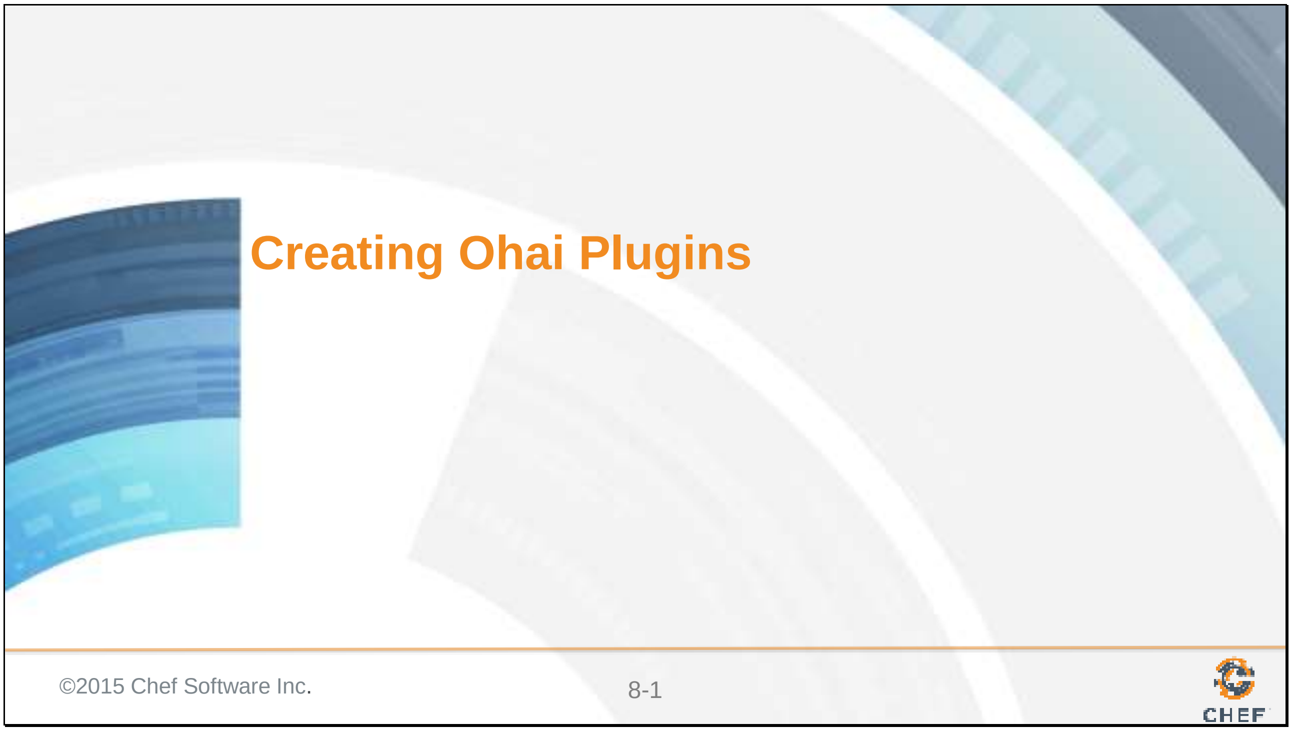
8: Creating Ohai Plugins

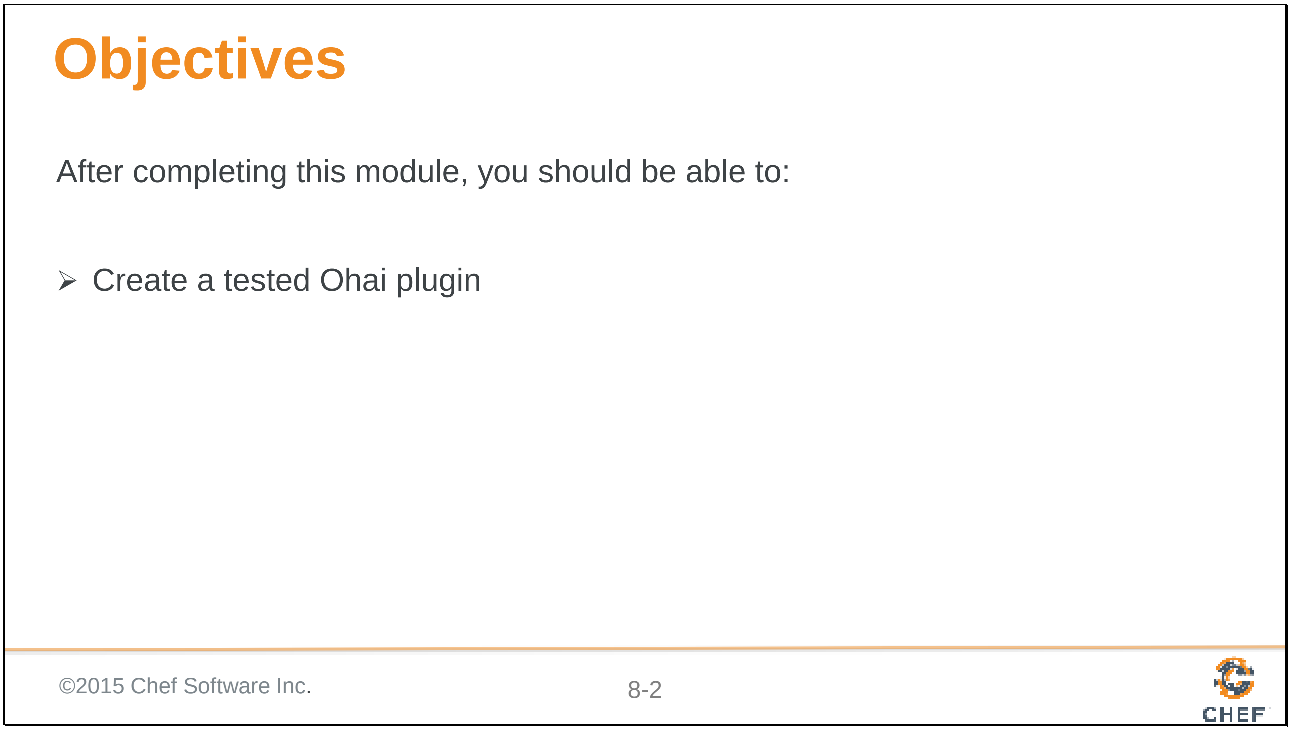


Creating Ohai Plugins

©2015 Chef Software Inc.

8-1

Slide 2



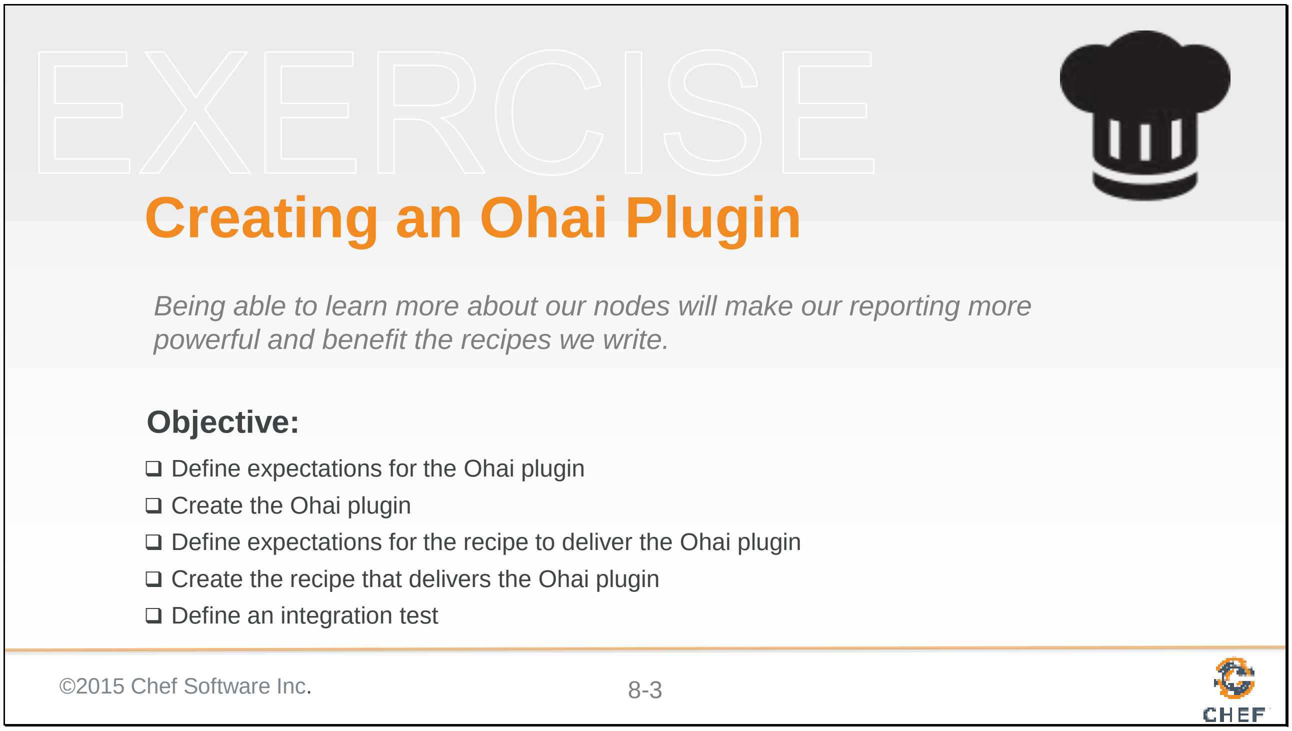
Objectives

After completing this module, you should be able to:

