



Red Hat CloudForms 4.5

Installing Red Hat CloudForms on OpenShift Container Platform

How to install and configure Red Hat CloudForms on an OpenShift Container
Platform environment

Red Hat CloudForms 4.5 Installing Red Hat CloudForms on OpenShift Container Platform

How to install and configure Red Hat CloudForms on an OpenShift Container Platform 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 how to install and configure Red Hat CloudForms on an OpenShift Container Platform environment. 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

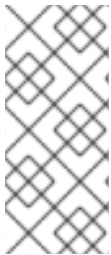
CHAPTER 1. INSTALLING RED HAT CLOUDFORMS	3
1.1. PREREQUISITES	3
1.1.1. Cluster Sizing	3
1.2. PREPARING TO DEPLOY CLOUDFORMS	4
1.3. DEPLOYING THE CLOUDFORMS APPLIANCE	6
1.4. VERIFYING THE CONFIGURATION	7
1.5. SCALING CLOUDFORMS APPLIANCES	8
1.6. OBTAINING POD ACCESS AND ROUTES	9
CHAPTER 2. MANAGING RED HAT CLOUDFORMS WITH OPENSIFT	10
2.1. CONFIGURING CUSTOM SSL CERTIFICATES FOR CLOUDFORMS	10
CHAPTER 3. LOGGING IN AFTER INSTALLING RED HAT CLOUDFORMS	11
3.1. CHANGING THE DEFAULT LOGIN PASSWORD	11
CHAPTER 4. TROUBLESHOOTING DEPLOYMENT	12
4.1. UNINSTALLING RED HAT CLOUDFORMS FROM A PROJECT	13

CHAPTER 1. INSTALLING RED HAT CLOUDFORMS

Red Hat CloudForms can be installed on OpenShift Container Platform in a few quick steps.

This procedure uses a template to deploy a multi-pod Red Hat CloudForms appliance with the database stored in a persistent volume on OpenShift Container Platform. It provides a step-by-step setup, including cluster administrative tasks as well as regular user information and commands.

The ultimate goal of the deployment is to be able to deconstruct the Red Hat CloudForms appliance into several containers running on a pod or a series of pods.



NOTE

Running Red Hat CloudForms on OpenShift Container Platform is available as a technology preview in this release of Red Hat CloudForms. For more information on the support scope for features marked as technology preview, see [Technology Preview Features Support Scope](#). Additionally, at this time only one Red Hat CloudForms instance can be run at one time, and upgrading is unavailable.

1.1. PREREQUISITES

To successfully deploy a Red Hat CloudForms appliance on OpenShift Container Platform, you need a functioning **OpenShift Container Platform 3.5** or newer install with the following configured:

- NFS or other compatible volume provider
- A `cluster-admin` user
- A regular user (such as an application developer)



IMPORTANT

OpenShift Container Platform 3.5 is required for this installation. Red Hat has not tested this procedure with earlier versions of OpenShift Container Platform.

The CloudForms deployment uses `.yaml` template files to create the appliance, including `cfme-template.yaml`, which is the CloudForms template used for the deployment, and `cfme-pv-example.yaml` and `cfme-pv-app-example.yaml`, two pod volume files.

These templates are available in RPMs from Red Hat-provided image streams. To obtain the templates:

1. Configure image streams as described in [OpenShift Container Platform Installation and Configuration](#).
2. After loading the image streams and templates, the templates will be available on your OpenShift system in `/usr/share/openshift/examples/cfme-templates`.

The CloudForms template points to several image files to create the OpenShift pods that comprise the appliance. These image files are obtained from the [Red Hat Container Catalog](#) during deployment.

1.1.1. Cluster Sizing

To avoid deployment failures due to resource starvation, Red Hat recommends the following minimum cluster size for a test environment:

- 1 master node with at least 8 vCPUs and 12GB of RAM
- 2 nodes with at least 4 vCPUs and 8GB of RAM
- 25GB of storage for CloudForms physical volume use

These recommendations assume CloudForms is the only application running on this cluster. Alternatively, you can provision an infrastructure node to run registry, metrics, router, and logging pods.

Each CloudForms application pod will consume at least 3GB of RAM on initial deployment (without providers added). RAM consumption increases depending on the appliance use. For example, after adding providers, expect higher resource consumption.

1.2. PREPARING TO DEPLOY CLOUDFORMS

To prepare for deploying the CloudForms appliance to OpenShift Container Platform, create a project, configure security contexts, and create pod volumes.

