Community Cookbooks

Find, Explore and View Chef Cookbooks

1L

CHEF

Objectives

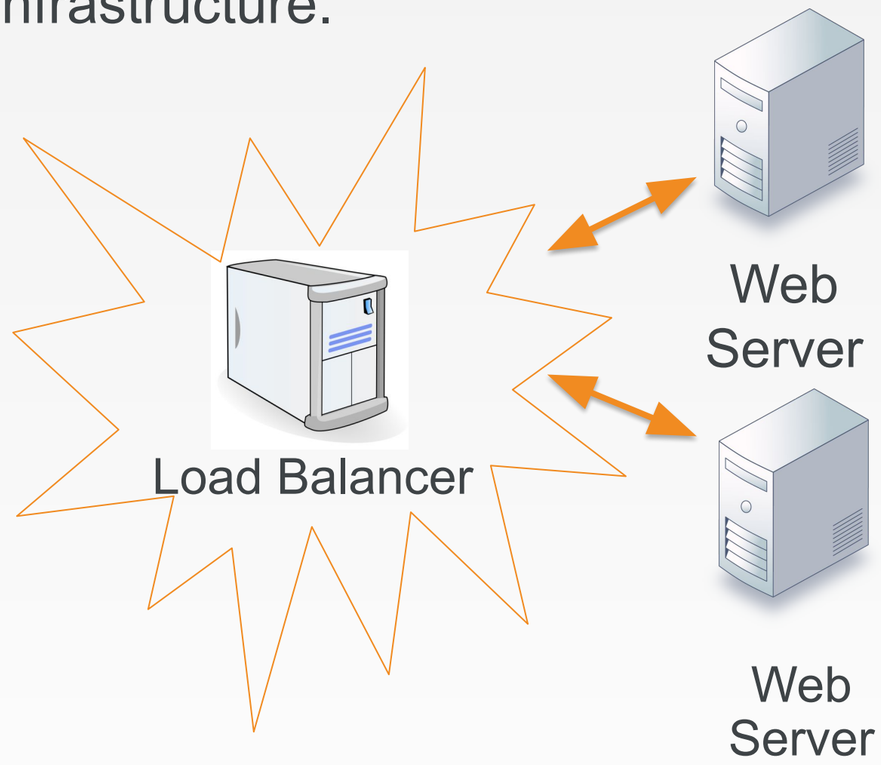
After completing this module, you should be able to

- Find cookbooks on the Chef Supermarket
- Create a wrapper cookbook for a community cookbook
- Utilize Custom Resources
- Create and upload a Policyfile to Chef Infra Server
- Bootstrap a new node that runs the Policyfile's cookbook
- Test the load balancer

Load Balancer

Adding a load balancer will allow us to better grow our infrastructure.

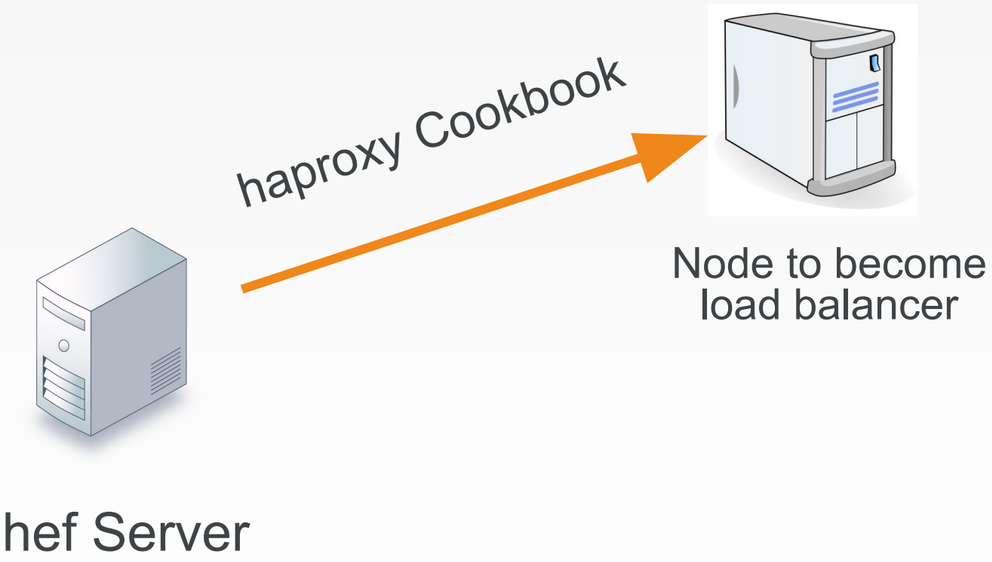
Receives requests and relays them to other systems.



Load Balancer

Work that needs to be accomplished to setup a load balancer within our infrastructure:

- Write a haproxy (load balancer) cookbook.
- We will need to establish a new node within our organization to which we apply that cookbook.



Community Cookbooks

Someone already wrote that cookbook?

Available through the community site called the Chef Supermarket

https://supermarket.chef.io

Types of Community Cookbooks

Chef Community Cookbooks fall into two broad categories: **Resource Cookbooks** and **Recipe Cookbooks.**

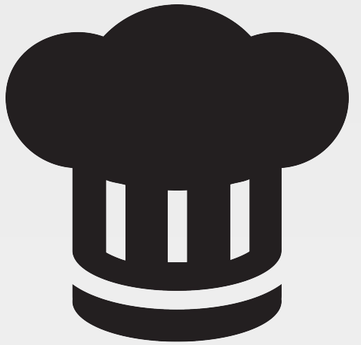
There is no strict naming convention, and the patterns can be mixed. Generally Resource Cookbooks provide extensions for Chef Infra by defining new Chef Resources, and Recipe Cookbooks use recipes to declare how a system should be configured. The difference is in what the end user uses in their wrapper cookbook: a custom resource or a recipe.

Types of Community Cookbooks

So far this class has focused on writing Recipe Cookbooks. We have written Recipes and declared Resources within those Recipes to specify how a system should be configured.

But you can extend Chef Infra by writing your own Resources! Now we will examine a Community Cookbook that provides an extension to Chef Infra by defining a new type of resource we can use inside of our Recipes.

