

Using and Managing vRealize Automation Service Broker

11 MARCH 2020

vRealize Automation 8.0

You can find the most up-to-date technical documentation on the VMware website at:

<https://docs.vmware.com/>

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

Copyright © 2022 VMware, Inc. All rights reserved. [Copyright and trademark information.](#)

Contents

1	What is vRealize Automation Service Broker	4
	How does vRealize Automation Service Broker work	5
2	What are the vRealize Automation Service Broker user roles	7
3	Setting up vRealize Automation Service Broker for your organization	9
	Adding Content to the Catalog	9
	Add vRealize Automation Cloud Assembly blueprints to the catalog	9
	Add CloudFormation templates to the catalog	12
	Add vRealize Orchestrator workflows to the catalog	15
	Add extensibility actions to the catalog	17
	Setting up policies	19
	How do I configure deployment leases using policies	20
	How do I configure day 2 actions using policies	24
	How do I configure deployment criteria in policies	28
	How are policies processed	30
	Customize an icon and request form	33
	Learn more about vRealize Automation Service Broker custom forms	36
	Add an email server to send notifications	49
	Working with the Infrastructure options	50
4	How do I deploy a catalog item	51
5	How do I manage my deployments	53
	Monitoring deployments	55
	What can I do if a vRealize Automation Service Broker deployment fails	57
	What actions can I run on deployments	58

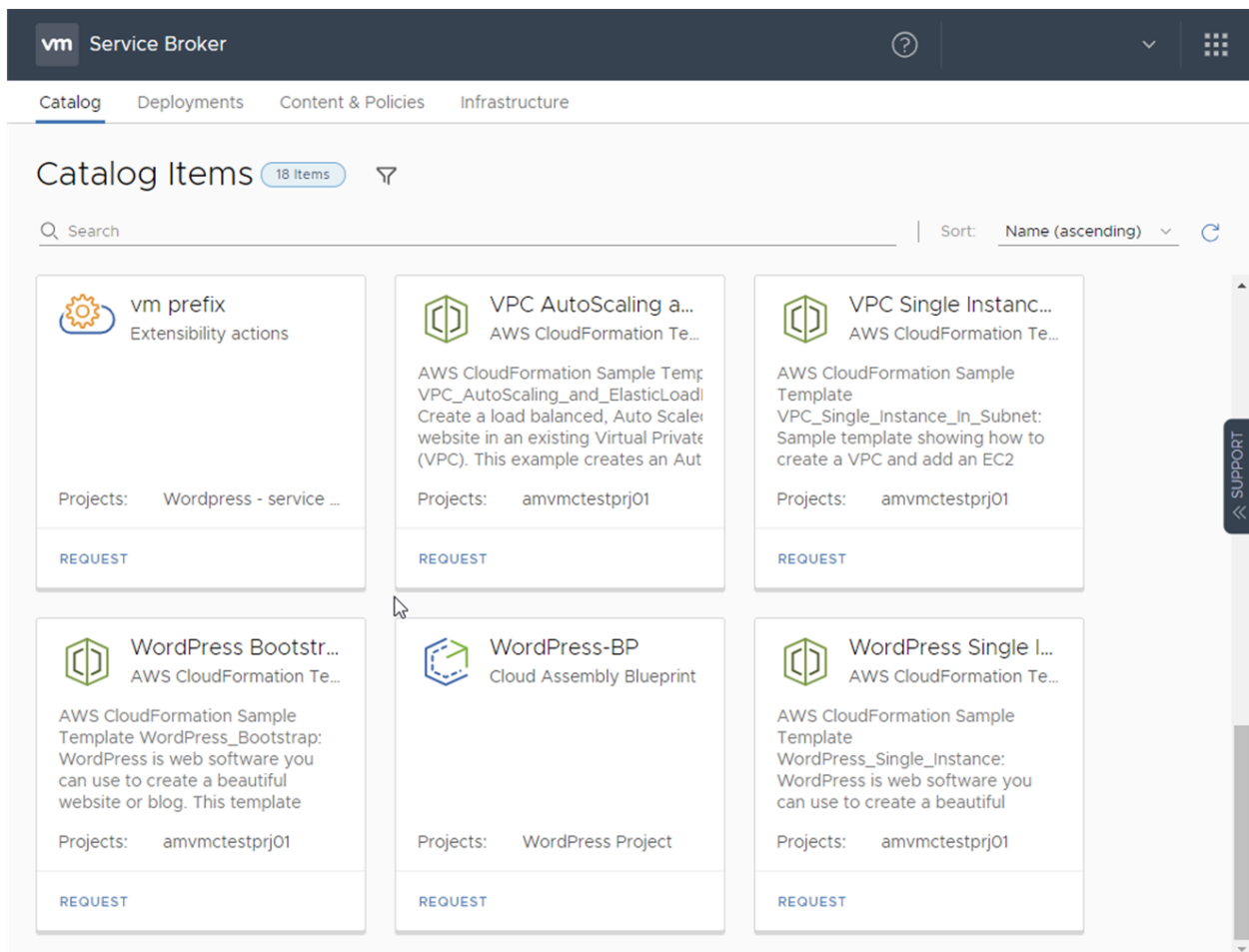
What is vRealize Automation Service Broker

1

The vRealize Automation Service Broker provides a single point where you can request and manage catalog items.

As a cloud administrator, you create catalog items by importing released vRealize Automation Cloud Assembly blueprints and Amazon Web Services CloudFormation templates that your users can deploy to your cloud vendor regions or datastores.

As a user, you can request and monitor the provisioning process. After deployment, you manage the deployed catalog items throughout the deployment lifecycle.



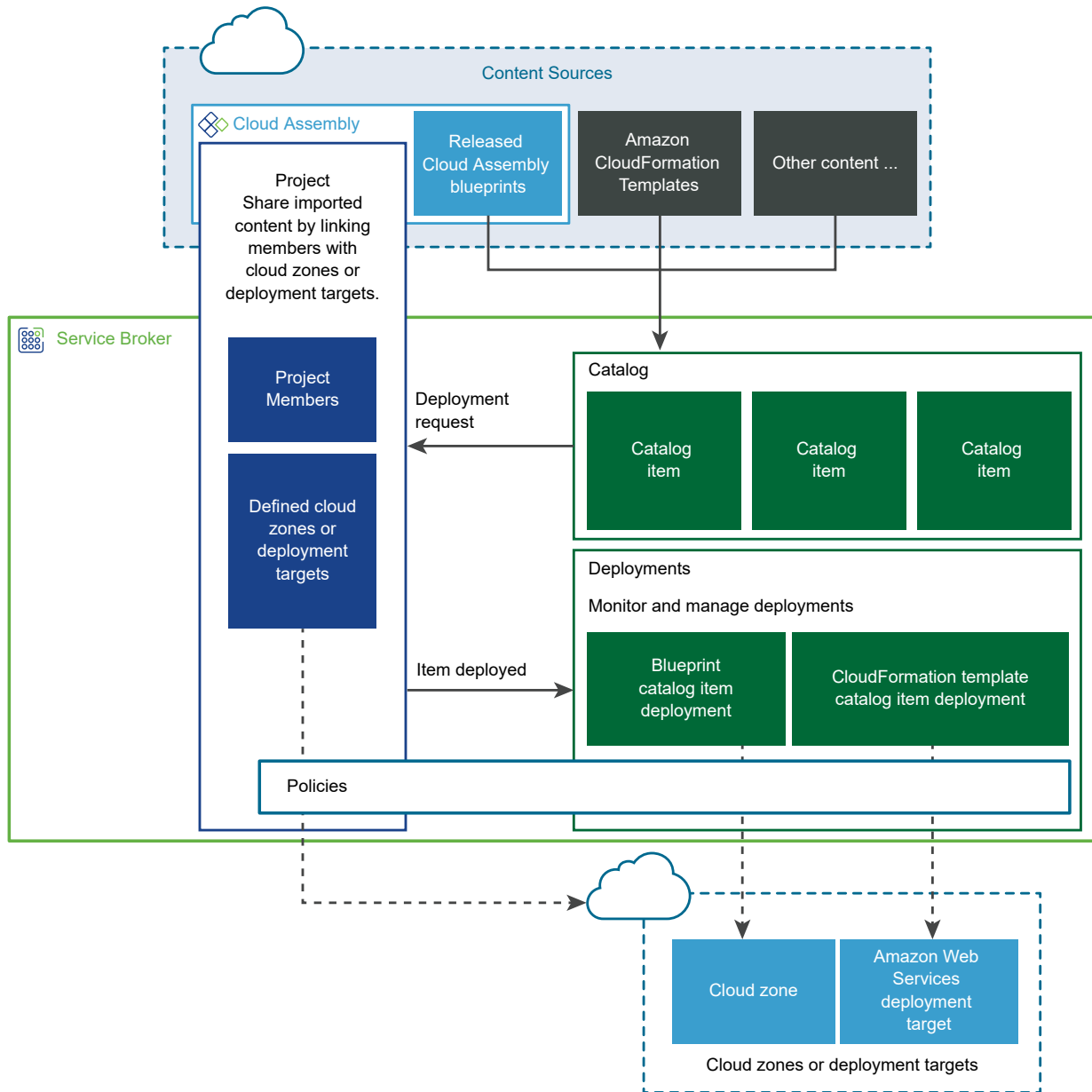
This chapter includes the following topics:

■ How does vRealize Automation Service Broker work

How does vRealize Automation Service Broker work

The vRealize Automation Service Broker is the simplified user interface that cloud administrators make available to users when the administrator's teams do not need full access to developing and building and the blueprints or templates.

You use vRealize Automation Service Broker to deploy blueprints and templates to cloud regions or datastores associated with projects.



To provide the blueprints and templates, the cloud administrator configures content sources. The content sources can include vRealize Automation Cloud Assembly blueprints and Amazon CloudFormation templates. The imported blueprints and templates become catalog items.

- The content sources are entitled to projects. Projects link a set of users with one or more target cloud zone regions or datastores.
- For example, UserA is a member of ProjectA and ProjectB, but not ProjectC. She sees only the imported blueprints or templates that were entitled to ProjectA and ProjectB.

When users requests a catalog item, where it deployed depends on the project selected. Projects might have one or more cloud zones.

- If UserA and UserB are members of ProjectA, they see the imported blueprints and templates as catalog items. And at deployment time they can deploy to ProjectA, which determines which cloud regions or datastores the catalog item is deployed to.

The availability of the catalog items is determined by project membership. Projects link users, catalog items, and cloud resources where the items are deployed.

After a successful request, your users can then manage their deployments by running actions, including dismiss or delete.

What are the vRealize Automation Service Broker user roles

2

Your user role in vRealize Automation Service Broker determines what you can see and do. Some roles are defined at the service organization level, and some are specific to vRealize Automation Cloud Assembly.

User Roles

User roles are defined for the organization in the vRealize Automation console. There are two types of roles, organization roles and service roles.

The organization roles are global and apply to all services in the organization. A user is assigned an Organization owner or Organization Member role.

For more information about the organization roles, see [Administering vRealize Automation](#).

The vRealize Automation Service Broker service roles, which are service-specific permissions, are also assigned at the organization level in the console.

Table 2-1. Service Roles

Role	Description
Service Broker Administrator	Must have read and write access to the entire user interface and API resources. This is the only user role that can perform all tasks, including creating a new project and assigning a project administrator.
Service Broker User	Any user who does not have the vRealize Automation Service Broker Administrator role. In a vRealize Automation Service Broker project, the administrator adds users to projects as project members. The administrator can also add a project administrator. The permission for these two roles are defined below.

Project roles and permissions

If you do not have a vRealize Automation Service Broker administrator role, you must be a member of a project to view the catalog and deploy items in your project.

Table 2-2. Project Administrator and Project Member Roles

Task	vRealize Automation Service Broker Administrator	Project Administrator	Project Member
Request catalog items in my project	Yes	Yes	Yes
Create projects	Yes	No	No
Update project name and description	Yes	Yes	No
Add users to my project	Yes	Yes	No
View provisioned deployments	Yes	Yes For all project members.	Yes For just your deployments, unless the project deployments are share with all project members.
Run deployment actions	Yes	Yes For all project members.	Yes For just your deployments, unless projects deployments are shared with all project members and you're entitled to run the day 2 actions.
Manage content sources	Yes	No	No
Share content	Yes	No	No
Customize request forms	Yes	No	No
Create policies	Yes	No	No
Add cloud zones	Yes	No	No
Add cloud accounts	Yes	No	No
Add integrations	Yes	No	No
Add cloud proxies	Yes	No	No

Setting up vRealize Automation Service Broker for your organization

3

To fully configure vRealize Automation Service Broker, you need to determine your catalog sources and apply governance using projects. As a cloud administrator, you can also apply policies and customize the catalog request form.

