Ohai Plugins

©2017 Chef Software Inc.

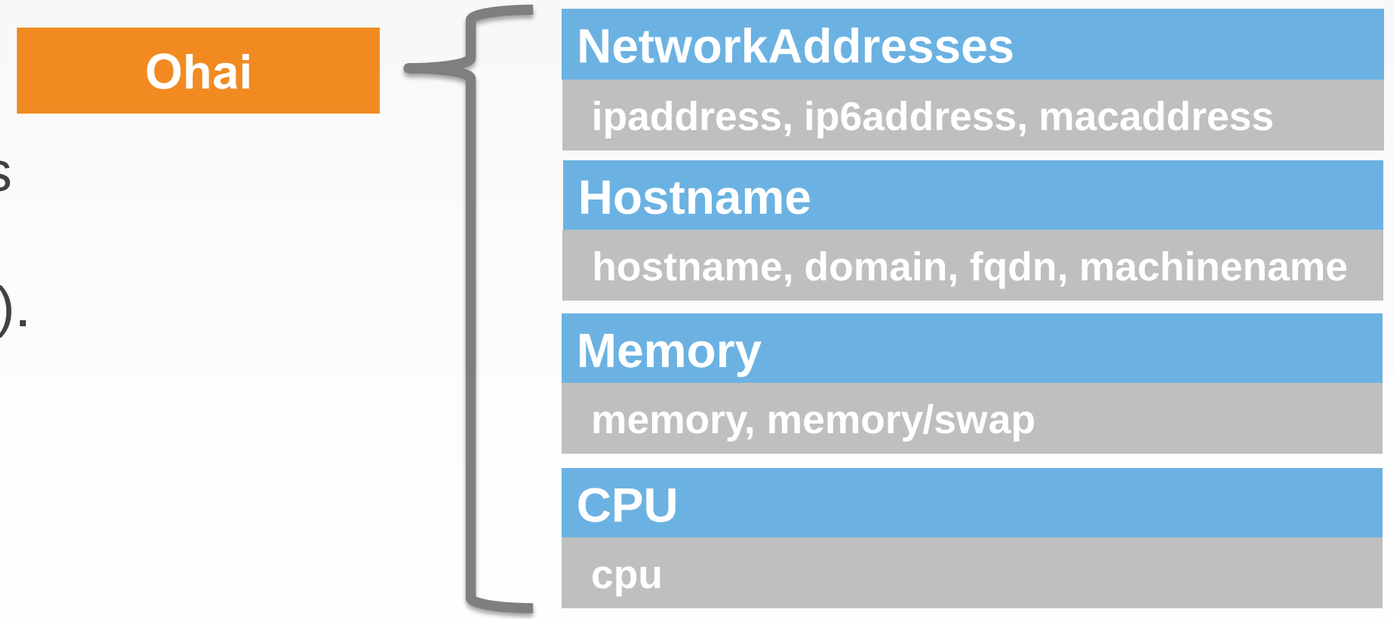
CHEF

CONCEPT

Ohai is Composed of Plugins

Ohai is packaged with a core set of plugins that are automatically loaded when executing Ohai.

These plugins provide the attributes we see in the JSON output (e.g. ipaddress, hostname, memory, cpu).



©2017 Chef Software Inc.

CHEF

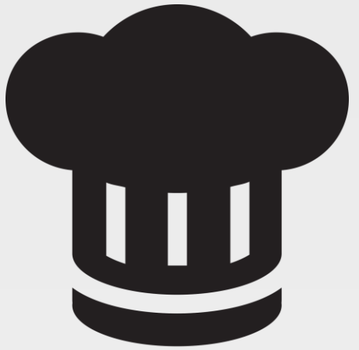
Objectives

After completing this module, you should be able to:

- Find Ohai's core plugins
- Express what a plugin provides, depends on, and how it collects its data

©2017 Chef Software Inc.

EXERCISE



Reviewing the Ohai Gem

Ohai is a Rubygem. First we need to learn about how a gem is structured.

