



Management Suites Business Unit | Cloud Solutions Engineering

VMware vRealize Code Stream Management Pack for IT DevOps – Operations Guide

Enabling IT DevOps on VMware vRealize Automation

© 2015 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. This product is covered by one or more patents listed at <http://www.vmware.com/download/patents.html>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. [Copyright and trademark information](#)

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

Contents

Operations Guide	6
Intended Audience	6
IT DevOps	6
The Software-Defined Data Center	6
Software-Defined Services	7
vRealize Code Stream Management Pack Concepts	7
IT DevOps	8
Managing Administrators.....	9
Testing Framework	11
Workflow Example.....	12
Test Workflow Requirements	14
Managing Content Endpoints	15
Add a Content Endpoint	15
Delete One or More Content Endpoints	18
Managing Content Packages	18
Management Pack Actions.....	18
Content Package Approvals.....	19
Single and Group Package Requests	19
Delete Packages	28
Delete Group Definition	29
Managing Captured or Released Content Packages.....	29
Managing Branches	31
Create a New Branch	31
Delete a Branch.....	32
Merge a Branch.....	32
Pipeline Process Flow Overview	33
Pipeline Process Flows for Content Package Request Actions.....	33
Pipeline Process Flow Prerequisites for Package Types.....	34

Troubleshooting	36
Package Persistence	36
vRealize Code Stream Pipeline Executions	37
Enable Debug Mode	38
Workflow Executions	39
Scheduled Workflow	40

List of Figures

Figure 1. Content Management Tests	12
Figure 2. Add a DNS A Record	12
Figure 3. Checks Action Exists Parameters.....	15
Figure 4. Previously Captured or Released Content Packages	29
Figure 5. Item Details	30
Figure 6. Package Actions	30
Figure 7. A Failed Pipeline	37
Figure 8. Process Request Queue.....	40
Figure 9. Suspend Task.....	41

List of Tables

Table 1. Content Package Types.....	8
Table 2. Content Endpoint Package Type Support	9
Table 3. Content Package Source	19
Table 4. Single and Group Package Requests.....	19
Table 5. Pipeline Failure Results	37
Table 6. Troubleshooting Workflow Executions.....	39

Operations Guide

This document provides operational guidance for using the vRealize™ Code Stream Management Pack for IT DevOps. It covers the following topics:

- Adding content endpoints to vRealize Automation and vRealize Orchestrator environments.
- Capturing, testing, and releasing software-defined content.
- Using branches to allow concurrent work on the same service by multiple developers.

Intended Audience

This guide is intended for users who are familiar with administering vRealize Automation and vRealize Orchestrator.

IT DevOps

The VMware vRealize Code Stream Management Pack for IT DevOps is an extensible, customizable framework that provides an efficient workflow to capture, apply version control, test, and manage the release of any software-defined content, such as workflows, blueprints, or templates, throughout multiple environments. This allows the automation of time consuming and error-prone manual processes so that management tasks are made repeatable, saving resources and improving workforce creativity and output.

The Software-Defined Data Center

VMware's concept of the Software-Defined Data Center (SDDC) abstracts every service in the data center from its underlying platform and defines services as software. Software-defined content can be treated just like code in any software application, meaning that modern software development methodologies can be adopted, and practices, such as agile development and DevOps, can be applied to the configurations and content within the SDDC. Although this approach lends itself to modern development practices and rapid service delivery, the reality is that it can be quite difficult to manage a whole data center worth of software-defined services.

The vRealize Code Stream Management Pack for IT DevOps eases the burden of managing software-defined content by providing administrators with the following abilities:

- Manage Software-Defined Data Center (SDDC) content as code, using SDDC techniques and tools.
- Apply version control, automated testing, and release automation to all SDDC content.
- Capture software-defined content automatically in a consistent format from multiple environments.
- Store content in a central JFrog Artifactory artifact repository.
- Partition the repository to allow multiple developers to work on the same content separately.
- Tag captured content to allow concurrent release to multiple environments.
- Run automated tests before allowing environmental testing or release of content.
- Roll back software-defined content.

- Leverage vRealize Automation Advanced Service Design (ASD) forms and vRealize Code Stream pipelines.
- (Optional) Require approval before deploying content to a test or release environment.

Software-Defined Services

The management pack streamlines the administration of SDDC content in a data center in the following ways:

- Groups of services are defined as code.
- Content is abstracted from its underlying platform.
- Content management is conducted as a programmable service.

Defining services programmatically makes it possible to provide vendor-agnostic services that are transferable across SDDC platforms. This makes it possible to employ modern development practices that provide continuous integration and delivery of software-defined content.

vRealize Code Stream Management Pack Concepts

Understanding the following terms is essential to knowing how the management pack works.

Term	Definition
Software-defined content	Blueprints, workflows, or configuration files that determine the behavior of the services provided by the SDDC platform.
Content package type	A particular type of software-defined content such as an action, blueprint, package, workflow, or template.
Content package	A file that contains definitions in a consistent format for software-defined services such as the following: <ul style="list-style-type: none"> • vRealize Orchestrator actions, packages, configuration elements, and workflows • vRealize Automation advanced services • IaaS blueprints
Content endpoint	An infrastructure endpoint in the SDDC that is targeted for the capture, test, and release of managed content. An example content endpoint is a vRealize Automation instance.
Dependency	Additional software-defined content that is required for proper functioning. You can choose to include dependencies with a content package so that all of the components for a service are available and up-to-date when you release the package. You can also choose not to include dependencies so that you can version control them separately or include them in another content package type.

Release version	<p>A content package intended for release to a production environment. You can identify a release version in the following ways:</p> <ul style="list-style-type: none">• By selecting Mark new version as release when you capture the package.• By editing the content package to change it to a release version. <p>If a package is not marked for release, it cannot be deployed to a release environment.</p>
Single package request	<p>A workflow that manages the capturing, testing, and release of a single content package of a single content type.</p>
Group package request	<p>A workflow that manages the defining, capturing, testing, and release of a selected group of content packages.</p>
Branch	<p>Storage area in the artifact repository. The area where captured content packages are stored by default is called the master branch.</p> <p>You can add a new branch that allows a content package to be captured and released independently of a package with the same name on another branch. You might create a new branch if you want to allow multiple developers to work on the same content package. In this case, you could create a separate branch for each person. You might also create a new branch if you have a new version of a service that a content package supports that needs to be developed in parallel, while retaining the original version of the service for maintenance updates and bug fixes.</p> <p>You can delete an existing branch and merge two branches together.</p>
Artifact repository	<p>The JFrog Artifactory repository that stores captured content packages. Packages in the repository are version controlled.</p>

IT DevOps

The vRealize Code Stream Management Pack for IT DevOps enables you to provide continuous integration and delivery of software-defined content for the following content package types.

Table 1. Content Package Types

Content Package Type	Description
ASD-Action	Captures a vRealize Automation Advanced Service Designer resource action.
ASD-Blueprint	Captures a vRealize Automation Advanced Service Designer service blueprint.
IAAS-Blueprint	(vSphere endpoints only) Captures a vRealize Automation infrastructure blueprint as well as referenced global profiles.
Linux-Files	Captures files from Linux operating systems.

