



OPSCODE

# *Infrastructure Automation with Chef*

*Speaker:*

**Aaron Peterson** Technical Evangelist

Some Content/Mixin Credit to:

**John Willis** VP Services

**Adam Jacob** CTO



OUR SERVERS WERE ABOUT TO CRASH, SO I WROTE A SUITE OF SCRIPTS TO KEEP THEM RUNNING.



Dilbert.com DilbertCartoonist@gmail.com

YOUR ACCOMPLISHMENTS ARE SUSPICIOUSLY HARD TO VERIFY.



4-7-10 ©2010 Scott Adams, Inc./Dist. by UFS, Inc.

SO, RECAPPING WHAT WE KNOW FOR SURE, YOU'RE AN INADEQUATE VERIFIER, AND YOU CAN'T RULE OUT THE POSSIBILITY THAT I'M AWESOME.

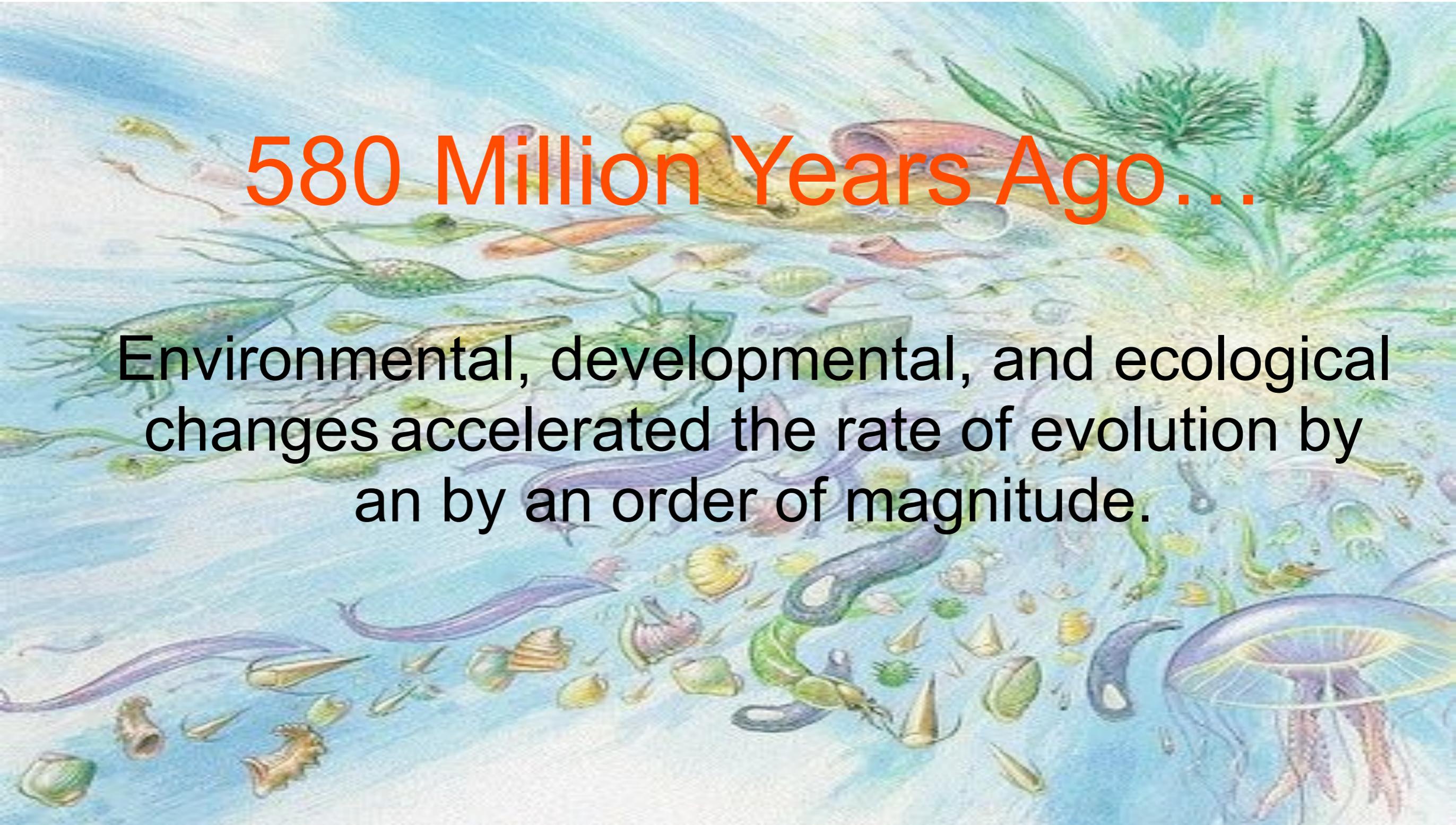




# Cambrian Explosion

580 Million Years Ago...

Environmental, developmental, and ecological changes accelerated the rate of evolution by an order of magnitude.



# Cambrian Explosion

**“The rate of evolution of any system is dependent upon the organization of its subsystems”**

**Herbert Simon, Theory of Hierarchy and componentization.**





## Cambrian Explosion

“A Period of Combinatorial Innovation”

**Hal Varian a Chief Economist for Google**

Recall component parts that innovators could combine or recombine to create new inventions: In the 1800s, it was interchangeable parts. In 1920, it was electronics. In the 1970s, it was integrated circuits.



# Cambrian Explosion

## Today

(John Willis) argues we are at the base of an explosion of innovation driven primarily by four factors:

- Abstract and fault tolerant components
- Unlimited infrastructure
- World wide collaboration
- Changes in intellectual property



Cambrian Explosion

# Abstract and Fault Tolerant Components

- Components are abstract
- Bits, software, virtualized
- Clustered
- Fault tolerant
- Parallelized



## Cambrian Explosion

# Unlimited Infrastructure

- You never run out of them
- So there's no shortage
- There are no inventory delays
- You can reproduce them
- You can duplicate them, you can spread them around the world



## Cambrian Explosion

# World Wide Collaboration

- You can have thousands and tens of thousands of innovators combining or recombining the same component parts to create new innovation.
