

RailsConf 2015

The Good Parts Edition

- * I'm going to cover the sessions I liked, and skip the sessions I didn't get much out of
- * All of the presentation are on Confreaks.tv and YouTube (RailsConf 2015)

Day One



Keynote

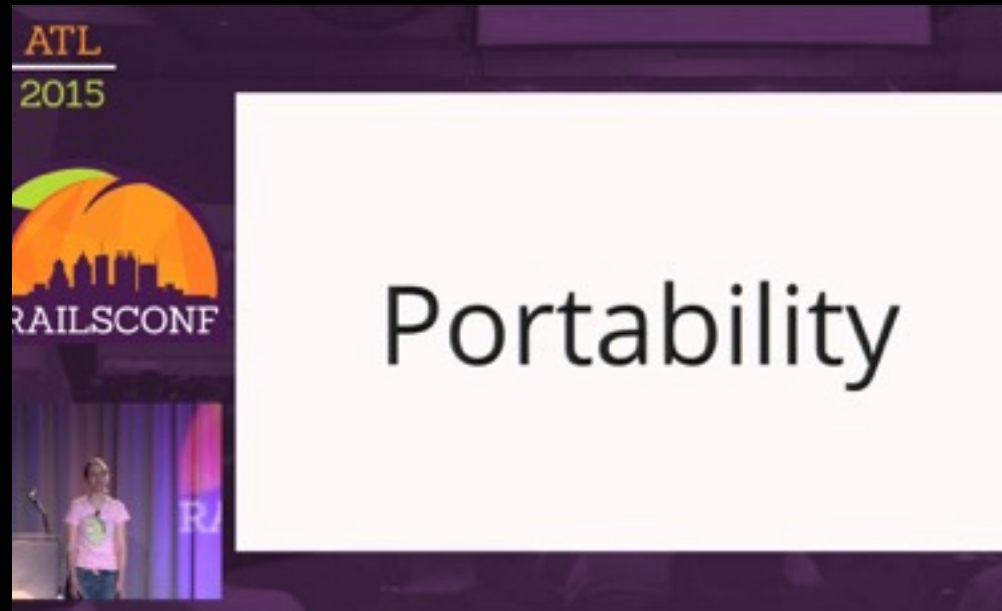
DHH has a backpack

- * Rails is designed for small teams to build cool things. It hasn't been designed to scale to large products or large teams, so YMMV.
- * Turbolinks 3 is coming (it has a built-in progress bar!)
- * ActionCable is fancy websocket framework thingy incubating at Basecamp. It looks interesting, but I reserve judgment until actual code is released.



