Community Cookbooks

Introducing the chef-client cookbook & wrapper cookbooks

Lesson Objectives

After completing the module, you will be able to

- Find, preview and download cookbooks from the Chef Supermarket
- Use knife to work with the Chef Supermarket site API
- Override community cookbook defaults using wrapper cookbooks
- Upload community cookbooks to Chef Server
- Run chef-client as a service/task

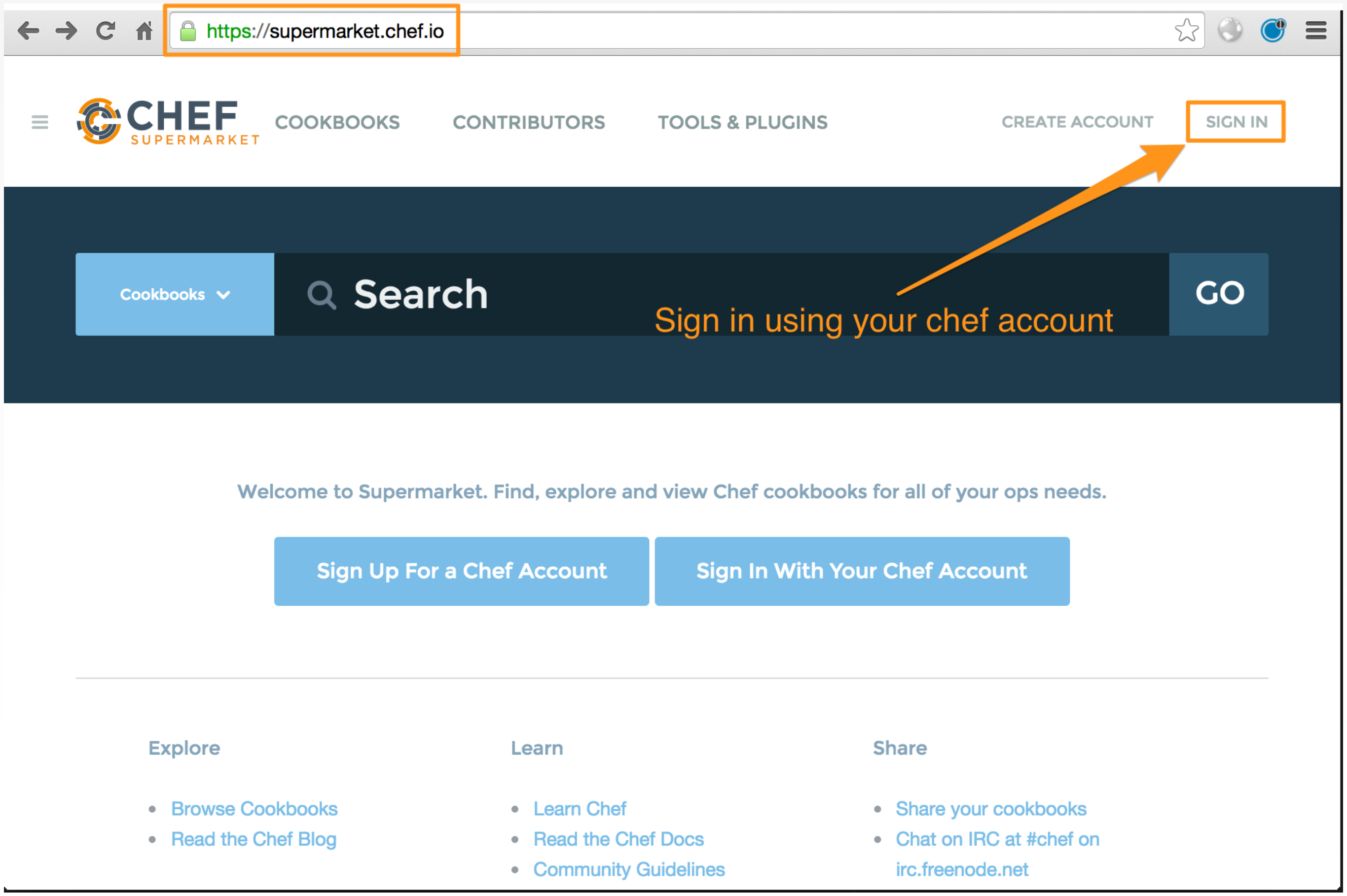
Community Cookbooks

Someone already wrote that cookbook?