EXERCISE



Group Lab: Load Balancer

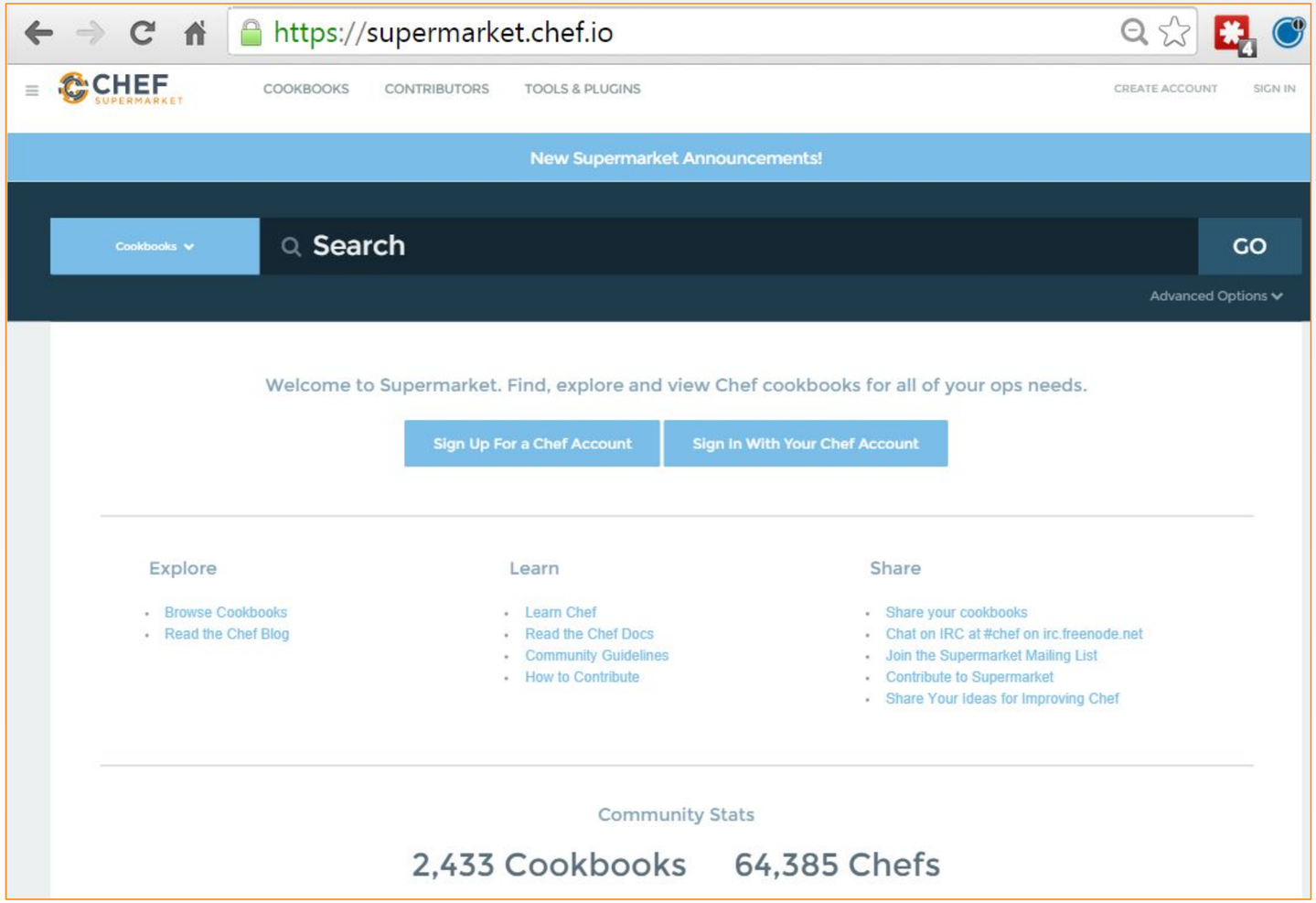
Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ❑ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ❑ Configure the load balancer to send traffic to the iis_web and apache_web nodes
- ❑ Create myhaproxy Policyfile
- ❑ Upload myhaproxy Policyfile.lock to the Chef Infra Server (uploads cookbooks)
- ❑ Bootstrap a new node that runs the myhaproxy (load balancer) cookbook

GL: Community Cookbooks

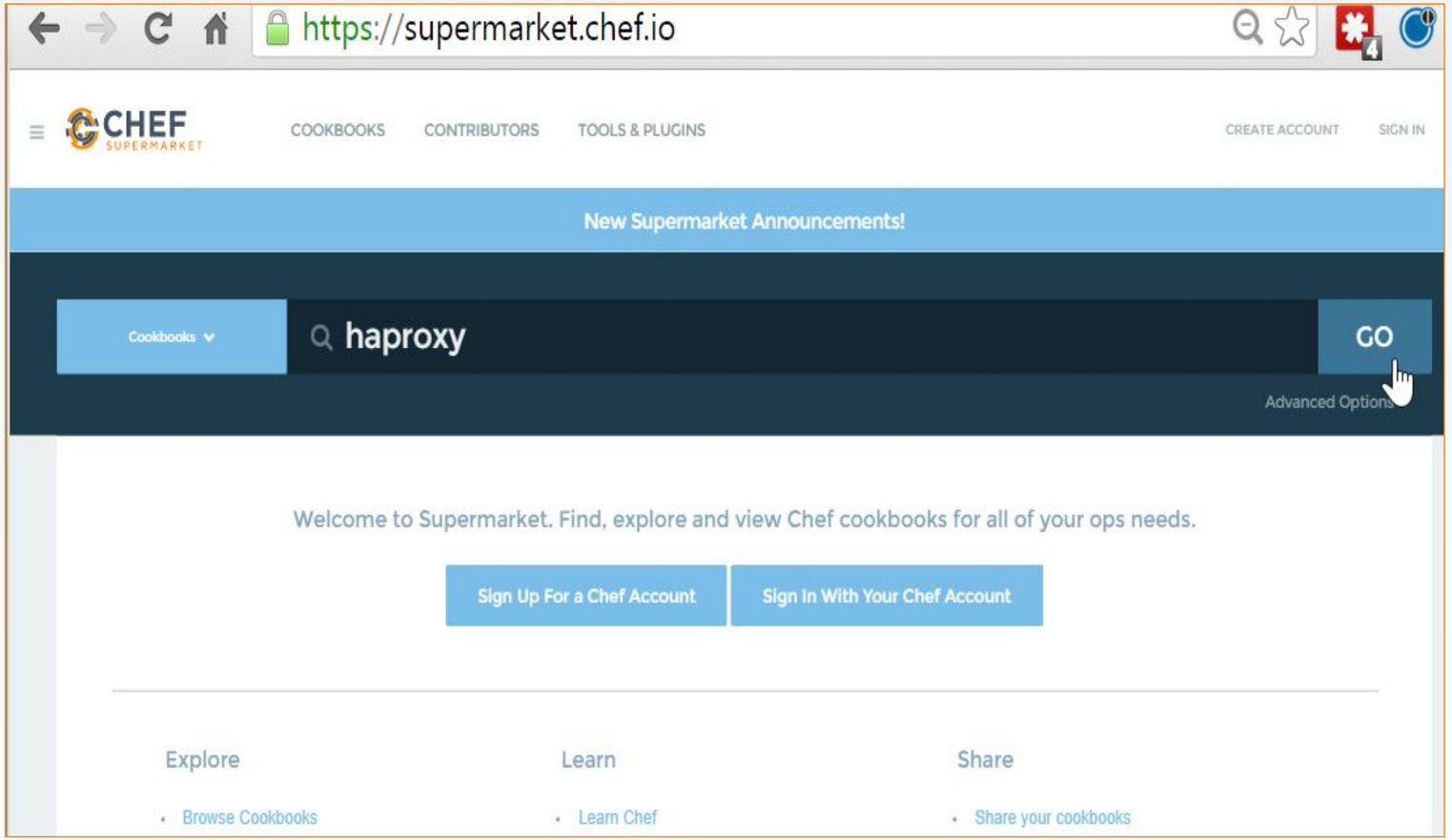
- Community cookbooks are managed by individuals.
- Chef does not verify or approve cookbooks in the Supermarket.
- Cookbooks may not work for various reasons.
- Still, there are real benefits to community cookbooks.



