CHEF™ Fundamentals

Instructor

franklin@getchef.com

Workstation

http://www.getchef.com/download-chef-client

Server

https://manage.opscode.com

Node

https://use.cloudshare.com/Class/dvn6c

passphrase: learn chef with me

Workstation Setup

Install Chef - http://www.getchef.com/chef/install

Get Chef - http://www.getchef.com

```
$ curl -L http://www.getchef.com/chef/install.sh | sudo bash
$ cd chef-repo
$ ls -al
$ ls .chef
```

chef-repo/.chef/knife.rb

```
current_dir = File.dirname(__FILE__)
log_level                :info
log_location             STDOUT
node_name                "USERNAME"
client_key               "#{current_dir}/USERNAME.pem"
validation_client_name "ORGNAME-validator"
validation_key           "#{current_dir}/ORGNAME-validator.pem"
chef_server_url      "https://api.opscode.com/organizations/ORGNAME"
cache_type               'BasicFile'
cache_options( :path =>"#{ENV['HOME']}/.chef/checksums" )
cookbook_path            ["#{current_dir}/../cookbooks"]
```

```
$ knife --version
$ knife client list
$ knife help list
```

Bonus Exercises

Exercise #1

Situation:

You want to keep your personal machines separate from your training environment.

Tasks:

- Create a second organization in your hosted Chef account called "**<username>-home**".
- Create a new Chef repo directory named "**chef-repo-personal**" and set it up to connect to your new organization.

- Create an "**editor-test**" client in your new personal organization. Run "**knife client list**". Now change back into your training organization's repo and run "**knife client list**" again. What's different?
- View your different organizations at http://manage.opscode.com
- Don't forget to change back into your training repo before we continue.

