Using Policy Groups to Reflect Environments

Separating your nodes with policy_group

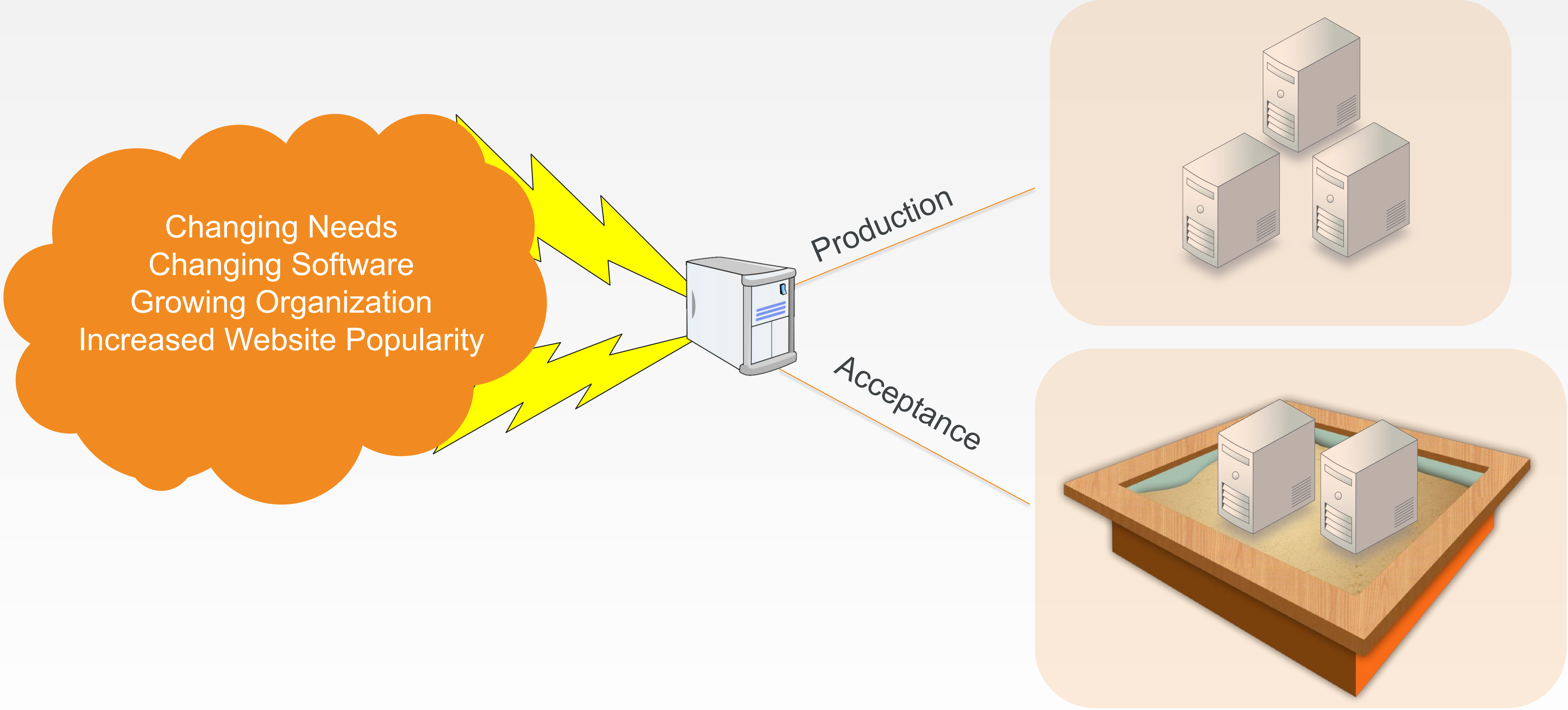
CHEF

Objectives

After completing this module, you should be able to

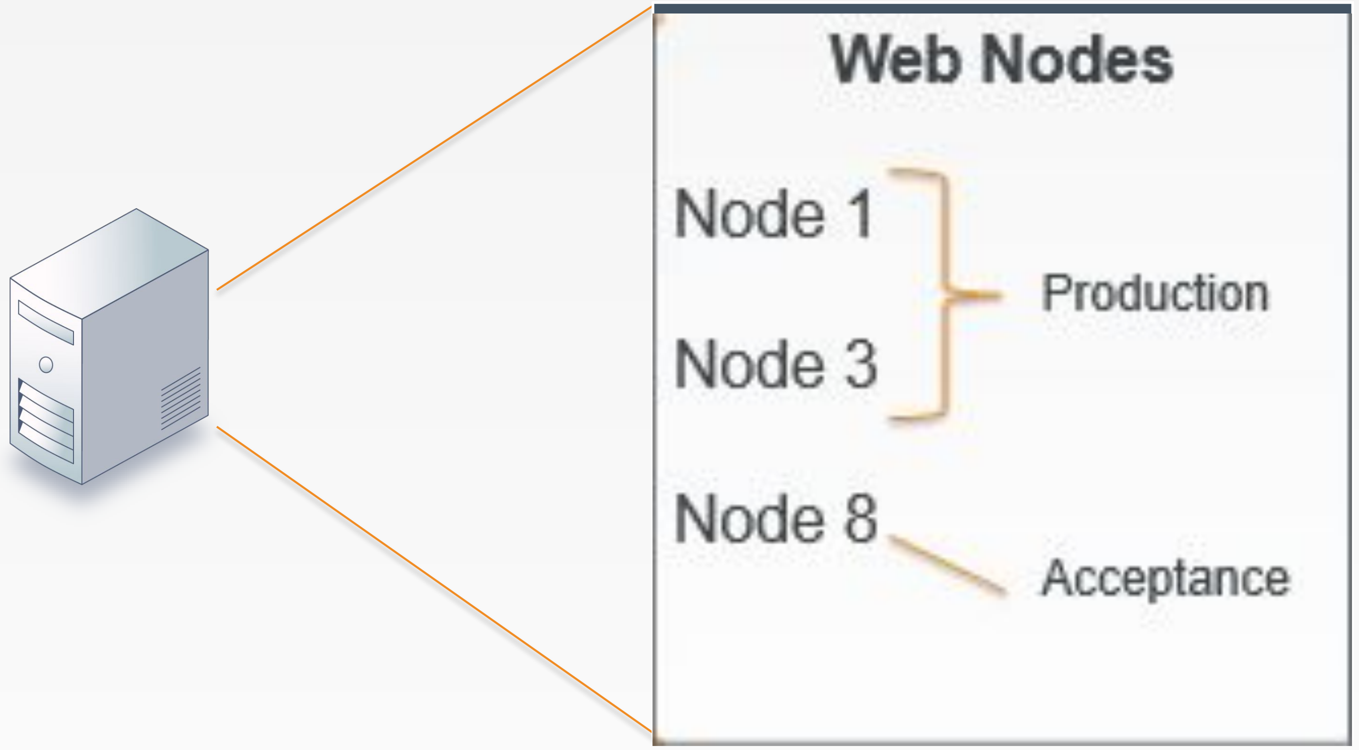
- Deploy a node to an environment via policy_group
- Update the load balancer's search query
- Test your load balancer to confirm that policy_group is separating your node from a group of nodes.