Objective:

- ❑ Review the basic structure of the Ohai gem
- ❑ Review the 'language' plugin
- ❑ Review the 'python' plugin

©2017 Chef Software Inc.

CHEF

CONCEPT

Ohai is Ruby Gem

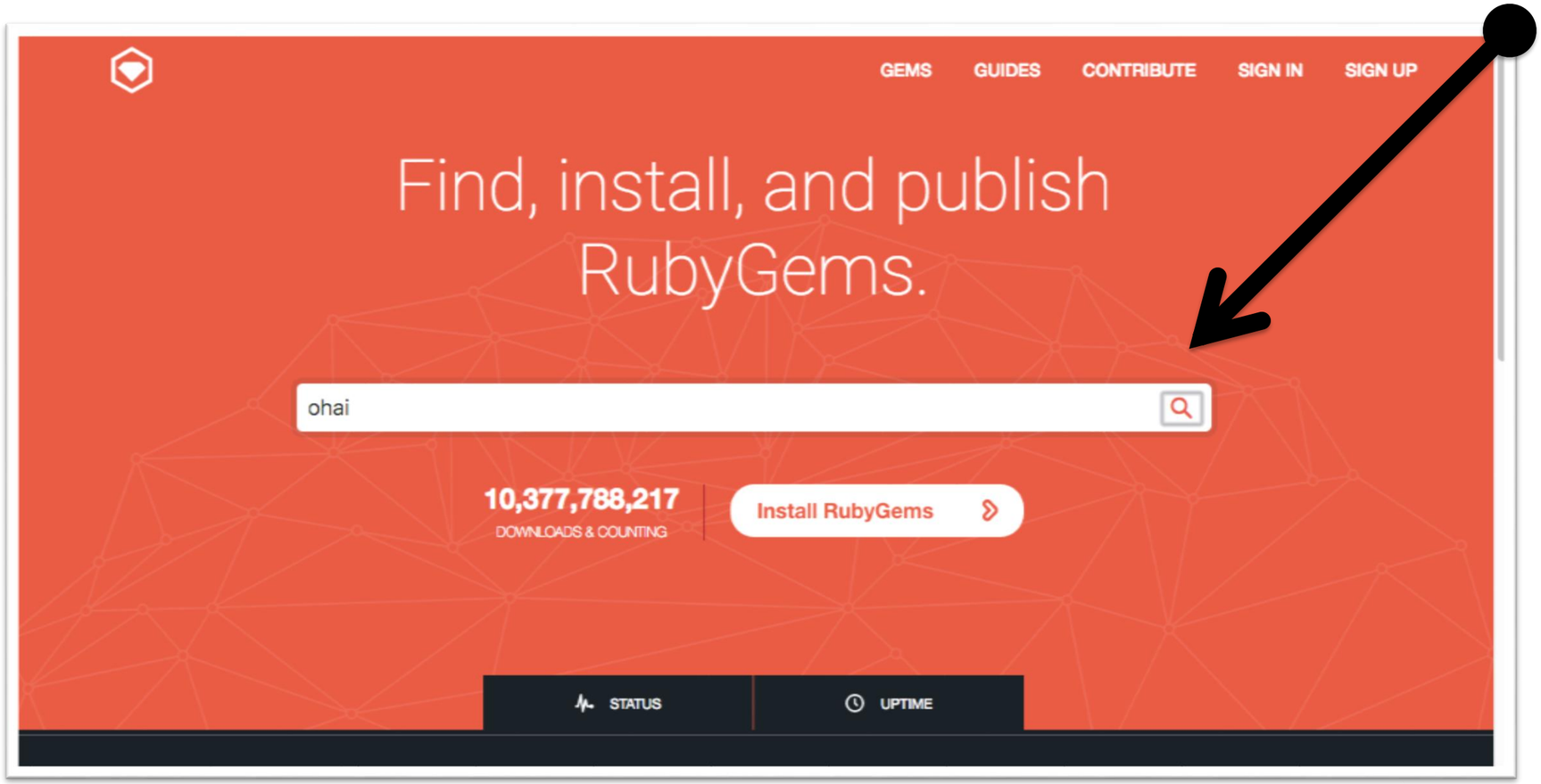
Ruby gems are the ways in which Ruby developers share the code that they develop with others. A Ruby gem is really a packaging structure similar to that of a Chef cookbook.

