Automated Infrastructure is on the Menu with Chef

OSCON 2010

Speakers:

Joshua Timberman  Technical Evangelist
Aaron Peterson  Technical Evangelist
Agenda

Chef 101
Getting Started
Cooking with Chef
Rate the tutorial and comment


Twitter IDs, hashtags

- #OSCON
- @opscode, #opschef
- @jtimberman, @metaxis

Questions
Developers?
Systems Administrators?

http://www.flickr.com/photos/timyates/2854357446/sizes/l/
The *Opscode Platform* is our Chef Server

http://www.opscode.com
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
Data-driven
Sane defaults
Hackability
TMTOWTDI
Multiple applications of an operation do not change the result
We start with **APIs**, you supply **data**
option :json_attribs,
  :short => "-j JSON_ATTRIBS",
  :long => "--json-attributes JSON_ATTRIBS",
  :description => "Load attributes from a JSON file or URL",
  :proc => nil

option :node_name,
  :short => "-N NODE_NAME",
  :long => "--node-name NODE_NAME",
  :description => "The node name for this client",
  :proc => nil

Defaults are **sane**, but easily changed
Open source and community
Tim Toady is a Perl motto
The world moves pretty fast

Primitives
Enable YOU
Know your systems

http://www.flickr.com/photos/gi/518613153/sizes/o/
You better be ready!

**Sysadmins are programmers**

**You need a 3GL**
Do I need to know Ruby?

A little
Simple syntax
Complex as you scale
A Tour of Chef
Chef Client runs on your systems
Clients talk to a Chef Server
Clients authenticate with RSA keys
Chef Server

Chef Server

API

Indexer

AMQP

SOLR

CouchDB

Search

Client Interaction

Data store
RESTful API w/ JSON responses
Chef can also stand alone - Chef Solo
We call each system you configure a **Node**
Nodes have Attributes

```json
{
  "kernel": {
    "machine": "x86_64",
    "name": "Darwin",
    "os": "Darwin",
    "version": "Darwin Kernel Version 10.4.0: Fri Apr 23 18:28:53 PDT 2010; root:xnu-1504.7.4~1/RELEASE_I386",
    "release": "10.4.0"
  },
  "platform_version": "10.6.4",
  "platform": "mac_os_x",
  "platform_build": "10F569",
  "domain": "local",
  "os": "darwin",
  "current_user": "jtimberman",
  "ohai_time": 1278602661.60043,
  "os_version": "10.4.0",
  "uptime": "18 days 17 hours 49 minutes 18 seconds",
  "ipaddress": "10.13.37.116",
  "hostname": "cider",
  "fqdn": "cider.local",
  "uptime_seconds": 1619358
}
```
Attributes are Searchable

```
$ knife search node 'platform:mac_os_x'
search(:node, 'platform:mac_os_x')
```
Nodes have a Run List

What Roles or Recipes to apply in Order
Nodes have a Run List

% knife node show web01-prod.example.com -r
{
  "run_list": [
    "role[production]",
    "role[webserver]"
  ]
}
Nodes have Roles
Roles

Describe the node

- webserver
- dbserver
- monitoring
- ... etc
Roles have a Run List

What Roles or Recipes to apply in Order
name "webserver"
description "Systems that serve HTTP traffic"

run_list(
  "role[base]",
  "recipe[apache2]",
  "recipe[apache2::mod_ssl]"
)

default_attributes(
  "apache" => {
    "listen_ports" => [ "80", "443" ]
  }
)

override_attributes(
  "apache" => {
    "max_children" => "50"
  }
)

Can include other roles!
% knife role show webserver
{
  "name": "webserver",
  "default_attributes": {
    "apache": {
      "listen_ports": [
        "80",
        "443"
      ]
    }
  },
  "json_class": "Chef::Role",
  "run_list": [
    "role[base]",
    "recipe[apache2]",
    "recipe[apache2::mod_ssl]"
  ],
  "description": "Systems that serve HTTP traffic",
  "chef_type": "role",
  "override_attributes": {
    "apache2": {
      "max_children": "50"
    }
  }
}
Roles are **Searchable**

```bash
$ knife search role 'max_children:50'
search(:role, 'max_children:50')
```
Chef manages Resources on Nodes
Resources...

