Stores a pointer to a https webserver (latest versions also allow ssh) where large file is stored
- https://docs.gitlab.com/ee/topics/git/lfs/
- https://git-lfs.github.com/




CHEMINS DE FER
DE L'OUEST

Huenry Hueffman
@HenryHoffman

I fucked up Git so bad it turned into Guitar Hero



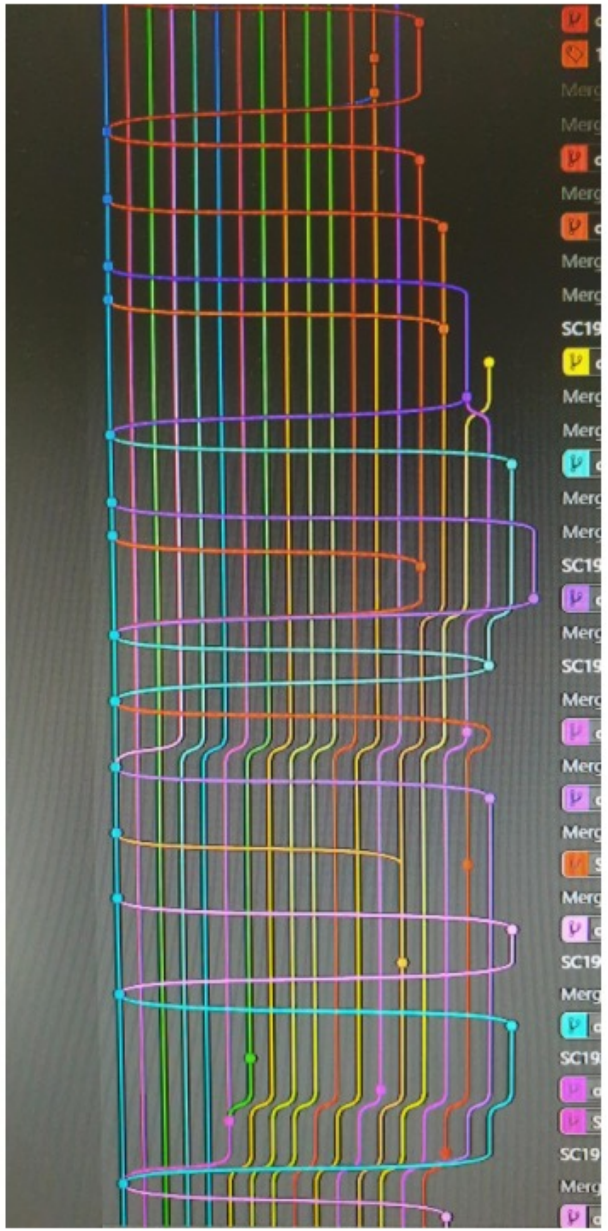
9:42 AM · Feb 1, 2016 · Twitter Web Client

13.6K Retweets 250 Quote Tweets 18.1K Likes

++
3.7k
--

Posted by u/Shabib_KBS 2 years ago

Fuck the Repo so hard it becomes a guitar hero level



190 Comments Share Save Hide Report 98% Upvoted


DON'T
PANIC

Hi there! You can get this same content without swears at dangitgit.com

https://ohshitgit.com/

Oh Shit, Git!?!

Git is hard: screwing up is easy, and figuring out how to fix your mistakes is fucking impossible. Git documentation has this chicken and egg problem where you can't search for how to get yourself out of a mess, *unless you already know the name of the thing you need to know about* in order to fix your problem.

So here are some bad situations I've gotten myself into, and how I eventually got myself out of them *in plain english*.

Oh shit, I did something terribly wrong, please tell me git has a magic time machine!?!

Demo Time