- It's a situation where the components are available for everyone.



Cambrian Explosion

# Changes in Intellectual Property

- Revolution in how we view intellectual property
- Open source
- Google – I can be an expert on anything in a matter of minutes... worst case one day.



# Infrastructure is Hard

# Fully Automated Infrastructure is Hard!



OPSCODE

Classic infrastructure & operations are major efforts to run.





There has been a recent shift, taking aspects of software development & agile methodology.

Some call this DevOps.



If you think  
agile isn't for  
you...  
you may be  
right!

...been a recent shift, taking aspects of  
development & agile methodology.  
...can this DevOps.



The rise of exceedingly dynamic businesses and infrastructures to support them created the need for DevOps....

But the sea change in tools, and methodologies that have resulted apply to far more than just those young ~~upstarts~~ ~~startups~~ whippersnappers!



“It **is** about the collaborative and communicative culture and the tools and process that arise from that culture.”

-John Allspaw (Etsy)

<http://www.rationalsurvivability.com/blog/?p=1890>



- Cloudcamp.com
  - src
  - JRE System Library [JavaSE-1.6]
- Opscamp.com
  - src
  - JRE System Library [JavaSE-1.6]
  - cookbooks
    - apache2
    - mysql
      - attributes
      - libraries
      - providers
      - recipes
        - client.rb
        - default.rb
        - server\_ec2.rb
        - server.rb
      - resources
      - templates
      - metadata.json
      - metadata.rb
      - README.rdoc

```
client.rb:1
#
# Cookbook Name:: mysql
# Recipe:: client
#
# Copyright 2008-2009, Opscode, Inc.
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#   http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or
# See the License for the specific language governing permissions
# limitations under the License.
#
p = package "mysql-devel" do
  package_name value_for_platform(
    [ "centos", "redhat", "suse" ] => { "default" => "mysql-devel"
    "default" => 'libmysqlclient15-dev'
  }
  action :nothing
end

p.run_action(:install)

package "mysql-client" do
```



