

Cookbook Attributes, Attribute Files and Dependencies

Setting Attributes within a Cookbook

Objectives

After completing this module, you should be able to

- Explain where cookbook attributes reside
- Create a wrapper cookbook
- Configure dependencies between cookbooks
- Create a new policyfile
- Upload the new policyfile and converge the Windows node

CONCEPT



Attribute Files

The Node object contains many automatic attributes generated by OHAI.

You can also maintain attributes within a cookbook.

These are like variables or parameters for your cookbook and allow recipes to be data driven.

<https://docs.chef.io/attributes.html>

CONCEPT



Best Practices

- ❖ Well-written cookbooks change behavior based on attributes.
- ❖ Ideally, you don't have to modify the contents of a cookbook to use it for your specific use case.
- ❖ Look at the attributes directory for things you can override through roles to affect behavior of the cookbook.
- ❖ Of course, well written cookbooks have sane defaults, and a README to describe all this.

Setting Attributes in Attribute Files

Cookbook attributes are set in the attributes file

`./cookbooks/<cookbook>/attributes/default.rb`

Format is:

```
default["attributename"] = "value"
```

We'll look at precedence later.

Example: Setting package name to an attribute



```
cookbooks/apache/attributes/default.rb
```

```
default['apache']['package_name'] = 'httpd'
```

```
cookbooks/apache/recipes/default.rb
```

```
package node['apache']['package_name'] do  
  action :install  
end
```

We can set the name of a particular package to an attribute and then call that attribute within a recipe

Example: Setting package name to an attribute



cookbooks/apache/attributes/default.rb

```
case node['platform']  
when 'ubuntu'  
  default['apache']['package_name'] = 'apache2'  
else  
  default['apache']['package_name'] = 'httpd'  
end
```

Implementing conditional statements allows us to alter the control flow permitting our cookbooks to be data driven.

CONCEPT



Reconfigure Welcome Message

Currently a welcome message is hard coded in both web server cookbooks.

What if we wanted to display a message that includes our company name utilizing a node attribute?

How could we implement this node attribute within both our 'myiis' and 'apache' cookbooks?

EXERCISE



GL: Reconfigure Welcome Message

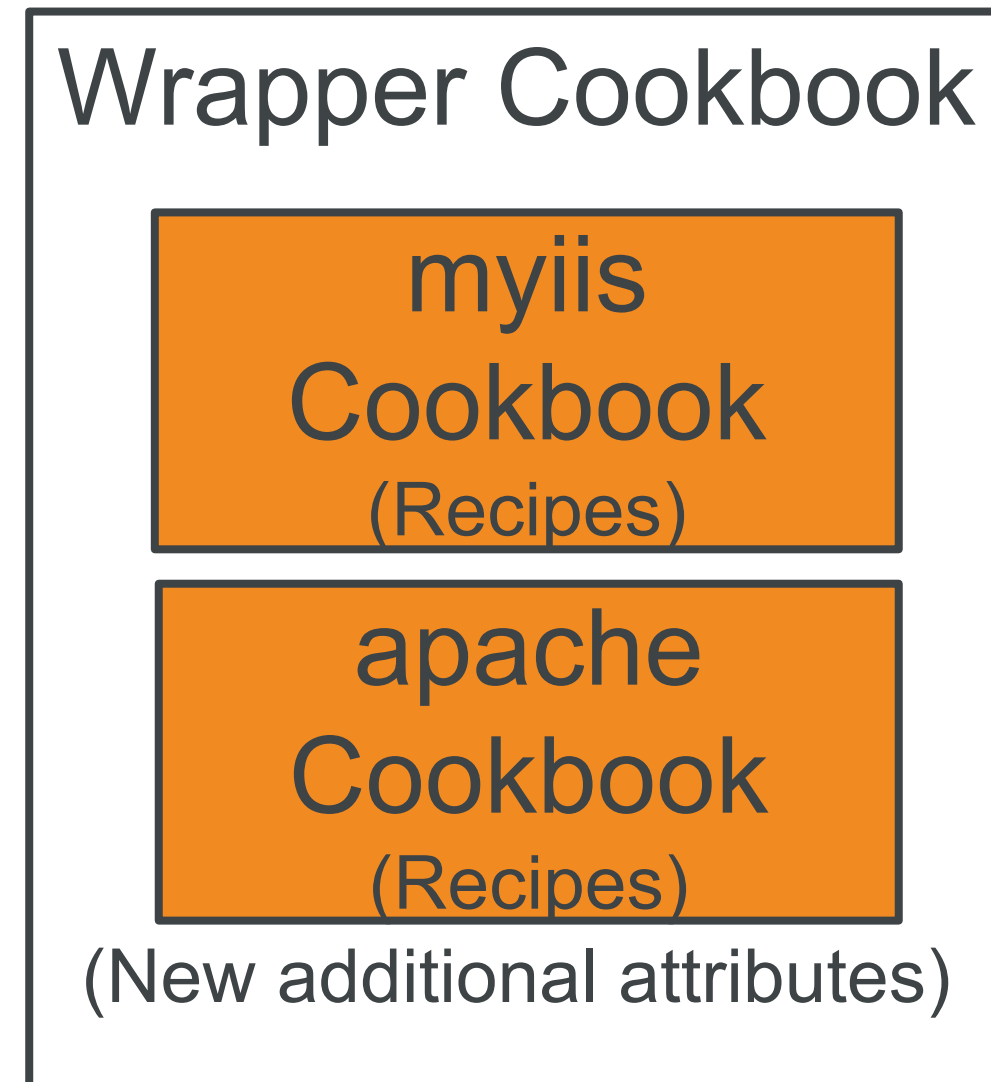
So we want both our web server cookbooks to display our company name...