Declare a description of the state a part of the node should be in

- Have a **type**

- Have a **name**

- Have **parameters**

- Take **action** to put the resource in the declared state

```yaml
package "apache2" do
  version "2.2.11-2ubuntu2.6"
  action :install
end

template "/etc/apache2/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  group "root"
  mode 0644
  action :create
end
```
Resources take action through Providers
Providers...

Know 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 parameter on a resource.

```ruby
package "apache2" do
  provider "Chef::Provider::Package::Dpkg"
  action :install
end
```

http://www.flickr.com/photos/affableslinky/562950216/
Chef::Platform

```ruby
:ubuntu => {
  :default => {
    :package => Chef::Provider::Package::Apt,
    :service => Chef::Provider::Service::Debian,
    :cron => Chef::Provider::Cron,
    :mdadm => Chef::Provider::Mdadm
  }
},
```
Recipes are lists of Resources
Recipes...

Apply resources in the order they are specified

- Evaluates resources in the order they appear
- Adds each resource to the Resource Collection

```ruby
package "apache2" do
  version "2.2.11-2ubuntu2.6"
  action :install
end

template "/etc/apache2/apache2.conf" do
  source "apache2.conf.erb"
  owner "root"
  group "root"
  mode 0644
  action :create
end
```
Order Matters
Recipes are just **Ruby!**

```ruby
extra_packages = case node[:platform]
  when "ubuntu", "debian"
    %w{
      ruby1.8
      ruby1.8-dev
      rdoc1.8
      ri1.8
      libopenssl-ruby
    }
  end

extra_packages.each do |pkg|
  package pkg do
    action :install
  end
end
```
Cookbooks are packages for Recipes
Cookbooks

Distributable
Infrastructure as Code
Version control repository
Common Cookbook Components

Recipes
Assets (files/templates)
Attributes
Metadata
Cookbook assets

Files

- Static assets
- Downloaded via cookbook_file
- File specificity

Templates

- Dynamic assets
- ERB (erubis)
- File specificity
Attributes

- Node attributes
- default, normal, override

```ruby
default[:apache][:listen_ports] = ["80","443"]
default[:apache][:keepalive] = "On"
default[:apache][:contact] = "ops@example.com"
default[:apache][:timeout] = 300

set[:apache][:log_dir] = "/var/log/apache2"
set[:apache][:user] = "www-data"

override[:apache][:dir] = "/etc/apache2"
```
Cookbooks

Metadata

cookbooks/django/metadata.rb

maintainer "Opscode, Inc."
maintainer_email "cookbooks@opscode.com"
license "Apache 2.0"
description "Installs DJango"
long_description IO.read(File.join(File.dirname(__FILE__), 'README.rdoc'))
version "0.8.0"

recipe "django", "Installs django and apache2 with mod_python"

%w{ ubuntu debian }.each do |os|
supports os
end

%w{ apache2 python }.each do |cb|
depends cb
end
Cookbooks are shareable!

cookbooks.opscode.com
Data bags store arbitrary data
A user data bag item...

```bash
% knife data bag show users jtimberman
{
  "comment": "Joshua Timberman",
  "groups": "sysadmin",
  "ssh_keys": "ssh-rsa SUPERSEKRATS jtimberman@cider",
  "files": {
    ".zshrc": {
      "mode": "0644",
      "source": "dot-zshrc"
    },
    ".vimrc": {
      "mode": "0644",
      "source": "dot-vimrc"
    }
  },
  "id": "jtimberman",
  "uid": 7004,
  "shell": "/usr/bin/zsh",
  "openid": "http://jtimberman.myopenid.com/"
}
```
Data Bags are Searchable

$ knife search users 'shell:/bin/bash'
search(:users, ‘/bin/bash’)
bash_users = search(:users, 'shell:/bin/bash')

bash_users.each do |u|
  user u['id'] do
    uid u['id']
    shell '/usr/bin/zsh'
    comment u['comment']
    supports :manage_home => true
    home '/home/#{u['id']}'
  end

directory '/home/#{u['id']}/.ssh' do
  owner u['id']
  group u['id']
  mode 0700
end

template '/home/#{u['id']}/.ssh/authorized_keys' do
  source 'authorized_keys.erb'
  owner u['id']
  group u['id']
  mode 0600
  variables :ssh_keys => u['ssh_keys']
end
end
Getting Started
Platforms

