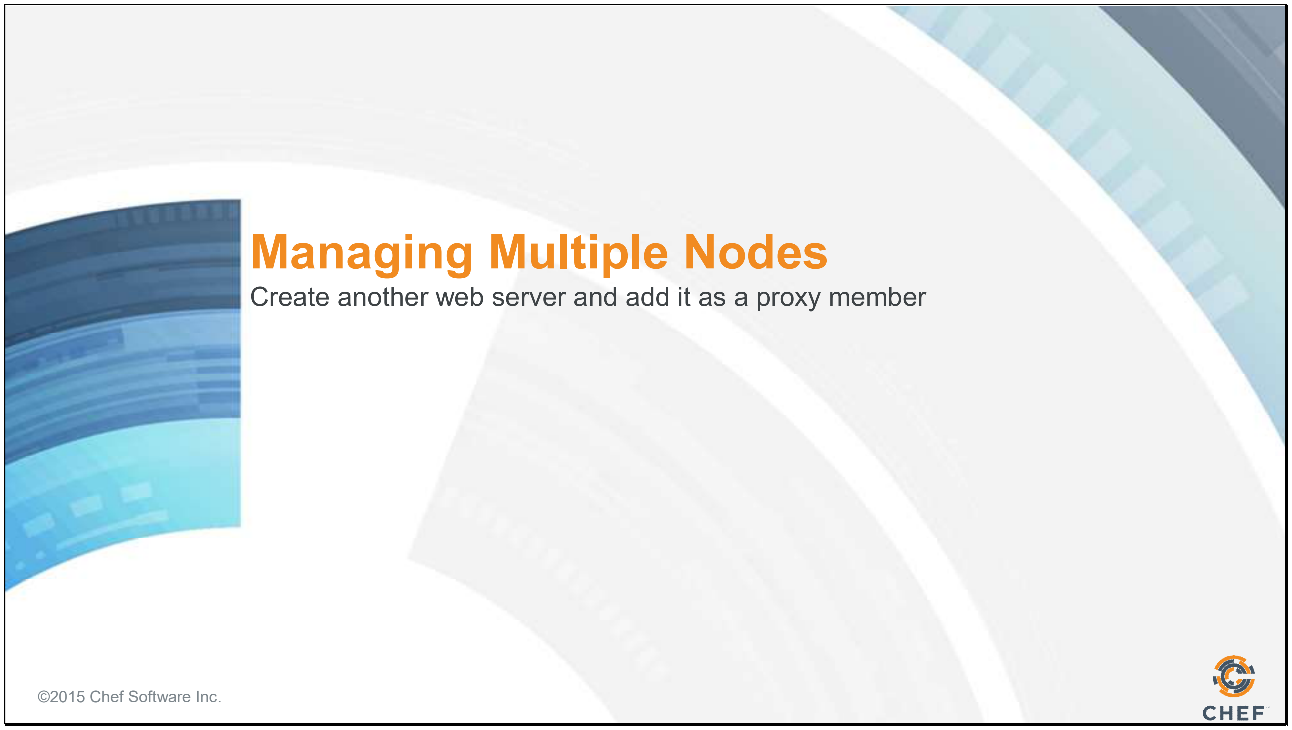
11: Managing Multiple Nodes

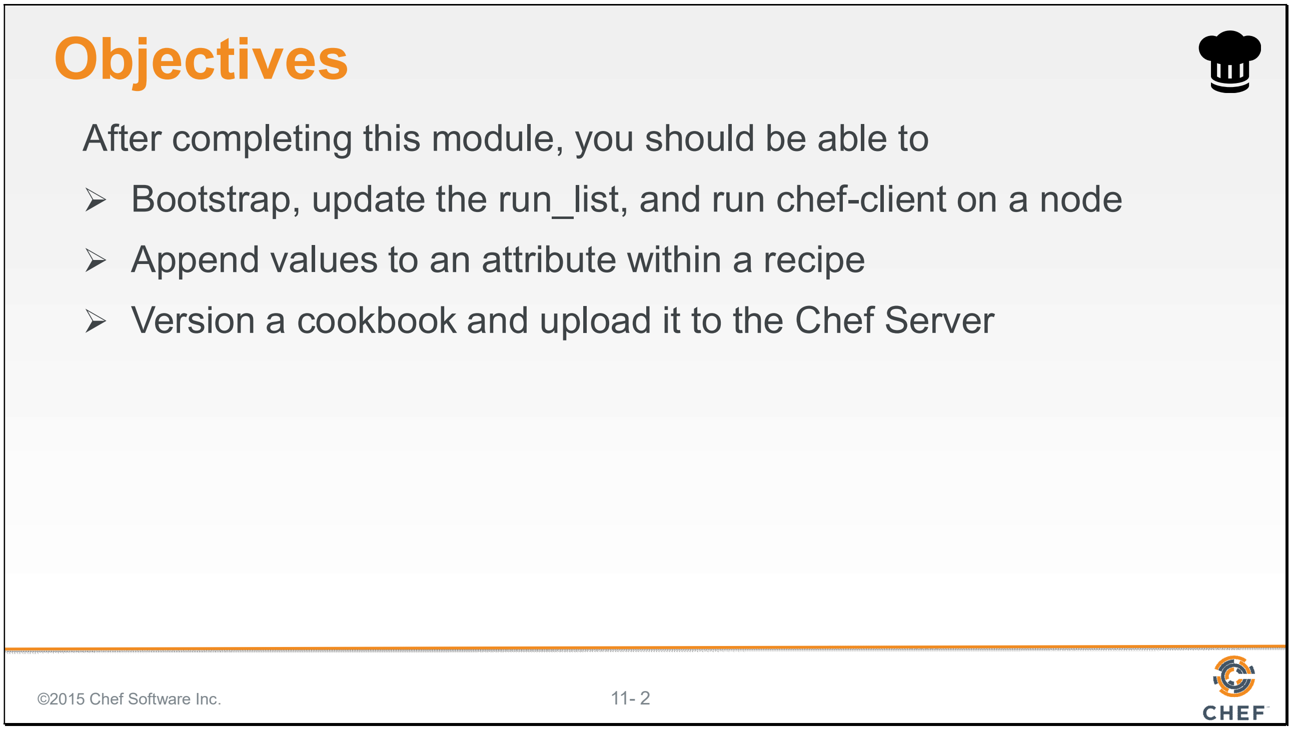


Managing Multiple Nodes

Create another web server and add it as a proxy member

This section's goal is to have you bootstrap another node, this time a web server, and add it to the proxy members.

