

GOING  
WITH  
BUDDHY



KING

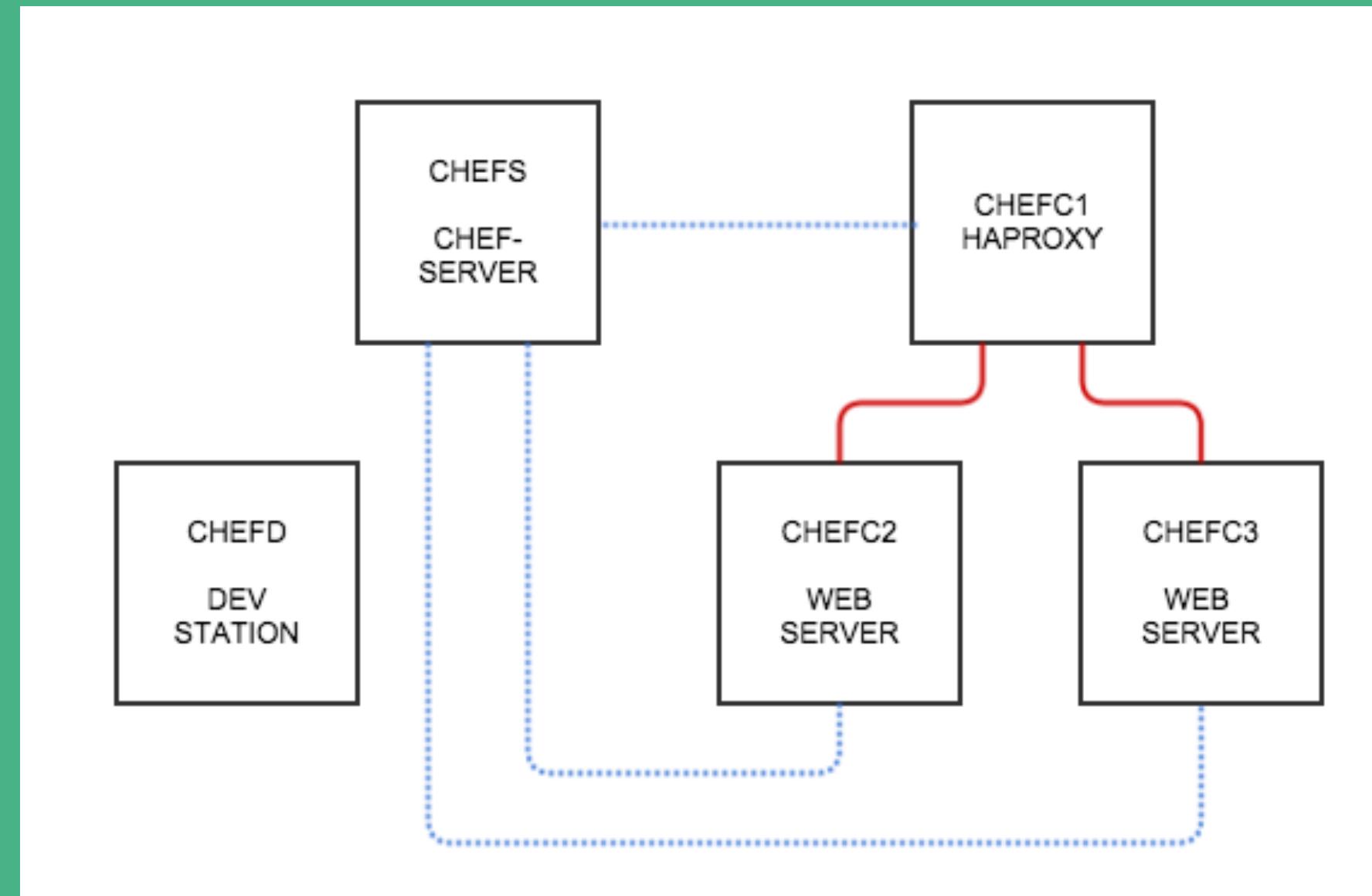
# REQUIREMENTS

1. Homebrew <http://brew.sh/> (brew doctor!)
2. brew-cask brew install caskroom/cask/brew-cask
3. Virtualbox brew cask install virtualbox
4. Vagrant brew cask install vagrant
5. Git brew install git

