

Chef Resources and Recipes

Writing an Apache cookbook

v2.1.6

Lesson Objectives

- After completing the lesson, you will be able to
 - Describe in detail what a **cookbook** is
 - Create a new cookbook
 - Explain what a **recipe** is
 - Describe how to use the **package**, **service**, and **cookbook_file** resources
 - **Upload a cookbook** to the Chef Server
 - Explain what a **run list** is, and how to set it for a node via knife
 - Explain the output of a chef-client run

What is a cookbook?

- A cookbook is like a “package” for Chef recipes.
 - It contains all the recipes, files, templates, libraries, etc. required to configure a portion of your infrastructure
- Typically they map 1:1 to a piece of software or functionality.

The Problem and the Success Criteria

- **The Problem:** We need a web server configured to serve up our home page.
- **Success Criteria:** We can see the homepage in a web browser.

Required steps

- Install Apache
 - Start the service, and make sure it will start when the machine boots
 - Write out the home page
-
- Please note in this course we're teaching Chef primitives, not web server management
 - This is probably not the Apache HTTP server configuration you would use in production

Exercise: Create a new Cookbook

```
$ knife cookbook create apache
```

```
** Creating cookbook apache
** Creating README for cookbook: apache
** Creating CHANGELOG for cookbook: apache
** Creating metadata for cookbook: apache
```

Exercise: Explore the cookbook

```
$ ls -la cookbooks/apache
```

```
total 24
drwxr-xr-x 13 opscode opscode 442 Jan 24 21:25 .
drwxr-xr-x  5 opscode opscode 170 Jan 24 21:25 ..
-rw-r--r--  1 opscode opscode 412 Jan 24 21:25 CHANGELOG.md
-rw-r--r--  1 opscode opscode 1447 Jan 24 21:25 README.md
drwxr-xr-x  2 opscode opscode  68 Jan 24 21:25 attributes
drwxr-xr-x  2 opscode opscode  68 Jan 24 21:25 definitions
drwxr-xr-x  3 opscode opscode 102 Jan 24 21:25 files
drwxr-xr-x  2 opscode opscode  68 Jan 24 21:25 libraries
-rw-r--r--  1 opscode opscode 276 Jan 24 21:25 metadata.rb
drwxr-xr-x  2 opscode opscode  68 Jan 24 21:25 providers
drwxr-xr-x  3 opscode opscode 102 Jan 24 21:25 recipes
drwxr-xr-x  2 opscode opscode  68 Jan 24 21:25 resources
drwxr-xr-x  3 opscode opscode 102 Jan 24 21:25 templates
```

Exercise: Edit the default recipe



OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
#  
# Cookbook Name:: apache  
# Recipe:: default  
#  
# Copyright 2013, YOUR_COMPANY_NAME  
#  
# All rights reserved - Do Not Redistribute  
#
```

Chef Resources

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Chef Resources

- Have a type

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Chef Resources

- Have a type
- Have a name

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Chef Resources

- Have a type
- Have a name
- Have **parameters**

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Chef Resources

- Have a type
- Have a name
- Have parameters
- Take **action** to put the resource into the desired state

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Chef Resources

- Have a type
- Have a name
- Have parameters
- Take action to put the resource into the desired state
- Can send **notifications** to other resources

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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Exercise: Add a package resource to install Apache to the default recipe



OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
#  
# Cookbook Name:: apache  
# Recipe:: default  
#  
# Copyright 2013, YOUR_COMPANY_NAME  
#  
# All rights reserved - Do Not Redistribute  
  
#  
  
package "httpd" do  
  action :install  
end
```

SAVE FILE!

189

So the resource we just wrote...

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

So the resource we just wrote...

- Is a **package** resource

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

So the resource we just wrote...

- Is a package resource
- Whose **name** is *httpd*

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

So the resource we just wrote...

