Roles

Giving your Nodes a Role

Objectives

After completing this module, you should be able to

- Assign roles to nodes so you can better describe them and configure them in a similar manner.

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.

Roles

When you assign a role to a node you do so in its run list.

This allows you to configure many nodes in a similar fashion.

GL: Roles for Everyone

We will give our nodes a role to better describe them and so we can configure them in a similar manner.

Objective:

- ❑ Give our load balancer node a "load_balancer" Role
- ❑ Give our web nodes a "web" Role

GL: What Can 'knife role' Do?

```
$ cd ~/chef-repo
$ knife role --help
```

```
** ROLE COMMANDS **
knife role bulk delete REGEX (options)
knife role create ROLE (options)
knife role delete ROLE (options)
knife role edit ROLE (options)
knife role env_run_list add [ROLE] [ENVIRONMENT] [ENTRY[,ENTRY]] (options)
knife role env_run_list clear [ROLE] [ENVIRONMENT]
knife role env_run_list remove [ROLE] [ENVIRONMENT] [ENTRIES]
knife role env_run_list replace [ROLE] [ENVIRONMENT] [OLD_ENTRY] [NEW_ENTRY]
knife role env_run_list set [ROLE] [ENVIRONMENT] [ENTRIES]
knife role from file FILE [FILE..] (options)
```

GL: Run 'knife role list'

```
$ knife role list
```


CHEF

GL: Create a Roles Directory

```
$ mkdir roles
```

GL: Create the load_balancer.rb

~/chef-repo/roles/load_balancer.rb

```
name 'load_balancer'
description 'Load Balancer'
run_list 'recipe[myhaproxy]'
```

GL: Upload it to the Chef Server

```
$ knife role from file load_balancer.rb
```

```
Updated Role load_balancer!
```

GL: Validate Chef Server Received It

```
$ knife role list
```

```
load_balancer
```

GL: View Details of the Role

```
$ knife role show load_balancer
```

```
chef_type:            role
default_attributes:
description:          Load Balancer
env_run_lists:
json_class:           Chef::Role
name:                 load_balancer
override_attributes:
run_list:             recipe[myhaproxy]
```

GL: Run 'knife node --help'

```
$ knife node --help
```

```
** NODE COMMANDS **
knife node bulk delete REGEX (options)
knife node create NODE (options)
knife node delete NODE (options)
knife node edit NODE (options)
knife node environment set NODE ENVIRONMENT
knife node from file FILE (options)
knife node list (options)
knife node run_list add [NODE] [ENTRY[,ENTRY]] (options)
knife node run_list remove [NODE] [ENTRY[,ENTRY]] (options)
knife node run_list set NODE ENTRIES (options)
knife node show NODE (options)
```

GL: Set the load_balancer Role to node2

```
$ knife node run_list set node2 "role[load_balancer]"
```

```
node2:
  run_list: role[load_balancer]
```

GL: Verify the Run List

```
$ knife node show node2
```

```
Node Name:   node2
Environment: _default
FQDN:        ip-172-31-0-128.ec2.internal
IP:          54.210.192.12
Run List:    role[load_balancer]
Roles:
Recipes:     myhaproxy, myhaproxy::default, haproxy::default, haproxy::install_package
Platform:    centos 6.6
Tags:
```

GL: Converge All the Load Balancer Nodes

```
$ knife ssh "role:load_balancer" -x USER -P PWD "sudo chef-client"
```

```
ec2-54-210-192-12.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-192-12.compute-1.amazonaws.com resolving cookbooks for run list:
["myhaproxy"]
ec2-54-210-192-12.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-192-12.compute-1.amazonaws.com   - build-essential
ec2-54-210-192-12.compute-1.amazonaws.com   - cpu
ec2-54-210-192-12.compute-1.amazonaws.com   - haproxy
ec2-54-210-192-12.compute-1.amazonaws.com   - myhaproxy
ec2-54-210-192-12.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-192-12.compute-1.amazonaws.com Converging 9 resources
ec2-54-210-192-12.compute-1.amazonaws.com Recipe: haproxy::install_package
ec2-54-210-192-12.compute-1.amazonaws.com   * yum_package[haproxy] action install (up
to date) ...
```

Roles for Everyone

We will give our nodes a role to better describe them and so we can configure them in a similar manner.

Objective:

- ✓ Give our load balancer node a "load_balancer" Role
- ❑ Give our web nodes a "web" Role

Lab: Define a Web Role

- ❑ Create a role named 'web' that has the run list 'recipe[apache]'
- ❑ Set node1's run list to be "role[web]"
- ❑ Set node3's run list to be "role[web]"

Lab: Create the web.rb File

~/chef-repo/roles/web.rb

```
name 'web'
description 'Web Server'
run_list 'recipe[apache]'
```

Lab: Upload the web.rb File

```
$ knife role from file web.rb
```

```
Updated Role web!
```

Lab: Verify the Role on the Chef Server

```
$ knife role list
```

```
load_balancer
web
```

Lab: Verify Specific Information About the Role

```
$ knife role show web
```

```
chef_type:            role
default_attributes:
description:          Web Server
env_run_lists:
json_class:           Chef::Role
name:                 web
override_attributes:
run_list:             recipe[apache]
```

Lab: Set node1's Run List

```
$ knife node run_list set node1 "role[web]"
```

```
node1:
  run_list: role[web]
```

Lab: Set node3's Run List

```
$ knife node run_list set node3 "role[web]"
```

```
node3:
  run_list: role[web]
```

Lab: Converge All Web Nodes

```
$ knife ssh "role:web" -x USER -P PWD "sudo chef-client"
```

```
ec2-54-175-46-24.compute-1.amazonaws.com  Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list:
["apache"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list:
["apache"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
ec2-54-175-46-24.compute-1.amazonaws.com  Synchronizing Cookbooks:
ec2-54-175-46-24.compute-1.amazonaws.com    - apache
```

Lab: Define a Web Role

- ✓ Create a role named 'web' that has the run list 'recipe[apache]'
- ✓ Set node1's run list to be "role[web]"
- ✓ Set node3's run list to be "role[web]"

Roles for Everyone

We will give our nodes a role to better describe them and so we can configure them in a similar manner.

Objective:

- ✓ Give our load balancer node a "load_balancer" Role
- ✓ Give our web nodes a "web" Role

Discussion

What are the benefits of using roles? What are the drawbacks?

Roles can contain roles. How many of these nested roles would make sense?

CHEF™