Details About the System

Finding and Displaying Information About Our System

CHEF

Objectives

After completing this module, you should be able to

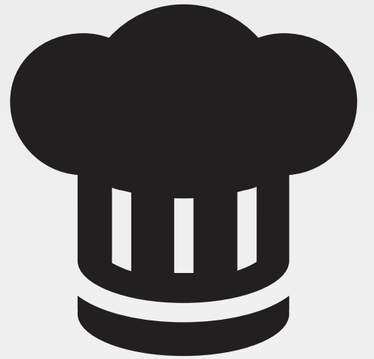
- Capture details about a system
- Use the node object within a recipe
- Use Ruby's string interpolation
- Update the version of a cookbook

Managing a Large Number of Servers

Have you ever had to manage a large number of servers that were almost identical?

How about a large number of identical servers except that each one had to have host-specific information in a configuration file?

CHEF



Details About the Node

Displaying system details in the home page definitely sounds useful.

Objective:

- Update the index.html file contents, in the "webserver" cookbook, to include node details

CHEF

Some Useful System Data

- IP Address
- hostname
- memory
- CPU - MHz

GL: Finding the IP Address

```
> ipconfig
```

```
Windows IP Configuration

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . : ec2.internal
   Link-local IPv6 Address . . . . . : fe80::2da8:4ba7:45e2:e863%21
   IPv4 Address. . . . . . . . . . . : 172.31.21.21
   Subnet Mask . . . . . . . . . . . : 255.255.240.0
   Default Gateway . . . . . . . . . : 172.31.16.1
```


CHEF

GL: Finding the Hostname

```
> hostname
```

```
WIN-KRQSVD3RFM7
```


CHEF

GL: Finding the Total Memory

```
> wmic ComputerSystem get TotalPhysicalMemory
```

```
TotalPhysicalMemory
8052654080
```


CHEF

GL: Finding the CPU MHz

```
> wmic cpu get name
```

```
Name
Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz
```


CHEF

DISCUSSION

Capturing System Data

What are the limitations of the way we captured this data?

How accurate will our data be when we deploy it on other systems?

Are these values we would want to capture in our tests?

CHEF

Hard Coded Values

The values that we have derived at this moment may not be the correct values when we deploy this recipe again even on the same system!

DISCUSSION

Data In Real Time

How could we capture this data in real-time?

CHEF

CONCEPT

Ohai!

Ohai is a tool that already captures all the data that we similarly demonstrated finding.

http://docs.chef.io/ohai.html

CHEF

Ohai!

```
> ohai
```


```
{
  "kernel": {
    "name": "Linux",
    "release": "2.6.32-431.1.2.0.1.el6.x86_64",
    "version": "#1 SMP Fri Dec 13 13:06:13 UTC 2013",
    "machine": "x86_64",
    "os": "GNU/Linux",
    "modules": {
      "veth": {
        "size": "5040",
        "refcount": "0"
      },
      "ipt_addrtype": {
```


CHEF

CONCEPT

All About The System

Ohai queries the operating system with a number of commands, similar to the ones demonstrated.

The data is presented in JSON (JavaScript Object Notation).

http://docs.chef.io/ohai.html

CHEF

CONCEPT

ohai + chef-client = <3

chef-client and chef-apply automatically executes ohai and stores the data about the node in an object we can use within the recipes named node.

