cd and Generate the Cookbook

```
$ cd ~/chef-repo
$ chef generate cookbook cookbooks/myhaproxy
```

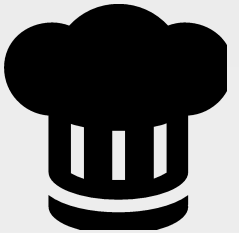
```
Compiling Cookbooks...
Recipe: code_generator::cookbook
  * directory[C:/Users/sdelfante/chef-repo/cookbooks/myhaproxy] action create
    - create new directory C:/Users/sdelfante/chef-repo/cookbooks/myhaproxy
  * template[C:/Users/sdelfante/chef-repo/cookbooks/myhaproxy/metadata.rb] action
create_if_missing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


CHEF

Create a Dependency in the Cookbook

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

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

depends 'haproxy', '= 2.0.0'
```



Load Balancer

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

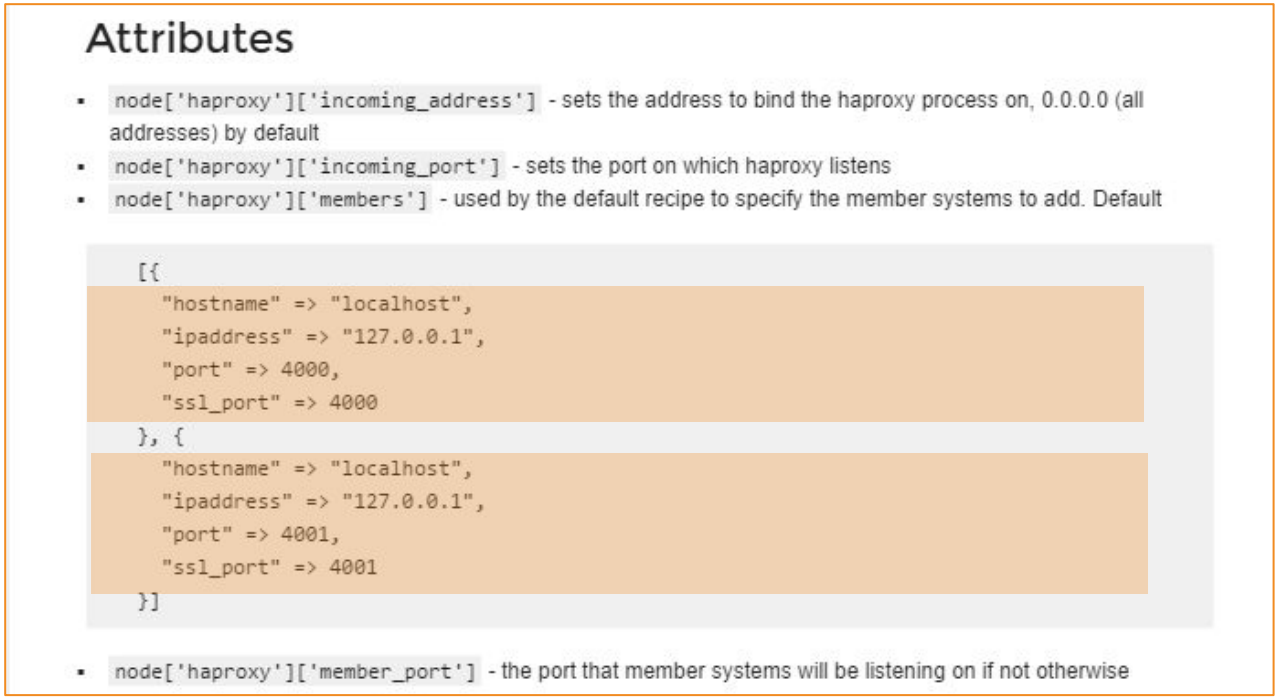
Objective:

- ✓ Find or create a cookbook to manage a load balancer
- ✓ Configure the load balancer to send traffic to the new node
- ✓ Upload cookbook to Chef Server
- ✓ Bootstrap a new node that runs the haproxy cookbook

Supermarket Cookbooks

Currently, the haproxy cookbook assumes that there are two different services running on the localhost at port 4000 and port 4001.

