Abstract

Table of Contents

MAKING OPEN SOURCE MORE INCLUSIVE ......................................................... 7

CHAPTER 1. INTRODUCTION TO RED HAT SOFTWARE CERTIFICATION PROGRAM ........................................... 8
  1.1. THE RED HAT CERTIFICATION PROGRAM OVERVIEW ................................................................. 8
  1.2. GETTING HELP AND GIVING FEEDBACK ......................................................................................... 8

CHAPTER 2. ONBOARDING CERTIFICATION PARTNERS ................................................................. 10
  2.1. ONBOARDING EXISTING CERTIFICATION PARTNERS ..................................................................... 10
  2.2. ONBOARDING NEW CERTIFICATION PARTNERS ............................................................................ 10

PART I. NON-CONTAINER CERTIFICATION ................................................................. 12

CHAPTER 3. INTRODUCTION TO NON-CONTAINERIZED PRODUCT CERTIFICATION ............................... 13

CHAPTER 4. CERTIFICATION WORKFLOW FOR NON-CONTAINERIZED APPLICATION ............................... 14
  4.1. CERTIFICATION ONBOARDING AND OPENING YOUR FIRST PROJECT ........................................... 14
  4.2. CERTIFICATION TESTING ........................................................................................................... 15
  4.3. PUBLISHING THE CERTIFIED APPLICATION .............................................................................. 15

CHAPTER 5. CREATING A NON-CONTAINERIZED APPLICATION PROJECT ............................................... 17

CHAPTER 6. CONFIGURING THE NON-CONTAINERIZED APPLICATION PROJECT ...................................... 18
  6.1. COMPLETE THE PRE-CERTIFICATION CHECKLIST ........................................................................ 18
  6.1.1. Provide details about your certification ....................................................................................... 18
  6.1.2. Complete your company profile ................................................................................................ 18
  6.1.3. Validate the functionality of your product on Red Hat Enterprise Linux .................................... 19
  6.1.4. Creating a product listing .......................................................................................................... 19
  6.1.5. Attach a completed product listing .............................................................................................. 21
  6.2. MANAGING PROJECT SETTINGS ................................................................................................ 22

CHAPTER 7. SETTING UP THE TEST ENVIRONMENT FOR NON-CONTAINERIZED APPLICATION TESTING ...... 24
  7.1. SETTING UP A SYSTEM THAT ACTS AS A SYSTEM UNDER TEST ..................................................... 24

CHAPTER 8. DOWNLOADING THE TEST PLAN FROM RED HAT CERTIFICATION PORTAL ........................... 26

CHAPTER 9. RUNNING CERTIFICATION TESTS BY USING CLI AND DOWNLOADING THE RESULTS FILE .... 27
  9.1. RUNNING THE CERTIFICATION TESTS USING CLI ......................................................................... 27
  9.2. REVIEWING AND DOWNLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN ...................... 27

CHAPTER 10. UPLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN TO RED HAT CERTIFICATION PORTAL .................................................................................................................. 28

CHAPTER 11. RECERTIFICATION ................................................................................................. 29

CHAPTER 12. PUBLISHING THE CERTIFIED APPLICATION ......................................................................... 30

APPENDIX A. RUNNING THE CERTIFICATION TESTS BY USING COCKPIT ............................................. 31
  A.1. CONFIGURING THE SYSTEM AND RUNNING TESTS BY USING COCKPIT ...................................... 31
  A.1.1. Setting up the Cockpit server ...................................................................................................... 31
  A.1.2. Adding system under test to Cockpit ........................................................................................... 32
  A.1.3. Getting authorization on the Red Hat SSO network .................................................................... 32
  A.1.4. Downloading test plans in Cockpit from Red Hat certification portal ....................................... 33
  A.1.5. Using the test plan to prepare the system under test for testing .................................................. 33
  A.1.6. Running the certification tests using Cockpit ............................................................................. 34
  A.1.7. Reviewing and downloading the results file of the executed test plan ...................................... 34
A.1.8. Submitting the test results from Cockpit to the Red Hat Certification Portal | 35

PART II. CONTAINER CERTIFICATION

CHAPTER 13. WORKING WITH CONTAINERS

13.1. INTRODUCTION TO CONTAINERS | 37
13.2. CONTAINER CERTIFICATION WORKFLOW
   13.2.1. Certification on-boarding and opening your first project | 38
   13.2.2. Certification testing | 39
   13.2.3. Publishing the certified container on the Red Hat Ecosystem Catalog | 40
13.3. TESTING MULTI-ARCH CONTAINER CERTIFICATION USING PREFLIGHT
   13.3.1. Building and pushing multi-arch container images using Podman | 41

CHAPTER 14. CREATING A CONTAINER APPLICATION PROJECT | 42

CHAPTER 15. CONFIGURING THE CONTAINER PROJECT

15.1. COMPLETE THE PRE-CERTIFICATION CHECKLIST
   15.1.1. Accept the Red Hat Container Appendix | 44
   15.1.2. Provide details about your container | 44
   15.1.3. Complete your company profile | 45
   15.1.4. Complete export control questionnaire | 45
   15.1.5. Submit your container for verification | 45
   15.1.6. Attach a completed product listing
      15.1.6.1. Creating a new product listing | 46
      15.1.6.2. Attaching a container project to an existing product listing | 46
      15.1.6.3. Attaching multiple container projects to a single product listing | 46
      15.1.6.4. Removing attached container project from an existing product listing | 47
15.2. VIEWING THE IMAGE TEST RESULTS | 47
15.3. MANAGING THE CONTAINER PROJECT SETTINGS | 48

CHAPTER 16. RUNNING THE CERTIFICATION TEST SUITE | 50

CHAPTER 17. PUBLISHING THE CERTIFIED CONTAINER | 52

PART III. OPERATOR CERTIFICATION

CHAPTER 18. WORKING WITH OPERATORS

18.1. INTRODUCTION TO OPERATORS | 54
18.2. CERTIFICATION WORKFLOW FOR OPERATORS
   18.2.1. Certification on-boarding | 54
   18.2.2. Certification testing | 54
   18.2.3. Publishing the certified Operator on the Red Hat Ecosystem Catalog | 55

CHAPTER 19. CREATING AN OPERATOR BUNDLE PROJECT | 57

CHAPTER 20. CONFIGURING THE OPERATOR BUNDLE

20.1. COMPLETE THE PRE-CERTIFICATION CHECKLIST
   20.1.1. Accept the Red Hat Container Appendix | 59
   20.1.2. Provide repository details used for pulling your container | 59
   20.1.3. Complete your company profile | 60
   20.1.4. Test your operator bundle data and submit a pull request
      20.1.4.1. Test locally with OpenShift | 60
      20.1.4.2. Test with Red Hat’s hosted pipeline | 60
   20.1.5. Attach a completed product listing | 61
   20.1.6. Validate the functionality of your CNI or CSI on Red Hat OpenShift | 61
20.2. VIEWING THE TEST RESULTS | 62
CHAPTER 21. RUNNING THE CERTIFICATION TEST SUITE LOCALLY

21.1. ADDING YOUR OPERATOR BUNDLE
   21.1.1. If you have certified this operator before -
   21.1.2. If you are newly certifying this operator -

21.2. FORKING THE REPOSITORY

21.3. INSTALLING THE OPENSHIFT OPERATOR PIPELINE
   21.3.1. Automated process
      21.3.1.1. Prerequisites
      21.3.1.2. Installing the pipeline through an Operator
      21.3.1.3. Executing the pipeline
   21.3.2. Manual process
      21.3.2.1. Installing the OpenShift Pipeline Operator
      21.3.2.2. Configuring the OpenShift (oc) CLI tool
      21.3.2.3. Creating an OpenShift Project
      21.3.2.4. Adding the kubeconfig secret
      21.3.2.5. Importing Operator from Red Hat Catalog
      21.3.2.6. Installing the certification pipeline dependencies
      21.3.2.7. Configuring the repository for submitting the certification results
         21.3.2.7.1. Adding GitHub API Token
         21.3.2.7.2. Adding Red Hat Container API access key
         21.3.2.7.3. Enabling digest pinning
         21.3.2.7.4. Using a private container registry

21.4. EXECUTE THE OPENSHIFT OPERATOR PIPELINE
   21.4.1. Running the Minimal pipeline
   21.4.2. Running the pipeline with image digest pinning
   21.4.3. Running the pipeline with a private container registry

21.5. SUBMIT CERTIFICATION RESULTS
   21.5.1. Submitting test results from the minimal pipeline
   21.5.2. Submitting test results with image digest pinning
   21.5.3. Submitting test results from a private container registry
   21.5.4. Submitting test results with image digest pinning and from a private container registry

CHAPTER 22. RUNNING THE CERTIFICATION SUITE WITH RED HAT HOSTED PIPELINE

22.1. FORKING THE REPOSITORY

22.2. ADDING YOUR OPERATOR BUNDLE
   22.2.1. If you have certified this operator before -
   22.2.2. If you are newly certifying this operator -

22.3. CREATING A PULL REQUEST
   22.3.1. Guidelines to follow

CHAPTER 23. PUBLISHING THE CERTIFIED OPERATOR

CHAPTER 24. TROUBLESHOOTING GUIDELINES

APPENDIX B. HELM AND ANSIBLE OPERATORS

PART IV. HELM CHART CERTIFICATION

CHAPTER 25. WORKING WITH HELM CHARTS
   25.1. INTRODUCTION TO HELM CHARTS
   25.2. CERTIFICATION WORKFLOW FOR HELM CHARTS
      25.2.1. Certification on-boarding
### 25.2.2. Certification testing
- Publishing the certified Helm chart on the Red Hat Ecosystem Catalog

### 26.2.2. Certification testing

### 26.2.3. Publishing the certified Helm chart on the Red Hat Ecosystem Catalog

### CHAPTER 26. VALIDATING HELM CHARTS FOR CERTIFICATION

#### 26.1. PREPARING THE TEST ENVIRONMENT

#### 26.2. RUNNING THE HELM CHART-VERIFIER TOOL
- By using Podman or Docker
  - Configuring the timeout option
  - Saving the report
- By using the binary file

### CHAPTER 27. CREATING A HELM CHART PROJECT

### CHAPTER 28. CONFIGURING THE HELM CHART PROJECT

#### 28.1. COMPLETE PRE-CERTIFICATION CHECKLIST
- Complete your company profile
- Provide details for your Helm chart
- Submit your chart through a pull request in GitHub
- Attach a completed product listing

#### 28.2. MANAGING PROJECT SETTINGS

### CHAPTER 29. SUBMITTING YOUR HELM CHART FOR CERTIFICATION

#### 29.1. SUBMITTING A HELM CHART WITHOUT THE CHART VERIFICATION REPORT
- Chart as a tarball
- Chart in a directory

#### 29.2. SUBMITTING A CHART VERIFICATION REPORT WITHOUT THE HELM CHART
- For submitting a signed report
- For submitting a report for a signed chart

#### 29.3. SUBMITTING A CHART VERIFICATION REPORT ALONG WITH THE HELM CHART
- For submitting a signed report
- For submitting a signed Helm chart

#### 29.4. SUMMARY OF CERTIFICATION SUBMISSION OPTIONS

#### 29.5. VERIFICATION STEPS

### CHAPTER 30. PUBLISHING THE CERTIFIED HELM CHART

### PART V. FUNCTIONAL CERTIFICATION FOR OPENSHIFT BADGES: CNF, CNI, CSI

### CHAPTER 31. CNF CERTIFICATION AND VENDOR VALIDATION

#### 31.1. WORKING WITH CLOUD-NATIVE NETWORK FUNCTION (CNF) CERTIFICATION
- Introduction to Cloud-native Network Function
- Certification workflow for CNF
  - Certification onboarding and opening your first project
  - Completing the checklist
  - Publishing the CNF product listing on the Red Hat Ecosystem Catalog

#### 31.2. CREATING A CNF PROJECT

#### 31.3. CONFIGURING THE CNF PROJECT
- Complete the checklist
  - Complete the pre-publication checklist for Vendor Validation
  - Complete your company profile
  - Provide details about your validation
  - Attach a completed product listing
  - Validate the functionality of your CNF on Red Hat OpenShift
  - Complete the Certification checklist to certify the Vendor Validated CNF project
31.3.2. Managing Project settings
31.4. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG
31.5. RECERTIFYING A CNF PACKAGE

CHAPTER 32. CNI CERTIFICATION
32.1. WORKING WITH CONTAINER NETWORK INTERFACE (CNI) CERTIFICATION
  32.1.1. Introduction to Container Network Interface
  32.1.2. Certification workflow for CNI
    32.1.2.1. Certification on-boarding and creating your first project
    32.1.2.2. Certification testing
    32.1.2.3. Publishing the CNI product listing on the Red Hat Ecosystem Catalog
  32.2. CREATING A CNI PROJECT
  32.3. CONFIGURING THE CNI PROJECT
  32.4. WORKING WITH THE OPENS SHIFT OPERATOR PIPELINE
  32.5. CONFIGURING YOUR TEST ENVIRONMENT FOR RUNNING THE CNI TESTS
  32.6. RUNNING THE CNI TESTS
  32.7. SUBMITTING YOUR CNI OPERATOR FOR CERTIFICATION
  32.8. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

CHAPTER 33. CSI CERTIFICATION
33.1. WORKING WITH CONTAINER STORAGE INTERFACE (CSI) CERTIFICATION
  33.1.1. Introduction to Container Storage Interface
  33.1.2. Certification workflow for CSI
    33.1.2.1. Certification on-boarding and creating your first project
    33.1.2.2. Certification testing
    33.1.2.3. Publishing the CSI product listing on the Red Hat Ecosystem Catalog
  33.2. CREATING A CSI PROJECT
  33.3. CONFIGURING THE CSI PROJECT
  33.4. WORKING WITH THE OPENS SHIFT OPERATOR PIPELINE
  33.5. CONFIGURING YOUR TEST ENVIRONMENT FOR RUNNING THE CSI TESTS
  33.6. ACCESSING THE CSI CERTIFICATION TESTS
  33.7. SETTING UP THE CSI TEST PARAMETERS
  33.8. RUNNING THE CSI TESTS
  33.9. SUBMITTING CSI TEST RESULTS
  33.10. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG
MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code and documentation. We are beginning with these four terms: master, slave, blacklist, and whitelist. Due to the enormity of this endeavor, these changes will be gradually implemented over upcoming releases. For more details on making our language more inclusive, see our CTO Chris Wright’s message.
CHAPTER 1. INTRODUCTION TO RED HAT SOFTWARE CERTIFICATION PROGRAM

Use this guide to certify and distribute your software application product on the Red Hat Enterprise Linux and Red Hat OpenShift platforms.

1.1. THE RED HAT CERTIFICATION PROGRAM OVERVIEW

The Red Hat software certification program ensures compatibility of your software application products targeting Red Hat Enterprise Linux and Red Hat OpenShift as the deployment platform.

The program has four main elements:

- **Project**: The online workflow where the progress and status of certification requests are tracked and reported.
- **Test suite**: Tests implemented as an integrated pipeline for software application products undergoing certification.
- **Publication**:
  - **Non-containerized products**: Certified traditional, non-containerized products are published on the Red Hat Ecosystem Catalog.
  - **Containers**: Certified containers are published on the Red Hat Ecosystem Catalog.
  - **Operators**: Certified Operators are published on the Red Hat Ecosystem Catalog and in the embedded OperatorHub.
  - **Helm Charts**: Certified Helm Charts are published on the Red Hat Ecosystem Catalog.
  - **Cloud-native Network Functions (CNFs)**: Vendor Validated and Certified CNF projects are attached to the product listings and are published on the Red Hat Ecosystem Catalog.
- **Support**: A joint support relationship between you and Red Hat to ensure customer success when deploying certified software application products.

1.2. GETTING HELP AND GIVING FEEDBACK

For any questions related to the Red Hat certification toolset, certification process, or procedure described in this documentation, refer to the KB Articles, Red Hat Customer Portal, and Red Hat Partner Connect.

**NOTE**

To receive Red Hat product assistance, it is necessary to have the required product entitlements or subscriptions, which may be separate from the partner program and certification program memberships.

**Opening a support case**

To open a support case, see How do I open and manage a support case?

To open a support case for any certification issue, complete the Support Case Form for Partner Acceleration Desk with special attention to the following fields:
- From the Issue Category, select **Product Certification**.
- From the Product field, select the required product.
- From the **Product Version** field, select the version on which your product or application is being certified.
- In the **Problem Statement** field, type a problem statement or issue or feedback using the following format:

  **{Partner Certification} (The Issue/Problem or Feedback)**

  - Replace *(The Issue/Problem or Feedback)* with either the issue or problem faced in the certification process or Red Hat product or feedback on the certification toolset or documentation.

  For example: **{Partner Certification}** Error occurred while submitting certification test results using the Red Hat Certification application.

**NOTE**

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

**Additional resources**

- To know more about the software certification program and platforms, see [Red Hat certified software](#).
- For a one-stop solution on all your certification needs, see [Red Hat Software Certification Quick Start Guide](#).
- For more information about program requirements and policies, see [Red Hat OpenShift Software Certification Policy Guide](#) and [Red Hat Enterprise Linux Software Certification Policy Guide](#).
CHAPTER 2. ONBOARDING CERTIFICATION PARTNERS

Use the Red Hat Partner Connect Portal to create a new account if you are a new partner, or use your existing Red Hat account if you are a current partner to onboard with Red Hat for certifying your products.

2.1. ONBOARDING EXISTING CERTIFICATION PARTNERS

As an existing partner you could be:

- A member of the one-to-many EPM program who has some degree of representation on the EPM team, but does not have any assistance with the certification process.
  OR

- A member fully managed by the EPM team in the traditional manner with a dedicated EPM team member who is assigned to manage the partner, including questions about the certification requests.

**NOTE**

If you think your company has an existing Red Hat account but are not sure who is the Organization Administrator for your company, email connect@redhat.com to add you to your company’s existing account.

**Prerequisites**

You have an existing Red Hat account.

**Procedure**

1. Access Red Hat Partner Connect and click Log in.

2. From the Certified technology portal section, click Log in for technology partners.

3. Enter your Red Hat login or email address and click Next.
   Then, use either of the following options:
   a. Log in with company single sign-on
   b. Log in with Red Hat account

4. From the menu bar on the header, click your avatar to view the account details.
   a. If an account number is associated with your account, then log in to the Red Hat Partner Connect, to proceed with the certification process.
   b. If an account number is not associated with your account, then first contact the Red Hat global customer service team to raise a request for creating a new account number. After that, log in to the Red Hat Partner Connect to proceed with the certification process.

2.2. ONBOARDING NEW CERTIFICATION PARTNERS

Creating a new Red Hat account is the first step in onboarding new certification partners.

1. Access Red Hat Partner Connect and click Log in.
2. From the **Certified technology portal** section, click **Log in for technology partners**

3. Click **Register for a Red Hat account**

4. Enter the following details to create a new Red Hat account:
   
   a. Select **Corporate** in the **Account Type** field.
      
      If you have created a Corporate type account and require an account number, contact the **Red Hat global customer service** team.

      **NOTE**
      
      Ensure that you create a company account and not a personal account. The account created during this step is also used to sign in to the Red Hat Ecosystem Catalog when working with certification requests.

   b. Choose a Red Hat login and password.

      **IMPORTANT**
      
      If your login ID is associated with multiple accounts, then do not use your contact email as the login ID as this can cause issues during login. Also, you cannot change your login ID once created.

   c. Enter your **Personal information** and **Company information**.

   d. Click **Create My Account**
      
      A new Red Hat account is created. Log in to the **Red Hat Partner Connect**, to proceed with the certification process.
PART I. NON-CONTAINER CERTIFICATION
CHAPTER 3. INTRODUCTION TO NON-CONTAINERIZED PRODUCT CERTIFICATION

The Red Hat Software certification program for traditional, non-containerized products helps Independent Software Vendors (ISV) to build, certify and distribute their application software on systems and server environments running Red Hat Enterprise Linux (RHEL) in a jointly supported customer environment. A strong working knowledge of RHEL is required.
CHAPTER 4. CERTIFICATION WORKFLOW FOR NON-CONTAINERIZED APPLICATION

NOTE

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

The following diagram gives an overview of certification workflow for non-containerized applications:

Figure 4.1. Certification workflow for non-containerized application

Task Summary

The certification workflow includes the following three primary stages-

1. Section 4.1, “Certification onboarding and opening your first project”
2. Section 4.2, “Certification testing”
3. Section 4.3, “Publishing the certified application”

4.1. CERTIFICATION ONBOARDING AND OPENING YOUR FIRST PROJECT

Prerequisites

Verify the functionality of your product on the target Red Hat platform, in addition to the specific certification testing requirements. If running your product on the targeted Red Hat platform results in a substandard experience, then you must resolve the issues prior to certification.

The Red Hat Partner Acceleration Desk (PAD) is a Products and Technologies level partner help desk...
service that allows our (prospective) technology partners a central location to ask non-technical questions pertaining to Red Hat offerings, partner programs, product certification, engagement process, and so on.

See PAD - How to open & manage PAD cases, to open a PAD ticket.

Through the Partner Subscriptions program, Red Hat offers free, not-for-resale software subscriptions that you can use to validate your product on the target Red Hat platform. To request access to the program, follow the instructions on the Partner Subscriptions site.

Procedure

Perform the steps outlined for the certification onboarding:

1. Join the Red Hat Connect for Technology Partner Program.
2. Add a non-containerized software product to certify.
3. Complete your company profile.
4. Complete the pre-certification checklist.
5. Create a non-containerized application project for your product.

Additional resources

For detailed instructions about creating your first application project, see Creating an application project.

4.2. CERTIFICATION TESTING

Follow these high-level steps to run a certification test:

- Log in to the Red Hat Certification portal.
- Download the test plan.
- Configure the system under test (SUT) for running the tests.
- Download the test plan to our SUT.
- Run the certification tests on your system.
- Review and upload the test results to the certification portal.

