



Red Hat CloudForms Beta Features Guide

Feature and use case information for Red Hat
CloudForms 4.0 Beta

CloudForms Documentation Team

Red Hat CloudForms Beta Features Guide

Feature and use case information for Red Hat CloudForms 4.0 Beta

CloudForms Documentation Team
Red Hat Customer Content Services
docs@redhat.com

Legal Notice

Copyright © 2015 Red Hat.

This document is licensed by Red Hat under the Creative Commons Attribution-ShareAlike 3.0 Unported License. If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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, MetaMatrix, 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 MySQLAB 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 document is the release notes for Red Hat CloudForms X.Y and contains a summary of new features.

Table of Contents

1. Introduction to the CloudForms 4.0 Beta	7
Upstream Projects.....	7
Available Interfaces.....	7
How to Provide Feedback.....	7
2. Deploying the CloudForms 4.0 Beta	8
3. Key Features in CloudForms 4.0 Beta	8
Multi-Tenancy.....	8
Kubernetes / Container Management.....	11
Microsoft Azure Provider (Tech Preview).....	12
New Self-Service UI.....	14
External Authentication Enhancements.....	15
Other Provider Enhancements.....	18
Automate Method Changes.....	19
REST API Changes.....	20
Deprecations and Removals.....	22

1. Introduction to the CloudForms 4.0 Beta

We are pleased to announce the public release of Red Hat CloudForms 4.0 Beta.

As a cloud management platform, CloudForms enables rapid, horizontal scaling of your cloud environment. Appliances and virtual machines are delivered in several different formats to ease integration with multiple different public and private cloud providers.

Red Hat CloudForms 4.0 is a major release which introduces numerous enhancements to existing cloud provider management, providers, security features to easily implement OpenSCAP and other standards, and content management. Additionally, CloudForms 4.0 introduces a redesigned web user interface and a new self-service user interface to make overall navigation and administration easier and more streamlined.

Upstream Projects

Red Hat CloudForms is based on the upstream [ManageIQ project](#). The CloudForms 4.0 Beta is based on ManageIQ Capablanca release; full upstream release details are available at <https://github.com/ManageIQ/manageiq/blob/capablanca/CHANGELOG.md>.

Available Interfaces

CloudForms 4.0 provides several different interfaces:

- A full administrative web-based user interface
- A self-service user interface for end users
- A REST API
- Methods to be used for automation

How to Provide Feedback

There are several ways to provide feedback on the CloudForms 4.0 Beta release.

Option 1 (Recommended): Contact [Red Hat Technical Support](#) and open a support case for Red Hat CloudForms, version 4.0.

Option 2: Open a Bugzilla.

Our bug tracking tool is available at <https://bugzilla.redhat.com/>. You must have a Bugzilla account to file an issue. When filing a bug, use these settings:

- Classification: Red Hat
- Product: Red Hat CloudForms Management Engine
- Version: 4.0
- Severity
 - LOW: Minor issues that don't affect usability.

- MEDIUM: Issues that affect usability, but for which there might be a reasonable workaround. Attempted workflows cannot be blocked.
- HIGH: Severe issues that block reasonable execution or lead to serious performance degradation. There may be no reasonable workaround for a HIGH-severity issue.
- URGENT: Urgent is generally reserved for critical cases of data loss, irreversible feature damage, or component crash without recovery.
- Description. Be as specific in the steps it takes to reproduce the issue or what was happening at the time the issue occurred.

An issue can be filed for a defect or for a requested enhancement, feature, or change in behavior (RFE).

2. Deploying the CloudForms 4.0 Beta

CloudForms Management Engine is available in a variety of virtual machine formats to enable rapid introduction into your environment:

- Red Hat Enterprise Linux OpenStack Platform
- Red Hat Enterprise Virtualization
- VMware vSphere

There are a variety of scenarios for deploying the beta version of CloudForms 4.0:

- You have an existing CloudForms 3.x deployment and plan to upgrade to CloudForms 4.0.
- You want to evaluate CloudForms 4.0.

