Create the CookbookPermalink

1. From your workstation, move to your chef-repo:

```
2. cd chef-repo
```

3. Create the cookbook. In this instance the cookbook is titled `lamp-stack`:

```
4. knife cookbook create lamp-stack
```

5. Move to your cookbook's newly-created directory:

```
6. cd cookbooks/lamp-stack
```

7. If you list the files located in the newly-created cookbook, you will see that a number of directories and files have been created:

```
8. attributes    definitions  libraries    providers  recipes    templates
9. CHANGELOG.md  files        metadata.rb  README.md  resources
```

   For more information about these directories see the Beginner's Guide to Chef.

default.rbPermalink

The `default.rb` file in `recipes` contains the "default" recipe resources.
Because each section of the LAMP stack (Apache, MySQL, and PHP) will have its own recipe, the `default.rb` file is used to prepare your servers.

1. From within your `lamp-stack` directory, navigate to the `recipes` folder:

```
2. cd recipes
```

3. Open `default.rb` and add the Ruby command below, which will run system updates:

   ~/chef-repo/cookbooks/lamp-stack/recipe/default.rb

```
 1 #
2 3# Cookbook Name:: lamp-stack
4 5# Recipe:: default
6 7#
8 9#
10
   execute "update-upgrade" do
     command "apt-get update && apt-get upgrade -y"
     action :run
   end
```

   Recipes are comprised of a series of *resources*. In this case, the *execute* resource is used, which calls for a command to be executed once. The `apt-get update && apt-get upgrade -y`commands are defined in the `command` section, and the `action` is set to `:run` the commands.

This is one of the simpler Chef recipes to write, and a good way to start out. Any other start-up procedures that you deem important can be added to the file by mimicking the above code pattern.

4. To test the recipe, add the LAMP stack cookbook to the Chef server:

```
knife cookbook upload lamp-stack
```

6. Add the recipe to your chosen node's run *list*, replacing `nodename` with your node's name:

```
knife node run_list add nodename "recipe[lamp-stack]"
```

Because this is the default recipe, the recipe name does not need to be defined after `lamp-stack` in the code above.

8. Access your chosen node and run the *chef-client*:

```
chef-client
```

It should output a successful Chef run. If not, review your code for any errors, usually defined in the output of the chef-client run.

ApachePermalink

Install and EnablePermalink

1. Create a new file under `/recipes` called `apache.rb`. This will contain all of your Apache configuration information.
2. Open the file, and define the *package* resource to install Apache:

~/chef-repo/cookbooks/lamp-stack/apache.rb

```
package "apache2" do
    action :install
end
```

Again, this is a very basic recipe. The *package* resource calls to a package (apache2). This value must be a legitimate package name. The action is *install* because Apache is being installed in this step. There is no need for additional values to run the install.

3. Apache will also need to be set to turn on at reboot, and start. In the same file, add the additional lines of code:

~/chef-repo/cookbooks/lamp-stack/apache.rb

```
service "apache2" do
    action [:enable, :start]
end
```

This uses the *service* resource, which calls on the Apache service; the *enable* action enables it upon startup, whereas *start* will start Apache. Save and close the `apache.rb` file.

4. To test the Apache recipe, update the LAMP Stack recipe on the server:

```
5. knife cookbook upload lamp-stack
```

6. Add the recipe to a node's run-list, replacing `nodename` with your chosen node's name:

```
7. knife node run_list add nodename "recipe[lamp-stack::apache]"
```

   Because this is not the `default.rb` recipe the recipe name, *apache*, must be appended to the recipe value.
8. From that **node**, run the chef-client:

```
9. chef-client
```

   If the recipe fails due to a syntax error, Chef will note it during the output.
10. After a successful chef-client run, check to see if Apache is running:

```
11. service apache2 status
```

    It should say that apache2 is running.