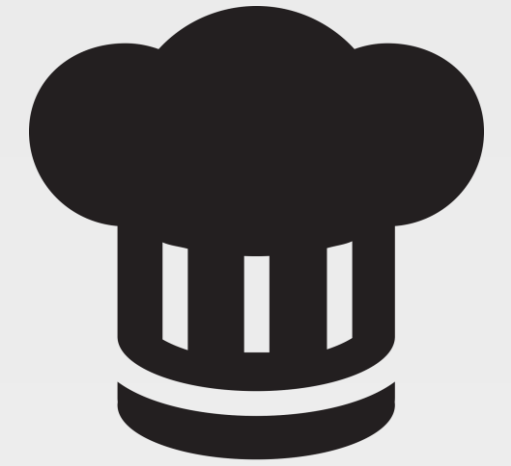# Creating a Custom Resource

Objectives

After completing this module, you should be able to:

➢ Create a custom resource file

➢ Define a custom resource action

➢ Extract Chef resources into a custom resource action implementation

➢ Create custom resource properties

EXERCISE

Review the Cookbook

We need to make sure everything works before we get started.

Objective:

❑ Checkout different branch of cookbook

❑ Review and execute the integration tests

❑ Review and execute the unit tests

❑ Review the default recipe

CHEF

Move in the apache cookbook

```
> cd ~/apache
```

Add the Changed Files to Staging

```
> git add .
```

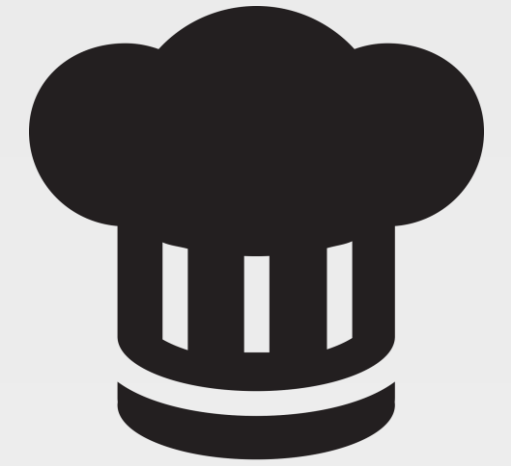# Commit the Staged Changes

```
> git commit -m "Saving TDD work"
```

Change Git Branches

```
> git checkout extending-cookbook
```

EXERCISE

Review the Cookbook

We need to make sure everything works before we get started.

Objective:

✓ Checkout different branch of cookbook

❑ Review and execute the integration tests

❑ Review and execute the unit tests

❑ Review the default recipe

Reviewing the Existing Integration Tests

`~/apache/test/smoke/default/default_test.rb`

```ruby
describe command('curl http://localhost') do
  its(:stdout) { should match(/Welcome home/) }
 end


describe command('curl http://localhost:8080') do
  its(:stdout) { should match(/Welcome admins/) }
end
```

Executing the Existing Integration Tests

```
> kitchen verify
```

```
Target:  ssh://vagrant@127.0.0.1:2222


  ✔   Command curl http://localhost stdout should match /Welcome
home/
  ✔   Command curl http://localhost:8080 stdout should match
/Welcome admins/


Summary: 2 successful, 0 failures, 0 skipped
      Finished verifying <default-centos-67> (0m0.74s).
-----> Kitchen is finished. (0m7.37s)
```
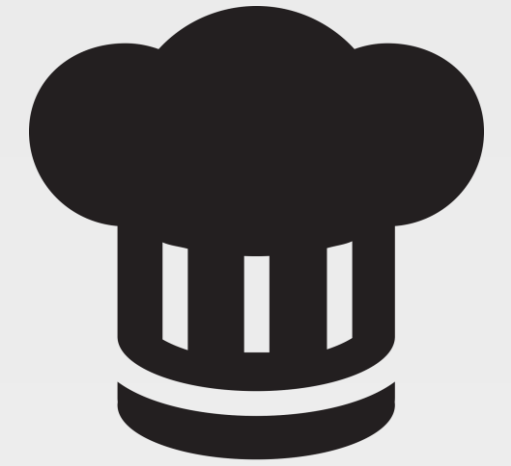
EXERCISE
Review the Cookbook

We need to make sure everything works before we get started.

