Chef Fundamentals

training@getchef.com

Nathen Harvey

- Community Director
- Co-host of the Food Fight Show Podcast

- @nathenharvey

Webinar Objectives and Style

CHEF
GETCHEF.COM

Multi-week Webinar Series

- After completing of this webinar series you will be able to
  - Automate common infrastructure tasks with Chef
  - Describe Chef's architecture
  - Describe Chef's various tools
 - Apply Chef's primitives to solve your problems

How to learn Chef

- You bring the domain expertise about your business and infrastructure

- Chef provides a framework for automating your infrastructure

- Our job is to work together to teach you how to model and automate your infrastructure with Chef

Chef is a Language

- Learning Chef is like learning the basics of a language
- 80% fluency will be reached very quickly
- The remaining 20% just takes practice
- The best way to **learn** Chef is to ***use*** Chef

Questions & Answers

- Ask questions in the chat window when they come to you
  - We'll answer as many questions as we can at the end of the session
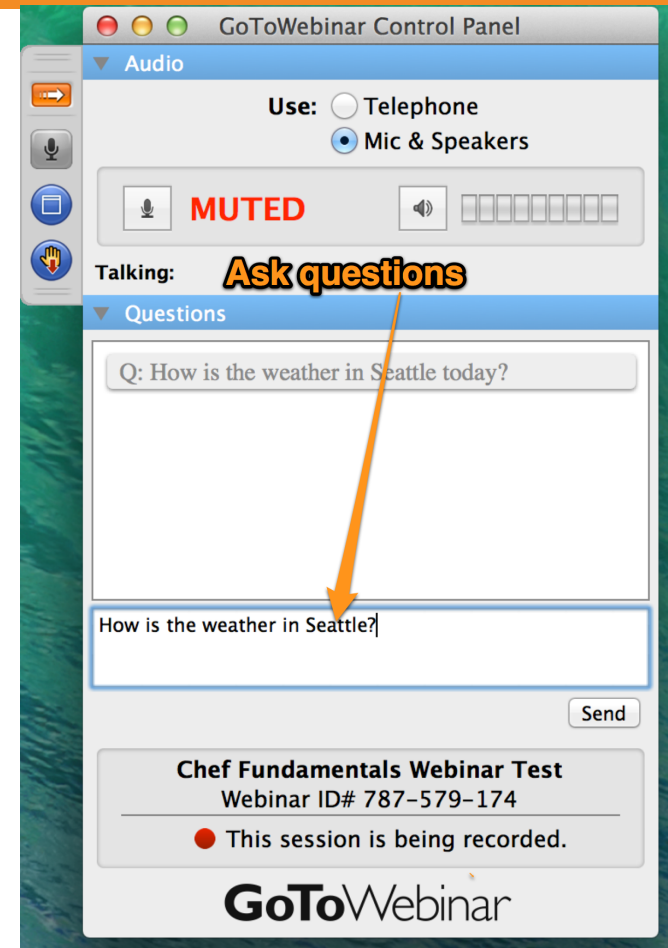
Questions & Answers

- Ask questions in the Google Discussion Forum

- This can be used during the webinar and outside of the webinar, too.

- https://groups.google.com/d/forum/learnchef-fundamentals-webinar

Slides and Video

- This webinar is being recorded. The video will be made available shortly after the session has ended.

- The slides used throughout this webinar will be made available at the end of each webinar.

- Watch http://learnchef.com for updates.

Agenda

Topics

- ~~Overview of Chef~~
- ~~Workstation Setup~~
- ~~Node Setup~~
- ~~Chef Resources and Recipes~~
- Working with the Node object - **Today**
- Roles - **Today**
- Common configuration with Data Bags
- Environments
- Community Cookbooks and Further Resources

Quick Recap

Where are we?

CHEF
GETCHEF.COM

In the last module

- Login to the node in your Chef Training Lab
- Install Chef nodes using "`knife bootstrap`"
- Build a web server configured to serve up our home page.

Where did my Node go?

- We still need a CentOS machine to manage
- The one we launched last time has likely expired
- Launch a new one using the Chef Lab

Launch Chef Training Lab

cloudshare

Home | Overview | CentOS 6.3 Server | Virtual Machines

Environment is ready

learnchef tutorial - CentOS

Your dedicated hands-on environment is just a click away.

