# Testing While Refactoring to Attributes





CHEF

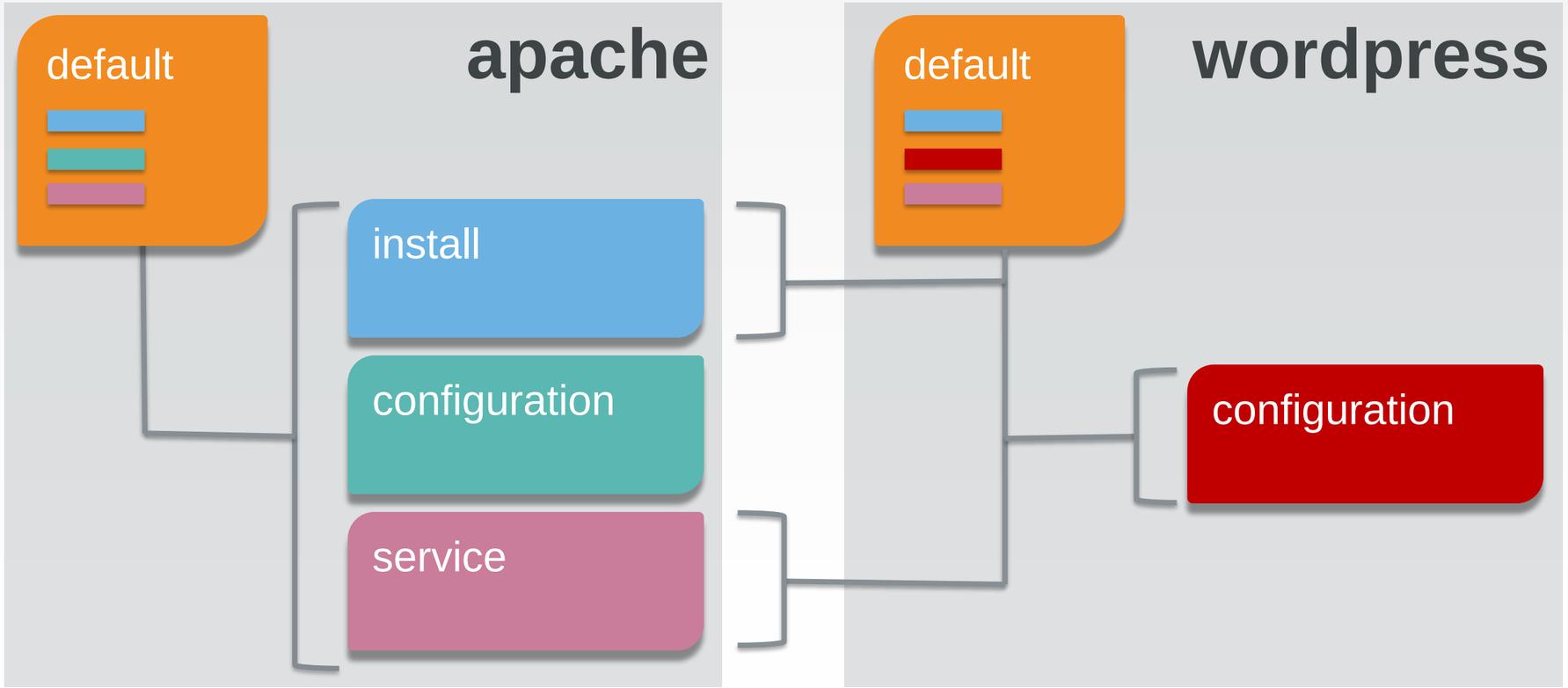
# Objectives

After completing this module, you should be able to:

- Refactor resources to use attributes
- Use Pry to explore the current state of execution
- Make changes to your recipes with confidence

