Chef Q&A

**www.zippyops.com**

172-172, 5th floor Old Mahabalipuram Road

(Above Axis Bank-PTC Bus Stop)  
Thuraipakkam  
Chennai 600097

**🖂**[**zippyops@gmail.com**](mailto:zippyops@gmail.com)

**✆+91 7010585768**

**1.What is a resource?**

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.

2. **What is a recipe?**

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.

**3. What happens when you don’t specify a resource’s action?**

When you don’t specify a resource’s action, Chef applies the default action.

4. **Are these two recipes the same?**

package 'httpd'

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

&&

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

package 'httpd'

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.

5. **The second recipe may not work as you’d expect because the service resource will fail if the package is not yet installed.**

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.

6. 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

**Are these two recipes the same?**

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

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

service ‘httpd’ do  
action [:stop, :disable]  
end

8. **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.

9. **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.

10. **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.

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

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

12. **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.

13. **Where can you get reusable cookbooks that are written and maintained by the Chef community?**

**14. What’s the command that enables you to interact with the Chef server?**

Knife

15. **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.

16. **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.

17. **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.

18. **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.

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

knife cookbook upload

20. **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.

21. **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.

22. **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>

23. **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!

24. **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.

25. **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.

**26. What is**chef**in**devops**?**

Chef is a configuration management tool for dealing with machine setup on physical servers, virtual machines and in the cloud.Many companies use Chef software to control and manage their infrastructure including Facebook, Etsy, Cheezburger, and Indiegogo.

27. **What is**chef**in automation?**

Chef is a powerful automation platform that transforms infrastructure into code.s The Chef server acts as a hub for configuration data.  
The Chef server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by the chef-client.

**27. What is chef DK?**

The Chef DK workstation is the location where users interact with Chef. On the workstation users author and test cookbooks using tools such as Test Kitchen and interact with the Chef server using the knife and chef command line tools.

28. **What are**chef client**nodes?**

Chef client nodes are the machines that are managed by Chef. The Chef client is installed on each node and is used to configure the node to its desired state.

29. **What is a chef server?**

The Chef server acts as a hub for configuration data. The Chef server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by Chef. Nodes use the Chef client to ask the Chef server for configuration details, such as recipes, templates, and file distributions.

30. **What are work stations in chef?**

A workstation is a computer running the Chef Development Kit (ChefDK) that is used to author cookbooks, interact with the Chef server, and interact with nodes.

The workstation is the location from which most users do most of their work, including:

Developing and testing cookbooks and recipes  
Testing Chef code  
Keeping the chef-repo synchronized with version source control  
Configuring organizational policy, including defining roles and environments, and ensuring that critical data is stored in data bags  
Interacting with nodes, as (or when) required, such as performing a bootstrap operation

**31. What are Cookbooks in chef?**

A cookbook is the fundamental unit of configuration and policy distribution. A cookbook defines a scenario and contains everything that is required to support that scenario:

Recipes that specify the resources to use and the order in which they are to be applied  
Attribute values  
File distributions  
Templates  
Extensions to Chef, such as custom resources and libraries

**32.** **What is chef repo?**

The chef-repo is a directory on your workstation that stores:

Cookbooks (including recipes, attributes, custom resources, libraries, and templates)  
Roles  
Data bags  
Environments  
The chef-repo directory should be synchronized with a version control system, such as git. All of the data in the chef-repo should be treated like source code.

33. **What is chef-client Run?**

A “chef-client run” is the term used to describe a series of steps that are taken by the chef-client when it is configuring a node.

34. **What is chef validator?**

chef-validator – Every request made by the chef-client to the Chef server must be an authenticated request using the Chef server API and a private key. When the chef-client makes a request to the Chef server, the chef-client authenticates each request using a private key located in /etc/chef/client.pem.