Debian
CentOS
SuSE
Gentoo
Solaris
Mac OS X
ArchLinux
OpenBSD
Windows
Red Hat
Scientific
Ubuntu
Fedora
FreeBSD
Ruby
Today’s Examples

Opscode Platform
Mac OS X 10.6.4
Ubuntu 10.04
RubyGems
The **Opscode Platform** is our Chef Server

http://www.opscode.com
RubyGems Installation

```
cd /tmp
wget http://production.cf.rubygems.org/rubygems/rubygems-1.3.7.tgz -O- | tar zxf -
cd rubygems-1.3.7 && sudo ruby setup.rb
ln -svf /usr/bin/gem1.8 /usr/bin/gem

sudo gem install chef
```
ELFF Yum Repo
Create Chef Repository

% git clone git://github.com/opscode/chef-repo.git
% cd chef-repo
% ls -la

drwxr-xr-x  13 jtimberman staff  442 Jul  7 16:48 ./
drwxr-xr-x   3 jtimberman staff  102 Jul  7 15:54 ../
drwxr-xr-x   5 jtimberman staff  170 Jul  7 17:55 .chef/
drwxr-xr-x  12 jtimberman staff  408 Jul  7 16:48 .git/
-rw-r--r--   1 jtimberman staff   23 Jul  7 16:48 .gitignore
-rw-r--r--   1 jtimberman staff  269 Jul  7 15:54 README
-rw-r--r--   1 jtimberman staff 2171 Jul  7 15:54 Rakefile
drwxr-xr-x   3 jtimberman staff  102 Jul  7 15:54 certificates/
drwxr-xr-x   7 jtimberman staff  238 Jul  7 17:03 config/
drwxr-xr-x   3 jtimberman staff  102 Jul  7 15:54 cookbooks/
drwxr-xr-x   3 jtimberman staff  102 Jul  7 15:54 data_bags/
drwxr-xr-x   3 jtimberman staff  102 Jul  7 15:54 roles/
drwxr-xr-x   3 jtimberman staff  102 Jul  7 15:54 site-cookbooks/
Platform Keys
Keys let you and your systems interact with the Opscode Platform.

Note: Opscode does not keep a copy of your keys.

Knife Configuration
Knife is Chef’s command-line tool. This is a pre-configured copy of your Knife config file.

What's Next?
Read the Getting Started Guide
setup command-line access and your first cookbook

Experienced with Chef?
manage your org with the Operations Console

Need Help?
If you have questions, or you’re stuck, we’re here to help.
Get Help
Setup User Environment

cp USERNAME.pem ~/chef-repo/.chef

cp ORG-validator.pem ~/chef-repo/.chef

cp knife.rb ~/chef-repo/.chef
Configure Knife

% cat .chef/knife.rb

```ruby
current_dir = File.dirname(__FILE__)
log_level :info
log_location STDOUT
node_name "oscon"
client_key "#{current_dir}/oscon.pem"
validation_client_name "oscon-validator"
validation_key "#{current_dir}/oscon-validator.pem"
chef_server_url "https://api.opscode.com/organizations/oscon"
cache_type 'BasicFile'
cache_options( :path => "#{ENV['HOME']}/.chef/checksums" )
cookbook_path ["#{current_dir}/../cookbooks"]
```

Per-directory configuration!
Explore Knife’s subcommands
Knife Sub-commands

`knife NOUN verb NOUN (options)`

```plaintext
knife client list
knife node show cider.local
knife role show webserver
knife search node "*::*" -i
knife --help
```
Configure Chef on workstation

% knife configure client /etc/chef
INFO: Creating client configuration
INFO: Writing client.rb
INFO: Writing validation.pem

% ls -l /etc/chef/
total 24
-rw-r--r--  1 jtimberman  staff   151 Jul  8 21:29 client.rb
-rw-r--r--@ 1 jtimberman  staff  1679 Jul  8 21:29 validation.pem
Chef::Config

log_level :info
log_location STDOUT
chef_server_url 'https://api.opscode.com/organizations/oscon'
validation_client_name 'oscon-validator'

http://wiki.opscode.com/display/chef/Chef+Configuration+Settings
Download **getting-started** cookbook

% knife cookbook site vendor getting-started
INFO: Downloading getting-started from the cookbooks site at version 0.2.0
...
INFO: Cookbook getting-started version 0.2.0 successfully vendored!
This site is currently in Alpha. Please use the feedback widget to the right to let us know what you think.

