



Cookbook Attributes, Attribute Files and Dependencies

Setting attributes within a cookbook

Objectives



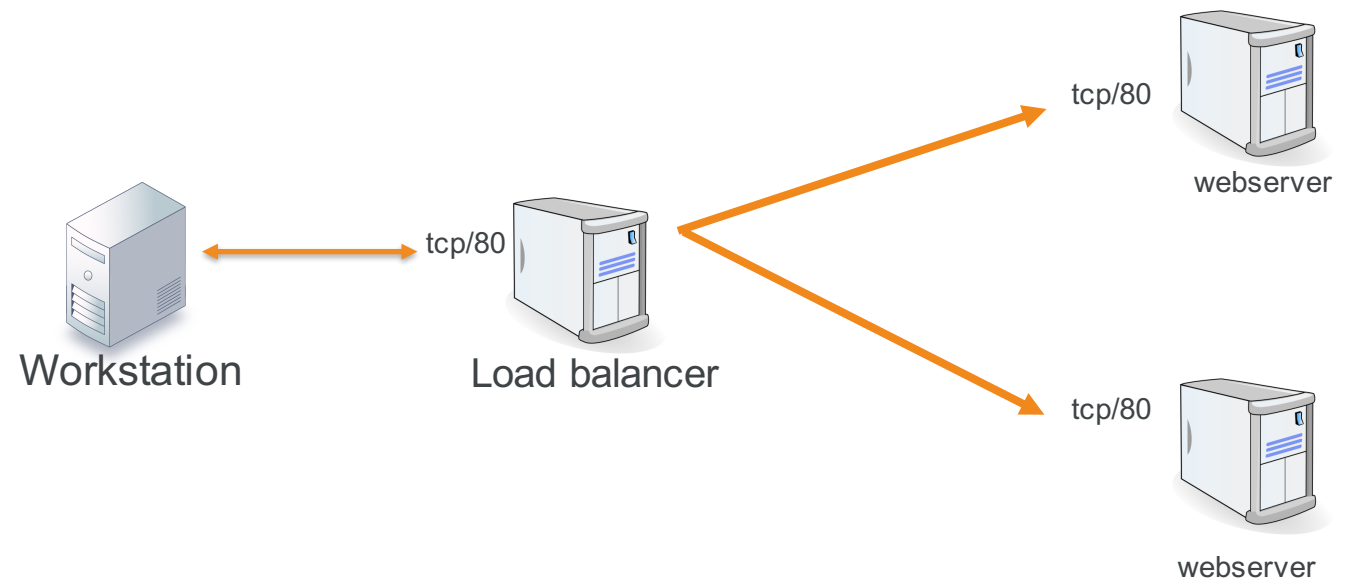
After completing this module, you should be able to

- Explain where cookbook attributes reside
- Configure dependencies between cookbooks
- Use `knife ssh` command