Objective:

- ✓ Checkout different branch of cookbook
- ✓ Review and execute the integration tests
- ❑ Review and execute the unit tests
- ❑ Review the default recipe

CHEF

Reviewing the Existing Unit Tests

~/apache/spec/unit/recipes/default_spec.rb

```ruby
require 'spec_helper'


describe 'apache::default' do
 context 'When all attributes are default, on an CentOS 6.9' do
    let(:chef_run) do
      runner = ChefSpec::ServerRunner.new(platform: 'centos', version: '6.9')
      runner.converge(described_recipe)
    end
    it 'converges successfully' do
      expect { chef_run }.to_not raise_error
    end


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

Reviewing the Existing Unit Tests

~/apache/spec/unit/recipes/default_spec.rb

```
# ... CONTINUES FROM THE PREVIOUS SLIDE ...


    it 'installs the necessary package' do
      expect(chef_run).to install_package('httpd')
    end


    it 'starts the necessary service' do
      expect(chef_run).to start_service('httpd')
    end


    it 'enables the necessary service' do
      expect(chef_run).to enable_service('httpd')
    end
# ... CONTINUES ON THE NEXT SLIDE ...
```

CHEF

Reviewing the Existing Unit Tests

~/apache/spec/unit/recipes/default_spec.rb

```
# ... CONTINUES FROM THE PREVIOUS SLIDE ...


  describe 'for the default site' do
    it 'writes out a new home page' do
      expect(chef_run).to render_file('/var/www/html/index.html').with_content('<h1>Welcome
home!</h1>')
    end
  end


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

CHEF

Reviewing the Existing Unit Tests

`~/apache/spec/unit/recipes/default_spec.rb`

```
# ... CONTINUES FROM THE PREVIOUS SLIDE ...
  describe 'for the admin site' do
    it 'creates the directory' do
      expect(chef_run).to create_directory('/srv/apache/admins/html')
    end


    it 'creates the configuration' do
      expect(chef_run).to render_file('/etc/httpd/conf.d/admins.conf')
    end


 it 'creates a new home page' do
        expect(chef_run).to
render_file('/srv/apache/admins/html/index.html').with_content('<h1>Welcome admins!</h1>')
      end
# ... CONTINUES ON THE NEXT SLIDE ...
```

CHEF

Reviewing the Existing Unit Tests

`~/apache/spec/unit/recipes/default_spec.rb`

```
# ... CONTINUES FROM THE PREVIOUS SLIDE ...
    end # describe admin site
  end # context
end # describe 'apache::default'
```
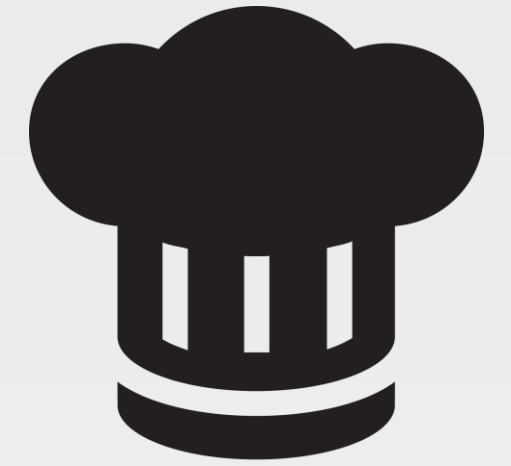
CHEF

Executing the Existing Unit Tests

```
> chef exec rspec
```

```
........

Finished in 0.92866 seconds (files took 2.76 seconds to load)
8 examples, 0 failures
```

EXERCISE

Review the Cookbook

We need to make sure everything works before we get started.