Orchestrator-Package	Captures a vRealize Orchestrator package containing software files and related information.
Orchestrator-Workflow	Captures a vRealize Orchestrator workflow containing system tasks and dependent actions.
Orchestrator-Action	Captures a vRealize Orchestrator Advanced Service Designer resource action.
Orchestrator-ConfigurationElement	Captures a vRealize Orchestrator Advanced Service Designer configuration element.

Each content endpoint can support more than one content package type depending on endpoint capabilities. For example, a content endpoint for vRealize Automation can support both Advanced Service Design blueprints and Infrastructure as a Service (IaaS) blueprints.

The following table shows the supported package types for vRealize Automation and vRealize Orchestrator content endpoints.

Table 2. Content Endpoint Package Type Support

Content Endpoint	Content Package Type
vRealize Automation	ASD-Action
	ASD-Blueprint
	IaaS-Blueprint
vRealize Orchestrator	Linux-Files
	Orchestrator-Action
	Orchestrator-ConfigurationElement
	Orchestrator-Package
	Orchestrator-Workflow

The management pack ensures that each captured package is correctly stored and retrieved. A package can have multiple content dependencies, which can also be stored with the package. The management pack can preserve package dependencies by ensuring that all dependent content is version controlled and stored with the package. Later the package and all of its dependent content can be released to a production environment in a single process.

The management pack archives multiple artifacts in a single Zip file, as defined by the package type. The management pack also generates a manifest file that identifies the location of every file in an archived package.

Managing Administrators

The tenant administrator specified at installation time is automatically granted access rights to both the vRealize Automation catalog tabs and the vRealize Code Stream tabs in the vRealize Automation console. This allows the tenant administrator to perform content procedures and view pipeline executions. In order for other users to access the management pack features, the tenant administrator must add users to the Content custom group.

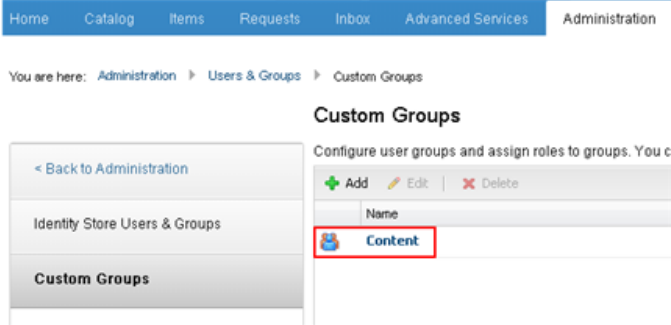
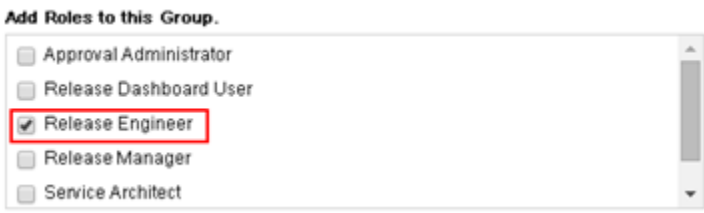
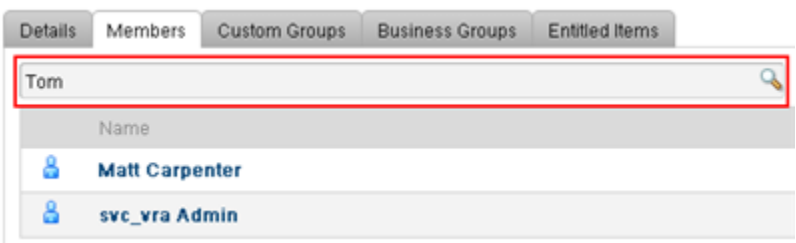
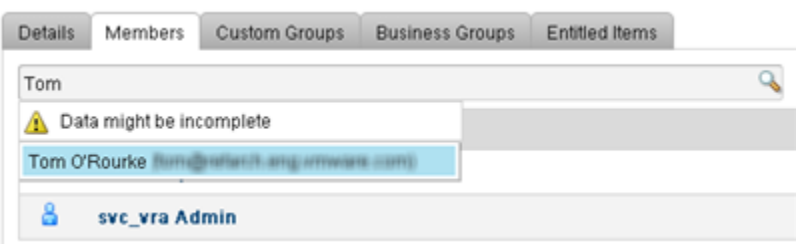
Add User to Content Custom Group

Add users to the Content custom group to allow access to management pack features.

Prerequisite

Log in to the vRealize Automation console as tenant administrator.

Procedure

Step	Reference
1. Click Administration tab. Select Users & Groups > Custom Groups , and click Content group in the right pane.	 <p>The screenshot shows the vRealize Automation console navigation bar with 'Administration' selected. Below it, the breadcrumb 'You are here: Administration > Users & Groups > Custom Groups' is visible. The main content area is titled 'Custom Groups' and includes a sub-header 'Configure user groups and assign roles to groups. You can'. There are three action buttons: 'Add', 'Edit', and 'Delete'. A table below lists the groups, with 'Content' highlighted by a red box.</p>
2. On the Details tab, select Release Engineer , and click Next .	 <p>The screenshot shows a dialog box titled 'Add Roles to this Group.' with a list of roles: 'Approval Administrator', 'Release Dashboard User', 'Release Engineer', 'Release Manager', and 'Service Architect'. The 'Release Engineer' role is selected, indicated by a checked checkbox and a red box around the row.</p>
3. On the Members tab, use the search text box to find the user that you want to add.	 <p>The screenshot shows the 'Members' tab of the 'Content' group. A search box at the top contains the text 'Tom' and is highlighted with a red box. Below the search box, a list of users is displayed: 'Matt Carpenter' and 'svc_vra Admin'.</p>
4. Click the name of the user you are looking for in the search results.	 <p>The screenshot shows the 'Members' tab with the search results expanded. A warning message 'Data might be incomplete' is displayed above the search results. The search results list 'Tom O'Rourke' with a red box around the name and 'svc_vra Admin'.</p>
5. (Optional) Repeat steps 3 and 4 to add additional users.	

6. Click **Update** when you are finished.

Testing Framework

vRealize Code Stream Management provides an integrated testing framework that you can use to validate captured packages as if each package were a piece of code. You can use one or more vRealize Orchestration workflows to validate your package using a process similar to a JUnit testing framework. You have two alternatives for a test environment.

- On the local vRealize Orchestrator appliance where you install the management pack.
- On a remote vRealize Orchestrator host where you add a content endpoint and select **Allow testing on this endpoint**.

Note: The `com.vmware.cse.clm.tests.package` must be imported to the remote vRealize Orchestrator instance for a remote test to succeed. For more information, see the *Installation Guide*.

