

Rails from the Ground Up!

with Jared Richardson
RoleModel Software

Introductions

Jared Richardson

Jay Hess

Matthew Dodds

Dave Woodall

Our Goal

Practical knowledge

Familiarity

Lots of typing

Follow along (for most slides)

Know how to learn

Outline

Foundations: Tools

Ruby: The basics

Ruby: With objects

Ruby: Network programming

Sinatra: Easier network programming

Rails: Introduction and overview

Rails: Deeper: writing our own app

Finally: Class Driven Details!

A Tool that Builds Tools

brew

Downloads source

Compiles & Installs

Scripts are in Ruby

Install Brew

Search “osx brew”

<http://mxcl.github.com/homebrew>

Install git

```
brew install git
```

Git

Distributed Version Control

New paradigm

Everyone has the repo

Git History

Created by Linus Torvalds

2005

For the linux kernel

Open source

Design Goals

Not CVS/SVN :)

Distributed workflow

Safe (e.g. corruption)

High Performance

Why Git (for us)?

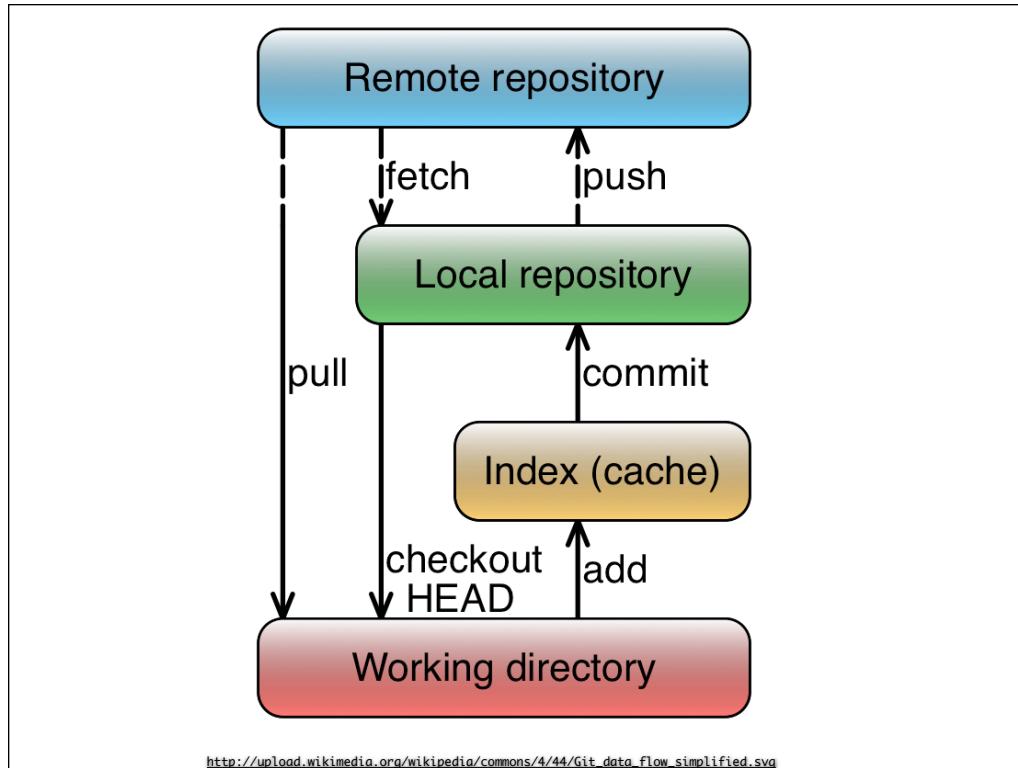
Community standard for Ruby/Rails

Merging

Speed

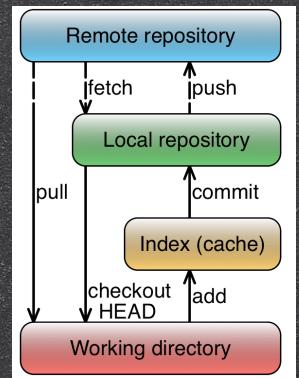
Portable

<https://github.com/>



Commands

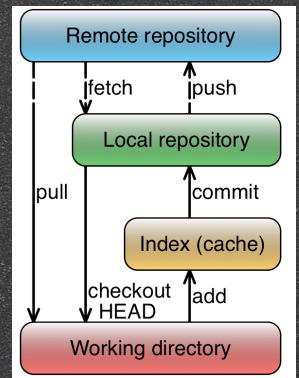
```
git pull <url>  
git add <file or directory>  
git commit -m <message>  
git push  
git checkout <filename>
```



