

KNIFE: Sous-commandes de knife

Table of Contents

- [KNIFE: Sous-commandes de knife](#)
 - [Syntaxe de commande de knife](#)
 - [Sous-commandes intégrées](#)
 - [Sous-commandes de verbe](#)

Knife est un outil de ligne de commande qui fournit une interface entre un chef repo local et le serveur Chef.

Knife aide les utilisateurs à gérer:

- Noeuds
- Livres de recettes et recettes
- Rôles, environnements et sacs de données
- Ressources dans divers environnements de cloud
- L'installation du chef-client sur des nœuds
- Recherche de données indexées sur le serveur Chef

Knife comprend une collection de sous-commandes intégrées qui fournissent toutes les fonctionnalités nécessaires pour effectuer des actions spécifiques sur tout objet d'une organisation, notamment les livres de recettes, les nœuds, les rôles, les poches de données, les environnements et les utilisateurs.

Syntaxe de commande de knife

Toutes les sous-commandes de couteau ont la syntaxe suivante:

```
knife sous-commande [ARGUMENT] (options)
```

Chaque sous-commande a son propre jeu d'arguments et d'options.



Tous les exemples de syntaxe de ce document montrent les variables au format ALL_CAPS. Par exemple -u PORT_LIST (où PORT_LIST est une liste de ports UDP publics et locaux séparés par des virgules) ou -F FORMAT (où FORMAT détermine le format de sortie, soit summary , text , json , yaml ou pp). Ces variables nécessitent souvent des valeurs spécifiques propres à chaque organisation.

Sous-commandes intégrées

Sous-commande	Description
knife_bootstrap	Utiliser la sous-commande knife bootstrap pour exécuter une opération d'amorçage qui installe le chef-client sur le système cible. L'opération d'amorçage doit spécifier l'adresse IP ou le nom de domaine complet du système cible.
knife_client	Utiliser la sous-commande knife client pour gérer une liste de clients API et leurs paires de clés publiques RSA associées. Cela permet aux demandes d'authentification d'être adressées au serveur Chef par toute entité utilisant l'API du serveur Chef, telle que le chef-client et knife
knife configure	Utiliser la sous-commande knife configure pour créer les fichiers client.rb et config.rb afin qu'ils puissent être distribués aux postes de travail et aux nœuds.
knife bootstrap	Utiliser la sous-commande knife cookbook pour interagir avec les livres de recettes situés sur le serveur Chef ou le chef repo local.
knife data bag	Utiliser la sous-commande knife data bag pour gérer les magasins arbitraires de données JSON disponibles au niveau mondial.
knife delete	Utiliser la sous-commande knife delete pour supprimer un objet d'un serveur Chef. Cette sous-commande fonctionne comme un ensemble de commandes knife cookbook delete knife data bag delete knife environment delete knife node delete knife role delete et de knife role delete , mais avec un seul verbe (et une seule action).
knife deps	Utiliser la sous-commande knife deps pour identifier les dépendances d'un nœud, d'un rôle ou d'un livre de recettes.
knife diff	Utiliser la sous-commande knife diff pour comparer les différences entre les fichiers et les répertoires sur le serveur Chef et dans le chef repo. Par exemple, pour comparer des fichiers sur le serveur Chef avant de télécharger ou de télécharger des fichiers à l'aide des sous-commandes de knife upload et de knife download knife upload , ou pour vous assurer que certains fichiers de plusieurs environnements de production sont identiques. Cette sous-commande est similaire à la commande git diff qui peut être utilisée pour comparer ce qui se trouve dans chef-repo avec ce qui est synchronisé dans un référentiel git.
knife download	Utiliser la sous commande knife download pour télécharger les rôles, les livres de recettes, les environnements, les nœuds et les sacs de données du serveur Chef vers le répertoire de travail en cours. Il peut être utilisé pour sauvegarder des données sur le serveur Chef, inspecter l'état d'un ou de plusieurs fichiers ou pour extraire les modifications apportées en dehors du processus que les utilisateurs peuvent avoir apportées aux fichiers sur le serveur Chef, par exemple si un utilisateur effectuait une modification. qui a contourné le contrôle de code de version. Cette sous-commande est souvent utilisée avec knife diff , qui permet de voir exactement quels changements seront téléchargés, puis knife upload , qui fait le contraire du knife download de knife download .
knife edit	Utiliser la sous-commande knife edit pour modifier des objets sur le serveur Chef. Cette sous-commande fonctionne de la même manière qu'un ensemble de commandes knife cookbook edit knife data bag edit , knife environment edit knife node edit knife role edit et knife role edit , mais avec un seul verbe (et une seule action).
knife environnement	Utiliser la sous-commande knife environment pour gérer les environnements au sein d'une organisation unique sur le serveur Chef.