Keeping Your Infrastructure Current

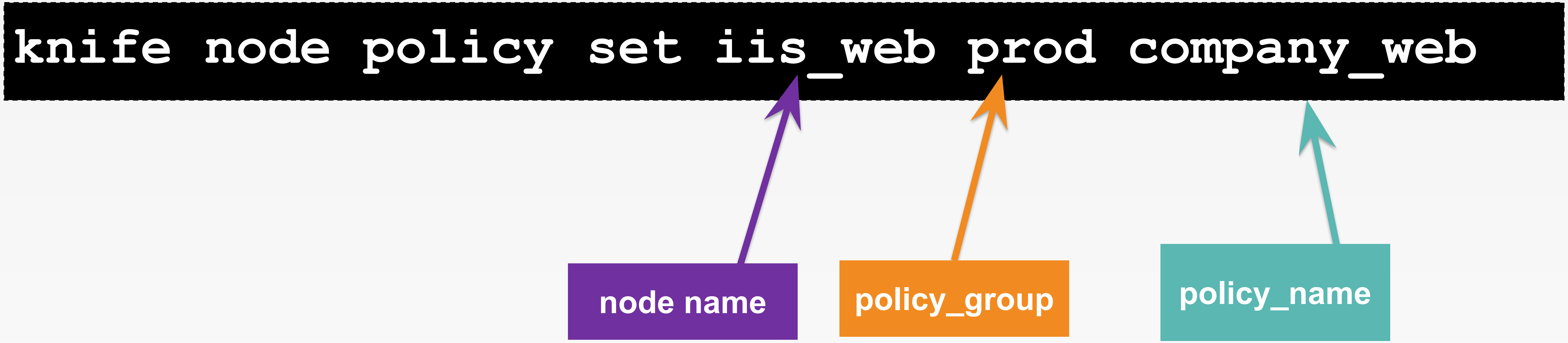


policy_group Environments

Environments can define different functions of nodes that live in the same infrastructure



Assigning a Node to an Environment

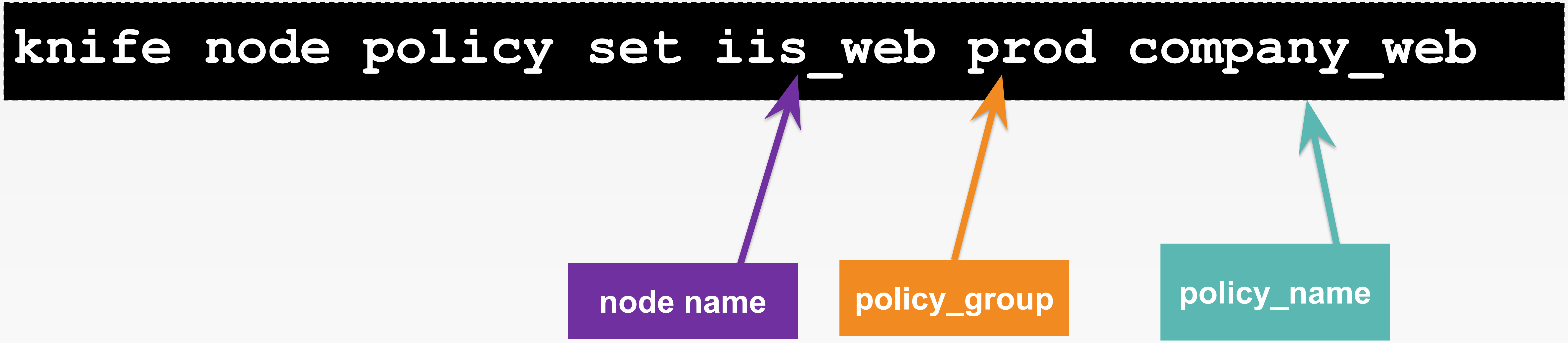


```
knife node policy set iis_web prod company_web
```

Assigning a node to an environment is as simple as specifying a policy_group in the `knife node policy set...` command.

In this example we assigned the **iis_web** node to the prod (production) environment. In this module, you will move your **iis_web** node to a new environment called **acceptance** and see the results.

Assigning a Node to an Environment



```
knife node policy set iis_web prod company_web
```

node name

policy_group

policy_name

We will leave our load balancer and our **apache_web** node in the **prod** environment so the load balancer will serve up only nodes in the **prod** environment.

Reminder: The first time you specify a policy group, that policy group name will be instantiated in Chef Infra Server. Then you can reuse it for other nodes.

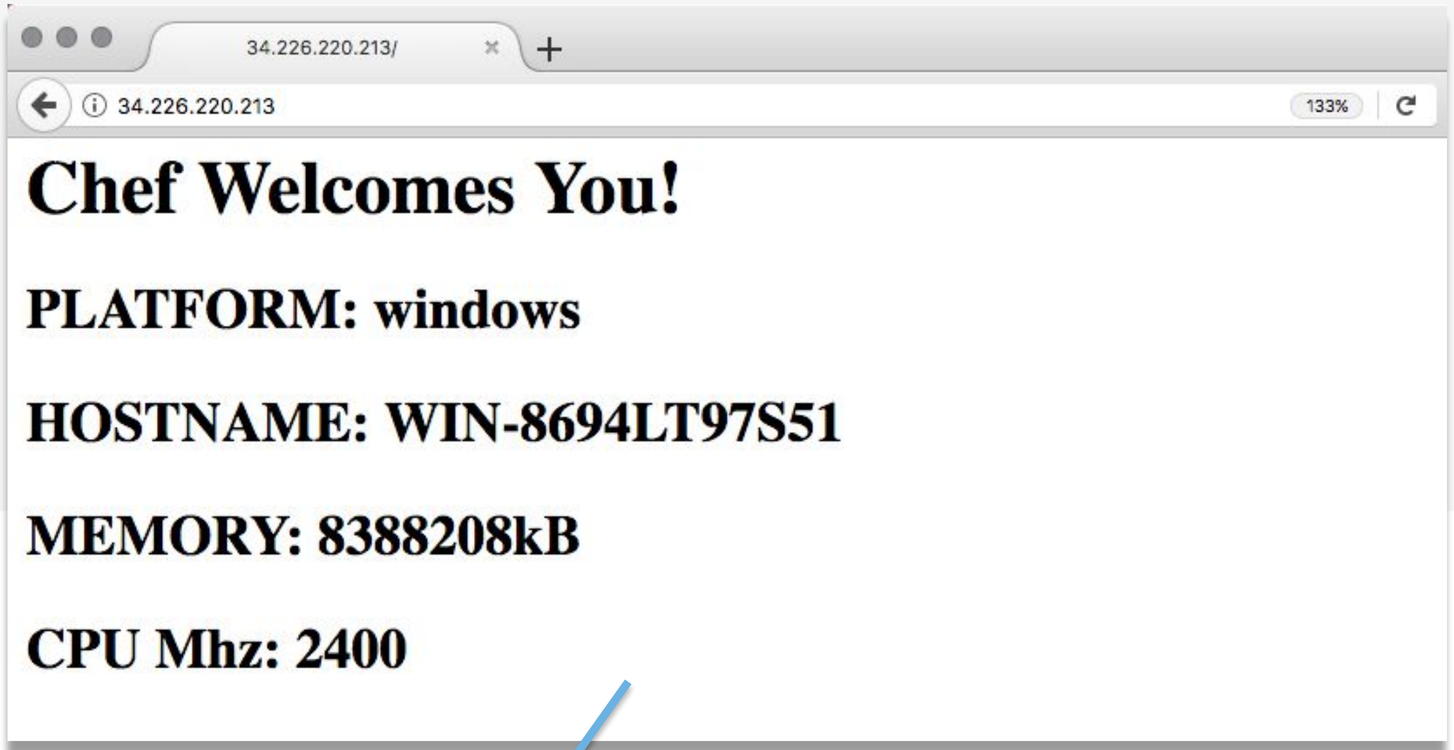
Group Lab: Using policy_group

Let's create an acceptance policy_group environment for our nodes

Objective:

- □ Test your current load balancer's behavior
- □ Assign the **iis_web** node to acceptance
- □ Update the load balancer's search criteria to **exclude** nodes in 'acceptance'
- □ Converge the load balancer node
- □ Test your load balancer