Additional resources

For detailed instructions about certification testing, see Setting up the test environment for non-containerized application testing.

4.3. PUBLISHING THE CERTIFIED APPLICATION

When you complete all the certification checks successfully, you can submit the test results to Red Hat. Upon successful validation, you can publish your product on the Red Hat Ecosystem Catalog.
Additional resources

For detailed instructions about publishing your application, see Publishing the certified application.
CHAPTER 5. CREATING A NON-CONTAINERIZED APPLICATION PROJECT

Procedure

Follow the steps to create a non-containerized application project:

1. Log in to Red Hat Partner Connect portal. The Access the partner portals web page displays.

2. Navigate to the Certified technology portal tile and click Log in for technology partners.

3. Enter the login credentials and click Login. The Red Hat Partner Connect web page displays.

4. On the page header, select Product certification and click Manage certification projects. My Work web page displays the Product Listings and Certification Projects, if available.

5. Click Create Project.

6. In the What platform do you want to certify? dialog box, select your desired platform and click Next. For example, select the Red Hat Enterprise Linux radio button for creating a non-containerized project.

7. In the What do you want to certify? dialog box, select Non-containerized application radio button and click Next.

8. On the Create non-containerized Red Hat Enterprise Linux project web page, provide the following details to create your project.

   IMPORTANT

   You cannot change the RHEL version after you have created the project.

   a. Project Name - Enter the project name. This name is not published and is only for internal use.

   b. Red Hat Enterprise Linux (RHEL) Version- Select the specific RHEL version on which you wish you to certify your non-containerized application.

9. Click Create project.
CHAPTER 6. CONFIGURING THE NON-CONTAINERIZED APPLICATION PROJECT

After the non-containerized project is created, the newly created application project web page displays.

The new application project web page comprises of the following tabs:

- **Overview** - Contains the pre-certification checklist.
- **Settings** - Allows you to view the configured project details.

Additionally, to perform the following actions on the application project, click the **Actions** menu on the non-containerized application project web page:

- Open Support Case
- Archive Project

6.1. COMPLETE THE PRE-CERTIFICATION CHECKLIST

For certifying a non-containerized application, you must complete the pre-certification checklist and then publish your application project. The **Overview** tab of the project contains the pre-certification checklist. The pre-certification checklist consists of a series of tasks that you must complete to certify and publish your application project.

Before you publish your application project, perform the following tasks in the checklist:

1. Provide details about your certification
2. Complete your company profile
3. Validate the functionality of your product on Red Hat Enterprise Linux
4. Creating a product listing
5. Attach a completed product listing

6.1.1. Provide details about your certification

1. Navigate to Provide details about your certification tile to configure the project details that are displayed in the catalog. This will allow users to pull your application image.
2. Click Review. You are navigated to the xref:[Settings] tab.
3. Edit the required project details.
4. Click Save.

6.1.2. Complete your company profile

Ensure that your company profile is up-to-date. This information is published in the catalog along with your certified non-containerized application product.

To verify the information,
1. Navigate to Complete your company profile tile.
2. Click Review in your checklist.
3. To make any changes, click Edit.
4. After updating your details, click Save.

6.1.3. Validate the functionality of your product on Red Hat Enterprise Linux

This feature allows you to perform the following functions:

- run the Red Hat Certification Tool locally
- download the test plan
- share the test results with the Red Hat certification team and
- interact with the certification team, if required.

To validate the functionality of your product on the Red Hat Enterprise Linux:

1. Navigate to Validate the functionality of your product on Red Hat Enterprise Linux tile and click Start. A new project gets created in the Red Hat Certification portal, and you are redirected to the appropriate project portal page.

6.1.4. Creating a product listing

NOTE
This section is applicable for Operator and Helm certification: Ensure that the product listing is completed in the pipeline for the Operator testing.

The Product Listing functions as a source of detailed technical information of your product and sales tool for showcasing your product’s features and advantages to potential customers.

Fields marked with an asterisk (*) are mandatory.

Red Hat recommends completing as many optional fields as feasible across all tabs in the listing. Providing more information enables our shared customers to make well-informed choices regarding the suitability of your product for their specific requirements.

To enter information in marketing-related fields, collaborate with your product manager, marketing representative, or other individuals well-versed with the product.

Procedure

1. Log in to the Red Hat Partner Connect Certified Technology Portal.
2. On the header bar, click Product certification > Manage products.
3. Click Create Product.
   A Create new listing dialog appears.
4. Enter the Product name.
5. Optional: Enter the major version you want to certify, and select one of the Listing Type radio buttons. A new page with your Product name opens.

**General** tab:

This tab provides basic details of the product, including product name and description.

1. Enter the Product Name.
2. Optional: Upload the Product Logo according to the defined guidelines.
3. Enter a Brief description and a Long description.
4. Optional: Enter the Marketing contact email address.

**Features & Benefits** tab:

This tab lists the important features of your product.

1. Optional: Enter the Title and Description.
2. Optional: Click + Add new feature, and click Save.

**Quick start & Config** tab:

This tab lets you add links to any quick start guide or configuration document to help customers deploy and start using your product.

Optional: Enter Quick start & configuration instructions and click Save.

**Linked resources** tab:

This tab lets you add links to supporting documentation to help our customers use your product. The information is mapped to and appears under the Documentation section on the product’s catalog page.

While Red Hat requires at least three resources, we encourage you to add more, if available.

Select the Type drop-down menu, and enter the Title and Description of the resource.

**FAQs** tab:

This tab lets you add frequently asked questions and answers, in case your product’s purpose, operation, installation, or other attributes have a bit of complex and intricate details.

Enter Question and Answer.

**Support** tab:

This tab lets you provide contact information of your Support team.

1. Enter the Support description, Support web site, Support phone number, and Support email address.
2. Click Save.

**Legal** tab:
This tab lets you provide the product-related license and policy information.

1. Enter the **License Agreement URL** for the product and **Privacy Policy URL**.

2. Click **Save**.

**SEO tab:**

This tab enables improved discoverability of your product for our mutual customers, enhancing visibility both within the Red Hat Ecosystem Catalog search and on internet search engines. Providing a higher number of search aliases (key and value pairs) will increase the discoverability of your product.

1. Select the **Product Category**.

2. Enter the **Key** and **Value** to set up Search aliases, and click **Save**.

**Certification Projects tab:**

This tab allows you to attach a project for each component of your product. The tab displays any previously attached projects linked to your Product Listing. Up to 10 projects are listed on a page - to view more, use the navigation button.

Alternatively, to attach a project to the Product Listing, you can complete the **Attach a completed product listing** item in the Pre-Certification Checklist of a Container, Operator, or Helm Chart project.

**NOTE**

One project can be applicable and attached to multiple products listings.

**NOTE**

All attached projects must be completed and published before the product listing can be published.

1. To attach a new project to the Product Listing, click **Attach Project**. A list of projects displays.

2. Select the checkbox beside the Project Name to select a project, and click **Attach**. All the selected projects will get attached to the Product Listing.

You can use additional options to select projects such as, Select none, Select page, and Select All by clicking the checkbox drop-down button beside the Search by Name search box.

Alternatively, to search for specific projects, type the project name in the **Search by Name** field.

### 6.1.5. Attach a completed product listing

This feature allows you to either create a new product listing, or to attach the project to an existing RHEL product listing for your new project.

1. Navigate to **Attach a completed product listing**

2. From the **Select method** drop-down menu, select **Attach or edit**. The **Attach product listing** page displays.
3. Decide whether you want to attach your project to an existing product listing or if you want to create a new product listing:
   a. To attach your project to an existing product listing:
      i. From the Related product listing section, click Select a product listing drop-down arrow to select the required product listing.
      ii. Click Save.
   b. To create a new product listing:
      i. Click Create new product listing.
      ii. In the Product Name text box, enter the required product name.
      iii. Click Save.

4. From the Select method drop-down menu, click View product listing to navigate to the new product listing and enter all the required product listing details.

5. Click Save.

NOTE
Make sure to complete all the items on the Pre-certification checklist except Validate the functionality of your product on Red Hat Enterprise Linux step, before submitting your application for certification.

After completing all the steps, a green check mark appears beside the tiles to indicate that configuration is complete.

6.2. MANAGING PROJECT SETTINGS

You can view and edit the project details through the Settings tab.

Enter the required project details in the following fields:

- **Project name** - Enter the project name. This name is not published and is only for internal use.
- **Red Hat Enterprise Linux (RHEL) Version** - Specifies the RHEL version on which you wish to certify your non-containerized application.

IMPORTANT
You cannot change the RHEL version after you have created the project.

- **Technical contact email address** - Enter your project maintainers email addresses separated by a comma.
- **Click Save.**
NOTE

All the fields marked with asterisk * are required and must be completed before you can proceed with the certification.
CHAPTER 7. SETTING UP THE TEST ENVIRONMENT FOR NON-CONTAINERIZED APPLICATION TESTING

The first step towards certifying your product is setting up the environment where you can run the tests.

The test environment consists of a system in which all the certification tests are run.

7.1. SETTING UP A SYSTEM THAT ACTS AS A SYSTEM UNDER TEST

A system on which the product that needs certification is installed or configured is referred to as the system under test (SUT).

Prerequisites

- The SUT has RHEL version 8 or 9 installed. For convenience, Red Hat provides kickstart files to install the SUT’s operating system. Follow the instructions in the file that is appropriate for your system before launching the installation process.

Procedure

1. Configure the Red Hat Certification repository:
   a. Use your RHN credentials to register your system using Red Hat Subscription Management:
      
      ```
      $ subscription-manager register
      ```
   b. Display the list of available subscriptions for your system:
      
      ```
      $ subscription-manager list --available*
      ```
   c. Search for the subscription which provides the Red Hat Certification (for RHEL Server) repository and make a note of the subscription and its Pool ID.
   d. Attach the subscription to your system:
      
      ```
      $ subscription-manager attach --pool=<pool_ID>
      ```
      Replace the pool_ID with the Pool ID of the subscription.

   NOTE

   You don’t have to attach the subscription to your system, if you enable the option Simple content access for Red Hat Subscription Management. For more details, see How do I enable Simple Content Access for Red Hat Subscription Management?

   e. Subscribe to the Red Hat Certification channel:
      i. On RHEL 8:
         
         ```
         $ subscription-manager repos --enable=cert-1-for-rhel-8-<HOSTTYPE>-rpms
         ```
         Replace HOSTTYPE with the system architecture. To find out the system architecture, run:
         ```bash
         hostnamectl
         ```
$ uname -m

Example:

$ subscription-manager repos --enable=cert-1-for-rhel-8-x86_64-rpms

ii. On RHEL 9:

$ subscription-manager repos --enable=cert-1-for-rhel-9-<HOSTTYPE>-rpms

Replace HOSTTYPE with the system architecture. To find out the system architecture, run

$ uname -m

Example:

$ subscription-manager repos --enable=cert-1-for-rhel-9-x86_64-rpms

f. Install the software test suite package:

$ dnf install redhat-certification-software
CHAPTER 8. DOWNLOADING THE TEST PLAN FROM RED HAT CERTIFICATION PORTAL

Procedure

1. Log in to Red Hat Certification portal.

2. Search for the case number related to your product certification, and copy it.

3. Click Cases → enter the product case number.

4. Optional: Click Test Plans.
   The test plan displays a list of components that will be tested during the test run.

5. Click Download Test Plan.
   - If you plan to use CLI to run the tests, see Running certification tests by using CLI and downloading the results file.
   - Otherwise, if you plan to use Cockpit to run the tests, see the Appendix.
CHAPTER 9. RUNNING CERTIFICATION TESTS BY USING CLI AND DOWNLOADING THE RESULTS FILE

To run the certification tests by using CLI you must download the test plan to the SUT. After running the tests, download the results and review them.

9.1. RUNNING THE CERTIFICATION TESTS USING CLI

Procedure
1. Run the following command:

   # rhcert-run

2. When prompted, choose whether to run each test by typing yes or no. You can also run particular tests from the list by typing select.

9.2. REVIEWING AND DOWNLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN

Procedure
1. Run the following command:

   # rhcert-save

2. Download the results file by using the rhcert-save command to your local system.

Additional resources
For more details on setting up and using cockpit for running the certification tests, see the Appendix.
CHAPTER 10. UPLOADING THE RESULTS FILE OF THE EXECUTED TEST PLAN TO RED HAT CERTIFICATION PORTAL

Prerequisites

- You have downloaded the test results file from either the SUT or Cockpit.

Procedure

1. Log in to Red Hat Certification portal.
2. On the homepage, enter the product case number in the search bar. Select the case number from the list that is displayed.
3. On the Summary tab, under the Files section, click Upload.

Red Hat will review the submitted test results file and suggest the next steps.

Additional resources

For more information, visit Red Hat Certification portal.
CHAPTER 11. RECERTIFICATION

As an existing partner you must recertify your application:

- on every major and minor release of the Red Hat Enterprise Linux
- on every major and minor release of your application

**NOTE**

To recertify your application, it is mandatory to create a new certification request for recertification.

To recertify your application, submit a new certification request through the Red Hat Certification tool or create a new project in the Red Hat Partner Connect. Run the certification tests on SUT and proceed with the regular certification workflow, like a new certification.
CHAPTER 12. PUBLISHING THE CERTIFIED APPLICATION

After submitting your test results through the Red Hat certification portal, your application is scanned for vulnerabilities within the project. When the scanning is completed, the publish button will be enabled for your application on the Product Listings page. After filling-in all the necessary information, click the publish button, your application will be available on the Red Hat Ecosystem Catalog.

IMPORTANT

The Red Hat software certification does not conduct testing of the Partner’s product in how it functions or performs on the chosen platform. Any and all aspects of the certification candidate product’s quality assurance remains the partner’s sole responsibility.
Using cockpit to run the certification tests is optional.

Use the following procedure to set up and run the certification tests by using cockpit.

**A.1. CONFIGURING THE SYSTEM AND RUNNING TESTS BY USING COCKPIT**

To run the certification tests by using Cockpit you need to upload the test plan to the SUT first. After running the tests, download the results and review them.

*NOTE*

Although it is not mandatory, Red Hat recommends you to configure and use Cockpit for the certification process. Configuring cockpit greatly helps you to manage and monitor the certification process on the SUT.

**A.1.1. Setting up the Cockpit server**

*Cockpit* is a RHEL tool that lets you change the configuration of your systems as well as monitor their resources from a user-friendly web-based interface.

*NOTE*

- You must set up Cockpit either on the SUT or a new system.
- Ensure that the Cockpit has access to SUT.

**Prerequisites**

- The Cockpit server has RHEL version 8 or 9 installed.
- You have installed the Cockpit plugin on your system.
- You have enabled the Cockpit service.

**Procedure**

1. Log in to the system where you installed Cockpit.
2. Install the Cockpit RPM provided by the Red Hat Certification team.
   
   ```bash
   # dnf install redhat-certification-cockpit
   ```

   By default, Cockpit runs on port 9090.

**Additional resources**
For more information on installing and configuring Cockpit, see Getting Started using the RHEL web console on RHEL 8, Getting Started using the RHEL web console on RHEL 9 and Introducing Cockpit.

A.1.2. Adding system under test to Cockpit

Adding the system under test (SUT) to Cockpit lets them communicate by using passwordless SSH.

Prerequisites

- You have the IP address or hostname of the SUT.

Procedure

1. Enter `http://<Cockpit_system_IP>:9090/` in your browser to launch the Cockpit web application.
2. Enter the username and password, and then click Login.
3. Click the down-arrow on the logged-in cockpit user name→Add new host.
   The dialog box displays.
4. In the Host field, enter the IP address or hostname of the system.
5. In the User name field, enter the name you want to assign to this system.
6. Optional: Select the predefined color or select a new color of your choice for the host added.
7. Click Add.
8. Click Accept key and connect to let Cockpit communicate with the SUT through passwordless SSH.
9. Enter the Password.
10. Select the Authorize SSH Key checkbox.
11. Click Log in.

Verification

On the left panel, click Tools →Red Hat Certification
Verify that the SUT you just added displays below the Hosts section on the right.

A.1.3. Getting authorization on the Red Hat SSO network

Procedure

1. Enter `http://<Cockpit_system_IP>:9090/` in your browser’s address bar to launch the Cockpit web application.
2. Enter the username and password, and then click Login.
3. Select Tools → Red Hat Certification in the left panel.
4. On the Cockpit homepage, click Authorize, to establish connectivity with the Red Hat system.
   The Log in to your Red Hat account page displays.
5. Enter your credentials and click **Next**.  
The Grant access to rhcert-cwe page displays.

6. Click **Grant access**. A confirmation message displays a successful device login. You are now connected to the Cockpit web application.

A.1.4. Downloading test plans in Cockpit from Red Hat certification portal

For Non-authorized or limited access users:

- To download the test plan, see [Downloading the test plan from Red Hat Certification portal](#).

For authorized users:

**Procedure**

1. Enter **http://<Cockpit_system_IP>:9090/** in your browser’s address bar to launch the Cockpit web application.

2. Enter the username and password, and then click **Login**.

3. Select **Tools → Red Hat Certification** in the left panel.

4. Click the **Test Plans** tab. A list of **Recent Certification Support Cases** will appear.

5. Click **Download Test Plan**. A message displays confirming the successful addition of the test plan.

6. The downloaded test plan will be listed under the **File Name** of the **Test Plan Files** section.

A.1.5. Using the test plan to prepare the system under test for testing

Provisioning the system under test (SUT) includes the following operations:

- setting up passwordless SSH communication with cockpit
- installing the required packages on your system based on the certification type
- creating a final test plan to run, which is a list of common tests taken from both the test plan provided by Red Hat and tests generated on discovering the system requirements.

For instance, required software packages will be installed if the test plan is designed for certifying a software product.

**Prerequisites**

- You have downloaded the test plan provided by Red Hat.

**Procedure**

1. Enter **http://<Cockpit_system_IP>:9090/** in your browser address bar to launch the Cockpit web application.

2. Enter the username and password, and then click **Login**.

3. Select **Tools → Red Hat Certification** in the left panel.
4. Click the Hosts tab, and then click the host under test on which you want to run the tests.

5. Click Provision.
   A dialog box appears.
   
a. Click Upload, and then select the new test plan .xml file. Then, click Next. A successful upload message is displayed.
   Optionally, if you want to reuse the previously uploaded test plan, then select it again to reupload.

   **NOTE**
   During the certification process, if you receive a redesigned test plan for the ongoing product certification, then you can upload it following the previous step. However, you must run `rhcert-clean all` in the Terminal tab before proceeding.

   b. In the Role field, select System under test and click Submit. By default, the file is uploaded to path: `/var/rhcert/plans/<testplanfile.xml>`

A.1.6. Running the certification tests using Cockpit

**Prerequisites**

- You have prepared the system under test.

**Procedure**

1. Enter `http://<Cockpit_system_IP>:9090/` in your browser address bar to launch the Cockpit web application.

2. Enter the username and password, and click Login.

3. Select Tools → Red Hat Certification in the left panel.

4. Click the Hosts tab and click on the host on which you want to run the tests.

5. Click the Terminal tab and select Run.
   A list of recommended tests based on the test plan uploaded displays. The final test plan to run is a list of common tests taken from both the test plan provided by Red Hat and tests generated on discovering the system requirements.

6. When prompted, choose whether to run each test by typing yes or no.
   You can also run particular tests from the list by typing select.

A.1.7. Reviewing and downloading the results file of the executed test plan

**Procedure**

1. Enter `http://<Cockpit_system_IP>:9090/` in your browser address bar to launch the Cockpit web application.

2. Enter the username and password, and then click Login.
3. Select **Tools → Red Hat Certification** in the left panel.

4. Click the **Result Files** tab to view the test results generated.
   
   a. Optional: Click **Preview** to view the results of each test.
   
   b. Click **Download** beside the result files. By default, the result file is saved as `/var/rhcert/save/hostname-date-time.xml`.

---

**A.1.8. Submitting the test results from Cockpit to the Red Hat Certification Portal**

**Procedure**

1. Enter `http://<Cockpit_system_IP>:9090/` in your browser’s address bar to launch the Cockpit web application.

2. Enter the username and password, and then click **Login**.

3. Select **Tools → Red Hat Certification** in the left panel.

4. Click the **Result Files** tab and select the case number from the displayed list.
   
   a. For the authorized users click **Submit**. A message displays confirming the successful upload of the test result file.
   
   b. For non-authorized users see, **Uploading the results file of the executed test plan to Red Hat Certification portal**.

The test result file of the executed test plan will be uploaded to the Red Hat Certification portal.
PART II. CONTAINER CERTIFICATION
13.1. INTRODUCTION TO CONTAINERS

Containers include all the necessary components like libraries, frameworks, and other additional dependencies that are isolated and self-sufficient within their own executable. A Red Hat container certification ensures supportability of both the operating system and the application layers. It provides enhanced security by vulnerability scanning and health grading of the Red Hat components, and lifecycle commitment whenever the Red Hat or partner components are updated.

However, containers running in privileged mode, or privileged containers, stretch their boundaries and interact with their host to run commands or access the host’s resources. For example, a container that reads or writes to a filesystem mounted on the host must run in privileged mode.

Privileged containers might create a security risk. A compromised privileged container might also compromise its host and the integrity of the environment as a whole.

Moreover, privileged containers are susceptible to incompatibilities with the host as operating system interfaces such as commands, libraries, ABI, and APIs might change or deprecate over time. This can put privileged containers at risk of interacting with the host in an unsupported way.

Containers must run in unprivileged mode unless approved by Red Hat during the certification process as described in the policy guide.

You must ensure that your containers can run on any supported hosts in the customer’s environment. Red Hat encourages you to adopt a continuous integration model that lets you test your containers with public betas or earlier versions of Red Hat products to maximize compatibility.

