Community Cookbooks

Find, Explore and View Chef Cookbooks

1L

Objectives

After completing this module, you should be able to

- Find cookbooks on the Chef Supermarket
- Create a wrapper cookbook for a community cookbook - haproxy
- Replace the existing default values
- Run the loadbalancer in a bootstrapped node

This is a no lab chapter

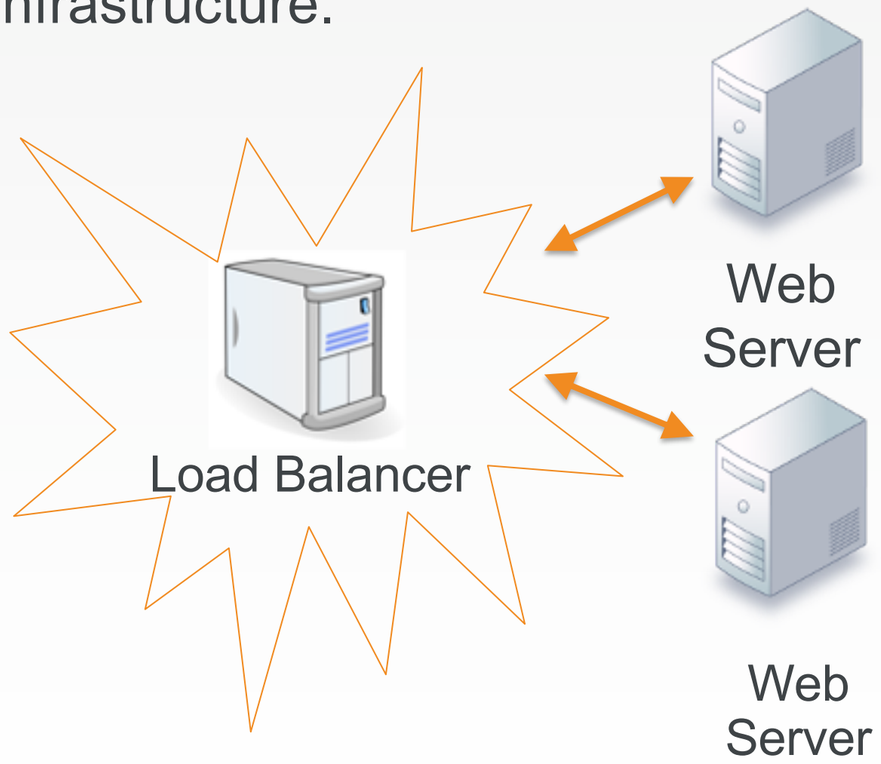
Note:- The community cookbook for loadbalancer (haproxy) mentioned in this chapter works with windows, not with centos.

We will go through the process of using community cookbooks in this chapter, but we will create loadbalancer in the next chapter from scratch haproxy without community cookbooks

Load Balancer

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

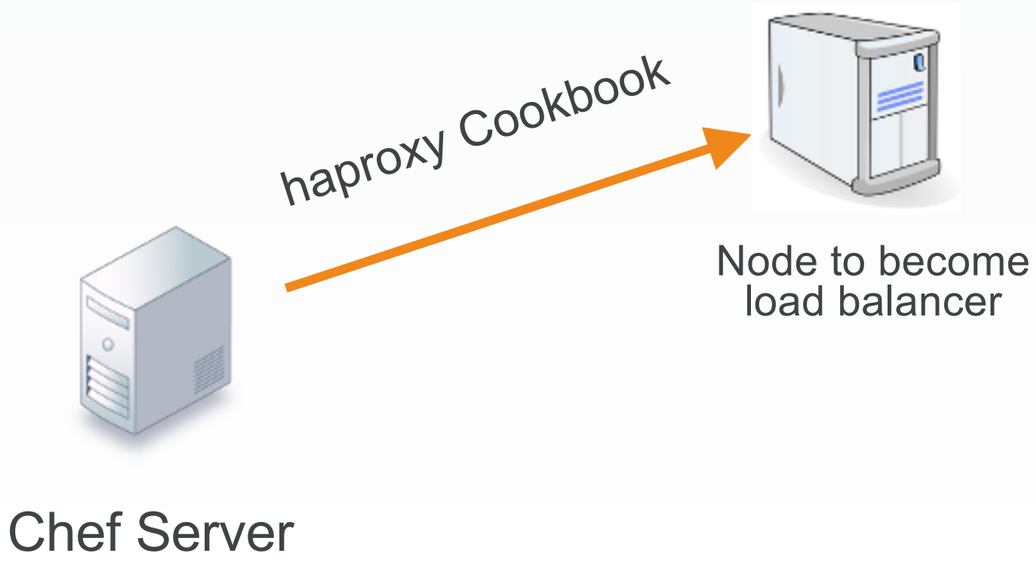
Receives requests and relays them to other systems.