Create a Local Repo

```
mkdir rails_class  
cd rails_class  
git init  
touch README
```

```
Initialized empty Git repository in ~/workspaces/ruby_class/.git/
```

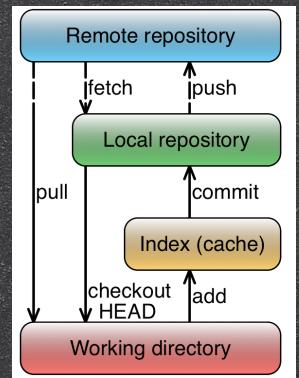


Add Files

`git add README`

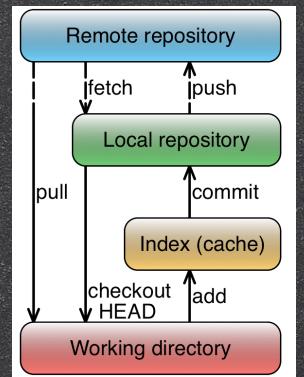
OR

`git add .`



Create a Remote Repo

```
git remote add <scp creds>  
git push
```



Client Configuration

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

More Help

```
git help  
git help <command>
```

Check Out Code

```
git clone git://github.com/  
jaredrichardson/RailsFromTheGroundUp.git
```

What Did You Get?

```
git log
```

```
git log <file_name>
```

Update Your Config

Edit first two lines

Move to `~/.gitconfig`

Now...

return to ruby_class_one

git log

git l

Add a File

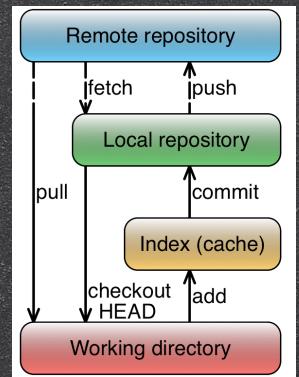
Create new_file_<initials>

git add <file>

git commit -m “<message>”

git push

But wait...

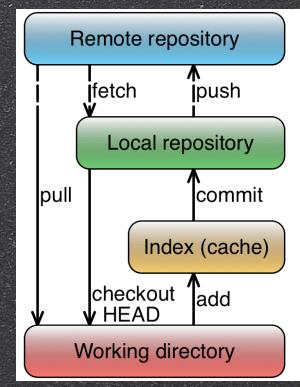


You Don't Have Write Access to the Repo!

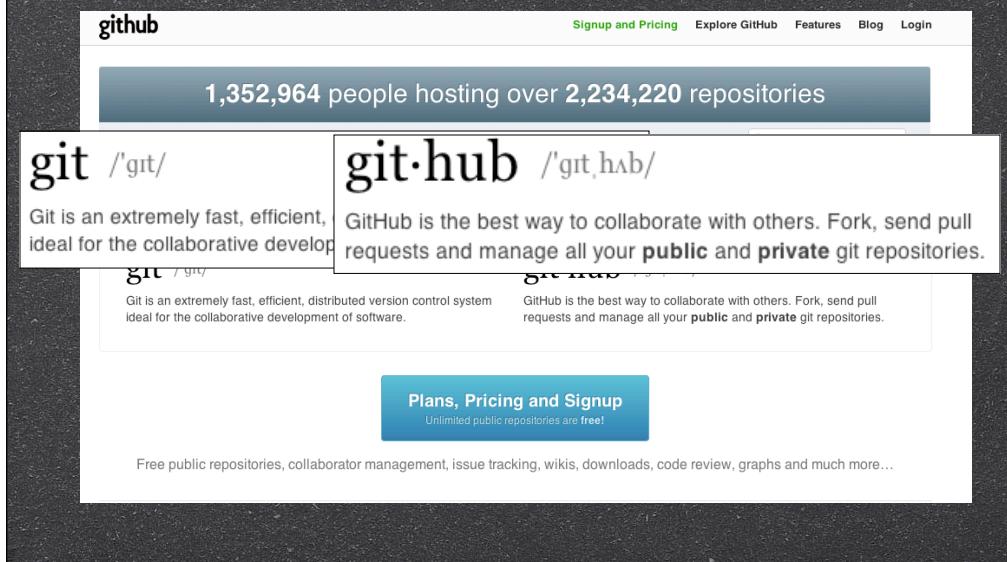
What step would fail?