- Objective:**
- ☐ Create a 'company_web' wrapper cookbook that can apply either the 'apache' or 'myiis' default recipe based on platform
 - ☐ Create a node attribute that contains your company name
 - ☐ Implement the edit_resource method to update the template resource for both the 'myiis' and 'apache' server recipes
 - ☐ Create Policyfile and lock.
 - ☐ Upload Policyfile.lock to the Chef server
 - ☐ Converge the node

Wrapper Cookbooks

A wrapper cookbook is a new cookbook that encapsulates the functionality of the original cookbook(s).

It can access all of the recipes, cookbook components, and attributes found in the original cookbook(s) and implement them in new ways.



<https://docs.chef.io/supermarket.html#wrapper-cookbooks>

<https://www.chef.io/blog/2013/12/03/doing-wrapper-cookbooks-right/>

GL: Generate the Wrapper Cookbook



```
$ cd ~/chef-repo
```

```
$ chef generate cookbook cookbooks/company_web
```

```
Generating cookbook company_web  
- Ensuring correct cookbook content
```

```
Your cookbook is ready. To setup the pipeline, type `cd  
cookbooks/company_web`, then run `delivery init`
```

Do this on your laptop

Create Dependency on apache

 ~/chef-repo/cookbooks/company_web/metadata.rb

```
name 'company_web'
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'All Rights Reserved'
description 'Installs/Configures company_web'
long_description 'Installs/Configures company_web'
version '0.1.0'
chef_version '>= 14.0'
depends 'apache'
depends 'myiis'
```

GL: Include Recipe Based on Platform



```
~/chef-repo/cookbooks/company_web/recipes/default.rb
```

```
#  
# Cookbook:: company_web  
# Recipe:: default  
#  
# Copyright:: 2019, The Authors, All Rights Reserved.  
  
case node['platform']  
when 'windows'  
  include_recipe 'myiis::default'  
else  
  include_recipe 'apache::default'  
end
```

EXERCISE



GL: Reconfigure Welcome Message

So we want both our web server cookbooks to display our company name...