Need Help?
If you have questions, or you're stuck, we're here to help. Get Help

Popular Tags
Ruby db dns logs opsc ode rails security ssh ubuntu web
git checkout -b \
chef-vendor-#{name_args[0]}
Upload getting-started to Chef Server

% knife cookbook upload getting-started
INFO: Saving getting-started
INFO: Validating ruby files
INFO: Validating templates
INFO: Syntax OK
INFO: Generating Metadata
INFO: Uploading files
...
INFO: Upload complete!
Apply getting-started Recipe to workstation

% knife node run list add cider.local "recipe[getting-started]"
{
    "run_list": [  
        "recipe[getting-started]"
    ]
}
Run **chef-client**!

% sudo chef-client
[Thu, 08 Jul 2010 21:35:49 -0600] INFO: Starting Chef Run
[Thu, 08 Jul 2010 21:35:56 -0600] INFO: Chef Run complete in 6.650602 seconds

% cat /tmp/chef-getting-started.txt
Welcome to Chef!

This is Chef version 0.9.6.
Running on mac_os_x.
Version 10.6.4.
Inside the getting-started cookbook
Cooking with Chef
Chef Resources in Depth

cookbook_file  link  remote_file

template  ruby_block  service

execute  user

package  bash

deploy  git

http_request  log
Resources have parameters and actions
Resource Parameters and Actions

Most resources have defaults
Defaults are sane

- [http://wiki.opscode.com/display/chef/Resources](http://wiki.opscode.com/display/chef/Resources)
packages

package "apache2" do
  action :install
end

Actions: install, upgrade, remove, purge
Package shortcuts

Map to providers

gem_package
dpkg_package
rpm_package

And more!
services

service "apache2" do
  action [ :enable, :start ]
end

Actions: enable, disable, start, stop, restart, reload
init script capabilities

service "apache2" do
  supports(
    :status => true,
    :restart => true,
    :reload => true
  )
  action [ :enable, :start ]
end
files

file "*/etc/`chef/client.rb" do
  owner "root"
  group "root"
  mode 0644
  action :create
end

Actions: create, delete, touch
file content!

```ruby
file "./tmp/example" do
  content "This is a file!"
end

file "./tmp/example2" do
  content IO.read("./etc/hosts")
end
```

Content is a string
remote_file

remote_file "tmp/nginx-0.7.67.tar.gz" do
  source "http://sysoev.ru/nginx/nginx-0.7.67.tar.gz"
  action :create_if_missing
end

Actions: create, create_if_missing
cookbook_file

cookbook_file "/etc/perl/CPAN/Config.pm" do
  source "Config-5.10.1.pm"
  owner "root"
  group "root"
  mode 0644
end

Actions: create, create_if_missing, delete
template

template "/etc/apache2/ports.conf" do
  source "ports.conf.erb"
  owner "root"
  group "root"
  mode 0644
end

Actions: create
local templates

template "*/tmp/config.conf" do
  local true
  source "*/tmp/config.conf.erb"
end
templates are ERB

<% node[:apache][:listen_ports].each do |port| -%>
  Listen <%= port %>
  NameVirtualHost *:<%= port %>
<% end -%>
Cookbook and Template

File Specificity
preferences = [
    File.join("host-#{fqdn}", "#{file_name}")],
    File.join("#{platform}-#{version}", "#
    {file_name}"),
    File.join("#{platform}", "#{file_name}"),
    File.join("default", "#{file_name}"))
]

host-node[:fqdn]
node[:platform]-node[:version]
node[:platform]
default

files/web1prod.example.com
files/ubuntu-9.10
files/ubuntu
files/default
execute

execute "apt-get update" do
  action :run
end

Actions: run
bash "compile/nginx_source" do
  cwd "/tmp"
  code <<-EOH
    tar zxf nginx-0.7.67.tar.gz
    cd nginx-0.7.67 && ./configure
    make && make install
  EOH
end