1. As a regular user, log in to OpenShift:

```
$ oc login -u <user> -p <password>
```

2. Create a project with your desired parameters. The **<project_name>** is mandatory, but **<description>** and **<display_name>** are optional:

```
$ oc new-project <project_name> \  
--description="<description>" \  
--display-name="<display_name>"
```

3. Add the **cfme-anyuid** service account to the **anyuid** security context constraint (SCC). Because the CloudForms image requires the root user, the **cfme-anyuid** service account for your namespace (project) must be added to the **anyuid** SCC before pods using the service account can run as root.

- a. As the admin user, add the **cfme-anyuid** service account by running:

```
$ oc adm policy add-scc-to-user anyuid system:serviceaccount:  
<your-namespace>:cfme-anyuid
```

- b. Verify that the **cfme-anyuid** service account is now included in the **anyuid** SCC:

```
$ oc describe scc anyuid | grep Users  
Users:      system:serviceaccount:<your-namespace>:cfme-anyuid
```

4. Add your default service account to the **privileged** security context. The default service account for your namespace (project) must be added to the **privileged** security context constraints (SCCs) before they can run privileged pods.

- a. As the admin user, add the default service account by running:

```
$ oc adm policy add-scc-to-user privileged system:serviceaccount:  
<your-namespace>:default
```


- b. Verify that your default service account is now included in the **privileged** security context constraints (SCCs):

```
$ oc describe scc privileged | grep Users
Users:
  system:serviceaccount:openshift-
infra:build-controller,system:serviceaccount:management-
infra:management-admin,system:serviceaccount:management-
infra:inspector-
admin,system:serviceaccount:default:router,system:serviceaccount:
default:registry,system:serviceaccount:<your-namespace>:default
```

5. Prepare persistent storage for the deployment. (Skip this step if you have already configured persistent storage.)

A regular CloudForms deployment needs at least two persistent volumes (PVs) to store CloudForms data. As the admin user, create two persistent volumes: one to host the CloudForms PostgreSQL database, and one to host the application data.

Example NFS-backed volume templates are provided by `cfme-pv-db-example.yaml` and `cfme-pv-server-example.yaml`, available from [GitHub](#).



NOTE

For NFS-backed volumes, ensure your NFS server firewall is configured to allow traffic on port 2049 (TCP) from the OpenShift cluster.

Red Hat recommends setting permissions for the pv-app (privileged pod volume) as 777, uid/gid 0 (owned by root). For more information on configuring persistent storage in OpenShift Container Platform, see the [OpenShift Container Platform Installation and Configuration](#) guide.

- a. Configure your NFS server host details within these files, and edit any other settings needed to match your environment.
- b. Run the following commands to create the two persistent volumes:

```
$ oc create -f cfme-pv-db-example.yaml
$ oc create -f cfme-pv-server-example.yaml
```

- c. Verify the pod volumes were created successfully:

```
$ oc get pv
NAME          CAPACITY   ACCESSMODES   RECLAIMPOLICY   STATUS
CLAIM  REASON    AGE
cfme-pv01    15Gi       RWO           Recycle         Available
30s
cfme-pv02    5Gi        RWO           Recycle         Available
19s
```



NOTE

Red Hat recommends validating NFS share connectivity from an OpenShift node before attempting a deployment.

6. Increase the maximum number of imported images on ImageStream.
By default, OpenShift Container Platform can import five tags per image stream, but the CloudForms repositories contain more than five images for deployments.

You can modify this setting on the master node at `/etc/origin/master/master-config.yaml` so OpenShift can import additional images.

- a. Add the following at the end of the `/etc/origin/master/master-config.yaml` file:

```
...
imagePolicyConfig:
  maxImagesBulkImportedPerRepository: 100
```

- b. Restart the master service:

```
$ systemctl restart atomic-openshift-master
```

1.3. DEPLOYING THE CLOUDFORMS APPLIANCE

To deploy the appliance on OpenShift Container Platform, create the Red Hat CloudForms template and verify it is available in your project.

1. As a regular user, create the Red Hat CloudForms template:

```
$ oc create -f templates/cfme-template.yaml
template "cloudforms" created
```

2. Verify the template is available with your project:

```
$ oc get templates
NAME          DESCRIPTION
PARAMETERS    OBJECTS
cloudforms    CloudForms appliance with persistent storage  18 (1
blank)        12
```

3. (Optional) Customize the template's deployment parameters. Use the following command to see the available parameters and descriptions:

```
$ oc process --parameters -n <your-project> cloudforms
```

To customize the deployment configuration parameters, run:

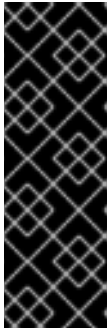
```
$ oc edit dc/<deployconfig_name>
```

