Managing Multiple Nodes

Create another web server and add it as a proxy member

CHEF

Objectives

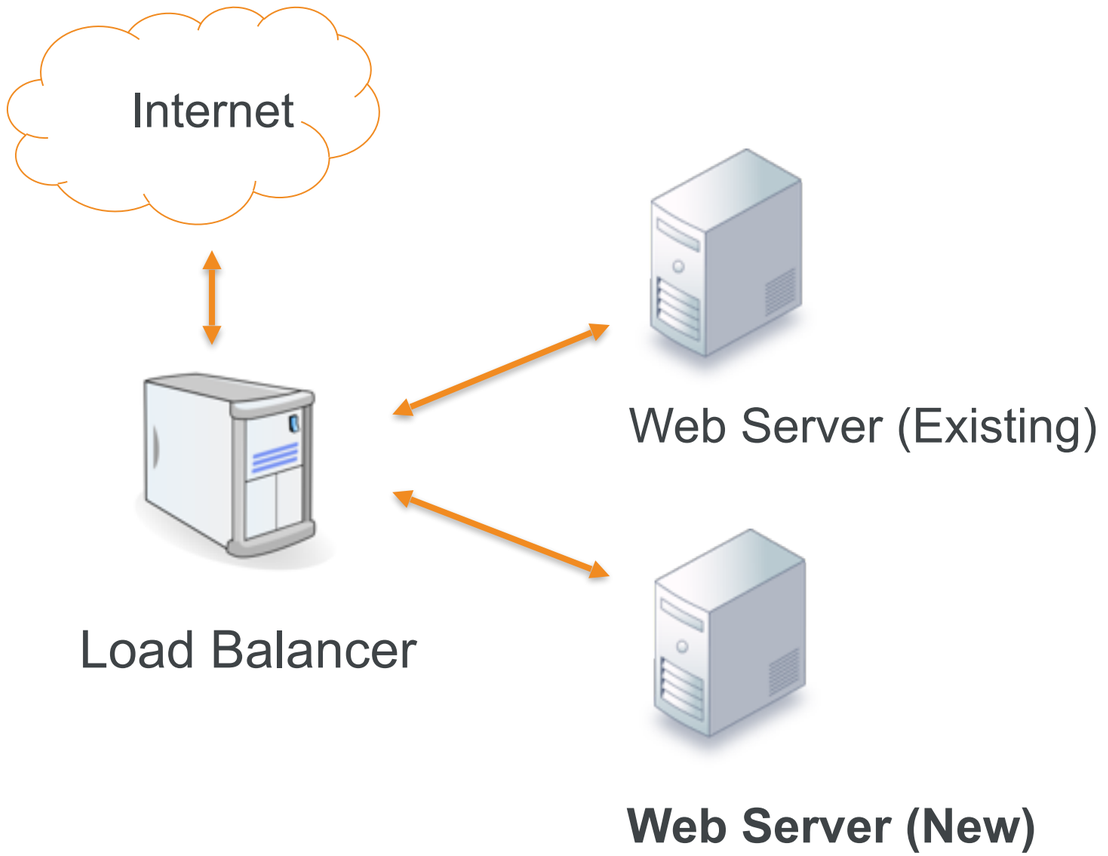
After completing this module, you should be able to

- 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

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.



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

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 [2016-09-16T17:36:14+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
```

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:
```

Lab: Set the Run List

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

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


CHEF

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
```

Verify that the New Node Serves the Page

CHEF

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

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.

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
```

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'
```

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'
```

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 .
```

Lab: Upload the Cookbook to Chef Server