Start Using
This Environment

We believe you shouldn't have to waste time copying gigabytes of software, shipping machines, or traveling, just to get your IT into people's hands.

When you click the 'Start Using' button to your right, you'll have instant access worldwide to a full, enterprise-grade IT

Lab - Login

```
$ ssh root@<EXTERNAL_ADDRESS>
```

```
The authenticity of host 'uvo1qrwls0jdgs3blvt.vm.cld.sr
(69.195.232.110)' can't be established.
RSA key fingerprint is d9:95:a3:b9:02:27:e9:cd:
74:e4:a2:34:23:f5:a6:8b.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'uvo1qrwls0jdgs3blvt.vm.cld.sr,
69.195.232.110' (RSA) to the list of known hosts.
chef@uvo1qrwls0jdgs3blvt.vm.cld.sr's password:
Last login: Mon Jan  6 16:26:24 2014 from
host86-145-117-53.range86-145.btcentralplus.com
[chef@CentOS63 ~]$
```

Checkpoint

- At this point you should have
  - One virtual machine (VM) or server that you'll use for the lab exercises
  - The IP address or public hostname
  - An application for establishing an ssh connection
  - 'sudo' or 'root' permissions on the VM

"Bootstrap" the Target Instance

```
$ knife bootstrap <EXTERNAL_ADDRESS> -x root -P chef -N "module3" -r "recipe[apache]"
```

```
Bootstrapping Chef on uvo1qrwls0jdgs3blvt.vm.cld.sr
...
...
uvo1qrwls0jdgs3blvt.vm.cld.sr Creating a new client identity for
module3 using the validator key.
uvo1qrwls0jdgs3blvt.vm.cld.sr resolving cookbooks for run list: []
uvo1qrwls0jdgs3blvt.vm.cld.sr Synchronizing Cookbooks:
uvo1qrwls0jdgs3blvt.vm.cld.sr Compiling Cookbooks...
uvo1qrwls0jdgs3blvt.vm.cld.sr [2014-01-28T11:03:14-05:00] WARN: Node
module3 has an empty run list.
uvo1qrwls0jdgs3blvt.vm.cld.sr Converging 0 resources
uvo1qrwls0jdgs3blvt.vm.cld.sr Chef Client finished, 0 resources updated
```

Exercise: Verify that the home page works

- Open a web browser
- Type in the the URL for your test node
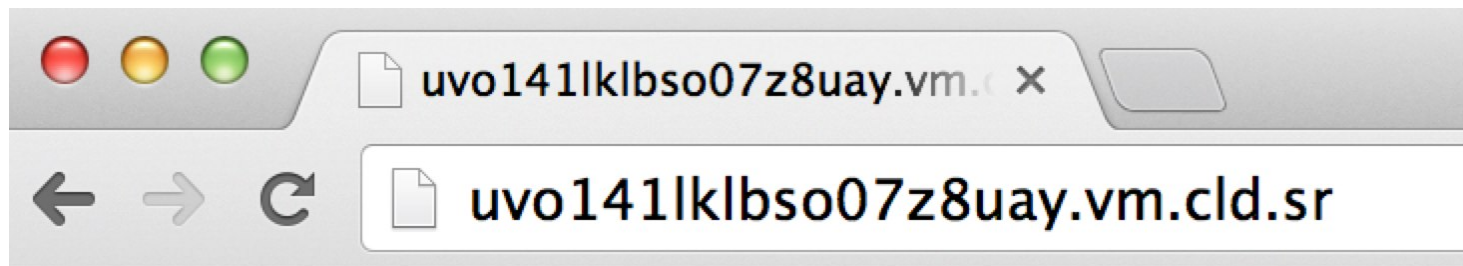
Introducing the Node object

Attributes

CHEF
GETCHEF.COM

Lesson Objectives

- After completing the lesson, you will be able to
  - Explain what the node object represents in Chef
  - Show details about a node
  - Describe what node attributes are
  - Retrieve a node attribute

What is the Node object

- A node is any physical, virtual, or cloud machines that is configured to be maintained by a Chef