©2017 Chef Software Inc.

Searching for a Gem on Rubygems

Steps

1. Visit https://rubygems.org
2. Within the search field enter: **ohai**
3. Press enter or click the magnified glass at the right-side of the search box.
4. Click the 'Source Code' link
5. Click on 'Clone or download' and then copy the git URL.

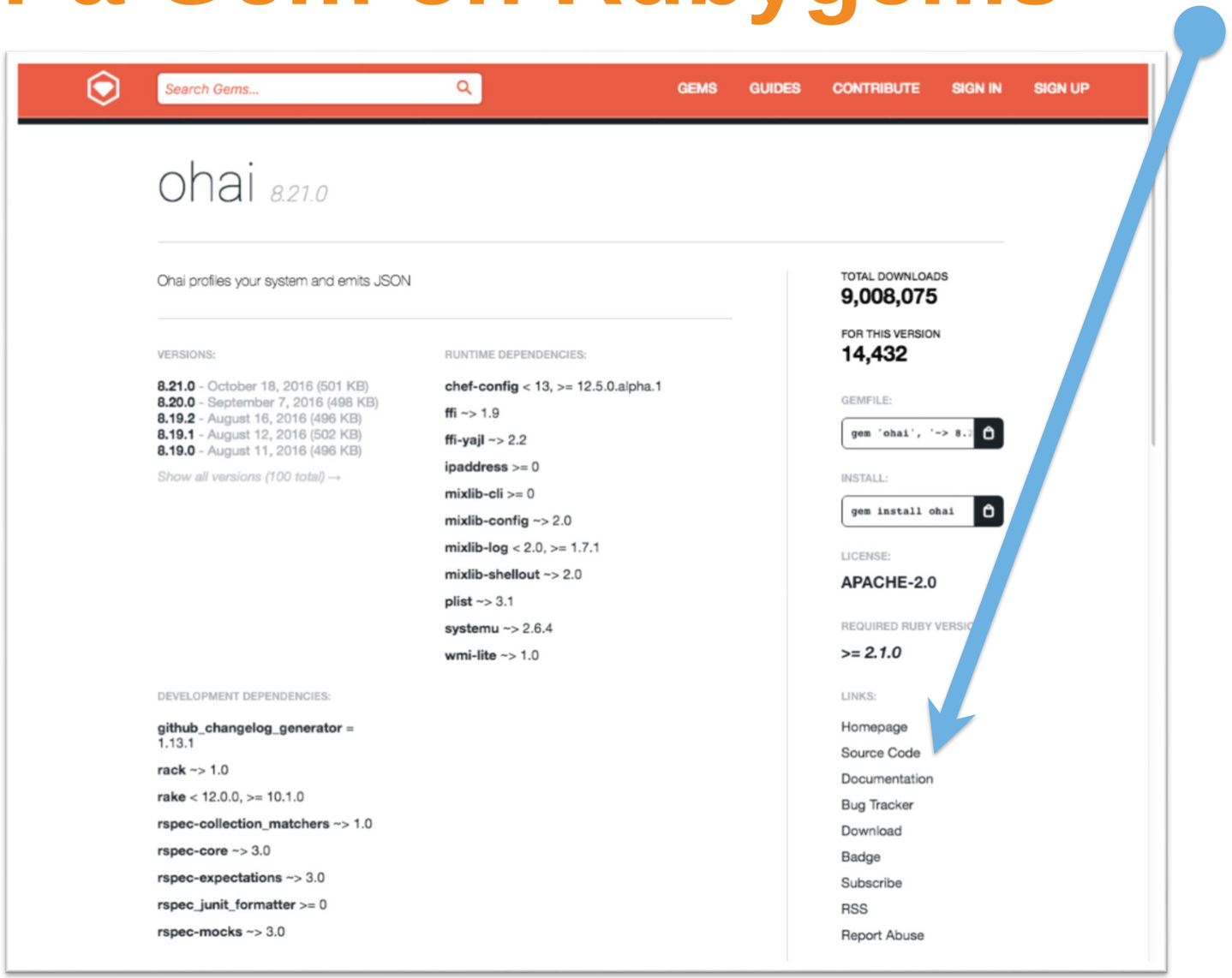


©2017 Chef Software Inc.

