

# Creating Ohai Plugins



## Objectives

After completing this module, you should be able to:

- Create a tested Ohai plugin

After completing this module you will be able to

# EXERCISE



## Creating an Ohai Plugin

*Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.*

### Objective:

- ☐ Define expectations for the Ohai plugin
- ☐ Create the Ohai plugin
- ☐ Define expectations for the recipe to deliver the Ohai plugin
- ☐ Create the recipe that delivers the Ohai plugin
- ☐ Define an integration test

## Installing the chefspec-ohai gem



```
> chef gem install chefspec-ohai
```

```
Successfully installed chefspec-ohai-0.1.0  
1 gem installed
```

## Adding the Gem to the Spec Helper

 ~/apache/spec/spec\_helper.rb

```
require 'chefspect'  
require 'chefspect/berkshelf'  
require 'chefspect/ohai'
```

## Creating a Directory for Plugins Tests



```
> mkdir ~/apache/spec/unit/plugins
```

## Defining the First Expectation for Plugin



~/apache/spec/unit/plugins/apache\_modules\_spec.rb

```
require 'spec_helper'

describe_ohai_plugin :Apache do
  let(:plugin_file) { 'files/default/apache_modules.rb' }

  it 'provides apache/modules' do
    expect(plugin).to provide_attribute('apache/modules')
  end
end
```

## Executing the Tests to See Failure



```
> chef exec rspec
```

```
F.....
```

```
Failures:
```

```
1) Apache provides 'apache/modules'
```

```
Failure/Error: let(:plugin_source) { File.read(plugin_file) }
```

```
Errno::ENOENT:
```

```
No such file or directory @ rb_sysopen -  
files/default/apache_modules.rb
```



# EXERCISE



## Creating an Ohai Plugin

*Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.*

### Objective:

- ✓ Define expectations for the Ohai plugin
- ☐ Create the Ohai plugin
- ☐ Define expectations for the recipe to deliver the Ohai plugin
- ☐ Create the recipe that delivers the Ohai plugin
- ☐ Define an integration test

## Creating the Plugin as a Cookbook File



```
> chef generate file apache_modules.rb
```

```
Recipe: code_generator::cookbook_file
```

- \* directory[/home/chef/apache/files/default] action create (up to date)
- \* template[/home/chef/apache/files/default/apache\_modules.rb] action create
  - create new file /home/chef/apache/files/default/apache\_modules.rb
  - update content in file /home/chef/apache/files/default/

## Defining the Plugin that Provides Values



```
~/apache/files/default/apache_modules.rb
```

```
Ohai.plugin :Apache do
  provides 'apache/modules'
end
```

## Executing the Tests to See Success



```
> chef exec rspec
```

```
.....
```

```
Finished in 5.85 seconds (files took 2.74 seconds to load)  
10 examples, 0 failures
```

## Defining an Expectation for Attribute Value



~/apache/spec/unit/plugins/apache\_modules\_spec.rb

```
require 'spec_helper'

describe_ohai_plugin :Apache do
  let(:plugin_file) { 'files/default/apache_modules.rb' }

  it 'provides apache/modules' do
    expect(plugin).to provide_attribute('apache/modules')
  end

  it 'correctly captures output' do
    allow(plugin).to receive(:shell_out).with('apachectl -t -D
DUMP_MODULES').and_return(double(stdout: 'OUTPUT'))
    expect(plugin_attribute('apache/modules')).to eq('OUTPUT')
  end
end
```

## Executing the Tests to See Failure



```
> chef exec rspec
```

```
.F.....
```

```
Failures:
```

```
1) Apache correctly captures output  
   Failure/Error: expect(plugin_attribute('apache/modules')).to  
   eq("OUTPUT")
```

```
NoMethodError:
```

```
undefined method `[]' for nil:NilClass
```

## Capturing the Apache Modules Content



~/apache/files/default/apache\_modules.rb

```
Ohai.plugin :Apache do
  provides 'apache/modules'

  collect_data :default do
    apache(Mash.new)
    modules_cmd = shell_out('apachectl -t -D DUMP_MODULES')
    apache[:modules] = modules_cmd.stdout
  end
end
```

## Executing the Tests to See Success



```
> chef exec rspec
```

```
.[2017-10-05T17:00:43-05:00] WARN: Plugin Definition Error:  
<files/default>: collect_data already defined on platform default  
.....
```

```
Finished in 5.84 seconds (files took 3.01 seconds to load)  
11 examples, 0 failures
```



# EXERCISE



## Creating an Ohai Plugin

*Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.*

### Objective:

- ✓ Define expectations for the Ohai plugin
- ✓ Create the Ohai plugin
- ☐ Define expectations for the recipe to deliver the Ohai plugin
- ☐ Create the recipe that delivers the Ohai plugin
- ☐ Define an integration test

## Adding a Dependency on the Ohai Cookbook



~/apache/metadata.rb

```
name 'apache'
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'all_rights'
description 'Installs/Configures apache'
long_description 'Installs/Configures apache'
version '0.1.0'

depends 'ohai'

# If you upload to Supermarket you should set this so your cookbook
# gets a `View Issues` link
# issues_url 'https://github.com/<insert_org_here>/apache/issues' if respond_to?(:issues_url)

# If you upload to Supermarket you should set this so your cookbook
# gets a `View Source` link
# source_url 'https://github.com/<insert_org_here>/apache' if respond_to?(:source_url)
```

