Sometimes Tools Matter

John E. Vincent
DevOpsDays Goteborg 2011

We all know about DevOps

We all know about DevOps

So what's the big deal?

“With XXXXX you are be able to do easily common task on your local or remote machine. The aim of XXXXX is to become the unique and universal tool language that permit you to make your own brew, apt-get or yum package, same syntax for all your machine.”

Not Puppet

Not Chef

Not Capistrano

“With XXXXX you are be able to do easily common task on your local or remote machine. The aim of XXXXX is to become the unique and universal tool language that permit you to make your own brew, apt-get or yum package, same syntax for all your machine.”

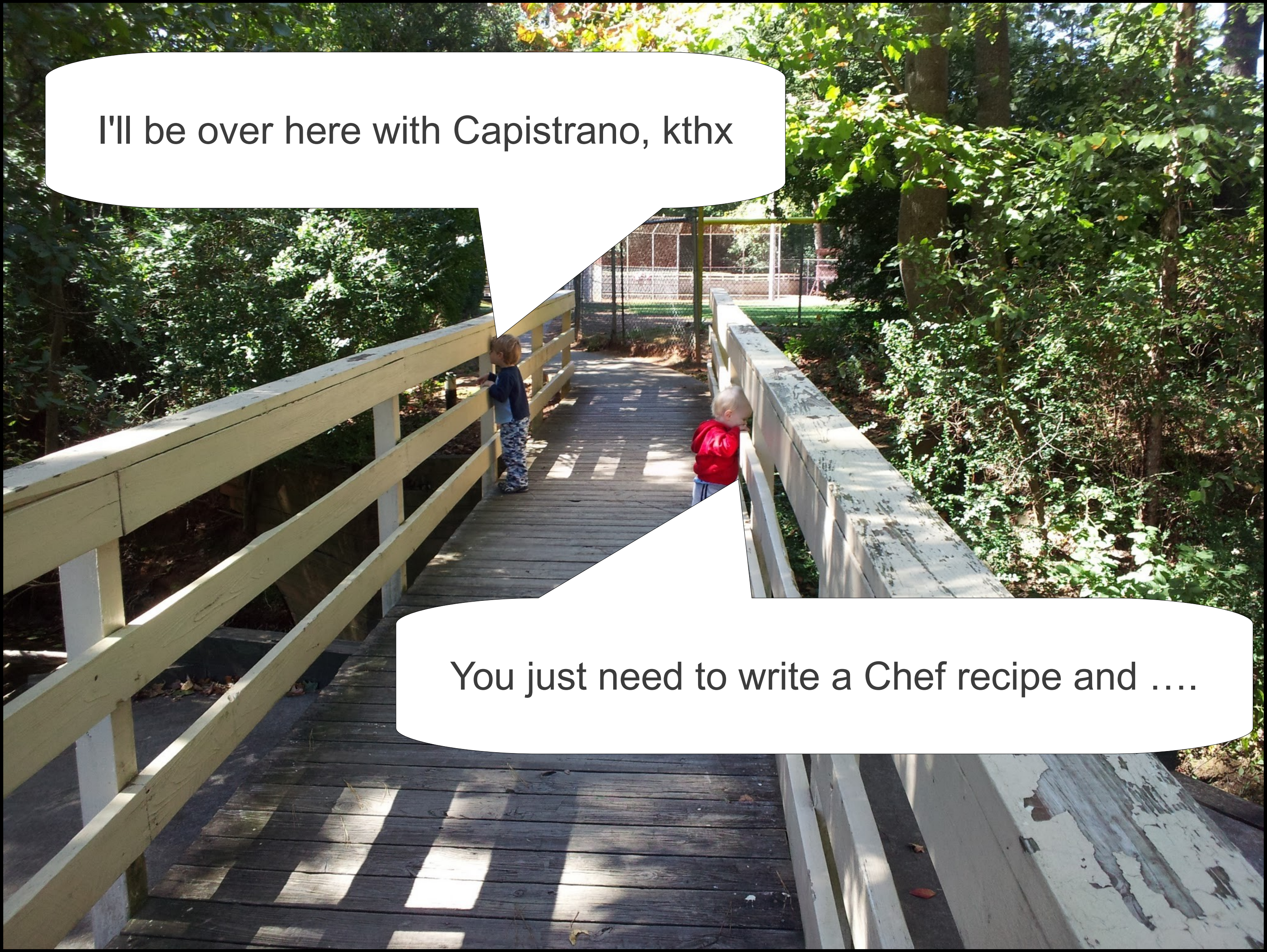
Not Fabric

Not DeployML

Not CFengine

YANFT
(yet another f*ing tool)

We're doing something wrong.
Something is missing.