- The 'node object' is the representation of that physical node within Chef (e.g. in JSON)

The Chef Node Object

- Nodes are the objects that you manage with Chef
- They have a few different properties
  - attributes
  - run_list
  - chef_environment

Node Data on the Chef Server

Chef Server

Search Index

PostgreSQL

Workstation
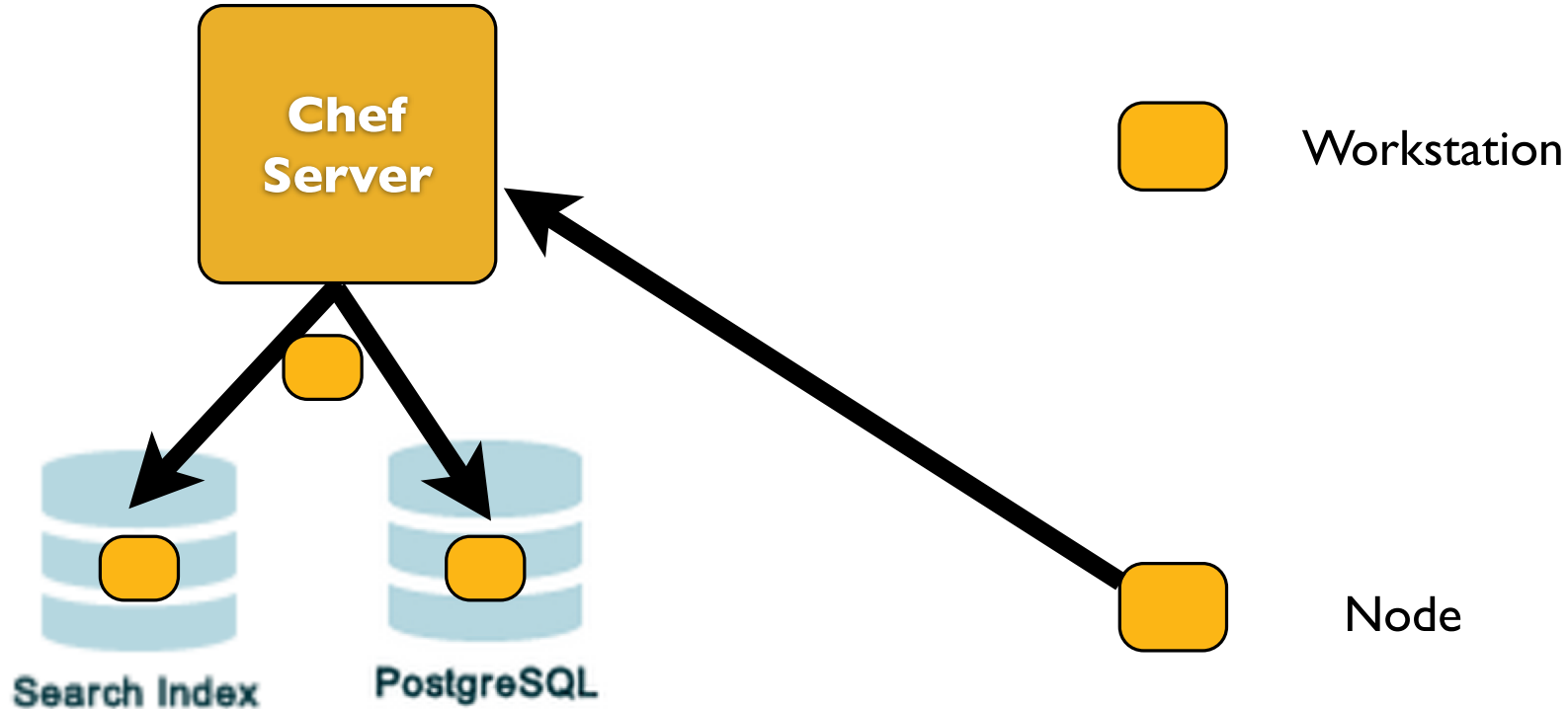
Node

The Chef Node Object

- Chef Server stores node object data
- It becomes searchable through the API with knife and with recipes
- Some of the data comes from `ohai`, which takes an inventory of the system and emits JSON data
- You can add data to the node through attributes in cookbooks, roles, directly on a node, etc

Node Attributes Syntax

- Node attributes are hashes (of hashes)
- Reference hash keys with square brackets and keys as strings
  - `node["hostname"]`
  - `node["kernel"]["machine"]`

Update the home page template

```
<p>Hello, world!</p>
<p>My name is <%= node['hostname'] %></p>
```

SAVE FILE!