Objective:

- ✓ Checkout different branch of cookbook
- ✓ Review and execute the integration tests
- ✓ Review and execute the unit tests
- ❑ Review the default recipe

Reviewing the Default Recipe

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
package 'httpd'


file '/var/www/html/index.html' do
  content '<h1>Welcome home!</h1>'
end


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

Reviewing the Default Recipe

```ruby
# ... CONTINUES FROM THE PREVIOUS SLIDE ...
directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end

template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
  variables(document_root:'/srv/apache/admins/html', port: 8080)
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end
# ... CONTINUES ON THE NEXT SLIDE ...
```
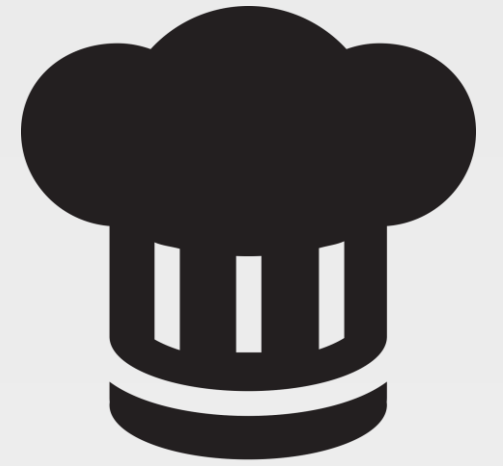
Reviewing the Default Recipe

`~/apache/recipes/default.rb`

```
# ... CONTINUES FROM THE PREVIOUS SLIDE ...

service 'httpd' do
  action [:enable, :start]
end
```
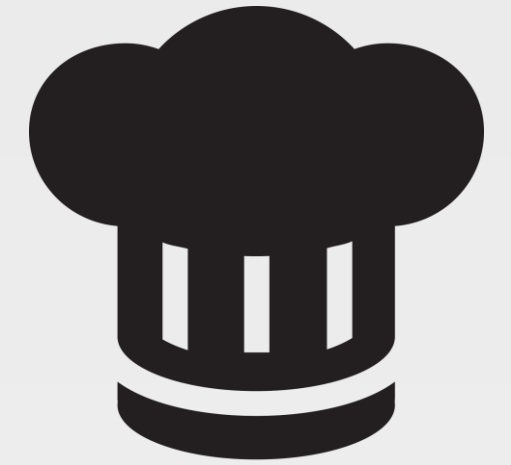
CHEF

EXERCISE
Review the Cookbook

We need to make sure everything works before we get started.

Objective:

- ✓ Checkout different branch of cookbook
- ✓ Review and execute the integration tests
- ✓ Review and execute the unit tests
- ✓ Review the default recipe

EXERCISE

Creating a Custom Resource

This will make our recipe much cleaner.

Objective:

❑ Create a custom resource that create the admin site
❑ Allow the custom resource to have configurable properties
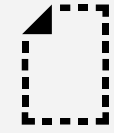
11-23

Generating a Custom Resource

```
> chef generate lwrp vhost
```

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

CHEF

Defining the Create Action
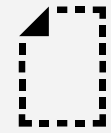
```
action :create do


end
```

Implementing the Create Action

```ruby
action :create do
  directory '/srv/apache/admins/html' do
    recursive true
    mode '0755'
  end

  template '/etc/httpd/conf.d/admins.conf' do
    source 'conf.erb'
    mode '0644'
    variables(document_root: '/srv/apache/admins/html',port: 8080)
    notifies :restart, 'service[httpd]'
  end

  file '/srv/apache/admins/html/index.html' do
    content '<h1>Welcome admins!</h1>'
  end
end
```

CHEF

Refactoring the Default Recipe

~/apache/recipes/default.rb

```
#
# Cookbook Name:: apache
# Recipe:: default
#
# Copyright (c) 2017 The Authors, All Rights Reserved.
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome home!</h1>'
end

directory '/srv/apache/admins/html' do
  recursive true
  mode '0755'
end
```

CHEF

Refactoring the Default Recipe

`~/apache/recipes/default.rb`

```ruby
template '/etc/httpd/conf.d/admins.conf' do
  source 'conf.erb'
  mode '0644'
  variables(
    document_root: '/srv/apache/admins/html',
    port: 8080
  )
  notifies :restart, 'service[httpd]'
end

file '/srv/apache/admins/html/index.html' do
  content '<h1>Welcome admins!</h1>'
end

service 'httpd' do
  action [:enable, :start]
end
```