13.2. CONTAINER CERTIFICATION WORKFLOW

NOTE

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

The following diagram gives an overview of container certification workflow:
Figure 13.1. Container certification workflow

Task Summary

The certification workflow includes the following three primary stages-

1. Section 13.2.1, “Certification on-boarding and opening your first project”
2. Section 4.2, “Certification testing”
3. Section 13.2.3, “Publishing the certified container on the Red Hat Ecosystem Catalog”

13.2.1. Certification on-boarding and opening your first project

Prerequisites

Verify the functionality of your product on the target Red Hat platform, in addition to the specific certification testing requirements. If running your product on the targeted Red Hat platform results in a substandard experience then you must resolve the issues prior to certification.

The Red Hat Partner Acceleration Desk (PAD) is a Products and Technologies level partner help desk service that allows our (prospective) technology partners a central location to ask non-technical questions pertaining to Red Hat offerings, partner programs, product certification, engagement process, and so on.

See PAD - How to open & manage PAD cases, to open a PAD ticket.

Through the Partner Subscriptions program, Red Hat offers free, not-for-resale software subscriptions that you can use to validate your product on the target Red Hat platform. To request access to the program, follow the instructions on the Partner Subscriptions site.

You must construct your container images so that they meet the certification criteria and policy. For more details, see image content requirements.

Procedure
Follow these high-level steps to certify your containerized software:

1. Join the Red Hat Partner Connect for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Complete your company profile.
4. Create your certification project by selecting your desired platform, for example - Red Hat OpenShift and then choose Container Image.
5. Complete the pre-certification checklist including the export compliance questionnaire for your container images, if applicable.

Additional resources

For detailed instructions about creating your first container project, see Creating a container application project.

13.2.2. Certification testing

Follow these high-level steps to run a certification test:

1. Build your container image.
2. Upload your container image to your chosen registry. You can choose any registry of your choice.

NOTE

You can perform Red Hat Container certification by using a custom container registry. This enables you to provide an access token to the registry, which thereby helps to verify the availability of the container images for users. Also, it ensures that the container image can undergo scanning by the security scanner and can be published on the Red Hat Ecosystem Catalog. Custom registries employ diverse authentication methods, and the Red Hat Software certification program supports the following authentication methods along with the standard OCI registry API:

- Bearer Authentication
- OAuth2
- Basic Authentication

For more details about the authentication methods, see Supported auth methods.

3. Download the Preflight certification utility.
4. Run Preflight with your container image.
5. Submit results on Red Hat Partner Connect.

Additional resources

For detailed instructions about certification testing, see Running the certification test suite.
13.2.3. Publishing the certified container on the Red Hat Ecosystem Catalog

Certified container images are delivered to customers through the Red Hat Connect Image Registry, which you can then run on a supported Red Hat container platform. Your product and its images get listed on the Red Hat Container Catalog using the listing information that you provide.

Additional resources

- For more details about publishing your certified container image, see Publishing the certified container.
- For more information about containers, see:
  - Containers and UBI Technical Track
  - Choosing the right container image
  - Everything you need to know about Red Hat Universal Base Image

13.3. TESTING MULTI-ARCH CONTAINER CERTIFICATION USING PREFLIGHT

Follow these steps to perform a multi-arch container certification test:

Procedure

1. Build your multi-arch container images. See Building and pushing multi-arch container images using Podman for more information.
2. Upload your container images to your chosen registry. You can select any OCI registry of your choice.

NOTE

You can perform Red Hat Container certification by using a custom container registry. This enables you to provide an access token to the registry, which thereby helps to verify the availability of the container images for users. Also, it ensures that the container image can be scanned by the security scanner and published on the Red Hat Ecosystem Catalog. Custom registries employ diverse authentication methods, and the Red Hat Software certification program supports the following authentication methods along with the standard OCI registry API:

- Bearer Authentication
- OAuth2
- Basic Authentication

For more details about the authentication methods, see Supported auth methods.

3. Download the Preflight certification utility. Ensure that you have the latest version to benefit from any updates or improvements.
4. Run preflight with your multi-arch container image. Preflight will automatically run and submit results for all architectures if the supplied image is a manifest list.
5. Review and address the preflight certification results.

6. Submit results on Red Hat Partner Connect.

13.3.1. Building and pushing multi-arch container images using Podman

Follow the instructions to build and push multi-arch images using Podman:

**Prerequisites**

1. Podman is installed on your system.

2. You have a Dockerfile that defines the image you want to build for multiple architectures.

3. You have a Quay.io account or any other container registry account.

**Procedure**

1. Prepare Your Dockerfile.

2. Build and push the multi-arch container Images. Check the podman-manifest documentation for instructions on building and pushing the multi-arch container images.
CHAPTER 14. CREATING A CONTAINER APPLICATION PROJECT

Prerequisites

- Build your container by using UBI or RHEL as your base image.
- Upload your container to a public or private registry of your choice.

Procedure

Follow these steps to create a container application project:

1. Log in to Red Hat Partner Connect portal. The Access the partner portals web page displays.

2. Navigate to the Certified technology portal tile and click Log in for technology partners.

3. Enter the login credentials and click Login. The Red Hat Partner Connect web page displays.

4. On the page header, select Product certification and click Manage certification projects. My Work web page displays the Product Listings and Certification Projects, if available.

5. Click Create Project.

6. In the What do you want to certify? dialog box, select your desired platform and click Next. For example, select the Red Hat OpenShift radio button for creating a container project.

7. In the What do you want to certify? dialog box, select Container image radio button and click Next.

8. On the Create container image certification project web page, provide the following details to create your project.

    **IMPORTANT**
    
    You cannot change the project name or its distribution method after you have created the project.

    a. **Project Name** - Enter the project name. This name is not published and is only for internal use.

    b. **OS Content Type** - Select the type of image that you want to use for your container project:

        i. **Red Hat Universal Base Image** - You can distribute UBI-based container images through the Red Hat Container registry or any other third-party registry.

        ii. **Red Hat Enterprise Linux** - You can distribute RHEL-based container images through the Red Hat Container registry only.

    c. **Distribution Method** - Select the container registry that you will use for distributing your container images. Customers will pull your container images from this location and in all the following methods your container images remain hosted on a registry that you manage. Red
Hat recommends **Quay.io** to host your images, but you can use any Kubernetes-compatible registry.

i. **Red Hat Container Registry** - Select this option, if you want Red Hat to distribute your containers through Red Hat’s container registry. Images with this distribution method are hosted on your own container registry, but are distributed to customers through a Red Hat registry proxy address. When you select this option, customers will have access to your containers without adding registries to their configuration, but you will not have visibility on customer-specific download metrics or other usage data from the proxy.

ii. **Your own Container Registry** - Select this option to publish your certified containers on your own registry. When using your own third-party registry, customers will need to authenticate to your registry, to pull your certified containers, and use your product. In disconnected environments, customers will need to add your registry to their Red Hat platforms to install your certified containers.

**IMPORTANT**

Red Hat recommends self-hosting on your own registry because you are able to access your entire container metrics and have full control on the access of your product. Red Hat recommends using **Quay.io** for this purpose, however you can use any Kubernetes compatible registry.

d. Click **Create project**
CHAPTER 15. CONFIGURING THE CONTAINER PROJECT

After the container project is created, the newly created container project web page displays.

The container project web page comprises of the following tabs:

- **Overview** - Contains the pre-certification-checklist.
- **Images** - Displays image test results that you submit from the preflight tool.
- **Settings** - Allows you to configure the registry and repository details.

Additionally, to perform the following actions on the container project, click the Actions menu on the container project web page:

- Open Support Case
- Archive Project

15.1. COMPLETE THE PRE-CERTIFICATION CHECKLIST

Certified containers are applications that meet Red Hat’s best practices for packaging, distribution and maintenance. Certified containers imply a commitment from partners to maintain their images up-to-date and represent the highest level of trust and supportability for Red Hat customers container-capable platforms, including OpenShift.

For certifying containers, you must complete the pre-certification checklist and then publish your container image. The Overview tab of the container project contains the pre-certification checklist. The pre-certification checklist consists of a series of tasks that you must complete, to certify and publish your container image.

Before you publish your container image, perform the following tasks in the checklist:

1. **Accept the Red Hat Container Appendix**
2. **Provide repository details about your container**
3. **Complete your company profile**
4. **Complete export control questionnaire**
5. **Submit your container for verification**
6. **Attach a completed product listing**

15.1.1. Accept the Red Hat Container Appendix

1. To publish any image, agree to the terms regarding the distribution of partner container images.

2. Navigate to the **Accept the Red Hat Container Appendix** tile and click **Review Accepted Terms**. The Red Hat Partner Connect Container Appendix document displays. Read the document to know the terms related to the distribution of container images.

15.1.2. Provide details about your container
1. Navigate to **Provide details about your container** tile, to enter your repository details that are displayed in the catalog. This will allow users to pull your container image.

2. Click **Add details**. You are navigated to the **Settings** tab.

3. Enter all the required repository information.

4. After filling-in all the fields, click **Save**.

**NOTE**

All the fields marked with asterisk * are required and must be completed before you can proceed with container certification.

### 15.1.3. Complete your company profile

Keep your company profile up-to-date. This information gets published in the Catalog along with your certified product.

To verify:

1. Navigate to **Complete your company profile** tile.

2. Click **Review** in your checklist.

3. To make any changes, click **Edit**.

4. Click **Save**.

**NOTE**

Make sure to complete all the items of the Pre-certification checklist except **Test your operator bundle** data before submitting your Operator Bundle image.

After completing all the steps, a green check mark appears beside the tiles to indicate that configuration is complete.

### 15.1.4. Complete export control questionnaire

**Export control questionnaire** contains a series of questions through which the Red Hat legal team evaluates the export compliance by third-party vendors.

Partner’s legal representative must review and answer the questions. Red Hat takes approximately five business days to evaluate the responses and based on the responses Red Hat approves partner or defers decision or requests more information.

**NOTE** - If you are using a version of **UBI** (Universal Base Image) to build your container image, you can host your image in a private repository. This allows you to skip the Export Compliance. This Export Compliance form is required only if you are hosting your images on the [Red Hat Container Catalog](#).

### 15.1.5. Submit your container for verification

Set up preflight locally and navigate to **Submit your container for verification** tile to submit the software certification test results.
15.1.6. Attach a completed product listing

You can either create a new product listing, or attach a project to an existing product listing.

15.1.6.1. Creating a new product listing

Procedure

1. Navigate to the Attach a completed product listing tile.

2. From the Select method drop-down menu, select Attach or edit. The Attach product listing pop-up window displays.

3. Click Create new product listing.

4. In the Product Name text box, enter the required product name.

5. From the Product listing type, select the required product type.

6. Click Save.

7. From the Select method drop-down menu, click View product listing to navigate to the new product listing and enter all the required product listing details.

8. Click Save.

Verification

Go to your project page on the Partner Connect portal and navigate to Certification Projects > Attached Listings column. You can see the new attached product listing.

15.1.6.2. Attaching a container project to an existing product listing

Procedure

1. Navigate to the Attach a completed product listing tile.

2. From the Select method drop-down menu, select Attach or edit. The Attach product listing pop-up window displays.

3. From the Related product listing section, click Select drop-down arrow to select one or more product listings.

4. Click Save.

Verification

Go to your project page on the Partner Connect portal and navigate to Certification Projects > Attached Listings column. You can see all the attached product listings.

15.1.6.3. Attaching multiple container projects to a single product listing

If your product consists of multiple container projects, you can attach all required container projects to a single product listing in the Listing Details.

Procedure
1. On the My Work page select the required product listing.

2. In the Listing Details sidebar menu of the product page click the Certification Projects option.

3. Click Attach Project.

4. In the Attach certification projects pop-up window select one or more container projects.

5. Click Attach.

**Verification**

Go to your project page on the Partner Connect portal and navigate to Certification Projects > Attached Listings column. You can see the attached product listing.

**15.1.6.4. Removing attached container project from an existing product listing**

If your product no longer uses a container project that is attached to the product listing, you can remove it.

**Procedure**

1. On the My Work page click the required product listing.

2. In the Listing Details sidebar menu of the product page click the Certification Projects option.

3. Select all the attached container projects you want to remove and click Remove.

**15.2. VIEWING THE IMAGE TEST RESULTS**

Images tab displays the image test results that you submit from the preflight tool.

It displays the following details of your container image:

- Specific image ID
- Certification test - pass/fail - click on it for more details.
- Health Index - Container Health Index is a measure of the oldest and most severe security updates available for a container image. 'A' is more up-to-date than 'F. See Container Health Index grades as used inside the Red Hat Container Catalog for more details.
- Architecture - specific architecture of your image, if applicable.
- Created - the day in which the submission was processed.
- Actions menu allows you to perform the following tasks:
  - Delete Image - click this option to delete your container image when your image is unpublished.
  - Sync Tags - when you have altered your image tag, use this option to synchronize the container image information available on both Red Hat Partner Connect and Red Hat Container catalog.
  - View in Catalog - When your container image is published, click this option to view the published container image on the Red Hat Ecosystem Container catalog.
15.3. MANAGING THE CONTAINER PROJECT SETTINGS

You can configure the registry and repository details by using the **Settings** tab.

Enter the required details in the following fields:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Container registry namespace</td>
<td>This field is non-editable and is auto-populated from your company profile. For example, <em>mycompany</em>.</td>
</tr>
<tr>
<td>Outbound repository name</td>
<td>Repository name that you have selected or the name obtained from your private registry in which your image is hosted. For example, <em>ubi-minimal</em>.</td>
</tr>
<tr>
<td>Repository summary</td>
<td>Summary information displayed in the Ecosystem Catalog listing, available at <strong>Technical Information → General information → Summary</strong>.</td>
</tr>
<tr>
<td>Repository description</td>
<td>Repository description displayed in the Ecosystem Catalog listing, available at <strong>Overview and Technical Information → General information → Description</strong>.</td>
</tr>
<tr>
<td>Application categories</td>
<td>Select the respective application type of your software product.</td>
</tr>
<tr>
<td>Supported platforms</td>
<td>Select the supported platforms of your software product.</td>
</tr>
<tr>
<td>Host level access</td>
<td>Select between the two options:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Unprivileged</strong> - If your container is isolated from the host. or</td>
</tr>
<tr>
<td></td>
<td>- <strong>Privileged</strong> - If your container requires special host-level privileges.</td>
</tr>
</tbody>
</table>

**NOTE**

If your product’s functionality requires root access, you must select the privileged option, before running the preflight tool. This setting is subject to Red Hat review.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release Category</td>
<td>Select between the two options:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Generally Available</strong> - When you select this option, the application is generally available and supported.</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>- <strong>Beta</strong> - When you select this option, the application is available as a pre-release candidate.</td>
</tr>
<tr>
<td>Project name</td>
<td>Name of the project for internal purposes.</td>
</tr>
<tr>
<td>Auto-publish</td>
<td>When you enable this option, the container image gets automatically published on the Red Hat Container catalog, after passing all the certification tests.</td>
</tr>
<tr>
<td>Technical contact email address</td>
<td>Primary technical contact details of your product.</td>
</tr>
</tbody>
</table>

**NOTE**

Note: All the fields marked with asterisk * are required and must be completed before you can proceed with container certification.
CHAPTER 16. RUNNING THE CERTIFICATION TEST SUITE

Follow the instructions to run the certification test suite:

Prerequisites

- You have a Red Hat Enterprise Linux (RHEL) system.
- You can use Podman to log in to your image registry. For example:

  $ podman login --username <your_username> --password <your_password> --authfile ./temp-authfile.json <registry>

- You have set up your container project on the Red Hat Partner Connect portal. The pre-certification checklist must at least be in progress.
- You have a pyxis API key.

Procedure

1. Build your container image by using Podman.

   **NOTE**

   Using Podman to build container images is optional.

2. Upload your container to any private or public registry of your choice.

3. Download the latest Preflight certification utility.

4. Perform the following steps to verify the functionality of the container being certified:

   a. Run the Preflight certification utility:

      $ preflight check container \
      registry.example.org/<namespace>/<image_name>:<image_tag>

   b. Review the log information and change the container as needed. For more information, see the troubleshooting information page.
      If you find any issues, either submit a support ticket or run the following command:

      $ preflight support

      Red Hat welcomes community contributions. If you experience a bug related to Preflight or the Red Hat Partner Connect Portal, or if you have a suggestion for a feature improvement or contribution, please report the issue. Before reporting an issue, ensure to review the open issues to avoid duplication.

   c. Run the container certification utility and make changes until all the tests pass.

5. Submit the certification test results by running the following command:

   $ preflight check container \
   registry.example.org/<namespace>/<image_name>:<image_tag> \

After you submit your test results to the Red Hat Partner Connect portal, Red Hat will scan the layers of your container for package vulnerabilities.

6. Review your certification and vulnerability test results in the certification project UI by navigating to the Images tab in the Red Hat Partner Connect portal. For more information, see Viewing the image test results.

Additional resources

If you are certifying a RHEL application, validate the functionality of your product by following the Non-container certification workflow.

You can also certify your RHEL application container by using the Red Hat Certification tool, which has the built-in pre-flight tool, thereby enabling you to validate your container.

Procedure

Follow the steps to use the built-in preflight tool:

1. Install the preflight package:
   ```bash
   # dnf install redhat-certification-preflight
   ```

2. Run rhcert and follow the instructions:
   ```bash
   # rhcert-run
   ```

3. Review and save the test results:
   ```bash
   # rhcert-save
   ```
CHAPTER 17. PUBLISHING THE CERTIFIED CONTAINER

After you submit your test results from the preflight tool on your Partner Connect project, your container images are scanned for vulnerabilities within the project. When the scanning is successfully completed, the publish button will be enabled for your image. After you click the publish button, your image will be available on the Red Hat Ecosystem Catalog.

IMPORTANT

The Red Hat software certification does not conduct testing of the Partner’s product in how it functions or performs on the chosen platform. Any and all aspects of the certification candidate product’s quality assurance remains the partner’s sole responsibility.
CHAPTER 18. WORKING WITH OPERATORS

NOTE
Certify your operator image or necessary container image as a container application project before proceeding with Red Hat Operator certification. All containers referenced in an Operator Bundle must already be certified and published in the Red Hat Ecosystem Catalog prior to beginning to certify an Operator Bundle.

18.1. INTRODUCTION TO OPERATORS

A Kubernetes operator is a method of packaging, deploying, and managing a Kubernetes application. Our Operator certification program ensures that the partner’s operator is deployable by the Operator Lifecycle Manager on the OpenShift platform and is formatted properly, using Red Hat certified container images.

18.2. CERTIFICATION WORKFLOW FOR OPERATORS

NOTE
Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

Task Summary
The certification workflow includes three primary steps-

1. Section 18.2.1, “Certification on-boarding”
2. Section 4.2, “Certification testing”
3. Section 18.2.3, “Publishing the certified Operator on the Red Hat Ecosystem Catalog”

18.2.1. Certification on-boarding
Perform the steps outlined for the certification on-boarding:

1. Join the Red Hat Connect for Technology Partner Program.
2. Add a software product to certify.
3. Enter your company profile.
4. Complete the pre-certification checklist.
5. Create an OpenShift Operator project bundle for your product.

18.2.2. Certification testing
To run the certification test:

1. Fork the Red Hat upstream repository.
2. Install and run the Red Hat certification pipeline on your test environment.

3. Review the test results and troubleshoot, if any issues.

4. Submit the certification results to Red Hat through a pull request.

5. If you want Red Hat to run all the tests then create a pull request. This triggers the certification hosted pipeline to run all the certification checks on Red Hat infrastructure.

18.2.3. Publishing the certified Operator on the Red Hat Ecosystem Catalog

When you complete all the certification checks successfully, you can submit the test results to Red Hat. You can turn on or off this result submission step depending on your individual goals. When the test results are submitted, it triggers the Red Hat infrastructure to automatically merge your pull request and publish your Operator.

The following diagram gives an overview of testing your Operator locally:

Figure 18.1. Overview of testing your Operator locally

Red Hat recommends you to choose this path for testing your Operator.

Additional resources

For more details about operators, see:

- Operators
- Operator Framework
- Operator Capability Levels
• Packaging Applications and Services with Kubernetes Operators
CHAPTER 19. CREATING AN OPERATOR BUNDLE PROJECT

Prerequisites

Certify your operator image or necessary container image as a container application project before creating an operator bundle.

Procedure

1. Log in to Red Hat Partner Connect portal. The Access the partner portals web page displays.
2. Navigate to the Certified technology portal tile and click Log in for technology partners.
3. Enter the login credentials and click Login. The Red Hat Partner Connect web page displays.
4. On the page header, select Product certification and click Manage certification projects. My Work web page displays the Product Listings and Certification Projects, if available.
5. Click Create Project.
6. In the What platform do you want to certify on? dialog box, select the Red Hat OpenShift radio button and click Next.
7. In the What do you want to certify? dialog box, select Operator Bundle Image radio button and click Next.
8. On the Create operator bundle image certification project web page, provide the following details to create your project.

   **IMPORTANT**

   You cannot change the project name and its distribution method after you have created the project.

   a. **Project Name**: Enter the project name. This name is not published and is only for internal use.

   b. **Specialized Certification** - This feature allows you to certify a specialized operator.

      i. Select My operator is a CNI or CSI checkbox, if you want to certify a specialized operator.

      ii. Select the required operator:

          A. Container Network Interface (CNI)

          B. Cloud Storage Interface (CSI)

   c. **Publication Options** - Select one of the following options for publishing your operator:

      i. **Web catalog only (catalog.redhat.com)** - The operator is published to the Red Hat Container Catalog and is not visible on Red Hat OpenShift OperatorHub. This is the default option when you create a new project and this option is suitable for partners who do not want their operator publicly installable within OpenShift, but require a proof of
certification. Select this option only if you have a distribution, entitlement, or other business requirements that is not otherwise accommodated within the OpenShift In-product Catalog (Certified) option.

ii. **OpenShift In-product Catalog (Certified)** - The operator is listed on the [Red Hat Container Catalog](https://catalog.redhat.com) and published to the certified operator index embedded in the OperatorHub of OpenShift.

d. Click **Create project**
CHAPTER 20. CONFIGURING THE OPERATOR BUNDLE

After the project is created, the newly created Operator Bundle project web page displays.

