Using the vRealize Orchestrator Chef Plug-In 1.0

VMware, Inc
3401 Hillview Ave
Palo Alto, CA 94304
www.vmware.com

Contents

Using the vRealize Orchestrator Chef Plug-In 1.0

Using the vRealize Orchestrator Chef Plug-In 1.0 tells you how to install, configure, and use the VMware vRealize® Orchestrator™ plug-in for Chef, a configuration management tool.

Intended Audience

This information is intended for users who want to install and configure the Chef plug-in, use the plug-in API, and the workflow library. The information in *Using the vRealize Orchestrator Chef Plug-In 1.0* assumes each user is experienced with VMware virtual machine (VM) technology, with vRealize Orchestrator workflow development, and with Chef.

For more information about vRealize Orchestrator, see http://www.vmware.com/support/pubs/orchestrator_pubs.html.

VMware Technical Publications Glossary

VMware Technical Publications Glossary provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, go to https://www.vmware.com/pdf/master_glossary.pdf.

Introduction to the vRealize Orchestrator Chef Plug-In 1.0

The vRealize Orchestrator Chef plug-in provides interaction between vRealize Orchestrator and Chef. The Chef plug-in interacts with the Chef server REST API and the node guest OS much like the Knife command line tool, which provides an interface between a local chef-repo and the Chef server. The workflows provided with this plug-in perform the following Chef tasks:

- Register and manage multiple Chef servers in the vRealize Orchestrator inventory.
- Install and configure the chef-client agent on Linux using SSH, on Windows using PowerShell, or on vSphere using in-guest scripting with VMware Tools.
- Create and manage attributes and run-lists on Chef nodes and roles.
- Trigger an on-demand chef-client to run on Linux using SSH and Windows using PowerShell or on vSphere using in-guest scripting with VMware Tools.
- Chef node cleanup using the client's own key.

Role of vRealize Orchestrator with the Chef Plug-In

vRealize Orchestrator is a development and process-automation platform that provides a library of extensible workflows to manage the VMware vCenter infrastructure and other technologies.

vRealize Orchestrator provides integration with third party management and administration solutions through its open plug-in architecture. The Chef plug-in is one example of a configuration management solution that you can integrate with vRealize Orchestrator.

After you install the plug-in using the vRealize Orchestrator Configuration interface, you can use the vRealize Orchestrator client to create and run workflows and access the Chef API.

Chef Plug-In Prerequisites

To install and use the Chef plug-in, your system must meet the following prerequisites.

vRealize Automation Support Matrix

vRealize Automation Version	vRealize Orchestrator Version	vRealize Orchestrator Deployment	Notes
7.0.0	7.0.0	Standalone or Embedded	Event Broker service triggers vRealize Orchestrator workflows. Classic IaaS extensibility workflows are still supported.

6.2.3	6.0.3	Standalone or Embedded	Uses classic IaaS extensibility workflows.
6.2.2	6.0.2	Standalone or Embedded	Uses classic IaaS extensibility workflows.
6.2.1	6.0.1	Standalone or Embedded	Uses classic IaaS extensibility workflows.

Chef Support Matrix

Chef Server Version	On Prem	Hosted	OpenSource	Notes
12.x	Supported	Supported	N/A	Only one type of Chef server.
11.x	Supported	Supported	Not Supported	Chef Organization support required.
Earlier Versions	Not Tested	Not Tested	Not Supported	

Virtual Machine Guest Operating System Support Matrix

Operating System	SSH	Remote PowerShell	VMware Tools	Notes
CentOS 6	Supported		Supported	
CentOS 7	Supported		Supported	
Debian Linux 6	Supported		Supported	
Debian Linux 7	Supported		Supported	
RHEL 6	Supported		Supported	
RHEL 7	Supported		Supported	
Windows Server 2008 R2		Supported	Supported with UAC Disabled	Remote PowerShell workflows are UAC compatible
Windows Server 2012		Supported	Supported with UAC Disabled	Remote PowerShell workflows are UAC compatible
Windows Server 2012 R2		Supported	Supported with UAC Disabled	Remote PowerShell workflows are UAC compatible

vRealize Orchestrator

Verify that you have a running instance of vRealize Orchestrator, version 6.0.2 or later, either in a standalone deployment or embedded with vRealize Automation.

Verify that you can log in to the vRealize Orchestrator configuration interface at https://<*orchestrator_server FQDN or IP address*>:8283.

For information about setting up vRealize Orchestrator, see *Installing and Configuring VMware vRealize Orchestrator* at https://www.vmware.com/support/pubs/orchestrator_pubs.html.

Chef Server

Verify if you need to use a proxy server to connect to the Chef server from your vRealize Orchestrator server.

