9: The Chef Server

You accomplished a lot so far. You created two cookbooks; one to setup workstations with your tools and a second cookbook that set up a web server that delivered a "Hello, world!" message with some pertinent information about your system.

Slide 2

Objectives

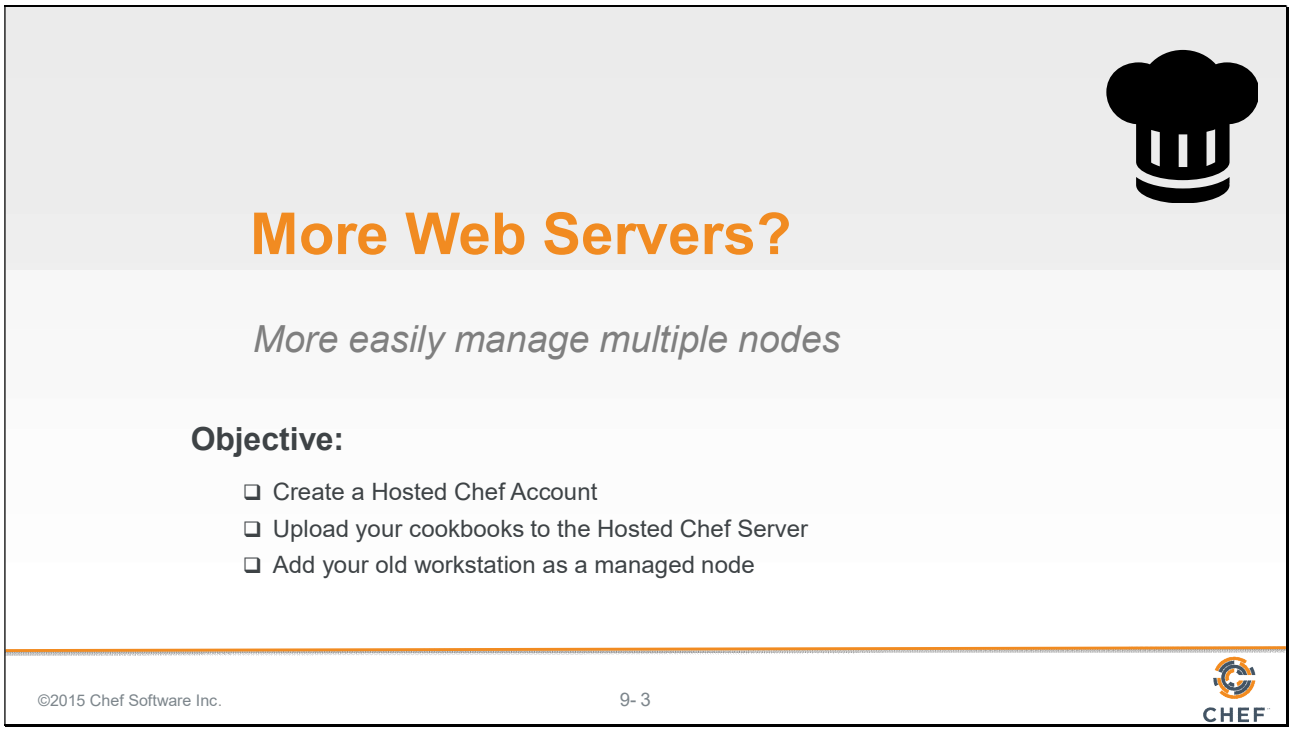
After completing this module, you should be able to

- Connect your local workstation (laptop) to a Chef Server
- Upload cookbooks to a Chef Server
- Bootstrap a node
- Manage a node via a Chef Server

CHEF

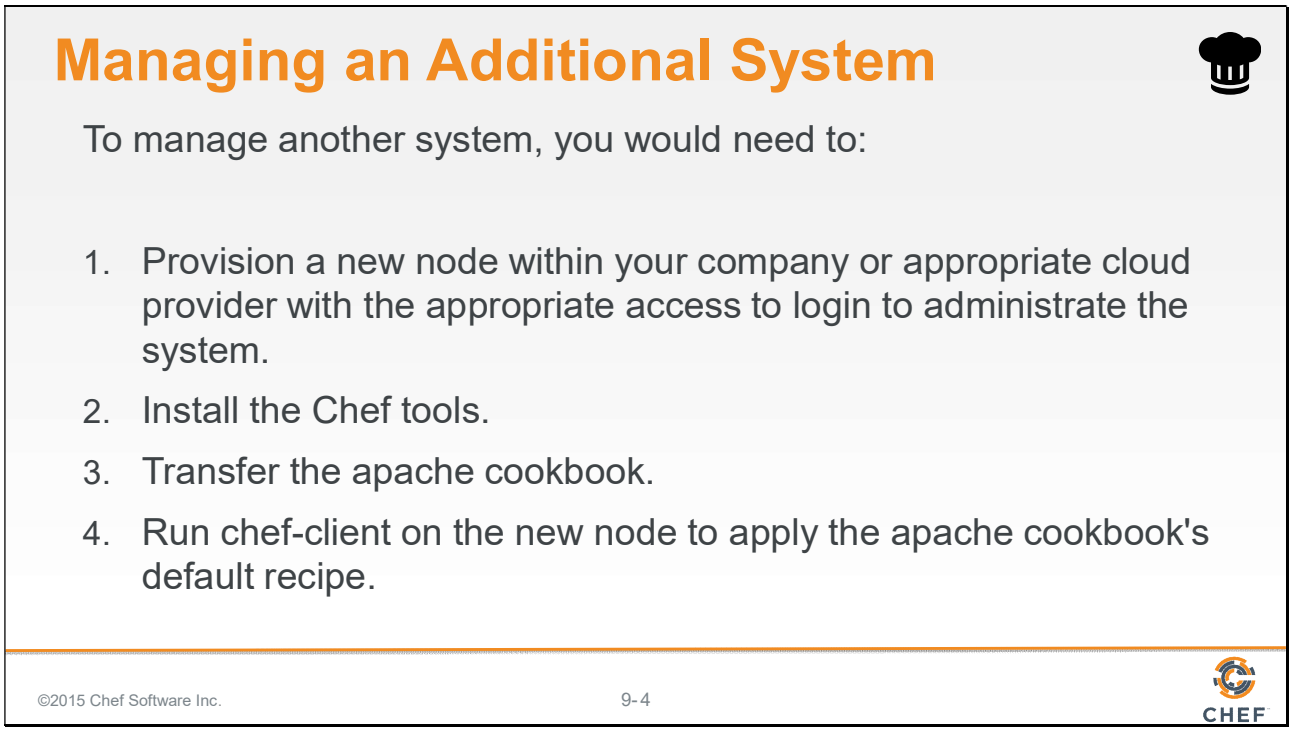
In this module you will learn how to connect your local workstation to a Chef Server, upload cookbooks to a Chef Server, bootstrap a node, manage a node via a Chef Server.

Slide 3



Currently, your cookbook exists on one webserver. If you wanted to setup additional web servers to serve additional traffic for your soon-to-be highly successful website, what steps would you need to take to setup an identical system?

Slide 4



As an exercise, roughly estimate the time it would take to accomplish this series of steps of preparing another node.

A new system would require us to provision a new node within your company or appropriate cloud provider with the appropriate access to login to administrate the system.

Install the Chef tools.

Transfer the apache cookbook.

Run chef-client locally to apply the apache cookbook's default recipe.

Instructor Note: This exercise is to show the value of using a Chef Server with regard to managing multiple systems. It can be done with the group, with individuals, or done in pairs.

Slide 5

Managing Additional Systems

Installing the Chef tools, transferring the apache cookbook, and applying the run list is not terribly expensive.

- Chef provides a one-line curl install.
- You could use **git** to clone the repository from a common **git** repository.
- Applying the run list.

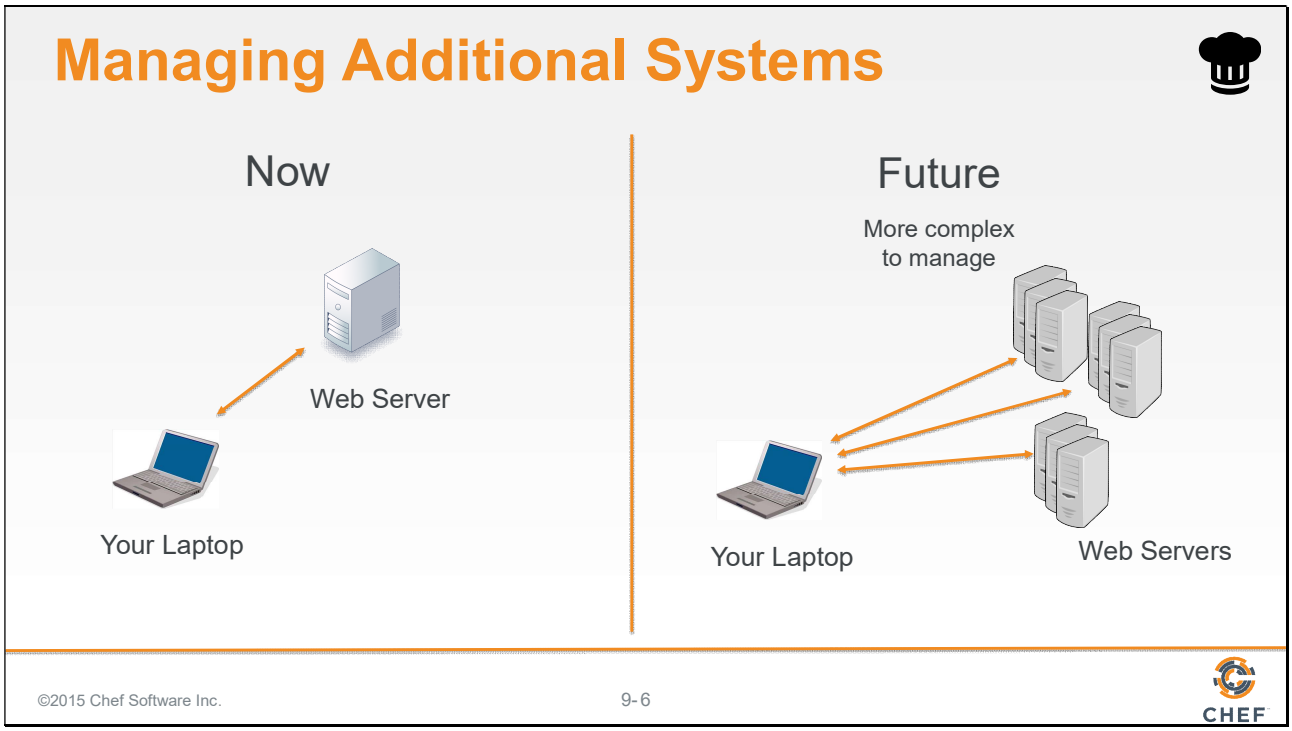
The cost of installing the Chef tools, transferring the apache cookbook, and applying the run list is not terribly expensive.

Chef provides a one-line curl install for the Chef Development Kit (ChefDK).

You could use git to clone the repository from a common git repository. Another option is to archive the cookbook and then using SCP to copy over the contents. A third might be to mount a file share. There are a myriad ways to transfer the cookbooks to the new instance.

Then applying the run list requires the execution of a command on that system.

Slide 6



So the overall time required to setup a new instance is not a massive time investment. This manual process will definitely take its toll when requirements demand you manage more than a few additional nodes.

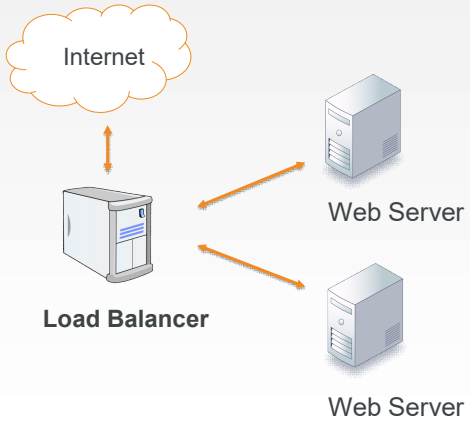
Some may think 10 minutes is not so bad. But what if there were 10 new nodes? 20 new nodes?

As the popularity of your site grows, one server will not be able to keep with all of the web requests. You will need to provision additional machines as demand increases.

Slide 7

Managing User Traffic

A load balancer can forward incoming user web requests to other nodes.



Let's change topics for a moment to managing user web traffic.

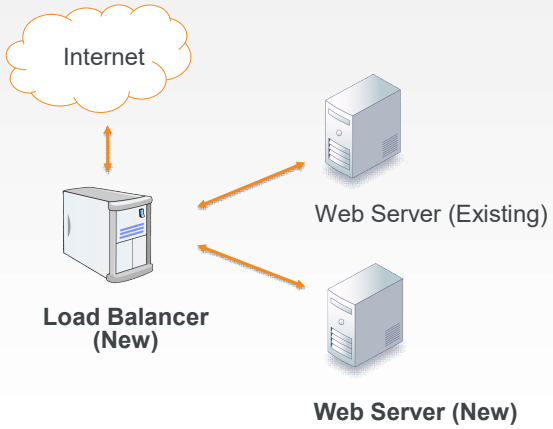
In addition to the complexities of configuring and managing multi-server infrastructure, such as web servers, you also need to develop a way to route incoming traffic to each of those web servers and other nodes. There are many ways that you can route the traffic from one node to a group of similar nodes. This can be done with services by some of the major cloud providers or it can be done with another instance running as a load balancer. A load balancer allows us to receive incoming requests and forward those requests to other nodes. A load balancer allows us to receive incoming requests and forward those requests to other nodes.

Instructor Note: The preceding three slides covered the complexity of configuring and managing multi-server infrastructure. This slide and page is now talking about managing user **web traffic**, via a load balancer.

Slide 8

Managing User Traffic

Today you will set up a new load balancer that will direct web requests to similarly-configured nodes.

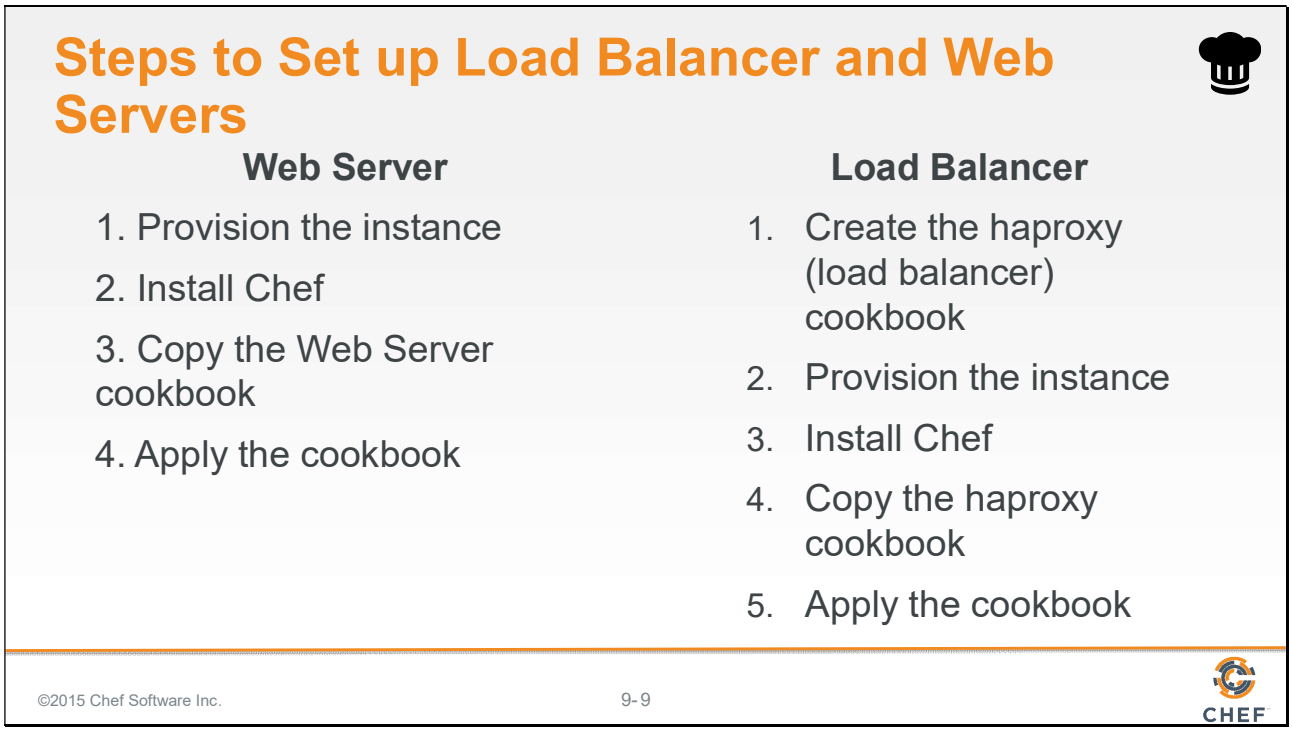


Today you are going to set up a load balancer that will direct web requests to similar configured nodes. Those nodes will be running your default web page that you deploy with the apache cookbook's default recipe.

