Separating Environments

Objective:

- ❑ Use Search to separate out the environments

- ❑ Update myhaproxy cookbook's version number

Expected Situation

What do we expect to happen when we set a web node to a specific environment?

Balancing Nodes

Which cookbook handles balancing the requests between web nodes?

Which recipe within that cookbook sets up the request balancing between the two nodes?

Search Criteria

How are we currently searching for web nodes?

How can we further refine our search results?

Search Criteria

`~/chef-repo/cookbooks/myhaproxy/recipes/default.rb`

```
#
# Cookbook Name:: myhaproxy
# Recipe:: default
#
# Copyright (c) 2016 The Authors, All Rights Reserved.

all_web_nodes = search('node','role:web')


members = []


#...
```

GL: Modify the myhaproxy default.rb

`~/chef-repo/cookbooks/myhaproxy/recipes/default.rb`

```ruby
#
# Cookbook Name:: myhaproxy
# Recipe:: default
#
# Copyright (c) 2016 The Authors, All Rights Reserved.

all_web_nodes = search('node',"role:web AND chef_environment:#{node.chef_environment}")

members = []

#...
```

GL: Version the myhaproxy metadata.rb

~/chef-repo/cookbooks/myhaproxy/metadata.rb

```
name              'myhaproxy'
maintainer        'The Authors'
maintainer_email  'you@example.com'
license           'all_rights'
description       'Installs/Configures myhaproxy'
long_description  'Installs/Configures myhaproxy'
version           '1.0.1'


depends 'haproxy', '= 2.0.0'
```

CHEF

GL: Run 'berks install'

```
$ cd cookbooks/myhaproxy
$ berks install
```

```
Resolving cookbook dependencies...
Fetching 'myhaproxy' from source at .
Fetching cookbook index from https://supermarket.chef.io...
Using build-essential (2.2.3)
Installing haproxy (2.0.0)
Using cpu (0.2.0)
Using myhaproxy (1.0.1) from source at .
```

GL: Run 'berks upload'

```
$ berks upload
```

```
Skipping build-essential (2.2.3) (frozen)

Skipping cpu (0.2.0) (frozen)

Skipping haproxy (2.0.0) (frozen)

Uploaded myhaproxy (1.0.1) to:
'https://api.opscode.com:443/organizations/vogue'
```

A Brief Recap

We restricted the production environment to specific cookbook version.

We created an acceptance environment with no cookbook restrictions.

We set web1 and load-balancer to the production environment, and web2 to the acceptance environment.

We updated the myhaproxy's default recipe to include environment search criteria.

And we changed the version number in the myhaproxy metadata.rb file.

Update Production Environment

❏ Update the environment named production:

`'myhaproxy'` cookbook version equal to `'1.0.1'`

Lab: Update production.rb

```
name 'production'
description 'Where we run production code'

cookbook 'apache', '= 0.2.1'
cookbook 'myhaproxy', '= 1.0.1'
```

Lab: cd and Run 'knife environment...'

```
$ cd ~/chef-repo
$ knife environment from file production.rb
```

```
Updated Environment production
```

Lab: Verify the Version Number

```
$ knife environment show production
```

```
chef_type:          environment
cookbook_versions:
  apache:     = 0.2.1
  myhaproxy: = 1.0.1
default_attributes:
description:        Where we run production code
json_class:         Chef::Environment
name:               production
override_attributes:
```

Login to Load Balancer

```
$ vagrant ssh load-balancer
```

```
Last login: Sat Dec 31 02:59:27 2016 from 10.0.2.2
[vagrant@load-balancer ~]$
```

Converge the Load Balancer

```
[vagrant@load-balancer ~]$ sudo chef-client
```

```
Starting Chef Client, version 12.17.44
resolving cookbooks for run list: ["myhaproxy"]
Synchronizing Cookbooks:
  - myhaproxy (0.1.0)
  - haproxy (2.0.0)
  - build-essential (7.0.3)
  - seven_zip (2.0.2)
  - windows (2.1.1)
  - ohai (4.2.3)
....
```
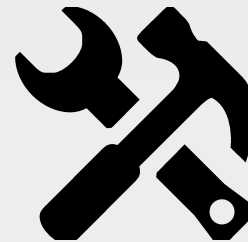
Return to your Workstation

```
[vagrant@load-balancer ~]$ exit
```

```
logout
Connection to 127.0.0.1 closed.
```
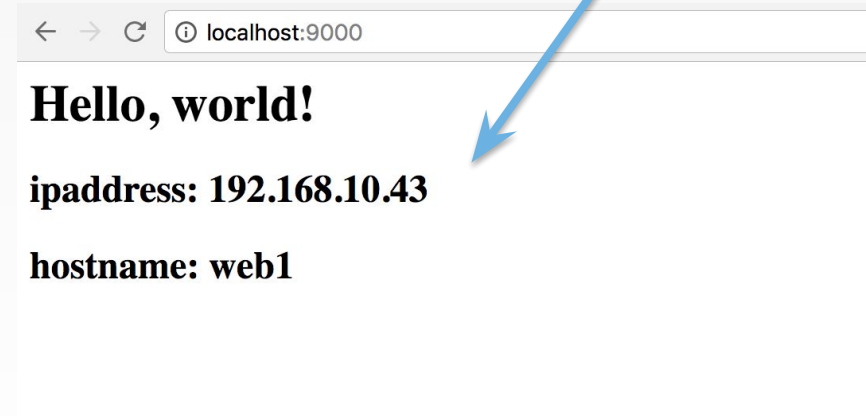
CHEF

Lab: Update Production

✓ Update the environment named production:

`'myhaproxy' cookbook version equal to '1.0.1'`

Hello, world!

ipaddress: 192.168.10.43

hostname: web1

Hello, world!

ipaddress: 192.168.10.43

hostname: web1

The load balancer will only send traffic to web1, since they are both assigned to the production environment.

CHEF

Discussion

What is the benefit of constraining cookbooks to a particular environment?

What are the benefits of **not** constraining cookbooks to a particular environment?