http://docs.chef.io/ohai.html

CHEF

CONCEPT

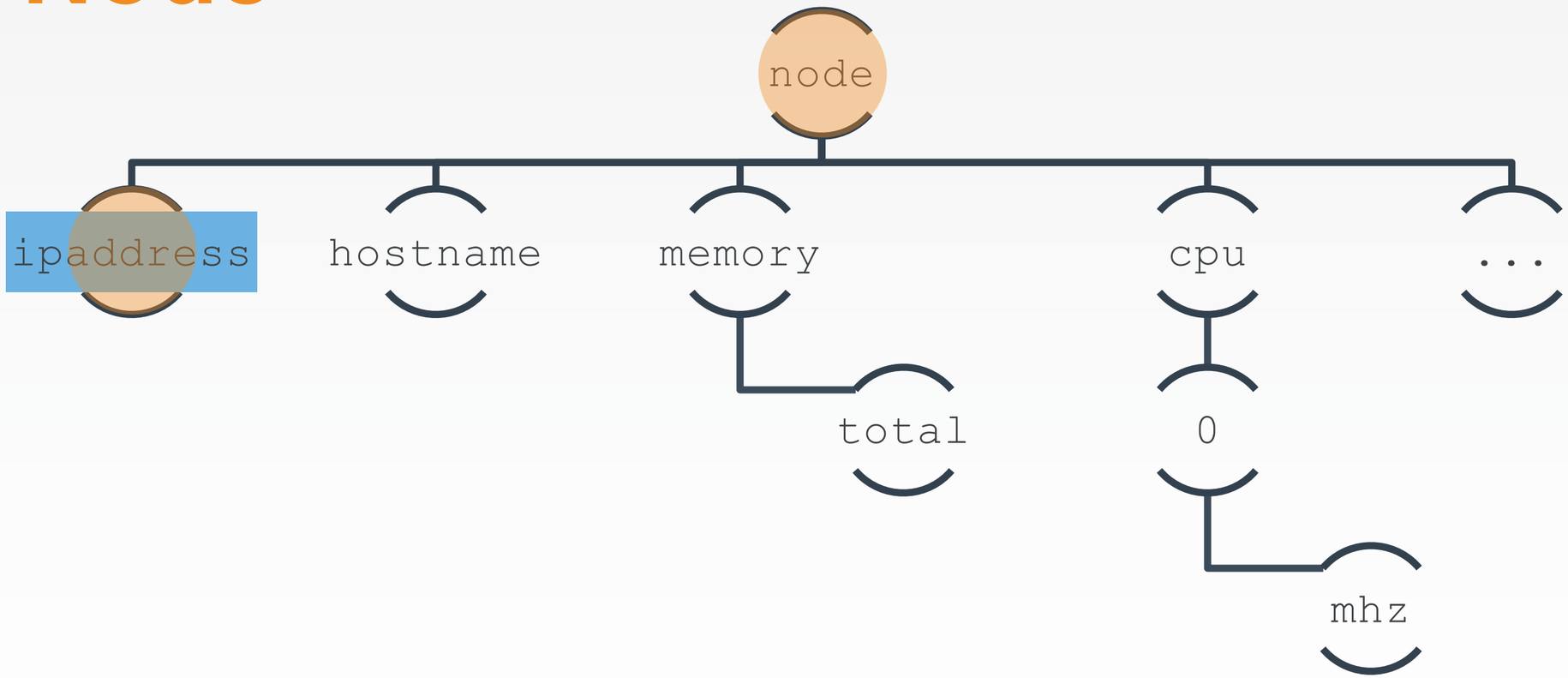
The Node Object

The node object is a representation of our system.
It stores all the attributes found about the system.

http://docs.chef.io/nodes.html#attributes

CHEF

The Node

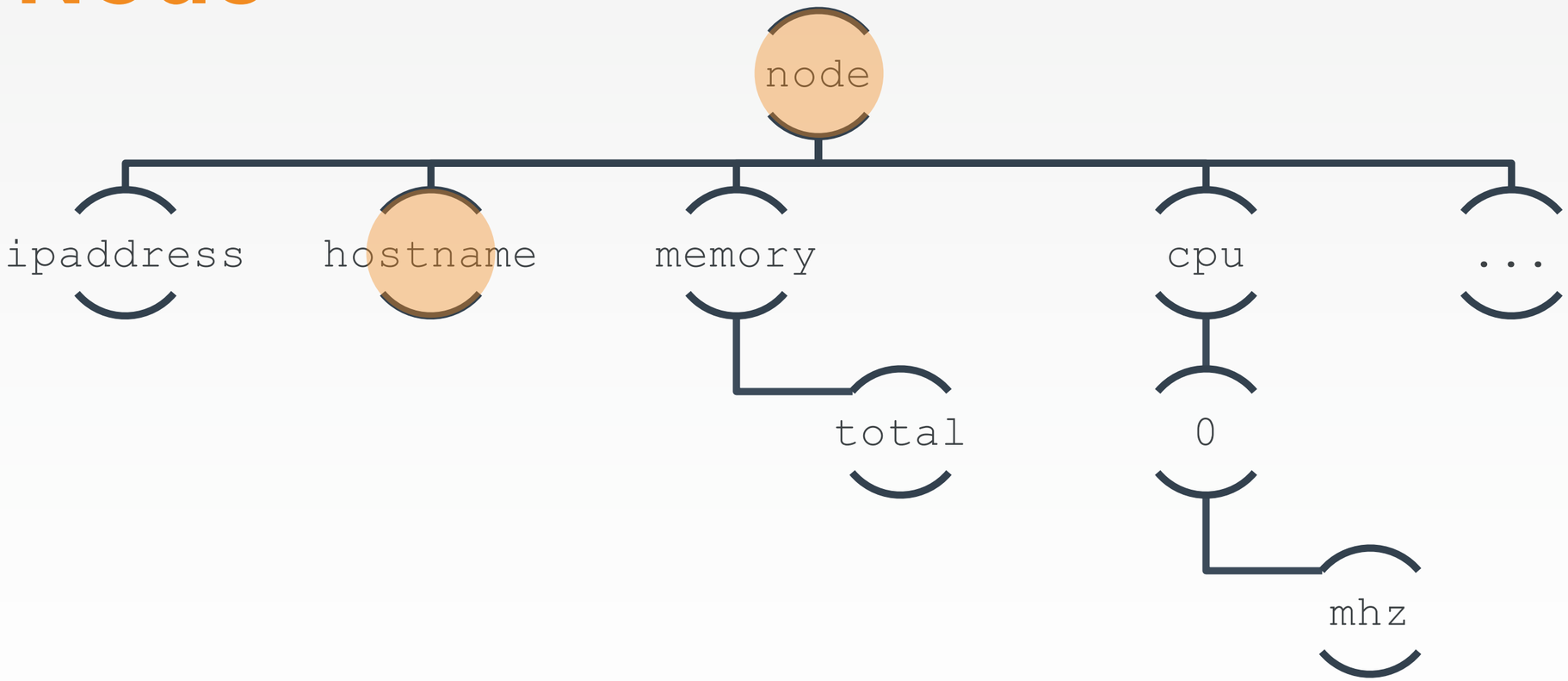



```
IPADDRESS: 104.236.192.102
```

```
"IPADDRESS: #{node['ipaddress']}"
```


