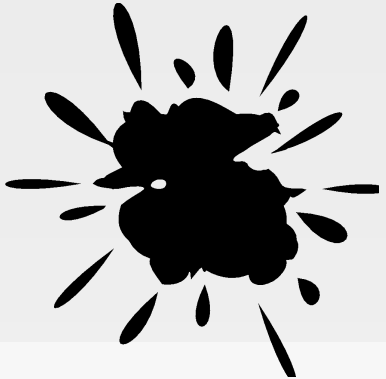


SSH Woes

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

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
```


CHEF

Run chef-client on all nodes

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

```
localhost Starting Chef Client, version 12.17.44
localhost resolving cookbooks for run list: ["myhaproxy"]
localhost Synchronizing Cookbooks:
localhost   - myhaproxy (0.2.0)
localhost   - haproxy (2.0.0)
localhost   - build-essential (7.0.3)
localhost   - seven_zip (2.0.2)
localhost   - windows (2.1.1)
localhost   - ohai (4.2.3)
localhost   - compat_resource (12.16.3)
localhost   - mingw (1.2.4)
localhost   - cpu (1.0.0)
....
```


CHEF

Verify the port and identity file

```
$ vagrant ssh-config load-balancer
```

```
Host load-balancer
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /Users/USER/chef-repo/.vagrant/machines/load-balancer/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
```


CHEF

Run chef-client on a Vagrant instance

```
$ knife ssh localhost "sudo chef-client" --manual-list --ssh-port PORT
--ssh-user vagrant --identity-file /PATH/TO/KEY
```

```
localhost Starting Chef Client, version 12.17.44
localhost resolving cookbooks for run list: ["myhaproxy"]
localhost Synchronizing Cookbooks:
localhost   - myhaproxy (0.2.0)
localhost   - haproxy (2.0.0)
localhost   - build-essential (7.0.3)
localhost   - seven_zip (2.0.2)
localhost   - windows (2.1.1)
localhost   - ohai (4.2.3)
localhost   - compat_resource (12.16.3)
localhost   - mingw (1.2.4)
localhost   - cpu (1.0.0)
....
```


CHEF

Run chef-client on Vagrant - short options

```
$ knife ssh localhost "sudo chef-client" -m -p PORT -x vagrant -i /PATH/TO/KEY
```

```
localhost Starting Chef Client, version 12.17.44
localhost resolving cookbooks for run list: ["myhaproxy"]
localhost Synchronizing Cookbooks:
localhost   - myhaproxy (0.2.0)
localhost   - haproxy (2.0.0)
localhost   - build-essential (7.0.3)
localhost   - seven_zip (2.0.2)
localhost   - windows (2.1.1)
localhost   - ohai (4.2.3)
localhost   - compat_resource (12.16.3)
localhost   - mingw (1.2.4)
localhost   - cpu (1.0.0)
....
```

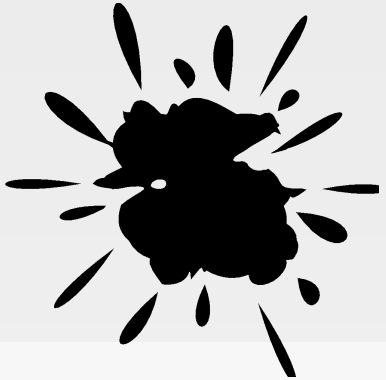

CHEF

Example - Run chef-client on Vagrant

```
$ knife ssh localhost "sudo chef-client" -m -p 2222 -x vagrant \
-i /Users/technotrainer/chef-repo/.vagrant/machines/load-balancer/virtualbox/private_key
```

```
localhost Starting Chef Client, version 12.17.44
localhost resolving cookbooks for run list: ["myhaproxy"]
localhost Synchronizing Cookbooks:
localhost   - myhaproxy (0.2.0)
localhost   - haproxy (2.0.0)
localhost   - build-essential (7.0.3)
localhost   - seven_zip (2.0.2)
localhost   - windows (2.1.1)
localhost   - ohai (4.2.3)
localhost   - compat_resource (12.16.3)
localhost   - mingw (1.2.4)
localhost   - cpu (1.0.0)
....
```


CHEF



SSH Woes

When using a Cloud Provider like Amazon EC2, knife ssh is a great way to issue a remote command to a set of nodes, not just a single node.

Example - Converge All Nodes

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