As a cloud administrator, you can also apply policies and customize the catalog request form.

This chapter includes the following topics:

- [Adding Content to the vRealize Automation Service Broker Catalog](#)
- [Setting up vRealize Automation Service Broker policies](#)
- [Customize a vRealize Automation Service Broker icon and request form](#)
- [Add an email server in vRealize Automation Service Broker to send notifications](#)
- [Working with the Infrastructure options in vRealize Automation Service Broker](#)

Adding Content to the vRealize Automation Service Broker Catalog

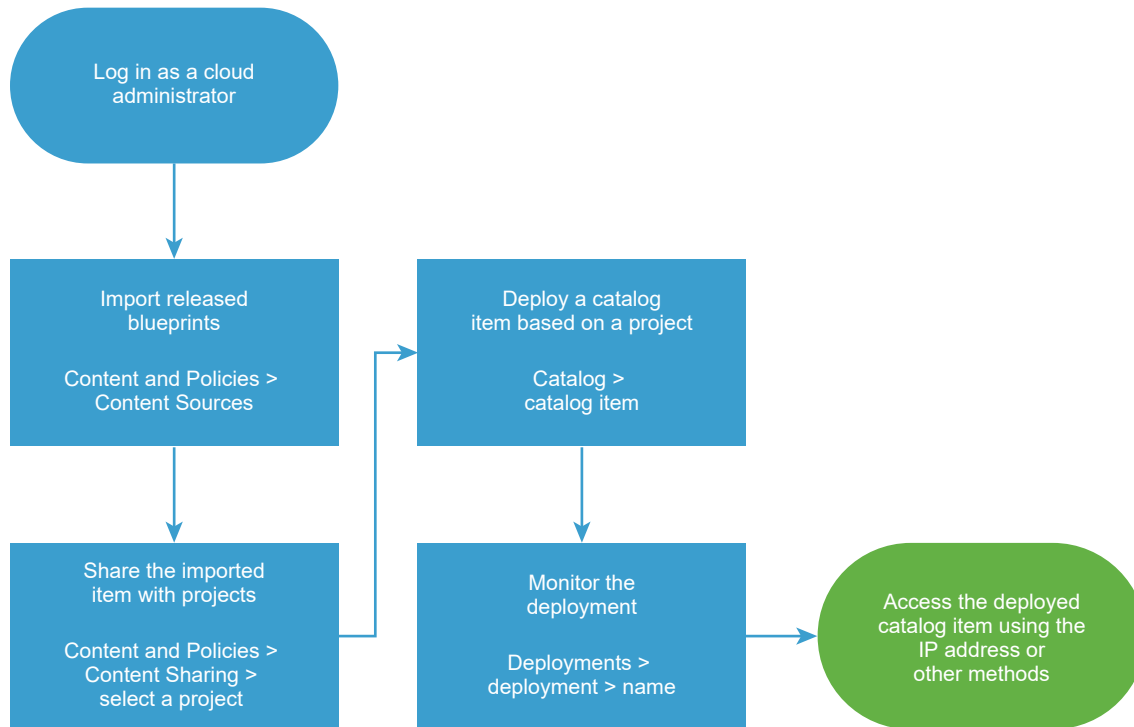
The requirements and process for setting up your vRealize Automation Service Broker catalog depends on the content that you are providing to your users.

Each process is provided as an end-to-end procedure. Identify the content that you are providing and add each relevant type. Ensure that the imported content is working properly outside of vRealize Automation Service Broker before you add it to the catalog.

After you add the content sources, the templates are refreshed every six hours. Any changes to the templates in your external sources are reflected in the catalog after a refresh.

Add vRealize Automation Cloud Assembly blueprints to the vRealize Automation Service Broker catalog

As a cloud administrator, you can make vRealize Automation Cloud Assembly blueprints available in the vRealize Automation Service Broker catalog by adding a vRealize Automation Cloud Assembly content source and sharing the blueprints. The blueprints are the specifications for services or applications that you can deploy to your cloud providers.



After you import the blueprints, you share them with project members so that they can deploy the blueprints. At the request time, the blueprint is deployed to cloud zone account region or datastore that supports the blueprint requirements.

Prerequisites

- Verify that the blueprints that you are importing are deployable and released in vRealize Automation Cloud Assembly before you import it. See [How to save different versions of a blueprint](#) in *Using and Managing vRealize Automation Cloud Assembly*.

Procedure

- 1 Import blueprints from vRealize Automation Cloud Assembly.

- a Select **Content and Policies > Content Sources**.
- b Click **New**, and then click **Cloud Assembly Blueprint**.
- c Enter the **Name** for this content source.
- d Select the **Source project** and then click **Validate**.

The validation process tests the connection and provides the number of released blueprints that are associated with the project in vRealize Automation Cloud Assembly.

- e Click **Create and Import**.

The Content Sources page lists your new source and the number of discovered and imported items.

2 Share the imported items with a project.

- a Select **Content and Policies > Content Sharing**.
- b Select the project that includes the users who should be able to deploy the blueprints.
- c Click **Add Items** and then select one or more blueprints to share with the project.

You can select all the items imported from a content sources or you can expand the source trees and select individual items.

- d Click **Save**.

The Content Sharing page lists all the items entitled to the selected project. The blueprints are also added to the catalog where the project members can request them.

3 Verify that the blueprint is available in the catalog to the members of the selected projects.

- a Click **Catalog**, locate the imported blueprint, and review the projects to ensure that the project you configured is included.
- b Click **Request** and provide any required information.

If the blueprint has more than one released version, select the version that you want to deploy.

- c Click **Submit**.

The provisioning process begins and the Deployments tab opens with your current request at the top.

4 Monitor the provisioning process to ensure successful deployment.

- a Click **Deployments** and locate your deployed catalog item.
- b Monitor the card status until it is successful.

Results

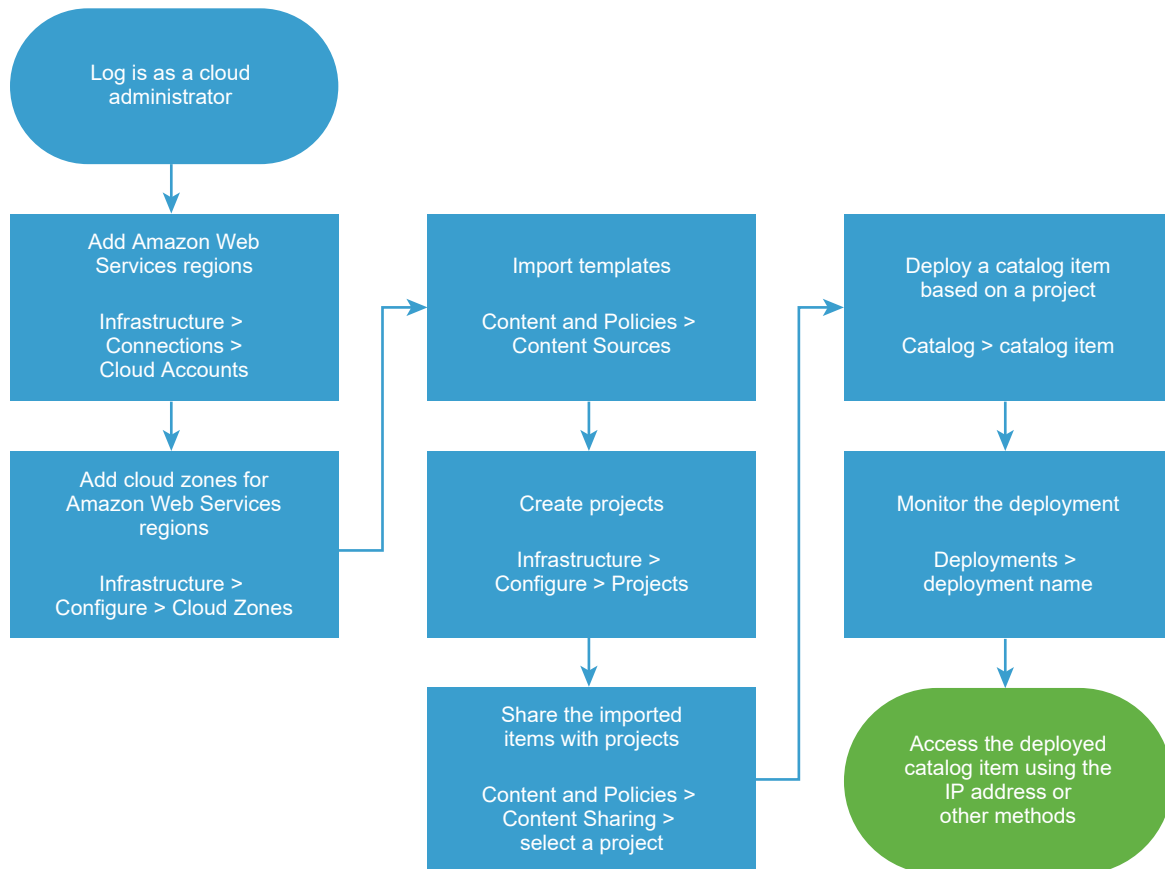
The released blueprints are imported into vRealize Automation Service Broker, shared in the catalog, and deployable.

What to do next

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if a vRealize Automation Service Broker deployment fails](#). If you are a vRealize Automation Cloud Assembly cloud administrator, you can also do more extensive troubleshooting in vRealize Automation Cloud Assembly [What can I do if a Cloud Assembly deployment fails](#) in *Using and Managing VMware Cloud Assembly*.
- If you want to control how long a deployment can exist, create a lease. See [Setting up vRealize Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize a vRealize Automation Service Broker icon and request form](#).

Add CloudFormation templates to the vRealize Automation Service Broker catalog

As a cloud administrator, you can populate the vRealize Automation Service Broker catalog with Amazon CloudFormation templates by adding one or more Amazon S3 buckets as content sources and sharing them with project members. The templates are the specifications for the services or applications that you can deploy to Amazon Web Services.



You can only add one bucket as a content source. To add multiple buckets, you create a content source for each bucket.

After you add the templates, you entitle project members to deploy the blueprints. At the request time, the blueprint is deployed to the cloud account region that you define when you add the content source.

Prerequisites

- Ensure that you know the name of the S3 bucket that contains your CloudFormation templates.
- If you are adding a private bucket, you must know the access key and the secret key.

Procedure

- 1 To deploy your CloudFormation templates, you must have at least one Amazon Web Services cloud account and select the regions.
 - a Select **Infrastructure > Connections > Cloud Accounts**.
 - b Click **Add Cloud Account** and then click **Amazon Web Services**.
 - c Enter the 20-digit **Access Key ID** and corresponding **Secret Access Key**.
 - d To verify the credentials, click **Validate**.
 - e Enter an account name.
Provide a name that you can identify when you share templates with projects.
 - f Select one or more regions in this account that you want to deploy templates to.
 - g Click **Create**.
- 2 Define cloud zones for the Amazon Web Services cloud account regions.
 - a Select **Infrastructure > Configure > Cloud Zones**, and then click **New Cloud Zone**.
 - b Select the **Account/region**, the **Name**, and the **Placement policy**.
 - c Click the **Compute** tab and verify or modify the resources that are included in the cloud zone.
 - d Click **Create**.
- 3 Import the templates.
 - a Select **Content and Policies > Content Sources**.
 - b Click **New**, and then click **AWS CloudFormation Template**.
 - c Enter the **Name** for this content source.
 - d Add the S3 bucket information.
 - e Click **Validate**.
If the bucket is public, the validation process verifies the name and the number of templates. If the bucket is private, the validation process verifies the name, the keys, and the number of templates.
 - f Select the **Deployment Target** Amazon Web Services cloud account and a region.
 - g Click **Create and Import**.
- 4 Add a project so that you can share the templates with project members.
 - a In vRealize Automation Service Broker, select **Infrastructure > Configure > Projects**, and then click **New Project**.
 - b Enter the project information on the **Summary** tab.

- c Click the **Users** tab and then click **Add Users**.

To add project users, the individuals or the groups must already be active service organization users.

- d If this project supports only CloudFormation templates, ignore the Provisioning tab.

CloudFormation templates are deployed to the target account and region that you defined when you imported the templates. If the project members can deploy other blueprints, templates, or content, you must add the target cloud zones for the content to the project.

- e Click **Create**.

The new project is added to your projects. It is also added to your associated vRealize Automation Cloud Assembly instance. If the project is for blueprints, you can add cloud zones in vRealize Automation Cloud Assembly. If the project is for templates, you do not need to add cloud zones.

5 Share the imported templates with a project.

- a Select **Content and Policies > Content Sharing**.
- b Select the project that includes the users who should be able to deploy the templates.
- c Select one or more Amazon Web Services content sources to share with the project.
- d Click **Save**.