GL: Test the Load Balancer



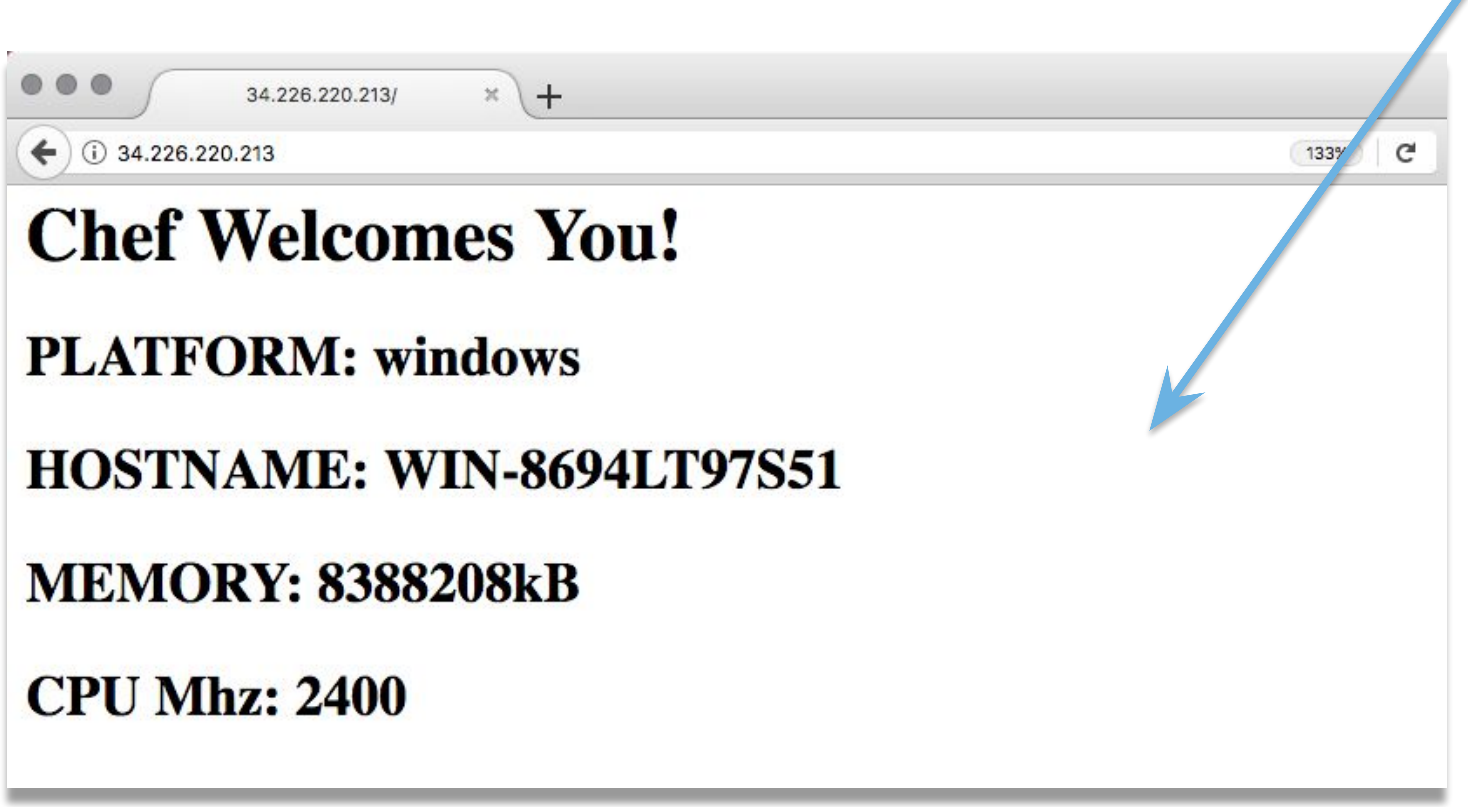
Chef Welcomes You!

PLATFORM: windows

HOSTNAME: WIN-8694LT97S51

MEMORY: 8388208kB

CPU Mhz: 2400



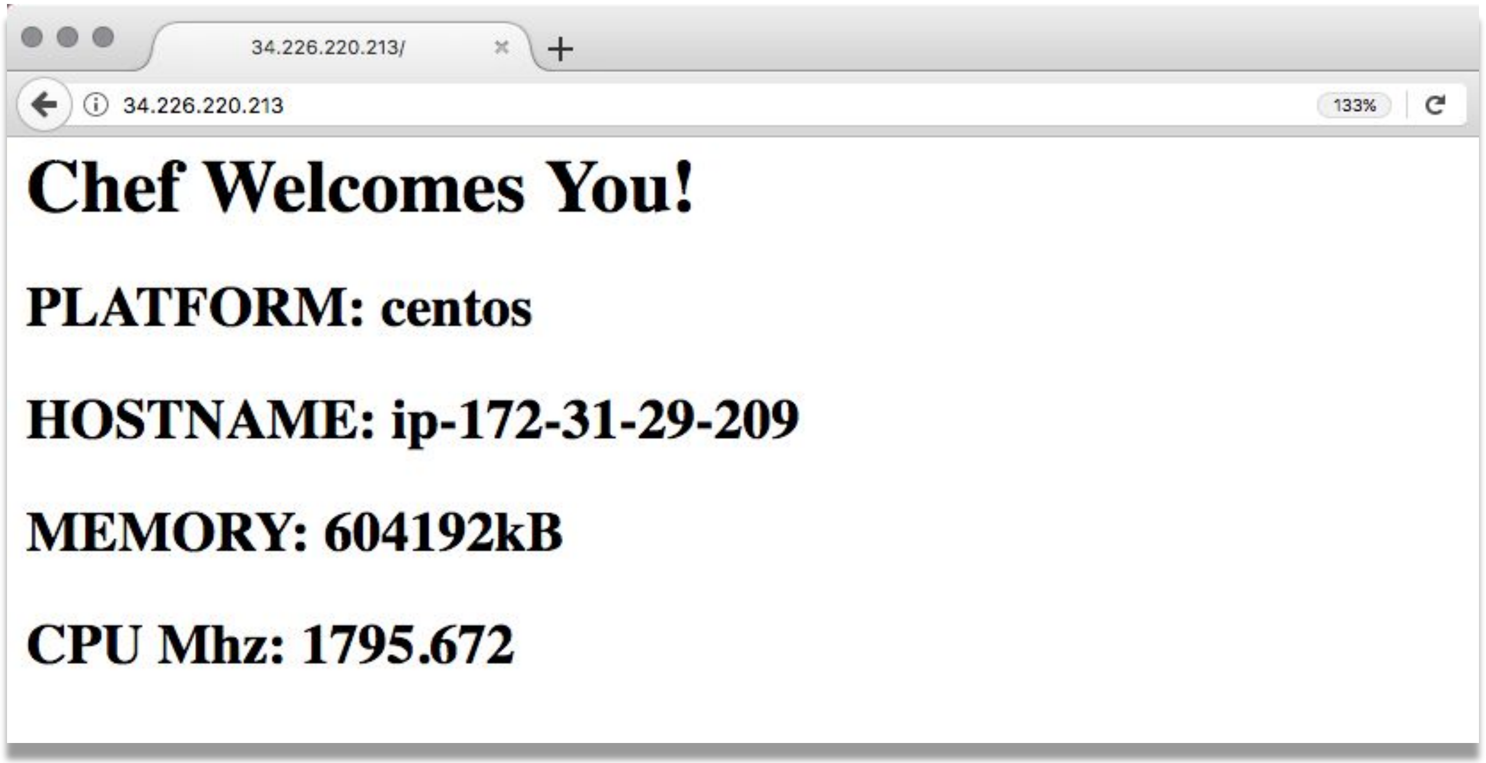
Chef Welcomes You!

PLATFORM: windows

HOSTNAME: WIN-8694LT97S51

MEMORY: 8388208kB

CPU Mhz: 2400



Chef Welcomes You!