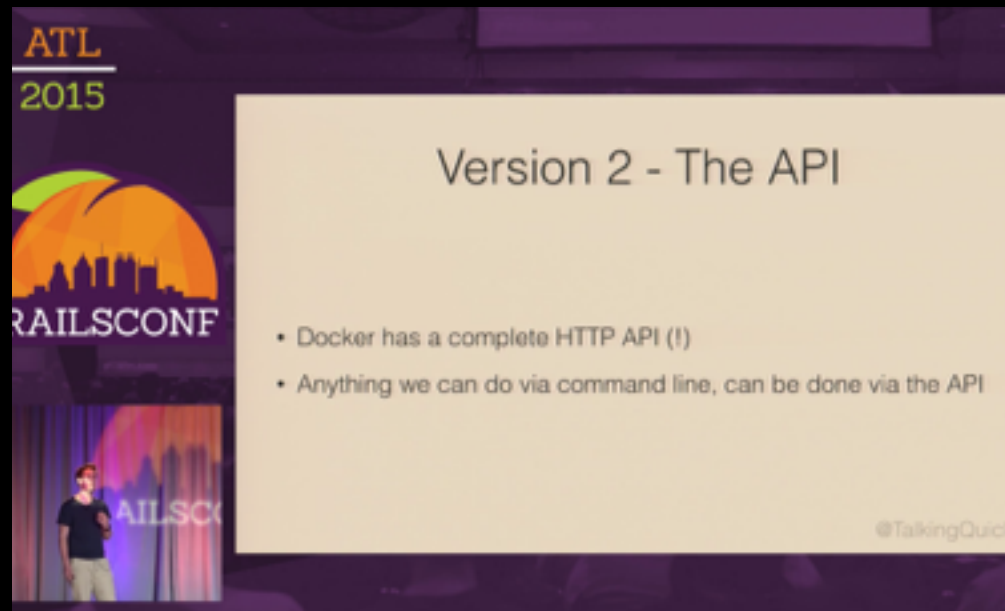
Speed Science

- * Diagnosing performance issues
- * Mainly: use the scientific method



DevOps for the Lazy

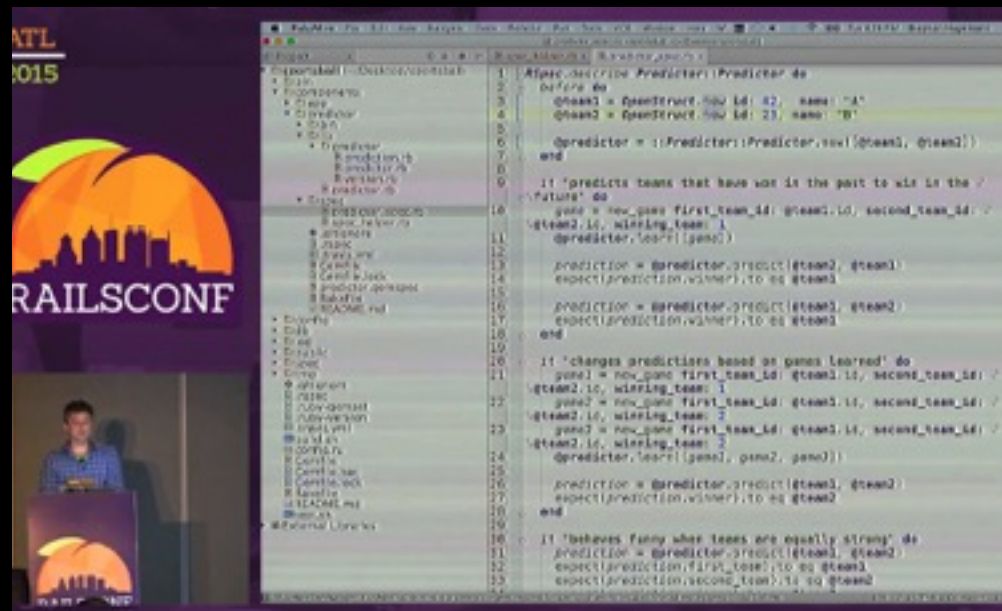
- * Docker is cool (knew that already, but whatever)
- * Kubernetes is an orchestration framework from Google for coordinating multiple docker images running on multiple hosts. Looks cool.
- * You can get a Linux VM from Google (it's not just App Engines! Who knew!?)
- * boot2docker is easy to setup



Docker...

...isn't just for deployment

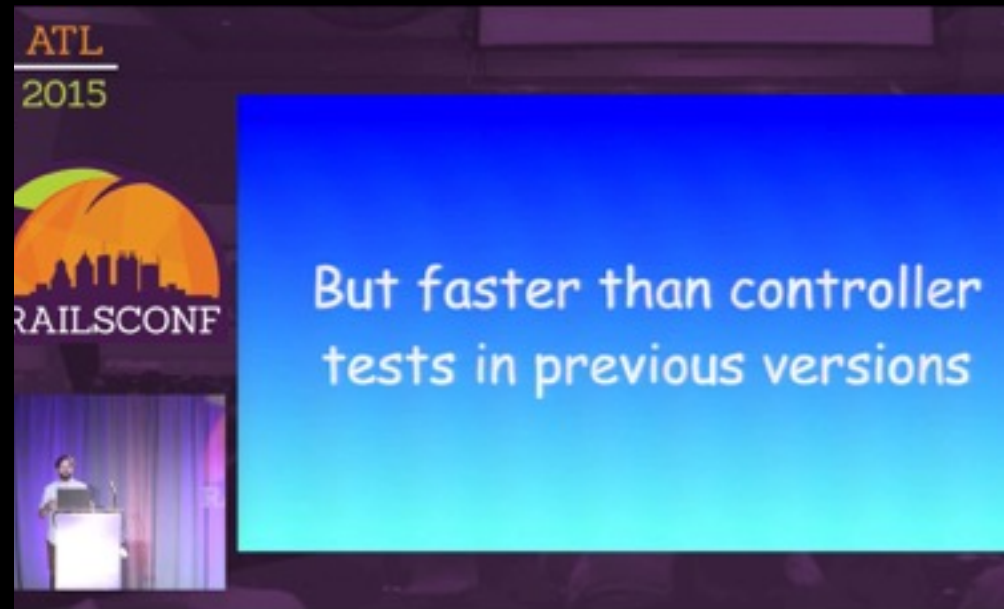
- * His talk was short. My summary is shorter.
- * ~250% better utilization with Docker over dedicated VMs
- * There are some nice Ruby gems for controlling docker
- * Don't bother watching the video