push

You need Github accounts



github.com



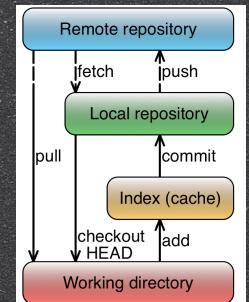
Clone a Repo

```
mkdir class
```

```
cd class
```

```
git clone
```

<https://github.com/jaredrichardson/RailsFromTheGroundUp>



Github

- 1.Create free github accounts
- 2.Create public/private keys
- 3.Put your public key in github
- 4.Clone the class repository

Sign up for GitHub

[Log in to an existing account](#)

\$0/mo

You are signing up for the **free** plan

The cost for this plan is **\$0 per month**. You can cancel, downgrade, or upgrade at any time.

Create your free personal account

Username

Email Address

We promise we won't share your email with anyone.

Password

Must contain one lowercase letter, one number, and be at least 7 characters long

Confirm Password

By clicking on "Create an account" below, you are agreeing to the
[Terms of Service](#) and the [Privacy Policy](#).

[Create an account](#)

facebook

twitter

mozilla

37signals

You're joining the smartest companies in the world

✓ Email support

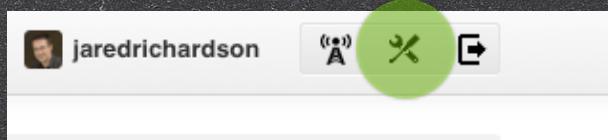
✓ Upgrade, downgrade or cancel at any time