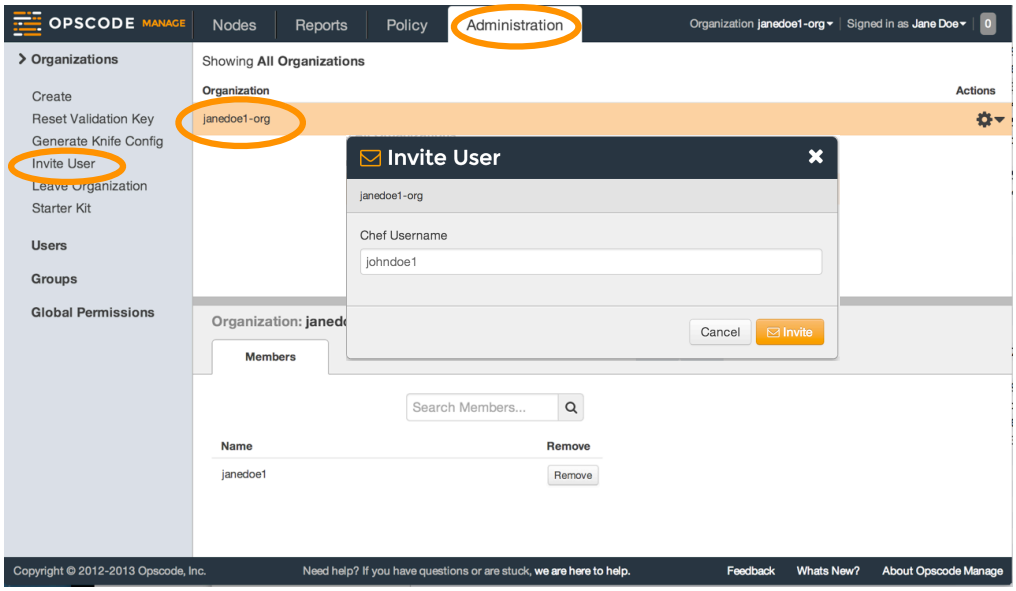
Organization Setup

Click the "Administration" tab,

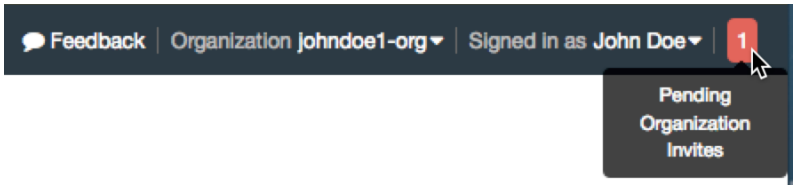
Select the appropriate Organization

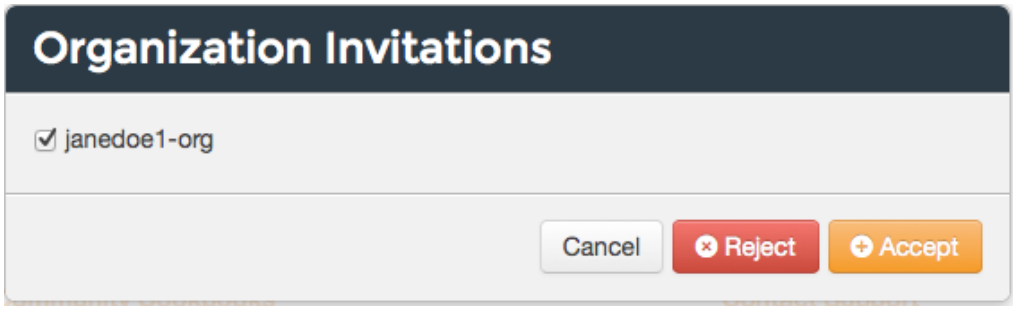
Click "Invite User" from the left menu

Enter your classmate's 'Chef Username' and click Invite

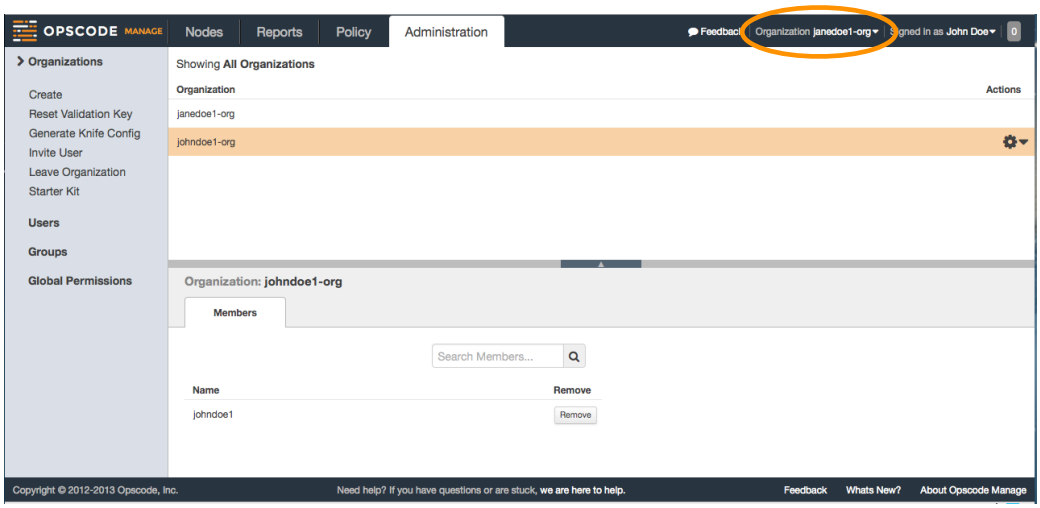


Click the notification, select the Organization and click 'Accept'

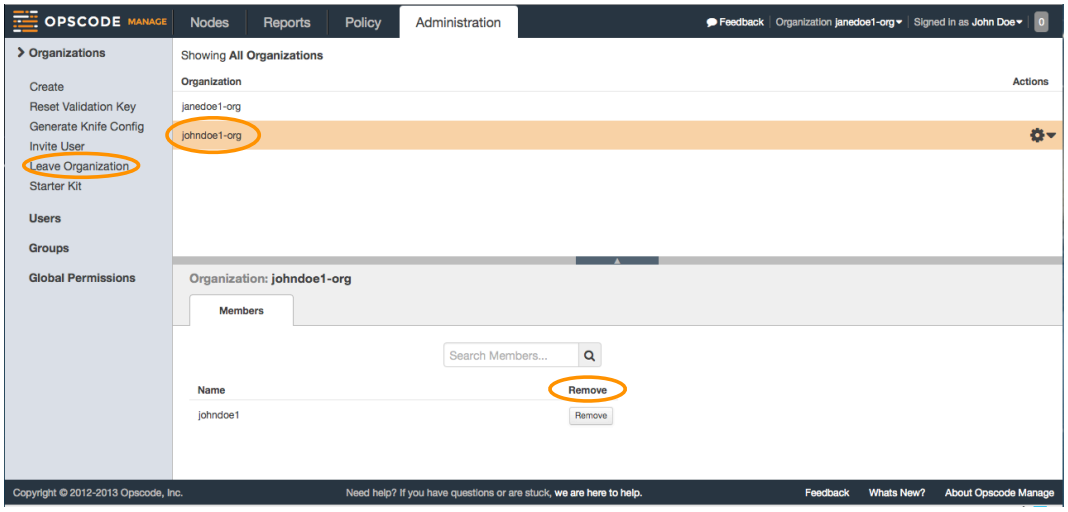




Select your classmate's organization from the drop down list and peruse their org