Adding the New Custom Resource

~/apache/recipes/default.rb

```
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome home!</h1>'
end

apache_vhost 'admins' do
  action :create
end

service 'httpd' do
  action [:enable, :start]
end
```

CHEF

Executing the Unit Tests

```
> chef exec rspec
```

```
Finished in 0.9673 seconds (files took 2.72 seconds to load)
8 examples, 3 failures

Failed examples:

rspec ./spec/unit/recipes/default_spec.rb:39 # apache::default When all attributes are
default, on an unspecified platform for the admin site creates the directory

rspec ./spec/unit/recipes/default_spec.rb:43 # apache::default When all attributes are
default, on an unspecified platform for the admin site creats the configuration

rspec ./spec/unit/recipes/default_spec.rb:47 # apache::default When all attributes are
default, on an unspecified platform for the admin site creates a new home page
```
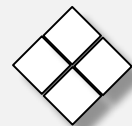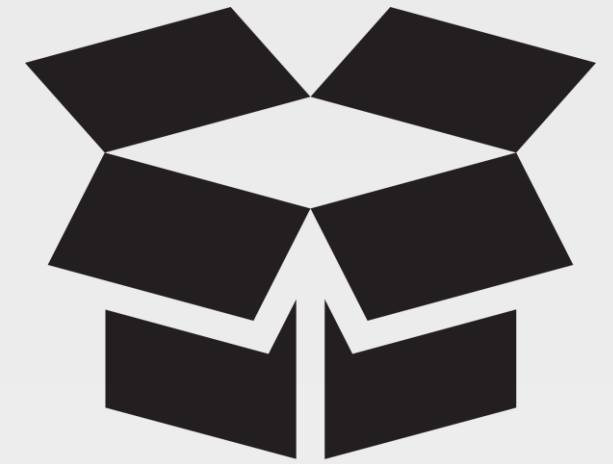
CONCEPT

Resource Collection
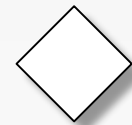
Resource Collection
⬦ other resources ...
◇ template '...'
⬠ directory '...'
◯ file '...'
⬦ other resources ...
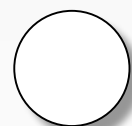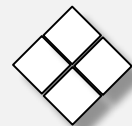
```
> rspec
```

```
This resource is missing from the resource
collection.
```

```
This resource is missing from the resource
collection.
```

```
This resource is missing from the resource
collection.
```

CHEF

CONCEPT

A Sub-Resource Collection

Resource Collection

◈ other resources ...

◇ **apache_vhost**

◈ other resources ...