Component-based Rails

- * The idea is to break your app apart into gems and engines, but keep them all as source within a single application
- * Very interesting concept. It's worth watching.
- * A fair amount of overhead to wire everything together. It really makes me appreciate Rails.

Day Two



Keynote

The Aaron Patterson stand-up hour

- * Hilarious standup routine. Lots of puns. He trolled DHH and Kent Beck.
- * Worth watching for the humor, but the actual "content" was all WIP stuff and not too memorable.



Crossing the Canyon of Cognizance

A Shared Adventure

- * The talk used a metaphor of a unicorn trying to climb a mountain to describe the journey of learning a new technology
- * It covers what it's like to learn something new, the stages you go through, how to get help and how to *give* help once you know what you're doing
- * The more I think about this talk, the better it gets for me
- * It's also a great example of a good conference talk. It was well-suited to the medium and not something I would have spent time on otherwise.



Lightning Talks

- * Five minute talks about random topics, presented by whoever signed up first
- * I skipped these and regret doing so. I did enjoy watching them later on YouTube.

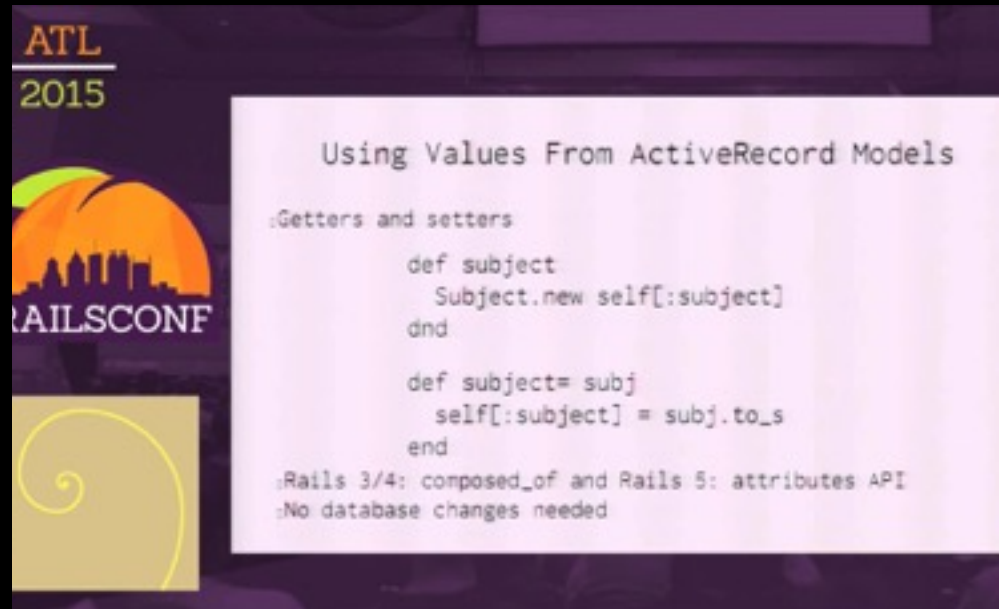
Day Three



Keynote

Mostly Meh

* Jeremy Evans (from Sacramento, maintains the sequel gem) got a Ruby Hero award. The rest was impossible to hear.