Slide 2



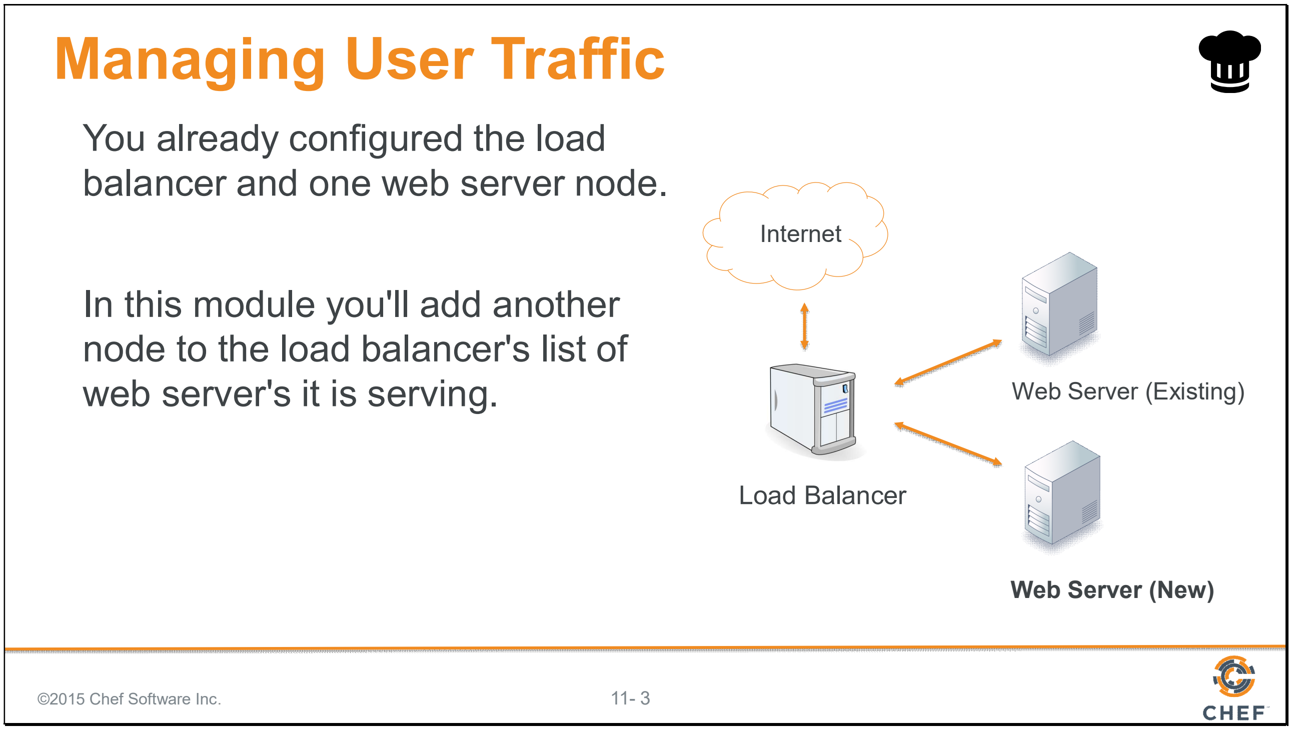
Objectives

After completing this module, you should be able to

- Bootstrap, update the run_list, and run chef-client on a node
- Append values to an attribute within a recipe
- Version a cookbook and upload it to the Chef Server

In this module you will learn how to bootstrap, update the run list, and run chef-client on a node. You will also learn how to update a default attribute within a recipe, version and upload a cookbook.

Slide 3



Managing User Traffic

You already configured the load balancer and one web server node.

In this module you'll add another node to the load balancer's list of web server's it is serving.

Internet

Load Balancer

Web Server (Existing)

Web Server (New)

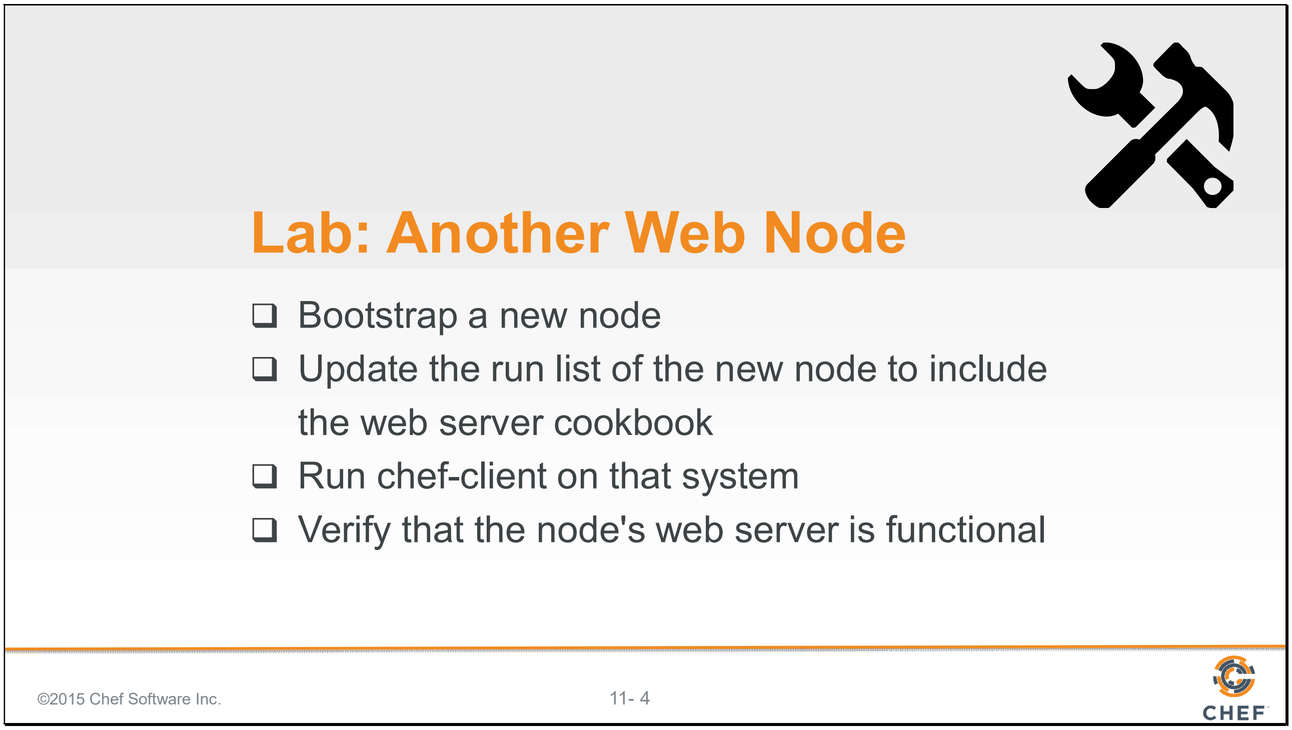
After completing this module, you will have configured three nodes:

Node 1: A web server

Node 2: The load balancer

Node 3: Another web server

Slide 4



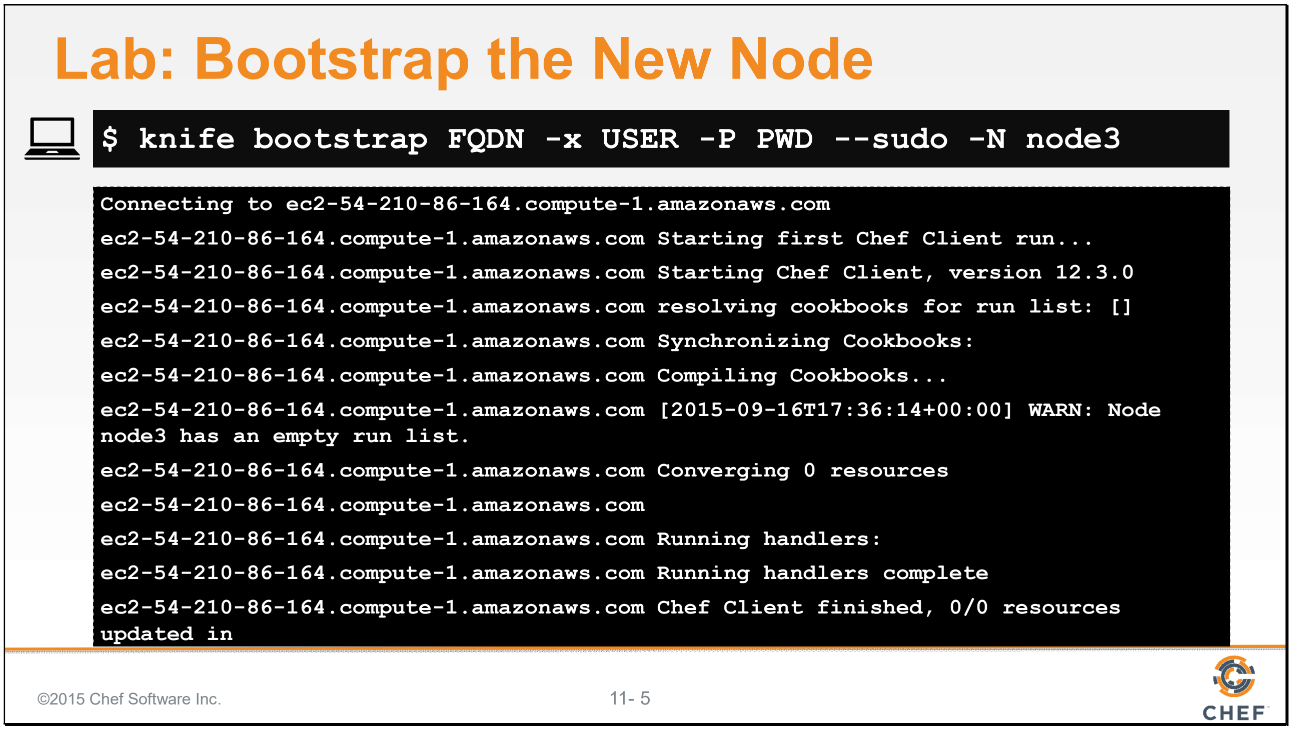
Lab: Another Web Node

- □ Bootstrap a new node
- □ Update the run list of the new node to include the web server cookbook
- □ Run chef-client on that system
- □ Verify that the node's web server is functional

Now it’s time to create a third node. The third node will be the second web server node.