This beta is intended for evaluation and planning only. It is not supported to upgrade from CloudForms 4.0 Beta to CloudForms 4.0 GA.

Installation documentation is provided for all appliances at <https://access.redhat.com/documentation/en/red-hat-cloudforms/>.

3. Key Features in CloudForms 4.0 Beta

The primary features and use cases in the beta release of CloudForms 4.0 are highlighted here. Some scenarios are supplied to make it easier to evaluate the given functionality.

Multi-Tenancy

CloudForms 4.0 introduces the concept of *multi-tenancy*. In CloudForms 3.x, the assumption was that all elements within the infrastructure and inventory were grouped within a single organization. Multi-tenancy allows for siloed organizations which have separate authentication and users, different branding, and different security.

In CloudForms, it is also possible to have relationships between tenants and projects. Tenants can be totally separate or they can be in a parent-child or peer relationship. Tenants in a relationship can share or inherit certain configuration.

Additionally, tenants can have *projects* as children. A project is a self-contained tenant (meaning it cannot have children) and is useful to allocate resources to a small group or team within a larger organization.

If an administrator does not want to have multiple tenants, meaning the CloudForms infrastructure is the same as in the 3.x version, then there is the concept of a *root tenant*. All existing configuration is moved under that root entry (to fit within the tenancy model) but is otherwise unaltered.

For the beta, the test cases are related to the new tenant structure:

- Setting the right user permissions
- Creating tenants and projects
- Managing quotas and services
- Managing entries, like tagging and searching

UI Changes

The left navigation menu is now structured as a hierarchical tree, with tenants, child tenants, and projects all listed. Tenants are identified by a small house icon. Projects are identified by a pencil/paper icon.

Use case 1: Assign tenancy administrator roles

1. Open the Configure item in the top menu, and select the Configuration subtab.
2. Open Access Control, and select Groups.
3. Select the group to edit or create a new group.
4. Select the tenant administrator and tenant quota administrator roles to assign to the group.
5. Save.

Users in that group should now be able to create and edit tenants and projects.

Use case 2: Add a new tenant or project

1. Open the Configure item in the top menu, and select the Configuration subtab.
2. Open Tenants.
3. Select the tenant entry to edit.
4. In the Configuration drop-down menu, select Add child tenant or Add project.
5. Enter the name of the new tenant / project.

Use case 3: Manage tenant quotas

There must be a user with the tenant quota administrator role to perform this.

1. Open the Configure item in the top menu, and select the Configuration subtab.
2. Open Tenants.
3. Select the tenant entry to edit.
4. In the Configuration drop-down menu, select Manage quotas.
5. Set any quota limits for the tenant.
 - a. CPU
 - b. Memory
 - c. Storage
 - d. The maximum number of allowed virtual machines
 - e. The maximum number of allowed templates

Use case 4: Provisioning to a tenant using OpenStack Orchestration

[Integration with AWS and OpenStack Heat](#) covers how to create catalogs, catalog items, and services and then deploy them using OpenStack Orchestration. In CloudForms 4.0, there is an additional configuration field to assign the catalogs and deployments to a given tenant.

1. Create a new catalog and assign it to a tenant.
 - a. Open the Services tab, and select the Catalog subtab.
 - b. Select Catalogs on the left.
 - c. In the Configuration drop-down menu, click Add a New Catalog.
 - d. Enter the name, description, and assign any items.
 - e. Enter the tenant.
2. Create a new Orchestration service or catalog item to add to the catalog.
 - a. Open the Services tab, and select the Catalog subtab.
 - b. Select Services or Catalog Item on the left.
 - c. In the Configuration drop-down menu, click Add a New
 - d. Enter the name, description, and any required configuration.
 - e. Assign the tenant.
3. Order the service.
 - a. Open the Services tab, and select the Catalog subtab.
 - b. Select Service Catalogs on the left.
 - c. Select the service catalog, and click the Order button.
 - d. Enter the stack name and any other parameters.
 - e. Click submit.