CHEF

The Node

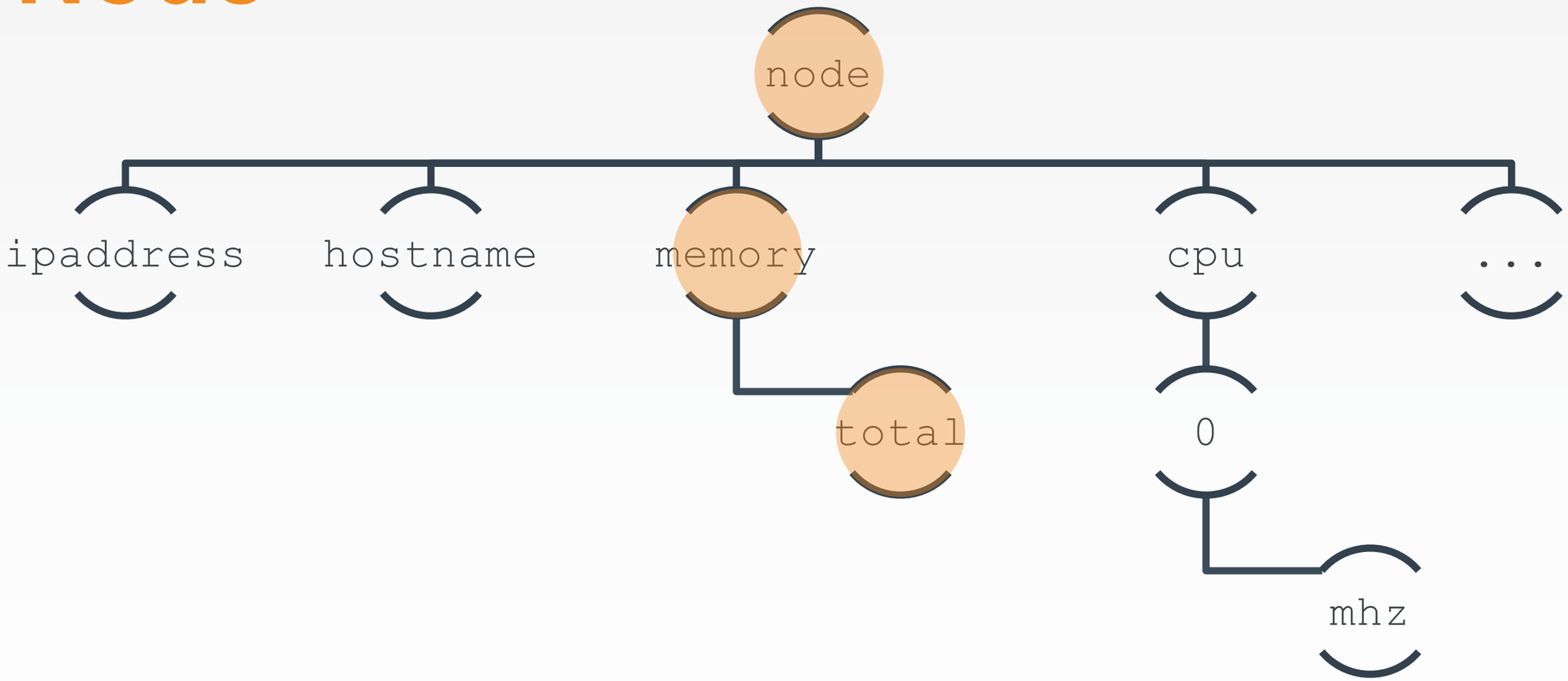



```
HOSTNAME: banana-stand
```

```
"HOSTNAME: #{node['hostname']}"
```


