



## **Red Hat CloudForms 4.5**

### **Integration with AWS CloudFormation and OpenStack Heat**

How to install and configure Amazon CloudFormation and OpenStack Heat in a Red Hat CloudForms environment



# Red Hat CloudForms 4.5 Integration with AWS CloudFormation and OpenStack Heat

---

How to install and configure Amazon CloudFormation and OpenStack Heat in a Red Hat CloudForms environment

Red Hat CloudForms Documentation Team  
cloudforms-docs@redhat.com

## Legal Notice

Copyright © 2018 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This guide provides instructions on the implementation of Amazon CloudFormation and OpenStack Heat in Red Hat CloudForms, and discusses the various areas of integration. Information and procedures in this book are relevant to CloudForms Management Engine administrators. If you have a suggestion for improving this guide or have found an error, please submit a Bugzilla report at <http://bugzilla.redhat.com> against Red Hat CloudForms Management Engine for the Documentation component. Please provide specific details, such as the section number, guide name, and CloudForms version so we can easily locate the content.

---

## Table of Contents

<b>CHAPTER 1. INTRODUCTION</b>	<b>3</b>
<b>CHAPTER 2. INTEGRATION WITH AWS CLOUDFORMATION AND OPENSTACK HEAT</b>	<b>4</b>
<b>CHAPTER 3. CLOUD ORCHESTRATION</b>	<b>5</b>
3.1. ADDING A NEW ORCHESTRATION TEMPLATE	5
3.2. EDITING ORCHESTRATION TEMPLATES	5
3.3. COPYING ORCHESTRATION TEMPLATES	6
3.4. DELETING ORCHESTRATION TEMPLATES	6
<b>CHAPTER 4. CLOUDFORMATION PROVISIONING VIA SERVICES</b>	<b>7</b>
4.1. ADDING A NEW CATALOG	7
4.2. ADDING A NEW SERVICE DIALOG	7
4.3. ADDING A NEW CATALOG ITEM	8
4.4. ORDERING SERVICE	9
4.5. ORCHESTRATION STACKS	11



## CHAPTER 1. INTRODUCTION

AWS CloudFormation enables users to orchestrate the instantiation of multi-instance services via templates. You can use CloudFormation's sample templates or create your own templates to describe the AWS resources, and any associated dependencies or runtime parameters, required to run your applications. Similarly, you can configure and monitor cloud resources in Red Hat Enterprise Linux OpenStack Platform using the Orchestration service. The Orchestration service provides a framework through which you can define an instance's resource parameters (for example, floating IPs, volumes, or security groups) and properties (for example, key pairs, image to be used, or flavor) using OpenStack Heat templates.

Instances deployed using templates through the orchestration service are known as stacks. A user can author the stack templates, or can upload them from other sources. Red Hat CloudForms has enabled CloudFormation and Heat integration, and now allows you to launch, delete, and update stacks using the dashboard.

## CHAPTER 2. INTEGRATION WITH AWS CLOUDFORMATION AND OPENSTACK HEAT

Red Hat CloudForms integration with AWS CloudFormation and OpenStack Heat provides an ability to:

- Inventory all **AWS CloudFormation** and **OpenStack Heat** stacks and elements into **CFME's VMDB**.
- Model the relationships of instances to their stacks, inclusive of the UI. Example, selecting an instance within a region that is within a stack, the UI shows this on the standard instance view.
- Model the stack and its elements in the UI.



### NOTE

When importing a template into Red Hat CloudForms, the selected elements are converted according to their type. For example, lists convert to list boxes, and single items convert to text boxes.





## CHAPTER 3. CLOUD ORCHESTRATION

Cloud Orchestration is a service that allows you to create, update and manage cloud resources and their software components as a single unit and then deploy them in an automated, repeatable way through a template. Templates use a human-readable syntax and can be defined in text files (thereby allowing users to check them into version control). Templates allow you to easily deploy and re-configure infrastructure for applications within your cloud. A user can author the stack templates, or can upload them from other sources.

### 3.1. ADDING A NEW ORCHESTRATION TEMPLATE

Use this procedure to add new orchestration templates using the dashboard UI.

1. Navigate to **Services** → **Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Click  **Configuration**, then click  **Create a new Orchestration Template**. The **Adding a new Orchestration Template** window is displayed.

Adding a new Orchestration Template



New Orchestration Template Information

Name	<input type="text"/>
Description	<div></div>
Template Type	Amazon CloudFormation 
Draft	<input type="checkbox"/>

3. In **Name**, enter a name for the new template.
4. In **Description**, enter a description for the template. Select Amazon CloudFormation or OpenStack Heat from the **Template Type** list. The default is Amazon CloudFormation.
5. You can select the **Draft** box to create a draft template.
6. Define your new template following the specification structure of the selected **Template Type**.
7. Click **Add**.



### 3.2. EDITING ORCHESTRATION TEMPLATES

Use this procedure to edit orchestration templates using the dashboard UI.