We will provide you with a new node for the following exercise.

Slide 5



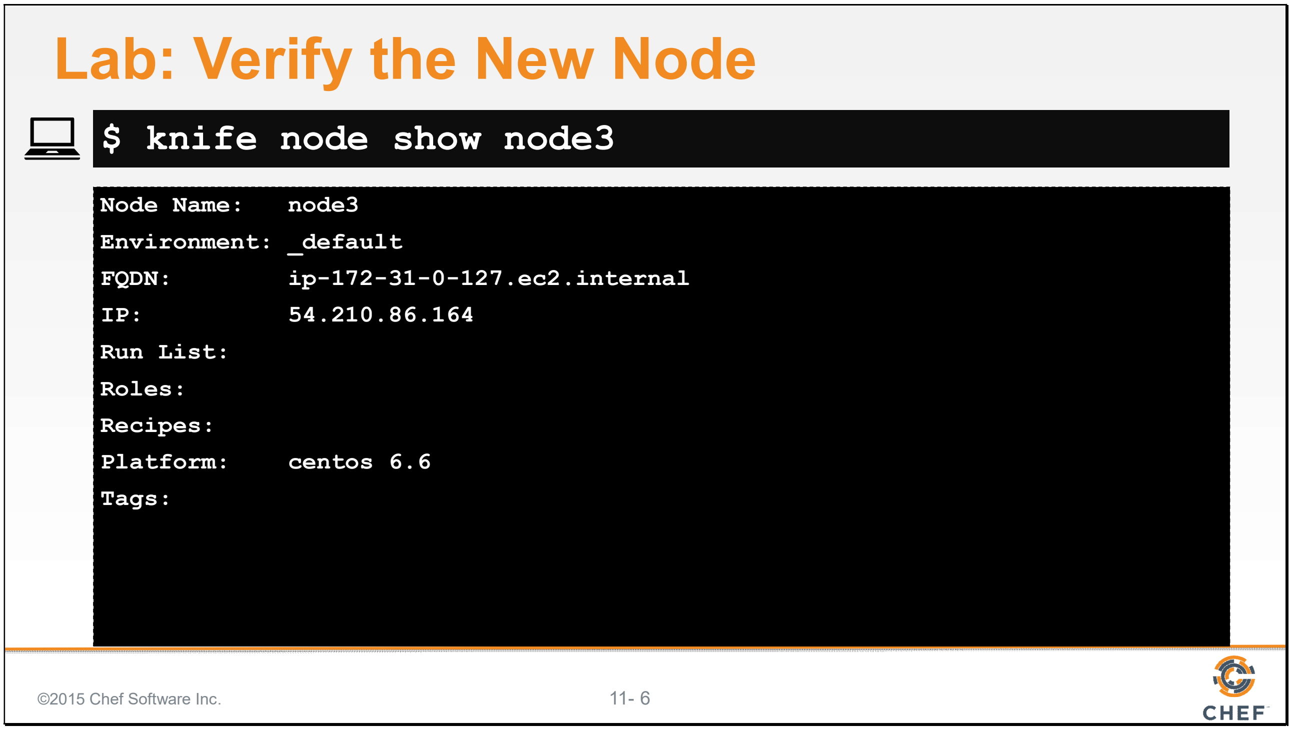
Lab: Bootstrap the New Node

```
$ knife bootstrap FQDN -x USER -P PWD --sudo -N node3
```

```
Connecting to ec2-54-210-86-164.compute-1.amazonaws.com
ec2-54-210-86-164.compute-1.amazonaws.com Starting first Chef Client run...
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list: []
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com [2015-09-16T17:36:14+00:00] WARN: Node
node3 has an empty run list.
ec2-54-210-86-164.compute-1.amazonaws.com Converging 0 resources
ec2-54-210-86-164.compute-1.amazonaws.com
ec2-54-210-86-164.compute-1.amazonaws.com Running handlers:
ec2-54-210-86-164.compute-1.amazonaws.com Running handlers complete
ec2-54-210-86-164.compute-1.amazonaws.com Chef Client finished, 0/0 resources
updated in
```

Bootstrap the new node and name it node3.

Slide 6



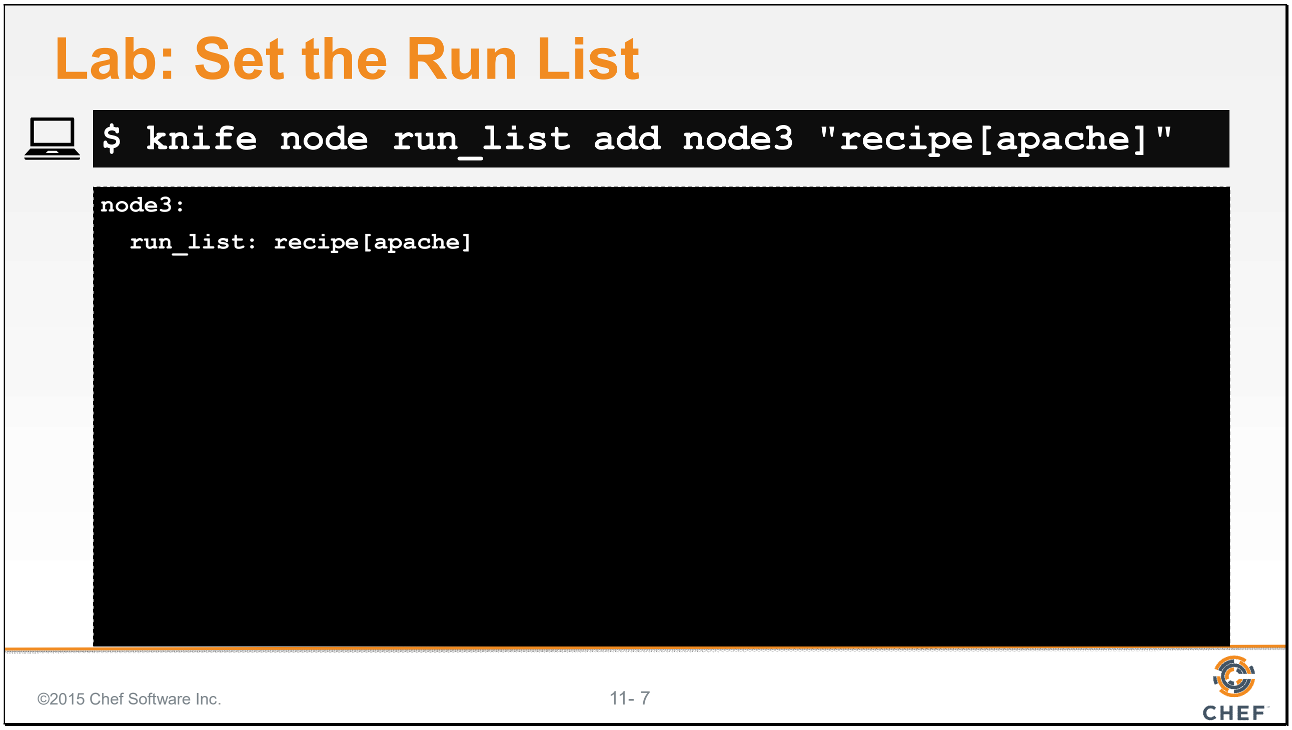
Lab: Verify the New Node

```
$ knife node show node3
```

```
Node Name:   node3
Environment: _default
FQDN:        ip-172-31-0-127.ec2.internal
IP:          54.210.86.164
Run List:
Roles:
Recipes:
Platform:    centos 6.6
Tags:
```

Verify that you bootstrapped the node.