- Objective:**
- ✓ Create a 'company_web' wrapper cookbook that can apply either the 'apache' or 'apache' default recipe based on platform
 - ☐ Create a node attribute that contains your company name
 - ☐ Implement the edit_resource method to update the template resource for both the 'apache' and 'apache' server recipes
 - ☐ Upload cookbook to the Chef server
 - ☐ Update the run list of the iis_web node to use the default recipe of the company_web cookbook and converge the node

GL: Generate the default Attribute File



```
$ cd ~/chef-repo  
$ chef generate --help
```

Available generators:

app	Generate an application repo
cookbook	Generate a single cookbook
recipe	Generate a new recipe
attribute	Generate an attributes file
template	Generate a file template
file	Generate a cookbook file

GL: Generate the default Attribute File



```
$ chef generate attribute --help
```

```
Usage: chef generate attribute [path/to/cookbook] NAME [options]
```

<code>-C, --copyright COPYRIGHT</code>	Name of the copyright holder - defaults to 'The Authors'
<code>-m, --email EMAIL</code>	Email address of the author - defaults to 'you@example.com'
<code>-a, --generator-arg KEY=VALUE</code>	Use to set arbitrary attribute KEY to VALUE in the code_generator cookbook
<code>-h, --help</code>	Show this message

GL: Generate the default Attribute File



```
$ chef generate attribute cookbooks/company_web default
```

```
Recipe: code_generator::attribute
```

```
  * directory[cookbooks/company_web/attributes] action create
```

```
    - create new directory cookbooks/company_web/attributes
```

```
  * template[cookbooks/company_web/attributes/default.rb] action  
create
```

```
    - create new file cookbooks/company_web/attributes/default.rb
```

```
    - update content in file
```

```
cookbooks/company_web/attributes/default.rb from none to e3b0c4
```

GL: Set the Company Name as an Attribute

 cookbooks/company_web/attributes/default.rb

```
default['company_web']['company_name'] = 'My company'
```

Cookbook name

Attribute name

** Not required, you can just give attribute name
But this is a standard practice

EXERCISE



GL: Reconfigure Welcome Message

So we want both our web server cookbooks to display our company name...

- Objective:**
- ✓ Create a 'company_web' wrapper cookbook that can apply either the 'apache' or 'apache' default recipe based on platform
 - ✓ Create a node attribute that contains your company name
 - ☐ Implement the edit_resource method to update the template resource for both the 'apache' and 'apache' server recipes
 - ☐ Create Policyfile and lock.
 - ☐ Upload Policyfile.lock to the Chef server
 - ☐ Converge the node

CONCEPT



Using the `company_name` Attribute

We are now able to apply a different default recipe based on whether the node's platform is Windows or Centos, but how do we update the respective template file to display the `company_name` attribute for both the 'apache' and 'apache' cookbooks?

edit_resource



A recipe can find a resource in the resource collection, and then edit it by using the `edit_resource` method. If a resource block with the same name exists in the resource collection, it will be updated with the contents of the resource block.

https://docs.chef.io/dsl_recipe.html#edit-resource

GL: View the server Recipes

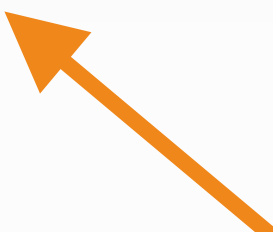


/apache/recipes/server.rb

```
powershell_script 'Install IIS' do
  code 'Add-WindowsFeature Web-Server'
end

template 'c:\inetpub\wwwroot\Default.htm' do
  source 'Default.htm.erb'
end

service 'w3svc' do
  action [:enable, :start]
end
```



/apache/recipes/server.rb

```
package 'httpd'

template '/var/www/html/index.html' do
  source 'index.html.erb'
end

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



We want to use a new source for the template resource for both our cookbooks

GL: Edit the Template resource for apache



```
~/chef-repo/cookbooks/company_web/recipes/default.rb
```

```
case node['platform']  
when 'windows'  
  include_recipe 'apache::default'  
  
  edit_resource(:template, 'c:\inetpub\wwwroot\Default.htm') do  
    source 'homepage.erb'  
    cookbook 'company_web'  
  end  
  