The Operator bundle web page comprises the following tabs:

- **Overview** - Contains the pre-certification-checklist.
- **Test Results** - Displays the test results after running the certification.
- **Update Graph** - Displays the OpenShift Version, Channel status, Update Paths and Other Available Channel details.
- **Settings** - Allows you to configure the registry and repository details.

Additionally, to perform the following actions on the Operator bundle, click the **Actions** menu on the Operator bundle web page:

- **Open Support case**
- **Archive Project**
- **Remove from Operator Hub**

20.1. COMPLETE THE PRE-CERTIFICATION CHECKLIST

The **Overview** tab of the Operator bundle project contains the pre-certification checklist. The pre-certification checklist consists of a series of tasks that you must complete, to certify and publish your Operator bundle.

Before you publish your Operator Bundle image, perform the following tasks in the checklist:

1. **Accept the Red Hat Container Appendix**
2. **Provide repository details used for pulling your container**
3. **Complete your company profile**
4. **Test your operator bundle data and submit a pull request**
5. **Attach a completed product listing**
6. **Validate the functionality of your CNI or CSI on Red Hat OpenShift**

20.1.1. Accept the Red Hat Container Appendix

Users must agree to the terms regarding the distribution of partner container images before they can publish any image.

Navigate to **Accept the Red Hat Container Appendix** tile and click **Review Accepted Terms**. Read the Red Hat Partner Connect Container Appendix document that displays for terms related to the distribution of container images.

20.1.2. Provide repository details used for pulling your container
1. Navigate to **Provide repository details used for pulling your container** tile, to enter your repository details that are displayed in the Catalog, so that customers can pull your container image, and click **Add details**.

2. On the **Settings** tab, enter all the required repository information, and click **Save**.

   **NOTE**

   All the fields marked with asterisk * are required and must be completed before you can proceed with Operator bundle certification.

20.1.3. **Complete your company profile**

Keep your company profile up-to-date. This information gets published in the Catalog along with your certified product.

To verify:

1. Navigate to **Complete your company profile** tile.

2. Click **Review** in your checklist.

3. To make any changes, click **Edit**.

4. Click **Save**.

   **NOTE**

   Make sure to complete all the items of the Pre-certification checklist except **Test your operator bundle** data before submitting your Operator Bundle image.

After completing all the steps, a green check mark appears beside the tiles to indicate that configuration is complete.

20.1.4. **Test your operator bundle data and submit a pull request**

To run the Operator certification suite, navigate to the **Test your operator bundle data and submit a pull request** tile and click **View Options**. It displays two tabs to determine how to test and certify your operator.

20.1.4.1. **Test locally with OpenShift**

Use the OpenShift cluster of your choice for testing and certification. This option allows you to integrate the provided pipeline to your own workflows for continuous verification and access to comprehensive logs for a faster feedback loop. This is the recommended approach. For more information, see **Running the certification test suite locally**.

20.1.4.2. **Test with Red Hat’s hosted pipeline**

This approach is separate from your OpenShift software testing from certification. After you have tested your operator on the version of OpenShift you wish to certify, you can use this approach if you don’t want the comprehensive logs, or are not ready to include it in your own workflows. For more information, see **Running the certification suite with the Red Hat hosted pipeline**.
Comparing certification testing options

In the long term, Red Hat recommends using the "local testing" option, also referred to as CI Pipeline, for testing your Operator. This method allows you to incorporate the tests into their CI/CD workflows and development processes, therefore ensuring the proper functioning of your product on the OpenShift platform and streamlining the future updates and recertifications for the Operator.

Although initially, learning about the method and debugging errors may take some time, it is an advanced method and provides the best and quickest feedback.

On the other hand, Red Hat recommends using the hosted pipeline, running on the Red Hat infrastructure option on a number of events, such as when working on an urgent deadline, or when enough resources and time is not available to use the tooling.

You can use the hosted pipeline simultaneously with CI/local pipeline as you learn to incorporate the local tooling long term.

20.1.5. Attach a completed product listing

This feature allows you to either create a new product listing or to attach the project to an existing OpenShift product listing for your new project.

1. Navigate to Attach a completed product listing tile.

2. From the Select method drop-down menu, select Attach or edit. The Attach product listing page displays.

3. Decide whether you want to attach your project to an existing product listing, or if you want to create a new product listing:

   a. To attach your project to an existing product listing:
      i. From the Related product listing section, click Select drop-down arrow to select the product listing.
      ii. Click Save.

   b. To create a new product listing:
      i. Click Create new product listing
      ii. In the Product Name text box, enter the required product name.
      iii. From the Product listing type, select the required product type, for example - OpenShift Operator.
      iv. Click Save.

4. From the Select method drop-down menu, click View product listing to navigate to the new product listing and enter all the required product listing details.

5. Click Save.

20.1.6. Validate the functionality of your CNI or CSI on Red Hat OpenShift
This feature is applicable for CNI and CSI operators only.

This feature allows you to run the certification test locally and share the test results with the Red Hat certification team.

To validate the functionality of your specialized CNI or CSI operator:

1. Select this option and click **Start**. A new project gets created in the Red Hat Certification portal and you are redirected to the appropriate project portal page.

2. On the Summary tab, navigate to the **Files** section and click **Upload**, to upload your test results.

3. Add any relevant comments in the **Discussions** section, and then click **Add Comment**.

Red Hat will review the results file you submitted and validate your specialized CNI or CSI operator. Upon successful validation, your operator will get approved and published.

**Additional resources**

- For detailed information, see CNI and CSI workflow.

**20.2. VIEWING THE TEST RESULTS**

After running the test certification suite, navigate to the **Test Results** tab on the Project header to view your test results.

It has two tabs:

- **Results** - Displays a summary of all the certification tests along with their results.

- **Artifacts** - Displays log files.

**20.3. WORKING WITH UPDATE GRAPH**

You can view and update OpenShift Version, Channel status, Update Paths and Other Available Channel details through the Update Graph feature.

Navigate to the **Update Graph** tab on the project header, to view and update the required details of your Operator project. See Operator update documentation tile, for more information on the upgrades.

**20.4. MANAGING PROJECT SETTINGS**

You can configure the registry and repository details through the Settings tab.

Enter the required details in the following fields:

- Container registry namespace
- Outbound repository name
- Authorized GitHub user accounts
- OpenShift Object YAML - Use this option to add a docker config.json secret, if you are using a private container registry.

- Red Hat Ecosystem Catalog details - It includes Repository summary, Repository description, Application categories and Supported platforms.

- Project Details - It includes Project name, Technical contact and email address.

**IMPORTANT**

This information is for internal use and is not published.

**NOTE**

All the fields marked with asterisk * are required and must be completed before you can proceed with Operator bundle certification.
CHAPTER 21. RUNNING THE CERTIFICATION TEST SUITE LOCALLY

By selecting this option, you can run the certification tooling on your own OpenShift cluster.

**NOTE**

Red Hat recommends you to follow this method to certify your operators.

This option is an advanced method for partners who:

- are interested in integrating the tooling into their own developer workflows for continuous verification,
- want access to comprehensive logs for a faster feedback loop,
- or have dependencies that are not available in a default OpenShift installation.

Here’s an overview of the process:

**Figure 21.1. Overview of running the certification test suite locally**

You use OpenShift pipelines based on Tekton, to run the certification tests, enabling the viewing of comprehensive logs and debugging information in real time. Once you are ready to certify and publish your operator bundle, the pipeline submits a pull request (PR) to GitHub on your behalf. If everything passes successfully, your operator is automatically merged and published in the Red Hat Container Catalog and the embedded operatorHub in OpenShift.

Follow the instructions to run the certification test suite locally:

**Prerequisites**

To certify your software product on Red Hat OpenShift test environment, ensure to have:

- The OpenShift cluster version 4.8 or later is installed.
NOTE

The OpenShift Operator Pipeline creates a persistent volume claim for a 5GB volume. If you are running an OpenShift cluster on bare metal, ensure you have configured dynamic volume provisioning. If you do not have dynamic volume provisioning configured, consider setting up a local volume. To prevent from getting Permission Denied errors, modify the local volume storage path to have the container_file_t SELinux label, by using the following command:

```
chcon -Rv -t container_file_t "storage_path(/.*)?"
```

- You have the kubeconfig file for an admin user that has cluster admin privileges.
- You have a valid operator bundle.
- The OpenShift CLI tool (oc) version 4.7.13 or later is installed.
- The Git CLI tool (git) version 2.32.0 or later is installed.
- The Tekton CLI tool (tkn) version 0.19.1 or later is installed.

Additional resources

For program prerequisites, see Red Hat OpenShift certification prerequisites.

21.1. ADDING YOUR OPERATOR BUNDLE

In the operators directory of your fork, there are a series of subdirectories.

21.1.1. If you have certified this operator before -

Find the respective folder for your operator in the operators directory. Place the contents of your operator Bundle in this directory.

NOTE

Make sure your package name is consistent with the existing folder name for your operator.

21.1.2. If you are newly certifying this operator -

If the newly certifying operator does not have a subdirectory already under the operator’s parent directory then you have to create one.

Create a new directory under operators. The name of this directory should match your operator’s package name. For example, `my-operator`.

- In this operators directory, create a new subdirectory with the name of your operator, for example, `<my-operator>` and create a version directory for example, `<V1.0>` and place your bundle. These directories are preloaded for operators that have been certified before.
- Under the version directory, add a **manifests** folder containing all your OpenShift manifests including your **clusterserviceversion.yaml** file.

**Recommended directory structure**

The following example illustrates the recommended directory structure.

```
- config.yaml
- operators
  - my-operator
    - v1.4.8
      - manifests
        - cache.example.com_my-operators.yaml
        - my-operator-controller-manager-metrics-service_v1_service.yaml
        - my-operator-manager-config_v1_configmap.yaml
        - my-operator-metrics-reader_rbac.authorization.k8s.io_v1_clusterrole.yaml
        - my-operator.clusterserviceversion.yaml
      - metadata
      - annotations.yaml
  - ci.yaml
```

<table>
<thead>
<tr>
<th>Configuration file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config.yaml</td>
<td>In this file include the organization of your operator. It can be <strong>certified-operators</strong>. For example, <strong>organization: certified-operators</strong></td>
</tr>
<tr>
<td>ci.yaml</td>
<td>In this file include your Red Hat Technology Partner project ID and the organization target for this operator. For example, <strong>cert_project_id: &lt;your partner project id&gt;</strong>. This file stores all the necessary metadata for a successful certification process.</td>
</tr>
<tr>
<td>Configuration file</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| annotations.yaml   | In this file include an annotation of OpenShift versions, which refers to the range of OpenShift versions. For example, `v4.8-v4.10` means versions 4.8 through 4.10. Add this to any existing content. For example, # OpenShift annotations com.redhat.openshift.versions: v4.8-v4.10. The com.redhat.openshift.versions field, which is part of the metadata in the operator bundle, is used to determine whether an operator is included in the certified catalog for a given OpenShift version. You must use it to indicate one or more versions of OpenShift supported by your operator. Note that the letter 'v' must be used before the version, and spaces are not allowed. The syntax is as follows:  
  - A single version indicates that the operator is supported on that version of OpenShift or later. The operator is automatically added to the certified catalog for all subsequent OpenShift releases.  
  - A single version preceded by '=' indicates that the operator is supported only on that specific version of OpenShift. For example, using `=v4.8` will add the operator to the certified catalog for OpenShift 4.8, but not for later OpenShift releases.  
  - A range can be used to indicate support only for OpenShift versions within that range. For example, using `v4.8-v4.10` will add the operator to the certified catalog for OpenShift 4.8 through 4.10, but not for OpenShift 4.11 or 4.12. |

**Additional resources**

- For more details, see Managing OpenShift Versions.
- For an example of an operator Bundle, see here.

**21.2. FORKING THE REPOSITORY**

1. Log in to GitHub and fork the RedHat OpenShift operators upstream repository.
2. Fork the appropriate repositories from the following table, depending on the Catalogs that you are targeting for distribution:
### 21.3. INSTALLING THE OPENSHIFT OPERATOR PIPELINE

**Prerequisites**

Administrator privileges on your OpenShift cluster.

**Procedure**

You can install the OpenShift Operator Pipeline by two methods:

- **Automated process** (Red Hat recommended process)
- **Manual process**

#### 21.3.1. Automated process

Red Hat recommends using the automated process for installing the OpenShift Operator Pipeline. The automated process ensures the cluster is properly configured before executing the CI Pipeline. This process installs an operator to the cluster that helps you to automatically update all the CI Pipeline tasks without requiring any manual intervention. This process also supports multitenant scenarios in which you can test many operators iteratively within the same cluster.

Follow these steps to install the OpenShift Operator Pipeline through an Operator:

> **NOTE**
>
> Keep the source files of your Operator bundle ready before installing the Operator Pipeline.

#### 21.3.1.1. Prerequisites

Before installing the OpenShift Operator Pipeline, in a terminal window run the following commands, to configure all the prerequisites:
NOTE
The Operator watches all the namespaces. Hence, if `secrets/configs/etc` already exist in another namespace, you can use the existing namespace for installing the Operator Pipeline.

1. Create a new namespace:
   `oc new-project oco`

2. Set `kubeconfig` environment variable:
   `export KUBECONFIG=/path/to/your/cluster/kubeconfig`

   **NOTE**
   This `kubeconfig` variable is used to deploy the Operator under test and run the certification checks.

   `oc create secret generic kubeconfig --from-file=kubeconfig=$KUBECONFIG`

3. Execute the following commands for submitting the certification results:
   - Add the github API token to the repository where the pull request will be created:
     `oc create secret generic github-api-token --from-literal=GITHUB_TOKEN=<github token>`
   - Add RedHat Container API access key:
     `oc create secret generic pyxis-api-secret --from-literal=pyxis_api_key=< API KEY >`

     This API access key is specifically related to your unique partner account on the Red Hat Partner Connect portal.

4. Prerequisites for running OpenShift cluster on bare metal:

   a. If you are running an OpenShift cluster on bare metal, the Operator pipeline requires a 5Gi persistent volume to run. The following yaml template helps you to create a 5Gi persistent volume by using local storage.
      For example:

      ```yaml
      apiVersion: v1
      kind: PersistentVolume
      metadata:
        name: my-local-pv
      spec:
        capacity:
          storage: 5Gi
        volumeMode: Filesystem
        accessModes:
          - ReadWriteOnce
        persistentVolumeReclaimPolicy: Delete
        local:
      ```
b. The CI pipeline automatically builds your operator bundle image and bundle image index for testing and verification. By default, the pipeline creates images in the OpenShift container registry on the cluster. To use this registry on bare metal, set up the internal image registry before running the pipeline. For detailed instructions on setting up the internal image registry, see Image registry storage configuration.

If you want to use an external private registry then provide your access credentials to the cluster by adding a secret. For detailed instructions, see Using a private container registry.

Additional resources

- For instructions on obtaining your API key, see Get API Key.
- For additional repository configurations, see Configuring the repository for submitting the certification results.

21.3.1.2. Installing the pipeline through an Operator

Follow these steps to add the Operator to your cluster:

1. Install the Operator Certification Operator.
   - Log in to your OpenShift cluster console.
   - From the main menu, navigate to Operators → OperatorHub.
   - Type Operator Certification Operator in the All Items - Filter by keyword filter/search box.
   - Select Operator Certification Operator tile when it displays. The Operator Certification Operator page displays.
   - Click Install. The Install Operator web page displays.
   - Scroll down and click Install.
   - Click View Operator, to verify the installation.

2. Apply Custom Resource for the newly installed Operator Pipeline.
   - Log in to your OpenShift Cluster Console.
   - From the Projects drop-down menu, select the project for which you wish to apply the Custom Resource.
   - Expand Operator Pipeline and then click Create instance.
The **Create Operator Pipeline** screen is auto-populated with the default values.

**NOTE**
You need not change any of the default values if you have created all the resource names according to the **prerequisites**.

- Click Create.

The Custom Resource is created and the Operator starts reconciling.

**Verification Steps**

1. Check the Conditions of the Custom Resource.
   - Log in to your OpenShift cluster console.
   - Navigate to the project for which you have newly created the Operator Pipeline Custom Resource and click the Custom Resource.
   - Scroll down to the **Conditions** section and check if all the **Status** values are set to **True**.

   **NOTE**
   If a resource fails reconciliation, check the **Message** section to identify the next steps to fix the error.

2. Check the Operator logs.
   - In a terminal window run the following command:
     ```bash
     oc get pods -n openshift-marketplace
     ```
   - Record the full podman name of the **certification-operator-controller-manager** pod and run the command:
     ```bash
     oc get logs -f -n openshift-marketplace <pod name> manager
     ```
   - Check if the reconciliation of the Operator has occurred.

**Additional resources**

1. To uninstall the Operator Pipeline Custom Resource:
   - Log in to your OpenShift Cluster Console.
   - Navigate to the **Operator Certification Operator** main page and click the Operator Pipeline that you wish to uninstall.
   - Click the Custom Resource overflow menu and select **Uninstall**.

2. To uninstall the Operator:
   - Log in to your OpenShift Cluster Console.
Navigate to **Operators → Installed Operators** and search for the Operator that you wish to uninstall.

- Click the overflow menu of the respective Operator and click **Uninstall Operator**.

### 21.3.1.3. Executing the pipeline

For executing the pipeline, ensure you have **workspace-template.yml** file in a templates folder in the directory, from where you want to run the **tkn** commands.

To create a **workspace-template.yml** file, in a terminal window run the following command:

```
cat <<EOF> workspace-template.yml
spec:
  accessModes:
  - ReadWriteOnce
resources:
  requests:
  storage: 5Gi
EOF
```

You can run the pipeline through different **methods**.

### 21.3.2. Manual process

Follow these steps to manually install the OpenShift Operator Pipeline:

#### 21.3.2.1. Installing the OpenShift Pipeline Operator

1. Log in to your OpenShift cluster console.
2. From the main menu, navigate to **Operators > OperatorHub**.
3. Type **OpenShift Pipelines** in the **All Items - Filter by keyword** filter/search box.
4. Select **Red Hat OpenShift Pipelines** tile when it displays. The Red Hat OpenShift Pipelines page displays.
5. Click **Install**. The Install Operator web page displays.
6. Scroll down and click **Install**.

#### 21.3.2.2. Configuring the OpenShift (oc) CLI tool

A file that is used to configure access to a cluster is called a kubeconfig file. This is a generic way of referring to configuration files. Use kubeconfig files to organize information about clusters, users, namespaces, and authentication mechanisms.

The **kubectl** command-line tool uses kubeconfig files to find the information it needs to choose a cluster and communicate with the API server of a cluster.

1. In a terminal window, set the KUBECONFIG environment variable:

   ```bash
   export KUBECONFIG=/path/to/your/cluster/kubeconfig
   ```
The kubeconfig file deploys the Operator under test and runs the certification checks.

Additional resources

For more information on kubeconfig files, see Organizing Cluster Access Using kubeconfig Files.

21.3.2.3. Creating an OpenShift Project

Create a new namespace to start your work on the pipeline.

To create a namespace, in a terminal window run the following command:

```
oc adm new-project <my-project-name> # create the project
oc project <my-project-name> # switch into the project
```

**IMPORTANT**

Do not run the pipeline in the default project or namespace. Red Hat recommends creating a new project for the pipeline.

21.3.2.4. Adding the kubeconfig secret

Create a kubernetes secret containing your kubeconfig for authentication to the cluster running the certification pipeline. The certification pipeline requires kubeconfig to execute a test deployment of your Operator on the OpenShift cluster.

To add the kubeconfig secret, in a terminal window run the following command:

```
oc create secret generic kubeconfig --from-file=kubeconfig=$KUBECONFIG
```

Additional resources

For more information on the kubeconfig secret, see Secrets.

21.3.2.5. Importing Operator from Red Hat Catalog

Import Operators from the Red Hat catalog.

In a terminal window, run the following commands:

```
oc import-image certified-operator-index \
   --from=registry.redhat.io/redhat/certified-operator-index \
   --reference-policy local \
   --scheduled \
   --confirm \
   --all
```
NOTE

If you are using OpenShift on IBM Power cluster for ppc64le architecture, run the following command to avoid permission issues:

```bash
oc adm policy add-scc-to-user anyuid -z pipeline
```

This command grants the anyuid security context constraints (SCC) to the default pipeline service account.

### 21.3.2.6. Installing the certification pipeline dependencies

In a terminal window, install the certification pipeline dependencies on your cluster using the following commands:

```bash
$git clone https://github.com/redhat-openshift-ecosystem/operator-pipelines
$cd operator-pipelines
$oc apply -R -f ansible/roles/operator-pipeline/templates/openshift/pipelines
$oc apply -R -f ansible/roles/operator-pipeline/templates/openshift/tasks
```

### 21.3.2.7. Configuring the repository for submitting the certification results

In a terminal window, run the following commands to configure your repository for submitting the certification results:

#### 21.3.2.7.1. Adding GitHub API Token

After performing all the configurations, the pipeline can automatically open a pull request to submit your Operator to Red Hat.

To enable this functionality, add a GitHub API Token and use `--param submit=true` when running the pipeline:

```bash
oc create secret generic github-api-token --from-literal GITHUB_TOKEN=<github token>
```

#### 21.3.2.7.2. Adding Red Hat Container API access key

Add the specific container API access key that you receive from Red Hat:

```bash
oc create secret generic pyxis-api-secret --from-literal pyxis_api_key=< API KEY >
```

#### 21.3.2.7.3. Enabling digest pinning

**NOTE**

This step is mandatory to submit the certification results to Red Hat.

The OpenShift Operator pipeline can automatically replace all the image tags in your bundle with image Digest SHAs. This allows the pipeline to ensure if it is using a pinned version of all the images. The pipeline commits the pinned version of your bundle to your GitHub repository as a new branch.