Slide 7



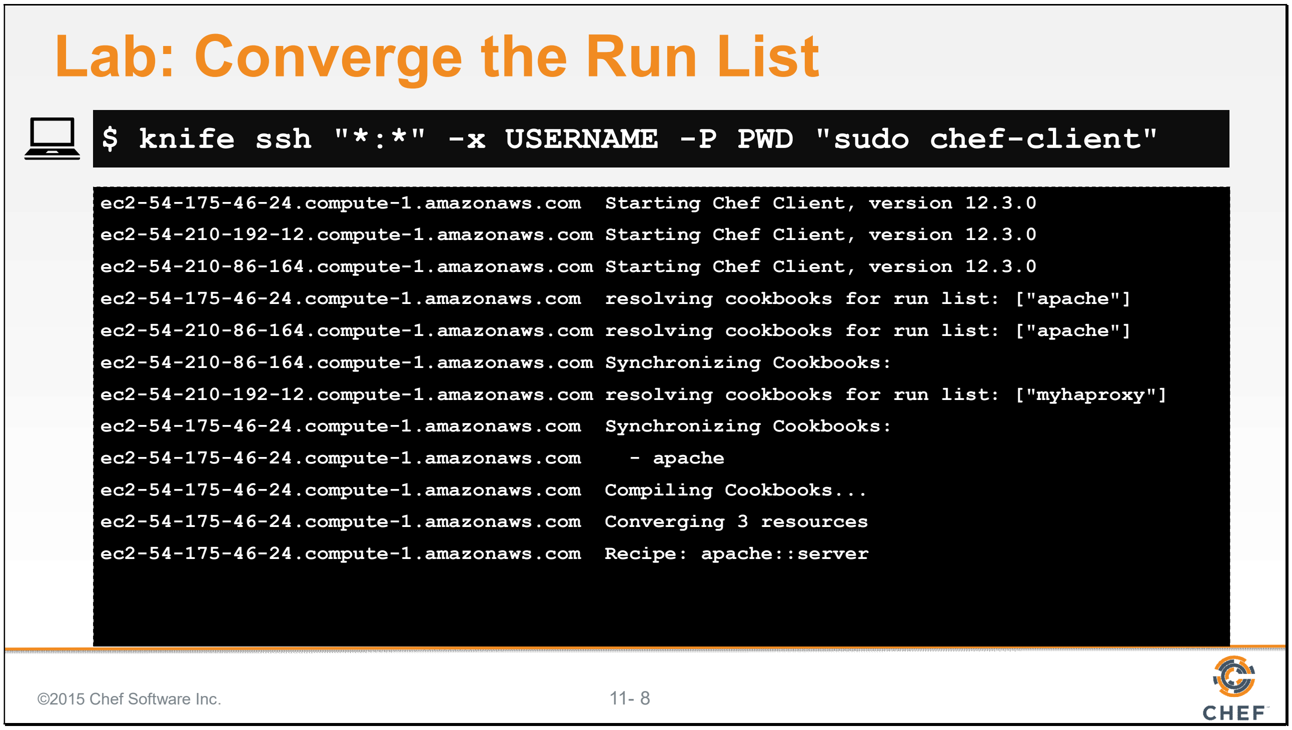
Lab: Set the Run List

```
$ knife node run_list add node3 "recipe[apache]"
```

```
node3:
  run_list: recipe[apache]
```

Set the run list for this node by running the apache cookbook's default recipe.

Slide 8



Lab: Converge the Run List

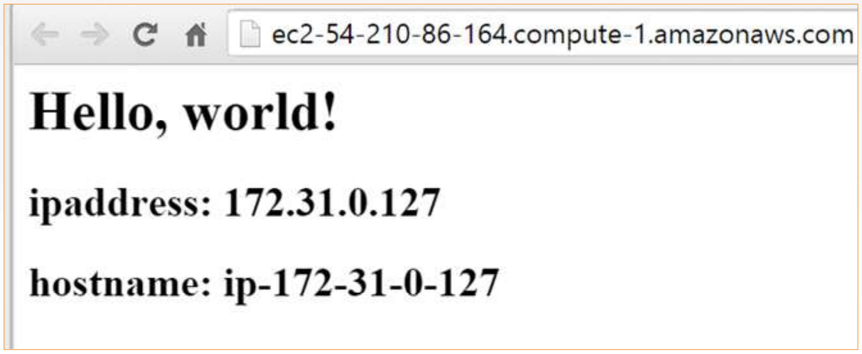
```
$ knife ssh "*:*" -x USERNAME -P PWD "sudo chef-client"
```

```
ec2-54-175-46-24.compute-1.amazonaws.com  Starting Chef Client, version 12.3.0
ec2-54-210-192-12.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list: ["apache"]
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list: ["apache"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-192-12.compute-1.amazonaws.com resolving cookbooks for run list: ["myhaproxy"]
ec2-54-175-46-24.compute-1.amazonaws.com  Synchronizing Cookbooks:
ec2-54-175-46-24.compute-1.amazonaws.com    - apache
ec2-54-175-46-24.compute-1.amazonaws.com  Compiling Cookbooks...
ec2-54-175-46-24.compute-1.amazonaws.com  Converging 3 resources
ec2-54-175-46-24.compute-1.amazonaws.com  Recipe: apache::server
```

Apply that run list by logging into that node and running sudo chef-client or remotely administer the node with the 'knife ssh' command as shown here.

Slide 9

Verify that the New Node Serves the Page



ec2-54-210-86-164.compute-1.amazonaws.com

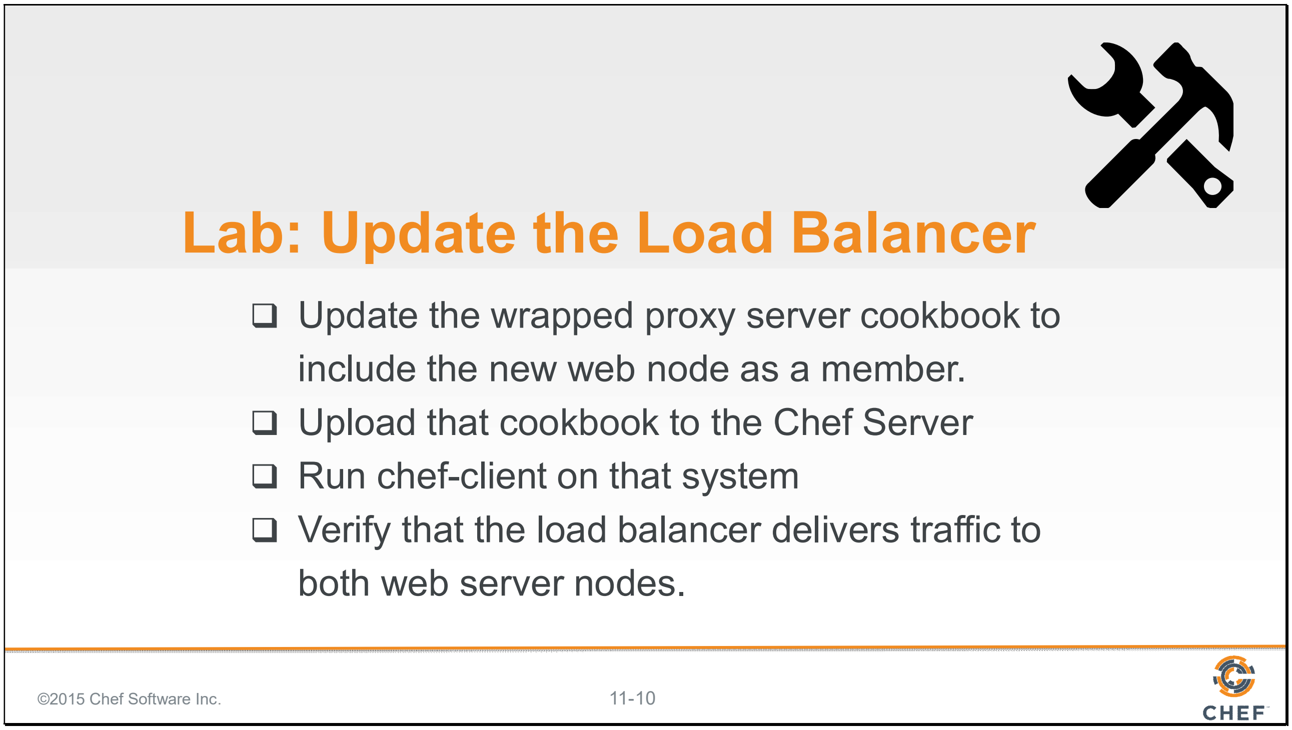
Hello, world!

ipaddress: 172.31.0.127

hostname: ip-172-31-0-127

Verify that the node serves up the default html page that contains the node's internal IP address and hostname.