Verify that you have a .pem key for the user that you want to use to connect to the Chef server, and optionally, that you have the common secret used in your organization for data bag encryption.

Install the Chef Plug-In

You install the Chef plug-in using the vRealize Orchestrator Configuration interface.

Prerequisites

- Log in to the vRealize Orchestrator Configuration interface at https://<*orchestrator_server FQDN or IP address*>:8283.
- Download the `.vmoapp` file for the Chef plug-in from VMware Solution Exchange.

Procedure

1. Click **General** in the left pane, and click **Install Application** in the right pane.
2. Upload the Chef Plug-in `.vmoapp` file.
    a. Click the magnifying glass icon (🔍).
    b. Navigate to the directory containing the `.vmoapp` file, and select the file.
    c. Click **Open**.
    d. Click **Install**.

   A message appears after a successful installation. The Chef plug-in is installed without a tab in the vRealize Orchestrator Configuration interface. The plug-in is configured in the vRealize Orchestrator client.
3. Click **Startup Options** in the left pane, and click **Restart service**.

Locate Chef Plug-In Workflows

You can use the following procedure to locate the Chef plug-in workflows.

Prerequisite

Log in to the vRealize Orchestrator client at https://<*orchestrator_server FQDN or IP address*>:8281.

Procedure

1. Select **Design** from the drop-down menu at the top of the client.
2. Click the Workflows icon ( ).
3. Select **Library** > **CHEF**.
   The location for specific workflows is indicated by the path mentioned in each of the following sections. For example, to locate workflows that add or delete a Chef host, you select **Library** > **CHEF** > **Configuration** in the left pane of the vRealize Orchestrator client.

Chef Plug-In Configuration Workflows

These workflows allow you to register and manage the Chef Server connection details in the vRealize Orchestrator inventory.

Add or Remove Chef Host Workflows

Library > CHEF > Configuration

Workflows for adding or removing the Chef host from the vRealize Orchestrator inventory.

Add Chef Host

Adds a CHEF:ChefHost object to the vRealize Orchestrator inventory.

Delete Chef Host

Removes a CHEF:ChefHost object from the vRealize Orchestrator inventory.

Common Parameters for Configuration Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
HOSTNAME	String	Hostname/IP Address of the Chef server.
CLIENTNAME	String	API client name used to connect to the Chef server.
ORGNAME	String	Chef organization to connect to.
CLIENTPEM	SecureString	PEM key for the API client for Chef server authentication.

ORGSECRET	SecureString	(Optional) Secret used for data bag encryption.
PROXYHOST	String	(Optional) Web proxy hostname, if needed.
PROXYPORT	Number	(Optional) Port number for web proxy host.
PROXYUSERNAME	String	(Optional) User name for proxy authentication.
PROXYPASSWORD	SecureString	(Optional) Password for proxy authentication.
IGNOREWARNINGS	Boolean	When set to true, the certificate is accepted silently and added to the trusted store.

Chef Plug-In Server Workflows

These workflows allow interaction between the Chef server REST API and the API client, cookbook, data bag, environment, node, and role endpoints.

API Client Workflows

Library > CHEF > Clients

Workflows for manipulating API clients.

Add New Client

Adds a new API client to a Chef server.
Returns the private and public keys for the new client.
The private key is only returned this one time. It is not stored on the Chef server.

Delete Client

Deletes a Chef API client from the server.
There is no response body from the Chef server.

Get Client Details

Retrieves API client details from the Chef server.

Get Clients

Retrieves an array of API client names from the Chef server.

Common Parameters for API Client Workflows

Name	Type	Description
HOST	CHEF:ChefHost	The Chef server connection to use.
CLIENT	String	The name of the API client to be added | deleted | queried.

Cookbook Workflows

Library > CHEF > Cookbooks

These workflows allow the retrieval of information about the cookbooks on a Chef server. Cookbooks define a scenario and provide everything required to support the scenario.

Get Cookbooks

Retrieves a list of cookbooks on the Chef server.

Get Cookbook Versions

Retrieves the versions of a particular cookbook on the Chef server.

Get Cookbook Version Details

Retrieves the recipes and additional details from a particular cookbook version on the Chef server.

Common Parameters for Cookbook Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
COOKBOOK	String	Name of the cookbook to be queried.
VERSION	String	Cookbook version to be queried.

Data Bag Workflows

Library > CHEF > Data Bags

Workflows for manipulating the data bags on a Chef server.

Add Data Bag Item

Adds a new data bag item or updates an existing data bag item on a Chef server.

Add New Data Bag

Adds a new data bag to a Chef server.

Delete Data Bag

Deletes a data bag on a Chef server.