Now either 'Leave Organization' you've been invited into, or remove your classmate from your organization

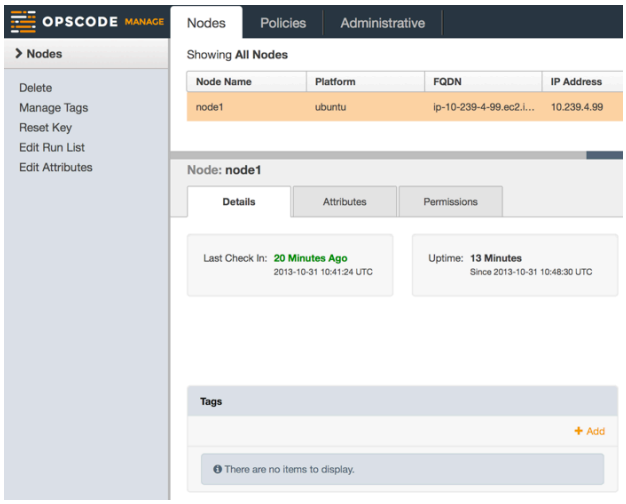


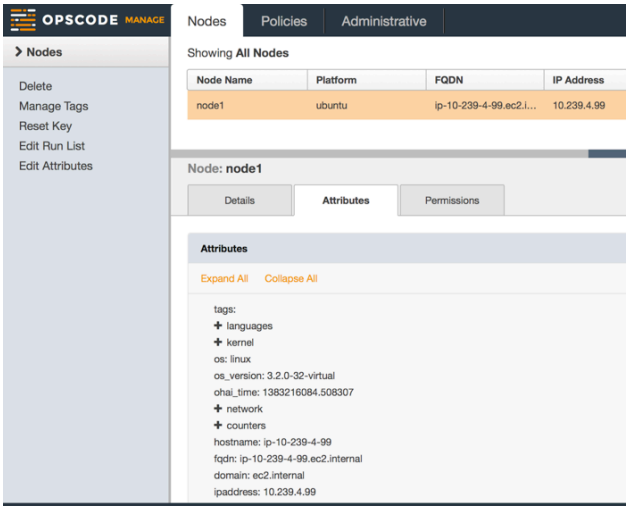
Node Setup

```
$ knife bootstrap <EXTERNAL_ADDRESS> --sudo -x chef -P chef -N "node1"
$ ssh chef@IPADDRESS
chef@node1:~$ ls /etc/chef
chef@node1:~$ which chef-client
chef@node1:~$ cat /etc/chef/client.rb
chef@node1:~$ sudo vi /etc/chef/client.rb
```

Set `log_level` to `:info`

View Node on Chef Server





Bonus Exercises

Exercise #1

Situation:

A junior admin accidentally deleted `client.pem`. Your job is to repair the damage.

Tasks:

- Delete `/etc/chef/client.pem` from your target node and run "**sudo chef-client**". What happens?
- How can you fix the target host so it can communicate with the server again? (*Hint*: Every node also has a client)

Exercise #2

Situation:

Your target node is not authenticating to the server and you're not sure why.

Tasks:

- Set the time on your target to midnight with this command:
  date +%T -s "00:00:00"
- Run "**sudo chef-client**". What happens? Why?
- Correct the time with this command: **ntpdate pool.ntp.org**

Chef Resources and Recipes

Writing an Apache cookbook

Create a cookbook and examine the contents of that cookbook

```
$ knife cookbook create apache

$ ls -la cookbooks/apache
```

Within the apache cookbook compose the default recipe and referenced cookbook file

cookbooks/apache/recipes/default.rb

```
package "httpd" do
  action :install
end

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

cookbook_file "/var/www/html/index.html" do
  source "index.html"
  mode "0644"
  action :create
end
```

cookbooks/apache/files/default/index.html

```
<html>
<body>
  <h1>Hello, world!</h1>
</body>
</html>
```

Upload the cookbook to the chef server and add the default recipe to the node's run_list

```
$ knife cookbook upload apache
$ ls -la cookbooks/apache
$ knife cookbook list
$ knife node run_list add node1 "recipe[apache::default]"
```