1. Navigate to **Services** → **Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Select the orchestration template you want to edit from the **All Orchestration Templates** list.
3. Click  **Configuration**, then click  **Edit selected Orchestration Template**. The **Edit selected Orchestration Template** window is displayed.
4. You can only edit the **Name** and **Description** of a read-only template as there can be stacks associated with the selected template. For templates that are not read-only, you can edit all content in the template as required.
5. Click **Save**.

### 3.3. COPYING ORCHESTRATION TEMPLATES

Use this procedure to copy an orchestration template to create a new template.

1. Navigate to **Services** → **Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Click  **Configuration**, then click  **Copy selected Orchestration Template**. The **Copy selected Orchestration Template** window is displayed.
3. You can copy the selected template to create a new template, and include the changes as required.





#### NOTE

In order to create the new template its content must be unique.

4. Click **Save**.

### 3.4. DELETING ORCHESTRATION TEMPLATES

Use this procedure to delete orchestration templates using the dashboard UI.

1. Navigate to **Services** → **Catalogs** and select **Orchestration Templates** in the accordion menu.
2. Select the orchestration template you want to delete from the **All Orchestration Templates** list.
3. Click  **Configuration**, then click  **Remove selected Orchestration Template**.
4. A warning window to confirm the permanent removal of the selected item from the VMDB appears.
5. Click **OK**.

This instantly deletes the selected orchestration template. Note that only non read-only templates can be removed.

## CHAPTER 4. CLOUDFORMATION PROVISIONING VIA SERVICES

After creating your template, you can add it as a catalog item to the **Service Catalog**. **Stacks** can then be created from templates and launched from the **Service Catalog**.

### 4.1. ADDING A NEW CATALOG

Use this procedure to add a new catalog using the dashboard UI.



#### Adding a new Catalog

##### Basic Info

Name	<input type="text"/>
Description	<input type="text"/>

##### Assign Catalog Items



Unassigned:	Selected:
<div></div>	<div></div>

1. Navigate to **Services** → **Catalogs** and select **Catalogs** in the accordion menu.
2. Click  **Configuration**, then click  **Add a New Catalog**. The **Adding a new Catalog** window is displayed.
3. In **Basic Info**, add **Name** and **Description** for the new catalog.
4. You can assign catalog items in **Assign Catalog Item**.
5. Click **Add**.

### 4.2. ADDING A NEW SERVICE DIALOG

Use this procedure to add a new service dialog based on the input parameters defined in the orchestration template.

1. Navigate to **Services** → **Catalogs** and click **Orchestration Templates** in the accordion menu.
2. From **All Orchestration Templates**, select the orchestration template you want to create a service dialog from.

- Click  **Configuration**, then click  **Create Service Dialog** from **Orchestration Template**. The **Adding a new Service Dialog from Orchestration Template** window is displayed.

Adding a new Service Dialog from Orchestration Template "aws-demo"



Service Dialog Information

Service Dialog Name	<input type="text"/>
---------------------	----------------------

- In **Service Dialog Information**, add a **Service Dialog Name**.
- Click **Save**.

## 4.3. ADDING A NEW CATALOG ITEM

Use this procedure to add a new service catalog item using the dashboard UI.

- Navigate to **Services** → **Catalogs** and select **Catalog Items** in the accordion menu.
- Click  **Configuration**, then click  **Add a New Catalog Item**. The **Adding a new Service Catalog Item** window is displayed.

Adding a new Service Catalog Item

New Catalog Item

Catalog Item Type	<div>&lt;Choose&gt; ▼ &lt;Choose&gt; Amazon Generic OpenStack Orchestration RHEV SCVMM VMware</div>
-------------------	---

- Choose **Orchestration** from **Catalog Item Type**.

4. In **Basic Info**, add **Name** and **Description**. Select the **Display** in **Catalog box**.

## Adding a new Service Catalog Item

### Basic Info

Name / Description	catalog-item-1 / catalog-item-1	<input checked="" type="checkbox"/> Display in Catalog
Catalog	Heat Catalog	
Dialog	hot-dialog	
Orchestration Template	hot-demo	
Provider	<Choose>	
Provisioning Entry Point (NS/Cls/Inst)	/Cloud/Orchestration/Provisioning/StateMachines/Pr	X

5. Select the **Catalog**, **Dialog**, and **Orchestration Template** from their respective list.

6. Select **Provisioning Entry Point**. The default is

/Cloud/Orchestration/Provisioning/StateMachines/Provision/default.

7. Click **Add**.

## 4.4. ORDERING SERVICE

Use this procedure to order a service catalog item using the dashboard UI.

1. Navigate to **Services** → **Catalogs** and select **Service Catalogs** in the accordion menu. From **All Services** catalogs, select the **catalog item** that you want to order. The **Service** window with the name and description of the service to be ordered is displayed.

### Service "orch\_item"



Name	orch_item
Description	desc
Long Description	

**Order**

- Click **Order**. The **Order Service** window with **Options** and **Parameter** is displayed.

