# 12: Roles



Slide 2



In this module you will give your nodes a role to better describe them so you can configure them in a similar manner.

Slide 3



Up until this point it has been a mouthful to describe the nodes within our organization. We have two nodes, node1 and node3, that have the apache cookbook's default recipe in their run list. We have one node, node2, which has the myhaproxy cookbook's default recipe in its run list.

The Chef Server allows us to create and manage roles. A role describes a run list of recipes that are executed on the node. A role may also define new defaults or overrides for existing cookbook attribute values. Similar to what we accomplished with the wrapper cookbook.

A node may have zero or roles assigned to it.

Slide 4



When you assign a role to a node you do so in its run list. This allows us to configure many nodes in a similar fashion because we no longer need to re-create a long run list for each node--we simply give it a role or all the roles it needs to accomplish its desired function.

Slide 5



In this section you will create a load\_balancer role and assign it to the run list of node2. You will also will create a web role and assign it to the run list of node1 and node3.

This is particularly powerful because we will no longer have to manage each of these identical nodes individually, instead we can make changes to the role that they share and all of the nodes that have this role will update accordingly.

Slide 6



Return to the base of your Chef repository and then run 'knife role --help' to see the available commands. Similar to other commands, you can see that 'knife role' supports the ability to list currently-defined roles.

Slide 7



When you run 'knife role list' you can see from its lack of response that you have no roles defined.

Slide 8



Create a **roles** directory if necessary. If you are using the Chef Starter Kit this directory may already exist.

Slide 9



Create a file named load\_balancer.rb. This is a ruby file that contains specific methods that allow you to express details about the role. You'll see that the role has a name, a description, and run list.

The name of the role as a practice will share the name of the ruby file unless it cannot for some reason. The name of the role should clearly describe what it attempts accomplish. The description of the role helps reinforce or clarify the intended purpose of the role. When selecting a role name that is not clear it is important that a helpful description is provided to help ensure everyone on the team understands its purpose. The run list defines the list of recipes that give the role its purpose. Currently the load\_balancer role defines a single recipe - the myhaproxy cookbook's default recipe.

Slide 10



Now you need to upload it to the Chef Server. This is done through the command 'knife role from file load\_balancer.rb'.

The knife tool understands that you are uploading a role file and will look within the roles folder to find a file named knife role from file load\_balancer.rb.

Slide 11



With the role uploaded, it is time to validate that the Chef Server received it correctly. We can do that by again asking the Chef Server for a list of all the roles on the system.

Slide 12



You can ask for more details about a specific role using the above command. In this example we are requesting specific details about the role named load\_balancer.

Slide 13



Run 'knife node --help' to see its options.

Slide 14



The last step is to redefine the run list for node2. We want the run list to contain only the load\_balancer role.

Previously, we used the command 'knife node run\_list add' to append a new item to the existing run list. There is also a command that allows us to remove an item from the run list. There is a command that allows us to set the run list to a value provided. This will replace the existing run list with a new one that we provide.

Slide 15



After you update the run list, you can verify that the node has the correctly-defined run list by running 'knife node show node2'.

Slide 16



You can use 'knife ssh' to run 'sudo chef-client' on all the nodes again to ensure that nothing has changed.

In this instance we only interested in having node2 run the command so we can get a little more creative with the search criteria and find nodes with the role load\_balancer. In this case there is only one result.

Within the results, nothing should change. Switching over to the role did not change the fundamental recipes that were applied to the node.

Slide 17



Now if you want to setup a new node in the future to act as a load balancer, you can now simply set the new node's run list to be the load\_balancer role and it will have identical functionality with all the other nodes that define this role.

Slide 18



In this lab, define a new role named 'web' that has the run list: including the apache cookbook's default recipe.

When you're done defining the role, upload it to the Chef Server, and then set the run list on node1 and node3 to the role that you have defined.

And for good measure, though nothing should have changed, run 'sudo chef-client' on both node1 and node3 to ensure that no functionality has been lost.

Instructor Note: Allow 10 minutes to complete this exercise

Slide 19



First we create a file named web.rb in the roles directory.

The name of the role is web. The description should be Web Server. The run list you define should contain the apache cookbook's default recipe.

Slide 20



You need to share the role with the Chef Server so upload that file.

Use the command 'knife role from file web.rb'. 'knife' knows where to look for that role to upload it.

Slide 21



Verify that the role can be found on the Chef Server.

Slide 22



Verify specific information about the role. Specifically, does it have the run list that we defined?

Slide 23



Set node1's run list to be the web role.

Slide 24



And we then set node3's run list to be the web role.

Slide 25



To verify that everything is working the same as before, run 'knife ssh' for both of these nodes. In this instance the query syntax is going to find all nodes with the role set to web.

Slide 26



With that we now have made it far easier to talk about our nodes. We can more casually describe a node as a 'web' server node or a 'load\_balancer' node.

In the future if we needed to ensure that these types of nodes needed to run additional recipes, we could return to the role file, update its run list, and then upload it to the Chef Server again.

Slide 27



Answer these questions.

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

Slide 28



Slide 29