Our topology



We now have a load balancer & two web servers, all listening on port 80

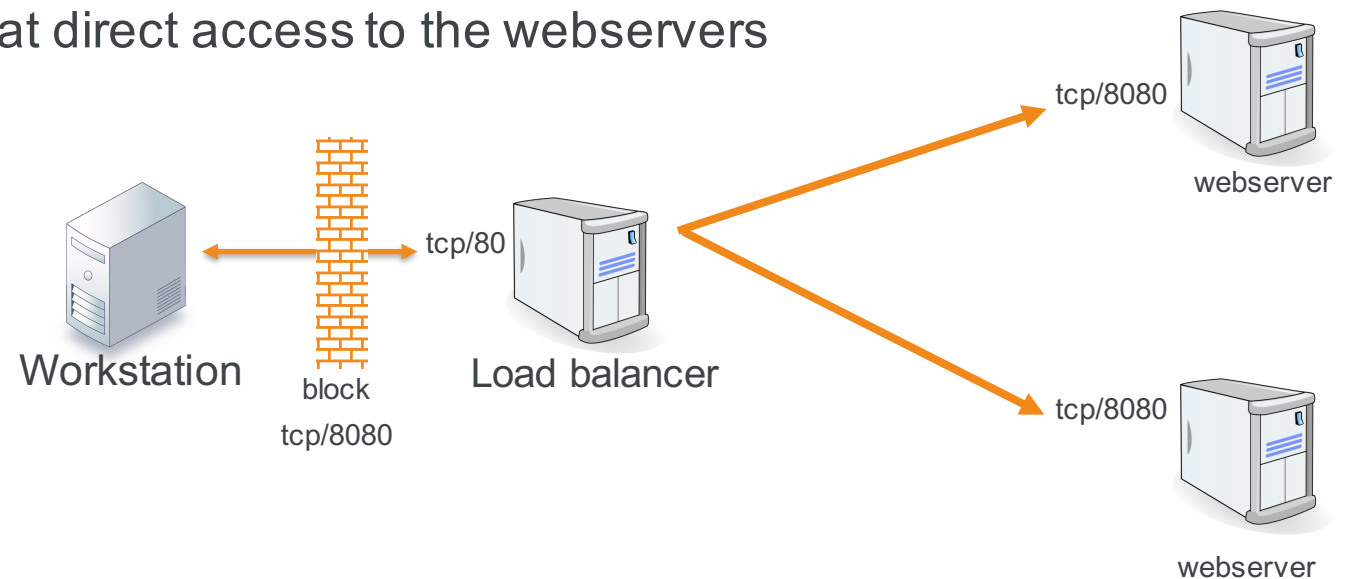


Our topology



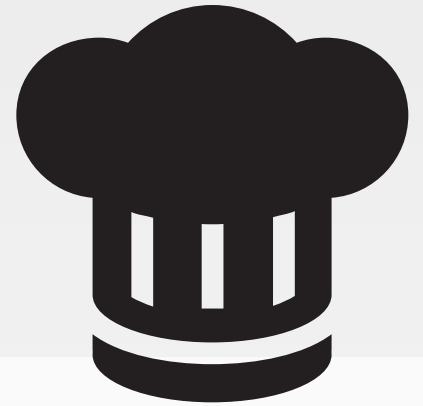
We now have a load balancer & two web servers, all listening on port 80

IT have mandated that direct access to the web servers should be blocked



LAB

Lab: Reconfigure Apache



*So we want HAProxy to serve content of port tcp/80,
and Apache to serve of tcp/8080*

Objective:

- ☐ Reconfigure Apache to serve content of port tcp/8080
- ☐ Reconfigure HAProxy to contact Apache on port tcp/8080



Reconfigure Apache

Apache is configured to listen on port 80 in the file `/etc/httpd/conf/httpd.conf`

We will create a template for this file in the apache cookbook and change the value to 8080 via an attribute

CONCEPT

Attribute Files



The Node Object contains many automatic attributes generate by OHAI

You can also maintain attributes within a cookbook

These are like variables or parameters for your cookbook and allow recipes to be data driven

DISCUSSION

Best Practices



- Well-written cookbooks change behavior based on attributes
- Ideally, you don't have to modify the contents of a cookbook to use it for your specific use case
- Look at the attributes directory for things you can override through roles to affect behavior of the cookbook
- Of course, well written cookbooks have sane defaults, and a README to describe all this.

Setting attributes in attribute files

Cookbook attributes are set in the attributes file


`./cookbooks/<cookbook>/attributes/default.rb`

Format is

```
default["attributename"] = "value"
```

We'll look at precedence later....

Lab: Bump the cookbook version number

 cookbooks/apache/metadata.rb

```
name                'apache'  
maintainer          'The Authors'  
maintainer_email    'you@example.com'  
license             'all_rights'  
description         'Installs/Configures apache'  
long_description    'Installs/Configures apache'  
version             '0.3.0'
```

Lab: Generate the attributes file



```
$ cd chef-repo
```

```
$ chef generate attribute cookbooks/apache default
```

```
Compiling Cookbooks...
```

```
Recipe: code_generator::attribute
```

```
  * directory[cookbooks/apache/attributes] action create
```

```
    - create new directory cookbooks/apache/attributes
```

```
  * template[cookbooks/apache/attributes/default.rb] action create
```

```
    - create new file cookbooks/apache/attributes/default.rb
```

```
    - update content in file cookbooks/apache/attributes/default.rb from none to  
e3b0c4
```

```
      (diff output suppressed by config)
```


Lab: Set the port value as an attribute

 cookbooks/apache/attributes/default.rb

```
default['apache']['port'] = 8080
```

Its good practice to include the name of the cookbook in the attribute name – helps trace where the value is set, although this is not enforced