- Is a package resource
- Whose name is *httpd*
- With an install action

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

Notice we didn't say how to install the package

- Resources are **declarative** - that means we say *what* we want to have happen, rather than *how*
- Resources take action through **Providers** - providers perform the *how*
- Chef uses the **platform** the node is running to determine the correct **provider** for a resource

Package Resource

```
package "git"
```



```
yum install git
```

```
apt-get install git
```

```
pacman sync git
```

```
pkg_add -r git
```

***Providers are
determined
by node's platform***

Exercise: Add a service resource to ensure the service is started and enabled at boot



OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

```
...
# All rights reserved - Do Not Redistribute
#
package "httpd" do
  action :install
end

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

SAVE FILE!

196

So the resource we just wrote...

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

So the resource we just wrote...

- Is a **service** resource

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

So the resource we just wrote...

- Is a service resource
- Whose **name** is *httpd*

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

So the resource we just wrote...

- Is a service resource
- Whose **name** is *httpd*
- With two **actions**:
 - enable
 - start

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

Order Matters

- Resources are executed in order



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

template "/etc/haproxy/haproxy.cfg" do
  source "haproxy.cfg.erb"
  owner "root"
  group "root"
  mode "0644"
  notifies :restart, "service[haproxy]"
end

service "haproxy" do
  supports :restart => true
  action [:enable, :start]
end
```

Exercise: Add a cookbook_file resource to copy the home page in place



OPEN IN EDITOR: cookbooks/apache/recipes/default.rb

...

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

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

SAVE FILE!

202

So the resource we just wrote...

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

So the resource we just wrote...

- Is a **cookbook_file** resource

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

So the resource we just wrote...

- Is a `cookbook_file` resource
- Whose **name** is: `/var/www/html/index.html`

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

So the resource we just wrote...

- Is a `cookbook_file` resource
- Whose **name** is: `/var/www/html/index.html`
- With two **parameters**:
 - **source** of `index.html`
 - **mode** of “0644”

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

Full contents of the apache recipe

```
#  
# Cookbook Name:: apache  
# Recipe:: default  
#  
# Copyright 2013, YOUR_COMPANY_NAME  
#  
# All rights reserved - Do Not Redistribute  
  
#  
  
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
```

Exercise: Add index.html to your cookbook's files/default directory



OPEN IN EDITOR: cookbooks/apache/files/default/index.html

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

SAVE FILE!

208

Exercise: Upload the cookbook

```
$ knife cookbook upload apache
```

```
Uploading apache [ 0.1.0 ]  
Uploaded 1 cookbook.
```

The Run List

- The Run List is the ordered set of recipes and roles that the Chef Client will execute on a node
 - Recipes are specified by “**recipe[*name*]**”
 - Roles are specified by “**role[*name*]**”

Exercise: Add apache recipe to test node's run list

```
$ knife node run_list add node1 "recipe[apache]"
```

```
node1:  
  run_list: recipe[apache]
```