GL: Searching in the Supermarket

STEPS

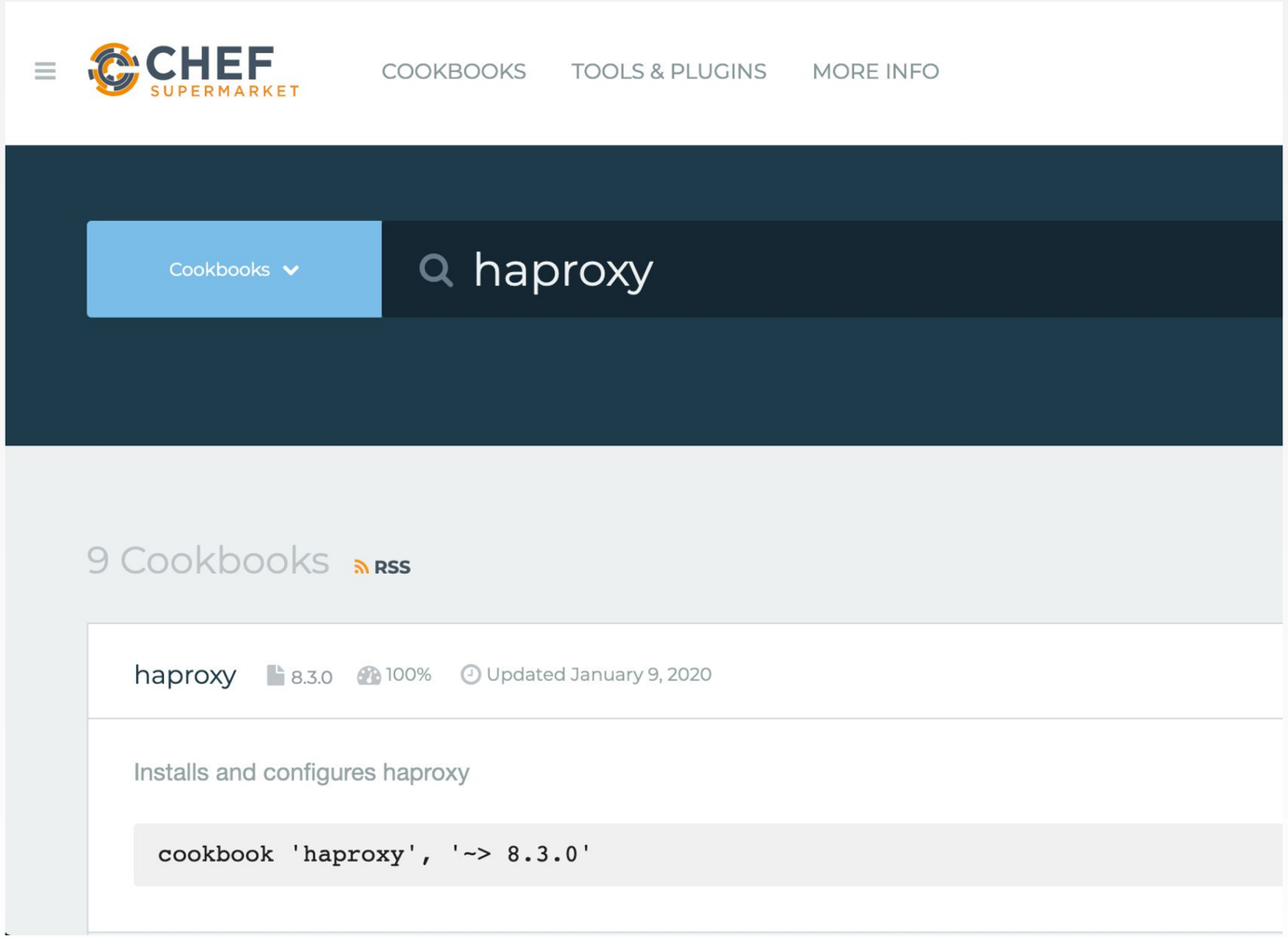
1. Visit **supermarket.chef.io**
2. Select the search field and type in **haproxy** in the search field. Then click the **GO** button.
3. Click the resulting **haproxy** link.



GL: Searching in the Supermarket

STEPS

1. Visit **supermarket.chef.io**
2. Select the search field and type in **haproxy** in the search field. Then click the **GO** button.
3. Click the resulting **haproxy** link.