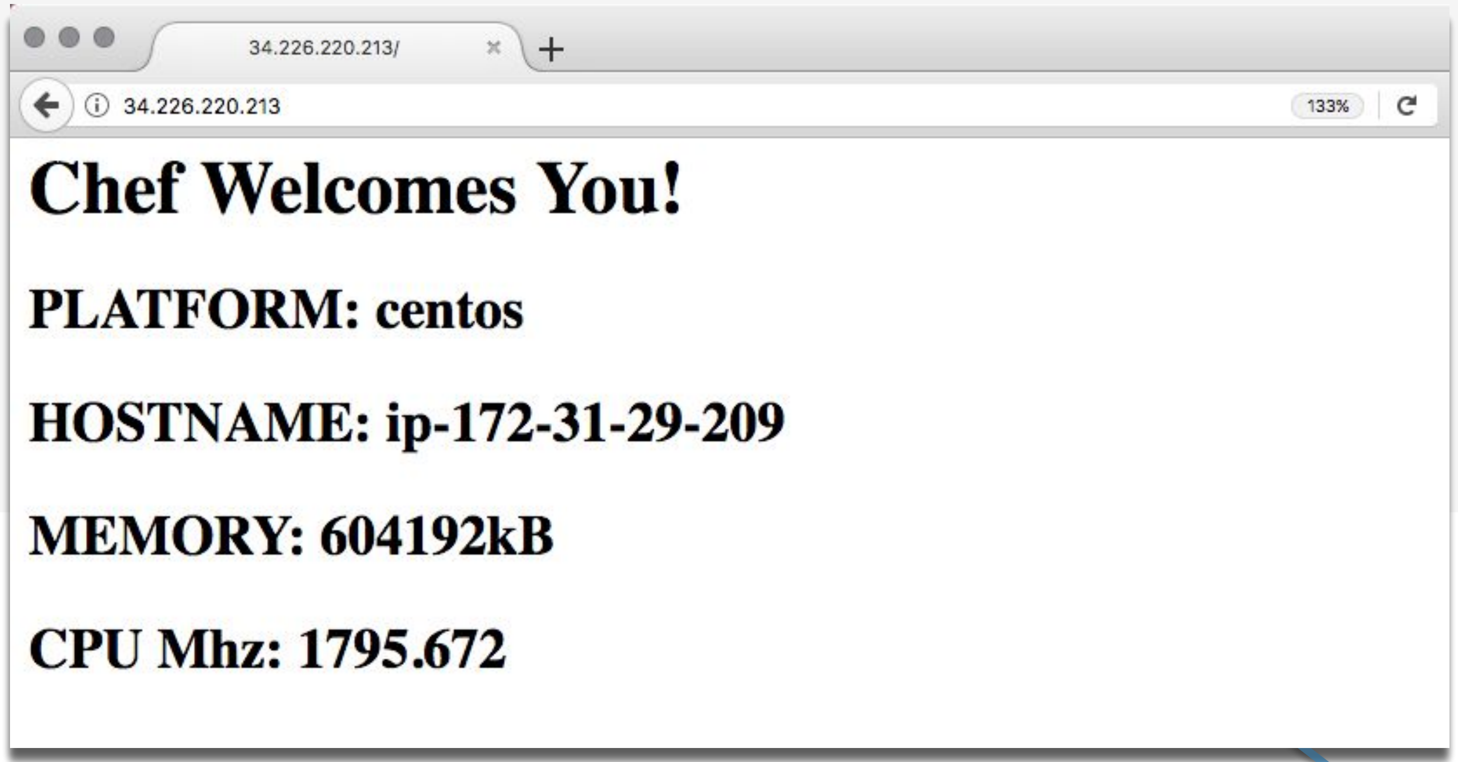
PLATFORM: centos

HOSTNAME: ip-172-31-29-209

MEMORY: 604192kB

CPU Mhz: 1795.672

GL: Test the Load Balancer



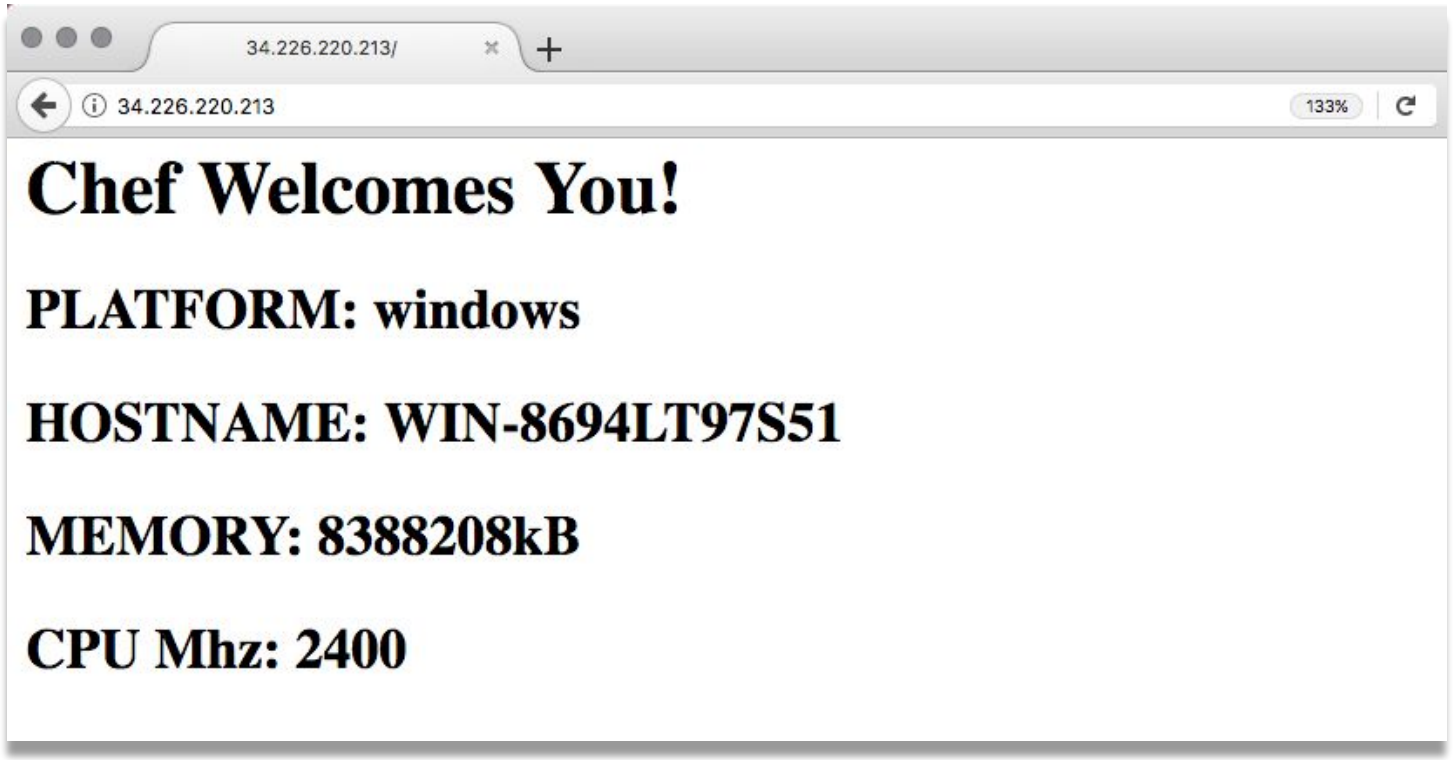
Chef Welcomes You!