Delete Data Bag Item

Deletes a data bag item on a Chef server. Returns the last defined attributes for the data bag item.

Delete Data Bag Item Attributes

Deletes the value of a single attribute of a data bag item on a Chef server.

Get Data Bag Item

Retrieves the keys or values of a data bag item on a Chef server. Attempts to decrypt the values of encrypted items with the org secret set on the CHEF:ChefHost inventory object.

Get Data Bag Item Attribute

Retrieves the value of a single data bag item attribute on a Chef server.

Get Data Bag Items

Retrieves a list of item names from a data bag on a Chef server.

Get Data Bags

Retrieves a list of data bag names from a Chef server.

Common Parameters for Data Bag Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
DATABAG	String	Data bag name.
ITEM	String	Data bag item name.
ITEM_JSON	String	The value of a data bag item in JSON format. Ex: {“string1”:”value1”, “number”:34, “array”:[“one”,”two”,”three”] “obj”: {“att1”:”val1”,”att2”:”val2”}
ENCRYPTED	Boolean	When true, the data bag item is encrypted on the Chef server using the org secret defined on the Chef host inventory object.
KEY	String	Individual data bag item attribute identifier.

Environment Workflows

Library > CHEF > Environments

Workflows for managing environments on a Chef server.

Add New Environment

Adds a new environment to a Chef server.

Get Environment Details

Retrieves details of a particular environment on a Chef server.

Get Environments

Retrieves a list of environments from a Chef server.

Remove Cookbook Version Constraint on Environment

Removes a version constraint for a cookbook in an environment on a Chef server.

Set Cookbook Version Constraint on Environments

Adds or updates a cookbook version constraint for an environment on a Chef server.

Common Parameters for Environment Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
ENVIRONMENT	String	Name of the environment to be managed.
DESCRIPTION	String	Description of the environment.
COOKBOOK	String	Name of the Cookbook to be version restricted.

Bootstrapping Linux Nodes using SSH Workflows

Library > CHEF > Linux Node Management - SSH

Workflows for bootstrapping Linux nodes using SSH.

Linux SSH Chef-Client First Run Validatorless

Validatorless registration workflow eliminates the need to use the validation key to provision new Chef nodes with Knife. The workflow creates a client object for the node and then creates the node object with its own run-list, environment, and initial JSON attributes using the node's respective client key.

The workflow creates the following files on the target system using SSH:

- `/etc/chef/client.rb`
- `/etc/chef/client.pem`

The workflow then executes the first chef-client run.

For additional details about validatorless bootstrap with Chef, see: https://www.chef.io/blog/2015/04/16/validatorless-bootstraps/.

Linux SSH Delete Instance with Own Key

Deletes a Chef node and respective client using a client's key extracted from the guest using SSH.

The `client.pem` file is deleted from the guest if both the node and respective client are deleted. This allows the node to register with Chef again.

Linux SSH Get client.pem

Retrieves the contents of `/etc/chef/client.pem` using SSH.

Linux SSH Install Omnibus Chef Client

Installs the latest version of the Chef client with the omnibus installer using SSH.
For more information about the omnibus installer, see: https://docs.chef.io/install_omnibus.html.

Run Chef-Client Using SSH

Executes chef-client using SSH and checks for any errors or warnings. Returns the chef-client output.

Common Parameters for Linux Node Management - SSH Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
ENVIRONMENT	String	Chef environment used to build the node.
RUNLIST	Array/String	Chef run-list of roles and recipes used to build the node. Ex: `role[us_west],recipe[chef-client::cron]`
NODENAME	String	Name for this new node. This is required for validatorless first run.
SSLVERIFYNONE	Boolean	When true, disables verification of the Chef server SSL certificate between the chef-client and Chef server.
IPADDRESS	String	IP address or host name for the SSH connection to the new node.
PASSWORDAUTHENTICAITON	Boolean	When set to true, uses password authentication for SSH to the node. When set to false, uses key authentication.
USERNAME	String	SSH user name for the node.
PASSWORD	SecureString	SSH password for the node when passwordAuthentication is true.
PATH	String	Path to the SSH private key on the vRealize Orchestrator host when passwordAuthentication is false.
PASSPHRASE	SecureString	Passphrase for the SSH private key if needed when passwordAuthentication is false.

Bootstrapping Linux Nodes using VMware Tools Workflows

Library > CHEF > Linux Node Management - VMware Tools

Workflows for bootstrapping Linux nodes using VMware Tools.

Linux VMware Tools Chef-Client First Run Validatorless

Validatorless registration workflow eliminates the need to use the validation key to provision new Chef nodes with knife. The workflow creates a client object for the

node and then creates the node object with its own run-list, environment, and initial JSON attributes using the node's respective client key.
The workflow creates the following files on the target system using VMware Tools:

- `/etc/chef/client.rb`
- `/etc/chef/client.pem`

The workflow then executes the first chef-client run.