```
$ berks upload
```

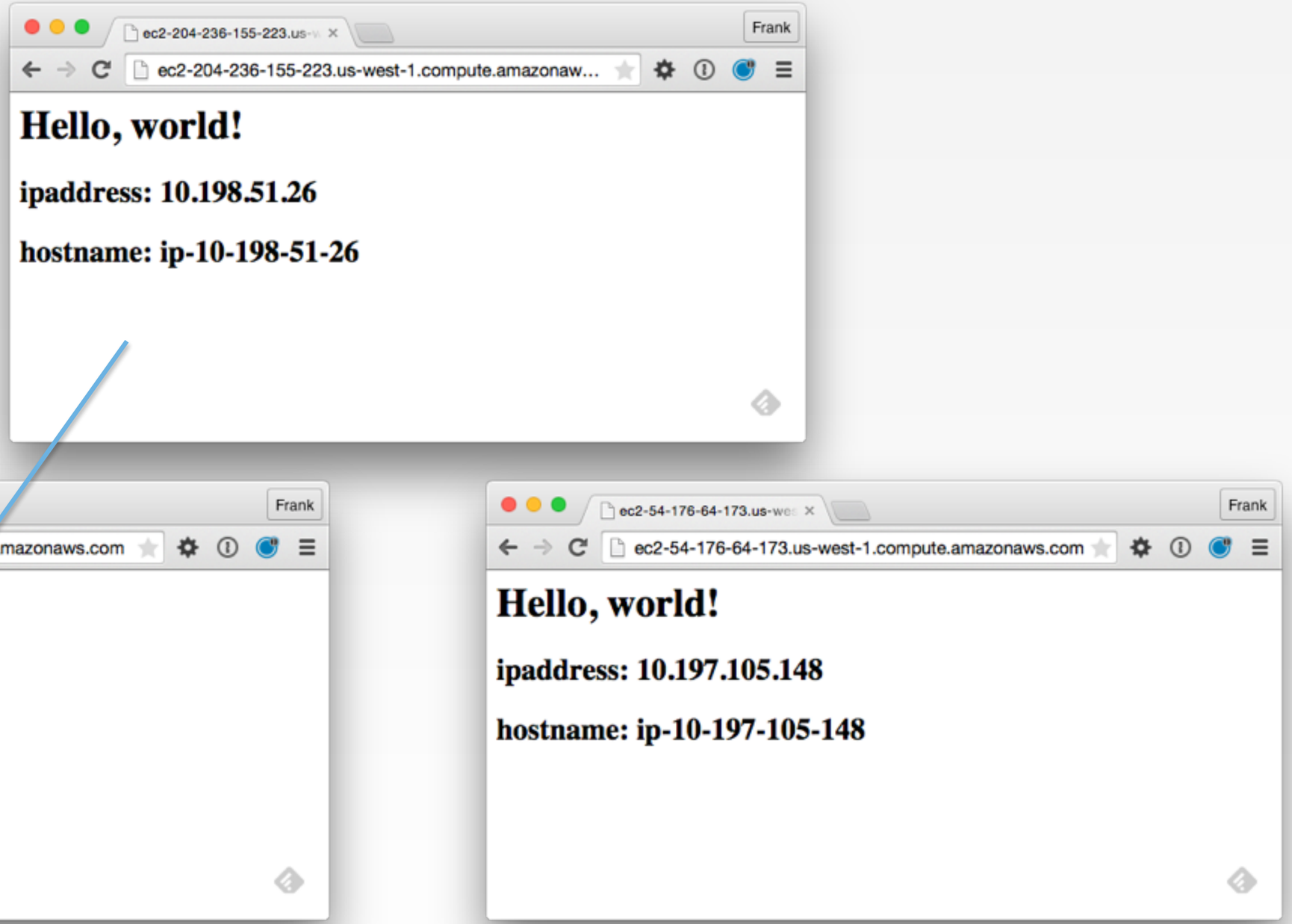
```
Uploaded build-essential (2.2.3) to: 'https://api.opscode.com:443/organizations/
steveessentials2'
Uploaded cpu (0.2.0) to: 'https://api.opscode.com:443/organizations/steveessentials2'
Uploaded haproxy (1.6.6) to: 'https://api.opscode.com:443/organizations/steveessentials2'
Uploaded myhaproxy (0.2.0)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