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.



CONCEPT

Community Cookbooks

Someone already wrote that cookbook?

Available through the community site called the Chef Supermarket

https://supermarket.chef.io

CHEF

EXERCISE

Group Lab: Load Balancer

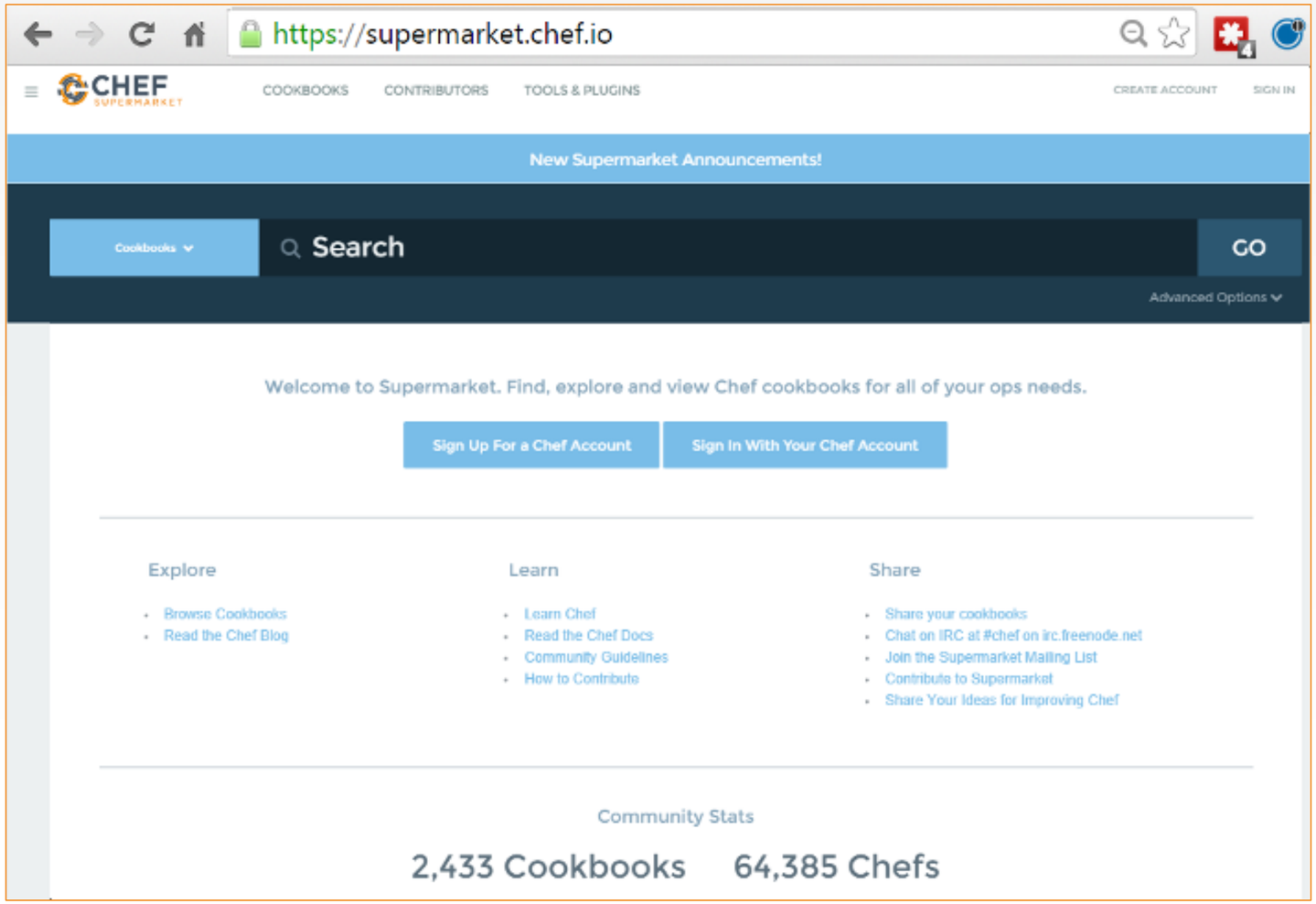
Adding a load balancer will allow us to better grow our infrastructure.

