Objectives:

- Create a custom resource in the apache cookbook

Custom resources help clarify recipes by hiding some implementation details which makes the code more concise.

A **VHOST** (virtual host) is an Apache configuration that allows multiple websites to run on a single web server. By using vhosts, one web server could serve hundreds or even thousands of different websites, based on each vhost's unique port number (eg. 8080 instead of the default 80).

Each request to a predefined, unique port number serves up its own **index.html** page, which is stored in a directory defined by the vhost configuration file
(eg. **/srv/apache/admins/index.html**)

Creating a new **VHOST:**

- Create a directory to house the new **index.html** page for this specific vhost configuration

- Add a configuration file defining:

  - The file locations from which the Apache service will serve files

  - The port number on which Apache will listen for requests

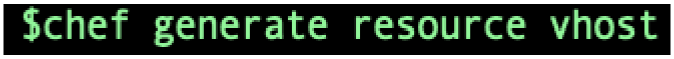
- Create the **vhost specific index.html** file

  - A custom resource is able to be reused and allows for multiple vhosts to be created quickly on the web server.

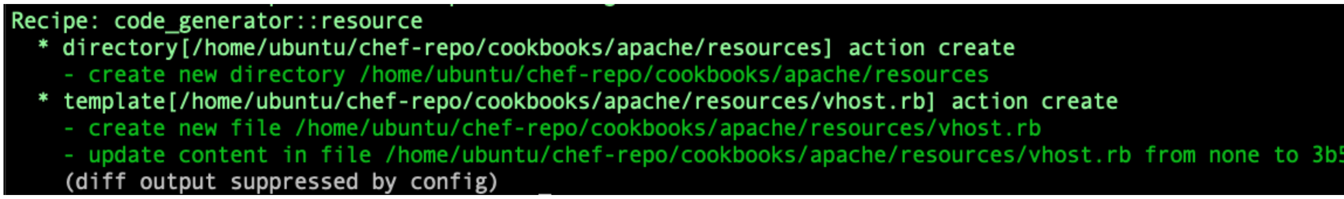
  - The custom resources are similar to those defined in the user cookbook’s default recipe.

Generating a custom resource using chef commands:

1. **From cookbooks/apache, execute:**



```
$chef generate resource vhost
```

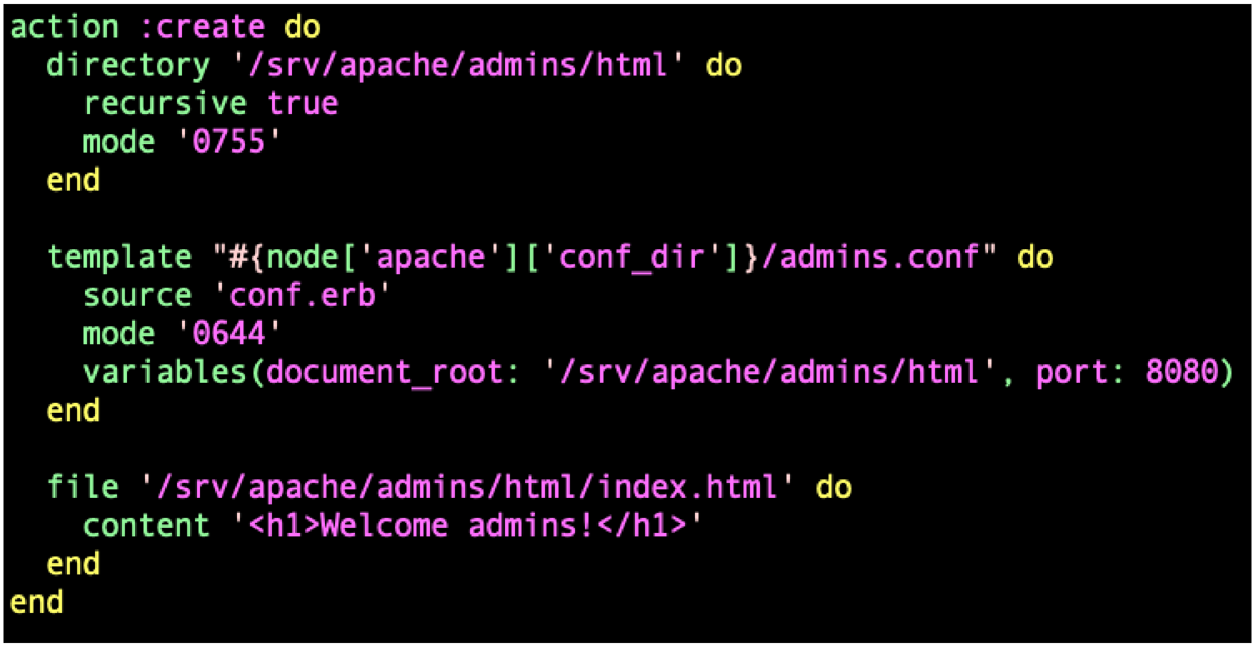


```
Recipe: code_generator::resource
  * directory[/home/ubuntu/chef-repo/cookbooks/apache/resources] action create
    - create new directory /home/ubuntu/chef-repo/cookbooks/apache/resources
  * template[/home/ubuntu/chef-repo/cookbooks/apache/resources/vhost.rb] action create
    - create new file /home/ubuntu/chef-repo/cookbooks/apache/resources/vhost.rb
    - update content in file /home/ubuntu/chef-repo/cookbooks/apache/resources/vhost.rb from none to 3b5
    (diff output suppressed by config)
```

