Ohai and the Node Object

Finding and Displaying Information About Our System

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

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?

Some Useful System Data

- platform
- hostname
- memory
- CPU - MHz

Demo: Finding Platform Info

```
> Get-WMIObject Win32_OperatingSystem
```

```
SystemDirectory : C:\Windows\system32
Organization    : Amazon.com
BuildNumber     : 9600
RegisteredUser  : EC2
SerialNumber    : 00252-70000-00000-AA535
Version         : 6.3.9600
```

Demo: Finding the Hostname

```
> $env:computername
```

```
WIN-KRQSVD3RFM7
```

Demo: Finding the Total Memory

```
> wmic ComputerSystem get TotalPhysicalMemory
```

```
TotalPhysicalMemory
8052654080
```

Demo: Finding the CPU MHz

```
> wmic cpu get name
```

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

Capturing System Data

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

How accurate will our recipe be if we hard code this information within our resources?

CONCEPT

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!

Data In Real Time

How could we capture this data in real-time?

CONCEPT

Ohai!

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

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

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

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

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

GL: Details About the Node

Displaying system details in the default web page definitely sounds useful.

Objective:

- ❑ Discover attributes about the system with Ohai
- ❑ Update the web page file contents, in the ”myiis" cookbook, to include system details
- ❑ Update the cookbook's version number
- ❑ Apply the updated recipe and verify the results

GL: Running Ohai!

```
> ohai
```

```
{
  "kernel": {
    "os_info": {
      "boot_device": "\\Device\\HarddiskVolume1",
      "build_number": "9600",
      "build_type": "Multiprocessor Free",
      "caption": "Microsoft Windows Server 2012 R2 Standard",
      "code_set": "1252",
      "country_code": "1",
      "creation_class_name": "Win32_OperatingSystem",
      "cs_creation_class_name": "Win32_ComputerSystem",
      "csd_version": null,
      "cs_name": "WIN-DCK5NTVVLBH",
```

GL: Running Ohai to Show the Platform

```
> ohai platform
```

```
[
  "windows”
]
```

GL: Running Ohai to Show the Hostname

```
> ohai hostname
```

```
[
  "WIN-NJ5OO7IAJNR"
]
```

GL: Running Ohai to Show the Memory

```
> ohai memory
```

```
{
  "swap": {
    "total": "8388608kB",
    "free": "8388608kB"
  },
  "total": "8388208kB",
  "free": "6841060kB"
}
```

GL: Running Ohai to Show the Total Memory

```
> ohai memory/total
```

```
[
  "8388208kB"
]
```

GL: Running Ohai to Show the CPU

```
> ohai cpu
```

```
{
  "0": {
    "cores": 2,
    "vendor_id": "GenuineIntel",
    "family": "1",
    "model": "16130",
    "stepping": "2",
    "physical_id": "CPU0",
    "model_name": "Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz",
    "description": "Intel64 Family 6 Model 63 Stepping 2",
    "mhz": "2400",
    "cache_size": " KB"
  },
  "total": 2,
  "cores": 2,
  "real": 1

}
```

GL: Running Ohai to Show the First CPU

```
> ohai cpu/0
```

```
{
  "cores": 2,
  "vendor_id": "GenuineIntel",
  "family": "1",
  "model": "16130",
  "stepping": "2",
  "physical_id": "CPU0",
  "model_name": "Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz",
  "description": "Intel64 Family 6 Model 63 Stepping 2",
  "mhz": "2400",
  "cache_size": " KB"
}
```

GL: Running Ohai to Show the First CPU MHz

```
> ohai cpu/0/mhz
```

```
[
  "2400"
]
```