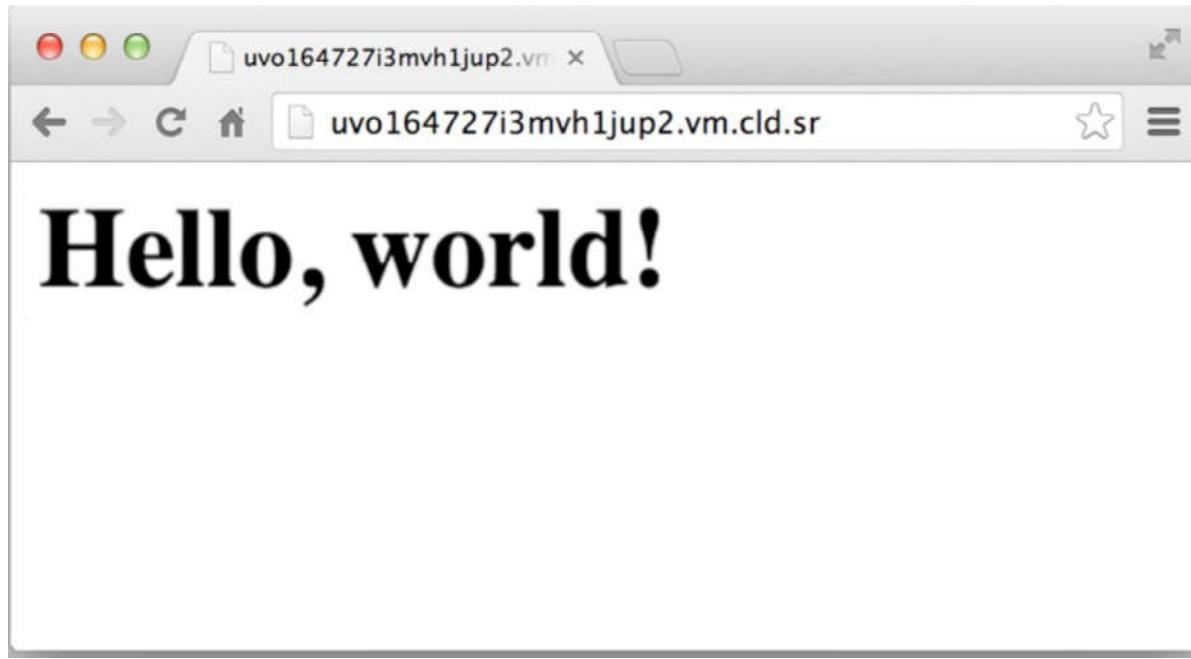
Exercise: Run Chef Client

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

```
[2014-01-21T12:22:40-05:00] INFO: Forking chef instance to converge...
Starting Chef Client, version 11.8.2
[2014-01-21T12:22:41-05:00] INFO: *** Chef 11.8.2 ***
[2014-01-21T12:22:41-05:00] INFO: Chef-client pid: 16346
[2014-01-21T12:22:41-05:00] INFO: Run List is [recipe[apache]]
[2014-01-21T12:22:41-05:00] INFO: Run List expands to [apache]
[2014-01-21T12:22:41-05:00] INFO: Starting Chef Run for node1
[2014-01-21T12:22:41-05:00] INFO: Running start handlers
[2014-01-21T12:22:41-05:00] INFO: Start handlers complete.
[2014-01-21T12:22:42-05:00] INFO: HTTP Request Returned 404 Object Not Found:
resolving cookbooks for run list: ["apache"]
[2014-01-21T12:22:42-05:00] INFO: Loading cookbooks [apache]
Synchronizing Cookbooks:
[2014-01-21T12:22:42-05:00] INFO: Storing updated
cookbooks/apache/recipes/default.rb in the cache.
[2014-01-21T12:22:42-05:00] INFO: Storing updated
cookbooks/apache/CHANGELOG.md in the cache.
[2014-01-21T12:22:43-05:00] INFO: Storing updated
cookbooks/apache/metadata.rb in the cache.
[2014-01-21T12:22:43-05:00] INFO: Storing updated
cookbooks/apache/README.md in the cache.
  - apache
Compiling Cookbooks...
Converging 3 resources
Recipe: apache::default
httpd-2.2.15-29.el6.centos from base repository
...
...
```

Exercise: Verify that the home page works

- Open a web browser
- Type in the URL for your test node



213

Congratulate yourself!

- You have just written your first Chef cookbook!
- (clap!)