Use case 5: Tagging tenants and projects

1. Open the Configure item in the top menu, and select the Configuration subtab.
2. Select the tenant in the navigation menu on the left.
3. In the tenant entry, open the Policy drop-down menu, and select Edit Tags.
4. Select the tag from the list, set a value, and save.

Kubernetes / Container Management

CloudForms 4.0 introduces container management along with image and flavor management for instances. The orchestration system can be Kubernetes or OpenShift, and it can work with both RHEV and VMware providers.

UI Changes

A new Containers area has been added to the top menu. This area has options to add and manage container providers and container images.

Use case 1: Adding a container provider

1. Open the Containers tab in the top menu, and select the Providers subtab.
2. Select the Configuration drop-down menu in the main window, and select Add a Container Provider.
3. Enter the name for the provider entry and set the type to OpenShift or Kubernetes. This opens up additional configuration fields.
4. Enter the name and connection information.
5. Save the configuration.

Use case 2: Topology Widget

1. Open the Containers tab in the top menu, and select the Providers subtab.
2. Select the provider from the list.
3. In the main window, click the Topology row in the Overview table on the right.

This displays an interactive topology graph, showing the relationships between the different elements.

- The topology graph includes pods, containers, services, nodes, virtual machines, and hosts within the overall container provider environment.
- CloudForms can identify virtual machines for Kubernetes and OpenShift container images which are running on either RHEV or VMware.
- Selecting any individual graph node will display a summary of details for the individual node.
- It is possible to drag elements to reposition the graph and to zoom in for details on a specific subset of items.

Use case 3: Running a SmartState analysis

1. Open the Containers tab in the top menu, and select the Container Images subtab.
2. Select the image from the list.
3. In the main window, click the Configuration drop-down menu, and select SmartState Analysis.

The image is scanned. The process will copy over any required files and install any required packages for the image. After reloading the image page, all new or updated packages are listed.

Microsoft Azure Provider (Tech Preview)

Microsoft Azure is being added a new provider type. This is a tech preview feature but it is available in the beta for evaluation. For information about our tech preview support policies, see <https://access.redhat.com/support/offerings/techpreview/>.

Use case 1: Set up CloudForms authentication to Azure

Azure uses OAuth2 to authentication web applications and services to its cloud. The overall Azure OAuth2 implementation is described in detail in the [Microsoft documentation](#). For CloudForms, the basic process is that the CloudForms server is registered with the central Azure Active Directory instance (not your organization's Active Directory instances). This generates a tenant ID, a client ID, and a client key which uniquely identify the CloudForms server. CloudForms connects to Azure through its resource web API, and it subsequently uses its client ID and client key to send and receive information about the Azure instances, sizes, and availability sets.

The procedure to register with the Azure Active Directory instance is covered in the Microsoft documentation here:

<https://msdn.microsoft.com/en-us/library/azure/dn798668.aspx>

Use case 2: Adding an Azure provider

1. Open the Clouds tab in the top menu, and select the Providers subtab.
2. Select the Configuration drop-down menu in the main window, and select Add a Cloud Provider.
3. Enter the name for the provider entry and set the type to Azure. This opens up additional configuration fields.
4. Enter the tenant ID.
5. Enter the client ID that was created when you authenticated the CloudForms server to the Azure server, and enter the generated client key.
6. Validate the configuration.
7. Save the configuration.

Use case 3: Viewing the inventory

Once the provider is configured, then CloudForms can query for any existing inventory and display and manage it within the CloudForms web UI. The inventory items which can be collected are availability sets, sizes or series, and instances.

Note: Gathering information on existing images is not yet enabled in the ARM REST API.