Upload the cookbook

```
$ knife cookbook upload apache
```

```
Uploading apache              [0.1.0]
Uploaded 1 cookbook.
```

Run the chef-client on your test node

```
root@module3:~$ sudo chef-client
```

```
Starting Chef Client, version 11.10.4
resolving cookbooks for run list: ["apache"]
Synchronizing Cookbooks:
  - apache
Compiling Cookbooks...
Converging 3 resources
Recipe: apache::default
  * package[httpd] action install (up to date)
  * service[httpd] action enable (up to date)
  * service[httpd] action start (up to date)
  * template[/var/www/html/index.html] action create
    - update content in file /var/www/html/index.html from 17d291 to 77b6e3
        --- /var/www/html/index.html    2014-02-21 08:18:54.710230641 -0500
        +++ /tmp/chef-rendered-template20140221-4650-ykt042    2014-02-21 08:41:45.428088171 -0500
        @@ -1,2 +1,3 @@
         <h1>Hello, world!</h1>
        +<p>My name is centos63</p>

Running handlers:
Running handlers complete

Chef Client finished, 1/4 resources updated in 8.151165056 seconds
```

Homepage
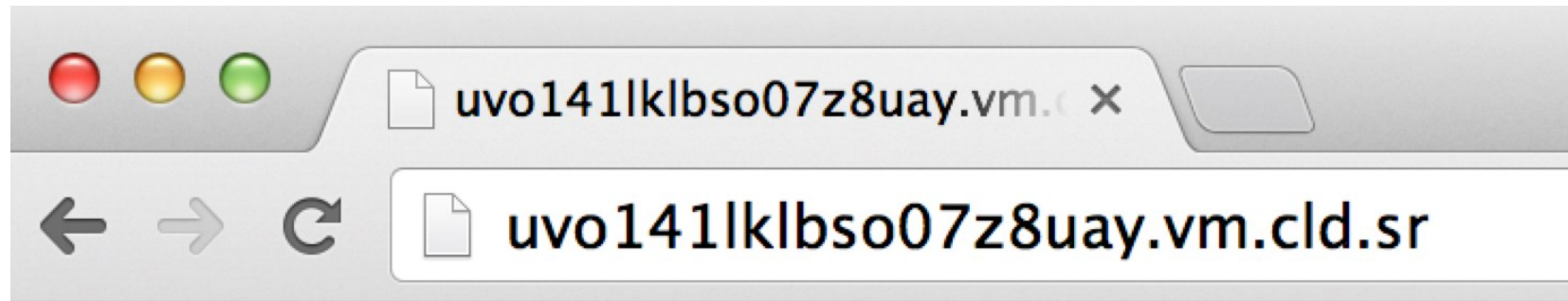
Run Ohai on the node

```
root@module3:~$ ohai
```

```
{
  "languages": {
    "ruby": {

    },
    "perl": {
      "version": "5.14.2",
      "archname": "x86_64-linux-gnu-thread-multi"
    },
    "python": {
      "version": "2.7.3",
      "builddate": "Aug 1 2012, 05:14:39"
    }
  },
  <SNIP>
```

Run Ohai on the node

```
root@module3:~$ ohai hostname
```

```
[
    "centos63"
]
```

Show the node object

```
$ knife node show module3
```

```
Node Name:    module3
Environment: _default
FQDN:         centos63.example.com
IP:           10.160.201.90
Run List:     recipe[apache]
Roles:
Recipes:      apache, apache::default
Platform:     centos 6.4
Tags:
```

Show specific node attribute

```
$ knife node show module3 -a hostname
```

```
module3:
  hostname: centos63
```

Setting Node Attributes

Setting attributes in recipes and attribute files

Lesson Objectives

- After completing the lesson, you will be able to
  - Describe where and how attributes are set
  - Declare an attribute with a recipe and set its value

What are Attributes?

