

Chef Resources and Recipes

```
$ knife cookbook create apache
$ ls -la cookbooks/apache
```

cookbooks/apache/recipes/default.rb

```
package "httpd" do
  action :install
end
```

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

```
cookbook_file "/var/www/html/index.html" do
  source "index.html"
  mode "0644"
end
```

cookbooks/apache/files/default/index.html

```
<html>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
```

```
$ knife cookbook upload apache
$ knife node run_list add node1 "recipe[apache]"
chef@node1:~$ sudo chef-client
```

Introducing the Node Object

```
$ knife node list
$ knife client list
$ knife node show node1
chef@node1:~$ sudo ohai | less
$ knife node show node1 -l
$ knife node show node1 -Fj
$ knife node show node1 -a fqdn
$ knife search node "*:*" -a fqdn
```

Node Attributes

cookbooks/apache/attributes/default.rb

```
default["apache"]["indexfile"] = "index1.html"
```

cookbooks/apache/files/default/index1.html

```
<html>
<body>
  <h1>Hello, world!</h1>
  <h2>This is index1.html</h2>
  <p>We configured this in the attributes file</p>
</body>
</html>
```

cookbooks/apache/recipes/default.rb

```
cookbook_file "/var/www/html/index.html" do
  source node["apache"]["indexfile"]
  mode "0644"
end
```

```
$ knife cookbook upload apache
chef@node1:~$ sudo chef-client
```

cookbooks/apache/recipes/default.rb

```
node.default["apache"]["indexfile"] = "index2.html"
cookbook_file "/var/www/html/index.html" do
  source node["apache"]["indexfile"]
  mode "0644"
end
```

cookbooks/apache/files/default/index2.html

```
<html>
<body>
  <h1>Hello, world!</h1>
  <h2>This is index2.html</h2>
  <p>We configured this in the recipe</p>
</body>
</html>
```

```
$ knife cookbook upload apache
```

```
chef@node1:~$ sudo chef-client
```

Attributes, Templates, and Cookbook Dependencies

Use knife to create a cookbook called 'motd' (command hidden)

cookbooks/motd/attributes/default.rb

```
default["motd"]["company"] = "Chef"
```

cookbooks/motd/recipes/default.rb

```
template '/etc/motd' do
  source 'motd.erb'
  owner 'root'
  group 'root'
  mode '0755'
end
```

cookbooks/motd/templates/default/motd.erb

```
This server is property of <%= node["motd"]["company"] %>
<% if node["pci"]["in_scope"] -%>
This server is in-scope for PCI compliance
<% end -%>
```

Use knife upload the 'motd' cookbook (command hidden)

Use knife to create a cookbook called 'pci' (command hidden)

cookbooks/pci/attributes/default.rb

```
default["pci"]["in_scope"] = true
```

Use knife upload the 'pci' cookbook (command hidden)

Use knife add 'recipe[motd]' to node1's run list (command hidden)

```
$ knife node show node1
chef@node1:~$ sudo chef-client
```

cookbooks/motd/metadata.rb

```
maintainer          "YOUR_COMPANY_NAME"
maintainer_email    "YOUR_EMAIL"
license             "All rights reserved"
description         "Installs/Configures motd"
long_description    IO.read(File.join(File.dirname(__FILE__),
`README.md`))
version            "0.1.0"
depends             "pci"
```

Use knife upload the 'motd' cookbook (command hidden)

Rerun 'chef-client' on node1 (command hidden)

```
chef@node1:~$ cat /etc/motd
$ knife search node "pci:*" -a pci
```

cookbooks/pci/attributes/default.rb

```
default["pci"]["in_scope"] = false
```

Use knife upload the 'pci' cookbook (command hidden)

Rerun 'chef-client' on node1 (command hidden)

```
chef@node1:~$ cat /etc/motd
$ knife node show node1 -a pci
```

Template Variables, Notifications, and Controlling Idempotency

cookbooks/apache/metadata.rb

```
maintainer          "YOUR_COMPANY_NAME"
maintainer_email    "YOUR_EMAIL"
license             "All rights reserved"
description         "Installs/Configures apache"
long_description    IO.read(File.join(File.dirname(__FILE__),
'README.md'))
version           "0.2.0"
```

cookbooks/apache/attributes/default.rb

```
default["apache"]["sites"]["clowns"] = { "port" => 80 }
default["apache"]["sites"]["bears"] = { "port" => 81 }
```

cookbooks/apache/recipes/default.rb

(See <https://gist.github.com/6781185>)

```

package "httpd" do
  action :install
end

service "httpd" do
  action [ :enable, :start ]
end

execute "mv /etc/httpd/conf.d/welcome.conf
/etc/httpd/conf.d/welcome.conf.disabled" do
  only_if do
    File.exist?("/etc/httpd/conf.d/welcome.conf")
  end
  notifies :restart, "service[httpd]"
end

node["apache"]["sites"].each do |site_name, site_data|
  document_root = "/srv/apache/#{site_name}"

  template "/etc/httpd/conf.d/#{site_name}.conf" do
    source "custom.erb"
    mode "0644"
    variables(:document_root => document_root, :port =>
site_data["port"])
    notifies :restart, "service[httpd]"
  end

  directory document_root do
    mode "0755"
    recursive true
  end

  template "#{document_root}/index.html" do
    source "index.html.erb"
    mode "0644"
    variables(:site_name => site_name, :port => site_data["port"])
  end
end
end

```

cookbooks/apache/templates/default/custom.erb

(See <https://gist.github.com/8955103>)


```

<% if @port != 80 -%>
  Listen <%= @port %>
<% end -%>

<VirtualHost *: <%= @port %>>
  ServerAdmin webmaster@localhost

  DocumentRoot <%= @document_root %>
  <Directory />
    Options FollowSymLinks
    AllowOverride None
  </Directory>
  <Directory <%= @document_root %>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
</VirtualHost>

```

cookbooks/apache/templates/default/index.html.erb

(See <https://gist.github.com/2866421>)

```

<html>
  <body>
    <h1>Welcome to <%= node["motd"]["company"] %></h1>
    <h2>We love <%= @site_name %></h2>
    <%= node["ipaddress"] %>:<%= @port %>
  </body>
</html>

```

Use knife upload the 'apache' cookbook (command hidden)

Rerun 'chef-client' on node1 (command hidden)

Troubleshoot the failure

Search

```
$ knife search node "*:*"
$ knife search node "ipaddress:10.*"
$ knife search node "*:*" -a ipaddress
$ knife search node "ipaddress:10.*" -a ipaddress
$ knife search node "ipaddress:10* AND
platform_family:rhel"
```

cookbooks/apache/recipes/ip-logger.rb

```
search("node", "platform:centos").each do |server|
  log "The CentOS servers in your organization have the following
  FQDN/IP Addresses:- #{server["fqdn"]}/#{server["ipaddress"]}"
end
```