- Create a tested Ohai plugin

After completing this module you will be able to

Slide 3



EXERCISE

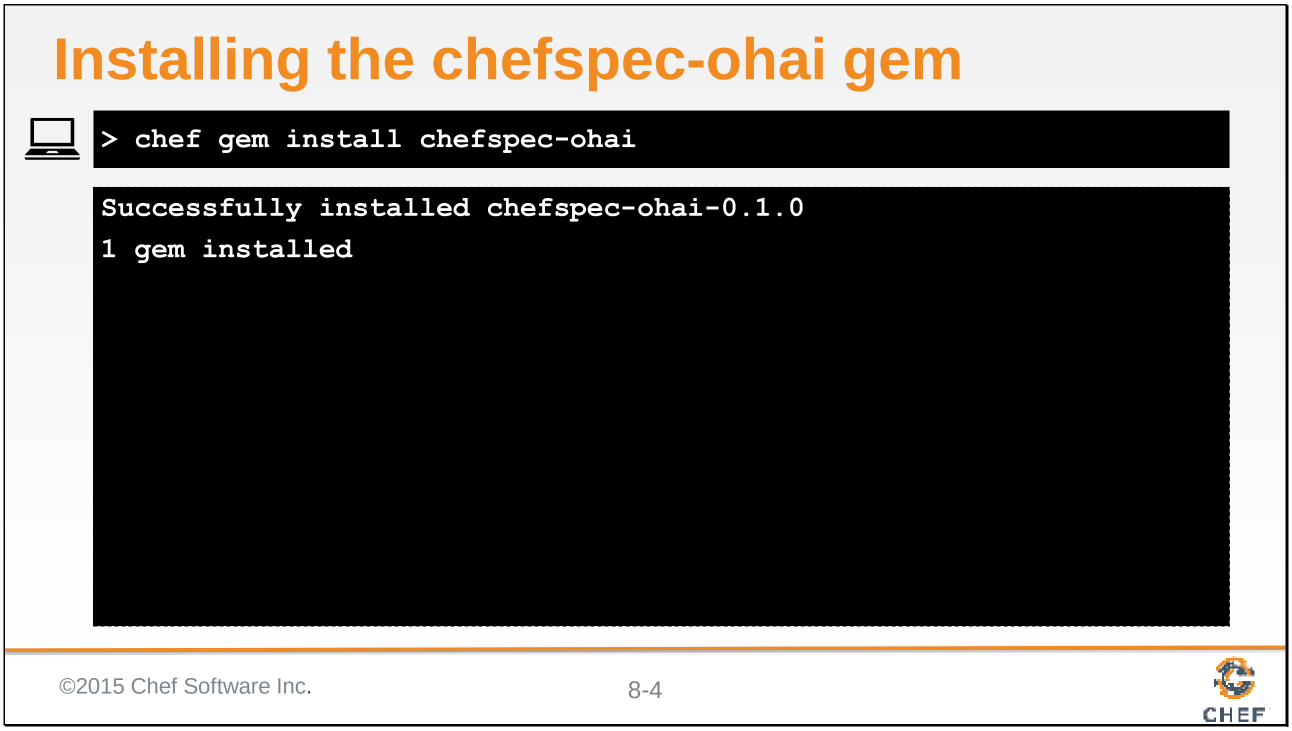
Creating an Ohai Plugin

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

Slide 4

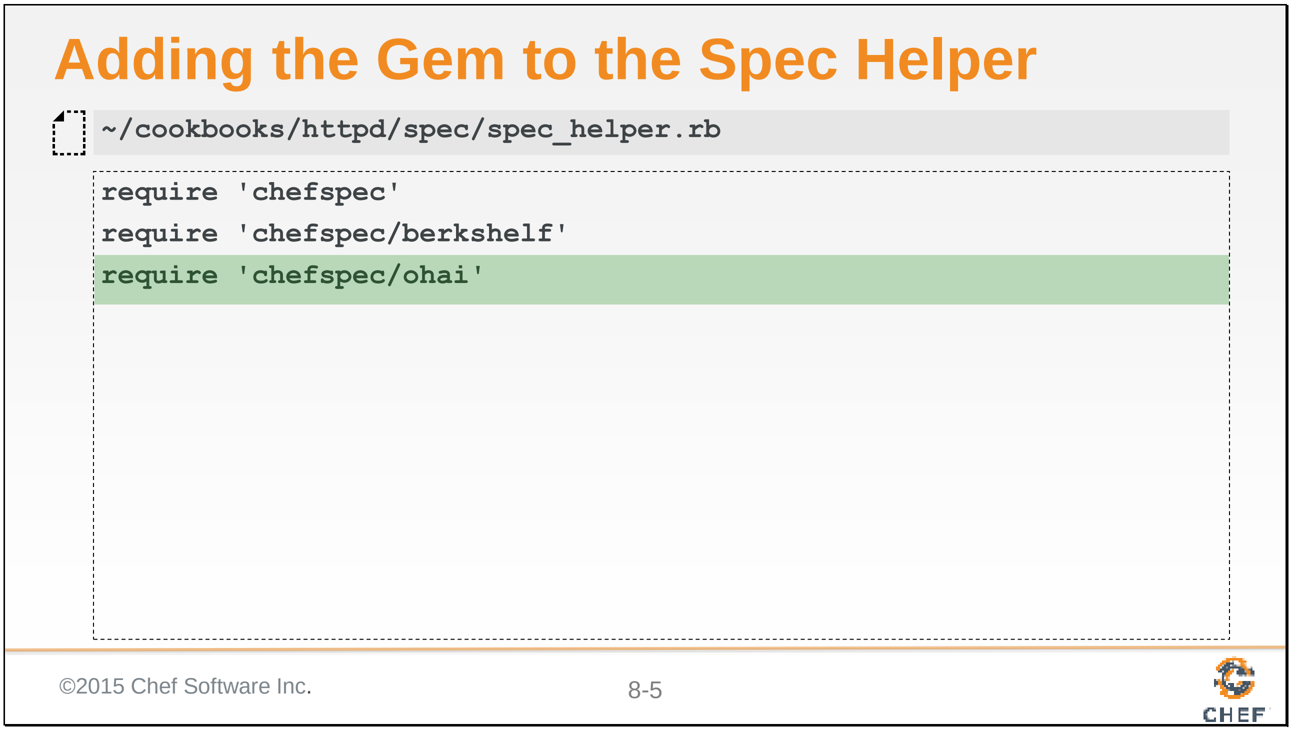


Installing the chefspec-ohai gem

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

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

Slide 5

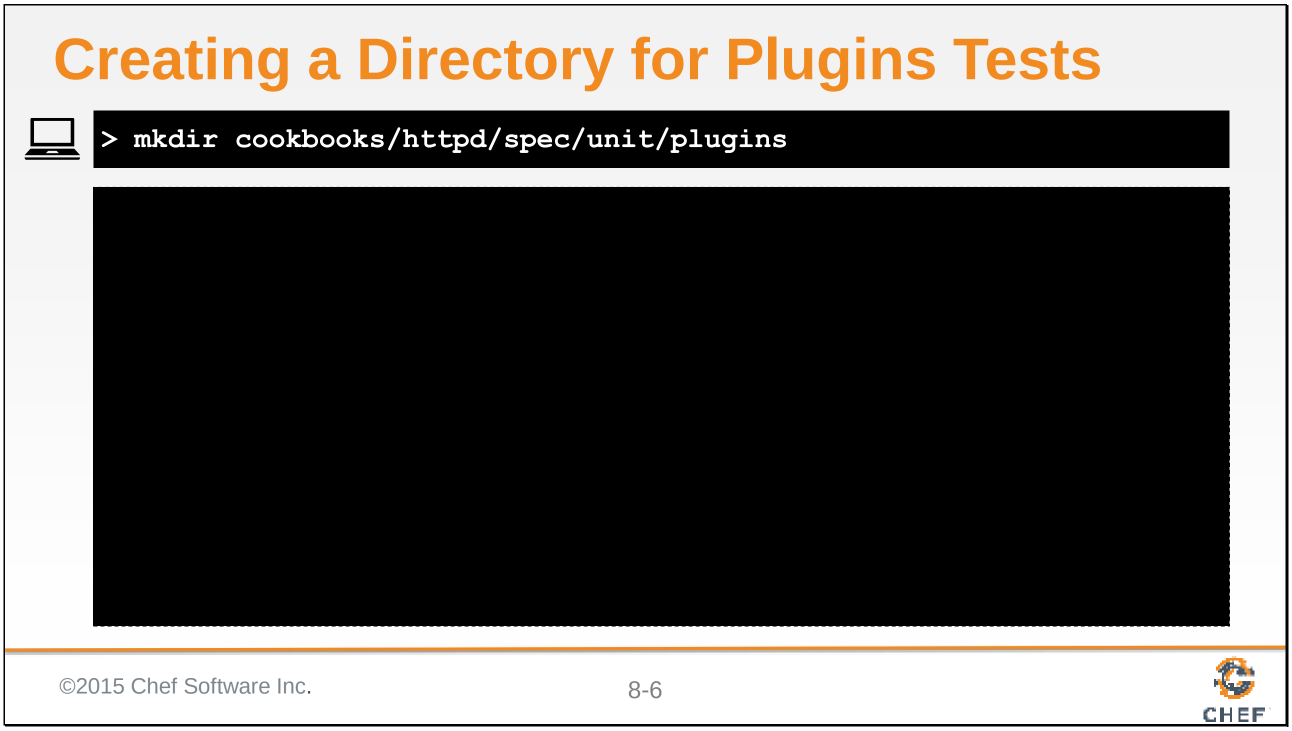


Adding the Gem to the Spec Helper

~/cookbooks/httpd/spec/spec_helper.rb

```
require 'chefspec'
require 'chefspec/berkshelf'
require 'chefspec/ohai'
```

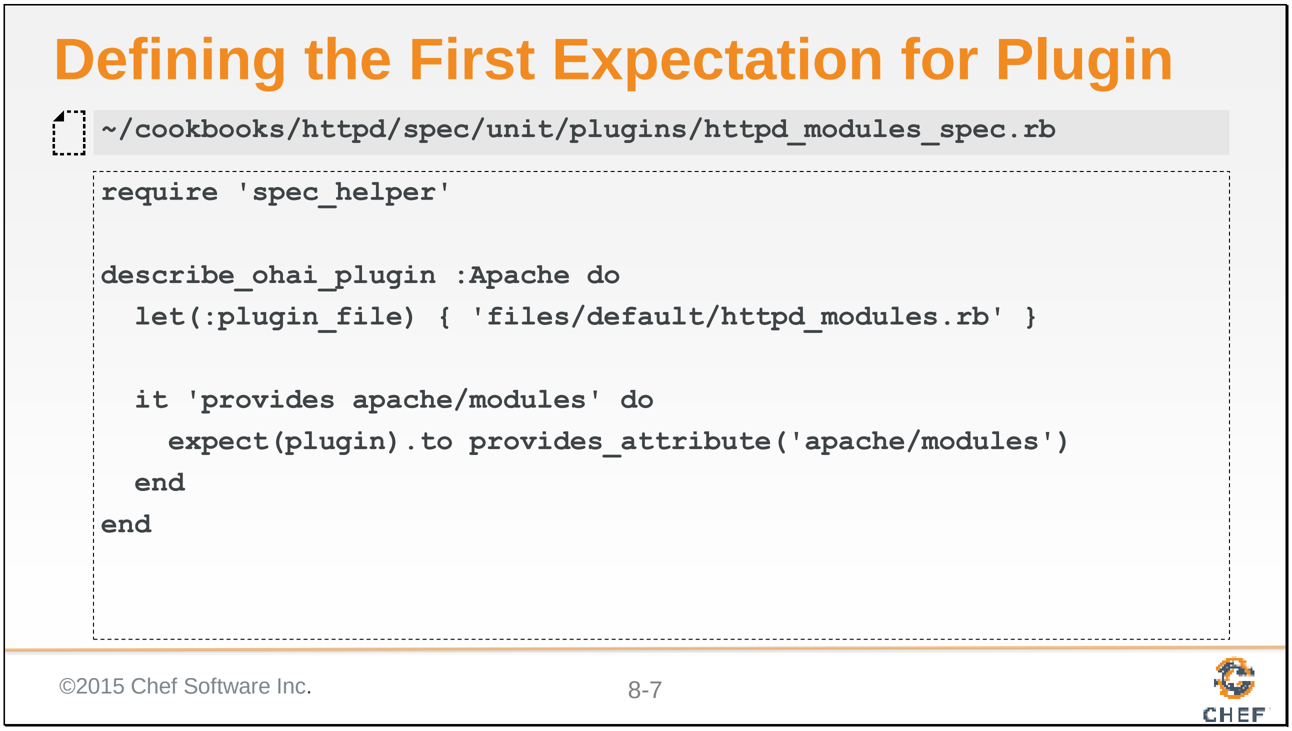
Slide 6



Creating a Directory for Plugins Tests

```
> mkdir cookbooks/httpd/spec/unit/plugins
```

Slide 7



Defining the First Expectation for Plugin

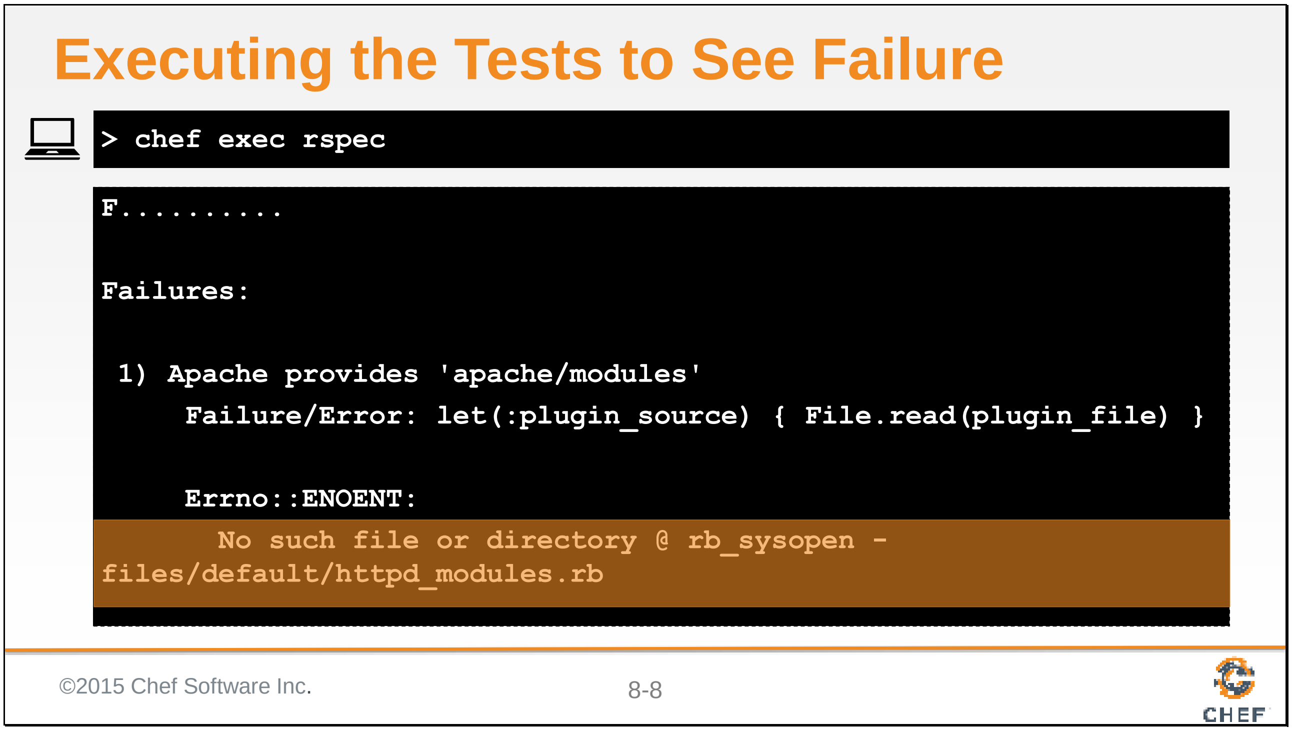
~/cookbooks/httpd/spec/unit/plugins/httpd_modules_spec.rb

```
require 'spec_helper'

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

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

Slide 8



Executing the Tests to See Failure