The first implementation for the custom resource will create an admin site exactly as was done in the default recipe for the apache cookbook.

These values are hard-coded to the admin site for now.

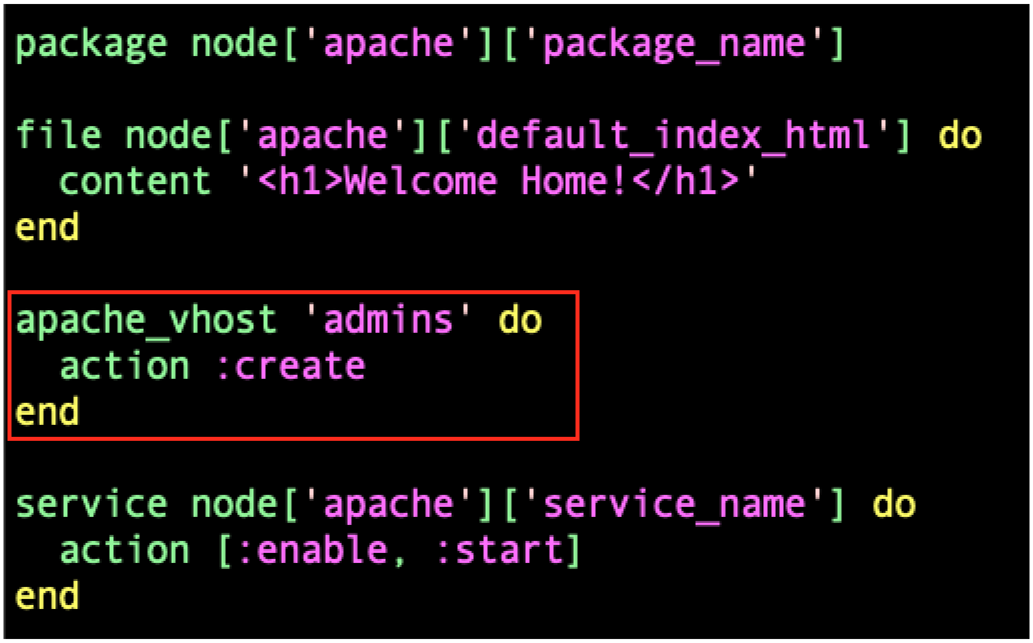
2. Update **/apache/resources/vhost.rb** as follows:



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

  template "#{node['apache']['conf_dir']}/admins.conf" do
    source 'conf.erb'
    mode '0644'
    variables(document_root: '/srv/apache/admins/html', port: 8080)
  end

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

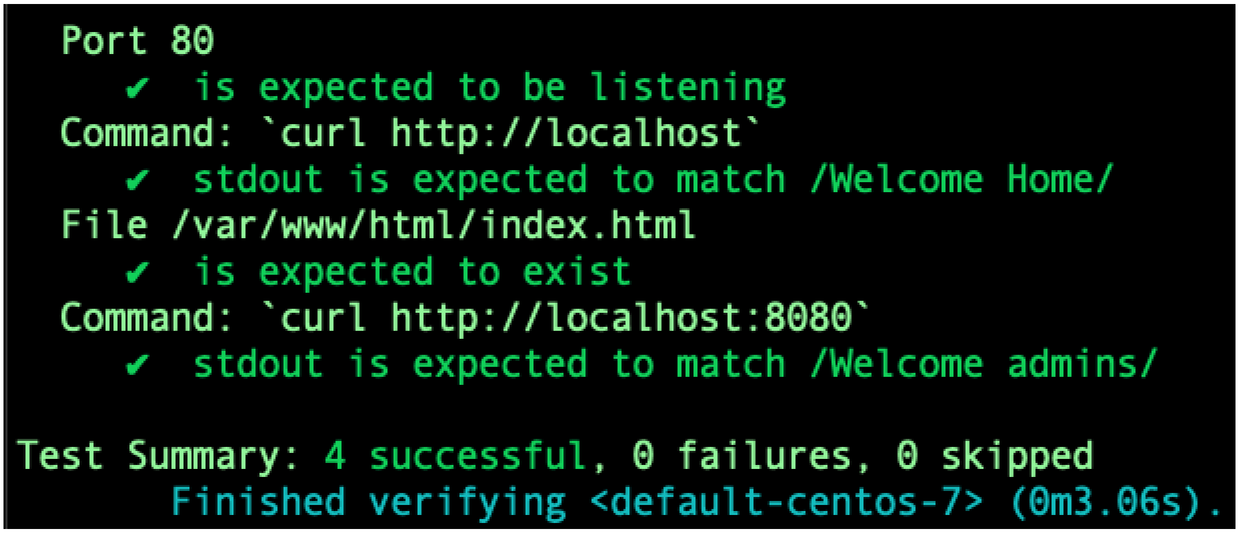
3. Add **apache_vhost** as a custom resource to **apache/recipes/default.rb**

