

Chef

Infrastructure as code

Background

- SaaS application
- Some clients had outgrown standard needs
- They want private instances
- They want run older but stable releases

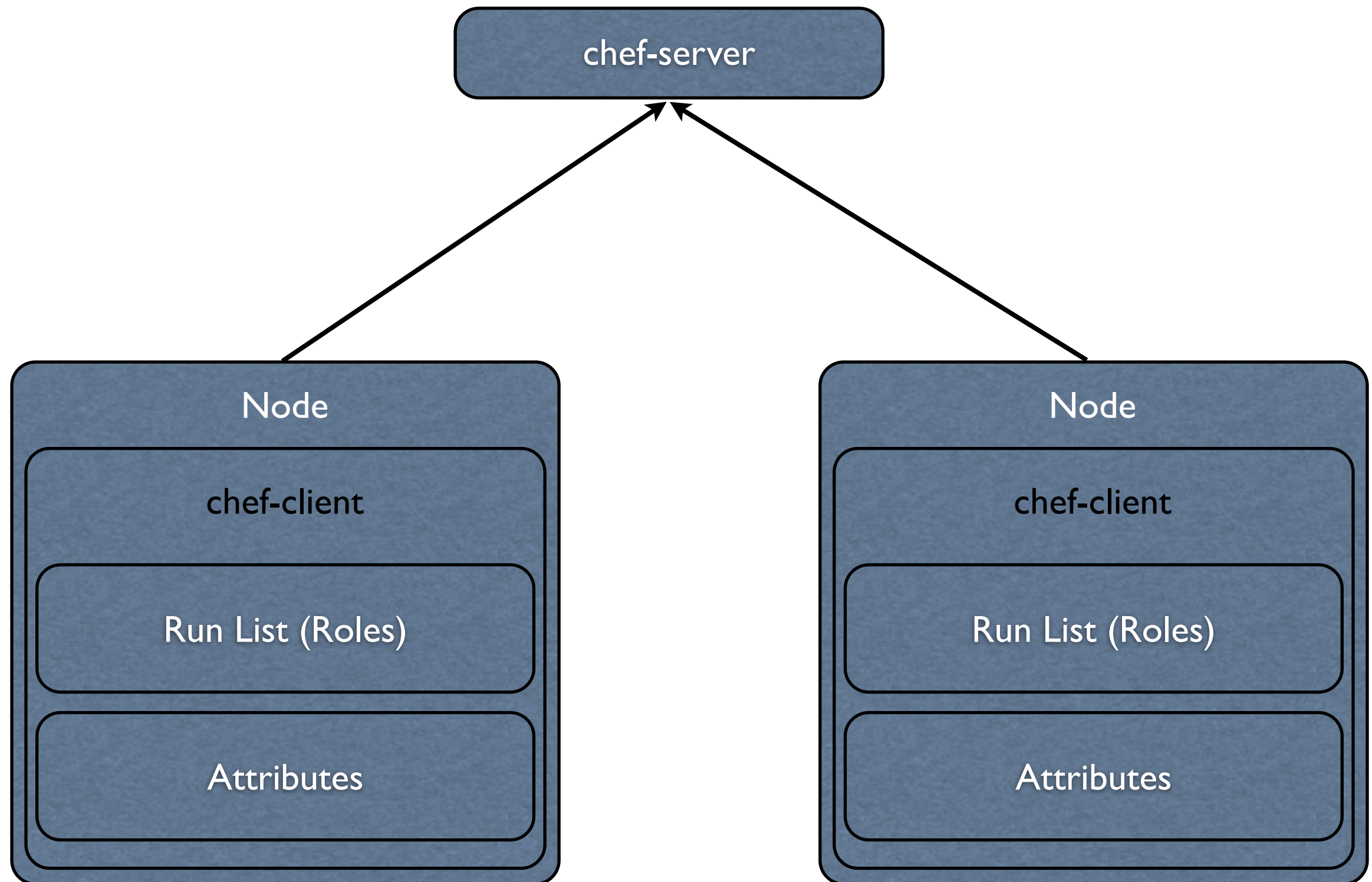
Problems

- Configuration management
- Monitoring
- Updates
- Deployment

Chef

- Systems integration framework
- Client-server architecture
- Idempotence
- Imperative approach
- Ruby everywhere
- Configuration as code

Basic terminology



Basic terminology

- Client
- Node
- Role
- Resource
- Recipe
- Cookbook
- Attribute

Basic components

- chef-client
- ohai
- chef-server
- chef-webui
- knife

Node lifecycle

- Bootstrap
- Configure
- Control

Bootstrap

```
$ knife bootstrap 178.79.166.70 -x root -P password -r 'role[base]' -d ubuntu10.04 -N new-node
```

chef-client run

- Authenticate node
- Synchronize cookbooks
- Compile
- Converge

Chef repository

```
chef-repo  
|-certificates  
|-config  
|-cookbooks  
|-data_bags  
|-roles  
|-script  
|---Rakefile
```