```
> chef exec rspec
```

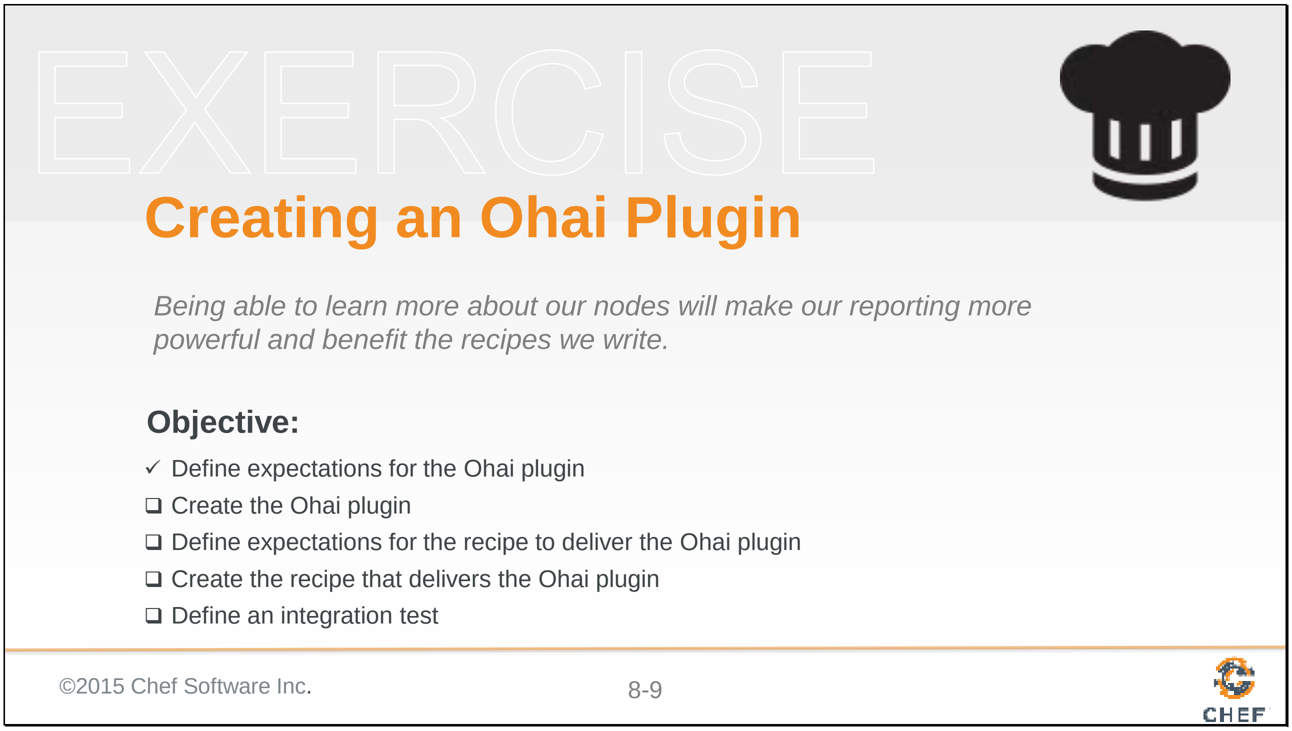
```
F.........

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/httpd_modules.rb
```

Slide 9



EXERCISE

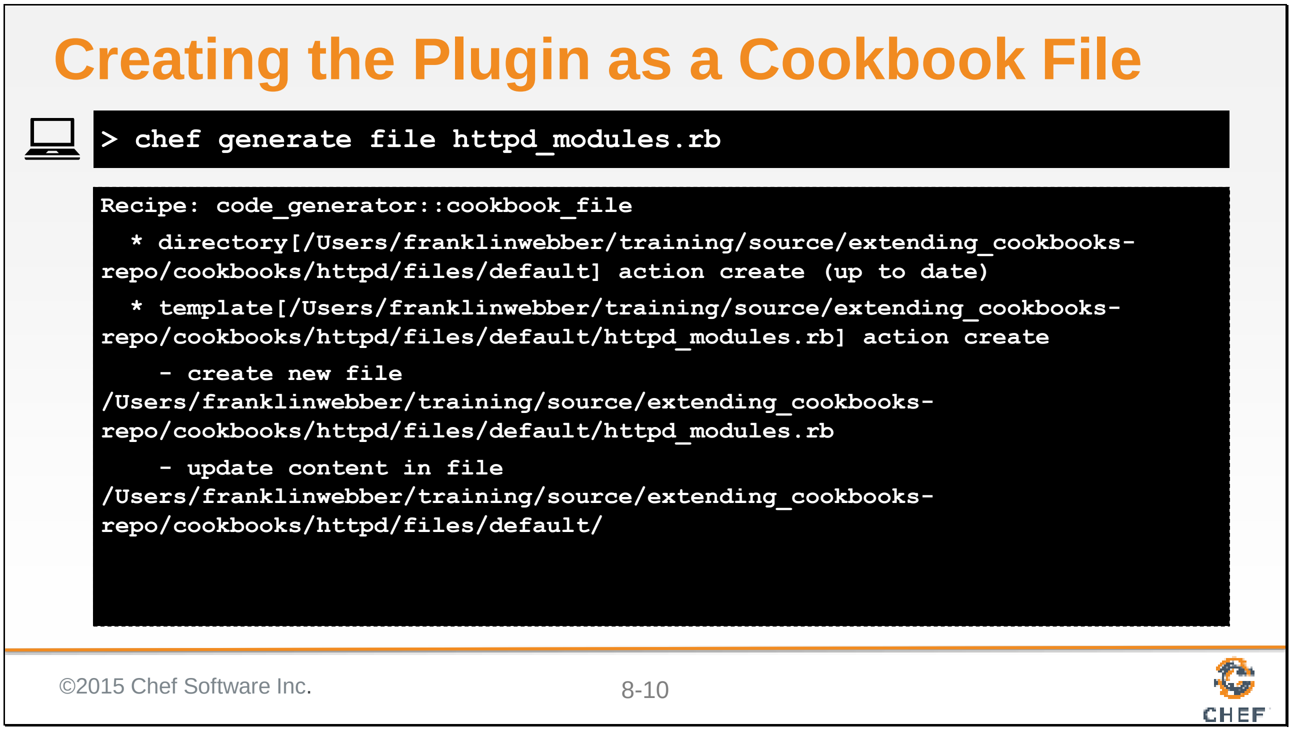
Creating an Ohai Plugin

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

Slide 10

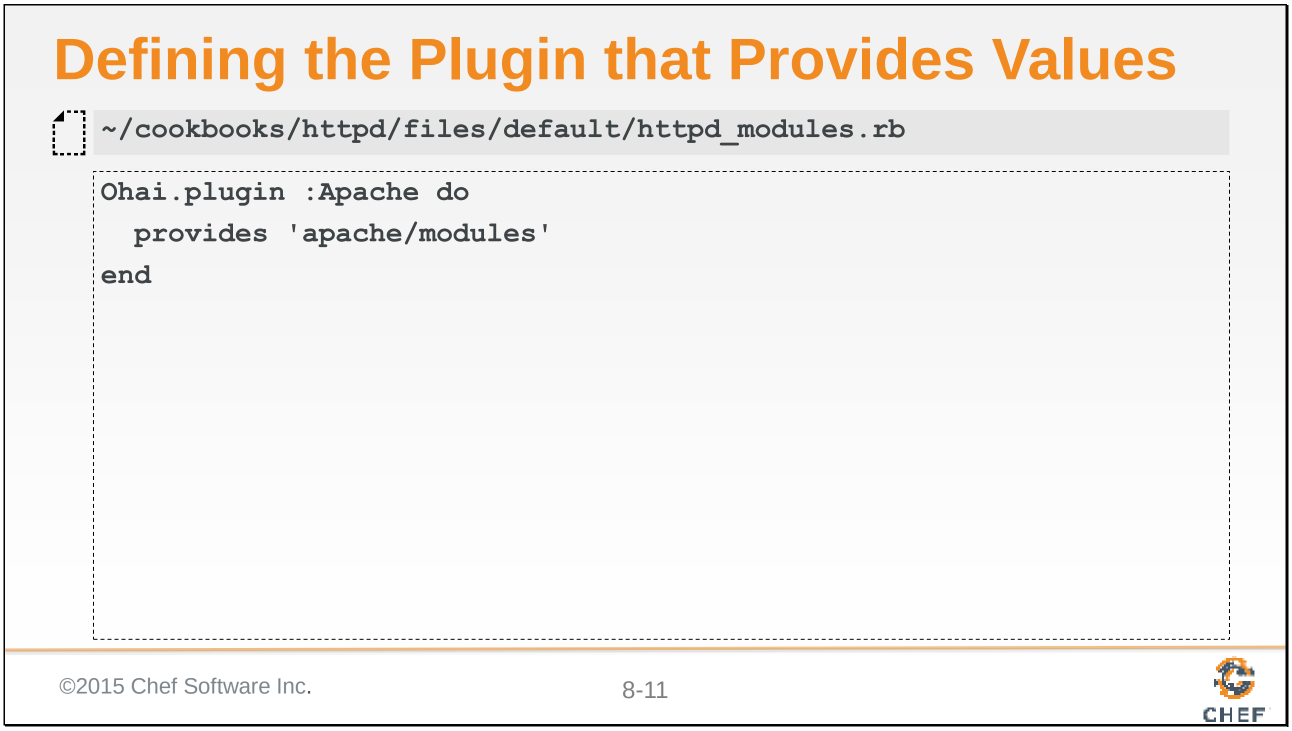


Creating the Plugin as a Cookbook File