35. **What is SSL\_CERT\_FILE in chef?**

Use the SSL\_CERT\_FILE environment variable to specify the location for the SSL certificate authority (CA) bundle that is used by the chef-client

**36. What are Knife Subcommands in chef?**

The chef-client includes two knife commands for managing SSL certificates:

Use knife ssl check to troubleshoot SSL certificate issues  
Use knife ssl fetch to pull down a certificate from the Chef server to the /.chef/trusted\_certs directory on the workstation.

**37. What is knife ssl check command in chef?**

Run the knife ssl check subcommand to verify the state of the SSL certificate, and then use the reponse to help troubleshoot issues that may be present.

**38. What is knife ssl fetch command in chef?**

Run the knife ssl fetch to download the self-signed certificate from the Chef server to the /.chef/trusted\_certs directory on a workstation.

**39. What are Data Bags?**

A data bag is a global variable that is stored as JSON data and is accessible from a Chef server. A data bag is indexed for searching and can be loaded by a recipe or accessed during a search.

40. **What is chef resources file?**

A file resource is used to manage files directly on a node.

A file resource block manages files that exist on nodes. For example, to write the home page for an Apache website:

file ‘/var/www/customers/public\_html/index.php’ do  
content ‘<html>This is a placeholder for the home page.</html>’  
mode ‘0755’  
owner ‘web\_admin’  
group ‘web\_admin’  
end

**41. What is apt\_package resource in chef?**

Use the apt\_package resource to manage packages on Debian and Ubuntu platforms.

apt\_package Syntax:

A apt\_package resource block manages a package on a node, typically by installing it. The simplest use of the apt\_package resource is:

apt\_package ‘package\_name’

**42. What is apt\_preference resource in chef?**

The apt\_preference resource allows for the creation of APT preference files. Preference files are used to control which package versions and sources are prioritized during installation. New in Chef Client 13.3

Syntax:

apt\_preference ‘package\_name’ do  
action :add  
end

**43. What is apt\_repository resource?**

 Use the apt\_repository resource to specify additional APT repositories. Adding a new repository will update APT package cache immediately.

apt\_repository ‘nginx’ do  
uri ‘http://nginx.org/packages/ubuntu/’  
components [‘nginx’]  
end

**44. What is apt\_update resource in chef?**

Use the apt\_update resource to manage APT repository updates on Debian and Ubuntu platforms.

**45. what is bff\_package resource in chef?**

Use the bff\_package resource to manage packages for the AIX platform using the installp utility. When a package is installed from a local file, it must be added to the node using the remote\_file or cookbook\_file resources.

**46. What is chef\_gem?**

Use the chef\_gem resource to install a gem only for the instance of Ruby that is dedicated to the chef-client. When a gem is installed from a local file, it must be added to the node using the remote\_file or cookbook\_file resources.

**47. What is chef\_acl resource in chef?**

Use the chef\_acl resource to interact with access control lists (ACLs) that exist on the Chef server.

Syntax: The syntax for using the chef\_acl resource in a recipe is as follows:

chef\_acl ‘name’ do  
attribute ‘value’ # see properties section below  
…  
action :action # see actions section below  
end

**48. What is chef\_client resource?**

A chef-client is an agent that runs locally on every node that is under management by Chef. When a chef-client is run, it will perform all of the steps that are required to bring the node into the expected state, including:

Registering and authenticating the node with the Chef server  
Building the node object  
Synchronizing cookbooks  
Compiling the resource collection by loading each of the required cookbooks, including recipes, attributes, and all other dependencies  
Taking the appropriate and required actions to configure the node  
Looking for exceptions and notifications, handling each as required.

**49.  What is chef\_container resource?**

 chef\_container resource is used to interact with container objects that exist on the Chef server.

**50. What is chef\_data\_bag\_item?**

A data bag is a container of related data bag items, where each individual data bag item is a JSON file. knife can load a data bag item by specifying the name of the data bag to which the item belongs and then the filename of the data bag item.

Use the chef\_data\_bag\_item resource to manage data bag items.

Syntax – The syntax for using the chef\_data\_bag\_item resource in a recipe is as follows:

chef\_data\_bag\_item ‘name’ do  
attribute ‘value’  
…  
action :action  
end