Searching for a Gem on Rubygems

Steps

1. Visit https://rubygems.org
2. Within the search field enter: **ohai**
3. Press enter or click the magnified glass at the right-side of the search box.
4. Click the 'Source Code' link
5. Click on 'Clone or download' and then copy the git URL.



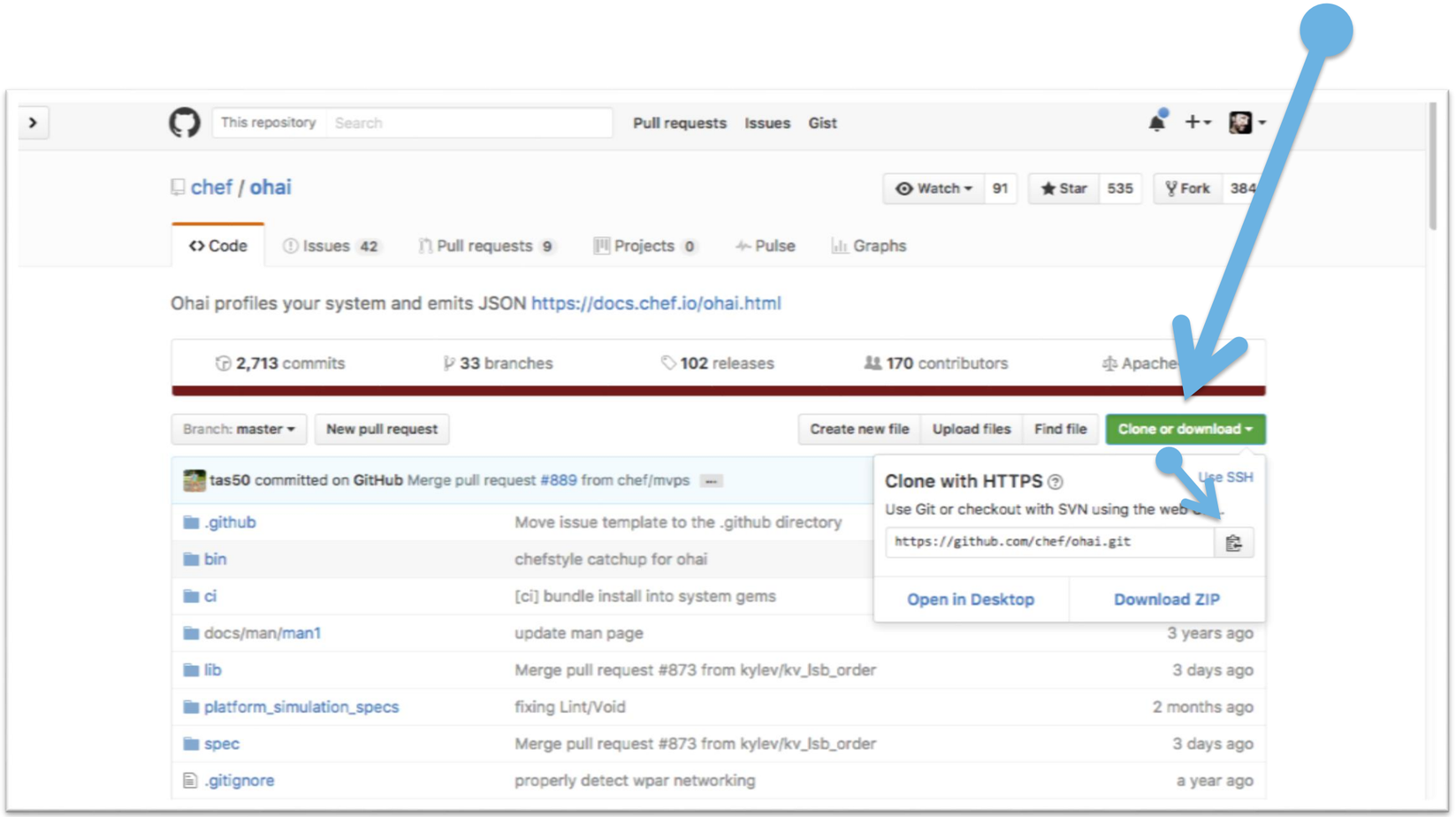
©2017 Chef Software Inc.