Slide 10



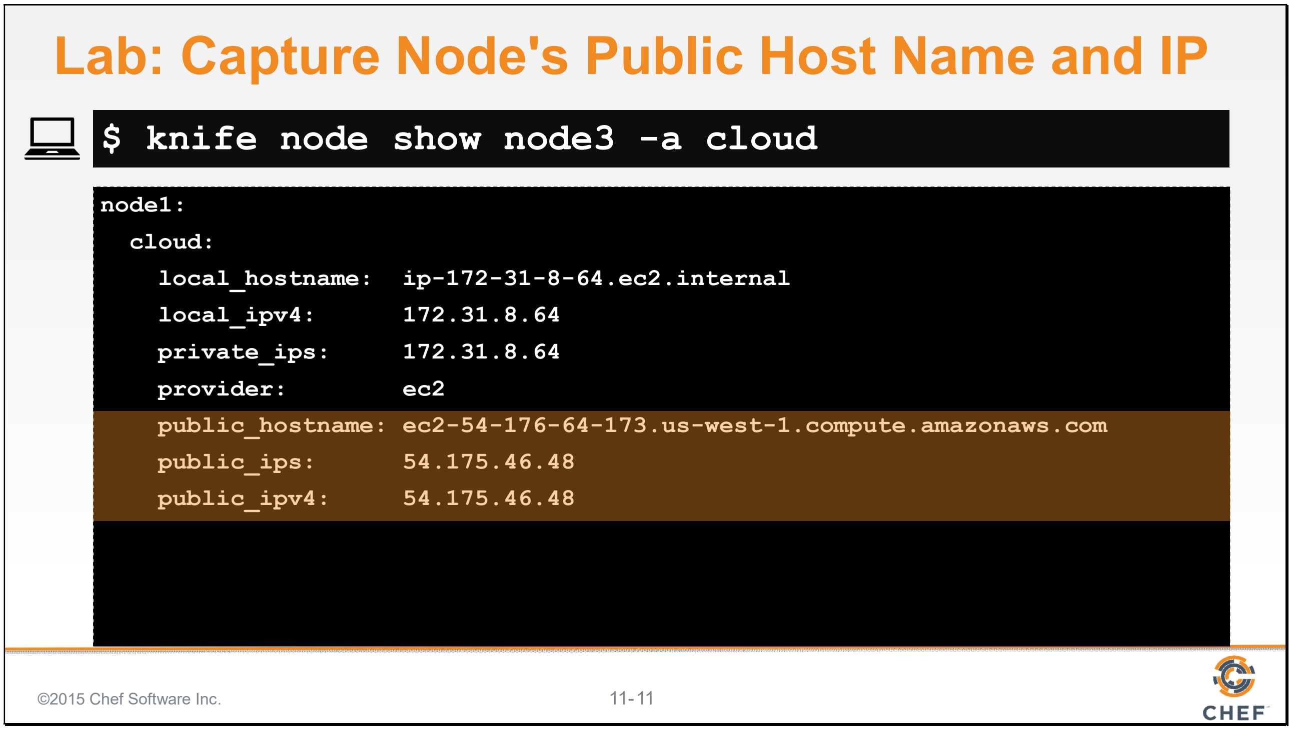
Lab: Update the Load Balancer

- □ 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.

CHEF

Now that you have the third node, it is time to add that node to the member's list for the load balancer.

Slide 11



Lab: Capture Node's Public Host Name and IP

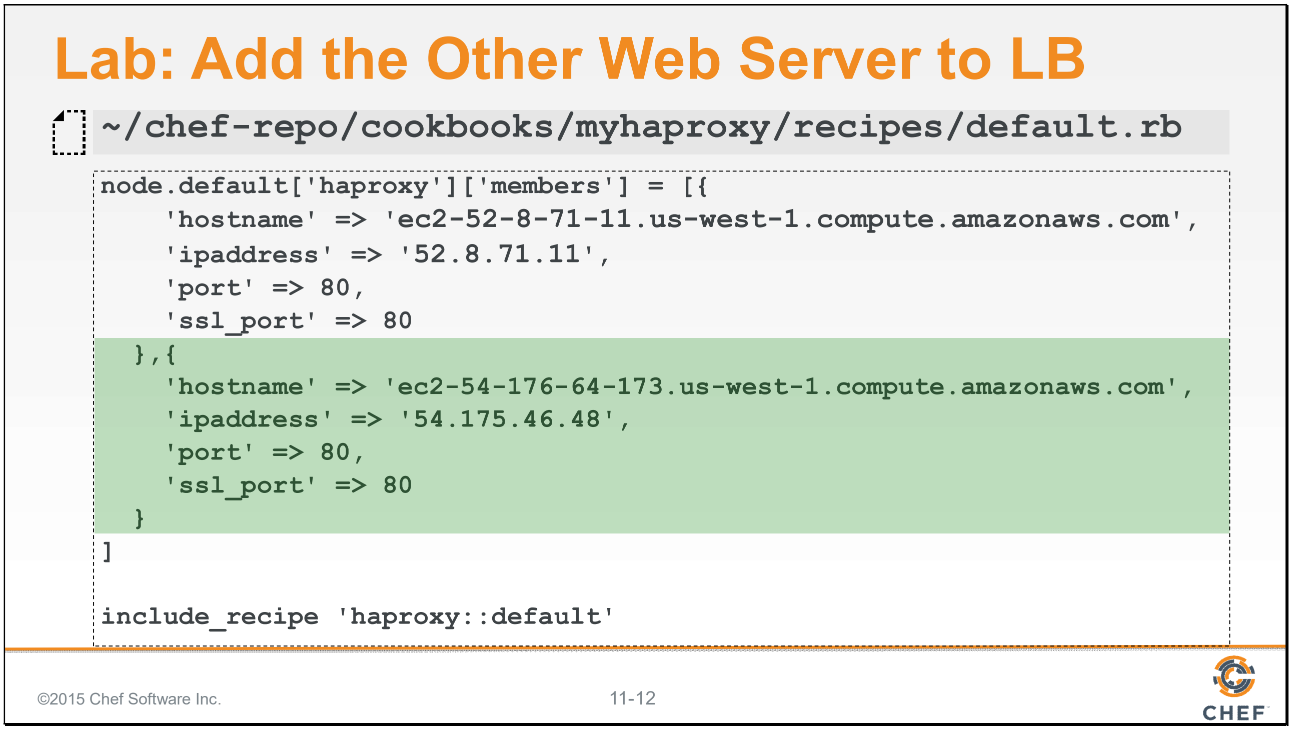
```
$ knife node show node3 -a cloud
```

```
node1:
  cloud:
    local_hostname:   ip-172-31-8-64.ec2.internal
    local_ipv4:       172.31.8.64
    private_ips:      172.31.8.64
    provider:         ec2
    public_hostname:  ec2-54-176-64-173.us-west-1.compute.amazonaws.com
    public_ips:       54.175.46.48
    public_ipv4:      54.175.46.48
```

If you use 'knife node show' to display the 'cloud' attribute for node3, you will see the local, private, and public connection information.

Capture and write down the public hostname and the public ipv4 address of node3. You will need this in the recipe you are going to write.