Content Sharing page lists all the items entitled to the selected project. The templates are also added to the catalog where the project members can request them.

6 Verify that the template is available in the catalog to the members of the selected projects.

- a Click **Catalog**, locate the imported CloudFormation templates, and review the projects to ensure that the project you configured is included.
- b Click **Request** and provide any required information.
- c Click **Submit**.

The provisioning process begins and the Deployments tab opens with your current request at the top.

7 Monitor the provisioning process to ensure successful deployment.

- a Click **Deployments** and locate your deployed catalog item.
- b Monitor the card status until it is successful.

Results

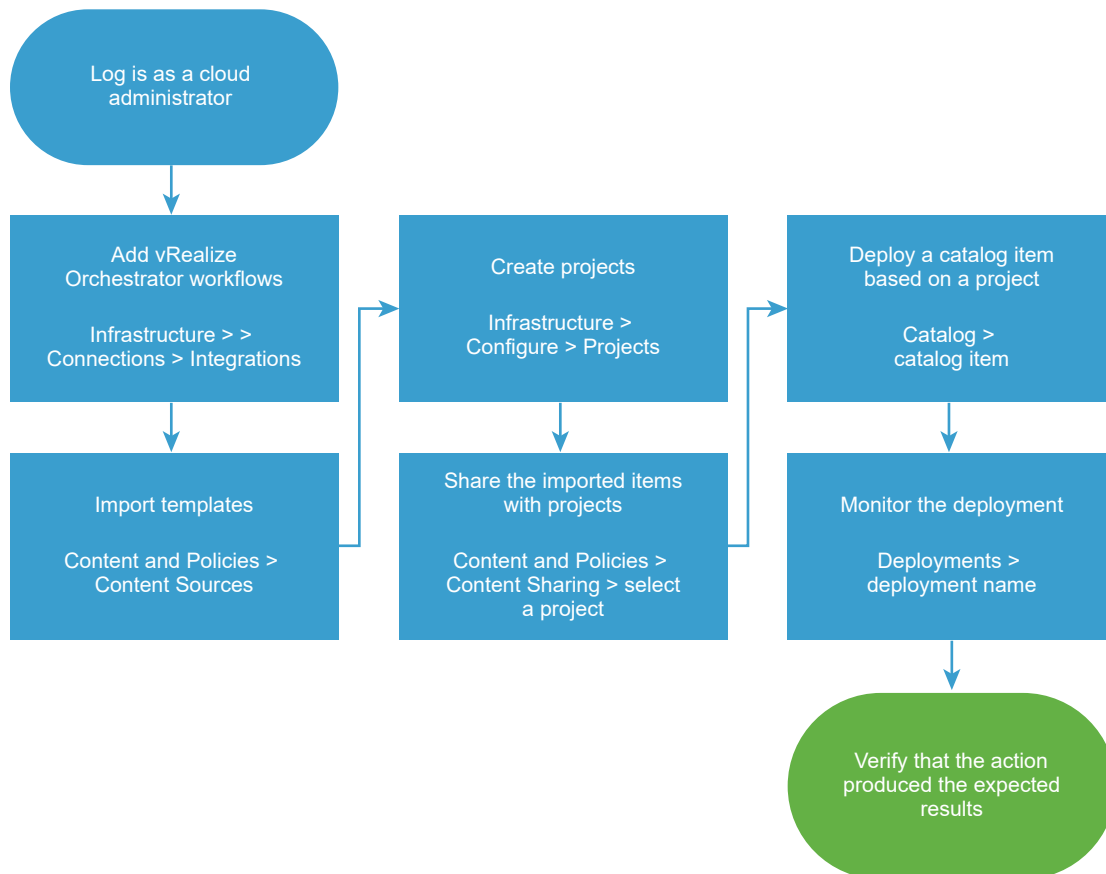
The templates are imported into vRealize Automation Service Broker and shared in the catalog.

What to do next

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if a vRealize Automation Service Broker deployment fails](#). If you are a vRealize Automation Cloud Assembly cloud administrator, you can also do more extensive troubleshooting in vRealize Automation Cloud Assembly [What can I do if a Cloud Assembly deployment fails](#) in *Using and Managing VMware Cloud Assembly*.
- If you want to control how long a deployment can exist, create a lease. See [Setting up vRealize Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize a vRealize Automation Service Broker icon and request form](#).

Add vRealize Orchestrator workflows to the vRealize Automation Service Broker catalog

As a cloud administrator, you can add vRealize Orchestrator workflows to the catalog. The workflows are created in vRealize Orchestrator to accomplish a simple or complex task.



Prerequisites

- Verify that you have vRealize Orchestrator workflows that can perform required tasks. See [Managing Workflows](#).

Procedure

- 1 If you do not have a vRealize Orchestrator integration configured in vRealize Automation Cloud Assembly, you can add the integration in vRealize Automation Service Broker.
 - a Select **Infrastructure > Connections > Integrations**.
 - b Click **Add Integration** and then click **vRealize Orchestrator**.
 - c Enter the URL for your vRealize Orchestrator instance.
 - d Select the **Cloud Proxy**.
 - e Enter a user name and password.
 - f To validate the credentials and URL, click **Validate**.
 - g Enter a name that identifies this instance when you create the content source.
 - h Click **Add**.
- 2 Import the workflow.
 - a Select **Content and Policies > Content Sources**.
 - b Click **New**, and then click **vRealize Orchestrator Workflow**.
 - c Enter the **Name** for this content source.
 - d Click **Add** and select the workflows that you want to make available in vRealize Automation Service Broker.
 - e Click **Create and Import**.
- 3 Share the imported workflow with a project.
 - a Select **Content and Policies > Content Sharing**.
 - b Select the project that includes the users who should be able to deploy the workflows.
 - c Click **Add Items** and then select one or more workflows to share with the project members.

You can select all the items imported from a content source or you can expand the source trees and select individual items.
 - d Click **Save**.
- 4 Verify that the workflow is available in the catalog to members of the selected project.
 - a Click **Catalog**, locate the imported workflow, and review the projects to ensure that the project you configured is included.
 - b Click **Request** and provide any required information.
 - c Click **Submit**.

The provisioning process begins and the Deployments tab opens with your current request at the top.

- 5 Monitor the provisioning process to ensure that the workflow runs successfully
 - a Click **Deployments** and locate your deployed request.
 - b Monitor the card status until it is successful.

Results

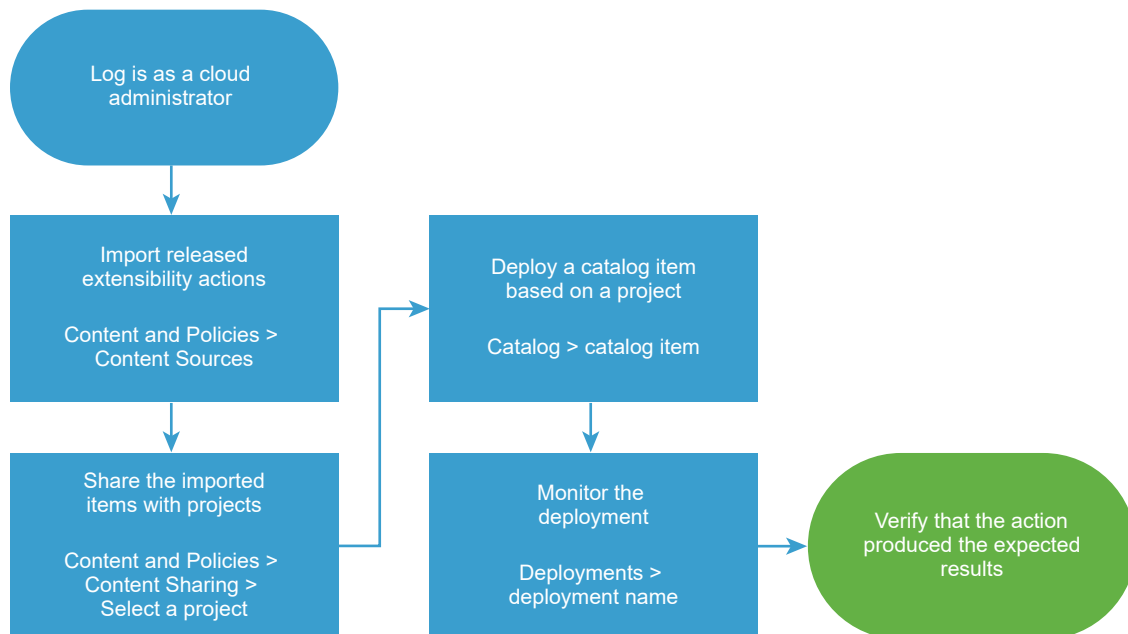
The vRealize Orchestrator workflows are imported into vRealize Automation Service Broker and shared in the catalog.

What to do next

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if a vRealize Automation Service Broker deployment fails](#). If you are a vRealize Automation Cloud Assembly cloud administrator, you can also do more extensive troubleshooting in vRealize Automation Cloud Assembly [What can I do if a Cloud Assembly deployment fails](#) in *Using and Managing VMware Cloud Assembly*.
- If you want to control how long a deployment can exist, create a lease. See [Setting up vRealize Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize a vRealize Automation Service Broker icon and request form](#).

Add extensibility actions to the vRealize Automation Service Broker catalog

As a cloud administrator, you can add vRealize Automation Cloud Assembly extensibility actions to vRealize Automation Service Broker as a content source. The extensibility actions are created and managed in vRealize Automation Cloud Assembly.



The actions are small scripts that perform lightweight tasks or steps. For example, rename a virtual machine or assign an IP address.

Prerequisites

- Verify that the actions you are adding are associated with a project, and that they are released. See [How do I create extensibility actions](#).

Procedure

1 Import the released extensibility actions.

- a Select **Content and Policies > Content Sources**, and click **New**.
- b Click **New**, and then click **Extensibility actions**.
- c Enter the **Name** for this content source.
- d Select the **Source project** and then click **Validate**.

The validation process verifies the number of released extensibility actions that are associated with the project in vRealize Automation Cloud Assembly.

- e Click **Create and Import**.

2 Share the imported actions with a project.

- a Select **Content and Policies > Content Sharing**.
- b Select the project that includes the users who should be able to deploy the extensibility actions.
- c Click **Add Items** and then select one or more actions to share with the project.

You can select all the items imported from a content source or you can expand the source trees and select individual items.

- d Click **Save**.

Content Sharing page lists all the items entitled to the selected project. The actions are also added to the catalog where the project members can request them.

3 Verify that the action is available in the catalog to the members of the selected projects.

- a Click **Catalog**, locate the imported extensibility action, and review the projects to ensure that the project you configured is included.
- b Click **Request** and provide any required information.
- c Click **Submit**.

The provisioning process begins and the Deployments tab opens with your current request at the top.

- 4 Monitor the provisioning process to ensure that the action runs successfully.
 - a Click **Deployments** and locate your deployed request.
 - b Monitor the card status until it is successful.

Results

The extensibility actions are imported into vRealize Automation Service Broker and shared in the catalog.

What to do next

- If the deployment fails, click the deployment name and begin troubleshooting. See [What can I do if a vRealize Automation Service Broker deployment fails](#). If you are a vRealize Automation Cloud Assembly cloud administrator, you can also do more extensive troubleshooting in vRealize Automation Cloud Assembly [What can I do if a Cloud Assembly deployment fails](#) in *Using and Managing VMware Cloud Assembly*.
- If you want to control how long a deployment can exist, create a lease. See [Setting up vRealize Automation Service Broker policies](#).
- To provide more or fewer user inputs at request time, you can create a custom form. See [Customize a vRealize Automation Service Broker icon and request form](#).

Setting up vRealize Automation Service Broker policies

To provide the background management of your deployments, you set up policies. Each vRealize Automation Service Broker policy is a set of rules or parameters that are applied to deployments, freeing the cloud administrator for other tasks.

Any policies that you create in vRealize Automation Service Broker are applied to the deployments in vRealize Automation Service Broker and in vRealize Automation Cloud Assembly.

Getting started with policies

To begin creating policies, select **Content and Policies > Policies > Definitions**. Any policy that you add is applied to current deployments and any new deployments.

vRealize Automation Service Broker includes the following policy types.

- Lease

You use lease policies to control how long deployments are available to users. For example, you create a policy where all deployments are destroyed after 30 days unless the user extends the lease.
- Day 2 actions

You use day 2 actions policies to control what changes users can make to deployments. For example, you create a policy that allows users to power off and power on their deployments.

To get you started, use the full use cases that are provided. The use cases guide you through the process of creating more than one policy. The use case provides contextual explanations of the choices and the desired behavior.

The use cases are followed by more in-depth information about how multiple policies are processed.

