Lab: Update the Load Balancer

- ❑ Update the wrapped proxy server cookbook to include the new web node as a member.
- ❑ Upload that cookbook to the Chef Server
- ❑ Run chef-client on that system
- ❑ Verify that the load balancer delivers traffic to both web server nodes.

Verify web1's hostname and IP

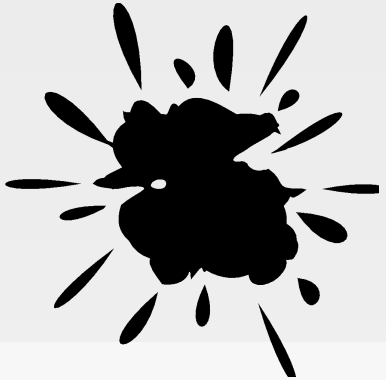
~/chef-repo/Vagrantfile

```
...
Vagrant.configure(2) do |config|

 config.vm.define 'web1' do |n|
    n.vm.box = 'bento/centos-7.2'
    n.vm.box_version = '2.2.9'
    n.vm.hostname = 'web1'
    n.vm.network :private_network, ip: '192.168.10.43'
    n.vm.provision :shell, inline: NODE_SCRIPT.dup
   set_hostname(n)
  end

 config.vm.define 'web2' do |n|
    n.vm.box = 'bento/centos-7.2'
    n.vm.box_version = '2.2.9'
    n.vm.hostname = 'web2'
    n.vm.network :private_network, ip: '192.168.10.44'
   ...
```

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 simply 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

NOTE: cloud web2's Public Host Name and IP

```
$ knife node show web2 -a cloud
```

```
node1:
  cloud:
    local_hostname:  ip-172-31-8-69.ec2.internal
    local_ipv4:      172.31.8.69
    private_ips:     172.31.8.69
    provider:        ec2
    public_hostname: ec2-54-175-46-25.compute-1.amazonaws.com
    public_ips:      54.175.46.25
    public_ipv4:     54.175.46.25
```


CHEF

Lab: Add the Other Web Server to LB

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

```
node.default['haproxy']['members'] = [{
    'hostname' => 'web1',
    'ipaddress' => '192.168.10.43',
    'port' => 80,
    'ssl_port' => 80
  },{
    'hostname' => 'web2',
    'ipaddress' => '192.168.10.44',
    'port' => 80,
    'ssl_port' => 80
  }
]

include_recipe 'haproxy::default'
```

Lab: Update the Version

~/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           '0.2.0'

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

Lab: CD and Then Run berks install

```
$ cd ~/chef-repo/cookbooks/myhaproxy
$ berks install
```

```
Resolving cookbook dependencies...
Fetching 'myhaproxy' from source at .
Fetching cookbook index from https://supermarket.chef.io...
Using build-essential (2.2.3)
Using cpu (0.2.0)
Using haproxy (2.0.0)
Using myhaproxy (0.2.0) from source at .
```

Lab: Upload the Cookbook to Chef Server

```
$ berks upload
```

```
Skipping build-essential (2.2.3) (frozen)
Skipping cpu (0.2.0) (frozen)
Skipping haproxy (2.0.0) (frozen)
Uploaded myhaproxy (0.2.0) to: 'https://api.opscode.com:443/organizations/ORGNAME'
```


CHEF

Login to Load Balancer

```
$ vagrant ssh load-balancer
```

```
Last login: Sat Dec 31 02:59:27 2016 from 10.0.2.2
[vagrant@load-balancer ~]$
```

Converge the Load Balancer

```
[vagrant@load-balancer ~]$ sudo chef-client
```

```
Starting Chef Client, version 12.17.44
resolving cookbooks for run list: ["myhaproxy"]
Synchronizing Cookbooks:
  - myhaproxy (0.1.0)
  - haproxy (2.0.0)
  - build-essential (7.0.3)
  - seven_zip (2.0.2)
  - windows (2.1.1)
  - ohai (4.2.3)
....
```

Verify the Load Balancing - curl 1

```
[vagrant@load-balancer ~]$ curl localhost
```

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>ipaddress: 192.168.10.43</h2>
    <h2>hostname: web1</h2>
  </body>
</html>
```

Verify the Load Balancing - curl 2

