CHEF: Ohai

Table of Contents

Introduction

Ohai est un outil utilisé pour collecter les données de configuration du système, qui sont fournies au chef client pour être utilisées dans les livres de recettes. **Ohai** est géré par le chef-client au début de chaque exécution pour déterminer l'état du système. **Ohai** comprend de nombreux plug-ins intégrés pour détecter les détails de configuration courants, ainsi qu'un modèle de plug-in pour écrire des plug-ins personnalisés.

Les types d'attributs collectés par **Ohai** incluent, mais ne sont pas limités à:

- Système opérateur
- Réseau
- Mémoire
- Disque
- CPU
- Noyau
- Noms d'hôtes
- Noms de domaine complets
- La virtualisation
- Métadonnées de fournisseur de cloud

Attributs

Les attributs collectés par Ohai sont des attributs de niveau automatiques, en ce sens qu'ils sont utilisés par le chef-client pour garantir que ces attributs restent inchangés après la configuration du noeud par le chef-client.

Présentation

Un attribut automatique est un détail spécifique concernant un nœud, tel qu'une adresse IP, un nom d'hôte, une liste de modules du noyau chargés, etc. Les attributs automatiques sont détectés par Ohai et sont ensuite utilisés par le chef-client pour s'assurer qu'ils sont gérés correctement lors de chaque exécution du chef-client.

Les Attributs automatiques les plus couramment utilisés :

Attribut	**Description**
node['platform']	La plate-forme sur laquelle un nœud est en cours d'exécution. Cet attribut aide à déterminer quels fournisseurs seront utilisés.
node['platform_version']	La version de la plateforme. Cet attribut aide à déterminer quels fournisseurs seront utilisés.
node['ipaddress']	L'adresse IP d'un noeud. Si le noeud a une route par défaut, il s'agit de l'adresse IPV4 de l'interface. Si le nœud n'a pas d'itinéraire par défaut, la valeur de cet attribut doit être nil . L'adresse IP de la route par défaut est la valeur par défaut recommandée.
node['macaddress']	L'adresse MAC d'un nœud, déterminée par la même interface que celle qui détecte le node['ipaddress'] .
node['fqdn']	Le nom de domaine complet pour un nœud. Ceci est utilisé comme nom d'un nœud, sauf indication contraire.
node['hostname']	Le nom d'hôte du noeud.
node['domain']	Le domaine pour le noeud.
node['recipes']	Une liste de recettes associées à un nœud (et une partie de la liste d'exécution de ce nœud).
node['roles']	
node['ohai_time']	L'heure à laquelle Ohai a été exécuté pour la dernière fois. Cet attribut n'est pas couramment utilisé dans les recettes, mais il est enregistré sur le serveur Chef et est accessible à l'aide de la sous-commande knife status .

Utilisation

Ouvrir chef-shell pour visualiser les attributs:

```
$ chef-shell -z

loading configuration: /etc/chef/client.rb
Session type: client
Loading................resolving cookbooks for run list: []
```

```
Synchronizing Cookbooks:
done.

This is the chef-shell.
 Chef Version: 14.7.17
 https://www.chef.io/
 https://docs.chef.io/

run `help' for help, `exit' or ^D to quit.

Ohai2u root@master2016!
chef (14.7.17)>
```

Passer en mode attributes_mode...

```
chef (14.7.17)>attributes_mode
```

...pour interroger l'attribut souhaité

```
 chef:attributes (14.7.17)> attributes[:cpu]
  => {"0"=>{"vendor_id"=>"GenuineIntel", "family"=>"6", "model"=>"13",
"model_name"=>"QEMU Virtual CPU
  version (cpu64-rhel6)", "stepping"=>"3", "mhz"=>"1861.914",
"cache_size"=>"4096 KB", "physical_id"=>"0",
  "core_id"=>"0", "cores"=>"1", "flags"=>["fpu", "de", "pse", "tsc", "msr",
"pae", "mce", "cx8", "apic",
  "mtrr", "pge", "mca", "cmov", "pse36", "clflush", "mmx", "fxsr", "sse",
"sse2", "syscall", "lm", "nopl",
  "pni", "cx16", "hypervisor", "lahf_lm"]},
"1"=>{"vendor_id"=>"GenuineIntel", "family"=>"6", "model"=>"13",
  "model_name"=>"QEMU Virtual CPU version (cpu64-rhel6)", "stepping"=>"3",
"mhz"=>"1861.914",
  "cache_size"=>"4096 KB", "physical_id"=>"1", "core_id"=>"0", "cores"=>"1",
"flags"=>["fpu", "de", "pse",
  "tsc", "msr", "pae", "mce", "cx8", "apic", "mtrr", "pge", "mca", "cmov",
"pse36", "clflush", "mmx", "fxsr",
  "sse", "sse2", "syscall", "lm", "nopl", "pni", "cx16", "hypervisor",
"lahf_lm"]}, "total"=>2, "real"=>2,
  "cores"=>2}
 chef:attributes (14.7.17)>
```

Il n'est pas nécessaire d'utiliser le nom complet du module car il existe une méthode fournie qui stocke le contenu sous forme de hachage sous la clé **xmen** .

Hints

Les hints Ohai sont utilisés pour indiquer à Ohai quelque chose sur le système sur lequel il tourne et/ou qu'il ne peut pas découvrir.