CONCEPT

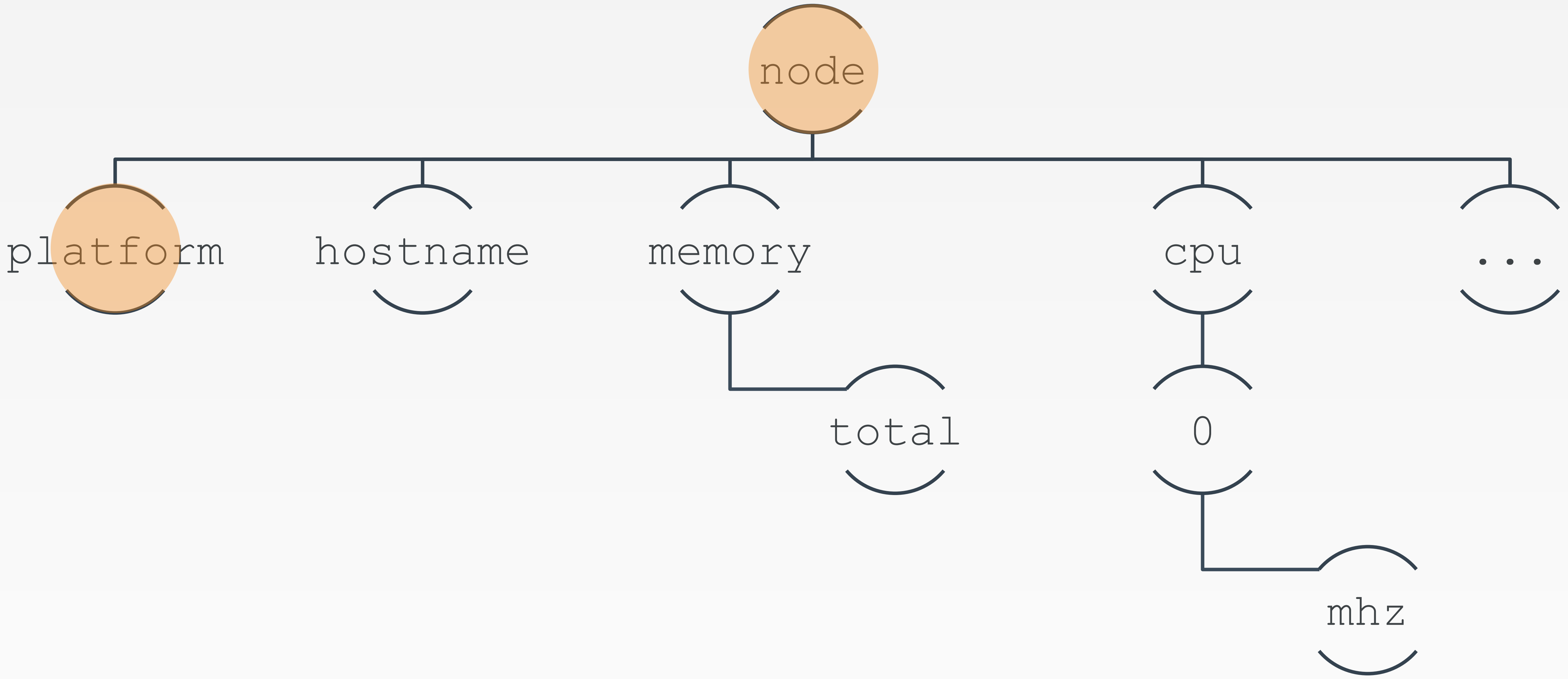
The Node Object

The node object is accessible within recipes as well as from the command line.

Let's take a look at the syntax.

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

The Node

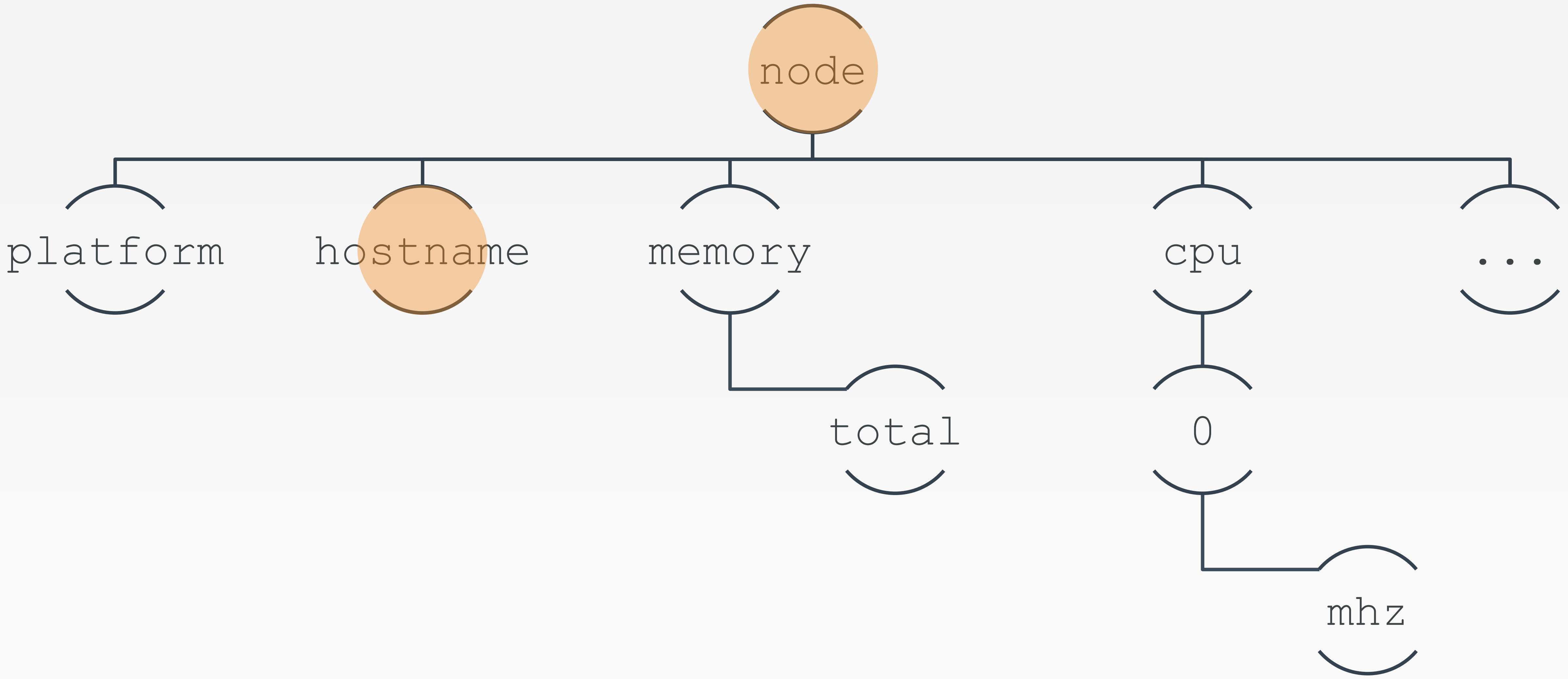


```
CLI:     ohai platform
```

```
RECIPE:  node['platform']
```

```
OUTPUT:  windows
```

