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
- Update the version of a cookbook

System Data in MOTD file

- Lets say you needed to update the MOTD file contents, in the "workstation" cookbook, to include node details
  - ❑ IP Address
  - ❑ hostname
  - ❑ memory
  - ❑ CPU - MHz

The Manual Way - Discover the IP Address

```
$ ifconfig
```

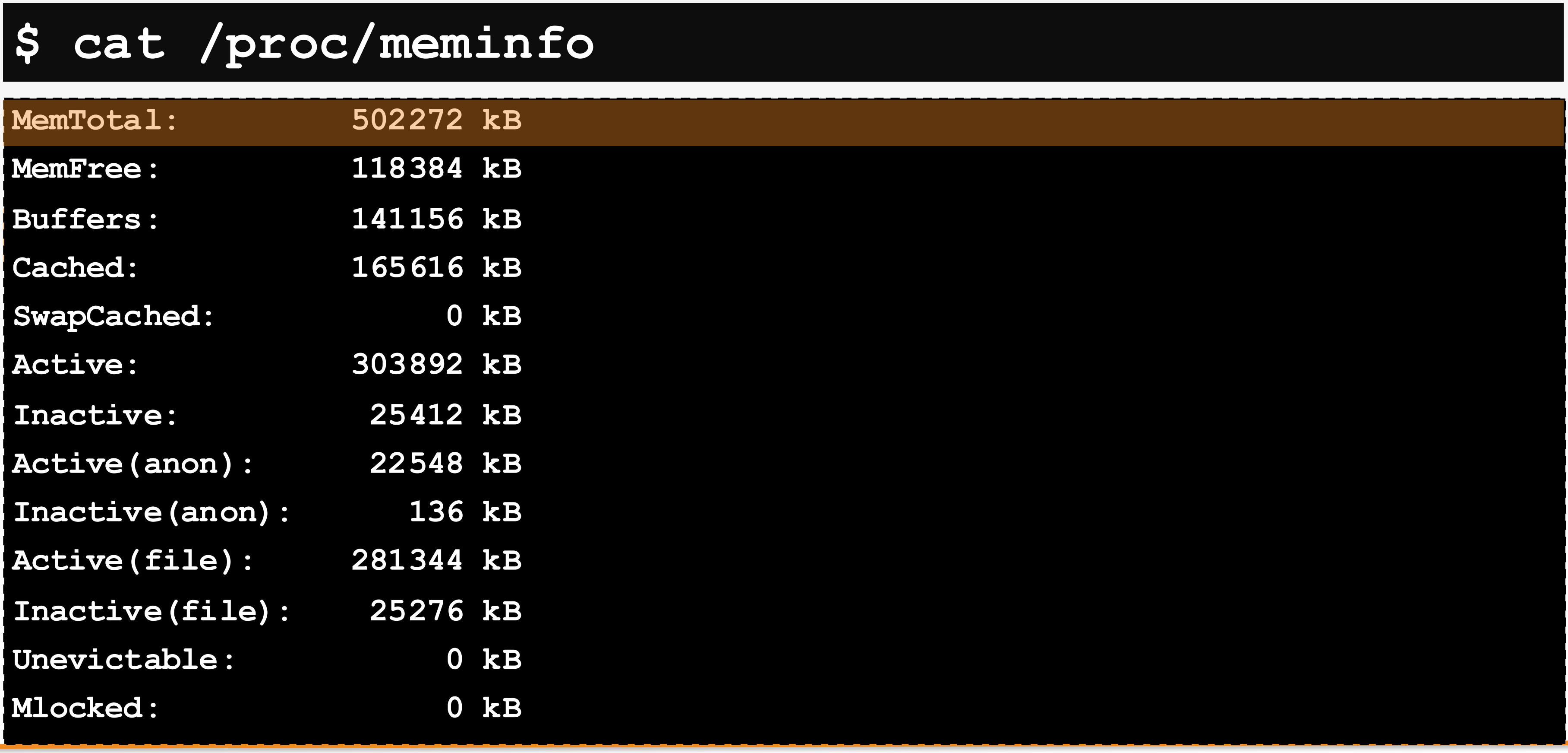
```
docker0   Link encap:Ethernet  HWaddr 56:84:7A:FE:97:99
          inet addr:172.17.42.1  Bcast:0.0.0.0  Mask:255.255.0.0
          inet6 addr: fe80::5484:7aff:fefe:9799/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:25870 errors:0 dropped:0 overruns:0 frame:0
          TX packets:128971 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1459392 (1.3 MiB)  TX bytes:190819384 (181.9 MiB)

eth0      Link encap:Ethernet  HWaddr 0A:4D:03:F7:91:D7
          inet addr:172.31.8.68  Bcast:172.31.15.255  Mask:255.255.240.0
          inet6 addr: fe80::84d:3ff:fef7:91d7/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

The Manual Way - Discover the Host Name