You have one system already configured as a web server. You will need to set up another web server.

You will also need to set up a node to act as the load balancer to both of these web servers.

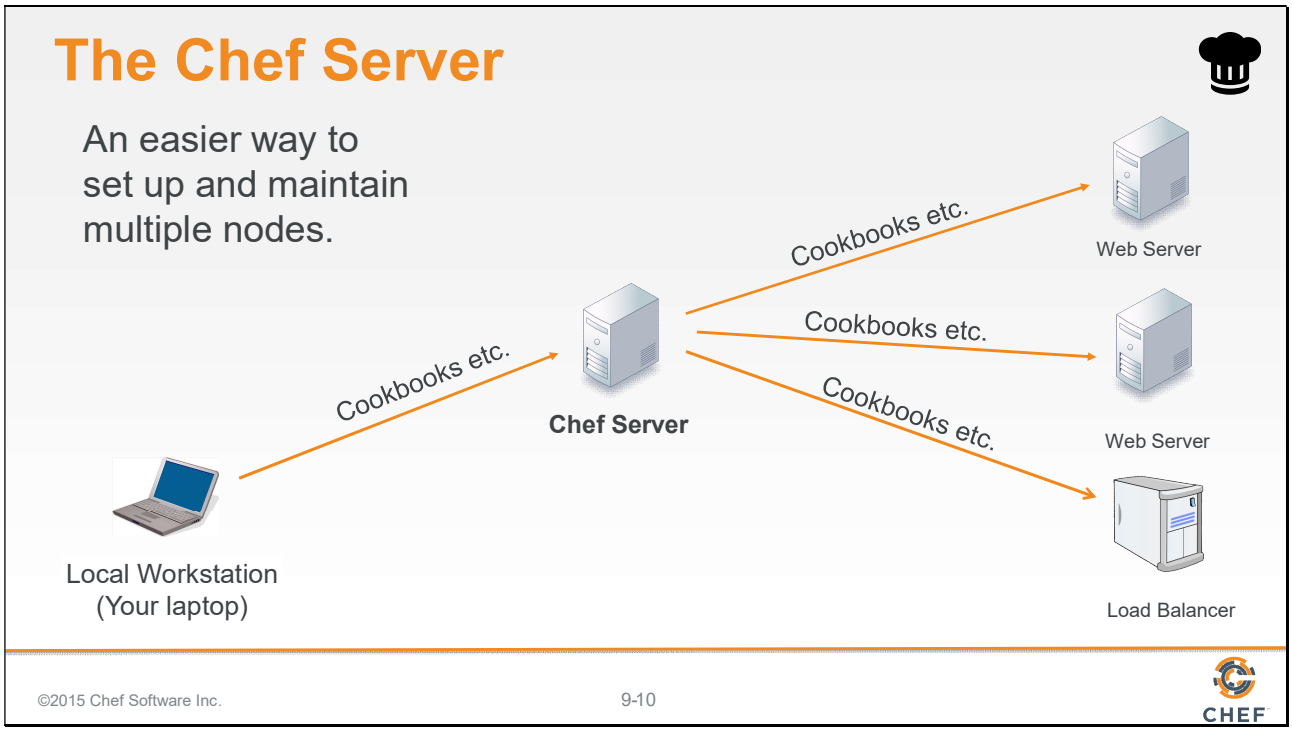
Slide 9



Whether you tackle installing, configuring, or running a load balancer or recreate a second instance running the apache cookbook's default recipe, you will need to solve the problem of how you can manage multiple systems. Each system would need to have Chef installed, the cookbooks copied onto each system, and a run list of the recipes to apply to each system.

Instructor Note: The left side is setting up a new web server that is like the one they created yesterday. The right is new work that the learner will be accomplishing today. Note that the haproxy is actually a load balancer.

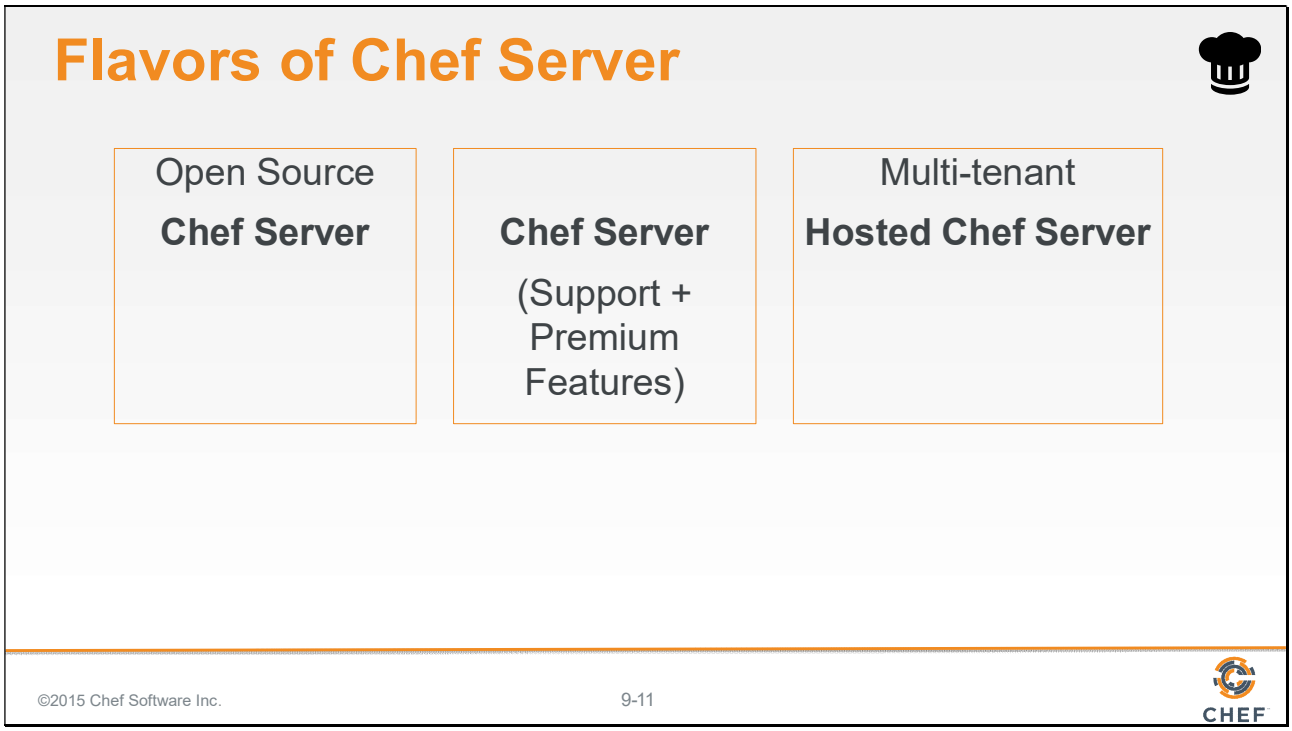
Slide 10



One way to solve that problem is with a Chef Server.

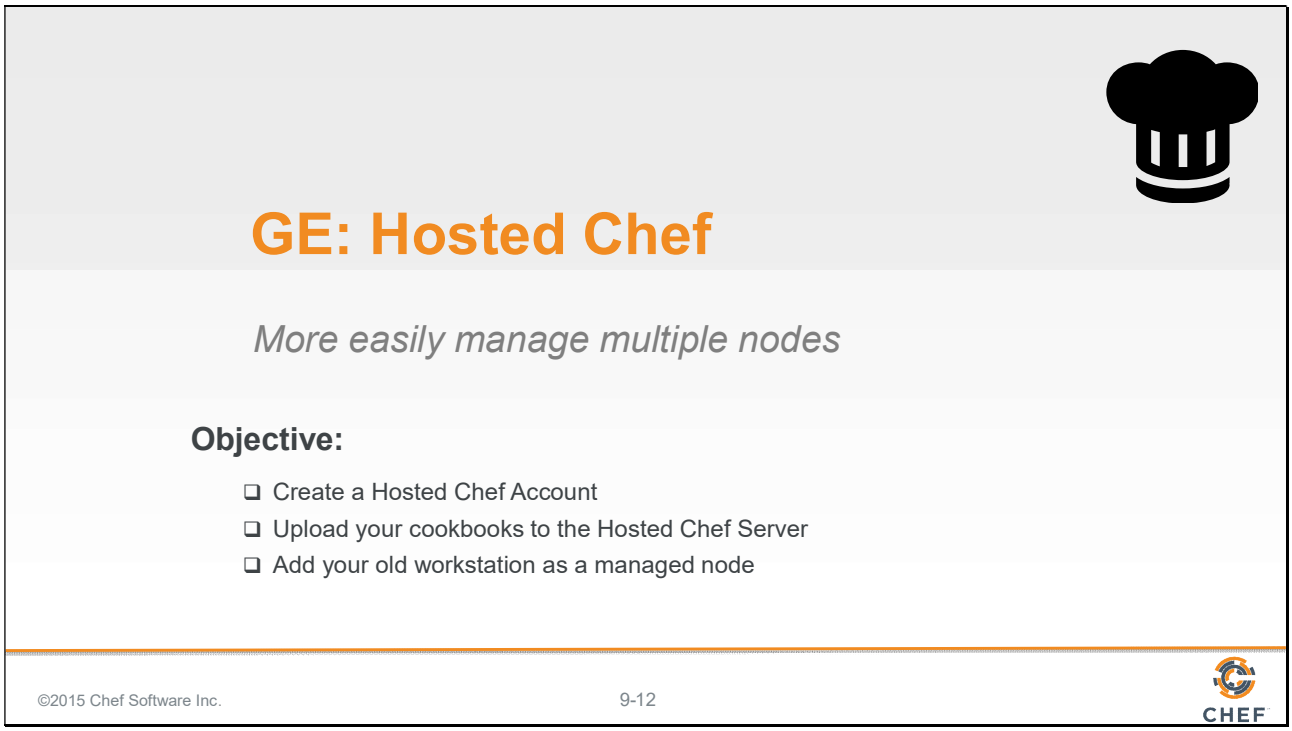
The Chef Server is designed to help us manage multiple nodes in this situation. The Chef Server acts as a hub for configuration data. The Chef server stores cookbooks, the policies that are applied to nodes, and metadata that describes each registered node that is being managed by 'chef-client'. Nodes, such as web servers, load balancers, etc., use 'chef-client' to ask the Chef server for configuration details, such as recipes, templates, and file distributions. The chef-client then does as much of the configuration work as possible on the nodes themselves (and not on the Chef server). In a production environment, the 'chef-client' runs in an automated mode—it polls the Chef Server for updates at set intervals and then applies any configuration changes. This scalable approach distributes the configuration effort throughout the organization.

Slide 11



At the core we offer Chef Server as an open source project freely available for anyone to deploy. We offer support and additional premium features. Lastly, we have Hosted Chef Server, which is a multi-tenant Chef Server that you host as a service. This by far is the quickest way to get started with and is free as long as you remain under the reasonable node amount.

Slide 12



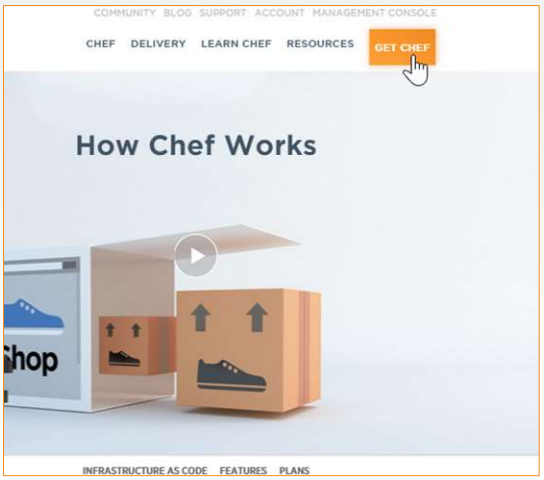
In the interest of getting things done with a relatively small node count, it seems like the Hosted Chef Server option is best.

Slide 13

GE: Signing Up for a Hosted Chef Account

Steps

1. Navigate to https://www.chef.io
2. From the resulting window, click **Get Chef**.



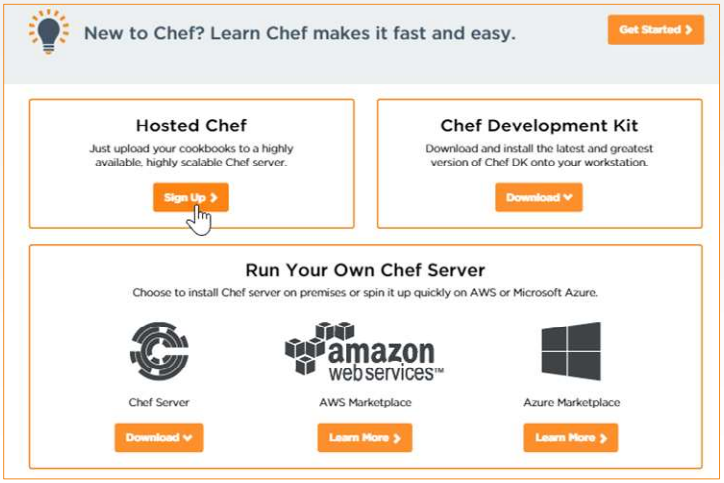
To get started with Hosted Chef Server, visit the Chef website and sign up for a Hosted Chef Account.

Slide 14

GE: Signing Up for a Hosted Chef Account

Steps

3. From the resulting window, click the Hosted Chef **Sign Up** button.



CHEF

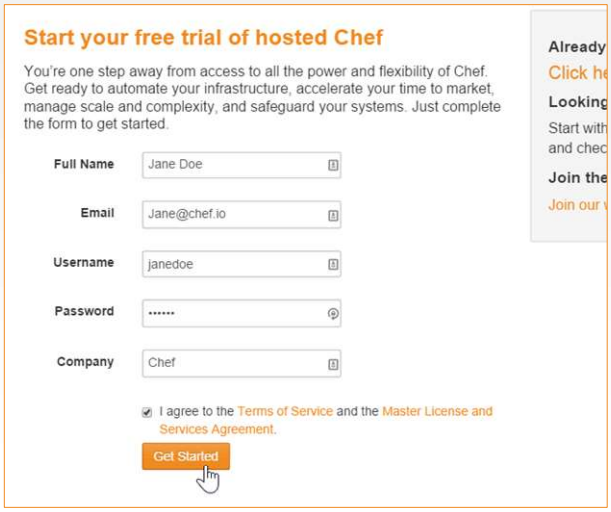
Slide 15

GE: Signing Up for a Hosted Chef Account

Steps

4. Fill out the form as indicated in this image using your name and a valid email address and then click **Get Started**.

Note: You should write down your new user name and remember your password.