# LAB BOOTSTRAP

1. git clone <http://github.com/caquino/chef-class>
2. cd chef-class
3. vagrant up

# LAB DIAGRAM



# LAB VM'S

» Server (SSH/WEB INTERFACE) hostname: chefs

» Client (SSH) hostname: chefc1, chefc2

» Development station (SSH) hostname: chefd

SSH Access: vagrant ssh <hostname> e.g: (chefs |  
chefc1 | chefc2 | chefd)

WEB-UI Access: <http://chefs.local>

# KNIFE CONFIGURATION

## 1. User private key web-ui

Access WEB-UI go to Users -> Create

Make sure you select admin

Save the private key at `chefd:~vagrant/.chef/<username>.pem`

## 2. Validation private key `chef:/etc/chef-server` save the file at `~/.chef/chef-validator.pem`

# KNIFE CONFIGURATION

```
vagrant@chefd:~$ vi ~/.chef/knife.rb
```

```
user = ENV['USER'] || 'YOUR USERNAME'  
node_name user  
client_key "#{ENV['HOME']}/.chef/#{user}.pem"  
validation_client_name "chef-validator"  
validation_key "#{ENV['HOME']}/.chef/chef-validator.pem"  
chef_server_url "https://chefs.local/"  
syntax_check_cache_path "#{ENV['HOME']}/.chef/syntax_check_cache"  
cookbook_path [ "#{ENV['HOME']}/chef/cookbooks" ]  
knife[:editor]="/usr/bin/vi"
```

# BOOTSTRAP

## BASIC

- » Register a new client on chef-server using `chef-validator.pem`
- » Creates a new node on the chef-server

```
vagrant@chefd:~$ knife bootstrap -x vagrant chefc1.local -N chefc1.local --sudo  
vagrant@chefd:~$ knife bootstrap -x vagrant chefc2.local -N chefc2.local --sudo  
vagrant@chefd:~$ knife bootstrap -x vagrant chefc3.local -N chefc3.local --sudo  
vagrant@chefd:~$ printf "%s\n" {1..3} | xargs -P 3 -I % knife bootstrap -x vagrant  
chefc%.local -N chefc%.local --sudo
```

# ENVIRONMENTS

- » map an organization's real-life workflow  
e.g: production, staging, testing, and development environments.
- » also associated with one (or more) cookbook versions.

```
vagrant@chefd:~$ knife environment create lab_environment
vagrant@chefd:~$ knife environment list
vagrant@chefd:~$ knife node environment_set chefcl.local lab_environment
vagrant@chefd:~$ knife node environment_set chefc2.local lab_environment
vagrant@chefd:~$ knife node environment_set chefc3.local lab_environment

vagrant@chefd:~$ printf "%s\n" {1..3} | xargs -P 3 -I % knife node
environment_set chefc%.local lab_environment
```

# NODES

- » physical, virtual, or cloud machine that is configured to be maintained by a chef-client.
- » attributes and a run\_list e.g:

```
vagrant@chefd:~$ knife node <show|edit|delete> <nodename> [-l]
vagrant@chefd:~$ knife node show chefc2.local
vagrant@chefd:~$ knife node show -l chefc1.local | less
vagrant@chefd:~$ knife node edit chefc1.local
```

# RUN\_LIST

- » defines configuration settings that are necessary for a node that is under management by Chef to be put into the desired state.
- » ordered list of roles and/or recipes

# ATTRIBUTES

- » ohai automatic attributes  
try it! vagrant@chefc1:~/
- » cookbooks code / attributes file
- » roles
- » environment
- » node

