

Refining a Custom Resource

Objectives

After completing this module, you should be able to:

- Set a custom resource's name to a property
- Set a default value for a custom resource property
- Define notifications correctly when creating custom resources

After completing this module you should be able to set a custom resource's name to a property, set a default value if the property is not provided, and define notifications correctly within the custom resource.

EXERCISE

Refactoring the Resources

**Objective:**

- Define a default action for the custom resource
- Set the resource name as the site_name property
- Set the default value of the site_port property
- Move the resource notifications to the recipe

Every resource that you have used within Chef has had a default action.

PROBLEM



Selecting a Default Action

Resources often have a default action. The default action is often the action that is the least surprising. For most resources it is an additive/restorative action that moves the system into the desired state.

Our custom resource should be no different. Having a default action that performs a non-surprising operation is important. Of the two actions that we have defined the create action seems like the current correct default action.

CONCEPT



The First Action is the Default

The first action within the custom resource definition is the default one. But you can explicitly define the default action within the resource definition.

If you defined the `create` action as the first action within a resource definition file it automatically is the default action. The first action is the default action and that probably makes sense to those reading the custom resource definition.

You may also explicitly declare the default action.

Defining a Default Action for the Resource

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

property :site_name, String
property :site_port, Fixnum

default_action :create

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

If the first action is the create action then you do not need to explicitly define the default action. However, you may decide that it makes it clearer for you or those you are collaborating with to specify this within the resource definition.

Removing the Default Action for Resources

```
~/apache/recipes/default.rb

# ... INITIAL RECIPE DEFINITION ...

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

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

With the default action defined all uses of the custom resource which explicitly defined it may have those lines removed.

Executing the Unit Test Suite



```
> chef exec rspec
```

```
.....
```

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

All the unit tests should pass.

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)
```

All the integration test should pass.

EXERCISE

Refactoring the Resources

**Objective:**

- ✓ Define a default action for the custom resource
- Set the resource name as the site_name property
- Set the default value of the site_port property
- Move the resource notifications to the recipe

Now we will look at tying the name provided to the custom resource to the site_name property.

PROBLEM



Clarity in the Custom Resource

There is some duplication in the declaration of the resource. The name of the resource and the site_name. We want to default to use the name specified in the resource as the site_name. This is similar to other resources.

We want to ensure our custom resource is clear and concise. At the moment when you define the resource within the recipe you specify a value as the name of the custom resource and then a property that matches that same name. This seems like a redundancy that we want to remove.

Tying the Resource Name to the Property

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

property :site_name, String, name_attribute: true
property :site_port, Fixnum

default_action :create

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

One property may have the 'name_attribute' option set to true. This property will be automatically populated from the name of the resource. Using the name of the resource will allow us to remove the need to specify that property which is repeating a value within the use of the custom resource.

Removing the site_name Property

```
~/apache/recipes/default.rb
```

```
package 'apache'

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

apache_vhost 'admins' do
  site_name 'users'
  site_port 80
end
```

Now that the site_name property is set as the name attribute we can remove the site name property from each use of the custom resource.

Removing the site_name Property

```
~/apache/recipes/default.rb
```

```
# ... INITIAL RECIPE DEFINITION ...

apache_vhost 'admins' do
  site_name 'admins'
  site_port 8080
end

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

Now that the site_name property is set as the name attribute we can remove the site name property from each use of the custom resource.

Executing the Unit Test Suite



```
> chef exec rspec
```

```
.....
```

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

All the unit tests should pass.

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)
```

All the integration test should pass.

EXERCISE

Refactoring the Resources

**Objective:**

- ✓ Define a default action for the custom resource
- ✓ Set the resource name as the site_name property
- Set the default value of the site_port property
- Move the resource notifications to the recipe

Resource properties can also have default values setup for them. Lets explore setting a default value for the site_port property.

CONCEPT



Setting Default Values for Properties

Properties can also have default values. When deploying a website the default port that one would expect to see that site is on port 80.

Setting a default value for a property allows you to define the resource and if you omit setting the property then the default value is used. We can use this behavior for our `site_port`, choosing to say that if you do not specify a port we want the port to be 80.

Setting a Default Value for the Property

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

property :site_name, String, name_attribute: true
property :site_port, Fixnum, default: 80

default_action :create

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

All properties may have a default value. To add one requires that you simply add the option 'default' with its corresponding default value. Here we are setting the 'site_port' value to 80.

Removing the site_port Property

```
~/apache/recipes/default.rb

package 'apache'

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

apache_vhost 'users' do
  site_port 80
end
```

This allows us to remove the value from a single resource.

Executing the Unit Test Suite



```
> chef exec rspec
```

```
.....
```

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

All the unit tests should pass.

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)
```

All the integration test should pass.

EXERCISE

Refactoring the Resources

**Objective:**

- ✓ Define a default action for the custom resource
- ✓ Set the resource name as the site_name property
- ✓ Set the default value of the site_port property
- ❑ Move the resource notifications to the recipe

Finally we want to properly address how the custom resource handles notifications.

PROBLEM



Resource Notifications

Defining a notification in a resource within a custom resource creates a fragile relationship. One that we want to address by removing any notifications to outside resources.

When defining notifications within a resource action if you reference a resource outside of the action implementation there is a chance that your code may break if that resource's name were to change or simply not be implemented at all.

If any resource were to take action within the custom resource then the custom resource considers itself as taking action. We often say that any changed resource events are sent the parent custom resource and from that custom resource you can define your notifications.

Removing the Notification from the Resource

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

# ... INITIAL RESOURCE DEFINITION ...

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 ...
```

First we remove the dependency on a resource not present within the action implementation of the custom resource.

Adding the Notification to the Resources

```
~/apache/recipes/default.rb

package 'apache'

apache_vhost 'welcome' do
  notifies :restart, 'service[httpd]'
  action :remove
end

apache_vhost 'users' do
  notifies :restart, 'service[httpd]'
end
```

We then add the notifications to the custom resource implementations within the default recipe.

Adding the Notification to the Resources

```
~/apache/recipes/default.rb

# ... INITIAL RECIPE DEFINITION ...

apache_vhost 'admins' do
  site_port 8080
  notifies :restart, 'service[httpd]'
end

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

We then add the notifications to the custom resource implementations within the default recipe.

Executing the Unit Test Suite



```
> chef exec rspec
```

```
.....
```

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

All the unit tests should pass.

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)
```

All the integration test should pass.

EXERCISE

Refactoring the Resources

**Objective:**

- ✓ Define a default action for the custom resource
- ✓ Set the resource name as the site_name property
- ✓ Set the default value of the site_port property
- ✓ Move the resource notifications to the recipe

We now have refactored the custom resource to have it behave more like other resources we are familiar within Chef.

DISCUSSION



Q&A

What questions can we answer for you?

What questions can we answer for you?