Slide 12



Lab: Add the Other Web Server to LB

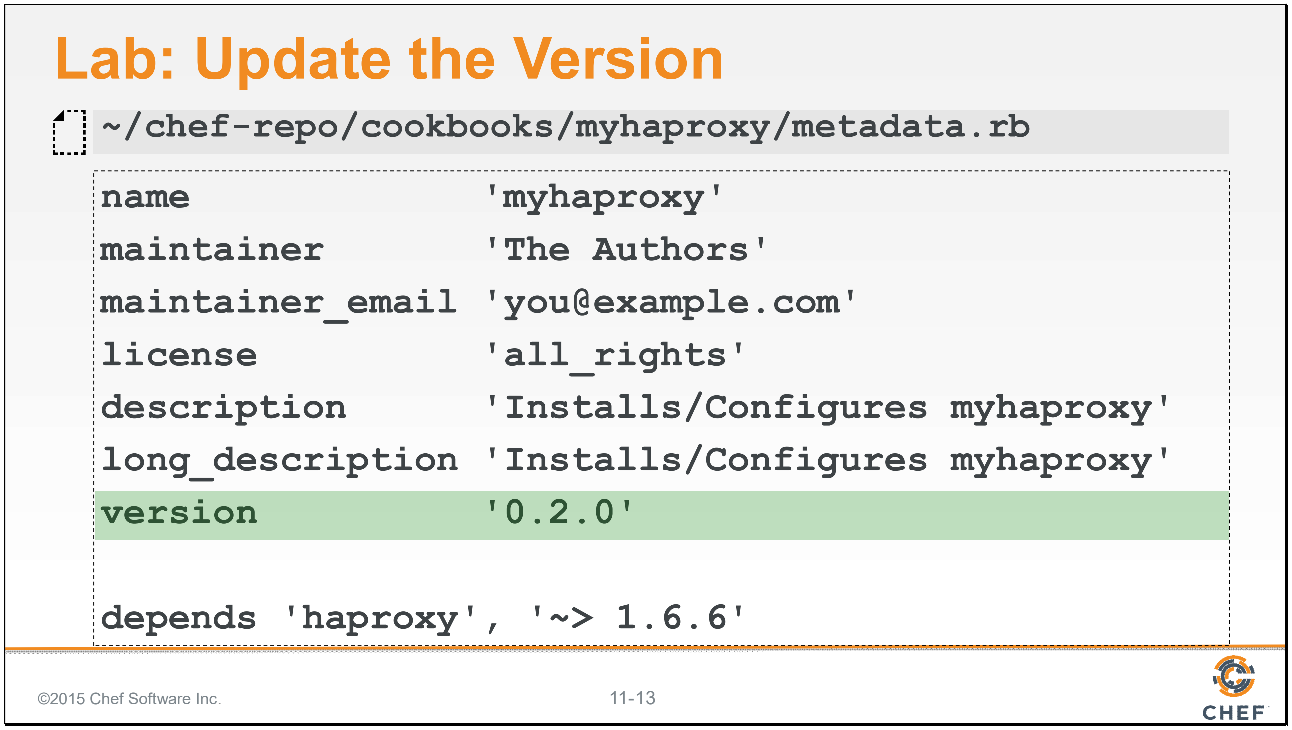
~/chef-repo/cookbooks/myhaproxy/recipes/default.rb

```
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
  },{
    'hostname' => 'ec2-54-176-64-173.us-west-1.compute.amazonaws.com',
    'ipaddress' => '54.175.46.48',
    'port' => 80,
    'ssl_port' => 80
  }
]

include_recipe 'haproxy::default'
```

Add the second web server (node3) to the load balancer's (LB) members list. You may need to run 'knife node show node3 -a cloud' to get the hostname and ipaddress values.

Slide 13



Lab: Update the Version

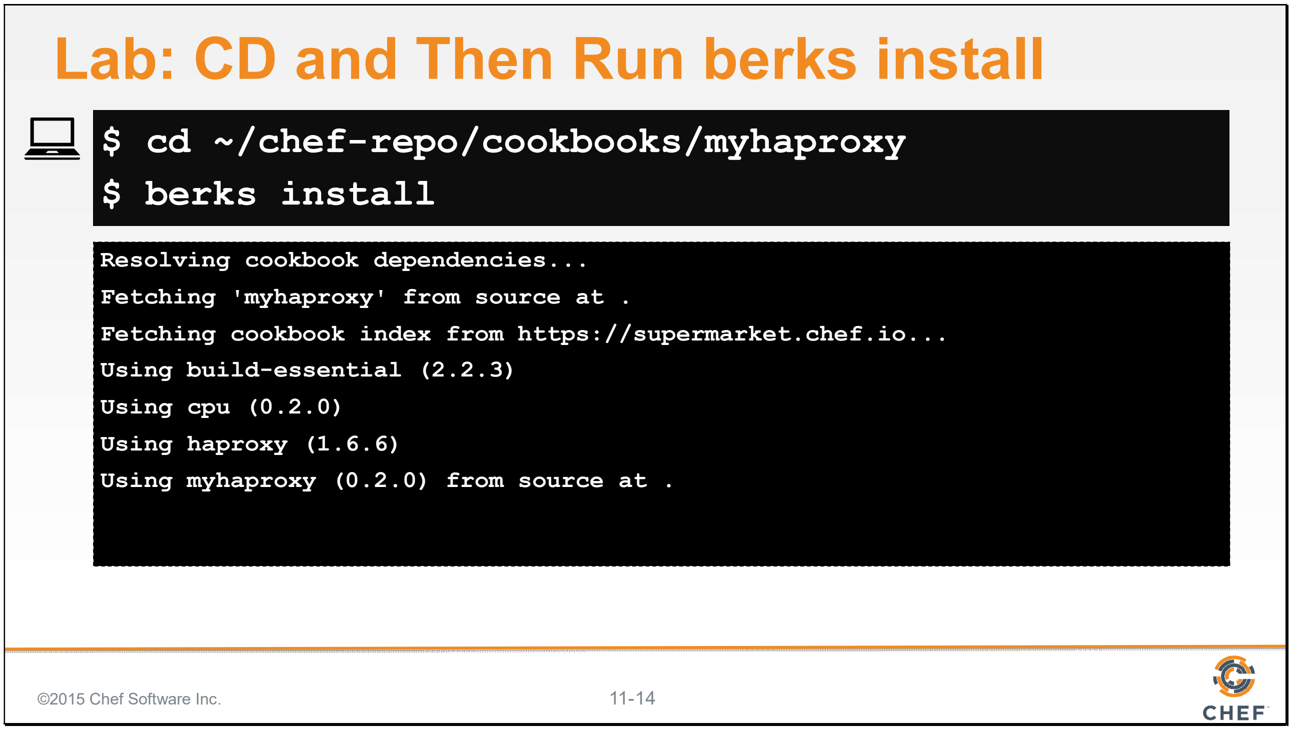
~/chef-repo/cookbooks/myhaproxy/metadata.rb

```
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', '~> 1.6.6'
```

Update the version number in myhaproxy cookbook's metadata.

Slide 14



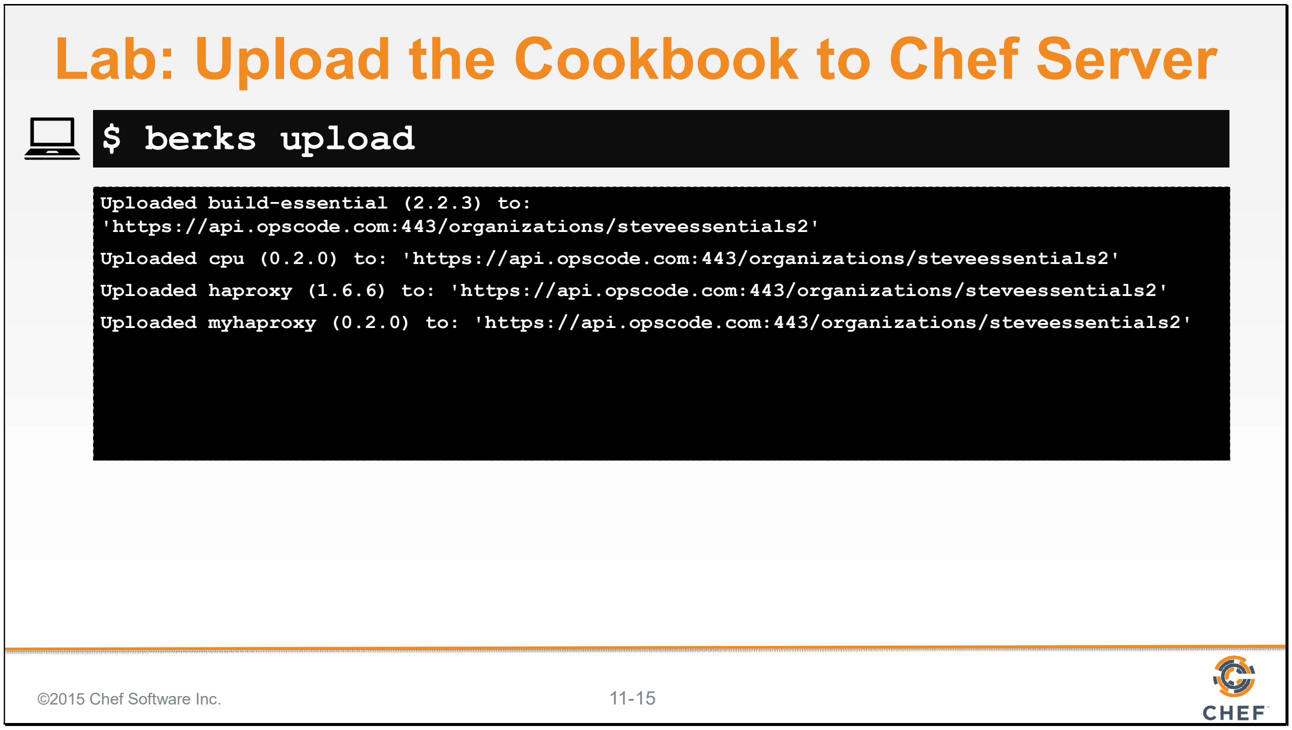
Lab: CD and Then Run berks install

```
$ 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 (1.6.6)
Using myhaproxy (0.2.0) from source at .
```

CHEF

Change into the 'myhaproxy' cookbook directory and then run 'berks install' to install any dependencies for the 'myhaproxy' cookbook.