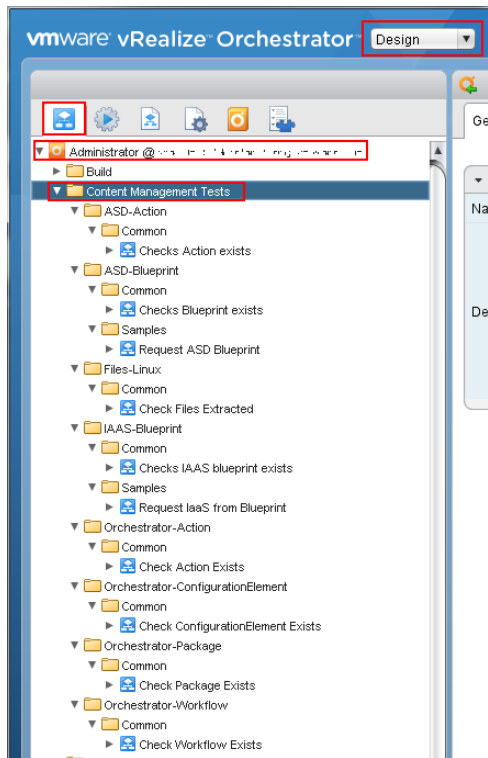
Testing a captured package is a two-step process:

1. You create and name a vRealize Orchestrator workflow, which functions as the test. For more information, see “Developing Workflows” in the VMware vSphere 6.0 Documentation Center at <http://pubs.vmware.com/vsphere-60/index.jsp#com.vmware.vrealize.orchestrator-dev.doc/GUID0BC34D09-9286-4C51-B713-D1069093A38F.html>
2. The test runs during the test stage of the Code Stream pipeline after you select **Run unit tests** on the **Test Details** tab.

Package tests are stored in vRealize Orchestrator that runs vRealize Code Stream Management. To navigate to this directory on the vRealize Orchestrator client, select **Design > Workflows > Administrator > Content Management Tests**. Additional sample workflows are provided in the `Samples` folders. To use these samples to create your own tests, you must copy and rename them.

The following figure shows the Content Management Tests directory in the vRealize Orchestrator client.

Figure 1. Content Management Tests



Any workflows in the `Common` folder under the package type name folder will always run when you test that package type. For example, the `Checks Action exists` workflow will always run when you test an `ASD-Action` package type.

Any workflow in the package type name folder, for example, `Content Management Tests/ASD-Action/` will only run when the workflow name contains the package name. For example, if `myWorkFlow_test1` and `myWorkFlow_test 2` are both in `Content Management Tests/ASD-Action`, both workflows run if the package name is `myWorkFlow`. If either workflow fails, the pipeline stops.

Workflow Example

The following table describes a test workflow with a sample called “Add a DNS A Record.”

Figure 2. Add a DNS A Record

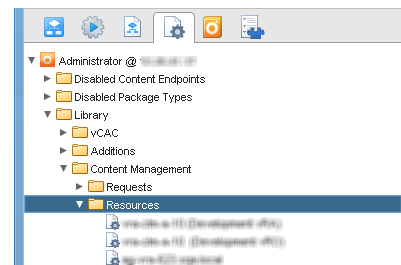
Input	Description	Sample Value
branch:	Branch containing the package being tested.	master
catalogRequestId:	Catalog request id of the single or group package request initiating the test.	18f60dca-df9a-4f57-b962-a1b4faff407

artifactUrls:	URLs for the content and manifest files for the package being tested.	<p>https://vra-clm-b-10.example.com/artifactory/installations-local/content/branches/master/packages/Orchestration/Orchestrator-Workflow/Add a DNS A Record/20/CLM-Package.zip</p> <p>https://vra-clm-b-10.example.com/artifactory/installations-local/content/branches/master/packages/Orchestration/Orchestrator-Workflow/Add a DNS A Record/20/CLM-Manifest.json</p>
version:	The CLM version of the artifact being tested.	20
packageName:	Name of artifact.	Add a DNS A Record
packageId:	Unique id of package being tested (same across versions).	3757a735-da3a-4b7a-abcc-e5fd1dc85369
packageType:	Type of package being tested e.g. ASD-Blueprint	Orchestrator-Workflow
endpointServer:	Server FQDN or IP of the environment where the tests are run.	vro-t1-clm-b-11.example.com
endpointUser:	User name to log in to the test environment	vcoadmin
endpointPassword:	User name password	*****

In addition, the following sources of data are passed into the test workflows.

Input	Description	Example								
Package Metadata	Any data stored in the package version under Artifactory is also passed in.	<p>An ASD-Blueprint called “Add a DNS A Record” for version 1 would be located under <code>/installations-local:content/branches/master/packages/Automation/ASD-Blueprint/Add a DNS A Record/1</code></p> <p>Properties matching “clm_metadata_” are passed in, so a property called “clm_metadata_service” results in a workflow input parameter called “service” with the relevant value.</p> <p>The following metadata properties for an ASD-Blueprint sample are additionally provided in the Test Workflows as input attributes.</p> <table border="1"> <thead> <tr> <th>Property</th> <th>Example Value</th> </tr> </thead> <tbody> <tr> <td>icon</td> <td>18b66845-9753-4f4b-81c9-7de6429907a8.png</td> </tr> <tr> <td>includeDependencies</td> <td>true</td> </tr> <tr> <td>service</td> <td>Infrastructure</td> </tr> </tbody> </table>	Property	Example Value	icon	18b66845-9753-4f4b-81c9-7de6429907a8.png	includeDependencies	true	service	Infrastructure
Property	Example Value									
icon	18b66845-9753-4f4b-81c9-7de6429907a8.png									
includeDependencies	true									
service	Infrastructure									

Resource Configuration Attributes	All attributes on the Configurations under the Content Endpoint (under <code>Library/Content Management/Resources</code>) of type string are passed in as workflow input parameters.
-----------------------------------	---



Test Workflow Requirements

The following are requirements for creating test workflows for a package type in vRealize Orchestrator.

- Tests can contain pertinent logic to test changes made to a package. vRealize Code Stream

Management Pack contains basic tests that check that the content exists on the target endpoint.

The following table shows the parameters used to check that the content exists.

Figure 3. Checks Action Exists Parameters

Name	Type
endpointUser	String
endpointPassword	SecureString
endpointServer	String
packageName	String
tenant	String

- The workflow throws an exception when a test fails.

vRealize Code Stream Management provides basic tests for each package type shipped with the framework. These basic tests can be used as a reference for the construction of your own tests.

Note: All good unit tests should be atomic so that they can stand up and teardown the test once completed. For example, if testing for the removal of a user in a group, the startup for the test will add the user to a group before running the removal test.

When failing a specific test, use a throw statement that is captured by the system. For example:
`throw ("Add some error message");`

The error message appears in the package logs, email notifications, and the vRealize Code Stream pipeline execution.

Managing Content Endpoints

You can add or delete a content endpoint using the vRealize Automation console.

Add a Content Endpoint

Add a content endpoint to an environment to capture or release a content package.

Prerequisites

- Verify that you have the login user name and password for the vRealize Automation or Orchestrator server where you want to add a content endpoint.
- Verify that the vRealize Code Stream Management Pack is installed.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Endpoints**.
2. On **Add Content Endpoint**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Add Content Endpoint** tab, provide the following information for the content endpoint that you are adding. Do one of the following:
 - To add a vRealize Orchestrator content endpoint, provide the following information.