CONCEPT

# Modular Cookbook Recipes



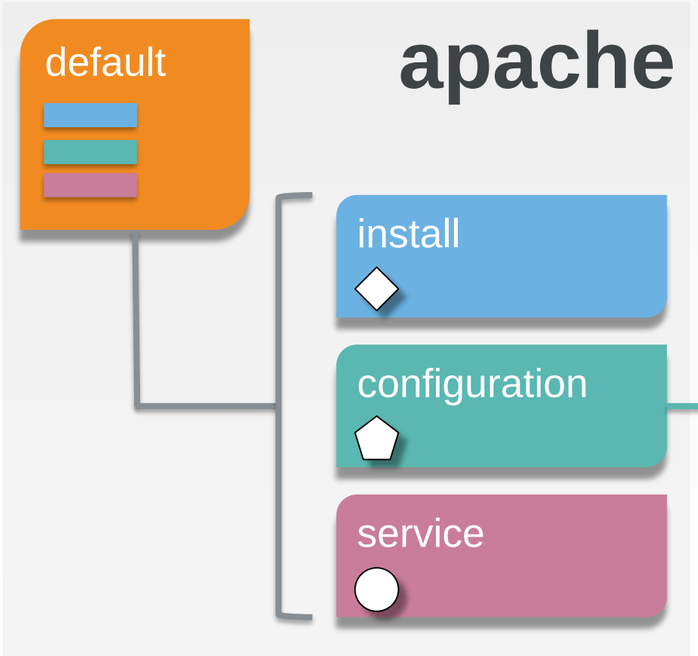





CHEF

CONCEPT

# Modular Cookbook Recipes




~/apache/recipes/configuration.rb

```
file '/var/www/html/index.html' do
  content '<h1>Welcome Home</h1>'
end
```