For more information about validatorless bootstrap with Chef, see: https://www.chef.io/blog/2015/04/16/validatorless-bootstraps/.

Linux VMware Tools Delete Instance with Own Key

Deletes a Chef node and respective client using a client's key extracted from the guest using VMware Tools.
The `client.pem` file is deleted from the guest if both the node and respective client are deleted. This allows the node to register with Chef again.

Linux VMware Tools Get client.pem

Retrieves the contents of `/etc/chef/client.pem` using VMware Tools.

Linux VMware Tools Install Omnibus Chef Client

Installs the latest version of the Chef client with the omnibus installer using VMware Tools.
For more information about the omnibus installer, see: https://docs.chef.io/install_omnibus.html.

Run Chef-Client Using VMware Tools

Executes chef-client on a Linux VM using VMware Tools and checks for any errors or warnings.

Common Parameters for Linux Node Management - VMware Tools Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
ENVIRONMENT	String	Chef environment used to build the node.
RUNLIST	Array/String	The Chef run-list of roles and recipes used to build the node. Ex: role[us_west],recipe[chef-client::cron]
NODENAME	String	The name for this new node. This is required for validatorless first run.
SSLVERIFYNONE	Boolean	When true, disables verification of the Chef server SSL certificate between the chef-client and Chef server.
VM	VC:VirtualMachine	vCenter virtual machine of the Linux node.
USERNAME	String	Guest user name.
PASSWORD	SecureString	Guest password.

Chef Server Node Workflows

Library > CHEF > Nodes

Workflows for manipulating nodes on a Chef server.

Add New Node

Adds a new node to a Chef server.

Delete Node

Deletes a node on a Chef server and returns the node's last known state. Typically a node has a respective API client as well. This workflow also deletes the client if **deleteClient** is set to true.

Get Node Details

Retrieves details of a particular node on a Chef server.

Get Nodes

Retrieves a list of nodes from a Chef server.

Tag Chef Node

Adds a tag to a Chef node. This is similar to the `knife tag create <node> <tag>` command.

Untag Chef Node

Removes a tag to a Chef node. This is similar to the `knife tag delete <node> <tag>` command.

Update Environment for Node

Updates the Chef environment for a node on a Chef server. The chef-client must run on the node for the update to happen.

Update Run-List on Node

Updates the Chef run-list for a node on a Chef server. Requires the chef-client to run on the node for the update to happen.

Common Parameters for Node Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
NODE	String	Name of the Chef node being managed.
TAG	String	Name of the knife tag to add or remove from a Chef node.
ENVIRONMENT	String	Chef environment to associate the node with.
RUNLIST	Array/String	Chef run-list of roles and recipes to build the node with. Ex: `role[us_west],recipe[chef-client::cron]`

Node Attribute Workflows

Library > CHEF > Nodes > Attributes

Workflows for manipulating node attributes on a Chef server.

Delete Single Node Attribute

Deletes a single node attribute from a node on a Chef server.

Set Node Attributes

Adds or replaces a set of attributes for a node on a Chef server.

Set Single Node Attribute

A simpler version of Set Node Attributes that will add or replace a single node attribute of a node on a Chef server.

Common Parameters for Node Attribute Workflows

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>HOST</td><td>CHEF:ChefHost</td><td>Chef server connection to use.</td></tr>
<tr><td>NODE</td><td>String</td><td>Name of the Chef node being managed.</td></tr>
<tr><td>ATTRIBUTECLASS</td><td>String</td><td>Chef attribute type to set on the node.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>Normal</td><td>Attribute persists in the node object.</td></tr>
<tr><td>Override</td><td>Attribute is automatically reset at the start of every chef-client run and has higher attribute precedence than default, force_default, and normal attributes.</td></tr>
</table>
</td></tr>
<tr><td>ATTRIBUTENAME</td><td>String</td><td>Single attribute name to query or set.</td></tr>
<tr><td>ATTRIBUTESJS</td><td>String</td><td>Collection of Attributes to set in JSON format. Ex:
{“Att1_String”:”value1”,
“Att2_Number”:34,
“Att3_boolean”:true,
“Att4_array”:[“one”,”two”,”three”]
“Att5_obj”: {“att1”:”val1”,”att2”:”val2”}}</td></tr>
<tr><td>ATTRIBUTEVALUEJS</td><td>String</td><td>The attribute value in JSON format. Ex:
string: “value1”
number: 34
Boolean: true
array: [“one”,”two”,”three”]
object: {“att1”:”val1”,”att2”:”val2”}

<b>Note</b>: the string value must be quoted. This allows a distinction between string, number and Boolean values.</td></tr>
</table>

Role Workflows

Library > CHEF > Roles

Workflows for managing roles on a Chef server.