[http://www.brooklynstreetart.com/theBlog/wp-content/uploads/2008/12/swedish\\_chef\\_bork-sleeper-cell.jpg](http://www.brooklynstreetart.com/theBlog/wp-content/uploads/2008/12/swedish_chef_bork-sleeper-cell.jpg)

# At a High Level...

- A **library** for configuration management
- A **configuration management** system
- A **systems integration** platform
- An **API** for your entire Infrastructure

# Principles

- Idempotent
- Reasonability
- Sane defaults, easily changed
- Hackability
- TMTOWTDI

# The world moves pretty fast

- **Primitives**
- Enable you to solve your problems
- Your **best skill** is knowing your systems



# So you better be ready

- You need a 3rd generation language
- You're already a programmer





OPSCODE

# What is Chef?

**Chef is a data-driven systems integration platform**

**Chef runs infrastructure as a business application**

# Chef enables Infrastructure as Code



```
1 package "sudo" do
2   action :upgrade
3 end
4
5 template "/etc/sudoers" do
6   source "sudoers.erb"
7   mode 0440
8   owner "root"
9   group "root"
10  variables(
11    :sudoers_groups => node[:authorization][:sudo][:groups],
12    :sudoers_users => node[:authorization][:sudo][:users]
13  )
14 end
```

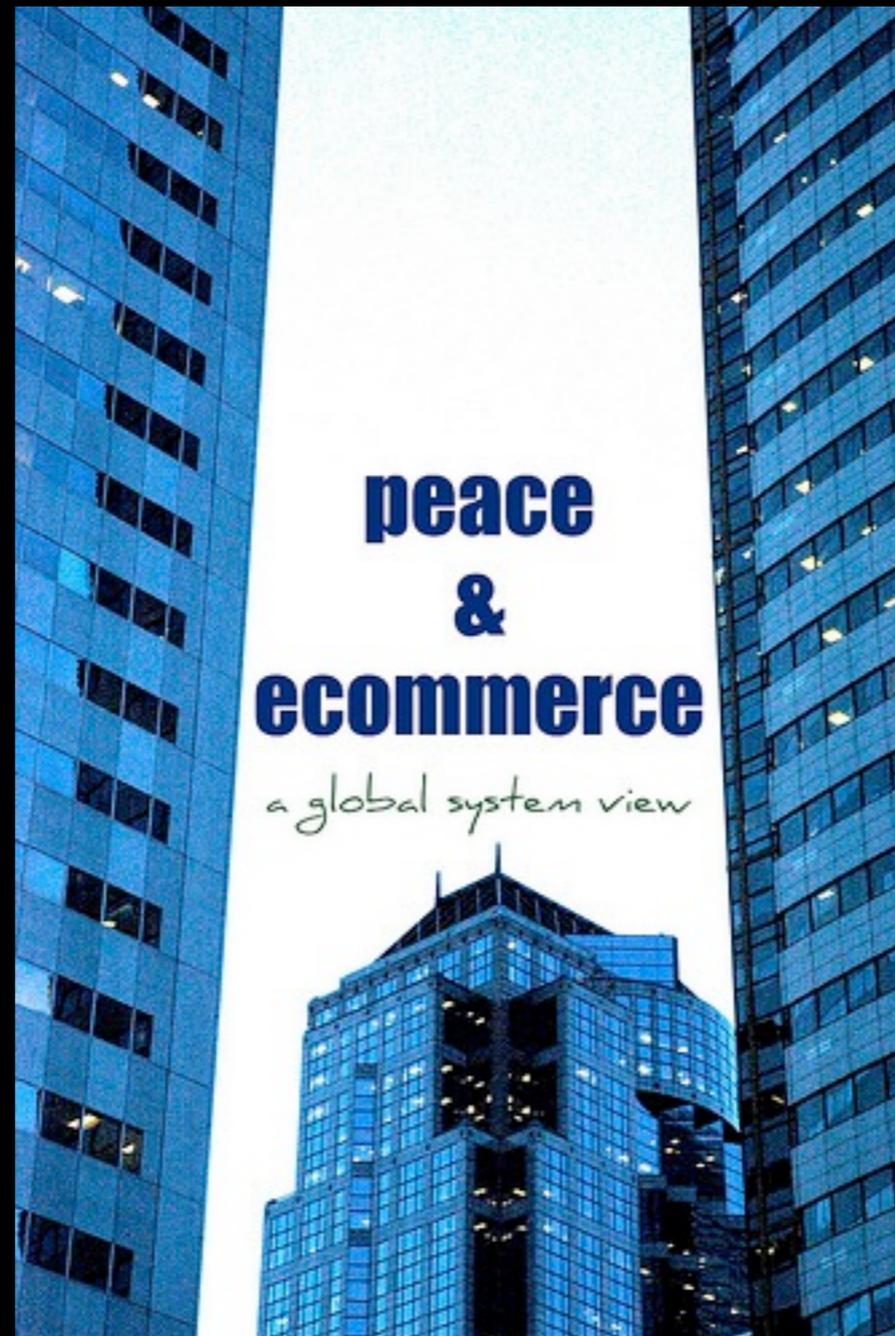
Manage configuration as idempotent **Resources**.

