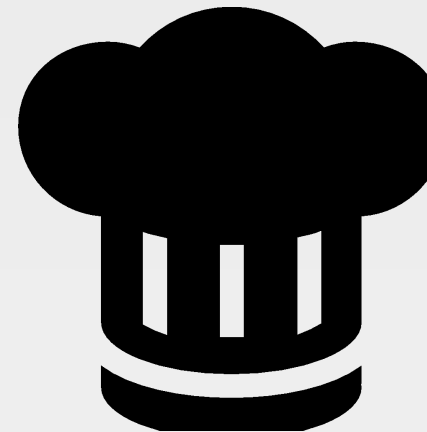


Lab

Building an Apache Webserver Cookbook

1L



Lab: Pre-built Linux Node

We will provide for you a Linux node with all the tools installed. You can do this lab on your own.

Objective:

- ☐ Login to the Linux Node

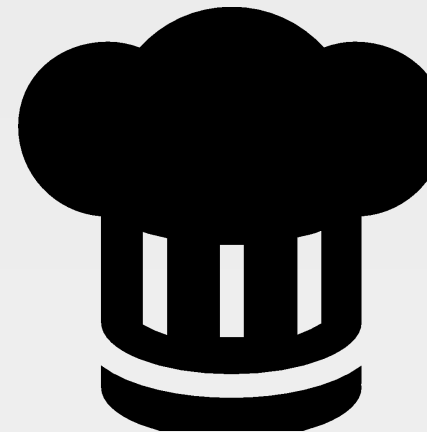
GL: Login to the Linux Node



```
> ssh IPADDRESS -l USERNAME
```

```
The authenticity of host '54.209.164.144 (54.209.164.144)' can't be
established.RSA key fingerprint is
SHA256:tKoTsPbn6ER9BLThZqntXTxIYem3zV/iTQWvhLrBIBQ.Are you sure you want to
continue connecting (yes/no)? yes
chef@54.209.164.144's password: PASSWORD
chef@ip-172-31-15-97 ~]$
```

You should ssh into the Linux node from your own laptop (much simpler) but if you're having trouble, the instructor might ask you to connect from the Windows workstation that you have already been using in the course.



Choose an Editor

You'll need to choose an editor to edit files:

In your 'participant guide', found below, you can find tips for using these editors:

nano

vi / vim

emacs



Lab: Setting up an Apache Web Server

- ☐ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ☐ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ☐ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ☐ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ☐ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ☐ Verify the site is available by running **`curl localhost`**

Lab: From Your Linux Node - Working within Home Directory



```
$ cd ~
```

Lab: Create a Cookbooks Directory



```
$ mkdir cookbooks
```

Lab: Accept the Chef License



```
$ sudo chef env
```

```
+-----+
      Chef License Acceptance

Before you can continue, 1 product license
must be accepted. View the license at
https://www.chef.io/end-user-license-agreement/

License that need accepting:
  * Chef Workstation

Do you accept the 1 product license (yes/no)?

> yes

Persisting 1 product license...
✓ 1 product license persisted.
```


Lab: Creating the apache Cookbook



```
> chef generate cookbook cookbooks/apache
```

Generating cookbook apache

- Ensuring correct cookbook content
- Committing cookbook files to git

Your cookbook is ready. Type ``cd cookbooks/apache`` to enter it.

There are several commands you can run to get started locally developing and testing your cookbook.

Type ``delivery local --help`` to see a full list of local testing commands.

Why not start by writing an InSpec test? Tests for the default recipe are stored at:

`test/integration/default/default_test.rb`

If you'd prefer to dive right in, the default recipe can be found at: `recipes/default.rb`



Lab: Setting up an Apache Web Server

- ✓ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ❑ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ❑ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ❑ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ❑ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ❑ Verify the site is available by running **curl localhost**

Lab: Creating the server Recipe



```
> chef generate recipe cookbooks/apache server
```

```
Recipe: code_generator::recipe
```

```
  * directory[cookbooks/apache/spec/unit/recipes] action create  
(up to date)
```

```
  * cookbook_file[cookbooks/apache/spec/spec_helper.rb] action  
create_if_missing (up to date)
```

```
  * template[cookbooks/apache/spec/unit/recipes/server_spec.rb]  
action create_if_missing
```

```
    - create new file
```

```
cookbooks/apache/spec/unit/recipes/server_spec.rb
```

```
    - update content in file
```

```
cookbooks/apache/spec/unit/recipes/server_spec.rb from none to  
e5ca2c
```