Available through the community site called Supermarket/

https://supermarket.chef.io

Log into Supermarket

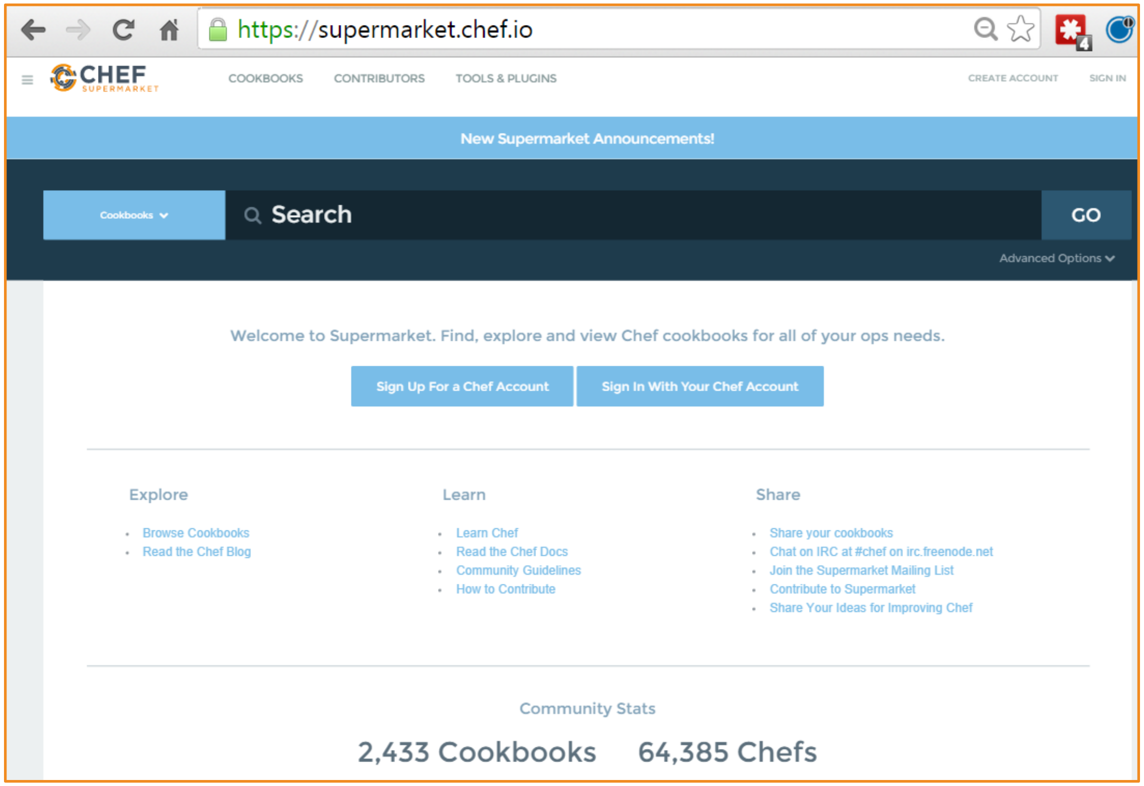


Welcome to Supermarket. Find, explore and view Chef cookbooks for all of your ops needs.

Sign in using your chef account

Community Cookbooks

- Community cookbooks are managed by individuals.
- Chef does not verify or approve cookbooks in the Supermarket.
- Cookbooks may not work for various reasons.
- Still, there are real benefits to community cookbooks.



https://supermarket.chef.io

Welcome to Supermarket. Find, explore and view Chef cookbooks for all of your ops needs.

Explore
- Browse Cookbooks
- Read the Chef Blog

Learn
- Learn Chef
- Read the Chef Docs
- Community Guidelines
- How to Contribute

Share
- Share your cookbooks
- Chat on IRC at #chef on irc.freenode.net
- Join the Supermarket Mailing List
- Contribute to Supermarket
- Share Your Ideas for Improving Chef

Community Stats

2,433 Cookbooks 64,385 Chefs

DISCUSSION

Step back: How how is chef-client configured

- How can I run chef-client as a service or Windows task?
- Where can I configure logging?
- How does chef-client know what Chef Server to connect to?
- How does chef-client authenticate with the Chef Server?
- How do I configure where chef-client caches?

Lab: View chef-client config directory

```
$ knife ssh "*:*" -x USER -P PWD "ls -F /etc/chef"
```

```
ec2... client.pem  client.rb  first-boot.json ohai/  validation.pem

ec2... client.pem  client.rb  first-boot.json ohai/  validation.pem

ec2... client.pem  client.rb  first-boot.json ohai/  validation.pem
```