Chef Resources and Recipes

Writing an Apache cookbook

Ask the node to check in with the Chef Server

```
chef@centos63:~$ sudo chef-client
```

Bonus Exercises

Take ownership of the cookbook

Cookbooks contain information about the maintainer, support and description. This is contained in the **metadata.rb** file of the cookbook. Update the **metadata.rb** file to include your name and email. Update the description field to briefly describe what the cookbook accomplishes.

cookbooks/apache/metadata.rb

```
name             'apache'
maintainer       'YOUR_COMPANY_NAME'
maintainer_email 'YOUR_EMAIL'
license          'All rights reserved'
description      'Installs/Configures apache'
long_description IO.read(File.join(File.dirname(__FILE__), 'README.md'))
version          '0.1.0'
```

Refining your resource definitions

Resources have actions that they perform by default. When you are using a resource in the default way, the action does not need to be stated. Find the three resources in the documentation at **https://docs.getchef.com**.

	current action	default action
package	_____	_____
service	_____	_____
cookbook_file	_____	_____

If a resource has no parameters and no actions (relying on the default values in all cases) you can exclude the **do...end** block and write the resource type and name. An example:

```
package "httpd"
```

Introducing the Node Object

```
$ knife node list
$ knife client list
$ knife node show node1
chef@node1:~$ sudo ohai | less
$ knife node show node1 -l
$ knife node show node1 -Fj
$ knife node show node1 -a fqdn
$ knife search node "*:*" -a fqdn
```

Node Attributes

cookbooks/apache/attributes/default.rb

```
default["apache"]["indexfile"] = "index1.html"
```

cookbooks/apache/files/default/index1.html

```
<html>
 <body>
  <h1>Hello, world!</h1>
   <h2>This is index1.html</h2>
    <p>We configured this in the attributes file</p>
 </body>
</html>
```

cookbooks/apache/recipes/default.rb

```
cookbook_file "/var/www/html/index.html" do
  source node["apache"]["indexfile"]
  mode "0644"
end
```

```
$ knife cookbook upload apache
chef@node1:~$ sudo chef-client
```

cookbooks/apache/recipes/default.rb

```
node.default["apache"]["indexfile"] = "index2.html"
cookbook_file "/var/www/html/index.html" do
  source node["apache"]["indexfile"]
  mode "0644"
end
```

cookbooks/apache/files/default/index2.html

```
<html>
 <body>
  <h1>Hello, world!</h1>
   <h2>This is index2.html</h2>
    <p>We configured this in the recipe</p>
 </body>
</html>

$ knife cookbook upload apache
chef@node1:~$ sudo chef-client
```

Attributes, Templates, and Cookbook Dependencies

Use knife to create a cookbook called 'motd' (command hidden)

cookbooks/motd/attributes/default.rb

```
default["motd"]["company"] = "Chef"
```

Add template resource to the motd cookbook's default recipe (cookbooks/motd/recipes/default.rb) for /etc/motd based on the source 'motd.erb'. (command hidden)

cookbooks/motd/templates/default/motd.erb

```
This server is property of <%= node["motd"]["company"] %>
 <% if node["pci"]["in_scope"] -%>
 This server is in-scope for PCI compliance
 <% end -%>
```

Use knife upload the 'motd' cookbook (command hidden)

Use knife to create a cookbook called 'pci' (command hidden)

cookbooks/pci/attributes/default.rb

```
default["pci"]["in_scope"] = true
```

Use knife upload the 'pci' cookbook (command hidden)
Use knife add 'recipe[motd]' to node1's run list (command hidden)

```
$ knife node show node1
chef@node1:~$ sudo chef-client
```

cookbooks/motd/metadata.rb

```
maintainer        "YOUR_COMPANY_NAME"
maintainer_email "YOUR_EMAIL"
license           "All rights reserved"
description       "Installs/Configures motd"
long_description IO.read(File.join(File.dirname(__FILE__),
'README.md'))
version           "0.1.0"
depends "pci"
```

Use knife upload the 'motd' cookbook (command hidden)
Rerun 'chef-client' on node1 (command hidden)

```
chef@node1:~$ cat /etc/motd
$ knife search node "pci:*" -a pci
```

cookbooks/pci/attributes/default.rb

```
default["pci"]["in_scope"] = false
```

Use knife upload the 'pci' cookbook (command hidden)

Rerun 'chef-client' on node1 (command hidden)

```
chef@node1:~$ cat /etc/motd
$ knife node show node1 -a pci
```

Bonus Exercises

Exercise #1

Situation:

You need to list the IP addresses of only linux nodes (CentOS or Ubuntu)

Tasks:

- Use the "**knife search**" command to create your list. (HINT: You can use the **-l** flag to get a list of all the attributes that are available to you.)