✓ Secure, reliable, always-available repository hosting

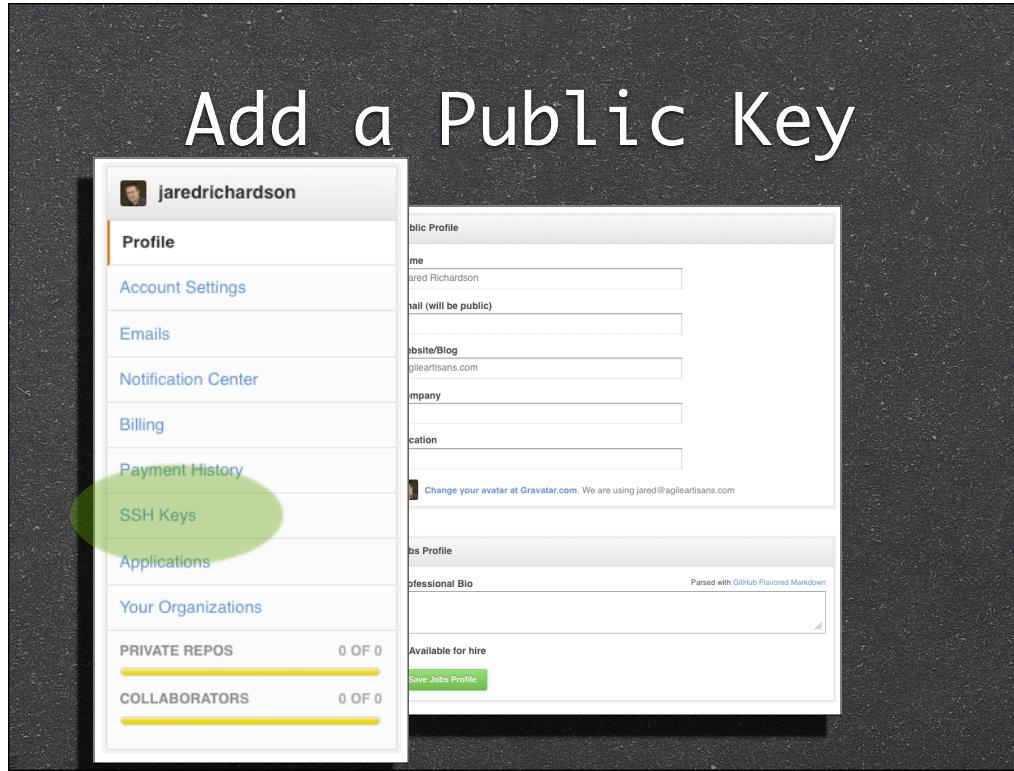
Add a Public Key

Top right hand corner

Click account settings



Add a Public Key



Creating an SSH Key

First, CHECK for an existing key!!

Open Terminal (on your Mac)

```
$ ls ~/.ssh
```

```
$ ssh-keygen -t rsa
```

```
$ ls ~/.ssh
```

```
$ chmod 700 ~/.ssh
```

```
$ chmod 600 ~/.ssh/id_rsa
```

Public/Private Key Pair

Private key => id_rsa

Public key => id_rsa.pub

Never share the private!

Share the public anywhere

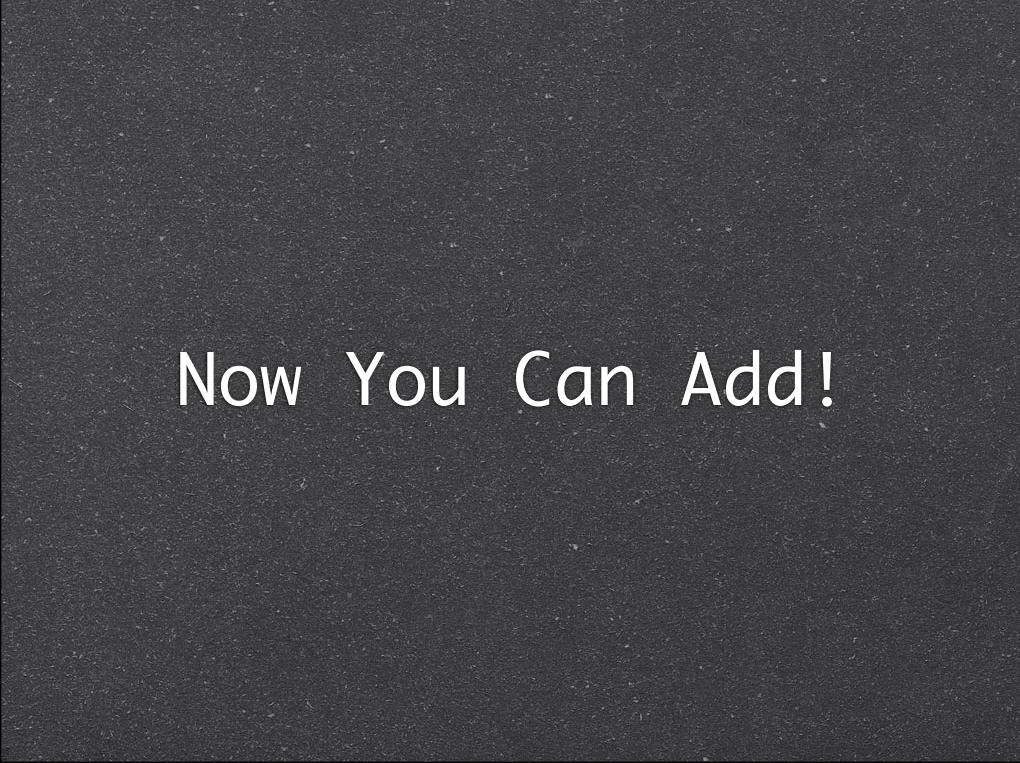
```
$ cat ~/.ssh/id_rsa.pub
```

Back to Github

Click SSH Keys

Click Add New SSH Key

Paste in your PUBLIC key



Now You Can Add!