Reading the output of a chef-client run

```
Starting Chef Client, version 11.8.2
[2014-01-06T07:06:00-05:00] INFO: *** Chef 11.8.2 ***
[2014-01-06T07:06:00-05:00] INFO: Chef-client pid: 10781
[2014-01-06T07:06:01-05:00] INFO: Run List is [recipe[apache]]
[2014-01-06T07:06:01-05:00] INFO: Run List expands to [apache]
[2014-01-06T07:06:01-05:00] INFO: Starting Chef Run for node1
[2014-01-06T07:06:01-05:00] INFO: Running start handlers
[2014-01-06T07:06:01-05:00] INFO: Start handlers complete.
```

- The run list is shown
- The expanded Run List is the complete list, after nested roles are expanded

Reading the output of a chef-client run

```
resolving cookbooks for run list: ["apache"]
[2014-01-06T07:06:02-05:00] INFO: Loading cookbooks [apache]
Synchronizing Cookbooks:
[2014-01-06T07:06:02-05:00] INFO: Storing updated cookbooks/apache/recipes/default.rb in the
cache.
[2014-01-06T07:06:02-05:00] INFO: Storing updated cookbooks/apache/metadata.rb in the cache.
  - apache
Compiling Cookbooks...
```

- Loads the cookbooks in the order specified by the run list
- Downloads any files that are missing from the server

Reading the output of a chef-client run

```
Converging 3 resources
Recipe: apache::default
 * package[httpd] action install[2014-01-06T07:51:48-05:00] INFO: Processing
   package[httpd] action install (apache::default line 9)
 [2014-01-06T07:51:55-05:00] INFO: package[httpd] installing
   httpd-2.2.15-29.el6.centos from base repository
     - install version 2.2.15-29.el6.centos of package httpd
```

- Checks to see if the package httpd is installed

Reading the output of a chef-client run

```
* service[httpd] action enable[2014-01-06T07:52:06-05:00] INFO: Processing
service[httpd] action enable (apache::default line 13)
[2014-01-06T07:52:06-05:00] INFO: service[httpd] enabled

- enable service service[httpd]

* service[httpd] action start[2014-01-06T07:52:06-05:00] INFO: Processing
service[httpd] action start (apache::default line 13)
[2014-01-06T07:52:07-05:00] INFO: service[httpd] started

- start service service[httpd]
```

- Checks to see if httpd is already enabled to run at boot - it is, take no further action
- Checks to see if httpd is already started - it is, take no further action

Idempotence

- Actions on resources in Chef are designed to be ***idempotent***
 - I.e. they can be applied multiple times but the end result is still the same - like multiplying by 1 in math!
 - Or in mathematical terms, $F(F(x))=F(x) \quad \forall x$
- Chef is a "desired state configuration" system - if a resource is already configured, no action is taken. This is called ***convergence***

Reading the output of a chef-client run

```
* cookbook_file[/var/www/html/index.html] action create[2014-01-06T07:52:07-05:00] INFO: Processing
cookbook_file[/var/www/html/index.html] action create (apache::default line 17)
[2014-01-06T07:52:07-05:00] INFO: cookbook_file[/var/www/html/index.html] created file /var/www/html/index.html

- create new file /var/www/html/index.html[2014-01-06T07:52:07-05:00] INFO: cookbook_file[/var/www/html/
index.html] updated file contents /var/www/html/index.html

- update content in file /var/www/html/index.html from none to 03fb1d
  --- /var/www/html/index.html      2014-01-06 07:52:07.285214202 -0500
  +++ /tmp/.index.html20140106-10796-1kxknbg    2014-01-06 07:52:07.868365963 -0500
  @@ -1 +1,6 @@
  +<html>
  +<body>
  +  <h1>Hello, world!</h1>
  +</body>
  +</html>[2014-01-06T07:52:07-05:00] INFO: cookbook_file[/var/www/html/index.html] mode changed to 644

- change mode from '' to '0644'
- restore selinux security context
```

- Checks for an index.html file
- There is already one in place, backup the file
- Set permissions on the file
- A diff of the written file is shown with the modified lines called out

Reading the output of a chef-client run

```
[2014-01-06T07:52:08-05:00] INFO: Chef Run complete in 23.477837576 seconds
[2014-01-06T07:52:08-05:00] INFO: Running report handlers
[2014-01-06T07:52:08-05:00] INFO: Report handlers complete
Chef Client finished, 4 resources updated
[2014-01-06T07:52:08-05:00] INFO: Sending resource update report (run-id:
952a8431-0994-468e-836c-0f7de7aa656e)
```