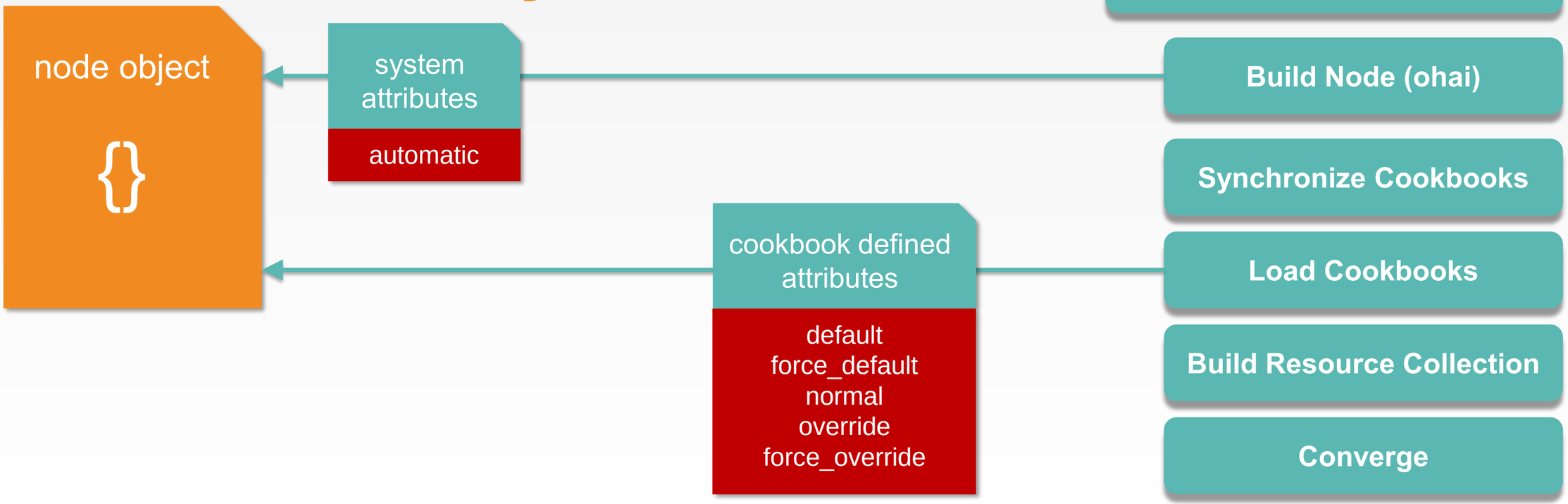




CHEF

CONCEPT

# The Node Object



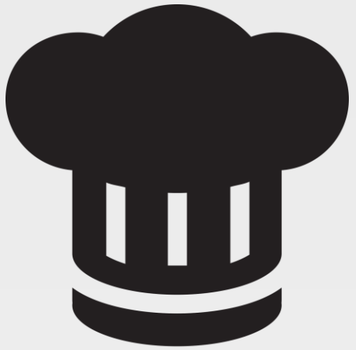

https://docs.chef.io/attributes.html - attribute-precedence





CHEF

EXERCISE



# Refactor to Use Attributes

*Time to remove all the hard-coded values and make them attributes.*

## Objective:

- ❑ Refactor the Install recipe to use a Node attribute
- ❑ Execute the tests and verify the tests fail
- ❑ Create the attributes file and add the Node attribute
- ❑ Execute the tests and verify the tests pass





CHEF

# Replace the Value with a Node Attribute

~/apache/recipes/install.rb

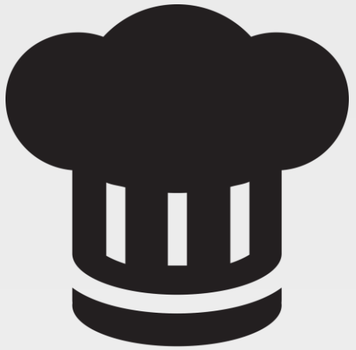
```
#
# Cookbook Name:: apache
# Recipe:: install
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
package node['apache']['package_name']
```





CHEF

EXERCISE



# Refactor to Use Attributes

*A change means a chance for us to run the tests!*

## Objective:

- ✓ Refactor the Install recipe to use a Node attribute
- ❑ Execute the tests and verify the tests fail
- ❑ Create the attributes file and add the Node attribute
- ❑ Execute the tests and verify the tests pass





CHEF

# Execute the Tests to See it Fail

```
> chef exec rspec
```

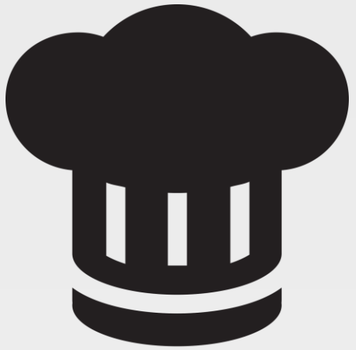
```
..FFFFFF...

Failures:

  1) apache::default When all attributes are default, on an Centos
6.9 converges successfully
     Failure/Error: expect { chef_run }.to_not raise_error

       expected no Exception, got #<NoMethodError: undefined
method `[]' for nil:NilClass> with backtrace:
         # /tmp/chefspec20180313-21260-
```





CHEF

EXERCISE



# Refactor to Use Attributes

*We definitely broke it! Now, let's fix it.*

## Objective:

- ✓ Refactor the Install recipe to use a Node attribute
- ✓ Execute the tests and verify the tests fail
- ❑ Create the attributes file and add the Node attribute
- ❑ Execute the tests and verify the tests pass

# Ask Chef How to Generate an Attributes File

```
> chef generate attribute --help
```

```
Usage: chef generate attribute [path/to/cookbook] NAME [options]
    -C, --copyright COPYRIGHT        Name of the copyright hol...
    -m, --email EMAIL                Email address of the auth...
    -a, --generator-arg KEY=VALUE    Use to set arbitrary      ...
    -I, --license LICENSE            all_rights, apache2, mit,...
    -g GENERATOR_COOKBOOK_PATH,      Use GENERATOR_COOKBOOK_PA...
        --generator-cookbook
```

# Use Chef to Generate a Default Attributes File

```
> chef generate attribute default
```

```
Compiling Cookbooks...
Recipe: code_generator::attribute
  * directory[/home/chef/apache/attributes] action create
    - create new directory /home/chef/apache/attributes
  * template[/home/chef/apache/attributes/default.rb] action
create
    - create new file /home/chef/apache/attributes/default.rb
    - update content in file
/home/chef/apache/attributes/default.rb from none to e3b0c4
    (diff output suppressed by config)
```





