Chef – First Recipe

Objectives:

Write a new cookbook

Use Chef resources

Successfully run chef on the node

Create a new Cookbook

Create a cookbook named: webserver

Webserver cookbook

This cookbook should do the following after running:

Create a directory /var/www

Create a file index.html that was created by a template.

Our Goal:

Have chef create and manage a file (index.html) built from the template

http://wiki.opscode.com/display/chef/Resources

Template file

Create a template file in **templates/default** named index.html.erb

```
<pre>
Platform: <%=node['platform']%>
Platform Version: <%=node['platform_version']%>
Default IP Address: <%=node['ipaddress']%>
Fully Qualified Domain Name: <%=node['fqdn']%>
Node's Run List: <%=node.run_list.to_s%>
</pre>
```

Directory resource

```
directory "/var/www" do
    action :create
end
```

Template resource

```
template "/var/www/index.html" do
    source "index.html.erb"
    action :create
end
```

Upload cookbook to server

Upload the cookbook to the chef server, so that nodes can see/consume it.

```
$ knife cookbook upload webserver
```

Modify Node run list

Open the node object:

```
$ knife node edit justin-client-test
```

Here we are adding the role 'webserver' to the node's run list.

A run list can be made up of 'recipes' or 'roles'.

```
{
  "name": "justin-client-test",
  "chef_environment": "_default",
  "normal": {
    "tags": [

    ]
  },
  "run_list": [
      "recipe[webserver]"
  ]
}
```

Node Run list. Cont.

Why is this important?

With out a run list the chef-server has no idea what that node is supposed to do.

Chef-client run

Now its time to run chef-client. There are 2 ways to do this.

1) Ssh to the node and run 'sudo chef-client'

```
sudo chef-client
```

Or:

2) Run the chef-client command from your local workstation

```
knife ssh "recipe:webserver" "sudo chef-server" \
-x centos -P cheftraining -a ipaddress
```

/var/www/index.html

```
<pre>
Platform: centos
Platform Version: 6.3
Defarult IP Address: 65.61.189.110
Fully Qualified Domain Name: bens-client
Node's Run List: role[base], role[webserver]
</pre>
```

Roles

Topics:

- **Roles**
- **Attributes**

Why are Roles Important?

You may have multiple recipes and instead of 1 node type you might have a webserver, a db, etc.

A role is an abstraction over a node type.

Introduction to Roles

Create a directory named: “**roles**” at the same level as the directory named cookbooks:

First Role

Create a role named: webserver.rb

First Role creation:

```
name "webserver"
description "Role for configuring a webserver."
run_list(
  "recipe[webserver]"
)
default_attributes(
)
```

Roles can be in either a ruby file, or a json file.

Upload the role to the server

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

In this example we are using the built in power of knife to read a role from a local file.

Another way to create a role is to run the command:

knife role create webserver

Modify the Node's Run list

Modify the node to look like:

```
{
  "name": "justin-client-test",
  "chef_environment": "_default",
  "normal": {
    "tags": [

    ]
  },
  "run_list": [
      "role[webserver]"
  ]
}
```

Here we are adding the role 'webserver' to the node's run list.

Re-run chef-client

Re-run chef-client

What is the output. What does the file index.html say?

Attributes in cookbooks

Create a file named default.rb in the attributes folder of the webserver cookbook.

Set a default level node attribute:

```
default['webserver']['origin'] = "This value is from the cookbook"
```

Update the template index.html.erb to add the a line:

```
<pre>
Platform: <%=node['platform']%>
Platform Version: <%=node['platform_version']%>
Default IP Address: <%=node['ipaddress']%>
Fully Qualified Domain Name: <%=node['fqdn']%>
Node's Run List: <%=node.run_list.to_s%>

Value of attribute: <%=node['webserver']['origin']%>
</pre>
```

Attribute in Role

Modify the role file to add the same attribute to the default attributes section.

```
name "webserver"
description "Role for configuring a webserver."
run_list(
   "recipe[webserver]"
)
default_attributes(
    "webserver" => {
        "origin" => "This value is from the role"
    }
)
```

Community Cookbooks

Topics

- **How to use community cookbooks**

Some times its necessary to create new cookbooks, but a lot of the time someone else has already done it:

http://community.opscode.com

Rackspace Email currently uses the following community cookbooks:

- Mysql
- Database
- Logrotate
- Iptables
- Yum
- Etc...