To enable this functionality, add a private key having access to GitHub to your cluster as a secret.
1. Use Base64 to encode a private key which has access to the GitHub repository containing the bundle.

```
base64 /path/to/private/key
```

2. Create a secret that contains the base64 encoded private key.

```
cat << EOF > ssh-secret.yml
kind: Secret
apiVersion: v1
metadata:
  name: github-ssh-credentials
data:
  id_rsa: |
    <base64 encoded private key>
EOF
```

3. Add the secret to the cluster.

```
oc create -f ssh-secret.yml
```

21.3.2.7.4. Using a private container registry

The pipeline automatically builds your Operator bundle image and bundle image index for testing and verification. By default, the pipeline creates images in the OpenShift Container Registry on the cluster. If you want to use an external private registry then you have to provide credentials by adding a secret to the cluster.

```
oc create secret docker-registry registry-dockerconfig-secret \
  --docker-server=quay.io \
  --docker-username=<registry username> \
  --docker-password=<registry password> \
  --docker-email=<registry email>
```

21.4. EXECUTE THE OPENSHEET OPERATOR PIPELINE

You can run the OpenShift Operator pipeline through the following methods.

**TIP**

From the following examples, remove or add parameters and workspaces according to your requirements.

If you are using Red Hat OpenShift Local, formerly known as Red Hat CodeReady Containers (CRC), or Red Hat OpenShift on IBM Power for ppc64le architecture, pass the following tekton CLI argument to every ci pipeline command to avoid permission issues:

```
--pod-template templates/crc-pod-template.yml
```

**Troubleshooting**

If your OpenShift Pipelines operator 1.9 or later doesn’t work, follow the procedure to fix it:
Prerequisites

Ensure that you have administrator privileges for your cluster before creating a custom security context constraint (SCC).

Procedure

For OpenShift Pipelines operator 1.9 or later to work and to execute a subset of tasks in the ci-pipeline that requires privilege escalation, create a custom security context constraint (SCC) and link it to the pipeline service account by using the following commands:

1. To create a new SCC:
   
   ```
   oc apply -f ansible/roles/operator-pipeline/templates/openshift/openshift-pipelines-custom-scc.yml
   ```

2. To add the new SCC to a ci-pipeline service account:
   
   ```
   oc adm policy add-scc-to-user pipelines-custom-scc -z pipeline
   ```

Additional resources

For more information on SCCs, see About security context constraints.

21.4.1. Running the Minimal pipeline

Procedure

In a terminal window, run the following commands:

```bash
GIT_REPO_URL=<Git URL to your certified-operators fork >
BUNDLE_PATH=<path to the bundle in the Git Repo> (For example - operators/my-operator/1.2.8)

tkn pipeline start operator-ci-pipeline
   --param git_repo_url=$GIT_REPO_URL
   --param git_branch=main
   --param bundle_path=$BUNDLE_PATH
   --param env=prod
   --workspace name=pipeline,volumeClaimTemplateFile=templates/workspace-template.yml
   --showlog
```

After running the command, the pipeline prompts you to provide additional parameters. Accept all the default values to finish executing the pipeline.

The following is set as default and doesn’t need to be explicitly included, but can be overridden if your kubeconfig secret is created under a different name.

```bash
--param kubeconfig_secret_name=kubeconfig
--param kubeconfig_secret_key=kubeconfig
```

If you are running the ci pipeline on ppc64le and s390x architecture, edit the value of the parameter param pipeline_image from the default value quay.io/redhat-isv/operator-pipelines-images:released to quay.io/redhat-isv/operator-pipelines-images:multi-arch.
Troubleshooting

If you get a Permission Denied error when you are using the SSH URL, try the GITHUB HTTPS URL.

21.4.2. Running the pipeline with image digest pinning

Prerequisites

Execute the instructions Enabling digest pinning.

Procedure

In a terminal window, run the following commands:

```bash
GIT_REPO_URL=<Git URL to your certified-operators fork >
BUNDLE_PATH=<path to the bundle in the Git Repo> (ie: operators/my-operator/1.2.8)
GIT_USERNAME=<your github username>
GIT_EMAIL=<your github email address>

tkn pipeline start operator-ci-pipeline \
    --param git_repo_url=$GIT_REPO_URL \
    --param git_branch=main \
    --param bundle_path=$BUNDLE_PATH \
    --param env=prod \
    --param pin_digests=true \
    --param git_username=$GIT_USERNAME \
    --param git_email=$GIT_EMAIL \
    --workspace name=pipeline,volumeClaimTemplateFile=templates/workspace-template.yml \
    --workspace name=ssh-dir,secret=github-ssh-credentials \
    --showlog
```

Troubleshooting

When you get an error - could not read Username for https://github.com, provide the SSH github URL for --param git_repo_url.

21.4.3. Running the pipeline with a private container registry

Prerequisites

Execute the instructions included under By using a private container registry.

Procedure

In a terminal window, run the following commands:

```bash
GIT_REPO_URL=<Git URL to your certified-operators fork >
BUNDLE_PATH=<path to the bundle in the Git Repo> (ie: operators/my-operator/1.2.8)
GIT_USERNAME=<your github username>
GIT_EMAIL=<your github email address>
REGISTRY=<your image registry.  ie: quay.io>
IMAGE_NAMESPACE=<namespace in the container registry>

tkn pipeline start operator-ci-pipeline \
    --param git_repo_url=$GIT_REPO_URL \
    --param git_branch=main \
    --param bundle_path=$BUNDLE_PATH \
    --param env=prod \
    --param pin_digests=true \
    --param git_username=$GIT_USERNAME \
    --param git_email=$GIT_EMAIL \
    --param registry=$REGISTRY \
    --param image_namespace=$IMAGE_NAMESPACE \
    --workspace name=pipeline,volumeClaimTemplateFile=templates/workspace-template.yml \
    --workspace name=ssh-dir,secret=github-ssh-credentials \
    --showlog
```
21.5. SUBMIT CERTIFICATION RESULTS

Following procedure helps you to submit the certification test results to Red Hat.

Prerequisites

1. Execute the instructions Configuring the repository for submitting the certification results.

2. Add the following parameters to the GitHub upstream repository from where you want to submit the pull request for Red Hat certification. It is the Red Hat certification repository by default, but you can use your own repository for testing.

   -param upstream_repo_name=$UPSTREAM_REPO_NAME #Repo where Pull Request (PR) will be opened  
   --param submit=true

   The following is set as default and doesn’t need to be explicitly included, but can be overridden if your Pyxis secret is created under a different name.

   --param pyxis_api_key_secret_name=pyxis-api-secret 
   --param pyxis_api_key_secret_key=pyxis_api_key

Procedure

You can submit the Red Hat certification test results for four different scenarios:

21.5.1. Submitting test results from the minimal pipeline

Procedure

In a terminal window, execute the following commands:

```
GIT_REPO_URL=<Git URL to your certified-operators fork >  
BUNDLE_PATH=<path to the bundle in the Git Repo> (ie: operators/my-operator/1.2.8)

  tkn pipeline start operator-ci-pipeline 
  --param git_repo_url=$GIT_REPO_URL 
  --param git_branch=main 
  --param bundle_path=$BUNDLE_PATH 
  --param upstream_repo_name=redhat-openshift-ecosystem/certified-operators 
```
21.5.2. Submitting test results with image digest pinning

In a terminal window, execute the following commands:

**Prerequisites**

Execute the instructions included for:

- Configuring the repository for submitting the certification results.
- Enabling digest pinning.

**Procedure**

```
GIT_REPO_URL=<Git URL to your certified-operators fork >
BUNDLE_PATH=<path to the bundle in the Git Repo> (ie: operators/my-operator/1.2.8)
GIT_USERNAME=<your github username>
GIT_EMAIL=<your github email address>

tkn pipeline start operator-ci-pipeline \
   --param git_repo_url=${GIT_REPO_URL} \
   --param git_branch=main \
   --param bundle_path=${BUNDLE_PATH} \
   --param env=prod \
   --param pin_digests=true \
   --param git_username=${GIT_USERNAME} \
   --param git_email=${GIT_EMAIL} \
   --param upstream_repo_name=red-hat-openshift-ecosystem/certified-operators \
   --param submit=true \
   --workspace name=pipeline,volumeClaimTemplateFile=templates/workspace-template.yml \
   --workspace name=ssh-dir,secret=github-ssh-credentials \
   --showlog
```

**Troubleshooting**

When you get an error - could not read Username for https://github.com, provide the SSH github URL for --param git_repo_url.

21.5.3. Submitting test results from a private container registry

In a terminal window, execute the following commands:

**Prerequisites**

Execute the instructions included for:

- Configuring the repository for submitting the certification results.
- By using a private container registry.
**Procedure**

```
GIT_REPO_URL=<Git URL to your certified-operators fork >
BUNDLE_PATH=<path to the bundle in the Git Repo> (ie: operators/my-operator/1.2.8)
GIT_USERNAME=<your github username>
GIT_EMAIL=<your github email address>
REGISTRY=<your image registry.  ie: quay.io>
IMAGE_NAMESPACE=<namespace in the container registry>

```tkn pipeline start operator-ci-pipeline \
    --param git_repo_url=$GIT_REPO_URL \
    --param git_branch=main \
    --param bundle_path=$BUNDLE_PATH \
    --param env=prod \
    --param pin_digests=true \
    --param git_username=$GIT_USERNAME \
    --param git_email=$GIT_EMAIL \
    --param registry=$REGISTRY \
    --param image_namespace=$IMAGE_NAMESPACE \
    --param upstream_repo_name=red hat-openshift-ecosystem/certified-operators \
    --param submit=true \
    --workspace name=pipeline,volumeClaimTemplateFile=templates/workspace-template.yml \
    --workspace name=ssh-dir,secret=github-ssh-credentials \
    --workspace name=registry-credentials,secret=registry-docker config-secret \
    --showlog
```

### 21.5.4. Submitting test results with image digest pinning and from a private container registry

In a terminal window, execute the following commands:

**Prerequisites**

Execute the instructions included for:

- Configuring the repository for submitting the certification results.
- Enabling digest pinning.
- By using a private container registry.

**Procedure**

```
GIT_REPO_URL=<Git URL to your certified-operators fork >
BUNDLE_PATH=<path to the bundle in the Git Repo> (ie: operators/my-operator/1.2.8)
GIT_USERNAME=<your github username>
GIT_EMAIL=<your github email address>
REGISTRY=<your image registry.  ie: quay.io>
IMAGE_NAMESPACE=<namespace in the container registry>

```tkn pipeline start operator-ci-pipeline \
    --param git_repo_url=$GIT_REPO_URL \
    --param git_branch=main \
    --param bundle_path=$BUNDLE_PATH \
    --param env=prod \
```
After a successful certification, the certified product gets listed on Red Hat Ecosystem Catalog.

Certified operators are listed in and consumed by customers through the embedded OpenShift operatorHub, providing them the ability to easily deploy and run your solution. Additionally, your product and operator image will be listed on the Red Hat Ecosystem Catalog.
CHAPTER 22. RUNNING THE CERTIFICATION SUITE WITH RED HAT HOSTED PIPELINE

If you want to certify your operator with the Red Hat Hosted Pipeline you have to create a pull request for the Red Hat certification repository.

Choose this path if you are not interested in receiving comprehensive logs, or are not ready to include the tooling in your own CI/CD workflows.

Here’s an overview of the process:

Figure 22.1. Overview of Red Hat hosted pipeline

The process begins by submitting your Operator bundle through a GitHub pull request. Red Hat then runs the certification tests using an in-house OpenShift cluster. This path is similar to previous Operator bundle certification. You can see the certification test results both as comments on the pull request and within your Red Hat Partner Connect Operator bundle project. If all the certification tests are successful, your Operator will be automatically merged and published to the Red Hat Container Catalog and the embedded OperatorHub in OpenShift.

Follow the instructions to certify your Operator with Red Hat hosted pipeline:

Prerequisites

- Complete the Software Pre-certification Checklist available on the Red Hat Partner Connect website.

- On the Red Hat Partner Connect website, click your Project Name and navigate to the Settings tab.
  - In the Authorized GitHub user accounts field, enter your GitHub username to the list of authorized GitHub users.
  - If you are using a private container registry, from the OpenShift Object YAML field, click Add, to add a docker config.json secret and click Save.

Procedure
NOTE

Follow this procedure only if you want to run the Red Hat OpenShift Operator certification on the Red Hat hosted pipeline.

### 22.1. FORKING THE REPOSITORY

1. Log in to GitHub and fork the RedHat OpenShift operators upstream repository.

2. Fork the appropriate repositories from the following table, depending on the Catalogs that you are targeting for distribution:

<table>
<thead>
<tr>
<th>Catalog</th>
<th>Upstream Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Certified Catalog</td>
<td><a href="https://github.com/redhat-openshift-ecosystem/certified-operators">https://github.com/redhat-openshift-ecosystem/certified-operators</a></td>
</tr>
</tbody>
</table>

3. Clone the forked certified-operators repository.

4. Add the contents of your operator bundle to the operators directory available in your forked repository.

If you want to publish your operator bundle in multiple catalogs, you can fork each catalog and complete the certification once for each fork.

**Additional resources**

For more information about creating a fork in GitHub, see [Fork a repo](https://github.com/docs/github/fork-a-repo).

### 22.2. ADDING YOUR OPERATOR BUNDLE

In the operators directory of your fork, there are a series of subdirectories.

#### 22.2.1. If you have certified this operator before -

Find the respective folder for your operator in the operators directory. Place the contents of your operator Bundle in this directory.

**NOTE**

Make sure your package name is consistent with the existing folder name for your operator.

#### 22.2.2. If you are newly certifying this operator -

If the newly certifying operator does not have a subdirectory already under the operator’s parent directory then you have to create one.

Create a new directory under operators. The name of this directory should match your operator’s package name. For example, `my-operator`.
In this operators directory, create a new subdirectory with the name of your operator, for example, `<my-operator>` and create a version directory for example, `<V1.0>` and place your bundle. These directories are preloaded for operators that have been certified before.

- Under the version directory, add a `manifests` folder containing all your OpenShift manifests including your `clusterserviceversion.yaml` file.

**Recommended directory structure**

The following example illustrates the recommended directory structure.

```
├── config.yaml
│   ├── operators
│   │   └── my-operator
│   │       └── v1.4.8
│   │           ├── manifests
│   │           │       ├── cache.example.com_my-operators.yaml
│   │           │       ├── my-operator-controller-manager-metrics-service_v1_service.yaml
│   │           │       ├── my-operator-manager-config_v1_configmap.yaml
│   │           │       ├── my-operator-metrics-reader_rbac.authorization.k8s.io_v1_clusterrole.yaml
│   │           │       └── my-operator.clusterserviceversion.yaml
│   │           └── metadata
│   │                   └── annotations.yaml
│   └── ci.yaml
```

**Configuration file**

<table>
<thead>
<tr>
<th>Configuration file</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>config.yaml</td>
<td>In this file include the organization of your operator. It can be <code>certified-operators</code>. For example, <code>organization: certified-operators</code></td>
</tr>
<tr>
<td>ci.yaml</td>
<td>In this file include your Red Hat Technology Partner project ID and the organization target for this operator. For example, <code>cert_project_id: &lt;your partner project id&gt;</code>. This file stores all the necessary metadata for a successful certification process.</td>
</tr>
</tbody>
</table>
In this file include an annotation of OpenShift versions, which refers to the range of OpenShift versions. For example, **v4.8-v4.10** means versions 4.8 through 4.10. Add this to any existing content. For example, 

```yaml
# OpenShift annotations
com.redhat.openshift.versions: v4.8-v4.10.
```

The `com.redhat.openshift.versions` field, which is part of the metadata in the operator bundle, is used to determine whether an operator is included in the certified catalog for a given OpenShift version. You must use it to indicate one or more versions of OpenShift supported by your operator.

Note that the letter 'v' must be used before the version, and spaces are not allowed. The syntax is as follows:

- A single version indicates that the operator is supported on that version of OpenShift or later. The operator is automatically added to the certified catalog for all subsequent OpenShift releases.

- A single version preceded by '=' indicates that the operator is supported only on that specific version of OpenShift. For example, using `=v4.8` will add the operator to the certified catalog for OpenShift 4.8, but not for later OpenShift releases.

- A range can be used to indicate support only for OpenShift versions within that range. For example, using `v4.8-v4.10` will add the operator to the certified catalog for OpenShift 4.8 through 4.10, but not for OpenShift 4.11 or 4.12.

### Additional resources

- For more details, see [Managing OpenShift Versions](#).

- For an example of an operator Bundle, see [here](#).

### 22.3. CREATING A PULL REQUEST

The final step is to create a pull request for the targeted upstream repo.
If you want to publish your Operator bundle in multiple catalogs, you can create a pull request for each target catalog.

If you are not familiar with creating a pull request in GitHub you can find instructions [here](#).

**NOTE**

The title of your pull request must conform to the following format: `operator my-operator (v1.4.8)`. It should begin with the word `operator` followed by your Operator package name, followed by the version number in parenthesis. When you create a pull request it triggers the Red Hat hosted pipeline and provides an update through a pull request comment whenever it has failed or completed.

### 22.3.1. Guidelines to follow

- You can re-trigger the Red Hat hosted pipeline by closing and reopening your pull request.
- You can only have one open pull request at a time for a given Operator version.
- Once a pull request has been successfully merged it can not be changed. You have to bump the version of your Operator and open a new pull request.
- You must use the package name of your Operator as the directory name that you created under operators. This package name should match the package annotation in the annotations.yaml file. This package name should also match the prefix of the clusterserviceversion.yaml filename.
- Your pull requests should only modify files in a single Operator version directory. Do not attempt to combine updates to multiple versions or updates across multiple Operators.
- The version indicator used to name your version directory should match the version indicator used in the title of the pull request.
- Image tags are not accepted for running the certification tests, only SHA digest are used. Replace all references to image tags with the corresponding SHA digest.
CHAPTER 23. PUBLISHING THE CERTIFIED OPERATOR

The certification is considered complete and your Operator will appear in the Red Hat Container Catalog and embedded OperatorHub within OpenShift after all the tests have passed successfully, and the certification pipeline is enabled to submit results to Red Hat.

Additionally, the entry will appear on Red Hat Certification Ecosystem.

IMPORTANT

The Red Hat OpenShift software certification does not conduct testing of the Partner’s product in how it functions or performs outside of the Operator constructs and its impact on the Red Hat platform on which it was installed and executed. Any and all aspects of the certification candidate product’s quality assurance remains the Partner’s sole responsibility.
CHAPTER 24. TROUBLESHOOTING GUIDELINES

For troubleshooting tips and workarounds, see Troubleshooting the Operator Cert Pipeline.
APPENDIX B. HELM AND ANSIBLE OPERATORS

- For information on building a Helm operator, see Building a Helm Operator.
- For information on building an Ansible operator, see Building an Ansible Operator.
PART IV. HELM CHART CERTIFICATION
NOTE

Certify your container application project before proceeding with Red Hat Helm chart certification. All the containers referenced in a Helm chart project must already be certified and published on the Red Hat Ecosystem Catalog before certifying a Helm chart project.

25.1. INTRODUCTION TO HELM CHARTS

Helm is a Kubernetes-native automation technology and software package manager that simplifies deployment of applications and services. Helm uses a packaging format called charts. A chart is a collection of files that describe a related set of Kubernetes resources. A running instance of a specific version of the chart in a cluster is called a release. A new release is created every time a chart is installed on the cluster. Each time a chart is installed, or a release is upgraded or rolled back, an incremental revision is created. Charts go through an automated Red Hat OpenShift certification workflow, which guarantees security compliance as well as best integration and experience with the platform.

25.2. CERTIFICATION WORKFLOW FOR HELM CHARTS

NOTE

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

The following diagram gives an overview of testing a Helm chart:

**Task Summary**

The certification workflow includes three primary steps:
25.2.1. Certification on-boarding

Prerequisites

Verify the functionality of your product on the target Red Hat platform, in addition to the specific certification testing requirements. If running your product on the targeted Red Hat platform results in a substandard experience then you must resolve the issues prior to certification.

The Red Hat Partner Acceleration Desk (PAD) is a Products and Technologies level partner help desk service that allows our (prospective) technology partners a central location to ask non-technical questions about Red Hat offerings, partner programs, product certification, engagement process, and so on.

See [PAD - How to open & manage PAD cases](#), to open a PAD ticket.

Through the Partner Subscriptions program, Red Hat offers free, not-for-resale software subscriptions that you can use to validate your product on the target Red Hat platform. To request access to the program, follow the instructions on the [Partner Subscriptions](#) site.

You must construct your container images so that they meet the certification criteria and policy. For more details, see [image content requirements](#).

Procedure

Follow these high-level steps to certify your Helm chart:

1. Join the [Red Hat Partner Connect](#) for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Complete your company profile.
4. Create your certification project by selecting your desired platform, for example – Red Hat OpenShift and then choose Helm chart.
5. Complete the pre-certification checklist.

Additional resources

For detailed instructions about creating your first Helm chart project, see [Creating a Helm chart project](#).

25.2.2. Certification testing

Follow these high-level steps to run a certification test:

1. Fork the [Red Hat upstream repository](#).
2. Install and run the chart verifier tool on your test environment.
3. Review the test results and troubleshoot, if any issues.
4. Submit the certification results to Red Hat through a pull request.

Additional resources

For detailed instructions about certification testing, see Validating Helm charts for certification.

25.2.3. Publishing the certified Helm chart on the Red Hat Ecosystem Catalog

Certified helm charts are published on the Product Listings page of the Red Hat partner connect portal, which you can then run on a supported Red Hat container platform. Your product along with its Helm chart gets listed on the Red Hat Container Catalog using the listing information that you provide.

Additional resources

- For more details about publishing your certified Helm chart, see Publishing the certified Helm chart.

- For more information about Helm charts, see:
  - What is Helm?
  - Helm charts
  - Technical workshop: OpenShift Helm chart certification
CHAPTER 26. VALIDATING HELM CHARTS FOR CERTIFICATION

You can validate your Helm charts by using the chart-verifier CLI tool. Chart-verifier is a CLI based open source tool that runs a list of configurable checks to verify if your Helm charts have all the associated metadata and formatting required to meet the Red Hat Certification standards. It validates if the Helm charts are distribution ready and works seamlessly on the Red Hat OpenShift Container Platform and can be submitted as a certified Helm chart entry to the Red Hat OpenShift Helm chart repository.