Lab: View chef-client configuration

```
$ knife ssh "*:*" -x USER -P PWD "cat /etc/chef/client.rb"
```

```
ec2...  log_location     STDOUT
ec2...  chef_server_url  "https://api.chef.io/organizations/jaf-291015"
ec2...  validation_client_name "ORG-validator"
ec2...  node_name "node3"
ec2...
ec2...  log_location     STDOUT
ec2...  chef_server_url  "https://api.chef.io/organizations/jaf-291015"
ec2...  validation_client_name "ORG-validator"
ec2...  node_name "node1"
ec2...
ec2...  log_location     STDOUT
ec2...  chef_server_url  "https://api.chef.io/organizations/jaf-291015"
ec2...  validation_client_name "ORG-validator"
ec2...  node_name "node2"
```

LAB

Introducing chef-client cookbook

The Chef-client cookbook allows you to manage and configure chef-client as a service on Linux based nodes, or as a Task on Windows nodes, configure logging, caching, etc

Bootstrapping installs chef-client executable
The chef-client cookbook is used to configure chef-client

Introducing chef-client cookbook

We will use the chef-client community cookbook to configured chef-client on each of our nodes to run periodically

Objective:

❑ Download chef-client cookbook & upload to Chef Server

❑ Configure each of our nodes to run chef-client as a service

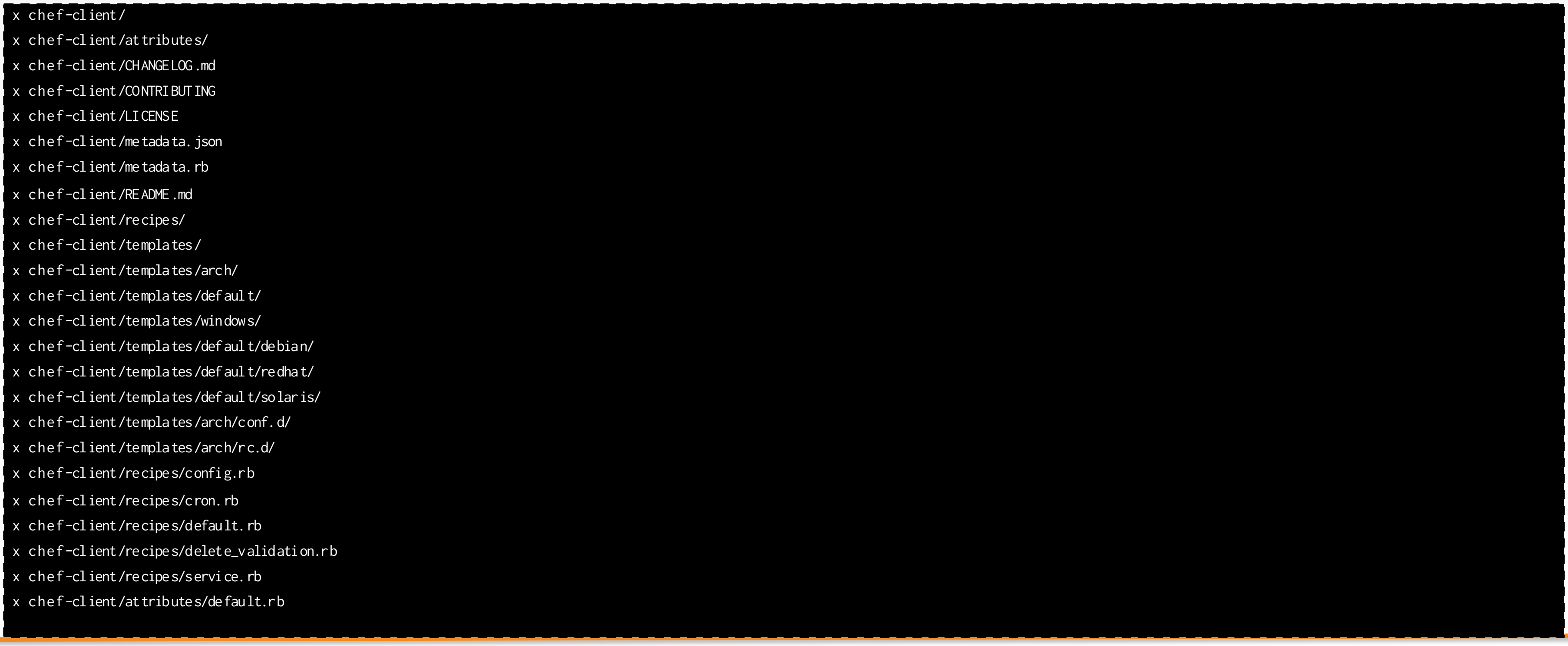
Lab: Download the chef-client cookbook

```
$ knife cookbook site download chef-client
```

```
Downloading chef-client from the cookbooks site at version 4.3.1 to /Users/.../chef-
repo/chef-client-4.3.1.tar.gz
Cookbook saved: /Users/.../chef-repo/chef-client-4.3.1.tar.gz
```