How do I configure vRealize Automation Service Broker deployment leases using policies

By using policy-based leases, you reduce the need to intervene manually to reclaim resources. You define lease policies so that you can control the amount of time that a deployment is available to your users. The lease policy use cases in this procedure provide a beginning point for learning about and implementing policies for your organization.

If you do not have any lease policies defined, then the deployments never expire. To reclaim the resources, you must manually destroy the deployments.

When does a lease policy go into effect?

- If the policy scope is Organization, then all the deployments in your organization are managed based on the defined policies.
- If the policy scope is a project, then the deployments that are associated with that project are managed based on the defined lease. Other projects are not affected.

Lease policies are applied when you:

- Create or update a lease policy. After lease policies are applied, they continuously evaluate the deployments in the background to ensure that they are in compliance with the defined leases.
- Request a catalog item in vRealize Automation Service Broker or a blueprint in vRealize Automation Cloud Assembly, and it is successfully deployed.
- Onboard workloads or resources in vRealize Automation Cloud Assembly so that you can manage them using vRealize Automation Service Broker, vRealize Automation Cloud Assembly, or vRealize Automation Code Stream.

In this use case, there are three policy definitions that illustrate how you can construct policies and the results when they are enforced. The last policy is not enforced, but the reasons are provided in the scenario results.

As you review the lease policies use case, you must also configure lease-specific options. The following descriptions provide a brief summary. Consult the signpost help for more information.

- Lease (days). The maximum number of days that the deployment resources are available before they are destroyed.
- Total lease (days). The total number of days before the deployment is destroyed and the resources are reclaimed.
- Grace period (days). The number of days the user has to renew the lease before the deployment is destroyed.

Procedure

1 Select **Content and Policies > Policies > Definitions > New Policy > Lease Policy**.

2 Configure Lease Policy 1.

As an administrator, you want to control costs by limiting the starting lease time for all deployments to 30 days, with the option to renew the lease for a total of 90 days.

a Define when the policy is valid.

Setting	Sample Value
Scope	Organization This policy is applied to everyone in your organization.
Deployment criteria	None
Enforcement type	Soft This enforcement type allows you to create other policies related to this lease that override this policy.

b Define the lease.

Setting	Sample Value
Lease	30
Total lease	90
Grace period	10

In this scenario, the deployment is shut down after 30 days and an email is sent to the user. If the user does not extend the lease, the deployment is destroyed after 10 days. If the user extends the lease for another 30 days, and then another 30, for a combined total of 90 days, the maximum lease time is reached and the deployment is shut down. It is destroyed 10 days later.

3 Configure Lease Policy 2.

As an administrator, you want to control costs by limiting the lease time on a expensive blueprint or template to two weeks. For this example, the blueprint name is `Multi-tier 5 machine with LB`.

a Define when the policy is valid.

Setting	Sample Value
Scope	Project MT5 This policy applied to deployments associated with this project.
Deployment criteria	BlueprintId eq Multi-tier 5 machine with LB Based on this criteria expression, only deployments for the referenced blueprint are considered for policy enforcement.
Enforcement type	Soft This soft enforcement still overrides the organization policy of 90 days in Policy 1 because the values are more meaningful at the project level.

b Define the lease policy.

Setting	Sample Value
Lease	14
Total lease	28
Grace period	3

In this scenario, both policies are applied, but Policy 2 takes precedence over Policy 1 because it is more specific. When applied, the deployment is shut down after 14 days. If the user does not extend the lease, it is destroyed three days later. If the user extends the lease for up to another 14 days, the deployment is shut down at the end of the second extension and it is destroyed three days later.

4 Review the configuration of Lease Policy 3.

As a project manager, you realize that one of your developers is working on a complex application. The developer requires the `Multi-tier 5 Machines with LB` blueprint and another blueprint, `Distributed Database Across Clouds`, but for a longer lease than defined in Policy 2.

Unless you understand how the policies are processed based on how they are defined, you might encounter unexpected results. Policy 3 is an example of how processing and precedence affect the result.

This policy, as provided, will not be enforced. This example provides an opportunity for you to see how leases are applied and enforced when there is more than one that applies.

- a Define when the policy is valid.

Setting	Sample Value
Scope	Project MT5 This policy is applied to deployments in this project.
Deployment criteria	(BlueprintId eq Multi-tier five machine with LB OR CatalogId eq Distributed Database Across Clouds) AND CreatedBy eq jan@mycompany.com. You use catalogID because it is a non- vRealize Automation Cloud Assembly template.
Enforcement type	Soft This soft enforcement still overrides the organization policy of 90 days in policy 1 because the values are more meaningful at the project level.

- b Define the lease policy.

Setting	Sample Value
Lease	21
Total lease	50
Grace period	3

In this scenario, Lease Policy 2 is applied, not Lease Policy 3.

- Lease 3 has a lease time that is less than or equal to 21 days, and the policy is applied. Lease 2 has a lease time that is less than or equal to 14 days, and the policy is applied.
 - Lease 2 is applicable and it does not violate the lease 3 policy. But, lease 2 is more restrictive, so it takes precedence. Lease policy 2 is more restrictive because it is for a shorter period of time.
 - When both lease definitions are true and applicable, the more restrictive policy is the one that is enforced.
- 5 To resolve the unexpected behavior in Lease Policy 3, you can implement one of the following solutions.
- To ensure that you can provide Jan with the needed policy, change the enforcement type to hard.
 - Alternatively, you could create a new project with access to the same resources, and then create Lease Policy 3 for that project. While this solution isolates the working policy, you must maintain a parallel project. The effort needed to set up and maintain the content sources, content sharing, and so on, are time consuming and subject to error.

What to do next

- For more examples of how the lease policies are processed and enforced, see [How are vRealize Automation Service Broker policies processed](#).
- Configure policies that are relevant to your organizations and projects. If you are just getting started with lease policies, begin with one lease policy at the organization level.
- To send an email to the deploying user, configure the email server for notifications. See [Add an email server in vRealize Automation Service Broker to send notifications](#).

How do I entitle deployment users to vRealize Automation Service Broker day 2 actions using policies

You define day 2 action policies so that you can control what changes your users can make to deployments and their component resources. By creating a list of permitted actions that all or some users can run on deployments, you ensure that the users cannot initiate any destructive or costly changes. The use cases related to day 2 actions policies are an introduction to the procedure.

When you entitle users to run day 2 actions, you select the individual actions that they can run. You are creating an inclusion list, not an exclusion list.

When does a day 2 actions policy go into effect?

- If you do not have any Day 2 Action policies defined, then no governance is applied and all users have access to all the actions. This initial lack of governance as you are starting out ensures that you and your users can exercise the day two actions in vRealize Automation Service Broker and vRealize Automation Cloud Assembly without the need to understand day 2 policies.
- After you determine that you are ready to control who has access to what actions, you add governance in the form of a single Day 2 Action policy. When the first policy goes into effect, the Day 2 Action policies are enforced for all users in vRealize Automation Service Broker and vRealize Automation Cloud Assembly. As a result, only the users for whom the first policy is true can run the selected actions. All others are excluded. They are excluded because the actions policies are a white list of users. By excluding all others, you are able to craft the policies to match your governance goals.
- To entitle other users, you must create policies that entitle them to run the actions you select.

As you create your policies, the way that you define Day 2 Actions policies must take sharing status into consideration.

To focus when the Day 2 Actions policies are applied, you can configure scope, role, and deployment criteria. These configurations control what deployments the policy is applied to and who can run the actions when the policy is enforced.

- What deployments the policy is applied to.
 - Scope determines whether the policy is applied to deployments at the organization or project level.

- Deployment criteria narrows the scope of the policy to particular aspects of deployments.
- Who can run what actions on those deployments.
 - Role entitles the members of selected role, within the selected scope and deployment criteria, to run the selected actions.

Day 2 policies are enforced when a user tries to manage a deployment using the Actions menu on the deployment or on the component resources.

As you review the day 2 actions policies use case, you must also select the actions. You must select the actions that support your cloud accounts.

- Actions are cloud specific. When you are entitling the users to make changes, consider what cloud accounts the entitled users are deploying to and ensure that you select all the cloud-specific versions of the actions. For example, add Cloud.AWS.EC2.Instance.Resize, Cloud.GCP.Machine.Resize, and Cloud.Azure.Machine.Resize to entitle users to resize those machines.
- Cloud agnostic actions, for example, Cloud.Machine.Resize, exist to accommodate resources where the on-boarding or migration process cannot identify the machine type. If you entitle users to the cloud agnostic actions, the actions appear in the action list, but running the actions has no effect.

Prerequisites

- For a list of possible actions, see [What actions can I run on vRealize Automation Service Broker deployments](#).
- For more information about constructing deployment criteria, see [How do I configure deployment criteria in vRealize Automation Service Broker policies](#).

Procedure

- 1 Select **Content and Policies > Policies > Definitions > New Policy > Day 2 Actions Policy**.

2 Configure Day 2 Policy 1.

As an administrator, you want to control storage costs by restricting the ability of users to request snapshots.

- a Define when the policy is valid.

Setting	Sample Value
Scope	Organization This policy applied to all deployments in your organization.
Deployment criteria	None
Enforcement type	Soft This enforcement type allows you to create other policies related to the snapshot actions that override this policy.
Role	Member This role applies the policy to all project members.

- b Select the actions that the users can run, but do not select any snapshot actions.

You explicitly entitle users to run actions. To exclude users from running snapshot actions, ensure that the actions are not selected.

In this scenario, none of the project members in your organization are entitled to create snapshots. Nor can your project administrators. Your next step is to create a policy that entitles the project administrators to create and manage snapshots.

3 Configure Day 2 Policy 2.

As an administrator, you want to give the project administrators the ability to create and manage snapshots.

- a Define when the policy is valid.

Setting	Sample Value
Scope	Organization This policy is applied to all deployments in your organization.
Deployment criteria	None
Enforcement type	Soft This enforcement type allows you to create other policies related to the snapshot actions that override this policy.
Role	Administrator This role applies the policy to the project administrators.

- b Select the snapshot actions that you want the administrators to run.

Project administrators are also entitled to run any actions that the members of their projects are entitled to run. You do not need to give them permission to member actions.

In this scenario, the project administrators are entitled to run the snapshot-related actions and all the actions that their project members are entitled to run.

4 Configure Day 2 Policy 3.

As a project administrator, you have two developers who are doing work that potentially makes a deployment unusable. You want to entitle them to snapshot and revert without your intervention. You entitle the two project members to use the snapshot actions.

- a Define when the policy is valid.

Setting	Sample Value
Scope	Project MT5 This policy applied to deployments associated with this project.
Deployment criteria	CatalogItemId eq Multi-tier five machine with LB and (createdBy eq jan@mycompany.com or createdBy kris@mycompany.com) Based on this criteria expression, only the deployments where Jan or Kris deployed a catalog item named Multi-tier five machine with LB are considered for policy enforcement.
Enforcement type	Hard This enforcement type ensures that the policy is enforced based on the definition.
Role	Member This role applies the policy to the catalog item defined in the deployment criteria.

- b Select the snapshot actions that you want the specified users to run.

Project administrators are also entitled to run any actions that the members of their projects are entitled to run.

In this scenario, Jan and Kris can use the snapshot actions on the Multi-tier 5 Machines with LB catalog item that either of them deploy. Although other members of the project can see the deployment, only Jan, Kris, and the project administrator can use the snapshot actions.

What to do next

- For more examples of how the policies are processed and enforced, see [How are vRealize Automation Service Broker policies processed](#).
- Configure policies that are relevant to your organizations and projects.

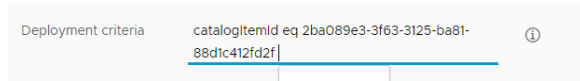
How do I configure deployment criteria in vRealize Automation Service Broker policies

The deployment criteria narrows the scope of a policy so that it is applied only to the deployments where the criteria is true. For example, you can use the deployment criteria to create a policy that is applied only to a particular catalog item or blueprint.

Deployment criteria user interface focus

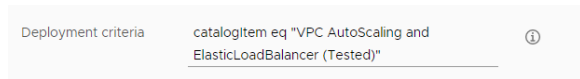
As you define the criteria, the apparent values change between the IDs and the human-readable names. When you are actively editing in text box, the ID is displayed. When you click outside the text box, the name replaces the ID, making it easier to understand your criteria expression.

Figure 3-1. Deployment criteria when the text box is active



When you click outside the text box, the name replaces the ID, making it easier to understand your criteria expression.

Figure 3-2. Deployment criteria when the text box is not the focus



Deployment criteria properties

To create a functional deployment criteria, you must understand the syntax.

The deployment criteria text box has various drop-down menus that provide the available properties and operators. How you construct your expression depends on the available values and on the order of operations.