# ROLES

- » zero (or more) attributes and a run-list
- » nodes can be part of zero (or more)
- » attributes and run\_lists shared accross nodes

e.g:

dbservers, webservers, base\_role

```
vagrant@chefd:~$ knife role list
```

```
vagrant@chefd:~$ knife role create lab_web_role
```

```
vagrant@chefd:~$ knife role create lab_lb_role
```

# CHEF-CLIENT HOW TO EXECUTE?

```
vagrant@chefd:~$ ssh chefc1.local sudo /usr/bin/chef-client
vagrant@chefd:~$ ssh chefc2.local sudo /usr/bin/chef-client
vagrant@chefd:~$ ssh chefc3.local sudo /usr/bin/chef-client
```

```
vagrant@chefd:~$ printf "%s\n" {1..3} | 
xargs -P 3 -I % ssh chefc%.local sudo /usr/bin/chef-client
```

# COOKBOOKS

- » fundamental unit of configuration and policy distribution.
- » defines a scenario, e.g: everything needed to install and configure MySQL
- » all of the components that are required to support that scenario, e.g: Attributes, Files, Recipes, Resources, Providers, Templates, Versions, Metadata

# COOKBOOKS LAB

```
vagrant@chefd:~/ $ knife cookbook create lab_nginx_cookbook  
vagrant@chefd:~/ $ knife cookbook create lab_haproxy_cookbook
```

# COOKBOOKS METADATA

```
vagrant@chefd:~/chef/cookbooks/lab_nginx_cookbook$ vi metadata.rb
```

```
name          'lab_nginx_cookbook'
maintainer    'Zendesk International'
maintainer_email 'caquino@zendesk.com'
license        'All rights reserved'
description    'Installs/Configures lab_nginx_cookbook'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version        '0.1.0'
```

# COOKBOOKS

## METADATA - SUPPORT AND DEPENDS

```
%{ ubuntu redhat }.each do |os|
  supports os
end
%{ ohai chef_handler }.each do |package|
  depends package
end
supports "ubuntu"
supports "redhat"
depends "ohai"
depends "chef_handler"
```

# COOKBOOKS RECIPES

```
vagrant@chefd:~/chef/cookbooks/lab_nginx_cookbook$ vi recipes/default.rb
```

```
package "nginx" do
  action :upgrade
end

file "/usr/share/nginx/www/index.html" do
  content node[:hostname]
end

service "nginx" do
  action [:enable, :start]
  supports :reload => true
end
```

# COOKBOOKS UPLOAD

```
vagrant@chefd:~/ $ knife cookbook upload lab_nginx_cookbook
```

# ROLES AND MULTIPLE NODES

```
vagrant@chefd:~$ knife role edit lab_web_role
{
...
  "run_list": [
    "recipe[lab_nginx_cookbook]"
  ],
...
}
}
```

# RUN LIST AND NODES

```
vagrant@chefd:~$ knife node run_list add chefc2.local "role[lab_web_role]"
vagrant@chefd:~$ knife node run_list add chefc3.local "role[lab_web_role]"
vagrant@chefd:~$ ssh chefc2.local sudo /usr/bin/chef-client
vagrant@chefd:~$ ssh chefc3.local sudo /usr/bin/chef-client
```

# COOKBOOKS METADATA

```
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ vi recipes/default.rb
```

```
name          'lab_haproxy_cookbook'
maintainer    'Zendesk International'
maintainer_email 'caquino@zendesk.com'
license        'All rights reserved'
description    'Installs/Configures lab_haproxy_cookbook'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version        '0.1.0'
```

# COOKBOOKS

## RECIPES - HAProxy

```
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ vi recipes/default.rb
```

```
package "haproxy" do
  action :upgrade
end

pool_members = search("node", "role:lab_web_role")

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  variables :pool_members => pool_members.uniq
  notifies :reload, "service[haproxy]", :delayed
end

cookbook_file "haproxy-default" do
  path "/etc/default/haproxy"
  owner "root"
  group "root"
  mode 0644
  notifies :restart, "service[haproxy]", :delayed
end

service "haproxy" do
  action [:enable, :start]
  supports :reload => true
end
```

# COOKBOOKS RECIPES - HAProxy

...

```
pool_members = search("node", "role:lab_web_role")

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  variables :pool_members => pool_members.uniq
  notifies :reload, "service[haproxy]", :delayed
end
```

...