```
$ hostname
```

```
banana-stand
```

The Manual Way - Discover the Memory



```
$ cat /proc/meminfo
```

```
MemTotal:          502272 kB
MemFree:           118384 kB
Buffers:           141156 kB
Cached:            165616 kB
SwapCached:             0 kB
Active:            303892 kB
Inactive:           25412 kB
Active(anon):       22548 kB
Inactive(anon):       136 kB
Active(file):      281344 kB
Inactive(file):     25276 kB
Unevictable:            0 kB
Mlocked:                0 kB
```

The Manual Way - Discover the CPU MHz

```
$ cat /proc/cpuinfo
```

```
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 62
model name      : Intel(R) Xeon(R) CPU E5-2630L v2 @ 2.40GHz
stepping        : 4
cpu MHz         : 2399.998
cache size      : 15360 KB
fpu             : yes
fpu_exception   : yes
cpuid level     : 13
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat
pse36
```

The Manual Way - Add info to the recipe

`cookbooks/workstation/recipes/setup.rb`

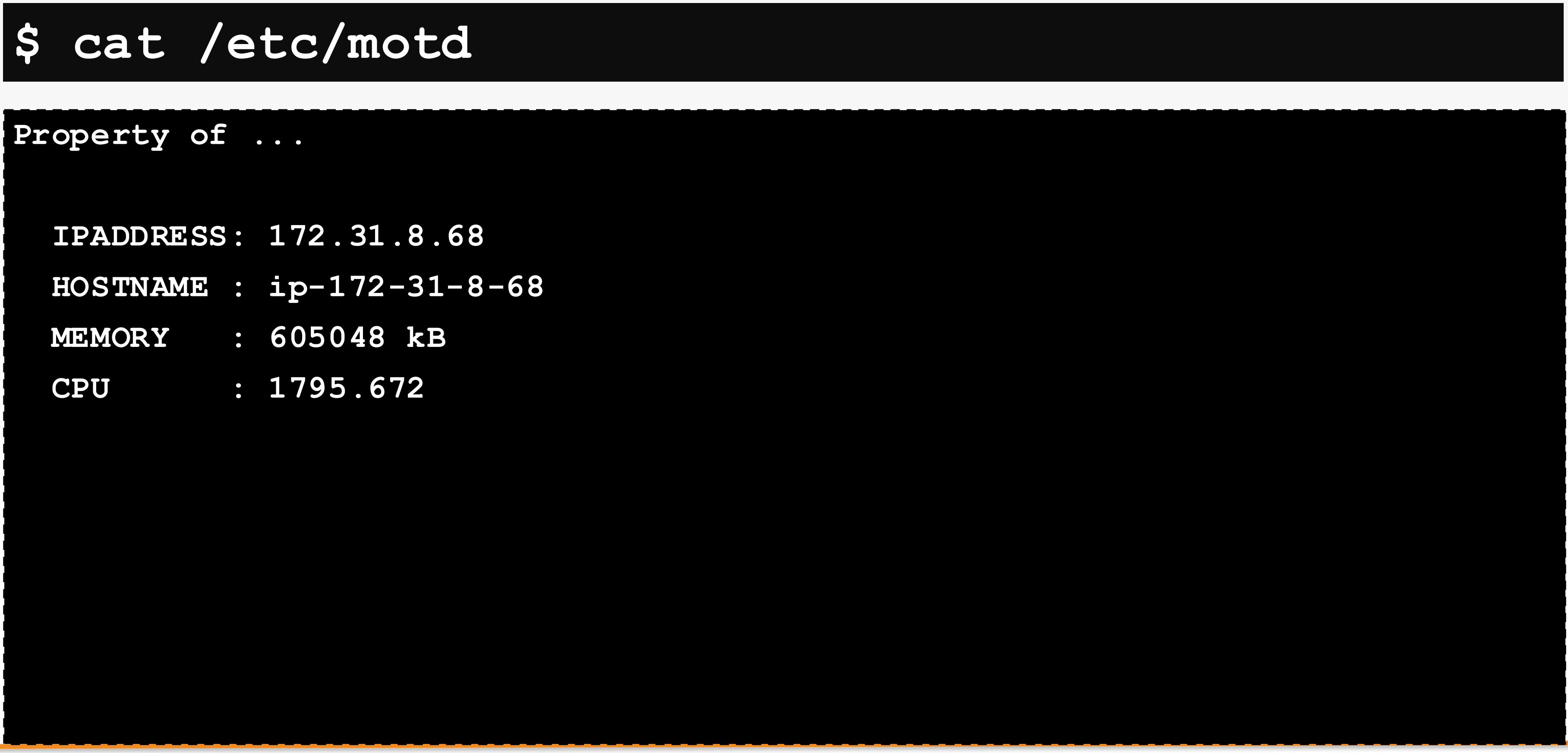
```
file '/etc/motd' do
  content 'Property of ...