Interpreters: bash, ruby, python, perl, csh
ruby_block

```
ruby_block "save the node" do
  block do
    node.save
  end
end
```

Action: create
scm: git...

git "/srv/apps/chef" do
  repository "git://github.com/opscode/chef.git"
  reference "0.9.6"
  action :checkout
end

Actions: sync, checkout, export
...and subversion

subversion "/srv/couchdb" do
  repository "http://svn.apache.org/repos/asf/couchdb/trunk"
  revision "HEAD"
  action :sync
end

Actions: sync, checkout, export
deploy

http://wiki.opscode.com/display/chef/Deploy+Resource

Actions: deploy, force_deploy, rollback
deploy "/srv/radiant" do
  repo "git://github.com/radiant/radiant.git"
  revision "HEAD"
  user "railsdev"
  migrate true
  migration_command "rake db:migrate"
  environment "production"
  restart_command "touch tmp/restart.txt"
  action :deploy
end
Meta-parameter

madness!
action :nothing
not_if & only_if
execute "runit-hup-init" do
  command "telinit q"
  only_if "grep ^SV /etc/inittab"
  action :nothing
end

execute "rabbitmqctl add_vhost /chef" do
  not_if "rabbitmqctl list_vhosts| grep /chef"
end

Enclose in quotes for shell commands or use a do..end or { } style ruby block
Resource notification

template "nginx.conf" do
  path "/etc/nginx/nginx.conf"
  source "nginx.conf.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, resources(:service => "nginx")
end
supports
Anatomy of a Chef Run
Anatomy of a Chef Run

Node discovery
Set the node name
Register with server
Anatomy of a Chef Run

Build node object

- `node.save`

Synchronize cookbooks

- `node.save`

Converge

- `node.save`
Chef Run Convergence

Compile

Execute
Development workflow with Chef
Development workflow

Gather requirements
Write recipes
Commit to repository
Run Chef in testing
Run Chef in production
Keeping things simple requires complicated effort.

Automating the Cloud with Chef
Add your Cloud credentials to knife.rb

vi ~/chef-repo/.chef/knife.rb

# Cloud credentials
knife[:aws_access_key_id] = ENV['AWS_ACCESS_KEY_ID']
nknife[:aws_secret_access_key] = ENV['AWS_SECRET_ACCESS_KEY']
Download some cookbooks

% knife cookbook site vendor nagios -d
INFO: Downloading nagios from the cookbooks site at version 0.3.3
...
INFO: Cookbook nagios version 0.3.3 successfully vendored!
INFO: Downloading apache2 from the cookbooks site at version 0.12.0
INFO: Cookbook apache2 version 0.12.0 successfully vendored!

Uses the “vendor branch” pattern, so you can make changes and track the upstream
Upload Cookbooks!

knife cookbook upload -a

These run as root, kids. Let’s not blindly trust the upstream too much!
Build some roles

% vi roles/monitoring.rb

name "monitoring"
description "Nagios monitoring server"

run_list(
  "role[base]",
  "recipe[nagios::server]"
)

override_attributes(
  "apache" => {
    "allowed_openids" => "http://jtimberman.myopenid.com/"
  }
)
Upload Roles

% knife role from file roles/monitoring.rb
WARN: HTTP Request Returned 404 Not Found: Cannot load role monitoring
WARN: Updated Role monitoring!

% ls roles
README base.rb monitoring.rb production.rb webserver.rb

% rake roles
(in /Users/jtimberman/Development/oscon/chef-repo)
WARN: HTTP Request Returned 404 Not Found: Cannot load role base
WARN: Updated Role base!
WARN: Updated Role monitoring!
WARN: HTTP Request Returned 404 Not Found: Cannot load role production
WARN: Updated Role production!
WARN: HTTP Request Returned 404 Not Found: Cannot load role webserver
WARN: Updated Role webserver!)
Launch a new Monitoring Server

```
knife ec2 server create 'role[monitoring]'`
Chef runs on your new server

sudo chef-client

INFO: Starting Chef Run

... INFO: Chef Run complete in 211.852033 seconds

Automatically.
Shef is Chef in IRB
Resources/Questions

www.opscode.com

IRC and Mailing lists

- irc.freenode.net #chef
- lists.opscode.com

Twitter:

- @opscode, #opschef
- @jtimberman, @metaxis

Questions?