CHEF

Searching for a Gem on Rubygems

Steps

1. Visit https://rubygems.org
2. Within the search field enter: **ohai**
3. Press enter or click the magnified glass at the right-side of the search box.
4. Click the 'Source Code' link
5. Click on 'Clone or download' and then copy the git URL.



©2017 Chef Software Inc.

CHEF

Returning to the Home Directory

```
> cd ~
```

©2017 Chef Software Inc.

CHEF

Cloning the Ohai Library

```
> git clone https://github.com/chef/ohai.git
```

```
Cloning into 'ohai'...
remote: Counting objects: 20571, done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 20571 (delta 7), reused 0 (delta 0), pack-reused 20540
Receiving objects: 100% (20571/20571), 4.46 MiB | 2.13 MiB/s, done.
Resolving deltas: 100% (13408/13408), done.
Checking connectivity... done.
```

©2017 Chef Software Inc.

CHEF

Viewing the Contents of the Project

```
> tree ohai
```

```
└── ohai
    ├── appveyor.yml
    ├── bin
    │   └── ohai
    ├── CHANGELOG.md
    ├── ci
    │   ├── jenkins_run_tests.bat
    │   └── jenkins_run_tests.sh
    ├── docs
    │   └── man
    │       └── man1
    │           └── ohai.1
    │
```

©2017 Chef Software Inc.

CHEF

Viewing the README

~/ohai/README.md

```
# ohai


[![Build Status Master](https://travis-ci.org/chef/ohai.svg?branch=master)](https://travis-
ci.org/chef/ohai) [![Build Status
Master](https://ci.appveyor.com/api/projects/status/github/chef/ohai?branch=master&svg=true&p
assingText=master%20-%20Ok&pendingText=master%20-%20Pending&failingText=master%20-
%20Failing)](https://ci.appveyor.com/project/Chef/ohai/branch/master) [![Gem
Version](https://badge.fury.io/rb/ohai.svg)](https://badge.fury.io/rb/ohai)


## Description


Ohai detects data about your operating system. It can be used standalone, but its primary
purpose is to provide node data to Chef.


Ohai will print out a JSON data blob for all the known data about your system. When used with
Chef, that data is reported back via node attributes.


Chef distributes ohai as a RubyGem. This README is for developers who want to modify the Ohai
source code. For users who want to write plugins for Ohai, see the docs:
```

©2017 Chef Software Inc.

CHEF

Viewing the Gem Specification

~/ohai/ohai.gemspec

```
$:.unshift File.expand_path("../lib", __FILE__)
require "ohai/version"

Gem::Specification.new do |s|
  s.name = "ohai"
  s.version = Ohai::VERSION
  s.platform = Gem::Platform::RUBY
  s.summary = "Ohai profiles your system and emits JSON"
  s.description = s.summary
  s.license = "Apache-2.0"
  s.author = "Adam Jacob"
  s.email = "adam@chef.io"
  s.homepage = "https://docs.chef.io/ohai.html"
```

©2017 Chef Software Inc.

CHEF

Viewing the lib Directory

```
> tree ohai/lib
```

```
ohai/lib
├── ohai
│   ├── application.rb
│   ├── common
│   │   └── dmi.rb
│   ├── config.rb
...
│   └── version.rb
└── ohai.rb
19 directories, 161 files
```

©2017 Chef Software Inc.