step into

Sub-Resource Collection

◇ **template '...'**

⬠ **directory '...'**

◯ **file '...'**
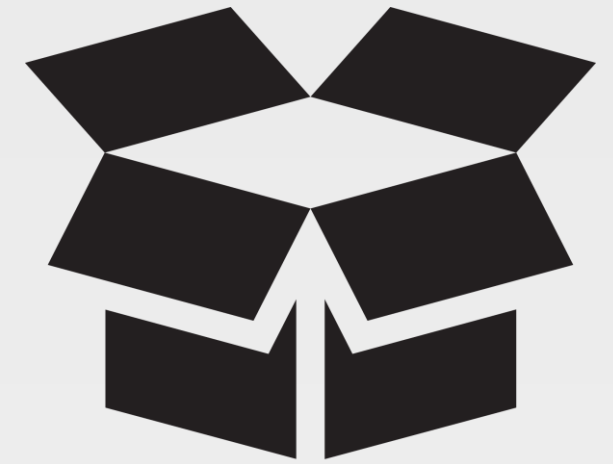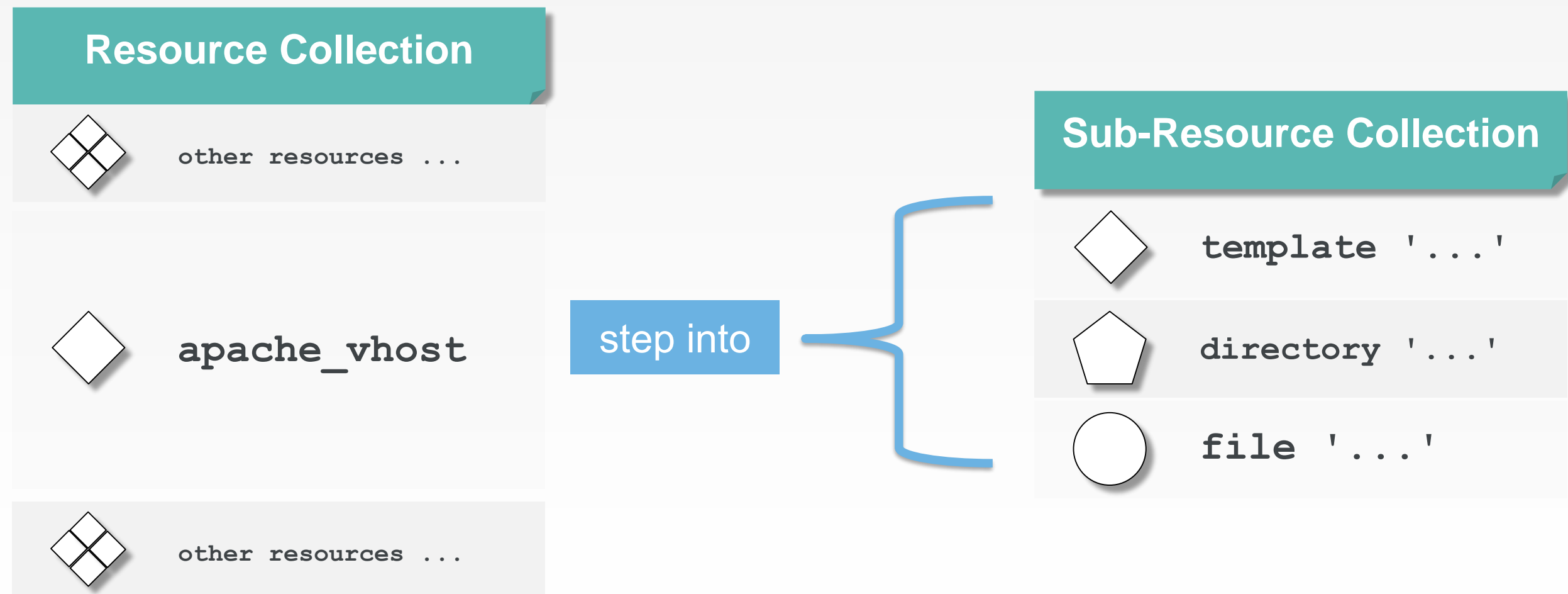
CHEF

Updating the Unit Tests to Verify Custom Resource

~/apache/spec/unit/recipes/default_spec.rb

```
require 'spec_helper'


context 'When all attributes are default, on an CentOS 6.9' do
    let(:chef_run) do
        runner = ChefSpec::ServerRunner.new(platform: 'centos', version: '6.9',
step_into: ['apache_vhost'])
        runner.converge(described_recipe)
    end
    it 'converges successfully' do
        expect { chef_run }.to_not raise_error
    end


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

CHEF

Executing the Unit Tests

```
> chef exec rspec
```

```
........


Finished in 0.98788 seconds (files took 2.95 seconds to load)
8 examples, 0 failures
```

Converging the Test Instance

```
> kitchen converge
```

```
        (up to date)
        (up to date)
           (up to date)
        (up to date)
         * service[apache] action start (up to date)


    Running handlers:
    Running handlers complete
    Chef Client finished, 0/8 resources updated in 05 seconds
    Finished converging <default-centos-67> (0m8.81s).
-----> Kitchen is finished. (0m9.80s)
```

Verifying the Test Instance

```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)
-----> Setting up <default-centos-67>...
       Finished setting up <default-centos-67> (0m0.00s).
-----> Verifying <default-centos-67>...
       Use `/home/chef/apache/test/smoke/default` for testing