PLATFORM: centos

HOSTNAME: ip-172-31-29-209

MEMORY: 604192kB

CPU Mhz: 1795.672



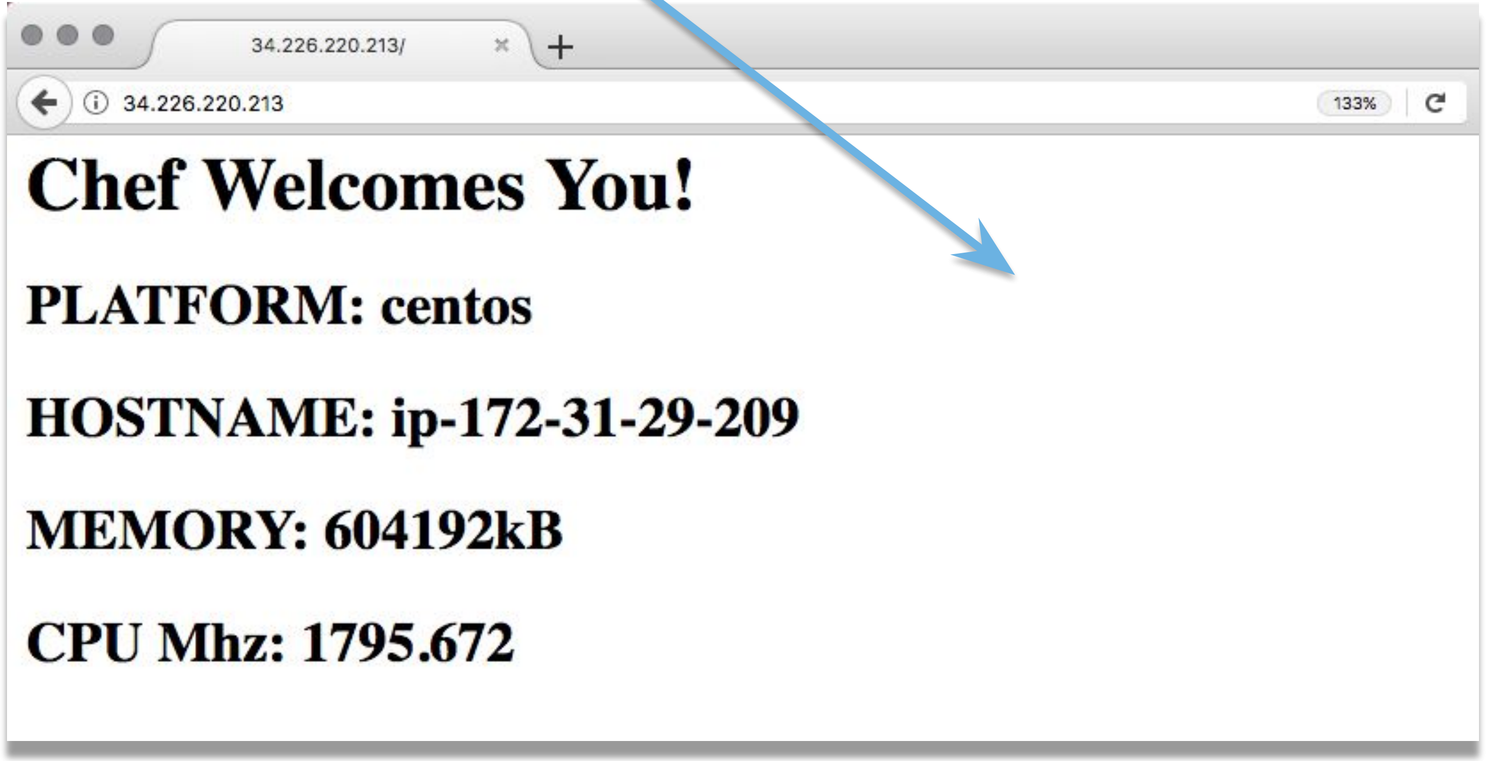
Chef Welcomes You!

PLATFORM: windows

HOSTNAME: WIN-8694LT97S51

MEMORY: 8388208kB

CPU Mhz: 2400



Chef Welcomes You!

PLATFORM: centos

HOSTNAME: ip-172-31-29-209

MEMORY: 604192kB

CPU Mhz: 1795.672

GL: Push the company_web.lock.json to a new acceptance Environment

```
~/chef-repo> chef push acceptance policyfiles/company_web.lock.json
```

```
Uploading policy company_web (1e97a11553) to policy group acceptance
Using     apache        0.1.0  (1388ab3a)
Using     chef-client   12.3.4 (7cb128f1)
Using     company_web   0.1.0  (085c5742)
Using     cron          7.0.3  (602e43b3)
Using     logrotate     3.0.4  (bd20a5c5)
Using     mychef_client 0.1.0  (f79fa661)
Using     myiis         0.2.1  (c7630da4)
```

New policy_group

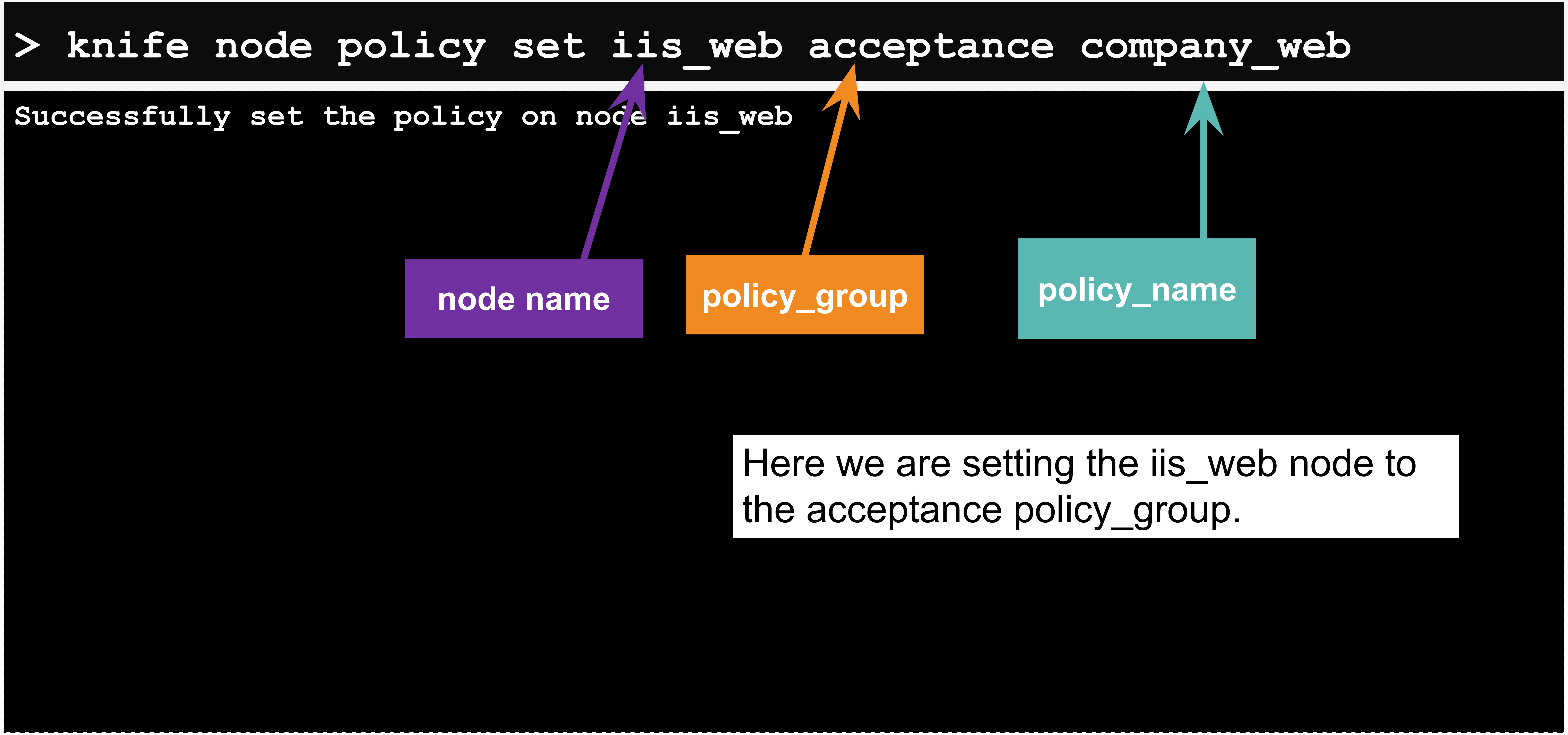
GL: Show the Policies on Chef Infra Server

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

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

* acceptance:   1e97a11553
* prod:         1e97a11553

myhaproxy
=========

* acceptance:   *NOT APPLIED*
* prod:         e633695adc
```

Here we can see that the **company_web** policy has been uploaded to Chef Infra Server and is in the **acceptance** policy_group.

GL: Assign the iis_web Node to acceptance

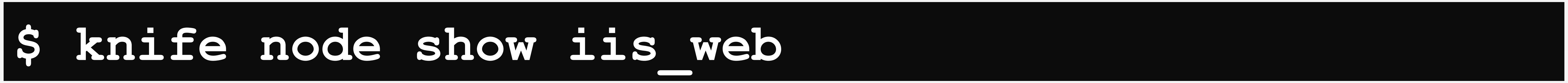


