What is a resource?

Answer- A resource represents a piece of infrastructure and its desired state, such as a package that should be installed, a service that should be running, or a file that should be generated.

Question: What is a recipe?

Answer- A recipe is a collection of resources that describes a particular configuration or policy. A recipe describes everything that is required to configure part of a system. Recipes do things such as:

install and configure software components.
manage files.
deploy applications.
execute other recipes.

Question: What happens when you don't specify a resource's action?

Answer- When you don't specify a resource's action, Chef applies the default action.

Question: Are these two recipes the same?

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

&&

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

Answer-

No, they are not. Remember that Chef applies resources in the order they appear. So the first recipe

ensures that the httpd package is installed and then configures the service. The second recipe configures the service and then ensures the package is installed.

Question: The second recipe may not work as you'd expect because the service resource will fail if the package is not yet installed.

Are these two recipes the same?

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

Answer-

No, they are not. Although both recipes ensure that the httpd package is installed before configuring its service, the first recipe enables the service when the system boots and then starts it. The second recipe starts the service and then enables it to start on reboot.

Are these two recipes the same?

file '/etc/motd' do
owner 'root'
group 'root'
mode '0755'
action :create
end

```
file '/etc/motd' do
action :create
mode '0755'
group 'root'
owner 'root'
end
```

Answer-

Yes, they are! Order matters with a lot of things in Chef, but you can order resource attributes any way you want.

Question -

Write a service resource that stops and then disables the httpd service from starting when the system boots.

Answer -

```
service 'httpd' do
action [:stop, :disable]
end
```

How does a cookbook differ from a recipe?

A recipe is a collection of resources, and typically configures a software package or some piece of infrastructure. A cookbook groups together recipes and other information in a way that is more manageable than having just recipes alone.

For example, in this lesson you used a template resource to manage your HTML home page from an external file. The recipe stated the configuration policy for your web site, and the template file contained the data. You used a cookbook to package both parts up into a single unit that you can later deploy.

How does chef-apply differ from chef-client?

chef-apply applies a single recipe; chef-client applies a cookbook.

For learning purposes, we had you start off with chef-apply because it helps you understand the basics quickly. In practice, chef-apply is useful when you want to quickly test something out. But for production purposes, you typically run chef-client to apply one or more cookbooks.

You'll learn in the next module how to run chef-client remotely from your workstation.

What's the run-list?

The run-list lets you specify which recipes to run, and the order in which to run them. The run-list is important for when you have multiple cookbooks, and the order in which they run matters.

What are the two ways to set up a Chef server?

Install an instance on your own infrastructure.
Use hosted Chef.

What's the role of the Starter Kit?
The Starter Kit provides certificates and other files that enable you to securely communicate with the Chef server.

Where can you get reusable cookbooks that are written and maintained by the Chef community?
Chef Supermarket, https://supermarket.chef.io.

What's the command that enables you to interact with the Chef server?
knife

What is a node?
A node represents a server and is typically a virtual machine, container instance, or physical server – basically any compute resource in your infrastructure that's managed by Chef.

What information do you need to in order to bootstrap?

You need:

your node's host name or public IP address.

a user name and password you can log on to your node with.

Alternatively, you can use key-based authentication instead of providing a user name and password.

What happens during the bootstrap process?

During the bootstrap process, the node downloads and installs chef-client, registers itself with the Chef server, and does an initial checkin. During this checkin, the node applies any cookbooks that are part of its run-list.

Which of the following lets you verify that your node has successfully bootstrapped?

The Chef management console.

knife node list

knife node show

You can use all three of these methods.

What is the command you use to upload a cookbook to the Chef server?

knife cookbook upload

How do you apply an updated cookbook to your node?

We mentioned two ways.

Run knife ssh from your workstation.

SSH directly into your server and run chef-client.

You can also run chef-client as a daemon, or service, to check in with the Chef server on a regular interval, say every 15 or 30 minutes.

Update your Apache cookbook to display your node's host name, platform, total installed memory, and number of CPUs in addition to its FQDN on the home page.

Update index.html.erb like this.

```
<html>
<body>
<h1>hello from <%= node['fqdn'] %></h1>

<pre>
<%= node['hostname'] %>
<%= node['platform'] %> - <%= node['platform_version'] %>
<%= node['memory']['total'] %> RAM
<%= node['cpu']['total'] %> CPUs
</pre>
</body>
</html>
```

Then upload your cookbook and run it on your node.

What would you set your cookbook's version to once it's ready to use in production?

According to Semantic Versioning, you should set your cookbook's version number to 1.0.0 at the point it's ready to use in production.

What is the latest version of the haproxy community cookbook?

To know the latest version of any cookbook on Chef Supermarket, browse to its page and view the latest version from the version selection box.