```
> chef generate file httpd_modules.rb
```

```
Recipe: code_generator::cookbook_file
  * directory[/Users/franklinwebber/training/source/extending_cookbooks-
repo/cookbooks/httpd/files/default] action create (up to date)
  * template[/Users/franklinwebber/training/source/extending_cookbooks-
repo/cookbooks/httpd/files/default/httpd_modules.rb] action create
    - create new file
/Users/franklinwebber/training/source/extending_cookbooks-
repo/cookbooks/httpd/files/default/httpd_modules.rb
    - update content in file
/Users/franklinwebber/training/source/extending_cookbooks-
repo/cookbooks/httpd/files/default/
```

Slide 11

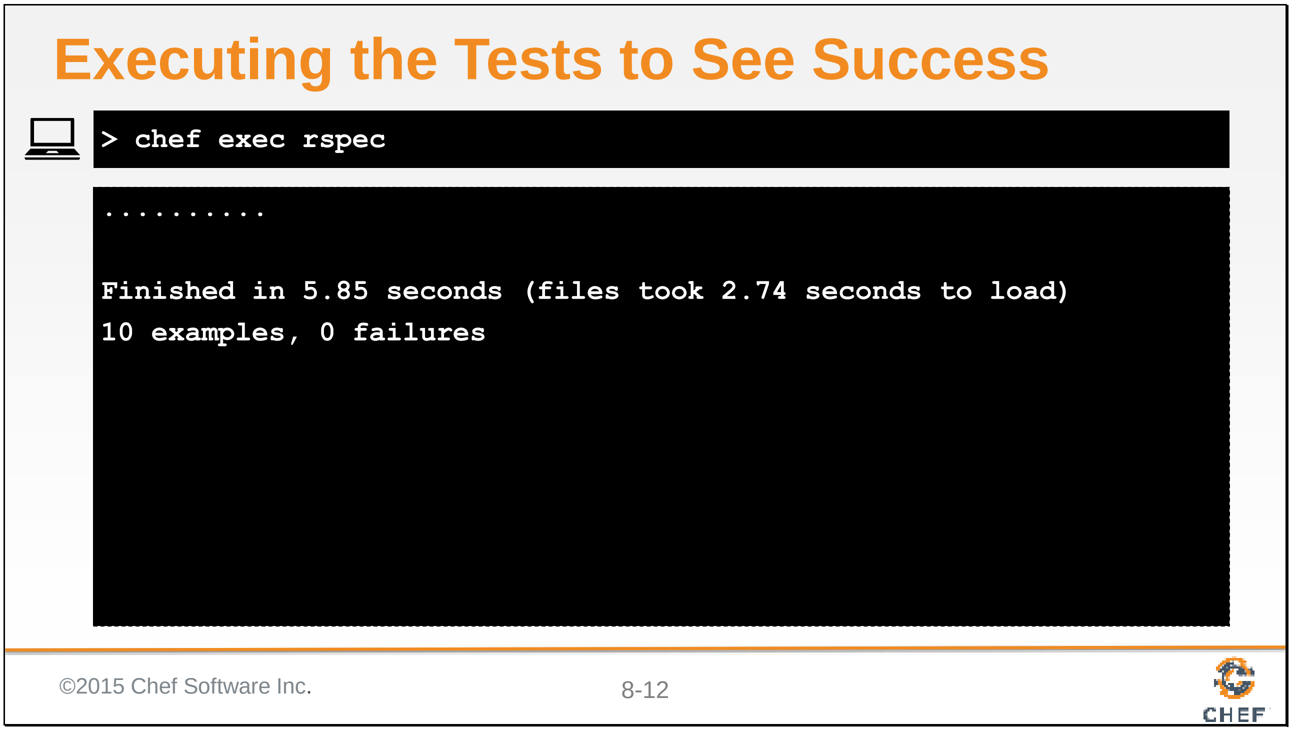


Defining the Plugin that Provides Values

~/cookbooks/httpd/files/default/httpd_modules.rb

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

Slide 12



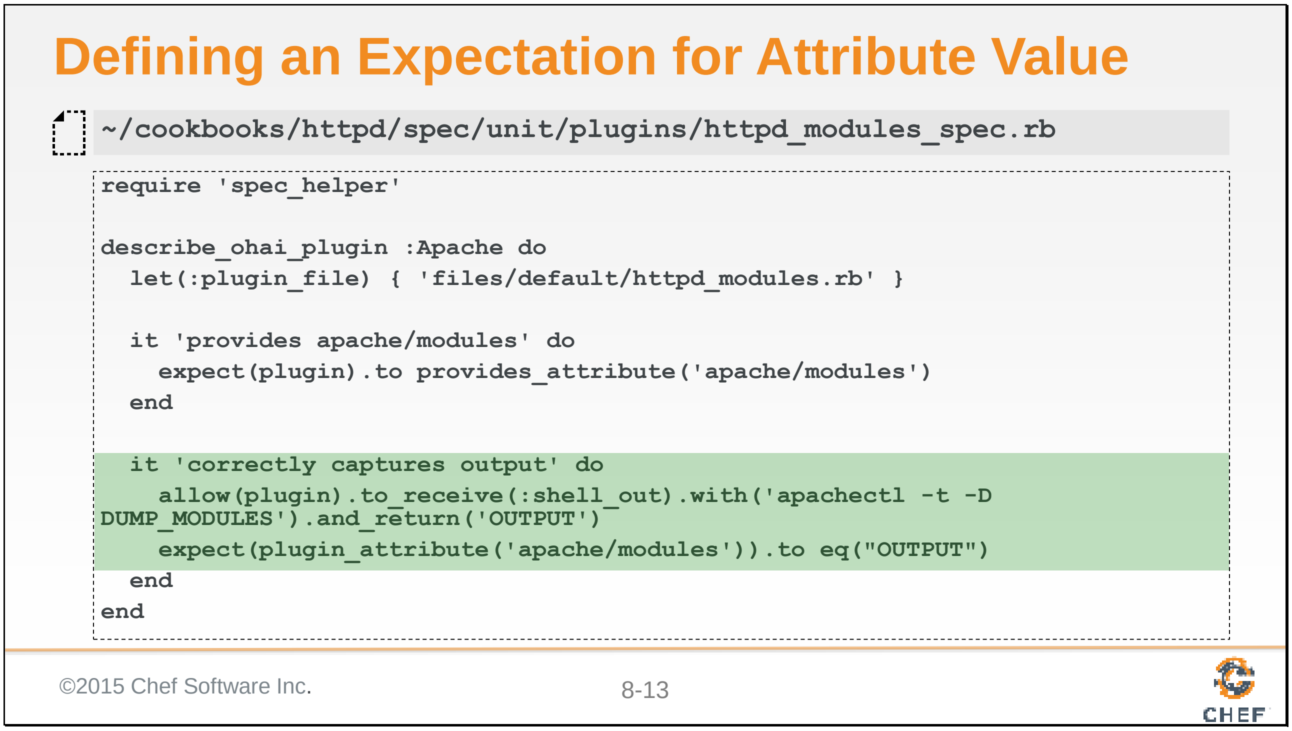
Executing the Tests to See Success

```
> chef exec rspec
```

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

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

Slide 13



Defining an Expectation for Attribute Value

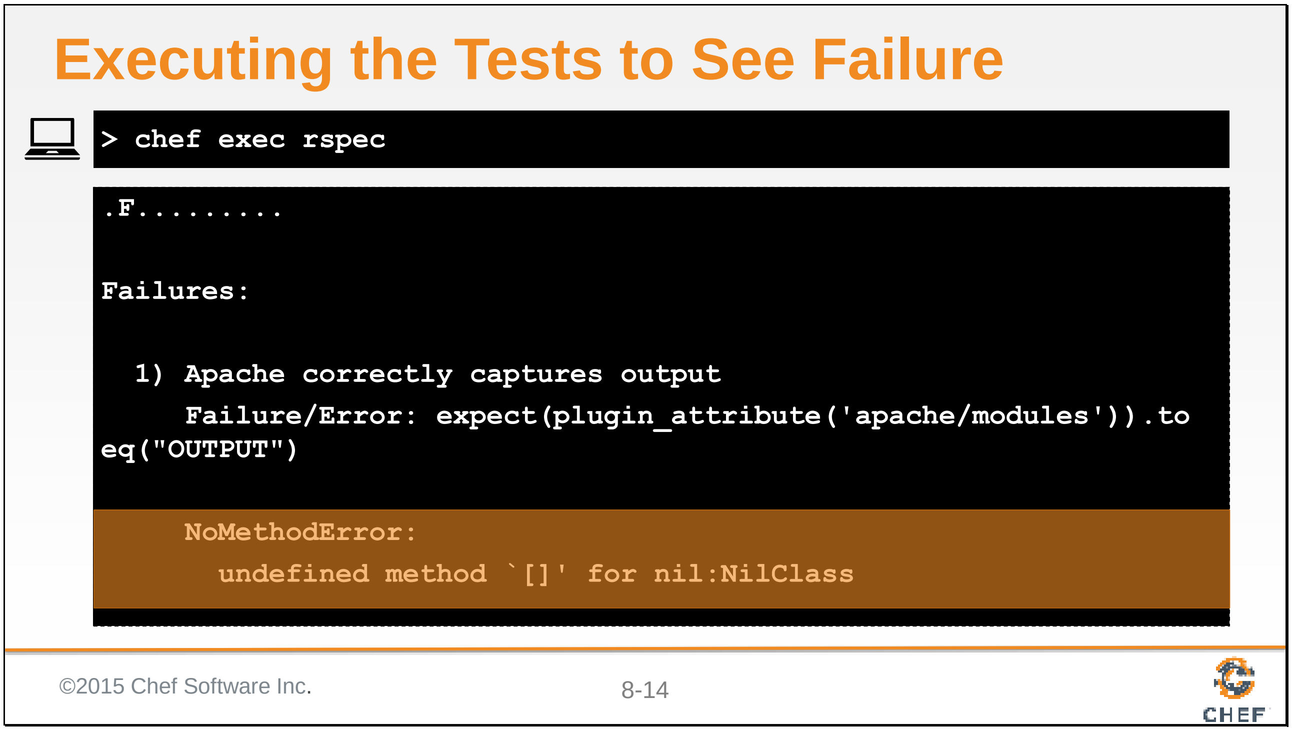
~/cookbooks/httpd/spec/unit/plugins/httpd_modules_spec.rb

```
require 'spec_helper'

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

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

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



Executing the Tests to See Failure

```
> chef exec rspec
```

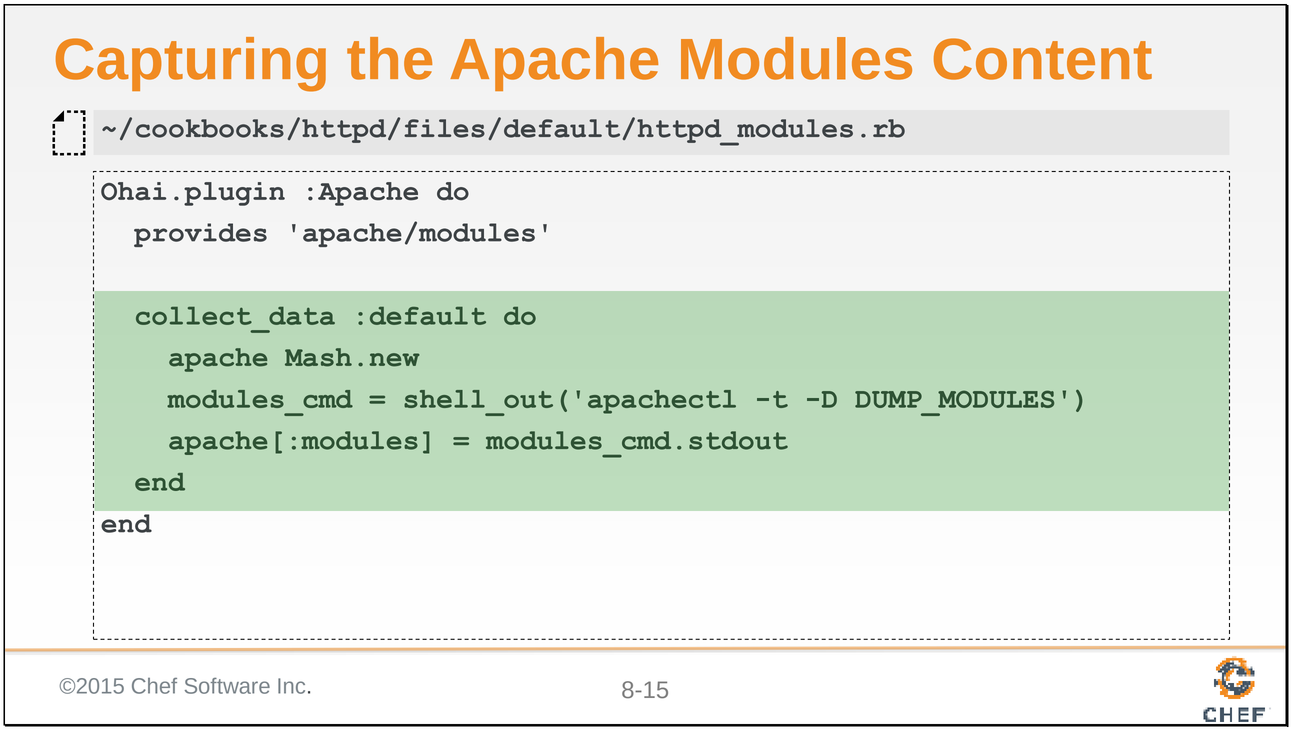
```
.F.........

Failures:

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

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

Slide 15

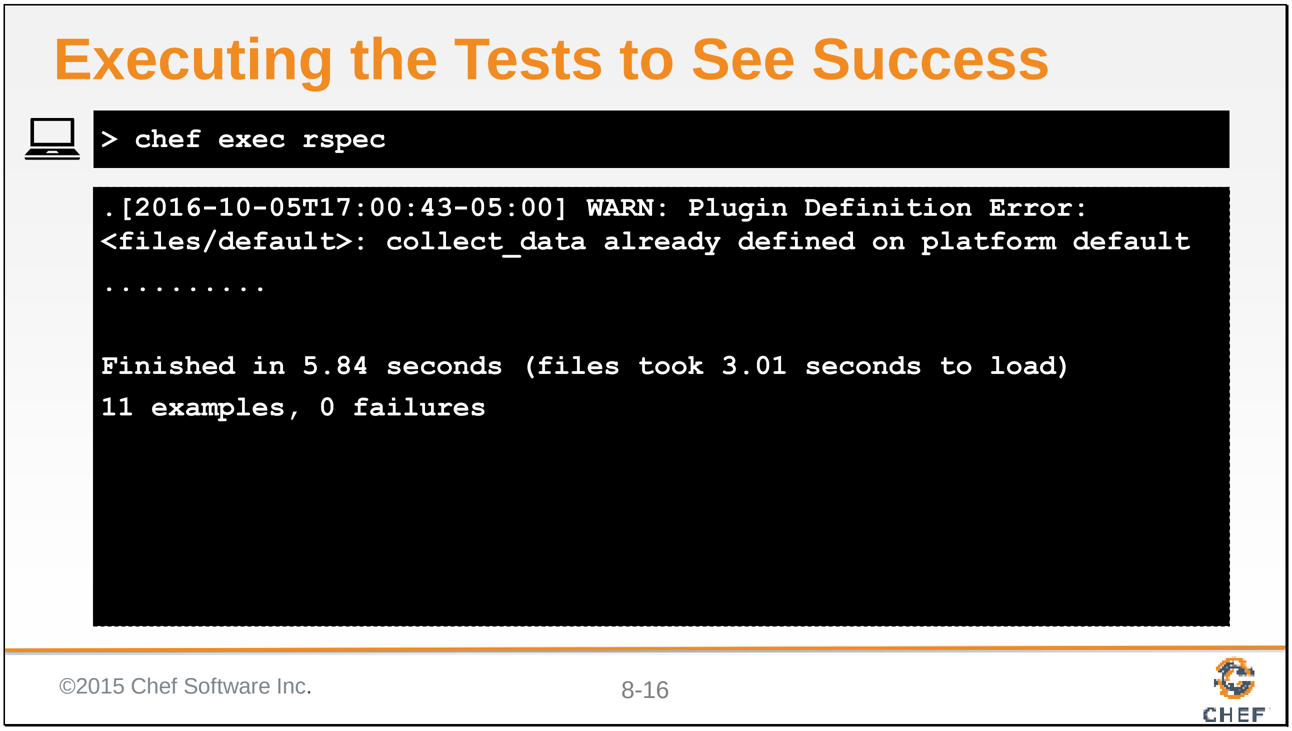


Capturing the Apache Modules Content

~/cookbooks/httpd/files/default/httpd_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
```