Lab: Converge the Cookbook

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

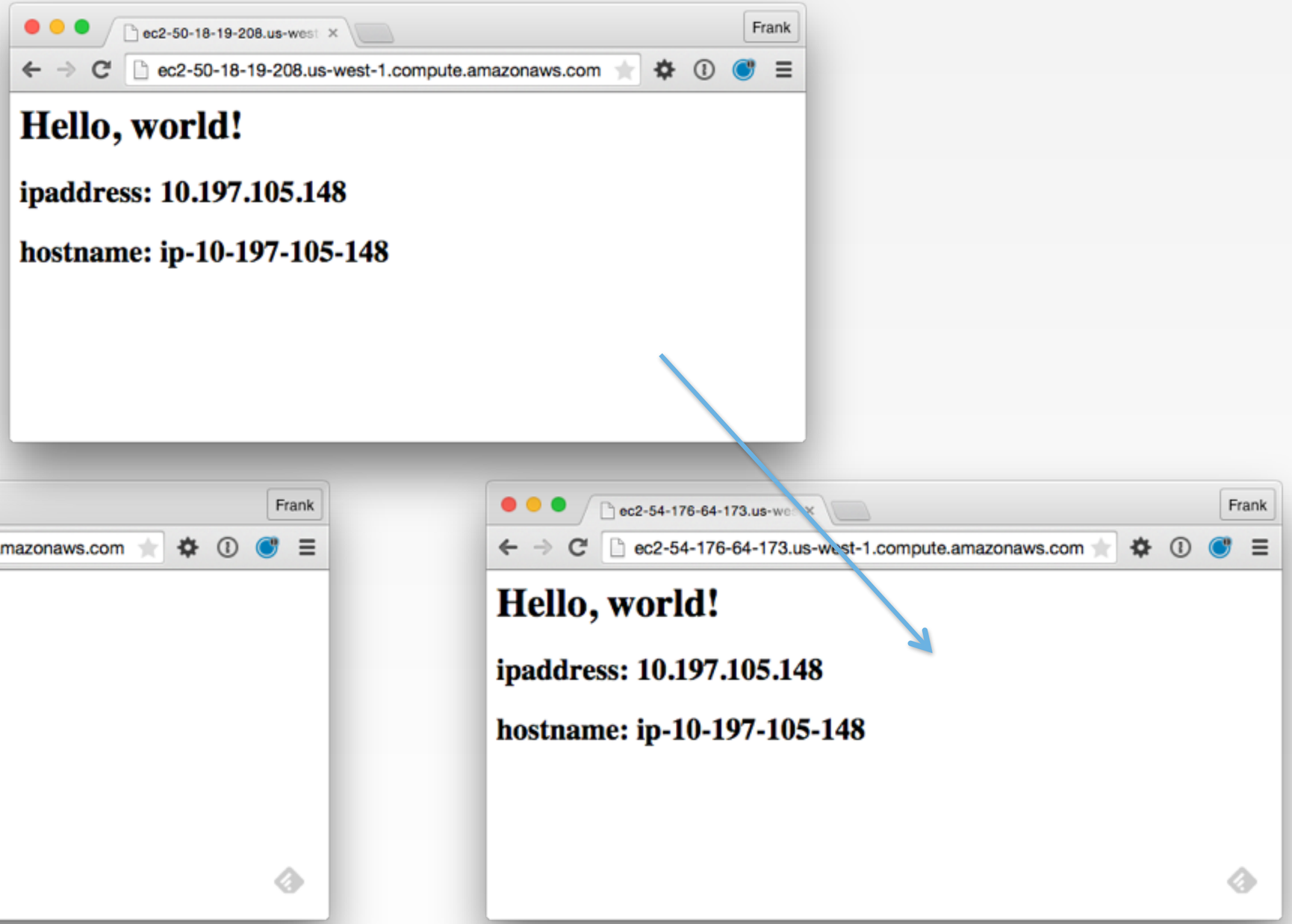
```
ec2-54-210-192-12.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-175-46-24.compute-1.amazonaws.com  Starting Chef Client, version 12.3.0
ec2-54-210-86-164.compute-1.amazonaws.com Starting Chef Client, version 12.3.0
ec2-54-210-192-12.compute-1.amazonaws.com resolving cookbooks 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
```

Lab: Test the Load Balancer



CHEF

Lab: Test the Load Balancer



CHEF

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

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?

CHEF™