After completing this lab, you should be able to **Create a tested Ohai plugin**

Ohai is packaged with a core set of plugins

- These plugins are automatically loaded when executing Ohai.
- These plugins provide the attributes we see in the JSON output (e.g. ipaddress, hostname, memory, cpu).
- All the data that Ohai collects is generated by plugins.

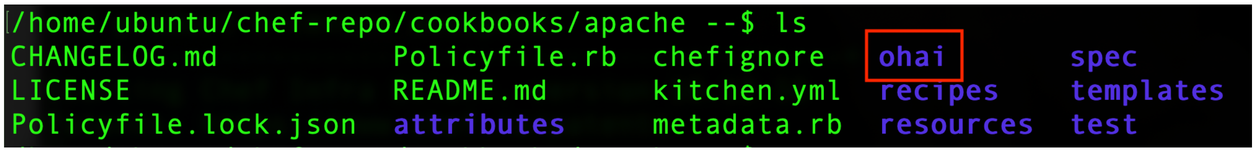
A cookbook consists of recipes and other optional components as files or directories. One of the components is **ohai**. Refer to this link for more information

https://docs.chef.io/cookbooks/#components

NOTE : Custom Ohai plugins can be written to load additional information about your nodes to be used in recipes. This requires Chef Infra Server 12.18.14 or later.

Lets create a plugin to get the operating system and internal IP addresses of VM.

1. Create a new directory with name **ohai** under apache cookbook:

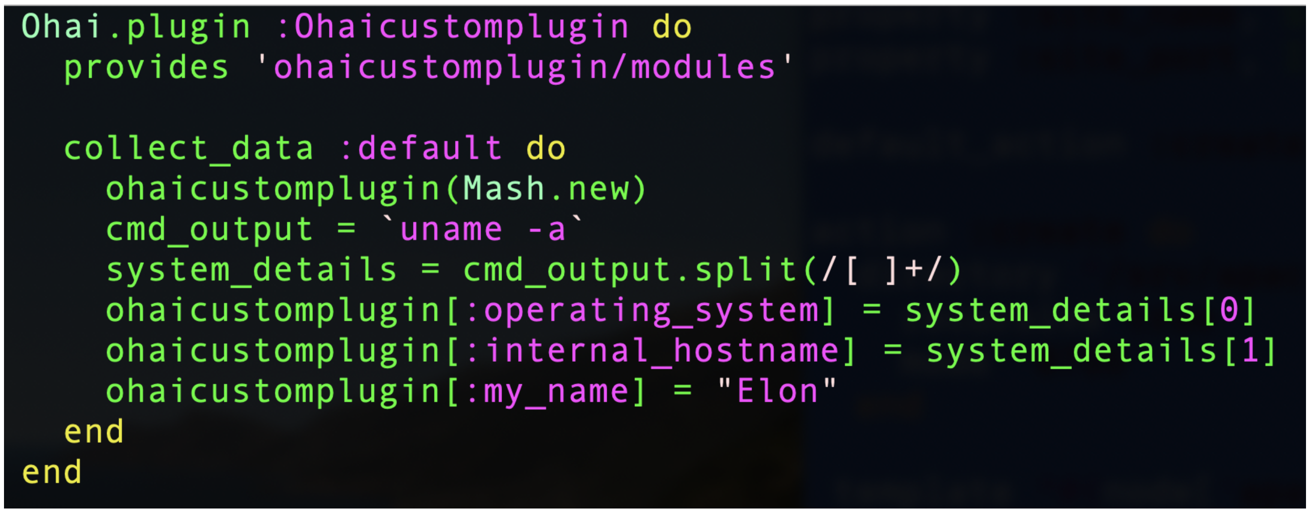


```
/home/ubuntu/chef-repo/cookbooks/apache --$ ls
CHANGELOG.md           Policyfile.rb  chefignore   ohai      spec
LICENSE                README.md      kitchen.yml  recipes   templates
Policyfile.lock.json   attributes     metadata.rb  resources test
```

2. Create a file to write a plugin code.

   $ vi ohai/ohai_custom_plugin.rb

   Add below code to this file:



```
Ohai.plugin :Ohaicustomplugin do
  provides 'ohaicustomplugin/modules'

  collect_data :default do
    ohaicustomplugin(Mash.new)
    cmd_output = `uname -a`
    system_details = cmd_output.split(/[ ]+/)
    ohaicustomplugin[:operating_system] = system_details[0]
    ohaicustomplugin[:internal_hostname] = system_details[1]
    ohaicustomplugin[:my_name] = "Elon"
  end
end
```

Code explanation :

1. A plugin starts with invoking a method on the **Ohai class** with a single parameter. That parameter provided is the symbol name of the plugin. All Ohai plugins must have a symbol name with the first letter capitalized.