Lab: Update the apache::server recipe

 cookbooks/apache/recipes/server.rb

```
...  
template '/var/www/html/index.html' do  
  source 'index.html.erb'  
end  
  
template '/etc/httpd/conf/httpd.conf' do  
  action :create  
  source 'httpd.conf.erb'  
  notifies :restart, 'service[httpd]'  
end  
  
service 'httpd' do  
  action [ :enable, :start ]  
end`
```

Lab: Generate the Template file



```
$ cd chef-repo
$ chef generate template cookbooks/apache httpd.conf
```

output suppressed by config)

```
MacBook-Pro-3:chef-repo johnfitzpatrick$ chef generate template cookbooks/apache
httpd.conf.erb
```

Compiling Cookbooks...

Recipe: code_generator::template

- * directory[cookbooks/apache/templates/default] action create (up to date)
- * template[cookbooks/apache/templates/default/httpd.conf.erb] action create

- create new file cookbooks/apache/templates/default/httpd.conf.erb

- update content in file cookbooks/apache/templates/default/httpd.conf.erb
from none to e3b0c4

(diff output suppressed by config)

Lab: Add the 'port' variable to the Template

 cookbooks/apache/templates/default/httpd.conf.erb

```
...  
MaxSpareThreads    75  
ThreadsPerChild    25  
MaxRequestsPerChild 0  
</IfModule>
```

If you are feeling hardcore, type it
<http://bit.ly/1Hdx7E1>

```
Listen 80
```

```
Listen <%= node['apache']['port'] %>
```

```
LoadModule auth_basic_module modules/mod_auth_basic.so  
LoadModule auth_digest_module modules/mod_auth_digest.so  
...
```

Copy entire contents of file, paste into the template, and edit line 27



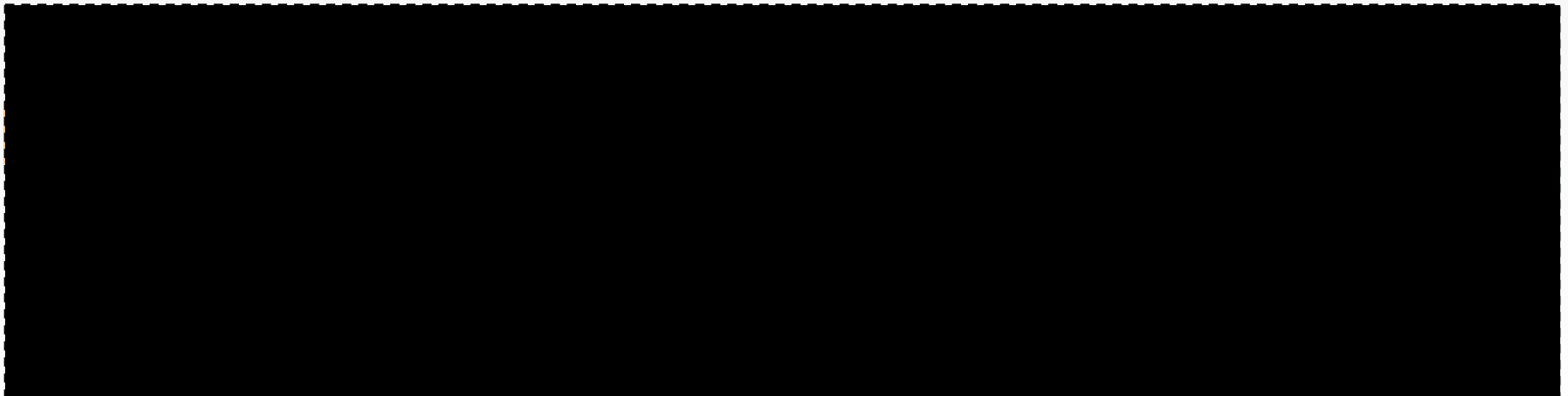
Lab: Upload the Cookbook

- ❑ Upload the apache cookbook to the Chef Server

Lab: Upload the Cookbook



```
$ cd chef-repo/cookbooks/apache
```



Lab: Upload the Cookbook



```
$ berks install
```

```
Resolving cookbook dependencies...  
Fetching 'apache' from source at .  
Fetching cookbook index from https://supermarket.chef.io...  
Using apache (0.3.0) from source at .
```