```
> knife node policy set iis_web acceptance company_web
```

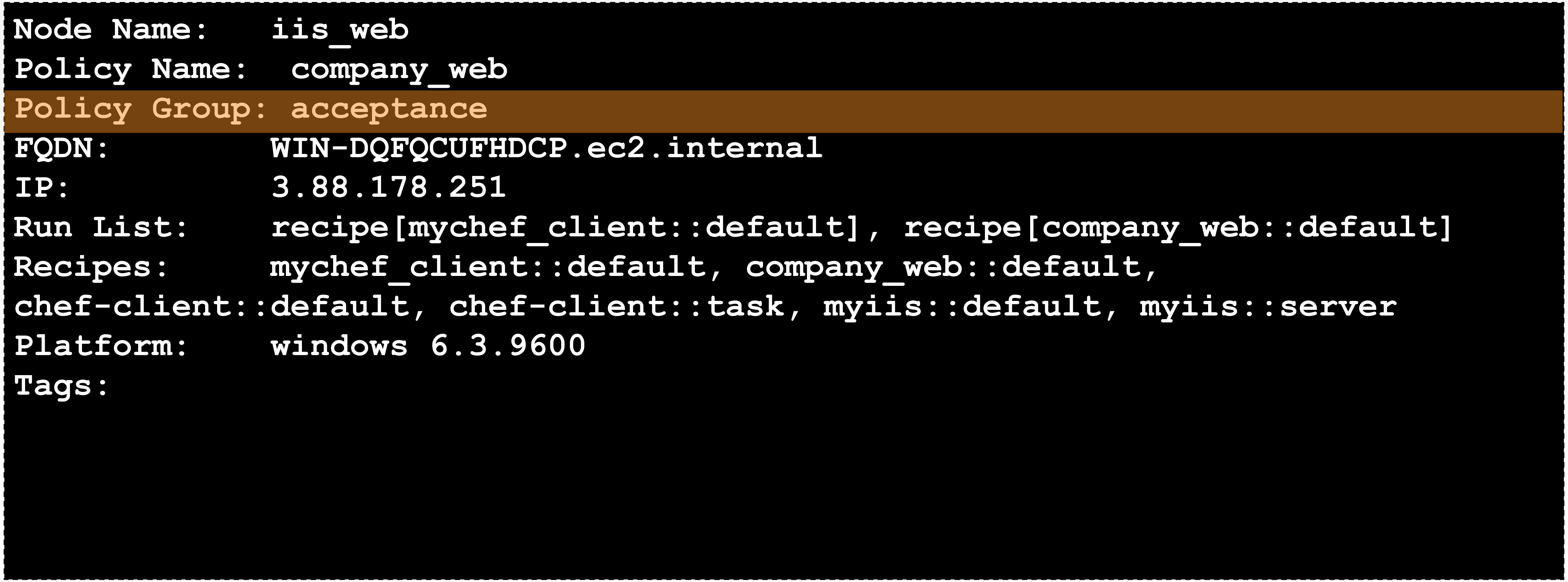
```
Successfully set the policy on node iis_web
```

Here we are setting the iis_web node to the acceptance policy_group.

GL: View Information About Your Node



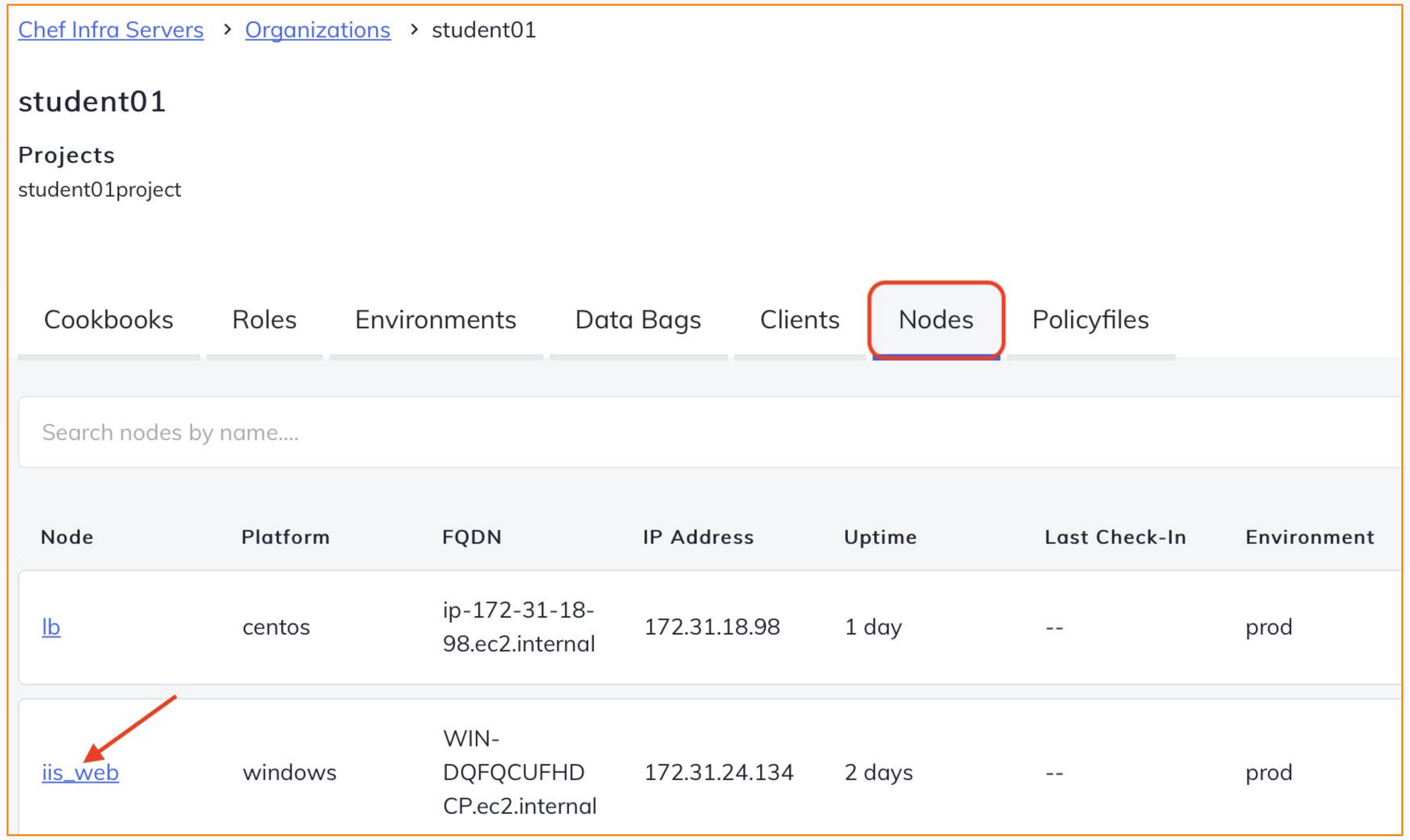
```
$ knife node show iis_web
```