The Node

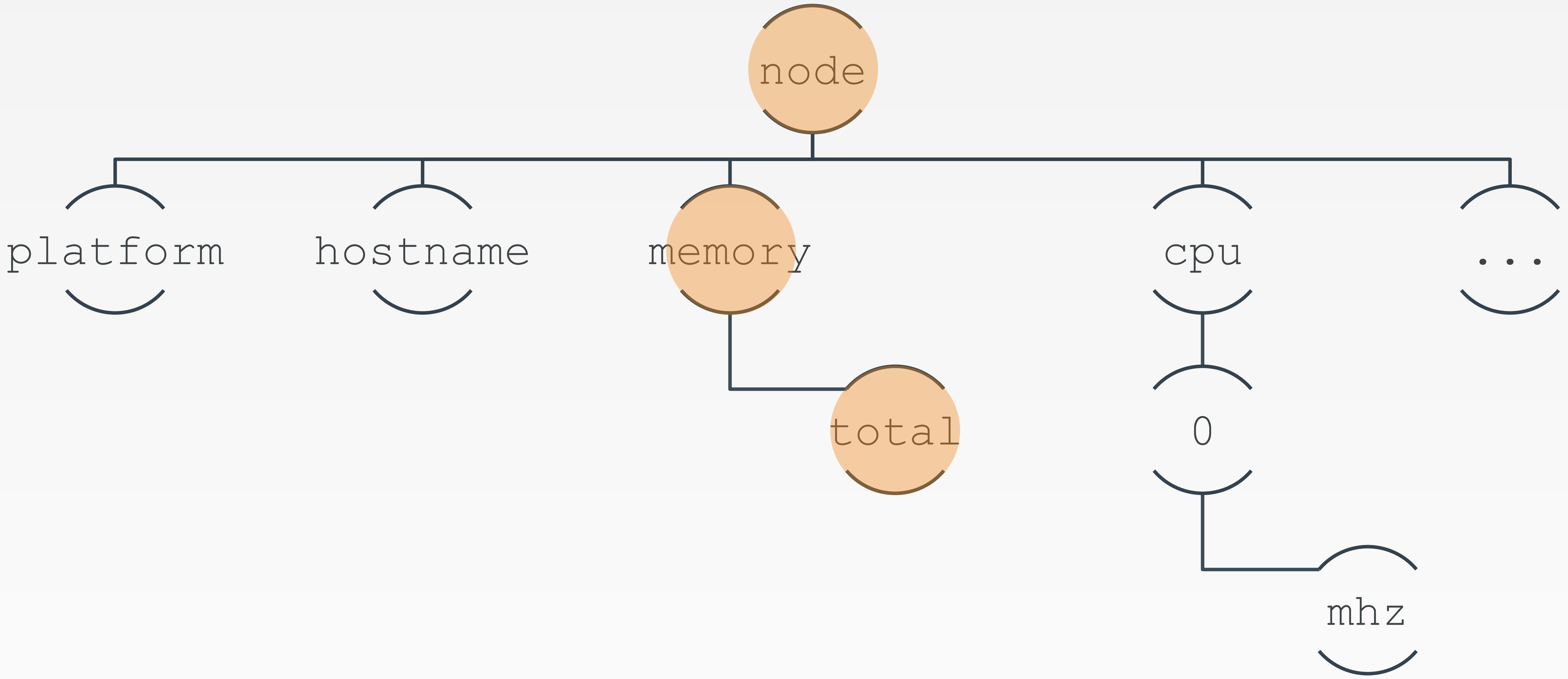


```
CLI:     ohai hostname
```

```
RECIPE:  node['hostname']
```