Target:  ssh://vagrant@127.0.0.1:2222

  ✔   Command curl http://localhost stdout should match /Welcome home/
  ✔   Command curl http://localhost:8080 stdout should match /Welcome admins/

Summary: 2 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-67> (0m0.70s).
-----> Kitchen is finished. (0m1.82s)
```

CHEF

Run a Complete Test on the Test Instance

```
> kitchen test
```

```
Target:  ssh://vagrant@127.0.0.1:2222

   ✔    Command curl http://localhost stdout should match /Welcome home/
   ✔    Command curl http://localhost:8080 stdout should match /Welcome admins/


Summary: 2 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-67> (0m0.70s).
-----> Destroying <default-centos-67>...
       ==> default: Forcing shutdown of VM...
       ==> default: Destroying VM and associated drives...
       Vagrant instance <default-centos-67> destroyed.
       Finished destroying <default-centos-67> (0m4.09s).
       Finished testing <default-centos-67> (2m21.40s).
-----> Kitchen is finished. (2m22.49s)
```
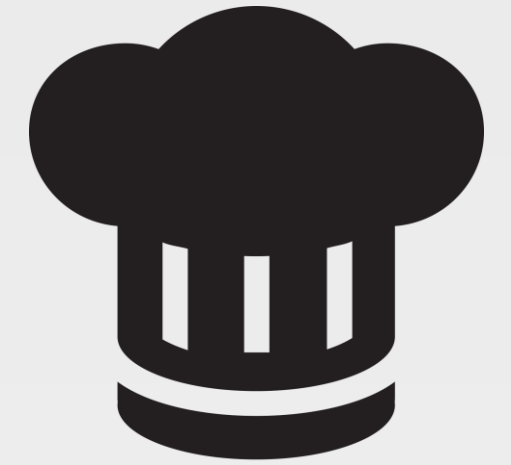
CHEF

EXERCISE

Creating a Custom Resource

This will make our recipe much cleaner.

Objective:

✓ Create a custom resource that create the admin site

❑ Allow the custom resource to have configurable properties

CONCEPT

Resource Properties

A property is defined with the following syntax.

```
~/apache/resources/vhost.rb

property :site_name, String
```

1 **2**

| property method | **1** name (symbol) | **2** type | Optional Parameters |

CONCEPT

Resource Properties

Properties are defined in the resource definition and then available within definition of the resource within the recipe.

~/apache/resources/vhost.rb	~/apache/recipes/default.rb
`property :site_name, String`	```apache_vhost 'admins' do site_name 'admins' action :create end```

Defining a Property to Manage the site_name

📄 `~/apache/resources/vhost.rb`

```ruby
property :site_name, String

action :create do
  directory "/srv/apache/#{new_resource.site_name}/html" do
    recursive true
    mode "0755"
  end

  template "/etc/httpd/conf.d/#{new_resource.site_name}.conf" do
    source "conf.erb"
    mode "0644"
    variables(document_root: "/srv/apache/#{new_resource.site_name}/html", port: 8080)
    notifies :restart, "service[httpd]"
  end

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

CHEF

Updating the Action to use the Property

`~/apache/resource/vhost.rb`

```ruby
action :create do
  directory "/srv/apache/#{new_resource.site_name}/html" do
    recursive true
    mode "0755"
  end

  template "/etc/httpd/conf.d/#{new_resource.site_name}.conf" do
    source "conf.erb"
    mode "0644"
    variables(document_root: "/srv/apache/#{new_resource.site_name}/html", port: 8080)
    notifies :restart, "service[httpd]"
  end

  file "/srv/apache/#{new_resource.site_name}/html/index.html" do
    content "<h1>Welcome #{new_resource.site_name}!</h1>"
  end
end
```

CHEF

Updating the Resource to use the Property

~/apache/recipes/default.rb

```
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome home!</h1>'
end


apache_vhost 'admins' do
  site_name 'admins'
  action :create
end


service 'httpd' do
  action [:enable, :start]
end
```