# COOKBOOKS TEMPLATES

```
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ mkdir -p templates/default/
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ vi templates/default/haproxy.cfg.erb
```

```
global
  user haproxy
  group haproxy
  daemon

listen http 0.0.0.0:80
<% @pool_members.each do |member| -%>
  server <%= member[:hostname] %> <%= member[:network][:interfaces][:eth1][:addresses].select
  { |address, data| data[:family] == 'inet' }.keys[0] %>
<% end -%>

<% if node["haproxy"]["enable_admin"] -%>
listen admin 0.0.0.0:22002
  mode http
  stats uri /
<% end -%>
```

# COOKBOOKS RECIPES - HAProxy

```
....  
cookbook_file "haproxy-default" do  
  path "/etc/default/haproxy"  
  owner "root"  
  group "root"  
  mode 0644  
  notifies :restart, "service[haproxy]", :delayed  
end  
....
```

# COOKBOOKS FILES

```
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ vi files/default/haproxy-default
```

```
# Set ENABLED to 1 if you want the init script to start haproxy.  
ENABLED=1  
# Add extra flags here.  
#EXTRAOPTS="-de -m 16"
```

# COOKBOOKS ATTRIBUTES

```
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ vi attributes/default.rb
```

```
default["haproxy"]["enable_admin"] = false
```

# COOKBOOKS UPLOAD

```
vagrant@chefd:~$ knife cookbook upload lab_haproxy_cookbook
```

# ROLES AND MULTIPLE NODES

```
vagrant@chefd:~$ knife role edit lab_lb_role
{
...
"run_list": [
    "recipe[lab_haproxy_cookbook]"
],
...
}
```

# RUN LIST AND NODES

```
vagrant@chefd:~$ knife node run_list add chefc1.local "role[lab_lb_role]"  
vagrant@chefd:~$ ssh chefc1.local sudo /usr/bin/chef-client
```

# NODE ATTRIBUTES

```
vagrant@chefd:~$ knife node edit chefcl.local
```

```
{
  "name": "chefcl.local",
  "chef_environment": "lab_environment",
  "normal": {
    "haproxy": {
      "enable_admin": true
    },
    "tags": [
    ]
  },
  "run_list": [
    "role[lab_lb_role]"
  ]
}
```

```
vagrant@chefd:~$ ssh chefcl.local sudo /usr/bin/chef-client
```

# ATTRIBUTES PRECEDENCE

	Attribute Files	Node / Recipe	Environment	Role
default	1	2	3	4
force_default	5	6		
normal	7	8		
override	9	10	12	11
force_override	13	14		
automatic			15	

# LAB RECAP

```
node[chefc1.local]
  role[lab_lb_role]
    recipe[lab_haproxy_cookbook]
```

```
node[chefc2.local]
  role[lab_web_role]
    recipe[lab_nginx_cookbook]
```

```
node[chefc3.local]
  role[lab_web_role]
    recipe[lab_nginx_cookbook]
```

# DATA BAGS

- » global variable that is stored as JSON data
- » indexed for searching and can be loaded by a recipe or accessed during a search.

e.g:

database\_clusters, users

# DATA BAGS

```
vagrant@chefd:~$ knife data bag create lab_environment_secrets haproxy
```

```
{  
  "id": "haproxy",  
  "admin": {  
    "port": 8081,  
    "username": "admin",  
    "password": "whatever"  
  }  
}
```

