CA Release Automation - Integrations

Chef

Date: 24-Nov-2016

Table of Contents

Chef

Action Pack for Chef 10.4.1406

The Action Pack for Chef contains actions that enable you to create workflow processes that perform operations in Chef.

Actions Implementation

The Chef actions use the Java CLI and are executed from the command line.

Actions Help

For a list of input and output descriptions for deployed actions, refer to:

- Chef Actions (see page 11) in this document
- The embedded actions help that is available in your installation at http://<yourdeployment>/nolio-app/actionshelp.jsp.

Supported Platforms:

- Chef Actions in this pack support Chef 11.8.0
- Chef actions in this pack support CA Release Automation 5.0 and later
- Windows 2008 R2 and Linux Red Hat 6

What's New

The following update was made for 10.4.1406.

- Password is listed in the action log as clear text.

Install the Action Packs

Install and use these actions to execute commands and tasks in Chef.

Follow these steps:

1. Verify the Prerequisites to Run Chef Actions (see page 10)
2. Deploy the action pack for Chef.
   For more information, see Manage Action Packs and Plug-ins (https://docops.ca.com/display/RAINT/Manage+Action+Packs+and+Plug-ins).
3. Use Chef actions in CA Release Automation processes.

Prerequisites to Run Chef Actions

To run Chef actions, verify the following prerequisites:

- Download and import the Action Pack.

Follow these steps

1. Download the action pack from CA Support (http://support.ca.com).
2. Extract the Action Pack zip file to get the action pack .jar and the dependencies .zip files.
3. Extract the dependencies.zip file.
4. Import both the .jar and the contents of the dependencies .zip into CA Release Automation using Actions Management.

- If the action pack for Chef is manually copied and imported:

1. Get the **javax.json-api-1.0.jar** and **javax.json-1.0.jar** files from one of the following locations:
   - CA FTP site
   - Maven Repository website:
     http://mvnrepository.com/artifact/javax.json/javax.json-api/1.0
     http://www.java2s.com/Code/Jar/j/Downloadjavaxjson10jar.htm
2. Add the files to the actionLibs directory under the product installation. The jar files are automatically downloaded and added to the actionLibs directory when the Chef Action pack is deployed using the Action Pack Manager.

Note: To use Chef actions, the agent must be installed on the system where Chef Workstation is installed or configured.

Chef Actions

The Action Pack for Chef provides the following actions:

Chef - Bootstrap Linux/Unix

Bootstraps chef-client on target Linux or Unix server

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Hostname or IP address**	String	Hostname or IP address of the target server
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** SSH Username**	String	SSH username of the target server
SSH Password	Password	SSH password to use when authenticating with the remote host
SSH Port	Integer	SSH port that is used by target
SSH Private Key Path	String	Private key file to authenticate target server in case password is not used.
Template File Path	String	The Path of the custom template on the workstation

Chef - Bootstrap Windows

Bootstraps chef-client on target windows server

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Hostname or IP address**	String	Hostname or IP address of the target server

Name	Type	Description
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Expected Target Connection Type	ConnectionTypeEnum	The type of connection
SSH Password	Password	SSH password to use when authenticating with the remote host
SSH Port	Integer	SSH port that is used by target
SSH Private Key Path	String	Private key file to authenticate target server in case password is not used.
SSH Username	String	SSH username of the target server
Target Password	Password	Password of the target windows server
Target UserName	String	Username of the target windows server
Template File Path	String	The Path of the custom template on the workstation
Winrm Port	Integer	Windows remote management listener port

Chef - Bulk Delete Cookbooks

Deletes cookbook files that match a pattern that is defined by a regular expression

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** REGEX**	String	Regular expression that defines a pattern to match cookbooks that are to be deleted. The regular expression must be within quotes and not surrounded by forward slashes (/).
Purge	Boolean	Select true to remove a cookbook (or cookbook version) entirely from the server. No copy of selected file is stored on the server. This disables other cookbooks that reference one or more files from the cookbooks being purged

Output Values

Name	Type	Description
Deleted cookbooks	String[]	List of deleted cookbooks

Chef - Bulk Delete Nodes

Delete one or more nodes that match a pattern that is defined by a regular expression.

Input Values

This is used to delete one or more nodes that match a pattern that is defined by a regular expression. The regular expression must be within quotes and not surrounded by forward slashes (/).