Viewing the ohai.rb file in the lib Directory

~/ohai/lib/ohai.rb

```
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.
#

require "ohai/version"
require "ohai/config"
require "ohai/system"
require "ohai/exception"
```

©2017 Chef Software Inc.

CHEF

Viewing the plugins directory

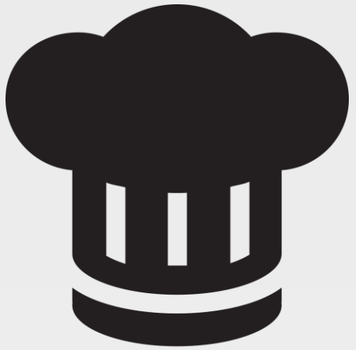
```
> tree ohai/lib/ohai/plugins
```

```
ohai/lib/ohai/plugins
├── aix
│   ├── cpu.rb
│   ├── filesystem.rb
│   ├── kernel.rb
│   ├── memory.rb
│   ├── network.rb
│   ├── os.rb
│   ├── platform.rb
│   ├── uptime.rb
│   └── virtualization.rb
├── azure.rb
├── bsd
│   ├── filesystem.rb
```

©2017 Chef Software Inc.

CHEF™

EXERCISE



Reviewing the Ohai Gem

Let's take a look at the structure of a plugin.

Objective:

- ✓ Review the basic structure of the Ohai gem
- ❑ Review the 'language' plugin
- ❑ Review the 'python' plugin

©2017 Chef Software Inc.

CHEF

CONCEPT

Recent Major Ohai Releases

Ohai 6

Released: April 13, 2011

Chef Version: 0.10.0

docs.chef.io/release/ohai-6

Ohai 7

Released: April 8, 2014

Chef Version: 11.12.0

docs.chef.io/release/ohai-7

Ohai 8

Released: Dec. 4, 2014

Chef Version: 11.18.0

docs.chef.io/release/ohai-8

©2017 Chef Software Inc.

CHEF

CONCEPT

Focus on Ohai 7

Ohai 7 refined the Domain Specific Language (DSL) created in the previous version of Ohai. Ohai 8 continues to use the same language.

Ohai 6 Ohai 7 Ohai 8

©2017 Chef Software Inc.

CHEF

Viewing the Languages Plugin

~/ohai/lib/ohai/plugins/languages.rb

```
Ohai.plugin(:Languages) do
  provides "languages"
  collect_data do
    languages Mash.new
  end
end
```

©2017 Chef Software Inc.

CHEF™

Viewing the Languages Plugin

~/ohai/lib/ohai/plugins/languages.rb

```
Ohai.plugin(:Languages) do
  provides "languages"
  collect_data do
    languages Mash.new
  end
end
```

Plugin Name
- Ruby Symbol
- First Letter Capitalized

Node attributes provided by the plugin

Code executed on all platforms and stored in the provided attribute(s).

©2017 Chef Software Inc.

Viewing the Language Plugin

~/ohai/lib/ohai/plugins/languages.rb

```
Ohai.plugin(:Languages) do
  provides "languages"
  collect_data do
    languages Mash.new
  end
end
```

The Languages plugin provides the node attribute 'languages' which is populated, on all platforms, with a Mash.

©2017 Chef Software Inc.

CHEF

CONCEPT

Hash

Using a String as a key

```
[1] pry(main)> content = Hash.new
=> {}
[2] pry(main)> content['name'] =
'Chef'
=> "Chef"
[3] pry(main)> content['name']
=> "Chef"
[4] pry(main)> content[:name]
=> nil
```

Using a Symbol as a key

```
[1] pry(main)> content = {}
=> {}
[2] pry(main)> content[:name] =
'Chef'
=> "Chef"
[3] pry(main)> content[:name]
=> "Chef"
[4] pry(main)> content['name']
=> nil
```

©2017 Chef Software Inc.