Put them together in **Recipes**.

Track it like **source code**.

**Configure** your servers.

# Infrastructure as Code



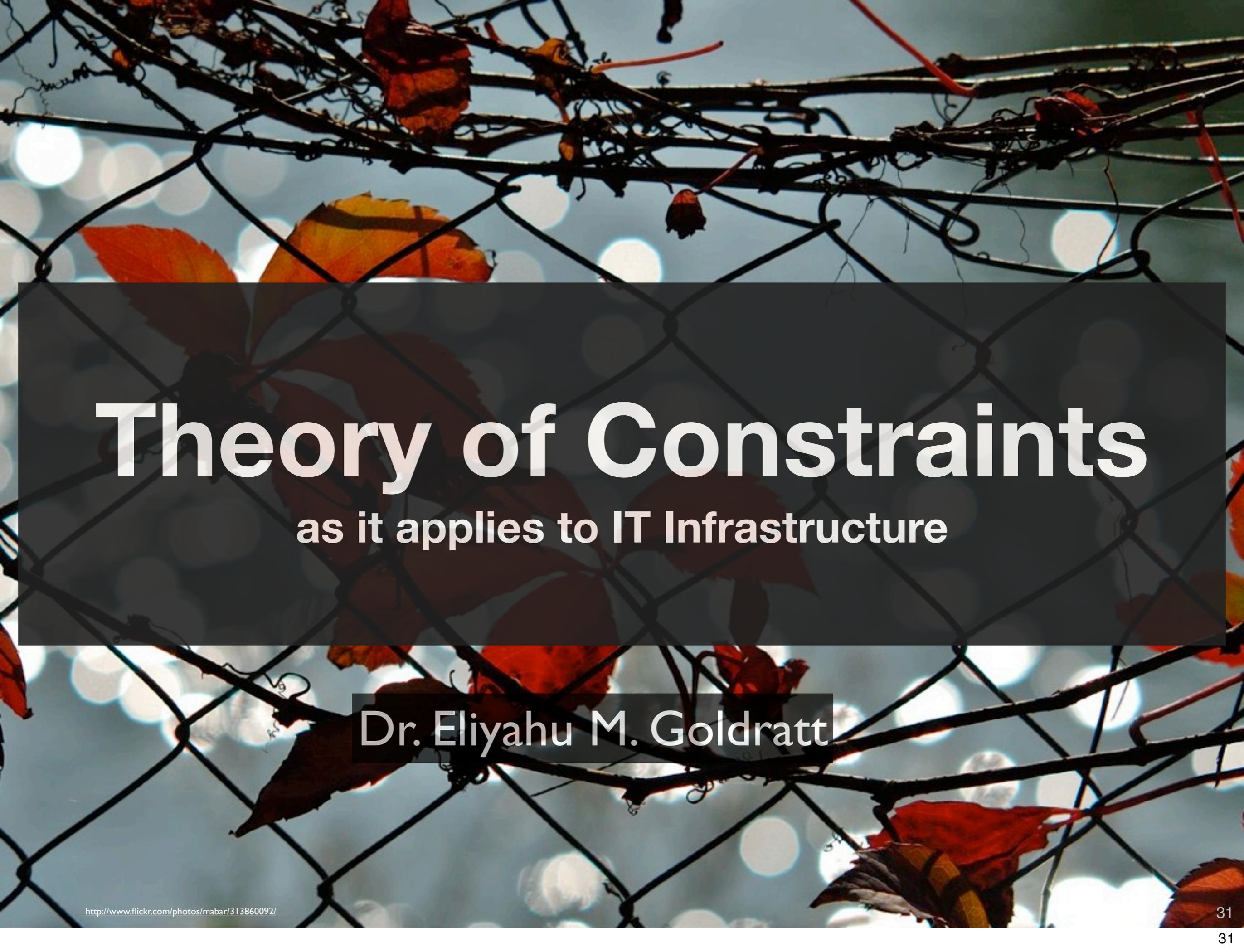
<http://www.flickr.com/photos/wonderlane/2306082998/>