Option	Description	Sample Value
Enter endpoint name	Enter a unique friendly name for this content endpoint.	Dev vRO Server
Select a content category	Select a category for grouping supported package types. For example, selecting Orchestration limits your choice of package types that relate only to orchestration.	Orchestration
Select supported package types	Select the package types that you want this content endpoint to support.	Orchestrator-Action Orchestrator-ConfigurationElement Orchestrator-Package Orchestrator-Workflow
Enter endpoint server FQDN, IP, etc.	Enter the fully qualified server name, IP address, or host name for the content endpoint server.	dev-vro.server.com
Enter endpoint user name	Enter the user name to access this content endpoint server.	admin@example.local
Enter endpoint password	Enter the user name password to access this content endpoint server.	*****
Endpoint tags	(Optional) Enter comma-separated tags associated with the endpoint so that this content endpoint can be used to test or release packages. Tags Best Practice: If each content endpoint shares the same set of tags, for example, "development-london," then selecting this tag for testing or releasing content ensures that each endpoint receives the content defined in the request.	development-london
Allow capturing from this endpoint	Select to allow capturing content packages from this endpoint.	
Allow testing on this endpoint	Select to allow testing content packages on this endpoint.	
Allow releasing to this endpoint	Select to allow releasing content packages to this content endpoint.	
Require approval for test or release	After you select this option, an email approval request is sent to selected users.	

- To add a vRealize Automation content endpoint, provide the following information.

Option	Description	Sample Value
Enter endpoint name	Enter a unique friendly name for this content endpoint.	Dev vRA Server
Select a content category	Select category from drop-down menu to display supported package types.	Automation
Select supported package types	Select the package types that you want this content endpoint to support.	ASD-Action ASD-Blueprint IAAS-Blueprint
Select the attached vRO servers	Select which vRealize Orchestrator servers are connected to this vRealize Automation endpoint. Note: The vRealize Orchestrator servers must exist in your deployment before you create a vRealize Automation content endpoint.	vro-content
Enter an endpoint server FQDN, IP, etc.	Enter the fully qualified name or IP address of the server.	dev-vra.server.com
Enter endpoint user name	Enter the user name password to access this content endpoint server.	admin@example.com
Enter endpoint password	Enter the user name password to access this content endpoint server.	*****
Endpoint tags	(Optional) Enter comma-separated tags associated with the endpoint so that this content endpoint can be used to test or release packages. Tags Best Practice: If each content endpoint shares the same set of tags, for example, "development-london," then selecting this tag for testing or releasing content ensures that each endpoint receives the content defined in the request.	development-london
Allow capturing from this endpoint	Select to enable capturing content on the endpoint.	
Allow testing on this endpoint	Select to enable testing on the endpoint.	
Allow releasing to this endpoint	Select to enable releasing packages to the endpoint.	
Require approval for test or release	Requires approval for testing and releasing to this endpoint. Appears when you select Allow Testing On This Endpoint or Allow Releasing To This Endpoint. When selected, an email approval request is sent to members of the test or release approval group.	
Enter a tenant	Enter the name of a vRealize Automation tenant to connect to.	Tenant1
Enter IaaS AD user	Enter Active Directory admin user name.	
Enter IaaS password	Enter Active Directory admin password.	

5. Click **Next**.
6. On the **Connection Test** tab, verify that the connection to the server is successful.
7. (Optional) Select **Test connection** to retest the connection to the server.
8. Click **Submit**.

Delete One or More Content Endpoints

Delete one or more content endpoints when you no longer need to use them.

Warning: This operation cannot be undone.

Prerequisites

- Verify that one or more content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Endpoints**.
2. On **Delete Content Endpoints**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Delete Content Endpoints** tab, move one or more selected endpoints to the right list box.
5. Click **Submit**.

Managing Content Packages

After you add a content endpoint to one or more environments, you can manage the software-defined content contained in each environment.

Management Pack Actions

You can use the vRealize Automation console to perform the following actions on one or more content packages:

- Capture content from endpoint
- Deploy to test and run unit tests
- Release content to production

The system stores the captured content in the binary repository where it becomes part of a continuous integration and delivery process. Requesting a content package starts a package life cycle management process where each change is version controlled and logged. If necessary, you can delete content packages from the repository and catalog items view

You can perform these three actions separately, together, or in combination. The content package source changes depending on the chosen actions.

Table 3. Content Package Source

Action	Content Package
Capture	From content endpoint to artifact repository
Test	From artifact repository
Release	From artifact repository
Capture	From content endpoint (read only)
Test	
Release	
Test	From artifact repository
Release	

Content Package Approvals

vRealize Code Stream has approval routing that can be configured to run before testing or releasing a package to an endpoint. When testing a package, you must release the package to the endpoint before the test can run. This allows you to prevent content from automatically deploying, which could be intrusive.

Two Active Directory Groups control who can approve and are configured in vRealize Orchestrator configurations: CLM Global Configuration and CLM Configuration.

For information about configuring gating rules, see “Configure Gating Rules” in *Using vRealize Code Stream* at <http://pubs.vmware.com/vrcs-11/topic/com.vmware.ICbase/PDF/using-vrealize-code-stream-11.pdf>

To view the status of a request approval in the vRealize Automation console, select **Inbox > Manual User Action**.

Single and Group Package Requests

You can perform these three actions on a single content package or on a group of content packages. The difference between these options is displayed in the following table.

Table 4. Single and Group Package Requests

Request Type	Contents
Single Package	A single content package of a single type.
Group Package	A group of selected content packages that can be of different types and from different endpoints.

Make a Single Package Request

Make a single package request when you want to manage a single content package of a single type. You can use the following procedures to capture, test, and release a single content package.

To capture, test, and release a group content package, see [Make a Group Content Package Request](#).

Capture Content from Endpoint

Capture a single content package from an endpoint to store it in the artifact repository.

Prerequisites

- Verify that one or more content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Single Package Request**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Choose Actions** tab, select **Capture content from endpoint**, and click **Next**.
5. On the **Content Details** tab, provide the information shown in the following table.

Option	Description
Select branch	Select a branch from the drop-down menu. By default a master branch is used to store a captured content package. You can create additional branches so that multiple developers can work on the same content package. For more information, see Managing Branches .
Select package type	Select the content package type from the drop-down menu that you want to request. The type you select determines how the system captures, tests, or releases a content package.
Select source endpoint	Select the source endpoint for the single package request from the drop-down menu. After you select the source endpoint, all associated content is displayed in the Select content package drop-down menu.
Select content package	Select the content package that you want to capture from the drop-down menu.
Include dependencies	Appears when content package includes dependent components. Select to capture these dependencies along with the content package. Including dependencies ensures that all parts of the component are included when you release the content package to a production environment.

6. Click **Next**.
7. On the **Additional Details** tab, do the following:
 - Enter comments explaining why you are capturing this content in the **Enter a version comment** text box.

- (Optional) Select **Mark new version as release** to indicate that this is a release version. In most cases, you will select this option. You would not select this option if you plan to check in a change that is not complete. You can change the package to a release version once it is ready.

8. Click **Submit**.

Deploy to Test and/or Run Unit Tests

Test the content package to ensure that it is ready for release.

Prerequisites

- Verify that one or more content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Single Package Request**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Choose Actions** tab, select **Deploy to test and/or run unit tests**, and click **Next**.
5. On the **Content Details** tab, provide the information shown in the following table.

Option	Description
Select branch	Select a branch from the drop-down menu. By default a master branch is used to store a captured content package. You can create additional branches so that multiple developers can work on the same content package. For more information, see Managing Branches .
Select package type	Select the content package type from the drop-down menu that you want to request. The type you select determines how the system captures, tests, or releases a content package.
Select content package	Select the content package that you want to test from the drop-down menu.
Show version log	Select to display the contents of the log file.
Version (Optional)	Select the package version that you want to test from the drop-down menu. If you do not select a version, the latest version is used.