Or, get the info from the knife cookbook site command, like this.

```
knife cookbook site show haproxy | grep latest_version
latest_version: http://cookbooks.opscode.com/api/v1/cookbooks/haproxy/versions/1.6.6
```

Create a second node and apply the awesome_customers cookbook to it. How long does it take?

You already accomplished the majority of the tasks that you need. You wrote the awesome_customers cookbook, uploaded it and its dependent cookbooks to the Chef server, applied the awesome_customers cookbook to your node, and verified that everything's working.

All you need to do now is:

Bring up a second Red Hat Enterprise Linux or CentOS node.
Copy your secret key file to your second node.
Bootstrap your node the same way as before. Because you include the awesome_customers cookbook in your run-list, your node will apply that cookbook during the bootstrap process.
The result is a second node that's configured identically to the first one. The process should take far less time because you already did most of the work.

Now when you fix an issue or add a new feature, you'll be able to deploy and verify your update much more quickly!

What's the value of local development using Test Kitchen?

Local development with Test Kitchen:

enables you to use a variety of virtualization providers that create virtual machine or container instances locally on your workstation or in the cloud.
enables you to run your cookbooks on servers that resemble those that you use in production.
speeds up the development cycle by automatically provisioning and tearing down temporary instances, resolving cookbook dependencies, and applying your cookbooks to your instances.

What is VirtualBox? What is Vagrant?

VirtualBox is the software that manages your virtual machine instances.

Vagrant helps Test Kitchen communicate with VirtualBox and configures things like available memory and network settings.

Verify that your motd cookbook runs on both CentOS 6.6 and CentOS 6.5.

Your motd cookbook is already configured to work on CentOS 6.6 as well as CentOS 6.5, so you don't need to modify it.

To run it on CentOS 6.5, add an entry to the platforms section of your .kitchen.yml file like this.

```
---
      driver:
      name: vagrant
provisioner:
      name: chef_zero
platforms:
      - name: centos-6.6
      driver:
      box: opscode-centos-6.6
      box_url: http://opscode-vm-
bento.s3.amazonaws.com/vagrant/virtualbox/opscode_centos-6.6_chef-
provisionerless.box
      - name: centos-6.5
      driver:
      box: opscode-centos-6.5
      box_url: http://opscode-vm-
bento.s3.amazonaws.com/vagrant/virtualbox/opscode_centos-6.5_chef-
provisionerless.box
suites:
      - name: default
      run_list:
```

```
    - recipe[motd::default]
    attributes:
```

In many cases, Test Kitchen can infer the box and box_url parameters, which specify the name and location of the base image, or box. We specify them here to show you how to use them.

Run kitchen list to see the matrix of test instances that are available. Here, we have two platforms – CentOS 6.5 and CentOS 6.6 – multiplied by one suite – default.

$kitchen list

```
Instance            Driver   Provisioner  Verifier  Transport  Last Action
default-centos-66   Vagrant  ChefZero     Busser    Ssh        <Not
Created>
default-centos-65   Vagrant  ChefZero     Busser    Ssh        <Not
Created>
```

Run kitchen converge to create the instances and apply the motd cookbook.

```
$kitchen converge
   -----> Starting Kitchen (v1.4.0)
   -----> Creating <default-centos-66>...
   Bringing machine 'default' up with 'virtualbox' provider...
   [...]
   Running handlers:
   Running handlers complete
   Chef Client finished, 1/1 resources updated in 10.372334751 seconds
   Finished converging <default-centos-66> (3m52.59s).
   -----> Creating <default-centos-65>...
   Bringing machine 'default' up with 'virtualbox' provider...
   [...]
   Running handlers:
```

```
   Running handlers complete
   Chef Client finished, 1/1 resources updated in 5.32753132 seconds
   Finished converging <default-centos-65> (10m12.63s).
 -----> Kitchen is finished. (19m47.71s)
```

Now to confirm that everything's working, run kitchen login. But this time, you need to provide the instance name so that Test Kitchen knows which instance to connect to.

```
$kitchen login default-centos-66
      Last login: Wed May 13 20:15:00 2015 from 10.0.2.2

      hostname:  default-centos-66
      fqdn:      default-centos-66
      memory:    469392kB
      cpu count: 1
      [vagrant@default-centos-66 ~]$ logout
      Connection to 127.0.0.1 closed.$kitchen login default-centos-65
      Last login: Wed May 13 20:28:18 2015 from 10.0.2.2

      hostname:  default-centos-65
      fqdn:      default-centos-65
      memory:    469452kB
      cpu count: 1
      [vagrant@default-centos-65 ~]$ logout
      Connection to 127.0.0.1 closed.
```