   In our example, plugin name is : **Ohaicustomplugin**

2. The remainder of the plugin is defined within the block of the 'plugin' method. The 'provides' method specifies what attribute or attributes the plugin will be added to the node object.

3. The **'collect_data'** method defines a block which contains the code that is executed on all platforms. This block of code will oftentimes set the values of the attributes the plugin provides.

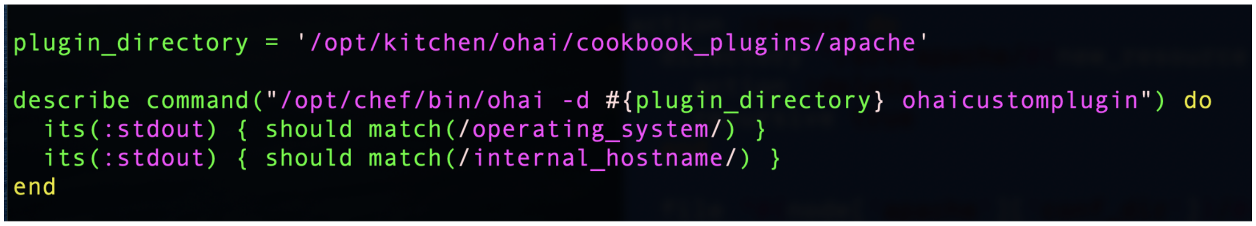
4. Core logic defined in **collect_data** method is getting executed as follow:
   a. Shell Command is executed “uname -a” (This command return information of machine)
   b. Command output is splitted using space.
   c. System details are getting saved in attributes.
      i. Operating_system
      ii. internal_hostname

Save this file and move to the next step.

3. Add an integration test case to test the custom plugin.

   $ vi ~/chef-repo/cookbooks/apache/test/integration/default/default_test.rb

   Add a new test case to test if the correct node attributes are present in the node object



```
plugin_directory = '/opt/kitchen/ohai/cookbook_plugins/apache'

describe command("/opt/chef/bin/ohai -d #{plugin_directory} ohaicustomplugin") do
  its(:stdout) { should match(/operating_system/) }
  its(:stdout) { should match(/internal_hostname/) }
end
```

4. Use ChefSpec and **$ kitchen verify** to verify your changes.

5. Validate the ohai plugin by logging in to the converged **centos-7** kitchen

6. Execute this command to see and validate the included attributes:

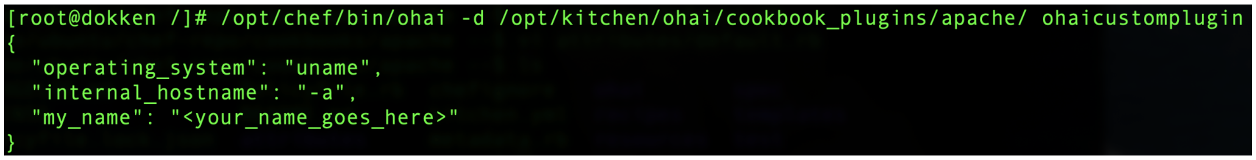
   $ /opt/chef/bin/ohai -d /opt/kitchen/ohai/cookbook_plugins/apache/ ohaicustomplugin

Command breakdown:

/opt/chef/bin/ohai == path to the ohai command

-d /opt/kitchen/ohai/cookbook_plugins/apache/ == pointer to the custom plugin path (versus the location of the core plugins that come with Ohai)

ohaicustomplugin == name of the plugin to display. without this, ALL ohai data would be displayed. Adding this specifies to ONLY display this one plugin. You could also run this command with something like "platform" or "ipaddress" at the end of the command and only see those attributes



```
[root@dokken /]# /opt/chef/bin/ohai -d /opt/kitchen/ohai/cookbook_plugins/apache/ ohaicustomplugin
{
  "operating_system": "uname",
  "internal_hostname": "-a",
  "my_name": "<your_name_goes_here>"
}
```

If you see the attributes as expected, this means that your plugin created the Ohai attributes, and when manually running Ohai, they are displayed.

Notify your instructor that you are done with the lab

END OF LAB