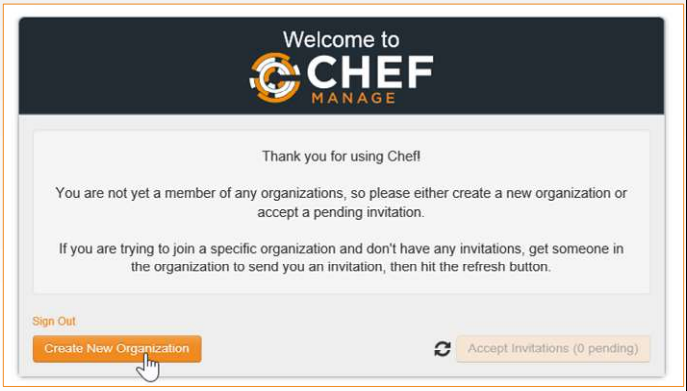
Instructor Note: Learners that already have an account will login instead of creating an account. The learner's account may be tied to an production organization. The learner can create a new organization with their existing account. If there are still concerns they can create a new login with a unique email address.

Slide 16

GE: Signing Up for a Hosted Chef Account

Steps

5. From the resulting page, click the **Create New Organization** button.



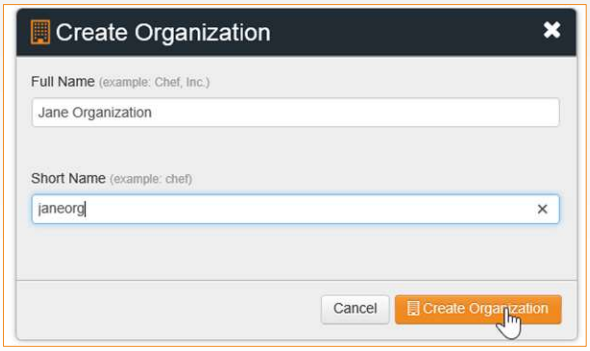
Instructor Note: Learners that already have an account will already have an organization. They are welcome to use that organization if it does not have any cookbooks or nodes from previous exercises or from their production systems. It is often easier to have them create a new organization for the purposes of this training. That can be done through the website.

Slide 17

GE: Signing Up for a Hosted Chef Account

Steps

6. Fill out the resulting Create Organization form and then click **Create Organization**.



CHEF

An organization is a structure within managed Chef that allows multiple companies or entities to exist on the same Chef Server without your paths ever crossing. You might think of it as like setting up a unique username for your organization.

All of the cookbooks, instances and other configuration details that you manage with Chef will be stored on the Chef Server for this particular organization. No other organization will have access to it.

Slide 18

GE: Signing Up for a Hosted Chef Account

Steps

7. From the resulting page, click your new organization to highlight it and then click **Starter Kit**.

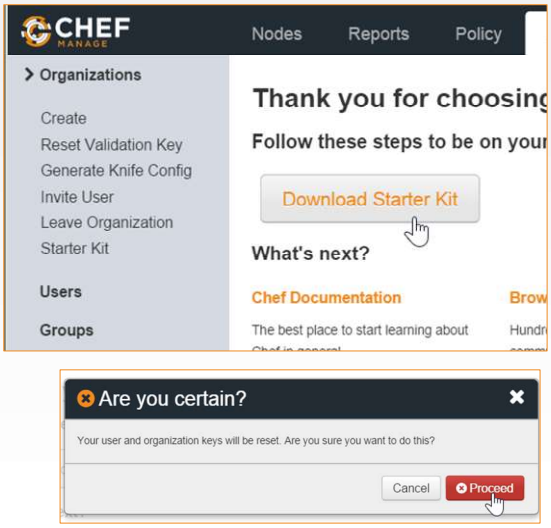
Instructor Note: By far the easiest way to get setup with Managed Chef Server is download the Starter Kit. The most important pieces of the starter kit are found in a hidden directory (".chef") which contains the organization key, user key, and organization configuration file.

Slide 19

GE: Signing Up for a Hosted Chef Account

Steps

8. From the resulting window, click the **Download Starter Kit** button.
9. Click the **Proceed** button when prompted.



The starter kit will warn that it will reset your organization key and personal key. If this is a new account and new organization this reset is totally fine. If you already have an account or this is an existing organization please understand that you are destroying the existing keys that already exist on a workstation.

Instructor Note: If the learner already has an account and an organization tied to that account this will reset their personal key and organizational key. This means that the other chef repository that they were previously maintaining will no longer be able to communicate with the Chef Server.

Slide 20

GE: Signing Up for a Hosted Chef Account

Steps

10. Open the downloaded zip file and copy chef-repo folder that's contained in the zip file.
11. Paste the chef-repo folder to a location on your laptop, such as your home directory.

Note: Ensure that the path to the chef-repo does not have a space in it. Examples:

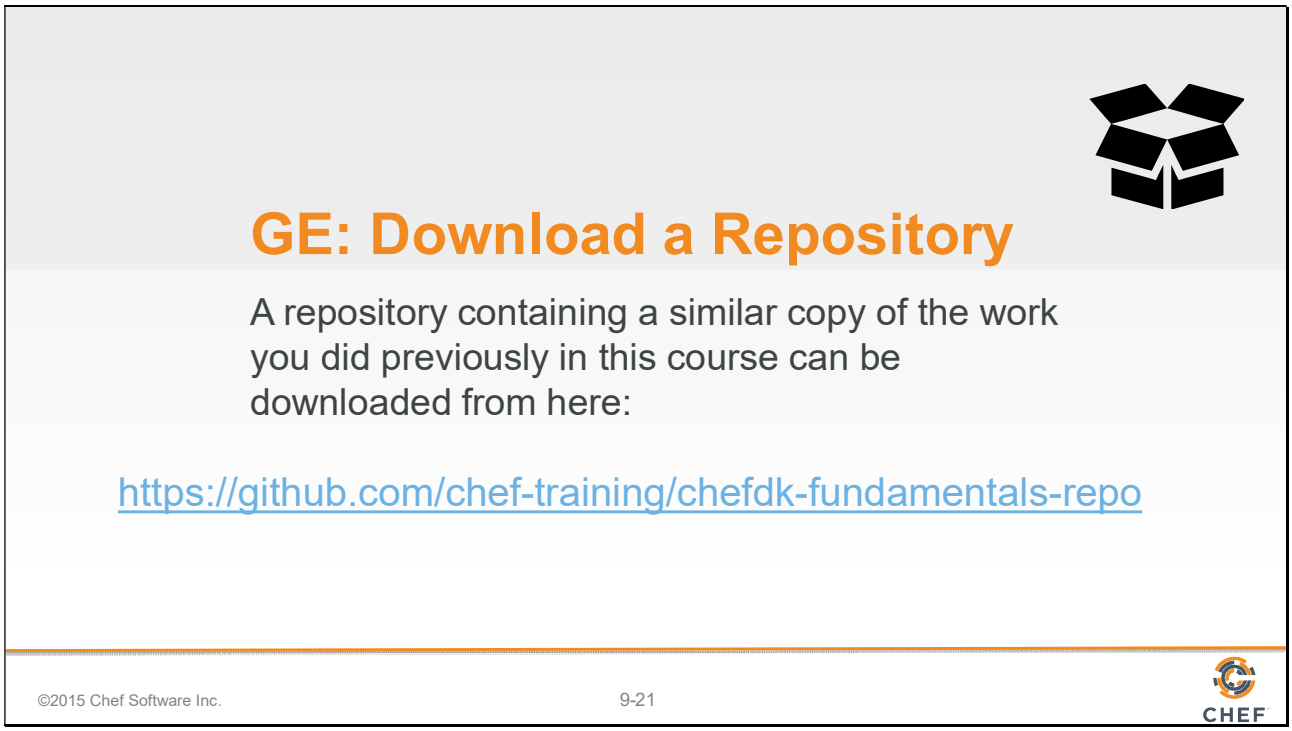
Mac: /home/*username*/chef-repo

Windows: C:\Users*username*\chef-repo

CHEF

Instructor Note: The reason that spaces are not suggested is that Ruby tools have a hard time with file paths that contain spaces.

Slide 21

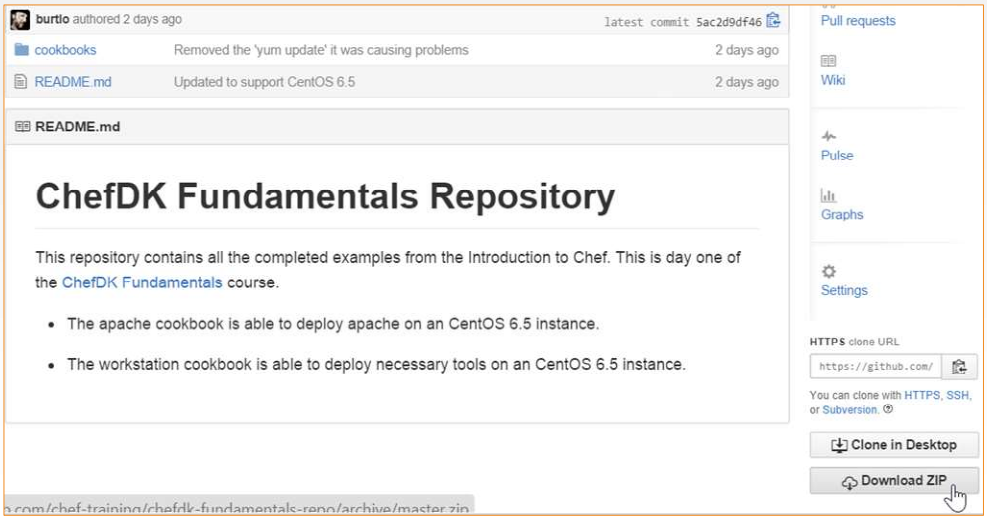


The cookbooks that were created during the first modules can be found here. These are not the exact cookbooks that you created but ones that have been completed with additional comments and details added.

Instructor Note: A learner may want to use the exact copy of the cookbooks that they developed. You may need to coordinate with the learners on using git or other methods to retrieve those cookbooks from those remote workstations.

Slide 22

GE: Download the Repository



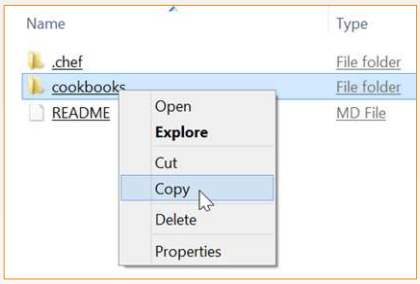
You may clone the repository or download the zip file. Both of those links can be found in the bottom right.

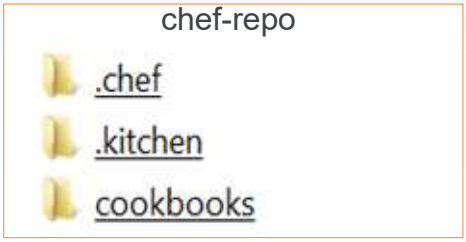
Slide 23

GE: Paste the cookbooks Folder

Steps

- Open the downloaded chefdk-fundamentals-repo-master zip file and then copy **only** the **cookbooks** folder that's contained in the zip file.
- Replace the **cookbooks** folder that's in your chef-repo folder with the copied cookbooks folder.





After you download and open the chefdk-fundamentals-repo archive, copy the included cookbooks folder and paste it into your chef-repo that you unzipped from the Start Kit. Let the new cookbooks folder (that you got from the chefdk-fundamentals-repo) overwrite the existing cookbooks folder that was in your chef-repo folder.

Important: If you had an existing chef-repo prior to class that you want to preserve, save a copy of your old cookbooks folder before pasting the new one into your chef-repo.

Instructor Note: The Starter Kit only contained a starter cookbook that has no value to the learner.

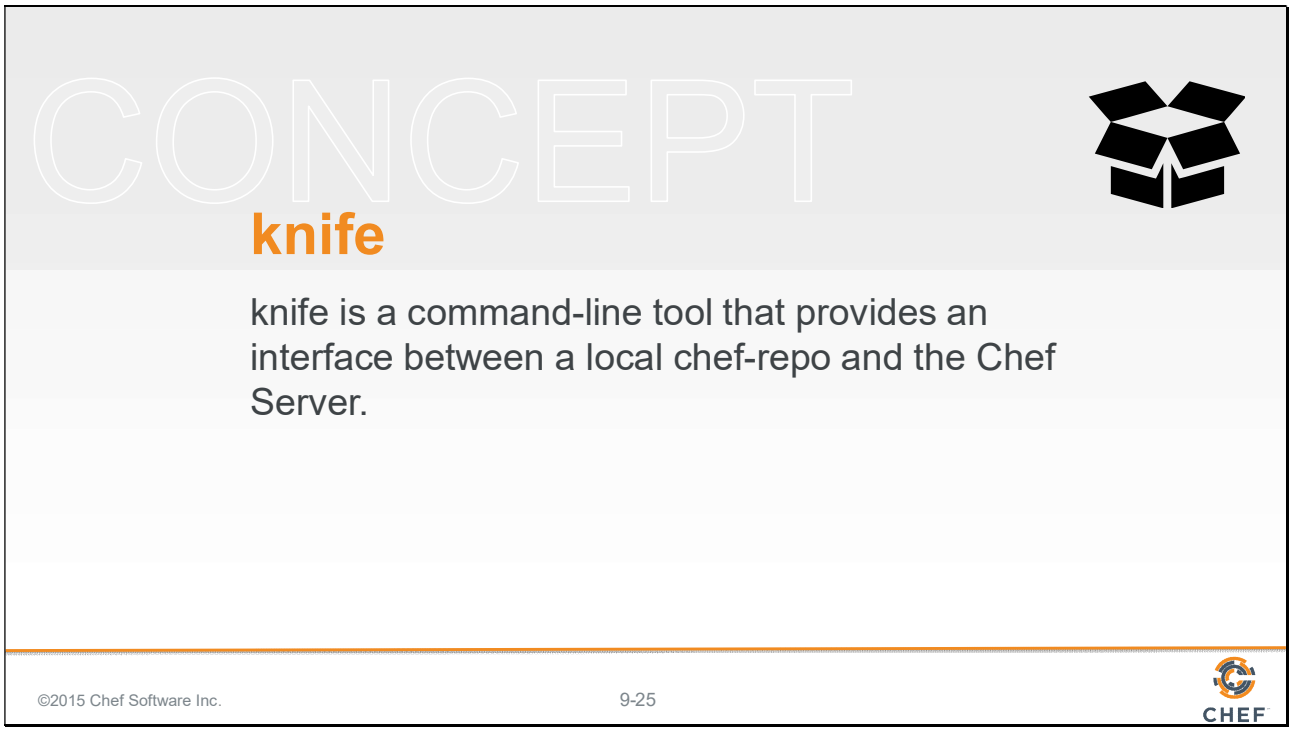
Slide 24

The starter kit contains the configuration to reach the Chef Server and your credentials to validate the communication between your workstation and the Chef Server.

To verify the connection with the Chef Server you will need to run commands within the repository you downloaded.

Open a terminal or command prompt and navigate to the chef-repo directory.