# COOK BOOKS

```
...
pool_members = search("node", "role:lab_web_role")
config = data_bag_item("lab_environment_secrets", "haproxy")
...
template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  variables ({
    :pool_members => pool_members.uniq,
    :admin => config['admin']
  })
  notifies :restart, "service[haproxy]", :delayed
end
...
```

# COOKBOOKS TEMPLATES

```
vagrant@chefd:~/chef/cookbooks/lab_haproxy_cookbook$ vi templates/default/haproxy.cfg.erb
```

```
global
  user haproxy
  group haproxy
  daemon

listen http 0.0.0.0:80
<% @pool_members.each do |member| -%>
  server <%= member[:hostname] %> <%= member[:network][:interfaces][:eth1][:addresses].select
  { |address, data| data[:family] == 'inet' }.keys[0] %>
<% end -%>

<% if node["haproxy"]["enable_admin"] -%>
listen admin 0.0.0.0:<%= @admin["port"] %>
  mode http
  stats uri /
  stats auth <%= @admin["username"] %>:<%= @admin["password"] %>
<% end -%>
```

# COOKBOOKS UPLOAD

```
vagrant@chefd:~$ knife cookbook upload lab_haproxy_cookbook  
vagrant@chefd:~$ ssh chefc1.local sudo /usr/bin/chef-client
```

# VERSION FREEZE

```
vagrant@chefd:~$ knife environment edit lab_environment
```

```
{
...
"cookbook_versions": {
    "lab_nginx_cookbook": "= 0.1.0"
},
...
}
```

# COOKBOOK

```
vagrant@chefd:~/chef/cookbooks/lab_nginx_cookbook$ vi recipes/default.rb
```

```
...
file "/usr/share/nginx/www/index.html" do
  content "Node: #{node[:fqdn]}"
end
```

```
...
```

# **COOKBOOK METADATA**

```
vagrant@chefd:~/chef/cookbooks/lab_nginx_cookbook$ vi metadata.rb
```

...

```
version "0.1.1"
```

...

# COOKBOOKS UPLOAD

```
vagrant@chefd:~$ knife cookbook upload lab_nginx_cookbook  
vagrant@chefd:~$ ssh chefc2.local sudo /usr/bin/chef-client  
vagrant@chefd:~$ ssh chefc3.local sudo /usr/bin/chef-client
```

# COOKBOOKS VERSIONS

```
vagrant@chefd:~$ knife cookbook show lab_nginx_cookbook
```

# COOKBOOK VERSION BUMP

```
vagrant@chefd:~$ knife environment edit lab_environment
```

```
{
...
"cookbook_versions": {
    "lab_nginx_cookbook": "= 0.1.1"
},
...
}
```

# CHEF CLIENT

```
vagrant@chefd:~$ ssh chefc2.local sudo /usr/bin/chef-client  
vagrant@chefd:~$ ssh chefc3.local sudo /usr/bin/chef-client
```

# CONCLUSIONS

## WHAT WE SAW?

- » Nodes (bootstrapping, run\_list, attributes)
- » Environments (creation, cookbooks freeze, attributes)
- » Roles (creation, run\_list, attributes)
- » Cookbooks (creation, upload, templates, files, recipes, versioning, search)
- » Data bags (creation, usage)

# RAPID REVIEW COMMANDS

```
knife client <edit|show|delete|list> client_name
```

```
knife user <edit|show|delete|add|list> user_name
```

```
knife node <edit|show|delete|list> node_name
```

```
knife cookbook <upload|show|delete|create|list> cookbook_name
```

```
knife environment <edit|show|delete|create|list> [from file] environment_name|file_name.json
```

```
knife role <edit|show|delete|create|list> [from file] role_name|file_name.json
```

```
knife data bag <edit|show|delete|create|list> [from file] data_bag_name item_name|file_name.json
```

# BONUS CONTENT

- » open source cookbooks

## ADVANCED CLASS - TODO

- » LWRP's
- » Template Helpers
- » Reporting Handlers
- » Testing (chefspec, serverspec, minitest-handler, berkshelf, kitchen)

THANKS