**About Knife**

knife is a command-line tool that provides an interface between a local chef-repo and the Chef server. knife helps users to manage:

* Nodes
* Cookbooks and recipes
* Roles
* Stores of JSON data (data bags), including encrypted data
* Environments
* Cloud resources, including provisioning
* The installation of the chef-client on management workstations
* Searching of indexed data on the Chef server
* knife includes the following subcommands:

| **Subcommand** | **Description** |
| --- | --- |
| [knife bootstrap](https://docs.chef.io/knife_bootstrap.html) | Use the knife bootstrap subcommand to run a bootstrap operation that installs the chef-client on the target system. The bootstrap operation must specify the IP address or FQDN of the target system. |
| [knife client](https://docs.chef.io/knife_client.html) | The knife client subcommand is used to manage an API client list and their associated RSA public key-pairs. This allows authentication requests to be made to the Chef server by any entity that uses the Chef server API, such as the chef-client and knife. |
| [knife configure](https://docs.chef.io/knife_configure.html) | Use the knife configure subcommand to create the knife.rb and client.rb files so that they can be distributed to workstations and nodes. |
| [knife cookbook](https://docs.chef.io/knife_cookbook.html) | The knife cookbook subcommand is used to interact with cookbooks that are located on the Chef server or the local chef-repo. |
| [knife cookbook site](https://docs.chef.io/knife_cookbook_site.html) | The knife cookbook site subcommand is used to interact with cookbooks that are located at <https://supermarket.chef.io>. A user account is required for any community actions that write data to this site. The following arguments do not require a user account: download, search, install, and list. |
| [knife data bag](https://docs.chef.io/knife_data_bag.html) | The knife data bag subcommand is used to manage arbitrary stores of globally available JSON data. |
| [knife delete](https://docs.chef.io/knife_delete.html) | Use the knife delete subcommand to delete an object from a Chef server. This subcommand works similar to knife cookbook delete, knife data bag delete, knife environment delete, knife node delete, and knife role delete, but with a single verb (and a single action). |
| [knife deps](https://docs.chef.io/knife_deps.html) | Use the knife deps subcommand to identify dependencies for a node, role, or cookbook. |
| [knife diff](https://docs.chef.io/knife_diff.html) | Use the knife diff subcommand to compare the differences between files and directories on the Chef server and in the chef-repo. For example, to compare files on the Chef server prior to an uploading or downloading files using the knife download and knife upload subcommands, or to ensure that certain files in multiple production environments are the same. This subcommand is similar to the git diff command that can be used to diff what is in the chef-repo with what is synced to a git repository. |
| [knife download](https://docs.chef.io/knife_download.html) | Use the knife download subcommand to download roles, cookbooks, environments, nodes, and data bags from the Chef server to the current working directory. It can be used to back up data on the Chef server, inspect the state of one or more files, or to extract out-of-process changes users may have made to files on the Chef server, such as if a user made a change that bypassed version source control. This subcommand is often used in conjunction with knife diff, which can be used to see exactly what changes will be downloaded, and then knife upload, which does the opposite of knife download. |
| [knife edit](https://docs.chef.io/knife_edit.html) | Use the knife edit subcommand to edit objects on the Chef server. This subcommand works similar to knife cookbook edit, knife data bag edit, knife environment edit, knife node edit, and knife role edit, but with a single verb (and a single action). |
| [knife environment](https://docs.chef.io/knife_environment.html) | The knife environment subcommand is used to manage environments within a single organization on the Chef server. |
| [knife exec](https://docs.chef.io/knife_exec.html) | The knife exec subcommand uses the knife configuration file to execute Ruby scripts in the context of a fully configured chef-client. Use this subcommand to run scripts that will only access Chef server one time (or otherwise very infrequently) or any time that an operation does not warrant full usage of the knife subcommand library. |
| [knife list](https://docs.chef.io/knife_list.html) | Use the knife list subcommand to view a list of objects on the Chef server. This subcommand works similar to knife cookbook list, knife data bag list, knife environment list, knife node list, and knife role list, but with a single verb (and a single action). |
| [knife node](https://docs.chef.io/knife_node.html) | The knife node subcommand is used to manage the nodes that exist on a Chef server. |
| [knife raw](https://docs.chef.io/knife_raw.html) | Use the knife raw subcommand to send a REST request to an endpoint in the Chef server API. |
| [knife recipe list](https://docs.chef.io/knife_recipe_list.html) | Use the knife recipe list subcommand to view all of the recipes that are on a Chef server. A regular expression can be used to limit the results to recipes that match a specific pattern. The regular expression must be within quotes and not be surrounded by forward slashes (/). |
| [knife role](https://docs.chef.io/knife_role.html) | The knife role subcommand is used to manage the roles that are associated with one or more nodes on a Chef server. |
| [knife search](https://docs.chef.io/knife_search.html) | Use the knife search subcommand to run a search query for information that is indexed on a Chef server. |
| [knife serve](https://docs.chef.io/knife_serve.html) | Use the knife serve subcommand to run a persistent chef-zero against the local chef-repo. (chef-zero is a lightweight Chef server that runs in-memory on the local machine.) This is the same as running the chef-client executable with the --local-mode option. The chef\_repo\_path is located automatically and the Chef server will bind to the first available port between 8889 and 9999. knife serve will print the URL for the local Chef server, so that it may be added to the knife.rb file. |
| [knife show](https://docs.chef.io/knife_show.html) | Use the knife show subcommand to view the details of one (or more) objects on the Chef server. This subcommand works similar to knife cookbook show, knife data bag show, knife environment show, knife node show, and knife role show, but with a single verb (and a single action). |
| [knife ssh](https://docs.chef.io/knife_ssh.html) | Use the knife ssh subcommand to invoke SSH commands (in parallel) on a subset of nodes within an organization, based on the results of a [search query](https://docs.chef.io/chef_search.html) made to the Chef server. |
| [knife ssl check](https://docs.chef.io/knife_ssl_check.html) | Use the knife ssl check subcommand to verify the SSL configuration for the Chef server or a location specified by a URL or URI. Invalid certificates will not be used by OpenSSL.  When this command is run, the certificate files (\*.crt and/or \*.pem) that are located in the /.chef/trusted\_certs directory are checked to see if they have valid X.509 certificate properties. A warning is returned when certificates do not have valid X.509 certificate properties or if the /.chef/trusted\_certs directory does not contain any certificates. |
| [knife status](https://docs.chef.io/knife_status.html) | Use the knife status subcommand to display a brief summary of the nodes on a Chef server, including the time of the most recent successful chef-client run. |
| [knife tag](https://docs.chef.io/knife_tag.html) | The knife tag subcommand is used to apply tags to nodes on a Chef server. |
| [knife upload](https://docs.chef.io/knife_upload.html) | Use the knife upload subcommand to upload data to the Chef server from the current working directory in the chef-repo. The following types of data may be uploaded with this subcommand:   * Cookbooks * Data bags * Roles stored as JSON data * Environments stored as JSON data   (Roles and environments stored as Ruby data will not be uploaded.) This subcommand is often used in conjunction with knife diff, which can be used to see exactly what changes will be uploaded, and then knife download, which does the opposite of knife upload. |
| [knife user](https://docs.chef.io/knife_user.html) | The knife user subcommand is used to manage the list of users and their associated RSA public key-pairs. |
| [knife xargs](https://docs.chef.io/knife_xargs.html) | Use the knife xargs subcommand to take patterns from standard input, download as JSON, run a command against the downloaded JSON, and then upload any changes. |