Objective:

- ❑ Find a Cookbook on the Chef Supermarket to Manage a load balancer
- ❑ Configure the load balancer to send traffic to the node1 node
- ❑ Upload cookbook to Chef Server
- ❑ Bootstrap a new node that runs the haproxy (load balancer) cookbook

GL: Community Cookbooks

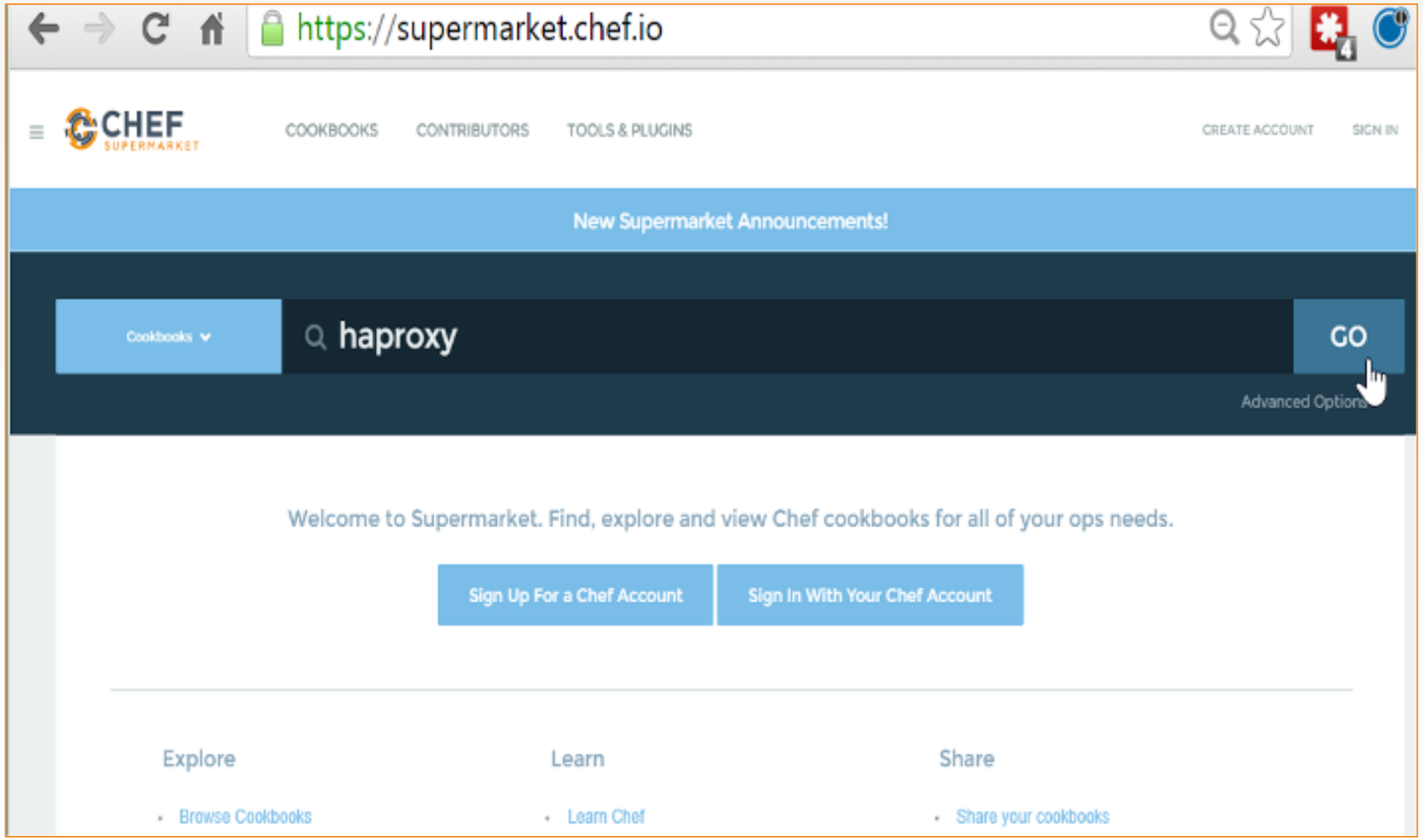
- Community cookbooks are managed by individuals.
- Chef does not verify or approve cookbooks in the Supermarket.
- Cookbooks may not work for various reasons.
- Still, there are real benefits to community cookbooks.