Add New Role

Adds a new role to a Chef server.

Delete Role

Deletes a role on a Chef server.

Get Role Details

Retrieves details of a particular role on a Chef server.

Get Roles

Retrieves a list of roles from a Chef server.

Update Run-List on Role

Updates the Chef run-list for a role on a Chef server. Requires the chef-client to run on each node that has the role in their run-list for the update to happen.

Common Parameters for Role Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
ROLE	String	The name of the Chef role being managed.
DESCRIPTION	String	A description of the Chef role on the Chef server.
RUNLIST	Array/String	Chef run-list of roles and recipes used to build the node. Ex: `role[us_west],recipe[chef-client::cron]`

Role Attribute Workflows

Library > CHEF > Roles > Attributes

Workflows for manipulating role attributes on a Chef server.

Delete Single Role Attribute

Deletes a single node attribute from role on a Chef server.

Set Role Attributes

Adds or replaces a set of attributes for a role on a Chef server.

Set Single Role Attribute

A simpler version of the Set Role Attributes workflow that adds or replaces a single role attribute for a role on a Chef server.

Common Parameters for Node Attribute Workflows

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>HOST</td><td>CHEF:ChefHost</td><td>The Chef server connection to use.</td></tr>
<tr><td>ROLE</td><td>String</td><td>The name of the Chef role being managed.</td></tr>
<tr><td>ATTRIBUTECLASS</td><td>String</td><td>Chef attribute type to set on the node.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>Default</td><td>Attribute is automatically reset at the start of every chef-client run and has the lowest attribute precedence.</td></tr>
<tr><td>Override</td><td>Attribute is automatically reset at the start of every chef-client run and has higher attribute precedence than default, force_default, and normal attributes.</td></tr>
</table></td></tr>
<tr><td>ATTRIBUTENAME</td><td>String</td><td>Single attribute name to query or set.</td></tr>
<tr><td>ATTRIBUTESJS</td><td>String</td><td>Collection of Attributes to set in JSON format. Ex:
{“Att1_String”:”value1”,
“Att2_Number”:34,
“Att3_boolean”:true,
“Att4_array”:[“one”,”two”,”three”]
“Att5_obj”: {“att1”:”val1”,”att2”:”val2”}}</td></tr>
<tr><td>ATTRIBUTEVALUEJS</td><td>String</td><td>The attribute value in JSON format. Ex:
string: “value1”
number: 34
Boolean: true
array: [“one”,”two”,”three”]
object: {“att1”:”val1”,”att2”:”val2”}

<b>Note</b>: the string value must be quoted. This allows distinction between string, number and Boolean values.</td></tr>
</table>

Bootstrapping Windows Nodes using Remote PowerShell (PS) Workflows

Library > CHEF > Windows Node Management - Remote PowerShell

Workflows for bootstrapping Windows nodes using remote PowerShell.

Windows PS Chef-Client First Run Validatorless

Validatorless registration workflow eliminates the need to use the validation key to provision new Chef nodes with Knife. The workflow creates a client object for the node and then creates the node object with its own run-list, environment, and initial JSON attributes using the node's respective client key.

The workflow creates the following files on the target system using remote PowerShell:

- `c:\chef\client.rb`
- `c:\chef\client.pem`

The workflow then executes the first chef-client run.

For more information about validatorless bootstrap with Chef, see: https://www.chef.io/blog/2015/04/16/validatorless-bootstraps/.

Windows PS Delete Instance with Own Key

Deletes a Chef node and respective client using a client's key extracted from the guest using remote PowerShell.
The `client.pem` file is deleted from the guest if both the node and respective client are deleted. This allows the node to register with Chef again.

Windows PS Get chef-client Version

Retrieves the installed version of the chef-client using remote PowerShell.

Windows PS Get client.pem

Retrieves the contents of `c:\chef\client.pem` using remote PowerShell.

Windows PS Install Chef Client

Uses WinRM to execute a PowerShell script that does the following:

- Downloads the chef-client MSI installer to the guest using a URL.
- Runs the installer.
- Confirms that the installation of the chef-client is successful.

Windows PS Run chef-client

Executes the chef-client using remote PowerShell and checks for any errors or warnings.