#Else Statement...
```

GL: Edit the Template resource for apache



```
~/chef-repo/cookbooks/company_web/recipes/default.rb
```

```
#When Statement...
```

```
else
```

```
  include_recipe 'apache::default'
```

```
  edit_resource(:template, '/var/www/html/index.html') do
```

```
    source 'homepage.erb'
```

```
    cookbook 'company_web'
```

```
  end
```

```
end
```


GL: View the default Recipe



```
~/chef-repo/cookbooks/company_web/recipes/default.rb
```

```
case node['platform']
when 'windows'
  include_recipe 'myiis::default'

  edit_resource(:template, 'c:\inetpub\wwwroot\Default.htm') do
    source 'homepage.erb'
    cookbook 'company_web'
  end
end

else
  include_recipe 'apache::default'
  edit_resource(:template, '/var/www/html/index.html') do
    source 'homepage.erb'
    cookbook 'company_web'
  end
end
end
```

Do not use this, but use this



GL: Generate the Template file



```
$ chef generate template cookbooks/company_web homepage
```

```
Recipe: code_generator::template
```

```
* directory[cookbooks/company_web/templates] action create
```

```
- create new directory cookbooks/company_web/templates
```

```
* template[cookbooks/company_web/templates/homepage.erb] action create
```

```
- create new file cookbooks/company_web/templates/homepage.erb
```

```
- update content in file cookbooks/company_web/templates/homepage.erb from  
none to e3b0c4
```

```
(diff output suppressed by config)
```

GL: Update the Template File

 cookbooks/company_web/templates/homepage.erb

```
<html>
  <body>
    <h1><%= node['company_web']['company_name'] %> Welcomes You!</h1>
    <h2>PLATFORM: <%= node['platform'] %></h2>
    <h2>HOSTNAME: <%= node['hostname'] %></h2>
    <h2>MEMORY:    <%= node['memory']['total'] %></h2>
    <h2>CPU Mhz:   <%= node['cpu']['0']['mhz'] %></h2>
  </body>
</html>
```

Note: We are adding all this code but we are also highlighting the `['company_web']['company_name']` attributes for the discussion below.

EXERCISE



GL: Reconfigure Welcome Message

So we want both our web server cookbooks to display our company name...

- Objective:**
- ✓ Create a 'company_web' wrapper cookbook that can apply either the 'apache' or 'apache' default recipe based on platform
 - ✓ Create a node attribute that contains your company name
 - ✓ Implement the edit_resource method to update the template resource for both the 'apache' and 'apache' server recipes
 - ☐ Create Policyfile and lock.
 - ☐ Upload Policyfile.lock to the Chef server
 - ☐ Converge the node

CONCEPT



Policyfile.rb and the Policyfile.lock.json

Now that we have our `company_web` cookbook in our chef-repo, we can create our `Policyfile.rb` and then generate our `Policyfile.lock.json` as we discussed in previous modules.

This time we'll name our Policyfile **`company_web`**.

GL: Generate the Policyfile and Name it company_web



```
> cd ~/chef-repo
```

```
> chef generate policyfile company_web
```

```
* template[/Users/sdelfante/chef-repo/company_web.rb] action create
  - create new file /Users/sdelfante/chef-repo/company_web.rb
  - update content in file /Users/sdelfante/chef-repo/company_web.rb from
none to 32a368
```

GL: Verify that the Policyfile Exists



```
> ls (or dir for Windows)
```

```
Policyfile.lock.json README.md      cookbooks  
Policyfile.rb      company_web.rb      roles
```

GL: Edit the New company_web.rb Policyfile

 `~/chef-repo/company_web.rb`

```
#...skipping for brevity...  
# https://docs.chef.io/policyfile.html  
# A name that describes what the system you're building with Chef does.  
name 'company_web'
```

```
# Where to find external cookbooks  
default_source :supermarket
```

```
# run_list: chef-client will run these recipes in the order specified.  
run_list 'company_web::default'
```

```
# Specify a custom source for a single cookbook:  
cookbook 'company_web', path: 'cookbooks/company_web'  
cookbook 'apache', path: 'cookbooks/apache'  
cookbook 'myiis', path: 'cookbooks/myiis'
```

Replace the contents of the company_web.rb below the `#https://docs.chef.io/policyfile.html` line with the code in green.

GL: Generate the company_web.lock.json