Sous-commande	Description
knife exec	Utiliser la sous-commande knife exec pour exécuter les scripts Ruby dans le contexte d'un chef-client entièrement configuré. Utilisez cette sous-commande pour exécuter des scripts qui n'accèdent qu'une seule fois au serveur Chef (ou très rarement) ou à tout moment où une opération ne garantit pas l'utilisation complète de la bibliothèque de sous-commandes knife.
knife list	Utiliser la sous-commande knife list pour afficher une liste d'objets sur le serveur Chef. Cette sous-commande fonctionne comme un ensemble de commandes knife cookbook list knife data bag list knife environment list knife node list knife role list et une knife role list , mais avec un seul verbe (et une seule action).
knife node	Utiliser la sous-commande knife node pour gérer les nœuds existants sur un serveur Chef.
knife raw	Utiliser la sous-commande knife raw pour envoyer une demande REST à un nœud final dans l'API du serveur Chef.
knife recipe list	Utiliser la sous-commande de knife recipe list pour afficher toutes les recettes figurant sur un serveur Chef. Une expression régulière peut être utilisée pour limiter les résultats aux recettes qui correspondent à un modèle spécifique. L'expression régulière doit être entre guillemets et ne pas être entourée de barres obliques (/).
knife role	Utiliser la sous-commande knife role pour gérer les rôles associés à un ou plusieurs nœuds sur un serveur Chef.
knife search	Utiliser la sous-commande knife search pour exécuter une requête de recherche pour les informations indexées sur un serveur Chef.
knife serve	Utiliser la sous-commande knife serve pour exécuter un chef-zéro persistant contre le chef-repo local. (chef-zero est un serveur Chef léger qui s'exécute en mémoire sur la machine locale.) Cela revient à exécuter l'exécutable chef-client avec l'option -local-mode . Le chef_repo_path est situé automatiquement et le serveur Chef se 8889 au premier port disponible entre 8889 et 9999 . knife serve récupère l'URL du serveur Chef local, afin de l'ajouter au fichier config.rb.
knife show	Utiliser la sous commande knife show pour afficher les détails d'un (ou plusieurs) objet sur le serveur Chef. Cette sous-commande fonctionne de la même manière que l' knife cookbook show au knife cookbook show , knife data bag show knife environment show knife node show knife role show et la knife role show , mais avec un seul verbe (et une seule action).
knife ssh	Utiliser la sous-commande knife ssh pour appeler des commandes SSH (en parallèle) sur un sous-ensemble de nœuds au sein d'une organisation, en fonction des résultats d'une requête de recherche effectuée sur le serveur Chef.
knife ssl check	Utiliser la sous-commande knife ssl check pour vérifier la configuration SSL du serveur Chef ou un emplacement spécifié par une URL ou un URI. Les certificats non valides ne seront pas utilisés par OpenSSL. Lorsque cette commande est exécutée, les fichiers de certificat (*.crt et / ou *.pem) situés dans le répertoire /.chef/trusted_certs sont vérifiés pour voir s'ils possèdent des propriétés de certificat X.509 valides. Un avertissement est renvoyé lorsque les certificats ne possèdent pas de propriétés de certificat X.509 valides ou si le répertoire /.chef/trusted_certs ne contient aucun certificat. Lorsque la vérification du certificat SSL d'un serveur distant est désactivée, le chef-client émet un avertissement semblable à «La validation SSL des demandes HTTPS est désactivée. Les connexions HTTPS sont toujours chiffrées, mais le chef-client n'est pas en mesure de détecter les réponses falsifiées ni les attaques de type «man-in-the-middle». »Pour configurer SSL pour le chef-client, définir ssl_verify_mode sur :verify_peer (recommandé) ou verify_api_cert sur true dans le fichier client.rb.