- Notice that a complete Chef-Run displays:
 - The time the client took to complete convergence
 - Status of report and exception handlers

Exercise: Re-run the Chef Client

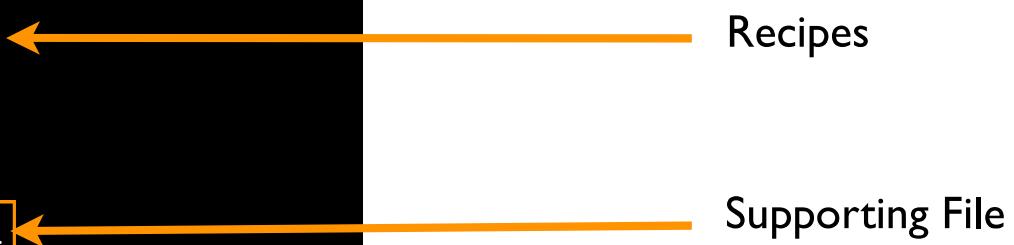
```
chef@node1:~$ sudo chef-client
```

```
...
resolving cookbooks for run List: ["apache"]
[2014-01-06T07:57:13-05:00] INFO: Loading cookbooks [apache]
Synchronizing Cookbooks:
  - apache
Compiling Cookbooks...
Converging 3 resources
Recipe: apache::default
  * package[httpd] action install[2014-01-06T07:57:13-05:00] INFO: Processing package[httpd] action install (apache::default line 9)
    (up to date)
  * service[httpd] action enable[2014-01-06T07:57:17-05:00] INFO: Processing service[httpd] action enable (apache::default line 13)
    (up to date)
  * service[httpd] action start[2014-01-06T07:57:17-05:00] INFO: Processing service[httpd] action start (apache::default line 13)
    (up to date)
  * cookbook_file[/var/www/html/index.html] action create[2014-01-06T07:57:17-05:00] INFO: Processing cookbook_file[/var/www/html/index.html] action create (apache::default line 17)
    (up to date)
[2014-01-06T07:57:17-05:00] INFO: Chef Run complete in 4.707139533 seconds
[2014-01-06T07:57:17-05:00] INFO: Removing cookbooks/apache/files/default/index.html from the cache; it is no longer needed by chef-client.
[2014-01-06T07:57:17-05:00] INFO: Running report handlers
[2014-01-06T07:57:17-05:00] INFO: Report handlers complete
Chef Client finished, 0 resources updated
[2014-01-06T07:57:17-05:00] INFO: Sending resource update report (run-id: 1c201d2c-797f-43ab-a904-1ba329b9aab8)
```

Recap - So resources are grouped into recipes

```
recipe[apache::default] {  
    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  
}
```

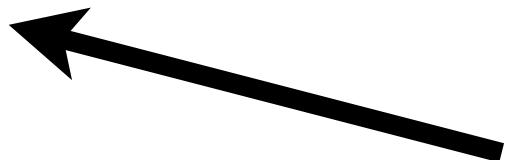
Cookbooks contain recipes and supporting files

```
& pwd  
/Users/you/chef-repo  
$ tree  
...  
└── cookbooks  
    ├── apache  
    │   └── recipes  
    │       ├── default.rb  
    │       └── server.rb  
    ├── files  
    │   └── default  
    │       └── index.html  
    ├── metadata.rb  
    ├── attributes  
    │   └── default  
...  