## Order Service "orch\_item"

### Options

Stack Name	<input type="text" value="heat-stack"/>
On Failure	<input type="button" value="Rollback"/> <input type="button" value="↕"/>
Timeout(seconds, optional)	<input type="text"/>

### Parameters

Private network name or ID	<input type="text" value="292fce8f-0365-4f63-88ee-"/>
Image name or ID	<input type="text" value="cirros"/>
Flavor	<input type="text" value="m1.small"/>
Key name	<input type="text" value="userkey"/>

- Enter stack name in **Stack Name**.
- The **On Failure** value is Rollback by default.
- Timeout** is optional. You can type the number of seconds to timeout the provision at the provider side.



#### NOTE

The number of seconds get converted (rounded) to minutes when ordering the provision through Red Hat Enterprise Linus OpenStack Platform. For example, 100 seconds rounds to two minutes.

- You can use the default parameter values from the template, or enter new values as appropriate.



#### NOTE

The Parameters vary per dialog; therefore, the parameters shown in the **Order Service** window may or may not exist depending on the dialog.

7. Click **Submit**.

The order request is submitted. After a request has been approved, the various stages of fulfillment are executed. You can see the progress status of the provisioning process in **Services** → **Requests**.

## 4.5. ORCHESTRATION STACKS

After ordering a service, you can see the progress state of the provisioning process in **Services** → **Requests**.

- Initially, the **Request State** shows **Pending** with its **Approval State** as **Pending Approval**.

### Requests

✔ Order Request was Submitted

#### Filter By

Requester:	Administrator
Approval State:	<input checked="" type="checkbox"/> Approved <input checked="" type="checkbox"/> Denied <input checked="" type="checkbox"/> Pending Approval
Type:	All
Request Date:	Last 7 Days
Reason:	
<input type="button" value="Apply"/> <input type="button" value="Reset"/> <input type="button" value="Default"/>	

Desc. by: Last Update

	Status	Request State	Request ID	Request	Request Type	Comple	Descript	Approva State	Approve By	Approve On	Created On	Last Update	Reason	Last Message	Region
?	Ok	Pending	22	Administra	Service Provision		Provisionir Service [orch_item from [orch_item	Pending Approval			05/28/15 23:54:58 UTC	05/28/15 23:54:58 UTC		Service_T - Request Created	Region 0

- After the request is **Approved**, the various stages of fulfillment are executed, and reflect accordingly under **Request State**.






Desc. by: Last Update															
	Status	Request State	Request ID	Request	Request Type	Comple	Descript	Approva State	Approve By	Approve On	Created On	Last Update	Reason	Last Message	Region
✔	Ok	Pending	22	Administra	Service Provision		Provisionir Service [orch_item from [orch_item	Approved	Administra	05/28/15 23:55:05 UTC	05/28/15 23:54:58 UTC	05/28/15 23:55:30 UTC	Auto-Appri	Creating Stack	Region 0

Desc. by: Last Update															
	Status	Request State	Request ID	Request	Request Type	Comple	Descript	Approva State	Approve By	Approve On	Created On	Last Update	Reason	Last Message	Region
✔	Ok	Finished	22	Administra	Service Provision	05/28/15 23:57:56 UTC	Provisionir Service [orch_item from [orch_item	Approved	Administra	05/28/15 23:55:05 UTC	05/28/15 23:54:58 UTC	05/28/15 23:57:56 UTC	Auto-Appri	Service Provisione Successful	Region 0

- After the **Request State** is **Finished**, you can see the stack entry created in **Compute** → **Clouds** → **Stacks**. In the screen capture below, you can see the heat-stack we created from the catalog item ordered from the **Service Catalog** as shown in the previous section.

## Orchestration Stacks

<input type="checkbox"/> (Check All) Asc. by: Name									
		Name <sup>A</sup>	Provider	Type	Status	Status Reason	Instances	Security Groups	Cloud Networks
<input type="checkbox"/>		BGMT3Az8Kn	ec2-east	OrchestrationStackA	ROLLBACK_COMP		0	0	0
<input type="checkbox"/>		bill528	qeblade38	OrchestrationStackC	CREATE_COMPLE	Stack create completed successfully	1	0	0
<input checked="" type="checkbox"/>		heat-stack	qeblade38	OrchestrationStackC	CREATE_COMPLE	Stack create completed successfully	1	0	0
<input type="checkbox"/>		newstack	ec2-east	OrchestrationStackA	CREATE_COMPLE		0	1	0
<input type="checkbox"/>		rhos_stack	qeblade38	OrchestrationStackC	CREATE_COMPLE	Stack create completed successfully	1	0	0

4. You can click on the stack to see a summary of its properties and relationships, and the instance(s) included in the stack. You can click on the instance(s) to see all instance details.

### heat-stack (Summary)

Properties	
Name	heat-stack
Description	Simple template to deploy a single compute instance
Type	OrchestrationStackOpenstack
Status	CREATE_COMPLETE
Status Reason	Stack create completed successfully

Relationships	
Cloud Provider	 qeblade38
Orchestration Template	 TOKkAEhF
Instances	 1
Security Groups	 0
Cloud Networks	 0
Parameters	 7
Outputs	 1
Resources	 1

You have now deployed instances and its associated collection of resources (called a stack) using an orchestration template.