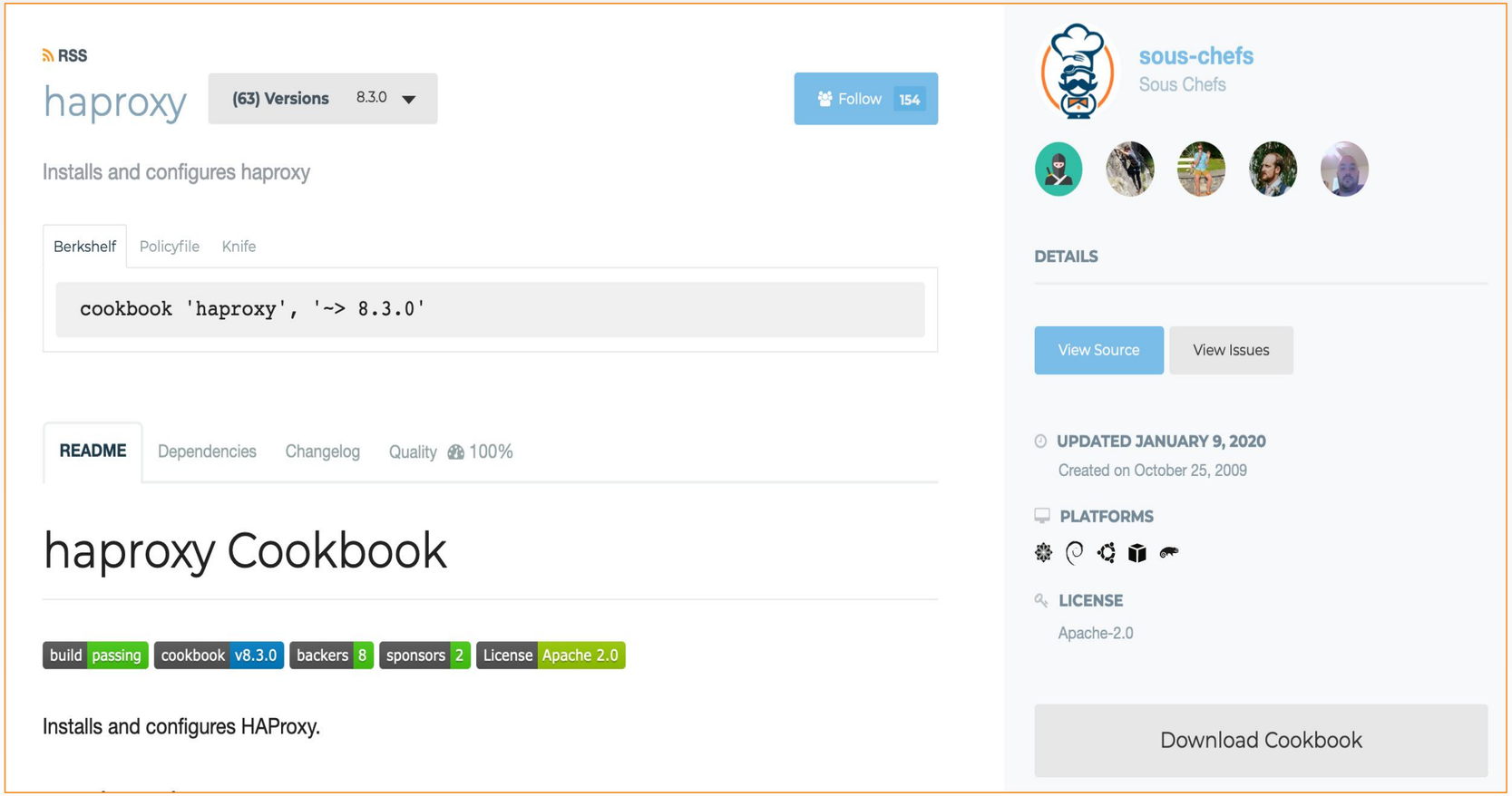


CHEF

GL: Supermarket Cookbooks

On the left, we are presented with the various ways we can install the cookbook...

On the right side we can see the individuals that maintain the cookbook...