Add a File

In student_files

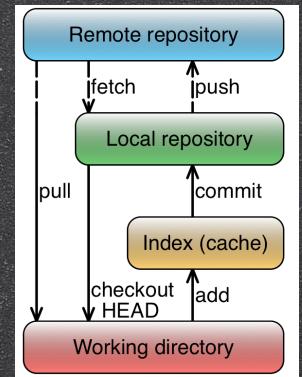
Create new_file_<initials>

Add text or copy mine

git add <file>

git commit -m "<message>"

git push



After Lunch...

Fix Matt's .gitconfig

Then start editing again!

Done?

```
git pull  
What do you see?
```

Log again

git l

blame

```
git blame <file>  
git blame -L <line>, +10 <file>
```

Edit Like Crazy!

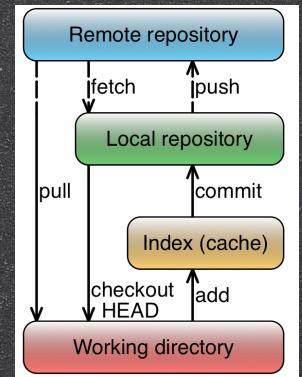
Everyone edit every file

git commit -m "message"

git push

git pull

git l



Collisions?

You need a stash!

Put your code aside

Pull in changes

Get your code off the stash



<http://www.flickr.com/photos/mogus/57114273>

stash

```
git stash save [label]  
... work... test...  
git pull  
git stash list  
git stash apply [label]  
OR git stash pop  
git stash drop [label] # or clear
```

Your Turn

```
git stash "this is my collision"  
git pull  
git stash pop  
git commit -m "Now I can commit my  
changes!"  
git stash clear
```

Bisect

```
git bisect start
```

```
<run your tests>
```

```
git bisect good
```

```
<run your tests>
```

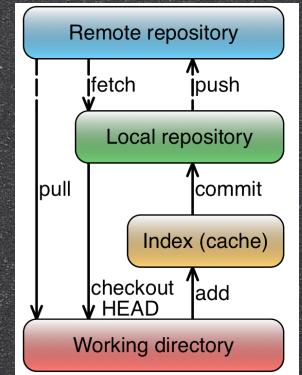
```
git bisect bad
```

```
....
```

```
git bisect reset
```

Git

Different
More granular
De facto standard



More git Links

<http://git-scm.com/>

<https://github.com/>

<http://devsundar.github.com/2012/02/09/Uses-of-git/>

<http://git-scm.com/>

Cloning and Creating a Patch

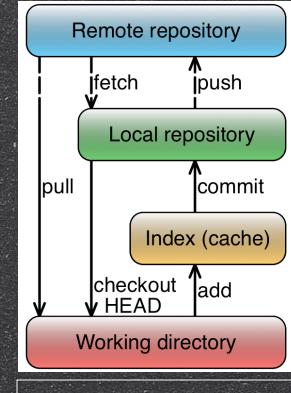
```
$ git clone git://github.com/git/hello-world.git
$ cd hello-world
$ (edit files)
$ git add (files)
$ git commit -m 'Explain what I changed'
$ git format-patch origin/master
```

Creating and Commiting

```
$ cd (project-directory)
$ git init
$ (add some files)
$ git add .
$ git commit -m 'Initial commit'
```

Git Cheat Sheet

<u>General Usage</u>	<u>New Repository</u>	<u>git bisect</u>
<code>git status</code>	<code>git init</code>	<code>git bisect start</code>
<code>git log</code>	<code>git push <url></code>	<code>git bisect good</code>
<code>git add</code>		<code>git bisect bad</code>
<code>git commit -m</code>		<code>git bisect reset</code>
<code>git push</code>		
<u>Download Changes</u>	<u>Copy a Repository</u>	
<code>git pull</code>	<code>git clone <url></code>	
<u>Revert a File</u>	<u>Stash and Retrieve</u>	
<code>git checkout <url></code>	<code>git stash <label></code>	
	<code><work and commit></code>	
	<code>git stash apply</code>	
		<code>git blame <file></code>



Next...

Ruby