Search

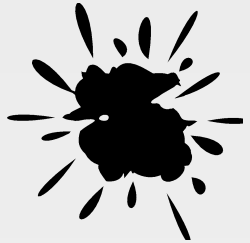
Update a Cookbook to Dynamically Use Nodes with the Web Role

CHEF

Objectives

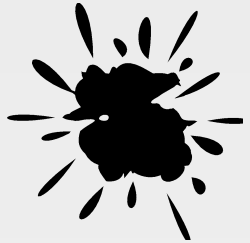
After completing this module, you should be able to

- Describe the query syntax used in search
- Build a search into your recipe code
- Create a Ruby Array and Ruby Hash
- Update the myhaproxy wrapper cookbook (for the load balancer) to dynamically use nodes with the web role



How Do We See Detail Across All Nodes?

`knife node show` only works with individual nodes so how do I see this info for multiple nodes?



How Do We See Detail Across All Nodes?

`knife node show` only works with individual nodes so how do I see this info for multiple nodes?

Search allows you to retrieve information across multiple nodes.

The Chef Server and Search

Chef Server maintains a searchable index of all nodes within our infrastructure.

Chef Server

Web Nodes	Proxy Nodes
Node 1	Node 2
Node 3	Node 5
Node 8	Node 6

What is search?

Use search to query data indexed on the chef server

```
$ knife search INDEX SEARCH_QUERY
```

The search runs on the server and is invoked from within a recipe or using knife

INDEX can be 'client', 'environment', 'node', 'role', (or the name of a data bag)

SEARCH_QUERY is of the format "attribute:value"

Querying ***:*** returns everything

GL: View Information for All Nodes

```
$ knife search node "*:*"
```

```
3 items found

Node Name:   web1
Environment: _default
FQDN:        web1
IP:          192.168.10.43
Run List:    role[web]
Roles:       web
Recipes:     workstation, workstation::default, apache, apache::default, workstation::setup, workstation::vagrant, apache::server
Platform:    centos 7.2.1511
Tags:

Node Name:   load-balancer
Environment: _default
FQDN:        load-balancer
IP:          10.0.2.15
Run List:    role[load-balancer]
....
```


CHEF

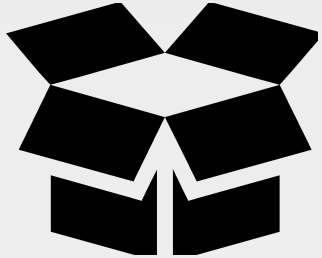
GL: Narrow the Search

```
$ knife search node "*:*" -a ipaddress
```

```
2 items found

web1:
 ipaddress : 192.168.10.43

load-balancer:
 ipaddress : 10.0.2.15

web2:
 ipaddress : 192.168.10.44
```


CHEF

CONCEPT

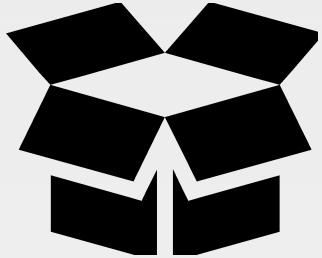


Search

So far we have seen how Chef is able to manage the policy of the nodes.

We have two web servers and one load balancer.

CONCEPT



Search

To add new servers as load balancer members, we would need to bootstrap a new web server and then update our load balancer's myhaproxy cookbook recipe.

That seems inefficient to have to update a cookbook recipe.

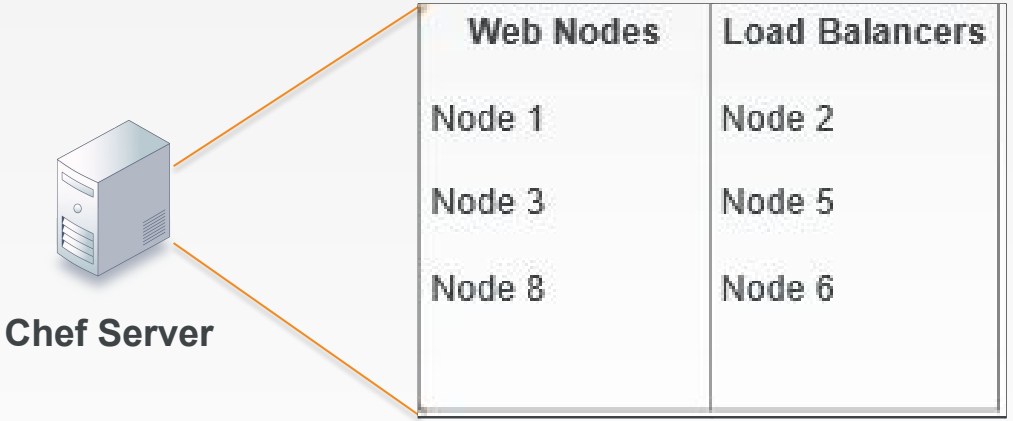
The Chef Server and Search

Chef Server maintains a representation of all the nodes within our infrastructure that can be searched on.

Search is a service discovery tool that allows us to query the Chef Server.

https://docs.chef.io/chef_search.html

https://docs.chef.io/chef_search.html#search-indexes



CHEF

The Chef Server and Search

We can ask the Chef Server to return all the nodes or a subset of nodes based on the query syntax that we provide it through `knife search` or within our recipes through `search`.