- Attributes represent information from your node as well as input variables in your cookbooks

- Attributes hold static node data (e.g. platform, memory installed, disk space, etc)

- Allow you to abstract out data in from cookbooks (e.g. directory paths, users, software version numbers)

- All attributes are set on the "node object", and are indexed for search on the server

Attribute Sources

- Attributes can be set at various levels (in increasing order of precedence)
  - Automagically on the node itself (by Ohai)
  - In roles
  - In environments
  - In cookbook recipes
  - In cookbook attribute files

Ohai - set automatically

```
"languages": {
  "ruby": {

  },
  "perl": {
    "version": "5.14.2",
    "archname": "x86_64-
linux-gnu-thread-multi"
  },
  "python": {
    "version": "2.6.6",
    "builddate": "Jul 10
2013, 22:48:45"
  },
  "perl": {
    "version": "5.10.1",
    "archname": "x86_64-
linux-thread-multi"
  }
},
```

```
"kernel": {
  "name": "Linux",
  "release": "3.2.0-32-virtual",
  "version": "#1 SMP Wed Oct 16
18:37:12 UTC 2013",
  "machine": "x86_64",
  "modules": {
    "isofs": {
      "size": "70066",
      "refcount": "2"
    },
    "des_generic": {
      "size": "16604",
      "refcount": "0"
    }
  },
  "os": "GNU/Linux"
},
"os": "linux",
"os_version":
"2.6.32-358.23.2.el6.x86_64",
"ohai_time": 1389105685.7735305,
```

```
"network": {
  "interfaces": {
    "lo": {
      "mtu": "16436",
      "flags": [
        "LOOPBACK", "UP","LOWER_UP"
      ],
      "encapsulation": "Loopback",
      "addresses": {
        "127.0.0.1": {
          "family": "inet",
          "netmask": "255.0.0.0",
          "scope": "Node"
        },
        "::1": {
          "family": "inet6",
          "scope": "Node"
        }
      },
    },
    "eth0": {
      "type": "eth",
      "number": "0",
```

Setting attributes in attribute files

- Attributes can be set in the cookbook's attributes file
  - `./cookbooks/<cookbook>/attributes/default.rb`
- Format is

precedence	attribute name	attribute value
↓	↓	↓

```
default['apache']['dir'] = "/etc/apache2"
```

- We'll look at precedence later....

Create an attributes file

OPEN IN EDITOR: cookbooks/apache/attributes/default.rb

```
default["apache"]["greeting"] = "World"
```

SAVE FILE!

Update the home page template

```erb
<h1>Hello, <%= node['apache']['greeting'] %>!</h1>
<p>My name is <%= node['hostname'] %></p>
```

SAVE FILE!

Upload the cookbook

```
$ knife cookbook upload apache
```

```
Uploading apache          [0.1.0]
Uploaded 1 cookbook.
```

Run the chef-client on your test node

```
root@module3:~$ sudo chef-client
```

```
Starting Chef Client, version 11.10.4
resolving cookbooks for run list: ["apache"]
Synchronizing Cookbooks:
  - apache
Compiling Cookbooks...
Converging 3 resources
Recipe: apache::default
  * package[httpd] action install (up to date)
  * service[httpd] action enable (up to date)
  * service[httpd] action start (up to date)
  * template[/var/www/html/index.html] action create
    - update content in file /var/www/html/index.html from 77b6e3 to 455d5b
        --- /var/www/html/index.html    2014-02-21 08:41:45.428088171 -0500
        +++ /tmp/chef-rendered-template20140221-4957-1j4w9sg    2014-02-21 08:47:51.921529076 -0500
        @@ -1,3 +1,3 @@
        -<h1>Hello, world!</h1>
        +<h1>Hello, World!</h1>
         <p>My name is centos63</p>
```

Checkpoint

- We have a node attribute

  - `node["apache"]["greeting"]`
- We've updated the index.html template to use this attribute.

- This will be used again soon!