  IPADDRESS: 104.236.192.102
  HOSTNAME : banana-stand
  MEMORY   : 502272 kB
  CPU      : 2399.998 MHz
'
  mode '0644'
  owner 'root'
  group 'root'
end
```

The Manual Way - Apply workstation Cookbook

```
$ sudo chef-client -zr "recipe[workstation]"
```

```
resolving cookbooks for run list: ["workstation"]
Synchronizing Cookbooks:
  - workstation
Compiling Cookbooks...
...
```

The Manual Way - Verify that the /etc/motd Has Been Updated



```
$ cat /etc/motd
```

```
Property of ...


  IPADDRESS: 172.31.8.68
  HOSTNAME : ip-172-31-8-68
  MEMORY   : 605048 kB
  CPU      : 1795.672
```

DISCUSSION

Capturing System Data

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

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

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

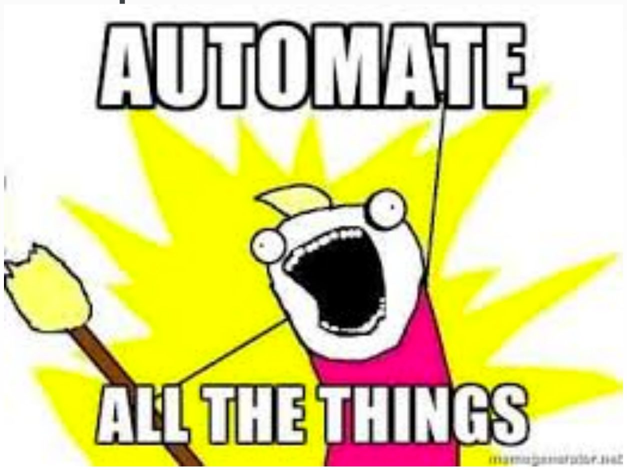
Hard Coded Values

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

We need to capture this data in real-time!



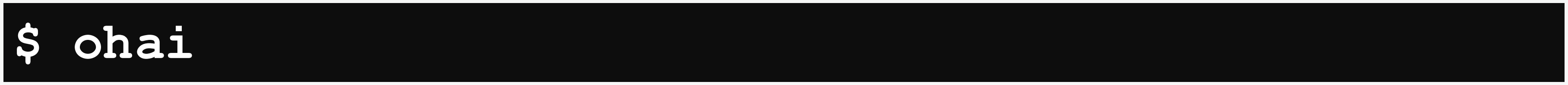
CONCEPT

Ohai!

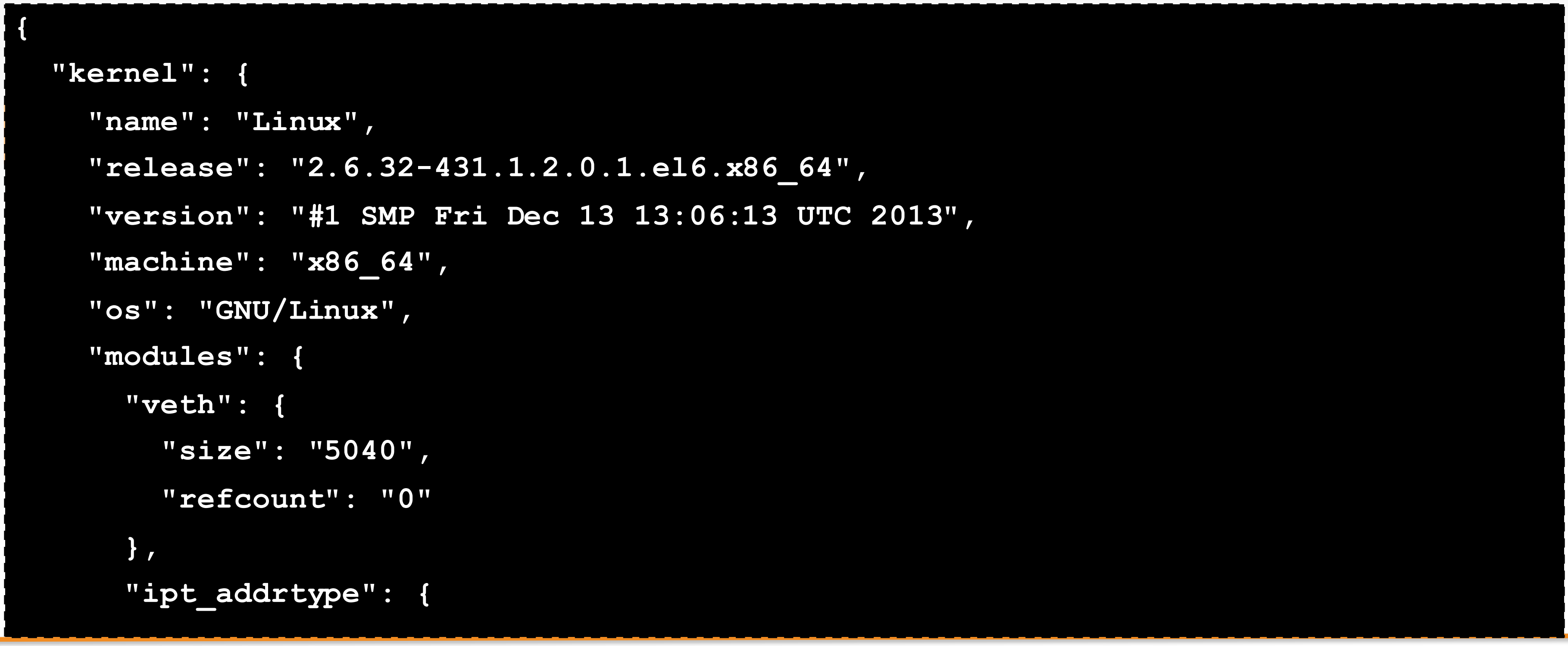
Ohai is a system profiling tool that captures all this data (and much more!)

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

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": {
```

CONCEPT

ohai + chef-client

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

The data is presented in JSON (JavaScript Object Notation)