CHEF

# View the Attributes File Generated

```
> tree attributes
```

```
attributes
└── default.rb

0 directories, 1 file
```





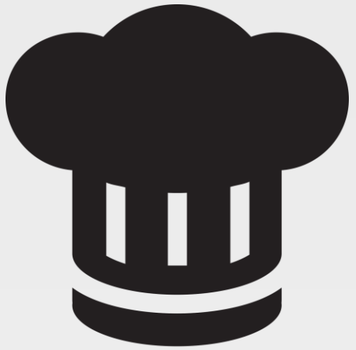
CHEF

# Add the Default Node Attribute

~/apache/attributes/default.rb

```
default['apache']['package_name'] = 'httpd'
```

EXERCISE



# Refactor to Use Attributes

*The work is done. Let's hope it's the right work. Run the tests!*

## Objective:

- ✓ Refactor the Install recipe to use a Node attribute
- ✓ Execute the tests and verify the tests fail
- ✓ Create the attributes file and add the Node attribute
- ❑ Execute the tests and verify the tests pass





CHEF

# Execute the Tests to See it Pass

```
> chef exec rspec
```

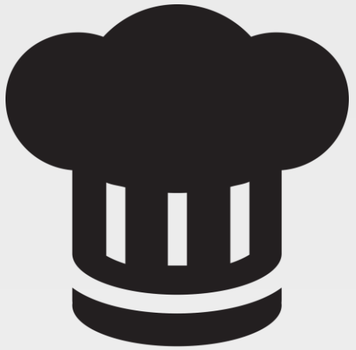
```
...........

Finished in 4.07 seconds (files took 3.93 seconds to load)
11 examples, 0 failures
```





CHEF

EXERCISE



# Refactor to Use Attributes

*We made a change and we know it works!*

## Objective:

- ✓ Refactor the Install recipe to use a Node attribute
- ✓ Execute the tests and verify the tests fail
- ✓ Create the attributes file and add the Node attribute
- ✓ Execute the tests and verify the tests pass





CHEF

PROBLEM

# What if We Made a Typo?

While implementing the node attribute what if made a mistake?





CHEF

# Typos Like This One Will Waste Time

~/apache/attributes/default.rb

```
default['apche']['package_name'] = 'httpd'
```





CHEF

CONCEPT

# Mental Model vs Actual Model

Faster feedback helps us build a greater mental model of the actual execution model. Tests that we define help strengthen it. However, tests are not very interactive as they are more like experiments. What we want is the ability to pause execution and look around.

CONCEPT

# Pry a Debugger

Pry is a Ruby debugger that allows you to define break points. These breakpoints allow you to pause operation and interact with the current process being able to interrogate the current state of the system.

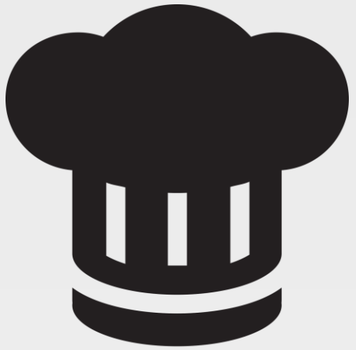
http://pryrepl.org





CHEF

EXERCISE



# Setup a Break Point

*Time to make trouble for ourselves.*

## Objective:

- ❑ Mutate the code and add the breakpoint
- ❑ Execute the tests to cause the breakpoint to trigger
- ❑ Remove the breakpoint and restore the code

# Create a Typo in the Defined Attribute

~/apache/attributes/default.rb

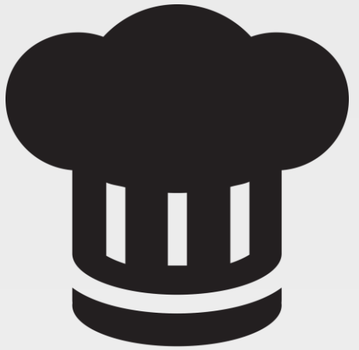
```
default['apche']['package_name'] = 'httpd'
```