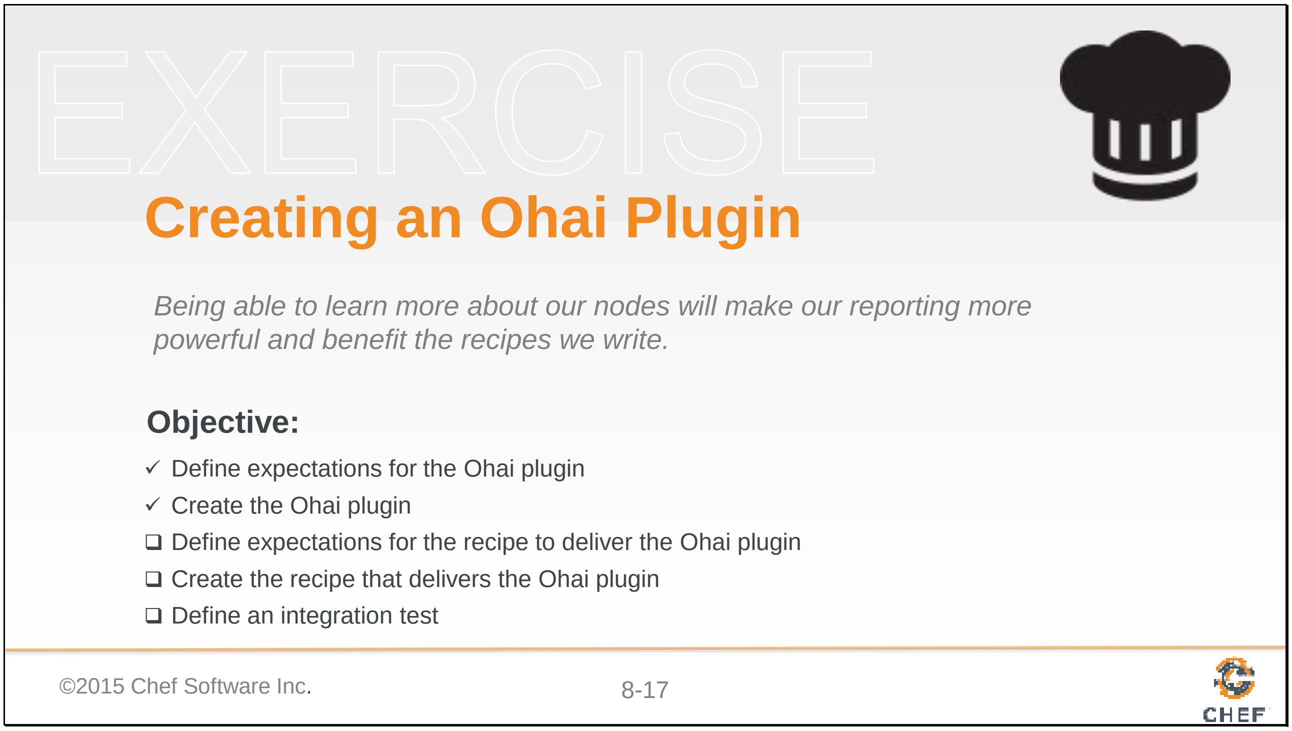
Executing the Tests to See Success

```
> chef exec rspec
```

```
.[2016-10-05T17:00:43-05:00] 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
```

Slide 17



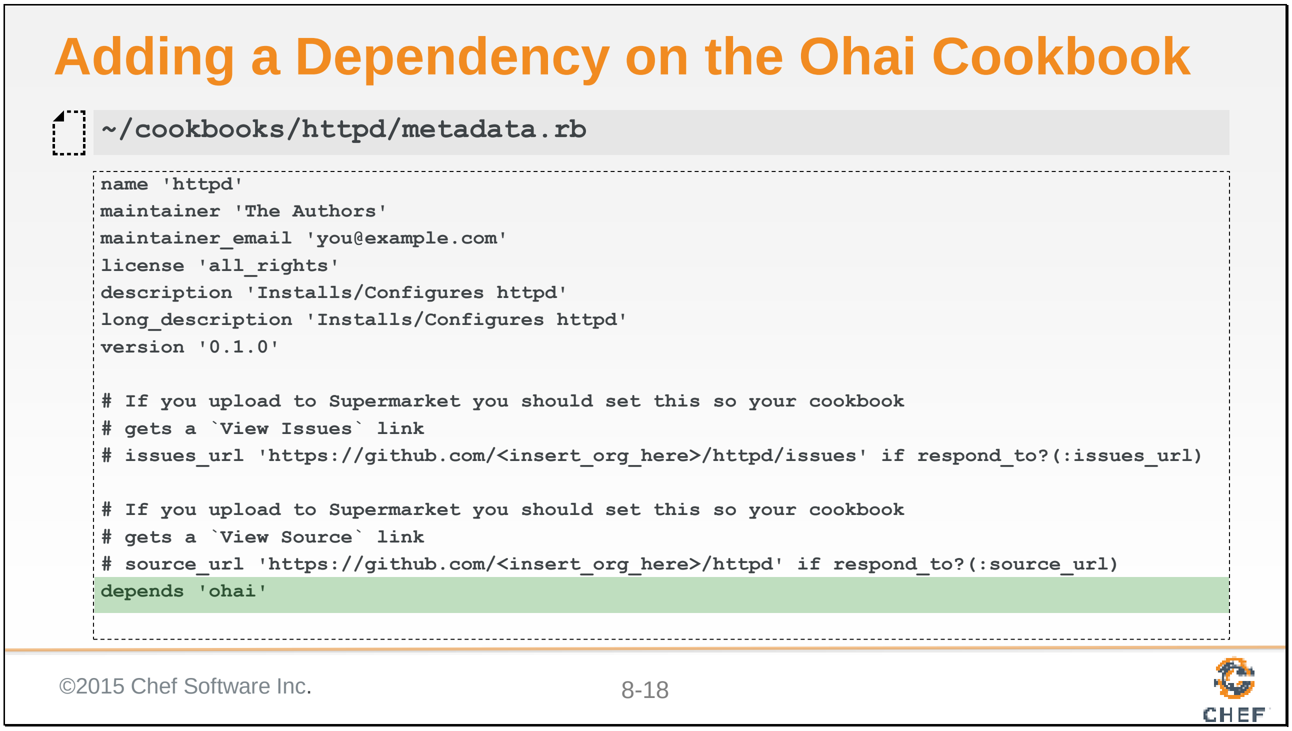
EXERCISE

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

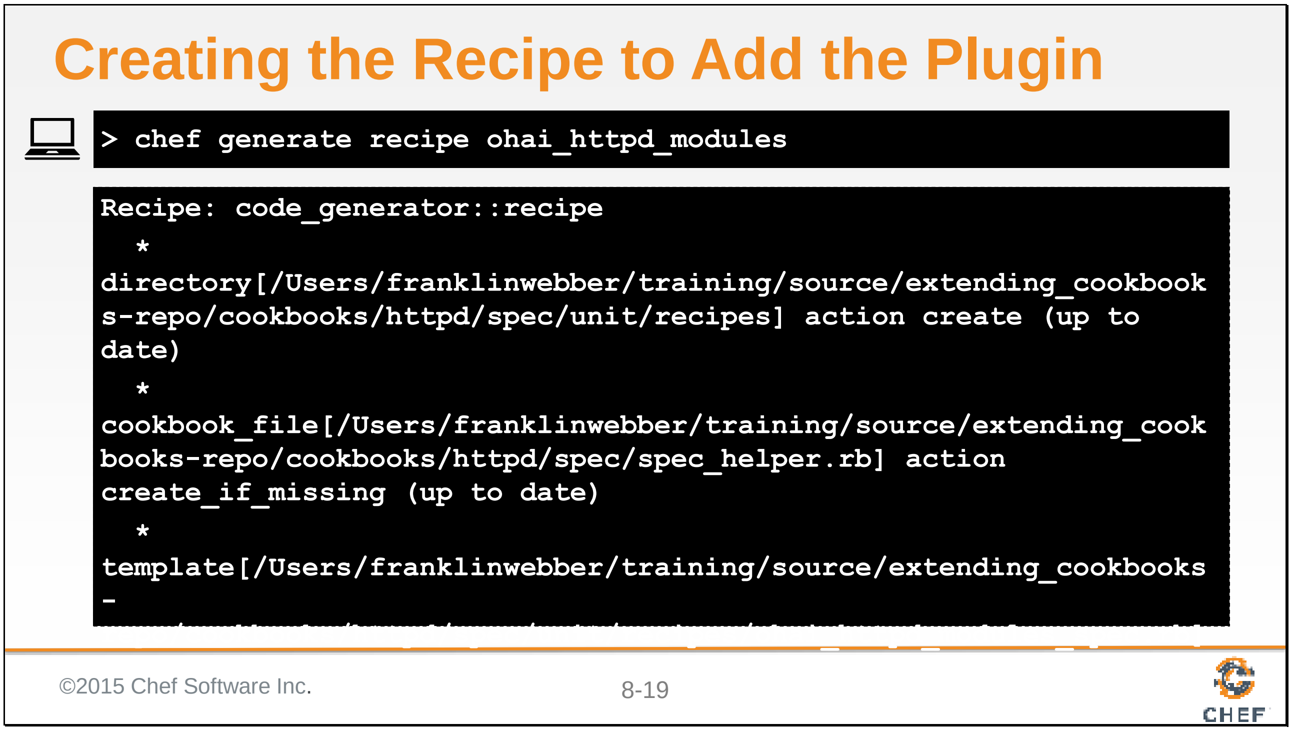
~/cookbooks/httpd/metadata.rb

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

# 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>/httpd/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>/httpd' if respond_to?(:source_url)
depends 'ohai'
```

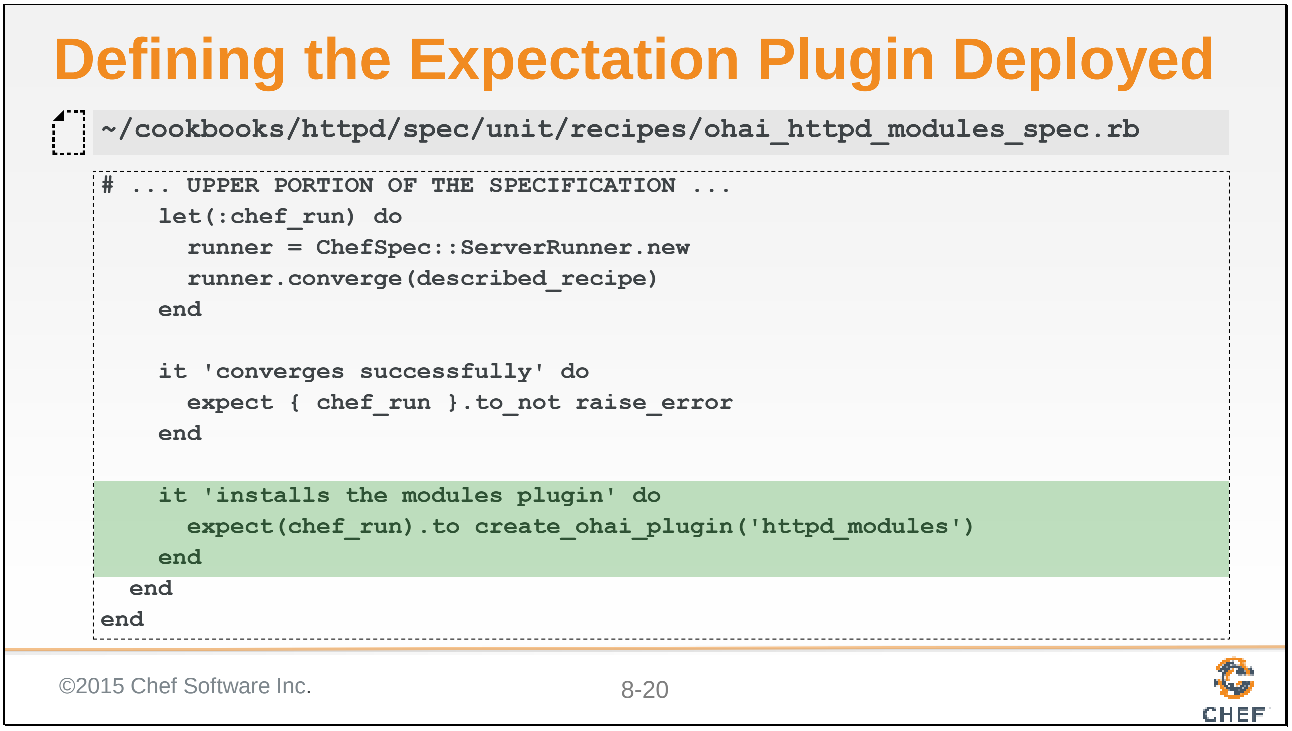
Slide 19



Creating the Recipe to Add the Plugin

