

Git for sysadmins

Brad Beyenhof

Git for sysadmins

-or-

I was an SCM engineer for a year
and lived to tell about it

Brad Beyenhof

I was placed in a new SCM group, not of my own accord

Git is an opaque system with a steep learning curve
But it's the "new hotness" everybody needs to know
I bet that's why a lot of you are here
Most of you probably want to see a slide that says...



This talk will
teach you to use git

You know you need version control, and you want to use git, so you want to know how to use it. Unfortunately, this is kind of a lie, because...



Git tutorials suck

Don Marti at ScaLE 12x

Anecdote with Chris a few years ago

I've given git tutorials

- to developers
- and cleaned up later

Commands don't actually tell you what you're doing
- cd,ls,mv,rm without file/directory structure?

You need to just do it... but I'll share vocabulary & concepts to help

If you know what you want, you can look up cmds

Also, every talk on git I've given has lots of Q&A

- Feel free to ask questions as we go along

Why use git, then?

- Track changes to any directory
- Journal of your work - documentation-ish
- No central server required
- Collaboration features are awesome

You might not see a need to collaborate now, but learning it on your own means you'll be able to use it to work with others

I'll be focusing here on:

- Configuration basics
 - Frontend basics
 - Backend basics

Install & configure git

From your local package repo (Debian default, CentOS rpmforge) or...

Install & configure git
<http://git-scm.com>

OS X has it built in, but older. Homebrew is up-to-date

Version 1.9 came out just a couple weeks ago, on VALENTINE'S DAY!

Install & configure git
<http://git-scm.com>
Add name & email

Either in a GUI git client, or at the command line:

Install & configure git

<http://git-scm.com>

Add name & email

```
git config --global user.name "Your Name"  
git config --global user.email you@foo.com
```

The `--global` will change your configuration on this machine (in `~/.gitconfig`), and will apply to all git repos you use here

Also, if you're at the command line, you'll want CLI colors:

Install & configure git

<http://git-scm.com>

Set CLI colors

```
git config --global user.name "Your Name"  
git config --global user.email you@foo.com  
git config --global color.ui auto
```

And that's it. We're ready to start our first git repo.

Git frontend basics

I'm going to foabout the same stuffnerally have cus
on CLI git commands; GUIs generally include just
about the same stuff.

Git frontend basics

workdir

Git frontend basics

workdir

`'git init'`

Git frontend basics

local repo – workdir

Local repo is stored in a '.git' directory here in the workdir

Git frontend basics

local repo – workdir

```
`git add $PATH`
```

\$PATH can be . If you want to add the whole thing

One other part you need to be aware of, though, is the index, or staging area.

Git frontend basics

```
repo — index ← workdir
```

Add files (or hunks) to the index, so you can craft your commit to only include specific things.

Git frontend basics

repo — index ← workdir

`'git commit'`

Git frontend basics

repo ← index ← workdir

Git frontend basics

repo ← index ← workdir

Lather, rinse, repeat.

Git frontend basics

repo ← index ← workdir

Lather, rinse, repeat.
Change, add, commit.

Git frontend basics

repo ← index ← workdir

Lather, rinse, repeat.
Change, add, commit.
Commit OFTEN.

This is the journaling part I mentioned earlier.

This is way better than commenting out code in order to write something different.

If you had a build/test environment, each commit would be a small, self-contained change. It may include changes to multiple files, but it should all still work.

Now, the backend...

Git frontend basics

DEMO

status

log

log --oneline

Git backend basics

Git has four types of objects:

Git backend basics

blob
tree
commit
tag

Git backend basics

```
      /b1  
t1-<  
      \b2
```

Git backend basics

```
          /b1  
c1-t1-<  
          \b2
```

Git backend basics

```
          /b1  
c1-t1-<  
          \b2  
          /  
t2-<  
          \b1'
```

Git backend basics

```
          /b1  
c1-t1-<  
          \b2  
          /  
c2-t2-<  
          \b1'
```

Git backend basics

```
                /b1
c1-t1-<
↑           \b2
|           /
c2-t2-<
                \b1'
```

Git backend basics

Git backend basics

c1

↑

c2

Git backend basics

c1
↑
c2
(master*)