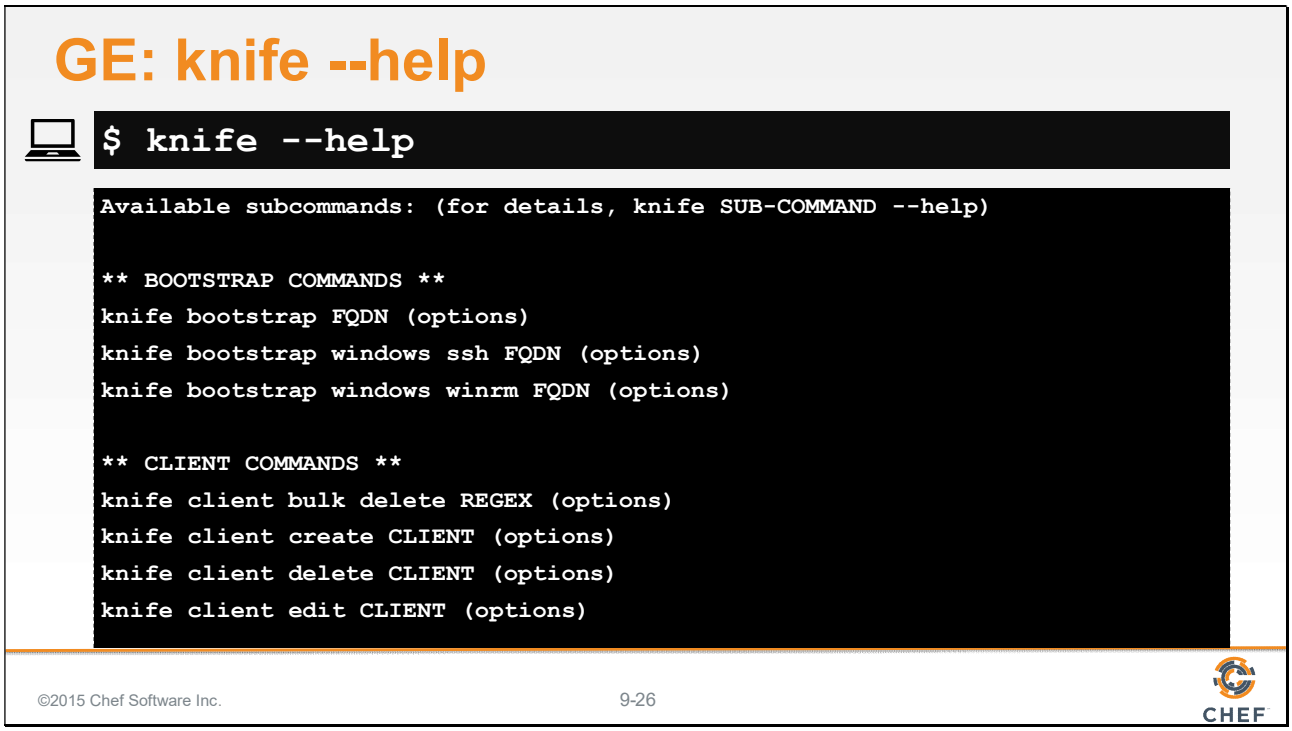
Slide 25



knife is a command-line tool that allows us to request and send information to the Chef Server.

knife helps users manage: nodes; cookbooks; roles; environments; and more. knife does this through a series of sub-commands.

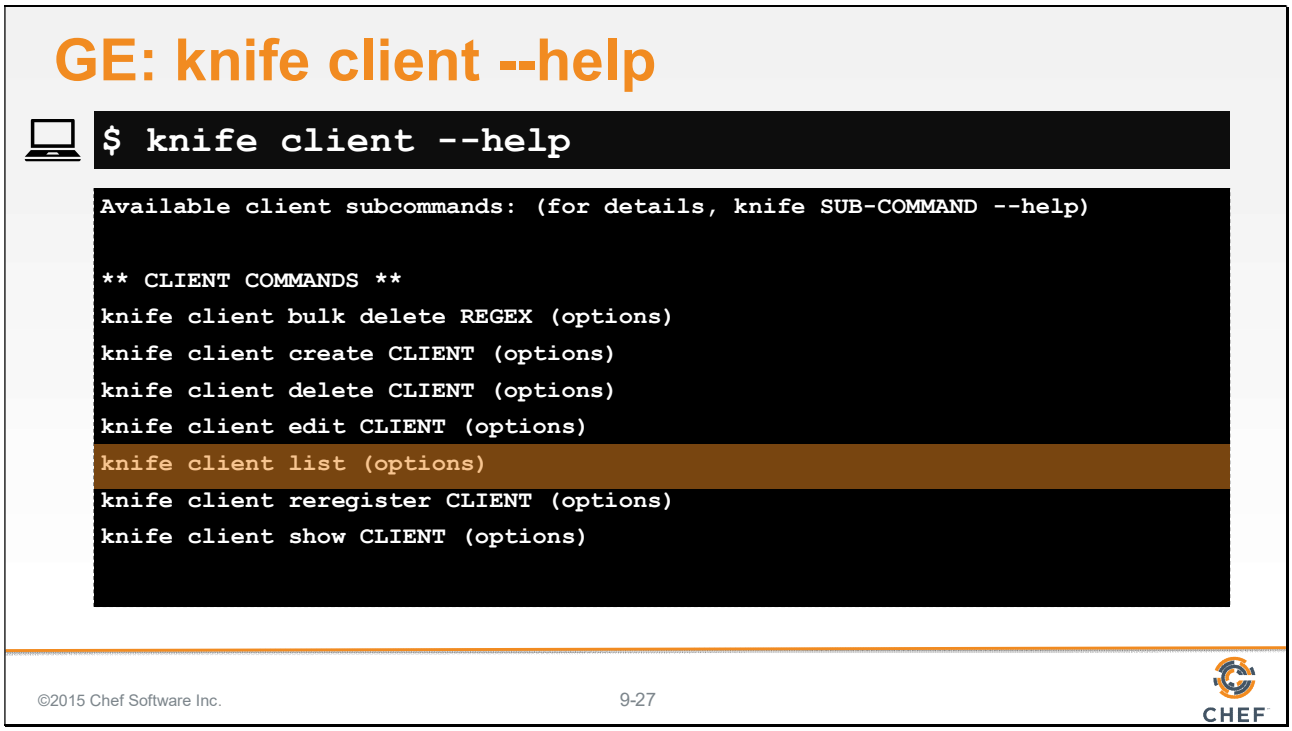
Slide 26



You can look at all the commands with 'knife –help'.

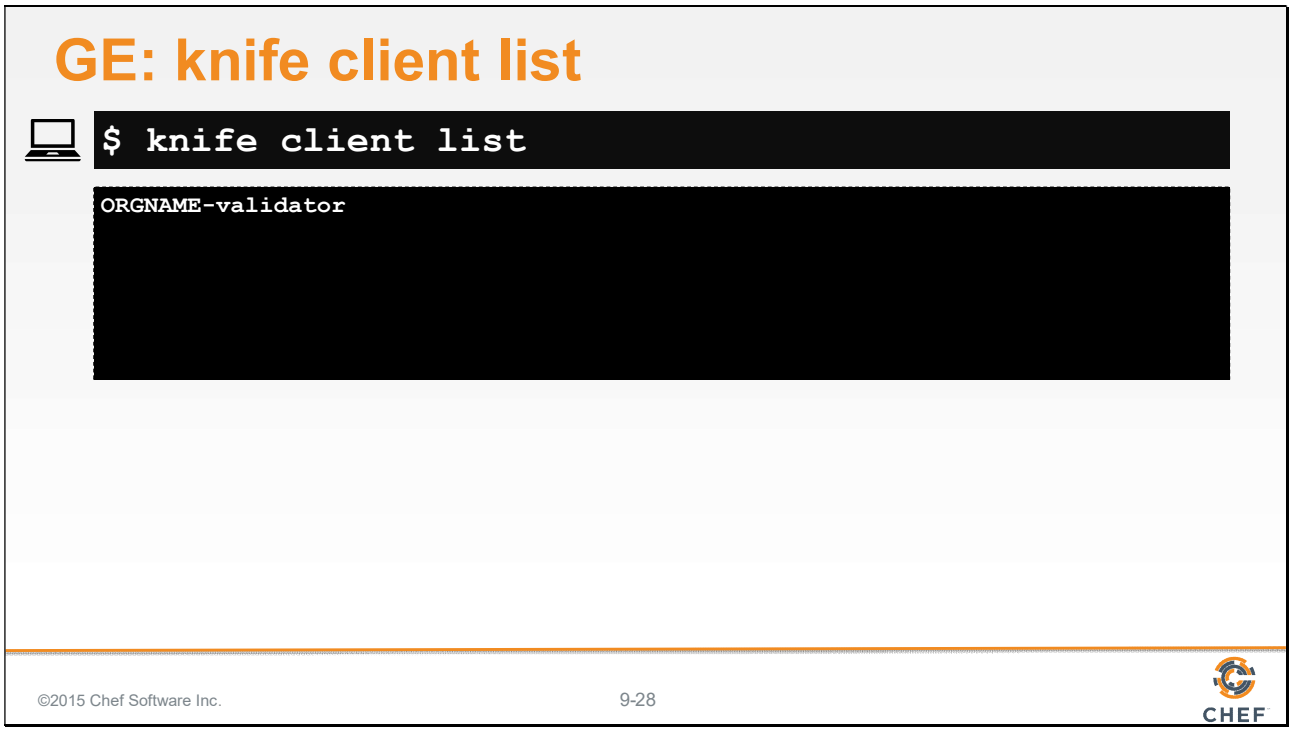
This will display all the sub-commands available. In your case you want to verify that the client list contains a single entry so you need to look for help for the specific command 'knife client --help'.

Slide 27



This will give us an even smaller subset of the commands related specifically to asking the Chef Server about client information. A general command is the list command which will output all the clients that the Chef Server currently maintains.

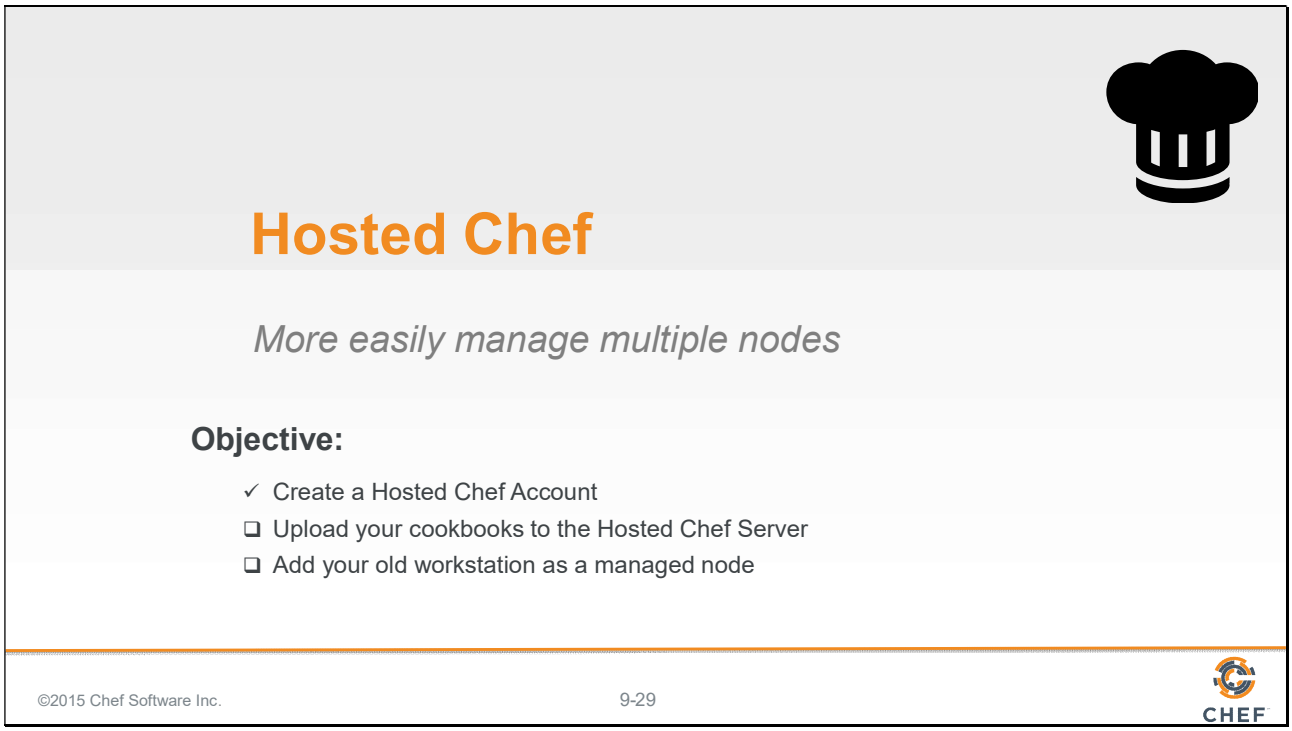
Slide 28



For your Chef Server account there should be a single client that is the organization name: validator. This is a special key that has access to the Chef Server. The important thing is that the result does not contain an error with the configuration or authenticating with the Chef Server.

If you receive an error ensure that you: typed the command correctly; executed the command within the chef repository directory; are connected to the internet and not blocking ssl connections from your own system's proxy servers or virtual private networks; and have a .chef directory, within the chef repository, which contains the knife configuration file (knife.rb), personal key, and organizational key.

Slide 29



With all that complete, you are now able to communicate with the Chef Server. At this point we will refer to the system in front of you, with the chef repository, the configuration, and the keys installed as your workstation.

When working with Chef with a Chef Server, the workstation is the location where you will compose your cookbook code. When that code is complete, you will then upload it to the Chef Server.

Slide 30

GE: knife cookbook --help

```
$ knife cookbook --help
```

```
** COOKBOOK COMMANDS **
knife cookbook bulk delete REGEX (options)
knife cookbook create COOKBOOK (options)
knife cookbook delete COOKBOOK VERSION (options)
knife cookbook download COOKBOOK [VERSION] (options)
knife cookbook list (options)
knife cookbook metadata COOKBOOK (options)
knife cookbook metadata from FILE (options)
knife cookbook show COOKBOOK [VERSION] [PART] [FILENAME] (options)
knife cookbook test [COOKBOOKS...] (options)
knife cookbook upload [COOKBOOKS...] (options)
```


Similar to asking the Chef Server about the list of available clients, you can also ask for information about cookbooks. You can find all the commands related to the cookbooks subcommand by running `knife cookbook --help`.

Similar to the list of clients, you can examine a list of cookbooks.

Slide 31

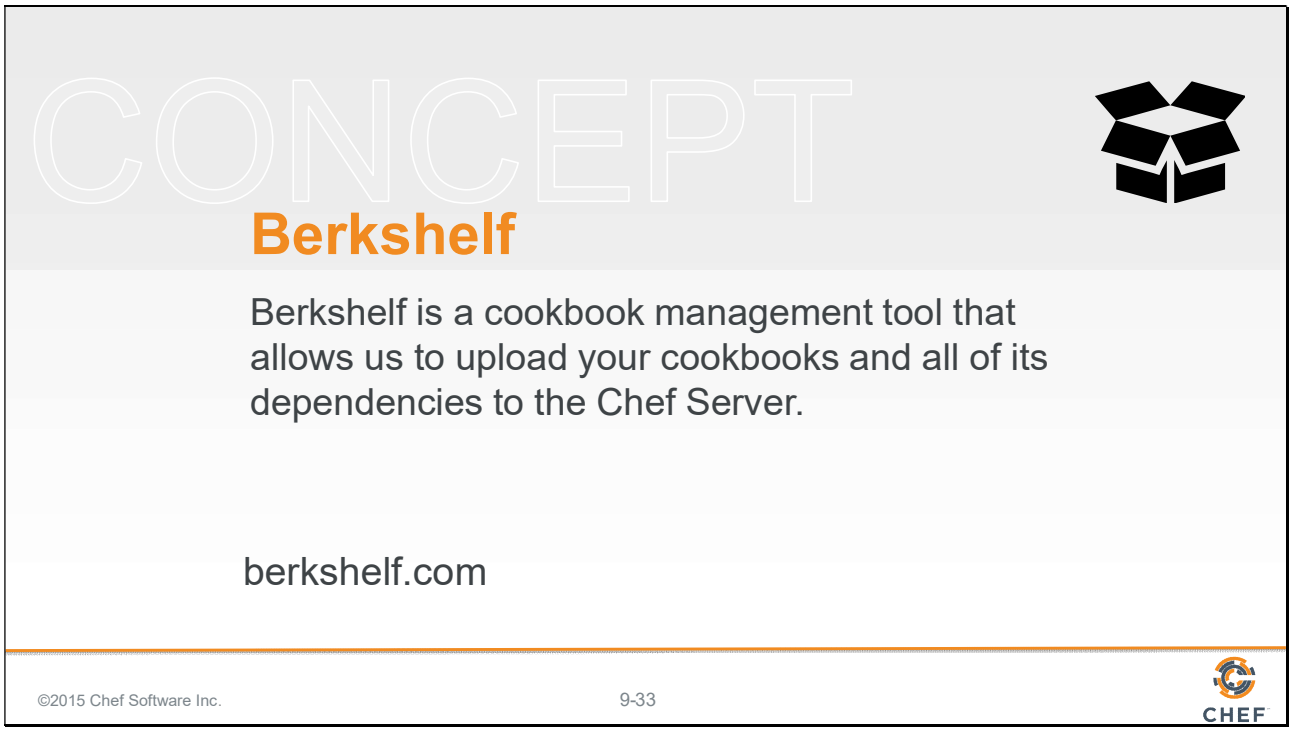
Running this command will return the cookbooks currently uploaded to the Chef Server. The empty response should come as no surprise.