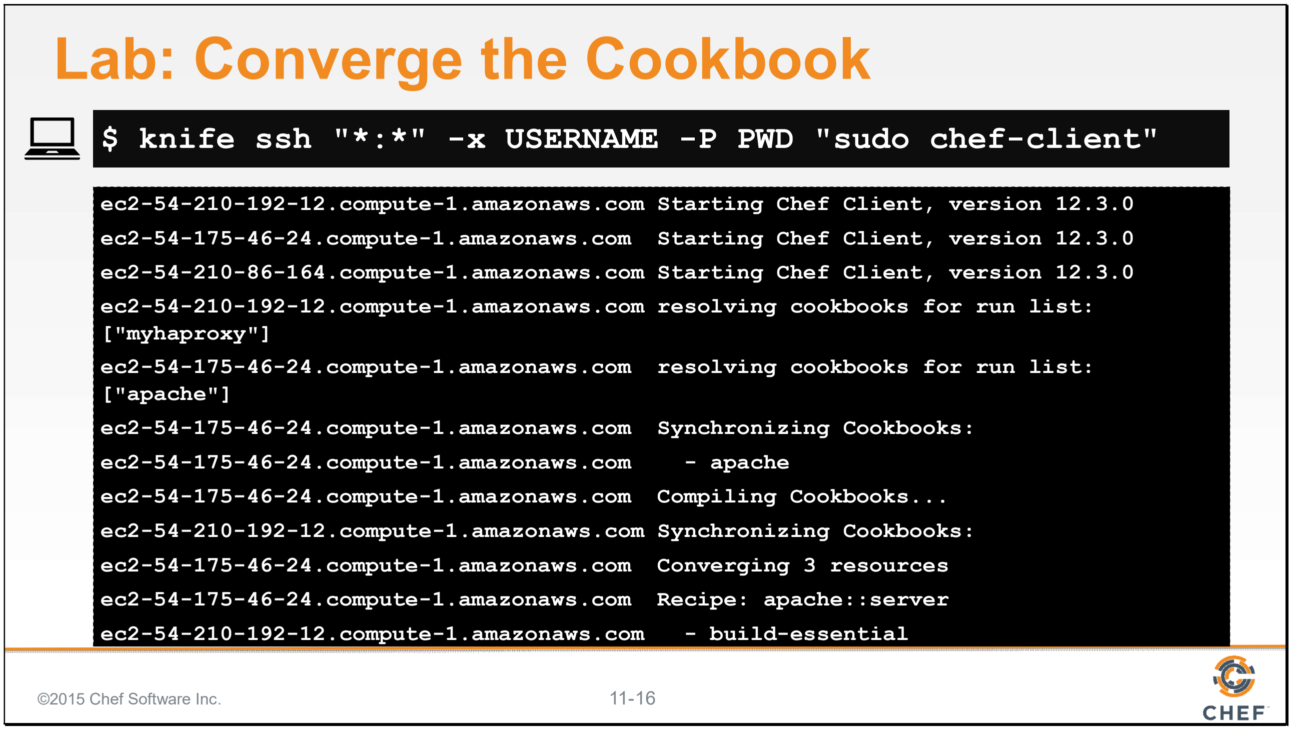
Lab: Upload the Cookbook to Chef Server

```
$ berks upload
```

```
Uploaded build-essential (2.2.3) to:
'https://api.opscode.com:443/organizations/steveessentials2'
Uploaded cpu (0.2.0) to: 'https://api.opscode.com:443/organizations/steveessentials2'
Uploaded haproxy (1.6.6) to: 'https://api.opscode.com:443/organizations/steveessentials2'
Uploaded myhaproxy (0.2.0) to: 'https://api.opscode.com:443/organizations/steveessentials2'
```

Run 'berks upload' to upload the myhaproxy cookbook to Chef Server.

Slide 16



Lab: Converge the Cookbook

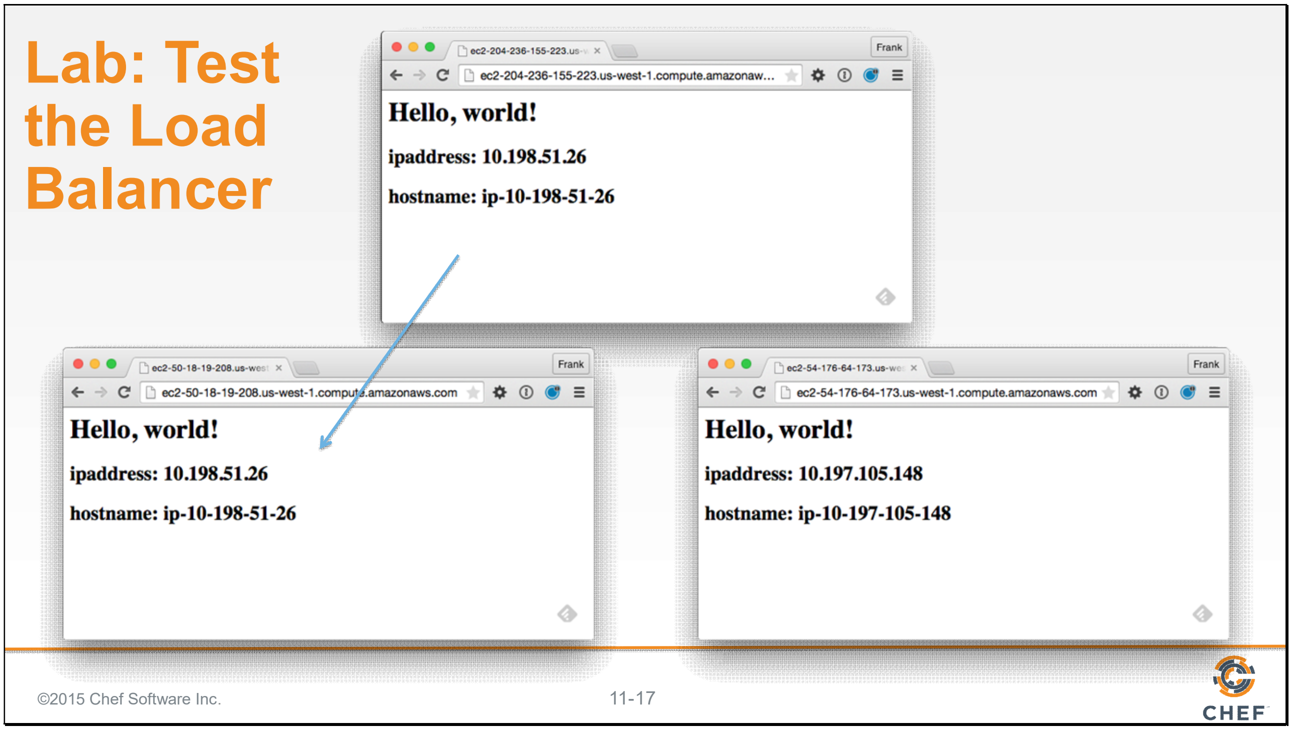
```
$ knife ssh "*:*" -x USERNAME -P PWD "sudo chef-client"
```

```
ec2-54-210-192-12.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-175-46-24.compute-1.amazonaws.com  Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-192-12.compute-1.amazonaws.com resolving cookbooks for run list:
["myhaproxy"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list:
["apache"]
ec2-54-175-46-24.compute-1.amazonaws.com  Synchronizing Cookbooks:
ec2-54-175-46-24.compute-1.amazonaws.com    - apache
ec2-54-175-46-24.compute-1.amazonaws.com  Compiling Cookbooks...
ec2-54-210-192-12.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-175-46-24.compute-1.amazonaws.com  Converging 3 resources
ec2-54-175-46-24.compute-1.amazonaws.com  Recipe: apache::server
ec2-54-210-192-12.compute-1.amazonaws.com   - build-essential
```

Converge the cookbook by logging into that node and running 'sudo chef-client' or remotely administer the node with the 'knife ssh' command as shown here.

