

CHEF: Chef Shell

Table of Contents

- [CHEF: Chef Shell](#)
- [Description](#)
 - [Modes](#)
 - [Usage](#)
 - [Options](#)
 - [Configuration](#)
- [Utilisation](#)
 - [En mode autonome](#)
 - [En tant que chef client](#)
- [Utilisation](#)
 - [Pour afficher les données de base](#)
 - [Pour le débogage](#)
 - [Débogage de recette](#)
 - [Débogage d'attributs](#)

chef-shell est un outil de débogage de recettes qui permet d'utiliser des points d'arrêt dans les recettes. chef-shell s'exécute en tant que session interactive Ruby (IRb). chef-shell prend en charge la syntaxe des fichiers de recettes et d'attributs, ainsi que les fonctionnalités de débogage interactif.

L'exécutable chef-shell est exécuté en tant qu'outil de ligne de commande

Description

Modes

chef-shell est un outil exécuté à l'aide d'une session Interactive Ruby (IRb). chef-shell prend actuellement en charge la syntaxe des fichiers de recettes et d'attributs, ainsi que les fonctionnalités de débogage interactif. chef-shell a trois modes d'exécution:

Mode	Description
Autonome	Mode par défaut. Aucun cookbokk n'est chargé et la liste de lecture est vide.
Solo	chef-shell agit en tant que client chef-solo. Il tente de charger le fichier de configuration chef-solo et les attributs JSON. Si les attributs JSON définissent une liste d'exécution, celle-ci sera honorée. Les cookbooks seront chargés de la même manière que le chef solo les charge. Le mode chef-solo est activé avec l'option de ligne de commande -s ou -solo et les attributs JSON sont spécifiés de la même manière que pour chef-solo, avec -j /path/to/chef-solo.json.

Mode	Description
Client	chef-shell agit en tant que chef-client. Lors du démarrage, il lit le fichier de configuration chef-client et contacte le serveur Chef pour obtenir des attributs et des cookbooks. La run list sera définie de la même manière que les listes normales de chef-client. Le mode chef-client est activé avec les options -z ou -client . On peut également spécifier le fichier de configuration avec -c CONFIG et l'URL du serveur avec -S SERVER_URL.

Usage

Cette commande a la syntaxe suivante:

```
chef-shell OPTION VALEUR OPTION VALEUR ...
```

Options

Cette commande a les options suivantes:

court	long	description
-a	-standalone	Exécuter chef-shell en mode autonome.
-c CONFIG	-config CONFIG	Fichier de configuration à utiliser.
-h	-help	Afficher l'aide pour la commande.
-j PATH	-json-attributes PATH	Chemin d'accès à un fichier contenant des données JSON. Utilisez cette option pour définir un objet run_list. Peut être utilisé en exécutant chef-client -j path/to/file.json. Dans certaines situations, cette option peut être utilisée pour mettre à jour normal attributs normal . Tout autre type d'attribut contenu dans ce fichier JSON sera traité comme un attribut normal . La définition d'attributs à d'autres niveaux de priorité n'est pas possible.
-l LEVEL	-log-level LEVEL	Le niveau de journalisation à stocker dans un fichier journal.
-s	-solo	Exécuter chef-shell en mode chef-solo.
-S CHEF_SERVER_URL	-server CHEF_SERVER_URL	URL du serveur Chef.
-v	-version	Version du chef-client.
-z	-client	Exécuter chef-shell en mode chef-client.

Configuration

chef-shell détermine le fichier de configuration à charger en fonction des éléments suivants:

- Si un fichier de configuration est spécifié à l'aide de l'option -c , chef-shell utilisera le fichier de configuration spécifié.
- Lorsque chef-shell est démarré avec une configuration nommée en tant qu'argument, chef-shell recherchera un fichier chef-shell.rb dans ce répertoire sous ~/.chef . Par exemple, si chef-shell est démarré en utilisant production comme configuration nommée, il chargera un fichier de configuration à partir de ~/.chef/production/chef_shell.rb

- Si une configuration nommée n'est pas fournie, chef-shell tentera de charger le fichier chef-shell.rb à partir du répertoire .chef . Par exemple: ~/.chef/chef_shell.rb
- Si aucun fichier chef-shell.rb n'est trouvé, chef-shell tentera de charger le fichier client.rb.
- Si aucun fichier chef-shell.rb n'est trouvé, chef-shell tentera de charger le fichier solo.rb.

Le fichier chef-shell.rb peut être utilisé pour configurer chef-shell de la même manière que le fichier client.rb est utilisé pour configurer le chef-client. Par exemple, pour configurer chef-shell afin qu'il s'authentifie auprès du serveur Chef, copier les paramètres **node_name** , **client_key** et **chef_server_url** du fichier config.rb:

```
node_name          'your-knife-clientname'
client_key          File.expand_path('~/.chef/my-client.pem')
chef_server_url     'https://api.opscode.com/organizations/myorg'
```

puis les ajouter au fichier chef-shell.rb. D'autres possibilités de configuration incluent la désactivation des plug-ins Ohai (ce qui accélérera le processus de démarrage chef-shell) ou l'inclusion de code Ruby arbitraire dans le fichier chef-shell.rb.

Utilisation

En mode autonome

Par défaut, chef-shell se charge en mode autonome et ne se connecte pas au serveur. On peut écrire un chef_shell.rb , utiliser un fichier knife.rb ou, sur un nœud réel, utiliser client.rb. L'utilisation de client.rb sur un nœud réel ne charge pas tout et ne se connecte pas au serveur Chef. Pour ce faire, on doit exécuter Chef Shell en tant que Chef Client (voir ci-dessous).

```
/opt/chef/bin/chef-shell

/opt/chef/bin/chef-shell -c /etc/chef/client.rb

/opt/chef/bin/chef-shell -c ~/.chef/knife.rb

/opt/chef/bin/chef-shell -c ~/.chef/chef_shell.rb
```

En tant que chef client

Chef-shell peut être exécuté en tant que chef-client pour vérifier une fonctionnalité disponible uniquement lorsque le chef-client se connecte au serveur, telle qu'une fonctionnalité de recherche ou l'accès à des données stockées dans des sacs de données.

```
/opt/chef/bin/chef-shell -z
```



Les autorisations pour accéder aux fichiers dans `/etc/chef/*` sont nécessaires.

Utilisation

Pour afficher les données de base

Données	Commandes
Nom du noeud	node.name
Environnement actuellement chargé	node.chef_environnement node.environnement
Run list	node.run_list
Attributs par défaut	node.default
Attributs normaux	node.normal
Attributs remplacé	node.override
Attributs automatiques	node.automatic
FQDN	node.fqdn node.automatic.fqdn
CPU	node.CPU node.automatic.CPU
Mémoire	node.memory node.automatic.memory

Pour le débogage

Débogage de recette

Pour déboguer une recette, passer d'abord en mode `recipe_mode` .

```
mode recipe_mode
```

Ensuite, on peut appeler une recette

```
include_recipe "mise en route"
```

Débogage d'attributs

Pour déboguer des attributs, passer d'abord en mode `attributs_mode` .

mode_attribut

Ensuite, copier le code d'attributs à tester dans le shell .

```
run_state['detected_environment'] =  
if /dev|qa|stg|prd/ =~ node['chef_environment']  
  /dev|qa|stg|prd/.match(node['chef_environment']).to_s.downcase  
else  
  # Consider _default as 'Dev'  
  'dev'  
end
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=vagrant:chef-shell>

Last update: **2025/02/19 10:59**