GL: Searching in the Supermarket

STEPS

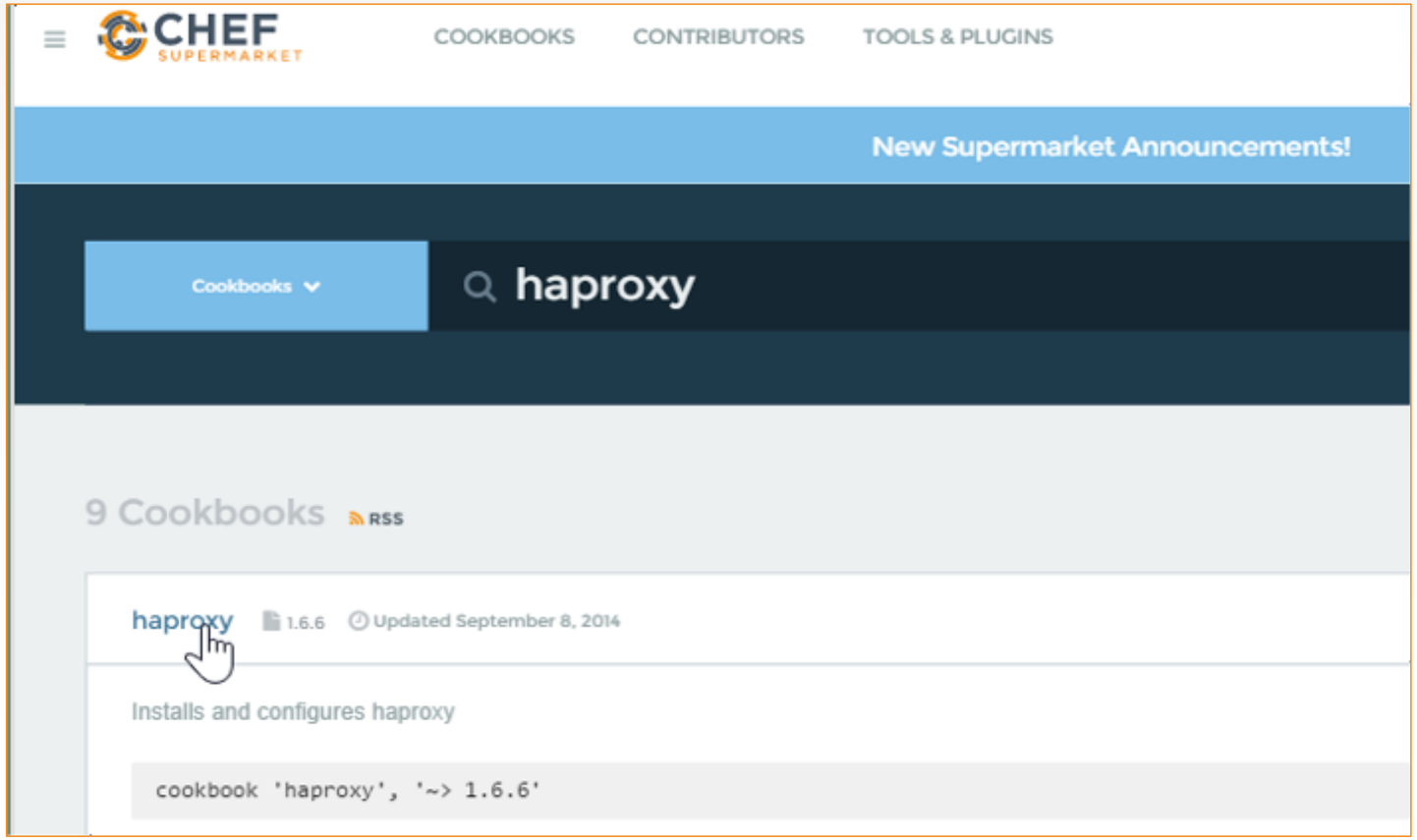
1. Visit **supermarket.chef.io**
2. Select the search field and type in **haproxy** in the search field. Then click the **GO** button.
3. Click the resulting **haproxy** link.