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

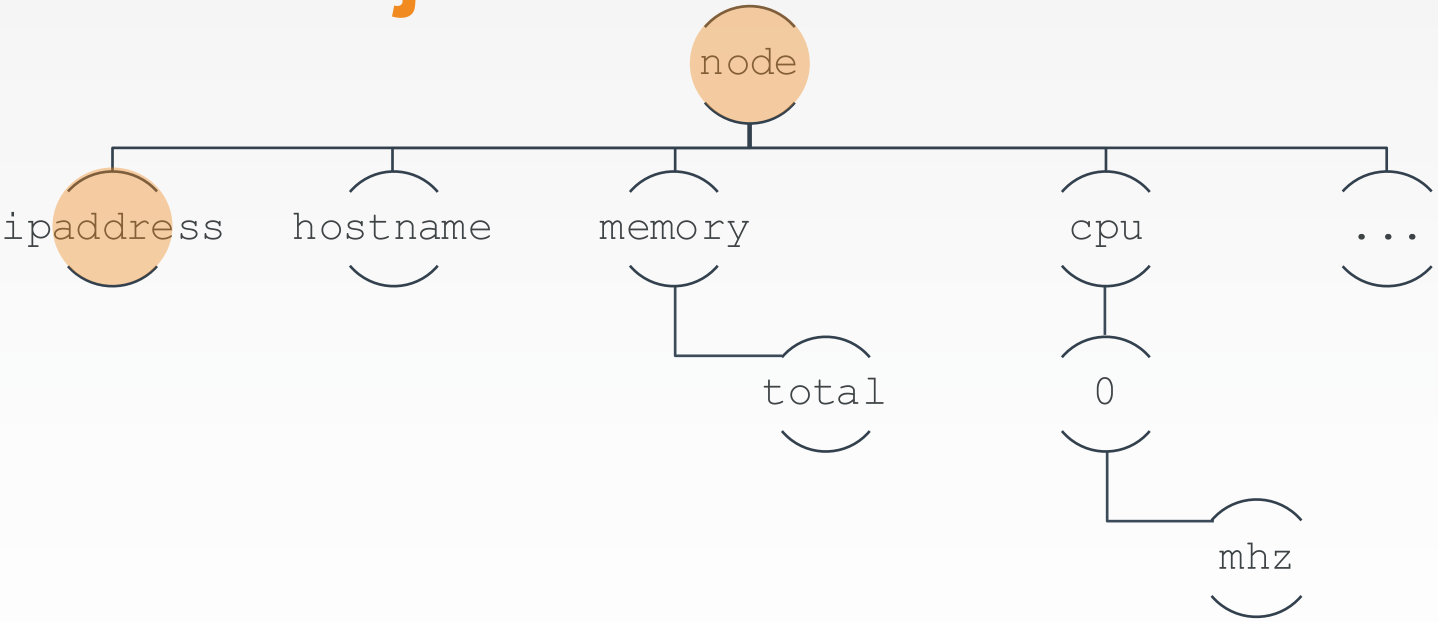
CONCEPT