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Regex**	String	Regular expression that defines a pattern to match cookbooks that are to be deleted. The regular expression must be within quotes and not surrounded by forward slashes (/).

Output Values

Name	Type	Description
Deleted Nodes	String[]	List of nodes that are deleted from Chef Server

Chef - Bulk Delete Roles

Delete one or more roles that match a pattern that is defined by a regular expression.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Regex**	String	This is used to delete one or more roles that match a pattern that is defined by a regular expression. The regular expression must be within quotes and not surrounded by forward slashes (/).

Output Values

Name	Type	Description
Deleted Roles	String[]	List of roles that are deleted from Chef Server

Chef - Configure Run List

Configure Run List for the given node from Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node
*** Run List Action**	RunListEnum	Run List operation type. Add or Remove.
*** Run List Item**	String	The comma-separated list of roles and/or recipes to be added to the Run List. **Examples:** role[ROLE_NAME]; recipe[COOKBOOK::RECIPE_NAME]; recipe[COOKBOOK::RECIPE_NAME],COOKBOOK::RECIPE_NAME,role[ROLE_NAME]
Insert After Run List Item	String	Specify the Run List Item. The run list item that is specified in the text field <B>Run List Item</B> would be inserted after this run list item.

Chef - Create Role

Create role in the Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Role Name**	String	Create role in the Chef Server.

Chef - Create User

Create user with the given user ID

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Password**	Password	The user password. Password must have at least 6 characters.
*** User Name**	String	Enter a valid user name
Is Admin	Boolean	True indicates that a user is created as an administrator.
Private Key File	String	Indicates that the private key is saved to a specified file name. Provide the file name with full path. **Example:** C:\Users\MyName\My_Private_Key.pem or /home/myname /my_private_key.pem

Chef - Delete a vSphere VM

Delete a VM from vSphere.

Input Values

Name	Type	Description
*** Chef Configuration file**	String	When this field is specified, the Server configuration settings (Protocol /Servername/Port/Username/Password) are optional and not used.
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux /Unix
*** Password**	Password	The password of the VMware account
*** Source VM Name.**	String	The name of a virtual server to clone.
*** User Name**	String	The user name of the VMware account

Name	Type	Description
*** VSphere Host or Cluster Name**	String	The name of the Host or Cluster.
*** VSphere Server Name**	String	The Vcenter server name. It can be IP address or name of the Vcenter server where the host is located.
Datacenter Name	String	The name of the datacenter.
Datastore Name	String	The name of the datastore for saving the cloned server.
Port	Integer	Port
Protocol	String	The protocol. Can be either "http" or "https".
Resource Pool	String	The name of the resource pool.

Chef - Delete Client

Delete a client from the Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Client Name**	String	Fully qualified host name of the Client
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix

Chef - Delete Node

Delete a node from the Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo

Name	Type	Description
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Chef - Delete Role

Delete the role from Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Role Name**	String	Delete roles from the chef server.

Chef - Delete User

Delete specified user

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** User Name**	String	Enter a valid user name

Chef - Deploy a vSphere VM Template

Deploy a VM Template from vSphere.

Input Values

Name	Type	Description
*** Chef Configuration file**	String	When this field is specified, the Server configuration settings (Protocol /Servername/Port/Username/Password) are optional and not used.
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** New VM Name.**	String	The name of a new cloned virtual server.
*** Password**	Password	The password of the VMware account
*** Source VM Name.**	String	The name of a virtual server to clone.
*** User Name**	String	The user name of the VMware account
*** VSphere Host or Cluster Name**	String	The name of the Host or Cluster.
*** VSphere Server Name**	String	The Vcenter server name. Can be IP address or name of the Vcenter server where the host is located.
Customizaion specification	String	Customize using specified existing customization specification.
Datacenter Name	String	The name of the datacenter.
Datastore Name	String	The name of the datastore for saving the cloned server.
New VM Path	String	The path of the new cloned VM according to the root folder Should be in a form of: Folder/sub_folder.
Port	Integer	Port
Power On Automatically	Boolean	Define whether to turn on the VM after deployment.
Protocol	String	The protocol. Can be either "http" or "https".
Resource Pool	String	The name of the resource pool.
Run List Item	String	The comma-separated list of roles and/or recipes to be added to the Run List. Examples: role[ROLE_NAME]; recipe[COOKBOOK::RECIPE_NAME]; recipe [COOKBOOK::RECIPE_NAME],COOKBOOK::RECIPE_NAME,role[ROLE_NAME]