```
[vagrant@load-balancer ~]$ curl localhost
```

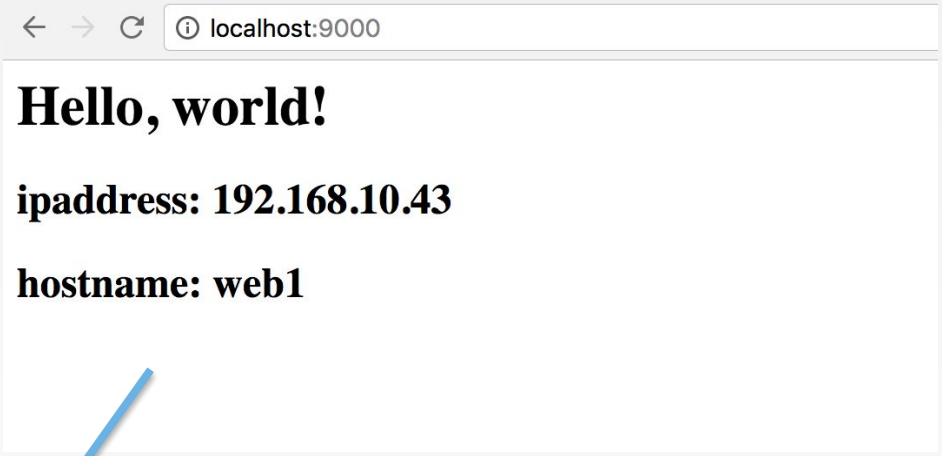
```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>ipaddress: 192.168.10.44</h2>
    <h2>hostname: web2</h2>
  </body>
</html>
```

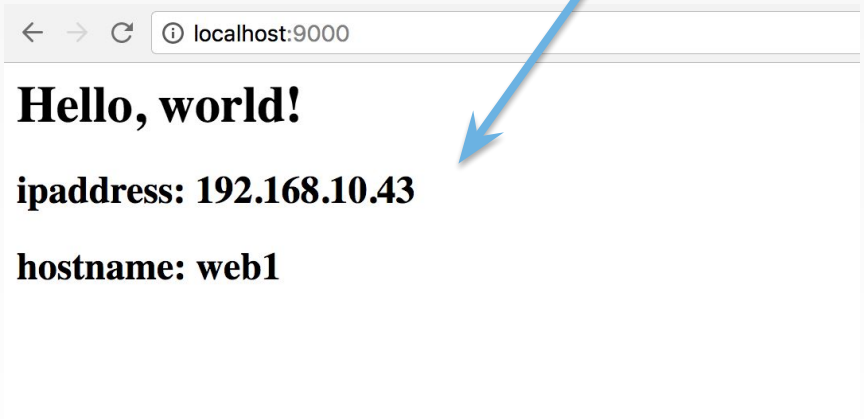
Return to your Workstation

```
[vagrant@load-balancer ~]$ exit
```

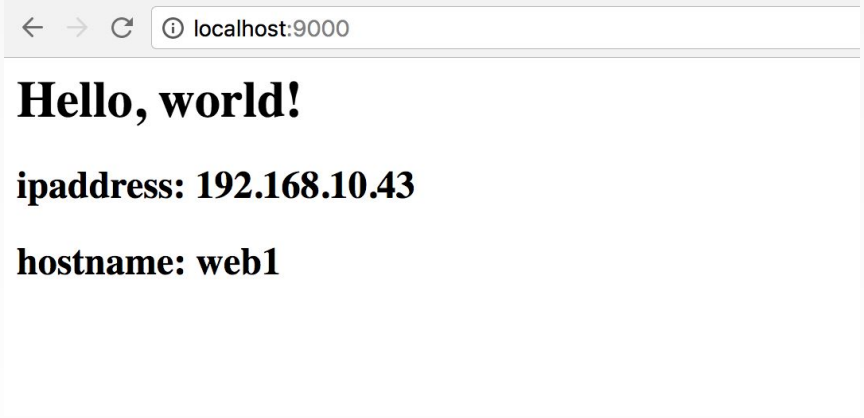
```
logout
Connection to 127.0.0.1 closed.
```

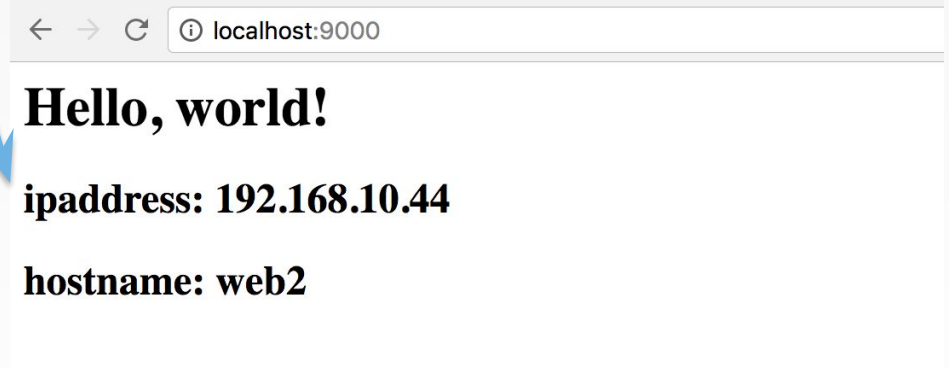
Lab: Test the Load Balancer





Lab: Test the Load Balancer





Lab: Update the Load Balancer

- ✓ Update the wrapped proxy server cookbook to include the new web node as a member.
- ✓ Upload that cookbook to the Chef Server
- ✓ Run chef-client on that system
- ✓ Verify that the load balancer delivers traffic to both web server nodes.

DISCUSSION

Discussion

What is the process to setup a third web node?

What is the process for removing a web node?

What is the most manual part of the process?