```
Node Name:   iis_web
Policy Name:  company_web
Policy Group: acceptance
FQDN:        WIN-DQFQCUFHDCP.ec2.internal
IP:          3.88.178.251
Run List:    recipe[mychef_client::default], recipe[company_web::default]
Recipes:     mychef_client::default, company_web::default,
chef-client::default, chef-client::task, myiis::default, myiis::server
Platform:    windows 6.3.9600
Tags:
```

View Windows Node Information (iis_web)

1. Click on Nodes tab and select iis_web.
2. View the updated policy group.



Chef Infra Servers › Organizations › student01

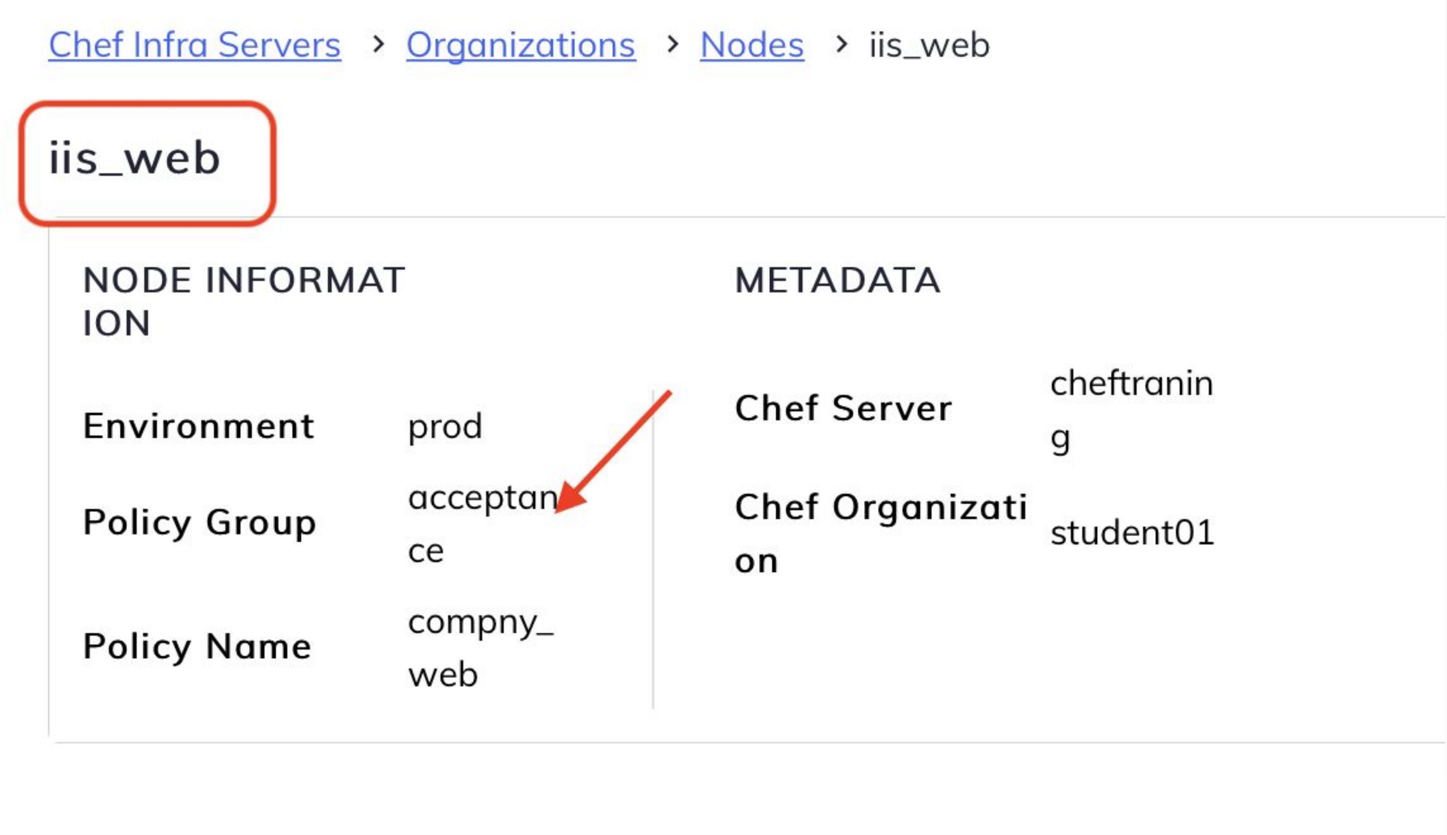
student01

Projects
student01project

Node	Platform	FQDN	IP Address	Uptime	Last Check-In	Environment
lb	centos	ip-172-31-18-98.ec2.internal	172.31.18.98	1 day	--	prod
iis_web	windows	WIN-DQFQCUFHDCP.ec2.internal	172.31.24.134	2 days	--	prod

View Windows Node Information (iis_web)

1. Click on Nodes tab and select iis_web.
2. View the updated policy group.



Chef Infra Servers › Organizations › Nodes › iis_web

iis_web

NODE INFORMATION		METADATA	
Environment	prod	Chef Server	cheftraning
Policy Group	acceptance	Chef Organization	student01
Policy Name	compny_web		

GL: Update the Search to Consider policy_group

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

```
...
web_nodes = search('node',"policy_name:company_web AND policy_group:#{node.policy_group}")
servers = []
web_nodes.each do |web_node|

  server = "#{web_node['cloud']['public_hostname']} #{web_node['cloud']['public_ipv4']}:80
maxconn 32"
  servers.push(server)
end

haproxy_backend 'servers' do
  server servers
end

haproxy_service 'haproxy'
```

Note: We are now using double quotes in the search string.

The double quotes are to interpolate the node.policy_group variable.

GL: Update the Search to Consider policy_group

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

```
web_nodes = search('node',"policy_name:company_web AND policy_group:#{node.policy_group}")
servers = []
web_nodes.each do |web_node|

  server = "#{web_node['cloud']['public_hostname']} #{web_node['cloud']['public_ipv4']}:80
maxconn 32"
  servers.push(server)
end

haproxy_backend 'servers' do
  server servers
end

haproxy_service 'haproxy' do
  subscribes :reload, 'template[/etc/haproxy/haproxy.cfg]', :delayed
end
```

Note: We are forcing the haproxy service to reload because we've updated the web server pool.

GL: Bump the myhaproxy Version in metadata.rb

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

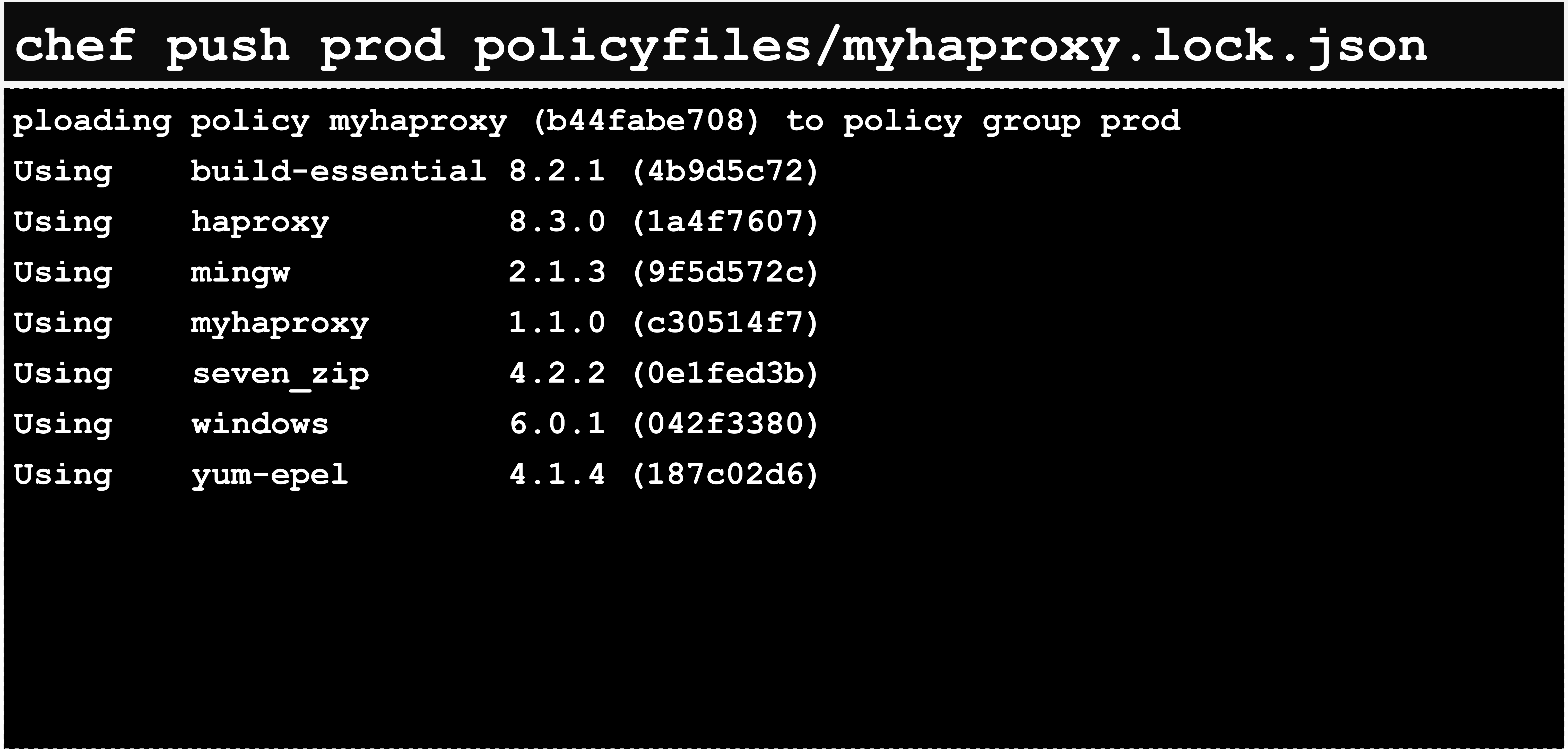
```
name 'myhaproxy'
maintainer 'The Authors'
maintainer_email 'you@example.com'
license 'All Rights Reserved'
description 'Installs/Configures myhaproxy'
version '1.1.0'
chef_version '>= 15.0'
depends 'haproxy', '~> 8.3.0'
```

GL: Update the myhaproxy.rb Policy