The drop-down menus include the following properties.

Property	Description
Name	Deployment name
CreatedBy	Name of the user who requested the deployment. The format is username@mycompany.com.
BlueprintId	Identifier for the vRealize Automation Cloud Assembly blueprint used to create the deployment.
CatalogItemId	Identifier for the vRealize Automation Service Broker catalog item that was used to request the deployment.
Parenthetical operators	Used to group expressions. The appropriate opening or closing parenthesis is provided in the drop-down menu.

What is the difference between `blueprintID` and `catalogItemId`?

- Use `blueprintID` when your policy is specific to vRealize Automation Cloud Assembly blueprints. For example, an Amazon Web Services template does not have a `blueprintID`.
- Use `catalogItemId` when your policy can include vRealize Automation Service Broker catalog items based on any blueprint, template, extensibility workflow, or other content type. For example, vRealize Automation Cloud Assembly blueprints and Amazon Web Services CloudFormation templates deployed from the catalog have a `catalogItemID`.

Expression operators

The drop-down menus include the following expression operators:

- `Eq.` Equals
- `NotEq.` Does not equal.

Boolean operators

You can use one of the following Boolean operators:

- `AND`
- `OR`

Order of operations for the expression

An expression is processed in the following order:

- 1 Expressions in parentheses
- 2 `AND`
- 3 `OR`

Use the following examples to understand the order.

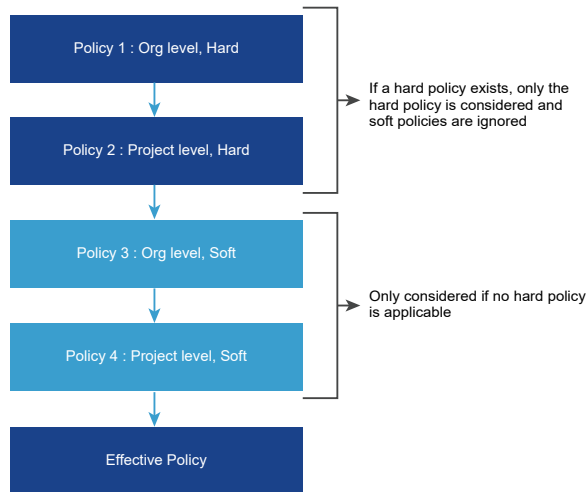
- `X OR Y AND Z`. In this example, `Y AND Z` is evaluated before `X OR Y`. Next, the `X OR` is evaluated against the results of `Y AND Z`.
- `(X OR Y) AND Z`. In this example, `X OR Y` is evaluated before `AND` because the expression in the parentheses is always evaluated first. Next the `AND Z` is evaluated against the results of `X OR Y`.

How are vRealize Automation Service Broker policies processed

Policies are processed based on the policy definition. In particular, the scope and the enforcement level determine which policy is valid when you have multiple policies that might apply to a single deployment.

How policies are ranked based on organization level and enforcement type

When a user, who is a member of a project, creates a deployment, there might be more than one policy that applies to that deployment.



To evaluate the policies, the system first identifies and ranks policies.

- 1 Are there any hard policies at either the organization and project level. If there are hard and soft policies, then only the hard policies are considered and ranked. If there are only soft policies, then the soft policies are ranked.
- 2 The ranking of all hard or all soft policies is ordered by scope, with organizational policies having a higher rank than project policies.
- 3 The final discriminating characteristic is the creation date, with older dates be ranked higher than newer dates.

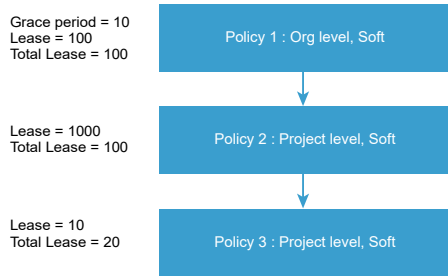
How policies are processed based on organization level and enforcement type

The policies are evaluated, ranked, and, where applicable, merged to produce an effective policy. An effective policy produces the intended results but is not always a specific named policy.

This section includes the following examples:

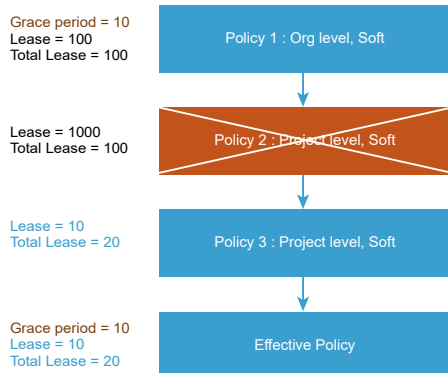
- Lease policies
- Day 2 actions policies

Review the following lease policy examples.



After identifying the policies to be considered and ranking them, the policies are then evaluated to identify the merge order.

- The highest ranking policy becomes the baseline. The second-level policy is applied on top of it, and so on.
- If a policy is incompatible with the preceding policies, then it is discarded from consideration. For example, the values are higher they the preceding policies.
- Any policy that is discarded is ignored. To see which policy is applied, select **Content and Policies > Policies > Enforcement**, locate the deployment, and review the decision notes.



Rather than applying one policy and excluding all the others above, the policies are merged and might include values from more than one individual policy.

In this example, the merging process excludes Policy 2 from consideration because the values are higher than Policy 1.

Next, Policy 3 is evaluated against Policy 1. The Lease and Total Lease values in Policy 3 are lower than Policy 1, so those values, in addition to the Grace period, become part of the effective policy.

Review the following day 2 actions policy examples.

- After identifying the policies to be considered and ranking them, the policies are then evaluated to identify the merge order.
 - The highest ranking policy becomes the baseline. The second-level policy is applied on top of it, and so on.
 - If a policy is enforced by preceding policies, for example, policy 3, then it is discarded from consideration.
 - Any policy that is discarded is ignored. To see which policy is applied, select **Content and Policies > Policies > Enforcement**, locate the deployment, and review the decision notes.

Lease policy management goal considerations

Now that you know how lease policies are processed, identify your policy management goals. By understanding how the policies are processed, you can meet your management goals without creating an excessive and unmanageable number of policies.

When deciding how to implement your policies, consider the following scenarios.

- Lease policy goals and enforcement examples
- Day 2 policy goals and enforcement examples

Table 3-1. Lease policy goals and enforcement examples

Management goal	Configuration Example	Behavior
Meaningful default organization-level policy that still allows the project-level policy values to influence the applied values.	Organization policy = Soft <ul style="list-style-type: none"> ■ Grace period: 10 ■ Lease: 100 ■ Total Lease: 100 Project 1 policy 1= Soft <ul style="list-style-type: none"> ■ Lease: 20 ■ Total Lease: 50 Project 2 policy 1= Soft <ul style="list-style-type: none"> ■ Lease: 10 ■ Total Lease: 30 	A member of project 1 requests a catalog item. Project 2 is not considered because it is not applicable to project 1 deployments. The merged effective policy is: <ul style="list-style-type: none"> ■ Grace period: 10 ■ Lease: 20 ■ Total Lease: 50
Always default to the organization-level policy.	Organization policy = Hard <ul style="list-style-type: none"> ■ Grace period: 10 ■ Lease: 100 ■ Total Lease: 100 Project 1 policy 1= Soft <ul style="list-style-type: none"> ■ Lease: 20 ■ Total Lease: 50 	A member of project 1 requests a catalog item. Project 1 policy 1 is not considered because the hard organization level project is a higher rank and the soft policy is not considered. The effective policy is: <ul style="list-style-type: none"> ■ Grace period: 10 ■ Lease: 100 ■ Total Lease: 100
All policies are defined at the project-level, with no organization-level default policy.	Project 1 policy 1 = Soft <ul style="list-style-type: none"> ■ Grace period: 10 ■ Lease: 100 ■ Total Lease: 100 Project 1 policy 2= Soft <ul style="list-style-type: none"> ■ Lease: 20 	A member of project 1 requests a catalog item. They are both soft policies, and they are both for project 1. The values are merged. The effective policy is: <ul style="list-style-type: none"> ■ Grace period: 10 ■ Lease: 20 ■ Total Lease: 100

The day 2 actions policies are used in these examples.

Table 3-2. Day 2 policy goals and enforcement examples

Management goal	Configuration Example	Behavior
Meaningful default organization-level policy that still allows the project-level policy values to influence the applied values.	Organization policy = Soft ■ Actions : Deployment.* Project 1 policy 1= Soft ■ Actions: Cloud.vSphere.Machine.* Project 2 policy 1= Soft ■ Actions: Cloud.Azure.Machine.*	A member of project 1 requests a catalog item. Project 2 is not considered because it is not applicable to project 1 deployments. The merged effective policy is: ■ Action : {Deployment.* ,Cloud.vSphere.Machine.*}
Always default to the organization-level policy.	Organization policy = Hard ■ Action : Deployment.* Project 1 policy 1= Soft ■ Action : Cloud.vSphere.Machine.*	A member of project 1 requests a catalog item. Project 1 policy 1 is not considered because the hard organization level project is a higher rank and the soft policy is not considered. The effective policy is: ■ Action : {Deployment.* }
All policies are defined at the project-level, with no organization-level default policy.	Project 1 policy 1 = Soft ■ Actions : Deployment.ChangeLease Project 1 policy 2= Soft ■ Action : Deployment.Delete	A member of project 1 requests a catalog item. They are both soft policies, and they are both for project 1. The values are merged. The effective policy is: ■ Action : {Deployment.ChangeLease , Deployment.Delete}

Customize a vRealize Automation Service Broker icon and request form

In vRealize Automation Service Broker, you can customize the icon that represents the content and the request form for imported blueprints or templates. When customizing the request form, you can also design the input parameters that allow the user requesting a catalog item to provide the values. You can customize how the custom options are presented in the form.

The icon that you provide helps you and your catalog consumers use visual queues to identify specific items. You are not required to customize a form if all you want is a custom icon. Nor are you required to customize the icon when you create a custom form.

When creating the custom form, the WordPress blueprint is used as the example in this use case. If you don't customize the request form, it is a simple list of parameters. See the following example.

New Request

WordPress-BP

Deployment Name *

Description

Project * Service Broker Project

Node Size *

Wordpress Cluster Size * 2

Select Image/OS *

Database Username *

Database Password *

Wordpress Archive Disk Size * 4

MySQL Data Disk Size * 4

CANCEL SUBMIT

In this use case, you customize the following options:

- Reduce the maximum number of Wordpress Cluster Size from 5 to 3.
- Specify operating system based on Node Size. For example, if size is small, then the operating system is coreos. If it medium, then the operating system is ubuntu.
- Set the MySQL Data Disk Size value to 5 and hide the option from the requesting users.

Prerequisites

- To add an icon, verify that you have an image that does not exceed 100 KB. The optimal size is no larger than 100x100 pixels.
- This use case assumes that you imported the WordPress use case blueprint from vRealize Automation Cloud Assembly, or that you have a blueprint or template that includes input parameters.

Procedure

- 1 Select **Content and Policies > Content**.
- 2 Locate the WordPress blueprint, click the menu to the left of the name, select **Customize icon**, and add the icon.

If all you want is a custom icon, you can stop here.

- 3 Locate the WordPress blueprint, click the menu to the left of the name, and select **Customize form**.

If the blueprint has input properties, they are listed in the Request Inputs pane on the left, and are added to the canvas.

- 4 Edit the form using the values provided in the following table.

For this field in the screenshot	Appearance	Values	Constraints
WordPress Cluster Size			Maximum value ■ Value source = Constant ■ Max value = 3
Select Image/OS		Default value ■ Value source = Conditional value ■ Expression = Set value = coreos If Node Size Equals small ■ Expression = Set value = ubuntu If Node Size Equals medium	
MySQL Data Disk Size	Visibility ■ Value source = Constant ■ Visible = No	Default value ■ Value source = Constant ■ Default value = 5	

- 5 Click and drag the fields to rearrange them on the form.
- 6 To turn on the custom form, toggle on the **Active custom form**.

- 7 Click **Save**.

Results

The request form is now similar to the following example.

New Request

WordPress-BP

Deployment Name * **WordPress Team Testing**

Description

Project * **Service Broker Project**

Node Size * **medium** ⓘ

Wordpress Cluster Size * **4** ⓘ ⓘ

Select Image/OS * **ubuntu**

Database Username * **abcd@myco.com**

Database Password * **a1b!**

Wordpress Archive Disk Size * **4** ⓘ

CANCEL **SUBMIT**

Notice that the Wordpress Cluster Size field indicates an error. The limit is 3, but the user entered a value of 4.