To view availability set (called *zones* in CloudForms) information:

1. Open the Clouds tab in the top menu, and select the Providers subtab.
2. Expand the Azure provider.
3. In the Azure provider details, select the Availability zones item under Relationships.

4. All discovered availability sets are displayed. Select a discovered availability set to display the details for the set.
5. Click the row that shows the number of instances to list all instances associated with the availability set.

To view sizes (called *flavors* in CloudForms):

1. Open the Clouds tab in the top menu, and select the Providers subtab.
2. Expand the Azure provider.
3. In the Azure provider details, select the Flavors item under Relationships.
4. All discovered sizes are listed. Select a discovered size to display its details.
5. The displayed size information includes:
 - CPU
 - Memory
 - Disk space
 - Number of instances configured using that size
6. Click the row that shows the number of instances to list all instances associated with the size.

To view instances:

1. Open the Clouds tab in the top menu, and select the Providers subtab.
2. Expand the Azure provider.
3. In the Azure provider details, select the Instances item under Relationships.
4. All discovered instances are displayed. Select a discovered instance to display the details for the instance.
5. The information discovered for the instance includes:
 - Name
 - IP address
 - Operating system
 - Power state
 - CPU
 - Memory
 - Disk space
 - Availability set
 - Size (flavor)

Note: Used disk space is not available. This issue is being investigated.

Use case 4: Stopping or starting an Azure instance (power operations)

1. Open the Clouds tab in the top menu, and select the Instances subtab.
2. Select the Azure instance.
3. Click Power, and select Start or Stop.

New Self-Service UI

A new self-service UI has been added so that non-administrative users can easily access their services, track requests, and manage their accounts.

Use case 1: Log in and see the dashboard

The self-service UI is running by default. It does not need to be started or configured.

1. The URL for the self-service UI is:
http://cloudforms-fqdn:3001/self_service/login
The default admin credentials will work in the self-service UI.
2. The dashboard lists a summary of services and requests:
 - a. The total number of services
 - b. Active services
 - c. Services expiring soon
 - d. The total number of requests
 - e. Pending requests
 - f. Denied requests

Use case 2: View the service catalog

1. In the left menu, select Service Catalog.
2. The service items are displayed as tiles on the catalog page. Select the service catalog to view.

Use case 3: Filter services and view the details

1. In the left menu, select My Services.
2. Optionally, set a string in the filter field at the top to filter the services **by name**.
3. Select a service from the results.
4. View the details.

Use case 4: Schedule a service to retire.

1. In the left menu, select My Services.
2. Optionally, set a string in the filter field at the top to filter the services by name.
3. Select a service from the results.
4. Remove the service.
 - a. Click the Remove button to retire the service immediately.
 - b. Click the Retire button to remove it on a schedule, and enter the date to retire it.

Use case 5: Filter and view request details

1. In the left menu, select My Requests.
2. Optionally, set a string in the filter field at the top to filter the requests **by name**.
3. Select a request from the results.
4. View the details.

External Authentication Enhancements

CloudForms has long used LDAP groups for managing users and roles. CloudForms 4.0 extends this and allows external authentication to Active Directory users or to external LDAP users. Additionally, CloudForms now supports two-factor authentication when using FreeIPA 4.1.

Use case 1: Authenticating Active Directory users through the web UI

Using System Security Services Daemon (SSSD) with Active Directory is described in more detail in the [Red Hat Enterprise Linux System-Level Authentication Guide](#).

The first part is configure the local system to use Active Directory for authentication so that local services (Apache and SSSD) can access the Active Directory domain.

1. Optionally, enable Network Manager and discover the domain. Otherwise, enter the Active Directory domain name manually in the configuration.