GL: Searching in the Supermarket

STEPS

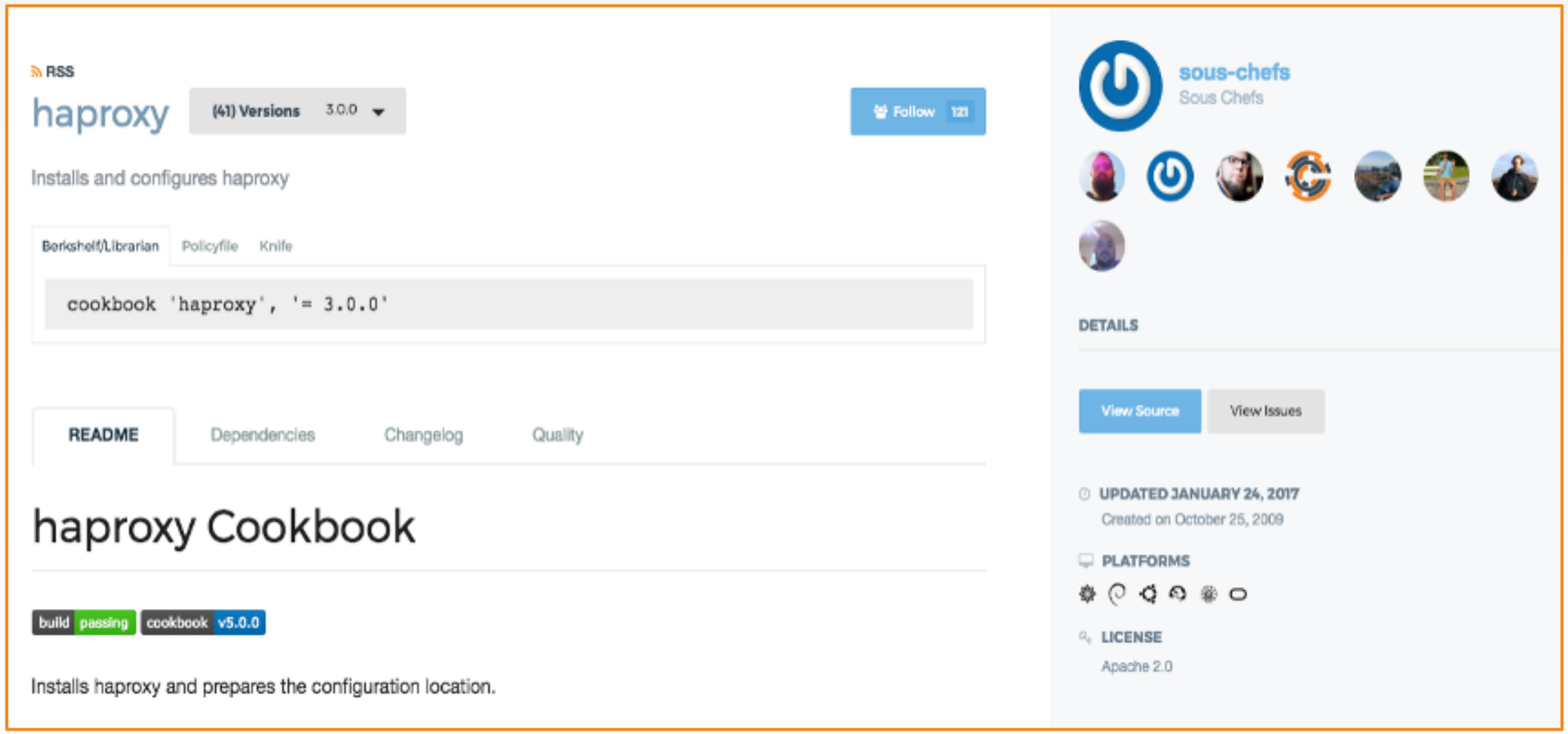
1. Visit **supermarket.chef.io**
2. Select the search field and type in **haproxy** in the search field. Then click the **GO** button.
3. Click the resulting **haproxy** link.