Name	Type	Description
SSH Password	Password	SSH password to use when authenticating with the remote host
SSH Username	String	SSH username of the target server
Template File Path	String	The Path of the custom template on the workstation

Chef - Download Cookbook

Downloads a cookbook from the server to the current working directory

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Cookbook Name**	String	Enter name of the cookbook
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Cookbook Version	String	Enter the version of the cookbook
Directory path	String	The directory into which the cookbook is downloaded
Force Cookbook	Boolean	Force option allow to upload the frozen cookbook
Latest Version	Boolean	Indicates that the most recent version of a cookbook is downloaded

Chef - Get Available Network Interfaces for a Node

Get list of network interfaces available on the node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**		Fully qualified host name of the Node

Name	Type	Description
	String	

Output Values

Name	Type	Description
Network Interfaces	String[]	List of network interfaces available on the node

Chef - Get Basic Node Information

Get the basic information about a Node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Chef Environment	String	Chef Environment of the Node
FQDN	String	Fully Qualified Domain Name of the Node
IP	String	IP Address of the Node
Platform	String	Platform of the Node
RECIPES	String	Recipes that are applied on the Node
Roles	String	Roles that are applied on the Node
Run _ List	String	Run List of the Node.
Tags	String	Tags that are applied on the Node

Chef - Get Chef Package Information on the Node

Get Chef package information about the node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Chef Root	String	Chef installed directory on the node
Chef Version	String	Version of the Chef that is installed on the Node
Ohai Root	String	Ohai installed directory on the node
Ohai Version	String	Version of the Ohai that is installed on the node

Chef - Get Cookbook List

To get the list of cookbooks that are currently available on the server

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Get all versions of cookbooks	Boolean	Select true to return the list of all available versions of each cookbook.
Get cookbooks with URI	Boolean	Select true to return the corresponding URIs of the cookbooks.

Output Values

Name	Type	Description
Returned cookbook list	String[]	List of returned cookbooks from the server

Chef - Get CPU Information by Index

Get the CPU information about a Node by CPU Index.

Input Values

Name	Type	Description
*** CPU Index**	Integer	CPU Index of the node. For Example, multi-processor nodes have cpu index as 0, 1, 2 etc. Here the index value can be 0, 1 or 2
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux /Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Cache Size(KB)	String	Denotes the size of the CPU Cache in KB
Cores	String	Number of cores in the CPU
Family	String	CPU Family
Model	String	CPU MODEL
Model Name	String	CPU Model Name
Physical ID	String	CPU Physical ID
Speed (mhz)	String	The CPU clock rate in mhz
Stepping	String	Stepping level of the CPU
Vendor ID	String	CPU Vendor

Chef - Get Environment List

View list of environments

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo

Name	Type	Description
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Environments with uri	Boolean	Select true to get the list of environments with complete uri

Output Values

Name	Type	Description
Environments	String[]	List of environments separated by comma(,) **Example:** [Environment1, Environment2] **Example with uri option:** [Environment1: https://Chef-Server-hostname/environments/Environment1, Environment2: https://Chef-Server-hostname/environments/Environment2]

Chef - Get Filesystem Information on the Node

Get the filesystem information about the node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** File System**	String	File system device name. For example, C:, D:, udev, proc
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Available space(KB)	String	Available space in the file system in KB
File System Type	String	File system type like ntfs, securityfs etc
Mount Point	String	File system-mounted directory
Size (KB)	String	Size of file system in KB
Used Space(KB)	String	Used space in the file system in KB

Chef - Get Filesystems on the Node

Get the list of available file systems on the node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
File System(s)	String[]	File system device name(s)

Chef - Get Kernel Details of a Node

Get OS kernel information for a node

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
OS Architecture	String	OS architecture information of the node
OS Name	String	Name of the OS installed on the node
OS Release	String	Release number of the OS installed on the node.