What to do next

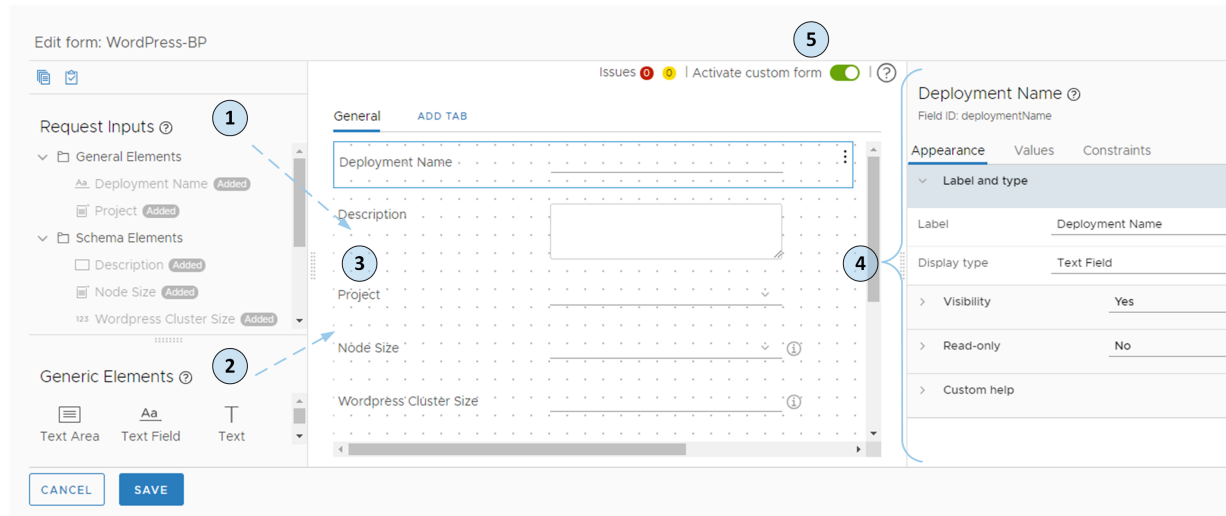
Request the item in the catalog and verify that the presentation and behavior is what you expected.

Learn more about vRealize Automation Service Broker custom forms

To create useful forms based on input parameters, you can design how the information appears at request time, how the parameters values are populated, and add any specialized constraints.

Custom request form designer

You use the form designer to create your custom form.



To create a custom form:

- 1 Drag elements (1 and 2) onto the design canvas (3).
- 2 Configure each element using the properties pane (4).
- 3 Activate the form (5).

The custom form designer supports data validation by adding constraints to a field. For constraints options that are applied as you create a form, see [Custom form designer field properties in vRealize Automation Service Broker](#). For a constraint example, see [Customize a vRealize Automation Service Broker icon and request form](#).

When you have a working form, you might want to reuse or share it rather than recreating it. Click **Actions** on the form designer and export as JSON or YAML, or import the form as needed.

Custom form designer field properties in vRealize Automation Service Broker

The field properties in vRealize Automation Service Broker determine how the fields look and what default values are presented to the user. You can also use the properties to define rules that ensure that users provide a valid entry when they enter the item in the catalog.

You configure each field individually. Select the field and edit the field properties.

Value source

For many of the properties, you can select from various value source options. Not all source options are available for all field types or properties.

- **Constant.** The value does not change. Depending on the property, the value might be a string, an integer, a regular expression, or selected from a limited list, for example Yes or No. For example, you can provide 1 as a default value integer, select No for the Read-only property, or provide the regular expression to validate a field entry.

- **Conditional value.** The value is based on one or more conditions. The conditions are processed in the order listed. If more than one condition is true, the last condition that is true determines the behavior of the field for that property. For example, you can create a condition that determines if a field is visible based on the value in another field.
- **External source.** The value is based on the results of a vRealize Orchestrator action. For example, calculate cost based on a scripted vRealize Orchestrator action. For an example, see [Using vRealize Orchestrator actions in the custom form designer in vRealize Automation Service Broker](#)
- **Bind field.** The value is the same as the field to which it is bound. The available fields are limited to the same field type. For example, you bind default value for an authentication needed check box field to another check box field. When one target field check box is selected in the request form, the check box on the current field is selected.
- **Computed value.** The value is determined based on how the operator processes the selected fields and values. Text fields use the concatenate operator. Integer fields use the selected add, subtract, multiply or divide operations. For example, you can configure an integer field to convert megabytes to gigabytes using the multiply operation.

Field appearance

You use the appearance properties to determine whether the field appears on the form and what label and custom help you want to provide to your catalog users.

Table 3-3. Appearance Tab Options

Option	Description
Label and type	<p>Provide a label and select a display type.</p> <p>The available display types depend on the field. Some fields support multiple text types and others only support integers. Possible values:</p> <ul style="list-style-type: none"> ■ Decimal ■ Drop Down ■ Image ■ Integer ■ Multi Select ■ Password ■ Radio Group ■ Text ■ Text Area ■ Text Fields <p>Drop-down and data grid fields include a Placeholder setting. The entered value appears as an internal label or instructions in the drop-down menu, or as a general label or instructions in the data grid.</p>
Visibility	<p>Show or hide a field on the request form.</p> <ul style="list-style-type: none"> ■ Constant. Select Yes to display the field on the form. Select No to hide the field. ■ Conditional value. Visibility is determined by the first expression that is true. For example, a field is visible if a check box is selected on a form. ■ External source. Visibility is determined by the results of the selected vRealize Orchestrator action.
Read-only	<p>Prevent users from changing the field values.</p> <ul style="list-style-type: none"> ■ Constant. Select Yes to display the value but prevent changes. Select No to allow changes. ■ Conditional value. Status is determined by the first expression that is true. For example, a field is read-only if the value in a storage field is greater than 2 GB. ■ External source. Status is determined by the results of the selected vRealize Orchestrator action.
Rows per page	<p>For data grid elements only.</p> <p>Enter the number of rows.</p>
Custom help	<p>Provide information about the field to your users. This information appears in signpost help for the field.</p> <p>You can use simple text or HTML, including href links. For example, <code>VMware Service Broker documentation</code>.</p>

Field values

You use the values properties to provide any default values.

Table 3-4. Values Tab Options

Option	Description
Columns	<p>For the data grid element only.</p> <p>Provide the label, ID, and value type for each column in your table.</p> <p>The default value for the data grid must include the header data that matches the defined columns. For example, if you have user_name ID for one column and user_role ID for another, then the first row is user_name,user_role.</p> <p>For configuration examples, see Using the data grid element in the vRealize Automation Service Broker custom form designer.</p>
Default value	<p>Populates the field with a default value based on the value source.</p> <p>Possible value sources depend on the field.</p> <ul style="list-style-type: none"> ■ Constant. The entered string. ■ Conditional value. The default value is determined by the first expression that is true. For example, the default value of a storage field is 1 GB if the memory field is less than 512 MB. ■ External source. Value is based on the results of the selected vRealize Orchestrator action. ■ Bind field. Value is the same as the selected field. ■ Computed value. Value is based on the results of the provided field values and the selected operator. For example, the default value of memory in MB is based on the memory in GB multiplied by 1024.
Value option	<p>Populates a drop-down, multi-select, radio group, or value picker fields.</p> <ul style="list-style-type: none"> ■ Constant. The format for the list is Value Label,Value Label,Value Label. For example, 2 Small,4 Medium,8 Large. ■ External source. Value is based on the results of the selected vRealize Orchestrator action.
Step	<p>For integer or decimal fields, define the incremental or decremental values.</p> <p>For example, if the default value is 1 and you set the step value to 3, then the allowed values are 4, 7, 10, and so on.</p>

Field constraints

You use the constraint properties to ensure that the requesting user provides valid values in the request form.

Table 3-5. Constraints Tab Options

Option	Description
Required	<p>The requesting user must provide a value for this field.</p> <ul style="list-style-type: none"> ■ Constant. Select Yes to require that the requesting user provides a value. Select no if the field is optional. ■ Conditional value. Whether the field is required is determined by the first expression that is true. For example, this field is required if the operating system family starts with Darwin in another field. ■ External source. Status is based on the results of the selected vRealize Orchestrator action.
Regular expression	<p>Provide a regular expression that validates the value and a message that appears when the validation fails.</p> <p>The regular expressions must follow JavaScript syntax. For an overview, see Creating a regular expression. For more detailed guidance, see Syntax.</p> <ul style="list-style-type: none"> ■ Constant. Provide a regular expression. For example, for an email address, the regular expression might be <code>^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\$</code> and the validation error message is <code>The email address format is not valid. Please try again.</code> ■ Conditional value. The regular expression that is used is determined by the first expression that is true.
Minimum value	<p>Specify a minimum numeric value. For example, a password must have at least 8 characters.</p> <p>Provide an error message. For example, <code>The password must be at least 8 characters.</code></p> <ul style="list-style-type: none"> ■ Constant. Enter the integer. ■ Conditional value. The minimum value is determined by the first expression that is true. For example, a minimum CPU value is 4 if the operating system does not equal Linux. ■ External source. Value is based on the results of the selected vRealize Orchestrator action.

Table 3-5. Constraints Tab Options (continued)

Option	Description
Maximum value	<p>Maximum numeric value. For example, a field is limited to 50 characters.</p> <p>Provide an error message. For example, <code>This description cannot exceed 50 characters.</code></p> <ul style="list-style-type: none"> ■ Constant. Enter the integer. ■ Conditional value. The maximum value is determined by the first expression that is true. For example, a maximum storage value is 2 GB if the deployment location equals AMEA. ■ External source. Value is based on the results of the selected vRealize Orchestrator action.
Match field	<p>This field value must match the selected field value.</p> <p>For example, a password confirmation field must match the password field.</p>

Using the data grid element in the vRealize Automation Service Broker custom form designer

If you use a data grid element in a custom form, the data that is presented in the table might be manually provided.

Example: Provided CSV data example

In this use case, you have a table of values that you provide in the custom request form. You provide the information in the table as a constant value source. The source is based on a CSV data structure where the first row defines the grid headers. The headers are the column IDs separated by a comma. Each additional row is the data that appears in each row in the table.

- 1 Add the Data Grid generic element to the design canvas.
- 2 Select the data grid and define the values in the properties pane.



Data Grid ?



Field ID: datagrid_ecdf4fe3



Appearance **Values** Constraints

Columns

ADD COLUMN

Label	Username	 
Id	username	
Type	String	▼

Label	Employee ID	 
Id	employeeid	
Type	Integer	▼

Label	Manager	 
Id	manager	
Type	String	▼

Default value Constant

Value Constant ▼

source

CSV

```
username,employeeid,manager
leonardo,95621,Farah
vindhya,15496,Farah
martina,52648,Nikolai
```

Label	ID	Type
Username	username	String
Employee ID	employeeid	Integer
Manger	manager	String

Define the CSV values.

```
username,employeeId,manager
leonardo,95621,Farah
vindhya,15496,Farah
martina,52648,Nikolai
```

- 3 Verify that the data grid displays the expected data in the blueprint request form.

<input type="checkbox"/>	Username	Employee ID	Manager
<input type="checkbox"/>	leonardo	95621	Farah
<input type="checkbox"/>	vindhya	15496	Farah
<input type="checkbox"/>	martina	52648	Nikolai

1 - 3 of 3

Example: External Source Example

This example uses the previous example but the values are based on a vRealize Orchestrator action. Although this is a simple action example, you can use a more complex action where you retrieve this information from a another database or system.

- 1 In vRealize Orchestrator, configure an action, `getUserDetails`, with an array similar to the following example.

The screenshot shows the vRealize Orchestrator interface for configuring an action named `getUserDetails`. The **Script** tab is selected, showing a JavaScript script that returns an array of user details: `return [{"username": "Fritz", "employeeId": 6096, "manager": "Tom"}]`. The **API Explorer** on the right shows the **Properties** section, where the **Return type** is set to **Array**. The **Inputs** section is also visible but empty.

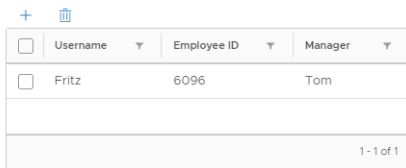
- a On the General tab, enter the name **getUserDetails** and provide a Module name.
- b On the Script tab, use the following script example.


```
return [{"username": "Fritz", "employeeId": 6096, "manager": "Tom"}]
```
- c In the Return type area, enter or select **Properties** as the type, and click **Array**.
- d Version and save the action.

- In vRealize Automation Service Broker, add the data grid and use the Values tab configure the data grid columns with the following values.

Label	ID	Type
Username	username	String
Employee ID	employeeid	Integer
Manger	manger	String

- In the Default value, Value source list, select **External source**.
- In Select action, enter **getUserDetails** and select the action you created in vRealize Orchestrator.
- Save the form.
- In the catalog, verify the table in the request form.