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...



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

haproxy Cookbook

build passing | cookbook v5.0.0

Installs haproxy and prepares the configuration location.

Requirements

- Chef 12.1+

Platforms

- Ubuntu 12.04+
- RHEL 6+, CentOS6+
- RHEL 7+, CentOS7+
- Debian 8+

CHEF

GL: Supermarket Cookbooks

These node attributes are different than the automatic ones defined by Ohai.

Attributes defined in a cookbook are not considered automatic.

Attributes

- `node['haproxy']['incoming_address']` - sets the address to bind the haproxy process on, 0.0.0.0 (all addresses) by default
- `node['haproxy']['incoming_port']` - sets the port on which haproxy listens
- `node['haproxy']['members']` - used by the default recipe to specify the member systems to add. Default

```
[{
  "hostname" => "localhost",
  "ipaddress" => "127.0.0.1",
  "port" => 4000,
  "ssl_port" => 4000
}, {
  "hostname" => "localhost",
  "ipaddress" => "127.0.0.1",
  "port" => 4001,
  "ssl_port" => 4001
}]
```

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

CONCEPT

Using Community Cookbooks

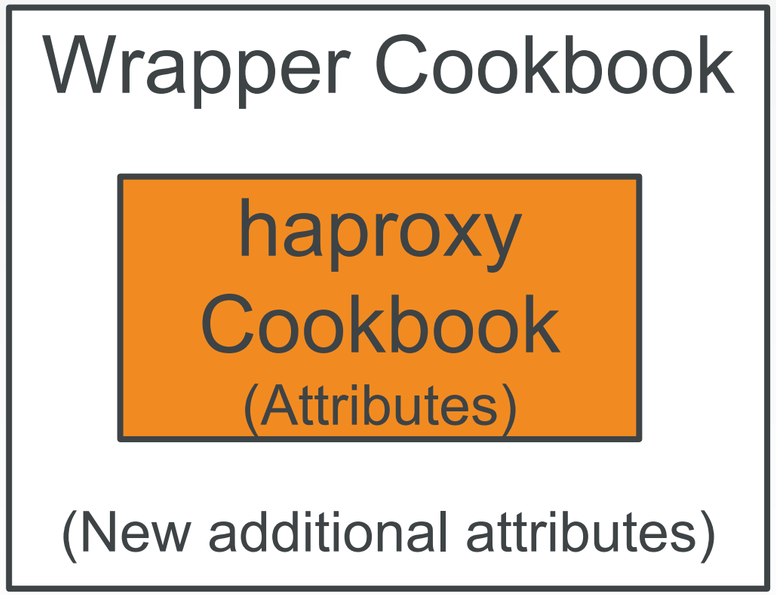
Chef Community Cookbooks can be used as-is but in most cases you will want to use them as a foundation as you write your own.