You want to change that. So you are going to upload each of your cookbooks to the Chef Server.

Slide 32

To upload a cookbook to the Chef Server you need to be within the directory of the cookbook. Let us start with the apache cookbook. Change directory into the apache cookbook directory which is within the cookbooks directory.

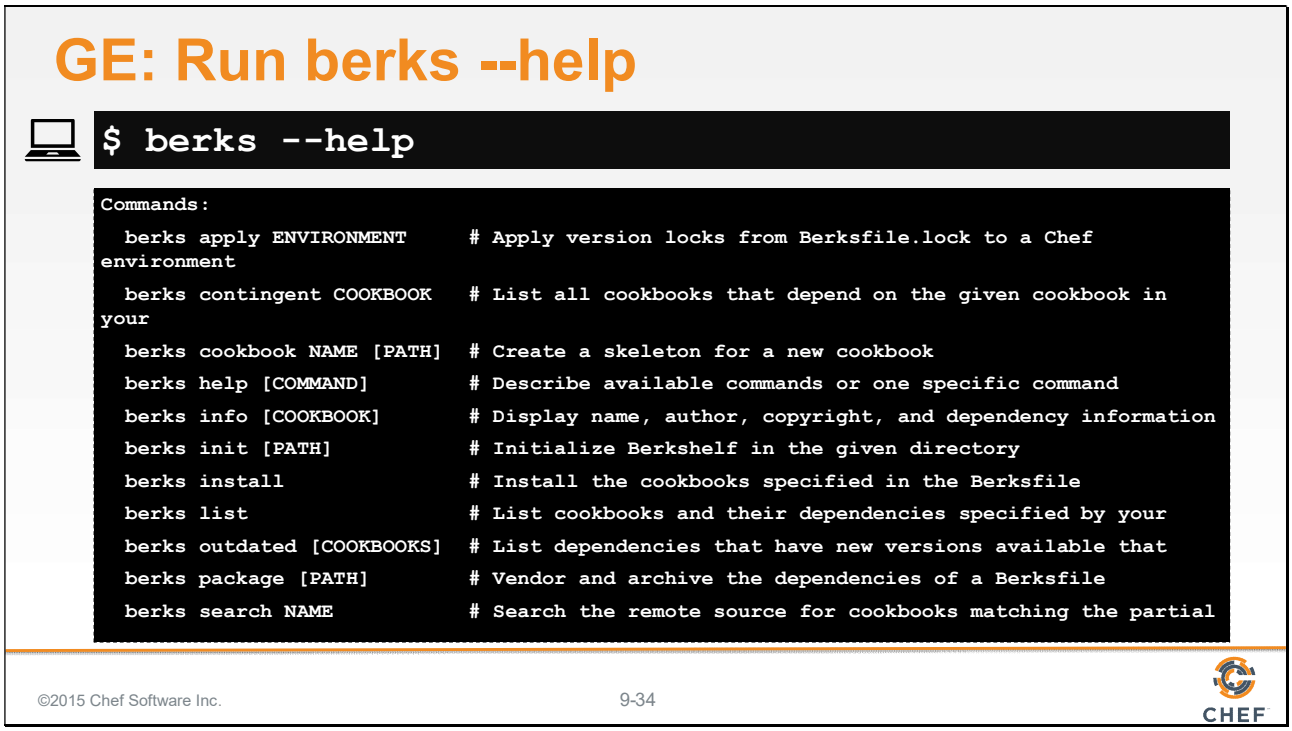
Slide 33



To upload the cookbook you will need to use another tool called Berkshelf.

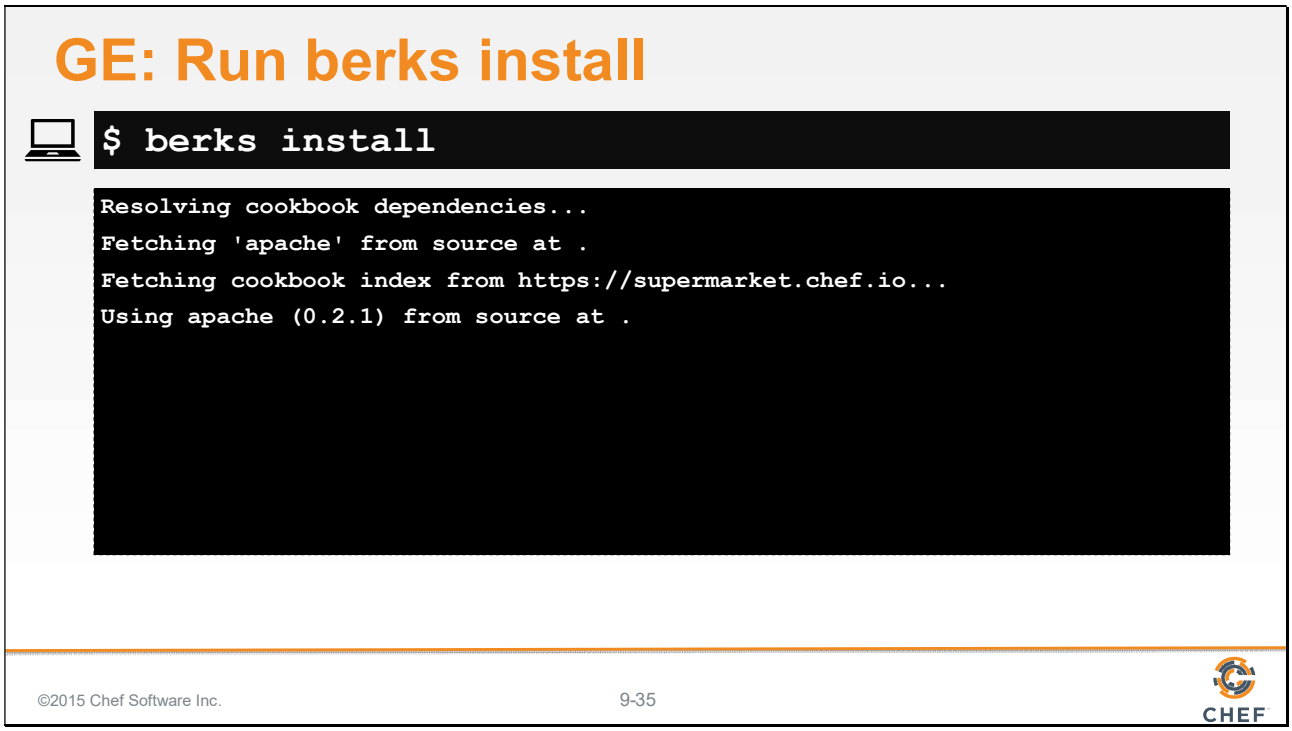
Berkshelf is a cookbook management tool that allows us to upload your cookbooks and all of its dependencies to the Chef Server. In this instance, your current cookbooks have no dependencies, but in the future when they do, Berkshelf will assist you in ensuring those are all uploaded.

Slide 34



Berkshelf is a command-line tool that you can ask to see available the commands.

Slide 35

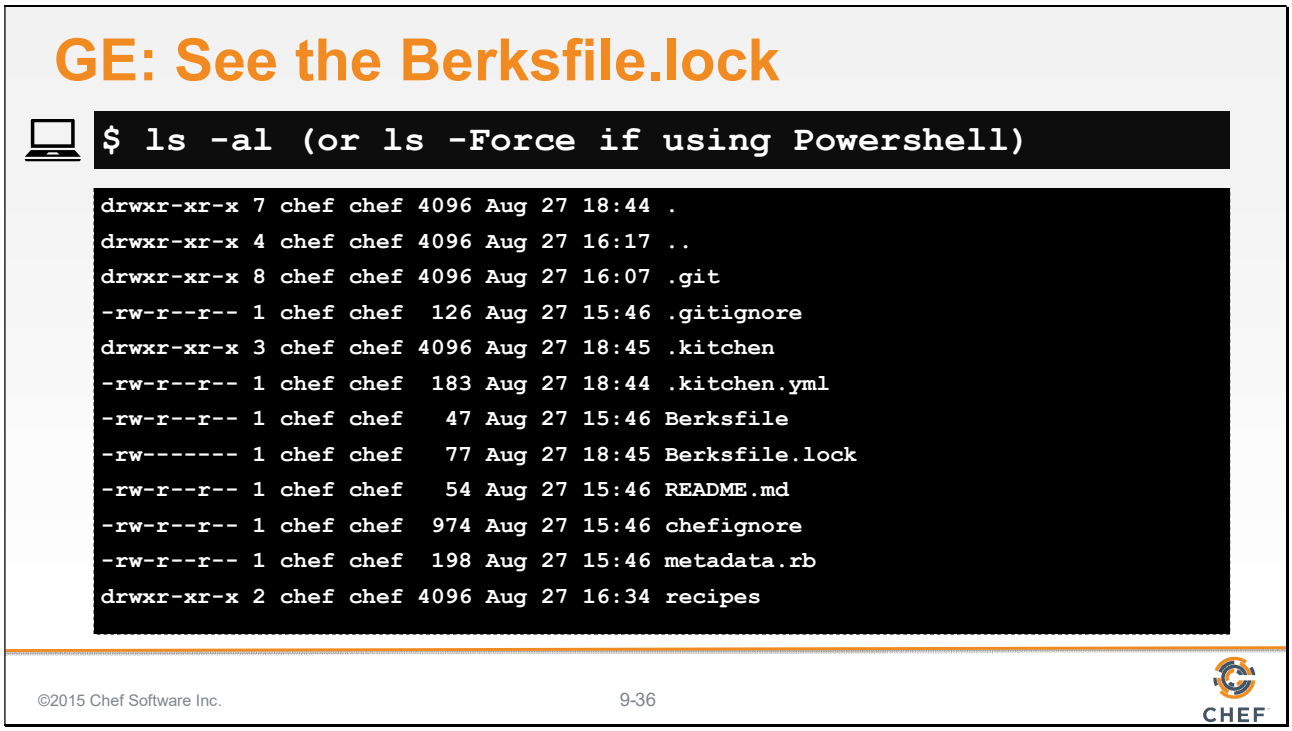


Berkshelf is used on a per-cookbook basis. As dependencies are often per cookbook you'll need to change into the directory of the cookbook.

You should install any dependencies that your cookbook might have. Again, in this instance there are no dependencies external to this cookbook but Berkshelf ensures that this is the case when it runs the 'berks install' command.

You'll see that it finds the current cookbook within your current directory, it contacts the Supermarket for any external dependencies, and then ...

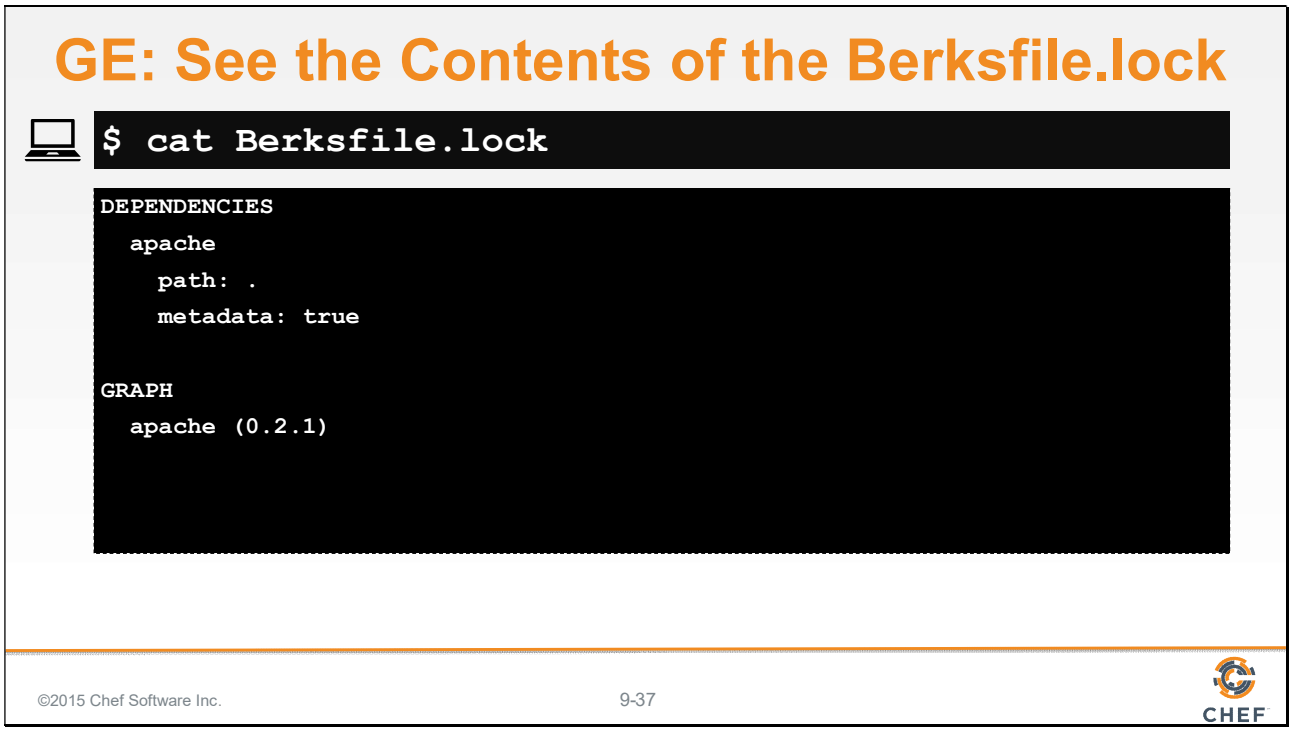
Slide 36



...it completes by writing a Berksfile.lock to the file system.

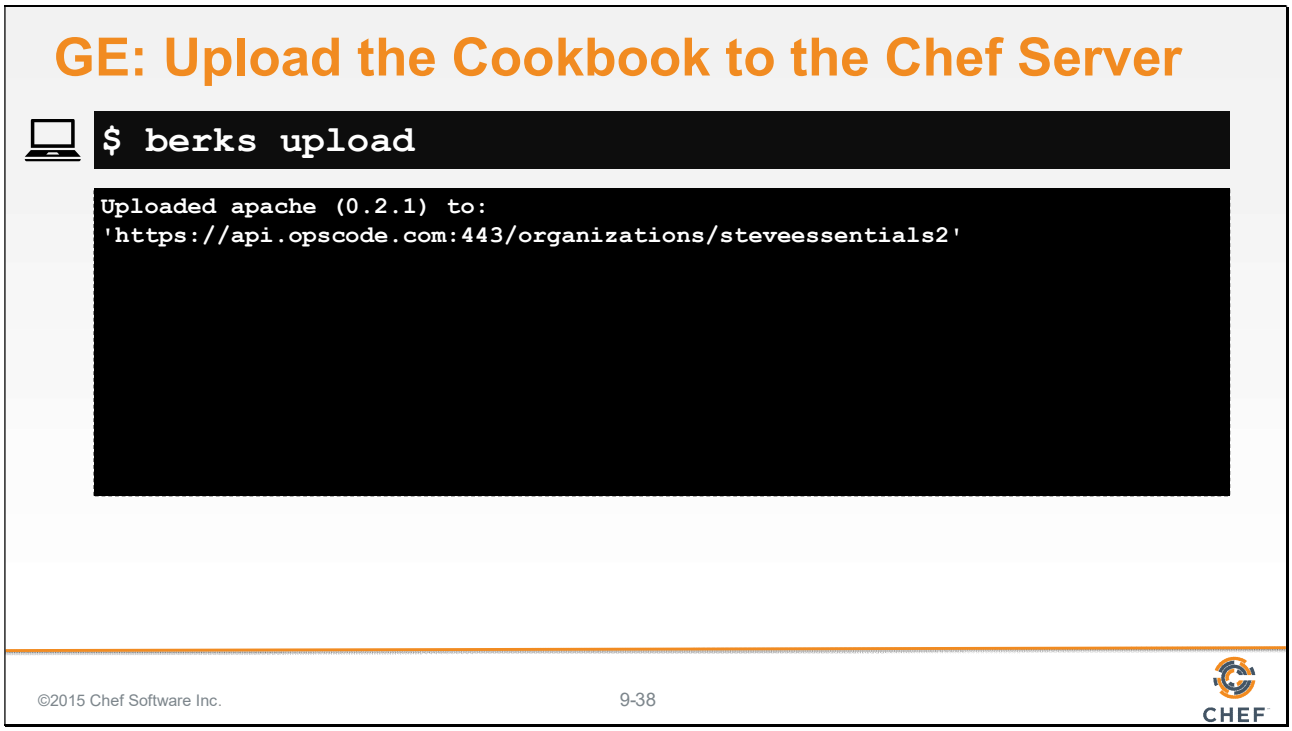
The Berksfile.lock is a receipt of all the cookbooks and dependencies found at the exact moment that you ran 'berks install'.