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

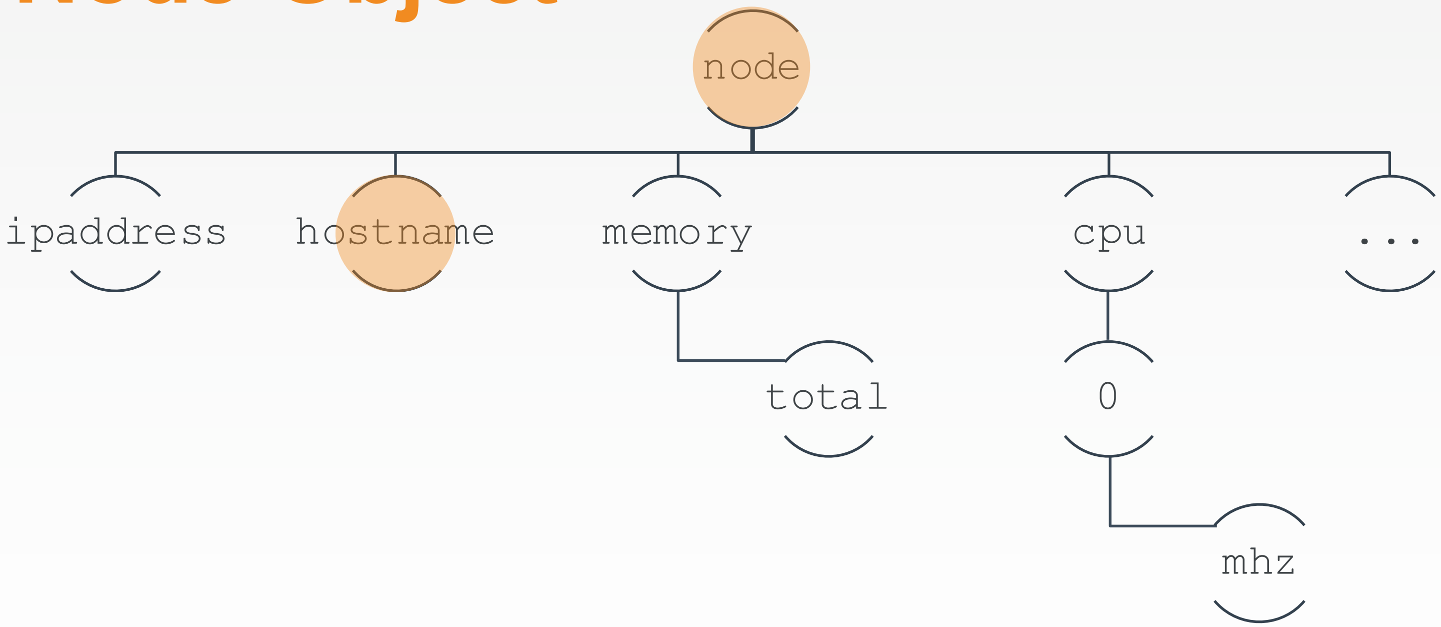
The Node Object



```
IPADDRESS: 104.236.192.102
```