Common Parameters for Windows Node Management – Remote PowerShell Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
ENVIRONMENT	String	Chef environment used to build the node.
RUNLIST	Array/String	Chef run-list of roles and recipes used to build the node. Ex: `role[us_west],recipe[chef-client::cron]`
NODENAME	String	The name for this new node. This is required for validatorless first run.
SSLVERIFYNONE	Boolean	When set to true, disables verification of the Chef server SSL certificate between the chef-client and Chef server.
IPADDRESS	String	IP address or host name for the WinRM and remote PowerShell connection to the new node.
TRANSPORTPROTOCOL	String	The WinRM transport protocol to use. Valid values: HTTP or HTTPS.
USERNAME	String	WinRM user name for the node.
PASSWORD	SecureString	WinRM password for the node.

AUTHENTICATION	String	Authentication scheme to use for WinRM. Valid values: Basic or Kerberos.
MSIURL	String	URL for the chef-client MSI installer.
INSTALLASSERVICE	Boolean	When set to true, the chef-client is installed as a Windows service during installation.

Bootstrapping Windows Nodes Using VMware Tools Workflows

Library > CHEF > Windows Node Management - VMware Tools

Workflows bootstrapping Windows nodes using VMware Tools.

Windows PS Chef-Client First Run Validatorless

Validatorless registration workflow eliminates the need to use the validation key to provision new Chef nodes with Knife. The workflow creates a client object for the node and then creates the node object with its own run-list, environment, and initial JSON attributes using the node's respective client key.

The workflow creates the following files on the target system using VMware Tools:

- `c:\chef\client.rb`
- `c:\chef\client.pem`

The workflow then executes the first chef-client run.

For additional details about validatorless bootstrap with Chef, see: https://www.chef.io/blog/2015/04/16/validatorless-bootstraps/.

Windows VMware Tools Delete Instance With Own Key

Deletes a Chef node and respective client using a client's key extracted from the guest using VMware Tools.

The system deletes the `client.pem` file from the guest if both the node and respective client are deleted. This allows the node to register with Chef again.

Windows VMware Tools Get client.pem

Retrieves the contents of `c:\chef\client.pem` using VMware Tools.

Windows VMware Tools Install Chef Client

Executes a PowerShell script using WinRM that does the following:

- Downloads the chef-client MSI installer to the guest using a URL.
- Runs the installer.
- Confirms that the installation of the chef-client is successful.

Run Chef-Client Using VMware Tools

Executes chef-client on a Windows VM using VMware Tools and checks for any errors or warnings.

Common Parameters for Windows Node Management - VMware Tools Workflows

Name	Type	Description
HOST	CHEF:ChefHost	Chef server connection to use.
ENVIRONMENT	String	Chef environment used to build the node.
RUNLIST	Array/String	The Chef run-list of roles and recipes used to build the node. Ex: `role[us_west],recipe[chef-client::cron]`
NODENAME	String	The name for this new node. This is required for validatorless first run.
SSLVERIFYNONE	Boolean	When set to true, disables verification of the Chef server SSL certificate between the chef-client and Chef server.
VM	VC:VirtualMachine	vCenter virtual machine on the Windows node.
USERNAME	String	Guest user name.
PASSWORD	SecureString	Guest password.
MSIURL	String	URL for the chef-client MSI installer.
INSTALLASSERVICE	Boolean	When set to true, the chef-client is installed as a Windows service.

vRealize Automation Extensibility Sample Workflows

Library > CHEF > vRA Extensibility Samples

The following workflow samples can be triggered during the vRealize Automation virtual machine life cycle. Bootstrapping of a VM occurs during the **Machine Provisioned** state. Chef Node cleanup on the Chef server occurs during the **Machine Disposing** state.

Two different styles of workflows are provided based on the version of vRealize Automation in use. Each style's configuration to call the workflows is different, but the Chef-specific custom properties are the same for both styles.

Event Broker Service (EBS) Style workflows

These workflows are supported by vRealize Automation 7. You must create two workflow subscriptions as a tenant administrator to enable these workflows.

The following table displays the configuration of each subscription. Use the vRealize Automation 7 console to create each subscription by navigating to **Administration** > **Events** > **Subscriptions**.

Subscription Settings to Call the EBS – Machine Provision – Chef Workflow

Subscription Setting	Choice or Value
EVENT TOPIC	Machine provisioning
CONDITIONS	Run based on conditions All of the following Data > Life cycle state > Life cycle state name Equals VMPSMasterWorkflow32.MachineProvisioned Data > Life cycle state > State phase Equals PRE
WORKFLOW	Orchestrator > Library > CHEF > vRA Extensibility Samples > vRA 7 EBS > EBS – Machine Provisioned - Chef
PRIORITY	Example: 10
TIMEOUT	Number of minutes. This depends how long the installation or chef-client run typically takes. Example: 60
BLOCKING	Selected The provisioning life cycle waits for this workflow to complete