The tool also validates a Helm chart URL and provides a report in YAML format with human-readable descriptions in which each check has a positive or negative result. A negative result from a check indicates a problem with the chart, which needs correction. You can also customize the checks that you wish to execute during the verification process.

NOTE

Red Hat strongly recommends using the latest version of the chart-verifier tool to validate your Helm charts on your local test environment. This enables you to check the results on your own during the chart development cycle, preventing the need to submit the results to Red Hat every time.

Additional resources

For more information about the chart-verifier CLI tool, see chart-verifier.

26.1. PREPARING THE TEST ENVIRONMENT

The first step towards certifying your product is setting up the environment where you can run the tests. To run the full set of chart-verifier tests, you require access to the Red Hat OpenShift cluster environment. You can install the chart-verifier tool and execute all the chart related tests in this environment. You can disable these tests by using several configurable command line options, but it is mandatory to run the tests for the certification to be approved by Red Hat.

NOTE

As an authorized Red Hat partner, you have free access to the Red Hat OpenShift Container Platform, and you can install a cluster in your own test environment using the Red Hat Partner Subscription (RHPS) program. To learn more about the benefits of software access as a part of the Red Hat Partner Connect program, see the program guide.

To set up your own test environment,

1. Install a fully managed cluster by using Red Hat Managed Services OpenShift cluster. This is a trial option that is valid only for 60 days.

2. Install a self-managed cluster that you can install in your cloud environment, datacenter or computer. Through this option you can use your partner subscriptions, also known as NFRs, for permanent deployments.

For more information on setting up your environment, see Try Red Hat OpenShift.

Additional resources
To know more about installing the cluster and configuring your helm charts, see:

- OpenShift Container Platform
- OpenShift cluster CLI Management
- Helm chart management in cluster

### 26.2. RUNNING THE HELM CHART-VERIFIER TOOL

The recommended directory structure for executing the chart-verifier tool is as follows:

```
├── src
│   ├── Chart.yaml
│   ├── README.md
│   ├── templates
│   │   ├── deployment.yaml
│   │   ├── _helpers.tpl
│   │   ├── hpa.yaml
│   │   ├── ingress.yaml
│   │   └── NOTES.txt
│   ├── serviceaccount.yaml
│   └── service.yaml
│       └── tests
│           └── test-connection.yaml
├── values.schema.json
└── values.yaml
```

**Prerequisites**

- A container engine in which Podman or Docker CLI is installed.
- Internet connection to check that the images are Red Hat certified.
- GitHub profile to submit the chart to the OpenShift Helm Charts Repository.
- Red Hat OpenShift Container Platform cluster.
- Before running the chart-verifier tool, package your Helm chart by using the following command:

```
$ helm package <helmchart folder>
```

This command will archive your Helm chart and convert it to a `.tgz` file format.

**Procedure**

You can run the full set of chart-verifier tool by using two methods:

- By using Podman or Docker
- By using the binary file (Linux only)

**26.2.1. By using Podman or Docker**
1. Run all the available checks for a chart available remotely using a universal resource identifier (uri), assuming that the kube config file is available at the location `${HOME}/.kube`:

```
$ podman run --rm -i                                  
  -e KUBECONFIG=/kube/config                   
  -v "${HOME}/.kube":/kube                     
  "quay.io/redhat-certification/chart-verifier" 
  verify                                        
  <chart-uri>
```

In this command, chart-uri is the location of the chart archive available on the https uri. Ensure that the archive must be in `.tgz` format.

2. Run all the available checks for a chart available locally on your system, assuming that the chart is available on the current directory and the kube config file is available at the location `${HOME}/.kube`:

```
$ podman run --rm
  -e KUBECONFIG=/kube/config                   
  -v "${HOME}/.kube":/kube                     
  -v $(pwd):/charts                             
  "quay.io/redhat-certification/chart-verifier" 
  verify                                        
  /charts/<chart>
```

In this command, chart-uri is the location of the chart archive available in your local directory. Ensure that the archive must be in `.tgz` format.

3. Run the following verify command to get the list of available options associated with the command along with its usage:

```
$ podman run -it --rm quay.io/redhat-certification/chart-verifier verify --help
```

The output of the command is similar to the following example:

```
Verifies a Helm chart by checking some of its characteristics

Usage:
    chart-verifier verify <chart-uri> [flags]

Flags:
  -S, --chart-set strings         set values for the chart (can specify multiple or separate values with commas: key1=val1,key2=val2)
  -G, --chart-set-file strings   set values from respective files specified via the command line (can specify multiple or separate values with commas: key1=path1,key2=path2)
  -X, --chart-set-string strings set STRING values for the chart (can specify multiple or separate values with commas: key1=val1,key2=val2)
  -F, --chart-values strings     specify values in a YAML file or a URL (can specify multiple)
    --debug                     enable verbose output
    --x, --disable strings      all checks will be enabled except the informed ones
    --e, --enable strings       only the informed checks will be enabled
    --helm-install-timeout duration  helm install timeout (default 5m0s)
    -h, --help                  help for verify
    --kube-apiserver string     the address and the port for the Kubernetes API server
    --kube-as-group stringArray group to impersonate for the operation, this flag can be
```
repeated to specify multiple groups.

- `--kube-as-user string` username to impersonate for the operation
- `--kube-ca-file string` the certificate authority file for the Kubernetes API server
- `--kube-context string` name of the kubeconfig context to use
- `--kube-token string` bearer token used for authentication
- `--kubeconfig string` path to the kubeconfig file
- `-n, --namespace string` namespace scope for this request
- `-V, --openshift-version string` set the value of certifiedOpenShiftVersions in the report
- `-o, --output string` the output format: default, json or yaml
- `-k, --pgp-public-key string` file containing gpg public key of the key used to sign the chart
- `-W, --web-catalog-only` set this to indicate that the distribution method is web catalog only (default: true)
- `--registry-config string` path to the registry config file (default: "/home/baiju/.config/helm/registry.json")
- `--repository-cache string` path to the file containing cached repository indexes (default: "/home/baiju/.cache/helm/repository")
- `--repository-config string` path to the file containing repository names and URLs (default: "/home/baiju/.config/helm/repositories.yaml")
- `-s, --set strings` overrides a configuration, e.g: `dummy.ok=false`
- `-f, --set-values strings` specify application and check configuration values in a YAML file or a URL (can specify multiple)
- `-E, --suppress-error-log` suppress the error log (default: written to "./chartverifier/verifier-<timestamp>.log")
- `--timeout duration` time to wait for completion of chart install and test (default: 30m0s)
- `-w, --write-to-file` write report to ".chartverifier/report.yaml" (default: stdout)

Global Flags:
- `--config string` config file (default is "/home/.chart-verifier.yaml")

4. Run a subset of the checks:

```bash
$ podman run --rm -i
\   -e KUBECONFIG=/kube/config
  -v "$HOME/.kube":/.kube
  "quay.io/redhat-certification/chart-verifier"
  verify -enable images-are-certified,helm-lint
  <chart-uri>
```

5. Run all the checks except a subset:

```bash
$ podman run --rm -i
\   -e KUBECONFIG=/kube/config
  -v "$HOME/.kube":/.kube
  "quay.io/redhat-certification/chart-verifier"
  verify -disable images-are-certified,helm-lint
  <chart-uri>
```

**NOTE**

Running a subset of checks is intended to reduce the feedback loop for development. To certify your chart, you must run all the required checks.

6. Provide chart-override values:
$ podman run --rm -i                                  
  -e KUBECONFIG=/kube/config                   
  -v "${HOME}/.kube":/.kube                     
  "quay.io/redhat-certification/chart-verifier"\ 
  verify --chart-set default.port=8080                   \ <chart-uri>

7. Provide chart-override values from a file located in the current directory:

$ podman run --rm -i                                  
  -e KUBECONFIG=/kube/config                   
  -v "${HOME}/.kube":/.kube                     
  -v $(pwd):/values                             
  "quay.io/redhat-certification/chart-verifier"\ 
  verify --chart-values /values/overrides.yaml              \ <chart-uri>

26.2.1. Configuring the timeout option

Increase the timeout value if the chart-testing process is delayed. By default, the chart-testing process takes about 30 minutes to complete.

$ podman run --rm -i                                  
  -e KUBECONFIG=/kube/config                   
  -v "${HOME}/.kube":/.kube                     
  -v $(pwd):/values                             
  "quay.io/redhat-certification/chart-verifier"\ 
  verify --timeout 40m                          \ <chart-uri>

NOTE

If you observe a delay in the chart-testing process, Red Hat recommends you to submit the report to the Red Hat certification team for verification.

26.2.1.2. Saving the report

When the chart-testing process is complete, the report messages are displayed by default. You can save the report by redirecting it to a file.

For example:

$ podman run --rm -i                                  
  -e KUBECONFIG=/kube/config                   
  -v "${HOME}/.kube":/.kube                     
  -v $(pwd):/values                             
  "quay.io/redhat-certification/chart-verifier"\ 
  verify --enable images-are-certified,helm-lint\ 
  <chart-uri> > report.yaml

Along with this command use the -w option to write the report directly to the file ./chartverifier/report.yaml. To get this file, you have to volume mount the file to /app/chartverifer.

For example:
$ podman run --rm -i                                  \
  -e KUBECONFIG=.kube/config                   \
  -v "${HOME}/.kube":/.kube                     \
  -v $(pwd)/chartverifier:/app/chartverifier    \
  -w                                            \n  "quay.io/redhat-certification/chart-verifier" \
  verify –enable images-are-certified,helm-lint \n  <chart-uri>

If the file already exists, it is overwritten by the new report.

26.2.1.3. Configuring the error log

By default, an error log is generated and saved to the file ./chartverifier/verify-<timestamp>.yaml. It includes the error messages, the results of each check and additional information about chart testing. To get a copy of the error log you have to volume mount the file to /app/chartverifer.

For example:

$ podman run --rm -i                                  \
  -e KUBECONFIG=.kube/config                   \
  -v "${HOME}/.kube":/.kube                     \
  -v $(pwd)/chartverifier:/app/chartverifier    \
  "quay.io/redhat-certification/chart-verifier" \
  verify –enable images-are-certified,helm-lint \n  <chart-uri> > report.yaml

If you want to store multiple logs to the same directory, you can store a maximum of 10 log files at a time. When the maximum file limit is reached, older log files are automatically replaced with the newer log files.

Use the -E or --suppress-error-log option to suppress the error log output.

NOTE

Error and warning messages are standard error output messages and are not suppressed by using the -E or --suppress-error-log option.

26.2.2. By using the binary file

NOTE

This method is applicable only for Linux systems.

1. Download and install the latest chart-verifier binary from the releases page.

2. Unzip the tarball binary by using the following command:

   $ tar zxfv <tarball>

3. Run the following command on the unzipped directory to perform all the Helm chart checks:

   $ ./chart-verifier verify <chart-uri>
In this command, **chart-uri** is the location of the chart archive available on your server. Ensure that the archive must be in `.tgz` format. By default, the chart-verifier tool assumes that the kube config file is available at the default location `$HOME/.kube`. Set the environment variable to `KUBECONFIG` if the file is not available at the default location.

The output of the chart-verifier includes the details of the tests executed along with a result status for each test. It also indicates whether each test is mandatory or recommended for Red Hat certification. For more detailed information, see [Types of Helm chart checks](#).

**Additional resources**

To know more about the chart-verifier tool, see [Helm chart checks for Red Hat OpenShift certification](#).
CHAPTER 27. CREATING A HELM CHART PROJECT

Prerequisites

Certify your chart’s container images as a container application project before creating a Helm chart project.

Procedure

1. Log in to Red Hat Partner Connect portal. The Access the partner portals web page displays.
2. Navigate to the Certified technology portal tile and click Log in for technology partners.
3. Enter the login credentials and click Login. The Red Hat Partner Connect web page displays.
4. On the page header, select Product certification and click Manage certification projects. My Work web page displays the Product Listings and Certification Projects, if available.
5. Click Create Project.
6. In the What platform do you want to certify on? dialog box, select the Red Hat OpenShift radio button and click Next.
7. In the What do you want to certify? dialog box, select Helm chart radio button and click Next.
8. On the Create Helm chart certification project web page, provide the following details to create your project.

   IMPORTANT
   You cannot change the project name and its distribution method after you have created the project.

   a. **Project Name:** Enter the project name. This name is not published and is only for internal use.

   b. **Chart name:** The name of your chart, which must follow Helm naming conventions.

   c. **Distribution Method** - Select one of the following options for publishing your Helm chart:

      i. **Helm chart repository charts.openshift.io** - The Helm chart is published to the Red Hat Helm chart repository, charts.openshift.io and the users can pull your chart from this repository.

      NOTE
      When you select the checkbox The certified helm chart will be distributed from my company’s repository, an entry about the location of your chart is added to the index of Red Hat Helm chart repository, charts.openshift.io.

      ii. **Web catalog only (catalog.redhat.com)** - The Helm chart is not published to the Red Hat Helm chart repository, charts.openshift.io and is not visible on Red Hat OpenShift
OperatorHub. This is the default option when you create a new project and this option is suitable for partners who do not want their Helm chart publicly installable within OpenShift, but require a proof of certification. Select this option only if you have a distribution, entitlement, or other business requirements that is not otherwise accommodated within the OpenShift In-product Catalog (Certified) option.

d. Click **Create project**

**Additional resources**

For more information on the distribution methods, see [Helm Chart Distribution methods](#).
CHAPTER 28. CONFIGURING THE HELM CHART PROJECT

When the project is created, your newly created Helm chart project web page displays.

The Helm chart web page comprises of the following tabs:

- **Overview** - Contains the pre-certification checklist.
- **Settings** - Allows you to configure the registry and repository details.

From the right of the Helm chart web page, click **Actions** menu to perform the following operations on the newly created Helm chart project:

- Open Support Case
- Archive Project

### 28.1. COMPLETE PRE-CERTIFICATION CHECKLIST

The **Overview** tab of the Helm chart project contains the pre-certification checklist. The pre-certification checklist consists of a series of tasks that you must complete to certify and publish your Helm chart.

Before you publish your Helm chart, perform the following tasks in the checklist:

- Complete your company profile
- Provide details for your helm chart
- Submit your chart through a pull request in GitHub
- Attach a completed product listing

#### 28.1.1. Complete your company profile

Keep your company profile up-to-date. This information gets published on the [Red Hat Ecosystem Catalog](https://redhatinc.com) along with your certified product.

To verify:

1. Navigate to **Complete your company profile** tile.
2. Click **Review** in your checklist.
3. To make any changes, click **Edit**.
4. Click **Submit**.

#### 28.1.2. Provide details for your Helm chart

1. Navigate to **Provide details for your Helm chart** tile to enter your repository details that are displayed on the [Red Hat Ecosystem Catalog](https://redhatinc.com), so that users can pull your Helm chart.
2. Click **Add details**. You are navigated to the **Settings** tab.
3. Enter all the required repository information.
4. After filling in all the details, click **Save**.

**NOTE**

All the fields marked with an asterisk * are required and must be completed before you can proceed with the Helm chart certification.

### 28.1.3. Submit your chart through a pull request in GitHub

After creating your Helm chart project on the Red Hat partner connect you have to submit your Helm chart for verification.

To submit your Helm chart:

1. Navigate to **Submit your chart through a pull request in GitHub** tile.
2. Click **Go to GitHub** You are redirected to the OpenShift Helm Charts Repository.
3. Submit a pull request.

The pull request is reviewed by the Red Hat certification team. After successful verification, your Helm chart is published on the Red Hat Ecosystem Catalog.

**Additional resources**

For detailed information on submitting your pull request, see 5. Submitting your Helm chart for certification.

### 28.1.4. Attach a completed product listing

This feature allows you to either create a new product listing, or to attach the project to an existing OpenShift product listing for your new project.

1. Navigate to the **Attach a completed product listing** tile.
2. From the Select method drop-down menu, select **Attach** or **edit**. The **Attach product listing** page displays.
3. Decide whether you want to attach your project to an existing product listing or if you want to create a new product listing:
   a. To attach your project to an existing product listing:
      i. From the Related product listing section, click **Select a product listing** drop-down arrow to select the required product listing.
      
      ii. Click **Save**.
   b. To create a new product listing:
      i. Click **Create new product listing**
      ii. In the **Product Name** text box, enter the required product name.
      iii. Click Save.
c. From the Select method drop-down menu, click View product listing to navigate to the new product listing and enter all the required product listing details.

d. Click Save.

NOTE

Make sure to complete all the items on the pre-certification checklist before submitting your application for certification.

After completing all the steps, a green check mark appears beside the tiles to indicate that configuration is complete.

28.2. MANAGING PROJECT SETTINGS

You can configure the registry and repository details through the Settings tab. When your Helm chart is verified, it is published on the Red Hat Ecosystem catalog along with the following details.

Enter the required details in the following fields:

NOTE

The following fields vary based on the selected distribution method.

- Chart name
- Container registry namespace - denotes the company name or abbreviation.
- Helm chart repository - denotes the location of your Helm chart repository.
- Any additional instructions for users to access your Helm chart - This information will be published on the Red Hat Ecosystem catalog.
- Public PGP Key - It is an optional field. Enter the key if you want to sign your certification test results.
- Authorized GitHub user accounts - denotes the GitHub users who are allowed to submit Helm charts for certification on behalf of your company.
- Short and Long repository descriptions and Application categories - This information will be used when listing your Helm chart on the Red Hat Ecosystem Catalog.
- Project Details - It includes your Project name, Technical contact and email address. This information will be used by Red Hat to contact you if there are any issues specific to your certification project.
- Click Save.

NOTE

All the fields marked with asterisk * are required and must be completed before you can proceed with Helm chart certification.
CHAPTER 29. SUBMITTING YOUR HELM CHART FOR CERTIFICATION

After configuring and setting up your Helm chart project on the Red Hat Partner Connect, submit your Helm charts for certification by creating a pull request to the Red Hat’s OpenShift Helm chart repository. In the pull request, you can either include your chart or the report generated by the chart-verifier tool or both. Based on the content of your pull request, the chart will be certified, and the chart-verifier will run if a report is not provided.

Prerequisites

Before creating a pull request, ensure to have the following prerequisites:

1. Fork the Red Hat’s OpenShift Helm chart repository and clone it to your local system. Here, you can see a directory already created for your company under the partner’s directory.

   NOTE
   The directory name is the same as the container registry namespace that you set while certifying your containers.

Within your company’s directory, there will be a subdirectory for each chart certification project you created in the previous step. To verify if this is set up correctly, review the OWNERS file. The OWNERS file is automatically created in your chart directory within your organization directory. It contains information about your project, including the GitHub users authorized to certify Helm charts on behalf of your company. You can locate the file at the location charts/partners/acme/awesome/OWNERS. If you want to edit the GitHub user details, navigate to the Settings page.

For example, if your organization name is acme and the chart name is awesome. The content of the OWNERS file is as follows:

```
chart:
  name: awesome
  shortDescription: A Helm chart for Awesomeness
  publicPgpKey: null
  providerDelivery: False
  users:
    - githubUsername: <username-one>
    - githubUsername: <username-two>
  vendor:
    label: acme
    name: ACME Inc.
```

The name of the chart that you are submitting must match the value in the OWNERS file.

2. Before submitting the Helm chart source or the Helm chart verification report, create a directory with its version number. For example, if you are publishing the 0.1.0 version of the awesome chart, create a directory as follows:

```
charts/partners/acme/awesome/0.1.0/
```
NOTE

For charts that represent a product supported by Red Hat, submit the pull request to the main branch with the OWNERS file located at the charts, redhat directory available in your organization directory. For example, for a Red Hat chart named awesome, submit your pull request to the main branch located at charts/redhat/redhat/awesome/OWNERS. Note that for Red Hat supported projects, your organization name is also redhat.

Procedure

You can submit your Helm chart for certification by using three methods:

1. Submit a Helm chart without the chart verification report
2. Submit a chart verification report without the Helm chart
3. Submit a chart verification report along with the Helm chart

29.1. SUBMITTING A HELM CHART WITHOUT THE CHART VERIFICATION REPORT

You can submit your Helm chart for certification without the chart verification report in two different formats:

29.1.1. Chart as a tarball

If you want to submit your Helm chart as a tarball, you can create a tarball of your Helm chart using the Helm package command and place it directly in the 0.1.0 directory.

For example, if your Helm chart is awesome for an organization acme

chart/partners/acme/awesome/0.1.0/awesome-0.1.0.tgz
chart/partners/acme/awesome/0.1.0/awesome-0.1.0.tgz.prov

29.1.2. Chart in a directory

If you want to submit your Helm chart in a directory, place your Helm chart in a directory with the chart source.

If you have signed the chart, place the providence file in the same directory. You can include a base64 encoded public key for the chart in the OWNERS file. When a base64 encoded public key is present, the key will be decoded and specified when the chart-verifier is used to create a report for the chart.

If the public key does not match the chart, the verifier report will include a check failure, and the pull request will end with an error.

If the public key matches with the chart and there are no other failures, a release will be created, which will include the tarball, the providence file, the public key file, and the generated report.

For example,

awesome-0.1.0.tgz
awesome-0.1.0.tgz.prov
If the **OWNERS** file does not include the public key, the chart verifier check is skipped and will not affect the outcome of the pull request. Further, the public key file will not be included in the release.

If the chart is a directory with the chart source, create a src directory to place the chart source.