Exercise #2

Situation:

The pretty MOTD banner is gone!

Tasks:

- Restore the **/etc/motd** banner from a backup. Where might the backup of this file be located? HINT: Look in the chef-client output. Run chef-client again. What happens?

Exercise #3
Situation:
The client wants the hostname of the machine to be automatically included on their homepage.
Tasks:
- Edit your **motd.erb** template, replacing “This server” with the node’s hostname using a node attribute and embedded Ruby.

Template Variables, Notifications, and Controlling Idempotency

cookbooks/apache/metadata.rb

```
maintainer       "YOUR_COMPANY_NAME"
maintainer_email "YOUR_EMAIL"
license          "All rights reserved"
description      "Installs/Configures apache"
long_description IO.read(File.join(File.dirname(__FILE__),
'README.md'))
version          "0.2.0"
```

cookbooks/apache/attributes/default.rb

```
default["apache"]["sites"]["clowns"] = { "port" => 80 }
default["apache"]["sites"]["bears"] = { "port" => 81 }
```

cookbooks/apache/recipes/default.rb

(See https://gist.github.com/6781185)

```
package "httpd" do
  action :install
end

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

execute "mv /etc/httpd/conf.d/welcome.conf
/etc/httpd/conf.d/welcome.conf.disabled" do
  only_if do
    File.exist?("/etc/httpd/conf.d/welcome.conf")
  end
  notifies :restart, "service[httpd]"
end

node["apache"]["sites"].each do |site_name, site_data|
  document_root = "/srv/apache/#{site_name}"

  template "/etc/httpd/conf.d/#{site_name}.conf" do
   source "custom.erb"
   mode "0644"
   variables(:document_root => document_root,:port => site_data["port"])
   notifies :restart, "service[httpd]"
  end

  directory document_root do
   mode "0755"
   recursive true
  end

  template "#{document_root}/index.html" do
   source "index.html.erb"
   mode "0644"
   variables(:site_name => site_name, :port => site_data["port"])
  end
end
```

cookbooks/apache/templates/default/custom.erb
(See https://gist.github.com/8955103)

```
<% if @port != 80 -%>
  Listen <%= @port %>
<% end -%>

<VirtualHost *:<%= @port %>>
  ServerAdmin webmaster@localhost

  DocumentRoot <%= @document_root %>
  <Directory />
     Options FollowSymLinks
     AllowOverride None
  </Directory>
  <Directory <%= @document_root %>
     Options Indexes FollowSymLinks MultiViews
     AllowOverride None
     Order allow,deny
     allow from all
     </Directory>
</VirtualHost>
```

cookbooks/apache/templates/default/index.html.erb
(See https://gist.github.com/2866421)

```
<html>
  <body>
    <h1>Welcome to <%= node["motd"]["company"] %></h1>
    <h2>We love <%= @site_name %></h2>
    <%= node["ipaddress"] %>:<%= @port %>
  </body>
</html>
```

Use knife upload the 'apache' cookbook (command hidden)
Rerun 'chef-client' on node1 (command hidden)

Troubleshoot the failure

Bonus Exercises

Exercise #1

Situation:

Someone borked your Apache cookbook, and now Apache won't start properly.

Tasks:

- Replace your "**custom.erb**" with the contents of this one: https://gist.github.com/scarolan/6091028
- Run "**knife cookbook upload apache**" on your workstation
- Stop apache on the target host with "**sudo service apache2 stop**"
- Attempt to run chef-client again. It will probably fail. Why did it fail? Fix it without manually changing any configurations on the target. You may only use chef-client to get Apache running again.

Exercise #2

Situation:

The marketing people are convinced that ponies are the new hotness.

Tasks:

- Create a new site running on port 83 for ponies. Don't use port 82, we are going to use that for something else later in the training.

Exercise #3 - Advanced

Situation:

The boss wants more pictures on our websites.

Tasks:

- Figure out how to add an image to each of your websites' home pages, using default attributes. You can search Google images and use a remotely hosted file if you wish.
  HINT: HTML formatting for images looks like this:
  <img src="http://www.example.com/path/to/image.jpg">
- Complete solution is here, don't peek unless you are completely stuck! https://gist.github.com/scarolan/6091430
- **NOTE:** The solution file is for Ubuntu systems but the method to get the images working is *identical*.