Lab: Extract chef-client cookbook tarball

```
$ tar -zxvf chef-client*.tar.gz -C cookbooks/
```



```
x chef-client/
x chef-client/attributes/
x chef-client/CHANGELOG.md
x chef-client/CONTRIBUTING
x chef-client/LICENSE
x chef-client/metadata.json
x chef-client/metadata.rb
x chef-client/README.md
x chef-client/recipes/
x chef-client/templates/
x chef-client/templates/arch/
x chef-client/templates/default/
x chef-client/templates/windows/
x chef-client/templates/default/debian/
x chef-client/templates/default/redhat/
x chef-client/templates/default/solaris/
x chef-client/templates/arch/conf.d/
x chef-client/templates/arch/rc.d/
x chef-client/recipes/config.rb
x chef-client/recipes/cron.rb
x chef-client/recipes/default.rb
x chef-client/recipes/delete_validation.rb
x chef-client/recipes/service.rb
x chef-client/attributes/default.rb
```

DISCUSSION

Examining the chef-client cookbook

We're going to use one recipe on our node from the chef-client cookbook.

`chef-client::service`

(via `chef-client::default`)

View the chef-client::default recipe

cookbooks/chef-client/recipes/default.rb

```
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

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

View the chef-client::service recipe

cookbooks/chef-client/recipes/service.rb

- The recipe supports a number of **service** providers and styles.
- It works on a lot of **platforms**.
- Everything is controllable through **attributes**.

```
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
```

DISCUSSION

Issue!

We cannot use berks with the cookbook downloaded using knife

It has no metadata.rb!

Stove refuses to upload metadata.rb because it's vulnerable to fairly serious abuse

Lab: Download & unpack chef-client cookbook

```
$ cd chef-repo
$ wget https://github.com/opscode-cookbooks/chef-
client/archive/master.zip
$ unzip master.zip -d cookbooks
$ mv cookbooks/chef-client-master/ cookbooks/chef-client/
```

So lets download cookbook another way

Or manually create metadata file from here

https://github.com/chef-cookbooks/chef-client/blob/master/metadata.rb

Lab: Lets view the dependencies

```
$ cd cookbooks/chef-client
$ cat Berksfile
```

```
source 'https://supermarket.chef.io'

metadata

group :integration do
  cookbook 'runit'
  cookbook 'apt'
  cookbook 'chef-server', '>= 3.0.0'
end
```

Lab: Install the Dependencies

```
$ berks install
```

```
Resolving cookbook dependencies...
Fetching 'chef-client' from source at .
Fetching cookbook index from https://supermarket.chef.io...
Installing apt (2.9.2) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
Installing apt-chef (0.2.0) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
Installing chef-ingredient (0.13.1) from https://supermarket.chef.io ([opscode]
https://supermarket.chef.io:443/api/v1)
Using chef-client (4.3.1) from source at .
Installing chef-server (4.1.0) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
Using chef_handler (1.2.0)
Using logrotate (1.9.2)
Installing cron (1.7.0) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
Using packagecloud (0.1.0)
Using runit (1.7.4)
Installing windows (1.38.2) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
Installing yum (3.8.2) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
Installing yum-chef (0.2.1) from https://supermarket.chef.io ([opscode] https://supermarket.chef.io:443/api/v1)
```

Whats in the Berksfile.lock?

```
$ cat Berksfile.lock
```

```
DEPENDENCIES
  apt
  chef-client
    path: .
    metadata: true
  chef-server (>= 3.0.0)
  runit

GRAPH
  apt (2.9.2)
  apt-chef (0.2.0)
    apt (>= 0.0.0)
  chef-client (4.3.1)
    cron (>= 1.2.0)
    logrotate (>= 1.2.0)
...
```

View the Berkshelf in your home directory

```
$ ls -lt .berkshelf/cookbooks/
```

```
drwxr-xr-x  12 YOU staff  408  2 Dec 15:46 chef_handler-1.1.6
drwxr-xr-x  12 YOU staff  408  2 Dec 15:46 cron-1.6.1
drwxr-xr-x  21 YOU staff  714  2 Dec 15:46 logrotate-1.8.0
drwxr-xr-x  15 YOU staff  510  2 Dec 15:46 windows-1.36.6
drwxr-xr-x   9 YOU staff  306 28 Nov 15:22 build-essential-2.1.3
drwxr-xr-x  25 YOU staff  850  6 Nov 16:27 build-essential-2.1.2
drwxr-xr-x  11 YOU staff  374  6 Nov 16:27 cpu-0.2.0
```

Upload cookbooks to Chef Server