<https://github.com/opscode/chef-repo>

Cookbook

cookbook

| -attributes

| -definitions

| -files

| -libraries

| -recipes

| -templates

| ---metadata.rb

Recipe

```
package "sendmail" do  
  action :install  
end
```

Recipe

```
%w{releases shared/config shared/log}.each do |dir|  
  directory "/var/www/apps/application/#{dir}" do  
    owner "deploy"  
    group "deploy"  
    mode 0755  
    recursive true  
  end  
end
```

Recipe

```
script "do_something_scary" do
  interpreter "bash"
  user "root"
  cwd "/tmp"
  code <<-EOS
    wget http://www.example.com/tarball.tar.gz
    tar -zxf tarball.tar.gz
  EOS
  not_if { File.exists? "/tmp/lock" }
end
```

Templates

```
# passenger.conf.erb
# Auto-generated. Local modifications will be overwritten.
passenger_root <%= node[:rvm_passenger][:root_path] %>;
passenger_ruby <%= node[:rvm_passenger][:ruby_wrapper] %>;
```

```
# passenger_nginx.rb
template "/etc/conf.d/passenger.conf" do
  source "passenger_nginx.conf.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, resources(:service => "nginx")
end
```


Resources

Cookbook File	HTTP Request	Remote Directory
Cron	Ifconfig	Remote File
Deploy	Link	Route
Directory	Log	Ruby Block
Env	Mdadm	SCM
Erlang Call	Mount	Script
Execute	Ohai	Service
File	Package	Subversion
Git	PowerShell Script	Template
Group	User	