Lab: Upload the Cookbook

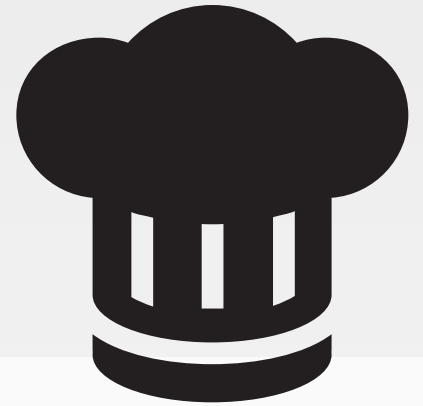


```
$ berks upload
```

```
Uploaded apache (0.3.0) to: 'https://api.opscode.com:443/organizations/ORGNAME'
```

LAB

Lab: Reconfigure Apache



*So we want HAProxy to serve content of port tcp/80,
and Apache to serve of tcp/8080*

Objective:

- ✓ Reconfigure Apache to serve content of port tcp/8080
- Reconfigure HAProxy to contact Apache on port tcp/8080

Lab: Bump the haproxy cookbook version

 cookbooks/haproxy/metadata.rb

```
name                'haproxy'  
maintainer          'The Authors'  
maintainer_email    'you@example.com'  
license             'all_rights'  
description         'Installs/Configures haproxy'  
long_description    'Installs/Configures haproxy'  
version             '0.2.0'
```

Lab: Configuring haproxy.cfg.erb

 cookbooks/haproxy/templates/default/haproxy.cfg.erb

```
...  
  
backend static  
    balance      roundrobin  
    server       static 127.0.0.1:4331 check  
  
backend app  
    balance      roundrobin  
    <% @webservers.each_with_index do |web, n| -%>  
    server <%= "app#{n}" %> <%= web['cloud']['public_ipv4'] %>:80 weight 1 maxconn 100 check  
    server <%= "app#{n}" %> <%= web['cloud']['public_ipv4'] %>:<%= node['apache']['port'] %>  
    weight 1 maxconn 100 check  
    <% end -%>
```

Add the port attribute as configured
in the apache cookbook

DISCUSSION

We've introduced a dependency



We now have the situation where HAProxy cookbook relies on an attribute that is configured in version 0.3.0 of the Apache cookbook

We need to manage this dependency – we do this in the haproxy cookbook `metadata.rb` file

Lab: Bump the cookbook version number

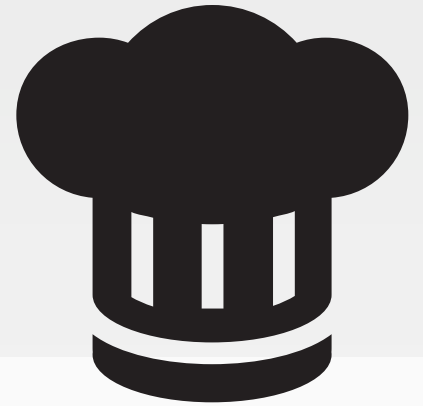
 cookbooks/haproxy/metadata.rb

```
name                'haproxy'
maintainer           'The Authors'
maintainer_email     'you@example.com'
license              'all_rights'
description           'Installs/Configures haproxy'
long_description     'Installs/Configures haproxy'
version              '0.2.0'

depends 'apache', '>= 0.3.0'
```


LAB

Lab: Reconfigure Apache



*So we want HAProxy to serve content of port tcp/80,
and Apache to serve of tcp/8080*

Objective:

- ✓ Reconfigure Apache to serve content of port tcp/8080
- ✓ Reconfigure HAProxy to contact Apache on port tcp/8080