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

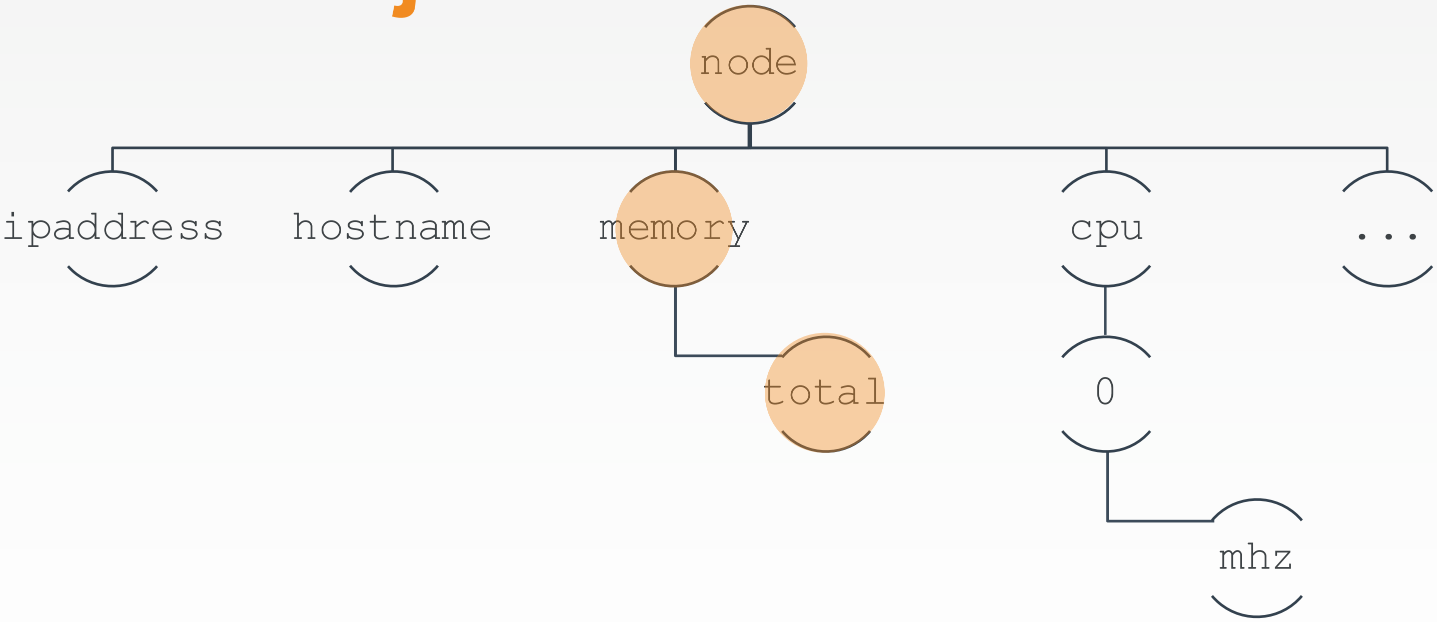
The Node Object



```
HOSTNAME: banana-stand
```