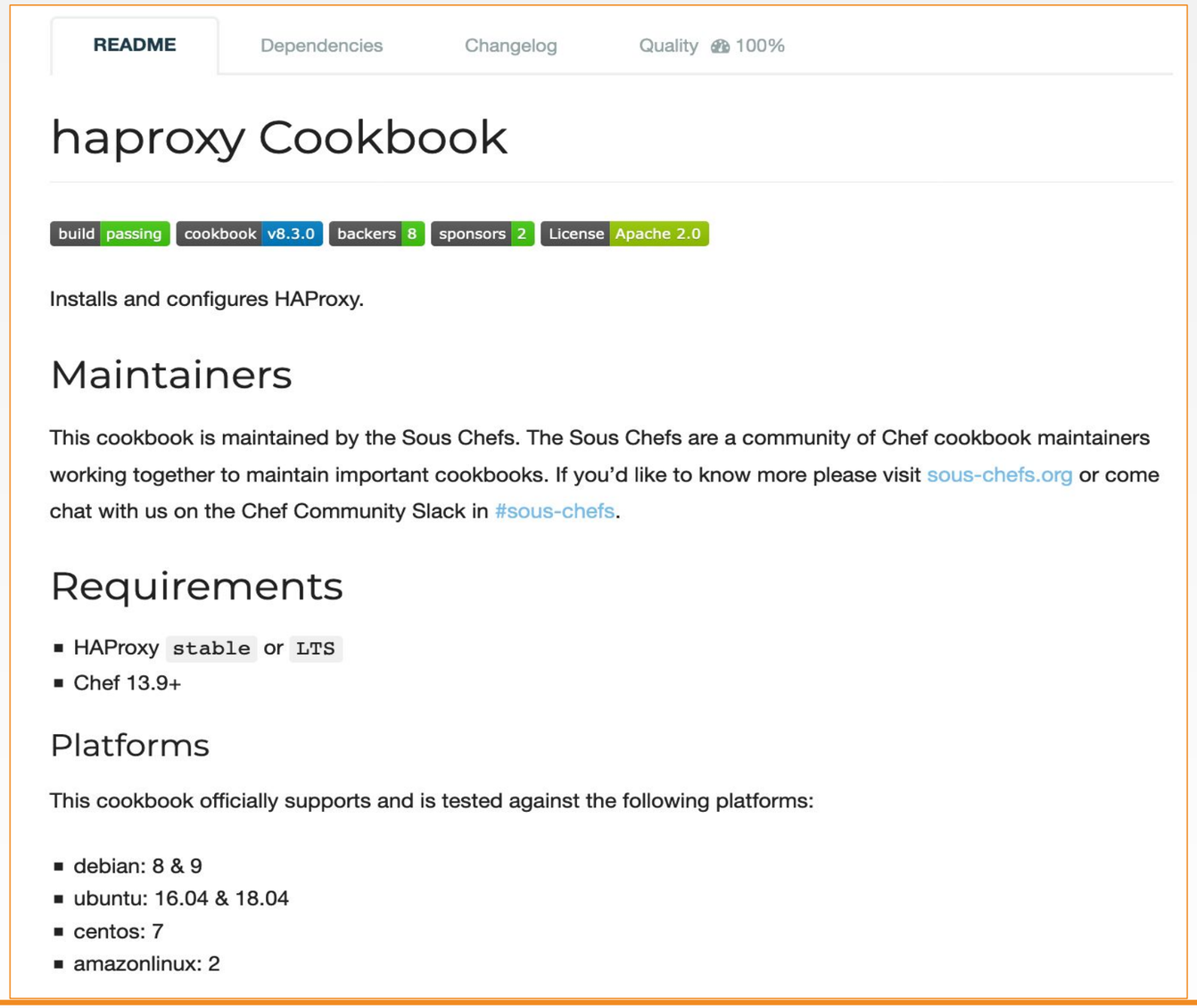


CHEF

GL: Supermarket Cookbooks

The area to focus most of your attention from the beginning is the README.

Reading and understanding the README at a glance is difficult. It is a skill that comes with time.



README Dependencies Changelog Quality 100%

haproxy Cookbook

build passing | cookbook v8.3.0 | backers 8 | sponsors 2 | License Apache 2.0

Installs and configures HAProxy.

Maintainers

This cookbook is maintained by the Sous Chefs. The Sous Chefs are a community of Chef cookbook maintainers working together to maintain important cookbooks. If you'd like to know more please visit sous-chefs.org or come chat with us on the Chef Community Slack in #sous-chefs.

Requirements

- HAProxy `stable` or `LTS`
- Chef 13.9+

Platforms

This cookbook officially supports and is tested against the following platforms:

- debian: 8 & 9
- ubuntu: 16.04 & 18.04
- centos: 7
- amazonlinux: 2

CHEF

GL: Supermarket Cookbooks

The README defines a number of new Resources that can be used if this Cookbook is included in a node's run-list.

These Custom Resources are defined with a Cookbook's resources/ directory. A well-written README will define the actions and properties a Custom Resource accepts.

Common Resource Features

HAProxy has many configurable options available, this cookbook makes the most popular options available as resource properties.

If you wish to use a HAProxy property that is not listed the `extra_options` hash is available to take in any number of additional values.

For example, the ability to disable listeners is not provided out of the box. Further examples can be found in either `test/fixtures/recipes` or `spec/test/recipes`. If you have questions on how this works or would like to add more examples so it is easier to understand, please come talk to us on the Chef Community Slack on the #sous-chefs channel.

```
haproxy_listen 'disabled' do
  bind '0.0.0.0:1337'
  mode 'http'
  extra_options('disabled': '')
end
```

https://docs.chef.io/custom_resources.html

CHEF

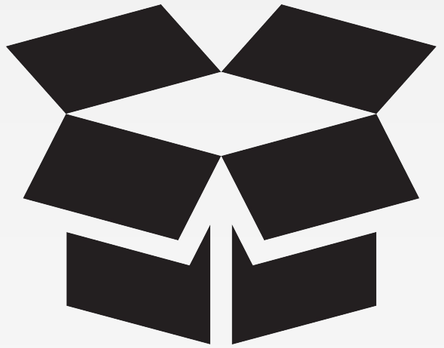
GL: Supermarket Cookbooks

Further down in the README you'll see the full list of Custom Resources that the haproxy Cookbooks provide.

We'll use of a number of these resources to easily configure a haproxy server and forward traffic to our web servers. This pattern provides an easy interface for consumers of the cookbook to change the way haproxy is deployed, without having to write all the logic yourself.

Resources

- haproxy_acl
- haproxy_backend
- haproxy_cache
- haproxy_config_defaults
- haproxy_config_global
- haproxy_fastcgi
- haproxy_frontend
- haproxy_install
- haproxy_listen
- haproxy_mailer
- haproxy_peer
- haproxy_resolver
- haproxy_service
- haproxy_use_backend
- haproxy_userlist



Using Community Cookbooks

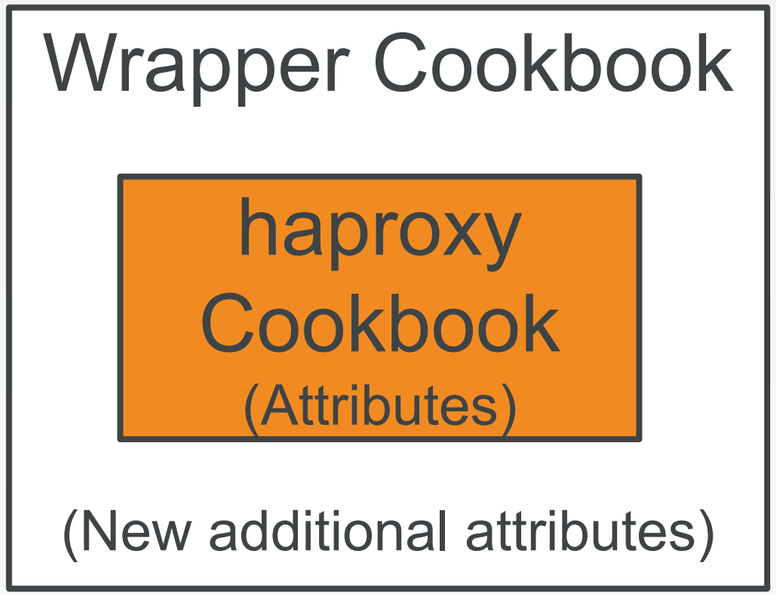
Chef Community Cookbooks can be used as-is but in most cases you will want to use them as a foundation as you write your own.

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

GL: Supermarket Cookbooks

Reminder: A wrapper cookbook is a new cookbook that encapsulates the functionality of the original cookbook.