Download the community cookbooks

Download the following cookbooks:

chef-client

sudo

apache2

yum

```
$ knife cookbook site download yum
```

Unpackage the cookbook and place it in your cookbooks directory.

```
$ tar xzf yum-0.8.0.tar.gz -C cookbooks/
```

Extending cookbook webserver using community cookbooks

Extend the webserver cookbook to use the apache2 recipe.

Use the **'include_recipe'** command.

```
include_recipe "apache2"

directory "/var/www" do
    owner "apache"
    action :create
end

template "/var/www/index.html" do
    source "index.html.erb"
    owner "apache"
    action :create
end
```

Extending metadata.rb file

When you use another cookbook it is important to update your metadata.rb file to reflect that.

```
maintainer        "YOUR_COMPANY_NAME"
maintainer_email "YOUR_EMAIL"
license           "All rights reserved"
description       "Installs/Configures webserver"
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version           "0.0.1"

depends          "apache2"
```

Upload the webserver cookbook to the chef-server, and run chef-client

View the file

In a web browser go to “http://<client_ip>”

Note: you may need to ssh to the client and shut down the firewall first.

Create a base role

Lets say you want all your servers to have the same starting point? What would you do?

Create a role named “base”

The role should have the following run_list:

```
name "base"
description "Base role applied to all nodes."
run_list(
  "recipe[yum::epel]",
  "recipe[chef-client::delete_validation]"
)

default_attributes(
)
```

Upload the role to the chef server

Update the node's run list

Update the node's run list to make is so that the role “base” is run first.

```
{
  "name": "justin-client-test",
  "chef_environment": "_default",
  "normal": {
    "tags": [

    ]
  },
  "run_list": [
    "role[base]",
    "role[webserver]"
  ]
}
```

Re run chef-client

Tired of having to type the password every time chef runs?

Chef to the rescue.....

More Community Cookbooks

Modify the base role to be:

```
name "base"
description "Base role applied to all nodes."
run_list(
  "recipe[yum::epel]",
  "recipe[chef-client::delete_validation]",
  "recipe[sudo]"
)

default_attributes(
  "authorization" => {
      "sudo" => {
            "users" => ["centos"],
            "groups" => ["adm","sysadmins"],
            "passwordless"=>true
        }
  }
)
```

Run chef

Upload the role to the chef-server and re-run chef-client on the node.

Cookbooks and Data Bags

Topics:

- **Creation and usage of data bags**

Data bags:

Are an arbitrary key value store located on the chef-server, that can be accessed directly by recipes running on nodes.

Why should you use them:

They are a good way to store sensitive data (data base username/passwords) and distribute them across many nodes.

Create a New user on the Node

Download cookbook 'users' and upload it to the chef-server

Updated role 'base' to add cookbook 'users'

```
name "base"
description "Base role applied to all nodes."
run_list(
   "recipe[yum::epel]",
   "recipe[chef-client::delete_validation]",
   "recipe[sudo]",
   "recipe[users::sysadmins]"
)
```

Data bag

Create a directory named data_bags

In the data_bags directory create a file <yourname>.json

```
{
   "id": "jwitrick",
   "groups": ["sysadmin"],
   "uid": 2001,
   "shell": "/bin/bash",
   "comment": "jwitrick",
   "ssh_keys": ""
}
```

Create a data bag item template

Create a data_bag on the chef server

In order to start using data bag items you first need to create the data bag container.

Upload to chef-sever

```
$ knife data bag from file users username.json
```

Here we are uploading the data bag to the chef-server using the 'from file' command.

Another way to accomplish this is by creating it directly on the chef-server using:

knife data bag create users username

Data bags cont.

Run chef-client again.

Now lets login using your new username:

knife ssh "role:base" "sudo chef-client -l debug" -x <your username> -i <path_to_private_key> -a ipaddress

QUESTIONS