```
$ berks upload
```

```
Uploaded apt (2.9.2) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded apt-chef (0.2.2) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded chef-client (4.3.2) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded chef-ingredient (0.11.3) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded chef-server (4.1.0) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded chef_handler (1.2.0) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded cron (1.7.3) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded logrotate (1.9.2) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded packagecloud (0.1.1) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded runit (1.7.6) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded windows (1.39.0) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded yum (3.8.2) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded yum-chef (0.2.2) to: 'https://api.chef.io:443/organizations/ORG'
```

View cookbooks on Chef Server

```
$ knife cookbook list
```

```
apache          0.3.0
apt             2.9.2
apt-chef        0.2.0
chef-client     4.3.1
chef_handler    1.2.0
cron            1.7.0
haproxy         0.2.0
logrotate       1.9.2
starter         1.0.0
windows         1.38.2
workstation     0.2.1
...
```

Introducing chef-client cookbook

We will use the chef-client community cookbook to configured chef-client on each of our nodes to run periodically

Objective:

✓ Download chef-client cookbook & upload to Chef Server

❑ Configure each of our nodes to run chef-client as a service

DISCUSSION

chef-client as a service

We will add the chef-client default recipe to the roles for each node

Lab: Add the recipe to the file web.rb

```
chef-repo/roles/web.rb
```

```
name 'web'
description 'Web Server'
run_list 'recipe[chef-client]', 'recipe[apache]'
default_attributes({
  "apache" => {
        "port" => 8181
      }
})
```

Lab: Add the recipe to the file proxy.rb

```
chef-repo/roles/loadbalancer.rb
```

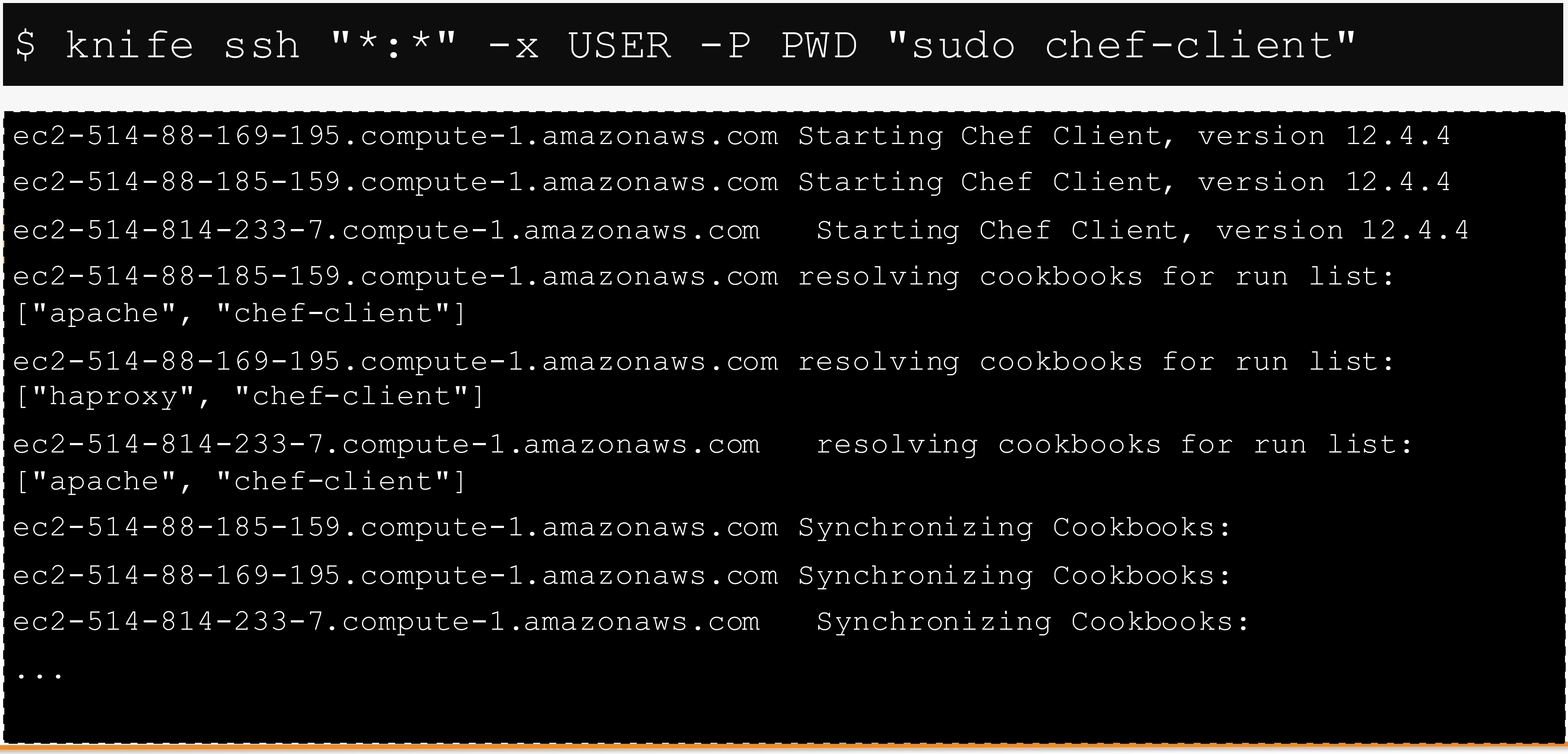
```
name 'loadbalancer'
description 'loadbalancer'
run_list 'recipe[chef-client]', 'recipe[haproxy]'
default_attributes({
  "apache" => {
        "port" => 8181
      }
})
```

Lab: Upload the roles files

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