Search

```
$ knife search node "*:*"
$ knife search node "ipaddress:10.*"
$ knife search node "*:*" -a ipaddress
$ knife search node "ipaddress:10.*" -a ipaddress
$ knife search node "ipaddress:10* AND platform:centos"
$ knife search node "ipaddress:[10.0.* TO 10.2.*]"
```

cookbooks/apache/recipes/ip-logger.rb

```
search("node","platform:centos").each do |server|
  log "The CentOS servers in your organization have the following
  FQDN/IP Addresses:- #{server["fqdn"]}/#{server["ipaddress"]}"
end
```

Use knife upload the 'apache' cookbook (command hidden)
Add the recipe ' apache::ip-logger' to node1's run list (command hidden)
Rerun 'chef-client' on node1 (command hidden)

Remove the recipe ' apache::ip-logger' from node1's run list (command hidden)

Recipe Inclusion, Data Bags, and Search

```
$ mkdir -p data_bags/users
$ knife data_bag create users
```

data_bags/users/bobo.json

```
{
  "id": "bobo",
  "comment": "Bobo T. Clown",
  "uid": 2000,
  "gid": 0,
  "home": "/home/bobo",
  "shell": "/bin/bash"
}
```

```
$ knife data_bag from file users bobo.json
```

Create another user in the users data bag called 'frank' (command hidden)

```
{
  "id": "frank",
  "comment": "Frank Belson",
  "uid": 2001,
  "gid": 0,
  "home": "/home/frank",
  "shell": "/bin/bash"
}
```

Use knife to upload frank's data_bag item(command hidden)

```
$ knife search users "*:*"
$ knife search users "id:bobo" -a shell
```

Create a data_bag called 'groups' (2 commands hidden)

data_bags/groups/clowns.json

```
{
  "id": "clowns",
  "gid": 3000,
  "members": [ "bobo", "frank" ]
}
```

Use knife to upload the 'clowns' data_bag item (command hidden)

Create a cookbook called 'users' (command hidden)

Edit the 'user' cookbook's default recipe and add the following

```
search(:users, "*:*").each do |user_data|
  user user_data["id"] do
    comment user_data["comment"]
    uid user_data["uid"]
    gid user_data["gid"]
    home user_data["home"]
    shell user_data["shell"]
  end
end
include_recipe "users::groups"
```

cookbooks/users/recipes/groups.rb

```
search(:groups, "*:*").each do |group_data|
  group group_data["id"] do
    gid group_data["gid"]
    members group_data["members"]
  end
end
```

Upload the 'users' cookbook (command hidden)
Use knife to add the 'users' cookbook's default receipt to node1's run list (command hidden)
Rerun 'chef-client' on node1 (command hidden)

```
chef@node1:~$ cat /etc/passwd
chef@node1:~$ cat /etc/group
```

Bonus Exercises

Exercise #1

Situation:

Frank and Bobo have user accounts but no home directories. (RHEL/CentOS users can skip this exercise.)

Tasks:

- Use your knowledge and the Chef documentation to figure out how to have user home directories created automatically.

Exercise #2

Situation:

A new junior admin named Zippy started work this morning. You need to give him a restricted account.

Tasks:

- Create a new user account for Zippy, and set his default shell to "**/bin/rbash**" instead of "**/bin/bash**". Set Zippy's uid to 2002, and his gid to 0. Don't forget to add him to the clowns group in your group data bag.