# Infrastructure as Code is...

**A technical domain revolving around building and managing infrastructure programmatically**



**Enable the reconstruction of  
the business from nothing  
but a source code  
repository, an application  
data backup, and bare metal  
resources.**



# Theory of Constraints

as it applies to IT Infrastructure

Dr. Eliyahu M. Goldratt

# Your Prime Constraint Should Be

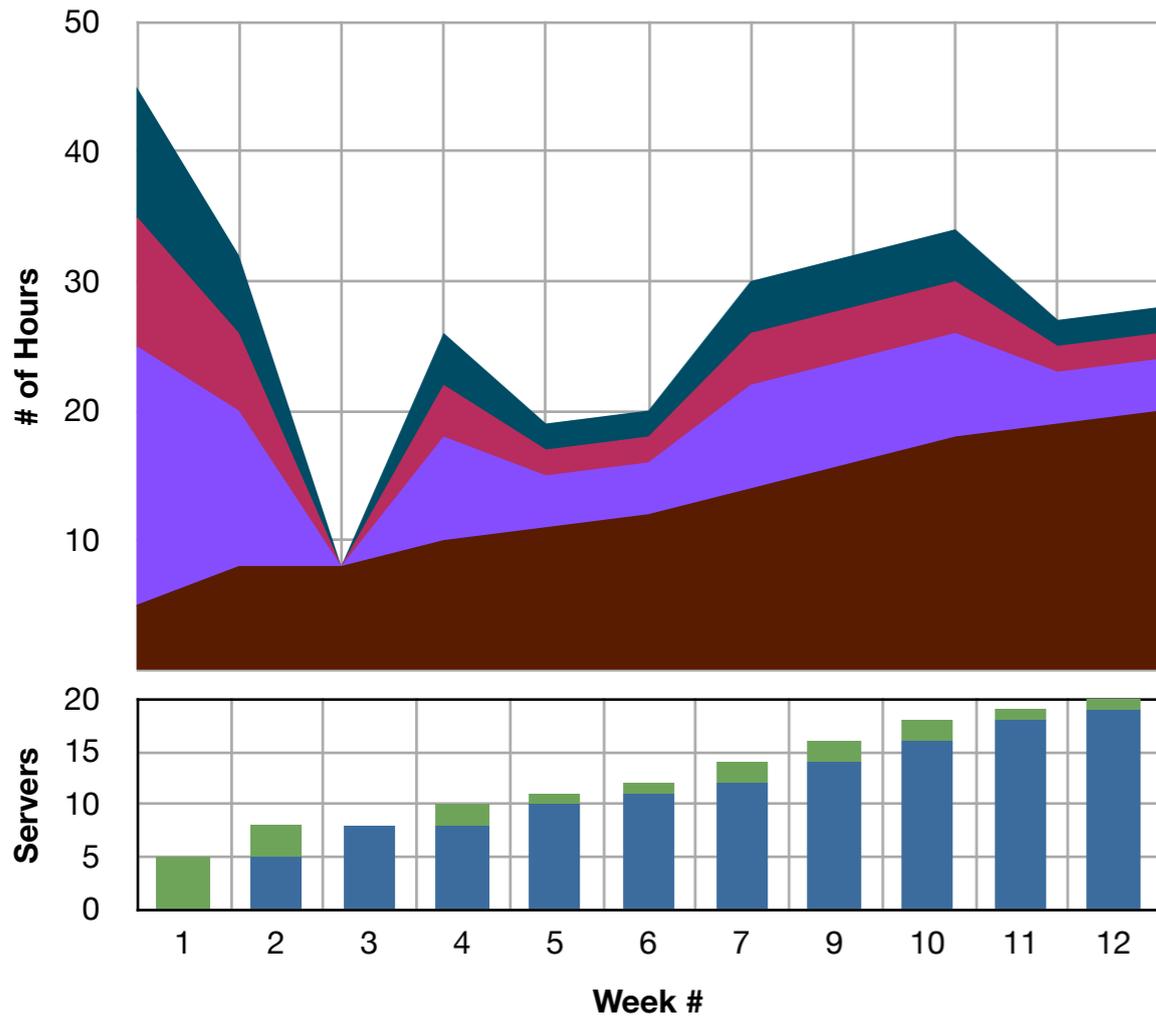
**The time it  
takes to  
restore your  
application  
data**