```
> chef generate recipe ohai_httpd_modules
```

```
Recipe: code_generator::recipe
  *
directory[/Users/franklinwebber/training/source/extending_cookbook
s-repo/cookbooks/httpd/spec/unit/recipes] action create (up to
date)
  *
cookbook_file[/Users/franklinwebber/training/source/extending_cook
books-repo/cookbooks/httpd/spec/spec_helper.rb] action
create_if_missing (up to date)
  *
template[/Users/franklinwebber/training/source/extending_cookbooks
-
```



Defining the Expectation Plugin Deployed

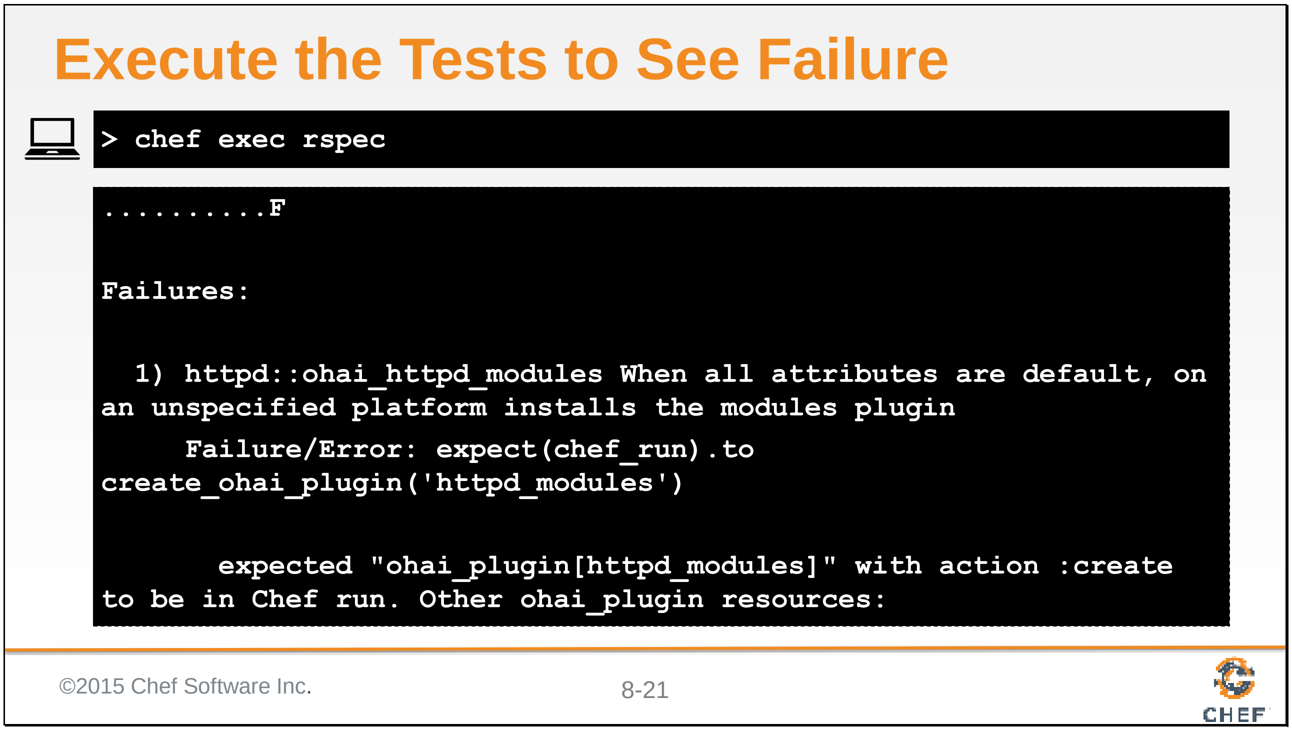
~/cookbooks/httpd/spec/unit/recipes/ohai_httpd_modules_spec.rb

```
# ... 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('httpd_modules')
    end
  end
end
```

Slide 21



Execute the Tests to See Failure

```
> chef exec rspec
```

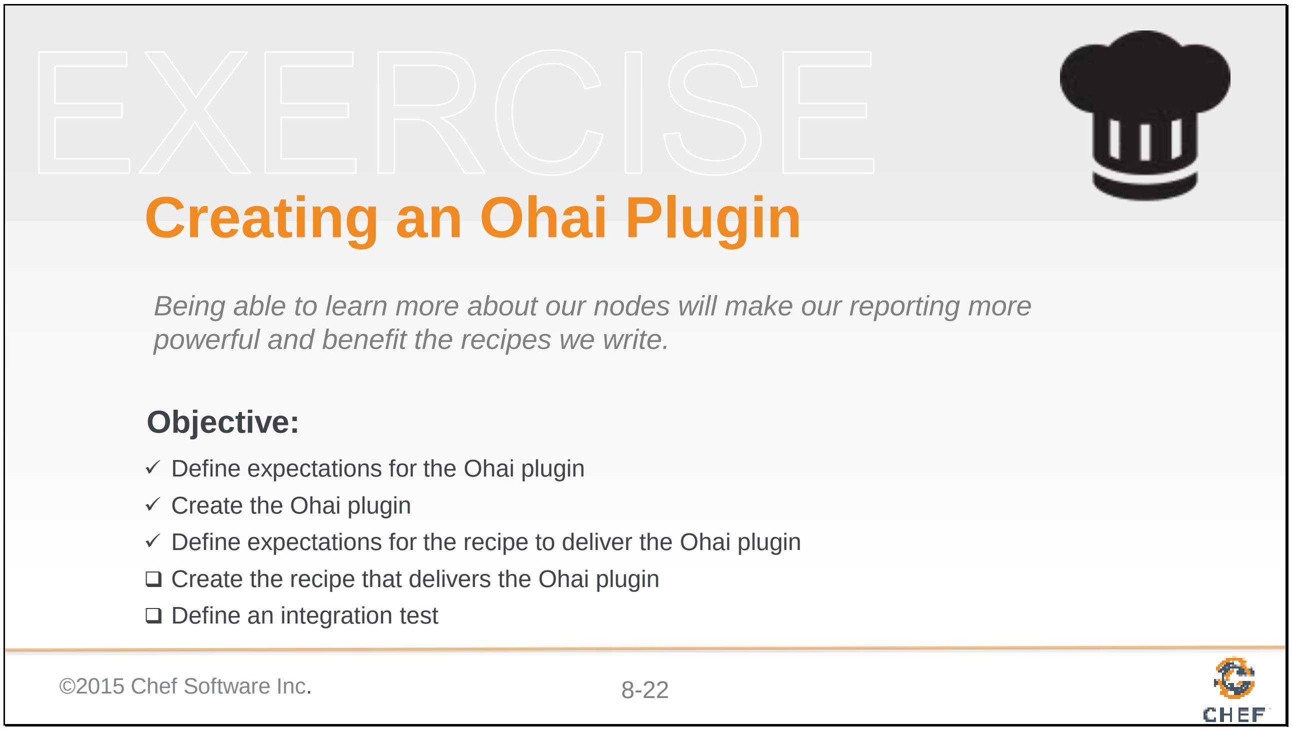
```
..........F

Failures:

  1) httpd::ohai_httpd_modules When all attributes are default, on
an unspecified platform installs the modules plugin
     Failure/Error: expect(chef_run).to
create_ohai_plugin('httpd_modules')