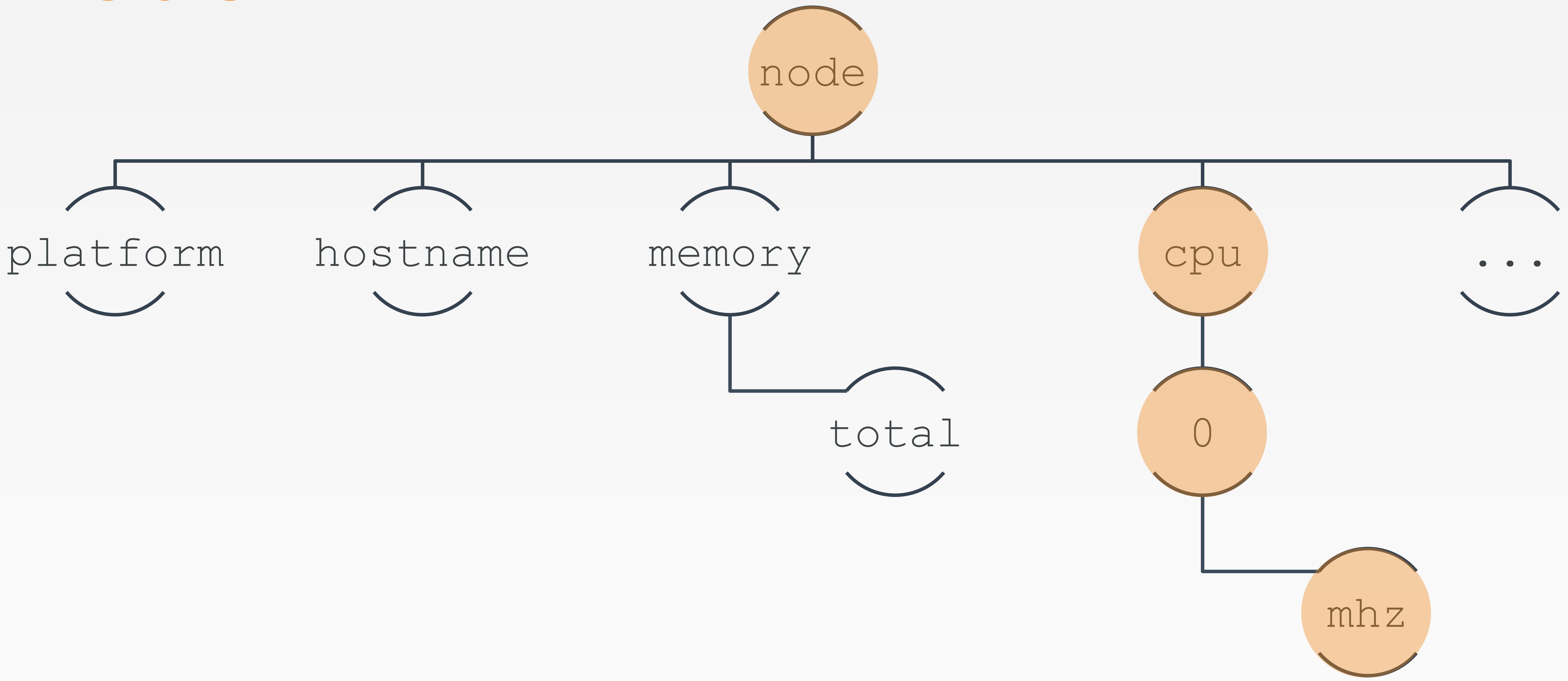
```
OUTPUT:  WIN-KRQSVD3RFM7
```

The Node



```
CLI:     ohai memory/total
RECIPE:  node['memory']['total']
OUTPUT:  7863920kB
```

The Node



```
CLI:     ohai cpu/0/mhz
RECIPE:  node['cpu']['0']['mhz']
OUTPUT:  2900
```

CONCEPT

String Interpolation

```
I have 4 apples
```

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

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

CONCEPT

String Interpolation

```
I have 4 apples
```

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

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

String Interpolation

```
I have 4 apples
```

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

GL: Details About the Node

Displaying system details in the default web page definitely sounds useful.