```
~/chef-repo> chef install company_web.rb
```

```
Building policy company_web
```

```
Expanded run list: recipe[company_web::default]
```

```
Caching Cookbooks...
```

```
Installing company_web >= 0.0.0 from path
```

```
Installing apache      >= 0.0.0 from path
```

```
Installing apache      >= 0.0.0 from path
```

```
Lockfile written to /Users/sdelfante/chef-repo/company_web.lock.json
```

```
Policy revision id:
```

```
1ca8efa6e5f06d30496441e4c4630685802d78d7885a9bc6245d68f63990aeb9
```

GL: Verify that the company_web.lock.json Exists



```
> ls (or dir for Windows)
```

```
Policyfile.lock.json README.md      company_web.rb      roles  
Policyfile.rb      company_web.lock.json  cookbooks
```

GL: Push the company_web.lock.json to Chef Infra Server



```
~/chef-repo> chef push prod company_web.lock.json
```

```
Uploading policy company_web (55529dbd15) to policy group prod  
Using      apache      0.1.0 (1388ab3a)  
Using      apache      0.2.1 (cd0db3ed)  
Uploaded company_web 0.1.0 (4bce960d)
```

GL: Verify the Policyfile.lock.json is on Chef Infra Server



```
~/chef-repo> chef show-policy
```

```
company_web
```

```
=====
```

```
* prod: 55529dbd15
```

```
apache
```

```
=====
```

```
* prod: 49eef2f1f1
```

Here we can see that the **company_web** policy has been uploaded to Chef Infra Server and is in the **prod** policy group.

Also notice the policy name that was derived from the contents of the **company_web.lock.json**.

EXERCISE



GL: Reconfigure Welcome Message

So we want both our web server cookbooks to display our company name...

- Objective:**
- ✓ Create a 'company_web' wrapper cookbook that can apply either the 'apache' or 'apache' default recipe based on platform
 - ✓ Create a node attribute that contains your company name
 - ✓ Implement the edit_resource method to update the template resource for both the 'apache' and 'apache' server recipes
 - ✓ Create Policyfile and lock.
 - ✓ Upload Policyfile.lock to the Chef server
 - ❑ Converge the node

GL: Apply the company_web Policy to Your Windows Node



```
> knife node policy set node1 prod company_web
```

```
Successfully set the policy on node node1
```

node name

policy_group

policy_name

'knife node policy set' takes 3 arguments:

- * node name
- * policy group
- * Policy name (Name inside file, not filename)

GL: View More Information About Your Node



```
$ knife node show node1
```

```
Node Name:    node1
Policy Name:   company_web
Policy Group:  prod
FQDN:         ip-172-...
IP:           34.195.38.226
Run List:     recipe[apache::default]
Recipes:      apache::default, apache::server
Platform:     centos 7.8.2003
Tags:
```

GL: Converge your Windows Node via winrm



```
$ knife ssh <IP_node1> -m -x centos -i <aws.pem path> "sudo chef-client"
```

...

Synchronizing Cookbooks:

34.195.38.226

34.195.38.226

34.195.38.226 - apache (0.2.1)

34.195.38.226 - apache (0.1.0)

34.195.38.226

34.195.38.226 - company_web (0.1.0)

...

* windows_service[w3svc] action start (up to date)

34.195.38.226 Running handlers complete

34.195.38.226 Chef Infra Client finished, 2/4 resources updated in 06 seconds

34.195.38.226 [2019-07-25T17:26:59+00:00] ERROR: Failed to post reporting data to server (HTTP 400), saving to c:/chef/cache/failed-reporting-data.json

The "Error" can be ignored. Our version of Hosted Chef still has Reporting installed, which has been deprecated in the latest versions of Chef Server.

GL: Verify that the New Node Serves the Page

My company Welcomes You!

PLATFORM: centos

HOSTNAME: ip-172-31-18-33

MEMORY: 1880312kB

CPU Mhz: 2300.117

Lab 12.1

<15 minutes>

Create a wrapper cookbook 'company_web'

Working Directory - chef_repo

- Generate a cookbook company_web - "chef generate cookbook cookbooks/company_web"
- Change company_web/metadata.rb to add dependencies - "depends apache", "depends myiis"
- Add to company_web/default.rb