I'll be over here with Capistrano, kthx
You just need to write a Chef recipe and ….

You can't solve cultural issues with tools

You can't solve cultural issues with tools
or can you?

Some Issues

- Repository mismatch
- Different languages
- Volatile configuration
- Visibility
- Sensitive information
- Testability
- Packaging

Caveats and Comments

- No single tool is going to solve all your problems – sorry.
- You may have already heard of some of these tools.
- You will likely have to “mold” some of these tools to fit your environment
- The tool itself is not the point, it is the end result
- I don't have all the answers...

Impedance mismatches

Sample Shop

- Operations – Puppet, EC2
- Development – Maven, Spring
- War files
- Spring properties files
- Metric assloads of XML
- Database changes

Issues

- Operations repository isn't application repository
- Developers need to test locally during development
- Properties files are artifacts
- Some settings are environment specific
- Some settings are “sensitive”
- How do new settings get in CM?
- Where do database changes fit?
- What about the rest of the business?

Know your Enemy

Know your Enemy

This sandwich kicked my ass.

Know your Enemy

This sandwich kicked my ass because I didn't know my enemy

The competition isn't between “devops” tools. It's between devops and “shell scripts covered in meat sauce”
(with apologies to @littleidea)

“shell scripts covered in meat sauce” == inertia

Configuration

Configuration Competition

- XML files (unless you run bcfg2)
- Key/Value property files (.ini style files)
- YAML
- JSON
- Hard coded “stuff”
- My-effing-SQL, Mongo-effing-DB

All of these things are “simpler” to understand.

Configuration Champions

- ZooKeeper, Nesoi, Noah
- Moves volatile configuration outside of application
- Can be populated by both operations and development*
- Service discovery
- No immediate need to learn a new DSL/Language
- Can still control access*
- Don't underestimate the power of environment variables
- You still need “proper” configuration management

Packaging

Packaging Protagonists

- War files
- Lein, Maven/Ivy, Rubygems, Agner, CPAN, Pypi
- RVM
- Homebrew

Not always simpler to understand but built into the community.

Packaging Princes

- FPM, brew2deb, fpm-cookery
- Makes building OS packages “painless”
- OS packages have value (rollback, versioning, validation)
- Need a single artifact that can be deployed in all environments

Development Environments

Development Defeat

- Windows/OSX for development, Linux/Solaris for production
- QA different than Production
- Different versions of critical libraries, jars, gems whatever
- Exploded war files vs. Packaged war files

“Works on my machine”

Development Dreams

- Vagrant, Veewee, Whirr
- Provides “production” in a box
- Developers should be “deploying” locally
- Operations needs to make modules, manifests, cookbooks (whatever) flexible

This is one of the hardest to accomplish

Visibility

Visibility Villains

- uptime, *top, iostat, vmstat, sar*
- ssh
- tail
- Host Obsession Disorder

Remember who the competition is...

Visibility Victors - part 1

- Statsd, Graphite, Graylog2, Logstash
- ruby-metrics, codahale metrics for JVM
- “If it moves, graph it”
- Disk is “cheap”
- Dashboards are the bomb. Become one with information radiators.
- Logstash can pipe the shit out of EVERYTHING
- You don't know what you don't need until you know what you have
- Overcompensate initially (but don't over do it!)

Visibility Victors - part 2

- Jenkins, Rundeck, MCollective
- Hosts don't matter, only services
- Sometimes you still need to get on a specific host....to reprovision it.
- Wrap the access for auditing and accountability
- DevOps is not about giving root access to developers
- The “myth” of security.

The “other” stuff

Shitty Stuff

- Manual database migrations
- Rollbacks
- The myth of the sensitive

Smart Stuff

- Flyway, Liquibase, Rails migrations
- Consider NoSQL/Schema-less stores
- Roll forward. Never roll back.
- You can never truly roll back anyway
- “Sensitive” data is a myth (for some values of sensitive)

There is, however, ONE tool that can solve almost every technological and cultural problem

Leffe
Abbaye de Abdij van
Leffe
Radieuse
Duvel
GULPENE
10%
GULPENE
GIGANTEN
GLADIA
ROBUUST BIE
GULPENE
GIGANTEN
GLADIAT
ROBUUST BI
10%
alc.vol.
Westma
TRAPPIS
Tripel