4. To deploy Red Hat CloudForms from template using default settings, run:

```
$ oc new-app --template=cloudforms
```

Alternatively, to deploy Red Hat CloudForms from a template using customized settings, add the `-p` option and the desired parameters to the command. For example:

```
$ oc new-app --template=cloudforms -p
DATABASE_VOLUME_CAPACITY=2Gi,POSTGRESQL_MEM_LIMIT=4Gi,APPLICATION_DOMAIN=hostname
```



IMPORTANT

The **APPLICATION_DOMAIN** parameter specifies the hostname used to reach the CloudForms application, which eventually constructs the route to the CloudForms pod. If you do not specify the **APPLICATION_DOMAIN** parameter, the CloudForms application will not be accessible after the deployment; however, this can be fixed by changing the route. For more information on OpenShift template parameters, see the [OpenShift Container Platform Developer Guide](#).

1.4. VERIFYING THE CONFIGURATION

Verify the deployment was successful by running the following commands as a regular user under the Red Hat CloudForms project:



NOTE

The first deployment can take several minutes to complete while OpenShift downloads the necessary images.

1. Confirm the Red Hat CloudForms pod is bound to the correct security context constraints.
 - a. List and obtain the name of the **cfme-app** pod:

```
$ oc get pod
NAME                                READY    STATUS    RESTARTS   AGE
cloudforms-1-fzwzm                 1/1      Running   0           4m
memcached-1-6iuxu                  1/1      Running   0           4m
postgresql-1-2kxc3                 1/1      Running   0           4m
```

- b. Export the configuration of the pod:

```
$ oc export pod <cfme_pod_name>
```

- c. Examine the output to verify that **openshift.io/scc** has the value **anyuid**:

```
...
metadata:
  annotations:
    openshift.io/scc: anyuid
...
```

2. Verify the persistent volumes are attached to the **postgresql** and **cfme-app** pods:

```
$ oc volume pods --all
pods/postgresql-1-437jg
  pvc/cfme-pgdb-claim (allocated 2GiB) as cfme-pgdb-volume
  mounted at /var/lib/pgsql/data
```

```
secret/default-token-2se06 as default-token-2se06
  mounted at /var/run/secrets/kubernetes.io/serviceaccount
pods/cfme-1-s3bnp
pvc/cfme (allocated 2GiB) as cfme-app-volume
  mounted at /persistent
secret/default-token-9q4ge as default-token-9q4ge
  mounted at /var/run/secrets/kubernetes.io/serviceaccount
```

3. Check the readiness of the CloudForms pod:



NOTE

Allow approximately five minutes once pods are in running state for Red Hat CloudForms to start responding on HTTPS.

```
$ oc describe pods <cfme_pod_name>
...
Conditions:
  Type      Status
  Ready     True
Volumes:
  ...
```

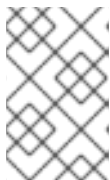
4. After you have successfully validated your CloudForms deployment, disable automatic image change triggers to prevent unintended upgrades.

By default, on initial deployments the automatic image change trigger is enabled. This could potentially start an unintended upgrade on a deployment if a newer image is found in the ImageStream.

Disable the automatic image change triggers for CloudForms deployment configurations (DCs) on each project with the following commands:

```
$ oc set triggers dc --manual -l app=cloudforms
deploymentconfig "memcached" updated
deploymentconfig "postgresql" updated

$ oc set triggers dc --from-config --auto -l app=cloudforms
deploymentconfig "memcached" updated
deploymentconfig "postgresql" updated
```



NOTE

The configuration change trigger is kept enabled; to have full control of your deployments, you can alternatively turn it off. See the [OpenShift Container Platform Developer Guide](#) for more information on deployment triggers.

1.5. SCALING CLOUDFORMS APPLIANCES

StatefulSets in OpenShift allow scaling of CloudForms appliances. See the [OpenShift Container Platform 3.5 Release Notes](#) for information on StatefulSets.



IMPORTANT

Each new replica (server) consumes a physical volume. Before scaling, ensure you have enough physical volumes available to scale.

The following example shows scaling using StatefulSets:

Example: Scaling to two replicas

```
$ oc scale statefulset cloudforms --replicas=2
statefulset "cloudforms" scaled
$ oc get pods
```

NAME	READY	STATUS	RESTARTS	AGE
cloudforms-0	1/1	Running	0	34m
cloudforms-1	1/1	Running	0	5m
memcached-1-mzeer	1/1	Running	0	1h
postgresql-1-dufgp	1/1	Running	0	1h

The newly created replicas will join the existing CloudForms region. For a StatefulSet with **N** replicas, when pods are being deployed, they are created sequentially, in order from {0..N-1}.