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

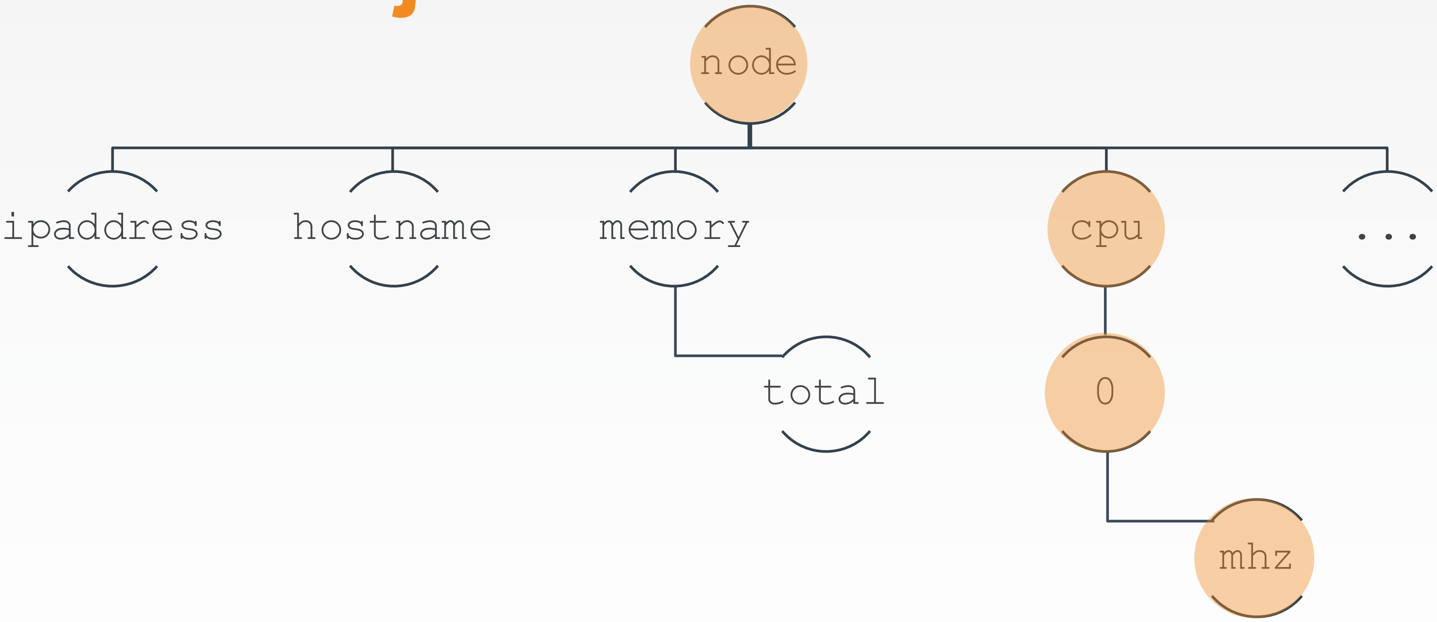
The Node Object



```
MEMORY: 502272kB
```

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

The Node Object



```
CPU: 2399.998MHz
```

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

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"
```

Lab: Using the Node's Attributes

cookbooks/workstation/recipes/setup.rb

```
# ... PACKAGE RESOURCES ...
file '/etc/motd' do
  content "Property of ...

  IPADDRESS: #{node['ipaddress']}
  HOSTNAME : #{node['hostname']}
  MEMORY   : #{node['memory']['total']}
  CPU      : #{node['cpu']['0']['mhz']}
"
  mode '0644'
  owner 'root'
  group 'root'
end
```

Lab: Return Home and Apply workstation Cookbook

```
$ cd chef-repo
$ sudo chef-client -zr "recipe[workstation]"
```

```
...
    +
    +  IPADDRESS: 172.31.8.68
    +  HOSTNAME : ip-172-31-8-68
    +  MEMORY   : 604308kB
    +  CPU      : 1795.672

Running handlers:
Running handlers complete
Chef Client finished, 1/2 resources updated in 08 seconds
```

Lab: Verify that the /etc/motd Has Been Updated

```
$ cat /etc/motd
```

```
Property of ...


  IPADDRESS: 172.31.8.68
  HOSTNAME : ip-172-31-8-68
  MEMORY   : 605048 kB
  CPU      : 1795.672
```

Changes Mean a New Version

Let's bump the version number

Objective:

❑ Update the version of the "workstation" cookbook

CONCEPT

Semantic Versions

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

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

http://semver.org

Lab: Update the Cookbook Version

cookbooks/workstation/metadata.rb

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

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?

DISCUSSION

Q&A

What questions can we help you answer?

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

CHEF™