# Add a Break Point in the Recipe

~/apache/recipes/install.rb

```
#
# Cookbook Name:: apache
# Recipe:: install
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
require 'pry'
binding.pry


package node['apache']['package_name']
```

EXERCISE



# Setup a Break Point

*Time to pry into the code and see what it is going on.*

## Objective:

- ✓ Mutate the code and add the breakpoint
- ❑ Execute the tests to cause the breakpoint to trigger
- ❑ Remove the breakpoint and restore the code





CHEF

# Execute the Test to Initiate Pry

```
> chef exec rspec spec/unit/recipes/install_spec.rb
```

```
From: /tmp/chefspec20180313-23174-
grz9vbfile_cache_path/cookbooks/apache/recipes/install.rb @ line 9
Chef::Mixin::FromFile#from_file:

    4: #
    5: # Copyright:: 2018, The Authors, All Rights Reserved.
    6: require 'pry'
    7: binding.pry
    8:
 => 9: package node['apache']['package_name']

    # ... CONTINUES ON THE NEXT SLIDE ...
```





CHEF

# Pry Provides an Interactive Prompt

```
> chef exec rspec spec/unit/recipes/install_spec.rb
```

```
    # ... CONTINUED FROM THE PREVIOUS SLIDE ...
    4: #
    5: # Copyright:: 2018, The Authors, All Rights Reserved.
    6: require 'pry'
    7: binding.pry
    8:
 => 9: package node['apache']['package_name']

[1] pry(#<Chef::Recipe>)>
```





CHEF

# Ask Pry for Help

```
[1] pry(#<Chef::Recipe>)> help
```

```
Help
  help                Show a list of commands or information about a specific command.

Context
  cd                  Move into a new context (object or scope).
  find-method         Recursively search for a method within a class/module or the curr...
  ls                  Show the list of vars and methods in the current scope.
  pry-backtrace       Show the backtrace for the pry session.
  raise-up            Raise an exception out of the current pry instance.
  reset               Reset the repl to a clean state.
```

To escape the help menu, type in q

# Execute Any Code As You Would in a Recipe

```
[2] pry(#<Chef::Recipe>)> node['apache']
```

```
=> nil
```





CHEF

# Explore the Different Node Attributes

```
[3] pry(#<Chef::Recipe>)> node['apche']
```

```
=> {"package_name"=>"httpd"}
```





CHEF

# Halt the Execution of the Test Immediately

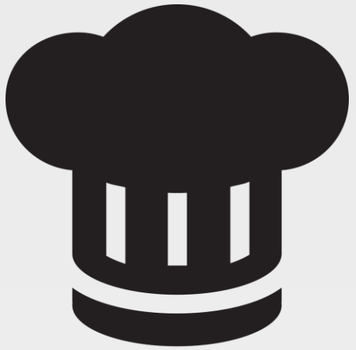
```
[4] pry(#<Chef::Recipe>)> exit!
```





CHEF

EXERCISE



# Setup a Break Point

*Time to pry into the code and see what it is going on.*

## Objective:

- ✓ Mutate the code and add the breakpoint
- ✓ Execute the tests to cause the breakpoint to trigger
- ❑ Remove the breakpoint and restore the code





CHEF

# Remove the Break Point from the Recipe

~/apache/recipes/install.rb

```
#
# Cookbook Name:: apache
# Recipe:: install
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
require 'pry'
binding.pry

package node['apache']['package_name']
```





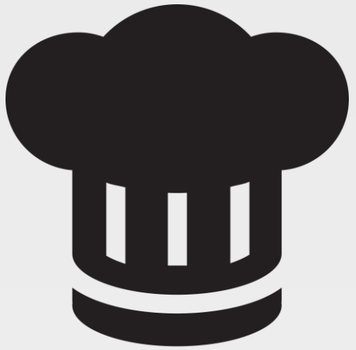
CHEF

# Fix the Change in the Attributes

~/apache/attributes/default.rb

```
default['apache']['package_name'] = 'httpd'                    +
```