Name	Type	Description
OS Type	String	Type of the OS installed on the node
OS Version	String	OS platform version

Chef - Get Node Attributes

Get the attributes of the node by name

Input Values

Name	Type	Description
*** Attributes**	String	A specific attribute name to retrieve or list of attributes that are seperated by comma(,)
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Attribute Names	String[]	The names of the attributes of retrieved values
Attribute Values	String[]	Attribute values in the same order as Attribute names output

Chef - Get Node Environment

Get the Chef Environment of the Node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Chef Environment	String	Chef Environment of the Node

Chef - Get Node Memory Information

Get the memory details of the Node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Available memory(KB)	String	Available size in KB
Cached Swap memory(KB)	String	Cached Swap memory size in KB
Cached memory(KB)	String	Cached memory in KB
Free Swap memory(KB)	String	Free Swap memory size in KB
Swap memory(KB)	String	Size of swap memory in KB
Total memory(KB)	String	Total memory size in KB

Chef - Get Node OS Platform Information

This action retrieves the OS details of a node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix

Name	Type	Description
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
OS Family	String	OS Platform family. For example, windows, suse
OS Platform	String	OS platform. For example, windows, linux
OS Uptime	String	OS platform uptime
OS Version	String	OS platform version

Chef - Get Node Run List

Get the Run List of the Node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Run _ List	String	Run List of the Node.

Chef - Get Node Virtualization Information

Get the virtualization details of the Node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo

Name	Type	Description
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Role	String	Role of the node in the virtualization environment
System	String	Virtualization technology provides name of the node. Example:VMWare

Chef - Get Nodes

Get the list of nodes from Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Nodes with uri	Boolean	Select true to get the name of the nodes with complete uri

Output Values

Name	Type	Description
Nodes	String []	Name of the nodes seperated by comma(,) **Example :** [hostname1, hostname2] **Example with uri option:** [hostname1: https://hostname1, hostname2: https://hostname2]

Chef - Get Number of CPU(s) on Node

Get the number of CPU(s) on a Node.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
Number of CPU(s)	Integer	Number of CPU(s) available on Node

Chef - Get Recipe List

View list of recipes

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Regex	String	Regular expression to limit the results to recipes that match a specific pattern Example1 : 'rpm', example2: 'java::'

Output Values

Name	Type	Description
Recipes	String[]	List of recipes

Chef - Get Role List

Get the list of roles from the server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Roles with uri	Boolean	Select true to get the list of roles with complete uri

Output Values

Name	Type	Description
List of roles	String[]	Shows the list of roles in the chef server separated by comma(,) For example, [role1,role2] If uri option is set to true the ouput is in the format: [role1:https://chefserver/roles/role1,role2:https://chefserver/roles/role2]

Chef - Get RPMS Installed on a Node

Get the list of rpms installed on the node with the release and version.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
*** Node Name**	String	Fully qualified host name of the Node

Output Values

Name	Type	Description
RPM Package Name(s)	String[]	Installed RPMs list from the node
RPM Release(s)	String[]	RPM Release(s) in the same order as RPM Package Name(s) output
RPM Version(s)	String[]	RPM Package version(s) in the same order as RPM Package Name(s) output

Chef - List User

View list of registered users

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux/Unix
Users with uri	Boolean	Select true to get the list of users with complete uri

Output Values

Name	Type	Description
Users	String[]	List of Users seperated by comma(,) **Example :** [User1, User2] **Example with uri option :** [User1: https://Chef-Server-hostname/users/User1, User2: https://Chef-Server-hostname/users/User2]

Chef - Upload Cookbook

Uploads the Cookbook from Chef Workstation to Chef Server.

Input Values

Name	Type	Description
*** Chef Repo Directory**	String	chef-repo directory path from where Knife Commands can be executed **Linux/Unix example:** {user.home}/chef-repo/.chef **Windows example:** {user.home}\chef-repo
*** Cookbook - Path**	String	Enter path for the Cookbook. For multiple paths add colon(:) in between paths as a separator
*** Knife Path**	String	The Path of Knife Command including knife.bat for Windows / knife for Linux /Unix
Cookbook Name	String	Enter the cookbook name. To upload multiple cookbook names it is separated by comma(,)
Dependency Cookbook Names	String	If any cookbook has dependencies enter the dependencies for that cookbook. For multiple dependency cookbooks are seperated by comma(,)
Force Cookbook	Boolean	Force option allow to upload the freezed cookbook

Name	Type	Description
Freeze Cookbook	Boolean	With Freeze option cookbook cannot be modified, any changes to the cookbook must be included as a new version. Only Force option overrides this setting.
Upload All Cookbooks	Boolean	To upload all cookbooks select true otherwise false