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

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 node1 node
- ❑ Upload cookbook to Chef Server
- ❑ Bootstrap a new node that runs the haproxy (load balancer) cookbook

GL: Returning to the Chef Repository Directory

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

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`
```

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 '>= 14.0
depends 'haproxy', '~> 3.0.0'
```

Match with any version of 3.x.x

CONCEPT

include_recipe

A recipe can include one (or more) recipes located in cookbooks by using the **include_recipe** method.

When a recipe is included, the resources found in that recipe will be inserted (in the same exact order) at the point where the **include_recipe** keyword is located.

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

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

```
#
# Cookbook Name:: myhaproxy
# Recipe:: default
#
# Copyright (c) 2016 The Authors, All Rights
Reserved.


include_recipe 'haproxy::manual'
```

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
```

Demo: Viewing the Node's IP Address

```
$ knife node show node1 -a ipaddress
```

```
node1:
  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 node1 -a cloud
```

```
node1:
  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 slide's task.

GL: Inserting Real Node Data into the Attributes

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

```
node['haproxy']['members'] = [
  {
    'hostname' => 'ec2-52-8-71-11.us-west-1.compute.amazonaws.com',
    'ipaddress' => '52.8.71.11',
    'port' => 80,
    'ssl_port' => 80
  }]
include_recipe 'haproxy::manual'
```

Replace the hostname value and the ipaddress value with your **node1** node's host name and IP address.

GL: Setting the Default Attributes Precedence Level

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

```
node.default['haproxy']['members'] = [{
    'hostname' => 'ec2-52-8-71-11.us-west-1.compute.amazonaws.com',
    'ipaddress' => '52.8.71.11',
    'port' => 80,
    'ssl_port' => 80
  }]

include_recipe 'haproxy::manual'
```

replace a default attribute in a recipe

GL: Viewing the Complete Recipe

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

```
node.default['haproxy']['members'] = [{
    'hostname' => 'ec2-52-8-71-11.us-west-1.compute.amazonaws.com',
    'ipaddress' => '52.8.71.11',
    'port' => 80,
    'ssl_port' => 80
  }]

include_recipe 'haproxy::manual'
```

replace a default attribute in a recipe

Node Attribute Precedence

	Attribute Files	Node/Recipe	Policy File
default	1	2	3
force_default	4	5	
normal	6	7	
override	8	9	10
force_override	11	12	
automatic from ohai highest priority			

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 node1 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

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 myhaproxy
```

```
* template[/Users/sdelfante/chef-repo/myhaproxy.rb] action create
    - create new file /Users/sdelfante/chef-repo/myhaproxy.rb
    - update content in file /Users/sdelfante/chef-repo/myhaproxy.rb from none
to f0eb38
    (diff output suppressed by config)
```

GL: Verify that the Policyfile Exists

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

```
Policyfile.lock.jsonREADME.md      company_web.rb      myhaproxy.rb
Policyfile.rb      company_web.lock.json    cookbooks      roles
```

GL: Edit the New myhaproxy.rb Policyfile

~/chef-repo/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 cookbook
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'
```

Replace the contents of the myhaproxy.rb below the **`#https://docs.chef.io/policyfile.html`** line with the code in green.

GL: Generate the myhaproxy.lock.json

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

```
Building policy myhaproxy
Expanded run list: recipe[myhaproxy::default]
Caching Cookbooks...
Installing myhaproxy >= 0.0.0 from path
Using       haproxy          3.0.4
Using       cpu              2.0.0
Using       build-essential 8.2.1
…

Lockfile written to /Users/sdelfante/chef-repo/myhaproxy.lock.json
Policy revision id:
e46185fa40c596e6bf916168d4cd75f1f903c9ea4742561c1da09e62310d3a0a
```

GL: Verify that the myhaproxy.lock.json Exists

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

```
Policyfile.lock.jsonREADME.md      company_web.rb      myhaproxy.lock.json
     roles
Policyfile.rb       company_web.lock.json    cookbooks      myhaproxy.rb
```