Roles

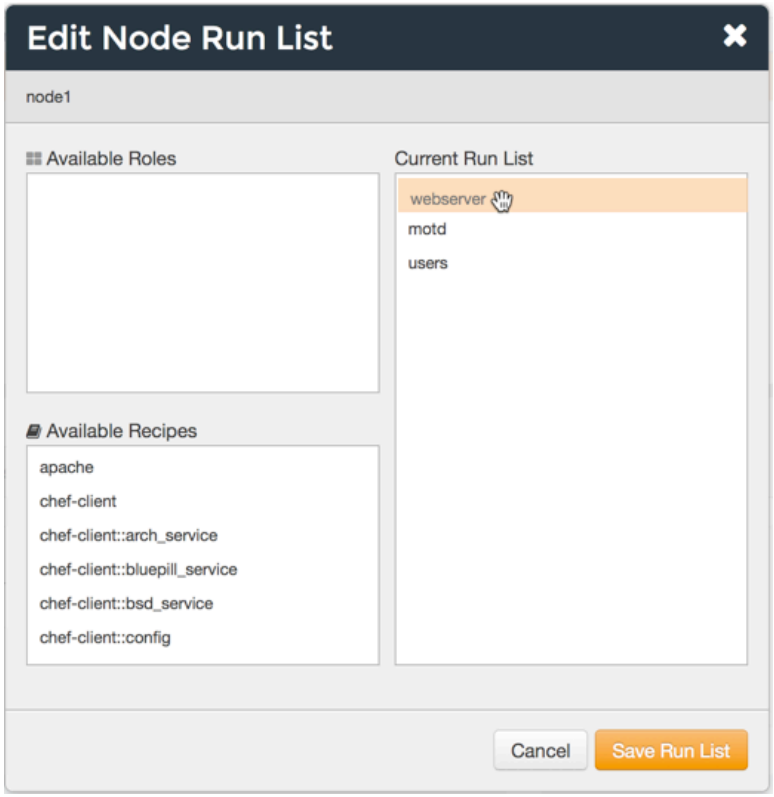
roles/webserver.rb

```
name "webserver"
description "Web Server"
run_list "recipe[apache]"
default_attributes({
  "apache" => {
    "sites" => {
      "admin" => {
        "port" => 8000
      }
    }
  }
})
```

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

Use knife to search for roles that have the apache cookbook's default recipe in its run_list (command hidden)

Replace recipe[apache] with role[webserver] in run list




```
chef@node1:~$ sudo chef-client
$ knife search node "role:webserver" -a apache.sites
```

roles/webserver.rb

```
name "webserver"
description "Web Server"
run_list "recipe[apache]"
default_attributes({
  "apache" => {
    "sites" => {
      "admin" => {
        "port" => 8000
      },
      "bears" => {
        "port" => 8081
      }
    }
  }
})
```

Use knife to upload this webserver role & rerun chef-client (commands hidden)
Use knife to Display the 'apache.sites' attribute on all nodes with webserver role (command hidden)
Edit the 'base' role (command hidden)

```
name "base"
description "Base Server Role"
run_list "recipe[motd]", "recipe[users]"
```

Upload the 'base' role to Chef server (command hidden)
Edit the 'webserver' role (command hidden)

```
name "webserver"
description "Web Server"
run_list "role[base]", "recipe[apache]"
default_attributes({
  "apache" => {
    "sites" => {
      "admin" => {
        "port" => 8000
      },
      "bears" => {
        "port" => 8081
      }
    }
  }
})
```

Upload the 'webserver' role to Chef server (command hidden)
Rerun 'chef-client' on node1 (command hidden)

<u>Bonus Exercises</u>
Exercise #1
Situation:
You need a dedicated NTP server inside the PCI environment to sync the clocks on all your hosts.

Tasks:
- Create a new role called "**ntp_server**". It's run list should be empty for now, we'll use it in a later exercise.

Environments

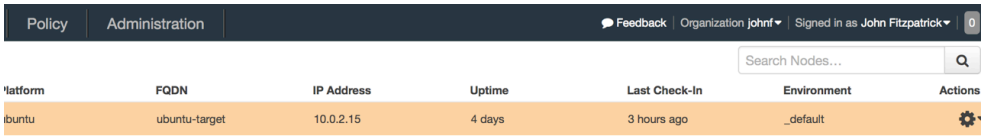
```
$ knife cookbook show apache
$ knife environment list
$ mkdir environments
```

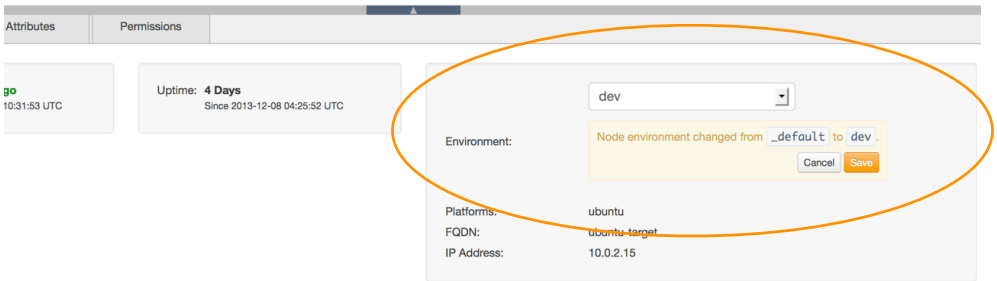
environments/dev.rb

```
name "dev"
description "For developers!"
cookbook "apache", "= 0.2.0"
```

```
$ knife environment from file dev.rb
$ knife environment show dev
```