6. Click **Next**.

7. On the **Test Details** tab, provide the information shown in the following table, and click **Next**.
Note: You must select either one or more endpoints or tags or a combination of both.

Option	Description
Select test content endpoints	Select one or more content endpoints to specify the environments where the tests are run.
Select test endpoint tags	Select one or more tags to specify the environments where the tests are run. Tags allow many environments to be deployed to at the same time, that is, a staging tag might include many environments.
Selected test endpoints	As you select content endpoints and tags, the selected content endpoints used for testing appear in the Selected test endpoints text box.
Deploy content	Select to deploy content package to the content endpoint before running the tests. If you also select Capture content from endpoint , the content package is in a read-only state. If content is already deployed, you can speed up testing by deselecting this option. If the content package is not deployed, you should leave this option selected.
Run unit tests	Select to run available unit tests on the package. If the unit tests fail, testing stops and the package is not released to a production environment. You should leave this option selected.
Select a unit test server	Select a vRealize Orchestrator unit test server to run the tests. If you select Default, the system uses the default server, which is the vRealize Orchestrator server where you installed the management pack.

8. On the Additional Details tab, click Submit.

Release Content to Production

Release the content package when it is ready for a production environment.

Prerequisites

- Verify that the appropriate content endpoint is added.
- Verify that **Capture content from endpoint** is selected or that the package is a release version.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Single Package Request**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Choose Actions** tab, select **Release content to production**, and click **Next**.

- On the **Content Details** tab, provide the information shown in the following table.

Option	Description
Select branch	Select a branch from the drop-down menu. By default a master branch is used to store a captured content package. You can create additional branches so that multiple developers can work on the same content package. For more information, see Managing Branches .
Select package type	Select the content package type that you want to request from the drop-down menu. The type you select determines how the system captures, tests, or releases a content package.
Select content package	Select the content package that you want to release from the drop-down menu.
Show version log	Select Show version log to display the contents of the log file.
Version (optional)	Select the package version from the drop-down menu that you want to test. If you do not select a version, the latest version is used.

- Click **Next**.
- On the **Release Details** tab, provide the information shown in the following table, and click **Next**.

Option	Description
Select release content endpoints	Select one or more content endpoints to specify the environments where the content package is released.
Select release endpoint tags	(Optional) Select one or more tags to specify the environments where the content package is released. Tags allow many environments to be accessed at the same time, that is, a production tag might include many environments.
Selected release endpoints	As you select the content endpoint or endpoint tags, the relevant content endpoints appear in the Selected release endpoints text box.
Enter a release comment	Enter a release comment that explains why you released this content package.

- On the **Additional Details** tab, click **Submit**.

Make a Group Package Request

You make a group package request when you want to manage a selection of multiple packages, which can consist of different package types and come from different content endpoints. You can use the following procedures to capture, test, and release a group package.

To capture, test, and release a single content package, see [Make a Single Package Request](#).

Capture Content from Endpoints

Capture a group content package when you want to store a group of packages in the artifact repository.

Prerequisites

- Verify that content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Group Package Request**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Choose Actions** tab, select **Capture content from endpoints**, and click **Next**.
5. On the **Content Details** tab, select a branch from the **Select branch** drop-down menu.
By default the master branch is used to store captured content packages. You can create additional branches so that multiple developers can work on the same content package. For more information, see [Managing Branches](#).
6. Perform one of the following procedures to select a group:

- **Select an Existing Group**

Select an existing group when the content packages were previously defined as a group.

- a. Select an existing group from the **Select group** drop-down menu.
- b. (Optional) For **Remove packages from group**, move the selected content packages to the right list box.
- c. (Optional) To add additional content packages to the selected group, provide the information shown in the following table.

Options	Description
Add packages of type	Select a content package type from the drop-down menu.
From source content endpoint	Select a source endpoint for the packages from the drop-down menu.
Select packages	Move the selected content packages to the right list box
Add more packages	Select to add additional packages to the group.

- d. Click **Next**.

- **Define a New Group**

Define a new group when the packages you want have not been defined as a group.

- a. Select **Define a New Group** from the **Select group** drop-down menu.
- b. In the **Save group as (optional)** text box, enter a name for your new group.
Note: You can skip this step if you want to perform content operations on content packages without saving the packages as a group.

- c. To add content packages to the group, provide the information shown in the following table.

Options	Description
Add packages of type	Select a package type from the drop-down menu.
From source endpoint	Select a source endpoint from the drop-down menu for the content packages.
Select packages	Move the selected content packages to the right list box.
Add more packages	Select to add additional content packages to the new group.

- d. Click **Next**.
- e. On the **Additional Details** tab, do the following:
- Enter comments in the **Enter a version comment** text box explaining why you are capturing this content.
 - (Optional) Select **Mark new version as release** to indicate that this is a release version. In most cases, you will select this option. You would not select this option if you plan to check in a change that is not complete. You can mark the package as a release version once it is ready.
- f. Click Submit.

Deploy to Test and/or Run Unit Tests

Use the following procedure to test multiple content packages.

Prerequisites

- Verify that content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Group Package Request**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Choose Actions** tab, select **Deploy to test and/or run unit tests**, and click **Next**.
5. On the **Content Details** tab, select a branch from the **Select branch** drop-down menu. By default the master branch is used to store captured content packages. You can create additional branches so that multiple developers can work on the same content package. For more information, see [Managing Branches](#).
6. Select an existing group from the **Select group** drop-down menu.

7. (Optional) For **Remove packages from group**, move the selected content packages to the right list box.
8. (Optional) To add packages to this group, provide the information shown in the following table.

Options	Description
Add packages of type	Select a content package type from the drop-down menu.
From source endpoint	Select a source endpoint for the packages from the drop-down menu.
Select packages	Move the selected content packages to the right list box.
Add more packages	Select to add additional content packages to this group.

9. Click **Next**.

On the **Test Details** tab, provide the information shown in the following table, and click **Next**.

Note: You must select either one or more endpoints or tags or a combination of both.

Detail	Description
Select test content endpoints	Select one or more Content Endpoints to specify the environments where the tests are run.
Select test endpoint tags	(Optional) Select one or more tags to specify the environments where the tests are run. Tags allow many environments to be deployed to at once, that is, a staging or production tag might include many environments.
Selected test endpoints	As you select the content endpoints and tags, the related content endpoints appear in the Selected test endpoints text box.
Deploy content	Select to deploy content package to the content endpoint before running the tests. If you also select Capture content from endpoint , the content package is in a read-only state. If content is already deployed, you can speed up testing by deselecting this option. If the content package is not deployed, you should leave this option selected.
Run unit tests	Select to run available unit tests on the package. If the unit tests fail, testing stops and the package is not released to a production environment.
Select a unit test server	Select a vRealize Orchestrator unit test server to run the tests. If you select Default, the system uses the default server, which is the vRealize Orchestrator server where you installed the management pack.

10. On the **Additional Details** tab, click **Submit**.

Release Content to Production

Use the following procedure to release multiple content packages to a production environment.

