

# Environments

Using Environments to Reflect Organization Patterns and Workflow

# Objectives

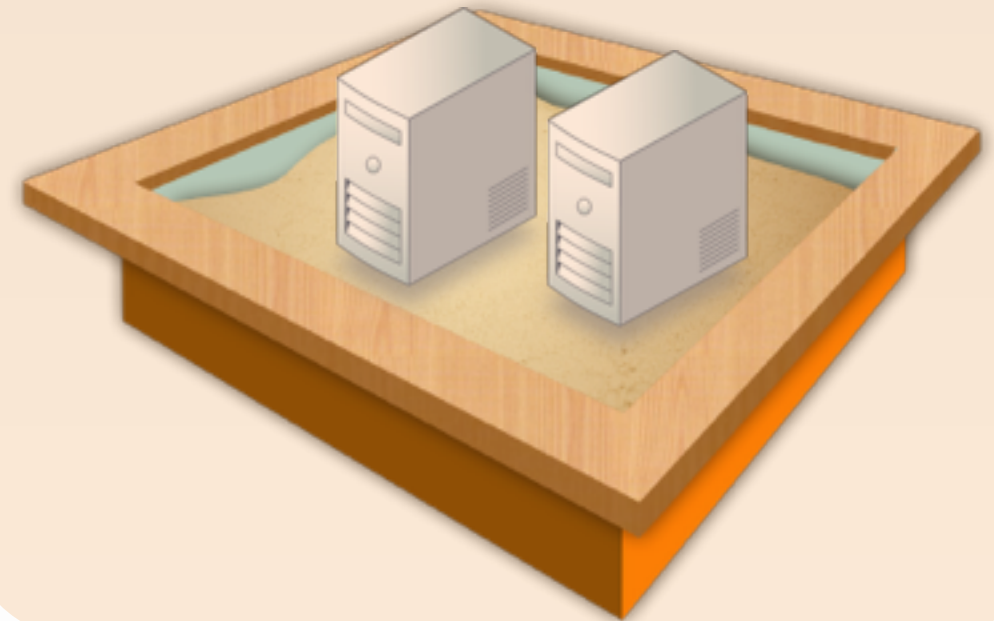
- After completing this module, you should be able to
- Create a production and acceptance environment
  - Deploy a node to an environment
  - Update a search query to be more exact

# Keeping Your Infrastructure Current

Changing Needs  
Changing Software  
Growing Organization  
Increased Website Popularity