Executing the Unit Tests

```
> chef exec rspec
```

```
........


Finished in 1.22 seconds (files took 7.58 seconds to load)
8 examples, 0 failures
```

Converging and Verifying the Test Instance

```
> kitchen converge && kitchen verify
```

```
-----> Verifying <default-centos-67>...
       Use `/home/chef/apache/test/smoke/default` for testing


Target:  ssh://vagrant@127.0.0.1:2222

  ✔    Command curl http://localhost stdout should match /Welcome home/
  ✔    Command curl http://localhost:8080 stdout should match /Welcome admins/

Summary: 2 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-67> (0m0.77s).
-----> Kitchen is finished. (2m42.37s)
```

CHEF

EXERCISE

Creating a Custom Resource

That's much better.

Objective:

✓ Create a custom resource that create the admin site

✓ Allow the custom resource to have configurable properties

CHEF

LAB

apache_vhost - site_port Property

❑ Create a custom resource property named **site_port** that is a *Integer*

❑ Within the apache_vhost's create action replace the hard-coded value 8080 with the **site_port** property

❑ Within the default recipe set the apache_vhost resource named 'admins' to have a site_port 8080

Defining a Property to Manage the site_port

~/apache/resource/vhost.rb

```
property :site_name, String
property :site_port, Integer


action :create do
  directory "/srv/apache/#{new_resource.site_name}/html" do
    recursive true
    mode "0755"
  end


  template "/etc/httpd/conf.d/#{new_resource.site_name}.conf" do
    source "conf.erb"
    mode "0644"
    variables(document_root: "/srv/apache/#{new_resource.site_name}/html", port: new_resource.site_port)
    notifies :restart, "service[httpd]"
  end
# ... REMAINDER OF CUSTOM RESOURCE ...
```

Updating the Resource to use the Property

~/apache/recipes/default.rb

```
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome home!</h1>'
end


apache_vhost 'admins' do
  site_name 'admins'
  site_port 8080
  action :create
end


service 'httpd' do
  action [:enable, :start]
end
```

Executing the Unit Tests

```
> chef exec rspec
```

```
........


Finished in 1.22 seconds (files took 7.58 seconds to load)
8 examples, 0 failures
```

Converging and Verifying the Test Instance

```
> kitchen converge && kitchen verify
```

```
-----> Verifying <default-centos-67>...
       Use `/home/chef/apache/test/smoke/default` for testing


Target:  ssh://vagrant@127.0.0.1:2222

  ✔    Command curl http://localhost stdout should match /Welcome home/
  ✔    Command curl http://localhost:8080 stdout should match /Welcome admins/


Summary: 2 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-67> (0m0.77s).
-----> Kitchen is finished. (2m42.37s)
```

LAB
apache_vhost - site_port Property

- ✓ Create a custom resource property named **site_port** that is a *Integer*
- ✓ Within the apache_vhost's create action replace the hard-coded value 8080 with the **site_port** property
- ✓ Within the default recipe set the apache_vhost resource named 'admins' to have a site_port 8080

Remove the Welcome Site

When apache installs itself it defines a default site on port 80. Up until this point we have relied on this site. We now want to remove that initial welcome site but we want to do that we a new action we will define on the custom resource we are defining.

apache_vhost Remove Action

❑ Define the 'remove' action for apache_vhost that defines the policy:

A directory named "/srv/apache/#{site_name}" is deleted

A file named "/etc/httpd/conf.d/#{site_name}.conf" is deleted

❑ Update the default recipe's policy to include a new resource:

An apache_vhost resource named 'welcome' is removed

CHEF

Defining the Resource's Remove Action

~/apache/resources/vhost.rb

```
# ... CREATE ACTION ...

action :remove do
  directory "/srv/apache/#{new_resource.site_name}" do
    action :delete
  end


  file "/etc/httpd/conf.d/#{new_resource.site_name}.conf" do
    action :delete
  end
end
```