In a moment, you'll need to change that.



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
}]
```

- `node['haproxy']['member_port']` - the port that member systems will be listening on if not otherwise

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

CHEF

Capture Node's Public Host Name and IP

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

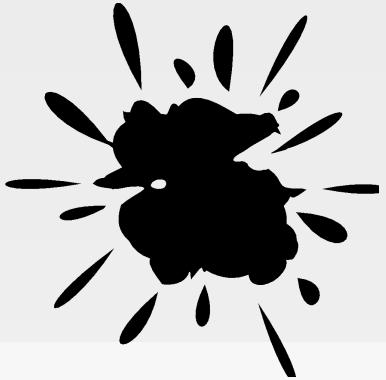
```
knife node show NODE (options)
    -a ATTR1 [--attribute ATTR2] ,   Show one or more attributes
        --attribute
    -s, --server-url URL             Chef Server URL
        --chef-zero-host HOST        Host to start chef-zero on
        --chef-zero-port PORT        Port (or port range) to start 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
    -e, --editor EDITOR              Set the editor to use for interactive commands
```

Capture Node's Public Host Name and IP

```
$ knife node show web1 -a ipaddress
```

```
web1:
  ipaddress: 192.168.10.43
```

NOTE:
Cloud Instances



If using a cloud provider, such as EC2, Azure or Google Compute, you'll need to discover the public ipaddress for your instance. You usually can't simple ask for the node['ipaddress']

Instead, use the cloud.public_ipv4 and cloud.public_hostname attributes

NOTE: cloud web1's Public Host Name and IP

```
$ knife node show web1 -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
```


CHEF

Edit the myhaproxy/recipes/default.rb

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

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



include_recipe 'haproxy::manual'
```

Edit the myhaproxy/recipes/default.rb

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

```
#

node['haproxy']['members'] = [
{
    'hostname' => 'localhost',
    'ipaddress' => '127.0.0.1',
    'port' => 4000,
    'ssl_port' => 4000
  }, {
    'hostname' => 'localhost',
    'ipaddress' => '127.0.0.1',
    'port' => 4001,
    'ssl_port' => 4001
}]

include_recipe 'haproxy::manual'
```

Erase 1 of the members

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

```
node['haproxy']['members'] = [
  {
    'hostname' => 'localhost',
    'ipaddress' => '127.0.0.1',
    'port' => 4000,
    'ssl_port' => 4000
  },
  {
    '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'
```

Create a node attribute node['haproxy']['members']

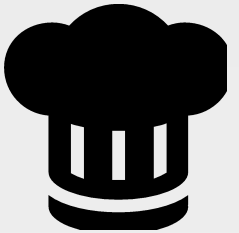
~/chef-repo/cookbooks/myhaproxy/recipes/default.rb

```
node.default['haproxy']['members'] = [{
    'hostname' => 'localhost',
    'ipaddress' => '127.0.0.1',
    'port' => 4000,
    'ssl_port' => 4000
  }]

include_recipe 'haproxy::manual'
```

GL: Edit the myhaproxy/recipes/default.rb

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

```
node.default['haproxy']['members'] = [{
    'hostname' => 'WEB1_PUBLIC_HOSTNAME',
    'ipaddress' => 'WEB1_PUBLIC_IPADDRESS',
    'port' => 80,
    'ssl_port' => 80
  }]

include_recipe 'haproxy::manual'
```



Load Balancer

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

Objective:

- ✓ Find or create a cookbook to manage a load balancer
- ✓ Configure the load balancer to send traffic to the new node
  - Upload cookbook to Chef Server
  - Bootstrap a new node that runs the haproxy cookbook