Prerequisites

- Verify that the content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Group Package Request**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Choose Actions** tab, select **Release content to production**, and click **Next**.
5. On the **Content Details** tab, select a branch from the **Select branch** drop-down menu. By default a master branch is used to store captured content packages. You can create additional branches so that multiple developers can work on the same content packages. For more information, see [Managing Branches](#).
6. Select an existing group from the **Select group** drop-down menu.
7. (Optional) For **Remove packages from group**, move the selected content packages to the right list box.
8. (Optional) To add packages to this group, provide the information shown in the following table.

Options	Description
Add packages of type	Select a content package type from the drop-down menu.
From source content endpoint	Select a source content endpoint from the drop-down menu for the content packages.
Select packages	Move the selected content packages to the right list box.
Add more packages	Select to add additional content packages to the group.

9. Click **Next**.

10. On the **Release Details** tab, provide the information shown in the following table, and click **Next**.

Detail	Description
Select release content endpoints	Select one or more content endpoints to specify the environments where the tests are run.
Select release endpoint tags	(Optional) Select one or more tags to specify the environments where the tests are run. Tags allow many environments to be deployed at the same time, that is, production might include many environments.
Selected release endpoints	As you select the endpoint and endpoint tags, the related content endpoints appear in the Selected release endpoints text box.
Enter a release comment	Enter a comment about why you are releasing this content to the selected endpoint.

11. On the **Additional Details** tab, click **Submit**.

Delete Packages

Delete one or more content packages when you no longer need to use them.

Note: This operation does not delete packages from the endpoints.

Warning: This operation cannot be undone.

Prerequisites

- Verify that one or more content endpoints are added.
- Verify that content packages are present in the deployment.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Packages**.
2. On **Delete Packages**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Delete Packages** tab, select a package type from the **Filter by content package type** drop-down menu.
5. For **Select content packages to delete**, move the selected content packages to the right list box.
6. Click **Submit**.

Delete Group Definition

Delete an existing group definition if you no longer need to use it.

Warning: This operation cannot be undone.

Prerequisites

- Verify that one or more content endpoints are added.
- Verify that group content packages are present in the deployment.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

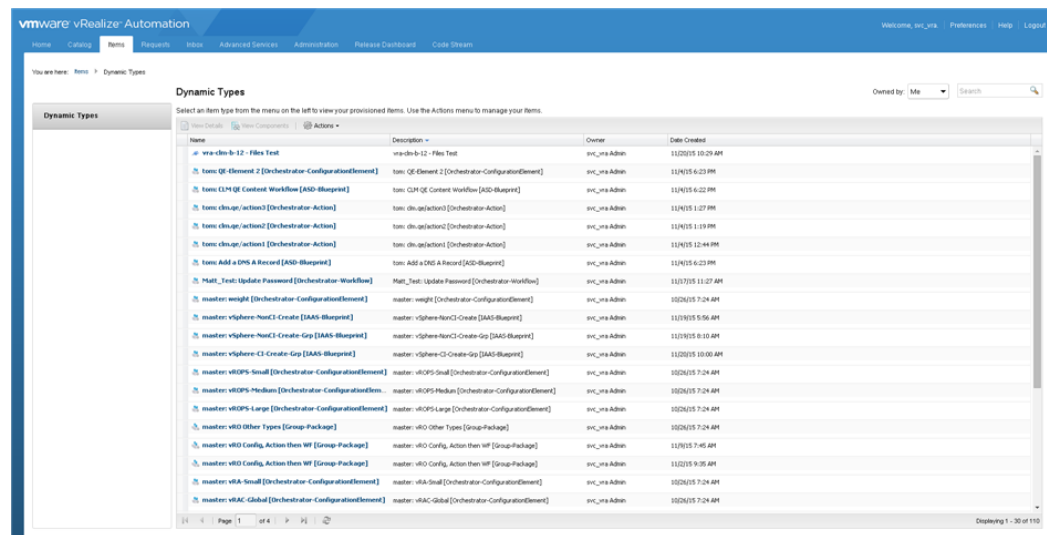
1. Select **Catalog > Packages**.
2. On **Delete Group Definition**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Delete Group Definition** tab, select a branch from the drop-down menu.
5. Select a group name from the drop-down menu.
6. Click **Submit**.

Managing Captured or Released Content Packages

View previously captured or released content packages when you want to perform a content package action. In the vRealize Automation console, select **Items > Dynamic Types**.

The following figure shows captured or released packages on the Dynamic Types page.

Figure 4. Previously Captured or Released Content Packages

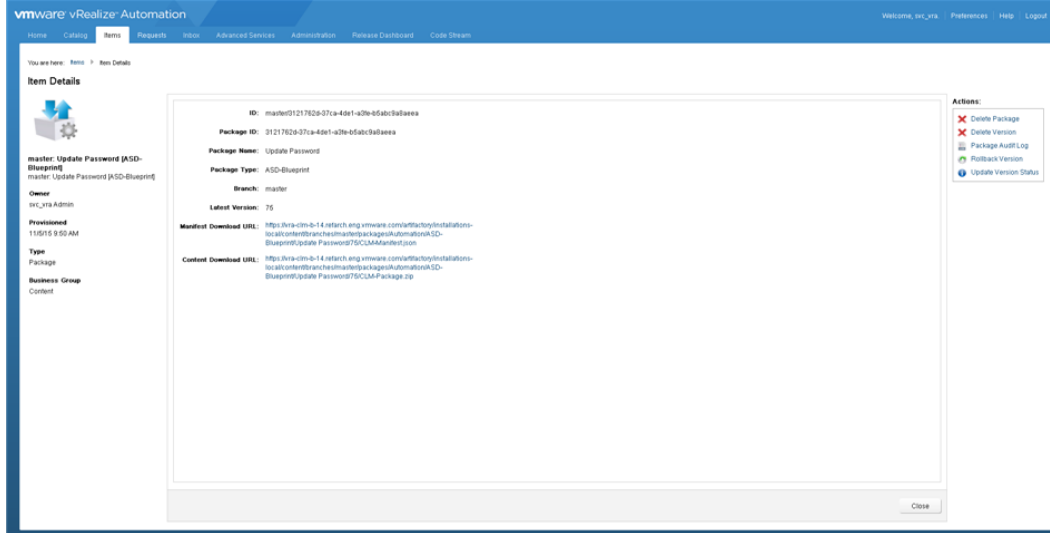


Click an item to view the details of one of the content packages on this page.

VMware vRealize Code Stream Management Pack for IT DevOps Operations Guide

The following figure shows an Item Details page with details about a single package. Here you can download the manifest or the content package file. You can also perform actions on the content package.

Figure 5. Item Details



Content Package Actions

The following table displays the actions you can perform on a content package from the Item Details page.

Figure 6. Package Actions

Option	Description
Delete Package	Removes a package and associated artifacts from the repository.
Delete Version	Removes a specified package version from the repository.
Package Audit Log	Shows a detailed list of all package versions.
Rollback Version	Rolls back to a different package version.
Update Version Status	Changes the release and test status of a package version so that a test version can be promoted to a release version or a release version can be demoted to a test version.

Managing Branches

A branch is an area in the artifact repository separate from the main working area, called the master branch.

By default, captured content is stored in the master branch. You can create a new branch if you want to allow developers to work on the same content package. In this case, you could create a separate branch for each person. You can also create a new branch if you have a new version of a service that a content package supports that needs to be developed in parallel, while retaining the older version of the service for maintenance updates and bug fixes. A package on one branch can be captured and released independently of a package with the same name on another branch.