Just kidding!
(sort of)

This is the point where you realize I've duped you

You are a tool

You are THE tool

A tale of two tools

This is Kelsey

Kelsey's Problem

- Operations used Puppet with Cobbler as ENC
- Application configuration was in ERB templates managed by Puppet
- Cultural issues ruled out development actually managing the templates
- System of Record was Cobbler

Kelsey's Solution

- Programming motherf**er
- Learned some Java (in a week)
- Wrote some classes, built some jars
- Delivered an XML-RPC library that development could use to query Cobbler for information

This is Jordan

Jordan's Problem

- Everyone has their own effing packaging format
- Distributions suck
- Debs suck to build
- RPMS suck moderately less to build

Jordan's Solution

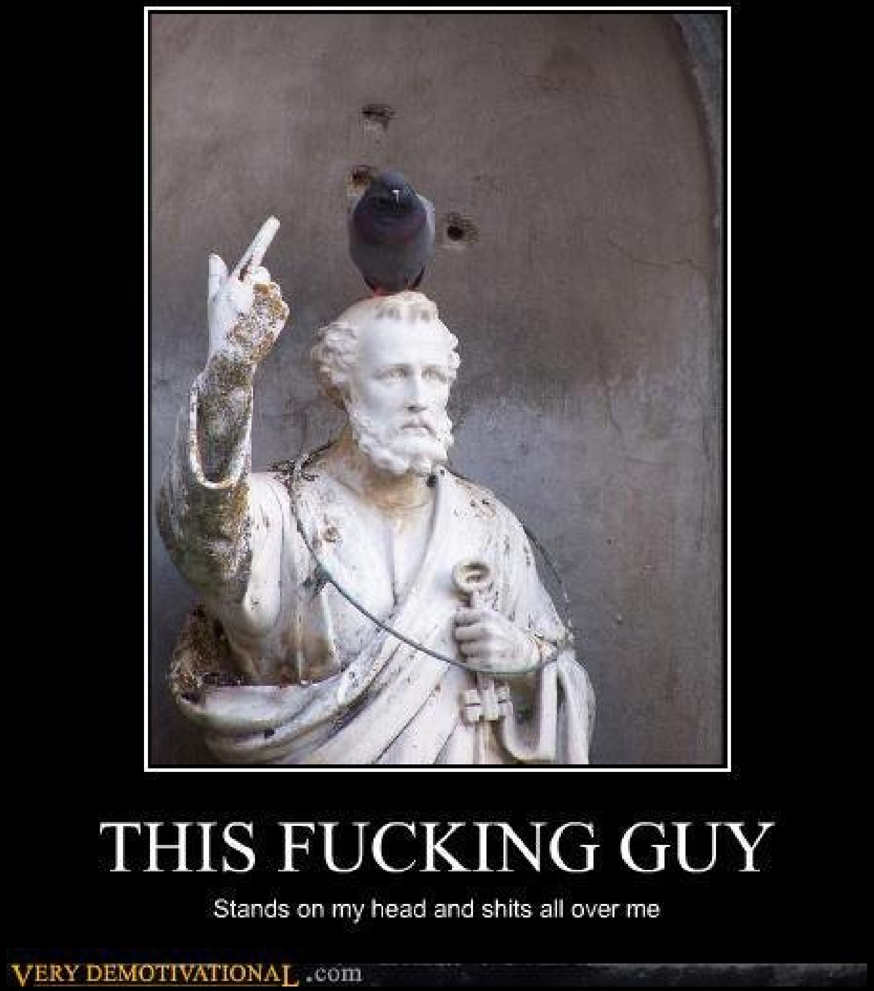
- Hate Driven Development
- Wrote FPM (eFfing package manager)
- Alien done right
- Converts debs, rpms, pypi, gems, compiled directories to native distro package format

Side note: If you ever say “I wonder if someone has done X” stop at the “I” and go to github.com/jordansissel

Real Talk

- Step outside your comfort zone
- Be a technologist not a specialist
- SysAdmins: Learn to write some code. Seriously.
- Developers: Learn the OS. Seriously.
- Everyone: Stop blaming other people
- Software can help ease communication but only you can prevent forest fires.

Go from this....


THIS FUCKING GUY
Stands on my head and shits all over me
VERY DEMOTIVATIONAL .com

and this

to this

Thank You!

Keep in touch!

- @lusis - twitter
- lusis – github
- John Vincent – linkedin
- blog.lusis.org
- john@lusis.org