1.6. OBTAINING POD ACCESS AND ROUTES

To open a shell on the CloudForms pod, run:

```
$ oc rsh <pod_name> bash -l
```

To obtain host information from the route:

```
$ oc get routes
```

NAME	HOST/PORT	PATH	SERVICE
cloudforms	cfme.apps.e2e.example.com	cloudforms:443-tcp	passthrough

```
app=cloudforms
```

A route should have been deployed via template for HTTPS access on the CloudForms pod. Examine the output and point your web browser to the reported URL/host (in this example, **cfme.apps.e2e.example.com**).

CHAPTER 2. MANAGING RED HAT CLOUDFORMS WITH OPENSIFT

This section includes common tasks to manage your Red Hat CloudForms deployment from OpenShift.

2.1. CONFIGURING CUSTOM SSL CERTIFICATES FOR CLOUDFORMS

By default, the route that is deployed as part of the template uses edge termination and the certificates that OpenShift is installed with. It is possible to change this in the OpenShift UI with the following steps:

1. Navigate to **Applications** → **Routes**.
2. Click on the route named **httpd**, then select **Actions** → **Edit**.
3. Scroll down to the **Certificates** section. Here you can upload or paste the required certificate files.
4. Click **Save**.

CHAPTER 3. LOGGING IN AFTER INSTALLING RED HAT CLOUDFORMS

Once Red Hat CloudForms is installed, you can log in and perform administration tasks.

Log in to Red Hat CloudForms for the first time after installing by:

1. Navigate to the URL for the login screen. (<https://xx.xx.xx.xx> on the virtual machine instance)
2. Enter the default credentials (Username: **admin** | Password: **smartvm**) for the initial login.
3. Click **Login**.

3.1. CHANGING THE DEFAULT LOGIN PASSWORD

Change your password to ensure more private and secure access to Red Hat CloudForms.

1. Navigate to the URL for the login screen. (<https://xx.xx.xx.xx> on the virtual machine instance)
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

CHAPTER 4. TROUBLESHOOTING DEPLOYMENT

Under normal circumstances, the deployment process takes approximately 10 minutes. If the deployment is unsuccessful, examining deployment events and pod logs can help identify any issues.

1. As a regular user, first retry the failed deployment:

```
$ oc get pods
NAME                                READY    STATUS    RESTARTS   AGE
cloudforms-1-deploy                0/1      Error     0           25m
memcached-1-yasfq                  1/1      Running   0           24m
postgresql-1-wfv59                 1/1      Running   0           24m
```

```
$ oc deploy cloudforms --retry
Retried #1
Use 'oc logs -f dc/cloudforms' to track its progress.
```

2. Allow a few seconds for the failed pod to get re-scheduled, then check events and logs:

```
$ oc describe pods <pod-name>
...
Events:
  FirstSeen LastSeen Count From                                SubobjectPath  Type  Reason
Message
-----
15m 15m 1 {kubelet ocp-eval-node-2.e2e.example.com}
spec.containers{cloudforms} Warning Unhealthy Readiness probe
failed: Get http://10.1.1.5:80/: dial tcp 10.1.1.5:80: getsockopt:
connection refused
```

Liveness and readiness probe failures, like in the output above, indicate the pod is taking longer than expected to come online. In this case, check the pod logs.

3. As the **cfme-app** container is **systemd** based, use **oc rsh** instead of **oc logs** to obtain journal dumps:

```
$ oc rsh <pod-name> journalctl -x
```

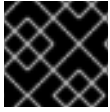
4. Transferring all logs from the **cfme-app** pod to a directory on the host for further examination can be useful for troubleshooting. Transfer the logs with the **oc rsync** command:

```
$ oc rsync <pod-name>:/persistent/container-deploy/log \
/tmp/fail-logs/
receiving incremental file list
log/
log/appliance_initialize_1477272109.log
log/restore_pv_data_1477272010.log
log/sync_pv_data_1477272010.log

sent 72 bytes  received 1881 bytes  1302.00 bytes/sec
total size is 1585  speedup is 0.81
```


4.1. UNINSTALLING RED HAT CLOUDFORMS FROM A PROJECT

If no longer needed, you can uninstall the Red Hat CloudForms pod from your project. Note the following commands do not remove SCC permissions, or the project itself.



IMPORTANT

Use this procedure if only Red Hat CloudForms exists in the project.

1. Inside the project, run the following as a regular user:

```
$ oc delete all --all
```

2. Wait approximately 30 seconds for the command to process, then run:

```
$ oc delete pvc --all
```