<input type="checkbox"/>	Username ▼	Employee ID ▼	Manager ▼
<input type="checkbox"/>	Fritz	6096	Tom

1 - 1 of 1

Using vRealize Orchestrator actions in the custom form designer in vRealize Automation Service Broker

When you customize a vRealize Automation Service Broker request form, you can base the behavior of some fields on the results of a vRealize Orchestrator action.

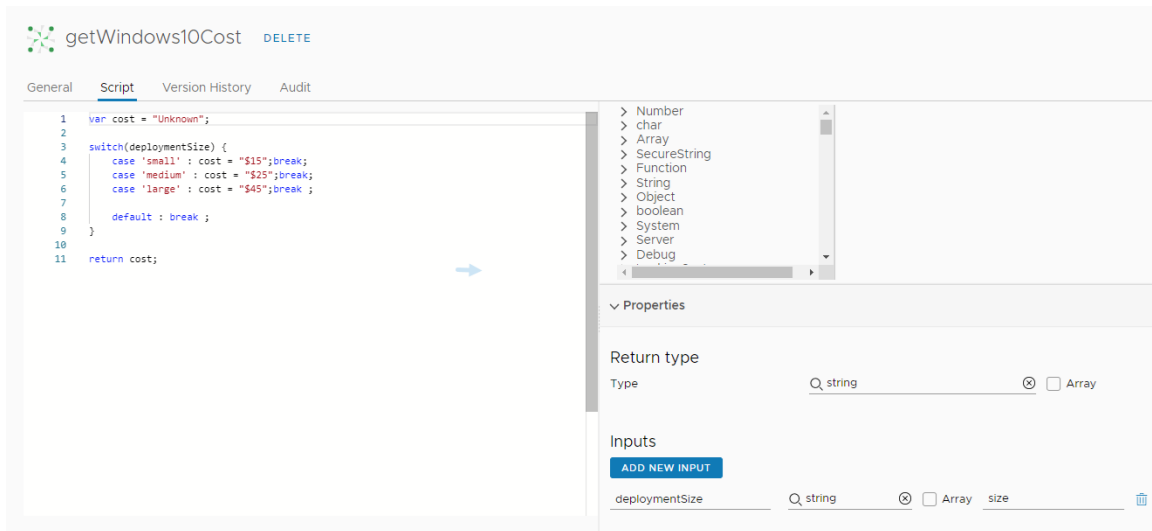
There are several ways that you can use vRealize Orchestrator actions. You might have an action that pulls the data from a third source, or you can use a script that defines the size and cost. This example uses a script.

The first example is based on manually added fields so that you understand the underlying process. The second example, uses the same premise, but instead relies on a blueprint field.

Example: Size and cost as manually added fields example

In this use case, you want the catalog user to select a virtual machine size, and then display the cost of that machine per day. To do this example, you have a vRealize Orchestrator script that correlates the size and cost. You then add a size field and a cost field to the blueprint custom form. The size field determines the value that appears in the cost field.

- In vRealize Orchestrator, configure an action named `getWindows10Cost`.



2 Add a script.

You can use the following example script.

```

var cost = "Unknown";

switch(deploymentSize) {
  case 'small' : cost = "$15";break;
  case 'medium' : cost = "$25";break;
  case 'large' : cost = "$45";break ;

  default : break ;
}

return cost;

```

3 Add deploymentSize as an input string.

4 In vRealize Automation Service Broker, add and configure a Size field to a blueprint custom form.

Configure the size field as drop-down element with Small, Medium, and Large values.

Size ⓘ
Field ID: WindowsMachine-size

Appearance **Values** Constraints

▼ Default value large

Value source Constant

▼ Value options Constant

Value source Constant

small|Small,medium|Medium,large|Large

On the Values tab, configure the following property values.

- Default value = **Large**
- Value options
 - Value source = **Constant**
 - Value definition = **small|Small,medium|Medium,large|Large**

- 5 Add the cost field as a text field to display the cost as defined in the vRealize Orchestrator action based on the value selected in the size field.

Cost ⓘ
Field ID: cost

Appearance **Values** Constraints

▼ Default value External source

Value source External source

Select action com.vmware.vra.customforms/getWindows10Cost

Action inputs

deploymentSize Field Size

On the Values tab, configure the following property values.

- Default value = External source
- Select action = <your vRealize Orchestrator actions folder>/getWindows10Cost
- Action inputs
 - deploymentSize. This value was configured in the action as the input.
 - Field
 - Size. This is the field that you previously created

- 6 Enable the custom form and save it.
- 7 To verify that it is working, request the item in the catalog. You should see the Cost field populated based on the selected Size value.

Example: Cost based on schema element example

In this use case, you want the catalog user to see the cost of that machine per day based on the flavor property in the blueprint. To do this example, you use the vRealize Orchestrators script from the previous example. But in this use case the cost is based on the flavor size that your user selected in the custom form when they request the vRealize Automation Service Broker catalog item.

The simple example blueprint includes a size input field where the user selects the flavor property.

```

1  formatVersion: 1
2  inputs:
3    size:
4      type: string
5      enum:
6        - small
7        - medium
8        - large
9      description: Size of Nodes
10     title: Select machine size
11   image:
12     type: string
13     enum:
14       - ubuntu
15       - centos
16       - windows
17     description: OS image
18     title: Select OS
19   resources:
20     Cloud_vSphere_Machine_1:
21       type: Cloud.vSphere.Machine
22       properties:
23         image: '${input.image}'
24         flavor: '${input.size}'
25

```

The custom form uses the field, named `Select machine size` in this example.

The cost deploymentSize input is based on the Select machine size field. SCREENSHOT

The screenshot shows a form with three input fields. The first field is 'Select machine size *' with a dropdown menu showing 'large'. The second field is 'Cost' with a text input showing '\$45'. The third field is 'Select OS *' with a dropdown menu showing 'windows'. Each field has an information icon (i) to its right.

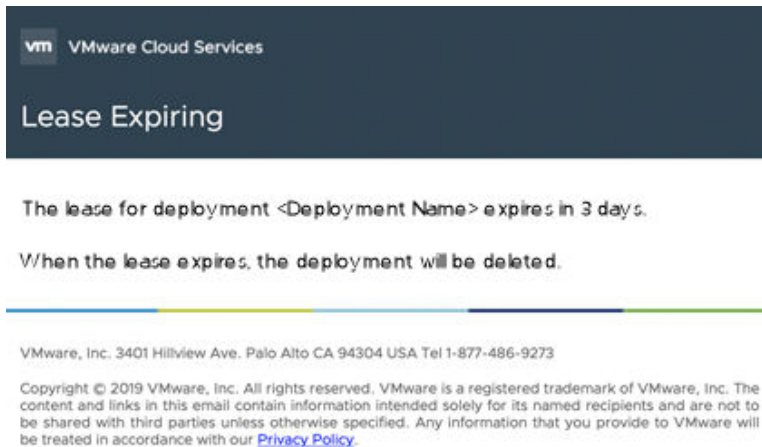
Add an email server in vRealize Automation Service Broker to send notifications

As a cloud administrator, you configure an email server if you want to send messages to users about events in vRealize Automation Service Broker and vRealize Automation Cloud Assembly. The messages are a courtesy that improves the experience of your consumers.

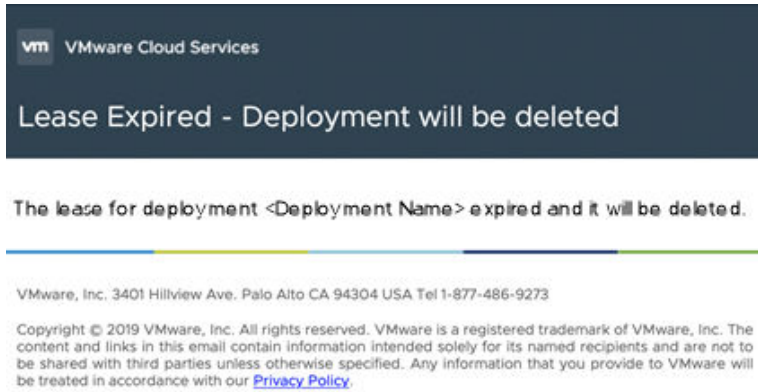
This email server is for outbound messages only.

Email messages are sent to users in the following scenarios.

- A deployment lease expires soon. The message is sent to the deployment owner three days before the lease expires.



- A deploy lease expired and the deploy is about to be deleted. The message is sent to the deployment owner 15–30 minutes before it is destroyed.



Prerequisites

- Verify that you know the credentials required to configure the email server. You must provide the server name and an email account that you want to be the message sender. If your email server requires authentication, you must also provide the user name and password.

Procedure

- 1 Select **Content and Policies > Notifications > Email Servers**.
- 2 Enter the information for each setting.
If you need assistance on a particular setting, consult the signpost help .
- 3 To verify the configured settings, click **Test Connection**.
- 4 To save, click **Create**.

What to do next

As the administrator, monitor the leases to ensure that the messages are sent to the deployment owners at the correct time.

Working with the Infrastructure options in vRealize Automation Service Broker

The Infrastructure tab that is provided in vRealize Automation Service Broker is available to administrators. As an administrator who is setting up the service catalog for your users, you use the options to create and manage configuration and connection information that is shared with vRealize Automation Cloud Assembly.

For more information about the various connection options, see [Setting up Cloud Assembly for your organization](#).

To better understand projects, and how it associates users with resources, see [Adding and managing Cloud Assembly projects](#).

When working with cloud zones, see [Learn more about Cloud Assembly cloud zones](#)

How do I deploy a vRealize Automation Service Broker catalog item

4

As a vRealize Automation Service Broker consumer, you deploy a catalog item that was imported from vRealize Automation Cloud Assembly, Amazon CloudFormation, and other sources so that you can deploy it as part of your work processes.

The catalog items that are provided to you by your cloud administrator. The items that are available depend on your project membership. If you are member of one project, you can see only the catalog items for that project. If you are member of several projects, you can see the catalog items those projects.

Projects also determine your options at deployment time.

The information provided in this article is general because each catalog item is unique. The variation depends on how the blueprint, template, and other items were constructed, including what variables are made available to you at request time.

Procedure

1 Click **Catalog**.

The available catalog items are available to you based on your project membership.

2 Locate the catalog item you plan to deploy.

You can use the filter, search, or sorting options to find the catalog item.

3 Click **Request**.

4 Provide any required information.

If the blueprint has more than one released version, select the version that you want to deploy.

A deployment name is required, as is a project. The project list includes those that you are a member of.

The form might have other options that you must configure, depending on how the blueprint was designed.

5 Click **Submit**.

The provisioning process begins and the Deployments tab opens with your current request at the top.

What to do next

Monitor your request. See [Monitoring vRealize Automation Service Broker deployments](#).

How do I manage my vRealize Automation Service Broker Deployments

5

As a vRealize Automation Service Broker consumer, you use the Deployment tab to manage your deployments, making changes to deployments, troubleshooting failed deployments, and destroying unused deployments.

The deployments are the provisioned instances of blueprints or templates. The Deployments tab displays your successful and failed deployments. You can use the page to manage your successful deployments or to begin troubleshooting any failed requests.

Working with deployment cards

You can locate and manage your deployments using the card list. You can filter or search for specific deployments, and then run actions on those deployments.

- 1 Filter your requests based on attributes.
- 2 Search for deployments based on keywords or requestor.
- 3 Sort the list to order by time or name.
- 4 Run deployment-level actions on the deployment, including deleting unused deployments to reclaim resources.

You can also see deployment costs, expiration dates, and status.

Deployments 14 items

Search for deployments by name or description

Sort: Created Time (descending)

Deployment Name	Resources	Cost	Status	Created	Expires	Actions
Redshift-AgnosticVsphe...	2 Resources my-cluster1 agnostic-machine-in-vsphere...	Month to date \$60...	Off	Created a month ago	Never Expir...	Change Lease, Delete, Power Off, Power On, Update
vSphere-With-Disk-Atta...	2 Resources photon-clone-vm-01-mcm62... photon-disk-01	Month to date \$15.13	On	Created a month ago	Never Expir...	Change Lease, Delete, Power Off, Power On, Update
Agnostic machine landin...	1 Resource agnostic-machine-aws-mcm...	Month to date \$2.78	Suspended	Created a month ago	Never Expir...	Change Lease, Delete, Power Off, Power On, Update
RDS Application	4 Resources instance20181216081356139... Default-AWS-mcm62234-93... cluster201812160813018264...	Month to date \$15...		Created a month ago	Never Expir...	Change Lease, Delete, Power Off, Power On, Update

Working with deployment details

You can use the Topology tab to understand the deployment structure and resources.