Lab: Defining the Policy in the server Recipe



```
~/cookbooks/apache/recipes/server.rb
```

```
#  
# Cookbook Name:: apache  
# Recipe:: server  
#  
# Copyright (c) 2017 The Authors, All Rights Reserved.  
package 'httpd'  
  
template '/var/www/html/index.html' do  
  source 'index.html.erb'  
end  
  
service 'httpd' do  
  action [:enable, :start]  
end
```

Note: You can use vi/vim, nano, or emacs to edit these files on Linux.

Nano is the easiest editor to use.



Lab: Setting up an Apache Web Server

- ✓ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ✓ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ❑ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ❑ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ❑ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ❑ Verify the site is available by running **`curl localhost`**

Lab: Creating the html Template



```
> chef generate template cookbooks/apache index.html
```

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

```
  * directory[cookbooks/apache/templates/default] action create
```

```
    - create new directory cookbooks/apache/templates/default
```

```
  * template[cookbooks/apache/templates/index.html.erb] action  
create
```

```
    - create new file cookbooks/apache/templates/index.html.erb
```

```
    - update content in file
```

```
cookbooks/apache/templates/index.html.erb from none to e3b0c4
```

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

Lab: Defining the index.html Template

`~/cookbooks/apache/templates/index.html.erb`

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



Lab: Setting up an Apache Web Server

- ✓ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ✓ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ✓ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ☐ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ☐ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ☐ Verify the site is available by running **`curl localhost`**

Lab: Including the server Recipe

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

```
#  
# Cookbook Name:: apache  
# Recipe:: default  
#  
# Copyright (c) 2016 The Authors, All Rights Reserved.  
  
include_recipe 'apache::server'
```



Lab: Setting up an Apache Web Server

- ✓ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ✓ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ✓ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ✓ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ☐ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ☐ Verify the site is available by running **`curl localhost`**

Lab: Applying the apache Cookbook's default Recipe



```
> sudo chef-client --local-mode --runlist "recipe[apache]"
```

```
Starting Chef Infra Client, version 17.3.48
resolving cookbooks for run list: ["apache"]
Synchronizing Cookbooks:
  - apache (0.1.0)
Installing Cookbook Gems:
Compiling Cookbooks...
Converging 3 resources
```

```
...
* service[httpd] action enable
  - enable service service[httpd]
* service[httpd] action start
  - start service service[httpd]
```

```
Running handlers:
```

```
Running handlers complete
```

```
Chef Infra Client finished, 4/4 resources updated in 13 seconds
```

Later in this course you'll learn how to set run lists using Policyfiles.

Lab: Setting up an Apache Web Server



- ✓ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ✓ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ✓ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ✓ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ✓ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ❑ Verify the site is available by running **`curl localhost`**

Lab: Verifying the Default Website is Available



```
> curl localhost
```

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>PLATFORM: centos</h2>
    <h2>HOSTNAME: ip-172-31-62-68</h2>
    <h2>MEMORY: 1013192kB</h2>
    <h2>CPU Mhz: 2400.234</h2>
  </body>
</html>
```

As you might imagine, you could also point a web browser to the external IP address of your Linux workstation/node to see similar output.

Lab: Setting up an Apache Web Server



- ✓ Create a **'cookbooks'** directory and generate a cookbook named **'apache'**
- ✓ Create a **'server'** recipe that defines the following policy:
 - The **package** named **'httpd'** is installed.
 - The **template** named **'/var/www/html/index.html'** is created with the source **'index.html.erb'**
 - The **service** named **'httpd'** is both started and enabled.
- ✓ Create a template named **'index.html.erb'** and populate it with a welcome message, the node's platform, hostname, total memory, and CPU speed.
- ✓ Use the `include_recipe` method to include the **'server'** recipe within the **'default'** recipe
- ✓ Use **'sudo chef-client'** to locally apply the apache cookbook's default recipe
- ✓ Verify the site is available by running **`curl localhost`**



Q&A

What questions can we help you answer?



CHEF™