You can copy the content on one branch to another branch. If there are multiple versions of a content package on the source branch, only the latest version is copied to the target branch and is marked version 1 in the target branch. You can also delete an existing branch and merge two branches together. During a merge, the system only copies packages from the source branch that are not present in the target branch. A merge takes place at a binary level, not at a code level, and so if a package on the source branch is newer than the same package on the target branch, the source branch package is copied to the target branch. This happens whether or not the package type contains code or is binary.

You can use the following procedures to manage branches.

Create a New Branch

Create a new branch when you want a separate area to store one or more content packages.

Prerequisites

- Verify that content endpoints are added.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Branches**.
2. On **Create Branch**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Create Branch** tab, enter a name for the new branch in the text box.
Note: This name must not contain any spaces.
5. Do one of the following:
 - Copy previously captured content.
 - a. Select **Copy content from another branch**.
 - b. Select a branch from the **Select a branch to copy from** drop-down menu.
 - c. Select **Create from release versions only** if you want to copy only packages designated as release versions.
 - Continue with step 6.
6. Enter a description of the new branch in the **Comment** text box.
7. Click **Submit**.

Delete a Branch

Delete a branch when you no longer need to use it.

Warning: This operation cannot be undone.

Prerequisites

- Verify that content endpoints are added.
- Verify that there are branches in addition to the master branch.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Branches**.
2. On **Delete Branch**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Delete Branch** tab, select a branch from the **Select a branch to delete** drop-down menu.
5. Enter the reason for deleting the branch in the **Comment** text box.
6. Click **Submit**.

Merge a Branch

Merge two branches when you want to add the content packages from another branch.

Prerequisites

- Verify that content endpoints are added.
- Verify that there are branches in addition to the master branch.
- Log in to the vRealize Automation console as a member of the Content custom group.

Procedure

1. Select **Catalog > Branches**.
2. On **Merge Branch**, click **Request**.
3. On the **Information** tab, click **Next**.
4. On the **Merge Branch** tab, select the source branch for the merge from the **Select a source branch** drop-down menu.
5. Select the target branch for the merge from the **Select a target branch** drop-down menu.
6. (Optional) Select **Only merge versions marked as release** to exclude packages that are not marked as release.
7. (Optional) Select **Run a test merge** to see the results of a merge without making any changes to the target branch.
8. (Optional) Select **Send report to all users** to inform notifications distribution list members of the merge results.
9. (Optional) Select **Send report to selected users** and enter comma-separated email addresses of users selected to receive the merge results.

10. (Optional) Select **Delete source after merge** to delete the source branch after it is merged with the target branch. This option is not available when you select **Run a test merge**.
11. Enter a comment in the **Comment** text box explaining why you are merging these branches.
12. Click **Submit**.

Pipeline Process Flow Overview

The vRealize Code Stream pipeline process flow uses vRealize Orchestrator actions that vary according to task. The following sections present the steps in the pipeline process flow for capture, testing, and release. Also included are the process flow specifics for the different content package types.

Pipeline Process Flows for Content Package Request Actions

The vRealize Code Stream pipeline performs the following steps for each content package request.

Capture Process Flow

1. vRealize Code Stream pipeline creates a new package version.
2. vRealize Code Stream pipeline executes capture workflow according to each package type. An example of a specific workflow type is seen in vRealize Orchestrator at this location: `/Library/Content Management/Package Types/Orchestrator/Orchestrator-Workflows/Capture vRO Workflow`.
3. Content export is run on the Content Transfer Server. This is typically done using CloudClient, but the actual process used depends on the package type. Content is exported to the Content Transfer Server in a temporary export folder located at `/storage/clm/export/<packageid>-v<version number>-Capture`
4. The vRealize Code Stream pipeline creates an artifact from the captured data.
5. The system imports the artifact to the JFrog's Artifactory artifact repository.
6. The system cleans the Content Transfer Server and removes the temporary capture folder.

Test Process Flow

1. If **Deploy to test and/or run unit tests** is selected in the single or group package request, the vRealize Code Stream pipeline executes the specific release workflow for each package type so that the content deploys to one or more the test content endpoints. An example of the specific workflow type executed for Orchestrator-Workflows is seen in vRealize Orchestrator at this location: `/Library/Content Management/Package Types/Orchestrator/Orchestrator-Workflows/Release vRO Workflows`.
Note: If **Capture content from endpoint** is also selected, the content is always deployed.
2. If **Run unit tests** is selected in the single or group package request, the vRealize Code Stream pipeline executes the specific test workflows for each package type so that the content is tested on the test content endpoint. For example, any workflows matching a valid package name in the specific Package Type folder in `/Content Management Tests` is executed.
3. If all tests pass, then the testing stage in the pipeline is successful.

4. If any tests fail (that is, throws an exception that is caught and logged by the system), then the pipeline fails.

Release Process Flow

The vRealize Code Stream pipeline executes the specific release workflow for each package type so that the content deploys to one or more release content endpoints.

An example of the specific workflow is seen in vRealize Orchestrator at this location:
Workflows/Administrator/Library/Content Management/Package
Types/Orchestrator/Orchestrator-Workflows/Release vRO Workflows.

Pipeline Process Flow Prerequisites for Package Types

This section provides details about how the system manages the pipeline process for the various package types.

vRealize Orchestrator Package Types

- Orchestrator-Workflow
- Orchestrator-Action
- Orchestrator-ConfigurationElement

Prerequisite

The only prerequisite for vRealize Orchestrator package types is correct permissions.

Note: For the Orchestrator-Workflow type, it is a best practice to have custom workflows outside the `Library` folder. If you have workflows in the `Library` folder, and would like to list them, you can modify the `ignoreFolders` attribute for each content endpoint in the vRealize Orchestrator client. If you select `ignoreFolders`, the system takes longer to list all the workflows in the client.

IAAS-Blueprint Package Type

Prerequisites

- You need valid IAAS Administrator permissions with business group level access.
- You also need valid IAAS Windows domain permissions so that the ODATA API interface running on the Windows IAAS machine can be used.
- Reservations need to be created, but this is outside the scope of this management pack.
- Business groups must be created first.
- Machine prefixes must be created first and are assigned according to the source package.

- A catalog entitlement must be created according to the business group.
- Only vSphere IAAS blueprints are supported for version 1.0 of the management pack.
- You must run a vSphere Clone data collection on the environment that you are deploying so that the vSphere templates are available.

Additional Flow Process Information

Capture

The following items are included when capturing content:

- IAAS blueprint definition
- Build profiles
- Storage reservation policies
- Host reservation policies
- Catalog icon
- Catalog service

Test and Release

- If not present, the service that the IAAS-Blueprint is assigned to in the vRealize Automation catalog is created.
- You can require that the IAAS-Blueprint be added to a valid entitlement for it to be accessible within the catalog.
- The catalog item is imported and made active to the service.
- The icon is assigned.

ASD Package Types

- ASD-Blueprint
- ASD-Action

Prerequisites

- Valid service architect administrator permissions.
- vRealize Orchestrator content endpoints must be assigned to a vRealize Automation content endpoint so that vRealize Orchestrator dependencies can be captured.
- A catalog entitlement needs to be created according to the business group.