Master is like SVN trunk

Git backend basics

```
c1  
↑  
c2  
(master*)
```

```
`git branch feature1`
```

Git backend basics

c1

↑

c2

(master*, feature1)

Git backend basics

c1

↑

c2

(master*, feature1)

```
`git checkout feature1`
```

Git backend basics

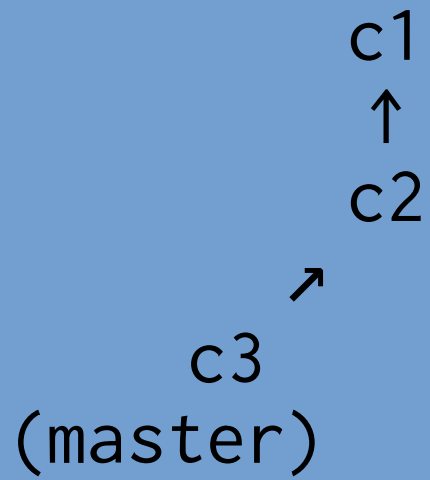
c1

↑

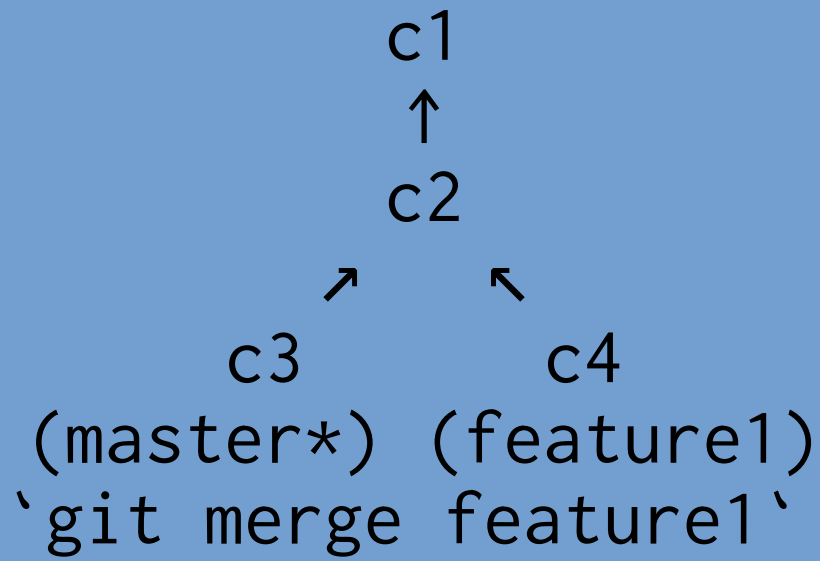
c2

(master, feature1*)

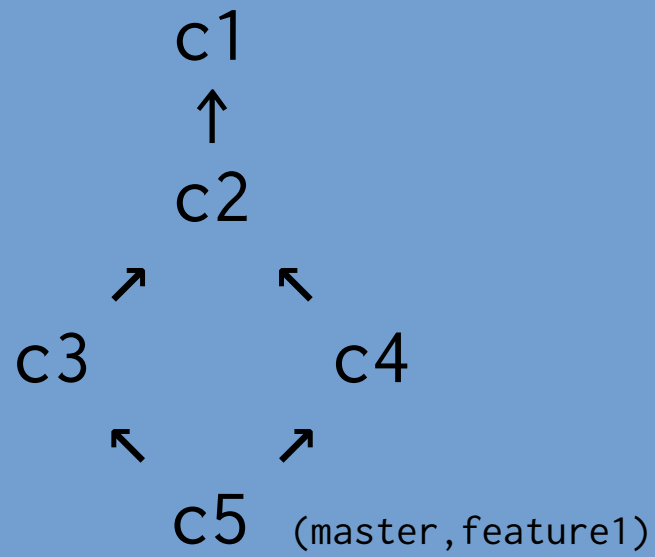
Git backend basics



Git backend basics



Git backend basics



C5 is a merge commit, with two parents

Resources

Pro Git (free)

Resources

Pro Git (free)
Version Control with Git
(O'Reilly)

Resources

Pro Git (free)
Version Control with Git
(O'Reilly)
Git Intro: Randal Schwartz
(Vimeo)

Q&A

- Branching
- Collaboration
- SVN migration