Providers

Resource

Interface

package

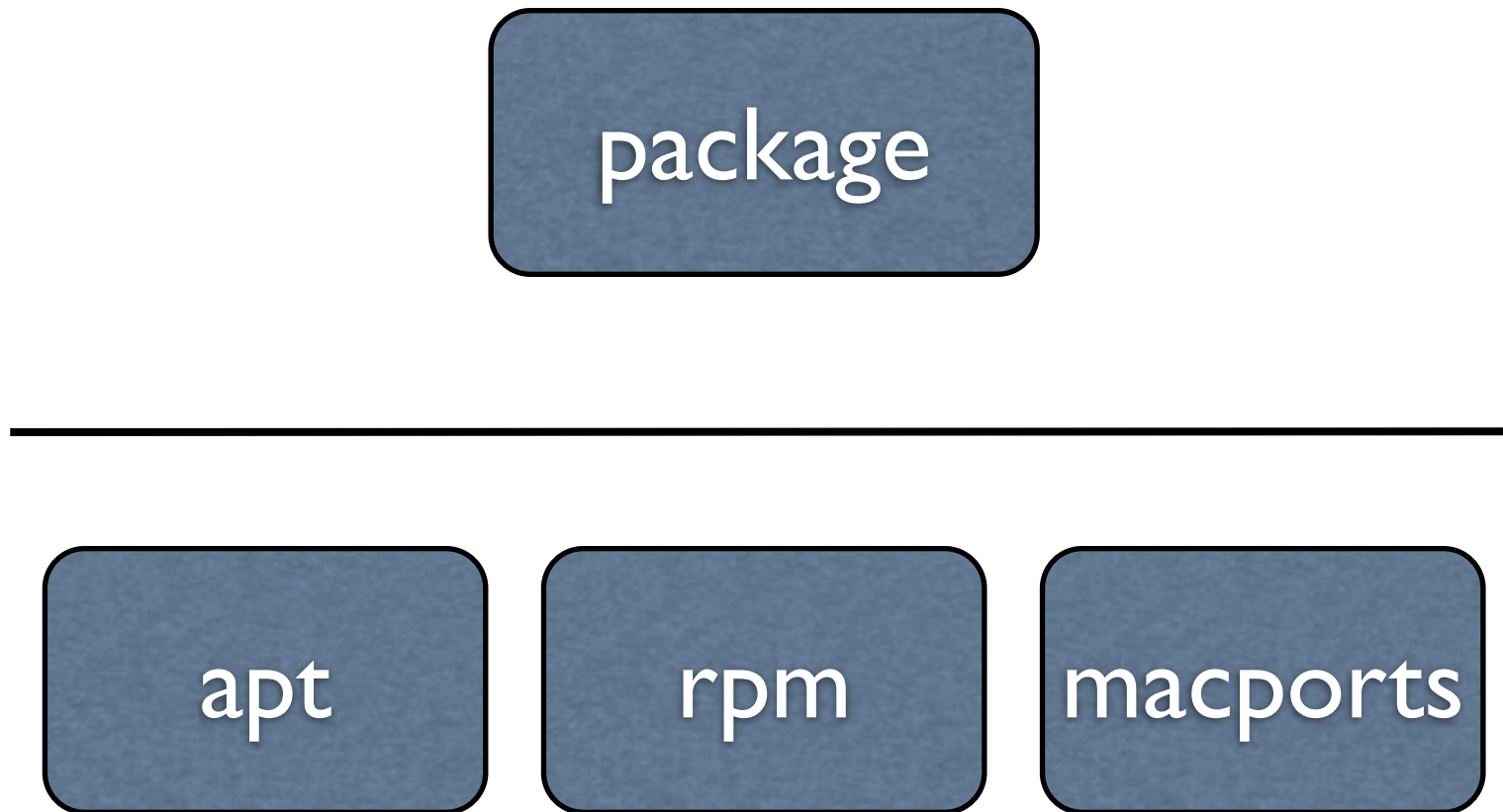
apt

rpm

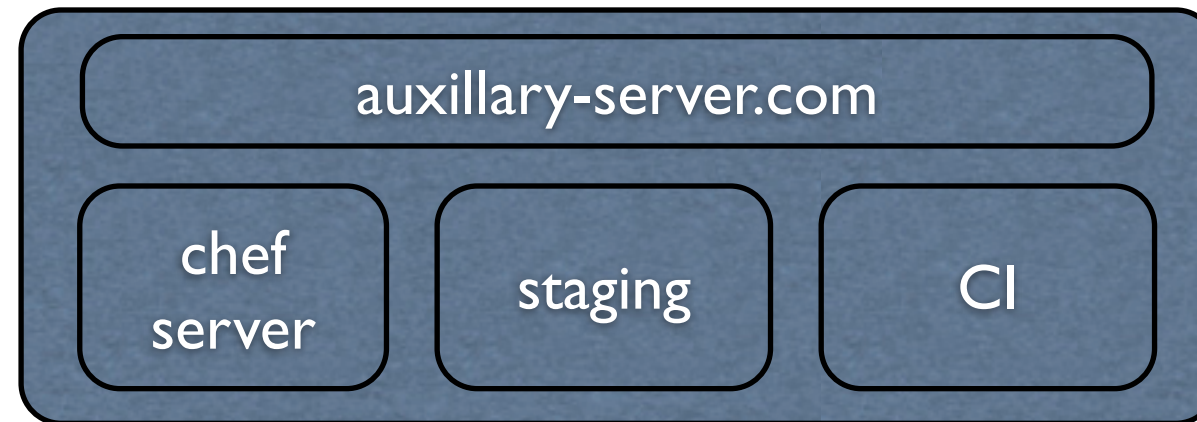
macports

Providers

Implementation



Our setup



VPN

Deployment

```
deploy_revision "/var/www/apps/application" do
  repo "git@github.com:iafonov/ha.git"
  environment "RAILS_ENV" => "production"
  branch "release7"
  action :deploy
  restart_command "touch tmp/restart.txt"
end
```

Deployment

```
$ rake deploy
```

```
$ rake deploy:production
```

```
$ rake rollback
```

```
$ rake rollback:production
```

Deployment Hint #1

(UNIX is your friend)

```
# /etc/ssh_config  
ForwardAgent yes
```

Deployment Hint #2

(UNIX is your friend)

```
file "/etc/sudoers.d/deploy_chef" do
  owner "root"
  group "root"
  mode 0440
  content <<-EOS
    Defaults          env_keep = "SSH_AUTH_SOCK"
    deploy ALL= NOPASSWD: /usr/bin/chef-client
  EOS
end
```

API

```
class Deployer
  def initialize(query_str)
    @nodes = Chef::Search::Query.new.search(:node, query_str)
    @ssh = SshWrapper.new.configure_session(@nodes)
  end

  def deploy
    set_action_and_update_nodes('deploy')
  end

  def rollback
    set_action_and_update_nodes('rollback')
  end
end

private
def set_action_and_update_nodes(action)
  @nodes.each {|node| node.set['groupinator']["deploy_action"] = action}
  @ssh.ssh_command("sudo chef-client")
end
end
```

```
Deployer.new("name:staging").deploy
```


Testing

- Vagrant - <http://vagrantup.com/>
- Linode/Amazon

Dark sides

- Complexity
- Lack of dry-run mode
- Complexity
- Lack of documentation

Links

- <http://wiki.opscode.com/display/chef/Home>
- <https://github.com/opscode/chef>
- <http://tickets.opscode.com/browse/CHEF>

<http://iafonov.github.com/>
@iafonov