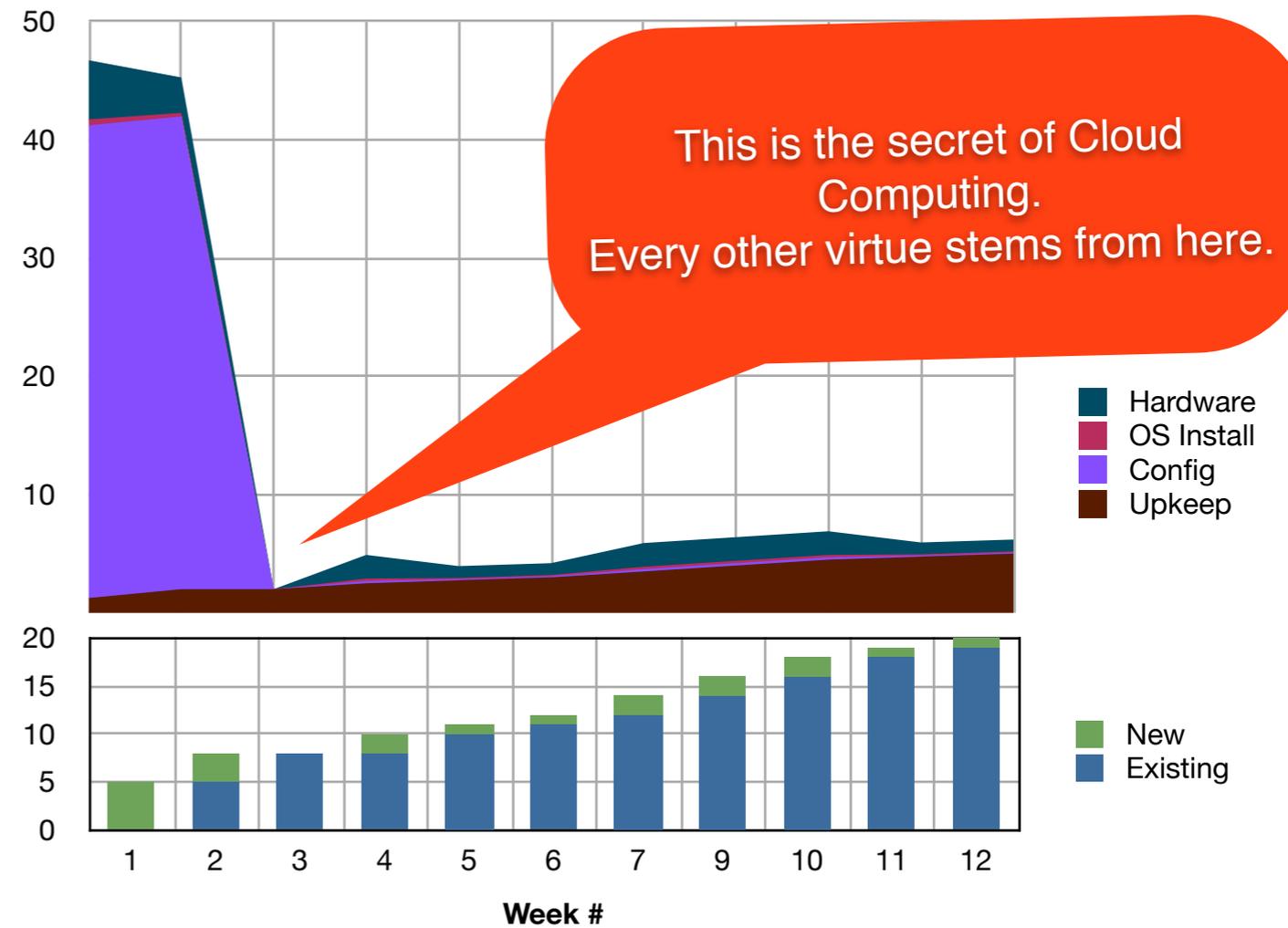
<http://www.flickr.com/photos/visualage/2126833132/sizes/o/>