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

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

```
Uploading policy myhaproxy (e46185fa40) to policy group prod
Uploaded build-essential 8.2.1 (4b9d5c72)
Uploaded cpu             2.0.0 (78364308)
Uploaded haproxy         3.0.4 (222f5e2b)
Uploaded mingw           2.1.0 (9f5d572c)
Uploaded myhaproxy       0.1.0 (370b798a)
Uploaded poise           2.8.2 (5eada1fb)
Uploaded poise-service   1.5.2 (d92d3eba)
Uploaded seven_zip       3.1.1 (a76d3fe4)
Uploaded windows         6.0.0 (afff0357)
```

GL: Verify the myhaproxy Policy is on Chef Infra Server

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

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

* prod:  55529dbd15

myhaproxy
=========

* prod:  e46185fa40

myiis
=====

* prod:  49eef2f1f1
```

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** .

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 node1 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

GL: Bootstrap a New Linux Node

```
$ knife bootstrap <ip_node2> -x centos -i aws.pem --sudo -N node2
```

node name

```
Connecting to 34.196.50.77
WARN: [SSH] PTY requested: stderr will be merged into stdout
The authenticity of host '34.196.50.77 ()' can't be established. fingerprint is
SHA256:q7AVSJ3Ai6fNFGr8u/uwnUGOMP1MZo7QQrG8KwLSUmI.
Are you sure you want to continue connecting
? (Y/N) Y
WARN: [SSH] PTY requested: stderr will be merged into stdout
Creating new client for lb
Creating new node for lb
Bootstrapping 34.196.50.77
…
Running handlers:
Running handlers complete
Chef Infra Client finished, 0/0 resources updated in 04 seconds
```

GL: Apply the myhaproxy Policy to Your Linux Node

```
> knife node policy set node2 prod myhaproxy
Successfully set the policy on node lb
```

node name

policy_group

policy_name

‘knife node policy set` takes 3 arguments:
* node name
* policy group
* Policy name (Name inside file)

GL: View More Information About Your Node

```
$ knife node show node2
```

```
Node Name:   node2
Policy Name:  myhaproxy
Policy Group: prod
FQDN:        ip-172-31-22-163.ec2.internal
IP:          34.196.50.77
Run List:
Recipes:
Platform:    centos 7.6.1810
Tags:
```

GL: Converge the Load Balancer

```
$ knife ssh IPADDRESS -m -x USER -P PASSWORD "sudo chef-client"
```

```
34.196.50.77 Starting Chef Infra Client, version 15.1.36
34.196.50.77 Using policy 'myhaproxy' at revision
'e46185fa40c596e6bf916168d4cd75f1f903c9ea4742561c1da09e62310d3a0a'
34.196.50.77 resolving cookbooks for run list: ["myhaproxy::default@0.1.0 (370b798)"]
34.196.50.77 Synchronizing Cookbooks:
34.196.50.77   - build-essential (8.2.1)
34.196.50.77   - cpu (2.0.0)
34.196.50.77   - haproxy (3.0.4)
34.196.50.77   - myhaproxy (0.1.0)
…
34.196.50.77 Running handlers:
34.196.50.77 Running handlers complete
34.196.50.77 Chef Infra Client finished, 15/21 resources updated in 07 seconds
34.196.50.77 [2019-07-29T15:48:19+00:00] ERROR: Failed to post reporting data to server (HTTP
400), saving to /var/chef/cache/failed-reporting-data.json
```

The "Error" can be ignored. Our version of Hosted Chef still has Reporting installed, which has been deprecated in the latest versions of Chef Server.

GL: Validate the Run List Has Been Set

```
$ knife node show node2
```

```
Node Name:   node2
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, haproxy::manual,
haproxy::install_package
Platform:    centos 7.6.1810
Tags:
```

GL: Confirm that the Website is Being Proxied

URL of load balancer.

Output from the node1 server.

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 node1 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

DISCUSSION

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?

DISCUSSION

Q&A

What questions can we help you answer?

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

CHEF

CHEF™