```
Updated Role web!
Updated Role loadbalancer!
```

Lab: Converge All Nodes



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

```
ec2-514-88-169-195.compute-1.amazonaws.com Starting Chef Client, version 12.4.4
ec2-514-88-185-159.compute-1.amazonaws.com Starting Chef Client, version 12.4.4
ec2-514-814-233-7.compute-1.amazonaws.com   Starting Chef Client, version 12.4.4
ec2-514-88-185-159.compute-1.amazonaws.com resolving cookbooks for run list:
["apache", "chef-client"]
ec2-514-88-169-195.compute-1.amazonaws.com resolving cookbooks for run list:
["haproxy", "chef-client"]
ec2-514-814-233-7.compute-1.amazonaws.com   resolving cookbooks for run list:
["apache", "chef-client"]
ec2-514-88-185-159.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-514-88-169-195.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-514-814-233-7.compute-1.amazonaws.com   Synchronizing Cookbooks:
...
```

Lab: Verify chef-client is running

```
$ knife ssh "*:*" -x USER -P PWD "ps awux | grep chef-
client"
```

```
ec2-514-88-185-159.compute-1.amazonaws.com root      10369  0.0 10.1 266928 61116 ?
Sl   12:54   0:00 /opt/chefdk/embedded/bin/ruby /usr/bin/chef-client -d -c
/etc/chef/client.rb -P /var/run/chef/client.pid -i 1800 -s 300
...
ec2-514-88-169-195.compute-1.amazonaws.com root      14922  0.0 10.1 267020 61092 ?
Sl   12:54   0:00 /opt/chefdk/embedded/bin/ruby /usr/bin/chef-client -d -c
/etc/chef/client.rb -P /var/run/chef/client.pid -i 1800 -s 300
...
ec2-514-814-233-7.compute-1.amazonaws.com   root      10202  0.0 10.1 267036 61096 ?
Sl   12:54   0:00 /opt/chefdk/embedded/bin/ruby /usr/bin/chef-client -d -c
/etc/chef/client.rb -P /var/run/chef/client.pid -i 1800 -s 300
...
```

Introducing chef-client cookbook

We will use the chef-client community cookbook to configured chef-client on each of our nodes to run periodically

Objective:

- ✓ Download chef-client cookbook & upload to Chef Server
- ✓ Configure each of our nodes to run chef-client as a service

DISCUSSION

Change default settings

Wait!

There has just been a mandate that every node in the infrastructure must run chef-client every 5 minutes

Further this must be the new default setting for anyone downloading the chef-client cookbook

DISCUSSION

Well-written cookbooks change behavior based on attributes

- Ideally, you don't have to modify the contents of a cookbook to use it for your specific use case
- Look at the attributes for things you can override through roles to affect behavior of the cookbook
- Of course, well written cookbooks have sane defaults, and a README to describe all this

Setting the chef-client run interval

```
...
default['chef_client']['log_file']    = 'client.log'
default['chef_client']['interval']    = '1800'
default['chef_client']['interval']    = '300'
default['chef_client']['splay']       = '300'
default['chef_client']['conf_dir']    = '/etc/chef'
default['chef_client']['bin']         = '/usr/bin/chef-client'
...
```

We need to change the value of the attribute `default['chef_client']['interval']` from to `1800` (seconds) to `300`

DISCUSSION

Wrapper Cookbooks

Don't use forked community cookbooks in production, or you will miss out on upstream changes, and will have to rebase

Instead use **wrapper cookbooks** to wrap upstream cookbooks and change their behavior without forking

Introducing wrapper cookbooks

We will create a wrapper cookbook to change the default chef-client converge interval to every 5 minutes. If necessary this could still be overwritten via a role

Objective:

❑ Create the wrapper cookbook

❑ Set the default run interval to 5 minutes

❑ Configure each of our nodes to run the default recipe from the wrapper cookbook

Lab: Create my-chef-client wrapper cookbook