# Tale of Two Startups

“Traditional” Operations



Operations - The “Secret Sauce”



# Wax Philosophical

- We are craftsmen and artists
- You need great brushes and tools
- But nobody remembers Gaudi's hammer



# SUCCESS

The ladder of success was not crowded at the top. Did I set it against the right building?

<http://www.flickr.com/photos/aloshbennett/619307160/sizes//>

- Can only be measured by the final solution



# SUCCESS

The ladder of success was not crowded at the top. Did I set it against the right building?

<http://www.flickr.com/photos/aloshbennett/619307160/sizes//>

- Hey, that looks a little rough around the edges and pieced-together... **BUT IT WORKS**

# A Tour of Chef





OPSCODE

# Some Principles of Chef's Design...

You specify the desired result and Chef takes action that will place the system in that state.



Chef prefers failure over non-deterministic "success" when something goes wrong.



Managed clients do most of the work and thus can ask useful questions about their surroundings and the rest of the infrastructure for systems integration.



**Chef exposes data and behavior through HTTP to enable integration with external tools.**



Apache licensed, Chef's foundation is a growing and vibrant world-class open source community with a culture of quality and reusability.

# Chef Cookbook Examples

Tomcat6

Glassfish

Apache2

Mysql

Dynomite

Hadoop

Riak

Postgresql

CouchDB

Java

Erlang

Solr

RabbitMQ

ZooKeeper

Subversion

Kickstart

Libvert

Xen

OpenIdap

OpenVPN

AWS

Rsync

Xfs

OpenSSL

**Chef Client** runs on your  
systems

# They talk to **Chef** **Servers**

**They can also stand  
alone - Chef Solo**

We call each system you  
configure a **Node**

# Nodes have a **Run List**

What **Roles** or **Recipes** to apply  
in **Order**

# Nodes have a **Run List**

```
1 "run_list": [  
2   "role[webserver]",  
3   "role[database_master]",  
4   "role[development]"  
5 ]
```

# Nodes have Roles

# Chef manages **Resources** on Nodes

# A Resource...

- Is of a given **type**.
- Has a **name**.
- Has **attributes**.
- Takes **actions** to bring the resource to a declared state.

```
20 package "sudo" do
21   version '1.6.8p12'
22   action :install
23 end
```

```
57 service "slapd" do
58   supports [ :restart, :status ]
59   action [ :enable, :start ]
60 end
```