Exercise: Verify new homepage works

- Open a web browser
- The homepage takes the attribute file value

Roles

Role-based Attributes and Merge Order Precedence

CHEF
GETCHEF.COM

Lesson Objectives

- After completing the lesson, you will be able to
    - Explain what Roles are, and how they are used to provide clarity
    - Discuss the Role JSON DSL
    - Explain how merge order affects the precedence hierarchy

What is a Role?

- So far, we've been just adding our recipe directly to a single node

- But that's not how your infrastructure works - think about how you refer to servers

  - *"It's a **web server**"*

  - *"It's a **database** server"*

  - *"It's a **monitoring** server"*

What is a Role?

- Roles allow you to conveniently encapsulate the run lists and attributes required for a server to "be" what you already think it is

- In practice, Roles make it **easy to configure many nodes identically** without repeating yourself each time

Create a webserver role

OPEN IN EDITOR: roles/webserver.json

```json
{
  "name" : "webserver",
  "default_attributes" : {
    "apache" : {
      "greeting" : "Webinar"
    }
  },
  "run_list" : [
    "recipe[apache]"
  ]
}
```

SAVE FILE!

Components of a role

- Roles *must* have a **name**
- Roles *may* have a **description**
- Roles *may* have a **run_list**, just like a node
- Roles *may* set node **attributes**
  - **default_attributes**
  - **override_attributes**

http://docs.opscode.com/essentials_roles.html

Exercise: Create the role

```
Updated Role webserver!
```

Exercise: Create the role

```
$ knife role from file webserver.json
```

```
Updated Role webserver!
```

Remove the apache recipe

```
$ knife node run list remove module3 "recipe[apache]"
```

```
module3:
  run_list:
```

Add the webserver role

```
$ knife node run list add module3 "role[webserver]"
```

```
module3:
  run_list: role[webserver]
```

Attributes can be set multiple places

- This is for flexibility
- Set a "sane default" that will be used in a cookbook
- Override it easily in a role (higher priority)
- There are **many places** where attributes come from
- In practice, you'll use **2-3 most** of the time.
- The others are there *when you need them*.

Checkpoint

- We now have a webserver role.

- We could apply just this role on more nodes to scale out our Virtual Hosting service.

- Roles are a great way to assign attributes for specific purposes

Run chef-client

```
root@module3:~$ sudo chef-client
```

```
* template[/srv/apache/bears/index.html] action create
  - update content in file /srv/apache/bears/index.html from d8f475 to 60fcf9
      --- /srv/apache/bears/index.html        2014-02-21 09:20:53.870834436 -0500
      +++ /tmp/chef-rendered-template20140221-6541-m57zkc      2014-02-21 09:28:53.857331666 -0500
      @@ -1,4 +1,4 @@
      -<h1>Hello, World!</h1>
      +<h1>Hello, SCaLE12x!</h1>
       <p>My name is centos63</p>
       <p>We love bears</p>
       <p>Served from 10.160.201.90:80</p>

* template[/etc/httpd/conf.d/clowns.conf] action create (up to date)
* directory[/srv/apache/clowns] action create (up to date)
* template[/srv/apache/clowns/index.html] action create
  - update content in file /srv/apache/clowns/index.html from 828461 to d9b384
      --- /srv/apache/clowns/index.html        2014-02-21 09:20:53.883834816 -0500
```

Homepage

Six Week Series

- Module 1 - Overview of Chef
- Module 2 - Node Setup, Chef Resources & Recipes
- Today - Working with the Node object & Roles
- June 10 - Common configuration data with Databags
- June 17 - Environments
- June 24 - Community Cookbooks and Further Resources

- * Topics subject to change, schedule unlikely to change

Sign-up for Webinar

- http://pages.getchef.com/cheffundamentalsseries.html

Additional Resources

- Chef Fundamentals Webinar Series

- https://www.youtube.com/watch?v=S5lHUpzoCYo&list=PL11cZfNdwNyPnZA9D1MbVqIdGuOWqbumZ

- Discussion group for webinar participants

- https://groups.google.com/d/forum/learnchef-fundamentals-webinar

Additional Resources

- Learn Chef
- http://learnchef.com

- Documentation
- http://docs.opscode.com