```
case node['platform']
when 'windows'
  include_recipe 'myiis::default'
else
  include_recipe 'apache::default'
end
```

- Make server.rb in apache to use template, if not already

If you have uploaded apache policy to the server,
And made any change in the policy,
upgrade metadata.rb in apache
Run chef update apache.lock.json

```
case node[:platform]
when "ubuntu", "debian"
  package "apache2" do
    action :install
  end
when "centos", "redhat", "amazon"
  package "httpd" do
    action :install
  end
end

case node[:platform]
when "ubuntu", "debian"
  service "apache2" do
    action [:start, :enable]
  end
when "centos", "redhat", "amazon"
  service "httpd" do
    action [:start, :enable]
  end
end

template '/var/www/html/index.html' do
  source 'index.html.erb'
end
```

Lab 12.2

<60 minutes>

Generate attribute file and use it !!

Work directory is chef-repo

- Create attribute in company_web cookbook - "chef generate attribute cookbooks/company_web default"
- Add this to the attribute file - `default['company_web']['company_name'] = 'Chef Company'`
- Use `edit_resource` in `company_web/recipes/default.rb` to use a template from `company_web`

```
case node['platform']
when 'windows'
  include_recipe 'myiis::default'

  edit_resource(:template, 'c:\inetpub\wwwroot\Default.htm') do
    source 'homepage.erb'
    cookbook 'company_web'
  end
else
  include_recipe 'apache::default'
  edit_resource(:template, '/var/www/html/index.html') do
    source 'homepage.erb'
    cookbook 'company_web'
  end
end
end
```

Homepage.erb ?
Doesn't exist yet

Lab 12.2 Contd.

Work directory is chef-repo

- Template in company_web - "chef generate template cookbooks/company_web homepage", with content

```
<html>
<body>
  <h1><%= node['company_web']['company_name'] %> Welcomes You!</h1>
  <h2>PLATFORM: <%= node['platform'] %></h2>
  <h2>HOSTNAME: <%= node['hostname'] %></h2>
  <h2>MEMORY:    <%= node['memory']['total'] %></h2>
  <h2>CPU Mhz:   <%= node['cpu']['0']['mhz'] %></h2>
</body>
</html>
```

```
name 'company_web'
default_source :supermarket
run_list 'company_web::default'

cookbook 'company_web', path: 'cookbooks/company_web'
cookbook 'myiis', path: 'cookbooks/myiis'
cookbook 'apache', path: 'cookbooks/apache'
```

- Generate company_web policyfile - "chef generate policyfile company_web"
- Install and push to server
"chef install company_web.rb"
"chef push prod company_web.lock.json"
- Check all policies - "chef show-policy"

If you have changed apache cookbook after uploading to server

- Upgrade version in apache metadata.rb
- chef update company_web.rb. → check company_web.lock.json will now have updated apache version
- chef push prod company_web.lock.json

Lab 12.2 Contd.

Work directory is chef-repo

- Check details of node1 - `"knife node show node1"`
- Set policy of node1 to company_web - `"knife node policy set node1 prod company_web"`
- Check details of node1, observe the difference
- Runlist: `knife ssh <IP_node1> -m -x centos -i <aws.pem path> "sudo chef-client"`
- Test in browser or curl localhost



Review

Attributes are like parameters to your cookbook, not hard-coded values in recipes or templates. Can you think of some other parameters that you might want to create attributes for?

Can you imagine in complex topologies where you could have multiple levels of dependencies between cookbooks?

EXERCISE



GL: Reconfigure Welcome Message

So we want both our web server cookbooks to display our company name...

- Objective:**
- ✓ Create a 'company_web' wrapper cookbook that can apply either the 'apache' or 'apache' default recipe based on platform
 - ✓ Create a node attribute that contains your company name
 - ✓ Implement the edit_resource method to update the template resource for both the 'apache' and 'apache' server recipes
 - ✓ Create Policyfile and lock.
 - ✓ Upload Policyfile.lock to the Chef server
 - ✓ Converge the node



Q&A

What questions can we answer for you?



CHEF™