Managing Multiple Nodes

Create another web server and add it as a proxy member

CHEF

Objectives

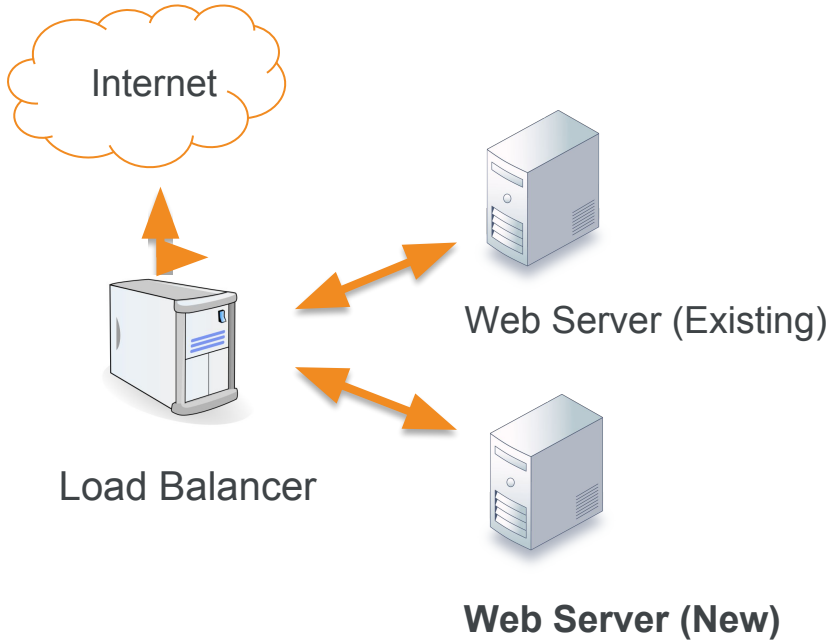
After completing this module, you should be able to

- 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 webserver 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 web2 -r "recipe[workstation],recipe[apache]"
```

```
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
```


CHEF

Verify the port and identity file for web1

```
$ vagrant ssh-config web2
```

```
Host web2
  HostName 127.0.0.1
  User vagrant
  Port 2201
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /Users/USER/chef-repo/.vagrant/machines/web2/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
```

Bootstrap Your Node

```
$ knife bootstrap localhost --ssh-port WEB2_PORT --ssh-user vagrant --sudo
--identity-file PATH_TO_KEY -N web2 --run-list "recipe[workstation],recipe[apache]"
```

```
Creating new client for web2
Creating new node for web2
Connecting to localhost
localhost -----> Installing Chef Omnibus (-v 12)
localhost downloading https://omnitruck-direct.chef.io/chef/install.sh
localhost   to file /tmp/install.sh.12058/install.sh
localhost trying wget...
localhost el 7 x86_64
localhost Getting information for chef stable 12 for el...
localhost downloading
https://omnitruck-direct.chef.io/stable/chef/metadata?v=12&p=el&pv=7&m=x86_64
localhost   to file /tmp/install.sh.12063/metadata.txt
localhost trying wget...
```


CHEF

Run 'knife node list' Again

```
$ knife node list
```

```
load-balancer
web1
web2
```

Lab: Verify the New Node

```
$ knife node show web2
```

```
Node Name:   web2
Environment: _default
FQDN:        web2
IP:          192.168.10.44
Run List:    recipe[workstation], recipe[apache]
Roles:
Recipes:     workstation, workstation::default, apache, apache::default,
workstation::setup, workstation::vagrant, apache::server
Platform:    centos 7.2.1511
Tags:
```


CHEF

Login to web2

```
$ vagrant ssh web2
```

```
Last login: Sat Dec 31 02:59:27 2016 from 10.0.2.2
[vagrant@web2 ~]$
```

Verify the state of your web application

```
[vagrant@web2 ~]$ curl localhost
```

```
<html>
  <body>
    <h1>Hello, world!</h1>
    <h2>ipaddress: 192.168.10.44</h2>
    <h2>hostname: web1</h2>
  </body>
</html>
```

Return to your Workstation

```
[vagrant@web1 ~]$ exit
```

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

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