Slide 37



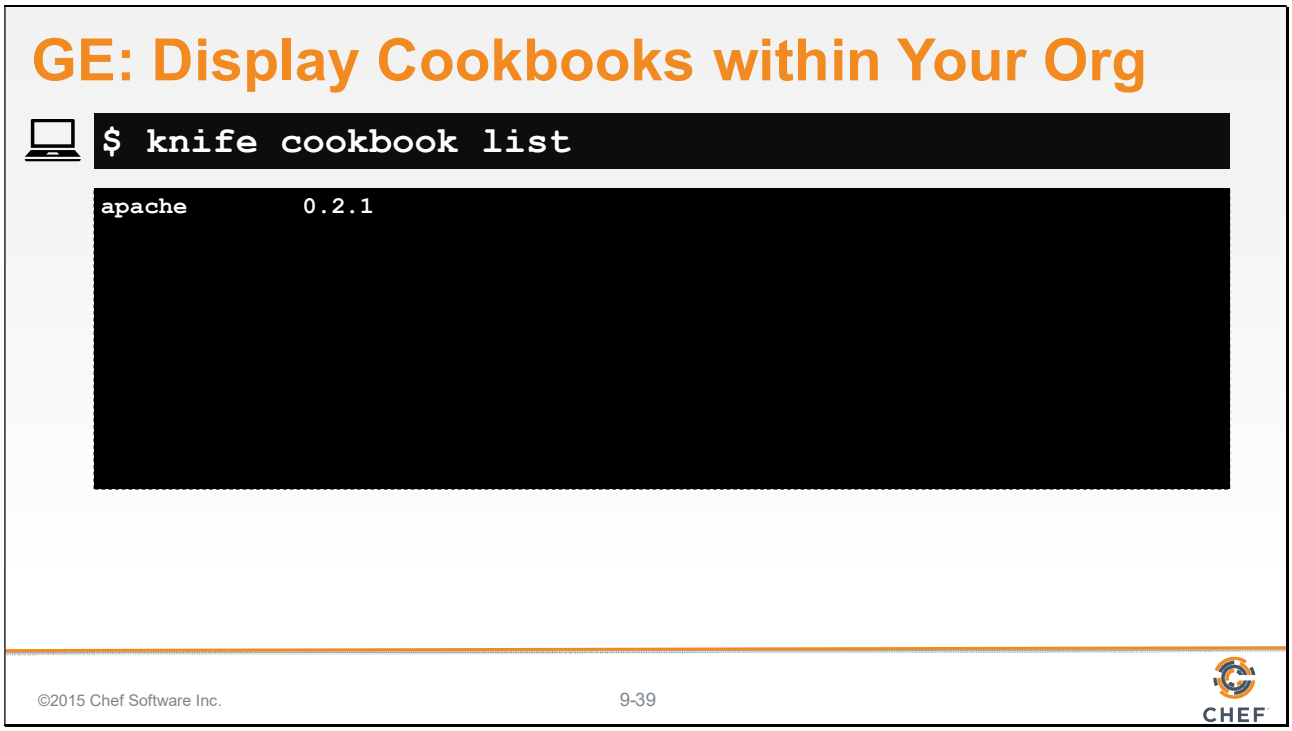
This lock file is useful to ensure that in the future you use the same dependencies when working with the cookbook.

Slide 38



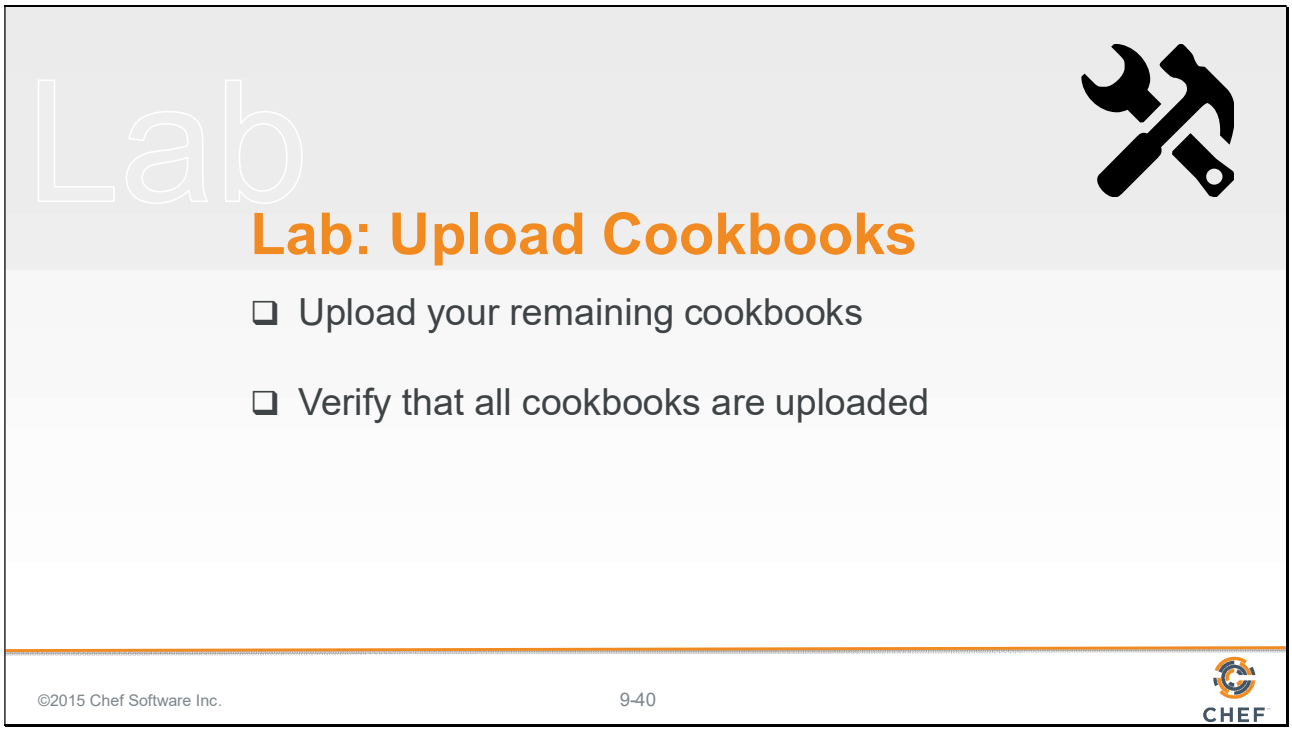
With the dependencies accounted for, it is time to upload the to the Chef Server. This is another sub-command that Berkshelf provides called 'upload'. Run the command to upload the apache cookbook to the Chef Server.

Slide 39



When that is complete you can return to the cookbook command that allows you to display the cookbooks within your organization by running this command. This will show you that the Chef Server has the apache cookbook that you have uploaded.

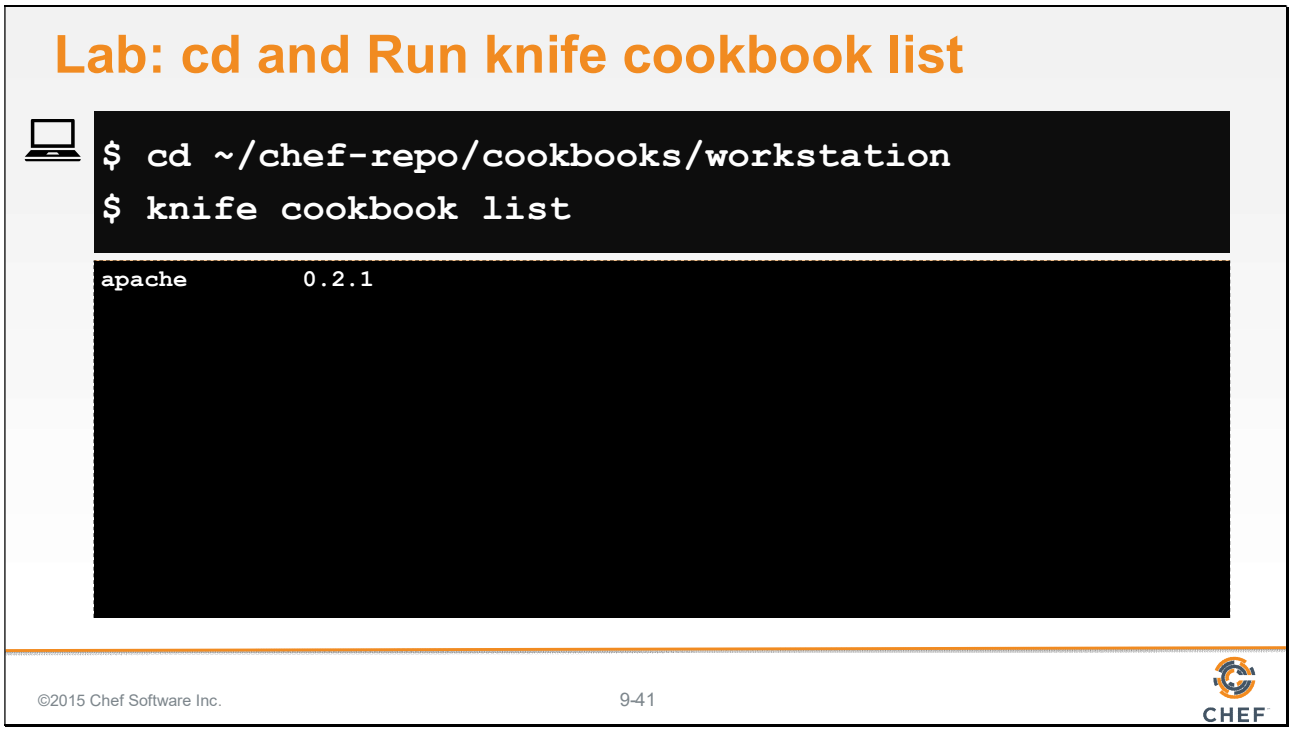
Slide 40



As a lab, upload the remaining cookbooks within the cookbooks directory. After you have done that verify that the cookbooks have been uploaded.

Instructor Note: Allow 5 minutes to complete this exercise.

Slide 41

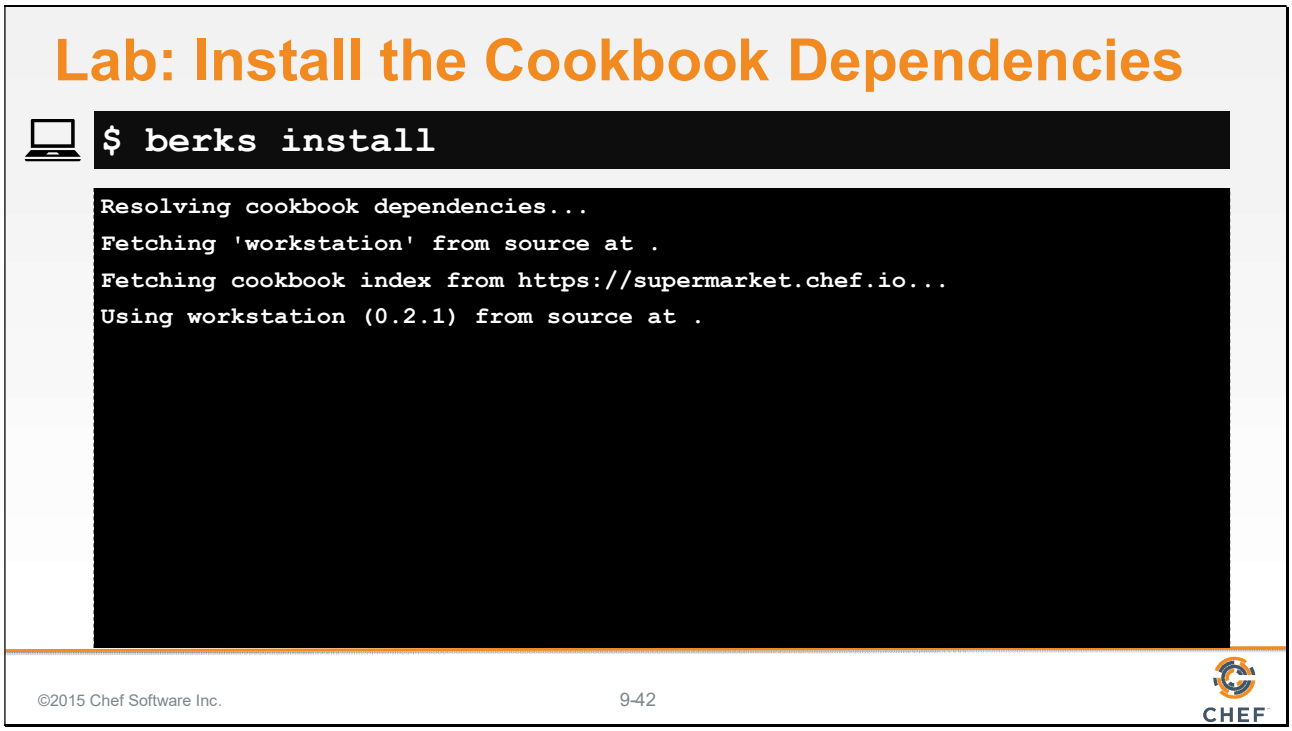


The one remaining cookbook is the workstation cookbook. Berkshelf is a cookbook management tool that examines the contents and dependencies of a single cookbook.

Change into the cookbooks directory.

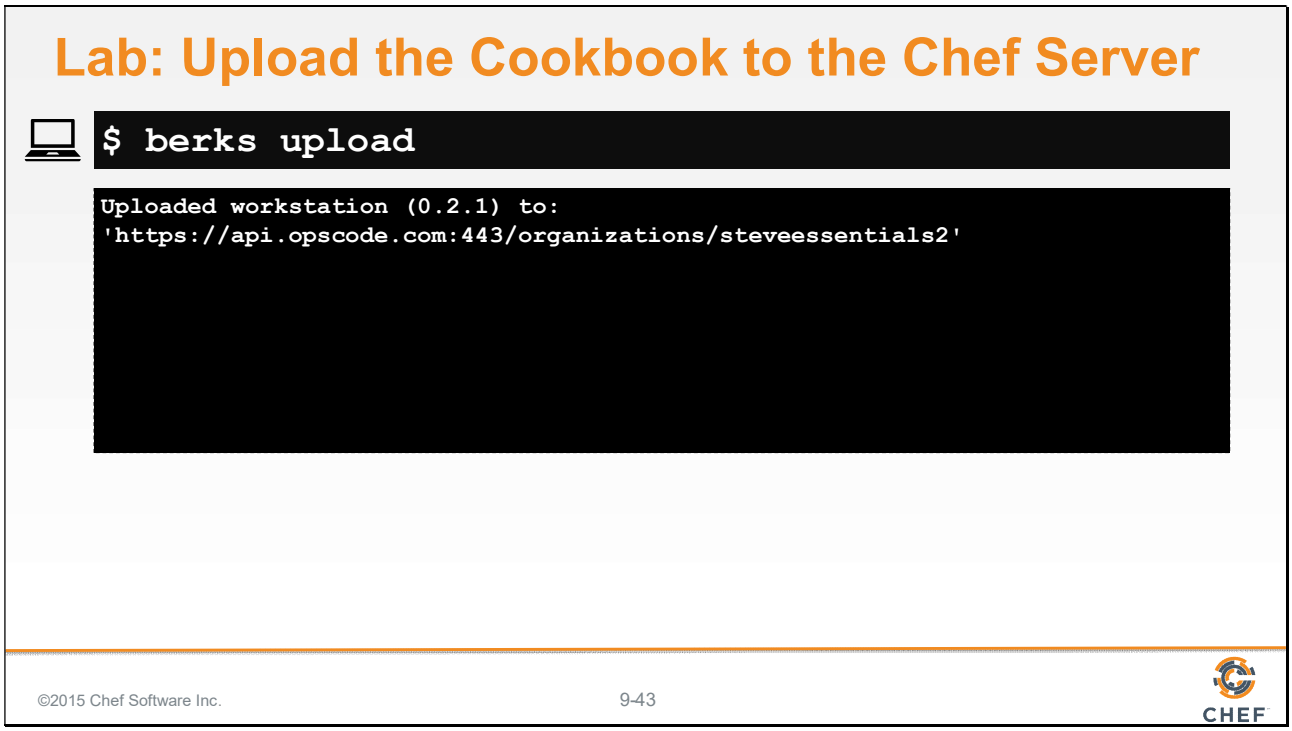
Verify that the cookbook is not currently uploaded.