Présentation

Ohai est un excellent moyen d'accéder aux informations appartenant au nœud lui-même par le biais de diverses commandes ou d'afficher ce qui se trouve sur le système de fichiers, mais il doit parfois être influencé par une source externe ou doit passer un appel réseau pour obtenir les informations (par exemple, Métadonnées EC2).

Les hints sont des fichiers JSON qui résident sur un nœud et fournissent des informations facultatives qu'Ohai ne peut pas collecter facilement .

Utilisation

Création des Hints dans le client

Les fichiers **hints** trouvent par défaut dans le répertoire `/etc/chef/ohai/hints/` . Utiliser l'option Ohai.config[:hints_path] dans le fichier client.rb pour personnaliser cet emplacement.

Les **hints** sont créés en plaçant des fichiers JSON dans le répertoire des hints (défaut est /etc/chef/ohai/hints/).

```
$ sudo mkdir -p /etc/chef/ohai/hints
$ echo '{"members": ["wolverine", "cyclops", "beast", "storm"]}' | sudo tee
/etc/chef/ohai/hints/xmen.json

{"members": ["wolverine", "cyclops", "beast", "storm"]}
```

Charger des hints ohai dans la commande knife bootstrap

Pour charger des hints dans le client lors du processus de bootstraping la syntaxe est la suivante :

```
--hint HINT_NAME[=HINT_FILE]
```

- **HINT_NAME** est le nom du conseil
- **HINT_FILE** est le nom du fichier de hint situé dans /path/to/HINT_FILE

Le fichier indiqué sera chargé dans le répertoire `/etc/chef/ohai/hints` du client

On peut spécifier plusieurs fichier en spécifiant plusieurs options **–hint-** dans la commande de bootstrap.

Visualiser le contenu

Ouvrir chef-shell pour visualiser les hints :

```
$ chef-shell -z

loading configuration: /etc/chef/client.rb
Session type: client
Loading................resolving cookbooks for run list: []
Synchronizing Cookbooks:
done.

This is the chef-shell.
 Chef Version: 14.7.17
 https://www.chef.io/
 https://docs.chef.io/

run `help' for help, `exit' or ^D to quit.

Ohai2u root@master2016!
chef (14.7.17)>
```

Visualiser le contenu du hint :

```
chef (14.7.17)> Ohai::Hints.hint?('xmen')
  => {"members"=>["wolverine", "cyclops", "beast", "storm"]}
chef (14.7.17)>
```

Il n'est pas nécessaire d'utiliser le nom complet du module car il existe une méthode fournie qui stocke le contenu sous forme de hachage sous la clé **xmen** .

Paramètres

Les paramètres de configuration Ohai peuvent être ajoutés au fichier client.rb.

Présentation

Les Paramètres dans client.rb sont:

Paramètre	Description
ohai.directory	Le répertoire dans lequel se trouvent les plugins Ohai.
ohai.disabled_plugins	Un tableau de plug-ins Ohai à désactiver sur un nœud. La liste des plugins inclus dans Ohai se trouve dans le ohai/lib/ohai/plugins.
ohai.hints_path	Le chemin d'accès au fichier contenant des astuces pour Ohai.
ohai.log_level	Le niveau de journalisation à stocker dans un fichier journal.

Paramètre	Description
ohai.log_location	L'emplacement du fichier journal.
ohai.plugin_path	Un tableau de chemins sur lesquels les plugins Ohai sont situés. Valeur par défaut: [<CHEF_GEM_PATH>/ohai-9.9.9/lib/ohai/plugins] . Lorsque des plug-ins Ohai personnalisés sont ajoutés, les chemins doivent être ajoutés au tableau.
ohai.version	La version d'Ohai.

Utilisation

Désactivation des plugins

Dans le client local, ajouter les lignes suivantes dans le fichier client.rb:

- pour désactiver un seul plugin:

  ohai.disabled_plugins=[:MyPlugin]

- pour désactiver plusieurs plugins:

  ohai . disabled_plugins = [: MyPlugin , : MyPlugin , : MyPlugin]

Lorsqu'un plugin est désactivé, le fichier journal chef-client contient des entrées similaires à:

```
  [ 2014 - 06 - 13 T23 : 49 : 12 + 00 : 00 ] DEBUG : plug-in désactivé
MyPlugin
```

Ajouter des plug-ins personnalisés

Lorsque des plug-ins Ohai personnalisés sont ajoutés, les chemins doivent être ajoutés au tableau.

- ajouter un seul plugin:

```
  ohai. plugin_path << '/etc/chef/ohai_plugins'
```

- ajouter plusieurs plugins:

```
  ohai. plugin_path += [
    '/etc/chef/ohai_plugins' ,
    '/chemin/vers/autre/plugins'
  ]
```

Automatisation du paramétrage

Pour automatiser la configuration dans les templates personnalisés:

- Localiser la ligne suivante au bas du fichier template.erb:

```
cat > /etc/chef/client.rb <<'EOP'
<%= config\_content %>
EOP
```

- En dessous, ajouter ce qui suit:

```
cat >> /etc/chef/client.rb <<'EOP'
Ohai::Config[:disabled\_plugins]=[:Passwd]
EOP
```