Additional Flow Process Information

Capture

The following items are included when capturing content:

- Advanced Service Designer Blueprint/Form definition

- vRO Actions used within the form
- vRO Workflow assigned to the Advanced Service Blueprint and any child actions/workflows associated
- Catalog Icon
- Catalog Service

Test and Release

- If not present, the service that the ASD-Blueprint or ASD-Action is assigned to in the vRealize Automation catalog is created.
- You can require that the ASD-Blueprint or ASD-Action is added to a valid entitlement for it to be accessible within the catalog.
- When deploying the content to an endpoint, all dependencies are imported first:
 - All vRealize Orchestrator actions
 - All vRealize Orchestrator workflows
 - The catalog item is imported and made active to the service
 - The icon is assigned

Files-Linux Package Type

Prerequisite

Valid credentials are required to connect remotely through SSH by using password authentication.

Additional Flow Process Information

Capture

When capturing content from a Linux machine, the search path is created. This path is stored in the `basePaths` attribute for the content endpoint. The attribute is located in the vRealize Orchestrator client under `Configurations/Library/Content Management/Resources`.

Test and Release

Permissions are set the same as the source configuration according to the default tar behavior.

Troubleshooting

Package Persistence

The vRealize Code Stream Management Pack stores all packages in JFrog's Artifactory under the top level folder `installations-local`. All packages are grouped by their package type and each package has multiple versions. The total number of packages stored is determined by `maxTestVersions` and `maxReleaseVersions` settings in package type configuration element in vRealize Orchestrator. If disk space becomes a problem, you can adjust the default policies for the number of old versions stored. This is selected in the package type configuration element settings.

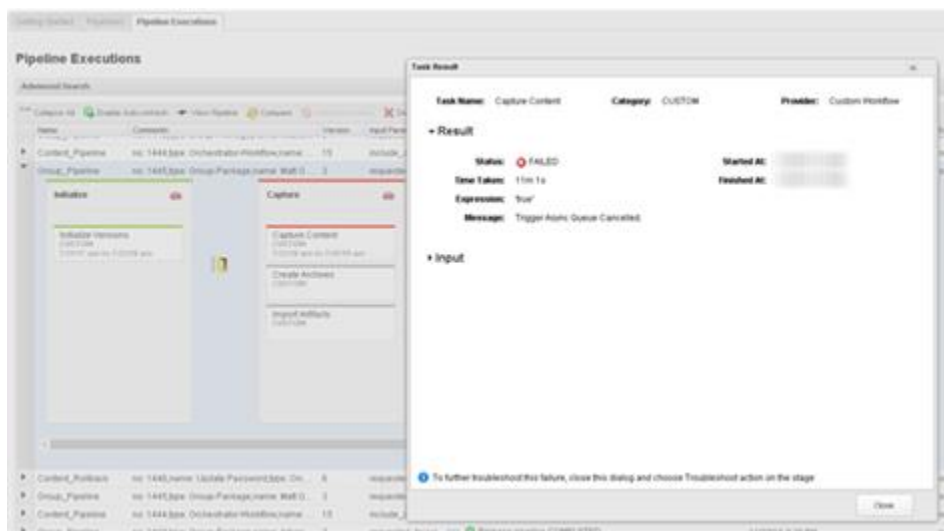
vRealize Code Stream Pipeline Executions

vRealize Code Stream Management uses vRealize Code Stream to manage the Capture, Test, Release, and Rollback stages. For information about Code Stream, see *Using vRealize Code Stream* at <http://pubs.vmware.com/vrcs-11/topic/com.vmware.ICbase/PDF/using-vrealize-code-stream-11.pdf>

When a content capture is submitted either through a single package or group package request, a vRealize Code Stream pipeline is started. You can view the process on the vRealize Automation console by selecting **Code Stream > Pipeline Executions**, and clicking a task to view the result.

The following figure shows a failed pipeline task resulting from a rejected approval.

Figure 7. A Failed Pipeline



If the pipeline fails at any point, the single package or group package request fails. The following table shows the difference between a pipeline failure on the first request and a failure on a subsequent request.

Table 5. Pipeline Failure Results

Failure Circumstance	Result
A capture, test and release request is being performed for the first time, which will create a new content package. A failure in the pipeline takes place at any time.	<p>The content package item on the Items tab in the vRealize Automation console is not available.</p> <p>There are no package logs available in the console, but logs are available in the repository.</p>

A capture, test and release request is being performed using an existing content package. A failure in the pipeline takes place at any time.

The content package item on the **Items** tab in the vRealize Automation console is available.

Package logs are available in the console making it possible for the package to be managed with logging.

All tasks in the vRealize Code Stream call v Realize Orchestration workflows. For specific details about the pipeline process, see [Pipeline Process Flow Overview](#).

If you need to do troubleshooting that requires additional output from vRealize Orchestrator workflows, use the following procedure to enable debug mode.


Enable Debug Mode

Enable debug mode when you want to generate additional output for troubleshooting.

Prerequisite

Log in to the vRealize Orchestrator client as a member of the Content custom group.



Procedure

1. Select **Design**. Click **Configurations** icon, and select **Library > Content Management > Global Configuration**.
2. Select **Configuration**, and click the **Attributes** tab in the right pane.
3. Select the **Edit** icon ().
4. For `debugMode`, select **Yes**.
5. Click **Save and close**.

Workflow Executions

After you enable debug mode, review the workflow executions in the following table.

Table 6. Troubleshooting Workflow Executions

Location	Reference	Contents
/Library/Content Management/Framework/Advanced Services/Service Blueprints		Workflows executed by the vRealize Automation Advanced Service Designer Catalog.
/Library/Content Management/Framework/Advanced Services/Resource Actions		The right-click action workflows that are executed by the vRealize Authentication Advanced Service Designer Catalog.

/Library/Content Management/
Package Types



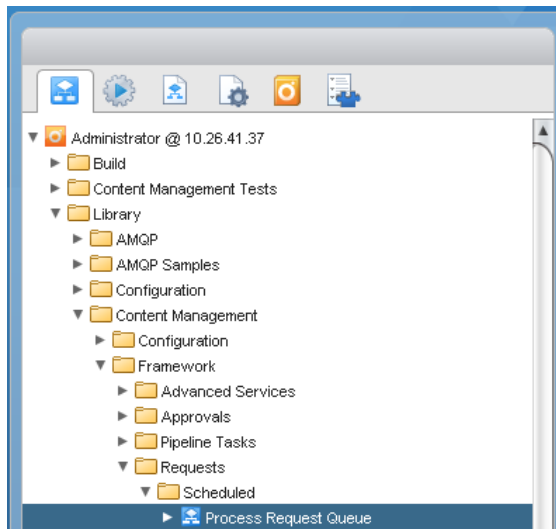
Various workflows that are executed when a capture, test, and release is performed for each package type.

Scheduled Workflow

The Process Requests Queue workflow found in the vRealize Orchestrator client at `Library/Content Management/Framework/Requests/Scheduled` needs to be running on a schedule to process requests that users make in the vRealize Automation console. The default is every 30 seconds.

The following figure shows the location of the Process Requests Queue.

Figure 8. Process Request Queue



To temporarily suspend the workflow, select **Run** and right-click the process name in the left pane. Select **Suspend task** to stop the process and Resume to restart the process. The following figure shows the location of the menu.

Figure 9. Suspend Task