4. Update apache/recipes/default.rb as follows:

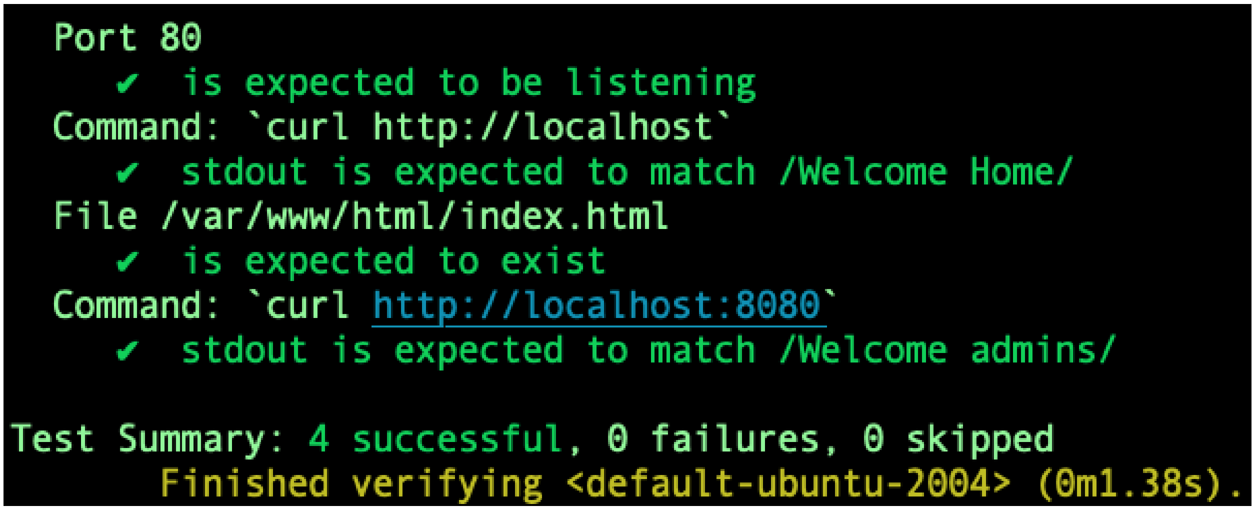


```
package node['apache']['package_name']

file node['apache']['default_index_html'] do
  content '<h1>Welcome Home!</h1>'
end

apache_vhost 'admins' do
  action :create
end

service node['apache']['service_name'] do
  action [:enable, :start]
end
```

5. Verify changes using Test Kitchen, it should pass all tests for both the Ubuntu and Centos platforms



```
  Port 80
     ✔  is expected to be listening
  Command: `curl http://localhost`
     ✔  stdout is expected to match /Welcome Home/
  File /var/www/html/index.html
     ✔  is expected to exist
  Command: `curl http://localhost:8080`
     ✔  stdout is expected to match /Welcome admins/

Test Summary: 4 successful, 0 failures, 0 skipped
       Finished verifying <default-centos-7> (0m3.06s).
```



```
  Port 80
     ✔  is expected to be listening
  Command: `curl http://localhost`
     ✔  stdout is expected to match /Welcome Home/
  File /var/www/html/index.html
     ✔  is expected to exist
  Command: `curl http://localhost:8080`
     ✔  stdout is expected to match /Welcome admins/

Test Summary: 4 successful, 0 failures, 0 skipped
       Finished verifying <default-ubuntu-2004> (0m1.38s).
```

The custom resource was created successfully if both sets of test summaries returned with 4 successful tests

Notify your instructor that you are done with the lab

END OF LAB