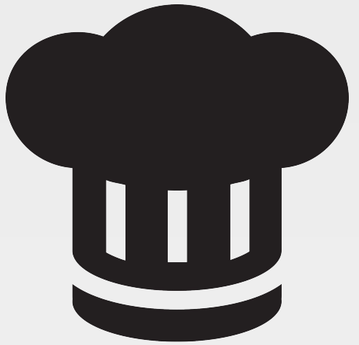
It can define new default values for the recipes.



https://docs.chef.io/supermarket.html#wrapper-cookbooks

https://www.chef.io/blog/2013/12/03/doing-wrapper-cookbooks-right/

EXERCISE



Group Lab: Load Balancer

Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ✔ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ❑ Configure the load balancer to send traffic to the iis_web and apache_web nodes
- ❑ Create myhaproxy Policyfile
- ❑ Upload myhaproxy Policyfile.lock to the Chef Infra Server (uploads cookbooks)
- ❑ Bootstrap a new node that runs the myhaproxy (load balancer) cookbook

GL: Returning to the Chef Repository Directory

```
$ cd ~/chef-repo
```


CHEF

GL: Generating a New Cookbook

```
$ chef generate cookbook cookbooks/myhaproxy
```

```
Generating cookbook myhaproxy
- Ensuring correct cookbook content
- Committing cookbook files to git

Your cookbook is ready. To setup the pipeline, type `cd cookbooks/myhaproxy`, then
run `delivery init`
```


CHEF

GL: Creating a Dependency in the Cookbook

~/chef-repo/cookbooks/myhaproxy/metadata.rb

```
name 'myhaproxy'
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'All Rights Reserved'
description 'Installs/Configures myhaproxy'
long_description 'Installs/Configures myhaproxy'
version '0.1.0'
chef_version '>= 15.0
depends 'haproxy', '~> 8.3.0'
```


CHEF

Custom Resources

A recipe can call any recipes or custom resources from a dependency that's defined with 'depends' in the metadata.rb file.

You can call a Custom Resource from any recipe in your wrapper cookbook.

GL: Include the haproxy's manual recipe in default recipe

~/chef-repo/cookbooks/myhaproxy/recipes/default.rb

```
#
# Cookbook Name:: myhaproxy
# Recipe:: default
#
# Copyright (c) 2020 The Authors, All Rights Reserved.

haproxy_install 'package'


haproxy_frontend 'http-in' do
  bind '*:80'
  default_backend 'server_backend'
end
...
```

GL: Viewing Help on the Node Show Subcommand

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

```
knife node show NODE (options)
    -a ATTR1 [--attribute ATTR2] ,   Show one or more attributes
        --attribute
    -s, --server-url URL             Chef Server URL
        --chef-zero-host HOST        Host to start chef-zero on
        --chef-zero-port PORT        Port (or port range) to start chef-zero on.
Port ranges
    -k, --key KEY                    API Client Key
        --[no-]color                 Use colored output, defaults to false on
Windows, true
    -c, --config CONFIG              The configuration file to use
        --defaults                   Accept default values for all questions
    -d, --disable-editing            Do not open EDITOR, just accept the data as is
```


CHEF

Demo: Viewing the Node's IP Address

```
$ knife node show iis_web -a ipaddress
```

```
iis_web:
  ipaddress: 172.31.8.68
```

This method of retrieving the IP address is not useful if you need the external IP address. We'll show you another way in a moment.

PROBLEM

Amazon EC2 Instances

The IP address and host name are unfortunately not how we can address these nodes within our recipes.

CHEF

GL: Viewing the Node's Cloud Details

```
$ knife node show iis_web -a cloud
```

```
iis_web:
  cloud:
    local_hostname:   ip-172-31-8-68.ec2.internal
    local_ipv4:       172.31.8.68
    private_ips:      172.31.8.68
    provider:         ec2
    public_hostname:  ec2-54-175-46-24.compute-1.amazonaws.com
    public_ips:       54.175.46.24
    public_ipv4:      54.175.46.24
```

You'll need this information for the next task.

CHEF

GL: Viewing the Node's Cloud Details

```
$ knife node show apache_web -a cloud
```

```
apache_web:
  cloud:
    local_hostname:   ip-172-31-57-169.ec2.internal
    local_ipv4:       172.31.57.169
    private_ips:      172.31.57.169
    provider:         ec2
    public_hostname:  ec2-34-196-10-17.compute-1.amazonaws.com
    public_ips:       34.196.10.17
    public_ipv4:      34.196.10.17
```

You'll need this information for the next task.

CHEF

GL: Inserting Real Node Data into the Attributes

~/chef-repo/cookbooks/myhaproxy/recipes/default.rb

```
...

haproxy_backend 'server_backend' do
  server [
  'ec2-54-175-46-24.compute-1.amazonaws.com 54.175.46.24:80 maxconn 32',
  'ec2-34-196-10-17.compute-1.amazonaws.com 34.196.10.17:80 maxconn 32'
  ]
end

haproxy_service 'haproxy'
```

Replace the hostname value and IP address values with your **iis_web** and **apache_web** node's public host name and public IP address.

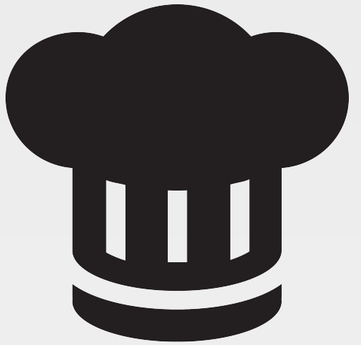
CHEF

GL: Viewing the Complete Recipe

~/chef-repo/cookbooks/myhaproxy/recipes/default.rb

```
haproxy_install 'package'

haproxy_frontend 'http-in' do
  bind '*:80'
  default_backend 'server_backend'
end

haproxy_backend 'server_backend' do
  server [
  'ec2-54-175-46-24.compute-1.amazonaws.com 54.175.46.24:80 maxconn 32',
  'ec2-34-196-10-17.compute-1.amazonaws.com 34.196.10.17:80 maxconn 32'
  ]
end

haproxy_service 'haproxy'
```

EXERCISE



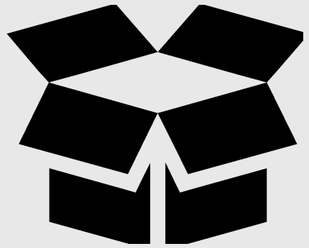
Group Lab: Load Balancer

Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ✔ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ✔ Configure the load balancer to send traffic to the iis_web and apache_web nodes
- ❑ Create myhaproxy Policyfile
- ❑ Upload myhaproxy Policyfile.lock to the Chef Infra Server (uploads cookbooks)
- ❑ Bootstrap a new node that runs the myhaproxy (load balancer) cookbook

CONCEPT



Policyfile.rb and the Policyfile.lock.json

Now that we have our myhaproxy cookbook in our chef-repo, we can create our Policyfile.rb and then generate our Policyfile.lock.json as we discussed in previous modules.

This time we'll name our Policyfile **myhaproxy**.