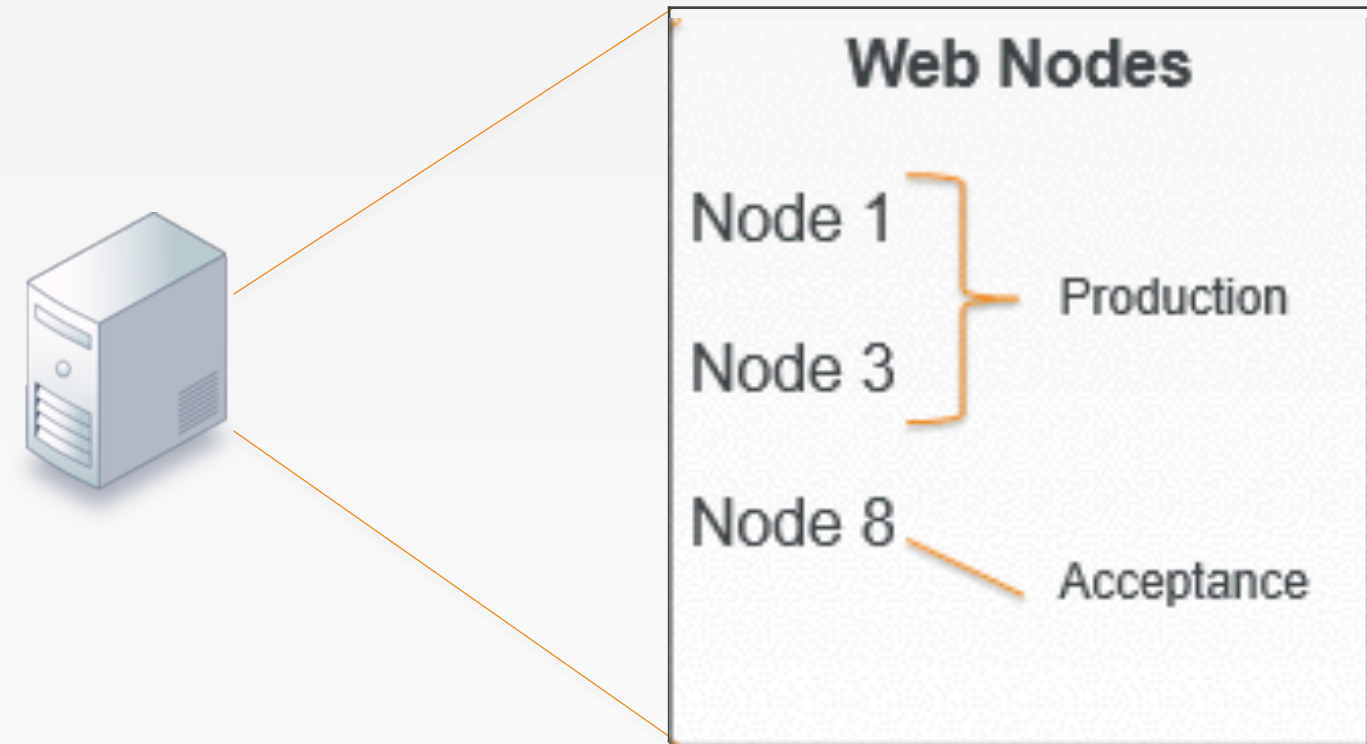
Production

Acceptance



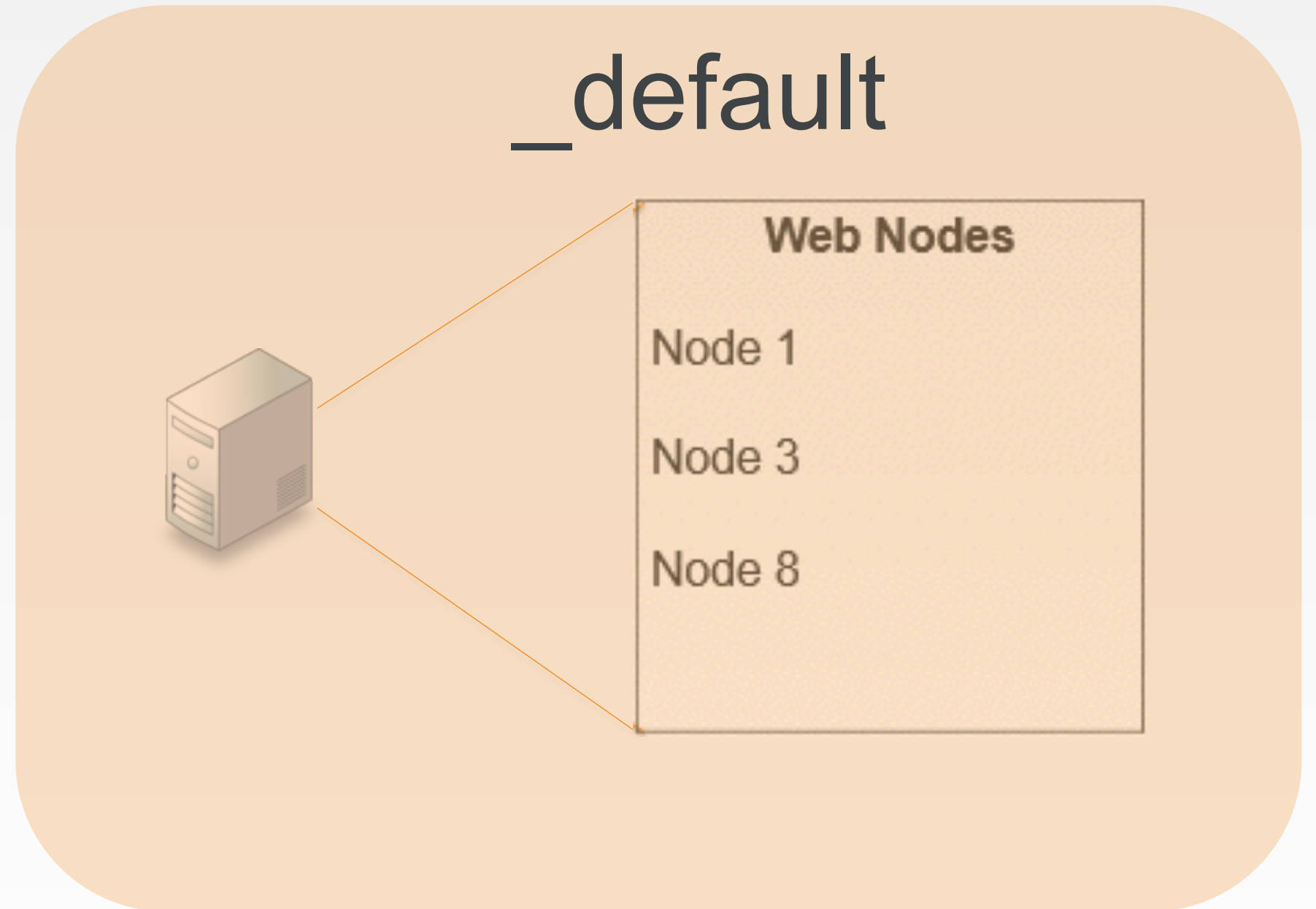
# Environments

Environments can define different functions of nodes that live on the same system.