```
$ cd chef-repo
$ chef generate cookbook cookbooks/my-chef-client
```

```
Compiling Cookbooks...
Recipe: code_generator::cookbook
  * directory[C:/Users/YOU/chef-repo/cookbooks/my-chef-client] action create
    - create new directory C:/Users/YOU/chef-repo/cookbooks/my-chef-client
  * template[C:/Users/YOU/chef-repo/cookbooks/my-chef-client/metadata.rb] action create_if_missing
    - create new file C:/Users/YOU/chef-repo/cookbooks/my-chef-client/metadata.rb
    - update content in file C:/Users/YOU/chef-repo/cookbooks/my-chef-client/metadata.rb from none to 899276
    (diff output suppressed by config)
  * template[C:/Users/YOU/chef-repo/cookbooks/my-chef-client/README.md] action create_if_missing
...
```

Introducing wrapper cookbooks

We will create a wrapper cookbook to change the default chef-client converge interval to every 5 minutes. If necessary this could still be overwritten via a role

Objective:

✓ Create the wrapper cookbook

❑ Set the default run interval to 5 minutes

❑ Configure each of our nodes to run the default recipe from the wrapper cookbook

Lab: Edit my-chef-client default recipe

`cookbooks/my-chef-client/recipes/default.rb`

```
#
# Cookbook Name:: my-chef-client
# Recipe:: default
#
# Copyright (c) 2015 The Authors, All Rights Reserved.

node.default['chef_client']['interval'] = '300'

include_recipe 'chef-client::default'
```

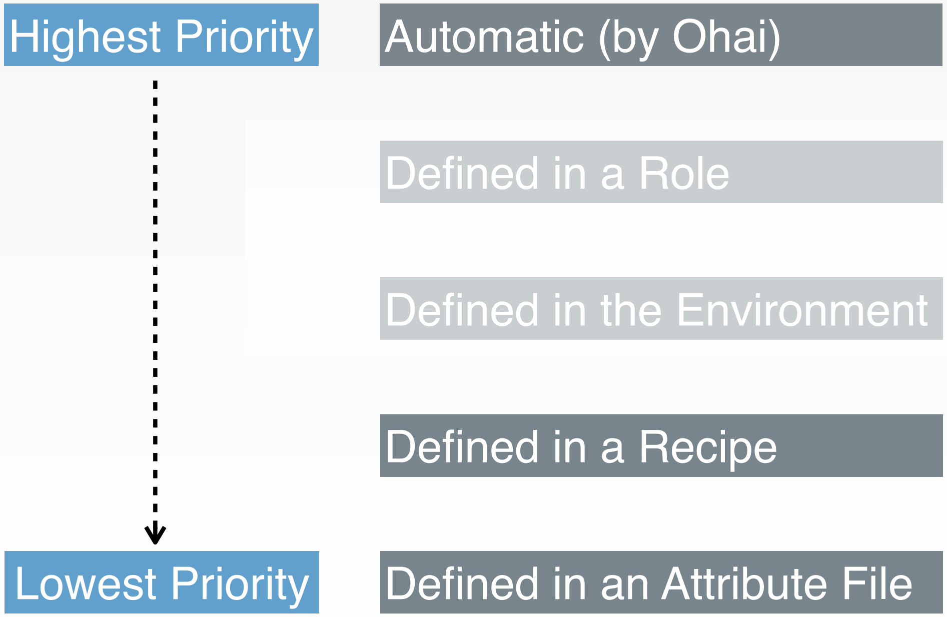
The recipe just sets the attribute value & calls the recipe `chef-client::default`

DISCUSSION

Precedence

Attributes set in a recipe trumps the value set in an attributes file!

Default Attribute Precedence



Highest Priority

Automatic (by Ohai)

Defined in a Role

Defined in the Environment

Defined in a Recipe

Lowest Priority

Defined in an Attribute File

Please note this is a simplified diagram, and the precedence shown can be overridden

Lab: Update my-chef-client metadata.rb file

cookbooks/my-chef-client/metadata.rb

```
name             'my-chef-client'
maintainer       'The Authors'
maintainer_email 'you@example.com'
license          'all_rights'
description      'Installs/Configures my-chef-client'
long_description 'Installs/Configures my-chef-client'
version          '0.1.0'

depends   'chef-client'
```

Lab: Lets view the dependencies

```
$ cd cookbooks/my-chef-client
```

Lab: Install the Dependencies

```
$ cd cookbooks/my-chef-client
$ berks install
```

```
Resolving cookbook dependencies...
Fetching 'my-chef-client' from source at .
Fetching cookbook index from https://supermarket.chef.io...
Installing chef-client (4.3.1) from https://supermarket.chef.io ([opscode]
https://supermarket.chef.io:443/api/v1)
Using cron (1.7.0)
Using chef_handler (1.2.0)
Using logrotate (1.9.2)
Using my-chef-client (0.1.0) from source at .
Using windows (1.38.2)
```

Upload cookbooks to Chef Server