GL: Generate the Policyfile and Name it myhaproxy

```
> cd ~/chef-repo
> chef generate policyfile policyfiles/myhaproxy
```

```
Recipe: code_generator::policyfile
  * template[/Users/sdelfante/chef-repo/policyfiles/myhaproxy.rb] action create
    - create new file /Users/sdelfante/chef-repo/policyfiles/myhaproxy.rb
    - update content in file /Users/sdelfante/chef-repo/policyfiles/myhaproxy.rb
from none to 2ae2aa
    (diff output suppressed by config)
```

GL: Verify that the Policyfile Exists

```
>ls policyfiles/ (or dir for Windows)
```

```
company_web.rb    company_web.lock.json    myhaproxy.rb
```


CHEF

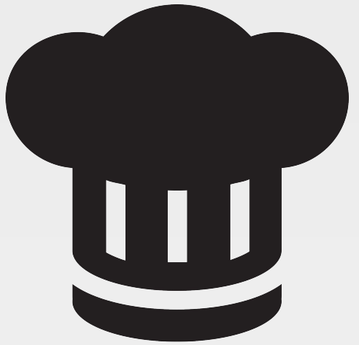
GL: Edit the New myhaproxy.rb Policyfile

~/chef-repo/policyfiles/myhaproxy.rb

```
#...skipping for brevity…
# https://docs.chef.io/policyfile.html
# A name that describes what the system you're building with Chef does.
name 'myhaproxy'

# Where to find external cookbooks:
default_source :supermarket

# run_list: chef-client will run these recipes in the order specified.
run_list 'myhaproxy::default'

# Specify a custom source for a single cookbook:
cookbook 'myhaproxy', path: '../cookbooks/myhaproxy'
```

Update the 'cookbook' section

EXERCISE



Group Lab: Load Balancer

Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ✔ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ✔ Configure the load balancer to send traffic to the iis_web and apache_web nodes
- ✔ Create myhaproxy Policyfile
- ❑ Upload myhaproxy Policyfile.lock to the Chef Infra Server (uploads cookbooks)
- ❑ Bootstrap a new node that runs the myhaproxy (load balancer) cookbook

GL: Generate the myhaproxy.lock.json

```
~/chef-repo> chef install policyfiles/myhaproxy.rb
```

```
Building policy myhaproxy
Expanded run list: recipe[myhaproxy::default]
Caching Cookbooks...
Installing myhaproxy >= 0.0.0 from path
Using       haproxy          8.3.0
Using       build-essential 8.2.1
Using       yum-epel         4.1.4
Using       seven_zip        4.1.4
Using       mingw            2.1.3
Using       windows          6.0.1

Lockfile written to /Users/robin/chef-repo/policyfiles/myhaproxy.lock.json
Policy revision id: 6e8a60de56e67ff84a7b7d8468c8ae63effd2f4d72afcc3480295954a24e0cdc
```


CHEF

GL: Verify that the myhaproxy.lock.json Exists

```
> ls policyfiles/ (or dir for Windows)
```

```
apache.lock.json     company_web.rb    README.md
apache.rb            myhaproxy.lock.json
company_web.lock.json  myhaproxy.rb
```


CHEF

GL: Push the myhaproxy.lock.json to Chef Infra Server

```
~/chef-repo> chef push prod policyfiles/myhaproxy.lock.json
```

```
Uploading policy myhaproxy (6e8a60de56) to policy group prod
Using    build-essential 8.2.1 (4b9d5c72)
Using    haproxy         8.3.0 (1a4f7607)
Using    mingw           2.1.3 (9f5d572c)
Using    seven_zip       4.2.2 (0e1fed3b)
Using    windows         6.0.1 (042f3380)
Using    yum-epel        4.1.3 (187c02d6)
Uploaded myhaproxy       0.1.0 (520e62bf)
```


CHEF

GL: Verify the myhaproxy Policy is on Chef Infra Server

```
~/chef-repo> chef show-policy
```

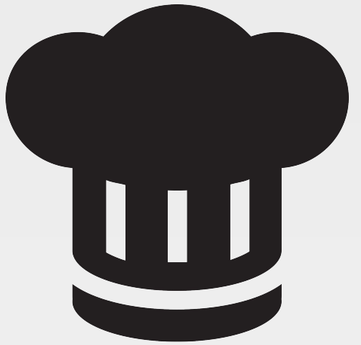
```
company_web
===========

* prod:   55529dbd15

myhaproxy
=========

* prod:   e46185fa40
```

Here we can see that the **myhaproxy** policy has been uploaded to Chef Infra Server and is in the **prod** policy group.

Also notice the policy name that was derived from the contents of the **myhaproxy.lock.json** .

CHEF

EXERCISE



Group Lab: Load Balancer

Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ✔ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ✔ Configure the load balancer to send traffic to the iis_web and apache_web nodes
- ✔ Create myhaproxy Policyfile
- ✔ Upload myhaproxy Policyfile.lock to the Chef Infra Server (uploads cookbooks)
- ❑ Bootstrap a new node that runs the myhaproxy (load balancer) cookbook

GL: Bootstrap a New Linux Node

```
$ knife bootstrap IPADDRESS -U USER -P PWD --sudo -N lb
  --policy-name myhaproxy --policy-group prod
```

```
 [34.196.50.77] Starting Chef Infra Client, version 17.3.48
 [34.196.50.77] Using policy 'myhaproxy' at revision
'ff07bcb7f5f57ac1cd6c2eb3f7c7785f8d5312c035725e3c54fb0d84292c6670'
 [34.196.50.77] resolving cookbooks for run list: ["myhaproxy::default@1.0.0 (60
 [34.196.50.77] Synchronizing Cookbooks:
 [34.196.50.77]
 [34.196.50.77] - build-essential (8.2.1)
 [34.196.50.77]   - haproxy (8.3.0)
 [34.196.50.77]   - mingw (2.1.3)
 [34.196.50.77] - myhaproxy (0.1.0)
 [34.196.50.77] - seven_zip (4.2.2)
 [34.196.50.77] - yum-epel (4.1.4)
 [34.196.50.77] - windows (6.0.1)
...
Running handlers:
 [34.196.50.77]
 [34.196.50.77] Running handlers complete [34.196.50.77] Chef Infra Client finished, 16/29
resources updated in 29 seconds
```