Subscription Settings to Call the EBS – Delete Chef Node Workflow

Subscription Setting	Choice / Value
EVENT TOPIC	Machine provisioning
CONDITIONS	Run based on conditions All of the following Data > Life cycle state > Life cycle state name Equals VMPSMasterWorkflow32.Disposing Data > Life cycle state > State phase Equals PRE
WORKFLOW	Orchestrator > Library > CHEF > vRA Extensibility Samples > vRA 7 EBS> EBS – Delete Chef Node
PRIORITY	Not applicable for Non-Blocking subscriptions.
TIMEOUT	Not Applicable for Non-Blocking subscriptions.
BLOCKING	Unselected. The provisioning life cycle does not wait for this workflow to complete. Fire and forget.

Create Property Group for Chef EBS Workflows

After the workflow subscriptions have been created, custom properties need to be associated with blueprints to bootstrap new VMs with Chef. The easiest way is to create a property group and associate the property group with the blueprint. The workflow **Library > CHEF > vRA Extensibility Samples > vRA 7 EBS > Create Property Group for Chef EBS Workflows** can create a property group with all of the required properties.

The workflow prompts the user with the following questions:

- Which vRealize Automation host do you want to create the Property Group on?
  Make sure this is a vCACCAFE:vCACHost object for the desired tenant and is authenticated as a tenant admin.
- What is the name of the new Property Group?
  You can have different property groups for different blueprints
- Which Chef server connection do you want to use?
  The system creates Chef nodes for new VMs in the same organization that the Chef server connection uses.
- Which Chef environment are the new nodes part of?
- What is the Chef run-list for these new nodes?
- Should the workflow attempt to install the chef-client in VMs guest OS?
  This should only be true if the VMs do not have the chef-client already installed in the template that they are cloned from.
- If you are installing the chef-client using the Windows MSI installer, what is the chef-client URL?
- If you are installing the chef-client on a Windows VM, do you want the chef-client installed as a service?
- What are the guest credentials for the VMs?

After the property group is created, the property group can be associated with a blueprint.

Classic IaaS Extensibility Workflows

These workflows are supported by vRealize Automation 6.2 and 7.0. The workflows use the vRealize IaaS plugin to configure vRealize Orchestrator.

Refer to the appropriate vRealize Automation documentation for information about preparing vRealize Automation to execute vRealize Orchestrator workflows during the virtual machine life cycle:

vRealize Automation 7.0 –
See Installing the vRealize Automation Plug-in (for vRealize Orchestrator standalone deployments) and

Configuring the vRealize Automation Plug-in for Machine Extensibility.

vRealize Automation 6.2
See Installing the vRealize Automation Plug-in (for vRealize Orchestrator standalone deployments) and
Configuring the vRealize Automation Plug-in for Machine Extensibility.

Ensure at least the following state change workflow stubs are enabled while executing the **Install vCO customization** workflow:

- WFStubMachineDisposing
- WFStubMachineProvisioned

Create Build Profile for Chef WF Stubs

Creates a vRealize Automation build profile to bootstrap new VMs with Chef.

The workflow prompts the user with the following questions:

- Which vRealize Automation IaaS host do you want to create the build profile on?
- What is the name of the new build profile?
- Which Chef server connection do you want to use?
  The system creates Chef nodes for new VMs in the same organization that the Chef server connection uses.
- Which Chef environment are the new nodes part of?
- What is the Chef run-list for these new nodes?
- Should the workflow attempt to install the chef-client in the VMs guest OS?
  This should only be true if the VMs do not have the chef-client already installed in the template that they are cloned from.
- If you are installing the chef-client using the Windows MSI installer, what is the chef-client URL?
- If you are installing the chef-client on a Windows VM, do you want the chef-client installed as a service?

After the build profile is created, the build profile can be associated with a blueprint.

Machine Provisioned - Chef

This example is triggered at the Machine Provisioned state of the vRealize Automation virtual machine life cycle. At the Machine Provisioned state, the VM is cloned and customized, and is on the network. If the custom property chef.installClient is set to true, the workflow installs the chef-client in the guest OS using VMware Tools.

The workflow performs a validatorless chef-client first run using VMware Tools.

Machine Disposing - Chef

This example removes the Chef node from the Chef server while a machine is disposed. The workflow uses the chef.orgName custom property to find the appropriate Chef server connection. It deletes both the Chef node and the respective client objects on the machine from the Chef server.