EXERCISE



# Setup a Break Point

*Time to pry into the code and see what it is going on.*

## Objective:

- ✓ Mutate the code and add the breakpoint
- ✓ Execute the tests to cause the breakpoint to trigger
- ✓ Remove the breakpoint and restore the code





CHEF

LAB

# Refactor Remaining Resources

- ❑ Refactor the resource to use a Node attribute
- ❑ Execute the tests and verify the tests fail
- ❑ Add the new Node attribute
- ❑ Execute the tests and verify the tests pass

*BONUS: Use pry to verify that the attribute has been set.*

❖ **Repeat this series of steps for the configuration recipe and service recipe**





CHEF

# Update the Recipe to use the Node Attribute

~/apache/recipes/service.rb

```
#
# Cookbook Name:: apache
# Recipe:: service
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
service node['apache']['service_name'] do
  action [:enable, :start]
end
```

# Execute the Tests to See it Fail

```
> chef exec rspec spec/unit/recipes/service_spec.rb
```

```
FFF


Failures:


  1) apache::service When all attributes are default, on an Centos 6.9
converges successfully
     Failure/Error: expect { chef_run }.to_not raise_error


       expected no Exception, got #<ArgumentError: You must supply a name when
declaring a service resource> with backtrace:
         # /tmp/chefspec20180313-17746-
14gwtwpfile_cache_path/cookbooks/apache/recipes/service.rb:6:in `from_file'
```





CHEF

# Add the Default Node Attribute

~/apache/attributes/default.rb

```
default['apache']['package_name'] = 'httpd'
default['apache']['service_name'] = 'httpd'
```

# Execute the Tests to See it Pass

```
> chef exec rspec spec/unit/recipes/service_spec.rb
```

```
...

Finished in 1.06 seconds (files took 4.33 seconds to load)
3 examples, 0 failures
```

# Update the Recipe to use the Node Attribute

~/apache/recipes/configuration.rb

```
#
# Cookbook:: apache
# Recipe:: configuration
#
# Copyright:: 2018, The Authors, All Rights Reserved.
file node['apache']['default_index_html'] do
  content '<h1>Welcome Home!</h1>'
end
```





CHEF

# Execute the Tests to See it Fail

```
> chef exec rspec spec/unit/recipes/configuration_spec.rb
```

```
FF

Failures:

  1) apache::configuration When all attributes are default, on an Centos 6.9
conve
     Failure/Error: expect { chef_run }.to_not raise_error

       expected no Exception, got #<ArgumentError: You must supply a name when
decurce> with backtrace:
```





CHEF

# Add the Default Node Attribute

~/apache/attributes/default.rb

```
default['apache']['package_name'] = 'httpd'
default['apache']['service_name'] = 'httpd'
default['apache']['default_index_html'] = '/var/www/html/index.html'
```

# Execute the Tests to See it Pass

```
> chef exec rspec spec/unit/recipes/configuration_spec.rb
```

```
..

Finished in 1.14 seconds (files took 4.03 seconds to load)
2 examples, 0 failures
```

LAB

# Refactor Remaining Resources

- ✓ Refactor the resource to use a Node attribute
- ✓ Execute the tests and verify the tests fail
- ✓ Add the new Node attribute
- ✓ Execute the tests and verify the tests pass

*BONUS: Use pry to verify that the attribute has been set.*

❖ **Repeat this series of steps for the configuration recipe and service recipe**





CHEF

# Discussion

What are the benefits of providing the package name and service name as node attributes?

What value does Pry provide to you as a Cookbook Developer?

DISCUSSION

# Q&A

What questions can we answer for you?





CHEF

# Morning

Introduction

Why Write Tests? Why is that Hard?

Writing a Test First

Refactoring Cookbooks with Tests

# Afternoon

Faster Feedback with Unit Testing

Testing Resources in Recipes

Refactoring to Attributes

**Refactoring to Multiple Platforms**





CHEF

CHEF™