For example,

A Path can be charts/partners/acme/awesome/0.1.0/src/

And the file structure can be

```plaintext
├── src
│   ├── Chart.yaml
│   ├── README.md
│   │   ├── templates
│   │       ├── deployment.yaml
│   │       │   ├── _helpers.tpl
│   │       │   ├── hpa.yaml
│   │       │   └── ingress.yaml
│   │       │       └── NOTES.txt
│   │       │       └── serviceaccount.yaml
│   │       │           └── service.yaml
│   │       │           └── tests
│   │       │               └── test-connection.yaml
│   │   ├── values.schema.json
│   │   └── values.yaml
└── src
```

### 29.2. SubmittinG A CHART VERIFICATION REPORT WITHOUT THE HELM CHART

Generate the report using the **chart-verifier** tool and save it with a file name report.yaml in the directory 0.1.0. You can submit two types of reports:

#### 29.2.1. For submitting a signed report

Before submitting your report for certification, you can add a **PGP public key** to the chart verification report. Adding a **PGP public key** is optional. When you add it to your report, you can find your public key in the **OWNERS** file under your chart directory within your organization directory. The **PGP public key** is available in the **publicPgpKey** attribute. The value of this attribute must follow **ASCII armor format**.

When submitting a chart verification report without the chart, you can sign your report and save the signature in **ASCII armor format**.

For example,

```
gpg --sign --armor --detach-sign --output report.yaml.asc report.yaml
```

NOTE

You can see a warning message on the console if the signature verification fails.
29.2.2. For submitting a report for a signed chart

For submitting the chart verification report for a signed chart, when you provide a PGP public key to the chart verifier tool while generating the report, it includes a digest of the key along with the report.

Also, when you include a base64 encoded PGP public key to the OWNERS file, a check is made to confirm if the digest of the decoded key in the OWNERS file matches the key digest in the report.

When they do not match, the pull request fails. But if the key digest matches with the report and there are no other errors when processing the pull request, a release is generated containing the public key and the report.

For example,

```bash
awesome-0.1.0.tgz.key
report.yaml
```

NOTE

A release is not generated if you have enabled the provider control delivery.

29.3. SUBMITTING A CHART VERIFICATION REPORT ALONG WITH THE HELM CHART

You can also submit a chart along with the report. Follow Submitting a Chart without Chart Verification Report procedure and place the source or tarball in the version number directory. Similarly, follow the steps in Submitting a Chart Verification Report without the Chart and place the report.yaml file in the same version number directory.

29.3.1. For submitting a signed report

You can sign the report and submit for verification. You can see a warning message on the console if the signature verification fails. For more information, see, ‘For submitting a signed report’ section of Submitting a Chart Verification Report without the Chart.

29.3.2. For submitting a signed Helm chart

For a signed chart you must include a tarball and a providence file in addition to the report file. For more information, see, ‘For submitting a report for a signed chart’ section of Submitting a Chart Verification Report without the Chart.

29.4. SUMMARY OF CERTIFICATION SUBMISSION OPTIONS

Follow the table that summarizes the scenarios for submitting your Helm charts for certification, depending on how you want to access your chart and also to check whether the chart tests have some dependencies on your local environment.
<table>
<thead>
<tr>
<th>Objective</th>
<th>Include Helm chart</th>
<th>Include chart verification report</th>
<th>Red Hat certification outcome</th>
<th>Methods to publish your certified Helm chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you want to perform the following actions:</td>
<td>Yes</td>
<td>No</td>
<td>The chart-verifier tool is executed in the Red Hat CI environment to ensure compliance.</td>
<td>Your customers can download the certified Helm charts from charts.openshift.io.</td>
</tr>
<tr>
<td>- Store your certified chart at charts.openshift.io.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Take advantage of Red Hat CI for ongoing chart tests</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If you want to perform the following actions:</td>
<td>Yes</td>
<td>Yes</td>
<td>The Red Hat certification team reviews the results to ensure compliance.</td>
<td>Your customers can download the certified Helm charts from charts.openshift.io.</td>
</tr>
<tr>
<td>- Store your certified chart at charts.openshift.io.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Aim to test your chart in your own environment since it has some external dependencies.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Objective</td>
<td>Include Helm chart</td>
<td>Include chart verification report</td>
<td>Red Hat certification outcome</td>
<td>Methods to publish your certified Helm chart</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------</td>
<td>----------------------------------</td>
<td>-----------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>If you don’t want to store your certified charts at charts.openshift.io.</td>
<td>No</td>
<td>Yes</td>
<td>The Red Hat certification team reviews the results to ensure compliance.</td>
<td>Your customers can download the certified Helm chart from your designated Helm chart repository. A corresponding entry is added to the index.yaml file at charts.openshift.io.</td>
</tr>
</tbody>
</table>

### 29.5. VERIFICATION STEPS

After submitting the pull request, it will take a few minutes to run all the checks and merge the pull request automatically. Perform the following steps after submitting your pull request:

1. Check for any messages in the new pull request.

2. If you see an error message, see Troubleshooting Pull Request Failures. Update the pull request accordingly with necessary changes to rectify the issue.

3. If you see a success message, it indicates that the chart repository index is updated successfully. You can verify it by checking the latest commit in the gh-pages branch. The commit message is in this format:

   ```
   <partner-label>-<chart-name>-<version-number> index.yaml (#<PR-number>) (e.g, acme-psql-service-0.1.1 index.yaml (#7)).
   ```

   You can see your chart related changes in the index.yaml file.

4. If you have submitted a chart source, a GitHub release with the chart and corresponding report is available on the GitHub releases page. The release tag is in this format: `<partner-label>-<chart-name>-<version-number>` (e.g., acme-psql-service-0.1.1).

5. You can find the certified Helm charts on the Red Hat’s official Helm chart repository. Follow the instructions listed here to install the certified Helm chart on your OpenShift cluster.
CHAPTER 30. PUBLISHING THE CERTIFIED HELM CHART

When you submit the Helm chart for validation through a pull request, the Red Hat certification team reviews and verifies your project for certification. After successful validation, your Helm chart project is certified through GitHub.

Follow the steps to publish your certified Helm chart:

1. Access the Partner connect web page. My Work web page displays the Product Listings and Certification Projects.

2. Navigate to the Product Listings tab and search for the required product listing.

3. Click the newly created product listing that you wish to publish. Review all the details of your product listing.

4. From the left pane, navigate to the Certification Projects tab.

5. Click Attach Project to attach your certified Helm chart to this listing. Also add the certified container project used by your Helm chart. Both the projects must be in Published status. The Publish button is enabled when you specify all the required information for the product listing along with the attached projects.

6. Click Publish.

Your certified Helm chart is now available for public access on the Red Hat Ecosystem Catalog.
PART V. FUNCTIONAL CERTIFICATION FOR OPENSHIFT BADGES: CNF, CNI, CSI

Red Hat OpenShift certification badges extend the Red Hat OpenShift Operator certification to specific functional areas or infrastructure services that are key in cloud-native deployments for our joint customers. These certifications are built on the foundation of container and operator certification, and by receiving a Red Hat OpenShift Certification Badge, partners can confirm that their solution is Kubernetes enabled and utilizes specific Kubernetes APIs for addressing the respective use cases.

The current OpenShift certification badges that are available are as follows:

- **Cloud-Native Network Functions (CNF)** — for the implementation of telecommunication functions deployed as containers.

- **Container Networking Interface (CNI)** — for the delivery of networking services through a pluggable framework.

- **Container Storage Interface (CSI)** — for providing and supporting a block or file persistent storage backend for Red Hat OpenShift.
CHAPTER 31. CNF CERTIFICATION AND VENDOR VALIDATION

31.1. WORKING WITH CLOUD-NATIVE NETWORK FUNCTION (CNF) CERTIFICATION

31.1.1. Introduction to Cloud-native Network Function

Cloud-native Network Functions (CNFs) are containerized instances of classic physical or Virtual Network Functions (VNFs) that have been decomposed into microservices supporting elasticity, lifecycle management, security, logging, and other capabilities in a Cloud-native format.

The CNF badge is a specialization within Red Hat OpenShift certification. It is available for products that implement a network function delivered in a container format with Red Hat OpenShift as the deployment platform. Red Hat offers two levels of CNF certification:

- **Vendor Validation** - Select this type of CNF certification, if your container base image is neither RHEL nor UBI. For this type of certification, Vendor Validate your CNF product by testing it internally, before publishing it as a Vendor Validated CNF product on the Red Hat Ecosystem catalog.

- **Certified CNF** - Select this type of CNF certification, if your container base image is RHEL or UBI. For this type of certification, Vendor Validate your CNF product, run the certification tests on your workload and then submit it for verification. After successful verification your CNF product gets listed as a Certified CNF product on the Red Hat Ecosystem catalog.

Products that meet the requirements and complete the certification workflow get listed on the Red Hat Ecosystem Catalog and are identified with the CNF badge. Partners will receive a logo to promote their product certification.

Additional resources

- For more information about CNF, see:
  - CNF and VNF certifications
  - About cloud-native network functions
  - Building CNF applications with OpenShift Pipelines

- To know more about the advantages of Vendor Validated CNF and Certified CNF, see Cloud-native network functions (CNF).

- To know about the requirements for pursuing a CNF certification, see Requirements for CNF.

- To know more about the best practices and common recommendations for OLM and SDK projects, see Operator Best Practices.

31.1.2. Certification workflow for CNF

**NOTE**

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.
The following diagram gives an overview of the certification process:

**Figure 31.1. Certification workflow for CNF project**

### Task Summary

The certification workflow includes the following three primary stages—

1. Section 4.1, “Certification onboarding and opening your first project”

2. Section 31.1.2.2, “Completing the checklist”

3. Section 31.1.2.3, “Publishing the CNF product listing on the Red Hat Ecosystem Catalog”

#### 31.1.2.1. Certification onboarding and opening your first project

**Prerequisites**

Ensure to check if your product meets the following requirements before proceeding with the certification process:

- Your product is generally available for public access.

- Your product is tested and deployed on Red Hat OpenShift.

- Your product is commercially supported on Red Hat OpenShift.

Verify the functionality of your product on the target Red Hat platform, in addition to the specific certification testing requirements. If running your product on the targeted Red Hat OpenShift Container platform results in a substandard experience, then you must resolve the issues prior to certification.

The Red Hat Partner Acceleration Desk (PAD) is a Products and Technologies level partner help desk service that allows our (prospective) technology partners a central location to ask non-technical questions pertaining to Red Hat offerings, partner programs, product certification, engagement process, and so on.
See PAD - How to open & manage PAD cases, to open a PAD ticket.

Through the Partner Subscriptions program, Red Hat offers free, not-for-resale software subscriptions that you can use to validate your product on the target Red Hat platform. To request access to the program, follow the instructions on the Partner Subscriptions site.

**NOTE**

Before proceeding with the certification, Red Hat recommends checking your container images and operators or Helm charts to see if they meet the certification criteria and policy. For more details, see Image content requirements, Operator requirements and Helm chart requirements.

**Procedure**

Perform the steps outlined for the certification onboarding:

1. Join the Red Hat Connect for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Complete your company profile.
4. Create your certification project by selecting your desired platform, for example- Red Hat OpenShift and then choose CNF Project.

**NOTE**

Create individual CNF projects for each partner product version and its corresponding Red Hat base version. If you want to certify your CNF project then create separate CNF projects for each attached CNF component such as container images and operator bundle or Helm chart.

**Additional resources**

For detailed instructions about creating your CNF project, see Creating a CNF project.

**31.1.2.2. Completing the checklist**

Perform the steps outlined for completing the checklist:

1. Provide details for your validation.
2. Complete your company profile.
3. Attach a completed product listing.
4. Validate the functionality of your CNF on Red Hat OpenShift for Vendor Validation.
5. Complete the Certification checklist for certifying your CNF project.

**Additional resources**

For more details about the checklist, see Configuring the CNF project.
31.1.2.3. Publishing the CNF product listing on the Red Hat Ecosystem Catalog

The Certified or Vendor Validated CNF project must be added as a component to your product’s Product Listing page on the Red Hat Partner Connect portal. Once published, your product listing is displayed on the Red Hat Ecosystem Catalog, by using the product information that you provide. You can publish both the Vendor Validated and Certified CNF products on the Red Hat Ecosystem Catalog with the respective labels.

Additional resources

- For more details about publishing your CNF project, see Publishing the CNF project.

31.2. CREATING A CNF PROJECT

Procedure

1. Log in to Red Hat Partner Connect portal.
   The Access the partner portals web page displays.

2. Navigate to the Certified technology portal tile and click Log in for technology partners.

3. Enter the login credentials and click Login.
   The Red Hat Partner Connect web page displays.

4. On the page header, select Product certification and click Manage certification projects.
   My Work web page displays the Product Listings and Certification Projects, if available.

5. Click Create Project.

6. In the What platform do you want to certify on? dialog box, select the Red Hat OpenShift radio button and click Next.

7. In the What do you want to certify? dialog box, select Cloud-native Network Function (CNF) radio button and click Next.

8. On the Create Cloud-native Network Function (CNF) certification project web page, provide the following details to create your project.

   a. Project Name: Enter the project name. This name is not published and is only for internal use. To change the project name after you have created the project, navigate to the Settings tab.

   NOTE

   Red Hat recommends including the product version to the project name to aid easy identification of the newly created CNF project. For example,
   
   `<CompanyName ProductName> 1.2 - OCP 4.12.2`.

   b. Click Create project.
Create individual CNF projects for each partner product version and its corresponding Red Hat base version. Also, create separate CNF projects for each attached certification component, such as container images, operator bundle, or Helm chart. You can create more than one CNF project for a product.

31.3. CONFIGURING THE CNF PROJECT

When the project is created, your newly created CNF chart project web page displays.

The CNF project web page comprises of the following tabs:

1. **Overview** - Contains the pre-publication and the certification checklists.
2. **Settings** - Allows you to configure the project details.

From the right of the CNF project web page, click **Actions** menu to perform the following operations, if you require now or in the future, on the newly created CNF project:

- Open Support Case
- Archive Project

31.3.1. Complete the checklist

The **Overview** tab of the CNF project contains the pre-publication and certification checklists. These checklists consist of a series of tasks that you must complete, to publish your CNF project.

Before you publish your CNF project, perform the following tasks:

1. Section 31.3.1.1, “Complete the pre-publication checklist for Vendor Validation”
2. Section 31.3.1.2, “Complete the Certification checklist to certify the Vendor Validated CNF project”

31.3.1.1. Complete the pre-publication checklist for Vendor Validation

31.3.1.1.1. Complete your company profile

Keep your company profile up-to-date. This information gets published on the Red Hat Ecosystem Catalog along with your certified product.

To verify your company profile, perform the following:

1. Navigate to the **Complete your company profile** tile and click **Edit**.
2. After filling in all the details, click **Submit**.
3. To make any changes, select the tile and click **Review**. The **Account Details** page displays wherein you can review and modify the entered **Company profile** information.
NOTE

All the fields marked with an asterisk * are required and must be completed before you can proceed with the pre-publication checklist.

31.3.1.1.2. Provide details about your validation

Navigate to the Provide details about your validation tile, to enter your project details that are displayed on the Red Hat Ecosystem Catalog, so that users can pull your CNF project.

To provide details about your validation, perform the following:

1. Click Start. You are navigated to the Settings tab.
2. Enter all the required project details.
3. After filling in all the details, click Save.
4. To make any changes, select the tile and click Edit.

31.3.1.1.3. Attach a completed product listing

This feature allows you to either create a new product listing, or to attach the project to an existing OpenShift product listing for your new CNF project.

1. Navigate to the Attach a completed product listing tile.
2. From the Select method drop-down menu, select Attach or edit. The Attach product listing page displays.
3. Decide whether you want to attach your project to an existing product listing or if you want to create a new product listing:
   a. To attach your project to an existing product listing:
      i. From the Related product listing section, click Select a product listing drop-down arrow to select the required product listing.
      ii. Click Save.
   b. To create a new product listing:
      i. Click Create new product listing
      ii. In the Product Name text box, enter the required product name.
      iii. Click Save.
   c. From the Select method drop-down menu, click View product listing to navigate to the new product listing and enter all the required product listing details.
   d. Click Save.

31.3.1.1.4. Validate the functionality of your CNF on Red Hat OpenShift

This feature allows the Red Hat CNF certification team to check if your product meets all the standards for Vendor Validation.
To validate the functionality of your CNF project, perform the following:

1. Select this option and click **Start questionnaire**. The **CNF Questionnaire** page displays.

2. Enter all your product and company information.

3. After filling in all the details, click **Submit**.

4. To make any changes, select the tile and click **Review process**. The **CNF Questionnaire** page displays, allowing you to review and modify the entered information.

After you click **Submit**, a new functional certification request is created. The Red Hat CNF certification team will review and validate the entered details of the CNF questionnaire. After successful review and validation, your functional certification request will be approved, and the **Certification Level** field in the Product Listing will be set to **Vendor Validated**.

After completing each step, a green check mark will appear beside each tile to indicate that particular configuration item is complete. When all items are completed in the checklist, the disclosure caret to the left of **Pre-publication Checklist** will be closed.

**Additional resources**

For detailed information about the validation process, see **CNF workflow**.

### 31.3.1.2. Complete the Certification checklist to certify the Vendor Validated CNF project

**NOTE**

Select this option only if you want to certify your CNF project.

This is an optional feature that allows you to certify your Vendor Validated project by using the Red Hat certification tool. For every Vendor Validated project, a new functional certification request will be created on the Red Hat Partner Certification portal. When you place a request for certification, your functional certification request will be processed by the CNF team for certification.

If you certify your Vendor Validated CNF project then it will be displayed on the **Red Hat Ecosystem Catalog** with the **Certified** label.

**Prerequisites**

1. Complete the Pre-Publication checklist before proceeding with the Certification checklist.

2. Certify your attached container images, operator bundles or helm charts before submitting your CNF project for certification.

**Procedure**

To certify your Vendor Validated CNF project, perform the following steps:

1. Navigate to the **Certify the functionality tool to certify your CNF** tile and click **Start**. A new functional certification request is created and will be redirected to your project on the **Red Hat Partner Certification (rhcert)** portal.

2. Run the **CNF certification test suite** or use **DCI OpenShift App Agent**. It consists of a series of test cases derived from best practices to evaluate whether your product adheres to these principles and satisfies the Red Hat certification standards.
3. To certify your CNF project, perform the following steps on your CNF project page on the Red Hat Partner Certification (rhcert) portal:

a. Navigate to the Summary tab,

i. To submit your CNF certification test results, from the Files section click Upload. Select the claims.json and tnf_config.yml files. Then, click Next. A successful upload message is displayed.

ii. Add your queries related to certification, if any, in the Discussions text box.

iii. Click Add Comment. By using this option, you can communicate your questions to the Red Hat CNF certification team. The Red Hat CNF certification team will provide clarifications for your queries.

b. In the Summary tab,

i. Navigate to the Partner Product category.

ii. Click the edit icon below the Partner Product Version option to enter your product version and then click the checkmark button. Your product version gets updated.

c. Navigate to the Properties tab,

i. Click the Platform list menu to select the platform on which you want to certify your CNF project. For example - x86_64

ii. Click the Product Version list menu to select the Red Hat product version on which you want to certify your CNF project. For example - Red Hat OpenShift Platform

iii. Click Update Values. The selected values are updated.

**NOTE**

All the versions of partner products are not certified for use with every version of Red Hat products. You need to certify each version of your product with the selected Red Hat base version. For example, if you certify your product version 5.11 with Red Hat OpenShift Container Platform version 4.13, you can use only the 5.11 version and not the later versions. Therefore, certify every version of your product individually with the latest version of the Red Hat base product.

The Red Hat CNF certification team will review and verify the details of your CNF project. When the Red Hat CNF certification team identifies issues or violations in the recommended best practices with your CNF, joint discussions will ensue to find the remediation options and timeline. The team also considers temporary exceptions if there is a commitment to fix the issues with an identified release target or timeline. All exceptions will be documented and published in a KIE base article listing all non compliant items before CNF gets listed on the Red Hat Ecosystem Catalog but the technical details will remain private.

**NOTE**

All the containers, operators or Helm charts referenced in your CNF project must be recertified before beginning to certify a CNF project in the prescribed order.
After successful verification by the Red Hat CNF certification team, your Vendor Validated CNF project will become certified, and will be automatically published on the Red Hat Ecosystem Catalog with the Certified label.

Additional resources

1. For more information about the CNF Certification test suite, see Overview and test catalog.
2. For more information about installing and configuring DCI OpenShift App Agent, see DCI OpenShift App Agent.

31.3.2. Managing Project settings

You can configure the CNF project details through the Settings tab. When your CNF project gets successfully verified by the Red Hat CNF certification team, your product listing is published on the Red Hat Ecosystem catalog along with the following details:

1. Project Details - This includes your Project name and Technical contact email address. This information will be used by Red Hat to contact you if there are any issues specific to your certification project.

   NOTE

   Red Hat recommends including the product version to the project name to aid easy identification of the newly created CNF project. For example, <CompanyName ProductName> 1.2 - OCP 4.12.2.

2. Click Add new contact, if you want to add an Additional Technical contact email address.
3. Click Save.

   NOTE

   All the fields marked with an asterisk * are required and must be completed before you can save your changes on this page.

31.4. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit your CNF project for validation after completing the pre-publication checklist, the Red Hat CNF certification team will review and verify the entered details of the CNF Questionnaire. If you want to certify your Vendor Validated CNF project, complete the Certification checklist.

The Red Hat certification team will review the submitted CNF test results. After successful verification, to publish your product on the Red Hat Ecosystem Catalog, navigate to the Product Listings page to attach the Vendor Validated or Certified CNF project.

Follow these steps to publish your product listing:

2. Navigate to the Product Listings tab and search for the required product listing.
3. Click the newly created product listing that you want to publish. Review all the details of your product listing.

4. From the left pane, navigate to the Versions & Certifications tab.

5. Click Attach New Version to add new project versions to your product listing.

   **NOTE**
   You must publish all the attached project versions before publishing your product version and product certification.

6. From the left pane, navigate to the Certification Projects tab.

7. Click Attach Project to attach your Vendor Validated or Certified CNF project to this listing. While attaching a certified CNF project, it is mandatory to add the certified container image and an operator bundle or Helm chart project used by your CNF project.

   **NOTE**
   All the attached projects must be in Published status.

   For Vendor Validated projects, this step is not required. The Publish button is enabled when you specify all the required information for the product listing, including the attached projects.