Slide 42



Run "berks install" to install all the cookbook dependencies.

Slide 43



Run "berks upload" to upload the cookbook and all its dependencies to the Chef Server.

Slide 44

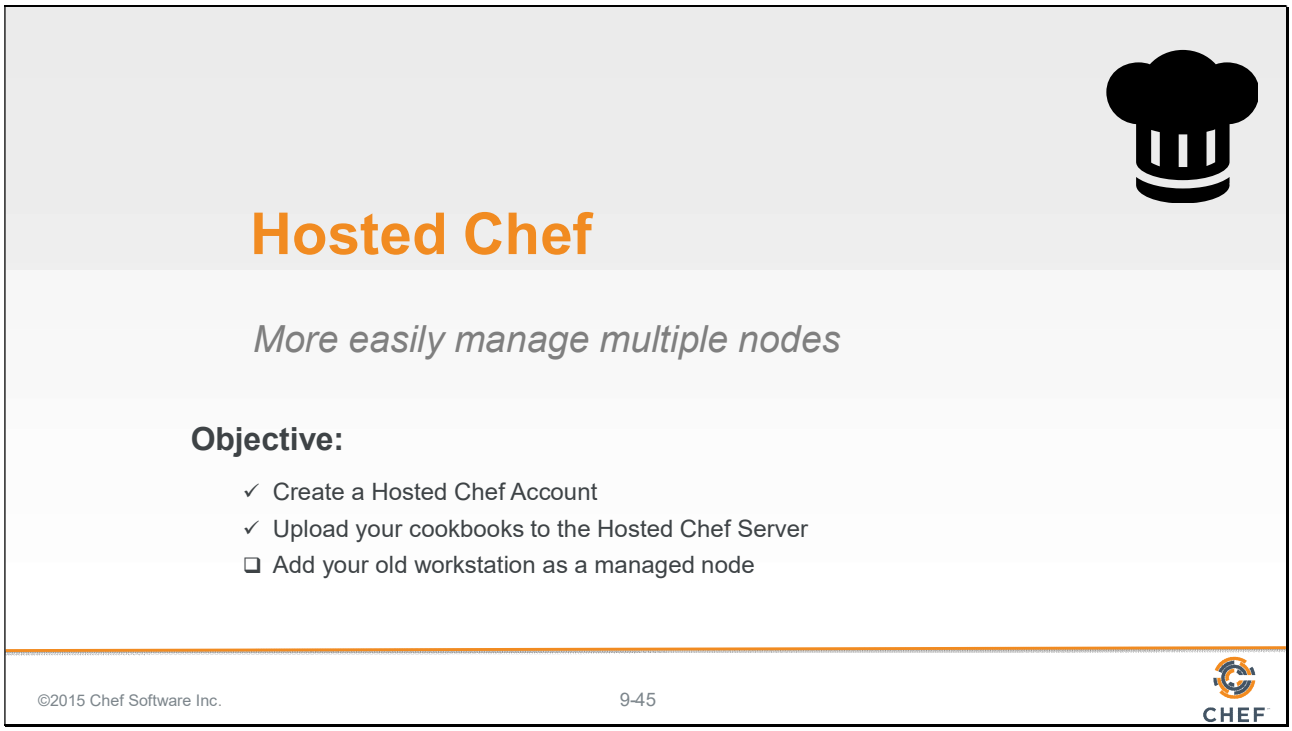
Lab: Is the workstation Cookbook Uploaded?

```
$ knife cookbook list
```

```
apache          0.2.1
workstation     0.2.1
```

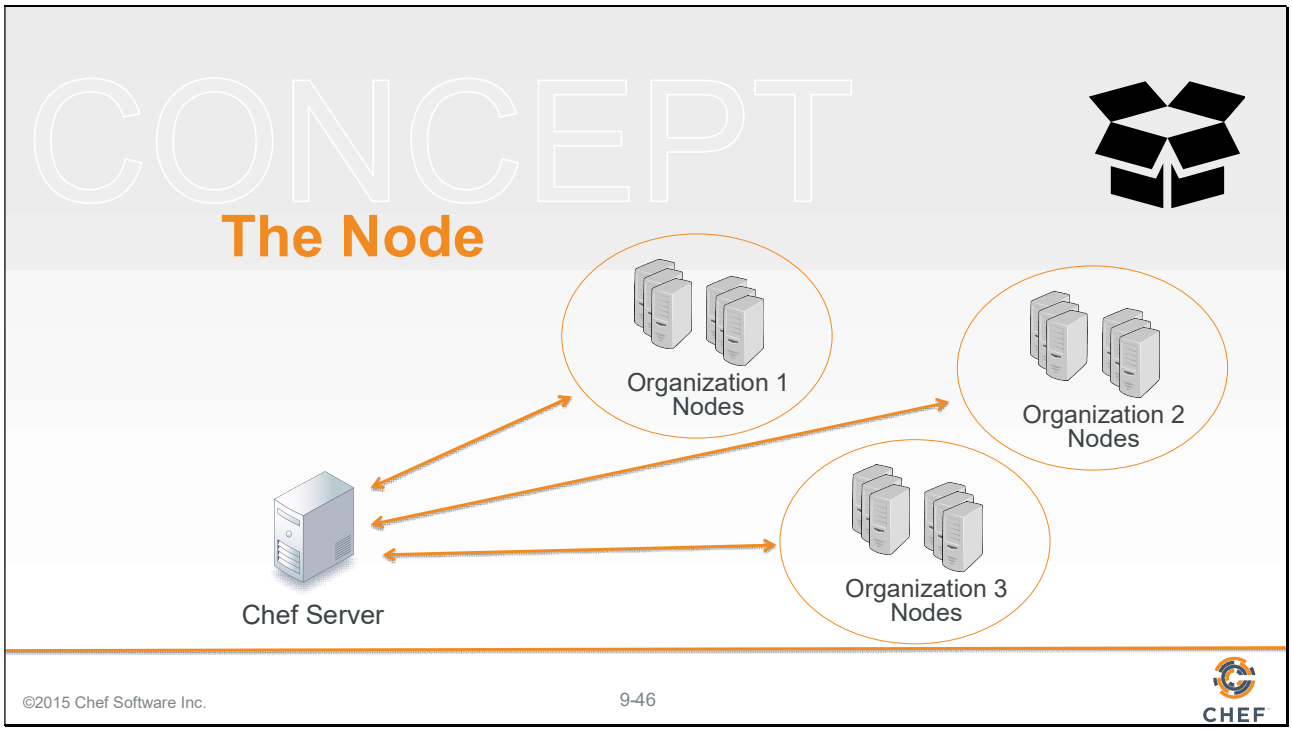

Lastly, run "knife cookbook list" to validate that the workstation cookbook is now uploaded to the Chef Server.

Slide 45



You have one remaining objective and that is to add an instance as a node within your organization.

Slide 46



As you know by now, a node is a server that Chef is managing. A node could be a web server, an application server, a database server, a load balancer, and so on.

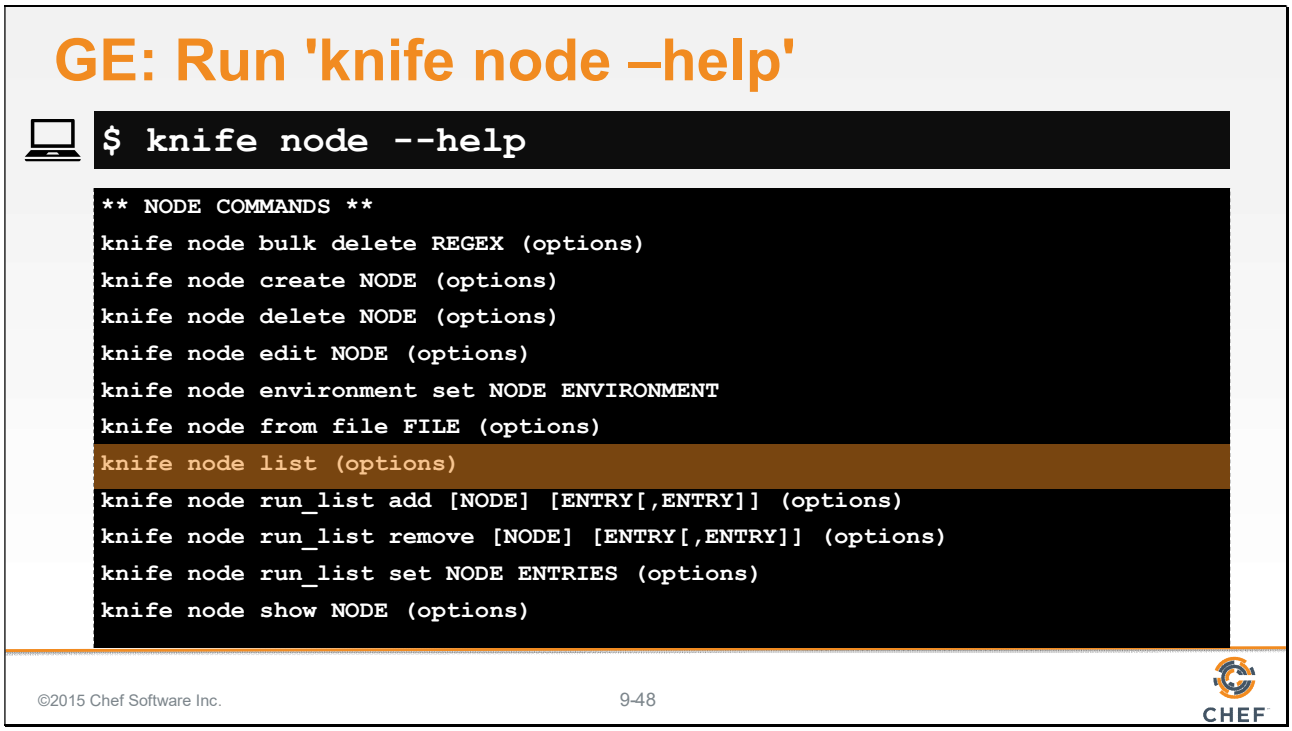
A node can only join one organization. To be a node means that it has Chef installed, has configuration files in place, and when you run the chef-client application with no parameters it will successfully contact the Chef Server and ask it for the run list that it should apply and the cookbooks required to execute that run list.

When a node is part of the organization you manage that information on the Chef Server as well. A Chef Server can manage multiple organizations. Managing that information in a Chef Server allows us to use for inventory, querying and searching.

Slide 47

Let's add the instance we used previously as a workstation now as a managed node. Return to the root of the chef repository.

Slide 48

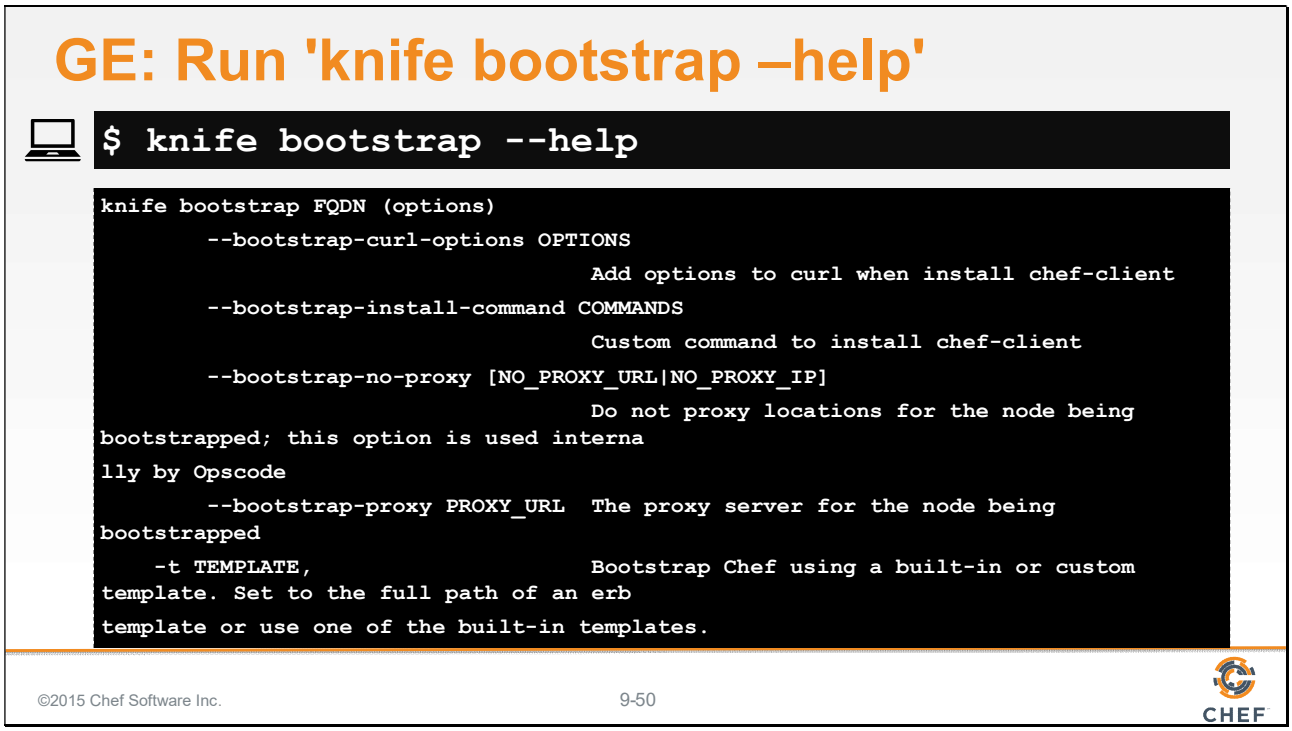


Verify that you have no existing nodes within your organization. You can use the 'knife node –help' command to see that you can ask for the list of all nodes within your organization with the list command.

Slide 49

Run "knife node list" to see that you have no nodes currently registered with your Chef Server. At this point the results should be blank.