```
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
ec2-54-210-86-166.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:
["workstation","apache"]
ec2-54-210-86-164.compute-1.amazonaws.com  resolving cookbooks for run list:
["workstation","apache"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list: ["myhaproxy"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - workstation
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
....
```


CHEF

Example - Converge All Webserver Nodes

```
$ knife ssh "role:web" -x USER -P PWD "sudo chef-client"
```

```
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
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list: ["apache"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list: ["apache"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
ec2-54-175-46-24.compute-1.amazonaws.com  Synchronizing Cookbooks:
ec2-54-175-46-24.compute-1.amazonaws.com    - apache
```

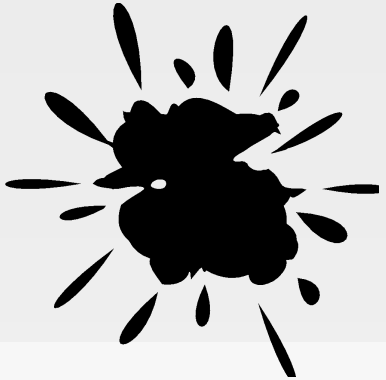

CHEF

Example - Converge All Proxy Nodes

```
$ knife ssh "role:load-balancer" -x USER -P PWD "sudo chef-client"
```

```
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:
["myhaproxy"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list:
["myhaproxy"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
  - myhaproxy (0.2.0)
  - haproxy (2.0.0)
  - build-essential (7.0.3)
  - seven_zip (2.0.2)
  - ohai (4.2.3)
  - compat_resource (12.16.3)
  - mingw (1.2.4)
```


CHEF



SSH Woes

When using a Cloud Provider like Amazon EC2, knife ssh allows us to specify what node attribute to use as the “ipaddress”

This allows us to ssh in using the node[‘cloud’] attribute, which contains the public ipaddress.

NOTE: cloud web1's Public Host Name and IP

```
$ knife node show web1 -a cloud
```

```
node1:
  cloud:
    local_hostname:  ip-172-31-8-68.ec2.internal
    local_ipv4:      172.31.8.68
    private_ips:     172.31.8.68
    provider:        ec2
    public_hostname: ec2-54-175-46-24.compute-1.amazonaws.com
    public_ips:      54.175.46.24
    public_ipv4:     54.175.46.24
```


CHEF

Example - Converge All Nodes

```
$ knife ssh "*:*" -x USER -P PWD "sudo chef-client" -a cloud.public_ipv4
```

```
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
ec2-54-210-86-166.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:
["workstation","apache"]
ec2-54-210-86-164.compute-1.amazonaws.com  resolving cookbooks for run list:
["workstation","apache"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list: ["myhaproxy"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - workstation
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
....
```


CHEF

Example - Converge All Webserver Nodes

```
$ knife ssh "role:web" -x USER -P PWD "sudo chef-client" -a cloud.public_ipv4
```

```
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
ec2-54-210-86-164.compute-1.amazonaws.com resolving cookbooks for run list: ["apache"]
ec2-54-175-46-24.compute-1.amazonaws.com  resolving cookbooks for run list: ["apache"]
ec2-54-210-86-164.compute-1.amazonaws.com Synchronizing Cookbooks:
ec2-54-210-86-164.compute-1.amazonaws.com   - apache
ec2-54-210-86-164.compute-1.amazonaws.com Compiling Cookbooks...
ec2-54-210-86-164.compute-1.amazonaws.com Converging 3 resources
ec2-54-210-86-164.compute-1.amazonaws.com Recipe: apache::server
ec2-54-175-46-24.compute-1.amazonaws.com  Synchronizing Cookbooks:
ec2-54-175-46-24.compute-1.amazonaws.com    - apache
```


CHEF