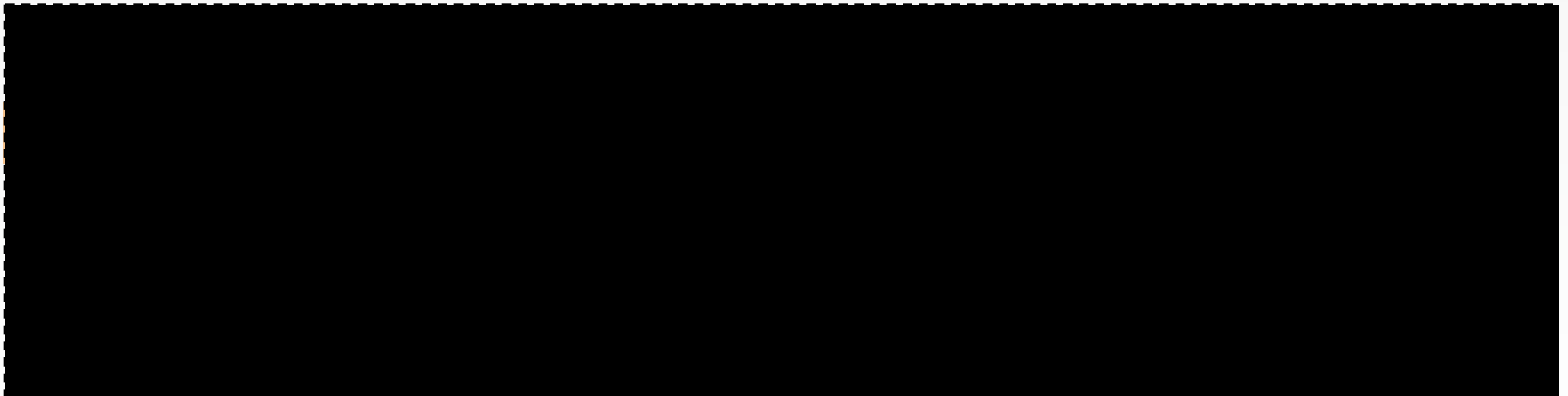
Lab: Upload the Cookbook

- ☐ Upload the haproxy cookbook to the Chef Server

Lab: Upload the Cookbook



```
$ cd chef-repo/cookbooks/haproxy
```



DISCUSSION

Configuring Berks



`berks install` retrieves dependencies from the Chef Supermarket by default

We want to override this to use the local apache cookbook instead of the community one

Lab: Edit Berksfile

 ./Berksfile

```
source "https://supermarket.chef.io"
```

```
source 'http://localhost:55555'
```

```
metadata
```

```
cookbook 'apache', path: '../apache'
```

Lab: Upload the Cookbook



```
$ berks install
```

```
Fetching 'apache' from source at ../apache  
Fetching 'haproxy' from source at .  
Using haproxy (0.2.0) from source at .  
Using apache (0.3.0) from source at ../apache
```

Lab: Upload the Cookbook



```
$ berks upload
```

```
Uploaded apache (0.3.0) to: 'https://api.opscode.com:443/organizations/ORGNAME'
```

```
Uploaded haproxy (0.2.0) to: 'https://api.opscode.com:443/organizations/ORGNAME'
```

SSH Woes



Logging into both systems is a pain. We can use another knife tool to allow us to send commands to all of our nodes.

Lab: Using knife ssh



```
$ knife ssh --help
```

```
knife ssh QUERY COMMAND (options)
```