node name

policy_name

policy_group

CHEF

GL: Validate the Run List Has Been Set

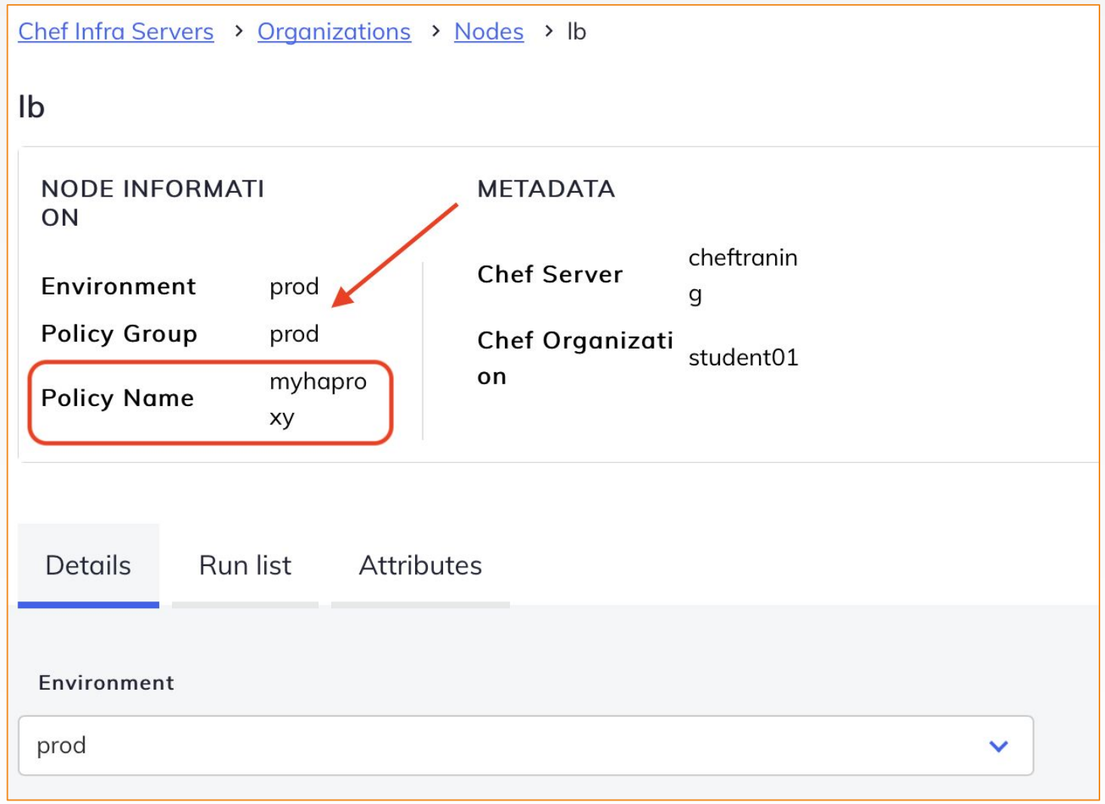
```
$ knife node show lb
```

```
Node Name:   lb
Policy Name:  myhaproxy
Policy Group: prod
FQDN:        ip-172-31-22-163.ec2.internal
IP:          34.196.50.77
Run List:    recipe[myhaproxy::default]
Recipes:     myhaproxy::default, yum-epel::default
Platform:    centos 7.6.1810
Tags:
```

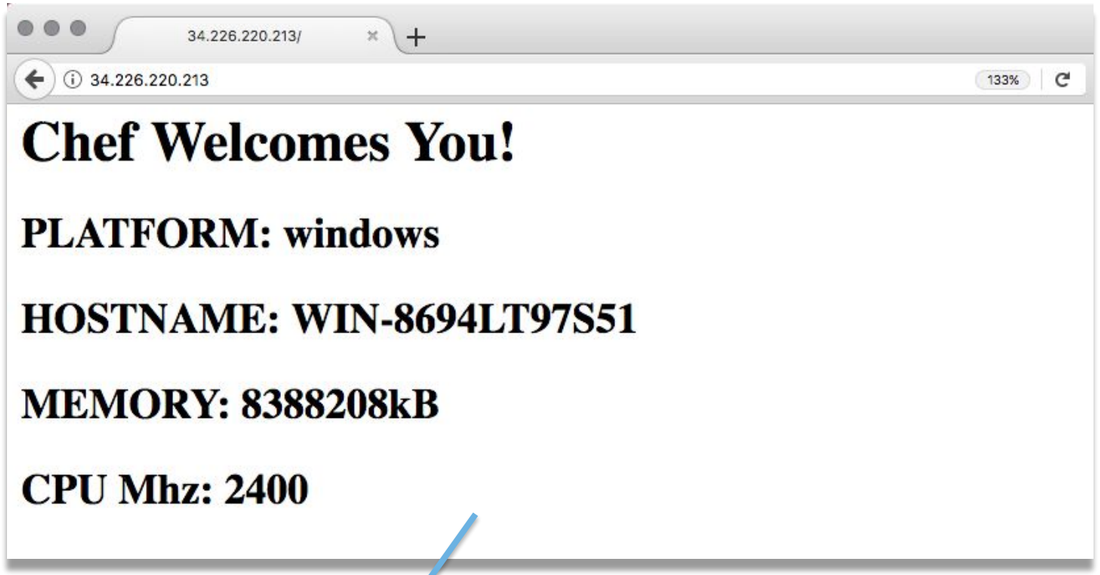

CHEF

Verify policy applied to lb

View Information associated with node.



Lab: Test the Load Balancer

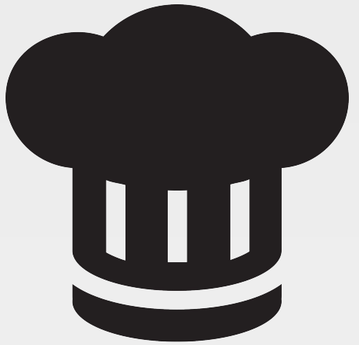


CHEF

Lab: Test the Load Balancer

CHEF

EXERCISE



GL: Load Balancer

Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ✔ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ✔ Configure the load balancer to send traffic to the iis_web node
- ✔ Create Policyfile and lock.
- ✔ Upload Policyfile.lock to the Chef server
- ✔ Bootstrap a new node that runs the haproxy (load balancer) cookbook
- ✔ Converge the node

Review Questions

1. What are the benefits of the Chef Super Market? And what are the drawbacks?
2. Why do you use a wrapper cookbook?
3. When might you decide to not wrap the cookbook?

Q&A

What questions can we help you answer?

- Chef Supermarket
- Wrapper Cookbooks
- Node Attributes
- knife ssh

CHEF™