Within the output you should see the haproxy configuration file will update with a new entry that contains the information of the second member (node3).

Slide 17



ec2-204-236-155-223.us-west-1.compute.amazonaw...
Frank
ec2-50-18-19-208.us-west-1.compute.amazonaws.com
Frank
ec2-54-176-64-173.us-west-1.compute.amazonaws.com
Frank
CHEF

Lab: Test the Load Balancer

Hello, world!

ipaddress: 10.198.51.26

hostname: ip-10-198-51-26

Hello, world!

ipaddress: 10.198.51.26

hostname: ip-10-198-51-26

Hello, world!

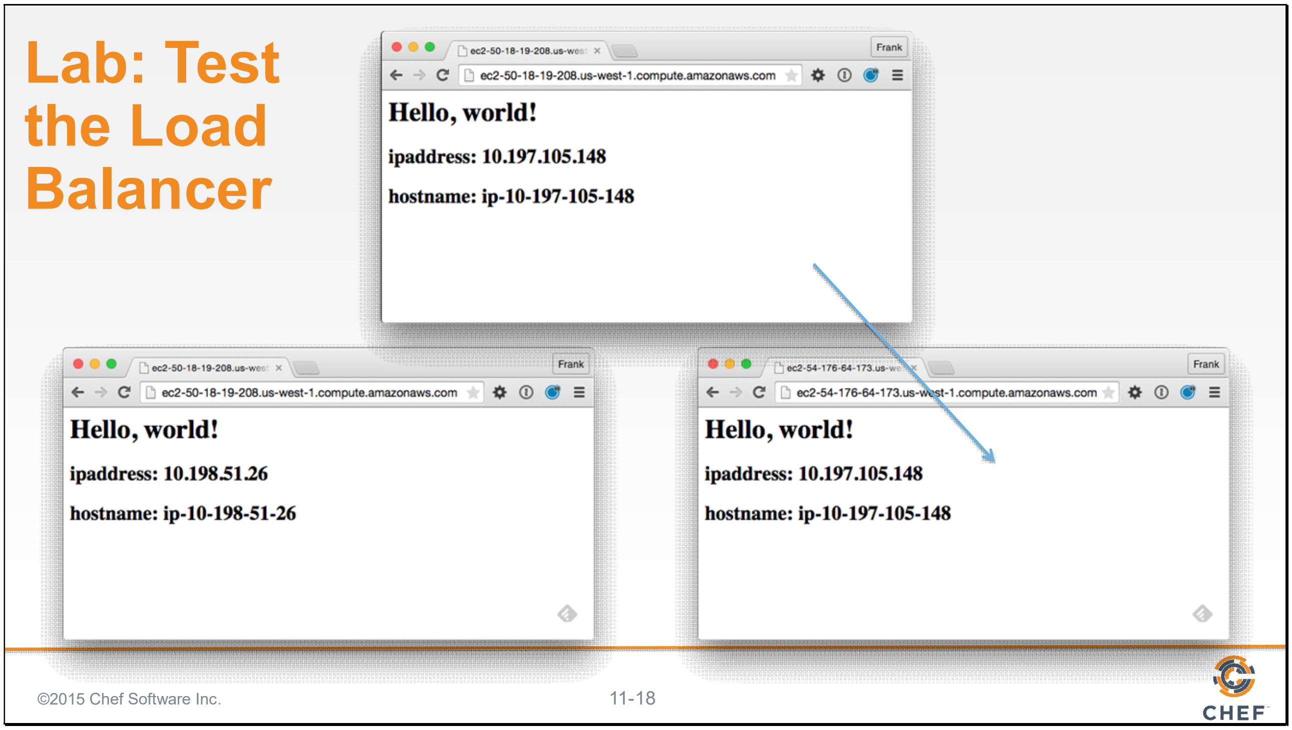
ipaddress: 10.197.105.148

hostname: ip-10-197-105-148

Point a web browser to the URL of your Proxy server and then click Refresh a few times. You should see each of your web server's HTML page as the Proxy server switches between each web server.

This is not a very scientific way of seeing that the proxy server is balancing requests between these two web nodes.

Slide 18


ec2-50-18-19-208.us-west
Frank
ec2-50-18-19-208.us-west-1.compute.amazonaws.com
ec2-54-176-64-173.us-we
ec2-54-176-64-173.us-west-1.compute.amazonaws.com
CHEF

Lab: Test the Load Balancer

Hello, world!

ipaddress: 10.197.105.148

hostname: ip-10-197-105-148

Hello, world!

ipaddress: 10.198.51.26

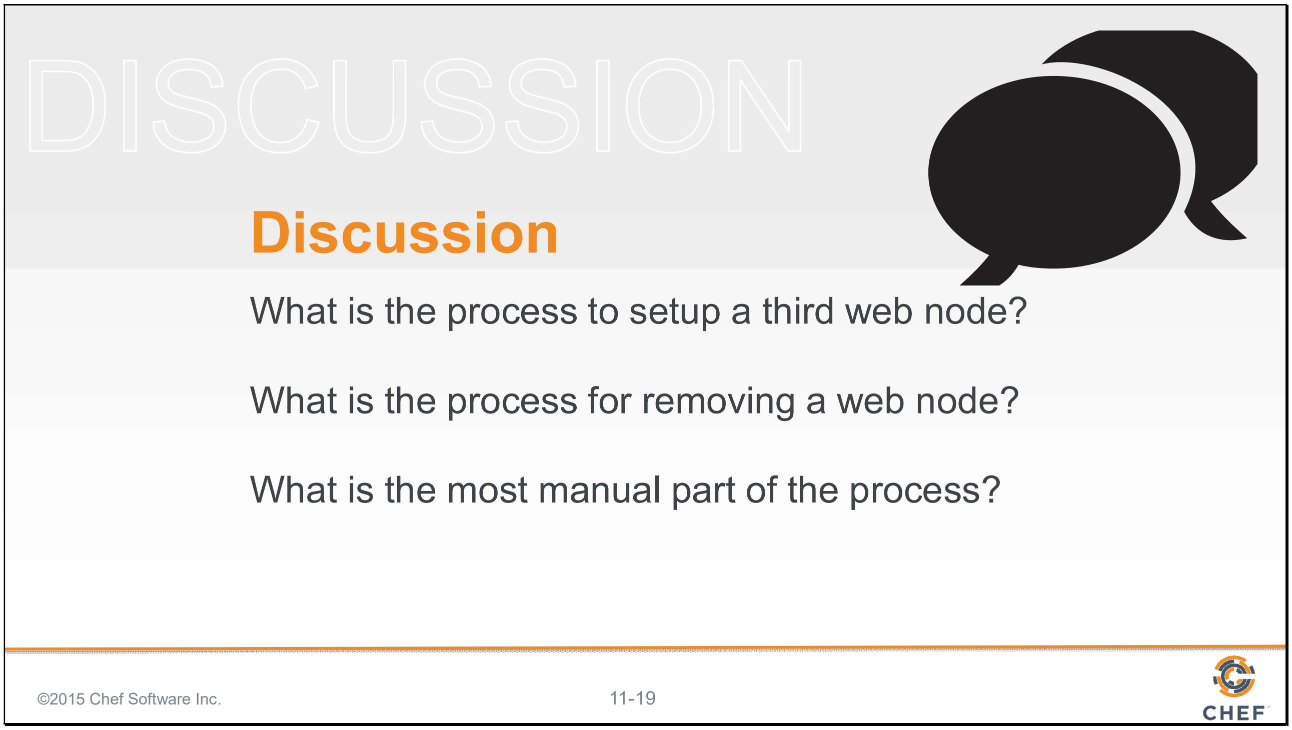
hostname: ip-10-198-51-26

Hello, world!

ipaddress: 10.197.105.148

hostname: ip-10-197-105-148

Slide 19



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?

Answer these questions.

With your answers, turn to another person and alternate asking each other asking these questions and sharing your answers.

Slide 20

DISCUSSION

Q&A

What questions can we help you answer?

CHEF

Slide 21

CHEF™