*Is a declarative description of the state you desire a part of your node to be in.*

```
20 package "sudo" do
21   action :upgrade ← Package Resource
22 end
23
24 template "/etc/sudoers" do
25   source "sudoers.erb"
26   mode 0440 ← Template Resource
27   owner "root"
28   group "root"
29   variables(
30     :sudoers_groups => node[:authorization][:sudo][:groups],
31     :sudoers_users => node[:authorization][:sudo][:users]
32   )
33 end
```

Resources take *action*  
through **Providers**

# A Provider...

*Knows how to actually perform the actions specified by a resource.*

Multiple providers  
per resource type.



Apt, Yum, Rubygems,  
Portage, Macports,  
FreeBSD Ports, etc.

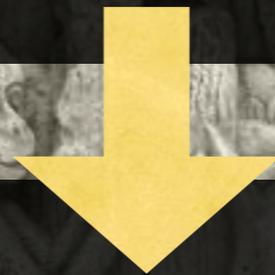
Can be overridden  
with the **provider**  
attribute on a  
resource.

```
33 package "sudo" do
34   provider Chef::Provider::Package::Yum
35   action :install
36 end
```

# Resources



# Platform



# Provider

life

<http://www.flickr.com/photos/acurbelo/2628837104/sizes/o/>

**Recipes** are lists of  
**Resources**

**Cookbooks** are  
*packages* for Recipes

# Cookbooks

## Attributes

### ▶ Node attributes

```
9 default[:nginx][:gzip] = "on"
10 default[:nginx][:gzip_http_version] = "1.0"
11 default[:nginx][:gzip_comp_level] = "2"
12 default[:nginx][:gzip_proxied] = "any"
13 []
14 default[:nginx][:keepalive] = "on"
15 default[:nginx][:keepalive_timeout] = 65
16
17 default[:nginx][:worker_processes] = cpu[:total]
18 default[:nginx][:worker_connections] = 4096
```

# Cookbooks

## Metadata

*cookbooks/god/metadata.rb*

```
1 maintainer      "Opscode, Inc."
2 maintainer_email "cookbooks@opscode.com"
3 license         "Apache 2.0"
4 description     "Installs and configures god and provides a define for monitoring
5 long_description IO.read(File.join(File.dirname(__FILE__), 'README.rdoc'))
6 version        "0.7"
7
8 %w{debian ubuntu}.each do |os|
9   supports os
0 end
1
2 %w{ ruby runit }.each do |cb|
3   depends cb
4 end
```

# Cookbooks

## Other Components

- ▶ **definitions**
- ▶ **libraries**
- ▶ **resources**
- ▶ **providers**

# Cookbooks are shareable!

**Search**   [Advanced Search](#)

Search results for 'category: \"Databases\"' Cookbooks

**mysql**  1 rating Category: Databases 578 downloads 14 followers  
Platforms: debian, ubuntu  
Installs and configures mysql for client or server

**database**  0 ratings Category: Databases 19 downloads 0 followers  
Sets up the database master or slave

cookbooks.opscodel.com

# Add your Cloudy credentials to knife

```
$ vi ~/.chef/knife.rb
```

```
10 knife[:rackspace_api_key] = "SECRET"  
11 knife[:rackspace_api_username] = "BOBO"  
12 knife[:aws_access_key_id] = "SUPERSEKRET"  
13 knife[:aws_secret_access_key] = "EVENMORESEKRET"  
14 knife[:aws_ssh_key_id] = "MYSSHKEYBRINGSALLTHEBOYSTOTHEYARD"
```

# Launch a new Rails Server

```
$ knife rackspace server create 'role  
[rails]'
```

```
$ knife ec2 server create 'role[rails]'
```

**Or just install Chef on  
your existing servers**

# Questions?

**[www.opscode.com](http://www.opscode.com)**

- ▶ **Twitter: @opscode, #opschef**
- ▶ **irc.freenode.net #chef**
- ▶ **lists.opscode.com**