# Environments

Every organization or infrastructure starts with the `_default` environment.





## GL: Production

*Let's create a reliable environment for our nodes.*

### Objective:

- ❑ Deploy Our Site to Production

## GL: Using 'knife environment --help'



```
$ cd ~/chef-repo  
$ knife environment --help
```

```
** ENVIRONMENT COMMANDS **
```

```
knife environment compare [ENVIRONMENT..] (options)
```

```
knife environment create ENVIRONMENT (options)
```

```
knife environment delete ENVIRONMENT (options)
```

```
knife environment edit ENVIRONMENT (options)
```

```
knife environment from file FILE [FILE..] (options)
```

```
knife environment list (options)
```

```
knife environment show ENVIRONMENT (options)
```

# GL: View List of Defined Environments



```
$ knife environment list
```

```
_default
```



# GL: Viewing the \_default Environment



```
$ knife environment show _default
```

```
chef_type:          environment
cookbook_versions:
default_attributes:
description:        The default Chef environment
json_class:         Chef::Environment
name:               _default
override_attributes:
```

# GL: Searching All of Our Nodes



```
$ knife search node "*:*"
```

```
3 items found
```

```
Node Name:    node1
```

```
Environment:  _default
```

```
FQDN:         ip-172-31-8-68.ec2.internal
```

```
IP:           54.175.46.24
```

```
Run List:     role[web]
```

```
Roles:        web
```

```
Recipes:      apache, apache::default, apache::server
```

```
Platform:     centos 6.7
```

```
Tags:
```

# GL: Create an environments Directory



```
$ mkdir environments
```

# GL: Create a New Environment File

```
~/chef-repo/environments/production.rb
```

```
name 'production'  
description 'Where we run production code'  
  
cookbook 'apache', '= 0.2.1'  
cookbook 'myhaproxy', '= 1.0.0'
```

# GL: Upload the production.rb File



```
$ knife environment from file production.rb
```

```
Updated Environment production
```

# GL: View the List of Environments



```
$ knife environment list
```

```
_default
```

```
production
```

# GL: View the Production Environment



```
$ knife environment show production
```

```
chef_type:          environment
cookbook_versions:
  apache:           = 0.2.1
  myhaproxy:        = 1.0.0
default_attributes:
description:         Where we run production code
json_class:          Chef::Environment
name:                production
override_attributes:
```

# GL: Viewing 'knife node --help'



```
$ knife node --help
```

```
** NODE COMMANDS **
```

```
knife node bulk delete REGEX (options)
```

```
knife node create NODE (options)
```

```
knife node delete NODE (options)
```

```
knife node edit NODE (options)
```

```
knife node environment set NODE ENVIRONMENT
```

```
knife node from file FILE (options)
```

```
knife node list (options)
```

```
knife node run_list add [NODE] [ENTRY[,ENTRY]] (options)
```

```
knife node run_list remove [NODE] [ENTRY[,ENTRY]] (options)
```

```
knife node run_list set NODE ENTRIES (options)
```

```
knife node show NODE (options)
```



# GL: Viewing 'knife node set --help'



```
$ knife node environment set --help
```

```
knife node environment set NODE ENVIRONMENT
```

```
  -s, --server-url URL           Chef Server URL
    --chef-zero-host HOST        Host to start chef-zero on
    --chef-zero-port PORT        Port (or port range) to start chef-zero on.
Port ranges like 1000,1010 or 8889-9999 will try all given ports until one works.
  -k, --key KEY                  API Client Key
    --[no-]color                 Use colored output, defaults to false on
Windows, true otherwise
  -c, --config CONFIG            The configuration file to use
    --defaults                   Accept default values for all questions
  -d, --disable-editing          Do not open EDITOR, just accept the data as is
  -e, --editor EDITOR            Set the editor to use for interactive commands
```

# GL: Using 'knife environment node set'



```
$ knife node environment set node1 production
```

```
node1:  
  chef_environment: production
```