## Creating the Recipe to Add the Plugin



```
> chef generate recipe ohai_apache_modules
```

```
Recipe: code_generator::recipe
  * directory[/home/chef/apache/spec/unit/recipes] action create
    (up to date)
  * cookbook_file[/home/chef/apache/spec/spec_helper.rb] action
    create_if_missing (up to date)
  *
  template[/home/chef/apache/spec/unit/recipes/ohai_apache_modules_s
    pec.rb] action create_if_missing
  ...
```

## Defining the Expectation Plugin Deployed



~/apache/spec/unit/recipes/ohai\_apache\_modules\_spec.rb

```
# ... UPPER PORTION OF THE SPECIFICATION ...  
let(:chef_run) do  
  runner = ChefSpec::ServerRunner.new  
  runner.converge(described_recipe)  
end  
  
it 'converges successfully' do  
  expect { chef_run }.to_not raise_error  
end  
  
it 'installs the modules plugin' do  
  expect(chef_run).to create_ohai_plugin('apache_modules')  
end  
end  
end
```

## Execute the Tests to See Failure



```
> chef exec rspec
```

```
.....F
```

```
Failures:
```

```
1) apache::ohai_apache_modules When all attributes are default,  
on an unspecified platform installs the modules plugin
```

```
Failure/Error: expect(chef_run).to  
create_ohai_plugin('apache_modules')
```

```
expected "ohai_plugin[apache_modules]" with action :create  
to be in Chef run. Other ohai_plugin resources:
```

# EXERCISE



## Creating an Ohai Plugin

*Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.*

### Objective:

- ✓ Define expectations for the Ohai plugin
- ✓ Create the Ohai plugin
- ✓ Define expectations for the recipe to deliver the Ohai plugin
- ☐ Create the recipe that delivers the Ohai plugin
- ☐ Define an integration test

## Defining the Ohai Plugin in the Recipe

 ~/apache/recipes/ohai\_apache\_modules.rb

```
#  
# Cookbook Name:: apache  
# Recipe:: ohai_apache_modules  
#  
# Copyright (c) 2017 The Authors, All Rights Reserved.  
include_recipe 'apache::default'  
ohai_plugin 'apache_modules'
```

## Executing the Tests to See Success



```
> chef exec rspec
```

```
.....
```

```
Finished in 5.77 seconds (files took 2.62 seconds to load)  
12 examples, 0 failures
```



# EXERCISE



## Creating an Ohai Plugin

*Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.*

### Objective:

- ✓ Define expectations for the Ohai plugin
- ✓ Create the Ohai plugin
- ✓ Define expectations for the recipe to deliver the Ohai plugin
- ✓ Create the recipe that delivers the Ohai plugin
- ☐ Define an integration test

## Appending the New Recipe to the Suite



~/apache/.kitchen.yml

```
# ... TOP OF KITCHEN CONFIGURATION ...  
suites:  
  - name: default  
    run_list:  
      - recipe[apache::default]  
      - recipe[apache::ohai_apache_modules]  
  verifier:  
    inspec_tests:  
      - test/smoke/default  
  attributes:
```

## Converging the Updated Default Suite



```
> kitchen converge
```

```
-----> Starting Kitchen (v1.11.1)
```

```
-----> Converging <default-centos-67>...
```

```
    resolving cookbooks for run list: ["apache::default",  
    "apache::ohai_apache_modules"]
```

```
    Synchronizing Cookbooks:
```

- ohai (4.2.2)
- apache (0.1.0)
- compat\_resource (12.14.7)

```
    Installing Cookbook Gems:
```

```
    Compiling Cookbooks...
```

```
    Recipe: apache::ohai apache modules
```

## Defining an Expectation for the Plugin



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

```
plugin_directory = '/tmp/kitchen/ohai/plugins'

describe command("ohai -d #{plugin_directory} apache") do
  its(:stdout) { should match(/core_module/) }
end
```

## Verifying the New Expectation



```
> kitchen verify
```

```
-----> Starting Kitchen (v1.11.1)
-----> Verifying <ohai-plugin-centos-67>...
        Use `/home/chef/apache/test/smoke/default` for testing
```

```
Target:  ssh://kitchen@localhost:32768
```

```
✓ Command ohai -d /tmp/kitchen/ohai/plugins apache stdout
should match /core_module \(static\) /
```

```
Summary: 1 successful, 0 failures, 0 skipped
```

# EXERCISE



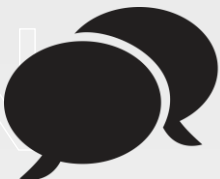
## Creating an Ohai Plugin

*Being able to learn more about our nodes will make our reporting more powerful and benefit the recipes we write.*

### Objective:

- ✓ Define expectations for the Ohai plugin
- ✓ Create the Ohai plugin
- ✓ Define expectations for the recipe to deliver the Ohai plugin
- ✓ Create the recipe that delivers the Ohai plugin
- ✓ Define an integration test

# DISCUSSION



## Q&A

What questions can we answer for you?