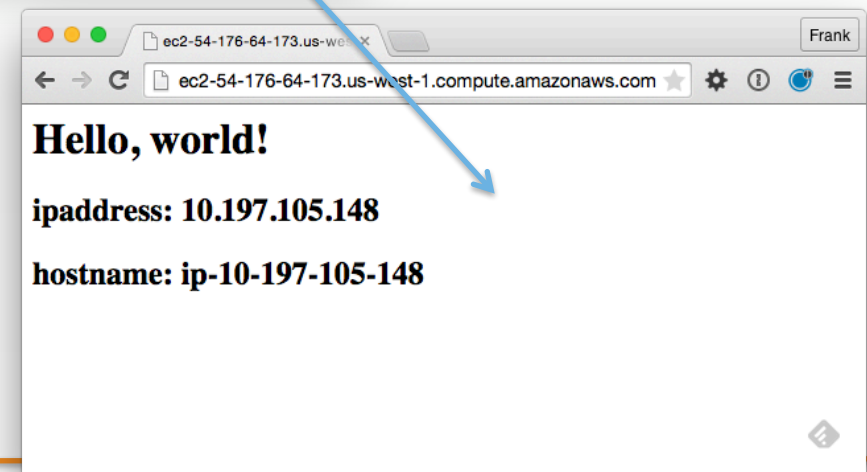
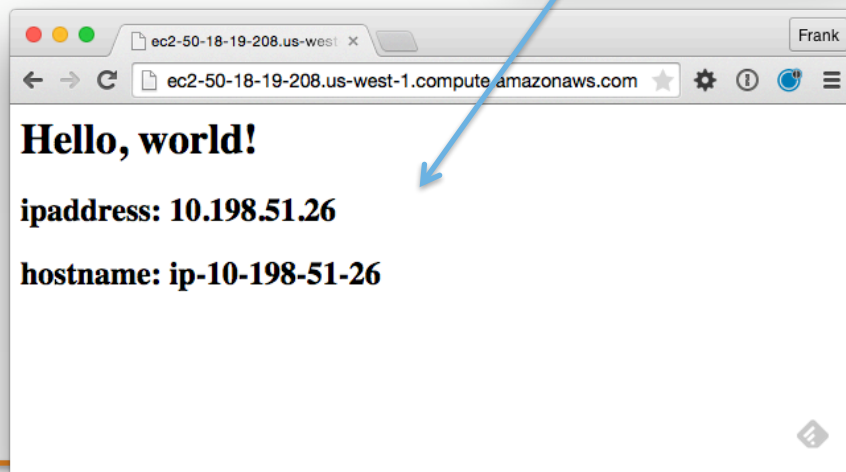
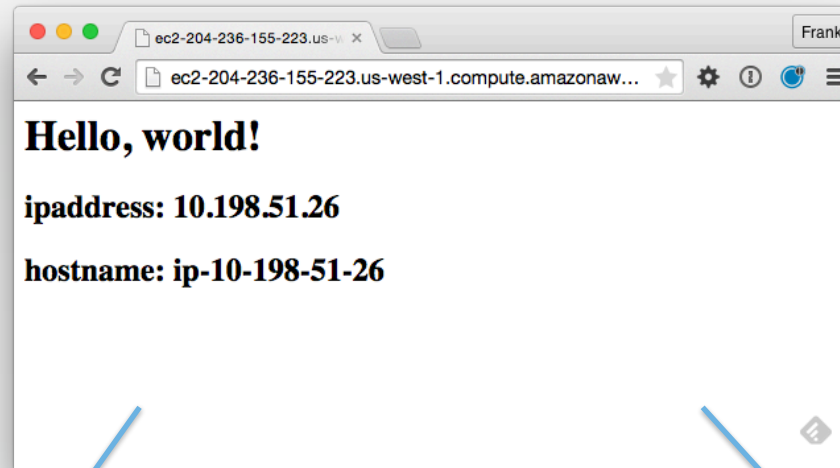
-a, --attribute ATTR	The attribute to use for opening the connection - default depends on the context
-s, --server-url URL	Chef Server URL
--chef-zero-host HOST	Host to start chef-zero on
--chef-zero-port PORT	Port (or port range) to start chef-zero on. Port ranges like 1000,1010 or 8889-9999 will try all given ports until one works.
-k, --key KEY	API Client Key
--[no-]color	Use colored output, defaults to false on Windows, true otherwise
-C, --concurrency NUM	The number of concurrent connections
-c, --config CONFIG	The configuration file to use
--defaults	Accept default values for all questions

Lab: Run chef-client on all nodes



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

```
ec2-54-88-169-195.compute-1.amazonaws.com Starting Chef Client, version 12.4.4
ec2-54-84-233-7.compute-1.amazonaws.com   Starting Chef Client, version 12.4.4
ec2-54-88-185-159.compute-1.amazonaws.com Starting Chef Client, version 12.4.4
ec2-54-88-169-195.compute-1.amazonaws.com resolving cookbooks for run list: ["haproxy"]
ec2-54-84-233-7.compute-1.amazonaws.com   resolving cookbooks for run list: ["apache"]
ec2-54-88-169-195.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-84-233-7.compute-1.amazonaws.com   Synchronizing Cookbooks:
ec2-54-88-169-195.compute-1.amazonaws.com   - haproxy
ec2-54-84-233-7.compute-1.amazonaws.com     - apache
...
```



DISCUSSION

Discussion



Attributes are like parameters to your cookbook – no hard-coded values in recipes or templates

Can you imagine in complex topologies, where you could have multiple levels of dependencies between cookbooks – berksel handles all of that out the box

DISCUSSION

Q&A



What questions can we help you answer?



CHEF™