```
$ berks upload
```

```
Skipping chef-client (4.3.1) (frozen)
Uploaded chef_handler (1.2.0) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded cron (1.7.0) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded logrotate (1.9.2) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded my-chef-client (0.1.0) to: 'https://api.chef.io:443/organizations/ORG'
Uploaded windows (1.38.2) to: 'https://api.chef.io:443/organizations/ORG'
```

Introducing wrapper cookbooks

We will create a wrapper cookbook to change the default chef-client converge interval to every 5 minutes. If necessary this could still be overwritten via a role

Objective:

✓ Create the wrapper cookbook

✓ Set the default run interval to 5 minutes

□ Configure each of our nodes to run the default recipe from the wrapper cookbook

Lab: Add the recipe to the file web.rb

```
chef-repo/roles/web.rb
```

```
name 'web'
description 'Web Server'
run_list 'recipe[chef-client]', 'recipe[apache]'
run_list 'recipe[my-chef-client]', 'recipe[apache]'
default_attributes({
  "apache" => {
          "port" => 8181
        }
})
```

Lab: Add the recipe to the file proxy.rb

chef-repo/roles/loadbalancer.rb

```
name 'loadbalancer'
description 'loadbalancer'
run_list 'recipe[chef-client]', 'recipe[haproxy]'
run_list 'recipe[my-chef-client]', 'recipe[haproxy]'
default_attributes({
  "apache" => {
        "port" => 8181
      }
})
```

Lab: Upload the roles files

```
$ cd chef-repo
$ knife role from file web.rb loadbalancer.rb
```

```
Updated Role web!
Updated Role loadbalancer!
```

Lab: Converge All Nodes

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

```
...
ec2-514-88-169-195.compute-1.amazonaws.com   * service[chef-client] action restart
ec2-514-88-169-195.compute-1.amazonaws.com     - restart service service[chef-
client]
ec2-514-814-233-7.compute-1.amazonaws.com
ec2-514-814-233-7.compute-1.amazonaws.com   Running handlers:
ec2-514-814-233-7.compute-1.amazonaws.com   Running handlers complete
ec2-514-814-233-7.compute-1.amazonaws.com   Chef Client finished, 2/15 resources
updated in 14.158836577 seconds
ec2-514-88-169-195.compute-1.amazonaws.com
ec2-514-88-169-195.compute-1.amazonaws.com Running handlers:
ec2-514-88-169-195.compute-1.amazonaws.com Running handlers complete
ec2-514-88-169-195.compute-1.amazonaws.com Chef Client finished, 2/14 resources
updated in 14.179414969 seconds
```

Lab: Verify chef-client is running

```
$ knife ssh "*:*" -x USER -P PWD "ps awux | grep chef-
client|"
```

```
ec2-514-88-169-195.compute-1.amazonaws.com root     15626  0.0 10.1 266936 61120 ?
Sl   13:21   0:00 /opt/chefdk/embedded/bin/ruby /usr/bin/chef-client -d -c
/etc/chef/client.rb -P /var/run/chef/client.pid -i 300 -s 300
...
ec2-514-88-185-159.compute-1.amazonaws.com root     11209  0.0 10.0 266904 61016 ?
Sl   13:21   0:00 /opt/chefdk/embedded/bin/ruby /usr/bin/chef-client -d -c
/etc/chef/client.rb -P /var/run/chef/client.pid -i 300 -s 300
...
ec2-514-814-233-7.compute-1.amazonaws.com   root     10906  0.0 10.1 267016 61096 ?
Sl   13:21   0:00 /opt/chefdk/embedded/bin/ruby /usr/bin/chef-client -d -c
/etc/chef/client.rb -P /var/run/chef/client.pid -i 300 -s 300
...
```

DISCUSSION

Using Community Cookbooks

Chef Community Cookbooks, can be used as is But in most cases you will want to use them as a reference as you write your own

Don't use forked community cookbooks in production, or you will miss out on upstream changes, and will have to rebase

Discussion

What are the benefits and drawbacks of the Chef Supermarket?

Is your team able to leverage community cookbooks? Is the team able to contribute to community cookbooks?

Why do you use a wrapper cookbook? When might you decide to not wrap the cookbook?

DISCUSSION

Q&A

What questions can we help you answer?

- Chef Supermarket
- Wrapper Cookbooks
- chef-client

CHEF™