- a. Enable Network Manager.

```
# sed -i 's/^NM_CONTROLLED=no$/NM_CONTROLLED=yes/'  
/etc/sysconfig/network-scripts/ifcfg-eth0  
# systemctl restart network
```

- b. Discover the realm.

```
# realm discover  
example.com  
type: kerberos  
realm-name: EXAMPLE.COM  
domain-name: example.com  
configured: kerberos-member  
server-software: active-directory  
client-software: sssd  
required-package: oddjob  
required-package: oddjob-mkhomedir  
required-package: sssd  
required-package: adcli  
required-package: samba-common  
login-formats: %U@example.com  
login-policy: allow-realm-logins
```

2. Add the system to the Active Directory realm.

```
# realm join example.com -U user  
Password for user: xxxxxxxx
```

3. Allow Active Directory systems to log into the local system.

```
# realm permit --all
```

4. Configure the local SSSD service to use the Active Directory domain as an identity provider and configure other services like NSS, PAM, and IFP to use the Active Directory Domain.

```
# vi /etc/sss/sss.conf
[domain/example.com]
    ad_domain = example.com
    krb5_realm = EXAMPLE.COM
    realmd_tags = manages-system joined-with-samba
    cache_credentials = True
    id_provider = ad
    krb5_store_password_if_offline = True
    default_shell = /bin/bash
    ldap_id_mapping = True
    use_fully_qualified_names = True
    fallback_homedir = /home/%d/%u
    access_provider = ad
    ldap_user_extra_attrs = mail, givenname, sn, displayname
[sss]
domains = example.com
config_file_version = 2
services = nss, pam, ifp
default_domain_suffix = example.com
```

```
[nss]
    homedir_substring = /home
```

```
[pam]
    default_domain_suffix = example.com
```

```
[ifp]
    default_domain_suffix = example.com
    allowed_uids = apache, root
    user_attributes = +mail, +givenname, +sn, +displayname
```

5. Make sure the Kerberos keytab is readable by the Apache daemon.


```
# chgrp apache /etc/krb5.keytab
# chmod 640 /etc/krb5.keytab
```
6. Create the Apache configuration files.


```
# TEMPLATE_DIR="/var/www/miq/system/TEMPLATE"
# cp ${TEMPLATE_DIR}/etc/pam.d/httpd-auth /etc/pam.d/httpd-auth
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/cfme-remote-user.conf /etc/httpd/conf.d/
# cp ${TEMPLATE_DIR}/etc/httpd/conf.d/cfme-external-auth.conf.erb
  /etc/httpd/conf.d/cfme-external-auth.conf
```
7. Update the Apache external authentication file to specify the Active Directory domain and the generated keytab.


```
<Location /dashboard/kerberos_authenticate>
    AuthType      Kerberos
    AuthName      "Kerberos Login"
```



```
KrbMethodNegotiate On
KrbMethodK5Passwd Off
KrbAuthRealms    example.com
Krb5KeyTab      /etc/krb5.keytab
KrbServiceName Any
Require           pam-account httpd-auth
```

```
ErrorDocument 401 /proxy_pages/invalid_sso_credentials.js
```

```
</Location>
```

8. Set SELinux permissions for the httpd authentication modules.

```
# setsebool -P allow_httpd_mod_auth_pam on
```

```
# setsebool -P httpd_dbus_sss on
```
9. Restart the SSSD and Apache services to reload the configuration.

```
# systemctl restart sssd
```

```
# systemctl restart httpd
```

After configuring the local system, then configure the CloudForms server to use Active Directory for external configuration. **This enables web-only authentication.**

1. Log into the web UI.
2. Open the Configure tab in the top menu, and select the Configuration subtab.
3. Select Authentication.
4. Set the mode to External (httpd). **Do not select enable single sign-on.**
5. For every UI and web service-enabled appliance (everything in the Access Control area).
6. Make sure the LDAP group for the appliance is created and has the right roles assigned.

Use case 2: Authenticating Active Directory users through Kerberos tickets

Configure the local system and CloudForms server exactly as in use case 2, only select the **Enable single sign-on** option. This allows CloudForms to use both passwords submitted through the UI and local Kerberos tickets to be presented for authentication.