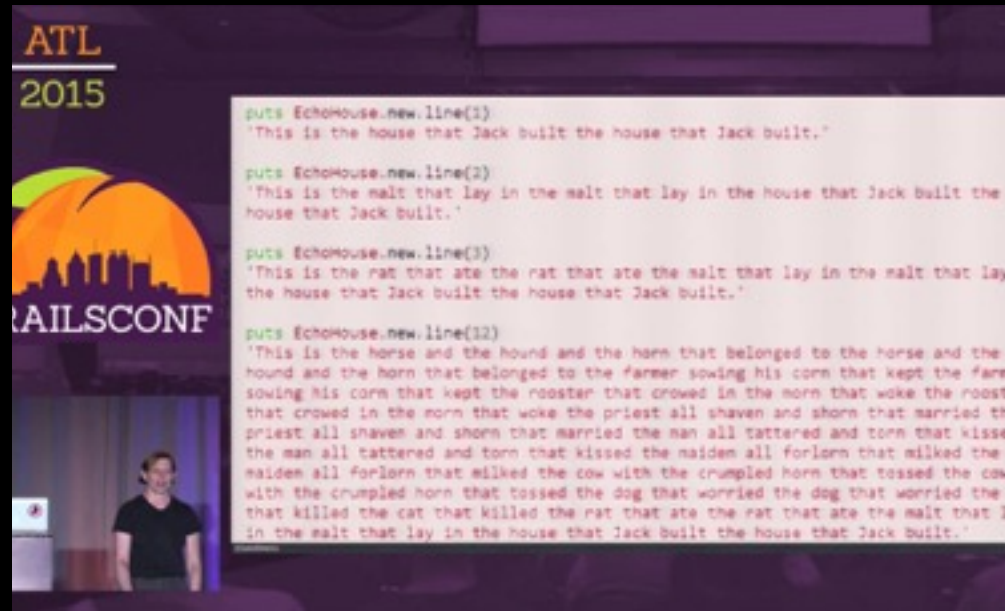
What Comes After MVC

- * The best technical talk at the conference for me
- * He walked through breaking a large model into:
 - * Value object, Entities, Adapters and Shells
- * A way to clean up a large codebase without sacrificing maintainability



Deploy and Manage Ruby on Rails Apps on AWS

- * A good lab. The content was very minimal. All the instructions were written down so you could go at your own pace. They had several people circulating to help with issues.
- * Amazon's OpsWorks is the best Chef frontend I've seen. I was very impressed.



Nothing is Something

- * Best talk at the conference. "Mind blowing" you might say.
- * The main thrust is recognizing when there's an invisible thing in your code that you can't see, and how to see it and make it real
- * The first half covered removing `nil` checks from your code by using null objects
- * The second half covered inheritance, when not to use it, and a better way to share functionality



Closing Keynote

Feeling at Ease at Work

- * Kent Beck trolled Aaron Patterson. It was hilarious.
- * Lots of good advice on how to work as a grown-up, and how that can make your professional feel less chaotic



Was it Worth It?

- * Yes
- * It was a vacation for my inner programmer. No job. No family. Just time to think and learn about code and coding.

Tips for Next Year

“Soft” Topics

- * In general, I got more out of the "soft" topic presentations than the highly technical ones
- * I think they're a better fit for the medium (it's hard to code and listen, it's hard to dive really deep in just 30 minutes)
- * Also, I think it takes more confidence as a speaker to present on a topic where you aren't just describing your last project and showing off code



“Intro to [X]”

- * Avoid these presentations unless you know nothing about the topic
- * They have about as much content as a long blog post, and I can read faster than you can talk.

“Doing [Y] with [Z]”

- * Avoid these unless you know you're going to need to do *Y*, and you think *Z* might be a good fit
- * These are very narrowly focused, so it's hard to pick out useful bits if you don't care about the main topic
- * This kind of talk is relatively easy to prepare and give, so let's just say it's the kind of talk I would give if it was my first time presenting at a large convention.



- * RailsConf 2016 is in Kansas City

Go watch some TV

<http://confreaks.tv/events/railsconf2015>

[https://www.youtube.com/results?
search_query=railsconf+2015](https://www.youtube.com/results?search_query=railsconf+2015)

[https://www.youtube.com/watch?
v=OMPfEXIITVE](https://www.youtube.com/watch?v=OMPfEXIITVE)

Seriously, go watch this