CHEF

The Node

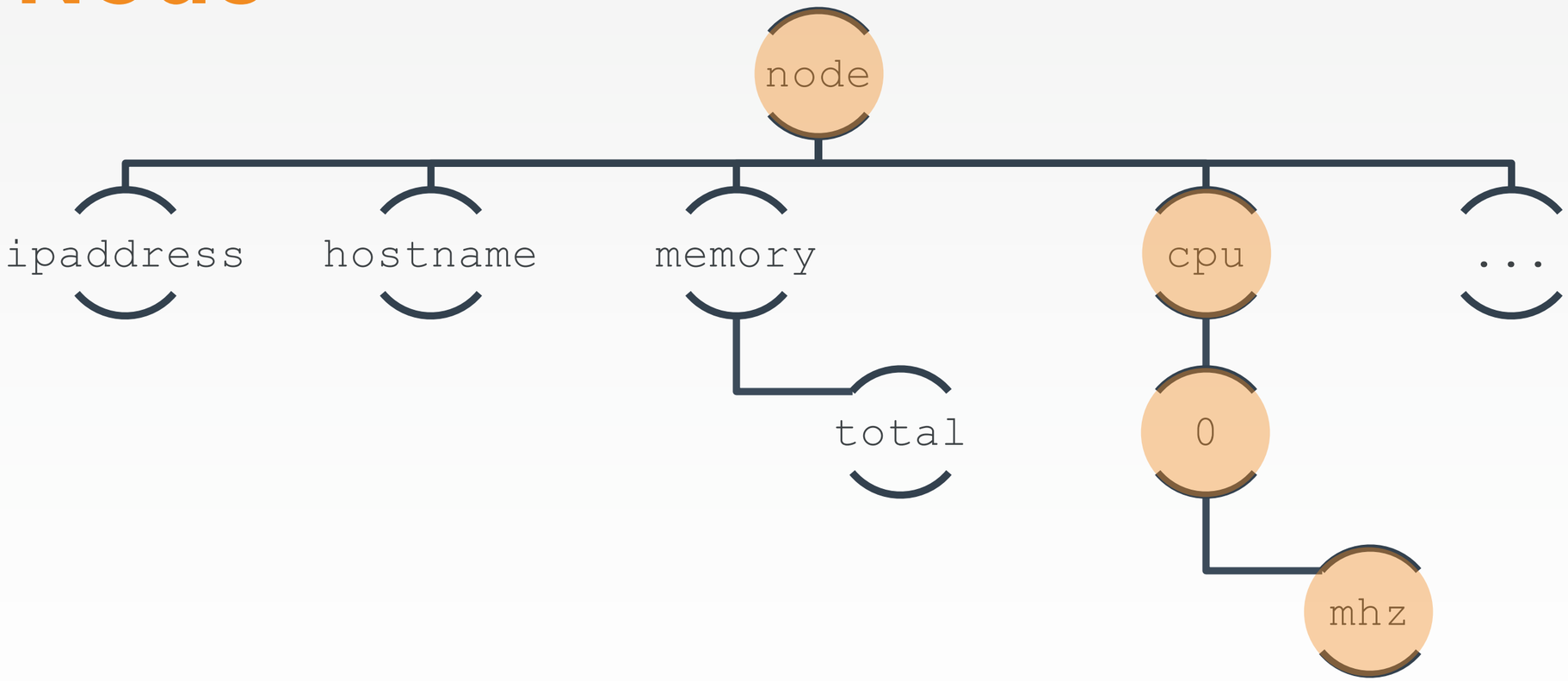



```
MEMORY: 502272kB
```

```
"Memory: #{node['memory']['total']}"
```


CHEF

The Node




```
CPU: 2399.998MHz
```

```
"CPU: #{node['cpu']['0']['mhz']}"
```


CHEF

CONCEPT

String Interpolation

```
I have 4 apples
```

```
apple_count = 4
puts "I have #{apple_count} apples"
```

http://en.wikipedia.org/wiki/String_interpolation#Ruby

CHEF

CONCEPT

String Interpolation

```
I have 4 apples
```

```
apple_count = 4
puts "I have #{apple_count} apples"
```

http://en.wikipedia.org/wiki/String_interpolation#Ruby

CHEF

CONCEPT

String Interpolation

```
I have 4 apples
```

```
apple_count = 4
puts "I have #{apple_count} apples"
```