The following custom properties apply to both the EBS and Classic IaaS Extensibility workflows.

vRealize Automation Custom Properties used with vRealize Automation Extensibility Samples

Name	Description
CHEF.ORGNAME	Name of the Chef organization. This is used to look up a corresponding Chef server connection.
CHEF.RUNLIST	A comma-delimited list of run-list entries to apply to the new node.
CHEF.ENVIRONMENT	The Chef environment where the system adds the node.
CHEF.NODENAME	(Optional) A custom node name for a VM. The name of the VM is the default name.
CHEF.INSTALLCLIENT	When set to true, the workflow installs the chef-client on the VM. When set to false, the VM template installs the chef-client on the VM.
CHEF.WINDOWS.INSTALLASSERVICE	When set to true, the workflow installs the chef-client on Windows as a service. When set to false, the workflow does not install the chef-client on Windows as a service.
CHEF.WINDOWS.MSIURL	URL for the chef-client MSI Installer. For more information, see: https://docs.chef.io/install_windows.html.
CHEF.GUEST.USERNAME	Guest OS user name for SSH, Remote PowerShell or VMware Tools usage. EBS workflows only.
CHEF.GUEST.PASSWORD	Guest OS password for SSH, Remote PowerShell or VMware Tools usage. EBS workflows only.
CHEF.WINDOWS.MSIURL	URL for the chef-client MSI Installer. For more information, see: https://docs.chef.io/install_windows.html.

Using Chef Plugin with vRealize Automation XaaS and Resource Actions

Chef as a Service (XaaS) in Converged Blueprints

Using XaaS and converged blueprints in vRealize Automation 7.0 makes it possible to give users more options at request time with their Chef-based services. This can allow users to choose the environment or runlist of new Chef nodes at request time. The environment dropdown and potential runlist entries are read from the Chef Server at request time.

The following node bootstrap workflows are a good fit for bootstrapping a node with XaaS:

- Library > CHEF >Linux Node Management > SSH > Linux SSH Chef-client first run Validatorless
- Library > CHEF > Windows Node Management > Remote PowerShell > Windows PS Chef-Client first run Validatorless

Tips for using Chef Plugin workflows with XaaS Converged Blueprints:

- Set the Chef Server input as a constant on the XaaS blueprint.
- Set the Runlist form type as **dual list**. Size large.
- For any XaaS form, input that is mapped from a custom property or field in the converged blueprint:
    - Mark it as not required in the XaaS blueprint form.
    - Map all fields from Blueprint Outputs or custom properties in the Converged Blueprint Designer.
    - Hide inputs that have been mapped to fields by setting **Visible** to **No** in the Converted Blueprint Designer.

The following table shows a typical field mapping of blueprint outputs to XaaS form inputs:

XaaS Input	Converged Blueprint Field
NODE NAME	Blueprint Outputs > IaaS Machine > Machine Name
IP ADDRESS	Blueprint Outputs > IaaS Machine > IP Address

Resource Actions

Suggested Day 2 Resource Action scenarios for the Chef Plugin:

Run chef-client on demand

By creating a wrapper workflow, you can run the chef-client on demand in the guest OS using one of the following communication protocols:

- SSH
- Remote PowerShell
- VM Tools

Related workflows:

- Library > CHEF > Linux Node Management > SSH > Run Chef-Client via SSH
- Library > CHEF > Linux Node Management > VMtools > Liunux VMtools Run Chef-Client
- Library > CHEF > Windows Node Management > PowerShell > Windows PS run chef-client
- Library > CHEF > Windows Node Management > VMtools > Windows VMtools run chef-client

The **Run Chef-Client** workflows in these folders have an output parameter that is the stdout of the chef-client run. This chef-client output can be checked for errors and emailed to the action requestor or machine owner using the following workflow:

- Library > Mail > Send Notification

Update Runlist or Environment of Node

Creating a wrapper workflow you can update the Runlist or Environment of a node. Related workflows:

- Library > CHEF > Nodes > Update Environment for Node
- Library > CHEF > Nodes > Update Runlist on Node

Optionally the workflow could run the chef-client on demand as suggested above.

Orchestrator Presentation Actions Provided with the Chef Plugin

vRealize Orchestrator workflows can incorporate dynamic workflow request forms. The dynamic attributes of a workflow request form are defined on the **Presentation** tab. Most of the workflows provided in the Chef Plug-in have dynamic request forms that can be used as samples. These dynamic request forms carry over to vRealize Automation XaaS and Resource Action forms. The Orchestrator Actions Module, **com.vmware.o11n.plugin.chef.helpers,** includes actions that can be used in workflow presentations. The request forms can assist with the following tasks:

- Creating a drop-down menu for data bags: **getDatabags**
- Creating a drop-down menu for Environments: **getEnvironments**
- Pre-format all roles and cookbook recipes for runlists: **getRunlistCandidates**

- Get the current runlist of a Node: **getNodeRunlist**
- Get the current environment of a Node: **getNodeEnvironment**