The History tab includes all the provisioning events and any events related to actions that you run after requested item is deployed. If there are any problems with the provisioning process, the History tab events will help you with troubleshoot the failures.

The Cost tab provides the current cost of some components since they were deployed.

The image displays three overlapping screenshots of the vRealize Automation Service Broker interface, illustrating the deployment and cost management of an EC2 instance with an EBS volume.

Top Screenshot: Deployment Overview

- Blueprint:** EC2 with EBS Attached
- Requestor:** skavdimatti
- Project:** Human Resources Tool Project
- Expires:** Never
- Last Updated:** January 11, 2019 4:24 AM
- Created On:** December 20, 2018 4:34 AM
- Topology:** Shows a diagram with two resources: **Cloud_AWS_Vo_1** (Volume) and **Cloud_AWS_EC_1** (EC2 Instance).
- Details for Cloud_AWS_Volume_1:**
 - Resource Name: attached-ebs-disk
 - CapacityGB: 1
 - Type: HDD

Middle Screenshot: Events for All Requests

Timestamp	Status	Resource Type	Resource Name	Details
Dec 20, 2018, 4:35:46 AM	REQUEST_FINISHED			
Dec 20, 2018, 4:35:45 AM	CREATE_FINISH	Cloud_AWS_EC2 Instance	Cloud_AWS_EC_2 Instance_1	Provisioning diagram
Dec 20, 2018, 4:35:45 AM	CREATE_IN_PROGRESS	Cloud_AWS_EC2 Instance	Cloud_AWS_EC_2 Instance_1	

Bottom Screenshot: Cost Analysis

Resource	Cost
Cloud_AWS_Volume_1	\$0.04
Storage	\$0.04
Cloud_AWS_EC2_Instance_1	\$2.78
Total	\$2.82

The Cost Analysis section also includes a topology diagram with cost labels: \$0.04 for the volume and \$2.78 for the EC2 instance.

This chapter includes the following topics:

- [Monitoring vRealize Automation Service Broker deployments](#)
- [What can I do if a vRealize Automation Service Broker deployment fails](#)
- [What actions can I run on vRealize Automation Service Broker deployments](#)

Monitoring vRealize Automation Service Broker deployments

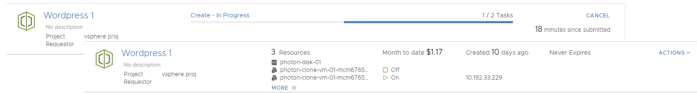
You monitor vRealize Automation Service Broker deployment requests to ensure that the resources are provisioned, that the provisioned resources are running, and to resize or destroy the resources as needed.

The Deployment tab provides information about the current state of the deployment and where the resources are deployed in your provider clouds.

How do I know that my deployment request succeeded

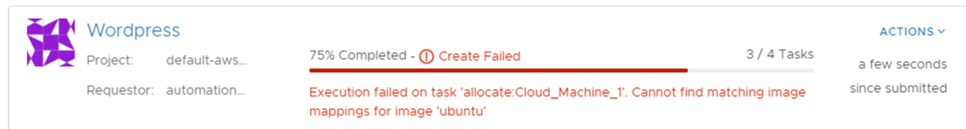
The deployment cards that appear on the Deployments tab show the state of the deployment, including in-progress (top) and completed (below). The card includes the number of deployed resources, how long it has been deployed, and the lease expiration date.

The cards also provide the IP addresses and the actions that you can run on the deployment.



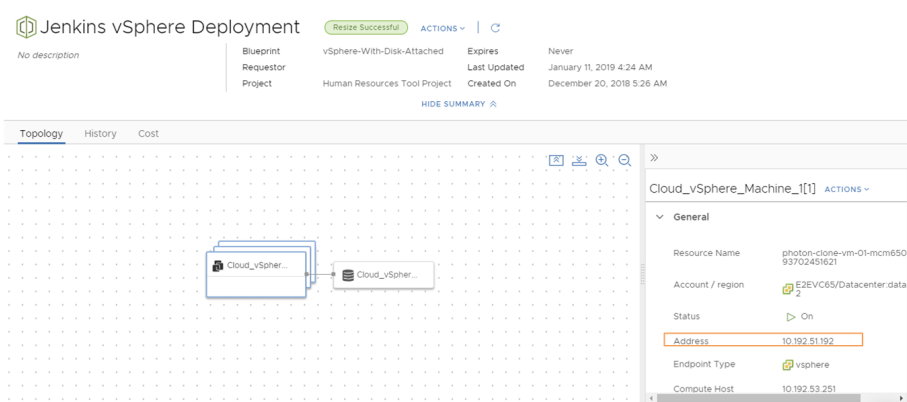
If a deployment fails, the cards show the error message for the point of failure and the process progress. To learn more about the failure, click the deployment name at review the History tab.

For more information about troubleshooting failed deployments, see [What can I do if a vRealize Automation Service Broker deployment fails.](#)



Where are my resources deployed

To access your successfully provisioned deployments, you might need more than the IP address provided on the card. Click the deployment name and review the deployment details on the Topology tab.



You most likely need the IP address for the primary component. As you click on each component, notice the information that is provided is specific to that component.

The availability of the external link depends on the cloud provider. Where it is available, you must have the credential on that provider to access the component.

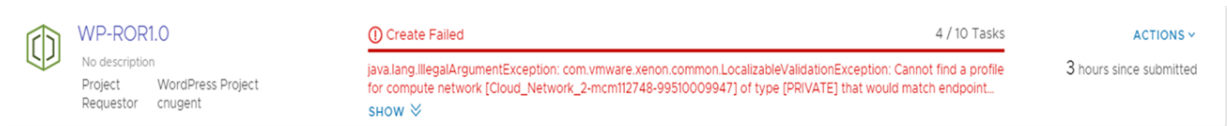
What can I do if a vRealize Automation Service Broker deployment fails

Your deployment request might fail for many reasons. It might be due to network traffic, a lack of resources on the target cloud provider, or a flawed deployment specification. Or, the deployment succeeded, but it does not appear to be working. You can use vRealize Automation Service Broker to examine your deployment, review any error messages, and determine whether the problem is the environment, the requested workload specification, or something else.

You use this workflow to begin your investigation. The process might reveal that the failure was due to a transient environmental problem. Redeploying the request after verifying the conditions have improved resolves this type of problem. In other cases, your investigation might require you to examine other areas in detail.

Procedure

- 1 To determine if a request failed, click the **Deployments** tab and locate the deployment card.



Failed deployments are indicated on the card.

- a Review the error message.
- b For more information, click the deployment name for the deployment details.

- 2 On the deployment details page, click the **History** tab.

WP - ROR1 Create Failed ACTIONS | [↗](#)

No description

Blueprint	Wordpress Basic	Expires	Never
Requestor	cnugent	Last Updated	February 22, 2019 1:54 PM
Project	WordPress Project	Created On	February 22, 2019 1:54 PM

[HIDE SUMMARY](#) [↗](#)

Topology **History** Cost

ALL REQUESTS (1)

2/22/19 1:54 PM CREATE cnugent **2.a**

Events for All Requests

Timestamp	Status	Resource Type	Resource Name	Details 2.b
Feb 22, 2019 1:55:09 PM	REQUEST_FAILED			No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist for the current project and they have been properly tagged.
Feb 22, 2019 1:55:08 PM	ALLOCATE_FAILED	Cloud.Machine	DBTier	No placement exists that satisfies all of the request requirements. Please check if suitable placements and cloud zones exist for the current project and they have been properly tagged.
Feb 22, 2019 1:55:02 PM	ALLOCATE_IN_PROGRESS	Cloud.Machine	DBTier	
Feb 22, 2019	ALLOCATE	Cloud.Net	WP-Netwo	

- Review the event tree to see where the provisioning process failed. This tree is useful when you modify a deployment, but the change fails.
- The **Details** provides a more verbose version of the error message.

What to do next

If you are unable to resolve your problem, contact your cloud administrator for additional assistance.

What actions can I run on vRealize Automation Service Broker deployments

After you deploy blueprints, you can run actions in vRealize Automation Service Broker to modify and manage the resources. The available actions depend on the resource type and whether the action is supported on a particular cloud account or integrated platform.

The available actions also depend on what your administrator entitled you to run.

As an administrator or project administrator, you can set up Day 2 Actions policies. See [How do I entitle deployment users to vRealize Automation Service Broker day 2 actions using policies.](#)

Table 5-1. List of possible actions

Action	Applies to these resource types	For these cloud accounts or integrations	Description
Add Disk	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Add additional disks to existing virtual machines.
Change Lease	Deployments	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	<p>Change the lease expiration date and time.</p> <p>When a lease expires, the deployment is destroyed and the resources are reclaimed.</p> <p>Lease policies are set in vRealize Automation Service Broker.</p>
Connect to Remote Console	Machines	<ul style="list-style-type: none"> ■ VMware vSphere 	<p>Open a remote session on the selected machine.</p> <p>Review the following requirements for a successful connection.</p> <ul style="list-style-type: none"> ■ As a deployment consumer, verify that the provisioned machine is powered on.
Create Snapshot	Machines	<ul style="list-style-type: none"> ■ Google Cloud Platform ■ VMware vSphere 	<p>Create a snapshot of the virtual machine.</p> <p>If you are allowed only two snapshots in vSphere and you already have them, this command is not available until you delete a snapshot.</p>
Delete	Deployments	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	<p>Destroy a deployment.</p> <p>All the resources are deleted and the reclaimed.</p> <p>If a delete fails, you can run the delete action on a deployment a second time. During the second attempt, you can select Ignore Delete Failures. If you select this option, the deployment is deleted, but the resources might not be reclaimed. You should check the systems on which the deployment was provisioned to ensure that all resources are removed. If they are not, you must manually delete the residual resources on those systems.</p>
	Machines and load balancers	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	Delete a machine or load balancer from a deployment. This action might result in an unusable deployment.
Delete Snapshot	Machines	<ul style="list-style-type: none"> ■ VMware vSphere ■ Google Cloud Platform 	Delete a snapshot of the virtual machine.
Edit Tags	Deployments	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	Add or modify resource tags that are applied to individual deployment resources.
Power Off	Deployments	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	Power off the deployment without shutting down the guest operating systems.

Table 5-1. List of possible actions (continued)

Action	Applies to these resource types	For these cloud accounts or integrations	Description
	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Power off the machine without shutting down the guest operating systems.
Power On	Deployments	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	Power on the deployment. If the resources were suspended, normal operation resumes from the point at which they were suspended.
	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Power on the machine. If the machine was suspended, normal operation resumes from the point at which the machine was suspended.
Reboot	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ VMware vSphere 	Reboot the guest operating system on a virtual machine. For a vSphere machine, VMware Tools must be installed on the machine to use this action.
Reconfigure	Load Balancers	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ Google Cloud Platform ■ VMware vSphere 	Change the load balancer protocol, port, health configuration, and member pool settings.
Remove Disk	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Remove disks from existing virtual machines.
Reset	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ VMware vSphere 	Force a virtual machine restart without shutting down the guest operating system.
Resize	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ Google Cloud Platform ■ VMware vSphere 	Increase or decrease the CPU and memory of a virtual machine.
Resize Boot Disk	Machines	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform ■ Microsoft Azure ■ VMware vSphere 	Increase or decrease the size of your boot disk medium.
Resize Disk	Storage disk	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Google Cloud Platform 	Increase the capacity of a storage disk.
Restart	Machines	<ul style="list-style-type: none"> ■ Microsoft Azure 	Shut down and restart a running machine.
Revert to Snapshot	Machines	<ul style="list-style-type: none"> ■ Google Cloud Platform ■ VMware vSphere 	Revert to a previous snapshot of the machine. You must have an existing snapshot to use this action.

Table 5-1. List of possible actions (continued)

Action	Applies to these resource types	For these cloud accounts or integrations	Description
Run Puppet Task	Managed resources	<ul style="list-style-type: none"> ■ Puppet Enterprise 	Run the selected task on machines in your deployment. The tasks are defined in your Puppet instance. You must be able to identify the task and provide the input parameters.
Shutdown	Machines	<ul style="list-style-type: none"> ■ VMware vSphere 	Shut down the guest operating system and power off the machine. VMware Tools must be installed on the machine to use this action.
Suspend	Machines	<ul style="list-style-type: none"> ■ Microsoft Azure ■ VMware vSphere 	Pause the machine so that it cannot be used and does not consume any system resources other than the storage it is using.
Update	Deployments	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	Change the deployment based on the input parameters.
Update Tags	Machines and disks	<ul style="list-style-type: none"> ■ Amazon Web Service ■ Microsoft Azure ■ VMware vSphere 	Add, modify, or delete a tag that is applied to an individual resource.