Use the UI to change your node's environment to "dev"





Rerun 'chef-client' on node1 (command hidden)

environments/production.rb

```
name "production"
description "For Prods!"
cookbook "apache", "= 0.1.0"
override_attributes({
 "pci" => {
   "in_scope" => true
 }
})
```

Use knife to upload this production environment (command hidden)
Use the UI to change your node's environment to "production" (Screenshot hidden)
Rerun 'chef-client' on node1 (command hidden)

<u>Bonus Exercises</u>
Exercise #1
Situation:
Mordac from the security team has insisted that there be a completely separate PCI dev environment.

Tasks:
- Create a new environment called "**dev_pci**". It should be identical to the production environment, but with Apache version 0.2.0 instead of 0.1.0. Try putting your target node in this environment to see what happens.

Using Community Cookbooks

```
$ knife cookbook site search chef-client
$ knife cookbook site show chef-client
$ knife cookbook site download chef-client
$ tar -zxvf chef-client*.tar.gz -C cookbooks/
```

cookbooks/chef-client/recipes/delete_validation.rb

```
unless chef_server?
  file Chef::Config[:validation_key] do
    action :delete
    backup false
    only_if { ::File.exists?(Chef::Config[:client_key]) }
  end
end
```

roles/base.rb

```
name "base"
description "Base Server Role"
run_list "recipe[chef-client::delete_validation]", "recipe[motd]",
"recipe[users]"
```

cookbooks/chef-client/recipes/default.rb

```
include_recipe "chef-client::service"
```

cookbooks/chef-client/recipes/service.rb

```
supported_init_styles = [
  'arch',
  'bluepill',
  'bsd',
  'daemontools',
  'init',
  'launchd',
  'runit',
  'smf',
  'upstart',
  'winsw'
]
init_style = node["chef_client"]["init_style"]

# Services moved to recipes
if supported_init_styles.include? init_style
  include_recipe "chef-client::#{init_style}_service"
else
  log "Could not determine service init style, manual intervention
required to start up the chef-client service."
end
```

Use knife to upload the 'chef-client' cookbook (command hidden)
Use knife to download the 'cron' cookbook (command hidden)
untar the 'cron' cookbook into the cookbooks directory (command hidden)
Use knife to upload the 'cron' cookbook (command hidden)
Use knife to upload the 'chef-client' cookbook (command hidden)
Use knife to download the 'logrotate' cookbook (command hidden)
untar the 'logrotate' cookbook into the cookbooks directory (command hidden)
Use knife to upload the 'logrotate' cookbook (command hidden)
Use knife to upload the 'chef-client' cookbook (command hidden)

Edit the 'base' role (command hidden)

```
name "base"
description "Base Server Role"
run_list "recipe[chef-client::delete_validation]", "recipe[chef-client]", "recipe[motd]", "recipe[users]"
```

Upload the 'base' role to Chef server (command hidden)Rerun 'chef-client' on node1 (command hidden)
Check the 'chef-cllient' service is running (command hidden)Use knife to download the 'ntp' cookbook (command hidden)
untar the 'ntp' cookbook into the cookbooks directory (command hidden)
Use knife to upload the 'ntp' cookbook (command hidden)

Edit the 'base' role (command hidden)

```
name "base"
description "Base Server Role"
run_list "recipe[chef-client::delete_validation]", "recipe[chef-client]", "recipe[ntp]", "recipe[motd]", "recipe[users]"
```

Upload the 'base' role to Chef server (command hidden)

Bonus Exercises

Exercise #1

Situation:

You need to sync all your hosts in with specific NTP servers.

Tasks:

- Without editing any cookbook code or node objects, set the default NTP servers to these North America NTP Pool hosts:

```
server 0.north-america.pool.ntp.org
server 1.north-america.pool.ntp.org
server 2.north-america.pool.ntp.org
server 3.north-america.pool.ntp.org
```

Exercise #2

Situation:

Mordac is back and says you have to have an internal NTP server for PCI hosts.

Tasks:

- Transform your target into an NTP server. *HINT*: Reading is fun!

Just Enough Ruby for Chef

Bonus Exercises

Exercise #1

Situation:

You need to learn more Ruby because you want to be a Chef ninja.

Tasks:

- Sign up for an account on codeacademy.com and start doing the exercises in the Ruby track. http://www.codecademy.com/tracks/ruby