```
chef update policyfiles/myhaproxy.rb
```

```
Attributes already up to date
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       4.2.2
Using      mingw           2.1.3
Using      windows         6.0.1
Lockfile written to
/Users/sdelfante/chef-repo/policyfiles/myhaproxy.lock.json
```

GL: Push the myhaproxy.rb Policy



```
chef push prod policyfiles/myhaproxy.lock.json
```

```
ploading policy myhaproxy (b44fabe708) to policy group prod
Using    build-essential 8.2.1 (4b9d5c72)
Using    haproxy         8.3.0 (1a4f7607)
Using    mingw           2.1.3 (9f5d572c)
Using    myhaproxy       1.1.0 (c30514f7)
Using    seven_zip       4.2.2 (0e1fed3b)
Using    windows         6.0.1 (042f3380)
Using    yum-epel        4.1.4 (187c02d6)
```

GL: Converge the Load Balancer node

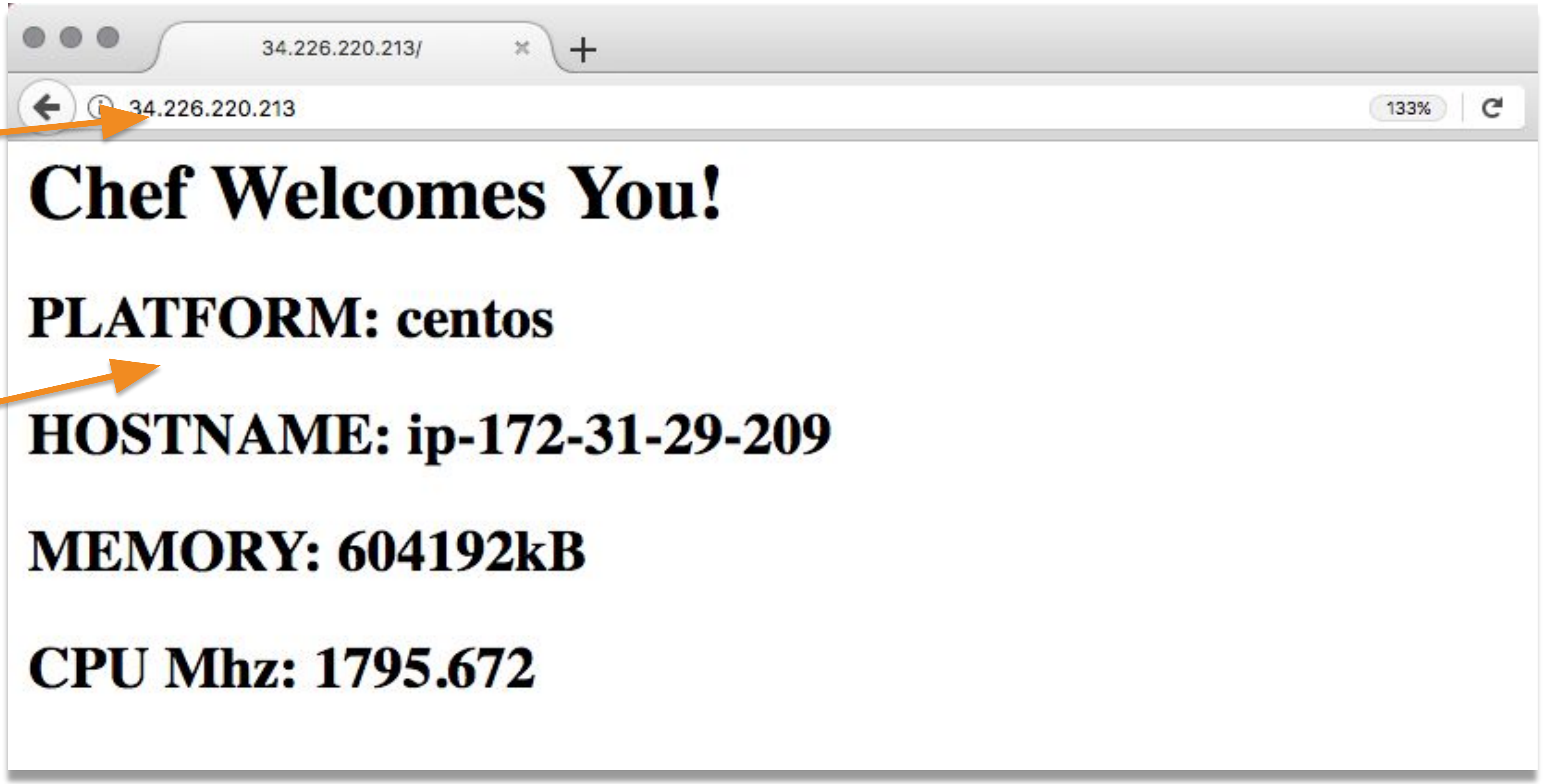
```
$ knife ssh 'name:lb' -x chef -P PWD 'sudo chef-client'
```

```
...
ec2-54-209-220-6.compute-1.amazonaws.com     - update content in file
/etc/haproxy/haproxy.cfg from 91c09e to 8f4138
ec2-54-209-220-6.compute-1.amazonaws.com     - suppressed sensitive resource
ec2-54-209-220-6.compute-1.amazonaws.com   * service[haproxy] action enable (up to
date)
ec2-54-209-220-6.compute-1.amazonaws.com   * service[haproxy] action start (up to
date)
ec2-54-209-220-6.compute-1.amazonaws.com   * haproxy_service[haproxy] action reload
ec2-54-209-220-6.compute-1.amazonaws.com     * service[haproxy] action reload
ec2-54-209-220-6.compute-1.amazonaws.com       - reload service service[haproxy]
ec2-54-209-220-6.compute-1.amazonaws.com      (up to date) ...
ec2-54-209-220-6.compute-1.amazonaws.com Running handlers:
ec2-54-209-220-6.compute-1.amazonaws.com Running handlers complete
ec2-54-209-220-6.compute-1.amazonaws.com Chef Infra Client finished, 2/26 resources
```

GL: Only the apache_web Node is Being Proxied

URL of load balancer.

Output from the apache_web server.



Chef Welcomes You!

PLATFORM: centos

HOSTNAME: ip-172-31-29-209

MEMORY: 604192kB

CPU Mhz: 1795.672

Group Lab: Using policy_group

Let's create an acceptance environment for our nodes

Objective:

- ✔ Test your current load balancer's behavior
- ✔ Assign the **iis_web** node to acceptance
- ✔ Update the load balancer's search criteria to **exclude** nodes in 'acceptance'
- ✔ Converge the load balancer node
- ✔ Test your load balancer

Review Questions

1. What is the benefit of constraining cookbooks to a particular environment?

2. What is the key item that defines an environment?

3. What does this bit of code in the load balancer do?

```
web_nodes = search('node',"policy_name:company_web AND policy_group:#{node.policy_group}")
```

DISCUSSION

Q&A

What questions can we help you answer?

CHEF™