Objective:

- ✔ Discover attributes about the system with Ohai
- ❑ Update the web page file contents, in the ”myiis" cookbook, to include system details
- ❑ Update the cookbook's version number
- ❑ Apply the updated recipe and verify the results

GL: Details About the Node

~\cookbooks\myiis\recipes\server.rb

```
# ... POWERSHELL_SCRIPT RESOURCE ...

file 'c:\inetpub\wwwroot\Default.htm' do
  content "<h1>Hello, world!</h1>
<h2>PLATFORM: #{node['platform']}</h2>
<h2>HOSTNAME: #{node['hostname']}</h2>
<h2>MEMORY:   #{node['memory']['total']}</h2>
<h2>CPU Mhz:  #{node['cpu']['0']['mhz']}</h2>"
end

# ... SERVICE RESOURCE ...
```

GL: Details About the Node

Displaying system details in the default web page definitely sounds useful.

Objective:

- ✔ Discover attributes about the system with Ohai
- ✔ Update the web page file contents, in the ”myiis" cookbook, to include system details
- ❏ Update the cookbook's version number
- ❏ Apply the updated recipe and verify the results

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

CONCEPT

Semantic Versions

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

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

http://semver.org

CHEF

Major, Minor, or Patch?

What kind of changes did you make to the cookbook?

CHEF™

GL: Update the Cookbook Version

```
~\cookbooks\myiis\metadata.rb
```

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

GL: Details About the Node

Displaying system details in the default web page definitely sounds useful.

Objective:

- ✔ Discover attributes about the system with Ohai
- ✔ Update the web page file contents, in the ”myiis" cookbook, to include system details
- ✔ Update the cookbook's version number
- ❑ Apply the updated recipe and verify the results

GL: Return Home and Apply myiis Cookbook

```
> cd ~
> chef-client --local-mode -r "recipe[myiis]"
```

```
...
Synchronizing Cookbooks:
  - myiis (0.2.0)
...
* powershell_script[Install IIS] action run
    - execute "C:\Windows\system32\WindowsPowerShell\v1.0\powershell.exe" -NoLogo -NonInteractive -NoProfile
-ExecutionPolicy Bypass -Input

Format None -File "C:/Users/ADMINI~1/AppData/Local/Temp/2/chef-script20200205-4560-giw545.ps1"
  * file[C:\inetpub\wwwroot\Default.htm] action create
    - update content in file C:\inetpub\wwwroot\Default.htm from 17d291 to afadfd
    --- C:\inetpub\wwwroot\Default.htm  2020-02-05 18:15:08.678449100 +0000
    +++ C:\inetpub\wwwroot/chef-Default20200205-4560-e2bb3h.htm 2020-02-05 18:29:46.178300200 +0000
    @@ -1,2 +1,6 @@
     <h1>Hello, world!</h1>
    +<h2>PLATFORM: windows</h2>
    +<h2>HOSTNAME: WIN-NJ5OO7IAJNR</h2>
    +<h2>MEMORY:   8283740kB</h2>
    +<h2>CPU Mhz:  2500</h2>
Running handlers:...
```

GL: Verify the Default Page Returns the Details

```
> Invoke-WebRequest localhost
```

```
StatusCode        : 200
StatusDescription : OK
Content           : <h1>Hello, world!</h1>
                    <h2>PLATFORM: windows</h2>
                    <h2>HOSTNAME: WIN-8694LT97S51</h2>
                    <h2>MEMORY:   8388208kB</h2>
                    <h2>CPU Mhz:  2400</h2>
RawContent        : HTTP/1.1 200 OK
                    Accept-Ranges: bytes
                    Content-Length: 137
...
```

GL: Details About the Node

Displaying system details in the default web page definitely sounds useful.

Objective:

- ✔ Discover attributes about the system with Ohai
- ✔ Update the web page file contents, in the ”myiis" cookbook, to include system details
- ✔ Update the cookbook's version number
- ✔ Apply the updated recipe and verify the results

Review

1. What is the node object and when is this generated?

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

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

Q&A

What questions can we help you answer?

- Ohai
- Node Object
- Node Attributes
- String Interpolation

CHEF™