CHEF

GL: Using the Node's Attributes

~\cookbooks\webserver\recipes\default.rb

```
...

file '/var/www/html/index.html' do
  content "<h1>Hello, world!</h1>

  <h2>IPADDRESS: #{node['ipaddress']}</h2>
  <h2>HOSTNAME: #{node['hostname']}</h2>
  <h2>MEMORY: #{node['memory']['total']}</h2>
  <h2>CPU: #{node['cpu']['0']['mhz']}</h2>
"
end

...
```

GL: Verify the Changes

- ❑ Change directory into the ”webserver" cookbook's directory
- ❑ Run kitchen converge for the ”webserver" cookbook
- ❑ Run kitchen verify for the ”webserver" cookbook

GL: Converge and Verify the Test Instance

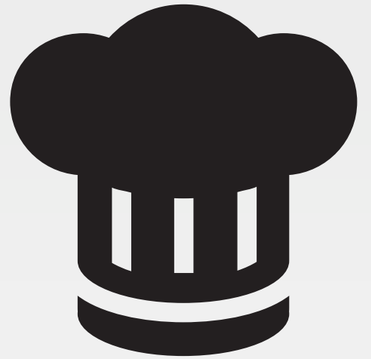
```
> cd ~\cookbooks\webserver
> kitchen converge
> kitchen verify
```

```
...
-----> Starting Kitchen (v1.13.2)
-----> Converging <default-centos-6>...
       Preparing files for transfer
       Preparing dna.json
       Resolving cookbook dependencies with Berkshelf 5.1.0...
       Removing non-cookbook files before transfer
       Preparing validation.pem
       Preparing client.rb
-----> Chef Omnibus installation detected (install only if missing)
```


CHEF

GL: Verify the Changes

- ✓ Change directory into the ”webserver" cookbook's directory
- ✓ Run kitchen converge for the ”webserver" cookbook
- ✓ Run kitchen verify for the ”webserver" cookbook



Changes Mean a New Version

Let's bump the version number and check in the code to source control.

Objective:

- ❑ Update the version of the ”webserver" cookbook
- ❑ Commit the changes to the ”webserver" cookbook to version control

CONCEPT

Cookbook Versions

A cookbook version represents a set of functionality that is different from the cookbook on which it is based.

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

CHEF

CONCEPT

Semantic Versions

Given a version number **MAJOR**.**MINOR**.**PATCH**, increment the:

- **MAJOR** version when you make incompatible API changes
- **MINOR** version when you add functionality in a backwards-compatible manner
- **PATCH** version when you make backwards-compatible bug fixes

http://semver.org

CHEF

DISCUSSION

Major, Minor, or Patch?

What kind of changes did you make to the cookbook?

CHEF

GL: Update the Cookbook Version

~\cookbooks\webserver\metadata.rb

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

COMMIT

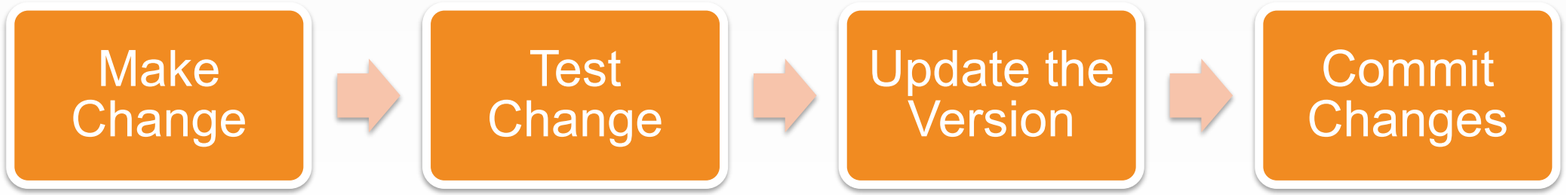
GL: Commit Your Work

```
> cd ~/cookbooks/webserver
> git add .
> git status
> git commit -m "Release version 0.2.0"
```


CHEF

CONCEPT

Development Workflow



CHEF

DISCUSSION

Discussion

What is the major difference between a single-quoted string and a double-quoted string?

How are the details about the system available within a recipe?

How does the version number help convey information about the state of the cookbook?

CHEF

DISCUSSION

Q&A

What questions can we help you answer?

- Ohai
- Node Object
- Node Attributes
- String Interpolation
- Semantic Versions

CHEF™