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

Slide 22



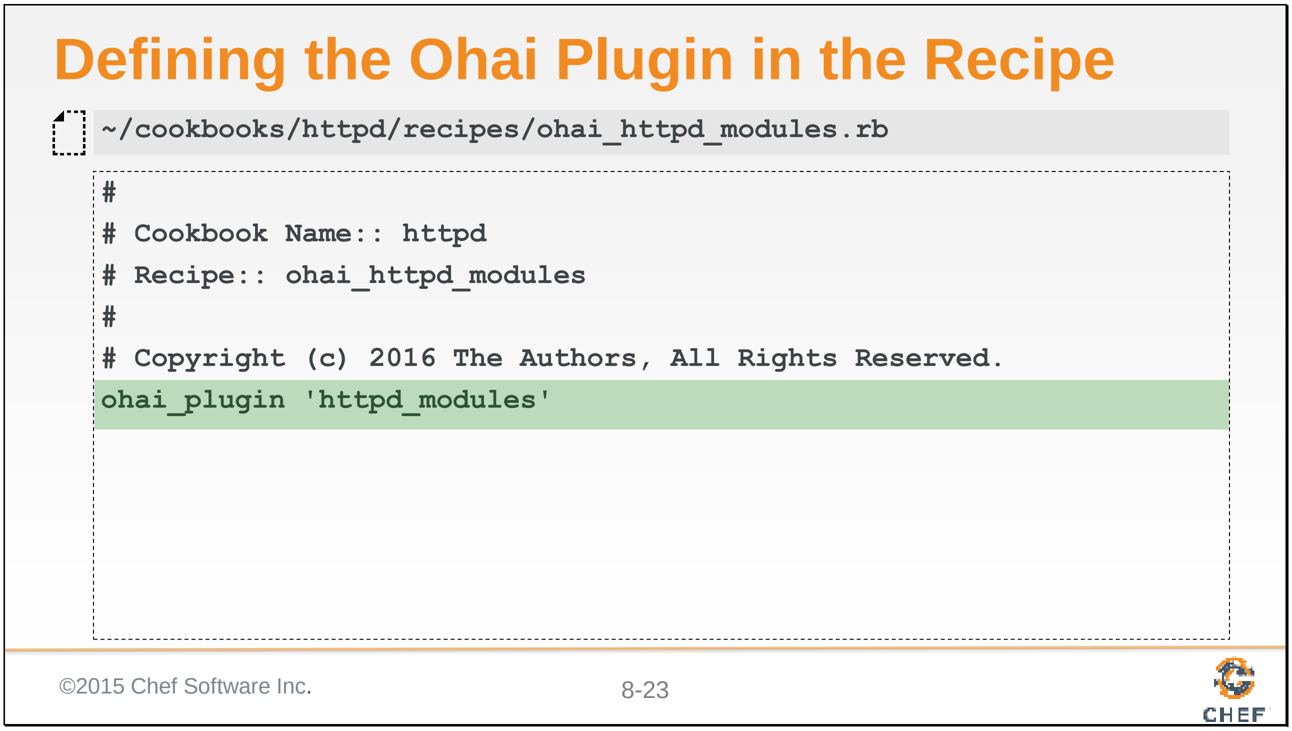
EXERCISE

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

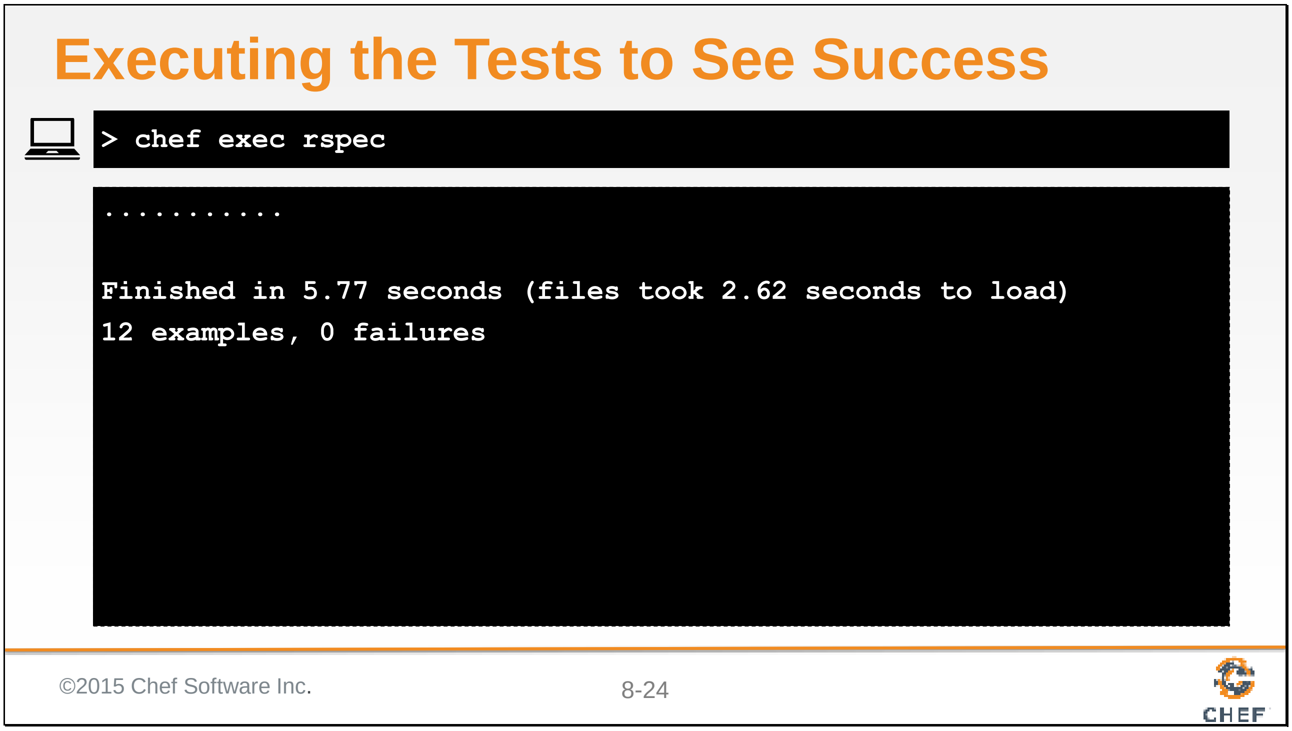


Defining the Ohai Plugin in the Recipe

~/cookbooks/httpd/recipes/ohai_httpd_modules.rb

```
#
# Cookbook Name:: httpd
# Recipe:: ohai_httpd_modules
#
# Copyright (c) 2016 The Authors, All Rights Reserved.
ohai_plugin 'httpd_modules'
```

Slide 24



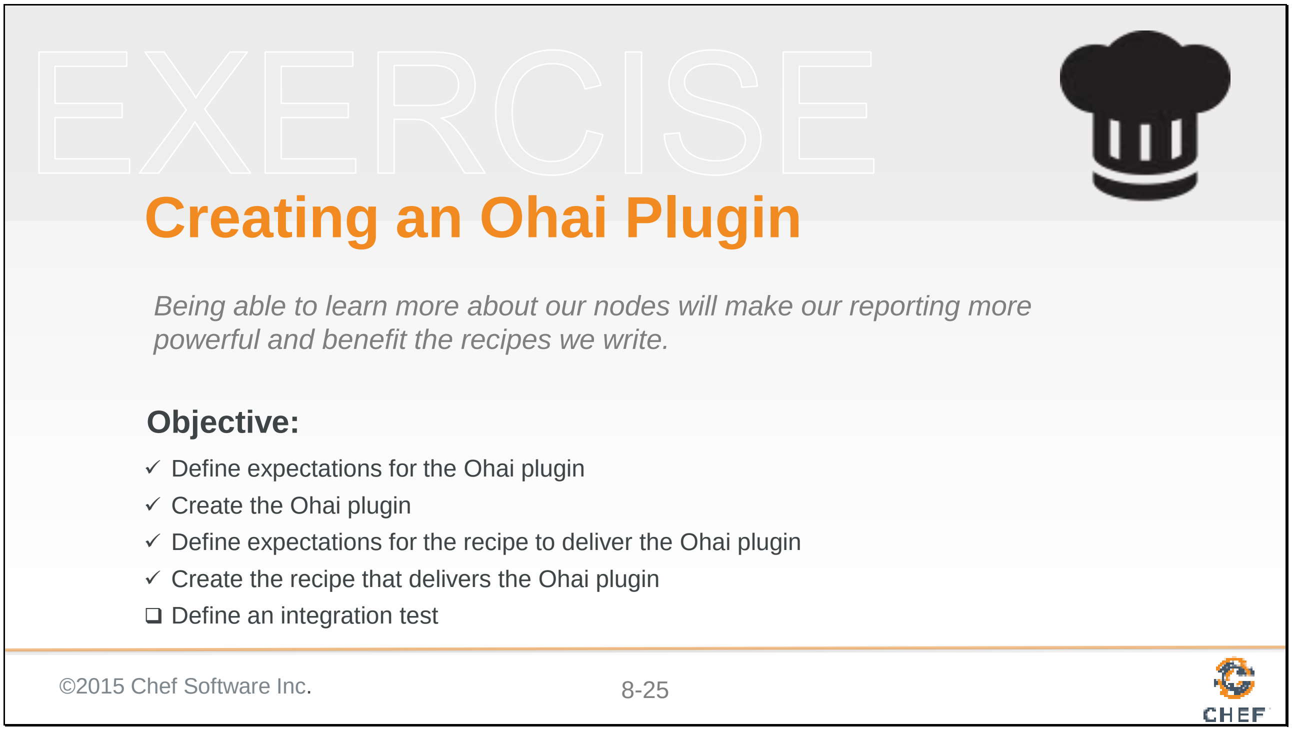
Executing the Tests to See Success

```
> chef exec rspec
```