CHEF

CONCEPT

Mash

Using a String as a key

```
[1] pry(main)> require 'chef'
=> true
[2] pry(main)> content = Mash.new
=> {}
[3] pry(main)> content['name'] = 'Chef'
=> "Chef"
[4] pry(main)> content['name']
=> "Chef"
[5] pry(main)> content[:name]
=> "Chef"
```

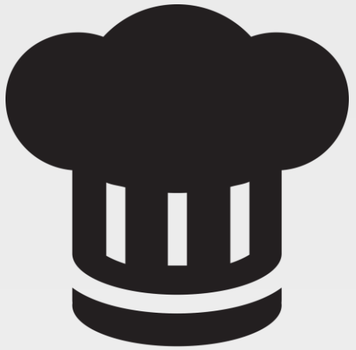
Using a Symbol as a key

```
[1] pry(main)> require 'chef'
=> true
[2] pry(main)> content = Mash.new
=> {}
[3] pry(main)> content[:name] = 'Chef'
=> "Chef"
[4] pry(main)> content[:name]
=> "Chef"
[5] pry(main)> content['name']
=> "Chef"
```

©2017 Chef Software Inc.

CHEF

EXERCISE



Reviewing the Ohai Gem

Now it is time to look at a more complex plugin.

Objective:

- ✓ Review the basic structure of the Ohai gem
- ✓ Review the 'language' plugin
- ❑ Review the 'python' plugin

©2017 Chef Software Inc.

CHEF

Viewing the Python plugin

~/ohai/lib/ohai/plugins/python.rb

```
Ohai.plugin(:Python) do
  provides "languages/python"

  depends "languages"

  collect_data do
    begin
      so = shell_out("python -c \"import sys; print (sys.version)\"")
      # Sample output:
      # 2.7.11 (default, Dec 26 2015, 17:47:53)
      # [GCC 4.2.1 Compatible Apple LLVM 7.0.2 (clang-700.1.81)]
      if so.exitstatus == 0
        python = Mash.new
        output = so.stdout.split
        python[:version] = output[0]
        if output.length >= 6
```

Node attributes provided by the plugin

This plugin depends on the attributes in the Languages to be defined

©2017 Chef Software Inc.

CHEF

Viewing the Python plugin

~/ohai/lib/ohai/plugins/python.rb

```
Ohai.plugin(:Python) do
  provides "languages/python"

  depends "languages"

  collect_data do
    begin
      so = shell_out("python -c \"import sys; print (sys.version)\"")
      # Sample output:
      # 2.7.11 (default, Dec 26 2015, 17:47:53)
      # [GCC 4.2.1 Compatible Apple LLVM 7.0.2 (clang-700.1.81)]
      if so.exitstatus == 0
        python = Mash.new
        output = so.stdout.split
        python[:version] = output[0]
        if output.length >= 6
```

©2017 Chef Software Inc.

CHEF

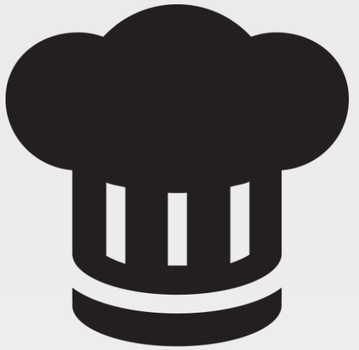
CONCEPT

collect_data for a specific Platform

Plugins can collect data in different ways across different platforms. When defining a collect_data block if you do not provide any arguments it is assumed the default and all platforms unless you define a collect_data block specific for a platform.

©2017 Chef Software Inc.

EXERCISE



Reviewing the Ohai Gem

We know where the plugins are located and what they look like. Now it's time to make one.

Objective:

- ✓ Review the basic structure of the Ohai gem
- ✓ Review the 'language' plugin
- ✓ Review the 'python' plugin

©2017 Chef Software Inc.

CHEF

Discussion

How are the structures of a Rubygem and a Cookbook similar to each other?

What are the requirements when specifying the name of a Ohai plugin?

What is the difference between a Ruby Hash and a Mash?

©2017 Chef Software Inc.

DISCUSSION

Q&A

What questions can we answer for you?

©2017 Chef Software Inc.

CHEF

CHEF™