Use case 3: Authenticating LDAP users through the web UI (using httpd and SSSD)

Make sure that FreeIPA is properly set up and configured and that SSSD is properly configured. The FreeIPA documentation is available at <http://www.freeipa.org/page/Documentation>.

1. Log into the web UI to configure the authentication method.
 - a. Open the Configure tab in the top menu, and select the Configuration subtab.
 - b. Select Authentication.
 - c. Set the mode to External (httpd). **Do not select enable single sign-on.**
 - d. For every UI and web service-enabled appliance (everything in the Access Control area).
 - e. Make sure the LDAP group for the appliance is created and has the right roles assigned.
2. Log into the appliance console to configure the identity provider.

- a. Run `appliance_console`, and press any key.
- b. In the Advanced Setting menu, select the menu item Configure External Authentication (`httpd`).
- c. Enter the information about the FreeIPA server:
 - The FQDN of the server, such as `ipaserver.example.com`
 - The FreeIPA server domain, such as `example.com`
 - The FreeIPA server Kerberos realm, such as `EXAMPLE.COM`
 - The default Kerberos principal (`admin`) and its password
- d. Enter `y` to proceed.

Use case 4: Using two-factor authentication through FreeIPA 4.1

Set up FreeIPA-based authentication exactly as in use case 3, but then enable two-factor authentication on FreeIPA. CloudForms does not evaluate or configure authentication itself when it is using an external provider, so whatever configuration is enabled on the identity provider is applied to those users when they authenticate to CloudForms.

Enable two-factor authentication on the FreeIPA server.

1. Either globally or per user, change the user authentication types to two factor authentication.
2. When the user logs into FreeIPA, add a time-based one time password token (TOPT) or counter-based token (HOTP) in the OTP Tokens tab.
3. When the token is added, it returns a QR code which can be scanned into FreeOTP or a similar application.
4. Users can synchronize tokens in the main FreeIPA login screen with their username, password, and two OTPs from the FreeOTP application.

Once two-factor authentication is configured in FreeIPA, then it is possible to use two factor authentication in the CloudForms UI.

Other Provider Enhancements

Several smaller enhancements have been added for OpenStack, Amazon, and RHEV providers.

Use case 1: Shelve an OpenStack instance

Shelving an instance is similar to removing it. It suspends it for a given amount of time, and it can then either be restarted or removed.

1. Open the Clouds tab in the top menu, and select the Instances subtab.
2. Select the OpenStack instance.
3. Click Power Operations, and select Shelve.
4. Set the time to shelve the instance, and save.

Use case 2: Create M4 and t2.large instance types for Amazon

Previously, CloudForms supported M3 instance types for Amazon. This release adds support for M4 and t2.large. Setting up images for CloudForms and Amazon is covered in <https://access.redhat.com/articles/1482553>.

Use case 3: Reconfigure the hardware for a RHEV instance

The ability to reconfigure the hardware for an OpenStack-hosted instance is already available in CloudForms. CloudForms 4.0 adds the ability to reconfigure the hardware for RHEV instances as well.

1. Open the Infrastructure tab in the top menu, and select the VMs subtab.
2. Select the RHEV instance.
3. In the Configuration drop-down menu, select Reconfigure this VM.
4. In the edit window, enter the settings for:
 - a. Memory
 - b. Processors (sockets and cores per socket)
5. Save. This returns the reconfiguration request, which is then processed as normal.

Use case 4: Reconfigure an Orchestration service catalog item

1. Open the Services tab in the top menu, and select the Service subtab.
2. Select the Orchestration service item.
3. In the Configuration drop-down menu, select Reconfigure this service.

General Improvements for OpenStack:

5. Improved naming for AMQP binding queues
6. Improved neutron performance
7. Simplified pagination
8. Able to delete unused RabbitMQ queues
9. Added support for Keystone v3

Automate Enhancements

Multiple State Machines

An automate state machine defines a workflow, such as integrating with IPAM or installing a database. Each state machine defines a specific workflow – this creates a modular environment where state machines can be reused in multiple automate instances to achieve different things without having to be duplicated.