# GL: Viewing node1's Attributes



```
$ knife node show node1
```

```
Node Name:    node1
```

```
Environment:  production
```

```
FQDN:         ip-172-31-8-68.ec2.internal
```

```
IP:           54.175.46.24
```

```
Run List:     role[web]
```

```
Roles:        web
```

```
Recipes:      apache, apache::default, apache::server
```

```
Platform:     centos 6.7
```

```
Tags:
```



## Lab: Set More Nodes to Production

- ❑ Set node2's environment to production

## Lab: Set node2's Environment to Production



```
$ knife node environment set node2 production
```

```
node2:  
  chef_environment: production
```

## Lab: Verify node2 is Set to Production



```
$ knife node show node2
```

```
Node Name:    node2
Environment:  production
FQDN:         ip-172-31-0-128.ec2.internal
IP:           54.210.192.12
Run List:     role[load_balancer]
Roles:        load_balancer
Recipes:      myhaproxy, myhaproxy::default, haproxy::default,
haproxy::install_package
Platform:     centos 6.6
Tags:
```



## Lab: Set More Nodes to Production

- ✓ Set node2's environment to production



# Production

*Let's create a reliable environment for our nodes.*

## Objective:

- ✓ Deploy our site to Production





## Lab: Acceptance Environment

- ❑ Create an environment named "acceptance" that has no cookbook restrictions.
- ❑ Move node3 into the acceptance environment
- ❑ Run chef-client on all the nodes

## Lab: Create a New Environment File

```
~/chef-repo/environments/acceptance.rb
```

```
name 'acceptance'  
description 'Where code and apps are tested'  
# No Cookbook Restrictions
```

# Lab: Upload the .rb File



```
$ knife environment from file acceptance.rb
```

```
Updated Environment acceptance
```

## Lab: Verify that the Environment was Set



```
$ knife environment list
```

```
_default  
production  
acceptance
```

# Lab: Verify the Contents of the Environment



```
$ knife environment show acceptance
```

```
chef_type:          environment
cookbook_versions:
default_attributes:
description:         Where code and applications are tested
json_class:          Chef::Environment
name:                acceptance
override_attributes:
```

## Lab: Set node 3 to the Acceptance Environment



```
$ knife node environment set node3 acceptance
```

```
node3:  
  chef_environment: acceptance
```

## Lab: Verify that the Environment Was Set



```
$ knife node show node3
```

```
Node Name: node3
```

```
Environment: acceptance
```

```
FQDN: ip-172-31-0-127.ec2.internal
```

```
IP: 54.210.86.164
```

```
Run List: role[web]
```

```
Roles: web
```

```
Recipes: apache, apache::default, apache::server
```

```
Platform: centos 6.6
```

```
Tags:
```

## Lab: Converge All the Nodes



```
$ knife ssh "*" "*" -x USER -P PWD "sudo chef-client"
```

```
ec2-54-175-46-24.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-192-12.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list: ["apache"]
ec2-54-210-192-12.compute-1.amazonaws.com resolving cookbooks for run list: ["myhaproxy"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
ec2-54-210-192-12.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-192-12.compute-1.amazonaws.com   - build-essential
```





## Lab: Acceptance Environment

- ✓ Create an environment named "acceptance" that has no cookbook restrictions.
- ✓ Move node3 into the acceptance environment
- ✓ Run chef-client on all the nodes



# 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

```
#  
# 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: Separate Environments

- ✓ Use Search to separate out the environments
- ❑ Update myhaproxy version number



# 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', '~> 1.6.6'
```

# 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 (1.6.6)  
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 (1.6.6) (frozen)
```

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



## GL: Separate Environments

- ✓ Use Search to separate out the environments
- ✓ Update myhaproxy version number



## A Brief Recap

We restricted the production environment to specific cookbook version.

We created an acceptance environment with no cookbook restrictions.

We set specific nodes to each of these environments.

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

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



## Lab: Update Production

- ❑ Update the environment named production:

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

# Lab: Update production.rb

```
~/chef-repo/environments/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:
```

# Lab: Converge All Nodes



```
$ knife ssh "*:*" -x USER -P PWD "sudo chef-client"
```

```
ec2-54-175-46-24.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-192-12.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list: ["apache"]
ec2-54-210-192-12.compute-1.amazonaws.com resolving cookbooks for run list: ["myhaproxy"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
ec2-54-210-192-12.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-192-12.compute-1.amazonaws.com   - build-essential
```



## Lab: Update Production

✓ Update the environment named production:

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

# DISCUSSION



## Discussion

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

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



**CHEF**™

---