```

Recipes

Supporting File

Cookbooks are installed as artifacts on Chef Server

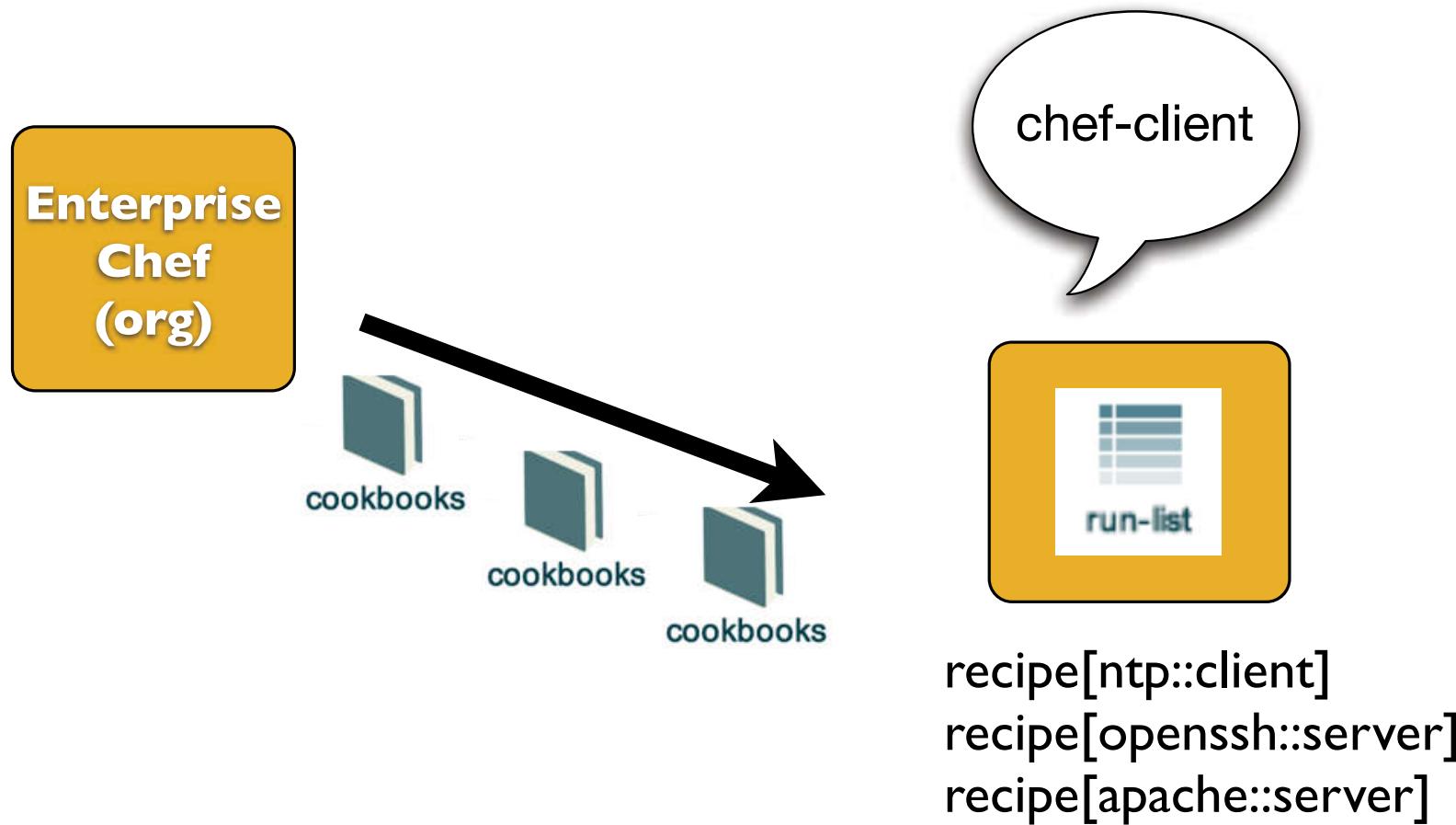


apache v0.1.0

ntp v0.2.0

openssh v2.4.7

Nodes have run_lists made of recipes



Questions

- What makes up a cookbook?
- How do you create a new cookbook?
- What is a recipe?
- What is a resource?
- How do you upload a cookbook to the Chef Server?
- What is a run list?