8. Click Publish.

Your new CNF product listing is now available for public access with respective Vendor Validated or Certified CNF labels on the Red Hat Ecosystem Catalog. The Certifications table on your product listings page displays the following details:

- **Product** - for example, Red Hat OpenShift Container Platform
- **Version** - selected Red Hat base product version. for example, 4.12 - 4.x
- **Architecture** - for example, x86_64
- **Partner product version** - for example, 5.11
- **Certification type** - for example, RHOCP 4 CNF
- **Level** - for example, Vendor Validated or Certified

You need to certify each version of your product with the selected Red Hat base version. Hence the Certifications table can have multiple versions of your product for the same Red Hat base version. For example,
### 31.5. RECERTIFYING A CNF PACKAGE

Recertification workflow is similar to the [regular CNF certification workflow](#). Red Hat recommends to recertify your application in the following scenarios:

- on every major release of the Red Hat OpenShift Container Platform.
- on every major release of your application.

**NOTE**

To recertify your application, it is mandatory to create a new certification request for recertification.

**Procedure**

1. Create a new CNF project on the [Red Hat Partner Connect](#).
2. In the **Project name** field enter the product name and its version. For example - `<CompanyName ProductName> 1.2 - OCP 4.12.2`
3. Complete the [pre-publication checklist](#), except the **CNF Questionnaire** and proceed with the [regular CNF certification workflow](#), like a new certification.
4. Submit a new certification request through the [Red Hat Partner Certification (rhcert) portal](#).
NOTE

Recertify the entire CNF package along with recertifying each CNF component individually as a standalone CNF. In case, if you have deployed your CNF application by using a separate helm chart or an operator, you must recertify each CNF separately. Also, if you are recertifying a new version of your CNF product on the same Red Hat product version, you don’t have to recertify the unaltered CNF image containers.

After successful verification by the Red Hat CNF certification team, the new version of your CNF package is recertified, and will be automatically published on the Red Hat Ecosystem Catalog with the Certified label.
CHAPTER 32. CNI CERTIFICATION

32.1. WORKING WITH CONTAINER NETWORK INTERFACE (CNI) CERTIFICATION

NOTE

Deploy and manage the CNI plug-in through an Operator before proceeding with the Red Hat CNI certification. Your operator and all the containers referenced in an Operator Bundle must already be certified and published on the Red Hat Ecosystem Catalog prior to certifying the CNI plug-in.

32.1.1. Introduction to Container Network Interface

Container Network Interface (CNI) is a specification to configure network interfaces in Linux containers. It consists of a specification and libraries for writing plug-ins to configure network interfaces in Linux containers, along with a number of supported plug-ins. CNI is mainly helpful with adding, connecting, deleting and disconnecting containers to networks.

The CNI badge is a specialization within Red Hat OpenShift certification. It is available to products that implement a network function delivered in a container format by using a CNI plug-in with Red Hat OpenShift as the deployment platform.

The products that meet the requirements and complete the certification workflow can be referred to and promoted as Certified CNI on the Red Hat OpenShift Container platform. After the certification is approved, the certified CNI product will be listed on the Red Hat Ecosystem Catalog as well as on the OperatorHub within the web console in Red Hat OpenShift. The certified CNI operators are identified with the CNI badge. Partners will receive a logo to promote their product certification.

Additional resources

- For more information about CNI, see:
  - Certified OpenShift CNI Plug-ins
  - CNI Specification
  - A brief overview of the Container Network Interface (CNI) in Kubernetes
  - Kubernetes CNI

32.1.2. Certification workflow for CNI

NOTE

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

The following diagram gives an overview of the certification process:
Task Summary

The certification workflow includes three primary steps-

1. Section 32.1.2.1, “Certification on-boarding and creating your first project”
2. Section 4.2, “Certification testing”
3. Section 32.1.2.3, “Publishing the CNI product listing on the Red Hat Ecosystem Catalog”

32.1.2.1. Certification on-boarding and creating your first project

Prerequisites

Verify the functionality of your product on the target Red Hat platform, in addition to the specific certification testing requirements. If running your product on the targeted Red Hat platform results in a substandard experience then you must resolve the issues prior to certification.

The Red Hat Partner Acceleration Desk (PAD) is a Products and Technologies level partner help desk service that allows our (prospective) technology partners a central location to ask non-technical questions about Red Hat offerings, partner programs, product certification, engagement process, and so on.

See PAD - How to open & manage PAD cases, to open a PAD ticket.

Through the Partner Subscriptions program, Red Hat offers free, not-for-resale software subscriptions that you can use to validate your product on the target Red Hat platform. To request access to the program, follow the instructions on the Partner Subscriptions site.

You must construct your container images so that they meet the certification criteria and policy. For more details, see image content requirements and Operator requirements.

Procedure
Follow these high-level steps to certify your CNI operator:

1. Join the Red Hat Partner Connect for Technology Partner Program.
2. Agree to the program terms and conditions.
3. Complete your company profile.
4. Create your certification project by selecting your desired platform, for example, Red Hat OpenShift and then choose Operator Bundle Image.
5. Complete the Pre-certification checklist.

Additional resources

For detailed instructions about creating and configuring your first CNI project, see Creating an Operator bundle project and Configuring the Operator bundle.

32.1.2.2. Certification testing

To run the certification test,

1. Fork the Red Hat upstream repository.
2. Install and run the Red Hat certification pipeline on your test environment.
3. Review the test results and troubleshoot, if any issues.
4. Submit the certification results to Red Hat through a pull request.

32.1.2.3. Publishing the CNI product listing on the Red Hat Ecosystem Catalog

When you complete all the certification checks successfully, you can submit the test results to Red Hat. You can turn on or off this results submission step depending on your individual goals. When the test results are submitted, it triggers the Red Hat infrastructure to automatically merge your pull request and publish your operator.

32.2. CREATING A CNI PROJECT

For detailed instructions about creating your first CNI project, see Creating an Operator bundle project.

32.3. CONFIGURING THE CNI PROJECT

For detailed instructions about configuring your CNI project settings and completing the Pre-certification checklist, see Configuring the Operator bundle.

32.4. WORKING WITH THE OPENSHIFT OPERATOR PIPELINE

Follow the steps to certify your operator before proceeding with the CNI certification:

1. Adding your Operator bundle
2. Forking the repository
3. Installing the OpenShift Operator Pipeline

4. Executing the OpenShift Operator pipeline

5. Submit certification results

Certified operators are listed in and consumed by customers through the embedded OpenShift operatorHub, providing them the ability to easily deploy and run your solution. Additionally, your product and operator image will be listed on the Red Hat Ecosystem Catalog.

### 32.5. CONFIGURING YOUR TEST ENVIRONMENT FOR RUNNING THE CNI TESTS

Before running the CNI tests, verify if the Red Hat OpenShift environment that you use for running the CNI tests meets the following criteria:

1. Use a Red Hat OpenShift version within the full support phase of the life cycle. Red Hat recommends using the latest supported release. For more information on OpenShift releases, see [Red Hat OpenShift Container Platform Life Cycle Policy](#).

2. Deploy the networking product only by using the documented installation procedure.

3. Install the Red Hat OpenShift Virtualization.

4. If you want to test Service Mesh, install the Red Hat Service Mesh operator and the Service Mesh Control Plane (SMCP).

5. Configure a host with access to the OpenShift cluster to use as the test client for running the CNI certification tests. This environment must include the utilities such as gcc, git, go, make, openssl and [sonobuoy](#).

**IMPORTANT**

Run your own product verification tests on the same configuration to ensure that your product functionality works as expected on the Red Hat OpenShift environment.

[Red Hat Partner Connect](#) offers free access to software as a program benefit. To know how to obtain a subscription to the Red Hat OpenShift environment, see [Red Hat Partner Connect Program Guide](#).

### 32.6. RUNNING THE CNI TESTS

1. To run the Red Hat OpenShift networking conformance test suite, place the `kubeconfig.yaml` file in the current working directory and run the following command:

   ```
   $ podman run -v `pwd`:/data:z --rm -it registry.redhat.io/openshift4/ose-tests sh -c "KUBECONFIG=/data/kubeconfig.yaml /usr/bin/openshift-tests run openshift/network/third-party -o /data/results.txt"
   ```

   The command uses the test suite that corresponds to the current minor release of Red Hat OpenShift, for example, 4.x. If you want to run the tests for a previous minor release, use an image tag to indicate the required version. For example, when running the tests for OpenShift 4.6, add the version number to the above command such as, ose-tests:v4.6. See the [ose-tests repository page](#) for information on tags available.
2. Follow the steps to run the Red Hat OpenShift Virtualization conformance test suite:
   a. Download the conformance tests specific to your environment by using the following command:

   ```
   $ curl -L https://github.com/kubevirt/kubevirt/releases/download/v<KubeVirt version>/conformance.yaml -o kubevirt-conformance.yaml
   ```

   In this command `<KubeVirt version>` corresponds to the OpenShift Virtualization version that you use. For more details, see the Version mapping table.

   b. Execute the tests by using the following command:

   ```
   $ sonobuoy run --skip-preflight --plugin kubevirt-conformance.yaml
   ```

   In this command `<KubeVirt version>` denotes the version of the kubevirt.

   c. Monitor the status of the test by using the following command:

   ```
   $ sonobuoy status
   ```

   d. When the test run is completed, fetch the results by using the following command:

   ```
   $ sonobuoy retrieve
   ```

   It generates a compressed tar file.

   **Verification steps**

   Verify that the tests are completed successfully, by using the following command:

   ```
   $ sonobuoy results <tarball>
   ```

   The output should look similar to this:

   ```
   Plugin: kubevirt-conformance
   Status: passed
   Total: 637
   Passed: 9
   Failed: 0
   Skipped: 628
   ```

3. Follow the steps to run the Red Hat OpenShift Service Mesh test suite:
   a. Clone the Maistra Test tool GitHub repository.

   b. Run the tests as per the instructions provided in the README.md file in the repository. Note that these tests may take approximately 3 hours to complete. If you face any issues that cause the suite to fail quickly, inspect the impacted pods by using the following command to check for ImagePullBackOff errors:

   ```
   $ describe
c. After successful test completion, results.xml and test.log files are generated. Submit the files along with the CNI conformance test results to the Red Hat certification team for verification.

### 32.7. SUBMITTING YOUR CNI OPERATOR FOR CERTIFICATION

Capture the output of the end-to-end CNI tests, the OpenShift Virtualization tests and the Service Mesh tests, if applicable and submit the results through the Red Hat Certification portal.

1. Log in to the Red Hat Certification portal.

2. On the homepage, enter the product case number in the search bar. Select the case number from the list that is displayed.

3. On the Summary tab, under the Files section, click Upload.

Before uploading the results ensure that all CNI tests are completed successfully. If a particular test does not apply to the certified product, include an explanation along with the result submission.

### 32.8. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit the CNI project for validation through the Red Hat certification portal, the Red Hat certification team reviews and verifies your project for certification. After successful verification, your certified product gets published on the Red Hat Ecosystem Catalog.

Follow the steps to publish your certified CNI operator:

1. Access the Partner connect web page. My Work web page displays the Product Listings and Certification Projects.

2. Navigate to the Product Listings tab and search for the required product listing.

3. Click the newly created product listing that you wish to publish. Review all the details of your product listing.

4. From the left pane, navigate to the Certification Projects tab.

5. Click Attach Project to attach your certified CNI operator to this listing. Also add the certified container project used by your CNI project. Both the projects must be in Published status. The Publish button is enabled when you specify all the required information for the product listing along with the attached projects.

6. Click Publish.

Your certified CNI operator is now available for public access on the Red Hat Ecosystem Catalog. The certified CNI operator will also be listed in the OperatorHub within the web console in OpenShift. Partners will receive a logo to promote their certified product on the Red Hat OpenShift platform.
NOTE

The Red Hat OpenShift software certification does not conduct functional or performance testing of the Partner’s product outside the Operator constructs and its impact on the Red Hat platform on which it was installed and executed. All aspects of the certification candidate product’s quality assurance remain the Partner’s sole responsibility.
CHAPTER 33. CSI CERTIFICATION

33.1. WORKING WITH CONTAINER STORAGE INTERFACE (CSI) CERTIFICATION

NOTE

Deploy and manage the CSI driver through an Operator, before proceeding with Red Hat CSI certification. Your operator and all the containers referenced in an Operator Bundle must already be certified and published on the Red Hat Ecosystem Catalog prior to certifying the CSI driver.

33.1.1. Introduction to Container Storage Interface

The Container Storage Interface (CSI) allows products on the OpenShift Container Platform to consume storage from storage back ends that implement the CSI interface as persistent storage. CSI drivers are typically shipped as container images. The CSI badge is a specialization within Red Hat OpenShift certification. It is available to storage products that integrate by using a CSI driver with Red Hat OpenShift as the deployment platform.

The products that meet the requirements and complete the certification workflow can be referred to and promoted as Certified CSI products on the Red Hat OpenShift Container platform. After the certification is approved, the certified CSI product will be listed on the Red Hat Ecosystem Catalog and the OperatorHub within the web console in Red Hat OpenShift. The certified CSI operators are identified with the CSI badge. Partners will receive a logo to promote their product certification.

Additional resources

- For more information about CSI, see:
  - CSI Specification
  - Using CSI
  - Kubernetes CSI Developer Documentation

33.1.2. Certification workflow for CSI

NOTE

Red Hat recommends that you are a Red Hat Certified Engineer or hold equivalent experience before starting the certification process.

The following diagram gives an overview of the certification process:
Figure 33.1. CSI certification workflow

Task Summary

The certification workflow includes three primary steps-

1. Section 32.1.2.1, “Certification on-boarding and creating your first project”
2. Section 4.2, “Certification testing”
3. Section 33.1.2.3, “Publishing the CSI product listing on the Red Hat Ecosystem Catalog”

33.1.2.1. Certification on-boarding and creating your first project

Prerequisites

Verify the functionality of your product on the target Red Hat platform, in addition to the specific certification testing requirements. If running your product on the targeted Red Hat platform results in a substandard experience then you must resolve the issues prior to certification.

The Red Hat Partner Acceleration Desk (PAD) is a Products and Technologies level partner help desk service that allows our (prospective) technology partners a central location to ask non-technical questions about Red Hat offerings, partner programs, product certification, engagement process, and so on.

See PAD - How to open & manage PAD cases, to open a PAD ticket.

Through the Partner Subscriptions program, Red Hat offers free, not-for-resale software subscriptions that you can use to validate your product on the target Red Hat platform. To request access to the program, follow the instructions on the Partner Subscriptions site.

You must construct your container images so that they meet the certification criteria and policy. For more details, see image content requirements and Operator requirements.

Procedure
Follow these high-level steps to certify your CSI operator:

1. Join the Red Hat Partner Connect for Technology Partner Program.

2. Agree to the program terms and conditions.

3. Complete your company profile.

4. Create your certification project by selecting your desired platform, for example, Red Hat OpenShift and then choose Operator Bundle Image.

5. Complete the Pre-certification checklist.

Additional resources

For detailed instructions about creating and configuring your first CSI project, see Creating an Operator bundle project and Configuring the Operator bundle.

33.1.2.2. Certification testing

To run the certification test,

1. Fork the Red Hat upstream repository.

2. Install and run the Red Hat certification pipeline on your test environment.

3. Review the test results and troubleshoot, if any issues.

4. Submit the certification results to Red Hat through a pull request.

33.1.2.3. Publishing the CSI product listing on the Red Hat Ecosystem Catalog

When you complete all the certification checks successfully, you can submit the test results to Red Hat. You can turn on or off this results submission step depending on your individual goals. When the test results are submitted, it triggers the Red Hat infrastructure to automatically merge your pull request and publish your operator.

33.2. CREATING A CSI PROJECT

For detailed instructions about creating your first CSI project, see Creating an Operator bundle project.

33.3. CONFIGURING THE CSI PROJECT

For detailed instructions about configuring your CSI project settings and completing the Pre-certification checklist, see Configuring the Operator bundle.

33.4. WORKING WITH THE OPENSHIFT OPERATOR PIPELINE

Follow the steps to certify your operator before proceeding with the CSI certification:

1. Adding your Operator bundle

2. Forking the repository
33.5. CONFIGURING YOUR TEST ENVIRONMENT FOR RUNNING THE CSI TESTS

Before running the CSI tests, verify if the Red Hat OpenShift environment that you use for running the CSI tests meets the following criteria:

1. Use a Red Hat OpenShift version within the full support phase of the life cycle. Red Hat recommends using the latest supported release. For more information on OpenShift releases, see Red Hat OpenShift Container Platform Life Cycle Policy.

2. Install the CSI driver by using its Operator.

3. Install the Red Hat OpenShift Virtualization.

4. Configure a RHEL host with access to the OpenShift cluster to use as the test client for running the CSI certification tests.

**IMPORTANT**

Run your own product verification tests on the same configuration to ensure that your product functionality works as expected on the Red Hat OpenShift environment.

33.6. ACCESSING THE CSI CERTIFICATION TESTS

The CSI certification tests are packaged in a container and are included in the OpenShift End-to-End repository. To retrieve the current version of the tests, navigate to the Red Hat Ecosystem catalog and follow the instructions available on the Get this image tab.

When accessing the OpenShift End-to-End Tests container, make sure to pull the corresponding tag of the OpenShift version that you are using for your product certification.

33.7. SETTING UP THE CSI TEST PARAMETERS

The CSI certification tests require the following files to be present on the client host:

- A kubeconfig.yaml file with credentials to access the OpenShift cluster under test. This file is automatically created during the OpenShift installation, but you can recreate a copy by using the following command:

  ```
  $ oc config view --raw > kubeconfig.yaml
  ```
A `manifest.yaml` file that describes the capabilities of your driver. This file is used to determine the tests that must be executed. For more information, see the example file.

### 33.8. RUNNING THE CSI TESTS

On the test client, place the `kubeconfig.yaml` and `manifest.yaml` files in the current working directory and run the following command:

```bash
$ podman run -v `pwd`:/data:z --rm -it registry.redhat.io/openshift4/ose-tests sh -c "KUBECONFIG=/data/kubeconfig.yaml TEST_CSI_DRIVER_FILES=/data/manifest.yaml /usr/bin/openshift-tests run openshift/csi --junit-dir /data/results"
```

If you execute the tests on a version of OpenShift previous to the latest release, make sure to add the right tag to the container image name: `registry.redhat.io/openshift4/ose-tests:<tag>`. See OpenShift End-to-End Tests repository page for a list of available tags.

**Verification Steps**

1. The output of the command includes a summary of the tests for the CSI capabilities and for container native virtualization (CNV) in OpenShift. Following is a sample output:

   ```
   Storage Capabilities (guaranteed only on full CSI test suite with 0 fails)
   =============================================================
   Driver short name: ceph-test
   Driver name: test.rbd.csi.ceph.com
   Storage class: ceph-rbd-sc.yaml
   Supported OpenShift / CSI features:
   Persistent volumes: true
   Raw block mode: true
   FSGroup: true
   Executable files on a volume: true
   Volume snapshots: true
   Volume cloning: true
   Use volume from multiple pods on a node: true
   ReadWriteMany access mode: true
   Volume expansion for controller: true
   Volume expansion for node: true
   Volume limits: true
   Volume can run on single node: true
   Topology: true
   Supported CNV features:
   Raw block VM disks: true
   Live migration: true
   VM snapshots: true
   Storage-assisted cloning: true
   
   The detailed results will be placed in the `results` subdirectory.
   ```

2. If you want to see a list of the tests that are run for CSI certification, run the following command:

   ```bash
   podman run -v `pwd`:/data:z --rm -it registry.redhat.io/openshift4/ose-tests sh -c "KUBECONFIG=/data/kubeconfig.yaml TEST_CSI_DRIVER_FILES=/data/manifest.yaml /usr/bin/openshift-tests run --dry-run openshift/csi"
   ```
NOTE

Execute separate tests for each supported storage protocol.

33.9. SUBMITTING CSI TEST RESULTS

Make sure to have the following ready before submitting your test results:

- Contents in the results directory
- manifest.yaml file
- Output of the following commands:
  
  ```
  $ oc get clusterversion -o yaml
  and
  $ podman image list registry.redhat.io/openshift4/ose-tests
  ```

Follow the procedure to submit the result files through the Red Hat Certification portal:

1. Log in to the Red Hat Certification portal.
2. On the homepage, enter the product case number in the search bar. Select the case number from the list that is displayed.
3. On the Summary tab, under the Files section, click Upload.

Before uploading the results ensure that all CSI tests are completed successfully. If a particular test does not apply to the certified product, include an explanation along with the result submission.

33.10. PUBLISHING THE PRODUCT LISTING ON THE RED HAT ECOSYSTEM CATALOG

When you submit the CSI project for validation through the Red Hat certification portal, the Red Hat certification team reviews and verifies your project for certification. After successful verification, your certified product gets published on the Red Hat Ecosystem Catalog.

Follow the steps to publish your certified CSI operator:

1. Access the Partner connect web page. My Work web page displays the Product Listings and Certification Projects.
2. Navigate to the Product Listings tab and search for the required product listing.
3. Click the newly created product listing that you wish to publish. Review all the details of your product listing.
4. From the left pane, navigate to the Certification Projects tab.
5. Click Attach Project to attach your certified CSI operator to this listing. Also add the certified container project used by your CSI project. Both the projects must be in Published status. The Publish button is enabled when you specify all the required information for the product listing along with the attached projects.
6. Click **Publish**.

Your certified CSI operator is now available for public access on the Red Hat Ecosystem Catalog. The certified CSI operator will also be listed in the OperatorHub within the web console in OpenShift. Partners will receive a logo to promote their certified product on the Red Hat OpenShift platform.

**NOTE**

The Red Hat OpenShift software certification does not conduct functional or performance testing of the Partner’s product outside the Operator constructs and its impact on the Red Hat platform on which it was installed and executed. All aspects of the certification candidate product’s quality assurance remain the Partner’s sole responsibility.