CHEF

Adding the Resource with Remove Action to the Recipe

~/apache/recipes/default.rb

```ruby
package 'httpd'

file '/var/www/html/index.html' do
  content '<h1>Welcome home!</h1>'
end

apache_vhost 'welcome' do
  site_name 'welcome'
  action :remove
end

apache_vhost 'admins' do
  site_name 'admins'
  site_port 8080
  action :create
end
# ... REMAINDER OF RECIPE ...
```

CHEF

LAB
apache_vhost Remove Action

✓ Define the 'remove' action for apache_vhost that defines the policy:

   A directory named "/srv/apache/#{site_name}" is deleted

   A file named "/etc/httpd/conf.d/#{site_name}.conf" is deleted

✓ Update the default recipe's policy to include a new resource:

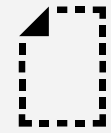   **An apache_vhost resource named 'welcome' is removed**

LAB

apache_vhost Remove Action

❏ Update the default recipe's policy:

Add an apache_vhost resource named 'users' is created with the site_port 80

❏ Update the ChefSpec tests to stop expecting the file resource and start expecting the new resources found within the apache_vhost resource named 'users'

❏ Update the InSpec tests to expect the default site to "Welcome users!"

Adding the Resource to create the users site

```ruby
apache_vhost 'welcome' do
  site_name 'welcome'
  action :remove
end

apache_vhost 'users' do
  site_name 'users'
  site_port 80
  action :create
end

apache_vhost 'admins' do
  site_name 'admins'
  site_port 8080
  action :create
  notifies :restart, 'service[httpd]'
end
```

Removing the Un-needed Unit Test Expectation

~/apache/spec/unit/recipes/default_spec.rb

```
# ... EXAMPLES DEFINED ABOVE ...


describe 'for the default site' do
  it 'writes out a new home page' do
    expect(chef_run).not_to
render_file('/var/www/html/index.html').with_content('<h1>Welcome home!</h1>')
  end
end


# ... EXAMPLES DEFINED BELOW ...
```

CHEF

Adding Expectations for the users Site

```ruby
# ... EXAMPLES DEFINED ABOVE ...
  describe 'for the users site' do
    it 'creates the directory' do
      expect(chef_run).to create_directory('/srv/apache/users/html')
    end

    it 'creates the configuration' do
      expect(chef_run).to render_file('/etc/httpd/conf.d/users.conf')
    end

    it 'creates a new home page' do
      expect(chef_run).to
render_file('/srv/apache/users/html/index.html').with_content('<h1>Welcome users!</h1>')
    end
  # ... EXAMPLES DEFINED ABOVE ...
```

Executing the Unit Test Suite

```
> chef exec rspec
```

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

Finished in 1.22 seconds (files took 2.56 seconds to load)
10 examples, 0 failures
```

Updating the Expectation for the users Site

`~/apache/test/smoke/default/default_test.rb`

```
describe command('curl http://localhost') do
  its(:stdout) { should match(/Welcome users/) }
end


describe command('curl http://localhost:8080') do
  its(:stdout) { should match(/Welcome admins/) }
end
```

CHEF

Converging and Verifying the Test Instance

```
> kitchen converge && kitchen verify
```

```
-----> Verifying <default-centos-67>...
       Use `/home/chef/apache/test/smoke/default` for testing


Target:  ssh://vagrant@127.0.0.1:2222


  ✔    Command curl http://localhost stdout should match /Welcome home/
  ✔    Command curl http://localhost:8080 stdout should match /Welcome admins/


Summary: 2 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-67> (0m0.77s).
-----> Kitchen is finished. (2m42.37s)
```

LAB

apache_vhost Remove Action

✓ Update the default recipe's policy:

   Add an apache_vhost resource named 'users' is created with the site_port 80

✓ Update the ChefSpec tests to stop expecting the file resource and start expecting the new resources found within the apache_vhost resource named 'users'

✓ Update the InSpec tests to expect the default site to "Welcome users!"

Discussion

What are the benefits of using a custom resource to manage the virtual hosts? What are the drawbacks of using a custom resource?

What does the resource collection look like when using a custom resource?

CHEF

DISCUSSION

Q&A

What questions can we answer for you?

CHEF