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

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

Slide 25



EXERCISE

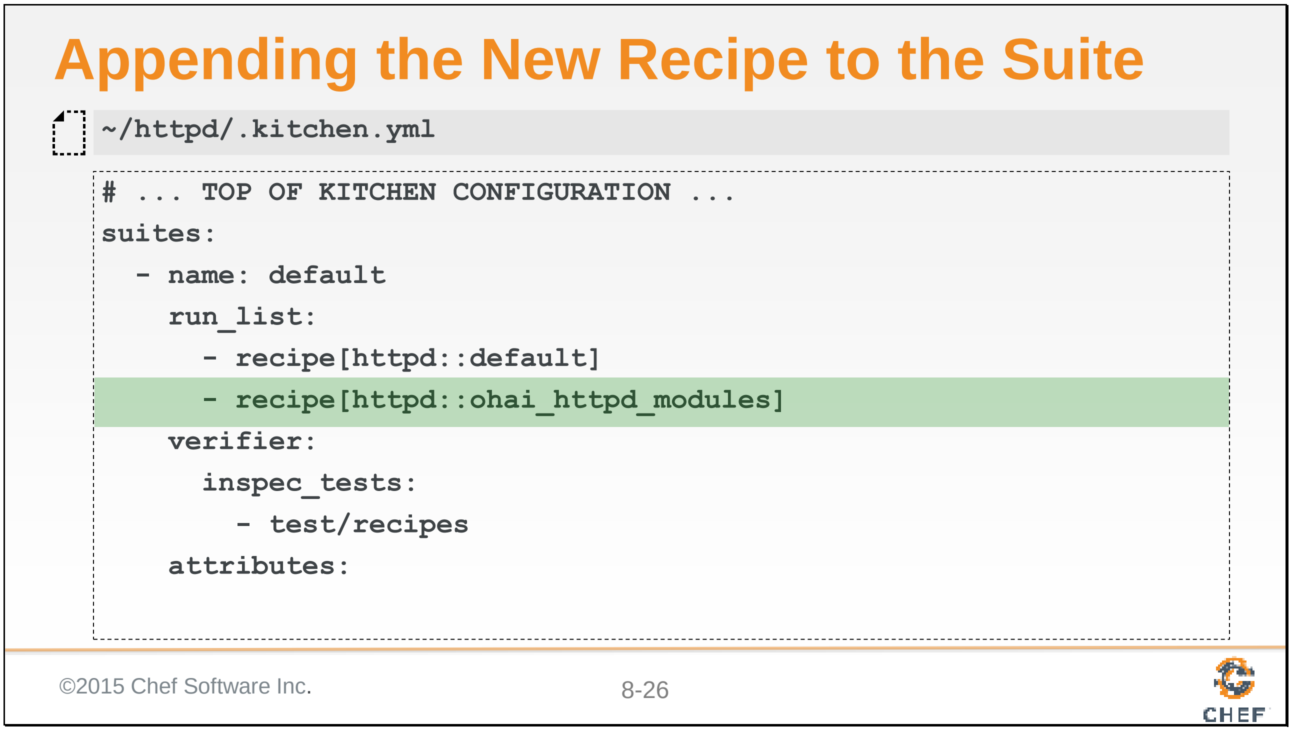
Creating an Ohai Plugin

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

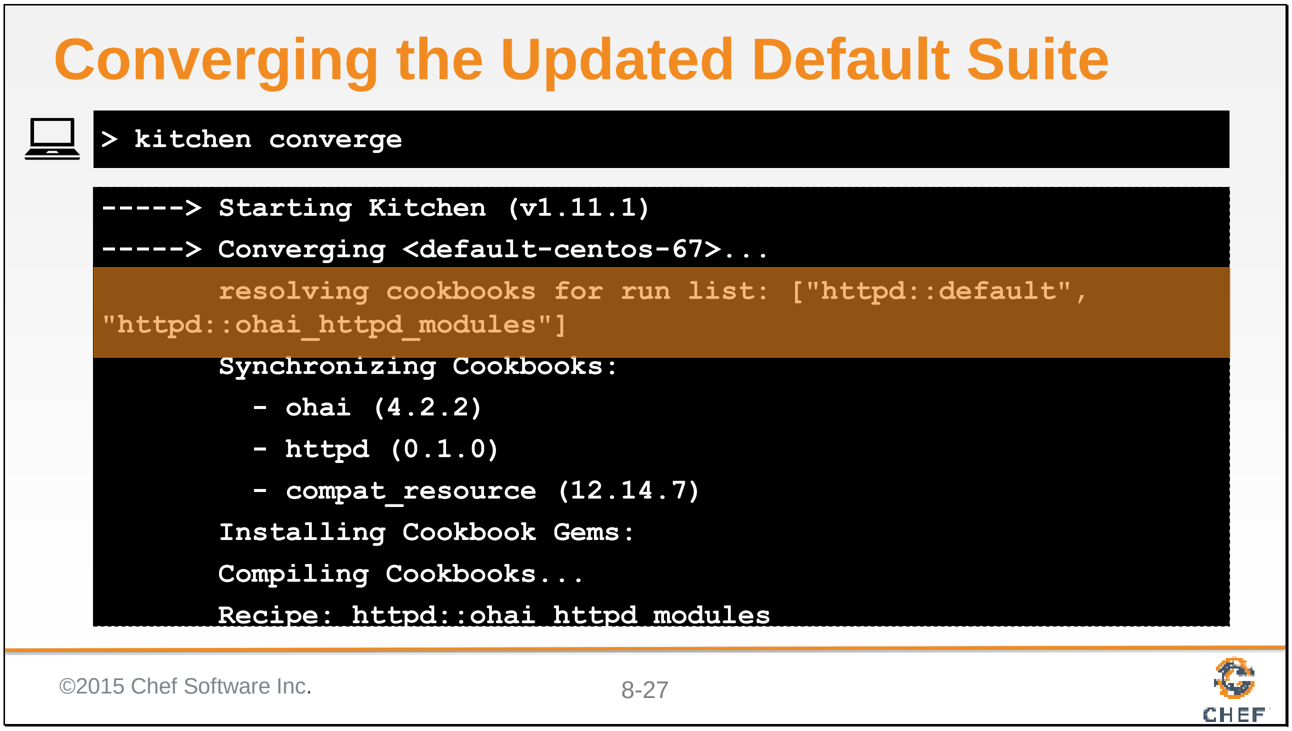
Slide 26



Appending the New Recipe to the Suite

~/httpd/.kitchen.yml

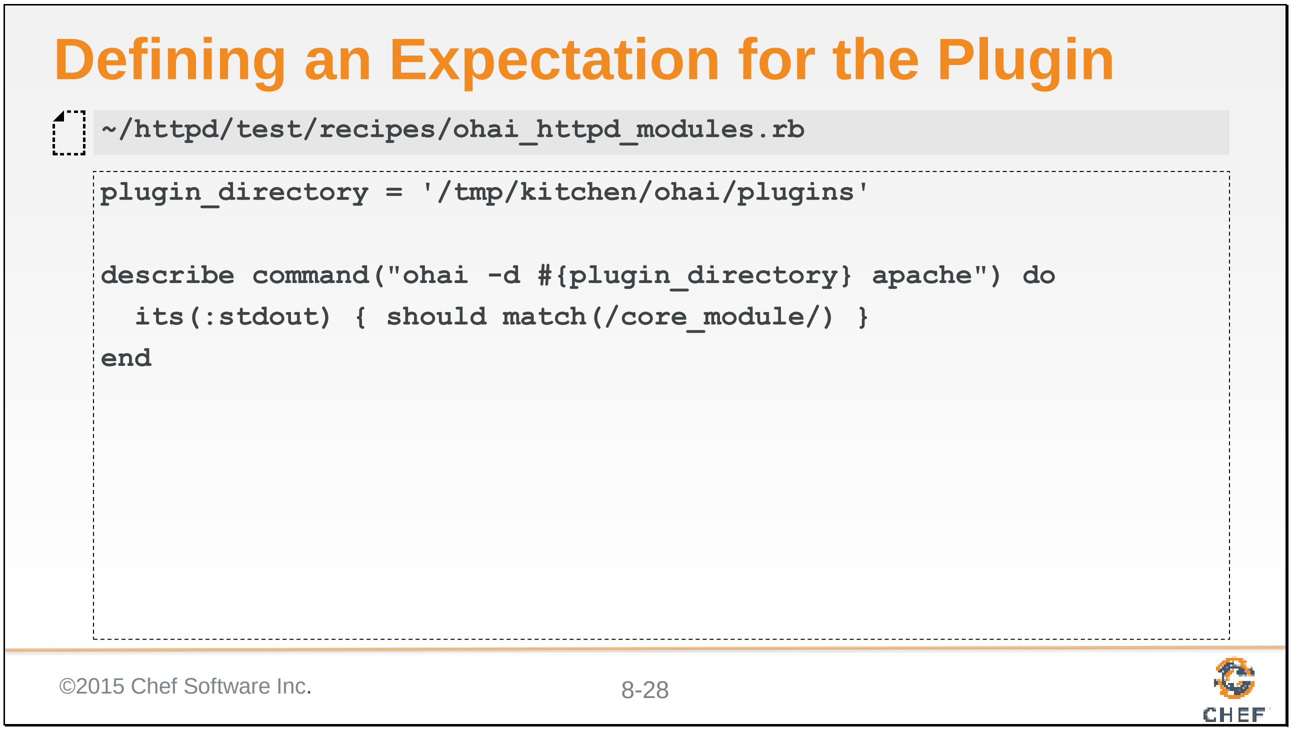
```
# ... TOP OF KITCHEN CONFIGURATION ...
suites:
  - name: default
    run_list:
      - recipe[httpd::default]
      - recipe[httpd::ohai_httpd_modules]
    verifier:
      inspec_tests:
        - test/recipes
    attributes:
```



Converging the Updated Default Suite

```
> kitchen converge
```

```
-----> Starting Kitchen (v1.11.1)
-----> Converging <default-centos-67>...
       resolving cookbooks for run list: ["httpd::default",
"httpd::ohai_httpd_modules"]
       Synchronizing Cookbooks:
         - ohai (4.2.2)
         - httpd (0.1.0)
         - compat_resource (12.14.7)
       Installing Cookbook Gems:
       Compiling Cookbooks...
       Recipe: httpd::ohai_httpd_modules
```



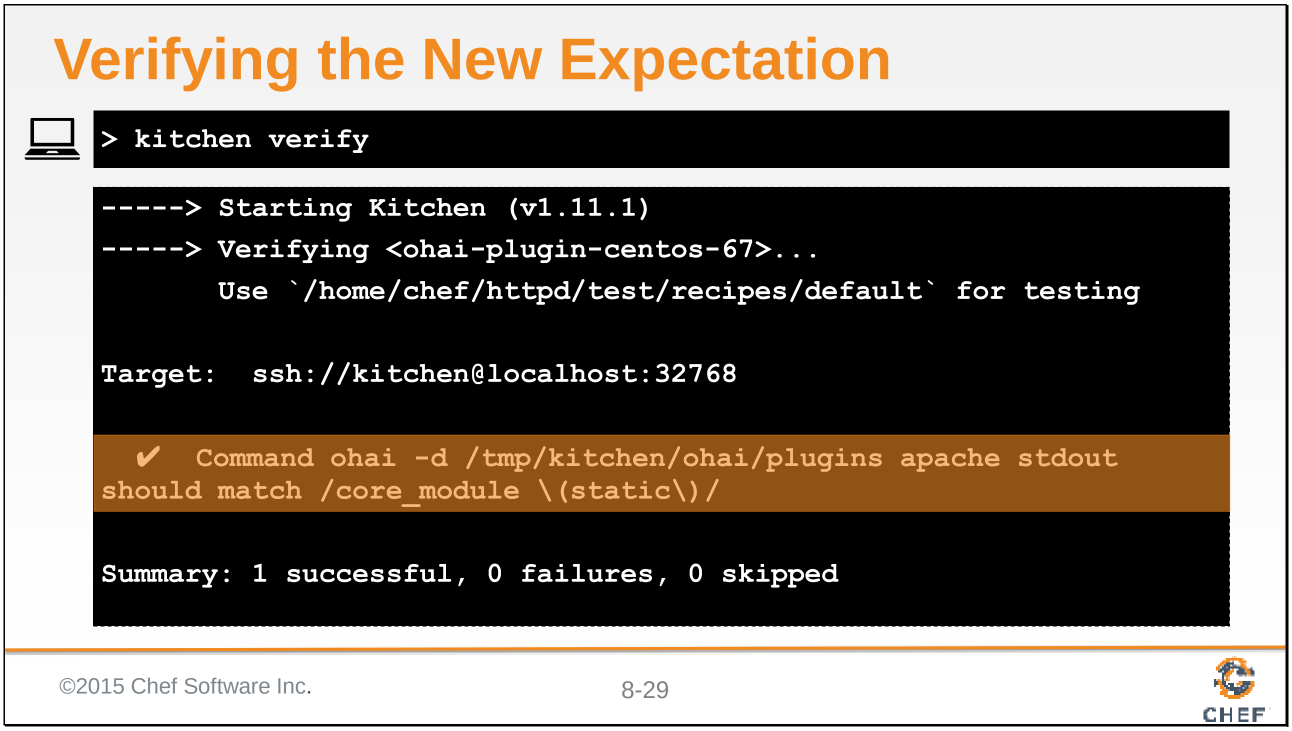
Defining an Expectation for the Plugin

~/httpd/test/recipes/ohai_httpd_modules.rb

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

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

Slide 29



Verifying the New Expectation

```
> kitchen verify
```

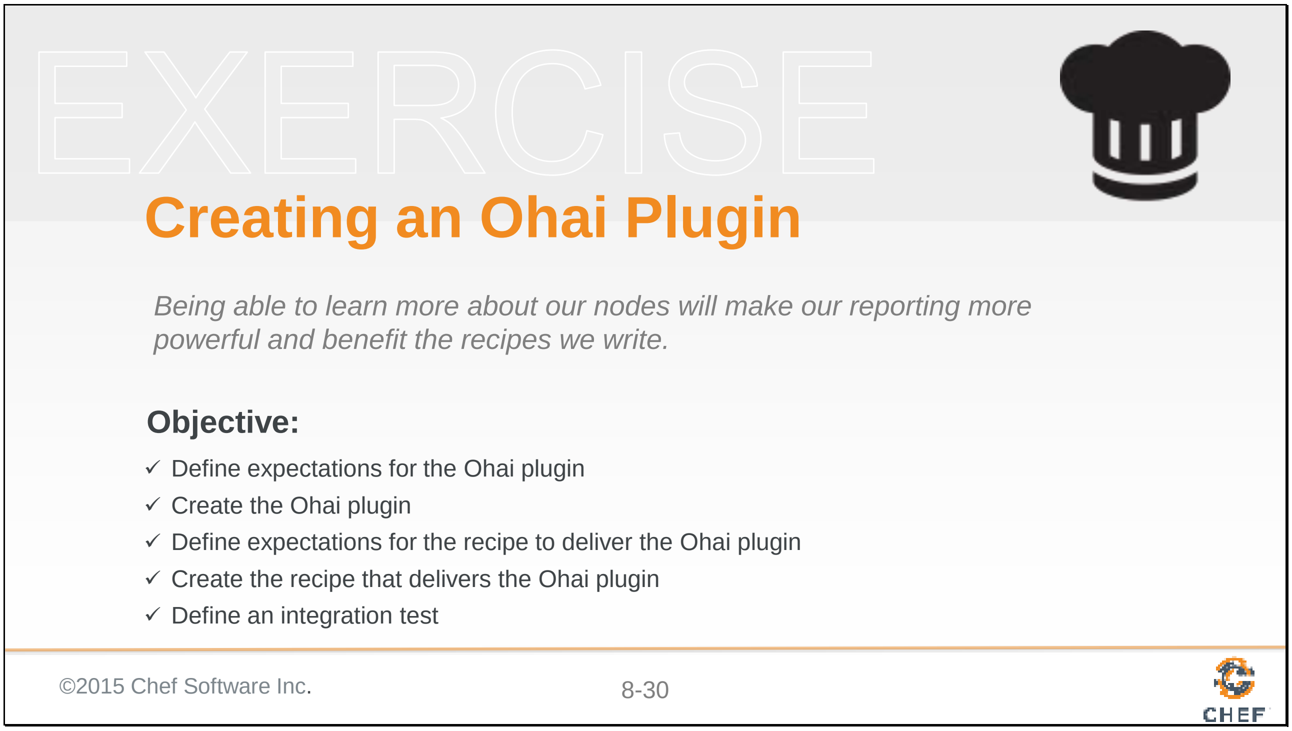
```
-----> Starting Kitchen (v1.11.1)
-----> Verifying <ohai-plugin-centos-67>...
       Use `/home/chef/httpd/test/recipes/default` for testing

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
```

Slide 30



EXERCISE

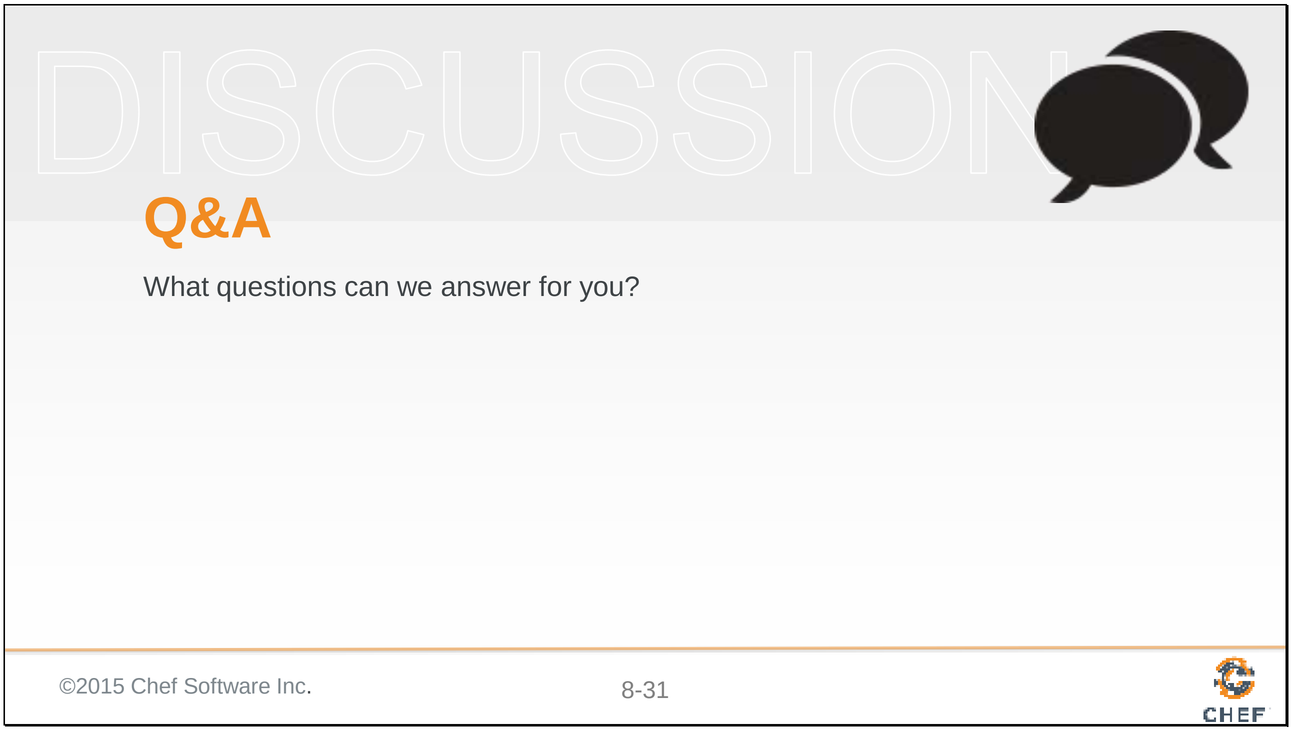
Creating an Ohai Plugin

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

Slide 31



DISCUSSION

Q&A

What questions can we answer for you?

Slide 32

CHEF™