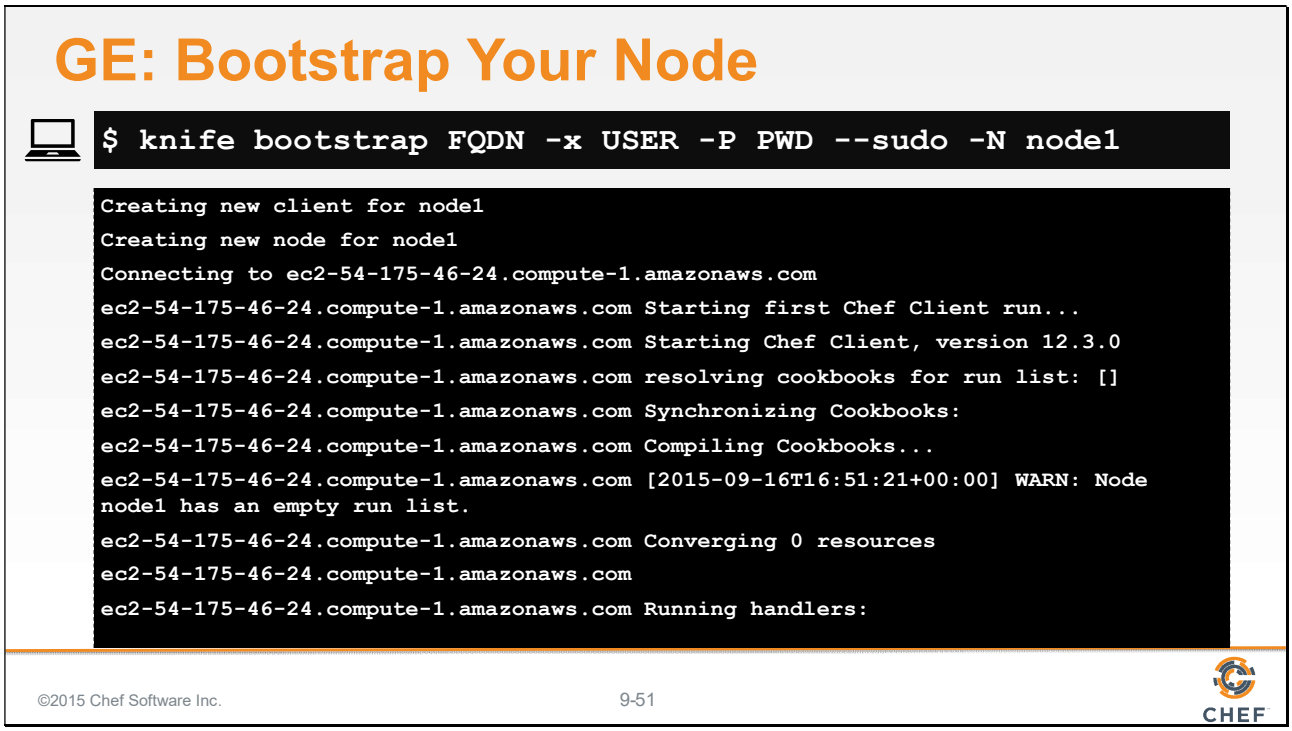
Slide 50



Knife provides a bootstrap subcommand that takes a number of options.

When you bootstrap an instance it is performing the following: Installing chef tools if they are not already installed; Configuring Chef to communicate with the Chef Server; Running chef-client to apply a default run list.

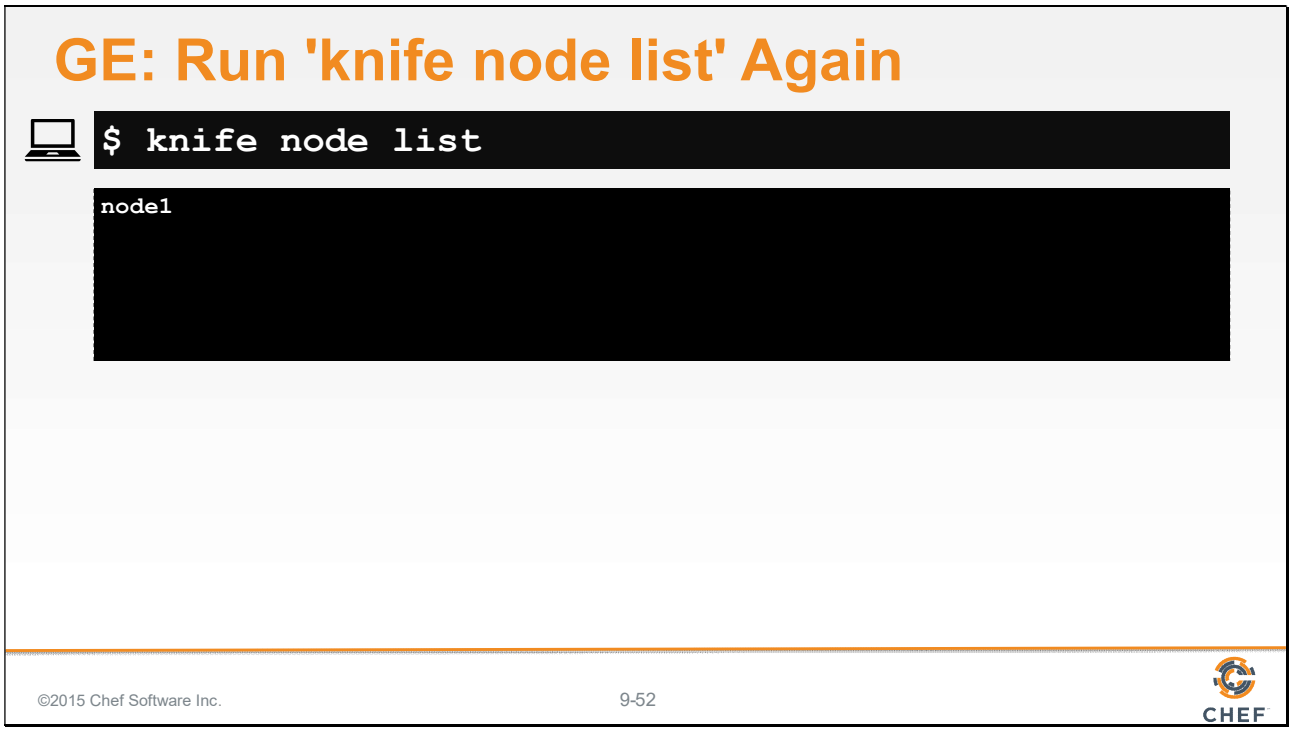
Slide 51



To communicate with the remote instance you need to provide it the credentials to connect to the system. Use the user name with the '-x' flag and the password '-P' flag. Include the '--sudo' flag because you are installing software and writing configuration to directories traditionally owned by the root user. Name the node with the '-N' flag. This is optional but makes it easier for us to communicate. When we ask you to look at the details of node 1 or login to node 1, it will be easier to remember than the fully-qualified domain name.

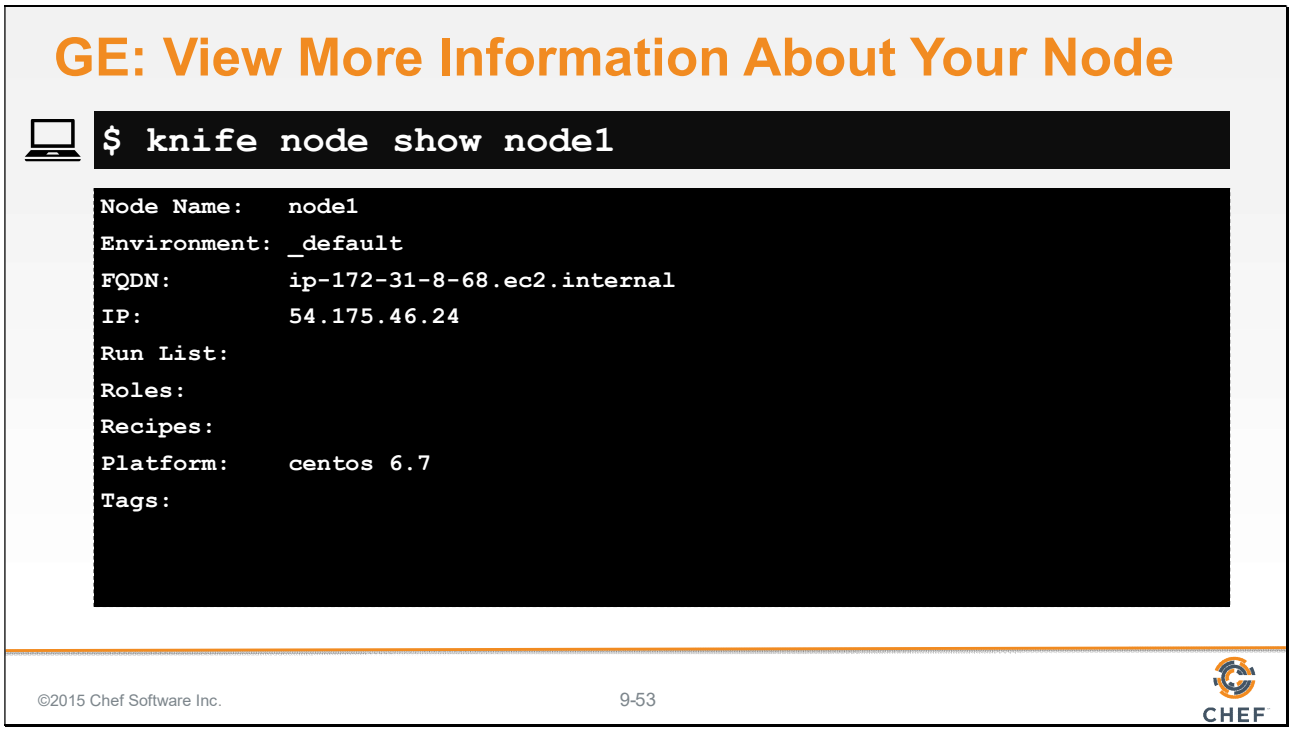
When executing the command, the output will tell us what it installed and ran.

Slide 52



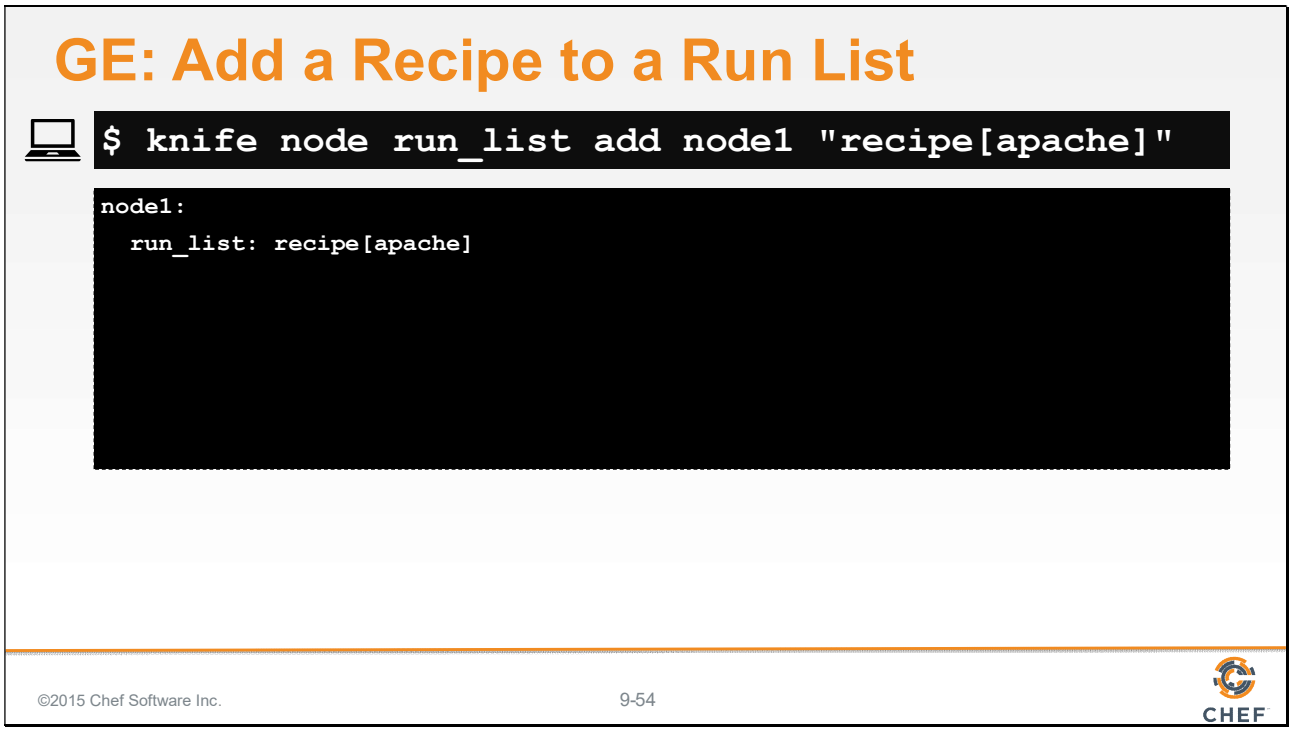
When bootstrapping is done, you can see that your organization knows about the new node by again running the command "knife node list". You now see that you have a new node, node1, uploaded to the Chef Server.

Slide 53



You can see more information about a particular node with the command 'knife node show node1'. This will display a summary of the node information that the Chef Server stores.

Slide 54



node1 does not have a list of recipes that it applies to the system by default. You can make Chef Server tell node1 to apply a specific run-list the next time node 1 runs 'chef-client'.

You can do that through the 'knife node run_list add' command. In this example, you are adding to node1's run-list the apache cookbook's default recipe.

Slide 55

Hosted Chef

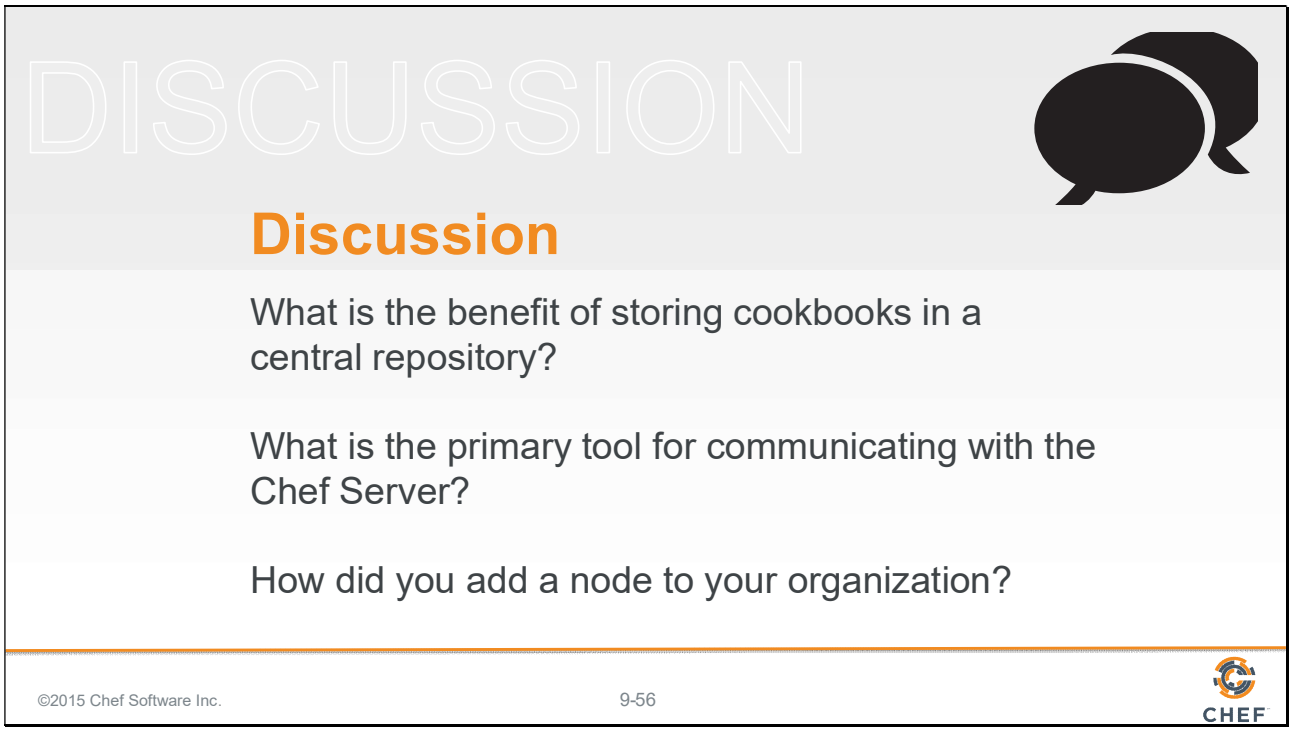
More easily manage multiple nodes

Objective:

- ✓ Create a Hosted Chef Account
- ✓ Upload your cookbooks to the Hosted Chef Server
- ✓ Add your old workstation as a managed node

CHEF

Slide 56



Answer these questions.

With your answers, turn to another person and alternate asking each other asking these questions and sharing your answers.

Slide 57

DISCUSSION

Q&A

What questions can you help you answer?

- Chef Server
- Managed Chef
- Berkshelf
- Bootstrapping Nodes

CHEF

With all of the objectives complete you are finished with this section. What question can you answer for you?

Slide 58