Sous-commande	Description
knife ssl fetch	Utiliser la sous-commande knife ssl fetch pour copier les certificats SSL d'un serveur HTTPS dans le répertoire trusted_certs_dir utilisé par couteau et le chef-client pour stocker les certificats SSL de confiance. Lorsque ces certificats correspondent au nom d'hôte du serveur distant, l'exécution de la commande knife ssl fetch est la seule étape requise pour vérifier un serveur distant auquel accéder via couteau ou le chef-client. Il incombe à l'utilisateur de vérifier l'authenticité de chaque certificat SSL avant de le télécharger dans le répertoire /.chef/trusted_certs . couteau utilisera n'importe quel certificat de ce répertoire comme s'il s'agissait d'un certificat SSL fiable et authentique à 100%. couteau ne pourra pas déterminer si un certificat de ce répertoire a été falsifié, falsifié, malveillant ou nuisible. Par conséquent, il est essentiel que les utilisateurs prennent les mesures appropriées avant de télécharger des certificats dans ce répertoire.
knife status	Utiliser la sous-commande knife status pour afficher un bref résumé des noeuds sur un serveur Chef, y compris l'heure de la dernière exécution réussie de chef-client.
knife tag	Utiliser la sous-commande knife tag pour appliquer des balises aux nœuds d'un serveur Chef.
knife supermarket	Utiliser la sous-commande knife supermarket pour interagir avec les livres de cuisine disponibles dans le supermarché Chef . Un compte d'utilisateur est requis pour toute action de communauté qui écrit des données sur ce site. Les arguments suivants ne nécessitent pas de compte d'utilisateur: download , search , install et list .
knife upload	Utiliser la sous-commande knife upload pour télécharger des données sur le serveur Chef à partir du répertoire de travail actuel du chef-repo. Les types de données suivants peuvent être téléchargés avec cette sous-commande: Livres de cuisine, Sacs de données, Rôles stockés sous forme de données JSON, Environnements stockés sous forme de données JSON (Les rôles et les environnements stockés sous forme de données Ruby ne seront pas téléchargés.) Cette sous-commande est souvent utilisée en conjonction avec knife diff , qui peut être utilisée pour voir exactement quels changements seront téléchargés, puis knife download , qui fait l'inverse du knife upload de knife upload.
knife user	Utiliser la sous-commande knife user pour gérer la liste des utilisateurs et leurs paires de clés publiques RSA associées.
knife xargs	Utiliser la sous-commande knife xargs pour extraire des modèles de l'entrée standard, télécharger en tant que JSON, exécuter une commande sur le JSON téléchargé, puis télécharger les modifications éventuelles.

Sous-commandes de verbe

knife inclut un ensemble de sous-commandes construites autour de verbes courants: **delete** , **deps** , **diff** , **download** , **edit** , **list** , **show** , **upload** , **xargs** . Ces sous-commandes permettent à knife d'émettre des commandes qui interagissent avec n'importe quel objet stocké dans le chef-repo ou stocké sur le serveur Chef. Certains principes importants derrière ce groupe de sous-commandes incluent:

- Une commande qui fonctionne avec chaque objet du chef-repo. Les sous-commandes spécifient l'action souhaitée (le «verbe»), puis le répertoire dans lequel réside cet objet (clients , cookbooks/ , data_bags/ , environments/ , nodes , roles/ et users). Par exemple: download cookbooks/

- Une commande qui fonctionne avec certains objets du serveur Chef, y compris acls , groups et containers
- Utilise le serveur Chef comme s'il s'agissait d'un système de fichiers, permettant ainsi au chef-repo du serveur Chef de se comporter comme un miroir du chef-repo du poste de travail. Le serveur Chef aura les mêmes objets que le chef repo local. Pour modifier les fichiers sur le serveur Chef, il suffit de télécharger des fichiers à partir du serveur Chef ou de télécharger des fichiers à partir du chef
- Le contexte à partir duquel une commande est exécutée est important. Par exemple, lorsque on travaille dans le répertoire roles/, knife show base.json knife renverra le rôle de base à partir du serveur Chef. A partir de la racine chef-repo, entrer knife show roles/base.json pour obtenir le même résultat.
- Les requêtes parallèles peuvent être adressées au serveur Chef et sont configurables commande par commande

From:

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

Permanent link:

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

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

