LIGHTWEIGHT CONFIGURATION MANAGEMENT

(for Fun and for Profit)

BACKGROUND

PUPPET

VAGRANT

CHEF

CAPISTRANO

PROVISIONER INDEPENDENT

OUR PATTERN, A WALKTHROUGH

DEMO

```
 git clone https://github.com/kuleszaj/chef-an-introduction
cd chef-an-introduction
bundle install
bundle exec cap berks
bundle exec cap vrails4 vg:up
bundle exec cap vrails4 bootstrap
bundle exec cap vrails4 chef
```

GEMFILE

```
# -*- mode: ruby -*-
# # vi: set ft=ruby :

source "http://rubygems.org"

# Required for capistrano
gem "capistrano"

# Not required, but may be useful for debugging
#gem "pry"

# Required/Useful for Chef
gem "chef","11.4.4"
gem "berkshelf"
gem "foodcritic"
```

BERKSFILE

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

site :opscode

cookbook "apt"
cookbook "apache2"
cookbook "database"
```

VAGRANTFILE

```
# -*- mode: ruby -*-
# vi: set ft=ruby :

@user_configuration = {}
load File.join(File.dirname(__FILE__),"config","stages.rb")

# This is specifically design for VirtualBox. You could fairly easily adapt it to another Vagrant provider such
Vagrant.configure("2") do |config|

  config.ssh.forward_agent = true

  # These types of options should be project specific:
  config.vm.box = "precise64"
  config.vm.box_url = "http://files.vagrantup.com/precise64.box"

  @user_configuration["vagrants"].each do |name|
    config.vm.define "#{name}" do |vagrant|
      vagrant.vm.hostname = "#{name}"
      vagrant.vm.network :private_network, type: :dhcp
    end
  end
end
```

CHEF CONFIG

```
chef
├── README.md
├── cookbooks
│   ├── apache2
│   ├── apt
│   ├── aws
│   ├── build-essential
│   ├── database
│   ├── mysql
│   ├── openssl
│   ├── postgresql
│   └── xfs
├── rails4.json
├── site-cookbooks
│   ├── README.md
│   └── rails4
├── solo.rb
└── vrails4.json
```

RAILS4.JSON

```
{
  "mysql":
    {
      "server_root_password": "somethingsecret",
      "server_repl_password": "somethingsecret",
      "server_debian_password": "somethingsecret"
    },
  "run_list":
    [
      "recipe[apache2]",
      "recipe[mysql::client]",
      "recipe[mysql::ruby]",
      "recipe[mysql::server]",
      "recipe[rails4]"
    ]
}
```

CAPISTRANO CONFIG

```
config/
├── README.md
├── base/
│   ├── berkshelf.rb
│   ├── bootstrap.rb
│   ├── chef.rb
│   ├── ssh.rb
│   ├── vg.rb
│   └── warning.rb
├── custom/
│   ├── README.md
│   └── custom_example.rb
├── deploy.rb
├── ssh_configs/
│   └── example_ssh_config
└── stages.rb
```

BENEFITS OF OUR APPROACH

- Lightweight (no master or server)
- Two important indirections
  - target node (local / remote)
  - Also easy to plug in another provisioning tool (we know Puppet and Chef both work, others probably do too)
- Uses tools most Rails developers are already familiar with (Ruby, Capistrano)
- Uses standard SSH config files to describe hosts (increasing portability and reusability)

THANKS!

CREDITS

"The Swedish Chef" image - Andrew Kuchling
http://www.flickr.com/photos/akuchling/190704045/

QUESTIONS?