State machines must exist within a single automate workspace to be available. Variables in the workspace are accessible to all state machines.

One state machine can call another state machine. Any state machine in the hierarchy can initiate an act, call another machine, skip an action, or retry an action. Any state machine can return an error that aborts a process or to continue to another state machine.

A design presentation on multiple state machines is available at <http://www.slideshare.net/ManageIQ/manage-iq-summit-john-hardy-state-machines>.

New Automate Methods

New automate methods have been added for Azure provider support, multi-tenancy, and container management.

REST API Changes

The full API reference is available in the ManageIQ upstream documentation at https://github.com/ManageIQ/manageiq_docs/tree/master/api/reference.

Many REST calls have been added in this release:

- Query services and service template images, including template image hrefs
 - GET /api/service_templates/:id?attributes=picture.image_href
 - GET /api/pictures
 - GET /api/pictures?expand=resources&attributes=image_href
 - GET /api/services?expand=resources&attributes=picture.image_href
 - GET /api/service_requests?expand=resources&attributes=picture.image_href
 - GET /api/service_templates?expand=resources&attributes=picture.image_href
- Query resource actions as a formal subcollection of service_templates
 - GET /api/service_templates/:id/resource_actions
 - GET /api/service_templates/:id/resource_actions?expand=resources&filter[]="action='Provision'"
- Query service dialogs
 - GET /api/service_dialogs
 - GET /api/service_templates/:id/service_dialogs
 - GET /api/services/:id/service_dialogs
- Query provisioning dialogs
 - GET /api/provision_dialogs
- Import reports, including multiple reports in a single request
 - POST /api/reports

```
{
  "action" : "import",
  "resource" : {
    "report" : {
      "menu_name" : "Test Report",
      "rpt_type" : "Custom",
      ...
    },
    "options" : { "save" : true }
  }
}
```

- }
- Create, read, update, and delete for user roles
 - /api/roles (create via POST)
 - /api/roles/:id (update through POST)
 - /api/roles/:id (delete through DELETE or POST)
- Query user roles and product features
 - /api/roles
 - /api/roles/:id
 - /api/roles/:id/features
 - /api/roles/:id/features/:feature_id
 - /api/features
 - /api/features/:id
- Querying chargeback rates
 - /api/chargebacks
 - /api/rates
 - /api/chargebacks/:id/rates
- Create, read, update, and delete for chargeback rates
 - /api/rates (create via POST)
 - /api/rates/:id (update through POST, PUT, and PATCH)
 - /api/rates/:id (delete through DELETE or POST)
- Run a report
 - POST /api/reports/:id
 - {
 - "action" : "run"
 - }
- Query reports results
 - /api/reports/:id/results
 - /api/reports/:id/results/:r_id
 - /api/results/:r_id
- Support for custom action buttons and dialogs
 - GET
/api/service_templates/:id?attributes=custom_actions,custom_action_buttons
 - GET /api/services/:id?attribtues=custom_actions,custom_action_buttons
- Categories and tags CRUD
 - POST /api/categories (actions: create, edit, delete)
 - POST /api/categories/:id (actions: edit, delete)
 - DELETE /api/categories/:id
 - POST /api/tags (actions: create, edit, delete)
 - POST /api/tags/:id (actions: edit, delete)
 - DELETE /api/tags/:id
- Support password updates
 - POST /api/users/:id
 - {
 - "action" : "edit",

```
“resource” : { “password” : “<new_password>” }  
}
```

Deprecations and Removals

- The SOAP API was deprecated in Red Hat CloudForms 3.2 and has been removed from the CloudForms 4.0 Beta.
- The option to filter conditions based on Ruby scripts has been removed from the Control page in the UI area and from reports.